POLITECNICO DI MILANO

Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria dell'Automazione

Control and optimization of the start-up of a combined cycle power plant

Relatore:
Prof. Riccardo Scattolini
Correlatori:
Ing. Damien Faille
Ing. Frans Davelaar

Candidato:
Fabio Righetti, n°724560

Anno Accademico 2009-2010

Io non ho talenti straordinari.
Sono solo appassionatamente curioso
A. Einstein

Contents

Ringraziamenti - Acknowledgements

Un sentito e doveroso ringraziamento va al Prof. Riccardo Scattolini, che mi ha dato, anzi regalato, la possibilità di svolgere questa Tesi, una esperienza che non avrei mai creduto possibile, oltre ad aver apportato il suo contributo dove e quando era necessario, sempre con un clima di apertura e disponibilità che non è facile trovare. Voglio inoltre ringraziare il Prof. Francesco Casella, per i chiarimenti e le delucidazioni su Dymola e sul tipo di processo in esame. Ringrazio inoltre l'azienda Electricité de France (EDF) per la disponibilità e l'accoglienza.

I would to thank Ing. Damien Faille for the opportunity to do this thesis in EDF - Chatou. His contribute in this thesis was very important, even his agreeable company during Paris period. I want to thank also Ing. Frans Davelaar for his attention and disponibility in every moment to solve my problem. I couldn't forget Adrian Tica for his total availability and his huge contribute in this thesis to realize all the simulations I needed.
In my Paris experience it's impossible to forget Adrian's company: without him my presence in France wouldn't be the same thing. Especially I want to thank him for KFC experience and for the time passed together!
I want also to thank other people in EDF: the others stagiaers (Fanny with her chair, Paula, PierAlban, David), Marité and Serge Moran, all the people of STEP group. Last but not the list, the group of FDM, that give me a little injection of football in France.

Dopo i ringraziamenti ufficiali, che accompagnano ogni tesi (perchè così è giusto, senza il contributo dei Prof. non se ne viene mai a campo in fondo) ci sono sempre quelli più intensi, più emozionanti, quelli personali...
Il modo giusto per iniziarli? Credo questo...

Una vita da mediano
a recuperar palloni
nato senza i piedi buoni
lavorare sui polmoni

una vita da mediano
con dei compiti precisi
a coprire certe zone
a giocare generosi, lì,sempre lì

Si, questi anni sono stati una grande partita, senza magari piedi buoni, ma lavorando sulla corsa, sui polmoni, sul fare, sulla voglia di vincere (con un

pochino di doping dai... chiamiamoli così). Sudore, fatica, difficoltà, ma anche tanta gioia, tante persone incontrate, conosciute, vissute.
Una grande partita... da commentare e ricordare, magari così...

Amici ascoltatori, ecco la formazione ufficiale: Federico, Ciro, Gabry, Maji, Giorgio. Essì perchè sono stati loro la squadra che è scesa sempre in campo e che mi ha accompagnato in questa grande sfida.
Federico, che da buon portiere parava le mie menate universitarie anche non capendoci un tubo, sempre pronto a chiamarmi anche quando tra mille esami, scritti, orali e tesi facevo fatico a sentirlo. Piuttosto che perdermi è arrivato a prendermi a pugni in più e più lezioni.
Ciro che anche se viene e va, alla fine sempre è qua. Quando qualcosa non quadra, quando bisogna lamentarsi, arrabbiarsi, ma soprattutto divertirsi, sdare... beh qualsiasi cosa ci sia da fare si sa sempre chi chiamare. Certo le polemiche non sono mai mancate...d'altronde se in campo qualcuno ti tocca o mi tocca...beh povero lui...tanto basta uno sguardo!
Gabry, che fa un primo tempo da centravanti, grazie ad appunti e lezioni private che nemmeno Pico de Paperis... per poi diventare, causa distanze di studi e tempo, un mediano nella mia vita, sempre lì a mediare, condividere, consigliare e correggere le mie (nostre) peggiori cose.I guizzi del fuoriclasse non sono persi peró: quando escono... AHHH... cosí forte!
Maji, te ne dispiacerà, ma il ruolo di mister è occupato e quindi giochi, niente direzione tecnica, nonostante i tanti momenti passati insieme a parlare di calcio (di cui non hai il mio tasso tecnico!), di cose scena e di quella cosa che finisce per no.
Giorgio, il trequartista, che tanto non torna, ma quando gli dai il pallone e stai con lui...le cose le senti diverse... sono più sghidde, più scena.
Un saluto particolare alla direzione tecnica, chi sui campi come Gandalf, sempre pronto ad aiutare, suggerire, sopportare, e chi da studio, come Simone, amico insostituibile in questi anni. E poi la partita di questi anni è iniziata...:

1' Entrano subito nella scena di gioco nuovi acquisti, nomi da calciomercato folle e stellare, come Coda, Mitch, Dea, Crespi, Ricky, Robertz e Isacco. Persone che usciranno dal campo a fine primo tempo, ma che rimarranno sempre nella vita ad accompagnarmi, pronti ad aiutare, a divertirsi e stare insieme, per il puro piacere di rivivere quel fantastico primo tempo.

15' L'inizio è dei migliori, dominio a centrocampo, solo un esame viene lasciato indietro il primo anno...i tanti tifosi che mi accompagnano e che mi accompagneranno negli anni sono già lì, seguono la squadra e ci si diverte, consigliano, incitano, ma soprattutto stanno con me. Grazie

a Giulia, Giorgione, Lorenzo, Civi, Alice, Cecilia e tutti gli altri che, anche se non nominati, non sono meno importanti. E poi c'è la curva, con i capi ultras Federico, Marco, Silvia e Ludovica: grazie ragazzi del '94, siete stati fonte di gioia in questa partita come poche altre cose.

30' La partita si fa dura... i meccanismi tattici non marciano a dovere..qualche esame si inceppa...speriamo bene, bisogna metterci il cuore verso la fine...intanto si pensa ai duri allenamenti sempre fatti... grazie a chi con me li ha condivisi, come Pave, Fafo, Tommy, Fra e tutti gli altri che da anni ormai seguono Ste per andare a far quel gran casino che ci contraddistingue.

45' Il primo tempo è finito, si è sbloccato il risultato nei minuti di recupero... tre anni sono andati, i più belli ed intensi che abbia mai potuto avere. 1-0, siamo in vantaggio, l'arbitro manda tutti a prendere un thé caldo (ma per me come sempre coca fredda!)

Nell'intervallo le scelte tecniche sono davvero difficili...si sente il bisogno di cambiare modulo...si sceglie quindi il PoliMi al posto del PoliTo...quanti pensieri, quante angosce nel farlo, l'assistente spirituale ha però lavorato bene: grazie Gimmy!

46' Inizia il secondo tempo, il cambio tattico, anche se difficile da capire a volte, produce i suoi frutti, anche grazie ai nuovi innesti, che regalano momenti divertenti e situazioni sempre nuove. Il bel gioco espresso e l'intesa con GiaGia, Stefano, Marco, Cepi, Meri (con la I!) e Valerio vale la citazione. Di particolare interesse è Petro (detto da alcuni Francesco..) giocatore scoperto a fondo in un vivaio parigino!

80' la partita perde un po' di tono, si fatica, ma ora c'è più consapevolezza (osiamolo dire.. più maturità). Il merito va anche a chi in tanti anni ha consigliato e spronato come una sorella maggiore, grazie a Federica, anche e soprattutto per le chiaccherate e le meritate prediche!

90' ragazzi non ci si crede... partita finita, ormai 2-0, è finita, andiamo a prenderci la coppa!

Che partita...ma senza la società, senza la mia famiglia come sarebbe stato possibile? Grazie mamma e papà, per l'appoggio e per averci creduto fino in fondo, e alla squadra primavera, Elena, dico di non mollare, ce l'ho fatta io, ma anche perchè c'eri tu. Non dimentichiamo, per la gioia di molti, la mascotte Peggy!
Un grazie particolare ai nonni/e, a coloro che di partite nella vita ne hanno

fatte molte e molto mi hanno quindi insegnato. Nel mio cuore lo spazio per voi ci sarà sempre, grazie Nonno Angelo, Nonno Ermanno, Nonna Gina e Nonna Tina, questa tesi è per voi.
Rimane il direttore d'orchestra, l'artefice di questa vittoria, il mio mister: Eleonora. Senza di te non ci sarebbe stata partita, sempre pronta a motivare e spronarmi, a consolarmi e a guidarmi o semplicemente, come sai fare tu, a stare con me. Grandi frasi romantiche non ti scrivo mai, grandi regalo non ne faccio mai, ma questo è il momento in cui più ti sento vicina e dentro alla mia vita, pronti a iniziare nuove e importanti partite sempre insieme! Abbiamo passato momenti fantastici, unici in questi anni, con follia e tanto amore. Ora chissà, nelle partite che mi attendono...di sicuro io il mio mister non lo esonero, lo confermo con un grande bacio, magari il mister di una vita intera... (ma mi deve far sempre vincere...ovvio!)
Un grazie particolare a chi per primo è sceso in campo, al capitano, cioè a me stesso. Alla fine ce l'ho fatta...direi che replico come alla triennale...GG Ingegnere, e GL e HF per il futuro.

Introduction

The main goal of this work is to develop and tune a simplified control-oriented model of a Combined Cycle Power Plant (CCPP), which is subsequently used to develop a start-up procedure minimizing the time required to reach full load conditions by limiting the thermo-dynamical stresses of the main process components. An already available and detailed plant model developed in the Dymola software environment is used as the reference system. However this model is not suitable for control purposes, due to the lack of efficient optimization tools in Dymola. For this reason, a simplified model, obtained as the combination of local linear estimated models, is developed in the Matlab/Simulink environment. In order to define the optimal start-up procedure, a minimum time optimal control problem is then defined and solved with reference to the simplified model. Finally, the results achieved are validated on the Dymola reference model.

The first part of the Thesis describes the Dymola model developed by SUPELEC Rennes, [15].
The second part presents the procedure adopted to obtain an estimated model of the plant in the Matlab/Simulink environment. It is then shown how to combine a number of local linear models so as to build a nonlinear model reliable in all the plant operating range.
The third part is devoted to describe the minimum time optimal control problem which allows one to compute the optimal load profile to be followed during the start-up phase of the plant to minimize the time required to reach full load operating conditions and to maintain the thermal stresses of the components within prescribed limits.

Finally, in the fourth part of the Thesis, the results obtained with the simplified estimated model are validated on the initial Dymola simulator.

Introduzione

In questo lavoro si presenta un metodo innovativo per la modellizzazione e l'ottimizzazione delle procedure di avviamento di un impianto a ciclo combinato, o CCPP (Combined Cycle Power Plant), per la produzione dellâenergia elettrica. Negli ultimi due decenni, gli impianti CCPP hanno avuto una grande diffusione in quanto molto più efficienti, e quindi economicamente più convenienti, degli impianti termoelettrici tradizionali, rispetto ai quali garantiscono anche emissioni decisamente ridotte a parità di energia prodotta. Tuttavia, tali caratteristiche positive dei sistemi CCPP sono conseguibili se gli impianti sono normalmente eserciti a pieno carico, o comunque a carichi elevati. Inoltre, la vita di un CCPP è strettamente legata agli stress termici e meccanici dei suoi componenti, stress che assumono i valori massimi durante le fasi di avviamento (start-up) e di spegnimento (shut-down). Per queste ragioni, è di particolare interesse lo studio delle procedure di avviamento, che garantiscano il raggiungimento delle condizioni di pieno carico nel tempo più breve possibile, mantenendo limitato il valore degli stress.
Nella prima parte della Tesi si descrivono i principali componenti degli impianti CCPP, e i diversi fenomeni che governano il loro funzionamento. Sono anche riportati i modelli matematici impiegati per lo sviluppo di simulatori dinamici adatti al progetto dei sistemi di controllo. Uno di questi simulatori, sviluppato in ambiente Dymola da Supelec (si veda [15]) è stato utilizzato in questo lavoro come impianto di riferimento. Il simulatore si basa su un modello semplificato del sistema, in cui si considera un solo livello di pressione allâinterno del generatore di vapore. Per la descrizione di un modello più dettagliato, a partire dal quale è stato dedotto il simulatore qui impiegato, si rimanda a [9]
Lâambiente Dymola è estremamente potente e consente lo sviluppo di simulatori molto dettagliati. Tuttavia, esso non è ancora dotato di tutti quegli strumenti software necessari per il progetto di un sistema di controllo e/o per l´ottimizzazione delle condizioni di funzionamento del sistema. Per questa ragione, nella seconda parte della Tesi, è stato ricavato un modello semplificato, identificato a partire dai dati forniti dal simulatore Dymola. Tale modello identificato è stato ottenuto per interpolazione di modelli lineari, identificati nelle diverse condizioni di funzionamento del sistema specificate dal valore assunto dal carico della turbina a gas (GT_Load).
La stima dei modelli lineari è stata effettuata con gli strumenti disponibili in ambiente Matlab, mentre l´interpolazione si basa sull´impiego di $membership\ functions$ che forniscono il grado di attendibilità di ogni modello lineare in funzione del GT_Load. Il modello complessivo è stato quindi implementato in ambiente Matlab/Simulink. Le sue caratteristiche sono una

relativa semplicità e la possibilità di effettuare simulazioni, anche di grandi transitori, in tempi sufficientemente ridotti.
Il modello identificato è stato quindi validato, rispetto al modello di partenza in Dymola, a fronte di variazioni di piccola e grande identità del carico (GT_Load). Tali esperimenti di validazione hanno permesso di verificare che con il modello identificato ottenuto per interpolazione è possibile seguire con ottima approssimazione lâandamento delle principali variabili di impianto durante i transitori di avviamento.
L´ultima parte della Tesi ha riguardato la definizione del profilo ottimo di carico (GT_Load) durante una procedura di avviamento. In particolare, con riferimento al modello stimato, il profilo del GT_Load è stato ottenuto risolvendo un problema di ottimizzazione in cui viene minimizzato il tempo di raggiungimento delle condizioni di pieno carico nel rispetto di vincoli sui valori massimi assunti dagli stress e di eventuali altri vincoli relativi a variabili di impianto. Per formulare correttamente il problema, si è assunto che il carico sia descritto da una opportuna funzione (funzione di Hill), parametrica rispetto a due variabili che rappresentano le incognite, insieme al tempo finale dello start-up, del problema di minimizzazione. Il profilo ottimo di carico è stato infine applicato al simulatore di partenza in Dymola. È stato così verificato che quasi tutte le variabili di impianto seguono effettivamente l´andamento previsto durante la fase di ottimizzazione, rispettando i vincoli imposti. Nell´unico caso in cui questo non avviene, per uno degli stress considerati, la discrepanza si può imputare a un modello identificato non sufficientemente preciso. Tuttavia, la procedura generale è ormai ben delineata ed eventuali approfondimenti futuri potranno portare a un raffinamento delle soluzioni trovate, già ora più che soddisfacenti.

1 Chapter 1: System Description and dynamic model

In this chapter the Combined Cycle Power Plant (CCPP) object of this work is first introduced. A dynamical model describing the main physical phenomena underlying its behavior is presented and its implementation in the Modelica simulation environment is discussed. Then, the main simplifications adopted to obtain a model suitable for simulation and control design are described.

1.1 Physical system

Combined Cycle Power Plants are systems designed to produce energy from thermal power with improved efficiency and reduced polluting emissions with respect to traditional thermal power plants. The main elements of a CCPP are the Gas Turbine (GT) system, the Heat Recovery Steam Generator (HRSG) and the Steam Turbine (ST). The dynamical models, taken from [4], of the main components of a CCPP are now briefly described.

1.1.1 The Gas Turbine system

The three-dimensional section of an industrial GT system is shown in Figure 1, while a schematic description is given in Figure 2. The main components of GT systems are the compressor, the combustion chamber (or burner) and the gas turbine. Figure 2 also includes the Inlet Guide Valve (IGV), the fuel valve, a mixer, located after the burner and associated to a bypass valve, the air cooler and the exhaust manifold (mixer), positioned after the gas turbine. From the figure, it is also possible to observe the air fluxes drained from different stages of the gas turbine: they play the role of cooling flow rates of the turbine blades. Although GTs are complex systems, for the purposes of this Thesis, their modeling can be simplified as shown in Figure 3, where the main elements considered in the following are shown. Moreover, since the GTs are characterized by a faster dynamics with respect to the one of the HRSG systems, their model can be based on algebraic equations. In fact, their significant dynamic phenomena are mainly due to the system actuators: the IGV and the fuel feeding system. The IGV, regulating the inlet air flux, is characterized by a quite slow dynamics, with a time constant of about ten seconds. The fuel valve, instead, exhibits a faster response, with a settling time even lower than one second. Both these two devices may be modeled as first order systems, while the time delay of the masses flow through the GT

may be neglected.

The Compressor

The compressor (see Figure 4) provides a high pressure air flow to the combustion chamber: the corresponding thermodynamic transformation is shown in Figure 5. With the same model it is possible to understand immediately the meaning of the compressor efficiency η_c, defined as:

$$\eta_c = \frac{h_{a_{ISO}} - h_{a_{IN}}}{h_{a_{OUT}} - h_{a_{IN}}} \tag{1}$$

where he definition of all the variables is reported in Table 1.

Input variables		
q_a	air flow rate	kg s^{-1}
q_{a_0}	air flow rate at nominal road	kg s^{-1}
$T_{a_{IN}}$	inlet air temperature	K
Output variables		
P_c	absorbed power	W
$T_{a_{OUT}}$	outlet air temperature	K
Auxiliary variables		
$h_{a_{IN}}$	inlet air enthalpy	J kg^{-1}
$h_{a_{OUT}}$	outlet air enthalphy	J kg^{-1}
$h_{a_{ISO}}$	air isentropic enthalphy	J kg^{-1}
$P_{a_{OUT}}$	outlet air pressure	Pa
$P_{a_{IN}}$	inlet air pressure	Pa
Parameters		
c_{p_c}	air equivalent specific heat	J K^{-1} kg^{-1}
η_c	efficiency	

Table 1: Compressor - variables and parameters.

To obtain a mathematical representation of the system, three general physical laws are here recalled:

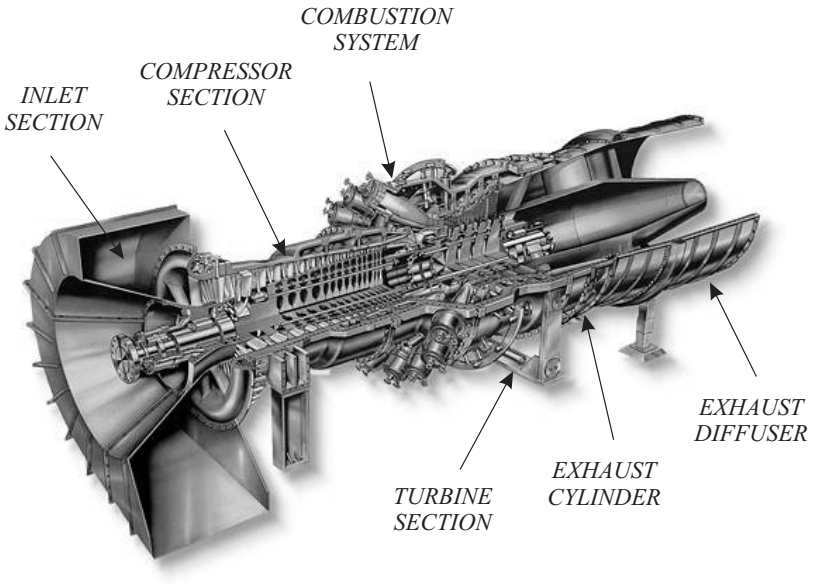


Figure 1: 3-D representation of a Gas Turbine System.

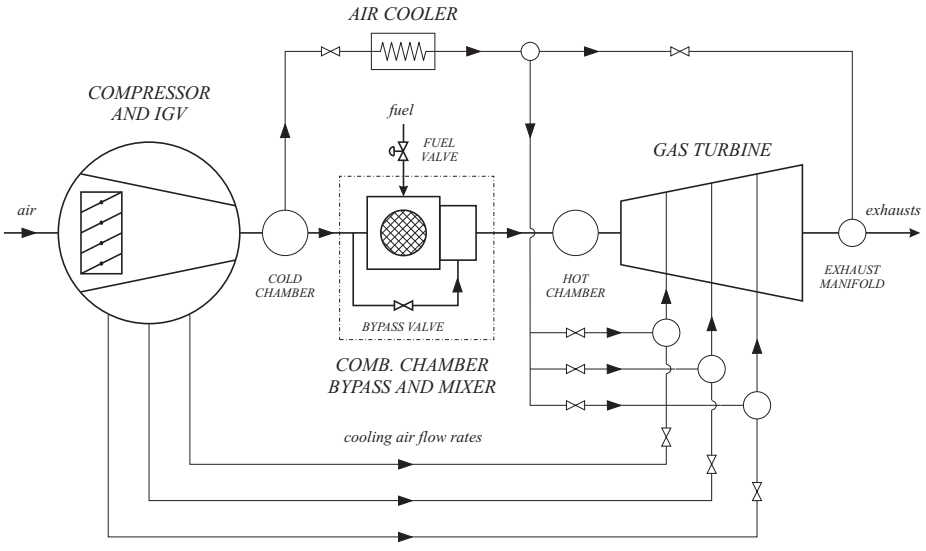


Figure 2: Detailed representation of a Gas Turbine system.

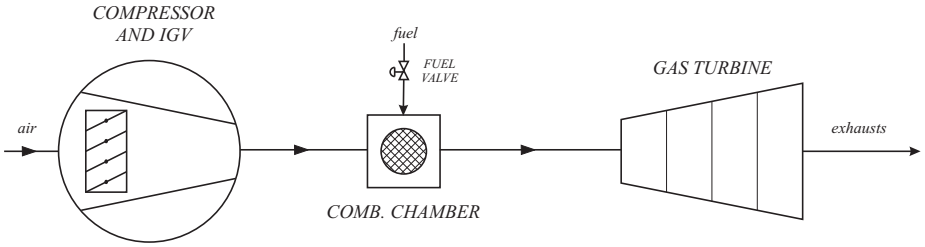


Figure 3: Rough modeling scheme of a Gas Turbine system.

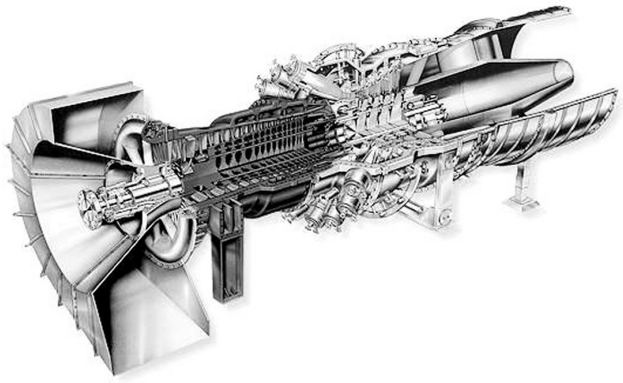


Figure 4: 3-D representation of a compressor.

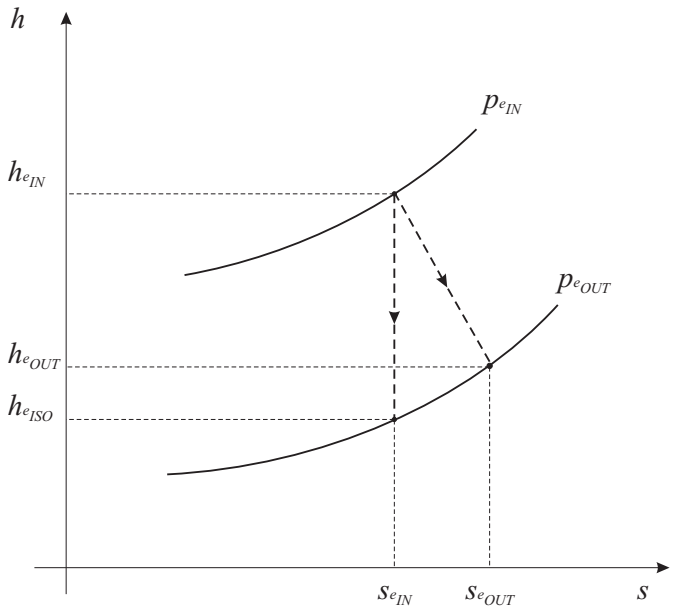


Figure 5: Thermodynamic transformation of a compressor system

1. the definition of isentropic transformation

$$Tds = dh - \frac{1}{\rho}dp = 0 \tag{2}$$

where T is the fluid temperature (K), s the specific entropy ($\mathrm{JK^{-1}kg^{-1}}$), h is the specific enthalphy ($\mathrm{Jkg^{-1}}$), ρ is the mass density (kg $\mathrm{m^{-3}}$) and p is the pressure (Pa).

