



Norwegian University of
Science and Technology

Optimal Operation of a Stand-Alone Power Supply using Artificial Intelligence

Øyvind August Rui

Master of Science in Energy and Environment

Submission date: June 2009

Supervisor: Tore Marvin Undeland, ELKRAFT

Norwegian University of Science and Technology
Department of Electrical Power Engineering

Problem Description

Due to global warming the world must shift to a more environmental friendly energy supply. At the same time billions of people do not yet have electrical power supply. Stand-alone systems will be an important contribution in the future.

To get an optimal operation of a stand-alone hybrid energy supply with renewable energy several things must be predicted, like the production of the renewable energy sources and the power load of the consumers. Artificial intelligent methods like artificial neural networks and fuzzy logic have proven to be suitable for such short term production. Secondly the power electronics converters must be controlled to obtain the wanted energy flow. Here artificial intelligent methods can also be used.

The project consists of presenting the suitable artificial intelligent methods. Calculations and/or simulations should be performed to test the performance. The results should be compared with more classical methods like PI control.

Design of the stand-alone plant is not included in the project. Laboratory work should be performed.

Assignment given: 02. February 2009

Supervisor: Tore Marvin Undeland, ELKRAFT

Sammendrag

Kunstig intelligens (KI) er en samlebetegnelse for mange ulike regneteknikker. De har til felles at de bruker ulineære algoritmer og at de er inspirert av prosesser i naturen, særlig hvordan mennesker tar avgjørelser. Bruken av KI for optimal drift av en stand-alone kraftforsyning har blitt undersøkt. Dette inkluderer prediksjon, estimering, optimalisering og regulering. En presentasjon av relevante KI teknikker har blitt gjort. KI teknikkene har blitt sammenlignet med klassiske tilnærminger som for eksempel PI-regulering. De nye teknikkene som ble undersøkt viste seg å være veldig kraftfulle og skulle blitt brukt mer enn de brukes i dag. KI er spesielt lovende for bruk innen overvåkning av systemer. Men det kan også brukes til å regulere omformere direkte. En regulator for en DC/DC boost omformer ble dimensjonert. Den viste seg å være betydelig bedre enn en klassisk PI-regulator. Om regnetiden er kortere eller lengre enn klassiske tilnærminger kommer an på anvendelsen. Sammenlignet med PI-regulatorer har KI-regulatorene lang regnetid. Sammenlignet med klassiske vindkraftpredikteringsteknikker er derimot KI teknikker veldig raske. En ulempe med KI er mangelen på regler for å avgjøre den indre strukturen i algoritmene.

Summary

Artificial intelligence (AI) is a collective term for several computing techniques. They have in common that they use non-linear algorithms and that they are inspired from different processes in the nature, in particular how human beings make decisions. The use of AI for optimal operation of a stand-alone power plant has been investigated. This includes prediction, estimation, optimization and control. A presentation of some relevant AI techniques are given. A comparison with classical approaches such as for example PI control was made. The new techniques that were investigated proved to be very powerful and should be used more frequently than it is used today. AI techniques are especially promising for supervisory control, but can also be used to control converters directly. A controller for a DC/DC boost converter was developed. It proved to be significantly better than a classical PI controller. Whether the computing time is shorter or faster than for classical approaches depends on the application. Compared to PI controllers the AI algorithms have a long computing time. Compared to classical wind power prediction techniques on the other hand AI techniques are very fast. A disadvantage with AI is the lack of rules for deciding the inner structure of the algorithms.

Acknowledgements

I want to thank everyone that have helped me with this project.

In particular I want to thank visiting Ph.D student Amin Hajizadeh for helping me with fuzzy controllers. His advices have been very useful. I also got a lot of help from Servicelabben during my practical work. In particular I want to thank Thor Lohse.

Thanks also to all fellow students and employs at ENO for the good social environment and useful technical discussions.

Last but not least I want to thank my supervisor Prof. Tore Undeland for letting me write a master theses on this subject.

Contents

1	Introduction	1
2	AI theory	2
2.1	Artificial neural networks	2
2.1.1	The perceptron	2
2.1.2	Logic operators AND, OR and exclusive-OR	3
2.1.3	Multilayer perception networks	4
2.1.4	Training of MLPs	5
2.1.5	Other networks	6
2.2	Fuzzy logic sets	6
2.2.1	Membership functions	6
2.2.2	Rules	8
2.2.3	Mamdani and Sugeno	9
2.2.4	Aggregation and defuzzification	9
2.2.5	Adaptive fuzzy logic	10
2.2.6	Graphical presentation	11
2.3	ANFIS	11
2.4	Genetic algorithms	12
3	Energy management	13
3.1	Power flow	13
3.2	Hybrid energy storage	16
3.3	Prediction	18
3.3.1	Wind power prediction	18
3.3.2	Load prediction	20
4	Control	22

