Coordinated operation and control of converter based substations

Structures, modules and real-time operation

Peter Dahlin

Department of Industrial Electrical Engineering and Automation
Lund University

Abstract

This master thesis project, commissioned by the University of Lund (Sweden), deals with the growing number of stochastic generators and how they will be handled in the future. A small-distributed network based on an information system and converter stations have been investigated in booth a simulation environment and in real time tests. The tests have been made to demonstrate that the concept of “plug and produce” is working and that it will keep the network in a stable condition at all times.

Contents

1 Introduction

This project presents and verifies principal design structures for scalable distributed power systems (DPS) were focus lays on operation & information flow and converter stations enabling “plug and produce” capabilities. The purpose is a general structure that should be appropriate independent of system capacity. The main needs in future market and power system requires a feasible solution that is adequate in scalability and flexibility for managing a high degree of automated functionality. The basic solution is to automate and downscale the common centralized structure for operating a power system to several DPSs that interact by well-defined hardware and software interfaces. This involves full decentralization that brings an increasing number of automated decision-makers decreasing the size of problems to be solved [1]

1.1 Background

Next coming large extensions of renewable energy sources (RESs) and at the same time high expectation in increased reliability will bring high demands on the power systems. RESs are likely to interact differently with the power system in the future dependent on capacity and localizations. The power systems would probably fully integrate RESs into energy and ancillary service markets. This will gradually change the power system to systems with highly stochastic behaviour in both production and consumption. With the aim of handling these coming changes and at the same time strive for properties enabling “plug and produce” facilities the system complexity will increase further. This work is an outcome of that future believes will bring a high number of converter stations that need to have a certain degree of coordination to meet coming power system demands.

1.2 Motivation

The expansion of renewable energy sources, the higher demands for interconnections between countries and the decentralized energy market will put higher demand on the power systems. The increasing number of stochastic energy sources, such as wind power plants, brings higher demand on controllability on the system.
The demands on reliability might lead to introduction of distributed systems (DPSs) with the possibility to separate specific control areas to be managed by their own. This will require bi-directional power flow with the possibility to be turned on and off fast.
The power system of the future will probably contain a great number of power converters based on transistor technology. These will be placed as an interface between new constructions and a power system and between different power systems.
Depending on design and structure, these power converter stations will bring a higher degree of controllability and thereby a possibility to higher reliability [2].

1.2.1 Increased reliability and survivability

Future power system structures might be a fully decentralized power system with a large number of small self-supporting control areas. The generators and loads should be able to connect and disconnect to the system. As they do, the power system should automatically adapt to the new circumstances and make the best out of each situation. New routines for the service “plug and produce” have to be introduced to make this possible.
To be able to sell or buy energy these small control areas could then be connected to each other in a possibly complex network. The amount of energy that can be transmitted is determined of how much the sending system can generate and how much the receiving one can take care of. To make it easier for the consumers to know from where they can buy the energy, all of the small power systems have to be logged on to a energy market where they declare how much they can sell/buy and to which prize.

1.2.2 Plug and produce

With the concept plug and produce come that producers or consumers that are to be connected to the power system should not have to be running some special installation programs to be able to log on the system. All they have to do is to connect to the system and start its control program and everything should adapt to the new unit and start to use it. To make this reality, a fast communication structure has to be implemented and what information that is to be transmitted have to well defined. All the units in a power system area have to be controlled from a common point. This point in the system takes care of all of the decision making of which unit that is supposed to do what.

1.3 Objectives

The purpose of the project is to build and test a distributed power system based on an information structure developed at the university. This information system has a high degree in modularity, scalability and adaptation and enables to handle the concept of plug and produce. The work should also demonstrate that it is possible to build a controllable and dynamical power system based on that information structure.
This power system should be tested both in the simulation environment and on real power converter stations. The power system should at all times remain stable and should be able to handle situations when a new unit is added to the system or if a unit disappear.

2 DPS Architecture

This chapter describes a power system and information structure that downscale the common centralised structure for operating a power system to several subsystems, which interact by well-defined hardware and software interfaces.

2.1 Design Criteria and Structure

Restructuring the power system with “the ideal” solution will be close to impossible due to the constant changes in demands and new possibilities. However, by designing a flexible and adaptable structure, it will provide means for extended durability. An important ambition in constructing future power systems should be attaining something close to “plug and produce” capabilities. Independent of connecting a generation or load unit, the system should automatically configure and, if appropriate, prepare the unit for operation. The unit responsible should also automatically be able to participate in the market offering purchase or sale bids. As complexity increases, a traditional solution is to divide a problem into sub problems with well-defined interfaces. This is also most likely what future power system will undergo by dividing the system in several subsystems or DPSs, primarily for reliability reasons, as shown in Figure 2.1.
The DPSs should have autonomic functionality and inform adjacent DPSs or detached market management organisations through primarily two communication interfaces. The *trading communication interface* (TCI), which includes several signals for managing the purchases and sales towards the market. The *operational communication interface* (OCI) that contains all necessary operation and information signals for real-time control.

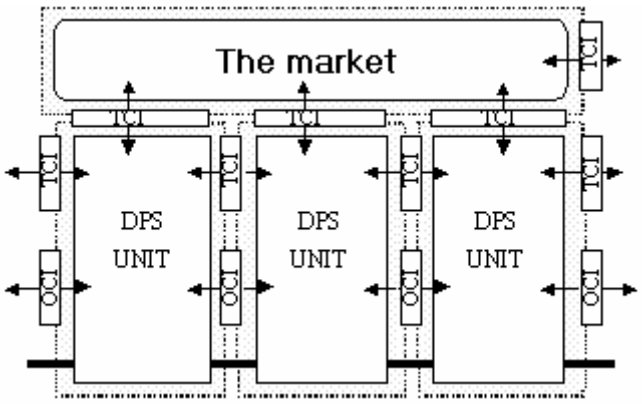


Figure 2.1. General market and power systems divided by distributed power systems interacting the markets

It is mainly the market that sets the prize for how much each DPS can buy or sell power. But if the market for some reasons can not be contactable by the DPS, all the DPS can negotiate for how much and to what prize the DPS can deliver power to there neighbor. The OCI signals is sent between the different DPSs so that they know how much power there adjacent DPSs might need in the future and start to prepare for that.

By downscale a power system as this; it will be more controllable and reliable in total. If there would be a failure in one of the DPSs that system would immediately be shut off. The failure would then not affect any of the other DPSs.
This report will not discuss the TCI signals any further but will focus a little bit more on the OCI signals [3].

2.2 Unit classifications

All DPS units have an internal structure including hardware and software with well-defined interfaces and different automated intelligent decision-making dependent on unit potential. Units may be constructed by other units and are easily interconnected by uniformed interfaces.

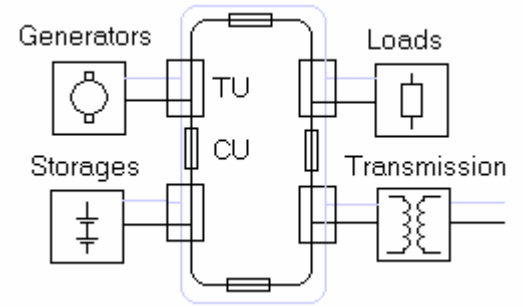


Figure 2.2. General DPS unit, information line (grey), power line (black)

The principal design of a DPS unit is constructed as shown in Figure 2.2 and includes the following units:

- *Connecting units* (CU) can be anything from small cables to long transmission lines.
- *Terminal units* (TU), which normally are substations in the power system but could also be intelligent outlets in smaller residential applications. The TUs are connected in series with the CUs as a ring grid where at least one TU is interconnected to the markets
- *Generator units* (GU) can be anything from a nuclear power plant, to a wind power plant. The thing that unites them is that they only can generate power
- *Load units* (LU) represents all consumers such as industries, residences etc. What they have in common is that they only can consume power.
- *Storage units* (SU) might be pumped hydro, hydrogen, SMES, super capacitors etc., which are for both long- and short-term usage. These units can work either as generator or as load depending if the storage is charging or discharging.
- *Tramsmission Exchange* units (EU) can be converters, transformers or breakers and are used to import or export power with another DPSs.

2.3 Interactions

The DPS base structure includes several TUs and CUs. The TUs are meant to be such general that the principal structure could be used anywhere from a common substation to a station that is capable to take care of the automated operation of the DPS.
All of the Terminal units are equipped with a production management program that is supposed to control all of the other units within the DPS. Every TU has a priority number and it is the TU with the highest priority that will take care of the automated operation of the system. The other TU:s will only work as ordinary substations. If the TU that is controlling the system for any reason would break or lose the contact with the others, another TU will take over and control the system instead.

Every producing or consuming unit in the DPS has a control program, production exchange, which continuously reports to the corresponding TU all the necessary information about how the unit is working and how large capacity it has.
It is also this program that receives the command signals, which tells the unit how it should operate. The command signals are checked so that they are appropriate to the unit before they are sent to the actual unit [3].

3 Management

To build a power system that is described above, there have to be a very well defined structure around it. The system has to be able to know what all of the units in it are doing and what the limits for the system are. There fore there have to be a good communication structure and communication signals.
Modules for autoimmunization have to be developed and that is what this chapter describes.

3.1 Communication Structure

An automated power system is very much dependent on a well defined communication system. Disturbances can occur at any time and they have to be taken care of immediately.
It is therefore important that the units within the DPS have a great deal of coordination and cooperation to make it all work.
The speed in the communication network is crucial to make the system dynamical and special consideration have been taken to make it as fast as possible.

To be able to coordinate the converters with each other, an automation module is needed. That automation module collects all the values from the converters and returns control signals to the converters. The communication system used in this project is built up according to Figure 3.1.

In this communication system there is only one module that takes care of the automation and that is the production management (PM) module. A system like this is quite vulnerable to disturbances. If the machine that takes care of the PM module would break or lose its contact to the other units the system would collapse. It would have been much

better if each converter had its own PM that could take over if the one in charge would break.
But this kind of setup is enough for the tests as we want do in this project.

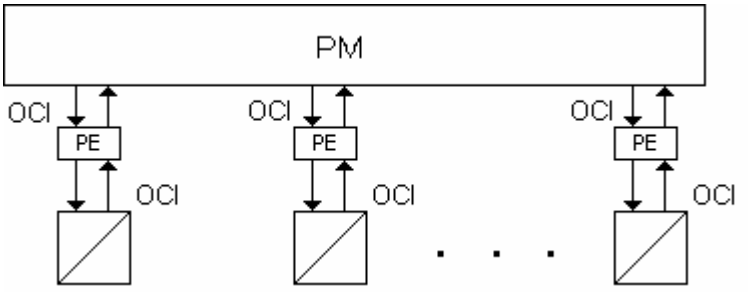


Figure 3.1. Overview of the communication system.

The operational communication interface (OCI) are sent throughout the system. These OCI are used between the PE block and all of the other blocks that are around it. These OCI are realized by a vector with the length of 37 signals. This vector contains all the information needed to run a fully automated power system.
The OCI signals that come from the PM are signals that tell the units in how it should operate. Before the signals reach the resource unit, they first have to go through the PE module. This module checks the signals to make sure that they do not exceed the capability of the unit.
The OCI signals that comes from the unit contains information about how the unit actually operates and how much power it consumes/generates. It also contains information about the capabilities of the units. An explanation of the OSI signal will be discussed later in this chapter.

The system is supposed to be of the type plug and produce. Meaning that there is not going to be a limit number of resource units connected to the PM block. The new unit should not have to run any special installation programs to be able to connect to the system and are also able to connect or disconnect at any time

3.1.1 OCI Signals

The OCI signals that are sent around in the system contain a vector with the size of 37 different values that describes everything in the system. All of the 37 signals in the vector are not used in this work but those used and their function will be described.

- The first number in the signal is not used here.
- The second one informs the system to which destination that this OCI is being sent. In this project this number will represent the last 2 numbers in the TCP/IP number on the receiving computer.
- The third signal informs the receiving computer from where this OCI is coming. It is later being used to respond to the right address.

The following 5 places in the vector are not used in this project.

The places from index 9 to 13 are used by the production management to set the different units in predefined mode of operation to work in.

- The command priority signal informs the units what priority that the DPS has. This information is only used by the exchange unit. It is using the signal to decide from which side that it will be controlled.
- Command = {“stop”; “ready”; “start”;} is the basic operational signal for start and stop of the units. The “stop” command is assign when the unit is supposed to be off. The “ready” command is used to prepare for a faster start and also as a check for no failure before the next coming “start”-command. In “ready”, the unit is in a hot standby mode. The “start” tells the unit to be in its operational mode.
- Govern = {“min”; “max”; “power control”; “voltage”; “droop”} specifies the different modes of control. The “min” mode is used to assign a generating unit in some type of ready state (idle running) but not completely out of order. The “max” mode always strives for the unit’s maximum generating output. If operating none stochastic generating units it is the same as the nominal output power. When an external power set point is used the mode is called “power control”, se the Set Point signal explanation below, where “voltage control” and “droop control” mode can be selected in case of low voltage on the grid.
- Set Point, 1 and 2, has no significant meaning unless the Govern signal is set to “power control”, “voltage”, or “droop”. Then it is used as a set point received from higher unit levels.

The places from index 14 to 18 are used by the production exchange to tell the rest of the units in the system what type of resource that it is. But it is only the signals on place 14 and 15 that is used in this project.

- Rtype = {“Generator”, “Load”, “Storage”, “Exchange”} each unit type is declared by a number from 3 to 6 and the number correspond to the order in which they are stated above.
- Resource priority - this signal is used to tell the system which priority that the unit has in comparison with other resource units.

The rest of the signals are used to inform the production management about the status and data from each unit

- Transaction status- not used in this project
- Command status – returns the value of which command mode as the unit is working.
- Govern status – returns the value of which govern mode as the unit is working.
- Direction status – returns the value of which direction the power flows has in the unit, importing or exporting.

The rest of the signals are data of the units' maximal limits and how much it currently is producing.

- ENom – Informs the system what the unit's nominal energy level is, only used by the storage unit.
- ECur – Informs the system how much energy the unit currently have in its storage, only used by the storage unit.
- EMax – Informs the system what the unit's Maximal energy level is, used by the storage unit and exchange unit.
- EMin – Informs the system what the unit's minimal energy level is, used by the storage unit and exchange unit.
- PNom – Informs the system what the unit's nominal power level is.
- PCur – Informs the system how much power the unit currently produces/consumes.
- PMax – Informs the system what the unit's maximal power level is.
- PMin – Informs the system what the unit's minimal power level is.
- PAux – Informs the system how much power that is needed to start up the unit.
- dpCur – Informs the system what the units current ramping rat is.
- dpMax – Informs the system what the units maximal ramping rate is.
- dpMin – Informs the system what the units minimal ramping rate is.
- VNom – Informs the system about what voltage level the unit want to work in.
- VCur – Informs the system about what voltage the unit has.

3.4 Control functions

To have a DPS running and consider it as a reliable system, the voltage level has to be stable in a DC case. Therefore one or more sources attached to it, need the ability to control the voltage.
There are mainly two ways to control the voltage, which belongs to the primary control.

The first is a PI-controller that will make sure that the voltage in the system will keep the voltage exactly at the voltage set point. This control function is good when there is only one unit that takes care of the voltage control part or if the units are far away from each other.
But if there are several units working in this controlling mode and they are close to each other, there is a risk that the integration parts in the controllers can start to compete with each other.

The second voltage control method is called droop control and has only a proportional part in its controller. Because of this, there will always be a stationary error in the voltage but it will stay stable at all times with the assumption that the capacity is enough.

If the system controlled is an AC-system, there is included a frequency function as well among the primary controlling functions. This project will focus on a DC-system and do not have to consider any frequencies.

The secondary control function considers the power control. But there are a few different types of control functions and they are.

- Minimal power control. This function is made to be used by the units that are not ably to regulate its power at 0W. This is the case in a nuclear power plant for example that have to produce a lot of power at a minimal when it is in its on state.
- Maximal power control. This is function telling the unit to always operate as hard as possible. Useful for wind power plants and solar power plants.
- Power control. This service is used for all power limits that are between the minimal and maximal power limit for a unit.

3.5 Strategy

There are mainly four different types of power units, generator units, load units, storage units or exchange units that are used as either sources or sinks. The order of in which the units are started and at what service they are started in depends of what strategy the operator of the DPS has.

For this work is the main objective to try to support the load units with as much power as they needs at all time. If generating units can't provide enough power to support the load unit completely the loads are not allowed to use more power then the system can provide.

The generating units are only capable to generate power and they are the main source in the DPS. It is these units that have the highest priority when it comes to support the load with power. If the generator can provide more power then what the loads needs, they can store energy in the storage units to have in reserve in cases when the generators are weak. They can also be used to provide power to support adjacent DPSs with power when they ask for it.

In this work the generators will try to fill up the storage units before the DPS is allowed to export any power. The reason for that is that the system will try to have the best possible opportunity to support its loads on there own and full storages will give that opportunity.
It is only when both the load units and the storage units are satisfied as the DPS is allowed to export power.

If however the power from the generators isn't enough the system can ask an adjacent DPS for assistance. The remaining amount of power that the DPS needs will then be imported from this DPS in order to not let the loads be short of power. If the storages have to low energy level, they will also be charged up but only if the load units have all the power requested.

The storage units can be used to store energy which later can be used as support if the generators are weak for the moment or the loads have a temporary peak in there demands.

They are only used as a support when the generators and exchange units are unable to deliver enough power for the loads. But they are charged at any time as the capacity of the generators are higher the demands of the loads.

To have a well-balanced power-system there have to be one/some generating units that is/are controlling the voltage, primary control, in the system.
These units' senses when there are units that are consuming power in the DPS and to automatically add power to the system in order to keep the voltage on the right level. The response for disturbances will therefore be taken care of very fast from these units.
But the control of how much power that each voltage controlling unit is supporting will get lost since each unit provides enough power to keep the voltage stable at the point where they are connected.
The idea for this project is to have enough primary controlling units working in droop mode and the rest of the generating units should work as power controllers.
A picture to illustrate how much power as the primary controllers should produce is shown in Figure 3.2.

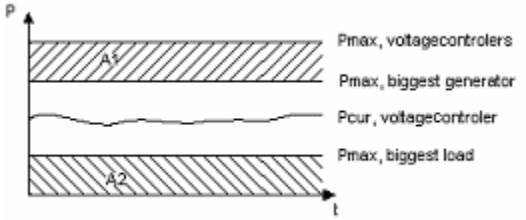


Figure 3.2. Picture of the power produced by the primary controlling units.

The power produced by these units is never allowed to enter the upper safety zone A1. This is because if the biggest generating unit suddenly would break. Then will the primary controlling units be able to quickly compensate for that loss of power by entering the zone A1. The system will then slowly adjust to the new circumstances by adding another generating unit in power control mode or reducing the set point for the loads. The power production of the primary controlling units will then go back into the allowed area. The security zone A2 is working in a similar way. But this zone is there to protect the system from sudden losses of load units.
The DPS is trying to keep the power produced by the primary controllers in the middle of these safety zones as much as possible but it is allowed to touch the lines of the safety zones during normal operation as well.
Then the sum of the safety zones are about as much as the maximal power productions of the primary controlling units and the zone, where these units are allowed to be in, will be very small. In these cases another generating unit will be started in droop mode and the margin for these units will become higher. In the cases when there is no more generating unit available to start up the system, a generating unit will change from power control mode to droop mode and solve the problem in that way.

The order for different unit types to be used for primary control is:

- Generator units.
- Storage units.
- Exchange units.

If there are any generators in the system that is not yet used, these ones are the first used for primary control. If there are no more generators available the storages are the second ones used for this purpose. Finally when all other options are used the exchange units are used for the primary control. The reason for having higher priority on the storages then on the exchange units is that the system becomes more flexible if the exchange units are working in power control mode as much as possible. The adjacent DPSs connected via the exchange units are also the last ones disturbed to avoid any possible cascade effects. If the exchange units are used to control the voltage on one power system, the option for voltage control for the other one disappears. So to maintain a high flexibility the exchange units is kept as power controllers for the longest time possible.
For dynamical purposes the system can automatically change a unit from power control to voltage control when needed. This is done when all of the free generating units are used and the priority, for in which order the unit types are changed, is the same one that is stated above.

The strategy for the start up is however a little bit different. The system will start up on only one generating unit, in droop mode, and when that unit almost has reached its maximal capacity another generating unit. That second generating unit is always started in droop mode as well, because of safety reasons. If any of the droop mode units breaks the other one will always be able to keep the voltage stable in the system.

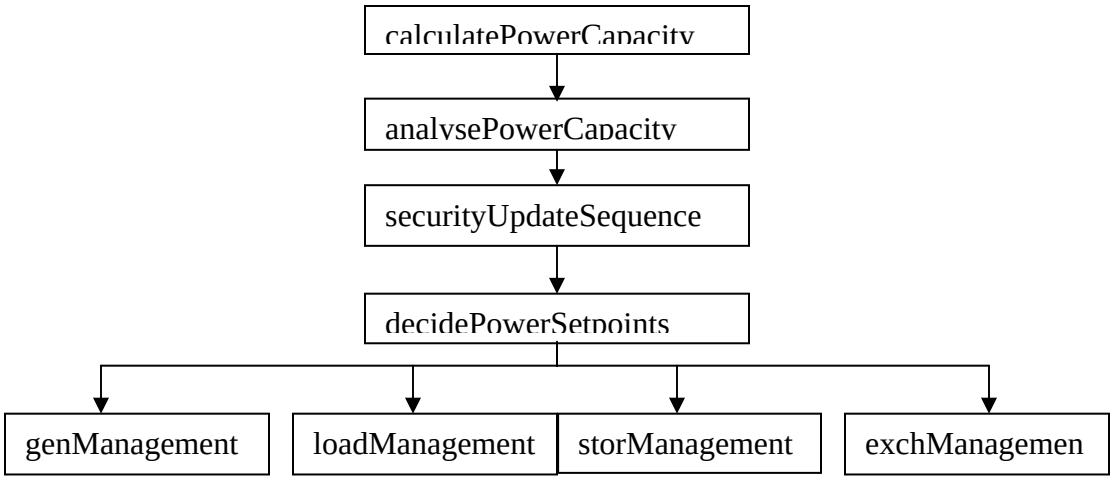
3.6 Module Structure

To make the DPS that is described above to work, a great deal of programming is necessary. The base for the project is a communication system, which is developed in the university.
On top of that system a DPS is created and the hart of that DPS is the production management module as it decides what each unit should do.
Another very important program is the production exchange module that picks out and checks the command signals that goes down to the unit that it is connected to.
Explanations of how these two works will here be discussed.

3.6.1 Production management

This is the hart of the power system and it is from this module that the control signals to all of the units in the system are created. To be able to maintain a high standard of safety to the power system, a safety routine is included in this module. The safety routine has its focus on the voltage level and different actions are being made if the voltage level is dropping too much.

The main sequence for this program is divided into several stand alone steppes, which are:



In the calculatePowerCapacity step all of the data concerning how much power each type of unit can produce, currently produces and minimal power that the units have to produce while it is on. The collected data is then used to calculate a lot of parameters that later can be used to decide the set points for each unit.
Parameters that are being calculated are:
Maximum capability: This parameter gives information about how much a certain group of units can provide.

- Operating Reserve: Provides additional capacity from electricity that can be used to serve customer demands.
- Raise Regulating Capability: Gives information about how much more capability that can be used for primary control.
- Falling Regulating Capability: Gives information about how much capability that currently is used for primary control.
- Current Generation: Capacity that currently is used.
- Minimum capability: The minimal capacity that the units need to be operating at while they are on.

The analysePowerCapacity part is checking what type of units that are connected to the system. The information is later used to decide the set points for the different unit types. This part of the program is also checking how large the safety zones for the units in droop mode have to be, see Figure 3.3. To do so the largest unit, which is operating as a generator, is found and decides how large the upper safety zone A1 should be. In the same way is the lower safety zone A2 being set by checking how large the larges unit that is operating as a load is.