2. the definition of specific heat c_p at constant pressure

$$c_p = \frac{dh}{dT} \tag{3}$$

3. the ideal gas law

$$\frac{p}{\rho} = \alpha T \tag{4}$$

where α is the thermodynamic constant

By proper substitutions, from equations (2), (3), and (4) it is possible to obtain the following equality:

$$c_p dT = \frac{\alpha T}{p}dp \tag{5}$$

which can be integrated, thus obtaining, for the compressor, the following relation between temperatures and pressures (related to the isentropic transformation case):

$$\frac{T_{\alpha ISO}}{T_{\alpha_{IN}}} = \left(\frac{p_{\alpha_{OUT}}}{p_{\alpha_{IN}}}\right)^{\frac{\alpha}{c_p}} \tag{6}$$

which can be assumed to be analogously valid for the enthalpies:

$$\frac{h_{\alpha_{ISO}}}{h_{\alpha_{IN}}} = \left(\frac{p_{\alpha_{OUT}}}{p_{\alpha_{IN}}}\right)^{\frac{\alpha}{c_p}} \tag{7}$$

By combining equation (7) with the compressor efficiency definition, it is possible to obtain the expression of the outlet temperature and of the entalphy (a simplification is used, the pressure ratio $PR = \frac{p_{\alpha_{OUT}}}{p_{\alpha_{IN}}}$ is replaced with the expression $PR = PR_0\frac{q_\alpha}{q_{\alpha_0}}$, where PR_0 is defined as the nominal pressure ratio, and its value coincides with PR at nominal load). The expressions are:

$$h_{\alpha_{OUT}} = h_{\alpha_{IN}}\left(1 + \frac{1}{\eta_c}\left[\left(PR_0\frac{q_\alpha}{q_{\alpha 0}}\right)^{e_c} - 1\right]\right) \tag{8}$$

$$T_{\alpha_{OUT}} = T_{\alpha_{IN}} \left(1 + \frac{1}{\eta_c}\left[\left(PR_0 \frac{q_\alpha}{q_{\alpha_0}}\right)^{e_c} - 1\right]\right) \tag{9}$$

where $e_c = \frac{\alpha}{c_p}$.
With further simplifications and substitutions, it is possible to obtain the value of the mechanical power absorbed by the compressor:

$$P_c = q_\alpha \frac{1}{\eta_c}\left[\left(PR_0 \frac{q_\alpha}{q_{\alpha_0}}\right)^{e_c} - 1\right] c_{p_c} T_{\alpha_{IN}} \tag{10}$$

The Gas Turbine

The Gas Turbine (Figure 6) model relies on the same physical principles and simplification already introduced for the compressor. Figure 7 shows the thermodynamic cycle of the turbine. Differently from the compressor case, where the mechanical power is needed to raise the air pressure, here the mechanical power is obtained by the exhaust pressure drop. Table 2 reports the list of all the model variables and parameters.

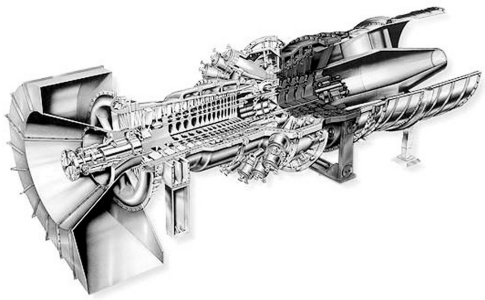


Figure 6: 3-D representation of Gas Turbine

The gas turbine efficiency is defined as

$$\eta_t = \frac{h_{e_{IN}} - h_{e_{OUT}}}{h_{e_{IN}} - h_{e_{ISO}}}$$

and, analogously to (6) and (7), one has:

$$\frac{T_{e_{ISO}}}{T_{e_{IN}}} = \left(\frac{p_{e_{OUT}}}{p_{e_{IN}}}\right)^{\frac{\alpha}{c_p}} \tag{11}$$

Input variables		
q_e	exhaust flow rate	kg s^{-1}
q_{e_0}	exhaust flow rate at nominal road	kg s^{-1}
$T_{e_{IN}}$	inlet exhaust temperature	K
Output variables		
P_t	generated mechanical power	W
$T_{e_{OUT}}$	outlet exhaust temperature	K
Auxiliary variables		
$h_{e_{IN}}$	inlet exhaust enthalpy	J kg^{-1}
$h_{e_{OUT}}$	outlet exhaust enthalphy	J kg^{-1}
$h_{e_{ISO}}$	exhaust isentropic enthalphy	J kg^{-1}
$P_{e_{OUT}}$	outlet exhaust pressure	Pa
$P_{e_{IN}}$	inlet exhaust pressure	Pa
Parameters		
c_{p_t}	exhaust equivalent specific heat	J K^{-1} kg^{-1}
η_t	efficiency	

Table 2: Gas Turbine - variables and parameters.

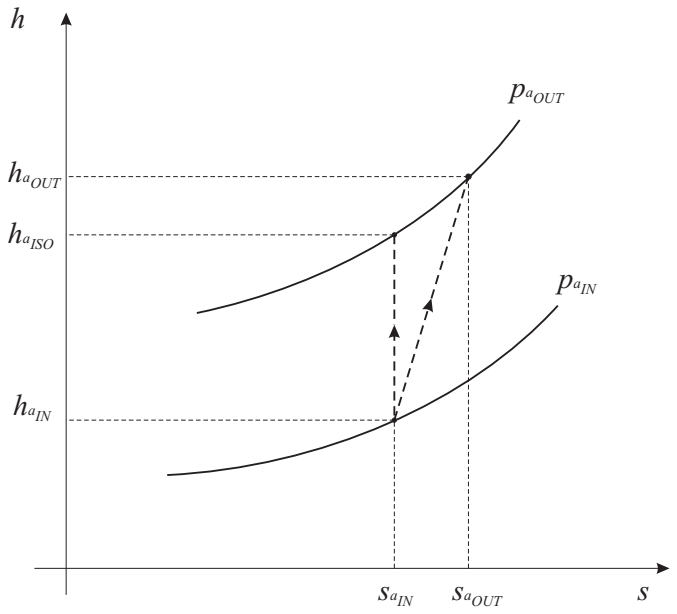


Figure 7: Thermodynamic transformation of a turbine system.

where $T_{e_{IN}} = T_{e_{MIX}}$, and

$$\frac{h_{e_{ISO}}}{h_{e_{IN}}} = \left(\frac{p_{e_{OUT}}}{p_{\alpha_{IN}}}\right)^{\frac{\alpha}{c_p}} \tag{12}$$

Differently from the compressor case, the value of the exponent $e_t = \frac{\alpha}{c_p}$ can be identified according to the normal working conditions. Letting

$$\frac{p_{e_{OUT}}}{p_{e_{IN}}} = \frac{p_{\alpha_{OUT}}}{p_{\alpha_{IN}}} = \frac{1}{PR} \tag{13}$$

and assuming that

$$\frac{p_{e_{OUT}}}{p_{e_{IN}}} = \frac{1}{PR_0}\frac{q_{e_0}}{q_e} \tag{14}$$

it is possible to express the exhaust enthalphy and temperature at the turbine outlet as

$$h_{e_{OUT}} = h_{e_{IN}}\left(1 - \eta_t\left[1 - \left(\frac{1}{PR_0}\frac{q_{e_0}}{q_e}\right)^{e_t}\right]\right) \tag{15}$$

$$T_{e_{OUT}} = T_{e_{IN}}\left(1 - \eta_t\left[1 - \left(\frac{1}{PR_0}\frac{q_{e_0}}{q_e}\right)^{e_t}\right]\right) \tag{16}$$

The expression of the power generated by the turbine is

$$P_t = q_e \eta_t \left[1 - \left(\frac{1}{PR_0} \frac{q_{e_0}}{q_e} \right)^{e_t} \right] c_{p_t} T_{e_{IN}} \tag{17}$$

Combustion chamber, bypass and mixer

A very simplified formulation has been adopted both for the burner and the subsequent air-exhausts mixer. The burner is described by the following two equations

$$T_{e_{cc}} = T_{a_{cc}} + k_f \frac{q_f}{q_{a_{cc}}} \tag{18}$$

$$q_{e_{cc}} = q_{a_{cc}} + q_f \tag{19}$$

(see Tables 3 and 4 for a complete variables list), where $q_{a_{cc}} = q_a - q_{a_{A.C.}} - q_{a_{T.C.}}$, while the mixer is modeled as

$$T_{e_{MIX}} = \frac{q_{a_{BYPASS}} T_{a_{BYPASS}} + q_{e_{cc}} T_{e_{cc}}}{q_{a_{BYPASS}} + q_{e_{cc}}} \tag{20}$$

$$q_{e_{MIX}} = q_{e_{cc}} + q_{a_{BYPASS}} \tag{21}$$

where $T_{a_{BYPASS}} = T_{a_{cc}} = T_{a_{OUT}}$. As previously mentioned, the bypass air flow rate value is assumed to vary proportionally with the compressor inlet air flow rate:

$$q_{a_{BYPASS}} = k_{a_{BYPASS}} q_a$$

1.1.2 The Heat Recovery Steam Generator

The HRSG represents the more complex subsystem of a CCPP. Its modular nature suggests to perform an independent model of the elementary (dynamic) blocks composing the boiler (heat exchangers, evaporators and steam turbine stages), in order to subsequently obtain a complete model (and simulator) of the steam generator by assembling them according to the plant topology. An algebraic solution has been adopted for both the steam fluid dynamics and the water hydrodynamics (except for the regulating actuators, valves and pumps). This choice takes also advantage of the separate modeling of the thermodynamic and fluid dynamic phenomena: the link between

Input variables		
$q_{a_{cc}}$	inlet air flow rate	kg s^{-1}
q_f	air flow rate at nominal road	kg s^{-1}
$T_{a_{cc}}$	inlet air temperature	K
Output variables		
$q_{e_{cc}}$	outlet exhaust flow rate	kg s^{-1}
$T_{a_{OUT}}$	outlet exhaust temperature	K
Parameters		
k_f	fuel equivalent burning power	K

Table 3: Combustion chamber - variables and parameters.

Input variables		
$q_{a_{BYPASS}}$	bypass air flow rate	kg s^{-1}
$q_{e_{cc}}$	comb. chamber outlet exhaust flow rate	kg s^{-1}
$T_{a_{BYPASS}}$	inlet air temperature	K
$q_{e_{cc}}$	comb. chamber outlet exhaust temperature	K
Output variables		
$q_{e_{MIX}}$	outlet exhaust flow rate	kg s^{-1}
$T_{e_{OUT}}$	outlet exhaust temperature	K
Parameters		
$k_{a_{BYPASS}}$	bypass air flow rate ratio	K

Table 4: Mixer and bypass - variables and parameters.

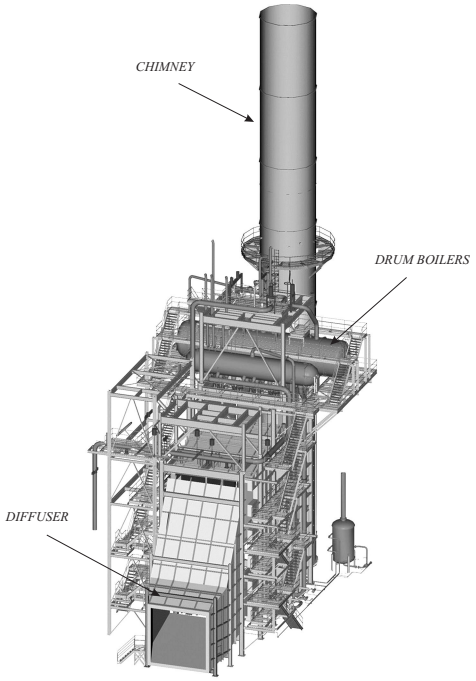


Figure 8: The Heat Recovery Steam Generator

them is represented by the drum boilers pressure. A three-dimensional representation of a HRSG is proposed in Figure 8.

Heat exchangers

A schematic representation of a heat exchanger is usually adopted: a rectilinear metal pipeline with an outer exhausts side and an inner water (in the economizers case) or steam (for superheaters and reheaters). Through the metal, the heat exchange between the two fluxes occurs: commonly, the countercurrent solution (i.e. with the two fluxes flowing in opposite directions) is preferred, by virtue of its greater efficiency, while the co-current configuration may be adopted, for example, when it is necessary to protect the metal from too high thermal stresses.
In the plant considered in this Thesis, a countercurrent solution has been used. The following analysis, describing the adopted simplified heat exchanger model, refers to the exhausts-to-steam countercurrent case: with minor modifications the same modeling approach can be used also for the economizers.
Countercurrent exhaust-to-steam heat exchanger
As regards the fluid dynamics, the pressure drop through the exchanger has

not been taken into account: a unique value of the fluid pressure, corresponding to the inlet section and provided by the solution of the fluid dynamic system (discussed in Section 3.9), has been considered and counted among the model input variables.
The main simplifying assumptions adopted in the thermodynamic model are subsequently summarized:

- A lumped parameters modeling approach has been adopted, see [7], as shown in Figure 9. The ideally rectilinear pipeline has been divided into N equally sized cells, each one characterized by univocal (mean) values of the thermodynamic quantities. The model reliability is guaranteed by the proper selection of the number of cells N.

- As already pointed out, the exhausts side has been algebraically solved, i.e. the heat accumulation phenomena, on this side, have been neglected.

- The *infinite gamma* assumption has been adopted. According to it, the metal temperature of each cell is supposed to be equal to the corresponding steam (mean) one. Consequently, a convenient order reduction of the model is obtained by excluding the metal from the final model formulation, but virtually including its heat accumulation dynamics into the steam one, see [7].

- The heat transfer between steam (metal) and exhausts in each cell has been computed by regarding the outgoing steam temperature and the ingoing exhausts temperature. Moreover, for further simplicity, the steam mean temperature inside each cell has been assumed to be equal to the outlet one.

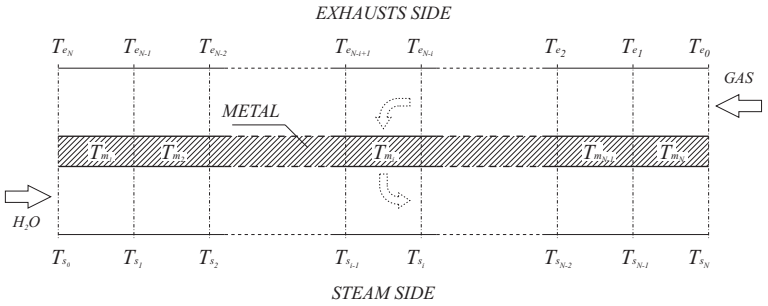


Figure 9: Scheme of a countercurrent exhaust-to-steam heat exchanger.

Steam-metal side
The i - *th* cell of the exchanger is considered: A detailed list of variables

and parameters is reported in Tables 5 and 6. The heat accumulation inside steam and metal is described by the following energy balances, respectively

$$\begin{cases} c_{p_s} m_{s_i} \frac{d\overline{T}_{s_i}}{dt} = q_s c_{p_s}(T_{s_{i-1}} - T_{s_i}) + \gamma_I S_{I_i}(T_{m_i} - \overline{T}_{s_i}) \\ c_m m_{m_i} \frac{dT_{m_i}}{dt} = \gamma_O S_{O_i}(T_{e_{N-1}} - T_{m_i}) - \gamma_I S_{I_i}(T_{m_i} - \overline{T}_{s_i}) \end{cases} \tag{22}$$

for $i = 1, \ldots, N$. By dividing both the equations in (22) by $q_s c_{p_s}$ and defining the constants:

$$\tau_{s_i} = \frac{c_{p_s} m_{s_i}}{q_s c_{p_s}} = \frac{m_{s_i}}{q_s} \tag{23}$$

and

$$\tau_{m_i} = \frac{c_m m_{m_i}}{q_s c_{p_s}} \tag{24}$$

which represent the two time constants associated to the steam and metal heat storing, respectively (note that $\tau_{m_i} >> \tau_{s_i}$), equations (22) can be rewritten as follows

$$\begin{cases} \tau_{s_i} \frac{d\overline{T}_{s_i}}{dt} = (T_{s_{i-1}} - T_{s_i}) + \frac{\gamma_I S_{I_i}}{q_s c_{p_s}}(T_{m_i} - \overline{T}_{s_i}) \\ \tau_{m_i} \frac{dT_{m_i}}{dt} = \frac{\gamma_O S_{O_i}}{q_s c_{p_s}}(T_{e_{N-1}} - T_{m_i}) - \frac{\gamma_I S_{I_i}}{q_s c_{p_s}}(T_{m_i} - \overline{T}_{s_i}) \end{cases} \tag{25}$$

The *infinite gamma* assumption is now introduced: the heat exchange coefficient γ_I between metal and steam is assumed to be virtually infinite. As a consequence, the metal and the steam mean temperatures inside a cell turn out to be identical, as it is easy to prove by rearranging the first equation in (22) and by dividing both its sides by $\gamma_I S_{I_i}$

$$\begin{aligned} & T_{m_i} - \overline{T}_{s_i} = \frac{1}{\gamma_I S_{I_i}} \left(c_{p_s} m_{s_i} \frac{d\overline{T}_{s_i}}{dt} - q_s c_{p_s}(T_{s_{i-1}} - T_{s_i}) \right) \\ & \lim_{\gamma_I \to \infty} T_{m_i} - \overline{T}_{s_i} = 0 \Rightarrow T_{m_i} = \overline{T}_{s_i} \end{aligned} \tag{26}$$

Once stated that $T_{m_i} = \overline{T}_{s_i}$, it is possible to describe the cell thermodynamics by means of a single equation, obtained by adding equations (25) together

$$(\tau_{s_i} + \tau_{m_i}) \frac{d\overline{T}_{s_i}}{dt} = (T_{s_{i-1}} - T_{s_i}) + \frac{\gamma_O S_{O_i}}{q_s c_{p_s}}(T_{e_{N-1}} - \overline{T}_{s_i}) \tag{27}$$

Exhaust side

For the exhausts side, it is assumed that no heat accumulation occurs (no energy variation characterizes the exhausts inside the cell volume), this allows for a simple algebraic modeling of the gases energy balance, i.e.

Input variables		
$p_{s_{IN}}$	inlet steam pressure	Pa
q_e	exhaust flow rate	kg s^{-1}
q_s	steam flow rate	kg s^{-1}
$T_{e_{IN}} = T_{e_0}$	inlet exhaust temperature	K
$T_{s_{IN}} = T_{s_0}$	inlet steam temperature	K
State variables		
T_{s_i}	i - *th* cell outlet steam temperature	K
Algebric variables		
$T_{e_{N-i}}$	i - *th* cell outlet exhaust temperature	K
Output variables		
$T_{e_{OUT}} = T_{e_N}$	outlet exhaust temperature	K
$T_{s_{OUT}} = T_{s_N}$	outlet steam temperature	K
Auxiliary Variables		
$h_{s_{IN}}$	inlet steam enthalphy	J kg s^{-1}
$h_{s_{OUT}}$	outlet steam enthalphy	J kg s^{-1}
E_{e_i}	i - *th* cell exhaust stored energy	J
ρ_{s_i}	i - *th* cell steam mass density	kg m^{-3}
T_{m_i}	i - *th* cell metal mean temperature	K
$\overline{T}_{s_i}$	i - *th* cell steam mean temperature	K

Table 5: Heat exchanger model - variables.

c_m	metal heat capacity	J K^{-1} kg^{-1}
c_{p_e}	exhaust specific heat	J K^{-1} kg^{-1}
c_{p_s}	steam specific heat	J K^{-1} kg^{-1}
γ_I	metal-steam (inner side) heat exch. coeff.	W m^{-2} K^{-1}
γ_O	exhaust-metal (outer side) heat exch. coeff.	W m^{-2} K^{-1}
Γ_e	exhaust-metal empiric heat exch. coeff.	W K^{-1} $kg^{-0.6}$ $s^{-0.6}$
Γ_s	steam-metal empiric heat exch. coeff.	W K^{-1} $kg^{-0.6}$ $s^{-0.6}$
i	cell index	
m_{m_i}	i - *th* cell metal mass	K
m_{s_i}	i - *th* cell steam mass	K
N	number of cell	
S_I	exchanger inner surface	m^2
S_{I_i}	i - *th* cell inner surface	m^2
S_{O_i}	i - *th* cell outer surface	m^2
V_{s_i}	i - *th* cell steam volume	m^3

Table 6: Heat exchanger model - parameters.

$$\frac{dE_{e_i}}{dt} = q_e c_{p_e}(T_{e_{N-i}} - T_{e_{N-i+1}}) - \gamma_O S_{O_i}(T_{e_{N-1}} - T_{m_i}) = 0 \tag{28}$$

By replacing T_{m_i} with T_{s_i} in (28), in accordance with (26) and the assumption of equality between $\overline{T}_{s_i} = T_{s_i}$, and by rearranging it in order to make $T_{e_{N-i+1}}$ explicit, one obtains

$$T_{e_{N-i+1}} = T_{e_{N-i}} - \frac{\gamma_O S_{O_i}}{q_s c_{p_s}}(T_{e_{N-1}} - T_{s_i}) \tag{29}$$

According to a commonly adopted empirical law, the quantity $\gamma_O S_{O_i}$ has to be replaced with the term $\Gamma_e q_e^{0.6}$, so as to introduce the proper dependance of the heat transfer on the exhausts flow rate.

Then, it is possible to write the final equations (one algebraic and one dynamic) for the thermodynamic behavior of each cell:

$$\begin{cases} \frac{dT_{s_i}}{dt} = \frac{1}{\tau_{s_i}+\tau_{m_i}}\left[(T_{s_{i-1}} - T_{s_i}) + \frac{\Gamma_e q_e^{0.6}}{q_s c_{p_s}}(T_{e_{N-i}} - T_{s_i})\right] \\ \\ T_{e_{N-i+1}} = T_{e_{N-i}} - \frac{\Gamma_e q_e^{0.6}}{q_e c_{p_e}}(T_{e_{N-1}} - T_{s_i}) \end{cases} \tag{30}$$

Economizers (exhaust-to-water heat exchangers)

As already pointed out, the proposed model can be applied also to the economizers provided that the following changes are made:

1. the water mass density ρ_ω is assumed to be constant ($\rho_\omega = 10^3$ kg m^{-3}), thus simplifying the computation of the time constants;

2. the introduction of a pressure-varying c_{p_w} coefficient is unnecessary.

As a consequence of these changes, the value of the inlet water pressure is not needed.

The Drum Boiler

A detailed scheme of the drum boiler (or evaporator) is provided in Figure 10. Its main components are:

- the drum, a cylindrical cavity containing saturated steam and water;
- the downcomers, the descending pipelines conveying the water from the drum to the re-circulation pump;
- the risers, i.e. the ascending pipelines which allow the thermal exchange between the hot exhausts and the steam-water fluid, thus giving rise to the steam production.

Concerning the process dynamics, the drum boiler behavior is described through the steam-water pressure and the water level inside the drum.

The steam production, is strictly correlated with the pressure dynamics. Based on mass and energy balances, the drum boiler model is the most complex component of the HRSG: its modeling required significant efforts to capture the true process nonlinearities. The reasons of such a complexity may be briefly summarized.

1. The strong coupling between the thermodynamic and the fluid dynamic processes. Suffices here to remember that all the thermodynamic properties of the saturated steam and water are functions of the drum boiler pressure. Moreover, by virtue of the interaction between the evaporator and the steam turbine, the outgoing steam flow rate directly depends on the pressure level established inside the cavity (and turns out to be approximatively proportional to the generated mechanical load).

2. The presence of both (saturated) steam and water (in the risers zone, in particular).

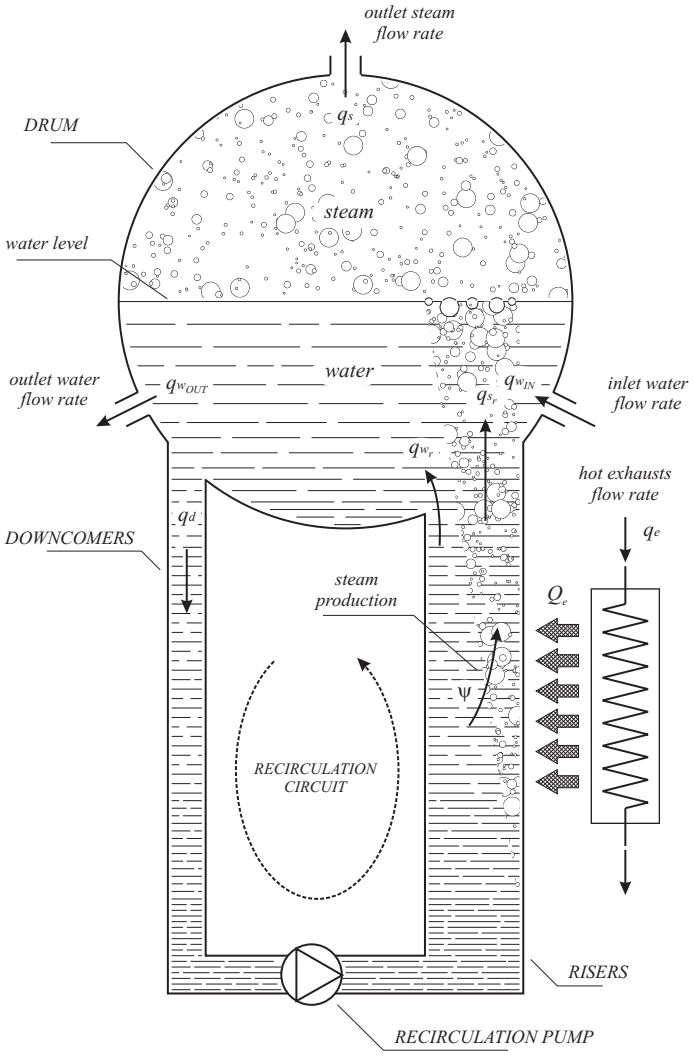


Figure 10: Evaporator scheme

A very complete model of the drum boiler has been developed in [6], where natural circulation drum boilers, characterized by strong nonlinearities and an accentuated "shrink and swell dynamics"have been considered. In this Thesis, the model described in [4] has been adopted. It is a $3rd-order$ model where the distribution of steam and water inside the drum has been treated in a simpler way than in [6]. For conciseness, the model is not reported here, and the interested reader is referred to [4] for details.

1.1.3 Steam turbine stage

As for the gas turbine, an algebraic model turns out to be suitable also for the steam turbine stages. Actually, the system under inspection includes both the steam turbine and the inlet (or "throttle") valve: the utilization of this latter depends on the adopted plant management strategy. Usually, the steam turbine valves are kept completely open (this working mode is commonly defined as *sliding pressure*), in order to guarantee the highest plant efficiency. Alternatively, it may be possible to adopt a different policy of plant regulation by keeping the throttle valves slightly closed at low loads, while opening them wide to exploit the so called steam supply when a load request occurs 10 . The system including the steam turbine and its throttle valve (corresponding to a pressure stage of the whole steam turbine) will be hereafter denoted as "Steam Turbine System"(STS). Its role is to generate the mechanical load by interacting with the steam generator. Figure 11 offers a schematic diagram of the STS, while Table 7 provides a list of all its variables and parameters.