4.1	Inverter	23
4.1.1	Voltage source inverter	23
4.1.2	Phase locked loop	23
4.1.3	Vector control	24
4.2	DC/DC converter	25
4.2.1	Transfer function model	25
4.2.2	Fuzzy control of DC/DC converters	27
4.2.3	Maximum power point tracking	28
4.2.4	Comparison of different controllers	32
4.3	Commercial products	33
5	Simulations on Boost converter	34
5.1	Converter setup	34
5.1.1	Parameters	34
5.1.2	Transfer function model	34
5.2	Controller design	35
5.2.1	PI controller	36
5.2.2	PI controller tuned with fuzzy controller	37
5.2.3	Fuzzy controller with integrator	39
5.3	Description of simulation case	41
5.4	Results and discussion	41
5.4.1	Voltage response	41
5.4.2	Parametric variations	42
5.4.3	Discussion	43
6	Practical work	44
6.1	dSPACE	45
6.2	Power circuit	45
6.2.1	Power supply	45
6.2.2	Converter	45
6.2.3	Load	47
6.3	Controller design	47
6.3.1	PI controller	47
6.3.2	PI with fuzzy tuning	47
6.4	Description of experiments	48
6.5	Results	48
6.5.1	PI controller	48
6.5.2	PI with fuzzy tuning	50

6.6	Discussion	53
6.7	Implementation of AI in a DSP	53
7	Conclusions and further work	54
7.1	Conclusions	54
7.2	Proposed further work	55
A	dSPACE	59
A.1	Getting started	59
A.2	Input and output blocks	60
A.3	Logging	60
B	Details of simulations	62

List of Figures

2.1	Perception	2
2.2	MLP	4
2.3	Membership function service	7
2.4	Membership function food	7
2.5	Membership function output tip	8
2.6	Command gensurf in Matlab	10
2.7	Simulation results for fuzzy block	11
3.1	Scheme stand-alone plant in [5]	14
3.2	ANN model from [5]	15
3.3	Battery model used in [18]	17
3.4	Fuel cell controller in [18]	17
3.5	Results of different prediction methods in [2]	19
4.1	Voltage source inverter, source [25]	23
4.2	Phase locked loop source [14]	24
4.3	Vector control schematic, source [25]	24
4.4	Schematic of boost converter	26
4.5	Fuzzy controller with output gain tuning, source [23]	27
4.6	Fuzzy controller with integrator in parallel, source [22]	28
4.7	Schematic diagram for MPPT, source [10]	29
4.8	Step-down converter, source [10]	30
4.9	Flow chart, source [10]	31
5.1	Circuit diagram boost converter	35
5.2	Bode plot of transfer function for boost converter	36
5.3	Surface plot PI_{tuned}	37
5.4	Optimization PI_{tuned}	38
5.5	Schematics $FLC_{para-int}$	39

5.6	Surface plot $FLC_{para-int}$	40
5.7	Optimization $FLC_{para-int}$	41
5.8	U_{in} , PI , PI_{tuned} and $FLC_{para-int}$	42
6.1	Experimental setup	44
6.2	Circuit diagram boost converter	46
6.3	Converter	46
6.4	Surface plot of fuzzy controller with 9 rules	47
6.5	Response of PI controller	49
6.6	Response of fuzzy tuned controller, $k_3 = 2, k_4 = 8$	51
6.7	Response of fuzzy tuned controller, $k_3 = 8, k_4 = 2$	52
B.1	Details of PI	63
B.2	Details of PI_{tuned}	64
B.3	Details of $FLC_{para-int}$	65

List of Tables

2.1	Training set for the logic function AND	3
2.2	Training set for the logic function OR	3
2.3	Training set for the logic function exclusive-OR	4
3.1	Power forecast classification	18
5.1	Parameters boost converter	35
5.2	Abbreviations of controllers	35
5.3	Parameters for PI	37
5.4	Parameters for PI_{tuned}	39
5.5	Parameters for $FLC_{para-int}$	40
B.1	Overview of the figures	62

Abbreviations

AI	: Artificial intelligence
ANN	: Artificial neural network
MLP	: Multi layer perception
ANFIS	: Adaptive neuro-fuzzy inference system
pu	: per unit
FL	: Fuzzy logic
GA	: Genetic algorithms
NWP	: Numeric weather prediction
MPPT	: Maximum power point tracking
SOC	: State of charge

Chapter 1

Introduction

Artificial intelligence (AI) is a wide term including a lot of computing techniques. These techniques have in common that they are based on how living creatures behave, in particular how a human being is thinking. Decisions are made based on reasoning and the algorithms can sometimes learn from previous measurements. The theory behind the relevant AI techniques is presented in Chapter 2.