The security update step is where the voltage level is checked. If the voltage level is to low, no unit except for the one in droop mode is allowed to operate. This is to prevent the system to get out of control.
It is also here that the maximal export and import power to and from the system is calculated and sent away to the units within the power system. This has to be done in order for the units to know what the limits for the system are. This is especially important for the exchange unit. This unit has to be able to inform the system on the other side how much power that it can provide to the different sides. This prevents one side from totally ruin the other side from all its energy without permission to do so.

The decideSetpoints sequence is the sequence where the major calculations are made. This part is in itself divided into two parts.
The fists part takes care of the primary controlling and decides which and how many units that should be in droop mode.
The second part takes care of all of the other units that do not operate in droop mode. This also calculates how much power that has to be added by the generator to support the loads with as much power as they need. If however the generators are not able to support the loads by them selves, the exchange units and the storage units add some extra power. If the case is that the generators can produce much more power then the loads can consume the system tries to store and export that extra amount of power.

A flow scheme of the primary control part is shown in appendix A
After the initial step where all the set points are set to zero comes a sequence where the number of units that are operating in droop mode are checked.
If there is no unit that is operating in droop mode the system is trying to start a generating unit in the system in droop mode to make sure that the voltage will be set. A source unit could either be a generator, storage or an exchange unit since all of them can add power to the system. But the start of a generating unit is not allowed without a unit that could act as a load connected to the system as well. There is no point in having a system operating if there is nobody that will use it. A sink unit can in this case be a load, storage or an exchange unit since all of them can consume power.
The system is also trying to start up another generating unit in droop mode if there only is one other droop-mode unit in the system and if that unit almost is working at 80% of its full capacity. This is due to safety reasons; a system with only one unit that takes care of the primary control is very dependent on that unit. If it breaks, the whole system goes down with it.
But to start another generating unit in droop mode when there already are 2 or more generating units that are working in droop mode, the regulating reserve has to be small. It is considered to be small when the sum of the safety zones, calculated in the analyzePowerCapasity part, is almost as big as the regulating reserve. Otherwise there will not be any more generating units added to the system.

Then the program continuous to calculate the set points for the different unit types, which are managing the secondary control.

A flow scheme for the secondary control is shown appendix A.
The idea for this structure is to be able to use the same blocks for the set point calculation even if not all of the different unit types are connected to the system.
First of all, the program checks if there is a load and a generator in the system. If there is, it set the set point for the generator to be large enough to feed the load with the power as it needs. If however the load needs more power then the generators can provide, the set point for the generators is set to its maximal value.
Thereafter the program checks if there is a generator and storage in the system. If there is, it also checks if the storage is full and if the generators are working at there full capacity. If not the set point for the storages are set to store some energy. The energy can however not be stored in a higher pace then what the generators can provide or what the storage can take. The set point for the generators is increased with the same amount of power as the storage is set to store.
This part is supposed to be able to work both alone and together with the part that takes care of the generator / load situation. That is why the flow scheme might look a little bit odd.
The next part takes care of the situation with both a generator and an exchange unit in the system. Power from the generator is exported when the need for power to the load and the storage is fulfilled. The amount of power that is exported is decided form the least value between how much this system can export and how much the other one can receive. The set point for the generator will be increased with the same amount of power as the exchange unit is set to export. If there is no load or storage connected to the system this part will still be able to work since the values for how much power the load and storage wants to have will be equal to zero.
This next coming part is about if there is a load and an exchange unit in the system. If the generators, which might be in the system, were not able to provide the loads with all the power needed, the exchange unit will try to provide the rest of the power. But if there is no generators in the system the exchange unit will try to support all the power to the load by itself. Setting the set point high enough to cover for the loads or to its maximal value does this.
Thereafter the situation with a load and the storage will be taken care of. The storage will try to support the load with power if the generators and exchange units can't do that. The storage will then try to add the extra amount of power that is needed to make the load consume as much as they need. If there are no generators or exchange units connected to the system, the storage unit will try to support the loads by itself. This part is just as the other parts, meant to be able to work in all kind of different systems. It does not matter if there are generators or exchange units or none, the system will be taken power from any of the other units if there is any. It is only when the other units, working as generators, can't support the load enough as the storages will add some power.

Finally the set point for the load will be set. And to never let the load get the chance to overload the system, its set point will be equal to the amount of power that is generated plus the raise regulating capacity.

With all of the small stand-alone parts all of the different system types will be taken care of. If all of the different unit types are connected to the system, all of the parts described

above will be used and if just some of the units are connected to the system not all of the parts will be used and the program knows by itself which ones to use.

GenManagement
It is not enough to know what the set point for the generators should be. It is also important to decide which generator it is that should fulfill the requirement and that is done in the part genManagement.
The decision of which generator it is that should do what is mainly decided by what priority each unit has. All of the generators are getting a number that tells how important the unit is. The lower the priority number is the more important is the generator.
The generators with the lowest priority number are the one that is started first. If there is more then one generator with the same priority number the power is split between them with respect to how large they are. Meaning that if there are two generators that should split a power set point between them and one of them is twice as large as the other. The larger one will also produces twice as much power in comparison to the smaller one. A flow scheme for how the generator management is working is showed in appendix A
The startup sequence for the generators is, first of all, to start to produce power at its minimum regulating capacity before it can start to follow the set point.
The shut down sequence is very similar to the startup sequence but this time it has to get down to its minimal regulating sequence before it can be shut off.
The signals that this part is sending to the converter contain information about what command mode, govern mode and set point the generator is supposed to fulfill.

LoadManagement
The load management is very similar to the genManagement but will instead decide which loads that will consume the energy. The one with highest priority is even here decided by a priority number, which works in the same way as for the generators. The startup and shut down sequence is also implemented in the same way as for the generators.

StorManagement and ExchManagement.
The program sequences that decide which units that are supposed to fulfill the set points for the storages and the exchange units do look very much the same as the one for the generators. The only thing that differs for the storage and the exchange unit is that some special consideration had to be made in order to make it possible for them to start with a negative set point
Except for the command, govern and set point signals an additional signal is also sent to the units. This signal tells the units in which direction the power flow is supposed to go in, export or import. This signal is crucial when it comes to the exchange unit, otherwise it would be very hard for the converter to know in which direction the power flow would be.

3.6.2 Production Exchange

It is this program that receives and processes the signals from the production-Management. This program finally decides which signals that are going to be sent down to the actual converter and which ones that should be sent back to the production

management. These signals are meant to give information about what the maximal capacity of the unit has at the moment and in which modes it is operating. These pieces of information are later going to be used to calculate new control signals for the converter.
A flow scheme for the productionExchange unit is shown in appendix A. This flow scheme shows how the program is working for the exchange unit, which needs to be able to communicate with two power systems and the converter.
In the selector block the program decides which one of the power system that is allowed to control the converter. This is done with a command priority signal that each system has to send. The power system with that highest priority gets permission to control the converter. But if any of the power systems instruct the converter to shut down, then the converter has to shut down. So even if one side has the authority to take command of how the converter is supposed to operate, both sides have to agree about when the converter is allowed to start.
The limiter is, if needed, limiting the set point signal down to the converters maximal operating value. This is done for security reasons. It should not be possible to run the unit with higher values then it is designed for. The set point command from the production management is never supposed to send a higher set point command then the unit can manage.
The mixing and cross coupling part manages the signals that is sent back to the production management. First, it collects the information from all of the inputs. Almost all of the signals that are coming in from side A are also going out from side B and vice versa. The exception from that rule is information concerning the recourse and the information about how much the converter can import and export. The import and export information is limited to the maximal capacity of both sides.
For example if side A can export 1000W but side B only can import 700W the import signal on the out put of side B will be 700W. Corresponding limitation is to the export signal is being implemented.
The signals coming in from side B is coupled to the A side in a similar way.

The productionExcange is supposed to work for all types of different units. Therefore, it is very flexible concerning the number of entrances and exits. For the generators, loads and exchange units, it only has two inputs and two outputs.
The program is able to handle that situation as well by setting suitable initial values on the side not connected to any power system.

4 The Power Converter

In order to be able to test the system that has been described in the chapter above some kind of power electronics devices between the AC and DC side are required. 3-phase converters are used as the power electronics in the tests. A short description of such a converter is done in this part.

4.1 Converter basics

The 3-phase converter basically consists of 3 half bridges in parallel with a capacitor as shown in figure 4.1.
Two transistors build up every half bridge and each transistor has one diode connected in parallel. In order to control the power to or from the capacitor a technique called pulls-width modulation (PWM) is used. This technique will briefly be described a little later in this report. When one of the transistors in a half bridge is on, the other one is always off. That is important since if both of the transistors in one half bridge were on they would shortcut the capacitor and that would destroy the transistors. To avoid that, both of the transistors are off for a short period of time, and then the transistors switch from on to off or vice versa.
The converter has a front that is connected to an inductive source. Due to that the load must be capacitive. The capacitor C_{dc} is also there to prevent disturbances from one side affecting the other.

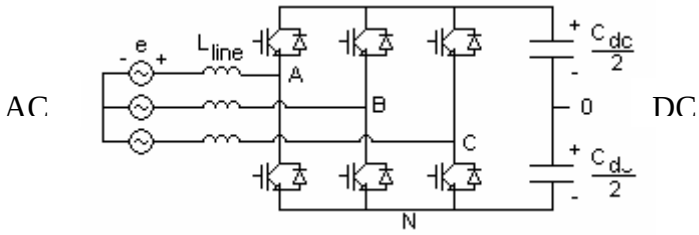


Figure 4.1. Schematic picture of a 3-phase converter.

When the voltage, held by the capacitor C_{dc}, is too low the converter will act as a 3-fhase-diode rectifier instead. In this case it doesn't matter if the transistor is on or off, the current will find its way to the capacitor through the diodes. This is a problem during the start-up when the voltage over the capacitor is zero [1].

When the converter acts as a diode rectifier, the voltage that appears over the capacitor is calculated below, eqi 4-1. This is also the smallest voltage the capacitor can have and still be controllable.

$$V_{C_{dc}} = \frac{3}{\pi}\int_{-\pi/6}^{\pi/6}\sqrt{2}\cdot V_{LL}\cdot\cos(\omega t)\quad d\omega t = \frac{3}{\pi}\sqrt{2}\cdot V_{LL} \approx 1.35\cdot V_{LL} \qquad \text{eqi (4-1)}$$

Where V_{LL} is the value of the voltage between the phases from the AC-grid.

4.1.1 PWM in three-phase converter

This section is written to give a very short explanation about the how a pulls-width modulation (PWM) for a 3-phase converter works. The inverter mode is when the power flow goes from the capacitor to the AC-grid.
Before discussing the PWM behavior, it is necessary to define a few terms. The triangular waveform in figure 4.2 is at a switching frequency f_s, which defines the frequency with which the converters are switched. The control signals ($v_{control}$ *A, B* and *C*) are used to moderates the switch duty ratio and has a frequency f_1, which is the fundamental frequency of the converter voltage output. The control signals A, B and C from figure 4.2 are supposed to control the characteristics of the line voltages and have the same amplitude and frequency, but control signals B and C are displaced ± 120° with respect to control signal A.
To avoid an unnecessary amount of disturbances and distortion the amplitude of the control signals are to be smaller than the amplitude of the triangle wave and $f_s >> f_1$.

How the line-to-line voltage V_{AB} is created is here discussed; the other phases are created in a similar way.
To create V_{AB} both $V_{Control\ A}$ and $V_{Control\ B}$ have to taken into consideration since the V_{AB} depends on the potentials in the points A and B.

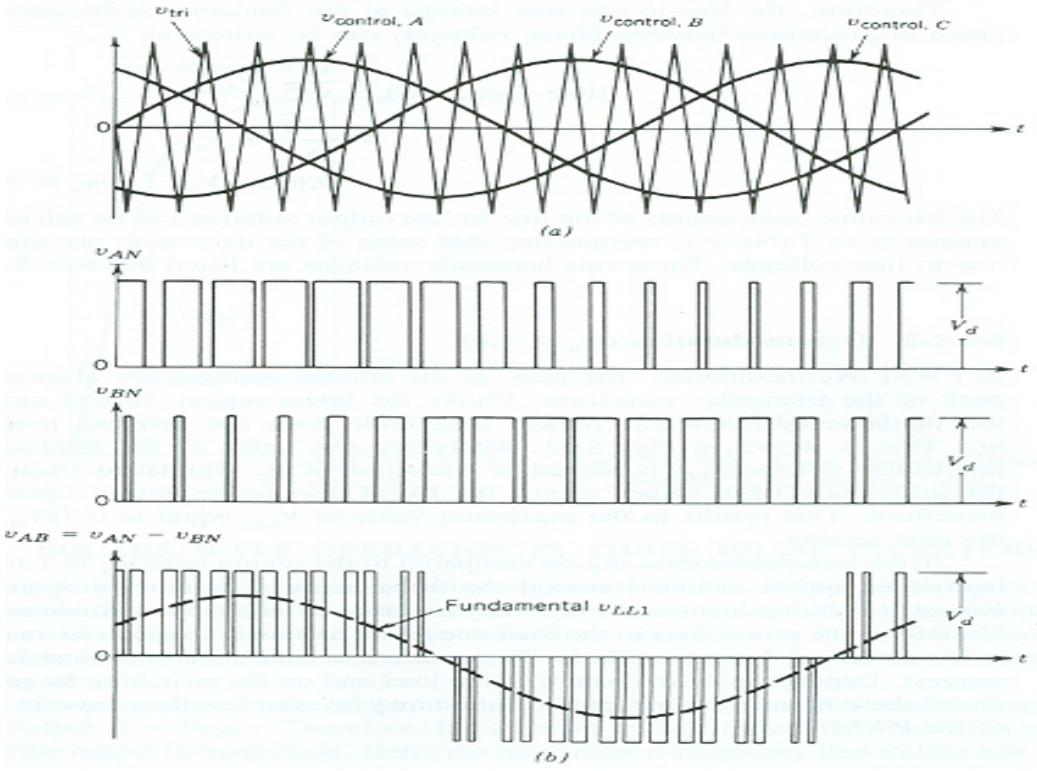


Figure 4.2. Three-phase PWM waveforms.

The control signal for the first half bridge is created by comparing $V_{Control\ A}$ with the triangle wave. If the control signal is greater then the triangle wave the control signal is on, and if the control signal is less then the triangle wave the control signal is off. If the control signal is on that means that the upper one of the transistors in the half bridge is on and the lower one is off. This makes the potential in point A follow the control signal at an average, this is illustrated in the upper picture of figure 4.2b. The potential for point B is calculated in the same way as the potential for point A.
V_{AB} is calculated by taking the difference between the two potentials V_A and V_B, meaning that whenever the potentials are the same V_{AB} will be equal to 0 and when they differ it will be V_{DC} (see the lowest graph in figure 4.2).
Using this method it is fully possible to shape the line-to-line voltage in any way desired on the AC- side of the converter, within limits, when it comes to frequency and amplitude. [1]

4.1.2 PWM in three-phase rectifiers.

The inverter mode is when the power flow goes from the capacitor to the AC-grid. Consider a three-phase system as in figure 4.3, which is redrawn and only showing one phase in figure 4.3. Here as well as in the inverter mode we are only interested in the fundamental frequency - this neglects all the distortions and harmonics.

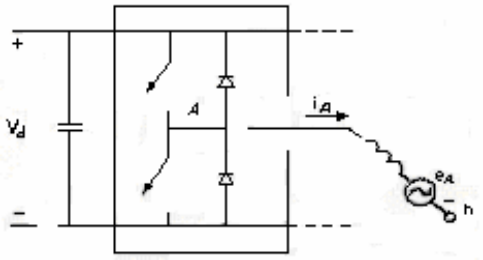


Figure 4.3 Circuit of one phase of the converter

As described above, it is possible to create any kind of waveform in the point A in figure 5.3. The phasor diagram shown in Figure 4.4 illustrates how it is used to make the converter work in inverter mode.

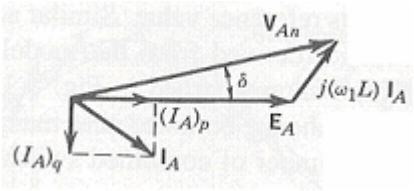


Figure 4.4. Inverter mode of operation

From Figure 4.4 we see that if the phase angle for the voltage in point A is ahead of the source, the current will flow towards the AC grid. The phase angle between V_{An} and E_A can be controlled by shifting the phase of the control signals in figure 4.4 above. How great this angle should be is determined by the size of the inductors on the AC-side of the converter [1].

An advantage of switched-mode converters is that they can make smooth transitions from inverter to rectifier mode. The rectifier mode in this report is when the power flow goes from the AC-grid to the capacitor in figure 4.5.
If the converter voltage V_{An} is now made to lag E_A by a certain angle δ, the phase diagram in Figure 4.5 shows how the active component $(I_A)_p$ of I_A is now 180° out of phase with E_A. This results in a condition when the capacitor is charging up and V_{DC} increases.

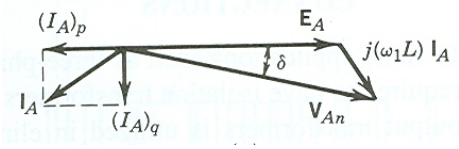


Figure 4.5. Inverter mode of operation

How great the phase angle has to be is determined by the size of the line inductor in this mode as well. [1]

4.2 Controllability

It is not possible to control the converter when each transistor should switch directly from the computer - it has to be done from the hardware on the converter itself. A program written in Simulink has to be downloaded to the converter to make it possible to control the switching.
The control program (see appendix C) mainly consists of three parts. One part acts as an interface to the superior system, the second one is the controlling part and the last part is the feedback part. The controlling part also sends down the PWM signal to the converter to make the transistors switch when they should.

4.2.1 Communication interface.

As mentioned earlier, the picture of the control program used by the converter to operate the switches is shown in Appendix C and the parts that have to do with the superior communication level are colored in cyan. Since the program is run by the converter and not by the computer itself, the communication from the computer is limited and the signals that are sent from the computer to the converter are in the form of set points and states.

These signals are connected to the block Superior control and consist of one set point signal, one command signal and one govern signal.

Command signal
The command signal has the information concerning which state the converter is supposed to be in. The possible states as it can be in are intern, start, ready and stop. In the off state, the converter is completely off and is not available for any kind of controlling. In the ready state, the controller is ready to start to control either voltage or power but since it doesn't know what to control yet it is just trying to keep the current going through the converter at a minimum. In the on state the controller is in its active state, meaning that it is controlling either power or voltage. The controller is in its intern state when the communication to the superior system doesn't work any more.
The ready state is not necessary to have in the lab for those tests that are done here. However the system is supposed to work even on large DPSs where each generator or load may need several minutes to start up. This means that it is necessary to have it in order to know which units really are ready or not.

Govern Signal
The govern signal is used to order the converter what the converter is supposed to control. Minimum power control is when, for example, a generator should produce as little power as possible without turning itself off.
Maximum power control is when you order the plant to use as much power as possible. This state is good to apply on a wind power plant where you have a variation in the maximal power and you want to use all of it.
Power control has to be an option and then of course it is power levels between the minimum and maximum that is of interest.
Voltage control is necessary if you want to keep a stable voltage level in the grid. There are two types of voltage controlling options that can be used.
One has both proportional part and an integrator part in the controller. This mode is slightly tricky if there is several units using it and if they are placed closely together. Because if there is an error in the measurements then the integrator parts can start to compete with each other and the signal control signals might be wrong.
The other one is called droop mode and is a voltage controller with only a proportional part. With this option there will be a stationary error in the voltage level but the voltage will remain stable.

The last of the signals that is sent down to the control program is the set point. It is this value that the controller is supposed to follow, both for voltage and power control.

State machine.
Letting the converter have several states is useful in case the unit as the converter is attached to needs to have some time for its startup. When the converter has changed from stop to ready then you know that the necessary measures are taken and the unit is ready to start to work.
The task of the state machine is to make sure that every demand is set before the converter can jump from one state to the other.

The normal procedure is that it jumps from stop to ready and from ready to start and then back to stop when it is time to shut the unit off. It is therefore not possible to go directly from state stop to start. But it is possible at any time to jump back to stop when the error signal is activated.

Control mode.
This block takes care of what it is that is supposed to be controlled. There are three things that are supposed to be controlled. The DC-current is supposed to be set to zero when the converter is in its ready state. The other two things that can be controlled are the power and the voltage when the converter is in its active state.

Govern mode
The task of the govern mode block is to deliver the correct set point to the control blocks. The signals it can deliver are min power, max power and set point. Which it chooses is decided by the govern signal.

4.2.2 Control Part.

The controlling part was not created during this research; instead some old material previously used for lab purposes was utilized. Below is a brief description of the functions of each block.
The controlling blocks, colored in yellow in appendix A, are the ones that take care of the controlling, starting with the block dc_link_vollage_control. This block is given the error of what it is supposed to control. If it is supposed to control the voltage it is fed by the voltage error, and if it is supposed to control the power it is fed by the power error. The block contains a PI-controller so – whatever it is controlling – the error will eventually go to zero. The output of this block is used as reference value of what DC-current is needed to compensate for the error and is fed to the block Current_reference. This block uses the DC-current reference to calculate how big the current on the AC-side should be, meaning that it transforms the DC-current reference to an AC-current reference and represents the result in a two-phase dq format with q part set to zero. The q part is set to zero because we don't want to add any reactive current in the system. This transforming step is necessary since the pulls-width modulation is produced from the AC-side.
The dq current reference is then fed to the block vector control. This block compares the dq-current reference with the actual dq-current and calculates the error between them. The error is then fed to a P-controller. The output of this P-controller is then used as a voltage reference in dq-format. This reference value is then transformed to three phase coordinates in the block vector_to_abc. These signals are then symmetrical iced to avoid saturation and are then fed to the block PWM. This block calculates the actual power with modulation, as described earlier, which tells the transistors in the converter if they should be on or off.

4.3 The Feedback Part.

From the green block in appendix A comes all of the feedback signals from the converter.

The signals that are being measured and sent back to the control program are two of the phase-currents, the flux vectors in αβ coordinates from the AC-side, the DC voltage and the DC current.
Since the load is a symmetric load it is only necessary to measure 2 of the 3 AC currents, as the third one can be calculated from the other two. The flux vector coordinates are used so the control program is in phase with the AC source. This is important when power is transferred from the DPS to the AC-grid.
There is a total number of 8 channels that can be used as an analogy to digital communication with the control program. 4 of these channels are synchronous sampled channels and 4 are asynchronous sampled. Since it is important that the AC-currents are measured at the same time, 2 of the synchronous channels are used for that. The other four measurements that are sent back are asynchronous sampled.

5 Test Scenarios.

The aim for this work is to build and test a number of small distributed test scenarios. To do so there have to be some well-chosen scenarios that are going to be tested to make sure that all of the different parts of the system is tested.
The DPS is supposed to be tested both in a simulation environment and in a laboratory environment in one of the laboratory rooms at the university of Lund. The very same test that is made in the simulations is also made in the laboratory.
This chapter will explain which tests scenarios that are going to be tested and what the purpose for each scenario is.

All of the different units, generator, load, storage and exchange unit will be used in at least one scenario.
In the description of the scenarios some pictures will be illustrate the configuration of the scenario. The same kind of picture will be used over and over again and the meaning of each symbol in the pictures is listed below.

Generator unit: eg wind power, solar power, hydro power

Load unit: eg. industries, motors, heating, residence, etc

Storage unit: eg batteries, pumped hydro, etc

Exchange unit: eg transformer, converter.
Connects one power system with another.

Limitations
In the laboratory, only converters were available. Because of that some adjustments have to be made in the strategy 1 to 5 enabling these tests.

- The system will only start up one unit that will control the voltage and it is always started in droop mode. The rest will be started in power control mode.
- The will not be any safety zones for the unit in droop mode. If the safety zones that are mentioned earlier would be applied here no other unit would get permission to start.