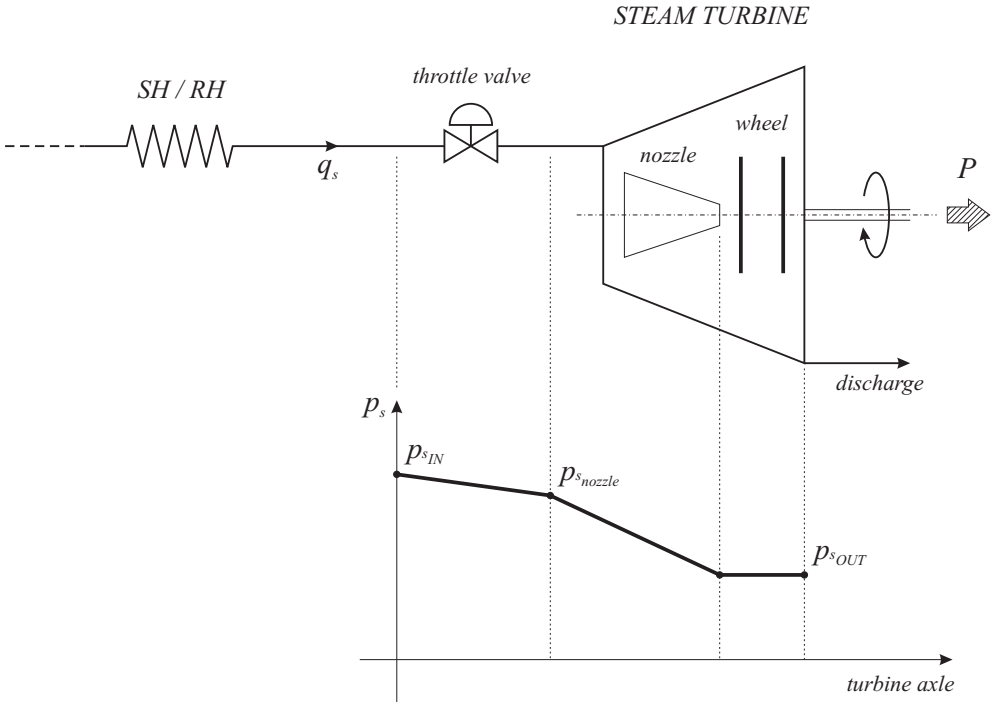


Figure 11: Diagram of the STS

The proposed algebraic model consists of three equations.

1. The first one deals with the steam flow rate. As previously remarked,

Input variables		
$p_{s_{IN}}$	inlet steam pressure	Pa
$p_{s_{OUT}}$	outlet steam pressure	Pa
$T_{s_{IN}}$	inlet steam temperature	K
ϑ	throttle valve opening	
Output variables		
P	generated mechanical power	W
q_s	steam flow rate	kg s^{-1}
$T_{s_{OUT}}$	outlet steam rate	K
Auxiliary Variables		
$h_{s_{IN}}$	inlet steam specific enthalphy	J kg s^{-1}
$h_{s_{ISO}}$	steam isentropic specific enthalphy	J kg s^{-1}
$h_{s_{OUT}}$	outlet steam specific enthalphy	J kg s^{-1}
$p_{s_{nozzle}}$	nozzle inlet steam pressure	Pa
$\rho_{s_{IN}}$	inlet steam mass density	kg m^{-3}
$T_{s_{ISO}}$	i - th cell steam mean temperature	K
Parameters		
A_n	nozzle section	m^2
A_v	throttle valve section	m^2
c_p	mean steam specific heat	J K^{-1} kg^{-1}
η_t	turbine efficiency	J kg s^{-1}

Table 7: Steam Turbine System - variables and parameters

the steam turbine interacts with the evaporator. This latter imposes the upstream pressure level on the basis of both the heat power provided by the hot exhausts to the risers and the (required) outlet steam flux. A pressure drop occurs through the superheaters chain: on the basis of the resulting inlet pressure, the STS determines the steam flow rate, which needs to be provided by the steam generator;

2. The second one regards the generated mechanical load;

3. The third one allows the computation of the outlet steam temperature.

As for the heat exchangers, a temperature-based model has been adopted. This choice affects the second and particularly the third model equation.

Steam flow rate

As proposed in [13] in detail, the steam flow rate through the throttle valve is described by the equation:

$$q_s = k_v A_v \sqrt{\rho_{s_{IN}} p_{s_{IN}}} \chi \left(\frac{p_{s_{nozzle}}}{p_{s_{IN}}} \right) \tag{31}$$

where $\chi()$ is a *universal* function, which applies to all valves, while k_v is a constant parameter. The superheated steam may be considered as a perfect gas, thus allowing to assume that:

$$\rho_{s_{IN}} = \frac{p_{s_{IN}}}{R T_{s_{IN}}}$$

where R is the thermodynamic constant of the steam-gas. The equation 31 became:

$$q_s = k_v A_v p_{s_{IN}} \sqrt{\frac{1}{R T_{s_{IN}}}} \chi \left(\frac{p_{s_{nozzle}}}{p_{s_{IN}}} \right) \tag{32}$$

The same relation is valid for the turbine nozzle too, i.e.

$$q_s = k_n A_n p_{s_{nozzle}} \sqrt{\frac{1}{R T_{s_{nozzle}}}} \chi \left(\frac{p_{s_{OUT}}}{p_{s_{nozzle}}} \right) \tag{33}$$

where k_n is the constant parameter (as shown in Figure 11, a simplified scheme of the steam turbine including a single internal stage has been assumed as the reference). As for the throttle valve, it is possible to assume that $T_{s_{nozzle}} \approx T_{s_{IN}}$, while, in accordance with the design features of the

steam turbines, the pressures ratio $p_{s_{OUT}}/p_{s_{nozzle}}$ is constant with the steam flow rate q_s. So, by comparing (32) with (33), one obtains

$$\frac{p_{s_{IN}}}{p_{s_{nozzle}}}\chi\left(\frac{p_{s_{nozzle}}}{p_{s_{IN}}}\right) \tag{34}$$

Equation (34) proves that the quantity $p_{s_{nozzle}}/p_{s_{IN}}$ can be regarded as a function of the only inlet section A_v , regulated through the corresponding throttle valve, i.e.

$$p_{s_{nozzle}}/p_{s_{IN}} = f_t(A_v)$$

Thanks to the most common design practice, which aims at an easier and faster turbine control, the function $f_t(A_v)$ exhibits an almost linear behavior with the valve control signal ϑ

$$f_t(A_v) \approx k_\vartheta \vartheta$$

where k_ϑ is a constant gain. It follows that

$$q_s = k_t \vartheta p_{s_{IN}} \sqrt{\frac{1}{T_{s_{IN}}}} \tag{35}$$

The coefficient k_t can be easily tuned on the basis of the nominal operating point.

Generated mechanical power

As for the gas turbine, the generated power is given by:

$$P = q_s(h_{s_{IN}} - h_{s_{OUT}}) \tag{36}$$

To match the adopted temperature-based modeling of thermodynamics, equation (36) needs to be modified as follows:

$$P = q_s c_p(T_{s_{IN}} - T_{s_{OUT}}) \tag{37}$$

where the equivalent specific heat c_p may be identified on the basis of the rated working conditions.

Outlet steam temperature

From a thermodynamic point of view, each steam turbine stage is analogous

to the gas turbine. Therefore, the outlet exhausts enthalpy and temperature equations, (15) and (16) respectively, may be regarded as a reference. However, in this case, it is not required to replace the outlet-inlet pressure ratio with an alternative function of the flow rate. Rather, it is necessary to consider the difference between the behavior of the superheated steam and a perfect gas (like the hot exhausts, in the gas turbine), which claims to carefully inspect the reliability of a temperature-based model formulation. By recalling the efficiency definition, here referred to the steam case:

$$\frac{h_{s_{IN}} - h_{s_{OUT}}}{h_{s_{IN}} - h_{s_{ISO}}}$$

the following expression of the steam enthalphy at the turbine outlet is obtained:

$$h_{s_{OUT}} = h_{s_{IN}} \left(1 - \eta_t \left[1 - \left(\frac{p_{s_{OUT}}}{p_{s_{IN}}}\right)^e\right]\right) \tag{38}$$

where the exponent $e = \alpha / c_p$ may be computed on the basis of the selected working point

$$e = \frac{\log\left(\frac{h_{s_{ISO}}}{h_{s_{IN}}}\right)}{\log\left(\frac{p_{s_{OUT}}}{p_{s_{IN}}}\right)} \tag{39}$$

Unlike the gas turbine, there are two different specific heat for input steam and for output steam. Because of this, a coefficient k_{iso} is defined:

$$k_{iso} = \frac{T_{s_{ISO}}}{T_{s_{IN}}} \frac{h_{s_{IN}}}{h_{s_{ISO}}}$$

Then it's possible to derive the required expression of the outlet steam temperature:

$$T_{s_{OUT}} = T_{s_{IN}} \left(1 - \frac{c_{p_{OUT}}}{c_{p_{ISO}}} \eta_t \left[1 - k_{iso} \left(\frac{p_{s_{OUT}}}{p_{s_{IN}}}\right)^e\right]\right) \tag{40}$$

1.1.4 Mixers and attemperators

The temperature-based modeling of a single-fluid (water or steam) mixer is a quite simple task, according to mass balances and the adiabatic mixing law. The basic configuration is shown in Figure 12: it roughly reproduces both the two steam-mixers situated after the HP and IP steam turbines outlet and the water-mixer located before the ECO-LP. This latter is equipped with a recirculation circuit, intended for the regulation of the inlet water temperature: the aim is to avoid too low temperatures of the exhausts leaving the HRSG, which cause corrosive condensation in the colder part of the boiler and pollutants production. A list of the system variables is provided in Table

8.

According to the mass balance, one has

$$q_{s,w_{OUT}} = q_{s,w_{IN_1}} + q_{s,w_{IN_2}} \tag{41}$$

while the adiabatic mixing law states that

$$h_{s,w_{OUT}} = \frac{q_{s,w_{IN_1}} h_{s,w_{IN_1}} + q_{s,w_{IN_2}} h_{s,w_{IN_2}}}{q_{s,w_{IN_1}} + q_{s,w_{IN_2}}} \tag{42}$$

which may be rewritten as

$$c_{p_{OUT}} T_{s,w_{OUT}} = \frac{q_{s,w_{IN_1}} c_{p_{IN_1}} T_{s,w_{IN_1}} + q_{s,w_{IN_2}} c_{p_{IN_2}} T_{s,w_{IN_2}}}{q_{s,w_{IN_1}} + q_{s,w_{IN_2}}}$$

By assuming

$$c_{p_{OUT}} \approx c_{p_{IN_1}} \approx c_{p_{IN_2}}$$

as it is reasonable, one obtains

$$T_{s,w_{OUT}} = \frac{q_{s,w_{IN_1}} T_{s,w_{IN_1}} + q_{s,w_{IN_2}} T_{s,w_{IN_2}}}{q_{s,w_{IN_1}} + q_{s,w_{IN_2}}} \tag{43}$$

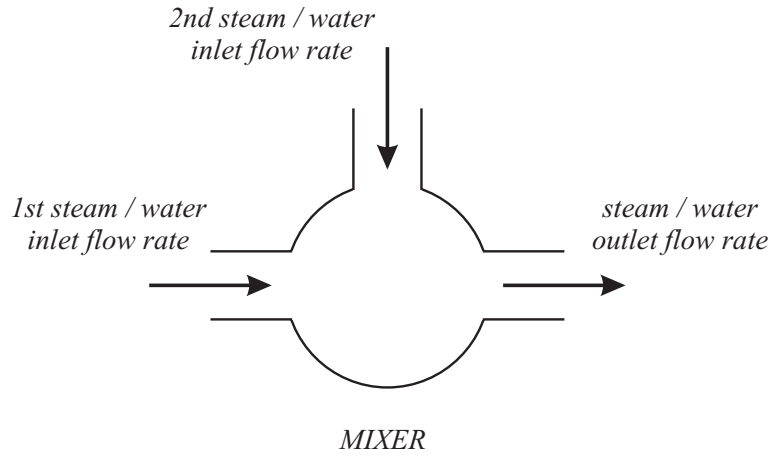


Figure 12: Basic configuration of a mixer

For the attemperators, which add an adjustable cold water flow rate to the superheated steam flux to reduce the steam temperature, the assumption about the approximation of specific heat ($c_{p_{OUT}} \approx c_{p_{IN_1}} \approx c_{p_{IN_2}}$) is not valid.

First inlet steam/water flux		
$c_{p_{IN_1}}$	specific heat	J kg^{-1} s^{-1}
$h_{s,w_{IN_1}}$	specific entalpy	J kg^{-1}
$q_{s,w_{IN_1}}$	flow rate	kg s^{-1}
$T_{s,w_{IN_1}}$	temperature	K
Second inlet steam/water flux		
$c_{p_{IN_2}}$	specific heat	J kg^{-1} s^{-1}
$h_{s,w_{IN_2}}$	specific entalpy	J kg^{-1}
$q_{s,w_{IN_2}}$	flow rate	kg s^{-1}
$T_{s,w_{IN_2}}$	temperature	K
Outlet steam/water flux		
$c_{p_{OUT}}$	specific heat	J kg^{-1} s^{-1}
$h_{s,w_{OUT}}$	specific entalpy	J kg^{-1}
$q_{s,w_{OUT}}$	flow rate	kg s^{-1}
$T_{s,w_{OUT}}$	temperature	K

Table 8: Mixer - variables and parameters

A possible way to perform a temperature-based modeling of the SH and RH attemperators consists of a proper rearrangement of (42), in accordance with a suitable local linearization of the enthalpy-temperature relation.
Actually, such a linearization is required only for the steam fluid. In fact, the value of enthalpy of the desuperheating water, provided by the LP drum boiler, is available through the adopted evaporator model, which relies on properly identified polynomial functions of the thermodynamic properties of the saturated steam and water. It is convenient to remember that the injection of a small cold water flow rate into the superheated steam flux implies only a cooling action of the steam (no state modification occurs).
A linearization of the inlet steamenthalpy is adopted

$$h_{s_{IN}}(T_{s_{IN}}) = h_{s_0} + c_{p_0} T_{s_{IN}} \tag{44}$$

All the model variables are summarized in Table 9. The outlet steam enthalpy is given by the adiabatic mixing law (42)

$$h_{s_{OUT}} = \frac{q_{s_{IN}} h_{s_{IN}}(T_{s_{IN}}) + q_{w_{IN}} h_{w_{IN}}}{q_{s_{IN}} + q_{w_{IN}}}$$

Finally, recalling (44), the outlet steam temperature may be computed as

$$T_{s_{OUT}} = \frac{1}{c_{p_0}}(h_{s_{OUT}} - h_{s_0}) \tag{45}$$

while the outlet steam flow rate is obtained from the mass balance

$$q_{s_{OUT}} = q_{s_{IN}} + q_{w_{IN}}$$

By means of the Steam Tables it is possible to perform the required linearization, which allows the proper tuning of the quantities c_{p_0} and h_{s_0} , on the basis of the nominal working conditions.

1.1.5 Model of the thermo-mechanical stresses in the turbine rotor

In the start-up phase of a CCPP, one of the main requirements is to limit the thermo-mechanical stresses of the turbine. In fact, the overall life of the plant mainly depends of the stress level reached in the start-up and shut-down phases, rather than in the total run time. For this reason, a reliable model of the stress of of paramount importance for the design of the start-up, i.e. of the main object of this work. In the following, the model developed in [9] and subsequently used in [12], is briefly summarized.
In our calculations leading to the final model, it is possible to assume a one

Inlet superheated steam		
$h_{s_{IN}}$	specific entalpy	J kg^{-1}
$q_{s_{IN}}$	flow rate	kg s^{-1}
$T_{s_{IN}}$	temperature	K
Inlet subcooled water		
$h_{w_{IN}}$	specific entalpy	J kg^{-1}
$q_{w_{IN}}$	flow rate	kg s^{-1}
Outlet steam/water flux		
$h_{s_{OUT}}$	specific entalpy	J kg^{-1}
$q_{s_{OUT}}$	flow rate	kg s^{-1}
$T_{s_{OUT}}$	temperature	K

Table 9: Attemperators - variables

dimensional radial model of the component we are interested in. The geometry of this desired component is a smooth cylindrical cavity and we assume that the total length is much larger than the diameter. In this context we can assume an axial-symmetrical temperature distribution. The assumption of axial-symmetric conditions simplifies the modeling and the solution of the necessary equations. An approximation of the one-dimensional radial temperature heat transfer can be expressed with cylindrical coordinates in the following form:

$$\frac{\rho c_p}{k}\frac{\delta T}{\delta t} = \frac{1}{r}\frac{\delta}{\delta r}\left(r\frac{\delta T}{\delta r}\right) \tag{46}$$

where ρ and c_p represent the density and the specific thermal conductivity of the material used in the component. T is the temperature of this material and r and t respectively represent the radial coordinate and time. For the numerical solution od this equation we may use spatial discretization involving the finite difference method. The right hand of equation (46) is hence discretized and rewritten as:

$$\frac{1}{r_i}\frac{\left[\left(r\frac{\delta T}{\delta r}\right)_{i+1/2} - \left(r\frac{\delta T}{\delta r}\right)_{i-1/2}\right]}{r_{i+1/2} - r_{i-1/2}} = \frac{1}{r_i\left(r_{i+1/2} - r_{i-1/2}\right)}\left(r_{i+1/2}\frac{T_{i+1} - T_i}{r_{i+1} - r_i} - r_{i-1/2}\frac{T_i - T_{i-1}}{r_i - r_{i-1}}\right) \tag{47}$$

The left-hand-side of (46) can be set equal to the equation (47), to obtain the results shown in (48).

$$\frac{\rho c_p}{k}\frac{\delta T}{\delta t} = A_i T_{i-1} + B_i T_i + C_i T_{i+1} \tag{48}$$

where A_i and C_i are the coefficients:

$$\begin{gathered} A_i = \frac{r_{i-1/2}}{r_i(r_i - r_{i-1})(r_{i+1/2} - r_{i-1/2})} \\ B_i = -(A_i + C_i) \\ C_i = \frac{r_{i+1/2}}{r_i(r_i - r_{i-1})(r_{i+1/2} - r_{i-1/2})} \end{gathered}$$

Then, the general conditions for the stress can be rewritten as

$$\begin{gathered} T_i = T_{int} \\ T_{N_r} = T_{ext} \\ \varphi_{int} = -k\frac{T_2 - T_1}{r_2 - r_1} \\ \varphi_{ext} = -k\frac{T_{N_r} - T_{N_{r-1}}}{r_{N_r} - r_{N_{r-1}}} \end{gathered}$$

The Fourier discretized component of the model of the internal and external stresses can be written as in equation (49)

$$\sigma_{T,r} = 0 \sigma_{T,\theta} = \frac{\alpha E}{1 - v}(T_m - T_{int/ext}) \sigma_{T,z} = \frac{\alpha E}{1 - v}(T_m - T_{int/ext}) \tag{49}$$

where α is the dilation constant, E is the Young Modulo, v is the Poisson ratio relatively to the material used in the component and T_m , T_{int} and T_{ext} are the medium, internal and external surface temperatures of the material. The mechanical stresses in the rotor can be written as in equation (50)

$$\sigma_{c,\theta} = \frac{3 + v}{4}\rho\omega^2 \left(r_{int/ext}^2 - \frac{1 - v}{3 + v} r_{int/ext} \right) \tag{50}$$

The external surface of the rotor is the part of the rotor that is in direct contact with the hot gases.

1.2 Plant data

In [4] and [8] CCPP models have been developed starting from the elementary models of the subsystems composing a CCPP, some of which have been described in the previous sections. Both in [4] and in [8], HRSG subsystems with three pressure stages have been developed and simulated in different environments (Matlab/Simulink in [4] and Modelica in [8]). These models and simulators, although very accurate, are not suitable for the development of innovative strategies for the start-up phase. In particular, the Simulink model described in [4] is computationally very demanding and not suited to represent the significant dynamic phenomena inside the CCPP at very low loads. On the contrary, the Modelica model described in [8], is more general, since it can completely describe the overall operating range of the CCPP. However, the model is even too complex for the project of the start-up procedure. For these reasons, in the framework of the EU project "Hierarchical and Distributed Control of Large Scale Systems", starting from the Modelica model first introduced in [9], [8] and already used in [12], SUPELEC has developed a simpler CCPP model, composed by a gas turbine (GT) and only one pressure stage in the HRSG, see [15].

Gas Turbine
The Gas Turbine system, which does not represent a limiting factor for the start-up since the HRSG dynamics is much slower than the GT one, is characterized by the following nominal data:

- maximal power of 235 [MW];
- nominal flue gas flow rate of 585.6 [kg/s];
- nominal fuel flow rate of 12.1 [kg/s].

Heat Recovery Steam Generator
The considered HRSG has a single pressure circuit (high pressure) with:

- nominal HP steam flow rate of 70.6 [kg/s];
- nominal HP steam pressure of 129.6 [bar].

The HP circuit has the following components:

- economizer;
- steam drum;
- evaporator;

- superheater.

Steam Lines
The system of pipes connecting the HRSG, the ST and the condenser, has a length of 130 [m] and is provided with an evacuation system to eliminate the condensates.

Input and output variables
Since the objective of this project is to design a control system for the plant, in particular with reference to the start-up phase, it is mandatory to have the possibility to monitor the main input and output variables. From the model developed by SUPELEC, it is possible to have full knowledge of the following variables.
Inputs:

- GT_Load: load of gas turbine;
- $Feed\_Drum$: feedwater flow rate for the HRSG circuit;
- $Feed\_DSH$: water flow that feeds the desuperheater;
- $Bypass\_ST$: position of the bypass valve;
- $Valve\_ST$: position of the turbine admission valve;
- $BreakerClosed$: generator grid breaker of the steam turbine.

Outputs:

- GT_Power: power produced by the gas turbine;
- GT_Fuel: fuel flow in the gas turbine;
- $Drum\_Level$: drum level of the HRSG;
- T_SH: temperature of steam in the superheater;
- $Steam\_Press$: steam pressure;
- $Header\_Stress$: thermodynamical stress of the header;
- T_SL: temperature of steam in the steam line;
- ST_Power: power produced by the steam turbine;
- $ST\_Frequency$: frequency of the steam turbine;

- ST_Stress: steam turbine rotor thermal stress;
- W_ST: steam flow.

Therefore, the model has six inputs and eleven outputs, but not all the inputs influence in the same way all the outputs. In the analysis phase it will be important to evaluate the sensitivity of the outputs with respect to the inputs, with particular care to the thermodynamic stress.
The range of the input variables is reported in the following:

- $0.075 \leq GT\_load(percentage) \leq 1$, note that the starting point is not zero to avoid singularities in the flue gas side model. These values correspond to the minimum load (17.625 [MW]) and maximum load (235 [MW]);
- $3.25 \leq Feed\_Drum$ [kg/s];
- $0 \leq Feed\_DSH$ [kg/s];
- $0 \leq Bypass\_ST \leq 1$ (close/open);
- $0 \leq Valve\_ST \leq 1$ (close/open);
- $BreakerClosed = boolean(1, 0)$; if it is equal to 1, the steam turbine is connected to the grid, otherwise it is 0.

1.3 Modelica Model

In order to develop an efficient simulator of the CCPP, the dynamic model previously described has been implemented in Modelica, a powerful object oriented language used for modeling large, complex, and heterogeneous physical systems.
The most important features of Modelica are:

- Object-oriented modeling. This feature allows one to create physical model components, employed to support hierarchical structuring, reuse, and evolution of complex models from several different domains.
- Acausal modeling. Modelica is based on equations instead of assignment statements. Direct use of equation gives a better reuse of model components, since the components adapt to the data flow context in which they are used.
- Physical modeling. The model components can correspond to real physical objects. The structure of the model is more natural since it implements the traditional concept based on block-oriented models.

The use of the modeling language Modelica in model based process control applications has many advantages. In fact, Modelica allows one to describe a large class of models (linear, nonlinear, hybrid etc.) and many efficient software tools [2], [1], [3] are available. On the other hand, there are few tools supporting Modelica for static and dynamic optimization, see e.g. [10] for a remarkable exception.

The CCPP model has been developed with the Dymola tool, see [3], and uses the components from the ThermoPower library, see [14], an open-source library for modeling thermal power plants at the system level, as well as to support the design and validation of control systems. The library has been developed according to following principles:

1. the models are derived from first principle equations or from known empirical correlations;
2. the model interface is totally independent of the modeling assumption adopted for each model, to achieve full modularity;
3. the level of detail of the models is flexible;
4. the inheritance mechanism is used with limitation, to maximize the code readability and modifiability.

The plant model has the following features:

- can represent the whole start-up sequence, [15];
- includes a model of the thermal stress in some critical components, which is the most limiting factor for the start-up time, [15];
- neglects phenomena and components which are not critical for the start-up sequence, in order to keep the complexity of the model at a reasonable level.

