Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Construction of a Simulator for the Siemens Gas Turbine SGT-600

Examensarbete utfört i reglerteknik

av

Lisa Nordström

LITH-ISY-EX--05/3713--SE

Linköping 2005-11-02

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET

Department of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Linköpings tekniska högskola
Institutionen för systemteknik
581 83 Linköping

Construction of a Simulator for the Siemens Gas Turbine SGT-600

Examensarbete utfört i reglerteknik
vid Linköpings tekniska högskola
av

Lisa Nordström
LITH-ISY-EX--05/3713--SE

Handledare: Markus Gerdin
Examinator: Torkel Glad
Linköping: 2005-11-02

Framläggningsdatum
2005-10-21

Publiceringsdatum (elektronisk version)

Institution och avdelning
Institutionen för systemteknik
Reglerteknik

Linköpings universitet

Språk
Language
Svenska
X Engelska/English

Rapporttyp
Report category
Licentiatavhandling
X Examensarbete
C-uppsats
D-uppsats
Övrig rapport

ISBN:

ISRN: LITH-ISY-EX--05/3713--SE

Serietitel

Serienummer/ISSN

URL för elektronisk version

Titel
Title
Construction of a Simulator for the Siemens Gas Turbine SGT-600
Utveckling av simulator för Siemens gasturbin SGT-600

Författare
Author
Lisa Nordström

Sammanfattning
Abstract
This thesis covers the development of a simulator for the Siemens Gas Turbine SGT-600. An explanation on how a gas turbine works is also given, as well as the principles behind the control system used by Siemens to control the turbine.

For Siemens Industrial Turbomachinery to be able to test its control system before delivering a gas turbine to the customer, a simulator is needed. The control system needs to be adjusted for every unique gas turbine, since there are several options for the customer to choose between when ordering the turbine. A control system standard is under development, which also needs to be tested in a simulator.

The framework for the simulator, i.e. the hardware and software that form the simulator system, was predefined to suit this specific purpose. The Siemens software SIMIT is used for developing the model. SIMIT is a real time simulation tool where models are constructed using blocks, similar to MATLAB Simulink.

A gas turbine is basically a heat engine that produces mechanical energy or electricity. The main task of the control system is to control the fuel flow to the combustion chamber and by that keeping the machine at desired speed.

The gas turbine model was developed using measurement data from a site in Hungary, where a gas turbine of the type SGT-600 is in service. The model is based on simplified relations between the signals. By analyzing measurement data and learning about the functionality of a gas turbine it was found out that the speed of the gas generator affected most other signals, like temperatures and pressures. The gas generator speed was found to be dependent on the heat flow, which is determined by the openings of the gas control valves.

As a result of this thesis a working simulator for the gas turbine SGT-600 has been developed. The simulator can be used for testing the control system standard and for testing the control system when adapting it to a specific delivery. It is also suitable for educational purposes, for example to instruct customers.

Nyckelord
Keywords
Gas Turbine, Simulator, Modeling, Siemens, Control System, SIMIT

Abstract

This thesis covers the development of a simulator for the Siemens Gas Turbine SGT-600. An explanation on how a gas turbine works is also given, as well as the principles behind the control system used by Siemens to control the turbine.

For Siemens Industrial Turbomachinery to be able to test its control system before delivering a gas turbine to the customer, a simulator is needed. The control system needs to be adjusted for every unique gas turbine, since there are several options for the customer to choose between when ordering the turbine. A control system standard is under development, which also needs to be tested in a simulator.

The framework for the simulator, i.e. the hardware and software that form the simulator system, was predefined to suit this specific purpose. The Siemens software SIMIT is used for developing the model. SIMIT is a real time simulation tool where models are constructed using blocks, similar to MATLAB Simulink.

A gas turbine is basically a heat engine that produces mechanical energy or electricity. The main task of the control system is to control the fuel flow to the combustion chamber and by that keeping the machine at desired speed.

The gas turbine model was developed using measurement data from a site in Hungary, where a gas turbine of the type SGT-600 is in service. The model is based on simplified relations between the signals. By analyzing measurement data and learning about the functionality of a gas turbine it was found out that the speed of the gas generator affected most other signals, like temperatures and pressures. The gas generator speed was found to be dependent on the heat flow, which is determined by the openings of the gas control valves.

As a result of this thesis a working simulator for the gas turbine SGT-600 has been developed. The simulator can be used for testing the control system standard and for testing the control system when adapting it to a specific delivery. It is also suitable for educational purposes, for example to instruct customers.

Preface

This thesis was written at the department of Controls Engeineering at Siemens Industrial Turbomachinery in Finspång, as a final thesis to complete the studies for Master of Science in applied physics and electrical engineering international, at Linköping University.

Thanks to Tomas Strömberg, instructor at Siemens, Markus Gerdin, instructor at the university, Torkel Glad, examiner, Maja Johansson, opponent, Emil Haraldsson, linguistic inspector and moral support and to Andreas Lindholm and Stefan Gunnarsson for invaluable help.

Table of contents

Figure- and chart-index

1 Introduction

This chapter will explain the purpose of this thesis and the reason why it was initiated. The scope will be concretized by declaring what is not covered and the structure will be commented on to provide guidance to the reader.

1.1 Aim

The purpose of this master thesis is to develop a simulator for the gas turbine SGT-600. The framework for the simulator has been fixed in two earlier final theses, i.e. the hardware and software has been tested and evaluated and adjusted for the application.

The simulator is supposed to be used for testing the control system before the delivery of a gas turbine. This will shorten developing time and reduce installing time at site, thus reducing costs. A control system standard is under development, which also needs to be tested in a simulator. The simulator can also be used for educational purposes, like training of new personnel and demonstration for the customer.

It should be possible to start the simulator from the operator station so that it automatically runs through the start sequences and reaches normal running state. From there different running states should be possible to simulate. Ideally the operator should be able to switch between the simulator and a real process and not notice any difference.

1.2 Background

The Finspång site, developing and producing gas and steam turbines, has since 2003 been owned by Siemens and is since 2004 called Siemens Industrial Turbomachinery AB.

Siemens is currently migrating from ABB's control system to Siemens own, called S7. For every new gas turbine that Siemens sells, the control system has to be adapted to the specific options included in the delivery. The control system needs to be tested against a simulator. Previously a rough model in the ABB system ADVANT had been used. Since the control system is being converted to S7, there was a need for a new simulator. In two earlier final theses a simulator for the gas turbine SGT-700 has been developed. Emphasis was on bringing forth the most suitable simulator setup. Siemens also needed a simulator for another gas turbine type, SGT-600, which is a predecessor to the gas turbine SGT-700.

1.3 Problems to be solved

The task to develop a simulator for the gas turbine SGT-600 can be divided into the following subtasks:

- Select an appropriate model type
- Find and analyze relevant measurement data.

- Acquire knowledge of the principles behind how a gas turbine works and how it is controlled
- Decide parameter values to model the relations between the signals
- Get the simulator setup system to work with the new control program for SGT-600
- Implement the model in SIMIT
- Test the model and evaluate the results

1.4 Scope

The gas turbine model should be of the type SGT-600, although it is desirable to improve the existing model of gas turbine type SGT-700.

The gas turbine model should be of Mechanical Drive type, and not Power Generation type. This was the natural choice since the control system code developed for the SGT-600 type is of Mechanical Drive type.

The simulator setup (the hardware and software that is used in the simulator system) should be the same as in the previous final theses. This is because the time frame given would not allow time to be spent on testing new software and hardware. The previous final theses writers had already gone through several alternative solutions before deciding on the chosen one.

The compressor load does not have to be implemented in this thesis. The part of the control program that controls the compressor is loaded into a separate PLC. Since only two PLC:s were available for this thesis, the compressor part of the control program will not be used. For that reason and due to time reasons the modeling of the compressor had to be left out. An additional reason for not modeling the compressor was that another final thesis covering the compressor was being planed.

The model will be of a gas turbine running on gas fuel only. It was desirable to model dual fuel (gas and liquid fuel), but since the control system code for the liquid fuel subsystem was not yet developed, there was no point in including it in the model, since it would not have been possible to test.

1.5 Structure of the thesis

This thesis has not been strictly divided into one theoretical part and one implementation part. Instead the theory is provided where it is needed. This was done to make the thesis more readable and interesting.

First the basic functionality of a gas turbine is explained, to give the reader an idea of what is to be simulated. Then the control system and the simulator system are explained, together with the hardware and software used. This is followed by a description of the construction process and an

illustration of the testing of the model. The thesis is concluded with a discussion about the results and suggestions on further development.

2 How a Gas Turbine Works

This chapter will give a basic understanding of how a gas turbine works. The theory is divided according to the main parts of the gas turbine. Some specifics on the SGT-600 type are added.

2.1 Overview

A gas turbine is a heat engine that converts chemical energy from the fuel into heat energy, which is converted into mechanical energy. The efficiency is between 25 and 45 percent. [1]

The main parts of the gas turbine are the gas generator and the power turbine. The gas generator consists of compressor, combustion chamber and compressor turbine (see figure 1 and 2). The purpose of the gas generator is to generate a flow of pressurized hot gas, driving the power turbine. The purpose of the power turbine is to convert the pressurized hot gas flow from the gas generator to mechanical energy, driving a load. [2]

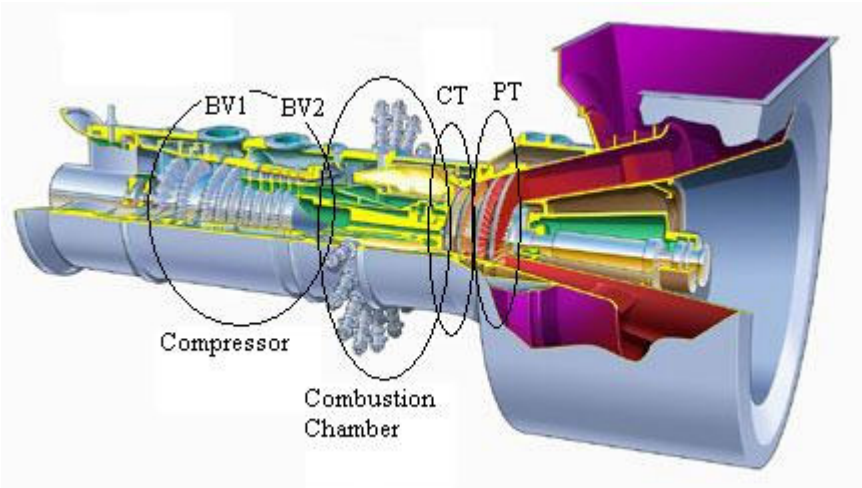

Figure 1 Gas Turbine SGT-600; Bleed Valve 1 (BV1), Bleed Valve 2 (BV2), Compressor, Combustion Chamber, Compressor Turbine (CT), Power Turbine (PT)

Compressor, combustion chamber and turbine are encapsulated in cylindrical casing, where the flow of air and gas is moving straight through. The compressor compresses air for the combustion chamber, where fuel is mixed with air and combusted. The hot gases pass through the compressor turbine, which is driving the gas generator and then expand through the power turbine. In an open gas turbine cycle, which is the completely dominating form, the air and combustion gases are let out in the atmosphere, whereas in a closed gas turbine cycle the gas is cooled and led back to the compressor. [1]

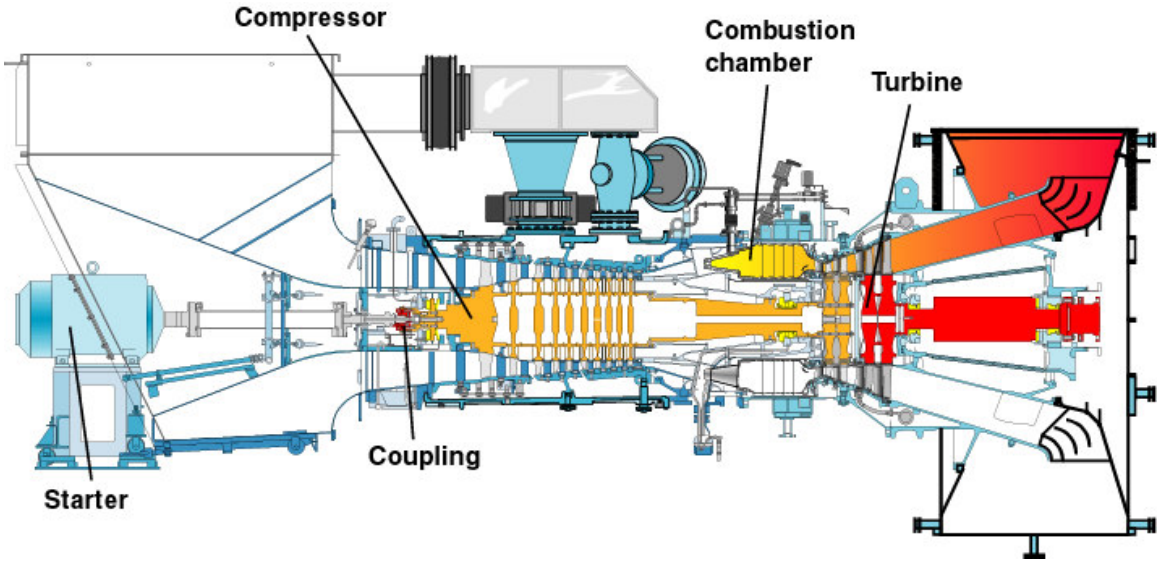

Figure 2 Overview of the gas turbine

If the gas turbine is to be used for power generation (PG), there is an AC generator connected to the power turbine. If it is to be used for mechanical drive (MD) there will usually be a compressor connected to the power turbine, which is used for compressing natural gas in a pipeline. Compressors are used at certain intervals in the pipeline to keep the gas flowing at a steady rate. Mechanical drive applications also include pumping up oil and driving ferries.