But all units in power control mode, except for the loads, will try to make the droop mode unit to produce power at half of its maximal capacity. This will at least give some safety margin to the system.

5.1 Scenario 1

This scenario contains only one converter, as shown in figure 5.1, and is only purpose test a startup of the DC power system. The unit that is used in this setup is going to be a generator, but a storage unit or an exchange unit could have worked just as well.

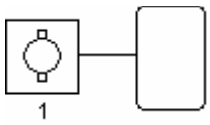


Figure 5.1. Setup for scenario 1

5.1.1 Goal

The goal of this scenario is to validate that the communication between the production management and the production exchange is working as it should. It is necessary to have this bit done before it is possible to set up more advanced scenarios.
The tests for this scenario can look a little bit different between the simulation and the laboratory test.
In the laboratory test is it very important to check that the converter really works as it is supposed to. Tests of all the different modes of the converter have to be made before the unit can be tested in the automated system.

5.2 Scenario 2.

This scenario contains 2 converters, one generator and one load, connected according to Figure 5.2. The first converter is responsible for the voltage level of the local DC-grid. During start up no converter except for the one that can control the voltage of the DPS is allowed to start up. The generator is started in its droop mode to set the voltage. When the DC voltage have reached its rated voltage limit the other converter, assign to act as a load, starts to consume power from the DC-grid, when it does the generator unit should compensate with adding as much power as the load is consuming.
The power that the generator can produce will be a fixed value but the power that the load wants to consume will vary in a sine-shaped curve. The idea of the test is that the power as the load want to consume will be higher then what the generator can produce in some occasions. This is done to check if the system is able to handle this situation without getting unstable.
If the load is asking for more power then the generator can provide the maximum set point for the load are going to be equal as the power provider in that moment.

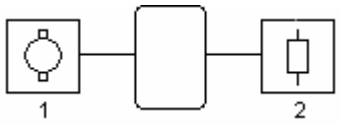


Figure 5.2. Setup for scenario 2

5.2.1 Goal

The goal of this scenario is mainly to verify that the parts of the program managing the load are working as they should. But it is also used to make sure that the droop mode for the generator is working as it should and can keep a stable voltage in the DC-grid.

The expected power curve looks like figure 5.3 where the power generated by the generator G1 follows the power consumed by the load. The power as the load L1 wants to consume is shown with a broken line and the actual with a full line. The power for the generator G1 is drawn in a full line on the positive side of the power axes.
The small bump in the beginning of it is there to illustrate the power needed to charge the grid. The power as the load wants to consume will be followed except for when the limit for what the generator can produce. Then the set point for the load will be locked to the maximal power as the generator can produce.

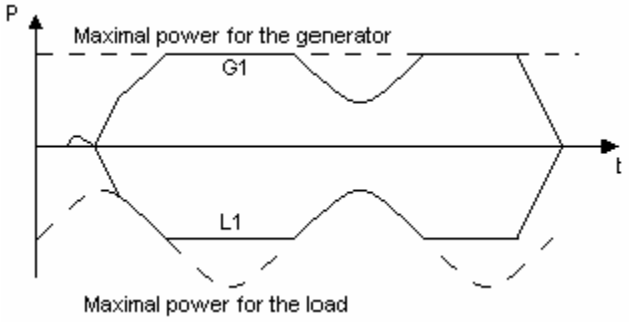


Figure 5.3. Expected power curve for scenario 2.

5.3 Scenario 3.

For this scenario 3 converters are connected according to figure 5.4.
This scenario contains 2 generators and one load. Just as in the first scenario there is one generator that controls the voltage of the DC-grid. Before the voltage has reached its rated voltage no other units are allowed to be activated.
The maximal power that each generator is able to produce is set to be a constant value while the power as the load want to consume will be sine-shaped just as in the previous scenario.
When the load consumes more power then half of the maximal power of what the droop mode generator can produce the second generator will add the rest.

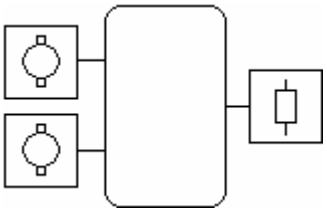


Figure 5.4. Setup for scenario 3

5.3.1 Goal

This scenario is meant to check that the generator controlling the voltage is producing power at half of its capacity as much as possible. But it does also check that the parts of the program that takes care of the generator works at it should. This can however not be the case when the load wants to consume very little power or very much power.
The expected power curve for this test is shown in Figure 5.5. The power produced by droop mode generator is showed in grey and the power produced by the other generator is showed in black on the positive side of the y-axis. The power as the load consumes is showed in black on the negative side of the y-axis.
The first thing that happened is that generator G1 charges the grid until it has reached its rated voltage level.
When that is done the load L1 starts to consume as much power as it wants to. When the power consumption of the load has reaches half of what the droop mode generator G1 can produce the other generator G2 adds some power to cower the rest of the power, as the load wants to consume.

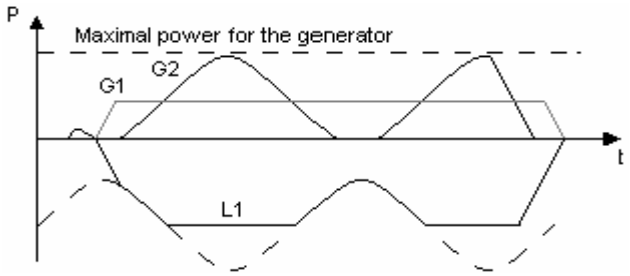


Figure 5.5. Expected power curve for scenario 3.

5.4 Scenario 4

The next scenario that is to be tested contains one generator that has a maximal capacity that is varying with respect to time and can therefore be seen as a wind power plant. Another unit should work as a storage and the last one is working as a load, see Figure 5.6. Both the load and the storage have constant power capacities.
The idea of this test is that the storage should store energy when the generator have a high capacity and add energy to the grid when the generator has a low capacity.
The generator manages the voltage control by operating in droop mode. Similar to the other scenarios, the system will try to keep the droop mode unit at half of its capacity.

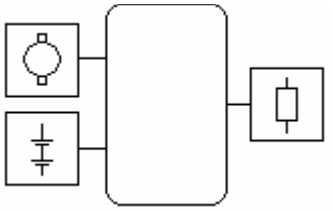


Figure 5.6. Setup for scenario 4

5.4.1 Goal

The purpose of this scenario is to test the storage module in order to see that it is working correctly.
The expected power curve looks like the one in figure 5.7. The grey curve on top of the graph is representing the amount of energy as the generator can produce at any given time.
The first thing that is supposed to happen is that the generator G1 sets the voltage on the grid. When that is done, the load L1 will start to consume the power, as it wants to consume. But when the load consumes more power then half of the capacity of the generator, the storage will start to add some power. The generated power of the storage should be enough to keep the generator to produce power at half of its capacity.
But since the storage is chosen to be small then it will get empty quite quickly. That is what happens when it goes to zero for the first time. The load will still get all the power needed but the safety distance upwards for the generator can no longer be maintained.
A bit later when the capacity of the generator is high, the storage will start to charge, but it will get full shortly after that and stop charging.

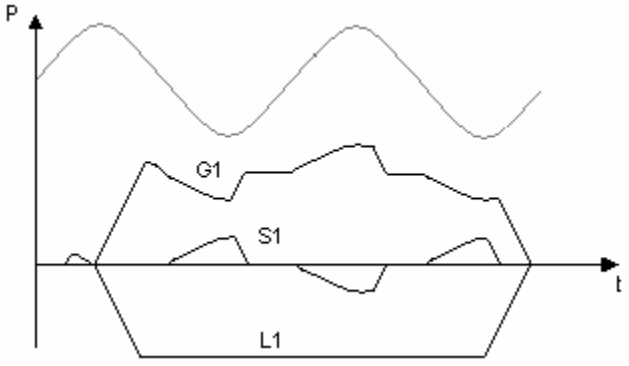


Figure 5.7. Expected power curve for scenario 4.

5.5 Scenario 5.

The scenario is very similar to scenario 4, it does also have 3 converters connected, as shown in figure 5.8. In this figure as well as in figure 4.3 there is one generator with a variable maximal production capability and one load, which is supposed to consume power at a constant level. But the third unit is a transmission unit, this unit is supposed to work as a link to a similar system as the one we are looking at.
This test is about testing this transmission module. This module works very much like the storage module meaning that it is supposed to be able to handle large amount of power in both directions. It does however differ a bit from the storage unit since it doesn't have any energy storage and can therefore not become empty or full.

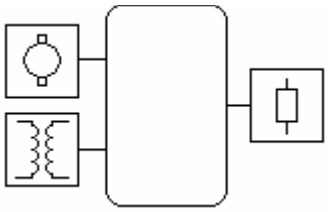


Figure 5.8. Setup for scenario 5

5.5.1 Goal.

The goal for this test is to show that the exchange unit works as it should. The test for the exchange unit is made in the same way as for the storage and the expected power curve for the test looks like Figure 5.9. The grey curve in figure 6.9 is representing the maximal power capacity of the generator G1.
The exchange unit E1 will always try to keep the power produced by the generator G1 at half of maximal capacity. Meaning that, it will try to add power to this power system when the generator is week and try to withdraw power from the system when the generator is strong.

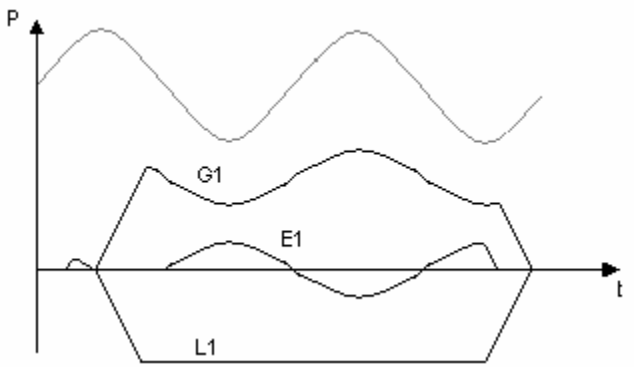


Figure 5.9. Expected power curve for scenario 5.

5.6 Scenario 6.

In this scenario are 4 units used according to Figure 5.10. Because of that, this scenario will not tested in the laboratory, only in the simulation program. 3 of the 4 units in this scenario act as generators and the fourth one as a load. All of the generators will have a constant power capacity while the capacity of the load will be varying. The power that the load will consume will look like a ramp, which will continue to get a higher and higher value all the time.

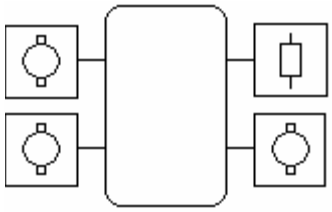


Figure 5.10. Setup for scenario 6

5.6.1 Goal

The goal for this scenario is to test the startup sequence that is described previously in the part about strategies. Another thing that this scenario is supposed to test is that it will make sure that the generator that is working in the safety distance upwards will be taken care of.
The expected power curve for this scenario looks like Figure 5.11.

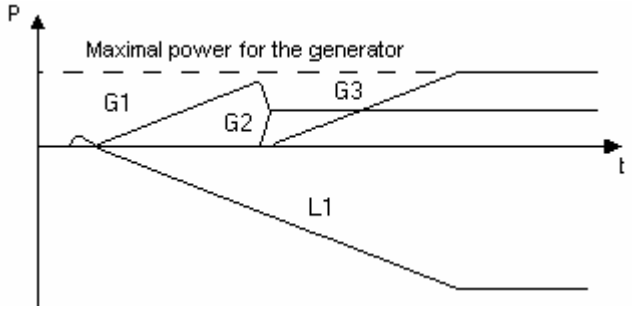


Figure 5.11. Expected power curve for scenario 6.

This figure shows that the first generator G1 starts in droop mode and supports the load with power. When the power level for the generator G1 reaches 80% of its maximal value another generator G2 will start, even this one in droop mode. Since both of these units are working in droop mode they will chare the power that is delivered to the load. The droop mode generators are not allowed to produce more power then half of their maximal capacity in order to have some safety margin if any of them breaks.
Since the power demand from the load still increases and the droop generators already produces power at half of their maximal capacity, the third generator G3 starts and supports the load with the extra power needed. This generator is started in power control

mode. The power that this unit is allowed to produce is equal to half of the maximal capacity of the droop mode generators. This is because the droop mode generators must be able to cover for generator 3 if this one suddenly would break or loose its contact to the grid.

5.7 Scenario 7.

In this scenario 4 units are connected according to Figure 5.12.
Tree of the units will be loads and the fourth one will be a generator. The power capacity for the generator will vary and the loads will have constant power demands but at different values.
All off the loads will have different priority numbers. So depending on how large the power capacity for the generator is the different loads will work at different power levels .

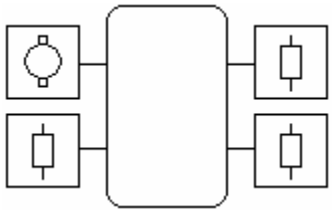


Figure 5.12. Setup for scenario 7

5.7.1 Goal

The goal for this scenario is to check that the right loads are turned on according to what have been described in the strategy part in previous chapter.
The expected power curve for this scenario looks like Figure 5.13

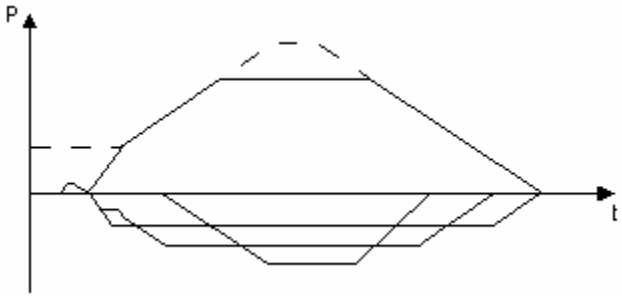


Figure 5.13. Expected power curve for scenario 7.

The dashed line in Figure 5.13 shows the maximal capacity for the generator and the solid line shows the power the generator should produce.
The solid lines, on the negative side of the power axes, show the power that the loads are supposed to consume.
The load with the highest priority will get its power that it wants to consume first. When the power capacity for the generator increases can even the loads with lower priority get there demands full filled.

The same thing happens when the power capacity for the generator decreases. It is the generator with the lowest priority that will get its power set point decreased first, in order to not overload the generator.

6 Verification by Simulations

In this chapter, small DC DPSs will be simulated based on the scenarios described in chapter 5. The simulation focuses on the dynamics of the system and on how well the coordination programs work. Measurement and result will be shown and discussed.
A description of the simulation tools and how the coordination programs work will also be explained.

6.1 Simulation tools

The simulation program used in this work is Dymola and the programming language used is called Modelica. Dymola/Modelica is a very dynamical environment that is based on graphical programming where the system is built up by modules that is connected to each other. The modules are provided by Modelica and contain all the necessary components that are needed to build up a system. It is however possible to make your own module by using some of the modules that are provided by Modelica. So basically what happens is that you make a lot of units to look like one. This is done to reduce the numbers of modules on the screen and to make it easier to maintain a good overview of the system. It is also possible to build up a module by using the program code. This code looks very much like c-code and is quite easy to use. By using this feature, there is no limitation to use only the standard modules that is provided by Modelica. The code however presented on screen as a module and is used in the same way as the standard modules.
It is also possible to link a program written in C++ to Modelica and make them interact with each other. This makes it possible to use the very same code in the simulations as later is going to be used in the actual system

6.1.1 Modelica blocks.

To make some reliable simulations of a power system the simulation structure must look like the real system.
To make it easy to have a good overview of the system it is important to try to make the simulation simple with few blocks on the screen. In order to keep it simple, all of the different units have been made as different modules that are used to build up a power system.
An example of how such a structure would look like in the simulations is shown in figure 6.1

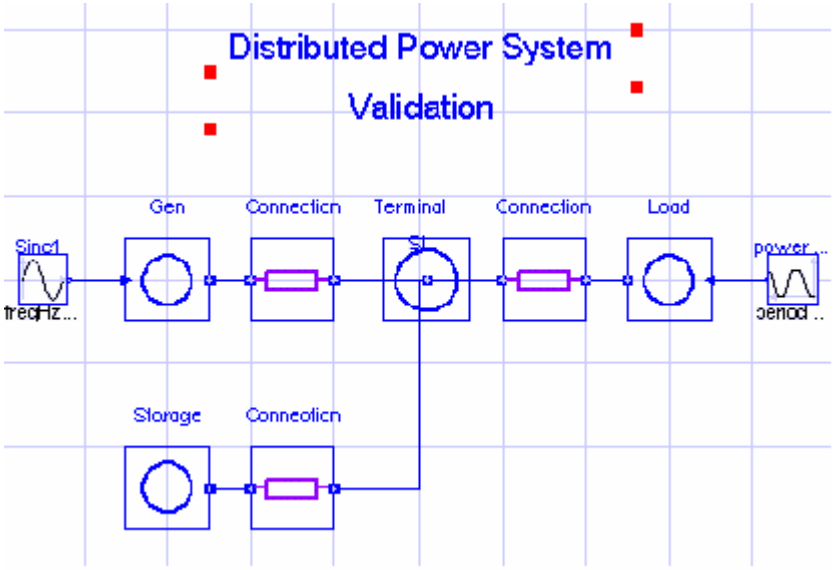


Figure 6.1. Figure of a small DPS in a simulation environment.

This system shows some of the different types of units that are used in this work, only the exchange unit is missing. But it is added in the same way as the others.
With a structure like this, it is easy to have a good overview of the system and new units can be added without any problems.
Each generating or consuming unit has to be connected to a connection unit before it is attached to the terminal unit. The connection unit adds some power line parameters into the system and can represent everything from a small connection to a high capacity power line.
It is the terminal unit that is the hart of each DPS. This is the link to the automated operating programs, production management (PM) module, where all the command signals comes from. Without any connection to this module the unit will not get any permission to start.

All of the units that are simulated must be properly built and classified to maintain balance in the system. This part of the chapter will describe the main structure of each unit that is used in the simulations.
Every unit that is to be used in the model has to be given a certain group identity number otherwise will the PM module not be able see any difference between the units.

The generator
A generation unit is a resource only capable of generating energy. In reality this could be a wind power plant, a water solar power plant etc.
The very same model used to simulate a generator is also going to be used to simulate a load.
How the generator looks in more detail and how the information block looks like is shown in Figure 6.2. To inform the system that this is a generator, some initial settings have to be made. This is done by double clicking with the mouse on the model and fill in

the numbers in the box that comes up. In this box is the group identity number for the unit and the nominal power as the unit wants to be working in is stated as well.

Some values about what nominal power the unit should have and how fast the unit will be able to change power levels will also be stated there

The left picture in Figure 6.2 shows how the generator looks like in more detail. The left side import port provides how much power the generator can produce at a certain point of time.
This piece of information is placed out side of the generator itself to make it more convenient to change that waveform.

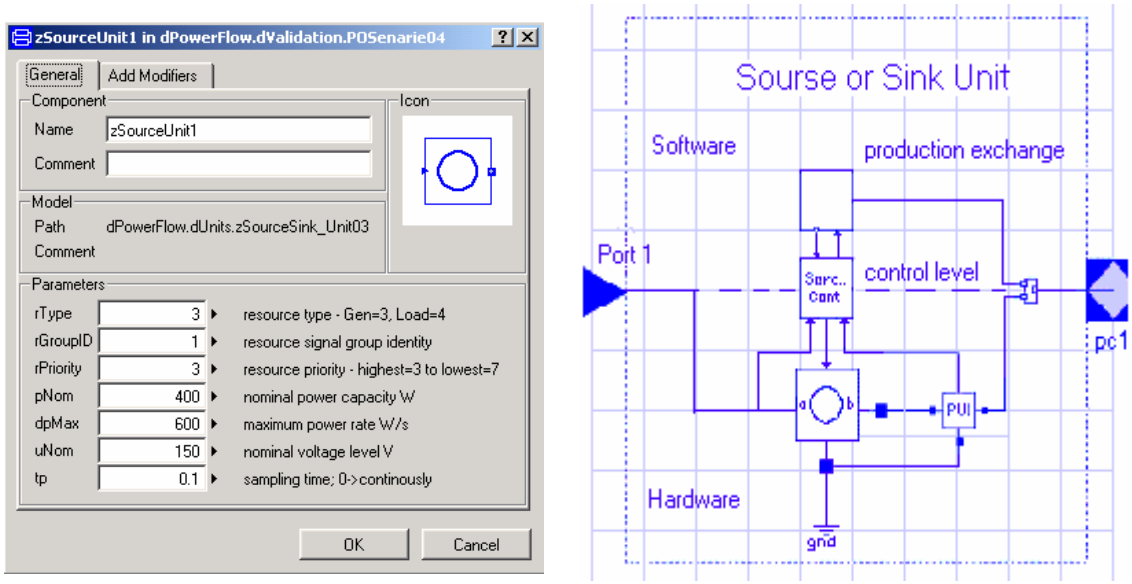


Figure 6.2. Shows how the information for the generator look like and how the generator model looks inside.

From the right picture in figure 6.2 it is shown how the generator looks inside. The generator is mainly divided in 3 parts and they are the three bigger blocks in the middle of the figure.
The block in the top is taking care of the communication to the production management module and checks all of the command signals that are coming from that module. The hart of this block, the power flow will be discussed later in this chapter when we take a closer look at this block. When the command signals are checked they are sent down to the control level.

The controlling part is taking care of the big block in the middle of the figure. This block will use the information that is coming from the level above it to calculate a set point for the converter that is situated below. The set point value will be a value between 0 and 1, where the value 1 means that the generator should work at its maximum. This block will

also send back information of in which mode of operation that the converter is working in and what the current voltage and amount of power that it produces.

The model of the converter is very simple and could be describe as a pump that pumps power into the system. Therefore, the maximal value that the generator can generate is also distributed to the converter model. The output, the thick line, from the converter is a power line in which the voltage and currents are held.
In the right direction from the converter is a small block that takes care of the measurement of the power, voltage and current in the point where it is placed.

The connection point in the very right in the generator model combines the information channel from the upper part of the model and the power line from the lower part of the model. The line that leaves the generator unit will then be a combination of a power line and an information line.

The load
The load unit model is an exact copy of the generation unit. The only thing that differs is that the operator has to tell the system that this is a load by typing a different number among the initial values in the information box. The maximal capacity value that has to be fed to the load must be a negative number to really point out that this is a load and not a generator. For an explanation of how the model works see the explanation for the generator.

The storage unit
The resource identity of this unit is a little bit different then for the other two mentioned above. Since the unit is supposed to simulate a storage it also has to be able to estimate how much energy that is stored. Some extra calculations have o be made due to that. This unit is also supposed to be able to have a bi-directional power flow since it must be able to both charge and discharge. Measurements of the voltage level and currents on both sides are therefore made on both sides of the converter.
The initial values for this unit are focused on the performance of the unit. It is here where the size of the storage capacity of the unit is decided and how much power as the unit can handle.

A figure of the model is shown in figure 6.3. All the energy stored in the unit is collected in a conductor which is place down in the right corner of that figure. The voltage level over the conductor will increase or decrease depending on if the conductor is charging or discharging. The energy level can however be estimated by using the equation 6.1.

$$E = \frac{C \cdot v^2}{2} \qquad \text{eqi (6.1)}$$

Where C is the capacitance of the conductor and v is the voltage level over the conductor.

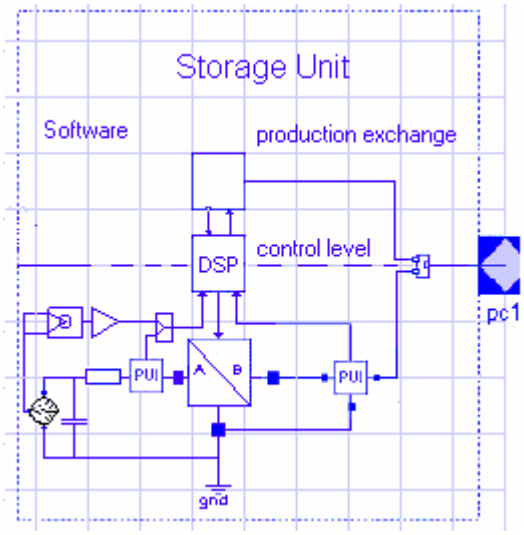


Figure 6.3. Picture of the storage unit.