The highest level of the CCPP Modelica simulator is shown in Figure 13; all the components (exploding the main model of Figure 13) are shown in Figure 14. In this scheme, it is possible to see the blocks corresponding to the gas turbine system, the HRSG, the steam line, the sink (substituting the intermediate pressure stage) and the steam turbine. The input/output blocks are standard Modelica blocks. A higher level of detail is presented in [9]

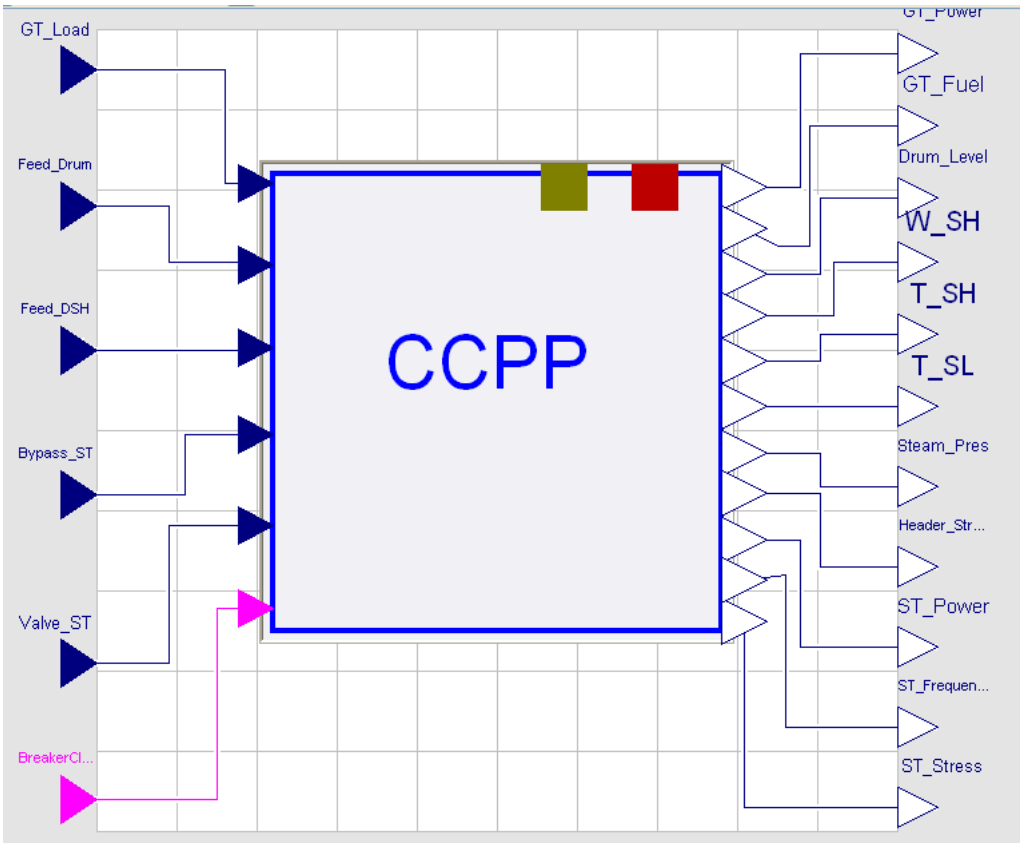


Figure 13: Modelica model of CCPP

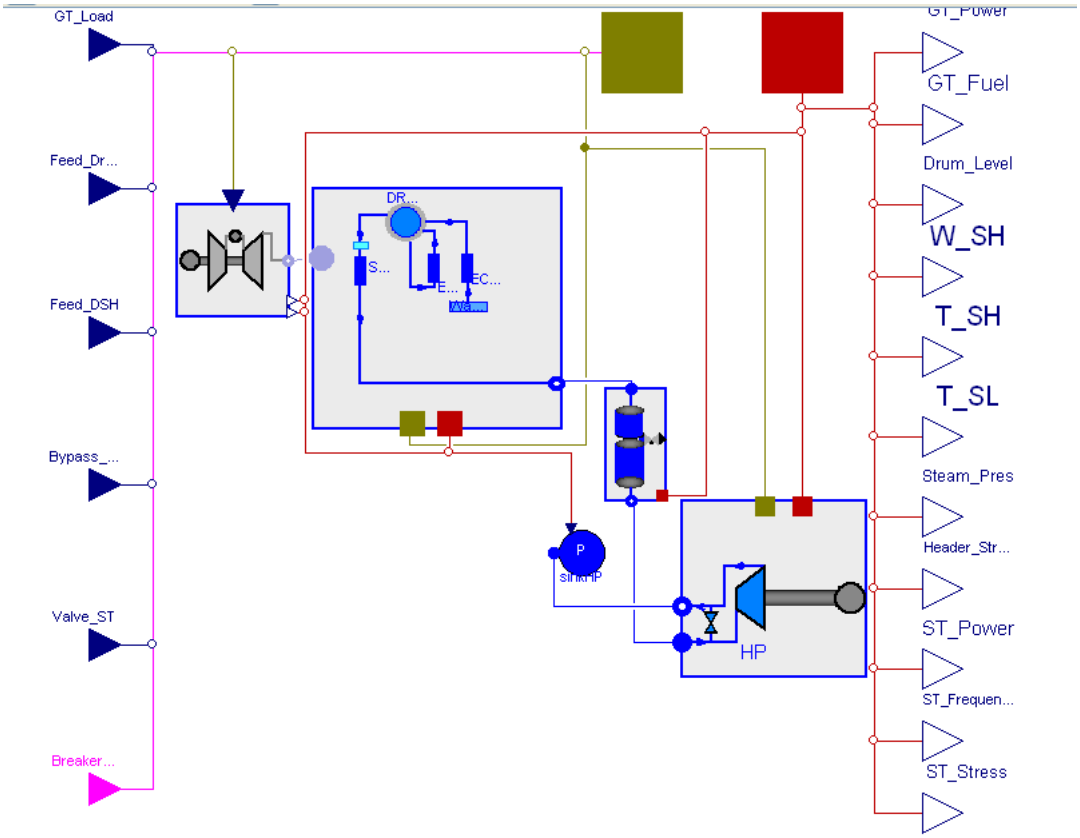


Figure 14: Dymola components

2 Chapter 2: Identification of a simplified model

The CCPP model described in the previous chapter is highly complex and characterized by strongly nonlinear dynamics. Moreover, as already noted, the Modelica environment is not yet equipped with adequate ancillary tools for control design and optimization. For these reasons, in this chapter attention is paid to the identification of a simpler model, to be implemented in the Matlab/Simulink environment, which will be subsequently used to determine the plant start-up procedure. The Modelica simulator will then be used as a reference model to produce the data for identification and validation.

2.1 Identification approach: interpolation of linear models

The identification of nonlinear process models is a challenging task: nonlinearities, changes in the directionality of the gain, discontinuities are phenomena typical of the CCPP behavior which is very difficult to capture with standard estimation procedures. Many different solutions have been proposed in the literature, for example nonlinear structures based on Neural Networks (NN) have been extensively studied. However, the models so obtained are typically *black-box* and do not guarantee a high level of transparency, so that the main dynamic phenomena are often hidden to the user. Moreover, the identification phase often relies on emphirical considerations, for instance in the definition of the model structure (number of neurons and hidden layers in NN, number and type of regressors in polynomial models).
For these reasons, the approach adopted in the following is different and quite easy to understand for anyone familiar with the use of linear (or linearized) models. The idea is simple and requires to decompose the nonlinear identification problem into a number of linear identifications using process information. As a matter of fact, a number of plant operating points is selected, possibly covering the overall operating range of the system. Then, at any operating point, a local linear model is estimated by means of standard identification procedures (based on the Least Squares method, for example). Finally, the overall identified plant model is obtained by suitably interpolating at any operating point the local linear models through the use of weights specified by properly chosen membership functions. Indeed, these weights define the validity, at any operating point, of the locally estimated transfer functions. This method, although pretty empirical, has already been successfully used in a number of applications, see [11], [5].

In the case of the CCPP system, the model to be identified has three inputs: GT_Load, $Valve\_ST$ and $Feed\_DSH$ (recall that the $Feed\_Drum$ is controlled with a local regulator to avoid problems of an excessive increase/decrease of the level inside the drum boiler, while the $Bypass\_ST$ and the $BreakerClosed$ are constant with the bypass valve totally open and the load grid connected). The gas turbine load is used as the scheduling variable, i.e. as the variable specifying the current operating condition. Therefore, the local linear models are identified at different steady-state conditions corresponding to a fixed value of the GT_Load. Interpolation of the local linear models is then performed depending on the value of the load. This choice is somehow obvious, since this variable governs the start-up and shut-down phases of the plant.

At each operating point, a local perturbation with the shape of a square signal has been imposed at any input of the Modelica model to generate the data subsequently used to identify with the Matlab Identification Toolbox a local linear model.
Then, letting y the (output) variable to be identified, $\overline{y}$ its nominal value in the considered operating point, $\overline{u}_i$ the nominal value of the $i-th$ input ($i = 1, 2, 3$), μ_i the imposed perturbation, δ_i the contribution of the $i-th$ input to the output and $G_i(s)$ the local (identified) linear transfer function relating the $i-th$ input to the δ_i component, the scheme of the overall identified local model for the generic output y is reported in Figure 15.

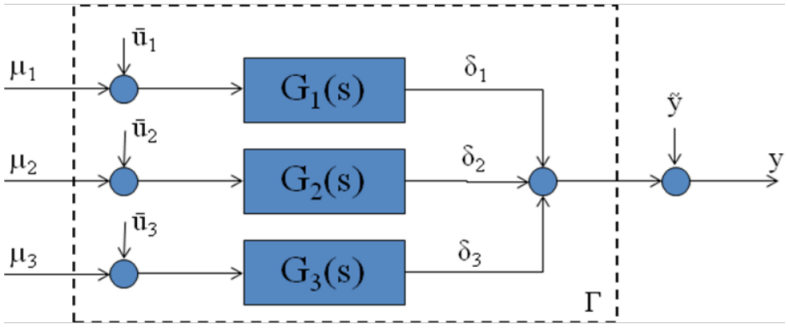


Figure 15: Scheme of the identified model at generic operating point.

Once a local linear model has been identified for any output variable of interest at any of the selected N operating points, a suitable interpolation procedure must be defined. To this regard, an approach based on the use of membership functions has been adopted. The membership functions must be selected so that for any value of the scheduling variable (the GT_Load) the weight associated to the model estimated at the nearest operating point

provides a bigger contribution than the others. The final identified model turns out to have the structure reported in Figure 16.

An example of the adopted membership functions is reported in Figure 17: it is important to note that in every point the sum of the membership functions is equal to one. Every function is associated to one of the operating points where local models have been identified, however, the slope of the membership functions must be properly tuned to obtain an overall model whose validity covers the complete operating range of the CCPP. The definition of the membership functions for each output variable requires a specific tuning, and the definition of their parameters can be time consuming. However, in future developments, the tuning of the membership functions could be automatized through an optimization procedure.

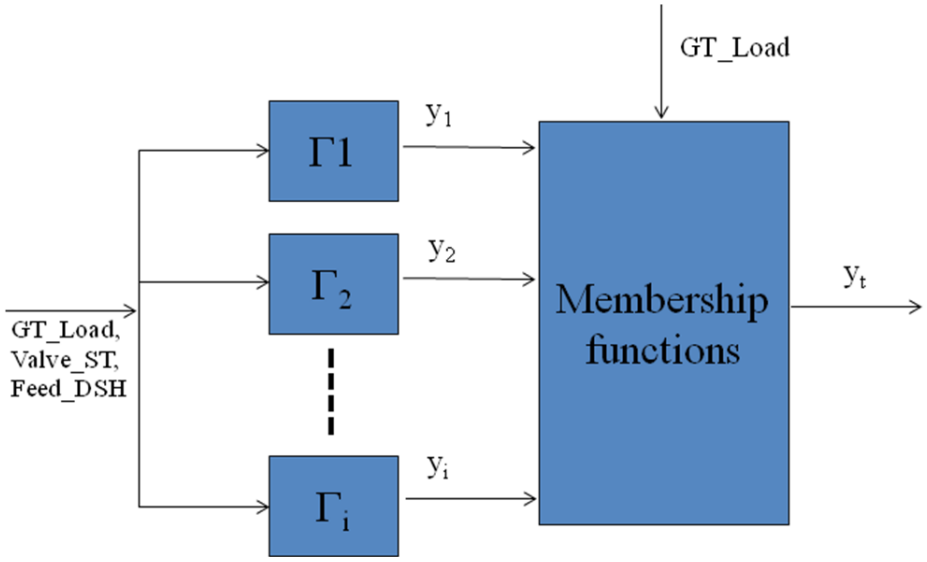


Figure 16: Interpolation with membership functions.

In conclusion, the identification algorithm for the CCPP can be summarized as follows:

1. determine the operating points where to perform local identifications;

2. simulate the Dymola model by forcing small perturbations around any operating point and store the corresponding transient to be used for identification;

3. identify the local linear models with the *Matlab Identification Toolbox*;

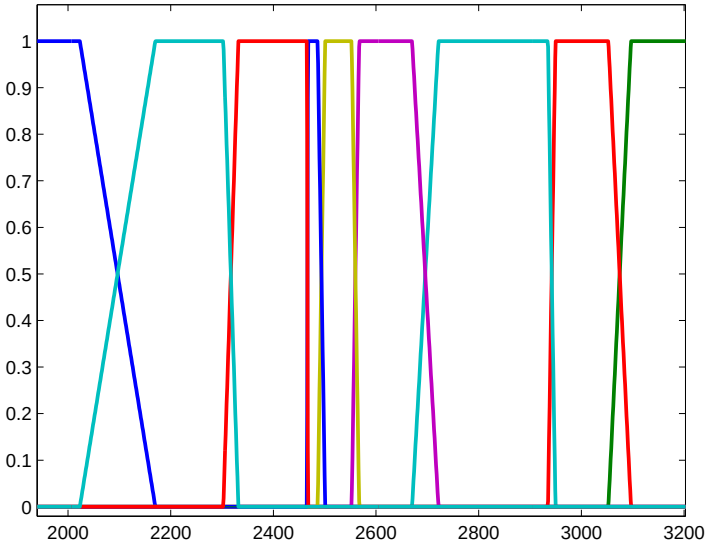


Figure 17: Example of membership functions.

4. select the membership functions and build the interpolation scheme with *Matlab Simulink*.

2.2 Selection of the operating points for identification

The choice of the operating points where to perform the identification is a tradeoff between physical properties of the plant and interpolation requirements. In fact, some points have particular physical relevance, but can not be sufficient to describe the dynamics of the plant. Initially, the following points were chosen:

- 100% of *GT_Load*, defines the full-load state of the model;
- 75% of *GT_Load*, intermediate point;
- 60% of *GT_Load*, under of this load level, the gas combustion is poor and the emissions start to increase;
- 50% of *GT_Load*, under of this level, the system needs a pressure regulator to stay at 60 bar;
- 40% of *GT_Load*, intermediate point;

- 15% of *GT_Load*, point near the lowest operating condition of the system (7.5%).

The previous operating points were not sufficient to estimate a satisfactory model describing the CCPP behavior. In particular, it was clear that critical phenomena occurred near the 60% of the load, where an inversion of some process gains appears. To increase the validity of the model, the following points were added:

- 65% of *GT_Load*;
- 57% of *GT_Load*;
- 25% of *GT_Load*.

2.3 Simulations on Dymola

At each operating point, the same perturbation has been given to the three inputs. It is essentially a square wave: this is mandatory if the simulations are performed with the open loop model (level loop open) to guarantee that the level in the drum does not exceed prescribed values. On the contrary, with the level in closed-loop, different perturbations could be considered. In Table 10, different values of the inputs are reported (note that the inputs corresponding to the *Feed_Drum*, the *BreakerClosed* and the *Bypass_ST* have been maintained constant at their nominal value in all the following simulations).

An example of the adopted inputs is shown in Figures 18, 19 and 20.

2.3.1 Simulation at 100% of GT_Load

An example of the simulations performed to produce the data subsequently used for the identification phase is now described. The inputs are those described in the Figures 18, 19 and 20 and the drum level is controlled with a standard PI regulator. The transients of the main output variables are shown in Figures 21, 22, 23, 24, 25, 26.

Point of Work	input	GT_Load	Valve_ST	Feed_DSH $[kg/s]$
100%	stable	1	1	0
	variation	0.95	0.95	3,209885
75%	stable	0.75	1	0
	variation	0.7	0.95	3,209885
65%	stable	0.65	1	0
	variation	0.6	0.95	3,209885
60%	stable	0.6	1	0
	variation	0.55	0.95	3,209885
57%	stable	0.57	1	0
	variation	0.52	0.95	3,209885
50%	stable	0.5	1	0
	variation	0.55	0.95	3,209885
40%	stable	0.4	0,872274	0
	variation	0.35	0,822274	3,209885
25%	stable	0.25	0,872274	0
	variation	0.2	0,822274	3,209885
15%	stable	0.15	0,286485	0
	variation	0.1	0,236485	0,9

Table 10: Inputs for identification.

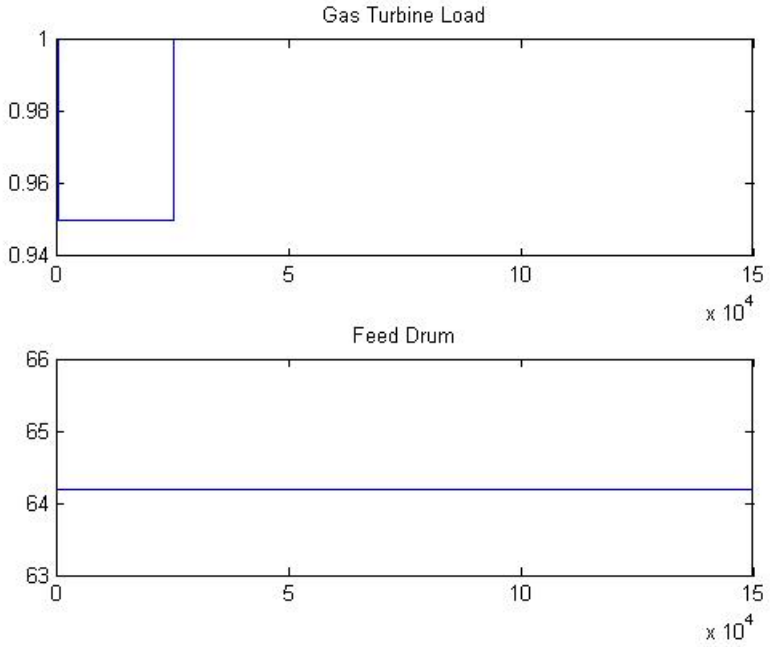


Figure 18: Inputs: example - 1

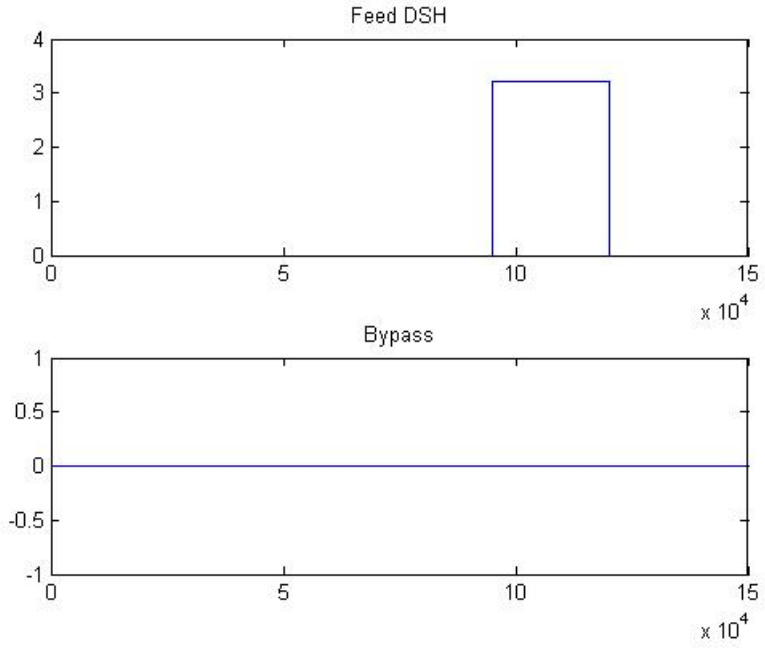


Figure 19: Inputs: example - 2

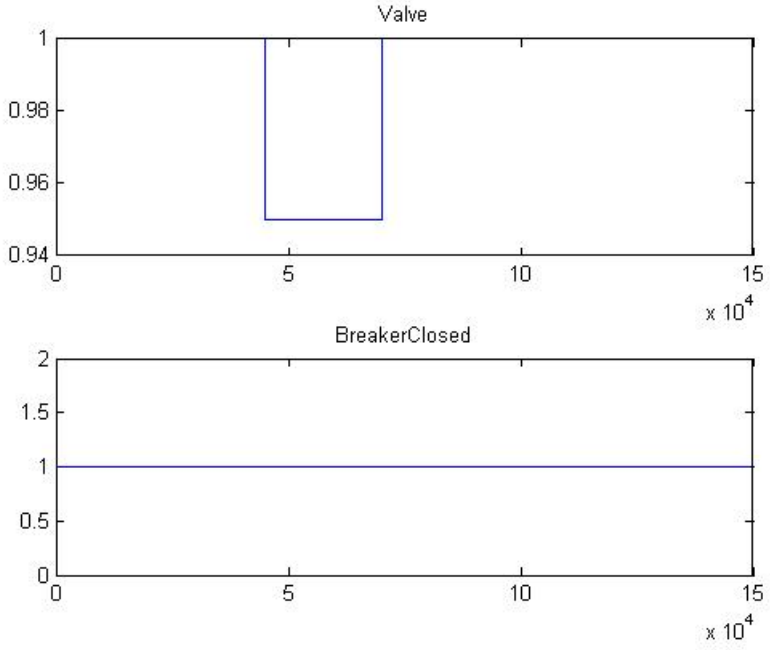


Figure 20: Inputs: example - 3

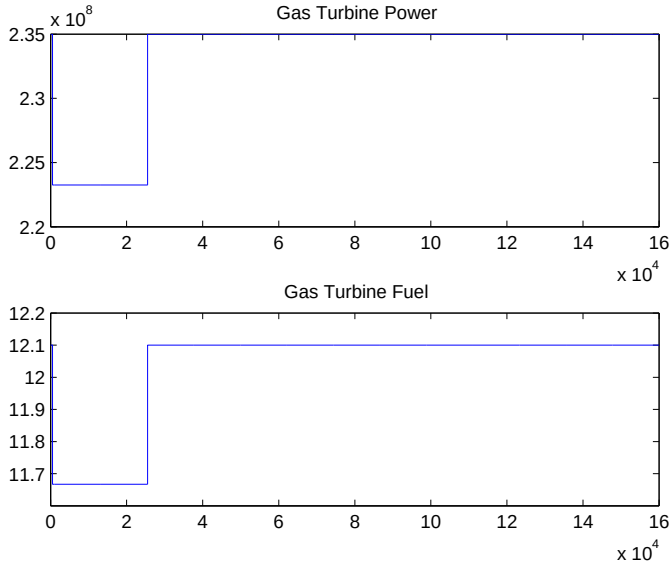


Figure 21: Gas Turbine Power and Gas Turbine Fuel at 100% of *GT_Load*.

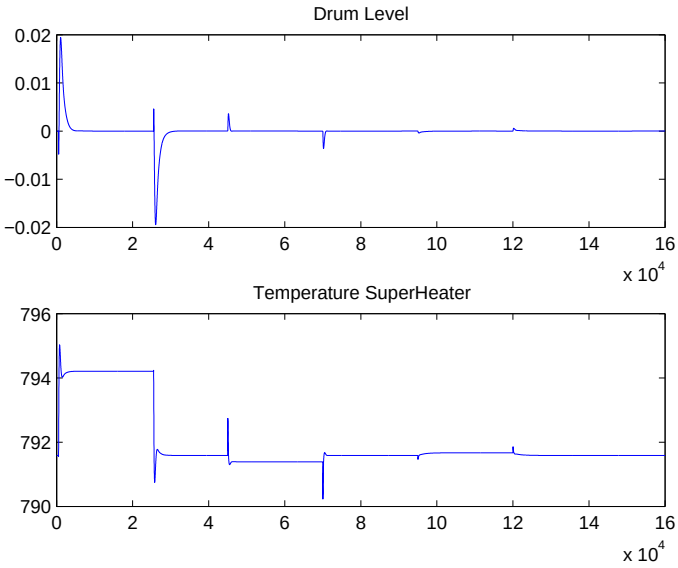


Figure 22: Drum Level and Temperature of SuperHeater at 100% of *GT_Load*.

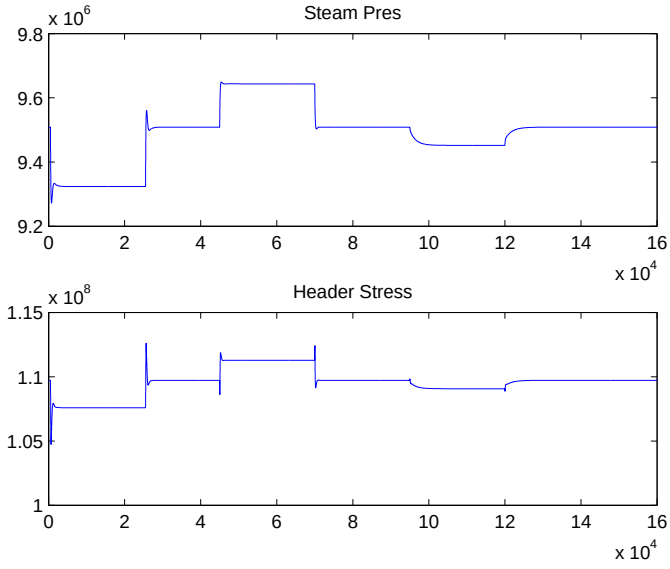


Figure 23: Steam Pressure and Header Stress at 100% of *GT_Load*.

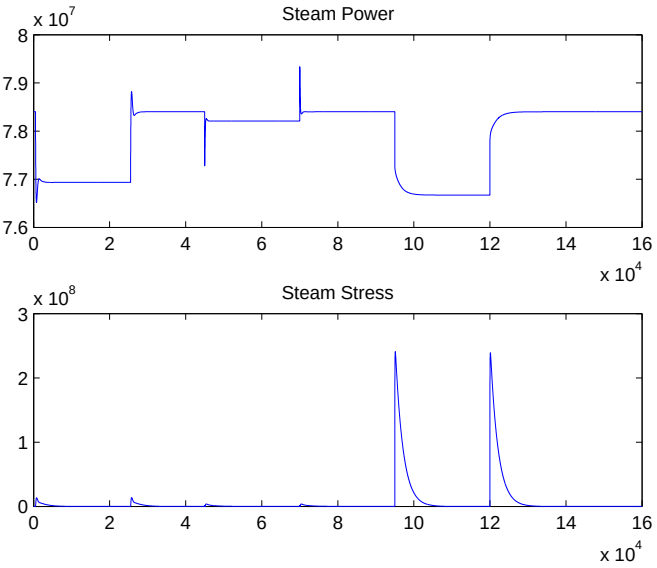


Figure 24: Steam Power and Steam Stress at 100% of *GT_Load*.