2.2 Compressor

The purpose of the compressor is to compress air for the combustion. An axial flow compressor consists of one or more rotor assemblies that are mounted between bearings in the casing. The compressor is a multi-stage unit, where the pressure is increased by each stage. Each stage consists of one vertical layer of rotating blades (see figure 3) and one of stator vanes. The stator vanes decrease the air velocity, increasing the pressure. They also aim the airflow at a correct angle to the next section of rotor blades. Sealings between the stages prevent the air from leaking. From the front to the rear the cross section area of the airflow is decreasing, so that the axial velocity remains constant as the volume decreases due to the compression. [1]

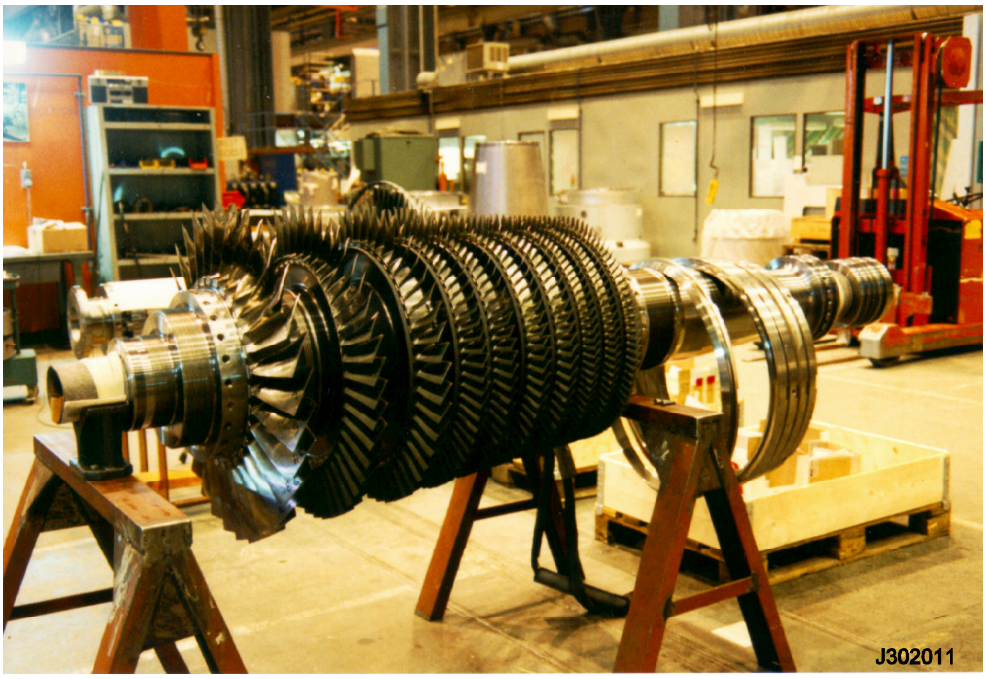

Figure 3 Compressor

Ambient air is taken through an inlet duct and passes a filter before reaching the compressor. The filter is necessary to prevent objects to enter the compressor and to minimize erosion and corrosion. The airflow to the compressor is controlled by two variable vane stages at the inlet of the compressor, called the inlet guide vane. The SGT-600 gas turbine has ten stages, divided into three parts: two low pressure sections consisting of the first five stages and a high pressure section consisting of the last five stages. [1]

The compressor has two cavities, one low pressure cavity between stage two and three and a high pressure cavity between stage five and six. The cavities are connected to bleed valves in the casing, the low pressure cavity to bleed valve one, which is a binary valve and the high pressure cavity to bleed valve two, which is controllable (see figure 4). [1]

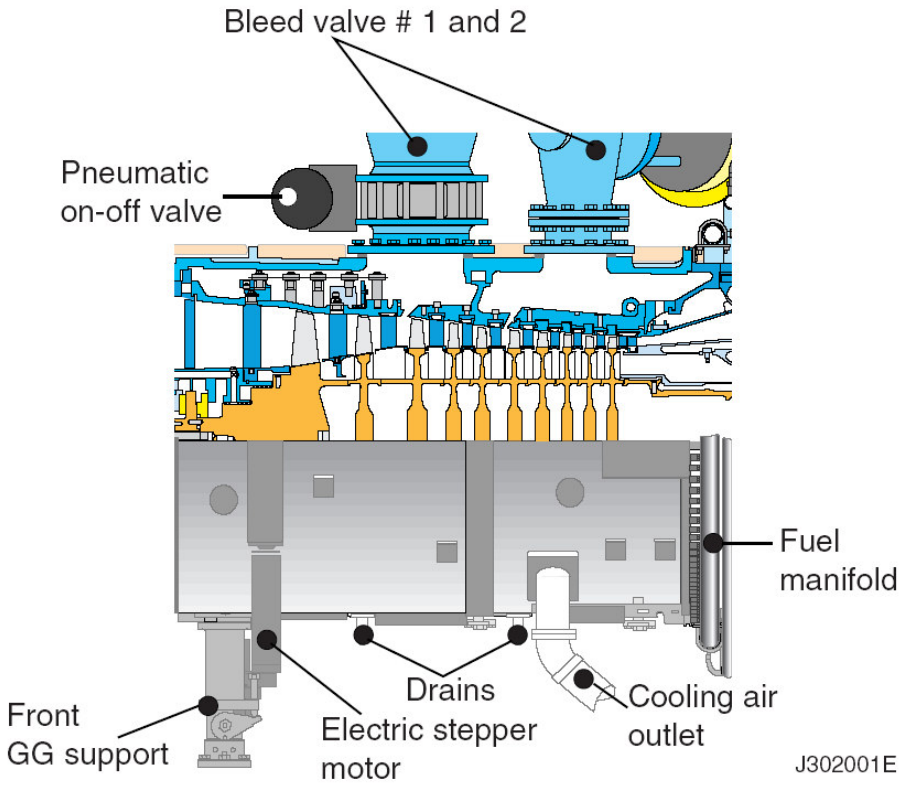

Figure 4 Bleed valves

The bleed valves are, together with the variable guide vanes, used for preventing stall and surging. Surging is a phenomenon that can occur during start and stop, when the volume air flow is low. The decrease of the cross section area of the flow in the compressor is optimized for full load (full speed), while at low speed the smaller compression ratio would actually require a less reduction of the cross section area. If the compressor rotor accelerates too quickly, the airflow velocity will be too small in relation to the blade velocity. Since the pressure rise does not correspond to the decreasing volume the small rear end of the compressor will not be able to complete the compression. This leads to a reverse airflow, which causes oscillations in the compressor. Surging is when the airflow through the whole compressor is broken down, stall when only some stages are affected. The oscillation can damage the compressor since it creates stress on the blades. [1]

The bleed valves open during start and stop to bypass some of the air to avoid surging. By controlling the variable guide vanes the airflow to the rear stages of the compressor can be decreased. At cooling down the bleed valves are closed so that all air is used for cooling. [1]

Air from the low pressure cavity is used as seal air to prevent oil leakage in bearing 1, 3 and 4 and air from the high pressure cavity is used for cooling the power turbine discs. [1]

2.3 Combustion chamber and ignition

In the combustion chamber the fuel injected through the burners is burnt with air supplied by the compressor. The combustion chamber is of annular form with 18 burners in the front end (see figure 5 and 6). The burners are connected to a section of the turbine inlet. The forward end of the burning chamber casing is connected to the compressor via a diffuser. The air from the compressor has a velocity of about 100 meters per second. The airflow must therefore be decelerated in order not to blow out the flame. In the combustion chamber a region of low axial velocity and re-circulating flow has to be created. Efficient combustion is necessary to obtain high thermal efficiency and to minimize the exhaust gas emissions. Flame temperature must be 1000-2000 C for efficient combustion. The combustor walls have to be cooled, since they cannot stand the high temperatures. Out of the airflow from the compressor only about 25% is supplied to the combustion zone at full load, the rest is used for cooling the combustor walls and to dilute the hot gases to a temperature low enough not to damage the turbine parts. [1]

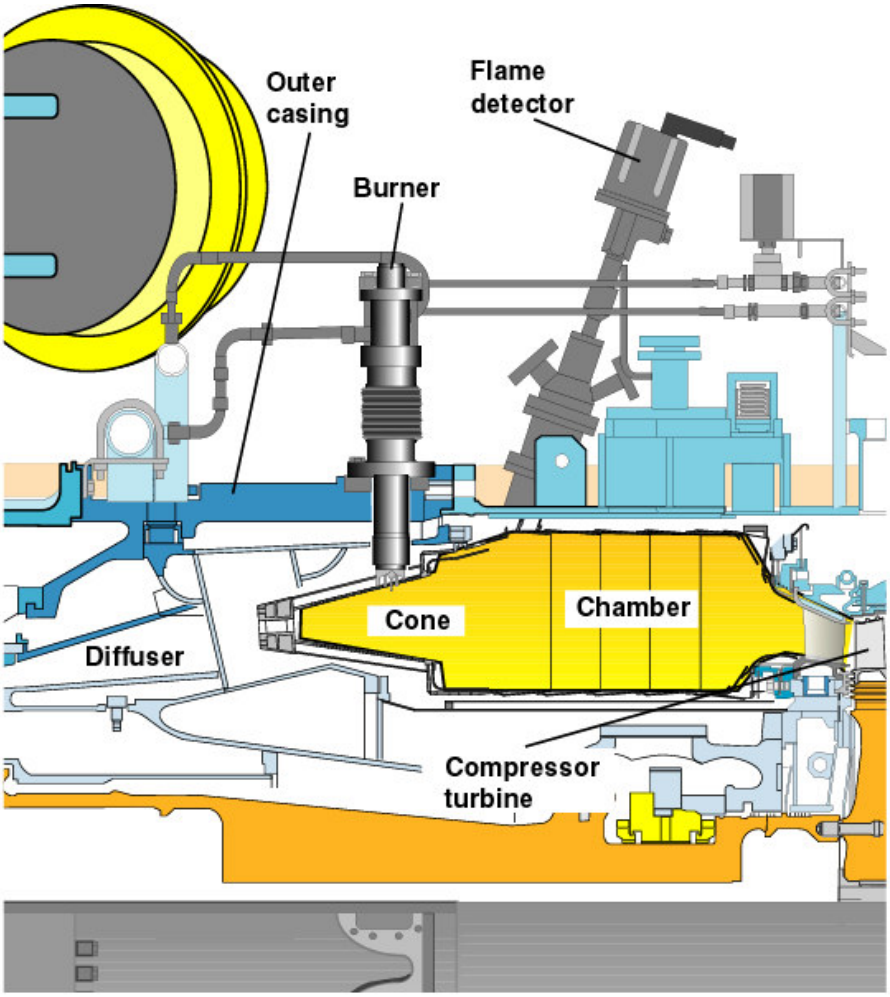

Figure 5 Combustion chamber with burner and flame detector

Figure 6 combustion chamber

During the start procedure, gas is led through a small pipe to an ignition burner, air is let in by opening of ignition air valves and an igniter ignites the mix of air and gas. Then gas is led through another small pipe and enters at the cone tip of burner six and a pilot flame is ignited. After that the main and the primary gas control valves start to open, letting gas from a larger pipe enter all the burners, igniting the main flame (See Figure 8). The primary gas control valve opens first and is used for controlling the amount of gas until a certain load is reached. Then the main valve starts to open as the primary valve starts to close, resulting in an even flame. The reason why the gas supply to the combustion chamber is controlled by two valves is that it gives the most stable flame form. Gas from the primary valve enters the burners at the cone tip, resulting in a long thin stable flame, while gas from the main valve enters at the cone sides, resulting in a wide unstable flame (see figure 7).