That estimation part is done by the components that are placed left of and above the capacitor.

Otherwise, the major structure is very similar to the structure of the generator and the load. The communication block is exactly the same as in the previous units and will not be commented any further in this part.

The control part is a little bit different in comparison to the generator and the load. The set point that comes out from the control block will in this case be a value between –1 and 1. This is to allow the converter to pump power in both directions. The –1 value will then tell the converter to charge as much energy as fast as possible and +1 tells the converter to discharge as fast as possible.

The converter model itself is acting as a pump that is capable of pumping power in both directions. How much power that has been pumped is depending on the set point and how much power as the unit can deliver or receive, depending on mode of operation.

Exchange unit

This unit is very similar to the storage unit since it has to be able to have a bi-directional power flow. The main differences are that that it cannot store any energy and do therefore not have to have any components for that. Another thing is that it has to be able to have contact with two different DPSs, which makes the communication block on top of the unit a bit different from the other units.
A picture of the exchange unit is shown in figure 6.4. Here it is clearly viable that the communication block is equipped with another communication port to be able to handle the information that is coming from the other DPSs. More of how the communication block is handling the different communication configuration is discussed later in this chapter.

The controlling part for this unit is working in the same way as in the storage unit with a set point to the converter, which is between –1 and +1.
The converter model used for the storage is also used for this unit and do therefore not have to be discussed more in this part.

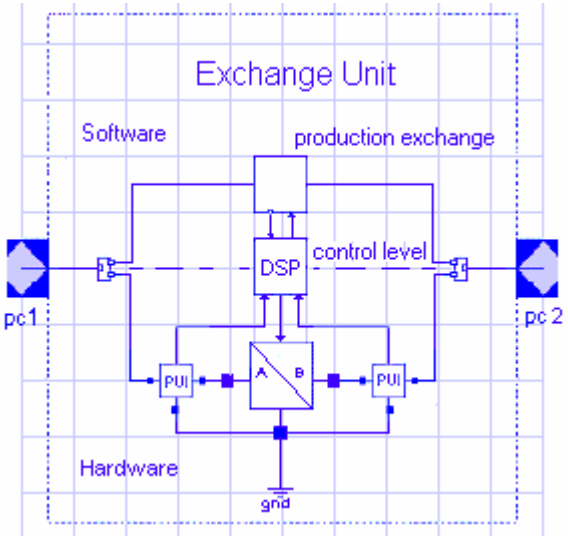


Figure 6.4. Picture of the exchange unit

Connection unit
The connection unit from in the DPS model is there to simulate a transmission line in the system. This is to separate the output conductor on the different generating and consuming units from each other and to make a more realistic model by adding some line characteristics to the system. A little bit more detailed picture of the connection unit is shown in figure 6.5

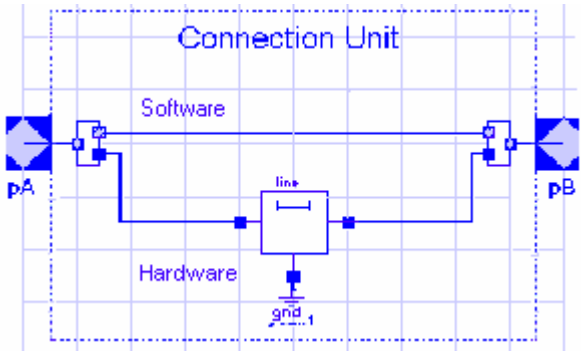


Figure 6.5. Picture of the connection unit

From that figure it is clearly shown that the communication line and the power line is going side by side between the different units. But before it is possible to add the line characteristics the power line has to be separated from the communication line.
All of the line parameters, such as resistance per meter and capacitance per meter and length of the cable, is set in the in the block in the middle of the figure.

When the characteristics have been added the two lines is attached to each other again to be sent away to the next unit.

Terminal unit
This is one of the most important units in the whole DPS. It is this module that every unit in the DPS has to be connected to be a part of the system. The terminal is shown in figure 6.7.

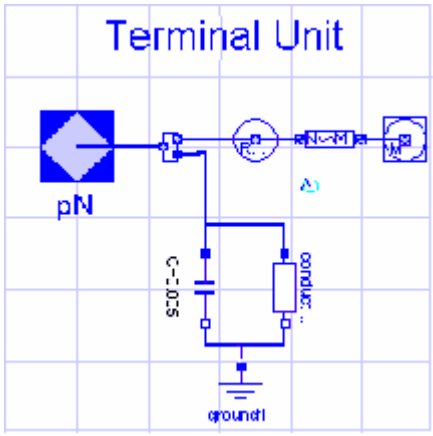


Figure 6.7. Picture of a terminal unit

In the simulation model the terminal unit has a very simple layout. First of all the power line and communication line is separated from each other.
The power line goes strait down to a capacitor that keeps a stable voltage level in that point and a conductor, which will add some more losses to the system.

The communication line goes to three blocks that together form a power flow, which is the connection to the entire controlling program. These three blocks will be further disused later in this chapter.

Interconnections and Interactions
All of the control algorithms and information system programs are being handled from programs that are written in C++ code. This code has to be linked to the simulation model in one way or another. In this work it is done with so called power nodes and power flows.

Power Flow
The power flows is placed in the communication block in the top of each producing or consuming unit, which is mentioned above. All of the signals that enter a unit have to go through these blocks. When a signal enters the communication block it is directly connected to the power flow, which is linking the signal to the production exchange program. Pictures of how this communication blocks looks are shown in Figure 6.8.
The signals have to go through some rectangular blocks, witch are called DFlows before they can attach to the powerFlow. The round block in the base of the powerFlow is called DNode and the purpose of this block is to keep track of all of the DFlows that is connected to it and send the signals from the DFlows to the PE module.

Two types of communication blocks are shown in the picture. The first one is used for the generators, loads and storages and the other one is used for the exchange unit.
The component within the dashed line in the centre of the figure is forming the power flow and all of the connections are connected to this module. It is the head of this module that is connecting the simulation program to the coordination programs. How this works is not a part of this report to describe. That part is described in a Phd project, which will be ready within a year.

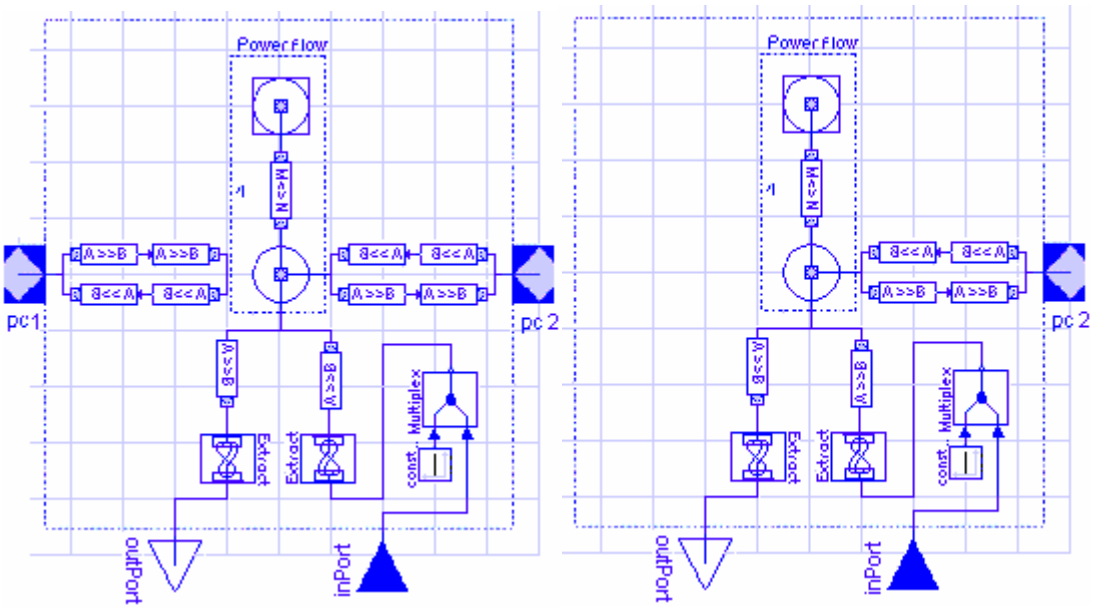


Figure 6.8. Pictures of the two different types of communication blocks for the units. The right one is in the exchange unit and the left on is for the other units.

The program for the power flow can however connect an unlimited amount of connections, so there is no need for adjustment when new connections for the exchange unit are made. The connections in the very right in the pictures belong to the power node. The reason to why they are placed here is that there is no fix number of connections connected to the power node. So if they are placed in the units, a new set of connections will be added when a new unit is added.

Power Node

The power node is placed in the terminal unit and looks very much like the power flow. It is also working in a very similar way as the power flow. The only difference is that the program is linking the signals to the production management program.
As discussed before, there are no connection blocks placed in the terminal unit. They are instead placed in each unit. That is to make it easier to connect and disconnect new units since the number of connections then will be of the right amount.

6.1.2 C++

The key to manage the dynamic design interconnected to a simulation environment (SE) is to use higher levels of abstraction as in object-oriented languages. An example for this approach is implemented in a C++ class library based on implementation corresponding to the DNode and DFlow models used in the Modelica layer.
The way that the Model in a simulation program is linked to the C++ classes is shown in Figure 6.9.

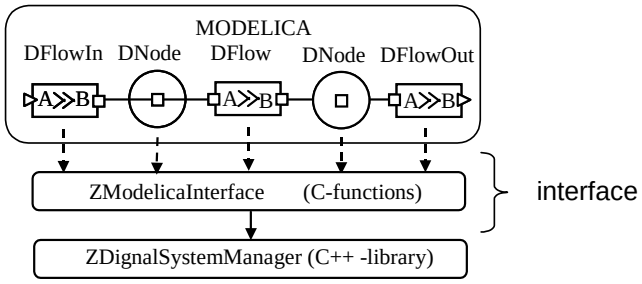


Figure 6.9. Modelica and C++ interconnection

All of the Modelica blocks calls on an interface class, ZModelicaInterface, written in C++. By using the ZModelicaInterface, the Modelica function calls are translated to method calls from the ZDataSystemManager class where all methods can be found in each step interfacing the C++ class library.

Using the C++ class libraries, a designer can describe components at a broad range of abstraction levels, which result from the ability to perform signal and control modification separately.
There are different levels of abstraction at which C++ can be used for the signal management system as depicted in Figure 6.10

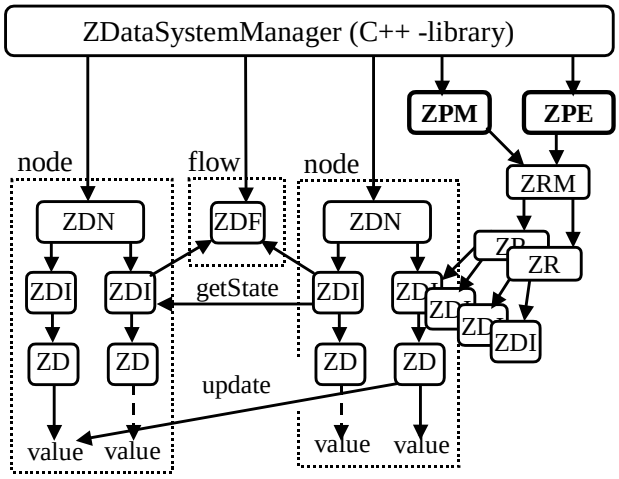


Figure 6.10. C++ structure

The C++ library mainly includes the following classes:

- The *ZDataNode* (ZDN) class is responsible for searching and joining together every signal in each connected input and output data interface. The class is receptive to changes of new interconnections and continuously controls all connections for not being defected.
- The *ZDataInterface* (ZDI) manages a pre-defined number of signals in a block that, for example, could correspond to a protocol. This object could be of several types such as inputs and outputs for communication but also parameters for configurations and specifications.
- The *ZData* (ZD) class is the object containing the particular value of the signal and its configuration. It could also be a reference pointing at another ZData.
- The *ZDataFlow* (ZDF) class is the actual configuration of the signals between two DNodes including the needed types and identities.
- The ZResourse (ZR) is the actual recourses in the system and is using the ZDataInterfaces to communicate with the rest of the system.
- The ZResourseManager (ZRM) class is responsible to keep track of all the recourses and make sure that the every recourse gets the value as it should have.
- The ZProductionManagement (ZPM) class is the coordination module in the system and decides what command signals each resource should have.
- The ZProductionExchange (ZPE) class is the last check that each resource will not get any command signals that is beyond the unit limits.

Exactly how this complex program structure works can be studied in the reference [4]. This report will only focus on the two blocks ZPE and ZPM in figure 6.10. These two blocks shows how the production exchanges module and production management module.
As shown in figure 6.10, the class ZProductionManagement module points at the class ZResource to get its calculated command signals for the resources. It is also from the ZResource block it checks in what status the resources are working in.
The most important functions in the ZProductionManagement are shown below.

```
class ZProductionManagement
{
private:
  double SPgen;
  double loadSP;
  double storSP;
  double exchSP;
  bool lowVoltage;
  bool enablePowerGeneratingUnits;
  //---------- constructors -------------------------------
  ZProductionManagement(double cv[], int cvSize);

  //----------- operational functions ---------------------
public:
```

```
  void       setValue(int index, double value);
  void       setValues(double* value, int valueSize);
  void      analysePowerCapacity(void);
  void      securityUpdateSequence(void);
  void      decidePowerSetpoints(void);
  void      calculatePowerCapacity(void);
  void      pGenerationCapacity(void);
  void      pLoadCapacity(void);
  void      pStorageCapacity(void);
  void      pExchangeCapacity(void);
  void      analyzePowerCapacity(void);
  void      updateSecurity(void);
  void      loadManagement(void);
  void      genManagement(void);
  void      storManagement(void);
  void      exchManagement(void);
  void      update(double timeStamp);
}; //end ZProductionManagement
```

The functions above are taking care of the tasks that is explained in chapter 3.6.1 that explains the flow scheme of the production management program.

The ZProductionExchange also use the ZResource class to read the command signals from the ZProductionManagement.
The ZProductionExchange then evaluate these command signals before they are forwarded to the recourse itself. The function that manages these operations is shown below.

```
class ZProductionExchange
{
private:

  ZString* m_objName;
  ZResourceManager* m_rm;
  bool m_rmOK;
  bool m_simOK;

public:
  //---------- constructors --------------------------
  ZProductionExchange(int cv[], int cvSize);

  //----------- operational functions ----------------------------
  void       setValue(int index, double value);
  void       setValues(double* value, int valueSize);
  void       unitSecurityUpdate(void);
  void       update(double timeStamp);
```

}; //end ZProductionExchange

The setValue and getValue functions are used to write and read from the ZResourceManager and the unitSequrityUpdate is called every time there are some new command signals that have to be checked.

6.2 Simulations.

This part is done to verify that all of the control programs and algorithms described above are working correctly. The scenarios that are tested here have been described in chapter 5. It is much better to discover failures in the system in simulations then in the laboratory and the simulations gives a great overview of all of the signals that is sent around.

6.2.1 Simulation 1

This first simulation is to make sure that all connections between Mmodelica and the program code is working as it should and to show how the startup sequence of the DPS looks like.
This system does only have one generator and its only purpose is to set the voltage to the system to the nominal value, which is 150V.
The result of the simulation is shown in figure 6.11.

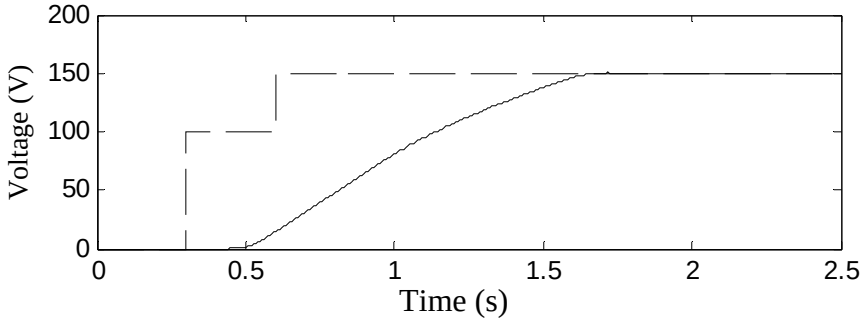


Figure 6.11. Result of simulation 1.

The dashed line shows the voltage set point for the system and the solid line shows the actual voltage in the DPS.
The generator is controlling the voltage in droop mode. The droop mode controller does only have a proportional part that normally causes a stationary error from the set point. But since there is no load in the system this error is very small.

The startup sequence of the generator is shown in figure 6.12

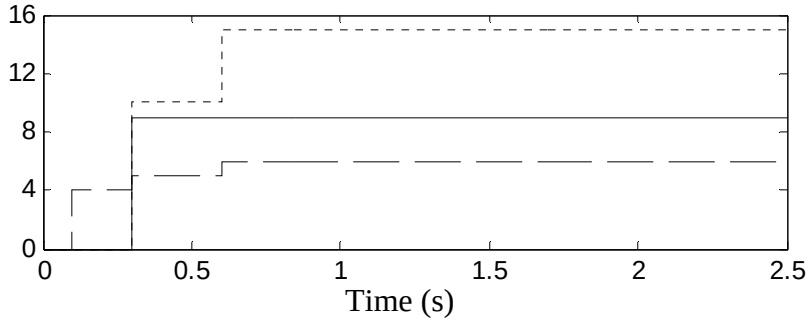


Figure 6.12. Result of start up signals from simulation 1.

The dashed line shows the command signal. It is this signal that orders the unit if the unit should be on ready or off. The solid line is the govern signal that tells the converter what control mode as it is supposed to operate.
The dotted line shows the set point that is sent to the unit. In this plot is the set point scaled down with a factor if 10 to make it fit in this figure.

The first thing that happens is that the unit gets a command signal that order it to be off, se the dashed line. The value of that signal is 4, and is sent to the unit with a delay of 0.1 seconds because that is the time it takes to registry the unit in the DPS.
Then the command signal order the unit to go to its ready state, represented by value 5. At the same time the govern signal changes value to 9 which order the converter in droop mode when started. The reason to why the set point is set to 100V at first is to make adjustments for when the system is going to be tested on the physical units.
The final step is that the command signal changes to the value 6, which order the unit to start. At the same time the set point will change to the nominal voltage value of the DPS.

That is the normal procedure to start up a unit. Units that is started up in power control mode will get a different govern signal value and the set point will go to its final value at once.
once.

6.2.2 Simulation 2

This scenario is following the description of the scenario 2 that is described in chapter 5. The main reason for simulating such a system is to verify that all of the algorithms that later are going to be used for the tests on physical units.
This test is mainly done to check that the control programs for the load are working and the result of the test is shown in Figure 6.13.
The load has a sinusoidal need for power that has an offset of –500 W and an amplitude of 300W, the frequency is 0.05 Hz. The maximal power as the generator can produce is fixed to 600W.
The reason for choosing these power and voltage levels is to make this simulation as close to the coming test on the real converters as possible.

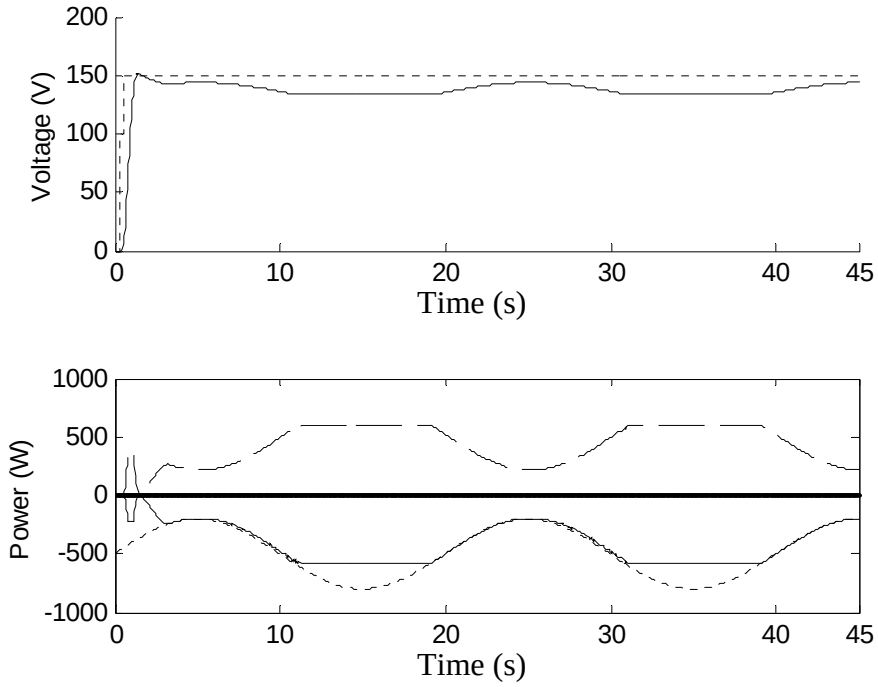


Figure 6.13. Result of simulation 2

The upper figure in Figure 6.13 shows the voltage in the DPS. The dotted line is representing the voltage set point in the system and the solid line shows the actual voltage. Here it is clearly shown that the generator is controlling the voltage in droop mode. Since the droop mode controller only has a proportional part there will be an error, which is proportional to how much power that the droop mode unit have to deliver to the DPS.
The lower figure in Figure 6.13 is showing the power values in the DPS. The dashed line shows how much power as the generator delivers to the DPS and the solid line shows how much power as the load consumes. The dotted line is representing how much power the load wants to consume.
As described in chapter 5.2 the maximal power the generator can produce is equal to 600W and the figure shows that it never produces more then that. The generator is nicely following the power, as the load wants to consume until the load wants to consume more power then the generator. When that happens, the set point for the load will be locked to 600W, which is on the capacity limit of the generator.
The result of this test looks good and this scenario can be tested on the physical converters, see chapter 7.

6.2.3 Simulation 3

As described in chapter 5 does this scenario contain 3 converters. Two of them will act as generators and the third one will act as a load. The specification of the load will be exactly as in the last scenario i.e a sinusoidal power curve with an offset of –500W and an amplitude of 300W and a frequency of o.o5 Hz. Both of the generators have a maximal capacity of 600W.this simulation is done to verify that the generation management program described above works as it should. The generator in working in

power controlling mode is supposed to keep the droop mode unit at half of its capacity, in this case 300W.
The result of this simulation is shown in Figure 6.14.

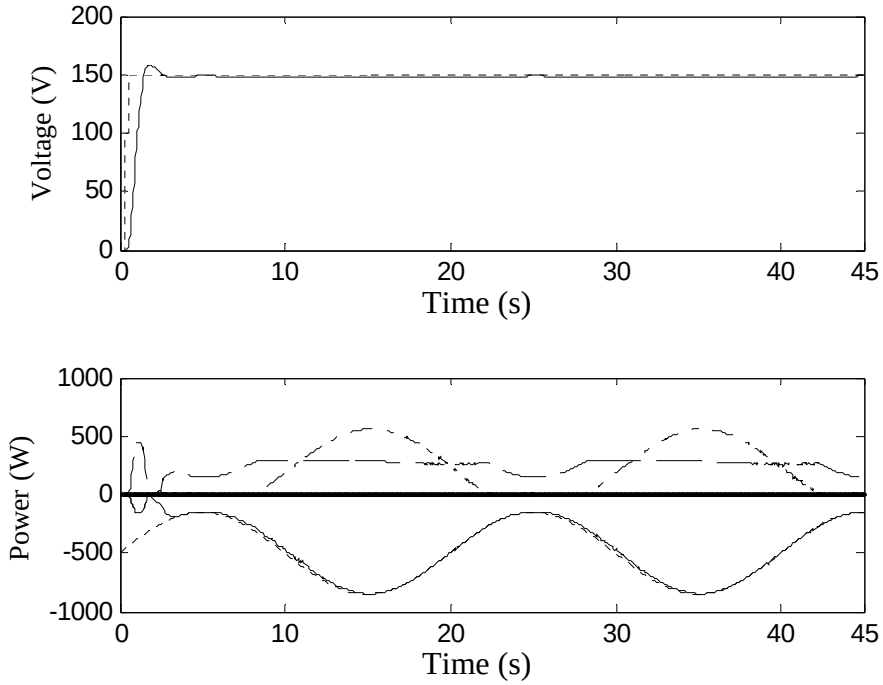


Figure 6.14. Result of the simulation of scenario 3.