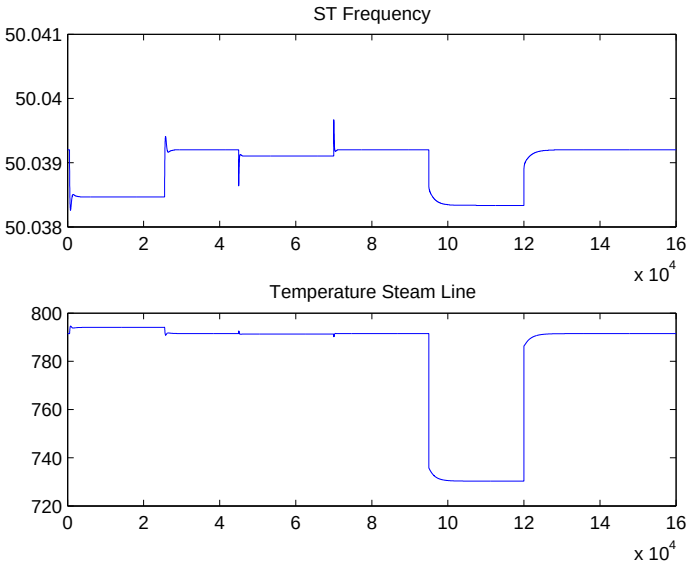


Figure 25: Steam Generator Frequence and Temperature of Steam Line at 100% of *GT_Load*.

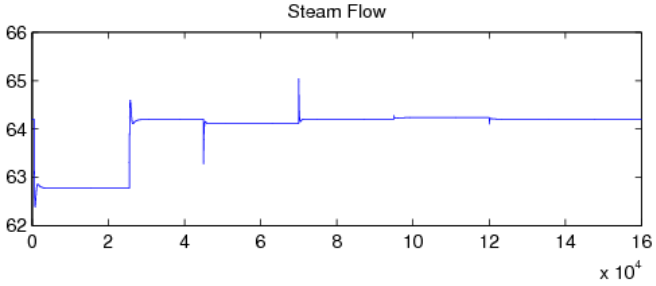


Figure 26: Steam Flow at 100% of *GT_Load*.

2.4 Identification with the *Matlab Identification Toolbox*

Due to the length of the simulations performed, it is not easy to interpret the corresponding results and to obtain useful information for the identification of the local linear models. As a matter of fact, being the input variations step changes, a preliminary visual inspection of the output variables can give a quite precise idea of the order and structure of the linear models to be estimated and of the main time constants. In order to extract these information from the available data, and to determine the sampling time to be used in the identification of discrete-time models, a simple Matlab GUI interface has been developed.
The empty GUI is shown in Figure 27: it gives the possibility to load the input and output variables of interest and to plot the transient response of the output in the first six operational points. The use of the GUI is very simple: first the data must be loaded with the *Load* Button (the data of all the simulations are gathered in *all_load.mat* file), then the input and output variables must be chosen with two drop down menus. On the left part of the screen the steady-state values are also reported for each load. Finally, in order to select the transient of interest, two buttons allow one to specify if he/she is interested in positive or negative step variations. Finally, a *Save* button is available to save the data. In Figure 28, an example of the use of the developed GUI is reported.

The identification of the transfer functions of the local linear models (see Chapter 2.1) has been performed with the *Matlab Identification Toolbox*, and in particular with the *Matlab* GUI *ident*. This toolbox allows one to select among different model structures, such as continuous-time or discrete-time, ARX, $Output - Error$ models, and to define the order of the transfer functions to be identified.
The main criterion adopted in this phase has been to estimate systems of low

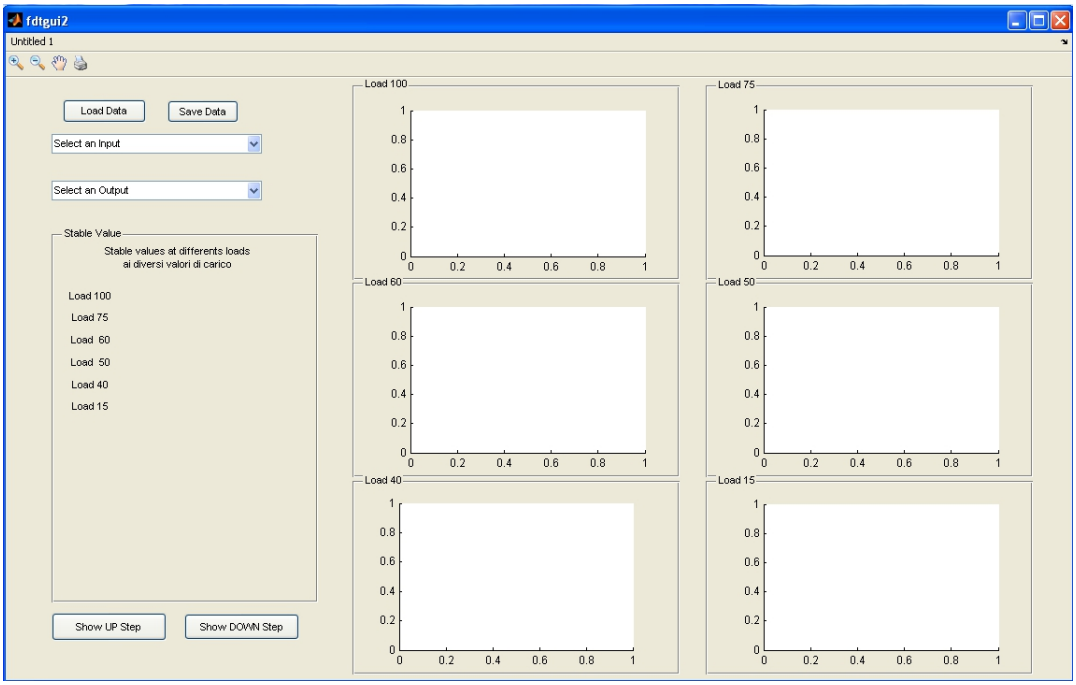


Figure 27: The developed Matlab GUI.

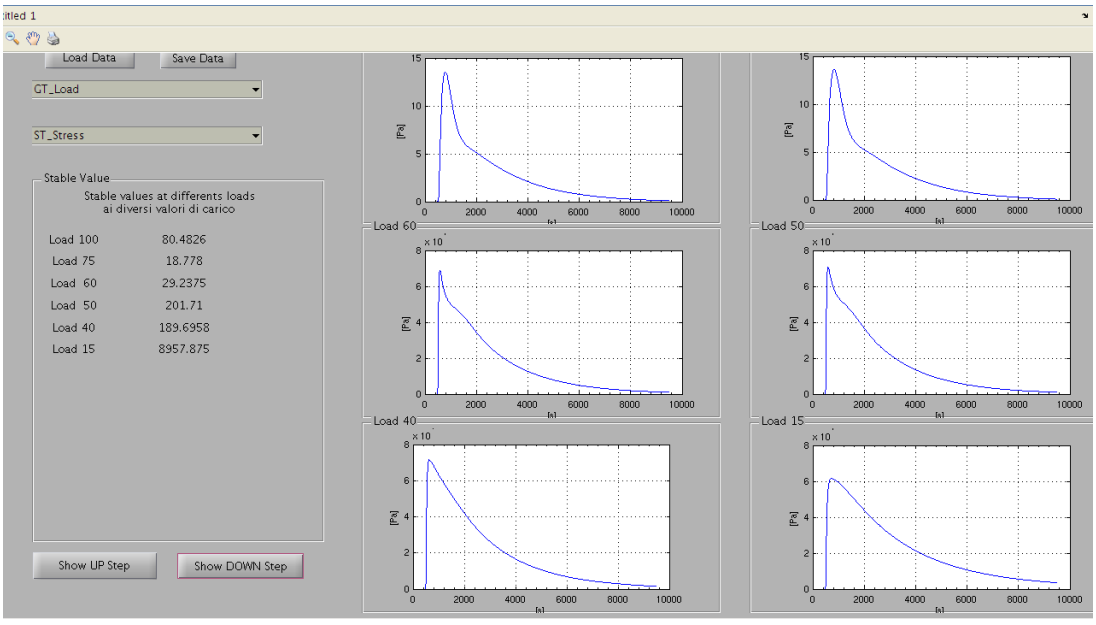


Figure 28: An example of the Matlab GUI.

order (first and second), mainly focusing on discrete-time models and *ARX* and *Output Error* structures. Indeed, the results achieved could be refined, but they can already be considered as fully satisfactory for the goals of the overall project.
An example of the results obtained with the identification procedure is shown in Figures 29 - 34. In particular, Figure 29 shows the GUI interface with a number of identified transfer functions between the *GT_Load* and the Temperature of SuperHeater (*TSH*). In these figures the x-axis is the time [s] and the y-axis the temperature [K]. It is important to note that all the values are normalized, to obtain better results. The same transfer function is considered in the Figures 30- 34 which compare the transients corresponding to the simulation data with those provided by the estimated models. In all the figures the black lines represent the data, while the other lines the identified models.

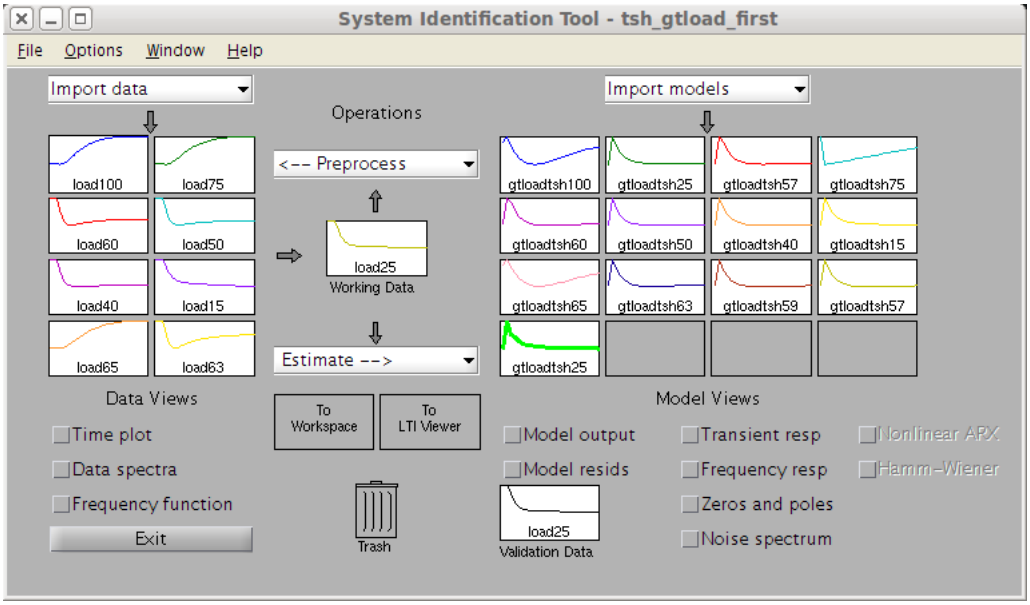
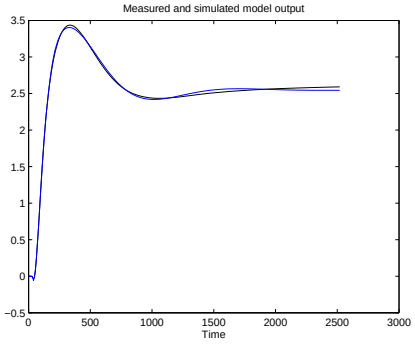
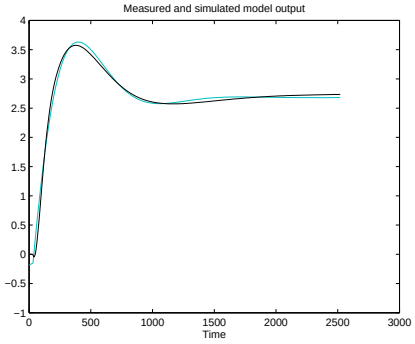


Figure 29: Identification GUI for GT_Load - TSH.

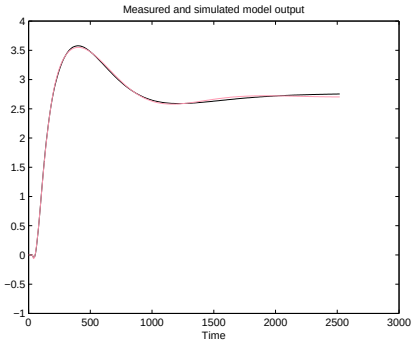


(a) TSH (K) at 100% of *GT_Load* (time in second)

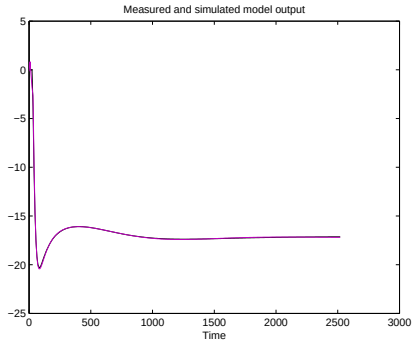


(b) TSH (K) at 75% of *GT_Load* (time in second)

Figure 30: Identification example - 1.

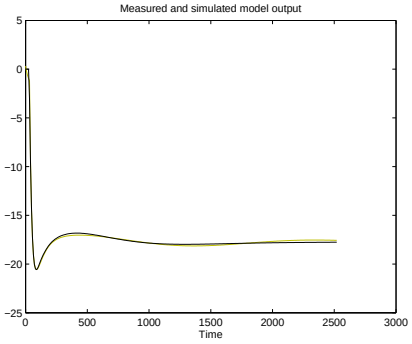


(a) TSH (K) at 65% of *GT_Load* (time in second)

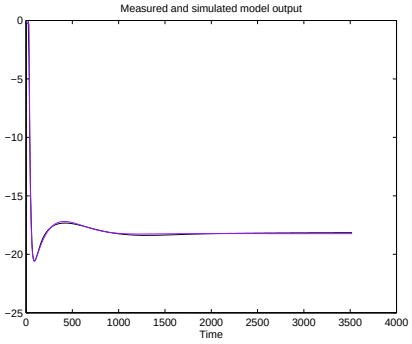


(b) TSH (K) at 60% of *GT_Load* (time in second)

Figure 31: Identification example - 2.

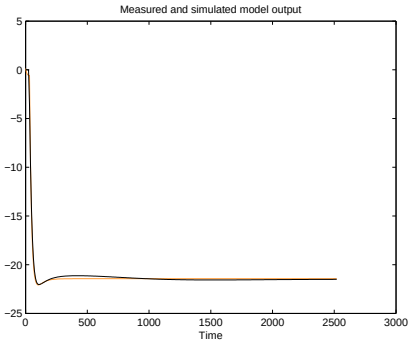


(a) TSH (K) at 57% of *GT_Load* (time in second)

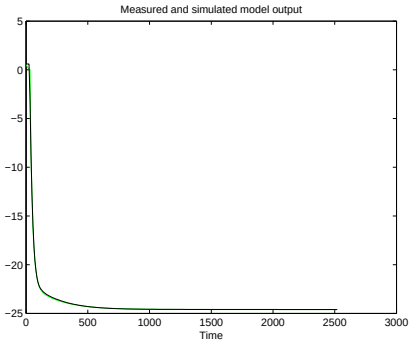


(b) TSH (K) at 50% of *GT_Load* (time in second)

Figure 32: Identification example - 3.

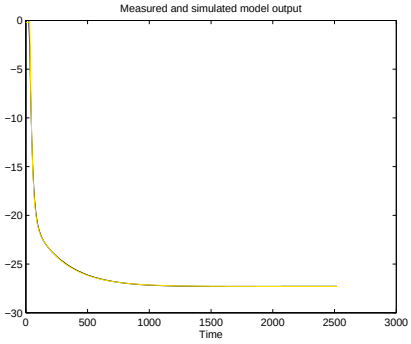


(a) TSH (K) at 40% of *GT_Load* (time in second)



(b) TSH (K) at 25% of *GT_Load* (time in second)

Figure 33: Identification example - 4.



(a) TSH (K) at 15% of GT_Load (time in second)

Figure 34: Identification example - 5.

2.5 Building the interpolation scheme with *Matlab Simulink*

Once all the transfer function describing the input-output relations at the considered operating points have been estimated, the adopted approach requires to build the overall model, including the membership functions. This is simple in the Simulink environment by means of the *Fuzzy Systems* toolbox, where a specific block for the definition of the membership functions is available. Then, a Simulink model can easily be built for any output variable. As an example, Figure 35 shows the Simulink model for the temperature TSH: it has the three system inputs, the main subsystem block and the output. The main block is expanded in Figure 36, where it is possible to see the blocks corresponding to the local linear models at any load, the block computing the membership functions and the connections among them. The weights (outputs of the membership functions) allow one to compute the final output as the interpolation of the contributions of the local models. The block computing the membership functions is expanded in Figure 37, while the local model at the 100% of the load is shown in Figure 38), which contains the identified transfer functions (see Figure 39). Note in particular that in the implementation of the transfer functions it is possible to use the blocks *ldmodel*, which allow one to use the models identified with the *ident* toolbox without any further elaboration.

The procedure now sketched to obtain the model of the variable TSH can be repeated for each output variable. In this Thesis, the models of four variables have been identified:

- TSH: temperature of superheater;

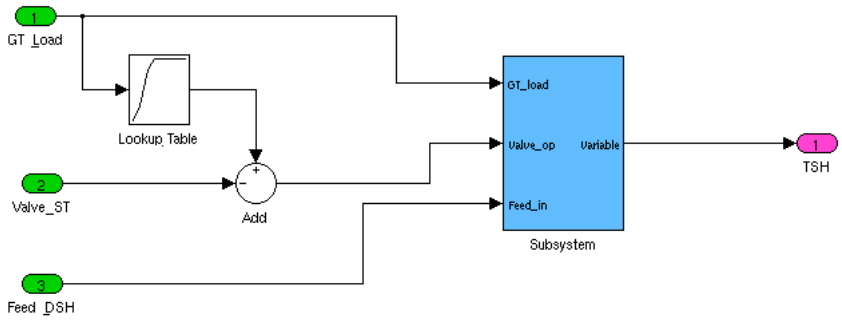


Figure 35: Main *Simulink* scheme for TSH.

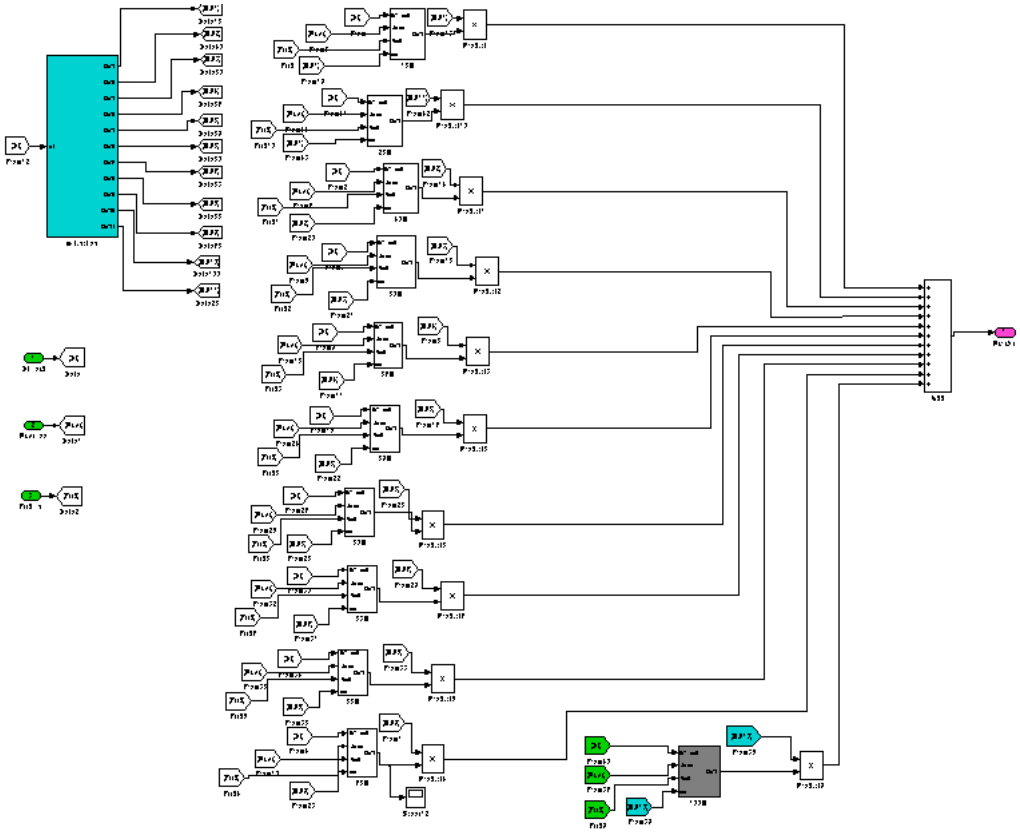


Figure 36: Subsystem *Simulink* scheme for TSH.

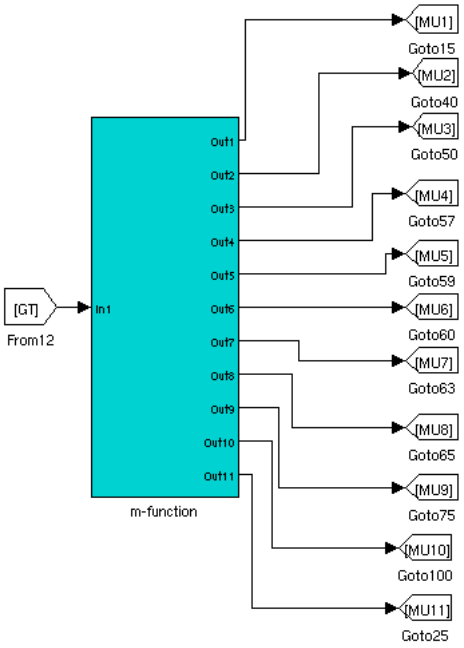


Figure 37: Membership functions *Simulink* scheme.

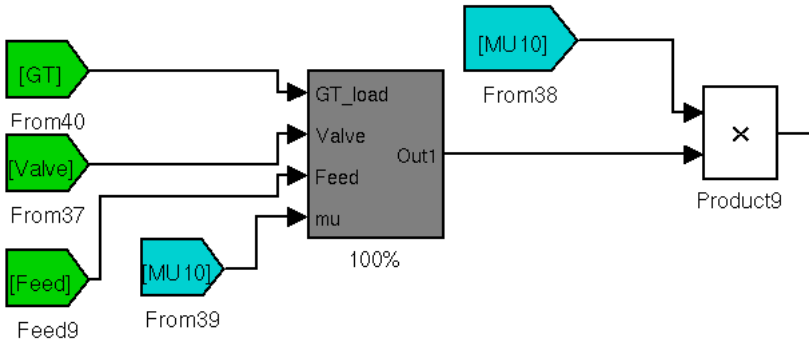


Figure 38: Local *Simulink* subsystem for TSH at 100% of GT_Load.

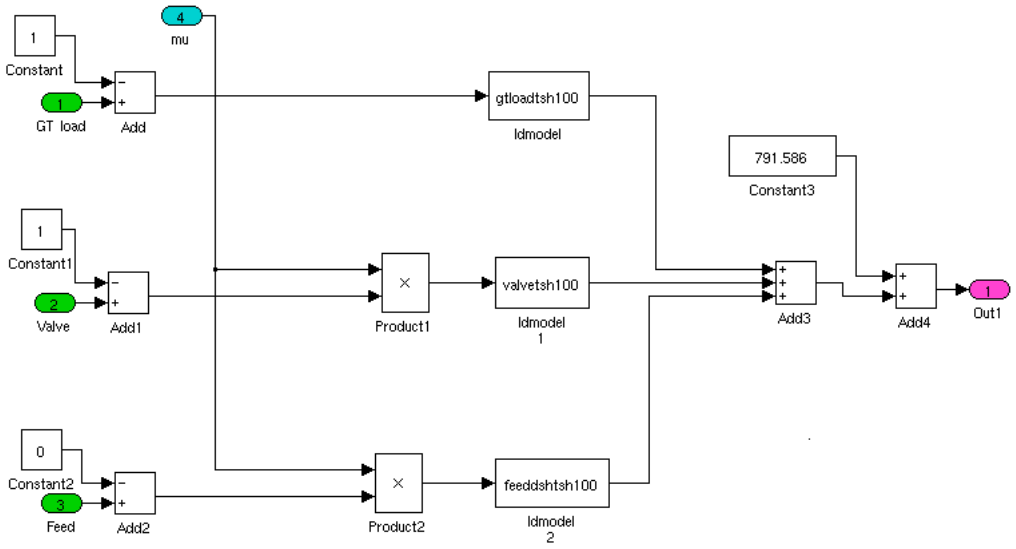


Figure 39: Detail of *Simulink* scheme to build the local model.

- ST_Press: steam pressure;
- $Header\_Stress$: stress of the header;
- ST_Stress: st*ress inside the steam turbine.

A possible future development can concern the modeling of all the plant output variables, paying attention to the fact that the power produced by the gas turbine and the gas turbine fuel have a very fast dynamics with respect to the other outputs. Then, these variables can be modeled with simple static relations.
The final *Simulink* scheme for the identified variables is shown in Figure 40.