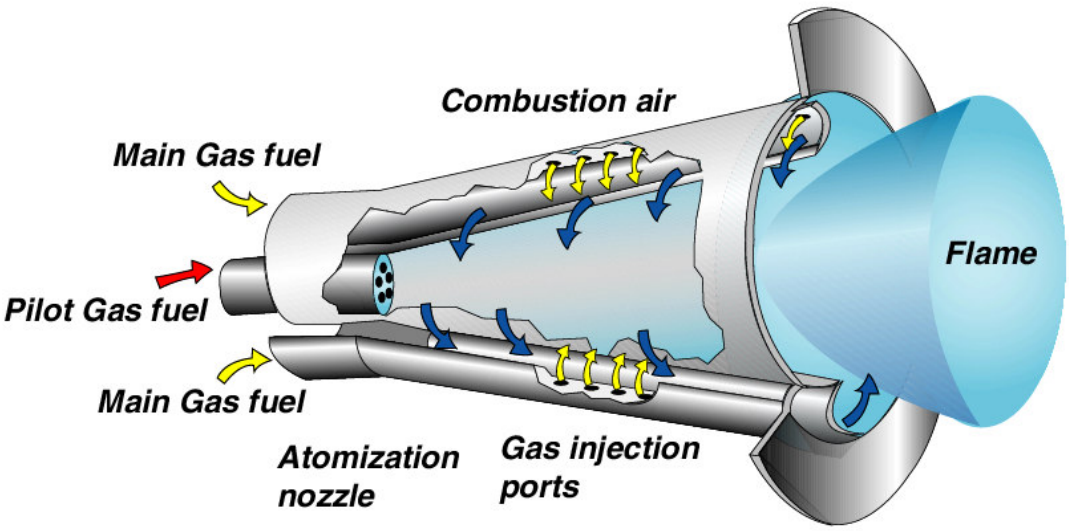

Figure 7 Double cone burner

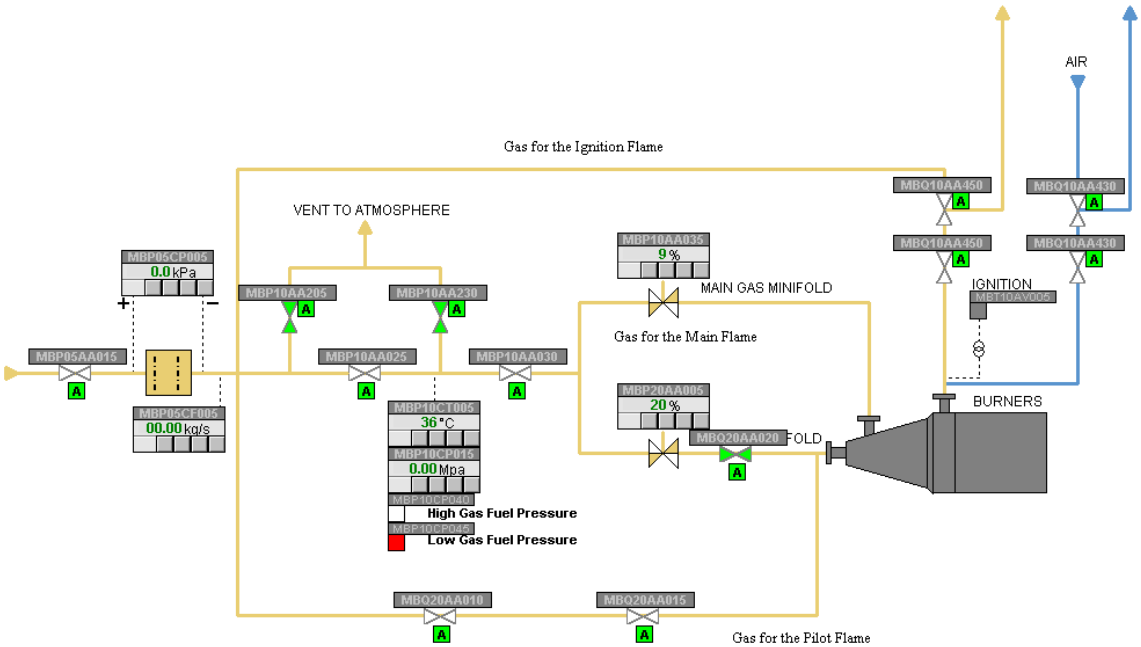

Figure 8 Operator Picture of the Fuel System showing the gas paths for the Ignition, Pilot and Main Flames

There are two flame detectors in the combustion chamber, one to detect the ignition and pilot flames during start and one to detect the main flame during start and normal operation.

There is a variable dilution system that decreases the airflow through the burners by providing a controlled bypass of air to the combustor exit. The purpose is to ensure complete combustion and therefore minimize the CO emissions and increase the efficiency on part load. Some of the air is by-passed from around the combustion chamber into an inner manifold, passed through six control valves to an outer manifold and fed back into the end of the combustion chamber. The valves are connected by a drive ring connected to a motor. Without the system, flame temperature would drop with increasing load, since the air-fuel ratio increases. A too low flame temperature results in an incomplete combustion with formation of CO. [1]

2.4 Compressor Turbine and Power Turbine

The two-stage compressor turbine provides power to drive the compressor and the power turbine gives useful mechanical output, driving a load. The turbines extract energy from the hot gases from the combustion chamber by expanding the gas to lower pressure and temperature. [1]

Each turbine stage consists of a row of stationary guide vanes, mounted to the turbine casing, followed by a row of moving blades, fitted to turbine discs (see Figure 9 and 10). Hot gas is expanded in the convergent passages between the guide vanes. Pressure energy is converted into kinetic energy and the gas is accelerated. When the gas reaches the turbine blades it is given a spin/swirl in the direction of the blades. The gas is forced to deflect and is further expanded, since the passages are convergent. During this process energy is absorbed, causing the turbine to rotate and provide power

for driving the turbine shaft. Since gas expansion, in contrast to compression, is a spontaneous process, fewer stages are needed to expand the gas to atmospheric pressure. [1]

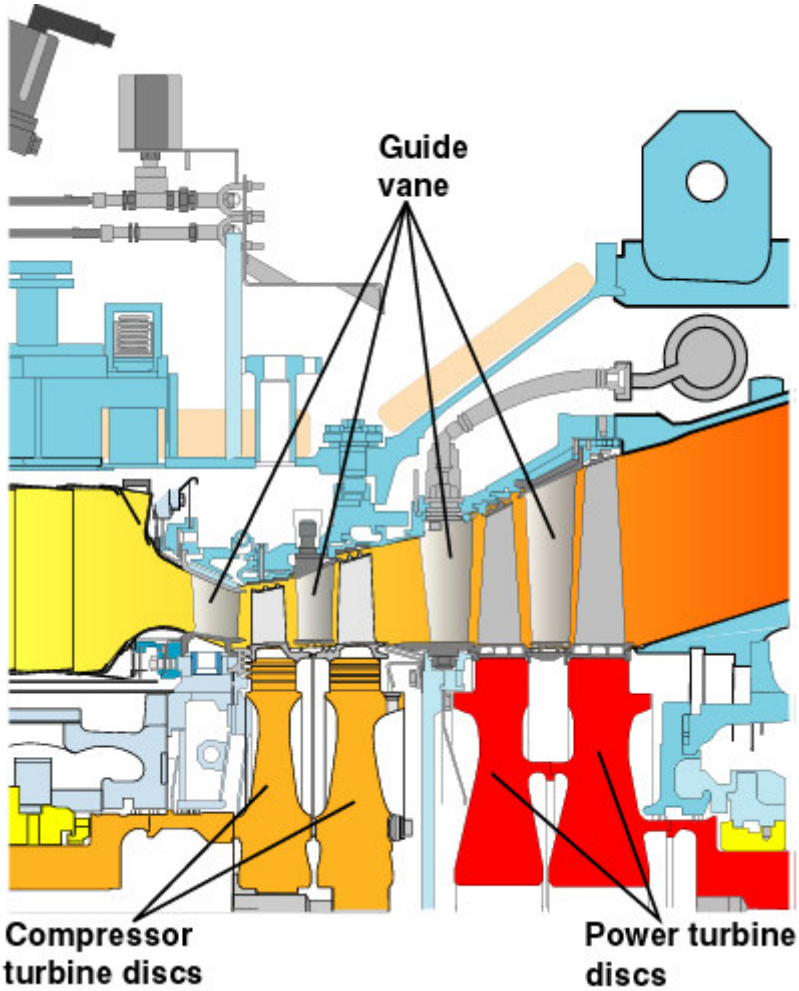

Figure 9 Compressor turbine and power turbine, showing guide vanes and turbine blades

Sealings prevent gas leakage between the stages and to shafts and bearings. Sealing air bled off from the compressor is lead off along the turbine discs, to cool them and prevent heat transfer to shafts and bearings. [1]

The gas generator rotor and the power turbine are carried by bearings. Oil is continuously supplied to the bearings during operation. [1]

In full operation the speed of the power turbine is about 7700 rpm. For power generation the AC-generator is connected via a gearbox to reduce the generator speed to 1500 rpm. In a two-shafted gas turbine, like the SGT-600, the power turbine is not mechanically interconnected to the gas generator. The speed of the gas generator is determined by the actual power demand in combination with ambient conditions, such as temperature and humidity. This allows a wider operating range in comparison with a single-shafted model. [1]

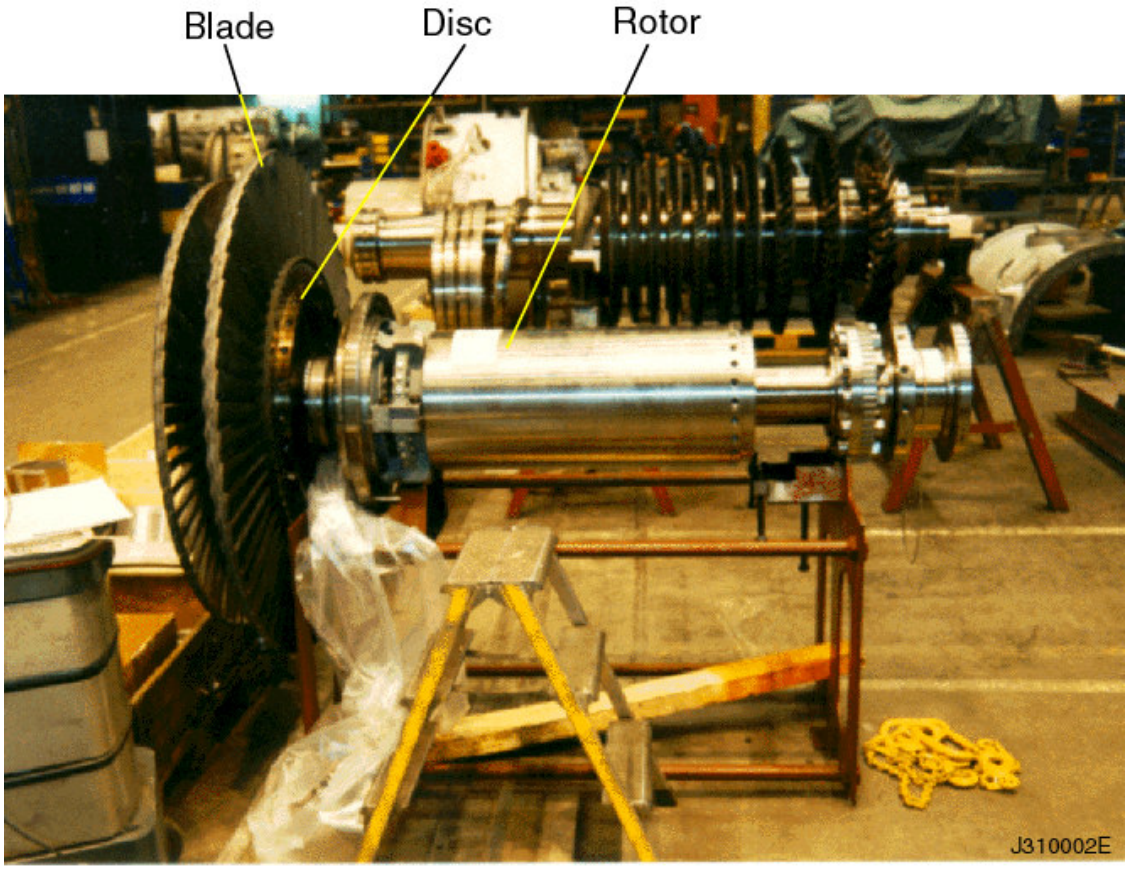

Figure 10 Power Turbine

2.5 Starting System

The compressor is set in motion by a start motor, connected to the compressor through a coupling. The starting system speeds up the gas generator to a speed necessary for purging (ventilation of the gas turbine and the exhaust system to get rid of uncombusted gas) and ignition. That speed is approximately 2300 rpm. After the purging the main fuel supply is opened to the combustion chamber and the gas turbine is ignited. The starting motor helps the gas turbine accelerate beyond self-sustaining speed (approximately 5400 rpm). As it reaches self sustaining speed the clutch disengages and the starting motor is switched off. [1]