The upper figure in the figure shows the voltage values in the DPS. The dashed line in the upper figure shows the voltage set point and the solid line shows the actual voltage. The properties for the generator working in droop mode are exactly the same as for the one in the previous simulation. But the voltage level does not vary that much because of the second generator.
The lower figure shows the power values in the DPS. The dashed line shows the power produced by the generator in droop mode and the dashed-dotted line shows the power produced by the other generator. The solid line shows the power that is consumed by the load and the dotted line shows the power that the load wants to consume.
Since the generators, together, can cover the load with power at all times, the load is able to get as much power as it wants. That is clearly shown in the figure where the solid and the dotted line are on top of each other.
It is also possible to see that the generator in droop mode actually is producing 300W almost all the time. The two dips in the power production of the droop mode generator are there because the load does not want any more power at those points of times. The thing that makes it possible for that to happen is that the other generator takes care of the rest of the power production just as it should.

6.2.4 Simulation 4.

This simulation is a simulation of scenario 4 from chapter 5. The scenario contains 3 converters, one generator, one load and one storage unit. The generator is supposed to do the voltage controlling part in the DPS. It has a variable maximal power capacity which is sinusoidal with an offset of 1000W, an amplitude of 350W and a frequency of 0.05 Hz. The load is a constant load with a power consumption of 500W. The storage has a maximal charging capacity of 150W and a maximal discharging capacity of 150W. The storage is chosen to be quite small in order to force the storage to become full, just to check that it will stop to charge when it gets full. The maximal storing capacity of the storage is 1000Ws, its minimal energy level when the storage is considered as empty is 200Ws and the initial energy level is the storage in 700Ws.
The reason for not choosing the minimal energy level to zero is in case the storage has a power overshoot when it is supposed to turn off from discharging. Then the storage would try to discharge from an empty storage, which would not be good.
The result from this simulation is shown in Figure 6.15.
The plot in the top of Figure 6.15 shows the voltage level in the DPS, the solid line shows the actual voltage level and the dotted line shows the set point for it. It is the generator that takes care of the voltage controlling part and it does it in droop mode.
The middle plot in Figure 6.15 shows the power values in the DPS. The dashed line shows the power produced by the generator and the dashed-dotted line shows the power caused by the storage and the solid line shows the power consumed by the load.
The dotted line shows the maximal power the generator can produce.
The plot in the bottom of the figure shows the energy levels of the storages. The dashed line shows the maximal storage limit, the dashed-dotted line shows the minimal storage level and the solid line shows the actual energy that is stored in the storage.

Since the generator is operating in droop mode, the system will try to keep the generator at half of its power-capacity. Trying to keep it like that, the storage will try to charge and discharge energy to compensate for the variations of the generator. In the beginning of middle plot is shown that the generator can produce a lot more power then the load wants and the storage will therefore start to charge. The storage will however become full shortly after starting to charge and its set point is set to zero.
The set point of the storage will remain at zero until after about 11 seconds when the maximal capacity of the generator has dropped a bit. The storage will then start to discharge to keep the distance between the maximal power that the generator can produce and the power that it actually produces at a safe level. It continues to do so until the maximal capacity level of the generator is high enough to allow the storage to charge again.

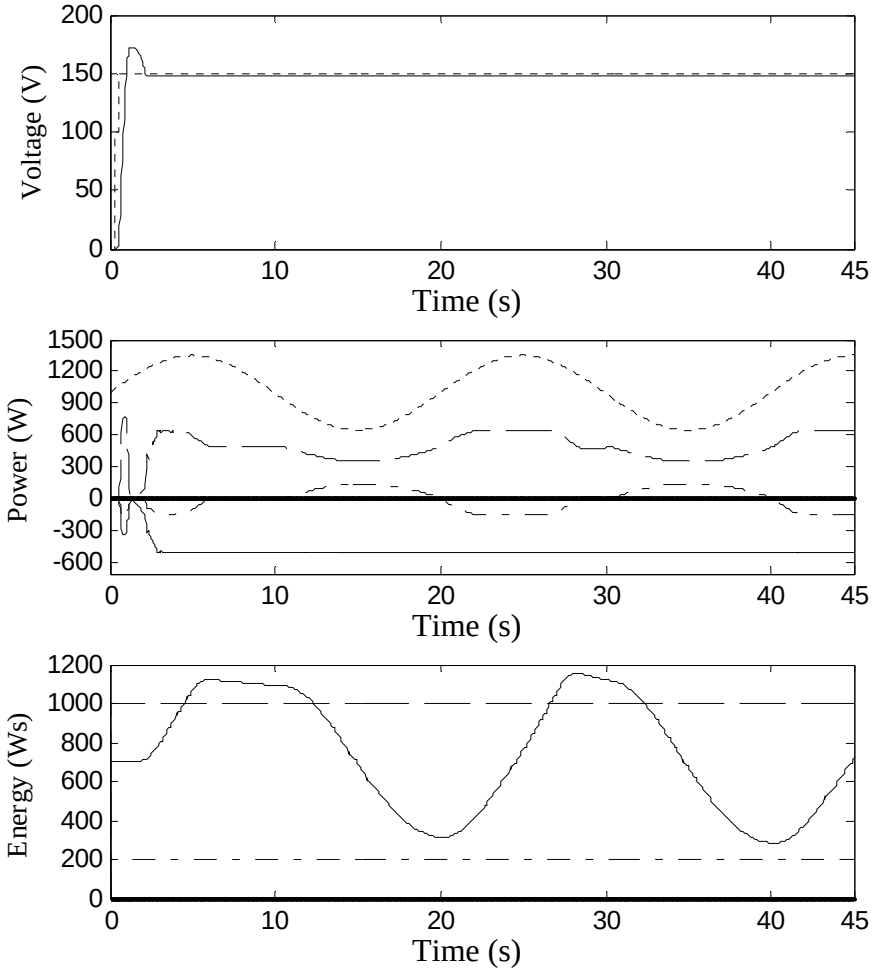


Figure 6.15. . Result of the simulation of scenario 4.

From the lowest plot in Figure 6.15 it is shown that the energy that is stored exceeds the maximal storage level. That is because it takes some time for the storage to stop charging when the set point is set to zero. It is not a strange thing and it is an expected phenomena.

6.2.5 Simulation 5

This simulation is supposed to simulate the scenario 5 described in chapter 5.
It contains 3 units, one generator, one load and one exchange unit.
The generator is supposed to handle the voltage controlling part in the DPS. It has a variable maximal power capacity which is sinusoidal with an offset of 1000W, an amplitude of 350W and a frequency of 0.05 Hz. The load is a constant load with a power consumption of 500W. The storage has a maximal import capacity of 150W and a maximal export capacity of 150W.
Just as in the in simulation 4,the system will try to keep the generator which is started in droop mode at half of its maximal capacity.
The exchange unit will always be able to export and import from the other system that it is connected to.

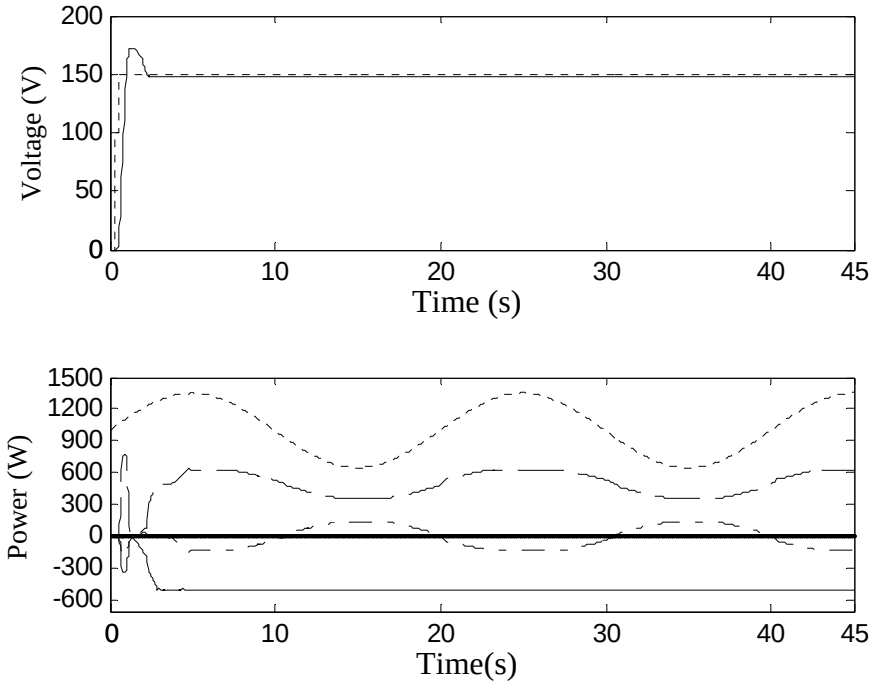


Figure 6.16. . Result of the simulation of scenario 5.

6.2.6 Simulation 6.

This simulation will simulate scenario 6 from chapter 5. The generators are of the same type with a constant maximal capacity of 600W. The load will however have a power capacity that is changing in time. The load will start to demand some power after 5 seconds and the demand will increase in a ramp like shape until it reaches 1500W.
The results from this experiment are shown in Figure 6.17.

The upper picture in Figure 6.17 shows the voltage in the DPS. The dotted line shows the voltage set point and the solid line shows the actual voltage. The voltage in this simulation is stable all the time, which is a good sign for well-balanced power levels.
The lower picture shows the power levels in the DPS.
The solid lines on the positive side of the power axis are showing the power produced by the generators and the solid line on the negative power axis is showing the power as the load is consuming. The dotted line on the negative side of the power axis is showing the power the load wants to consume.

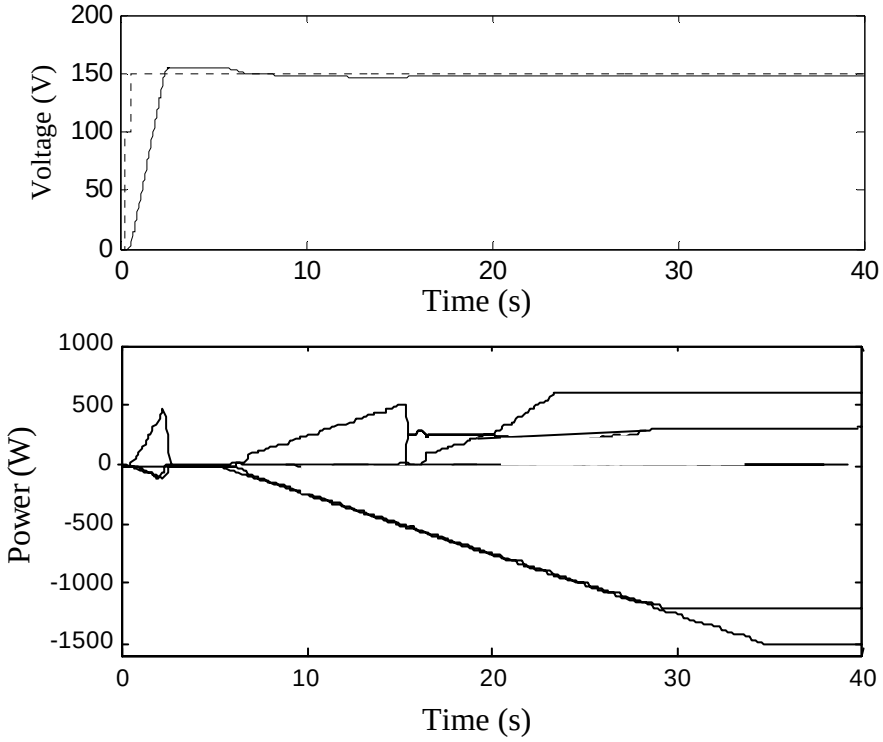


Figure 6.17. . Result of the simulation of scenario 6.

The first bumps in the picture are due to the power grid is charging up. 5 seconds into the simulation do the load start to consume power. To maintain the power balance in the DPS, the droop mode generator will add the same amount of power as the load consumes. When the generator has reached 80% of its maximal capacity will the next generator start. This generator starts in droop mode as well and will produce the same amount of power as the first generator. The droop generators should not produce more power then half of their capacity in order to maintain a good safety distance upwards to be able to cover each other in case one of them breaks. Because of that, the third generator will start shortly after the second generator. This generator will start in power control mode and the increase of power coming from that generator will follow the increasing of power as the load wants to consume. The third generator does however not produce more power then 600W since that is the highest value as the generators in droop mode can cover in case of a failure by that generator.
The load is not allowed to consume more then 1200W since that is the highest amount of power as the generators are allowed to provide.

6.2.7 Simulation 7.

This simulation is based on scenario 7 described in chapter 5. The simulation has 4 units and 3 of them are supposed to be loads and the last one is a generator. The 3 loads have constant but different power demands and they have different priorities as well. The load with the lowest priority has a power demand of 200W, the one with the middle priority has a power demand of 400W and the last load demands 500W.

The generator has a start capacity of 400W but that will be increased after a while to 1000W, stay at that capacity for a while and then decrease to zero. The result is shown in Figure 6.18.

The upper picture in Figure 6.18 shows the voltage in the DPS. The dotted line shows the voltage set point and the solid line shows the actual voltage.
The lower picture in Figure 6.18 shows the power curves in the system. The dotted line shows the maximal power as the generator can provide and the dashed line shows the actual power as the generator provides. The generator is started in droop mode and generates as much power as the loads wants, if it is within its capacity.
The solid lines on the negative axis show the power consumed by the different loads.

The bumps in the beginning of the picture are due to the charging of the system when the voltage rises. When the voltage is set, the loads will start to consume some power. But since the loads have different priority number will they get there power demand fulfilled at different times. The smallest load with lowest priority is the first one that is working on its full capacity, which is 200W. The generator can however produce 400W so the load with will consume the rest of what the generator can produce. When the power capacity of the generator increases, the load with the middle priority will use all of the extra power until its requirement is fulfilled. There after the load will be able to consume some power. The same thing happens when the capacity of the generator decreases, meaning that it is the load with the lowest priority that has to decrease its power consumption before while the others can continue to consume as much as they want.

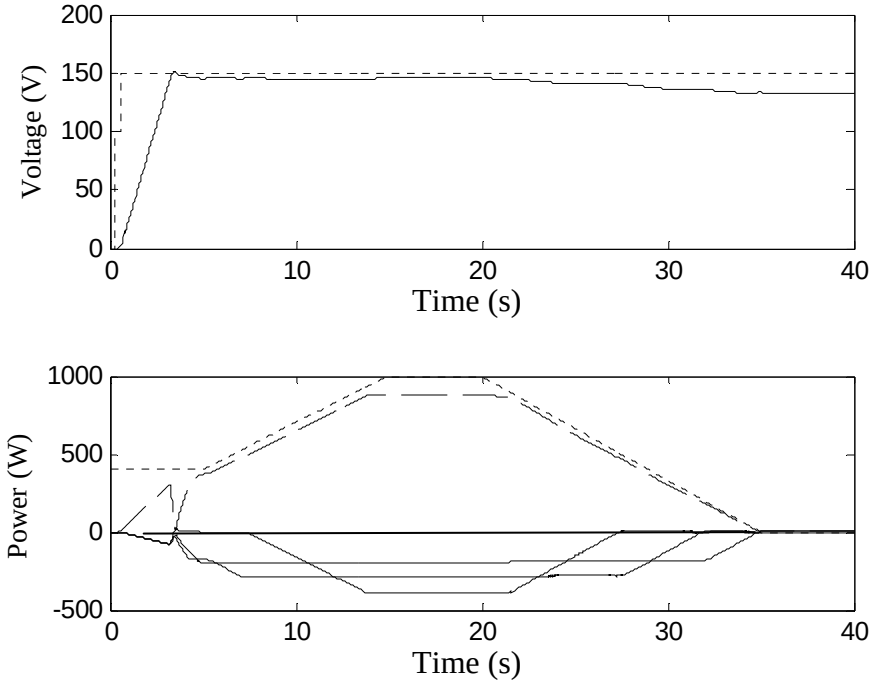


Figure 6.18. . Result of the simulation of scenario 7.

7 Experimental Validation

A small DC distributed power system is implemented on the basis of the result in chapter 3. The main reason for this setup is to verify that the simulations 1 to 5 in chapter 6 could be used to simulate a real system. Therefore the very same control algorithms used in the simulation are going to be used on a real system.
To be able do these in the laboratory, a communication system between the different converters has to be tested and implemented.
This chapter will explain how the communication system for the setup works and the results of the tests on the converters.

7.1 Communication System

An automated power system is very much dependent on a well-defined communication system. Disturbances can occur at any time and they have to be taken care of immediately.
It is therefore important that the units within the DPS have a great deal of coordination and cooperation.
The speed of the communication network is crucial to make the system dynamic~~al,~~ and special consideration has been taken to make it as fast as possible.

In this project there are a number of different kinds of information paths that have to be solved before a fully automated DPS can be set up in the laboratory environment.
The communication paths are:

- Converter-PC communication: This communication link is very important because without it the converter would be impossible to control. The communication is taken care of by a program in Matlab.
- PC-PC communication: The different converters have to be controlled by a computer. Since the production management program has to know what every converter is doing, all of the computers have to be able to communicate with each other.
- Matlab - C++ communication: The communication between all of the different units is done by Matlab but the control program is written in C++. Therefore it is necessary to find a way to use C++ files from Matlab.
- Matlab-Matlab communication: this is used when the production management and production exchange program are handled from the same computer. In this case it is unnecessary to use the global communication when it can be done with internal communication within the computer.

This part will describe the different communication paths in further detail.

7.1.1 Converter – PC communication

To be able to communicate with the converter a program named dSpace is used. DSpace works as an interface between the PC and the converter and it is through this program the Simulink program controls the converter. But dSpace could also be used to change set points and different variables that may be changed after the download is done. It is also possible to use dSpace to supervise certain values and signals to get information about what is going on in the converter.

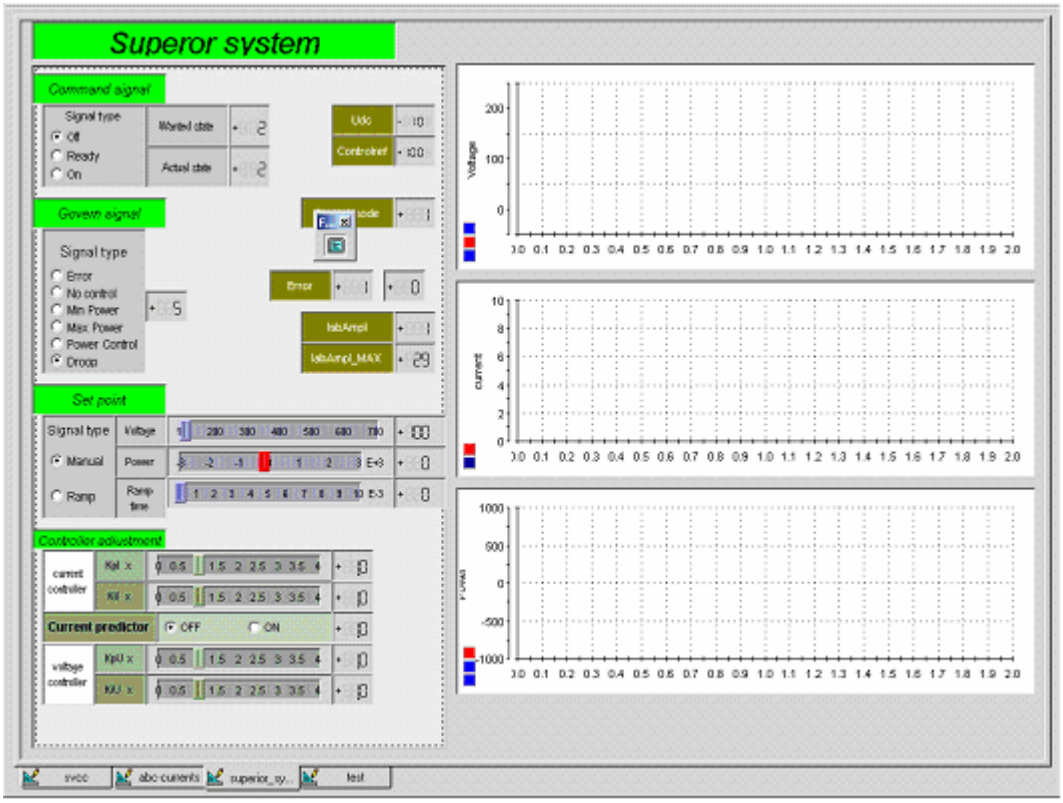


Figure 7.1. Picture of the surveillance window.

The most common and the most illustrative way to communicate with the converter is to use a program in dSpace that is called control desk. In that environment it is possible to build up your own control system if you want. The way I chose to build up my control system is shown in figure 7.1.

Up in the left corner it is easy to change the value of the command signal by simply clicking on one of the fields to tell the converter in which state you want it to be. It works in the very same way to change the govern signal. Using the mouse on one of the slide bars changes the value of the set point.
I have also chosen to show the DC-voltage, the DC-current and the power that is transferred by in the converter in the right side of the window. This is a good way to monitor the converter, however it is better to use graphs from the oscilloscope for presentation.

This is a good way to monitor and to change certain variables by hand. But a useful improvement would be to build a more automatized system with several converters. To make that possible it is necessary to be able to change the variables from another program written in some kind of high-level programming language.
There are basically two different kinds of methods that are possible to use. One of them is to use a library called CLIB, which is an interface between dSPACE and C. The other method is to use a library called MLIB, which is an interface between dSPACE and matlab. Both of the methods were investigated and the chosen method was MLIB, mostly because it is easiest to use.

The MLIB programs are quite easy to use as well. To use it the search path to the variable you want to find must first be traced and that is done by the following code.

```
mlib('SelectBoard','DS1104');
variables   = {'Model Root/Superior \ncontrol/Command/Value';...
               {' continue like this with whatever variables you want to find'};
var_desc    = mlib(' GetTrcVar ', variables );
```

Now when you want to change the values of the variables you have chosen, it is just necessary to write:

```
mlib('Write',var_desc,'Data',{The variables that have been initiated});
```

Or if you want to read them:

```
mlib('Read', var_desc);
```

Note that the constant var_desk is a number that tells the converter where it finds the variables that have been traced, and has to be used both when you write and read the variables.
Also keep in mind that the tracing part only needs to be done once, and it takes about 0.3 seconds to do it. But when it is done the reading and writing part takes less then 0.001 seconds to do.

7.1.2 PC – PC communication

Every converter is attached to one computer. Since the converters are supposed to transfer information between each other, it is easiest to let the computers to take care of the communication. The best way to implement communication is to use tcp/ip to be able to use the local network that connects all of the computers at the department. Since the rest of the programs, so far, are written in Matlab then it would be very good if it was possible to handle the communication from Matlab as well.
To do so, it is possible to use a library of m-files, which can be downloaded from the homepage http://www.mathworks.com/. This library takes care of tcp/ip communication from Matlab. The main file is written in C++ and the library has a lot of small m-file programs that uses the C++ file.