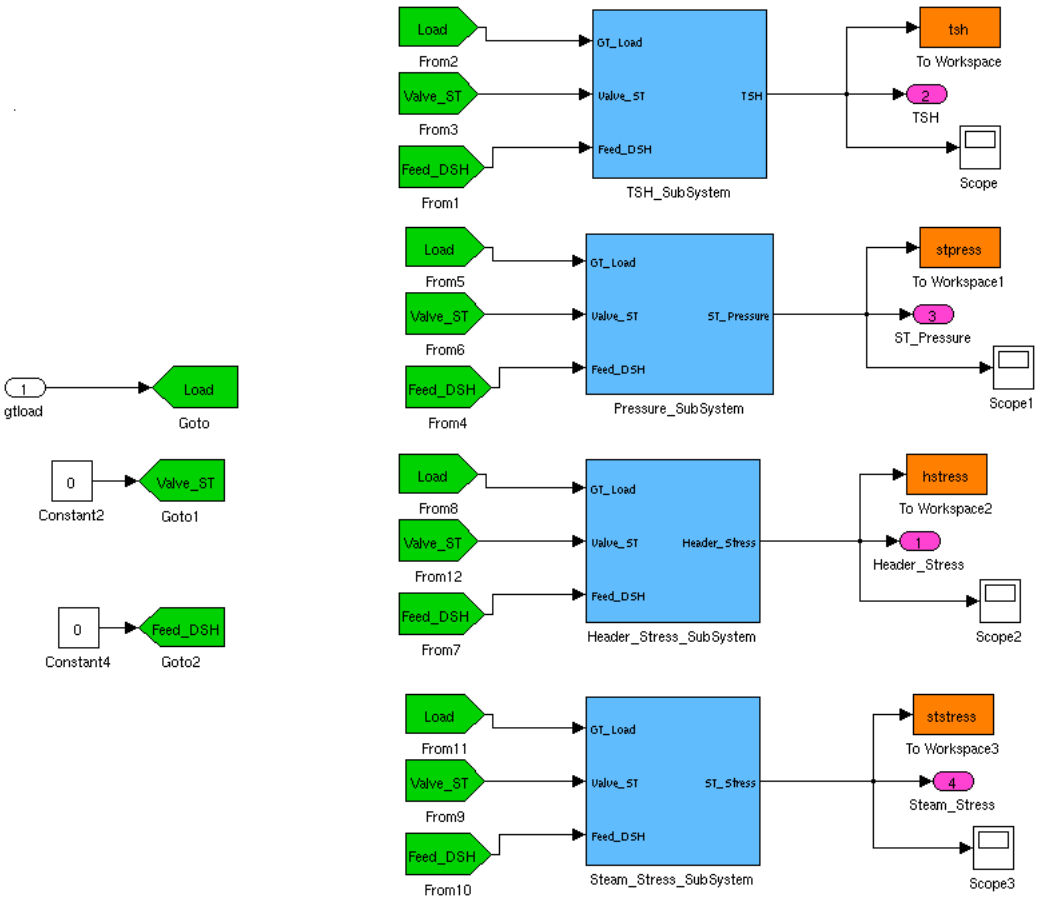


Figure 40: *Simulink* scheme for all identified variables.

3 Chapter 3: Validation of the identified models

The validation of the identified model developed in Matlab/Simulink has been performed by comparing its behavour with the one of the original Modelica model used to generate the data. Different sets of data have been considered to compare the two models both locally and with respect to large variations of the *GT_Load* over the whole operating range of the CCPP. In this second case, positive and negative ramp variations, from the 15% of the load to full load and viceversa, have been imposed.
In the following Figures 41, 42, 43, 44, 45 and 46 the transients of the steam pressure due to small variations of the *GT_Load* starting from different steady state operating conditions are reported. In these figures, and in all the figures of the chapter, the blue line shows the transients computed with Dymola, while the green line those produced by the Simulink model.

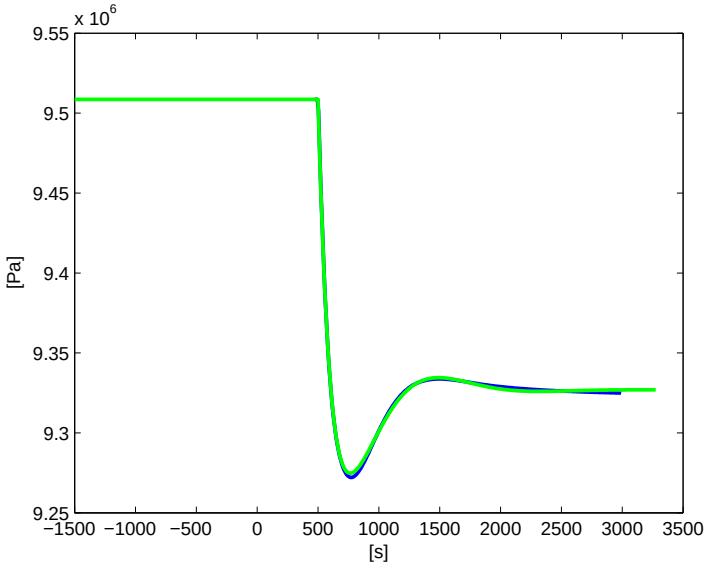


Figure 41: *GT_Load* - *ST_Pressure* transient at 100%.

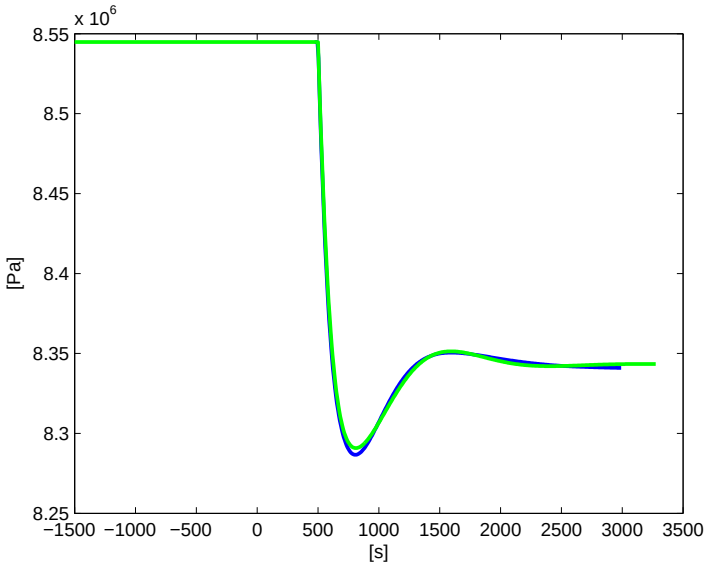


Figure 42: *GT_Load - ST_Pressure* transient at 75%.

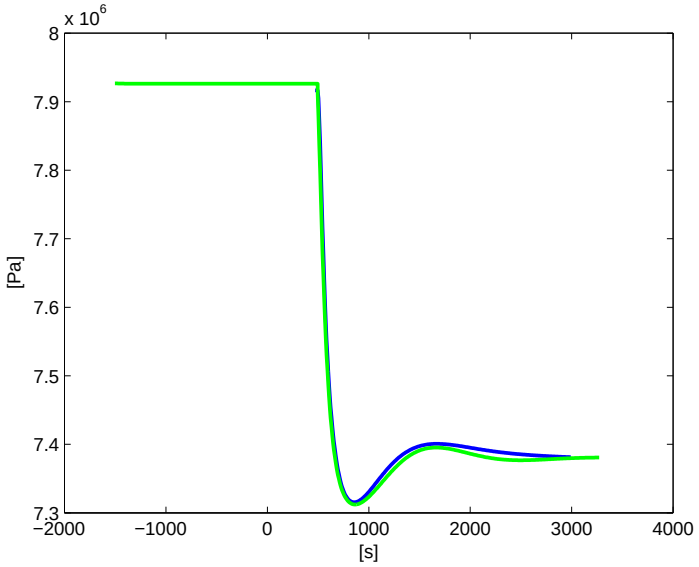


Figure 43: *GT_Load - ST_Pressure* transient at 60%.

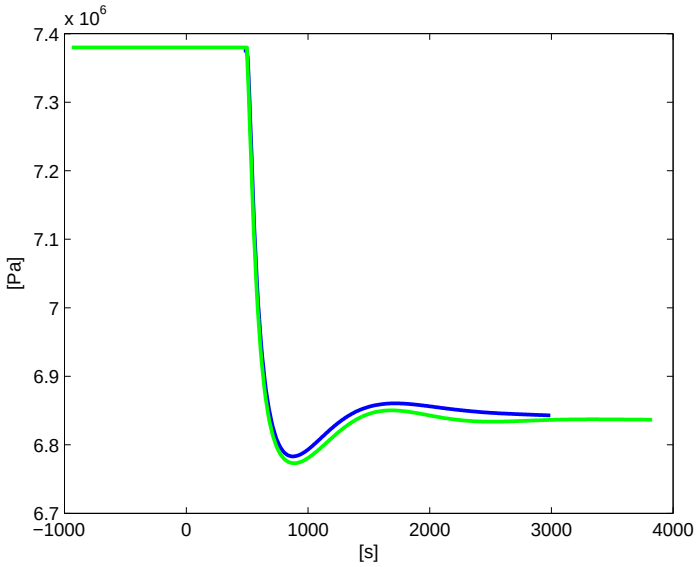


Figure 44: *GT_Load* - *ST_Pressure* transient at 50%.

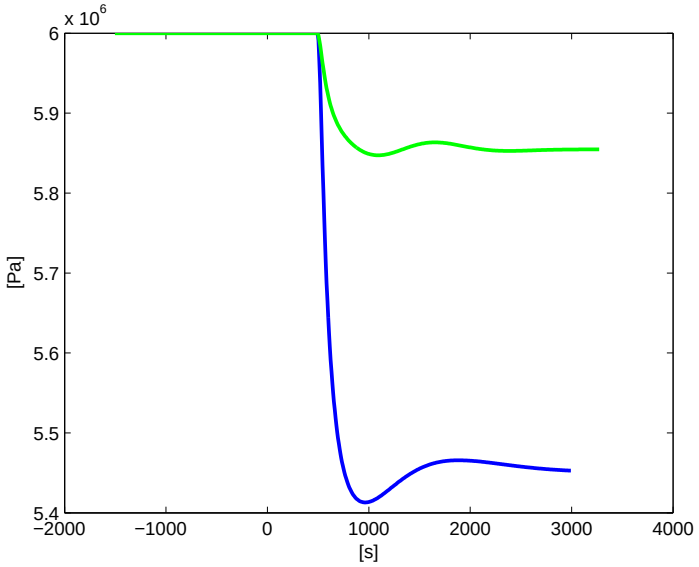


Figure 45: *GT_Load* - *ST_Pressure* transient at 40%.

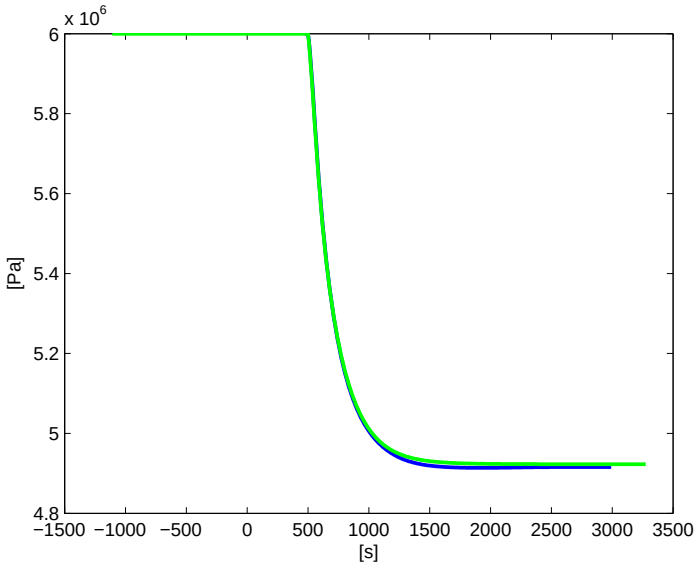


Figure 46: *GT_Load - ST_Pressure* transient at 15%.

A short comment must be done for the figure which represent the variation of the load from 40% to 35%: is possible to see a gap from Simulink and Dymola response. This because the dynamic of the valve is modelized only with a look-up table in function of the load, which approx the behavior of the valve. To have a better model the only way is to improve this function, with more point. By the way is not much important, because at this point of work the pressure controller is online. The transients produced by the opening of the admission valve in the two models are reported in Figures 47, 48, 49, 50, 51 and 52.

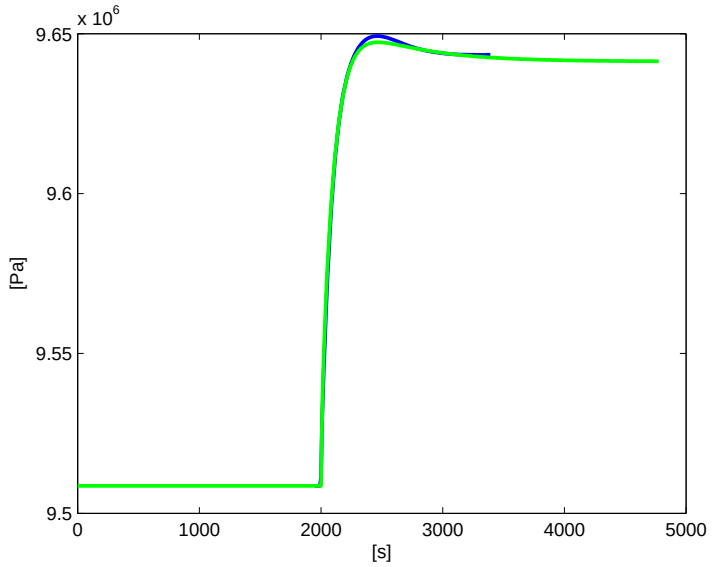


Figure 47: $Valve\_ST$ - $ST\_Pressure$ transient at 100%.

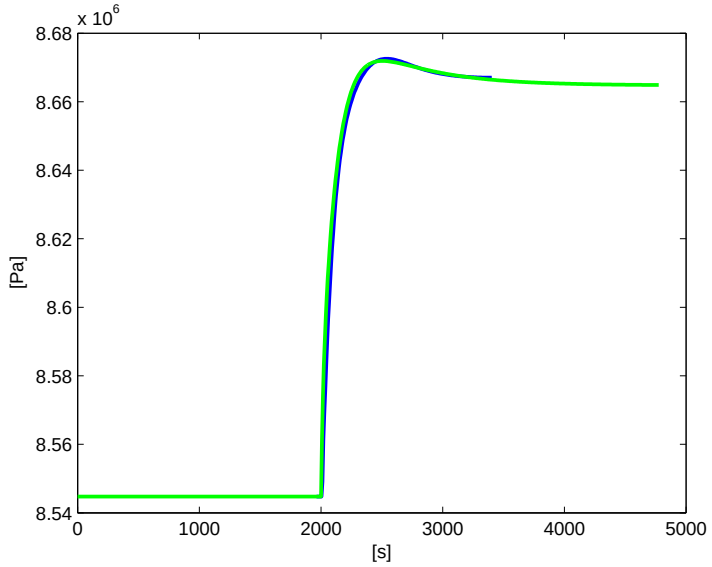


Figure 48: $Valve\_ST$ - $ST\_Pressure$ transient at 75%.

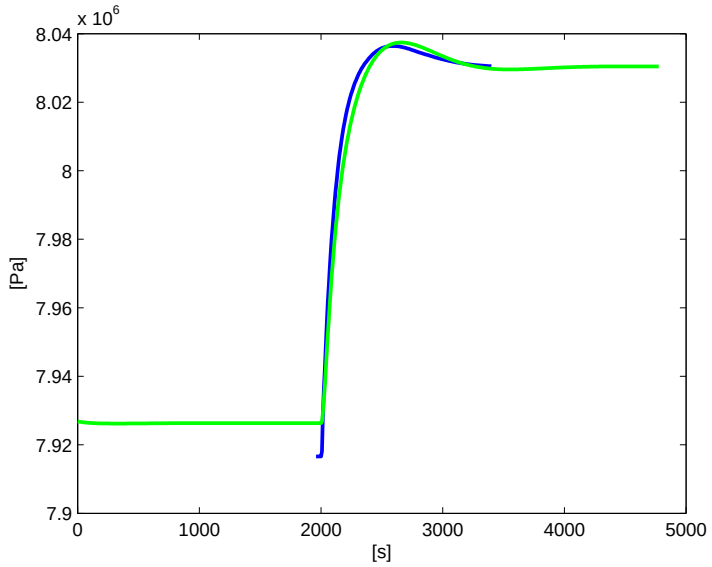


Figure 49: $Valve\_ST$ - $ST\_Pressure$ transient at 60%.

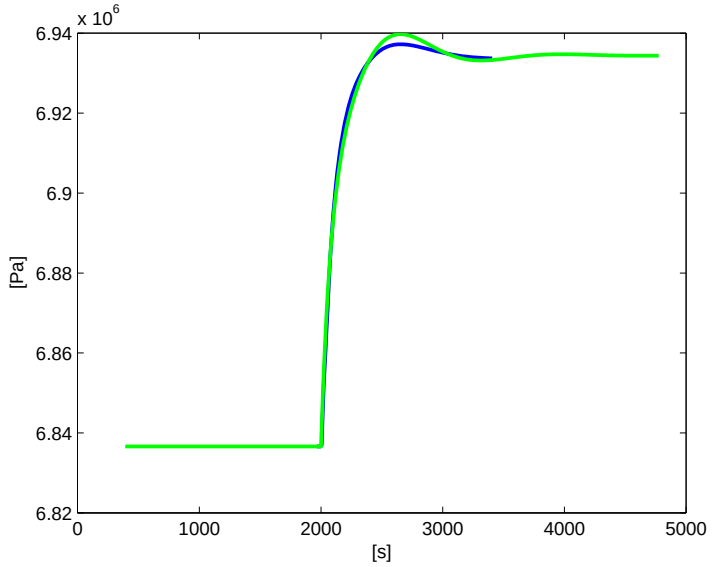


Figure 50: $Valve\_ST$ - $ST\_Pressure$ transient at 50%.

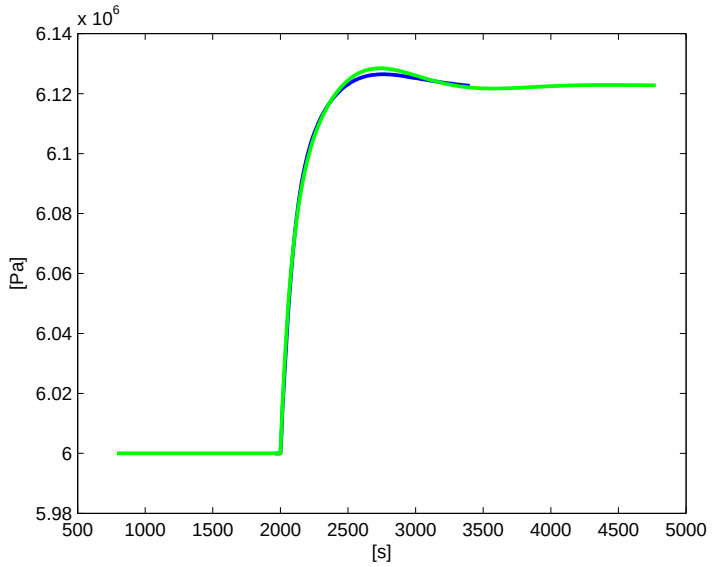


Figure 51: $Valve\_ST$ - $ST_Pressure$e transient at 40%.

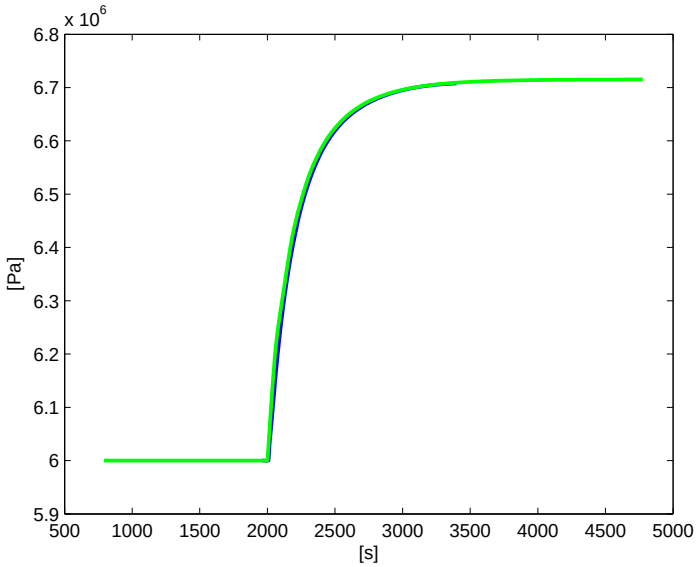


Figure 52: $Valve\_ST$ - $ST\_Pressure$ transient at 15%.

The small difference (about 0.2 bar) in Figure 49 between the responses in Dymola and Simulink is due to the interpolation between local linear models in Simulink: near the regime corresponding to the 60% of the load, many local models have been used to properly reproduce the change of sign of the

gain in some transfer functions; correspondingly, the membership functions are very dense. However, the difference between the two models is quite small and fully acceptable for the scopes of this work.
The Figures 53, 54, 55, 56, 57 and 58 show the transient between the Feed_DSH and the steam pressure.

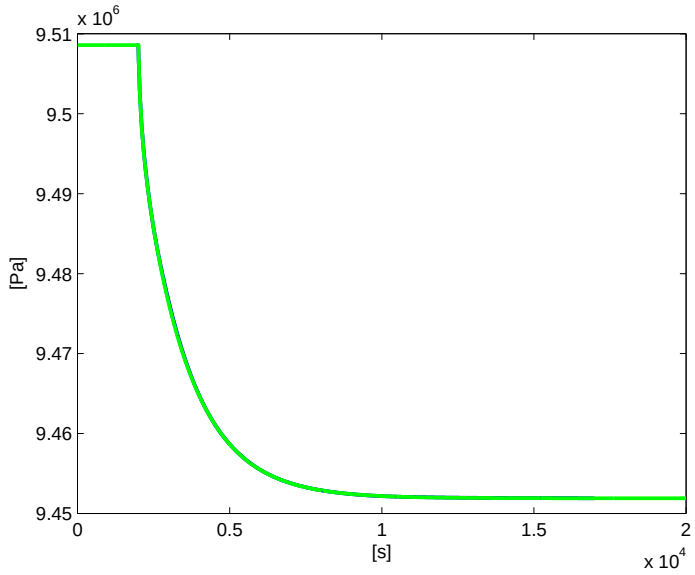


Figure 53: *Feed_DSH - ST_Pressure* transient at 100%.

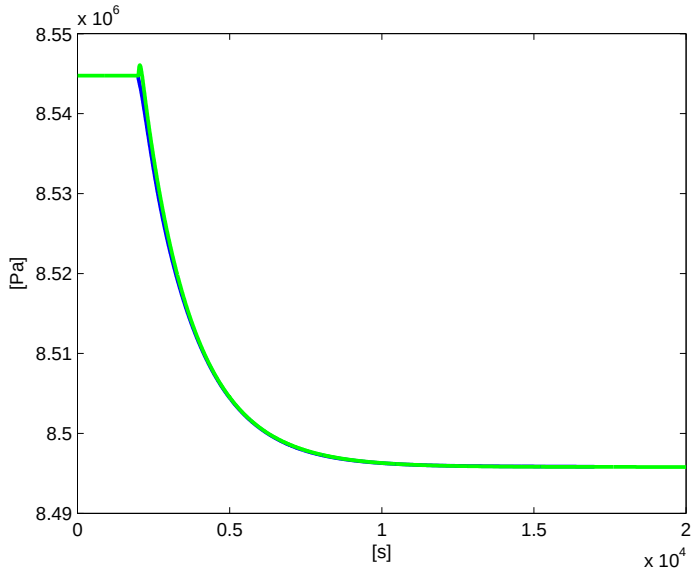


Figure 54: *Feed_DSH - ST_Pressure* transient at 75%.

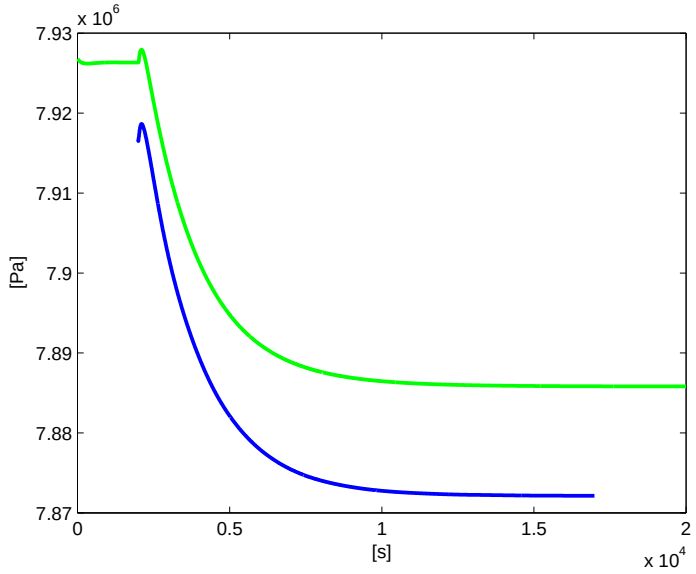


Figure 55: *Feed_DSH - ST_Pressure* transient at 60%.

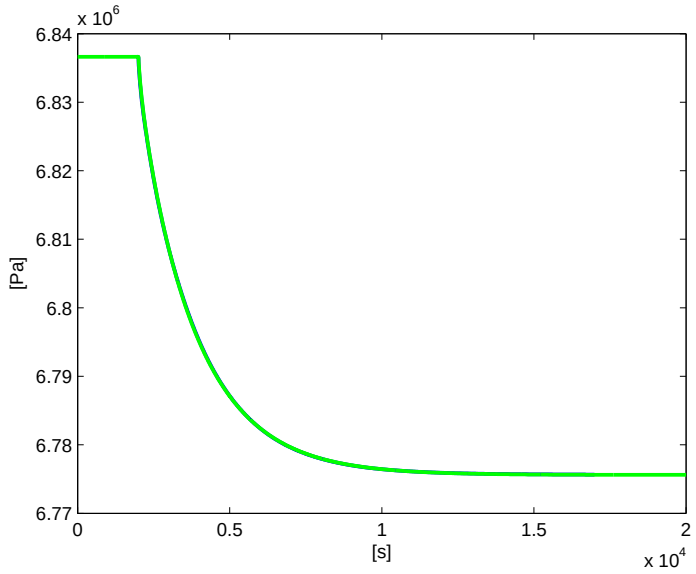


Figure 56: *Feed_DSH - ST_Pressure* transient at 50%.