When the gas turbine is to be stopped the load is decreased slowly. When it is reduced to about 400 kW the fuel supply is closed and the rotors will coast down. The starting system is activated and given a speed reference of approximately 230 rpm. When the gas generator reaches that speed the clutch will engage and the gas generator will be driven by the starting motor until it has cooled down, which takes ten hours. In case of power failure emergency batteries will drive the starting motor. The reason why the turbine has to rotate during cool down is that the rotors, which are supported by the bearings only at each end, will deform if they are brought to stand still while they are still hot. [1]

2.6 SGT-600

SGT-600, is a compact, low-weight heavy-duty industrial gas turbine with an electrical output of 24.8 MW. [3]

It is suitable for both mechanical drive applications and power generation. The applications also include combined cycle operation, cogeneration and marine propulsion. Combined cycle operation is when the gas turbine is connected to a steam turbine, where the waste gases from the gas turbine are used to heat a boiler that provides the steam turbine with steam. Cogeneration means that the gas turbine is equipped with a waste heat recovery unit (a boiler fan). The waste heat can be used for production of industrial process steam or for district heating for example. [3]

Siemens Industrial Turbomachinery in Finspång sells about 60 Gas Turbines every year. About half of them are of the type SGT-600. The reason for the high percentage of this type is that it is suitable for driving a compressor (mechanical drive), which is the most common application on the market.

3 The Control System

This chapter will give a brief explanation of the hardware and software used by Siemens to control a gas turbine and provide understanding of how the control system works.

3.1 Configuration overview

The gas turbine process is controlled by the control program, which is located in CPU:s that are part of programmable logic controllers (PLC:s). The PLC:s are connected with PROFIBUS to I/O modules that send and receive signals from sensors and to actuators on the gas turbine. The PLC:s are also connected to an operator station (a PC station with human-machine interface) via industrial Ethernet. From the operator station the process is operated and the different subsystems are monitored. (See figure 11 for an overview of the configuration.)

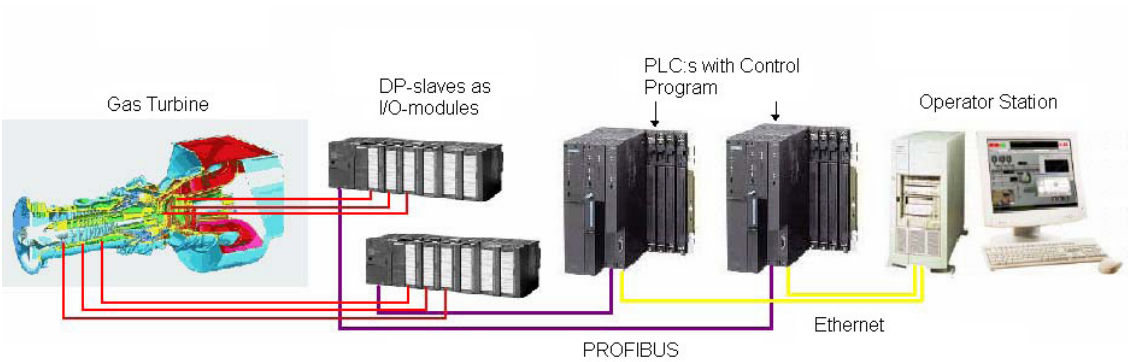

Figure 11 Overview of the gas turbine setup

The automation system (control system) registers the process variables, processes the data according to the instructions in the user program, issues control instructions and set points to the process, supplies the operator station with the data for visualization and registers actions on the operator station and forwards them to the process. [4]

3.2 PLC

PLC:s (Programmable Logic Controllers) are computers used for controlling the process. The control program code is loaded into the CPU: s of the PLC:s.

The PLC:s used are called S7-400. The most important components of the S7-400 are racks, power supply modules (PS), Central Processing Units (CPU:s), memory cards and Communication Processor (CP) for Ethernet and PROFIBUS interface. Racks provide mechanical and electrical connections between the S7-modules. Power Supply Modules convert the line voltage (120/230 VAC or 24 VDC) to the 5 VDC and 24 VDC operating voltages required to power the S7-400. CPU:s execute the user program. Memory cards store the user program and parameters. Communication processors enable data exchange between programmable controllers and/or computers by means of point-to-point connection. [5]

3.3 PCS 7 Software

PCS 7 (Process Control System) is a software suite used for creating the control program, configuring the hardware of the system and creating the user interface (see chapter 6.1 for information about the part that is used for creating operator pictures).

PCS 7 projects are created with the Engineering System (ES). The Engineering System includes the applications SIMATIC manager, HW Config, CFC and SFC Editor.

SIMATIC Manager is the central application that provides access to all other applications that are used to create and modify a PCS 7 project. From the SIMATIC manager HW Config and the CFC and SFC charts are reached for example. The PLC:s are also managed from here. The operating mode can be changed, settings adjusted and the events can be reviewed in the diagnostic buffer for example. [6]

Hardware Config is the configuration of the hardware of a system. Configuration refers to the arranging of racks, modules and distributed I/O racks in a station window (see figure 12). Racks are represented by tables where a specific number of modules can be inserted. Parameters for the modules are set, downloaded to the CPU and transferred to the respective modules. [7]

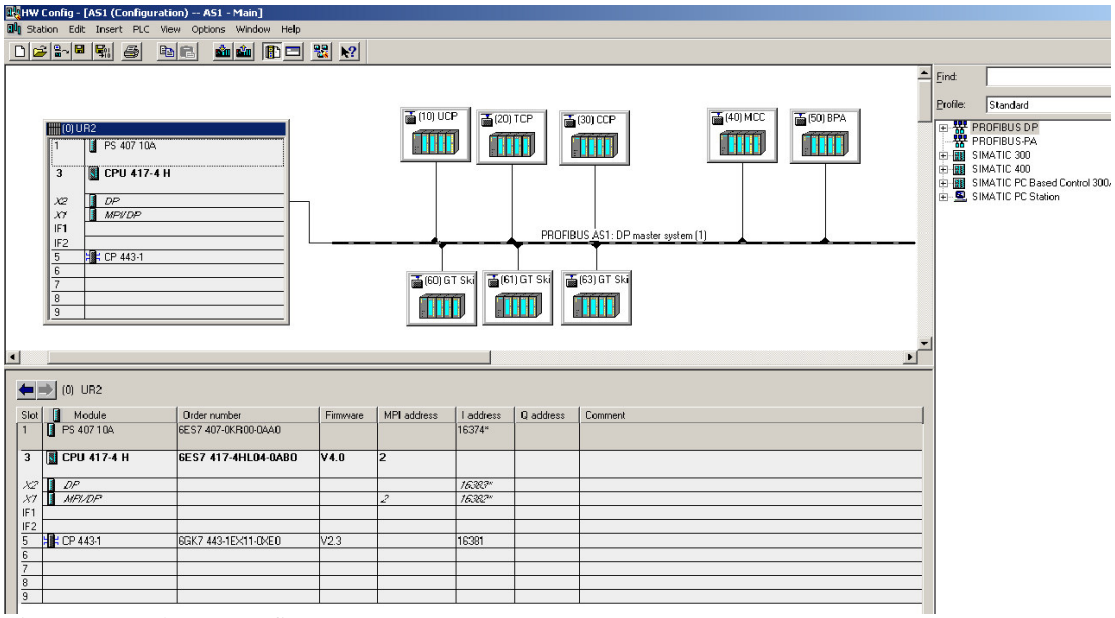

Figure 12 Hardware Config

CFC (Continuous Function Chart) is a graphic editor used to create the software structure of the CPU. Ready-made blocks are dragged to function charts, parameters are assigned to them and they are interconnected (see figure 13). When the required functions are created the code is compiled and downloaded to the CPU:s. [8]

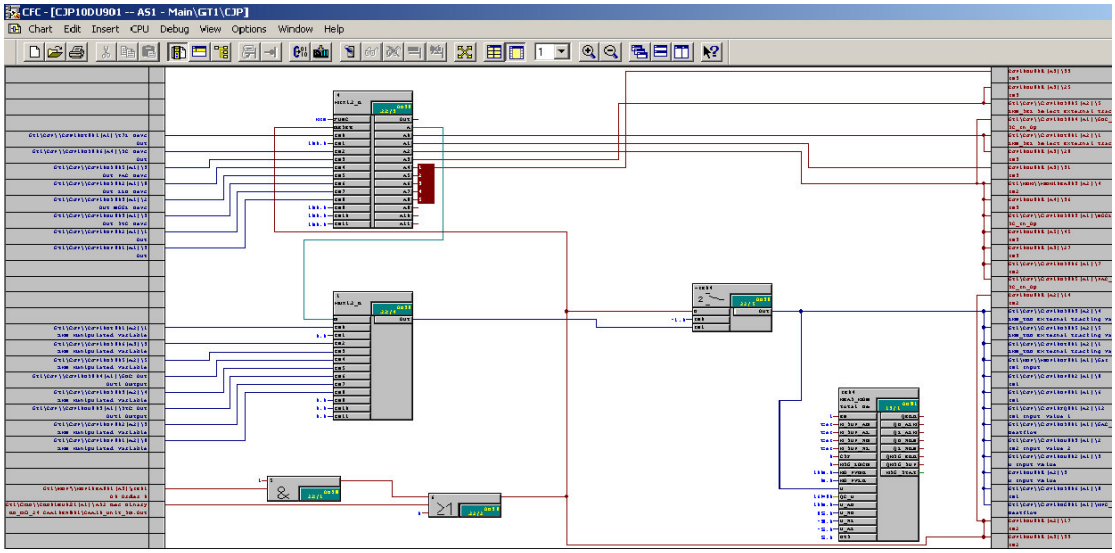

Figure 13 Example of a CFC chart from the control program

SFC (Sequential Function Chart) allows graphic configuration of sequential control systems. The functions created with CFC are controlled by operating and state changes and executed selectively. In a sequential control system the process is broken down to consecutive steps. Each step includes actions to be executed. Step transitions between the steps include conditions that have to be fulfilled to enable passing of control from one step to another (see figure 19). [9]

3.4 PROFIBUS

Conventional signal transmission between sensors/actuators in the field and input/output modules of the control system is implemented via parallel point-to-point connections with copper cables. Fieldbus systems permit digital communication between the automation system and the field devices on a single serial bus cable. This results in large savings on installation costs due to reduction in cabling and input/output hardware and a significantly greater amount of information can be transferred. [10]

PROFIBUS (process field bus) is the world leader among fieldbuses. It can be used in all sectors of the production and process industries for high-speed communication with high measurement accuracy. [10]

PROFIBUS with DP system (Decentralized Peripherals) permits fast communication with intelligent, distributed I/O devices. It provides high data transmission rates and short response times. [10]

3.5 DP-slaves

DP-slaves are hardware process devices (seen in figure 11) that handle the communication between the PLC:s and the sensors, actuators and other types of measurement points in the plant. They are called DP-slaves because they use the PROFIBUS-DP protocol to communicate with one or two masters (PLC:s). The DP-slaves are connected via PROFIBUS to PLC:s. Each slave holds a variable number of I/O:s. [11]

3.6 Fail-Safe

Fail-safe automation systems are used when a fault code could endanger human life, damage the plant or the environment. They detect errors in the process and automatically bring the plant to a safe state when a fault occurs. The safety mechanisms include for example that the configured safety functions are processed twice in different processor sections of the CPU and errors detected in a comparison of the results. Another example is that programming errors such as division by zero and value overflow are intercepted by special fail-safe CFC blocks. [15]

The current control program is partly in a fail-safe configuration. This affects the construction of the simulator in the way that the software used must support fail-safe, which will be discussed later in this thesis.