The files chosen from the library of m-files were:

tcpip_open	Opens a new tcpip connection.
tcpip_close	Closes an open tcpip connection.
tcpip_sendvar	Send matlab variable.
tcpip_getvar	Get matlab variable.
tcpip_servsocket	Creates a socket bonded to a port, waiting for connections
tcpip_listen	Checks/Gets connection connected to tcpip_servsocket .
tcpip_status	Returns status of open connection. Detects broken connections.

One important parameter, when it comes to communication, is speed. The first that was tested were the m-miles themselves, to see how long time each one needed to do their job. To do the tests only two computers were used, one to open the connection and to send and the other one to receive, see Figure 7.2. The result of this test is important in order to get an idea of the minimum time possible it takes to send a message. Each of these time tests were done 20 times in order to avoid faulty results that might be due to some unexplained disturbance in the system.

Figure 7.2. Setup for the first speed test.

The result from the test and a deeper explanation of the tcpip files will now follow.

Tcpip_open is trying to open a new connection to the other computer for the maximum time of 1.1 seconds. If it does not receive any answer during that time the program will continue to run, but it will not be possible to send any information to the chosen address. But with optimal conditions it takes less then one millisecond to set up the connection.

Tcpip_sendvar takes at an average 15.5 milliseconds to send a package with the size of one constant to a computer when the communication path is already made. If the file tcpip_sendvar is about to be executed while there is no open connection to the other computer, Matlab will immediately shut itself off and everything has to be restarted again. It is therefore important to make sure that there is an open connection to the receiving computer before trying to send something. The checking is done with the m-file tpip_status.

Tcpip_getvar needs in average 15.5 milliseconds to receive a package with the size of one constant.

The rest of the files tcpip_close, tcpip_servsocket, tcpip_listen and tcpip_status needs less then 1 milliseconds to use and are quite straightforward to use.

These times were done during a weekday and during work hours for the network at the university. The times are quite fast and will be able to meet the requirements for the small communication system for this work.

Now when we know that this program is fast enough to be used here, it is of interest to know how long it takes to send,-receive and store one message from one computer to another. The setup for the test is shown in Figure 7.2. The message was of the size of one constant and 300 messages were sent to get reliable results. The time needed for every message to reach its destination was saved in a vector to be analyzed.

The most important m-files used in this experiment for computer 1 were tcpip_open to open a new connection and tcpip_sendvar to send the package. The most important m-files for computer 2 were tcpip_listen to check if someone is trying to send something to the computer and tcpip_sendvar to receive what has been sent.
The most relevant times to show from these tests are the longest, shortest and the average time it took for one message to reach its destination. These times are shown in Table 7.1

Table 7.1

T average	15,6 milliseconds
T max	32 milliseconds
T min	15 milliseconds

The average time and the shortest time from table 7.1 are quite similar, which is good. It means that most of the messages are being sent at full speed, and that a few of them took slightly longer to reach their destination. These longer times can be explained with the fact that the tests are done on a communication network that is used by more computers than the one used in this test. Another thing that is important to note is that the shortest and average times are very close to how long it took for the m-files tcpip_sendvar and tcpip_getvar. That means that the only time-consuming elements were these files and they were working at full speed.

The next test was to check how long it took to send one package from one computer to another and then be sent back again to the first one, see figure 7.3

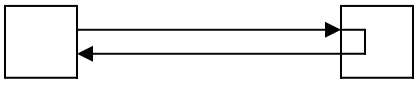


Figure 7.3. Setup for the second speed test.

This test was repeated 300 times in order to get reliable results. This experiment is basically done to test how fast how fast computer 2 can respond by sending its result back to computer 1.
The result of the test is shown in Table 7.2 and shows the minimum, maximum and the average time for one message to go both ways.

Tabell 7.2

T average	32,5 milliseconds
T max	344 milliseconds
T min	31,0 milliseconds

The average and minimum time for is about twice as large in comparison to the previous speed test, which was expected since the package has to go two ways instead of just one. But the maximal time it took to send one message was a lot longer then expected. But ~~on~~ a closer look at the results revealed that this was only on one occasion, and the most likely reason was that more then only these two computers were using the network.

The third test is conducted to see how well the program can cope with getting too much information. To try to drown one computer with information two other computers were trying to send their messages as fast as they could, see figure 7.3.

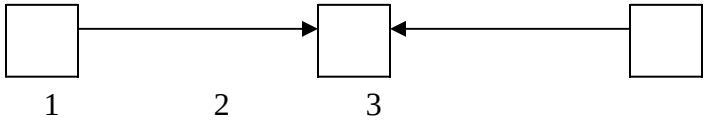


Figure 7.3. Setup of the overflow of information test.

Computer 1 was constantly trying to one constant to computer 2 and computer 3 was constantly trying to send another constant to computer 2. Both computer 1 and 3 was loaded with 300 messages to send to computer 2. Computer 2 on the other hand was saving all of the messages in a vector in order to see in which order the messages arrived. The times that was measured in this test was the average, minimal and the maximal time it took for each computer to receive or send one message, see table 7.3.
The program was written in such a way that both of the sending computers were trying to set up a connection with the receiving one. When one of the sending computers managed to set up a connection the other one can no longer do it. The computer that has the connection~~s~~ sends its message and when the message is received by computer 2 the connection is closed. Then a new competition begins between the sending computers to set up a new connection.

Tabel 7.3.

	Computer 1	Computer 2	Computer 3
T average (ms)	33,2	21,5	33
T max (ms)	562	437	1047
Tmin (ms)	15	15	15

An analysis of the order in which computer 2 received the messages was done and it appeared to be quite random. Sometimes there were 12 messages from either computer 1 or 3 in a row and sometimes just a few. This is of course not good since it would have been preferable for the messages to arrive in the order in which they were sent.

The average times were good and in line with what could be expected. The average time for computer 1 and three are about twice the minimal time. These minimal times are the absolute minimal time it is possible for ~~as~~ the m-files can accomplish, which is a good result. The maximal times for computer 1 and 3 cannot only be blamed on the fact that other people are using the web as well. Here we also have to consider that if computer 2 is receiving a message from one of the computers, the other one has to wait until the transmission is complete which takes some time.
But other people using the web can explain the reason for the very long time for computer 2. This theory gets some support due to the fact that only 2 times was around 400 ms, the rest was much shorter.

This coming test is quite similar as the previous one but this time another element is added: sending back the message to the source again, see Figure 7.5. This test is of great interest for this work. The communication procedure for the computer 2 will – if this test turns out well – be used in the PM module, and the communication program for computers 1 and 3 will be used for the PE module.

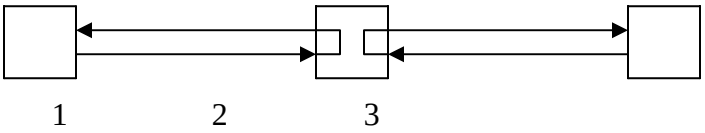


Figure 7.5. Setup for the first type of information system.

Just as in the other test, computer 1 and 3 do not send the same number as the other one to make it possible to know in which order the messages reaches computer 2. When they do reach computer 2 they will be saved in a vector for further analysis and then will be sent back to their source. Due to the fact that computer two is returning the messages as well, the system will be even more occupied. Since all 3 computers are trying to send information simultaneously, this means that the messages will collide with each other. This leads to bad communication times, shown quite clearly in Table 7.4

Tabel 7.4

	Computer 1	Computer 2	Computer 3
T average (ms)	86	43,1	64,7
T max (ms)	2218	1093	1656
Tmin (ms)	31	31	31

From table 7.4 it is possible to see that the average time for computer 2 has increased with 33% in comparison to test 2. However a drawback was that the maximum times it took for send a message were so very large. Times above 1 second are not acceptable. Also, the order in which the messages arrived to computer 2 was just as randomized as in the previous test.
The conclusion is that it is not possible to use this method since it is too unreliable to be used for converter coordination.

To be able to control the amount of information, it is better to let computer 2 send a message to any of the other two computers according to Figure 7.6. In this way the traffic on the web will be much more limited and easier to control.

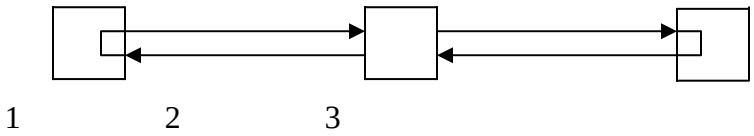


Figure 7.6. Setup for the second type of information system.

This system works like this. Computer 2 sends its message to computer 1 and waits for the answer. When the answer has arrived it sends a new message to computer 3 and waits for the answer, and so on..
Computers 1 and 3 have the same program and their only tasks are to wait for a signal from computer 2 and, when the signal comes, to respond by sending back whatever they are supposed to send back. The times for this system are shown in table 7.5.

Tabell 7.5.

	Computer 1	Computer 2	Computer 3
T average (ms)	79,1	39,4	79,2
T max (ms)	94	62	94
Tmin (ms)	78	31	78

From table 5.5 it is possible to see that the times for the computers are quite similar to the times in test 2, which is a good result, showing that the packages do not collide with each other. The times for computer 1 and 3 are also good; their times are quite fast and there is not a big gap between the shortest and the longest time.
This looks like it could be a good method to use for this project and since it is so stable it can even be expanded with a few more computers to be connected to computer 2.

7.1.3 Matlab – C++ communication

DSpace is controlling the converters and uses Matlab / Simulink as a base program for its controlling technique. It is also from Matlab that the communication between the converter and the computer is taking place. But the major control programs that are processing all the data and making the system work dynamically and efficiently are written in C++ code.
The Matlab programs are linked to the C++ programs according to Figure 7.7.

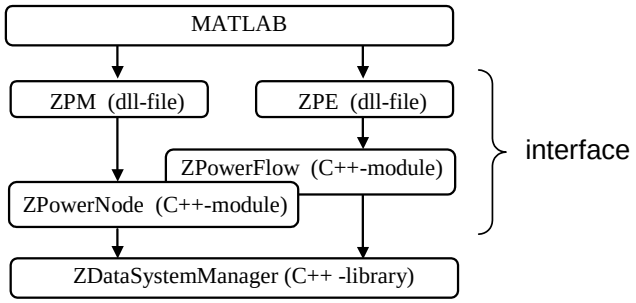


Figure 7.7. Matlab and C++ interconnection.

To be able to run that code from Matlab an interface class in C++ had to be created. That interface class then points at the functions that Matlab is going to be able to use. All of the C++ code had to be made as a .dll file to make it possible for Matlab to compile it.

There are mainly two different C++ classes that Matlab must be able to communicate with. Therefore two different .dll files have to be created.
The first one linked to the production management class. This class takes care of all the automation calculations and it is to this class that all the production exchange classes have to be connected.
The second one is linked to the exchange class. This class has to be used in every computer that controls a converter. This is because the controlling signals have to be processed through this class before being forwarded to the converter.

7.1.4 Matlab – Matlab communication

This communication link is only used when there are two Matlab programs running on the same time on one computer.
This communication is not controlled in any way, so both of the Matlab programs reads and write to the storages files as fast as they can.
There are two files that are used for the communication. One file is only used for communication from the production management part to tell the production management part what to do. The other file is used by the production exchange part to inform the production management its status. Each file is in other words used for a one-way communication link.

The files that are used are the standard functions fwrite and fread. These functions write and read data from a file on the computer. But before these functions can be used the file that is going to be used must be opened for read or write access. This is to prevent the file from being read and written to at the same time. The opening process is done with the function fopen.

7.2 Matlab

The whole system is controlled from Matlab in the laboratory tests. It is Matlab that takes care of the communication between the different computers and between the computer and the converter.

A picture for how everything is connected is shown in Figure 7.8.

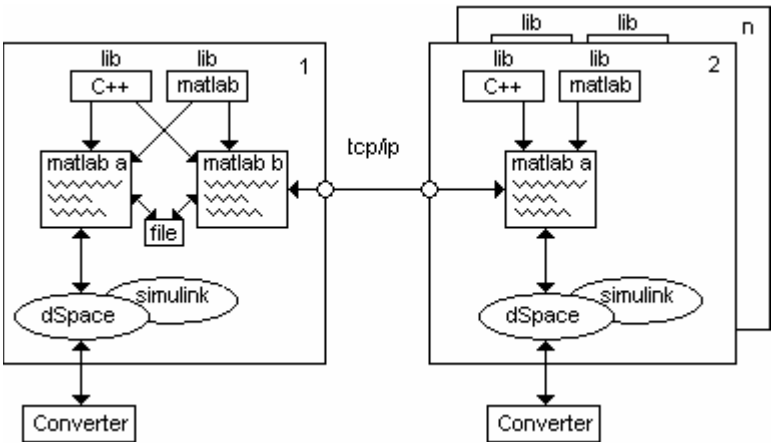


Figure 7.8. Schematic picture of the communication paths in the DPS

The converter is controlled through an interface program called dSpace. In order for dSpace to know how it is supposed to work a control program made in Simulink has to be downloaded to dSpace.
The large boxes in Figure 7.8 represent the different computers in the system. Computer 1 in the left side of the picture is described first and is also the one that takes care of the coordination of the system.
The best way to change some variables and to get feedback signals from the process while it is running is to use Matlab. The Matlab program a, in figure 7.8, is therefore continuously checking some values from the converter and updating others. It is this program that receives the information from the management program. The information is then linked to the production exchange class. The result from this class is then sent down to the converter and back to the management program.
To do this it uses a library of functions of both C++ and Matlab types.
Another Matlab program b, which is taking care of the automation and coordination part, is being run at the same time. It is this program that orders all the other programs what to do and when to do it. It is therefore from this program all the information to the converter is being sent, via a local file on the computer. It is linked to the production management class that is written in C++ and it is from this program the Matlab program gets the information that is sent to the other units.
Other computers i.e. computer 2 to computer n from Figure 7.8, can connect by using the TCP/IP protocols. These computers do however only need to have the exchange program running, since the automatisation part is done at computer 1.
A description of both of the Matlab programs is now going to be described.

7.2.1 Management interface

A flow scheme of the management interface program is shown in appendix B and an explanation follows:

The following section describes the flow scheme for the management interface program shown in appendix A, and subsequent discussions will refer to that figure. The initiation step will initiate some variables that are important to be set at a certain value before the program is started and open up a socket used for communication.

The next thing that happens is that the program listens to see if there is anyone that wants to connect to the system. It is unnecessary to start the main program if nobody is connected to it.
When a unit has finally connected to the program two data flows are created for the unit, one for input communication and one for output communication. These data flows are used to add the initial data that is sent by the unit to the production management program and its connection number will be added to the list of connections. Now the Matlab program can reach its main loop.

The main loop can basically be described in three parts and they are:

Data collecting.
System update.
Result sending.

The data collecting part is the tricky part for this Matlab program. First the program must check if the unit that it wants to collect the data from is attached to a remote computer, or is on the very same as the program is operating on.
If the unit is attached to another computer it first has to ask that computer for its data. Then it turns into listen mode and waits for the unit to answer. The unit answers by sending its current data and the data is stored. But if another unit answers instead of the one that is asked, then this means a new unit is trying to connect to the system. Its connection number is therefore added to the list of connections, data flows are created for the system and the initial data is collected and stored. When that is done, the program turns back to listen mode and waits for the data of the unit that was asked to send it. If the program has to wait for the responding unit for more then 3 seconds the unit is considered as disconnected and its contact number is removed from the contact list. That procedure is following the result from previous results where different communication structures were discussed.
But if the unit is attached to the computer where the program is executed from, there is no special communication technique. The program just reads and stores the data that has been written to the file and assumes that it is some recent data that it reads. After the data reading it checks if any new units are trying to connect to the system, if so it adds the new unit by adding its connection number to the list of connections, creates two data connections and the initial data is collected and stored.
This procedure goes on until the data from all of the connected units is stored.

The production management program does the system update part. It is here all of the computing takes place and it is this computing that decides what every unit is supposed to do. The program is written in C++ code and will be explained in another chapter.

In the result sending part, all of the results that were calculated in the system update part will be delivered to the corresponding units.

7.2.2 Exchange interface

The program that receives the signals from the management interface program is the Matlab program exchange interface.

The flow scheme for the exchange program is shown in in Appendix B
The program starts with some initial settings such as finding the search ways for the communication to the converter, and setting up the data flows for the production exchange program. Even some initial preparation for the TCP/IP communication is made in this stage. When that is done the program tries to connect itself to the management interface program. If it fails to do so the first time it tries again until it succeeds.
Then what happens depends on what situation the exchange interface program is in. If the program is executed on a different computer then the management interface program does get its command signal via a TCP/IP connection. If the command signal only contains 1 constant, it means that the exchange interface is supposed to send its values to the management interface program. But if the command signal contains 37 constants it means that this is some information about how the converter is supposed to work. The information is therefore stored and linked to the production exchange program. Then the information about how the converter currently is working will be stored and linked to the same program.
Thereafter the system will be updated and new command signals for the converter will be calculated and sent down to the converter. Even new signals that are supposed to be sent back to the management interface program are calculated but will not be sent until requested by the management interface.

7.3 Experimental Validation

A small DC DPS is implemented on the base of the result in chapter 6. The main reason for this setup is to verify that the simulation that are made in chapter 7 could be used to simulate a real system. Therefore the very same control algorithms that are used in the simulation are going to be used on a real system.

In the lab we do not have access to different generators, loads, storages or transmissions but we do however have several power converters to use. Since every power converter has the ability to make the power to flow in any direction through it, we can therefore simulate that each converter is one of the units mentioned earlier in this text. The power converter acting as generators are used to supply a local DC-network with power and to keep it on a stable voltage level by using a tree phase AC grid as a power source. At least

one other converter is acting as a load and is therefore taking power from the local DC-network and is dumping it back to the AC-grid. In these test we are going to use three converters and they are being connected. A schematic figure for how the unite are connected is shown in figure 7.9a and how the system looks in the real tests is shown in figure 7.9.b.

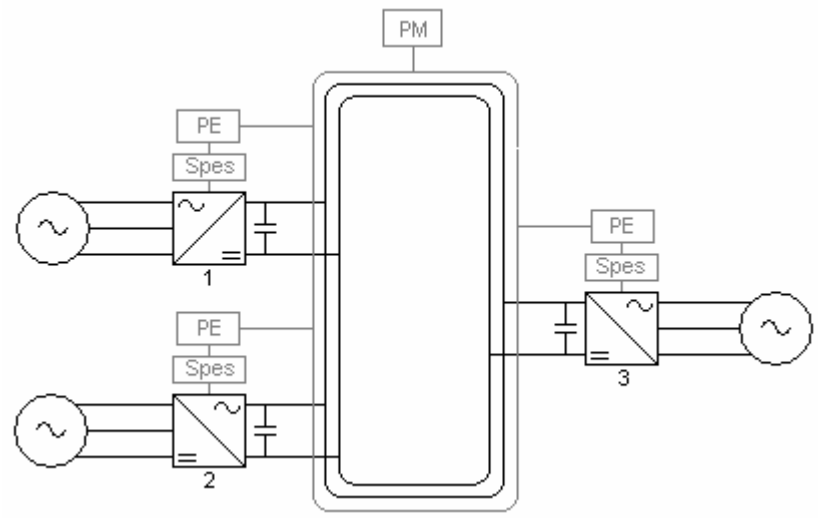


Figure 7.9. The configuration of the investigated DC DPS.

Since there are three converters in the setup it is possible to use the third one to be another generator or a storage store some energy when there is excess in the system or as an exchange unit. How this third unit is chosen and what part of the system as this unit will take in the system will be further discussed later in the report.

The converter must get some information about what kind of unit it is suppose to be and the block Spes in Figure 7.9 provides that information. Spes gives the information as a number and it also inform the converter where the upper and lower power limits are, how fast it is allowed to increase its power, what its nominal voltage level is and if it is a storage it also give information of how much energy it have in storage.

The block PM contains all of the automation information and algorithms. Its purpose it to make sure that the DC-grid have correct voltage level and it is trying to support the loads with as much power as possible with the generators that is connected. It is this block that decides what state the converters is supposed to be in and what set point each converter should have when they are in their active mode.

The PE block do finally provide the converter with the signals needed to make it do what it is supposed to do at a certain point of time. It never gives any set point that is higher than what is allowed from Spes.

7.3.1 Experimental setup.

The setup of this experiment is formed by three-phase converters of the type that is described in chapter 5. The converters are connected to a 135 V, 50 Hz three phase AC grid via a transformer and three inductors, one per phase according to Figure 7.10

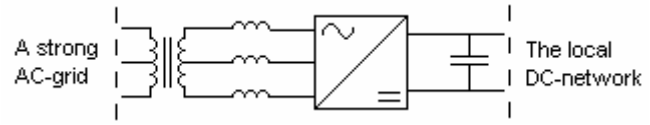


Figure 7.10. Shematic picture of one converter setup.

The transformers are three-phase transformers and are there to create a galvanic isolation between the AC-grid and the converters itself. This is very important thing to do because the controlling part of the converter is feed by a separate power source and if none of them are galvanic isolated they could add two different ground points to the converter and that can create problems.
The transformers are connected in YNy with the voltage ratio of 235 / 75. The voltage level on the secondary side has to be that low since the maximum allowed DC voltage is 200V. As discussed in previous chapters the minimal DC voltage that occurs on the DC-side is determined by the amplitude of the voltage on the AC-side by the ratio of V_{DC}/V_{LL} = 1.35. In this case the DC-bus voltage will be equal to 100 V for the converter line-to-line voltage 75V. The voltage on the DC-side therefore has to be between 100 and 200 volts.
The maximal rated power for the transformers is 2000VA, so this will set the limit for the power that can be added or withdrawn from the AC three-phase source. The line side filter inductors are inserted between the converter and the transformers are each of the size 3 mH.

Before all of the capacitors have reached 100 Volts it is not possible to control the current that enters the first converter that is turned on. Therefore is the converter that is supposed to start up the DC-system equipped with an inrush current protection device to protect diodes in the converters. The layout for this device is quite simple and an overview of it is shown in Figure 7.11

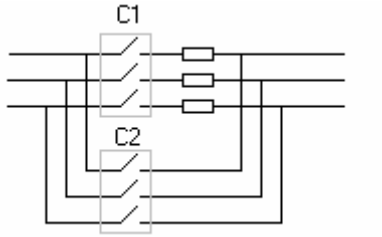


Figure 7.11. Schematic picture of the current inrush protection device.

At first when the DC power system is off and the voltage level is zero, both of the contactors C1 and C2 are open and do not allow any current to go through.
When the system is supposed to be turned on, the contactor C1 is the first one that gets shut and forces the inrush current to go through the resistors, which reduces the current. The resistors are of the size 5 Ω each and the current through them will not be higher then 20 ampere.
After about 0.2 seconds when the voltage has reached 100 volts on the DC side, the other contactor closes and shortcuts the resistors.
There is only one signal that is needed for this operation. The same signal that closes the first contactor C1 is delayed with a time relay and closes the other contactor as well. The time delay is set 0.2 seconds because that is the time needed to charge the capacitor to 100 volts.

The channels used for this operation is one of the digital channels that is available from the dSpace/Simulink interface blocks in Simulink.
There are a total number of 16 digital channels that could be used to send signals to and from Simulink and dSpace. But the dSpace interface card can be designed to handle 4 digital out signals and 4 digital in signals.
Two of the digital output signals (i.e digital output signal 15 and digital output signal 16) are locked to be used for special purposes.
Digital output signal 15 is dedicated to be an enable/disable signal for the inrush current device described above. The contactors in the inrush current devise needs 15 volts to shut. The signal that enters the dSpace interface card is either 0 or 10 volts. The signal that enters the dSpace card is therefore feed to a transistor via an optocoupler to be amplified to 15 volts.
The digital output (DO) signal 16 is dedicated to enable or disable the transistor switching. The best node to connect this signal on, is the node that is used by the over voltage and over current protection system on the card. The voltage level that is needed to turn on the switching is 9.2 volts. But to avoid disturbances between the card and the computer, the signal is first feed to a voltage-follower and after that it goes to an operational amplifier that amplifies the signal to 9.2 volts and the switching starts.