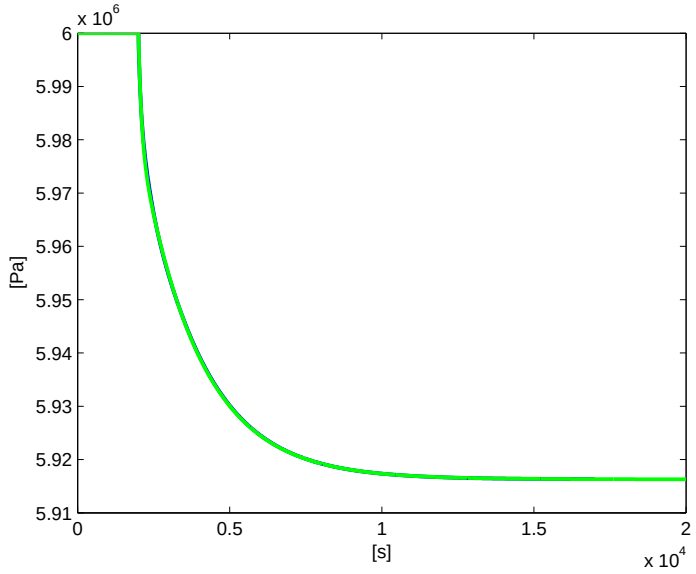


Figure 57: *Feed_DSH - ST_Pressure* transient at 40%.

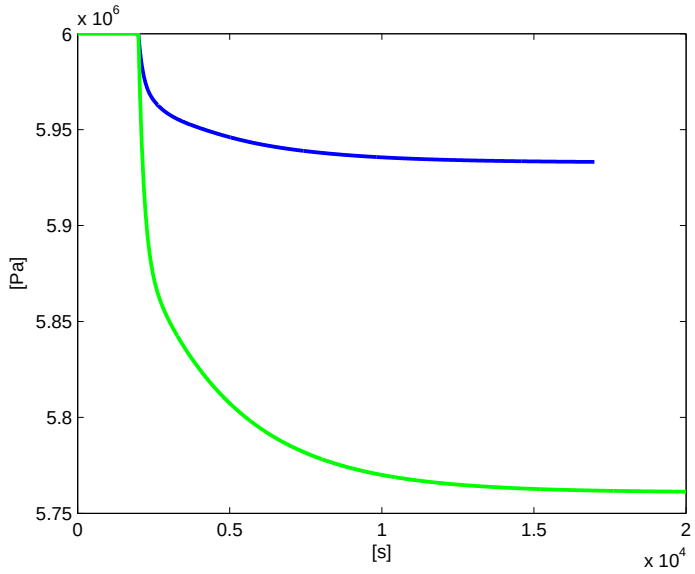


Figure 58: *Feed_DSH - ST_Pressure* transient at 15%.

Also in this case, it is possible to note a small gap between the Dymola and the Simulink responses at 60% of the load, due to the same causes previously described. On the contrary, Figure 58 shows that the two models have a significantly different behavior at low load. However, this difference (about 2 bar) is not critical, in fact like at 40% of load, in these conditions a pressure controller is normally online.

The final results are referred to the large ramp variations corresponding to the shutdown (Figure 59) and to the start-up (Figure 60) phases. Both are encompassed between 100% and 15% of *GT_Load*. The transients of the four outputs analyzed are shown in Figures 61, 62, 63 and 64 for the negative ramp, and in Figures 65, 66, 67, 68 for the positive one.

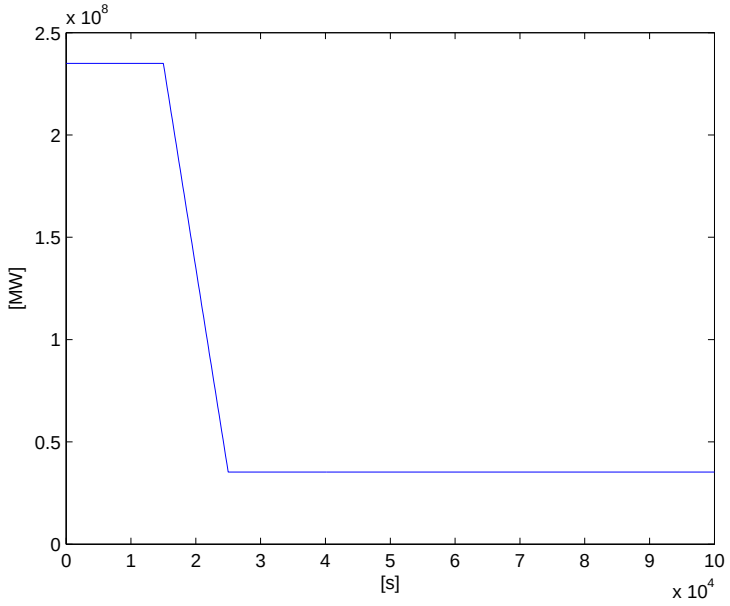


Figure 59: Shutdown ramp.

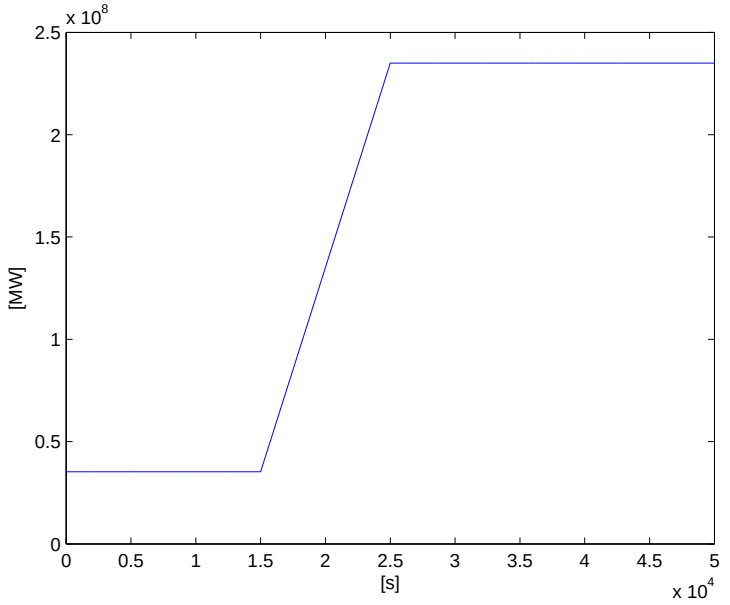


Figure 60: Start-up ramp.

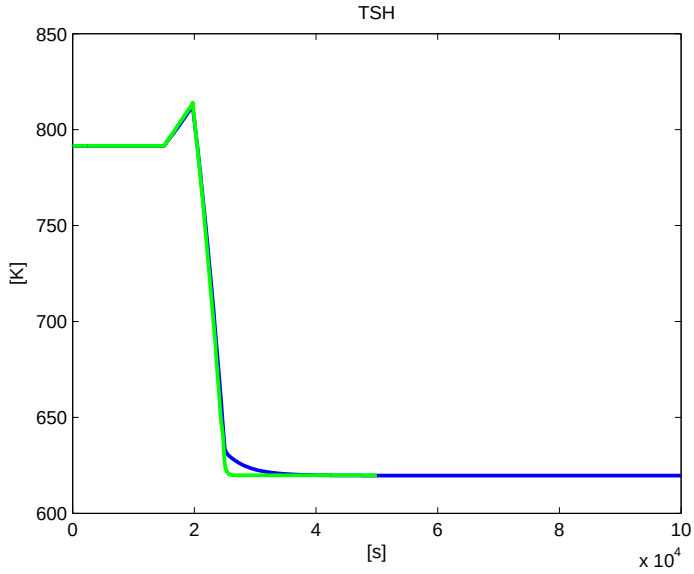


Figure 61: Shutdown temperature of superheater.

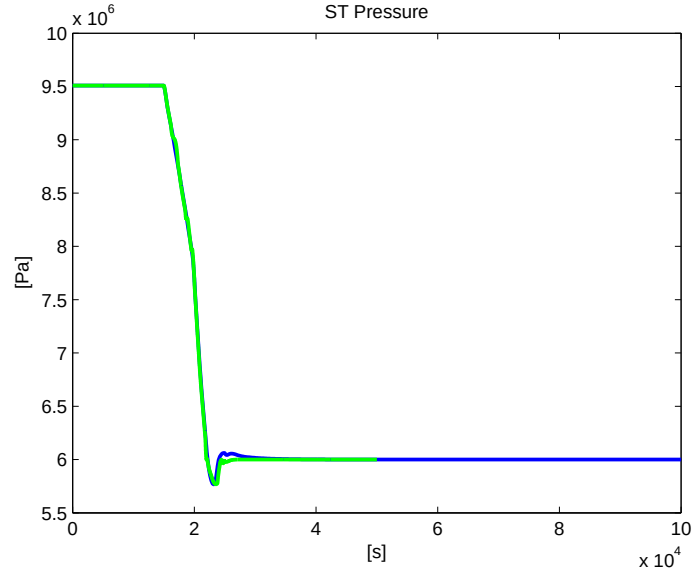


Figure 62: Shutdown steam pressure.

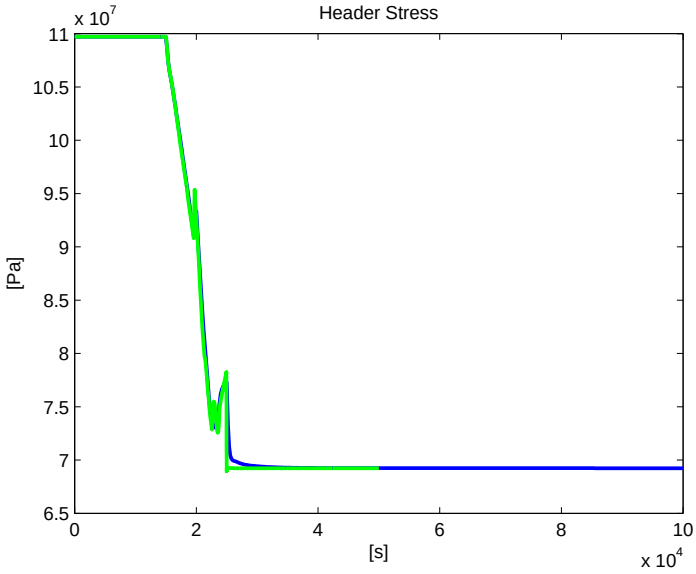


Figure 63: Shutdown header stress.

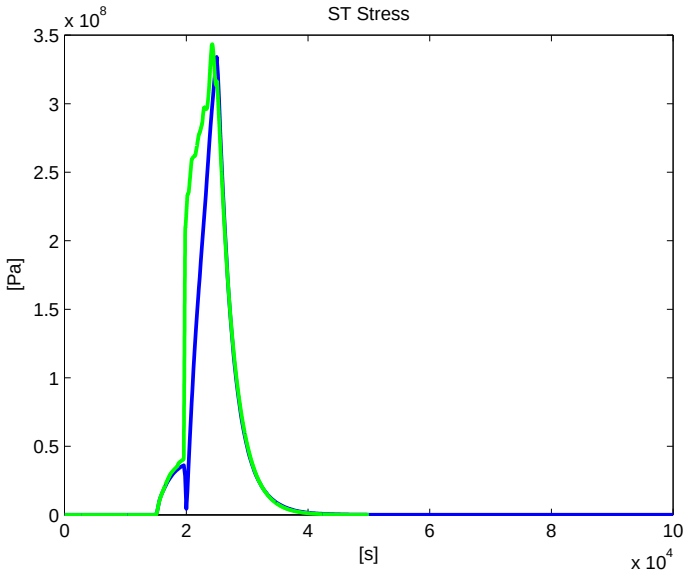


Figure 64: Shutdown steam turbine stress.

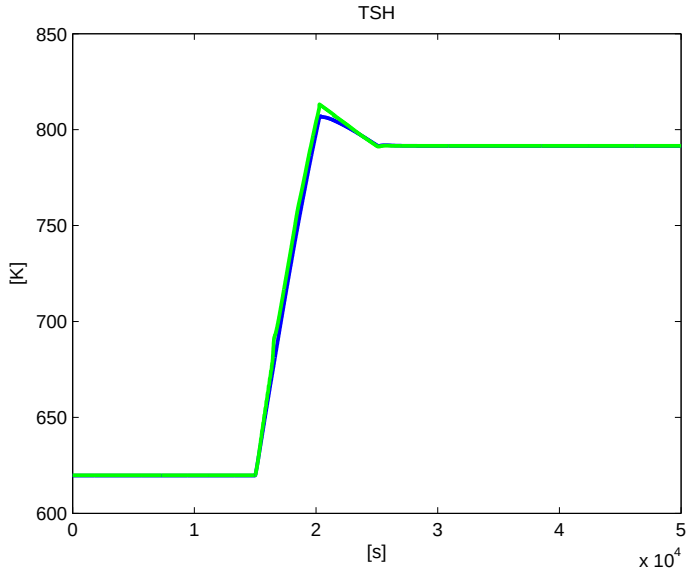


Figure 65: Start-up temperature of superheater.

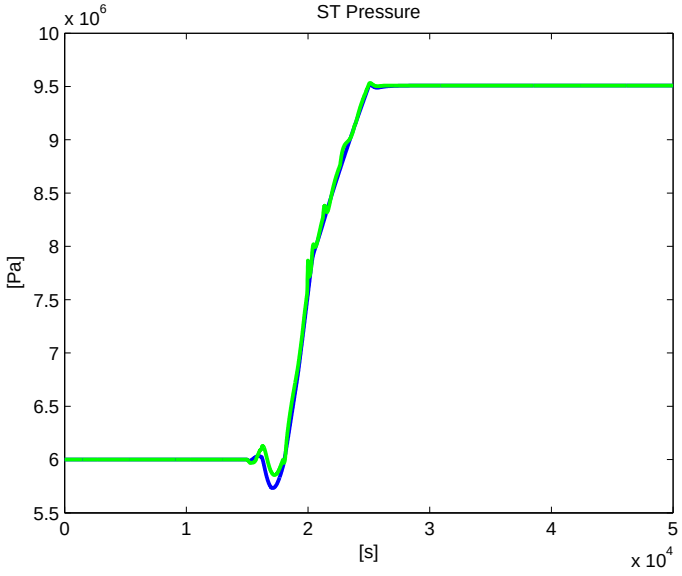


Figure 66: Start-up steam pressure.

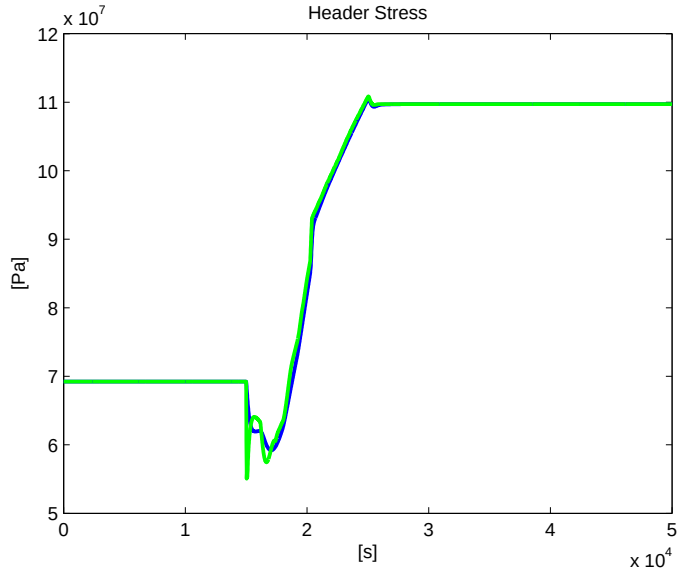


Figure 67: Start-up header stress.

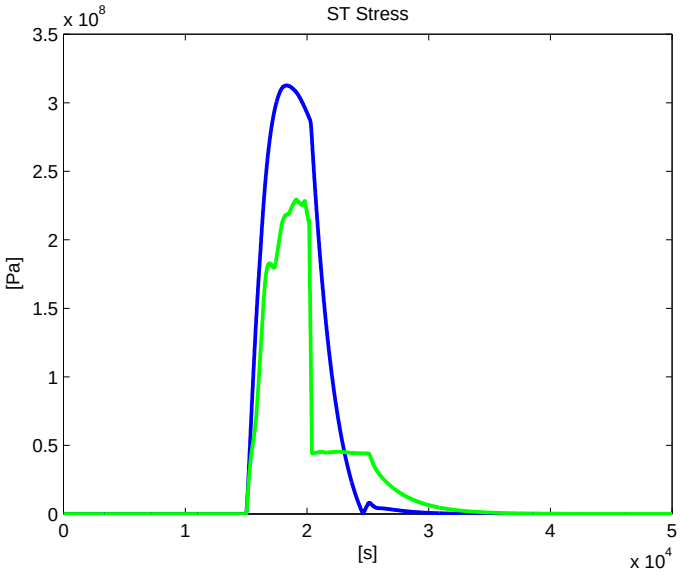


Figure 68: Start-up steam turbine stress.

It is possible to note that the responses to the two load variations are different. In the shutdown case, the Simulink responses fit very well the Dymola data. This is due to the fact that these data have been used for the definition of the membership functions. Therefore, the comparison of the

shutdown transients cannot be properly considered as a validation test. On the contrary, the analysis of the start-up experiments is made on data not used to identify the interpolated model. As expected (recall that the Dymola model is highly nonlinear), the fitting is slightly worse than the previous one. The worst response is shown in Figure 68, where the Simulink model is not able to properly reproduce the peak stress.

4 Chapter 4: Optimization

This chapter describes the optimization procedure developed to determine the *GT_Load* profile minimizing the start-up time while fulfilling a number of constraints on the main process variables. More specifically, it is required that during the start-up phase temperatures, pressures and stresses remain into prescribed limits for safety and economical reasons. The approach is based on the definition of a minimum-time control problem subject to constraints. Its solution is obtained by means of the joint use of the Simulink simulator previously developed and of the Matlab Optimization Toolbox.

4.1 Minimum-time optimization problem

The model of the Combined Cycle Power Plant can be given the general form of a set of continuous-time differential equations:

$$\dot{x} = f(x(t), L(t))$$

where x is the state vector, which contains the plant variables, and L represents the gas turbine load.
Denote by t_0 the initial time instant of the start-up procedure (conventionally it can be set $t_0 = 0$) and by t_f the final one. As for the load profile, it must be selected so that $L(t_0) = L_m$ and $L(t_f) = L_M$, L_m and L_M being the initial and final (full) load.
Assume now that the load is described by an increasing function $L(t, q)$ satisfying the boundary conditions above stated and where q is a vector of unknown parameters, which have to be selected through an optimization procedure. Then, the problem of computing the optimal load profile consists of finding the value of the parameter vector q, together with the final time t_f, which solves the following optimization problem:

$$\min_{q,t_f} J = \int_{t_0}^{t_f} dt \tag{51}$$

subject to the constraints:

$$\dot{x} = f(x(t), L(t, q)) \tag{52}$$
$$L(t_f, q) \geq L_M - \varepsilon_1 \tag{53}$$
$$|f(x(t_f), L(t_f, q))| \leq \varepsilon_2 \tag{54}$$
$$h(x(t)) \leq 0, t_0 \leq t \leq t_f \tag{55}$$

where ε_1 and ε_2 are arbitrary small values (ideally equal to zero) and the corresponding constraints are included to guarantee that at time t_f the system is (almost) in stationary conditions and has reached (almost) full load. The last (vector) constraint includes, in general form, all the constraints to be imposed to the plant variables, in particular on the stresses, during the start-up procedure.
Many functions $L(t, q)$ can be chosen, such as sigmoid functions. In this work, the following Hill function has been chosen:

$$L(t, q) = L_m + (L_M - L_m)\frac{t^h}{t^h + k^h} \tag{56}$$

where $q = [h\ k]$ is the parameter vector to be determined through the optimization procedure. An example of Hill function for different values of its parameters is shown in Figure 69.

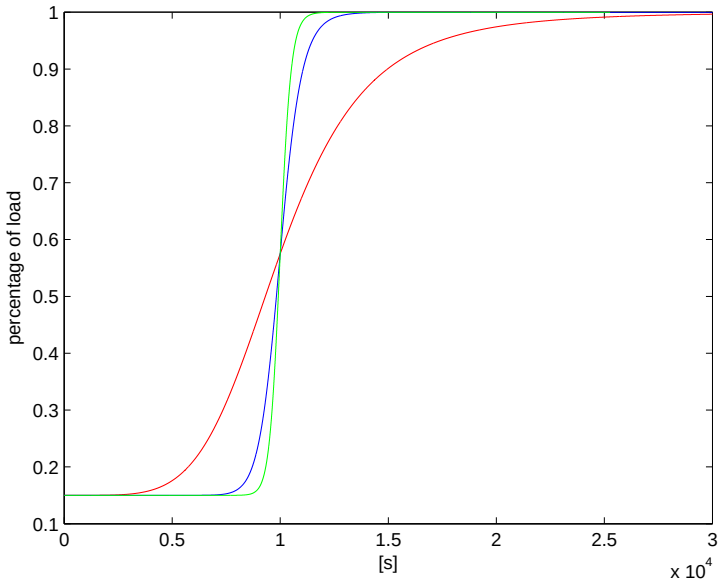


Figure 69: Hill function with L_m=0.15, L_M=1, t_0=0, t_f=30000, k=2 and h=5000 (blue), k=4 h=5000 (green), k=6 h=5000 (red)

4.2 Matlab Algorithm

For the solution of the minimum-time optimization problem, a program has been created in Matlab. This program is based on the minimization function *fmincon*, available in the Matlab Optimization Toolbox, which solves

nonlinear constrained minimization problems starting from an initial guess of the solution.

Fmincon has the following structure:

$$x = \text{fmincon}(@myfun, \text{x}_0, A, b, Aeq, beq, lb, ub, @mycon)$$

where $myfun$ is the function to be minimized, which accepts a vector x and returns a scalar f, the objective function evaluated at x. With the "@" symbols, $myfun$ is specified as a function defined through a file. A and Aeq are linear constraint matrices, b and beq their corresponding vectors. In our case, since the problem is non-linear, A, Aeq, b and beq are empty (in the Matlab syntax they are defined as empty matrices as follows: $A = [\,]$). The vectors lb and ub indicate, respectively, the lower and the upper bounds of the unknown elements of x, i.e. of the variables to be determined by the optimization algorithm. In the *mycon* function all nonlinear constraints are defined.

In order to solve the optimization problem for the CCPP, three files have been created:

1. *opt_hill*
2. *con_hill*
3. *fun_hill*

The *opt_hill* file is the main program which defines and manages the optimization problem. Its code is the following:

```
clear all
load total.mat
x0=[0.001 5 30000];
lb=[0 2.3 10800];
ub=[0.0051 10 100000];
A=[];
b=[];
Aeq=[];
beq=[];
 optimset('fmincon');
 opts=optimset('Display','Iter','TolFun',10^(-6),...
      'MaxFunEvals',1000,'MaxIter',10000,'TolX',10^(-20));
[K,fval,exitflag]= fmincon(@(K) fun_hill(K),x0,A,b,Aeq,
                           beq,lb,ub,@(K) con_hill(K));
```

This file fixes the initial values (for example, the initial guess for the final time of the start-up phase is set to 30000 seconds, see [12], [9]), and the upper and lower bounds for the parameters h and k of the Hill function. In call of $fmincon$, the name of the file computing the performance index to be minimized is specified (fun_hill), together with the name of the function computing the constraints (con_hill). With the *optimset* instruction some options for $fmincon$ are set.

The file fun_hill computes the cost function, i.e. the final time of the start-up:

```
function J = fun_hill(K)
J=K(3);
```

The file is composed only by one row of code because the cost function corresponds to one of the parameters to be determined.

The third file, *con_hill*, computes the constraints to be fulfilled in the optimization problem. Its code is:

```
function [C,Ceq] = con_hill1(K)
%% Hill function
K
lm=0.15;
lM=1;
tsim=(0:1:K(3));
for i=1:length(tsim)
    L(i)=lm+(lM-lm)*((tsim(i)^(K(1)*1000))/
        (tsim(i)^(K(1)*1000)+(K(2)*1000)^(K(1)*1000)));
end
tsim=(0:1:K(3))';
loadgt=L';
plot(tsim,L);

%% simulating model
[t1,X,Y]=sim('totalmodel',[tsim(1) tsim(end)],
                [],[tsim,loadgt]);

DERI=Y(end,:)-Y(end-1,:);
STRESSMAX_HEADER=max(Y(:,1))
STRESSMAX_STEAM=max(Y(:,4))

ERR=DERI*DERI'


Ceq=[];
C=ERR-1;
```

```
C=[C;STRESSMAX_HEADER-11e+7];
C=[C;STRESSMAX_STEAM-2.25e+8];
C=[C;-L(end)+lM-0.005]
```

In the first part, the file computes the Hill function, i.e. the GT load, corresponding the current values of the optimization variables, which will be the input for the Simulink model used to simulate the plant behavior. Then, the Simulink model is run and, based on the computed transients of the plant variables, the constraints on $Header\_Stress$ and $Steam\_Stress$ are evaluated, together with the constraints which require to reach almost the steady state conditions at the end of the simulation.
To obtain good results, many simulations are needed; a good idea is to start different simulations with different bounds and initial states, so as to explore all the feasibility space. The time for one simulation is about five minutes, but it strongly depends on the inputs. Another important aspect to consider is that $fmincon$ is not an efficient routine: sometimes problems occur and the tests crash. This typically happens when Matlab tries to move from a set of safe values. The weakness of $fmincon$ algorithm is balanced by its simplicity. Different solvers and optimization tools, such as Tomlab, should be tested to make the optimization phase more reliable and efficient.

4.3 Optimization Results

After the processing of the system with $fmincon$, a solution is founded which reduce the time of start-up and maintain the stresses under the imposed constraints. With this solution the system also arrive near 100% of GT_Load (note that in constraints there is a little parameter ε that allow the system to stay under the full load. This parameter is very small, but this solution allow Matlab to obtain more simply the result).
The functions built for the optimization give this optimal K vector of parameter:

$$K = 1.0e+004 * [0.000000327902446\ 0.000233591945258\ 1.231463112769526]$$

where the two first number are Hill function parameters, and the third is the time of simulation. These parameters generate the Hill function showed in Figure 70, which represent the optimal load curve.