3.7 The Control Program

The control program is divided into three parts: Main (including the Turbine Governor), Protection (failsafe configuration of critical signals) and Compressor (controls the load).

The turbine governor controls the fuel flow to the gas turbine so that the machine is kept at desired speed and does not run in forbidden operating areas and so that flame-out is avoided. It further controls the split between the primary and the main gas valves and it also controls bleed valve 2, the position of the inlet guide vanes and the combustion chamber bypass. [2]

There are eleven control functions involved in controlling the amount of fuel fed to the gas turbine (STC, NGGL, SC, MPC, GDC, T7L, GAC, LLD, PAC, MPPRC and MMPRC). STC and NGGL are used at start, SC is used at normal operation and the rest are used as limit functions. They all give a desired heat flow to the combustion chamber as output. The main inputs to the controllers are the measured speed of the gas generator and of the power turbine. There are two speed pickups for gas generator speed measurement and two for power turbine. The pickups are measuring the presence of cogs on a cogwheel on the turbine rotor. If over speed is sensed at one of the two pickups, the turbine is tripped. (For more information about trips see the last passage of this chapter). The maximum value from the two pickups is used in the governor. [2]

Only one controller is in operation at a time. A minimum selector selects the one of the control functions (STC, MPC, FLC, GAC, T7L, NGGL, LLD, PAC, MPPRC and MMPRC) that has the lowest desired heat flow as output. The output from that controller and the output from GDC are then compared in a maximum selector, so that the GDC will be selected to be in charge if it requires a higher heat flow than the previously selected controller. That means that the fuel flow is controlled by the channel requiring the smallest valve opening, except if the resulting fuel flow would not be enough to keep the flame alive. In that case the GDC will be in charge (see figure 14). [2]

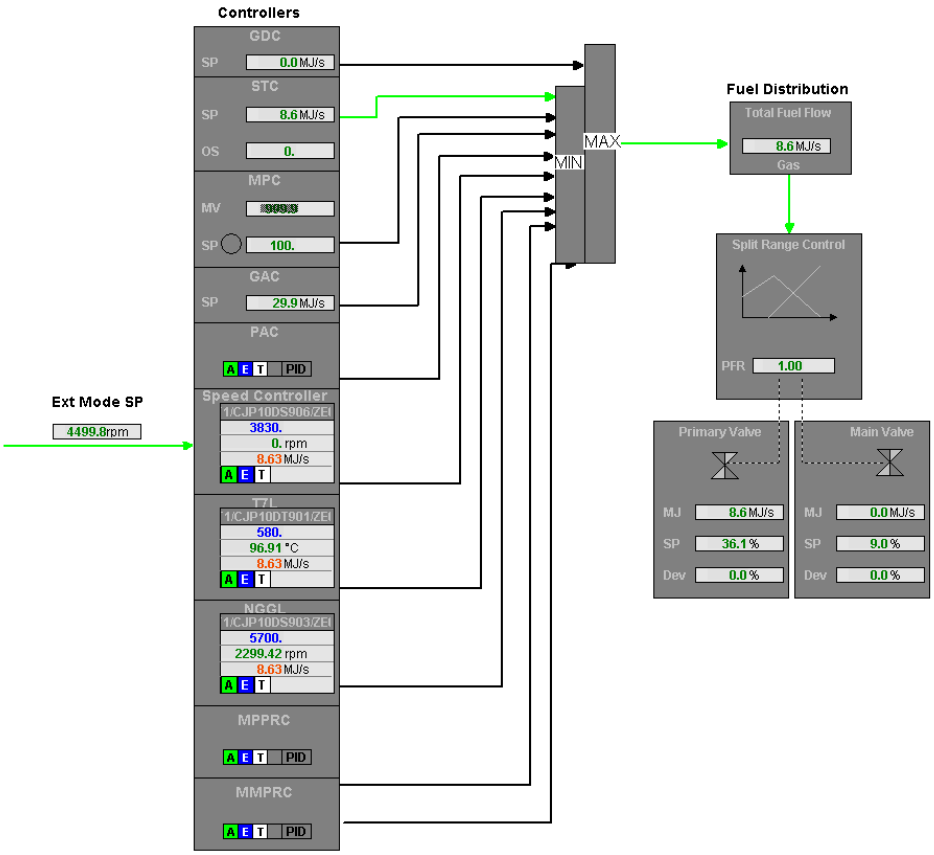

Figure 14 Turbine Governor as shown in the operator picture in WinCC, screen shot from startup

The starting control (STC) keeps the acceleration of the gas generator at a limited rate during startup, thereby preventing thermal stress of the turbine. Before ignition, the engine is running at purge speed (2300 rpm). It is ramped to 2700 rpm by the start motor. A “start kick” adds some extra fuel during ignition, to get ignition to all burners. The fuel flow is during start corrected for ambient temperature, i.e. increasing temperature reduces the amount of fuel. The fuel flow starts to increase with “STC ramp rate 1” and the start motor is released at 5000 rpm. The fuel ramp is changed to “STC ramp rate 2”. If the exhaust temperature reaches a certain point, the fuel flow is kept constant until the temperature has decreased. At 5600 rpm the gas generator speed limiter, NGGL, takes over the fuel control. [12]

The gas generator speed limiter (NGGL) takes over from STC at 5600 rpm and stays in controls until the power turbine reaches minimal speed and SC takes over. NGGL also controls that the maximum allowed speed of the gas generator will not be exceeded during operation, thus preventing the engine from severe damage. [2]

The speed controller (SC) is used for mechanical drive applications (whereas for power generation the frequency and load controller (FLC) is used instead). SC takes over from NGGL shortly before minimal speed of the power turbine and operates until full load is reached and T7L or NGGL takes over. The set point of the controller (the speed of the gas generator) is set by the operator or by the compressor performance controllers.

The maximum servo position control (MPC) is not used during normal operation. It is used for the operator to manually limit the maximum amount of fuel (in MJ/s) fed to the combustion chambers. It is also used as backup control if feedback error occurs. Then the actual desired heat flow becomes the set point for the MPC controller. [2]

The gas generator deceleration control (GDC) keeps the flame alive, by keeping the amount of fuel at minimum demand. This is very important if load rejections occur and call for fuel below the limit to sustain flame. At load rejection all valves are taken to a minimum, but the main gas control valve is brought to a level that sustains the flame. The GDC set point is dependent on the actual normalized gas generator speed. [2]

The exhaust temperature limiter (T7L) limits the exhaust temperature and therefore the maximal load, since the exhaust temperature increases with increasing load. The set point is a function of ambient air temperature, ambient humidity, compressor delivery pressure, exhaust gas pressure and compressor inlet pressure. [2]

The gas generator acceleration control (GAC) limits the acceleration of the gas generator (the derivative of the gas generator speed) during loading and thus preventing the turbine from surging and from transient over temperatures in the gas generator. [2]

The power turbine rotor acceleration control (PAC) controls that the maximum speed and the acceleration of the power turbine are not exceeded, to avoid engine damage. [12]

The loss of load detector (LLD) watches the change of speed of the power turbine. If it exceeds a certain value the LLD will order full closing of the fuel valves, which will activate the GDC. [2]

The maximum primary pressure ratio controller (MPPRC) and the maximum main pressure ratio controller (MMPRC) prevent that the differential pressures over the gas fuel valves get too low. This is done since too low differential pressures can cause the control of the gas flow through the control valve to become unstable. If the pressure after the valve exceeds 87 % of the pressure before the valve the controller will activate and prevent the valve to open any further, thus preventing the differential pressure to get any lower.

When the controller to be in charge has been selected in the turbine governor, its output (the desired heat flow) must be converted into gas control valve orders. First the primary fuel ratio is calculated, that is the share of the desired heat flow that is created by the primary gas fuel valve. Then the desired gas control valve positions for both valves (primary and main) are calculated using the effective area (which is calculated using the desired heat flow and the gas fuel temperature among others) and the pressures over the valves. The deviation of the actual valve positions from the desired are used in PID-regulators, whose outputs are the gas fuel valve orders.

The valve positions of bleed valve 1 and 2 and the guide van positions are functions of the normalized gas generator speed N_{norm}:

$$N_{norm} = N \times \sqrt{\frac{288}{t_2 + 273}}$$

where N is the gas generator speed and t_2 is an average of three measurements of the compressor inlet temperature. [12]

The emission control system should provide stable combustion at minimum CO and NOx emissions in the entire load range. The emission governing should also limit combustor pulsations. The system includes governing of the combustor bypass system, governing of bleed valve 2 and governing of pilot to fuel ratio (PFR). The combustor bypass system is used on part load and bypasses combustor air to the burners to increase the flame temperature. This reduces CO emissions, although NOx emissions are increased. Governing of bleed valve 2 is used on low part loads and recycles compressed air back to the compressor inlet. This increases the combustion air temperature which reduces CO emissions. The governing of pilot to total fuel ratio helps reducing NOx emissions and is used especially on full load. All systems use calculated flame temperature, which can be calculated either using the compressor pressure or the temperature increase in the compressor. [12]

The protection part of the control program causes the gas turbine to trip if it enters a dangerous state. A gas turbine trip interrupts the fuel flow to the gas generator. The gas generator speed decreases until it reaches about 230 rpm. Then the starting system will take over and drive the gas generator until it has cooled down. The causes of a gas turbine trip can be that there is no main flame during operation, there is no pilot flame during start up, any of the gas generator bearing temperatures is high high (i.e. at a dangerously high level), the vibration level at any of the bearing housings is high high, the axial position of the shaft is high high, the bleed valve position is incorrect, the shaft speed is high high or the exhaust temperature is high high. Some trips are immediate and others are delayed. An immediate trip causes immediate unloading, whereas delayed trips implies unloading during 30 or 90 seconds. [1]

4 Simulator Setup

This chapter will explain the simulator system and the hardware and software that are used. An explanation of why the actual solution was chosen and its pros and cons in comparison with other possible solutions will be given.

4.1 Overview

The framework for the simulator, i.e. the software and hardware to be used, was decided during two former final theses. The decision on which solution to use was based on investment costs, the demand of fast enough hardware to enable simulation of a gas turbine and the prerequisite to simulate the PROFIBUS communication between the simulator and the PLC:s (holding the control program). It was also important that the simulator should be easy to modify and develop for future applications.

The chosen solution includes the use of the Siemens software SIMIT in a normal PC as simulator program and a SIMBA I/O card to communicate with the PROFIBUS network, connected to the PLC:s.

In the simulator system, just as in the normal process, there will be an operator station connected with Ethernet to the PLC:s, into which the control program is loaded. In the gas turbine setup there are DP-slaves connected via PROFIBUS to the PLC:s. The DP-slaves handle I/O:s, connected to measurement points in the gas turbine, managing the communication to the control program in the PLC:s. In the simulation system the SIMBA I/O card, connected via PROFIBUS to the PLC:s, is simulating the I/O:s. It will send and receive signals to and from the simulator program and forward them to the control program, which will receive them and act as if they came from the real gas turbine. (See figure 15 for an overview of the simulator setup and the gas turbine setup).

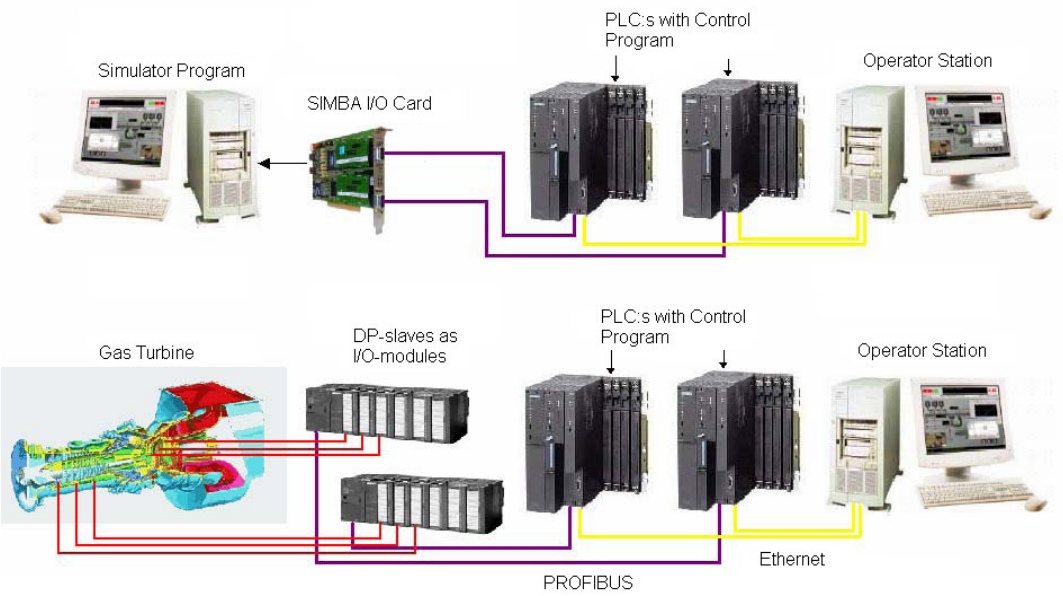

Figure 15 Overview of the simulator and the process

4.2 SIMIT

SIMIT is a modular system of blocks for modeling and simulation similar to MATLAB simulink. It is a real time simulation system, where data is processed and simulated in real time. It is therefore not possible to run tests using data collected from measurements from a real gas turbine, as can be done in simulink. This makes testing more difficult. However, it is possible to save output data from the simulation. The complexity of the components in the SIMIT library is somewhat limited, but new components can be created using the component editor.