Two of the phase-currents, the flux vectors in αβ coordinates from the AC-side, the DC voltage and the DC current are being measured and sent back to the control program. Since the load is a symmetric load, it is only necessary to measure 2 of the 3 AC currents, the third one can be calculated from the other two.
There are a total number of 8 channels that can be used as an analogue to digital communication with the control program. 4 of these channels are synchronous sampled channels and 4 are asynchronous sampled. Since it is important that the AC-currents are measured at the same time, 2 of the synchronous channels are used for that. The other four measurements that are sent back are asynchronous sampled.

7.4 Tests.

The main purpose to do these tests is to test how well the control algorithms work on a small system that exists in reality, and to see how flexible and dynamic the system is.
The tests that are made on the laboratory setup are based on the first 5 scenarios described in chapter 5.
Since the maximal voltage allowed on the system is 200 volts, and the minimal voltage is 100 volts, 150 volts would be a suitable voltage level for these tests.
The efficiency for the converter at this voltage level was measured to be approximately 0.66.
Since the transformer is limited to handle higher power levels then 2000W, it means that the maximal power that can be feed to the DC DPS is 1320W. To have some safety margins for the power levels, no tests that have been done have used levels higher then 800W.

7.4.1 Test of scenario 1.

This first test is a startup test but since this test uses real components there are things that can go wrong. To avoid that the startup is done manually the first time in order to make sure that everything works properly.
In this scenario not only the control program and its different control modes are checked, but also the communication between dSpace, the computer and the enable signals that take care of the inrush current devise and transistor switching is tested.
It is therefore important to go through all the steps in the list below before the converter can be used in the automatized system. The way that the converter is controlled in manual control with the control desk, is described in chapter 7.1.1. The manual startup is done with a resistive load of 50Ω connected in parallel with the capacitor; which is done to make it possible to test the power control mode of the converter.

1. The command mode of the converter changes state from state off to state ready. In the state ready an enable signal is sent to the inrush current device. This step makes sure that the voltage on the DC side of the converter reaches 100 volts without too much stress on the diodes of the converter.
2. The govern mode of the converter is set to voltage control, the set point is set to 100 V and the control mode of the converter is changed from ready to on. When

the converter is in its on state, an enable signal to the switching of the transistors goes on and the switching starts.

3. The set point for the voltage is now changed to 200 volts and then changed back to 100 volts again. The actual voltage over the capacitor should now follow the set point without any noticeable delays.
4. The govern mode of the converter is changed to droop control. This control mode is also supposed to control the voltage on the DC-side but the control algorithm does not have any integral part, which will lead to a small stationary error to the set point. The test of this control mode is made in the same way that is described in step 3.
5. In this step the power control part is going to be tested. Since the voltage level has to be between 100 and 200 volts and the load is 50Ω then power delivered from the converter has to be between 200W and 800W.
6. The govern signal is changed to power control and the set point for the power is set to 200W. The power set point is then slowly changed to 100W and then changed back to 25W.
7. The command signal for the converter is finally changed back to off. Now both of the enable signals go off, which turns off the switching and disconnects the inrush current device.

When the converter has been tested with all of these tests and shows good results, it is considered to be working properly and can be used in the automatic tests. Every new converter has to go through these tests before it is allowed to be used in the more advanced scenarios later on.

Now an automatic startup is going to be tested to show what the startup looks like. The test is done-on one computer. But since both the production management and production exchange have to be running at the same time, and both of them have to he controlled by Matlab, there have to be two Matlab programs running at the same time.
First the production management program is started. When it has started and is waiting for any units to contact it, the production exchange program is started. Now all the contacts are made and the setup looks like the one that is simulated in simulation 1 in chapter 6.
The result of the experiment is shown in Figure 7.12

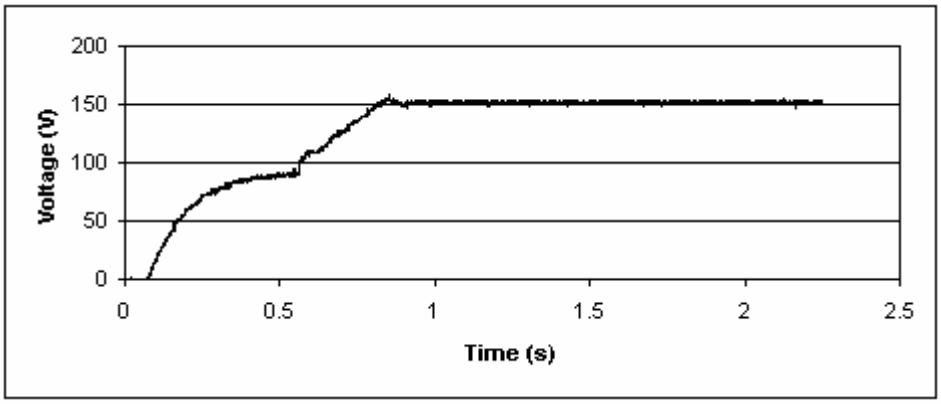


Figure 7.12: Measurements of scenario 1.

In Figure 7.12, it is clearly possible to see that between the times 0.1 seconds and 0.55 seconds the converter is in its ready mode because it is when the capacitor is charged up through the resistors in the inrush current devise. By the time 0.55 the converter turns to on state and assigns the set point 150 volts, which is the nominal voltage for this system. The control algorithm that is downloaded to the dSpace does however not allow rapid voltage changes and it will limit the step change in the voltage set point with a ramp with the slope of 200 volts/second. That is why it takes some time for the voltage to reach its set point at 150V.

The result of this test shows that all of the communication links within the computer work as they are supposed to. But the TCP/IP communication is not tested enough with this test, since the communication between the production management and production exchange is taken care of with a local storage file on the computer.

7.4.2 Test of scenario 2.

From scenario 2 in chapter 5 it is shown that there should be 2 converters: one that controls the voltage of the DC DPS, and another one that creates a disturbance by taking some power from the DC system.
This means that the second converter has to be able to put back some power into the AC-grid. The operation of that is not tested yet so this had to be tested manually. The manual power control test from scenario 1 only tested the possibility to take power from the AC-grid and not the other way around.
To check that it is possible to run the power from DC system to the AC grid both of the converters were started manually. The first converter is started in droop mode and controlled the voltage on the DC power system to 150 V. When the voltage was set the other converter was started in power control mode. The power set point was then smoothly decreased from 0 down to minus 800W. The test showed that there were no problems with this mode of operation and this two-converter system was now ready to be tested automatically.

Just as described in the specification of scenario 2, the maximal power that the converter acting as a load wants to consume, will change with respect to time. The change will be a

sin-shaped waveform with an offset of 500W, an amplitude of 300W and a frequency of 0.05 Hz. The generator on the other hand is set to have a maximal out power capacity of 625W.
The result is shown in Figure 7.13. The upper picture shows voltage in the system. The black curve in the Figure 7.13b is the power delivered to the system by the generator, the dark grey curve in Figure 7.13b is the power consumed by the load and the light grey curve is the maximum power that the load wants to consume.
Note the difference in the voltage level in the system when the generator is working hard in comparison to when it is working less. The voltage level differs a bit since the generator is in droop mode.
Just as was showed in the simulation, the load was not allowed to get more power then the generator was able to deliver.

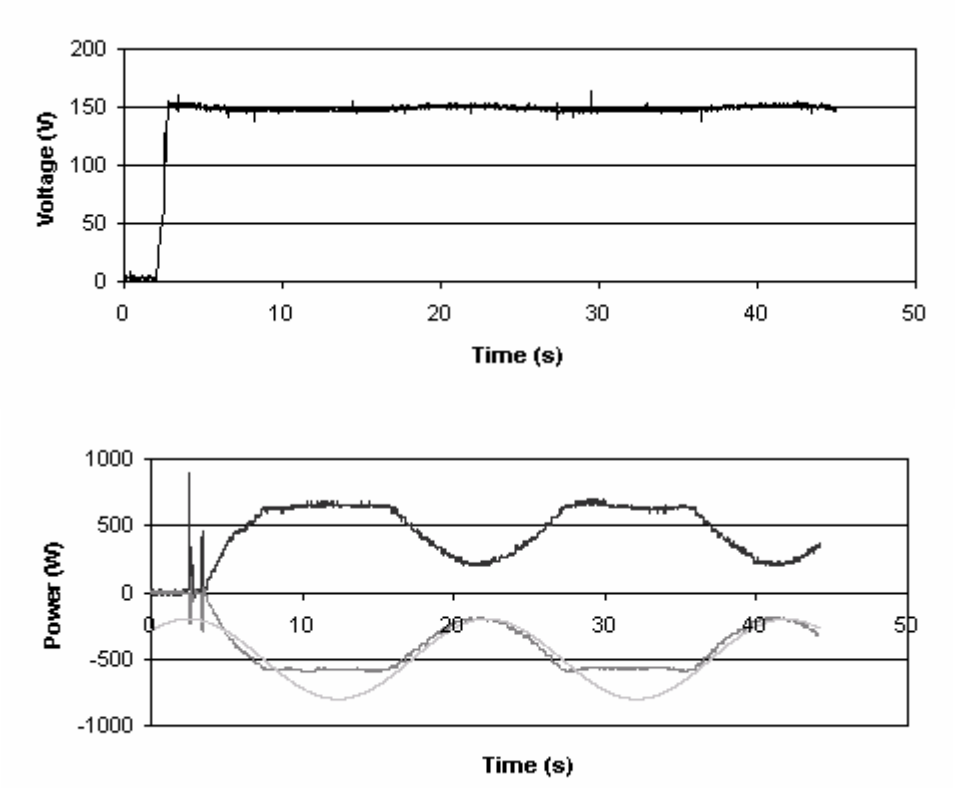


Figure 7.13. Measurements of scenario 2.

The first spike in the power curves in the beginning of Figure 7.13b is caused from the second contactor in the inrush current protection device. The spike occurs if the voltage over the capacitor has not yet reached 100 volts. So when the second contactor turns on it shortcuts the resistors, which will cause a higher current to go through the diodes. The second spike is created when I turn on the power switch for the converter that is acting as a load. The purpose for this switch is to enable the decoupling of the converter from the AC-grid. The switch can't be on from the beginning since that would set the voltage on the DC distributed power system to 100 volts, and I want to do that by using the inrush current device.

The result of this experiment is that it does fulfil the expectations. The first generator starts in droop mode and sets the voltage of the DC power system to 150V. When the voltage of the system has reached 90% of the nominal voltage, the second converter, set to be acting as a load, starts. This converter is started in power control mode and will try to withdraw power from the system according to the grey curve in Figure 7.13b. When the load converter is asking for more power, then the setpoint for the load unit is set to what the generator can produce at its maximum. But when the load is asking for less power then what the generator can produce at its maximum, the power set point for the load will be set according to how much power the load need.

This test also prove that the TCP/IP communication used by matlab works. Each converter is using a computer of its own, so in order to tell the production management program a TCP/IP communication for the remote computers is used. The time it takes between the control signals to the remote computer is updated every 0.2 seconds in general. The times are measured during the day, when other people are using the communication network as well, which is causing some delays. The time is however acceptable.

7.4.3 Test of scenario 3.

The third scenario that is described in chapter 3 contains 3 converters. One of them is acting as a load and the other two are supposed to act as generators.
The power that the load wants to consume will change with respect to time as in the previous experiment, thus will the pmax for the load be sine-shaped with an offset of –500W, an amplitude of 300W and the frequency 0.05 Hz.
The maximal power that the generators can produce are constant and with the size of 600W each. The specification is chosen as in the previous experiment, so it is possible to see the difference when another generator is added to the system.

The result of this experiment is shown in Figure 7.14. The curve in Figure 7.14a shows the voltage of the system. The black curve in Figure 7.14b shows the power generated from the generator that is controlling the voltage level of the DC power system. The grey curve is the power produced by the other generator and the dark grey curve is the power consumed by the load. The light grey curve shows the maximal power that the load wants to consume.

The system is started with the load already connected to the production management program, but it does not get permission to start until the voltage level is close to its nominal value. When the first generator was added to the system it was set in droop control mode and the voltage of the system increased to 150V. When the voltage was set the load got permission to start to consume some power.
After 9 seconds the other generator was added to the system. But since the power consumed by the load was low when it was added, it didn’t get permission to start until after 12 seconds.

As explained in the description of this scenario 3 in chapter 5, the system is always trying to keep the power generated from the generator(s) in droop mode at half of its capacity. In this experiment it would mean that the black curve in Figure 7.14b would have a constant value of 300W + losses in the system. This is almost what is shown in Figure 7.14b. The little peak in the blue power curve at the time 12 seconds is there because it takes some time for the second generator to start. It therefore have to add some more power then expected to avoid a voltage drop. The reason for the dip of the black curve at the time 27 seconds is that the load don't demand any more power.

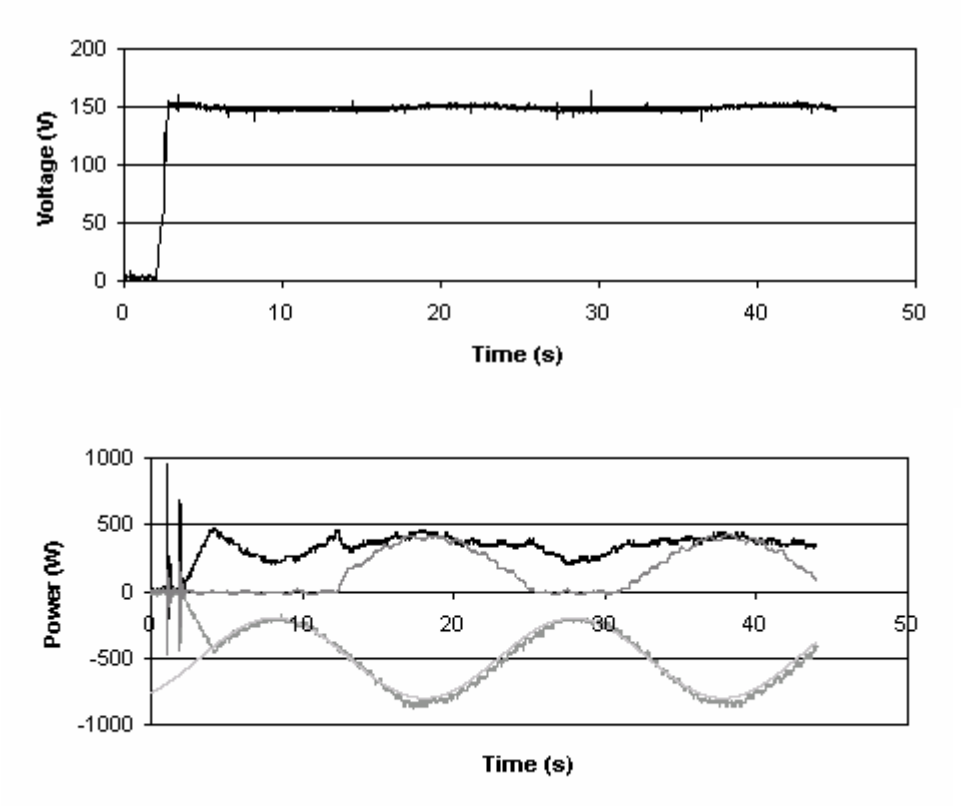


Figure 7.14. Measurements of scenario 3

7.2.1 Test of scenario 4

Scenario number 4 in chapter 5 is a scenario that is created to test the function of the storage unit. It is a 3-converter system with one generator, one load and one storage unit. The generator in this experiment has maximal output power capacity that varies with respect to time and could be seen as a wind power plant. The maximal power curve is sinusoidal with an offset of 1000W, amplitude of 350W and the frequency 0.05 Hz. The storage unit has a maximal output and input power capacity of 150W. The maximal amount of energy that the storage unit is able to store is 10000Ws, and when the energy stored is less then 200Ws it is considered as empty. The reason why zero is not chosen as the minimal amount of energy is because the system needs some time to detect that the storage is empty to set the set point to zero. The initial amount of energy that the storage

has when the experiment starts is 700Ws. This storage is chosen to be a small storage in order to be able to check that the storage really shuts down when it is empty or full.
The load in this experiment is chosen to have a constant maximal power capacity and it is chosen to be 500W.
The result of the experiment is shown in Figure 7.15. The Figure 7.15a , shows the voltage of the DC power system. The black curve in Figure 7.15b shows the power generated by the generator and the light grey curve shows the maximal power capacity that the generator can produce. The dark grey curve shows the power produced by the storage unit and the grey coloured curve represents the power consumed by the load.

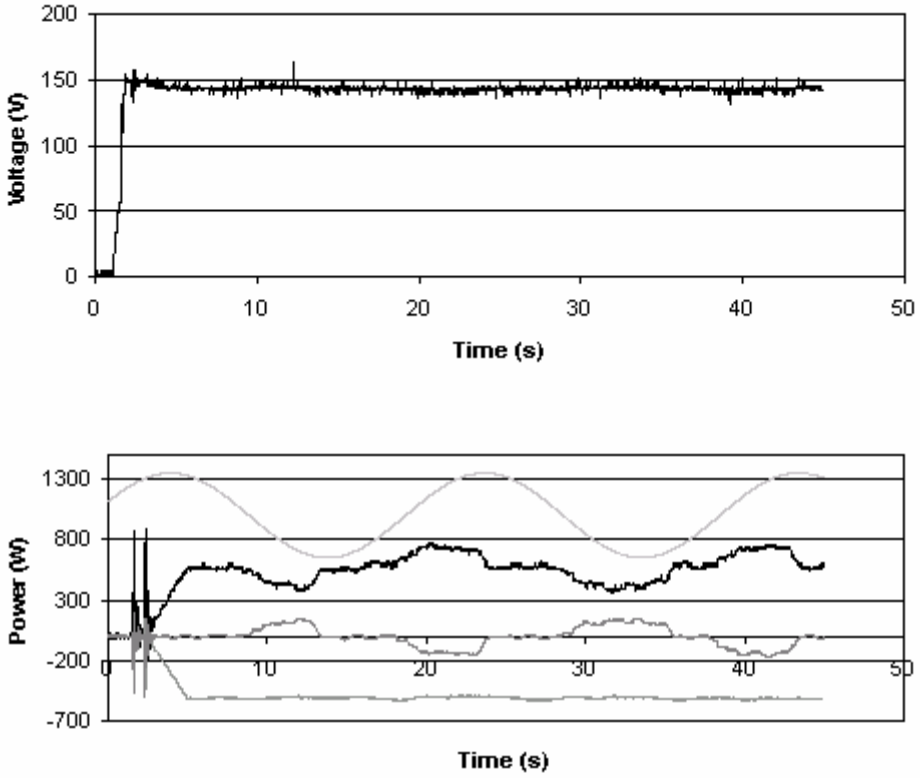


Figure 7.15. Measurements of scenario 4

In this test it is the generator that is first connected to the system, the second unit is the load. But it is not until the load is connected that the generator gets permission to set the voltage to the system. When the voltage in the system has reached 90% of the rated voltage, the load gets the permission to start to consume some power.
The storage unit was added to the system after about 8 seconds and then immediately starts to generate some power. The system is designed to always try to keep a safe distance between power produced by the droop generators and the maximal power that it can produce. That is why the storage is adding some power to the system even if the generator could manage to support the load by itself. But after 13 seconds the storage is empty and cannot deliver any more power. When 18 seconds have passed, the safety distance for the generator is high enough to let the storage store some energy, which could be used at a later occasion. The storage does however quite soon reach maximal

storing capacity. After 23.5 seconds the storage is full and cannot store any more energy so the power-set point for the storage is set to zero.

The result of this experiment went as expected, which shows that the storage module that is created in the laboratory does work.
The power-regulating controller for the storage does not however work as well as we would hope. The best way to see this is between the times between 30 and 34, when the set point is set to 150W but the power that is produced is quite choppy. The reason for this is probably that the dSpace interface card that belongs to the storage converter adds some disturbance to the control signals that goes to the converter. This card is a modified card that earlier has been used in a laboratory context. The modification did however not work properly. Lots of time has been spent to try to solve the problem but the error could not be found.
This error does not affect the main purpose of the experiment, which is to show that the algorithms are working and to show that the system is dynamic and adaptable.

7.2.5 Test of scenario 5.

The aim of this scenario is to make sure that the exchange module works the way it is supposed to. The scenario is described in chapter 5 and contains 3 converters. The setup is very close to the setup from experiment 4. The major difference is that the exchange module is not capable of storing energy. The maximal energy that the exchange unit can deliver or take from the DC-power system is for this test set to be 150W in both directions. Since this module needs two power systems to work properly and there only was one available in this experiment a simulated system had to be made. On the side of the converter that was not connected to a power system, some stationary signals that looked like a power system was created and the experiment was possible to perform.
The generator that sets the voltage of the DC-power system has a maximal power capacity that varies with respect to time. The variations follow a sinusoidal curve form with an offset of 1000W, amplitude of 350W and a frequency of 0.05 Hz.
The maximal power that the load wants to consume is constant and has a magnitude of 500W.

The result of the experiment is shown in Figure 7.16. In Figure 7.16a the voltage of the DC system is shown.
The black curve in the Figure 7.16b represents the power that is produced in by the generator and the light grey coloured curve shows the maximal power that the generator can produce.
The dark grey curve shows the power that is generated by the exchange unit and the grey curve shows the power consumed by the load.

In this experiment the generator was the first unit that was added in the information system. The second unit that was added to the system was the load. Just as in the previous experiments the generator is not allowed to start until another unit, that isn't a generator, is added to the information system. So the generator sets the voltage in the system when

the load was added. The load was started when the voltage reached 90% of the nominated voltage level. Since the generator could support the load with all the power demanded, the load was set to consume power at its maximal capacity.
The exchange unit was added to the information system after about 8 seconds and immediately starts to generate energy in/for the system. Just as described in the last experiment this is to, if possible, maintain a safety margin between the power produced by the unit in droop mode and the maximal power that the unit can produce.
This safety margin is set to be half of what the generator can produce at its maximum. In this way the unit that is set to droop will have a good safety margin both up and down. The exchange unit can however not produce more energy then 150W, which is what the unit does after a short while.
After a while the safety margin for the generator is big enough to not need any support from the exchange unit and the power set point for the exchange unit drops. A little while later the maximal power that the generator can produce is high enough to export some power. Since the exchange unit always is able to export in this setup, this is also what is happening. But the power that is exported is never higher then 150W.

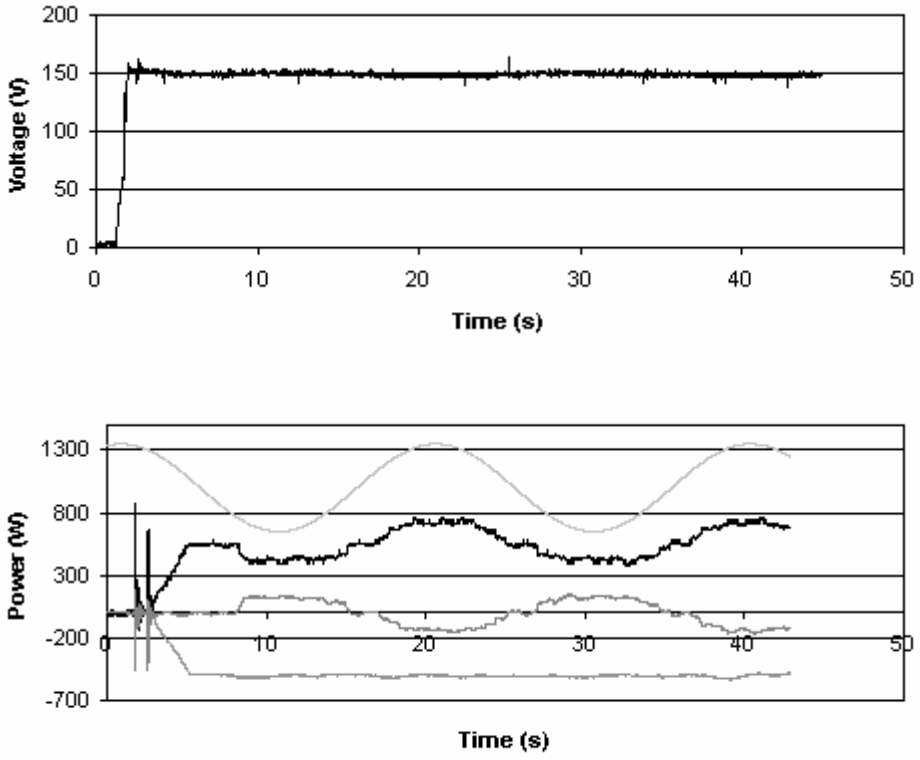


Figure 7.16: Measurements of scenario 5.