Starting from initial condition (x_0 = [10 5000 30000], then 30000 seconds to start-up the plant and arrive at full load) the optimization tool calculate a function that allow to start-up the plant in only 12315 seconds. In figures 71, 72, 73 and 74, whichs represents respectively the temperature of superheater, the steam pressure, the header stress and the steam turbine rotor stress.

In these figures is possible to note how the curves reach, in less time, the full load, with a good behavior for temperature of superheater and for steam pressure. Header stress and steam turbine rotor stress respect the imposed constraints, and

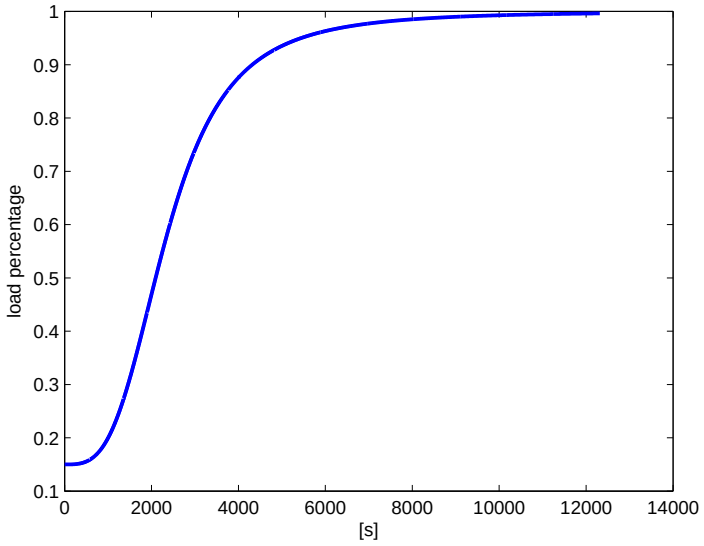


Figure 70: Optimal load curve

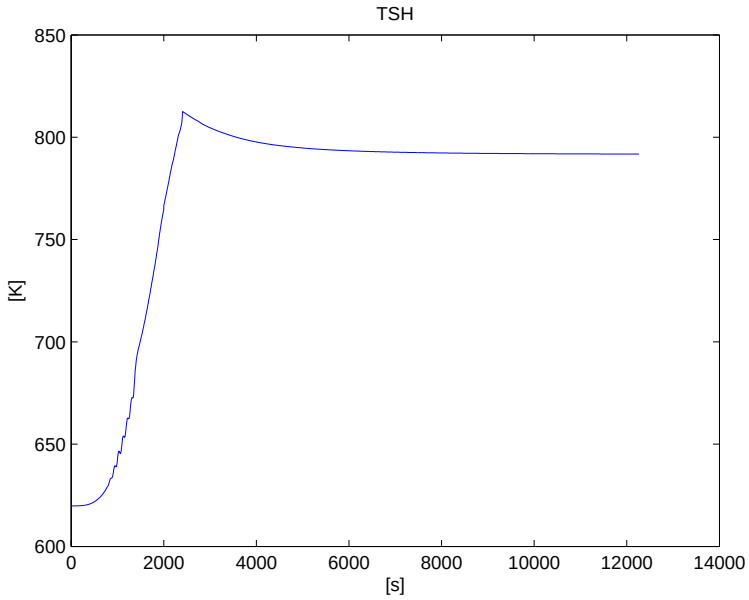


Figure 71: Optimized temperature of superHeater (TSH)

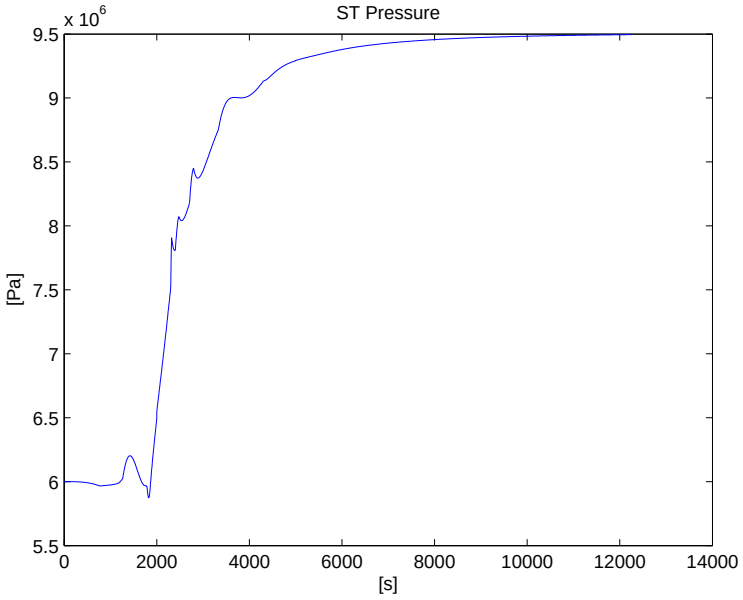


Figure 72: Optimized steam pressure

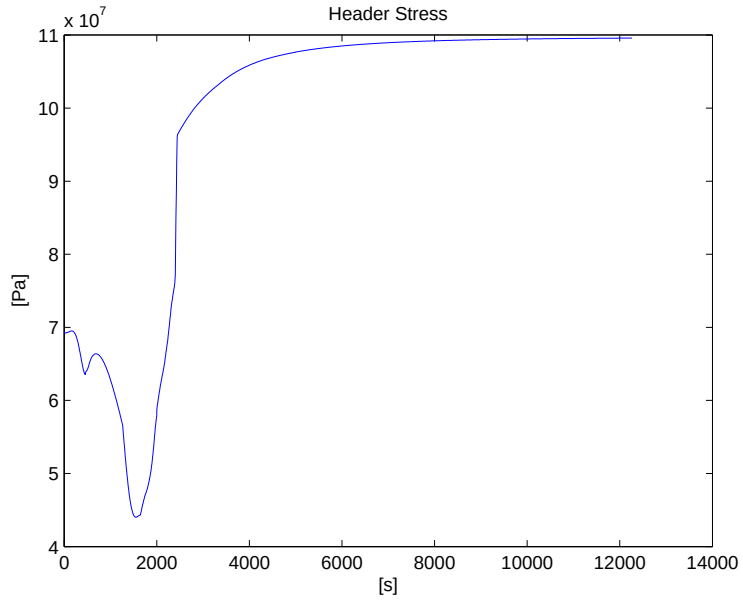


Figure 73: Optimized header stress

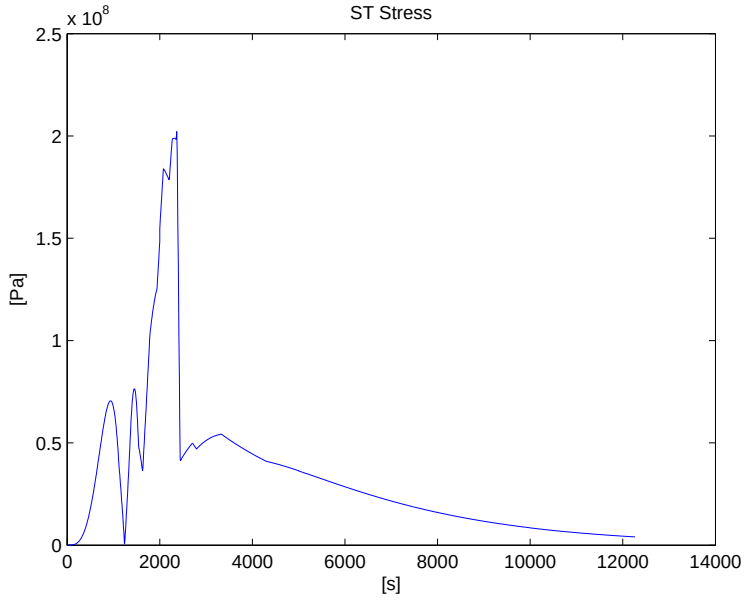


Figure 74: Optimized steam turbine rotor stress

for the first one values are lower than the stress generate from the load ramp.
The successive step is to validate this solution on Dymola model, to observe if the optimization algorithm is even good. The same load is given on Dymola model as a input, with the same admission valve dynamic even imposed by the load. Unfortunately the results are not very well as in Simulink model. Temperature of superheater, steam pressure and headerstress are very similar, as is possibile to see in Figures 75, 76 and 77, and respect the imposed constraints, as in Simulink model.

Steam stress has a very different behavior in Dymola model, as is possible to see in Figure 78: here the stress doesn't respect the constraint, reaching very high value (about 700 milions of Pascal instead 200). Probably this difference is given from the problem highlighted in Chapter 3, that is the non linearities of the model. The problems in Chapter 3 whichs does not allow to obtain a very good identification in start-up phase for steam stress are here mirrored, producing this bad result. In a future development a better identification can probably solve this, but will be necessary restart the optimization algorithm to obtain new values for the *GT_Load*

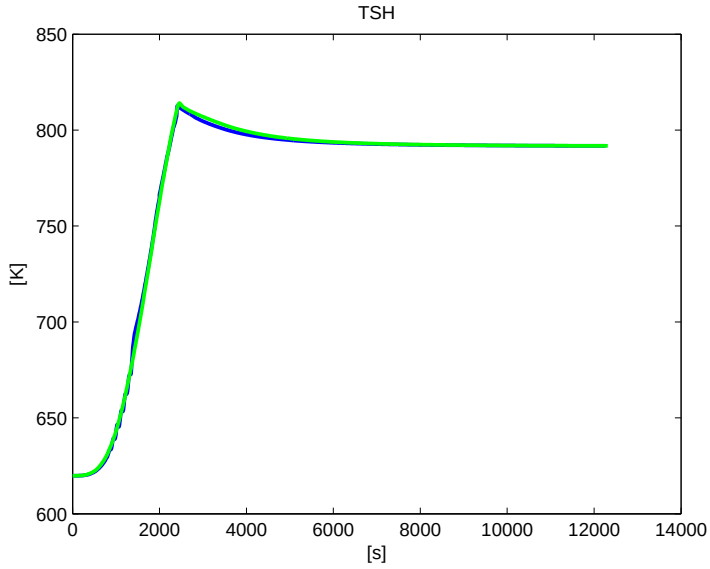


Figure 75: Optimized temperature of SuperHeater (TSH) (Simulink in blue, Dymola in green)

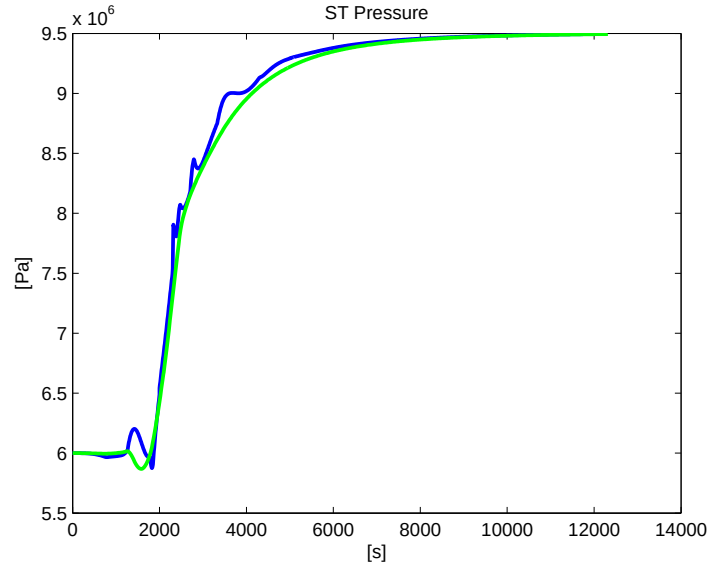


Figure 76: Optimized steam pressure (Simulink in blue, Dymola in green)

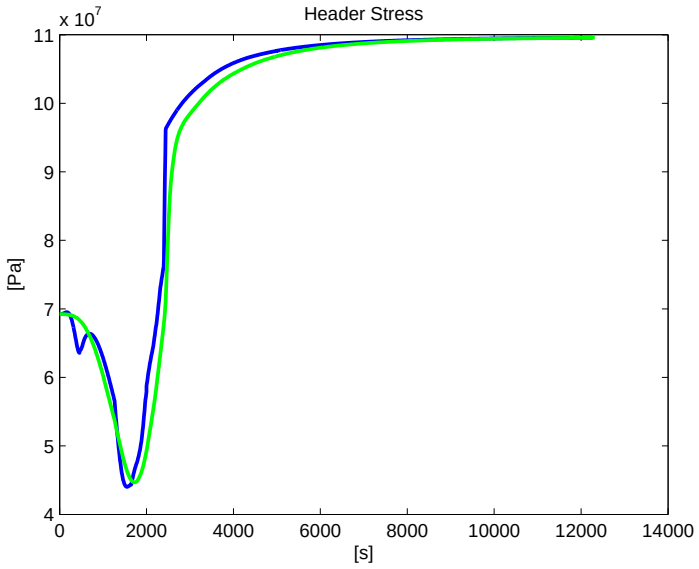


Figure 77: Optimized header stress (Simulink in blue, Dymola in green)

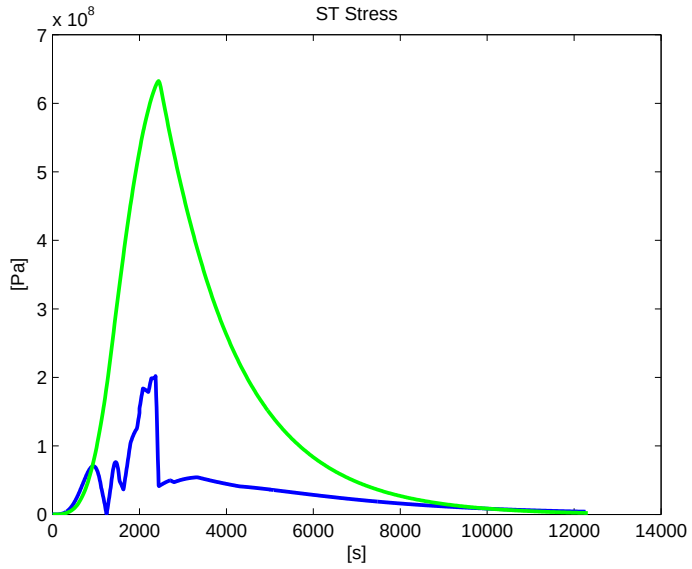


Figure 78: Optimized steam stress (Simulink in blue, Dymola in green)

5 Chapter 5: Conclusion

In this Thesis, a simplified model of a CCPP has been developed in the Matlab/Simulink environment, starting from a detailed Dymola model already available. The model has been derived by interpolation of a number of estimated linear models with local validity, and has been used to optimize the start-up phase of the plant. The simplified model has been validated with respect to large ramp-type variations of the gas turbine load, starting from low loads (15% of the full load) to full load conditions and vice versa. The transients of the main plant variables, namely temperatures, pressures and thermal stresses, have shown in general an excellent correspondence with those provided by the original Dymola model. This is a a significant result, since the behavior of the system is highly nonlinear and asymmetric for positive or negative variations of the load.
The second part of the Thesis has considered the design of the load profile to be followed during the start-up phase to reduce the overall time required to reach full load conditions while maintaining the process variables within prescribed ranges. This requirement is particularly important for the lifetime of the plant, in particular for what concerns the thermal stresses. This problem has been formulated as a mathematical programming problem, and more specifically, as a constrained minimum time problem. The load profile has been assumed to have the form of a Hill function, i.e. a parametrized continuous and increasing function, whose parameters are the unknowns of the optimization problem, together with the final time. The solution of this minimum time problem has been found with standard optimization tools available in the Matlab/Simulink environment. The computed optimal load profile has then been applied to the estimated and interpolated model; this test has verified that the start-up time is effectively reduced with respect to standard procedures and the constraints on the stresses are fulfilled. Finally, the computed optimal profile has been used to simulate the start-up with the Dymola model. The results achieved are largely expected, and confirm that the time required to reach full load conditions is reduced maintaining most of the variables in the prescribed limits. The only exception is due to the behavior of one of the stresses, whose value is too high. This is probably due to a not sufficiently precise identified model for that variable, and it is believed that a more reliable estimated model could completely solve the problem.
In conclusion, the approach developed in this Thesis can be considered efficient, and could probably be extended to analyze other energy problems related to plants with a strongly non linear behavior. Obviously, as it is standard in the development of a new approach, there are many possible future developments, such as:

- the development of a bigger model, which describes all the plant variables (at the moment only a subset of significant variables has been considered in the development of the identified model). This is very easy, because it is necessary only to repeat the identification of the local linear models and their interpolation for any new variable, while maintaining the same interpolation

structure.

- Consider the possibility to use other input variables in the optimization procedure.
- Include a model of the emissions of the gas turbine and constraint their value in the optimization problem for the determination of the load profile. This will become more and more important in the future, in view of the tight legal requirements on the emission of CCPP.
- Use an optimization procedure to tune the parameters of the *membership functions* which are used in the interpolation of local linear models. This would allow to save a lot of time in the design phase of the model, and probably would improve the achieved solutions.
- Use more efficient optimization tools than those applied in this work (the function *fmincon* in the Matlab optimization toolbox).
- Consider other load profiles. The Hill function is a flexible and easy choice, with a small number of parameters to be computed, but other functions could be considered. Another possibility would be to consider the cascade of two Hill functions to replicate the shape usually followed in industrial plants.
- Extend the approach to other energy system, developing a set of models very easy to simulate. This can be very convenient also in the project phase, because it gives the possibility to have simple and fast tools for simulation, control design and optimization.

A Simulations Procedure

This appendix reports a short manual describing how to perform the simulations with Dymola and hao to manage all the files developed. The first subsection, describes the methodology to obtain results from the SUPELEC Dymola model, in order to have similar results at different loads.

A.1 Simulations on Dymola

The Modelica model provided by SUPELEC has two possible initialization points: full load conditions or at 7.5% of load. In this case, the initialization is set by the *hot_start_up.txt* file, available in SUPELEC package, see [15]. In this section it is shown how to perform different simulations starting from steady-state conditions characterized by a constant load in the range 7.5%-100%. The simulations are always performed with a local regulator controlling the drum level. This is necessary for the identification procedure described in the following chapter to eliminate the integrating behavior of the level, which can cause troubles in the simulation phase. Moreover, also a regulator controlling the steam pressure is used to guarantee that this variable remains into prescribed limits.

The procedure to run the simulations is easy, but must be strictly followed to avoid errors and false results. As an example, in the following it is described how to simulate the system at values of the load greater or equal to the the 50%:

1. load the model, with the drum level in closed-loop and at the 100% of the load, with the file *dsin.txt* of default parameters provided by SUPELEC and with the file *dsu.txt* of constant inputs (the values of the variables at this operating point are presented in a table reported below);

2. "translate" the model in Dymola and run a very long simulation (for example of 100000 seconds) in Dymola, to obtain the steady-state at 100% of the load (see Figure 79 on how to set the simulation time);

3. in the folder where the model is placed, there is now a file called *dsfinal.txt*, which represents the final state at the end of the simulation. Copy and rename it as *full_load_controlled.txt*;

4. choose the operating point of interest. To do this, the model needs a new *dsu.txt* file, which, according to a ramp behavior, allows to reach the selected operational point. With this new input file (which must be placed in the same folder of the model), the Modelica model must be re-translated and re-simulated to generate a new initialization *.txt* file (it is important to pay attention at the settings of the initial conditions in Dymola, see Figure 80);

5. as before, the *dsfinal.txt* file represents the final state at the required load level. It is more useful to save this file with another name, for example

loadpercentage.txt. In Dymola it is also possible to find the stable value of the variable Feed_Drum at the considered load level;

6. with the file previously created, the initial conditions at desired operating point can be loaded. Now it is possible to perturb the system with a local transient near the chosen working point with a new *dsu.txt* file, which contains the perturbation inputs;

7. the results of the simulations are stored in the file *model_name_results.mat*. This file is very huge for long simulations. To avoid this problem, is necessary to select, in "Dymola simulation setup", only the input and output variables in the menu. In this way internal and auxiliary variables are not saved in the file of the results.

The procedure to run the simulation under of 50% of the load is similar, but attention must be paid to use the system with two control loops (level and pressure). In this case, not only the stable value of the Feed_Drum must be found at different loads, but also the opening of the valve controlling the steam pressure. In fact, the pressure loop is closed on the admission valve.

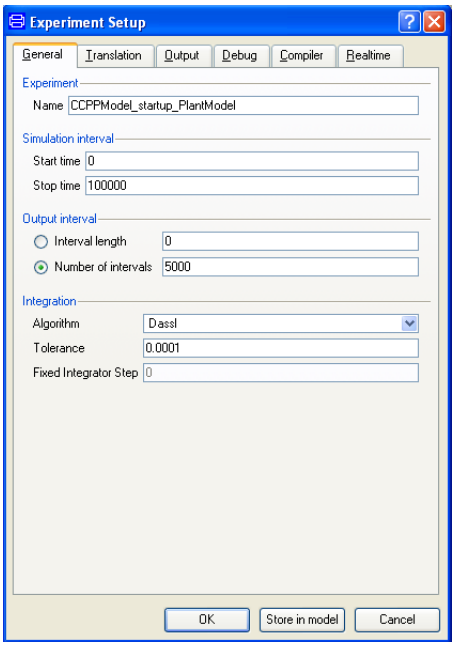


Figure 79: How to set the time in Dymola.

A.2 Files manual

The target to this part of the appendix is to become easy the work on the thesis files. A lot of files have been generated with Dymola, Matlab and Simulink: a

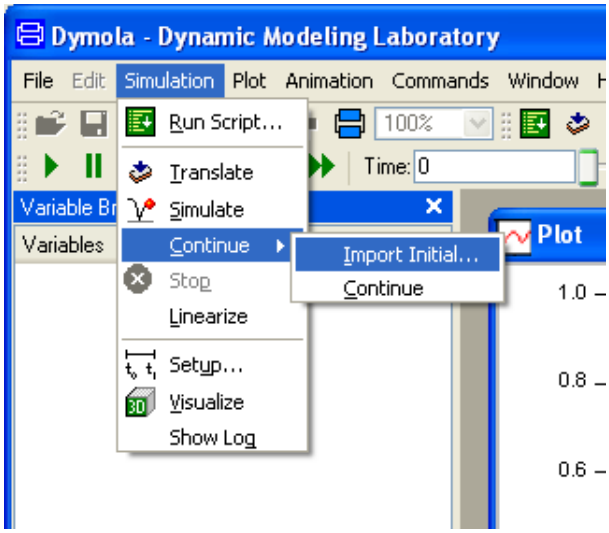


Figure 80: How to import the state in Dymola.

little manual to understand how the file are can be very useful.
In the main *Thesis* folder there are:

- Dymola Simulations
- GUI
- Identifications
- Simulink

In *Dymola simulations* folder all the different simulations made with Dymola have been gathered. There are *Closed_Loop1* and *Closed_Loop2*, whichs contain the tries to obtain the states at different load necessary to initialize the model (as is explained in A.1. In this folder another one is contained, called *Open Loop*: inside of this all different simulations with square wave are gathered. It is possible to find two different type of these, one with the total open loop model, one with the closed loop model on the drum level. This last it is considered for the identification and for all the thesis work.
In *GUI* folder are stored all the data from simulations, and there is the code to run the dedicated GUI. With Matlab is sufficient run the *fdtgui2.m* file to start it. How is explained in Chapter 2, is necessary to load the *all_load.mat* file to see the local transitory.
In *Identifications* folder are gathered all the identification done with Matlab Identification ToolBox. The ".sid" files can be loaded in *ident* mask to see directly the different identifications. To facilitate the read of these files, all are called like *output_input_first*. In this way is very easy take the proper variable to analyze. Inside

of this folder, others two are present: in one of this there are some Matlab script to obtain the data for the identifications, in another one the Matlab workspace data to start the them. Because the inputs are three and the outputs for, twelve files have been generated.
In *Simulink* folder are stored all the Simulink plant realized to reproduce the Dymola model in Matlab. There are one folder for each reconstruited output: this choice is very good, because is better tune every model alone. This avoid a lot of problems in project phase. The *Total* folder store the model which assemble the single output models. *Optimiz* and *Optimiz2* contain the code to start the optimization phase, the only difference between the first from the second is able to launch many simulations.
A little precisation to work on Simulink: all developed models require the load of properly workspace file, that contains all the *ldmodel* form (the local transfer function). These are present in every folder and also in *workspace* folder.

References

[1] *Mathmodelica*.

[2] *Openmodelica*.

[3] Dynasim AB, *Dymola v. 7.2*.

[4] Claudio Aurora, *Power plants: Modelling, simulation and control*, Ph.D. thesis, University of Pavia, 2004.

[5] Automatica 31, *Identification of nonlinear system structure and parameters using regime decomposition*, 1995.

[6] R.D. Bell and K.J. Astrom, *Automatica*, 2000.

[7] Francesco Casella, *Materiale didattico corso modellistica e controllo dei processi continui e delle reti di pubblica utilità*.

[8] Francesco Casella and Alberto Leva, *Modelling of thermo-hydraulic power generation processes using modelica, mathematical and computer modelling of dynamical systems, v. 12, n. 1, pp. 19-33*, 2006.

[9] Francesco Casella and Riccardo Scattolini, *Procedure di esercizio innovative per la gestione ottimale dei gruppi di produzione in fase di avviamento, relazione finale*, Tech. report, Dipartimento di Elettronica e informazione, Politecnico di Milano.

[10] DyOS, *Dyos user manual, release 2.1*, Aachen University, 2002.

[11] B. A. Foss and T. A. Johansen, *Constructing narmax models using armax models*, Int. J. Control, 1993.

[12] Morthazavi Hesam, *Optimization of the startup time in a combined cycle power plant considering the thermo mechanical stresses developed in the steam turbine*, Master's thesis, Politecnico di Milano, 2006.

[13] C. Maffezzoni, *Dinamica dei generatori a vapore*, Hartmann & Braun, 1988.

[14] Modelica Conference, *Modelica open library for power plant simulation: design and experimental validation*, 2003.

[15] Adrian Tica, *One pressure level ccpp model*, Tech. report, SUPELEC, December 2009.