4.3 Simba pro PCI card

The SIMBA pro PCI card is fitted into a PCI slot in a PC. One card has two connectors that can be connected to a PROFIBUS network. This means that it can be connected to two PLC:s. If more PLC:s are needed for the control program, another SIMBA pro PCI card will have to be purchased.

SIMBA pro itself can be used to create simulations of simple systems, in which case SIMIT is not necessary. For the gas turbine application, SIMIT, with more advanced components, is more suitable. [11]

4.4 Discussion on the chosen solution

The former final thesis writers revised several different solutions before deciding to use the one with SIMIT and SIMBA pro. One was to have an internal simulator in the same PLC:s as the control program. The programming language would then be the same as for the control program. This means that no additional programming language has to be learned by the users, as has to be done if SIMIT is used. A rough simulator of this type was already in use at Siemens at the time. One disadvantage with an internal simulator is that the simulator is generating extra code to the control program that has to fit into the memory of the PLC:s. Another important disadvantage is that the PROFIBUS communication cannot be tested if the simulator is internal. The disadvantages lead to the choice of an external simulator. [11]

When it was decided to use an external simulator two different configurations were revised. One was the chosen, with SIMIT and SIMBA pro. The other was to use SIMULINK and a PROFIBUS DP I/O card. The later solution was discarded because it was considered too complicated to get the communication between SIMULINK and the PROFIBUS card to work and that it does not support failsafe (see chapter 3.6) neither in SIMULINK nor in the PROFIBUS card. [11]

There are thus several advantages with the chosen solution. The control program does not have to be changed to run it with the simulator. The buss communication can be tested since PROFIBUS is used just as in the real process. It is easy to get the communication between the control program in the PLC:s and the simulator program to work. The SIMBA I/O card supports failsafe. SIMIT did not support failsafe at the time when the work with

this final thesis started. An update that would do so was due to be released in March 2005. [11]

A further discussion on the SIMIT simulator, advantages and disadvantages and problems encountered working with it can be found in the chapters 5.3, 5.4 and 7.2.

5 Construction

This chapter will describe how the simulator was constructed using the hardware, software and theory described in earlier chapters. The interaction between the simulator and the control system will be further explained and problems related to that communication will be discussed.

5.1 Selection of model type

The first decision to make was whether the model should be based on physical relations or on measurement data. A model based on physical relations was considered to be too complicated and not feasible to achieve for one person considered the given time frame. It was also assumed, after discussions with the control system constructors at Siemens, that a model based on measurement data would be good enough for the given purpose. The reasons above settled the choice in favor of the model based on measurement data.

A decision on how to model the relations between the signals had to be made later in the construction process (See Chapter 6.4).

5.2 Extraction and analysis of measurement data

To enable the development of a model of the gas turbine SGT-600, measurement data from a machine of that type was required. Since there was no such data from the Finspång testing site (the SGT-600 turbines had been tested only with liquid fuel there) data had to be gathered from another site. For this purpose the Conditioning Monitor System (the CMS system) proved to be very useful.

The Conditioning Monitor System (CMS) is a tool for collecting, presenting, analyzing and storing process data for extended periods. It enables continuous monitoring and long-time storage of operation parameters of a turbine or a power plant. Real-time and historical data can be presented for example as time trends or x-y plots. There are a large number of standard reports and plots specific for the actual gas turbine type available. It is also possible to make user specific trends and plots. There are tools to export data to Excel files. [13]

The only gas turbine of SGT-600 type available was TVK in Hungary. That meant that the model was going to be based on data collected from that specific gas turbine.

There was about ten months of logged data available form TVK, so certain time intervals had to be chosen to make the amount of data manageable. To get a good picture of the gas turbines functionality some normal starts, normal stops, trips and changes of load were chosen from the logged data. This selection of interesting events was made using Trends in CMS, where a number of selected signals in a chosen time interval can be watched as graphs (see Figure 16). To be able to find the events mentioned above in the data, the active load signal was analyzed in particular, since the load rises from zero during a start, drops to zero with a time delay of about five min-

utes during a normal stop and drops to zero immediately during a trip. There were plenty of those events in the data, although no delayed trips of 30 seconds or 90 seconds were found.

A selection of which signals to use had to be made to reduce the amount of data. It was assumed to be sufficient to analyze the digital signals directly in CMS, while the analog signals had to be transferred to an excel-format to enable further analyzing in MATLAB. All the analog signals that are sent between the control program and the I/O:s of the gas turbine were needed, which resulted in more than 100 signals.

In the CMS system data is logged every second, but not if the signal is constant or linearly changing. This results in data logged with non-regular intervals. Because of that the chosen data had to be interpolated to be of any use for analyzing in MATLAB. This was done using an Excel-program developed by the service-department at Siemens, which is responsible for the CMS system.

To make the construction of the model practicable, simplified relations between the signals had to be found, i.e. relations where one out-signal depends on only one in-signal. This was done by first getting a basic understanding of how a gas turbine works (se chapter 1) and then watching selected signals using Trends in CMS. It was found that the speed of the gas generator affected most of the other measured signals, such as temperatures and pressures. In reality those signals depend on a number of factors, like position of the bleed valves and inlet guide vane, activation of the combustion bypass system and ambient conditions. It was nevertheless observed that the behavior of most signals could be described fairly well using the speed of the gas generator. The speed of the gas generator was found to be dependent on the heat flow, which is dependent on the openings of the gas control valves. (See Figure 16)

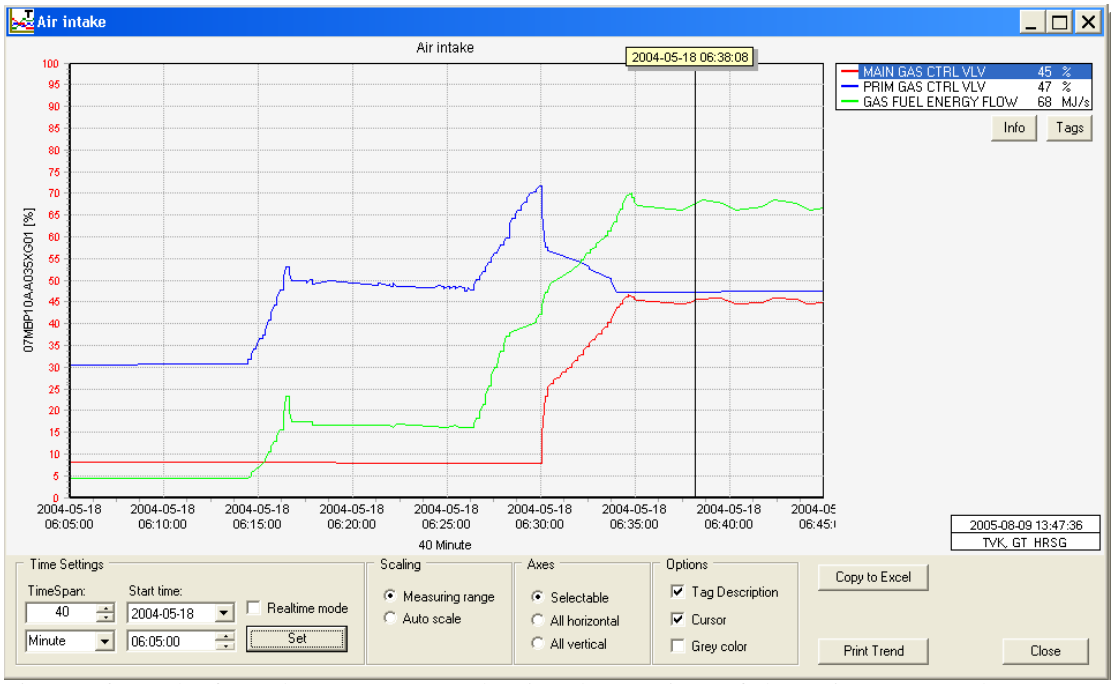

Figure 16 Graphs from the CMS system showing the openings of the main gas control valve (red), the primary gas control valve (blue) and the heat flow (green).

5.3 Building the model in SIMIT

The first approach to model the relations was to use transfer functions and identify parameters in MATLAB System Identification Toolbox. Since there was no ready made transfer function component in SIMIT, a new component had to be developed using the Component Type Editor. After several attempts to model the relations using the System Identification Toolbox it was found that since all relations were more or less non-linear the result was not satisfactory. It was decided not to use transfer functions with identified parameters as the main alternative due to time restrictions. To get a functioning model, with sufficient accuracy, was considered to be the main task. Another possibility would have been to try linearising the relations before identification, but this was, as said earlier, not within the given time restrictions.

Instead of using transfer functions with identified parameters it was decided to use registers to simply map one input-signal to one output-signal. For this purpose there are components in SIMIT called polygons, which are registers with maximum twenty entries that use interpolation between the points. This approach had been taken by the former final thesis writers, who had also developed a MATLAB-function that, given the in- and output-signal, picked out the twenty points that best described the relation between the signals. The chosen points could then be inserted in the SIMIT polygons. Sometimes it was sufficient to pick out the points manually from a plot of the output-signal depending on the input-signal, using the Ginput functionality in MATLAB. To get a dynamic behavior, the polygons were followed by PT-components, which delay and smooth the signal.

$$y_n = y_{n-1} + \frac{T_s}{T_1}(x_n - y_{n-1})$$

The chosen time delay, T_1, corresponds to the time it takes for a step response to reach 0.63. The approach was assumed to result in a model good enough for the given purpose, since a working model for the SGT-700 type had been constructed using the same method.

Binary signals were mainly modeled as answers on orders. For example the *Start Motor on* order signal from the control system would result in the setting of the *Start Motor on* indicator signal from the simulator. Buttons were included to enable manual setting of the signals, thus enabling simulation of faulty signals (See Figure 17). Some signals had to be modeled in a more complex way, for example the Main Flame indicator signal and the Pilot/Main Flame indicator signal (indicating ignition, pilot and main flame). The logic for those signals was developed using information from the operator picture for the fuel system (seen in Figure 8) and the system description.

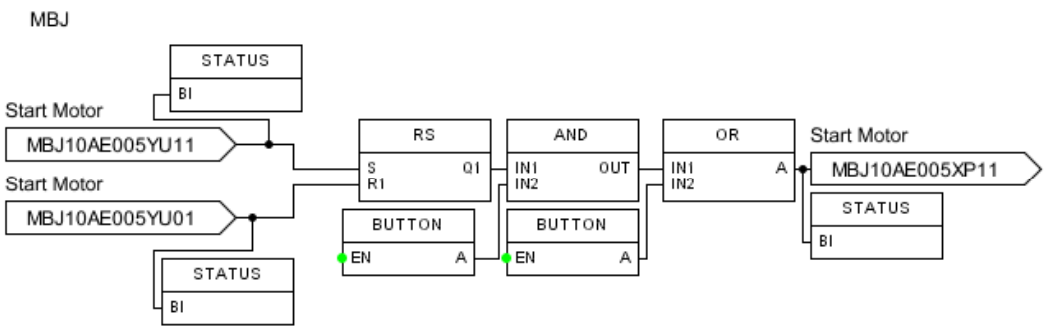

Figure 17 Start motor indicated on or off as answer on order from the control system. Signals are represented by symbolic names. MBJ stands for Starting System, YU11 is a digital on order, YU01 a digital off order and XP11 a digital on indication.