This experiment really shows that the concept of plug and produce really works in reality. The time between 5 and 8 seconds when only the generator and load was connected to the information system the generator supported the load by itself. It was not until the

exchange unit was introduced after 8 seconds that the generator could get some support to feed the load with power. This is something that is possible to see in Figure 7.16b.

7.2.6 Further tests

Further tests have been made to verify hat the concept of plug and produce is working with this setup.
The base of tests has been made out of 2 units, one acting as a load and one as a generator. That has been working according to scenario 2. A third unit has been added and withdrawn from the system and the result of the test has been investigated.

The first set of tests that were made for these purposes was a test without letting any converter start at all. The purpose for this test was only to check the signals that were sent between the computers. The reason the converters were not allowed to start in this test was that they didn't have to be on for the test and the programs were stopped at several places, which would lead to a system that would be out of control.
All of the three converters were added and taken away from the production management program in different orders. The only limitation in this system was the converter station running both the production exchange program and the production management program. When this converter is added to the system it is not possible to take it out from the system. That is because it is using a local file on the computer to handle the communication with the production management. This problem is of course possible to solve by separating the two modules to different computer stations, but was not implemented in this work.

The second category of test was made with the converters on. The same tests that were checked with the converters off were conducted, but this time to verify that the converters really act in a way consistent with previous tests.
The results of the tests show that the system can handle both the addition of another unit and the withdrawal of units from the system, and can adapt to the new condition that occurs.

8 Conclusions

In this report a small DCdistributed power system have been tested both in a simulation environment and in a laboratory environment where power converters have been acting as power units. The tests that have been done have verified that that the system is dynamic and adaptable, which include managing several different system configurations and that new power units can be connected or disconnected to the system at any time online.
The simulation results and the results that are being made in the laboratory are very similar. That means that results from coming simulations can be used as reliable results and that they are very likely the same in the real system.
The system is dynamically and can cope with disturbances that may appear at any time.
The pace of when the command signals are updated in the real system can be increased a lot by using a faster communication network or have a network which only purpose to be used for the power system. The communication in the real power system can also be increased by having the production management process on a separate computer.
The concept of plug and produce is working as it should. That is e.g. shown in the real time test 5.

References

[1] Mohan N, Underland T M, Robbins W P
Power Electronics “Coverers, Application and design”
USA 2003

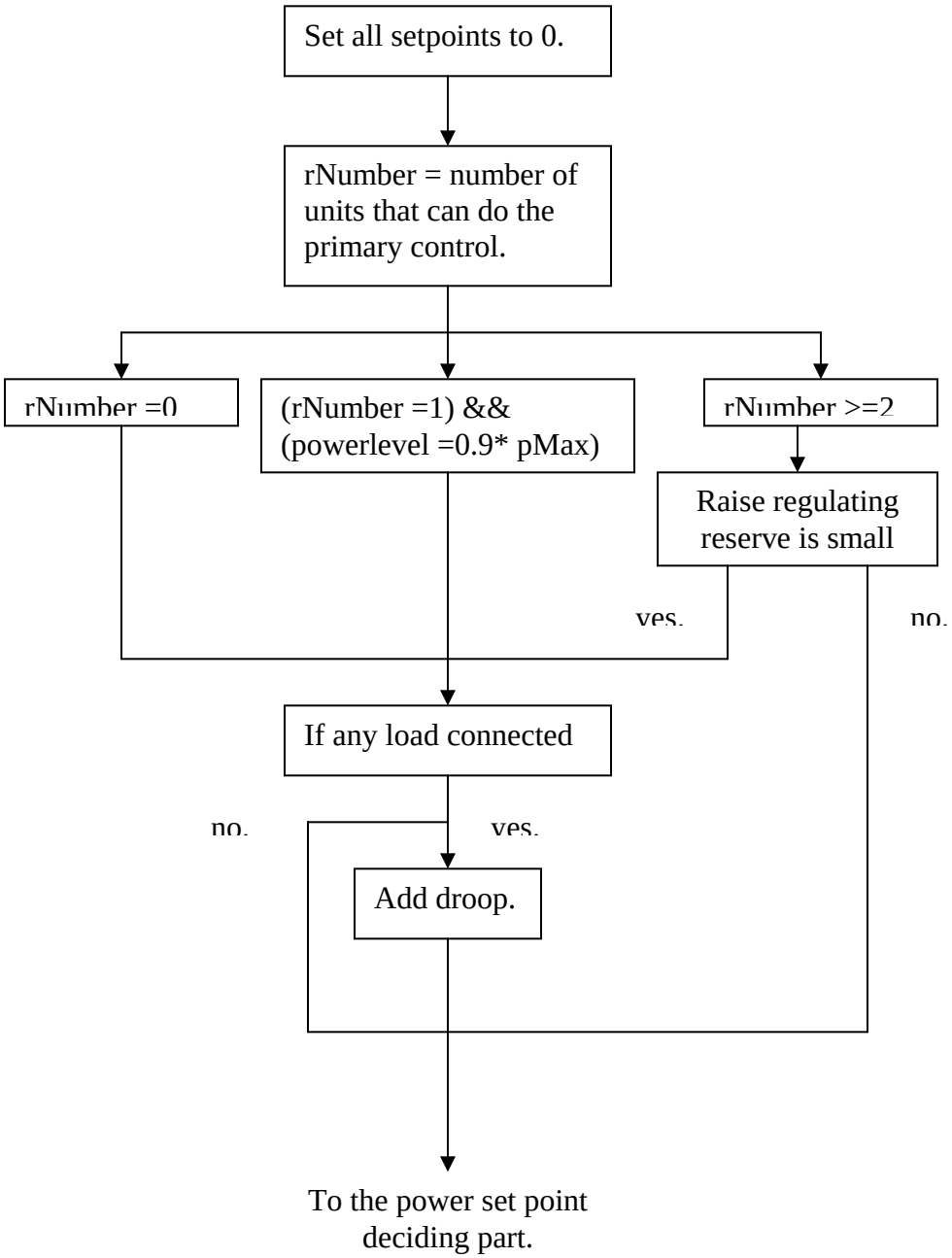
[2] Svensson J, Johnsson A, Karlsson P
Wind Power Plant Market and Operation Interaction “Principles for Information and Energy Management Systems”
Sweden 2002

[3] Svensson J, Johnsson A, Karlsson P
Information Structures for Scalable Distributed Power Systems
Sweden 2002

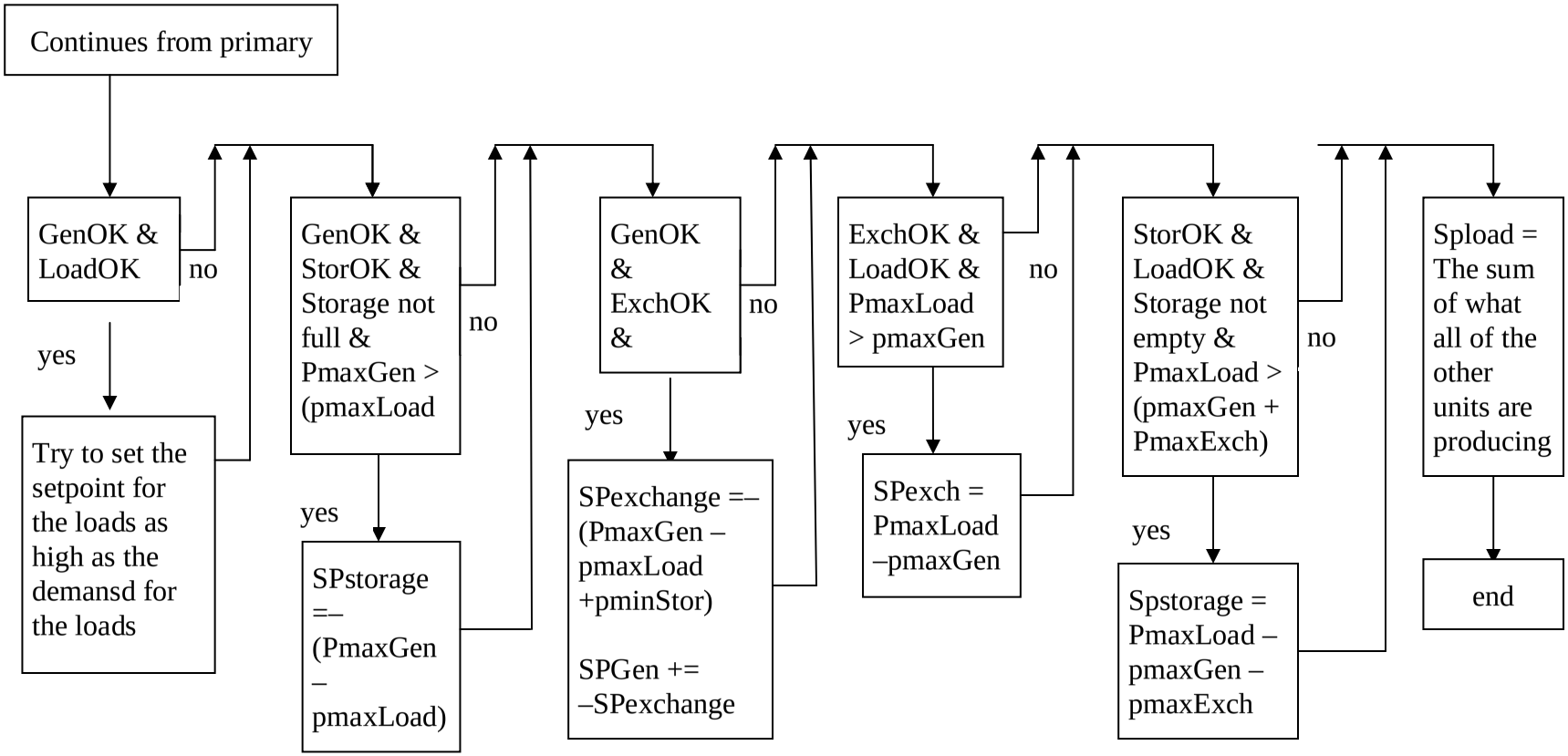
[4] Svensson J, Karlsson P
Adaptive signal management “A Modelica and C++ interaction example, the SignalFlow Library”
Sweden 2002

Appendix A

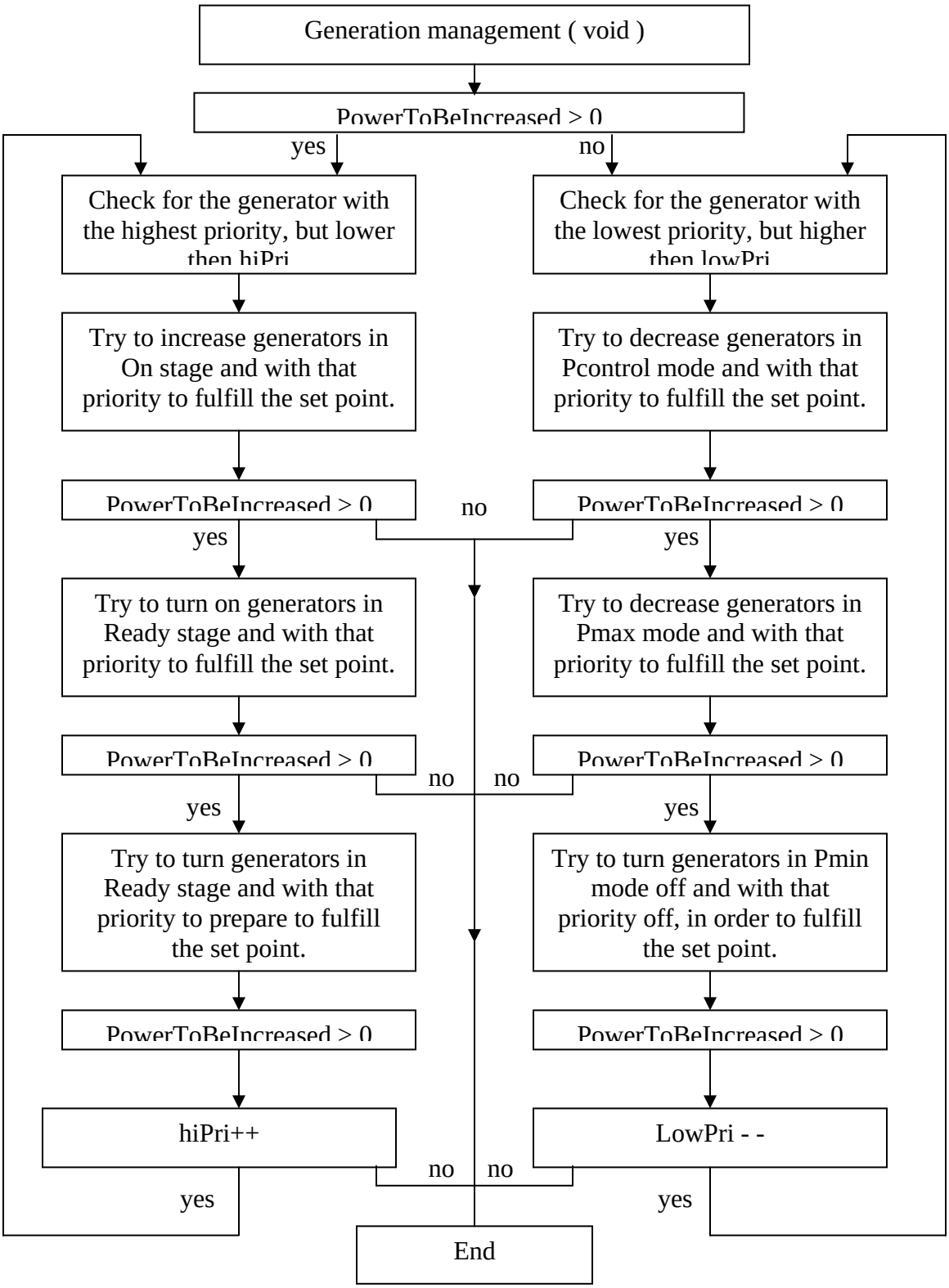
Primary control sequence



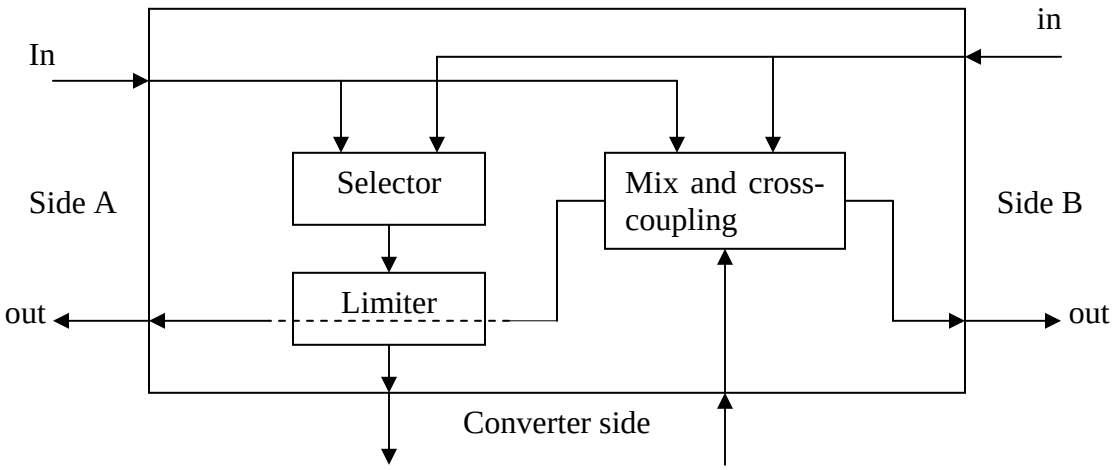
Secondary control sequence



Generation Management

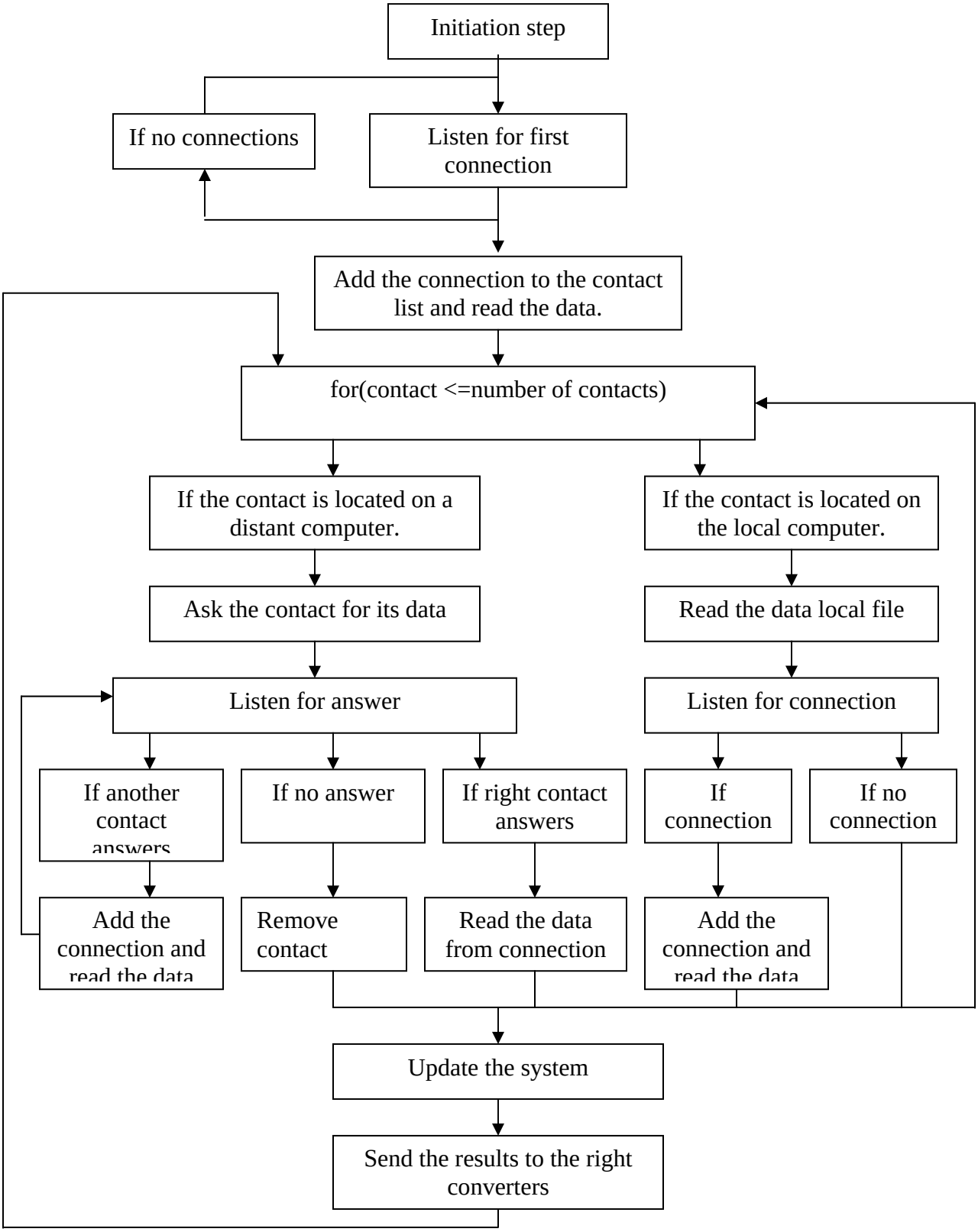


Production exchange

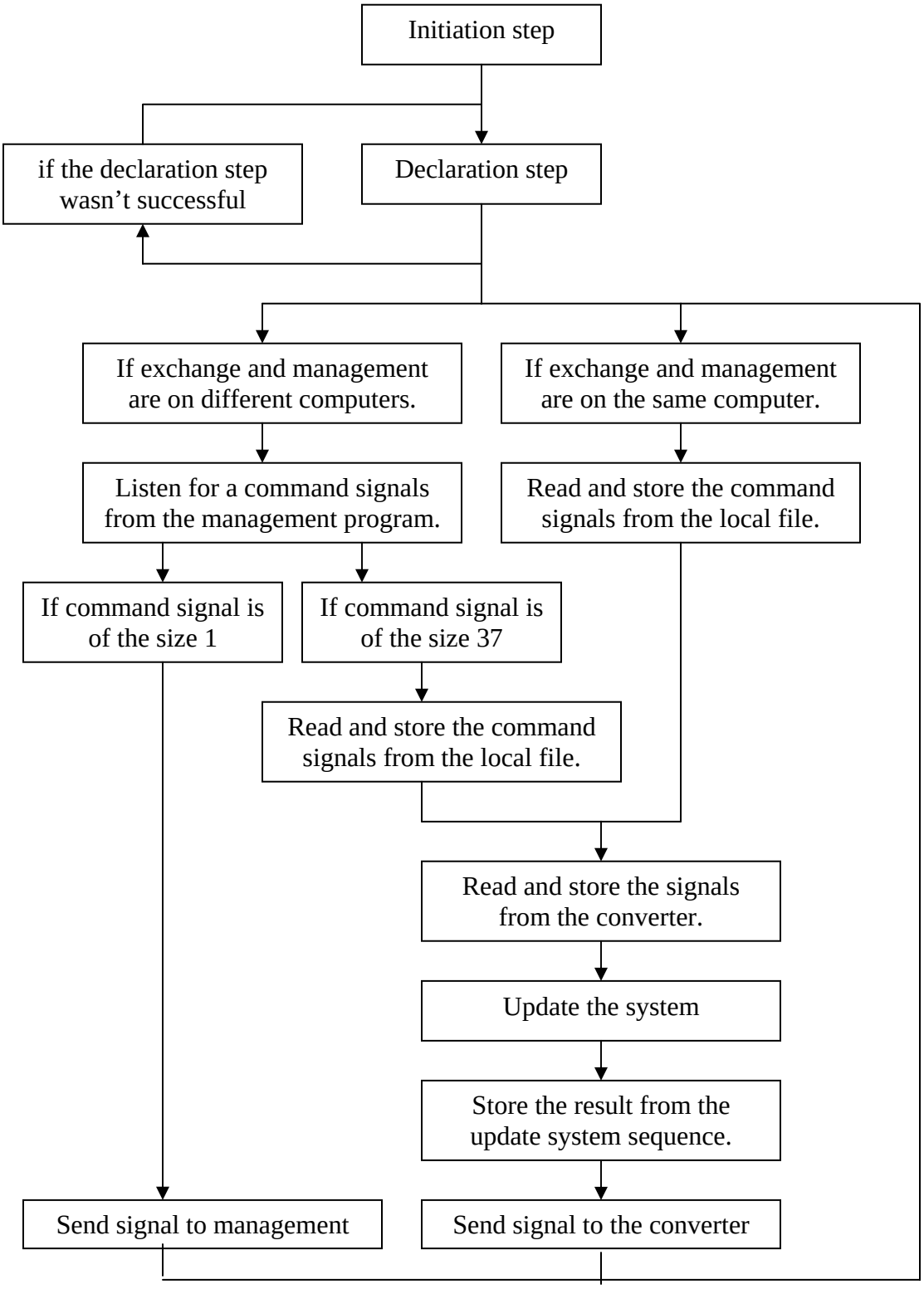


Appendix B

Flow scheme of management interface program



Flow scheme of exchange interface program



Appendix C

The Simulink model