The model was going to be of Mechanical Drive type, which means that a compressor is attached to the power turbine. It was not in the scope of this thesis to model the compressor (see chapter 1.4). It was nevertheless necessary to have a rough model of the load to be able to model the power turbine speed. An available MATLAB model of the compressor was used and translated into SIMIT code.

The signals used in the SIMIT model were grouped in the same way as in the control system, corresponding to the subsystems that a gas turbine consists of. Specific SIMIT diagrams were used to gather all signals and to convert them into global signals. The global signals could then be used on multiple different diagrams in the model. This was done to make the handling and changing of the model easier. The gathering of signals in one place makes it possible to further develop the model to make it possible to turn on and off specific signal groups and thereby adapt the model to the options chosen by the customer for a specific delivery.

Some measured signals from the gas turbine are not transmitted to the control system via PROFIBUS. This applies to the axial displacement (of

gas generator and power turbine) and bearing vibration signals, which are instead transmitted via modbus. Since they are not transmitted via PROFIBUS they are not included in the I/O-cards in the hardware configuration and are consequently not present in the signal list imported to SIMIT. They can therefore not be simulated in the SIMIT-simulator. The signals are nevertheless needed by the control system and must therefore be simulated. This was solved by internal simulation of these signals (in the PLC:s). The signals were modeled on CFC-charts that were added to the control program code. The solution is not ideal, since the purpose is to be able to test the control program in the simulator without changing it. The other possible solution would have been to add the signals to I/O-cards in the hardware configuration to be able to simulate them in SIMIT. This would have requested changes in the hardware configuration as well as in the control program code and was seen a less suitable solution.

During the work with the construction of the model it turned out that some of the measured signals from TVK were apparently not correct. Other signals were not even included in the data. The solution to this problem was to use data from the gas turbine type 10C, which had been used to develop the model of that turbine type. That made it possible to model the signals, even though not in a completely correct way since the data was from another turbine type.

5.4 Communication between the simulator and the control program

The signals that will be used in SIMIT have to be imported from SIMATIC. The output-signals from SIMATIC will be input-signals in SIMIT and vice versa. A new signal list has to be imported for every new SIMIT-project. If the signals are changed in the control program the signal list in SIMIT has to be modified or reimported. In SIMIT a signal list can be imported from SIMATIC to a gateway, from where they can be dragged-and-dropped onto the diagrams. The signals are identified through their addresses on the I/O cards, but they are represented by symbolic names, which make the SIMIT-model easier to grasp (see figure 17).

It proved to be a non-trivial task to import the signal list to SIMIT. A complete list should include addresses, symbolic names and for analog signals ranges and types (see figure 18). In the SIMIT-version used in the beginning a SIMBA-gateway could be created, to which one could import either a list with the addresses and the symbolic names or a list with the addresses, types and ranges, but not a complete list with everything. This was solved by the construction of a JAVA-program that combined the two lists into one complete. At the end of the twenty weeks dedicated to this thesis a new version of SIMIT arrived. This new version was necessary to use for reasons shown later in this chapter. In the new SIMIT-version the SIMBA-gateway had been exchanged for a ProfibusDP-gateway. It was no longer possible to import the types and ranges of the signals to the gateway. They must now be added manually to the signal list. The import of symbolic names did not work properly. The SIMIT support in Germany has been alerted to the problem and is working to solve it.

ProfibusDP

File ?

Settings | Binary In | Binary Out | Analog In | Analog Out | Hardware-View

Symbol	Address	Type	Start	End	N	Line	Slv	Mod	Act	System Remark
MBT10AA010YE01	S000QW512	unipolar	0.0	100.0	1	0	61	4	☑	
MBL05AA005YE01	S000QW514	unipolar	0.0	100.0	1	0	61	4	☑	
MBM10AA010YE01	S000QW516	unipolar	0.0	100.0	1	0	61	4	☑	
MBM10AA015YE01	S000QW518	unipolar	0.0	100.0	1	0	61	4	☑	
MBM10AA020YE01	S000QW520	unipolar	0.0	100.0	1	0	61	5	☑	
MBM10AA025YE01	S000QW522	unipolar	0.0	100.0	1	0	61	5	☑	
MBM10AA030YE01	S000QW524	unipolar	0.0	100.0	1	0	61	5	☑	
QW526	S000QW526	no standardisation			1	0	61	5	☑	
MBJ10AE005YE01	S000QW528	unipolar	0.0	5400.0	1	0	40	4	☑	
MBN10AP005YE01	S000QW530	no standardisation			1	0	40	4	☑	
MBN20AP005YE01	S000QW532	no standardisation			1	0	40	4	☑	
QW534	S000QW534	no standardisation			1	0	40	4	☑	
SAA10AN005YE01	S000QW536	no standardisation			1	0	40	11	☑	
SAG10AN005YE01	S000QW538	no standardisation			1	0	40	11	☑	
QW540	S000QW540	no standardisation			1	0	40	11	☑	
QW542	S000QW542	no standardisation			1	0	40	11	☑	
CFA10CE001YE01	S000QW544	unipolar	0.0	100.0	1	0	30	5	☑	
CFA10CE001YE02	S000QW546	unipolar	0.0	100.0	1	0	30	5	☑	
CFA10CE002YE01	S000QW548	no standardisation			1	0	30	5	☑	
CFA10CE003YE01	S000QW550	no standardisation			1	0	30	5	☑	
MBP10AA035YE01	S000QW552	bipolar	-100.0	100.0	1	0	63	10	☑	
MBP20AA005YE01	S000QW554	bipolar	-100.0	100.0	1	0	63	10	☑	
MBT10AE005YE01	S000QW556	bipolar	-100.0	100.0	1	0	63	10	☑	
MBM10AA005YE01	S000QW558	bipolar	-10.0	10.0	1	0	63	10	☑	

Figure 18 ProfibusDP gateway with signals imported from SIMATIC

When the work with this final thesis started in February 2005, the current version of SIMIT did not support failsafe. Since failsafe signals are used in the control program it is necessary that the simulator software can handle failsafe. A new version of SIMIT that would support failsafe was going to be released in March 2005 but was delayed until July 2005. When the simulator was run (using the SIMIT version that did not support failsafe) together with the control system the failsafe signals sent from SIMIT were received as invalid values in SIMATIC. This caused trips to activate which prevented the gas turbine from being started, which made it impossible to test the model. Attempts were made to work around the problem but the only result was that the signals could be sent properly for a short time, before they turned invalid. The model could therefore not be tested before the new version of SIMIT arrived. After installing the new failsafe-supporting version of SIMIT it seemed to work fine and the failsafe signals were received properly. The remaining problem was that when restarting the simulation after having made changes to and compiled the code, the failsafe signals were invalid again. The CPU:s had to be restarted several times before all the signals would turn valid. It seems that SIMIT still needs some improvement. The new version made it possible (although with a lot of trouble) to test the model, but if the simulator should be used for production purposes, a new more stable version of SIMIT is most certainly required. The SIMIT support has been alerted to the problem and they have localized an error in the IM-card. The suppliers are working on a new update.

Another problem concerning the communication that was discovered during the work with the simulator was that some of the I/O-cards in the hardware configuration did not work, i.e. the signals sent via these cards were not transmitted. Attempts were made to troubleshoot the cards and remove them from the hardware configuration and then replacing them. Since

the attempts were ineffective, another solution had to be found. The malfunctioning cards were moved to other I/O-modules in the hardware configuration. This solved the problem in the way that the signals are now transmitted properly, which is essential to get a functioning simulator.

6 Testing

This chapter will describe the environment used for testing and running the simulator. The sequences used to start the gas turbine process (or the simulator) will be described and the testing process will be accounted for.

6.1 WinCC

WinCC is used for creating operator pictures of the gas turbine process. In this thesis those operator pictures are used to start and stop the simulated gas turbine and to watch the simulated process. (The gas generator speed and the power turbine speed can be watched as trends, as well as temperatures and other signals. Incoming alarms and events are logged in lists. Subsystems of the process can be watched. See chapter 6.3 for more information on how WinCC was used during the testing process.)

WinCC (Windows Control Center) belongs to the OS (Operator Station) Engineering part of the PCS 7 software package. It is used for creating process pictures (configuring PSC 7 OS). OS clients are PC stations that enable control and monitoring of an automation process. They are connected to OS servers, which function as HMI connections to the automation system. The OS client has its own WinCC project and visualizes the process data generated on an OS server. [15]

6.2 Sequences

Sequences are used to start, run and stop the gas turbine. The main sequences are the Unit Sequence, the Turbine Sequence and the Gas Fuel Sequence. There are also sequences for the compressor and for liquid fuel, but they will not be explained further since they are not used to run the simulator (See Chapter 1.4).

The Unit Sequence starts the subsystems that need to be running before the turbine starts, like lubrication oil system and ventilation system. The safety system is reset and checked. It then sends an order to the Turbine Sequence to start the gas turbine and waits for the indication that the turbine is in operation. When the order is received it indicates that the unit is in service and continues running until a stop order is received from the operator station. When it gets the order to stop, the turbine is unloaded and the gas turbine is stopped and set in standby position, from where it can be started again. [14]

The Turbine Sequence (seen in figure 18) is started by order from the Unit Sequence. It starts with purging of the gas generator and sends an order to The Gas Fuel Sequence to start. When purging time has elapsed it ignites the pilot flame and then the main flame. The turbine is accelerated by the start motor until the gas generator reaches 5200 rpm. The turbine is then accelerated further without start motor. When the power turbine speed exeeds 95% of minimum speed, an indication is sent to the Unit Sequence that the turbine is in operation. When a stop order is received from the Unit Sequence an order is sent to the Gas Fuel Sequence to stop, the flame is put

out and the cool down procedure starts. An indication is sent to the Unit Sequence that the turbine has been stopped. [14]

The Gas Fuel Sequence is started by the turbine sequence. It starts with checking the fuel control valves by opening them, closing them and then setting them in start position. If the valve positions deviate too much from the control signals during the test, the start is aborted. After the purge time has elapsed the ventilation valves are checked. Then leak tests are performed on the Shut Off Valves before they are opened. The Shut Off Valves are used for shutting off the fuel supply when the turbine is stopped. The leak tests are executed by closing the valve in question and measuring the pressure after the valve. When the pressure is measured again after a certain time it is not allowed to differ more than a certain limit value from the first measured value. When the gas fuel sequence receives an order from the Turbine Sequence to stop, the Shut Off Valves and the Gas Fuel Isolation Valve are closed and a ventilation valve is opened. (See Figure 8 for an overview of the valves.) [14]

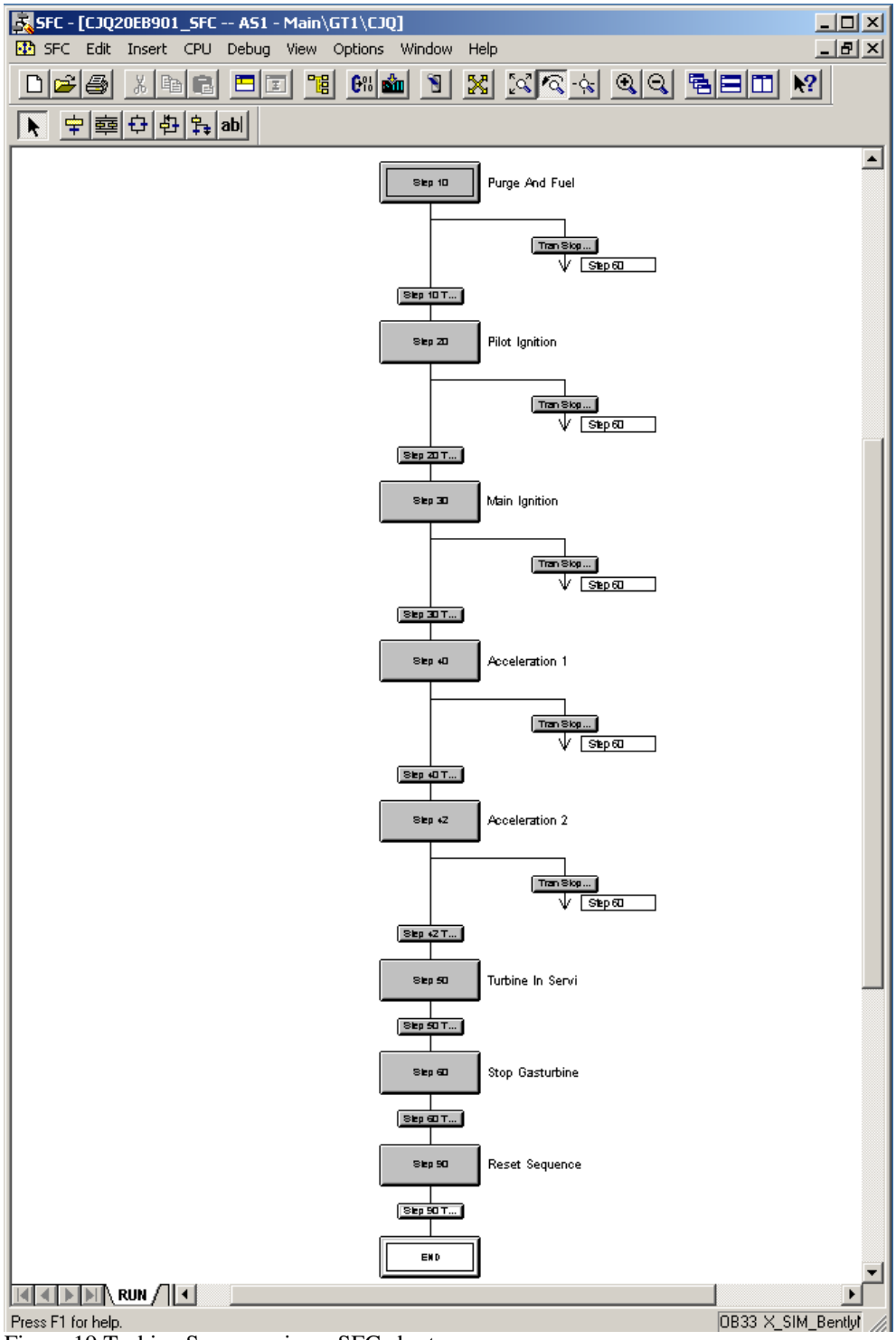

Figure 19 Turbine Sequence in an SFC chart

6.3 Testing the simulator

Testing was performed by starting the sequences from the operator pictures in WinCC. In WinCC overviews of the sequences can be watched, which makes it possible to see in which step the sequence is at the moment. It is also possible to see the conditions that have to be fulfilled to move on to the next step.

Before any testing could be done all conditions had to be removed that were connected to the compressor-part of the control program. (That is the part of the control system that controls the load, which in normal operation is situated in a third PLC. In this simulator it is not used, as explained in chapter 1.4.) This had to be done since the gas turbine otherwise would have stopped at one step waiting for an indication from a CPU that did not exist.

The aim of the testing process was to be able to run through the sequences to reach the state "Turbine in operation". Before that was fulfilled the gas turbine would stop at one step, not able to reach the next, due to different errors, such as errors in the logic of the model or errors that originated in malfunction I/O-cards in the hardware configuration (explained in chapter 6.4). It was then necessary to see which condition was not fulfilled to be able to identify the subsystem or the signal that was not modeled correctly. It was usually necessary to look at the CFC-charts of the subsystem in question, while running the process, to see the changing of state of the signals and systematically try to track the problem backwards chart by chart.

The testing got complicated and was slowed down because of the failsafe problem (discussed in chapter 6.4), since it was necessary to stop the simulation to be able to make changes in the model. It could then take long after restarting the simulation before all failsafe signals were properly transmitted.

After having corrected all found errors in the model and moved some I/O-cards in the hardware configuration, it was possible to get the simulator running, i.e. it would start and accelerate, but not reach "Turbine in operation". The remaining problem was that the speed of the gas generator and the speed of the power turbine changed to fast, which caused the control system to shut down the gas turbine.

PT-blocks were inserted to delay and smooth the speed signals. The delays that were needed to get it slow enough were so big that the system got unstable, i.e. the speed of the gas generator and the speed of the power turbine were fluctuating radically.

In the operator pictures in WinCC the turbine regulator can be watched, i.e. the actual set points and which regulator is in charge at the moment (see figure 13). The instability problem was found out to originate in the compressor (load) model. This could be seen as the fluctuation of the speed signals started when the Speed Controller (that controls the speed of the power turbine) took over during acceleration.

The compressor model was based on an existing MATLAB-model (see chapter 5.3) that had not been constructed to be run in real time. It was too fast, i.e. the speed of the power turbine changed too fast. This caused the Load Loss Detector to activate. To slow it down, PT-blocks had been inserted, as mentioned above. Attempts were made to adjust the time delays so that it would be long enough for the speed to change at a reasonable pace and short enough not to cause instability.

It was found out that it was not sufficient to use time delays in the compressor model. Limitations on how fast the power turbine speed could change were inserted. This together with tuning of the time delays resulted in a reasonable stable model. It was however necessary to make changes in the control program to increase the limit of the power turbine speed when the Load Loss Detector should activate. The model still has some minor instability problems in a certain range of the power turbine speed (further discussed in chapter 7.1).

7 Conclusion

In this chapter the result of the testing of the simulator will be presented. The limitations, advantages and possible usage will be discussed and suggestions for further development will be given.

7.1 Result

A working model for the gas turbine SGT-600 has been developed, based on the one previously developed for SGT-700.

The simulator can be started from WinCC and runs through the sequences much like a real gas turbine. The preparation systems are started, purging is performed, the gas is ignited and the gas generator starts to accelerate. After acceleration to a certain speed the Speed Controller takes over the control. A set point for the power turbine speed is set by the operator, which causes the speed to change and stabilize at the requested level. If the power turbine speed set point is set to a level high enough to result in a dangerously high exhaust gas temperature, the T7 limiter will take over the control and stabilize the exhaust gas temperature at a safe level. At shutting down the speed of the gas generator and the exhaust gas temperature are slowly decreasing. The temperature delays are much shorter than in reality, since there is no point in watching the simulator cooling off for hours.

The model has some minor instability problems in a certain range of the power turbine speed. The problem origins in the compressor model, as explained in chapter 6.3. To model the compressor (load) was however not in the scope of this thesis. Since a better model of the compressor is under development in another thesis, it was decided not to put more effort into fine-tuning the old model.

The model is realistic in the way that it is based on measurement data from a real gas turbine. It has nevertheless been adjusted to work with the control system. Time delays have been adjusted to get a stable model and do not always correspond to real measured time delays. It can seem strange to adapt the model to the control system, since the purpose of the model is to test the control system. But, since the control system had been tested (and the control parameters adjusted) against a real gas turbine it was assumed that if the model was tuned to fit the control system it would be close enough to the real turbine.

The combination of polygons and PT-blocks seem to work sufficiently to model the relations between the signals. The polygons handle non-linear relations and the PT-blocks add dynamics to the system. The main limitation of the model is instead the simplified relations between the signals, i.e. the output signals depend in reality on many more input signals than modeled in this thesis. To include all affecting input signals, it would be necessary to develop a mathematical model based on physical relations. This was however out of the scope of this thesis.

7.2 Discussion

The simulator can be used for testing the standard of the control system or to test the control system when adapting it to a specific delivery. To further serve the latter purpose it would be good to develop option packages to enable quick adjustment of the simulator to the unique composition of each gas turbine.

It is also specially suited for educational purposes, e.g. to teach the customers about gas turbine functionality, mainly because of the operator pictures (that have no equivalence in the SIMATIC-simulator, i.e. the simulator where the simulation takes place internally in the PLC:s). The operator pictures in SIMIT make it easy to control and watch the simulator and also to simulate different operating scenarios. If this should be done with the SIMATIC-simulator one has to change parameters in the code, which is not user friendly. The staff of the customer training department asked for a simulator that would be easy to carry around. This is not the case with this simulator, since the two PLC:s are quite heavy. The problem can however not be solved at the moment, since PLC:s must be used. All other simulator setups would have the same problem.

It is at the time of this writing not quite clear if and how the simulator will be used at SIEMENS Industrial Turbomachinery. It is clear that a simulator is needed, but not if it should be this one in SIMIT, the internal SIMATIC-simulator or yet another type. It is up to the heads of the departments concerned to decide which way to go. This final thesis has contributed with evaluation of the SIMIT software which will make the decision easier. The functionality of the model itself is similar to the one in SIMATIC.

The advantages of the SIMIT-simulator compared to the SIMATIC-simulator are:

- It is possible to test the PROFIBUS communication.
- It does not require extra memory of the CPU. This can be extra relevant in the future, since there are plans to fit the whole control program into one CPU.
- It is possible to make operator pictures in SIMIT, which can be used to control the simulator. This would be very useful if the simulator should be used for educational purposes.

The disadvantages are:

- The staff has to learn to use SIMIT, which is at the moment not used.
- The investment cost for SIMIT-licenses is high.
- SIMIT needs further development, which can cause problems when using it (as seen in this thesis).

- Even though most blocks are the same in SIMIT as in SIMATIC, user-made blocks in SIMATIC have to be made in SIMIT as well, which causes extra work.

7.3 Further Development

The problem with the malfunctioning I/O-cards mentioned in chapter 6.4 must be solved in a proper way, without moving the I/O-cards. Since the hardware configuration must not be changed, it is essential to get the I/O-cards to work in the I/O-modules where they are supposed to be. This should best be done by the people who have set up the hardware configuration.

A more realistic model of the compressor needs to be developed, to replace the rough model used in this simulator. (This is being done in another thesis at the department of Oil and Gas.) A generator model also needs to be developed, to enable simulation of Power Generation.

Option packages need to be developed, to enable adaptation of the simulator to specific deliveries. This has been prepared in this thesis by grouping and gathering of the signals on specific charts.

To extend the model to be valid for dual fuel, a model for the liquid fuel system needs to be developed. This can however not be done before the control system code for the liquid fuel system is ready.

It is necessary to update SIMIT to get better failsafe functionality and to be able to import the signal list properly. SIMIT support in Germany has been informed of the problems described in this thesis and is working to solve them.

Operator pictures needs to be developed in SIMIT if the simulator is to be used for educational purposes.

Simulators for other gas turbine types can be developed in the same way as this one. The simulator for SGT-700 needs to be modified to work with the new turbine governor and the new version of SIMIT. It might be useful to merge the two simulators into one, with a button for the user to switch between them.

References

1) Siemens internal documentation, GT10 Basic Training
2) Siemens internal documentation, System Description GT10B2
3) Siemens, (2005), http://www.powergeneration.siemens.com/en/oilgas/drives/gt/sgt600/index.cfm (Accessed 2005-08-11)
4) Siemens, (2005), PCS 7 V6.1 Engineering System, http://cache.automation.siemens.com/dnl/jk1NTM3MQAA_21407752_HB/PHB_PCS7_e.pdf (Accessed 2005-06-15)
5) Siemens, (2004), Automation System S7-400 Hardware and Installation, http://cache.automation.siemens.com/dnl/TQ1MTg5OQAA_1117849_HB/424ish_e.pdf (Accessed 2005-06-15)
6) Siemens, (2005), PCS 7 V6.1 Getting Started - Part 1, http://cache.automation.siemens.com/dnl/TYxNjcwNwAA_21405992_HB/ps7gs1b_e.pdf (Accessed 2005-06-15)
7) Siemens, (2004), Configuring Hardware and Communication Connections STEP 7, http://cache.automation.siemens.com/dnl/DM0NTMzMQAA_18652631_HB/S7hwV53_e.pdf (Accessed 2005-06-15)
8) Siemens, (2005), CFC for S7, http://cache.automation.siemens.com/dnl/TU1NTA2NwAA_21401430_HB/CFC_For_S7_e.pdf (Accessed 2005-06-15)
9) Siemens, (2005), SFC for S7, http://cache.automation.siemens.com/dnl/zg0NTY3OQAA_21400488_HB/s7sfcs7b_e.pdf (Accessed 2005-06-15)
10) Siemens, (2004), PROFIBUS Product Brief
11) Lindholm and Klang, (2005), Modelling and simulation of a gas turbine, LITH-ITN-ED-EX--05/009--SE
12) Siemens internal documentation, (2004), GT10B2 engine control specification
13) Siemens internal documentation, (2004), CMS User Guide, Version 2.2
14) Alstom internal documentation, (2003), GT10 Mechanical Drive Standard Sequence diagram
15) Siemens internal SIMATIC documentation

Figure 1 is from http://www.powergeneration.siemens.com/en/oilgas/drives/gt/sgt600/index.cfm (Accessed 2005-08-11, modified by the author)
Figure 2, 3, 4, 5, 7, 9, 10 are from [1]
Figure 11, 15 are from an internal Siemens document (modified by the author)
Figure 8, 12, 13, 14, 16, 17, 18, 19 are screen shots from the simulator or the control system
Figure 6 is from http://www.powergeneration.siemens.com/en/oilgas/drives/gt/sgt600/emissiontech/index.cfm (Accessed 2005-08-12)