Several aspects must be taken into account when operating a stand-alone power plant. The diesel generator should be operated in a way that minimizes the fuel consumption. At the same time the generator set and the converters should be controlled to maintain the stability of the grid at all time. This report includes a literature study of the use of artificial intelligence for energy management and converter control, presented in Chapter 3 and 4 respectively. Converter control is also further investigated by simulations and practical measurements, presented in Chapter 5 and 6 respectively. Comparisons with classical techniques are made throughout the report. The work also includes considerations on the implementation of AI in real time.

Chapter 2

AI theory

In this chapter mainly two artificial intelligence techniques are presented: Artificial neural networks and fuzzy logic sets. A brief introduction to genetic algorithms is also given.

2.1 Artificial neural networks

2.1.1 The perceptron

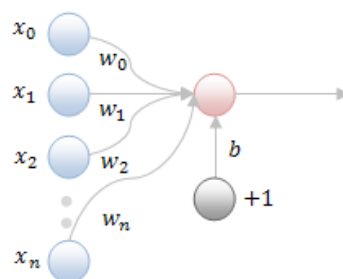


Fig. 2.1: Perception

The building brick in artificial neural networks (ANN) is the perceptron. A graphic presentation of a perceptron is given in Figure 2.1. A perceptron executes the mathematical function in Equation 2.1, where x is the input vector,

w the weighting vector and b the threshold. $\tanh(x)$ is defined in Equation 2.2. The $\tanh(x)$ function ensures that the output is always between -1 and 1.

$$y = \tanh\left[\sum_{i=1}^n x_i w_i - b\right] \quad (2.1)$$

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (2.2)$$

2.1.2 Logic operators AND, OR and exclusive-OR

In order to understand how neural networks work some simple functions can be implemented. Only a single perceptron is needed to implement the logic functions AND and OR. The perceptron is trained with measurements. First the perceptron can be trained to perform the function AND. The vectors in Table 2.1 can for example be inserted into the Matlab Neural Network Toolbox. The initially random weights will then be updated to suit the relations between the inputs and the outputs. This is what is called training.

Variable	Values
x_1	0 0 1 1
x_2	0 1 0 1
y	0 0 0 1

Table 2.1: Training set for the logic function AND

Variable	Values
x_1	0 0 1 1
x_2	0 1 0 1
y	0 1 1 1

Table 2.2: Training set for the logic function OR

The exact same neuron can later be trained with the vectors in Table 2.2 to perform the function OR. This shows one of the main ideas behind ANN. They model relations without changing the equations in the model. Only the

weights are changed. But the network has to be more advanced if the relations are more advanced. On single perceptron is not enough to implement the logic function exclusive-OR, with the training vectors given in Table 2.3. Then a network of perceptions must be used.

Variable	Values
x_1	0 0 1 1
x_2	0 1 0 1
y	0 1 1 0

Table 2.3: Training set for the logic function exclusive-OR

2.1.3 Multilayer perception networks

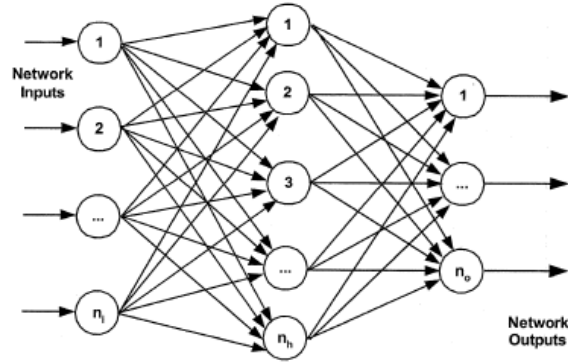


Fig. 2.2: MLP

Multilayer perception networks (MLP) are built up of perceptions. This is shown schematically in Figure 2.2. The Equations 2.3 and 2.4 represents a three layer MLP on matrix form. This is a MLP with r inputs, s neurons in the hidden layer and t outputs.

$$\begin{bmatrix} y_{1,1} \\ y_{1,2} \\ \vdots \\ y_{1,s} \end{bmatrix} = \tanh \left(\begin{bmatrix} w_{1,11} & w_{1,12} & \dots & w_{1,1r} \\ w_{1,21} & w_{1,22} & \dots & w_{1,2r} \\ \dots & \dots & \ddots & \vdots \\ w_{1,s1} & w_{1,s2} & \dots & w_{1,sr} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix} - \begin{bmatrix} b_{1,1} \\ b_{1,2} \\ \vdots \\ b_{1,r} \end{bmatrix} \right) \quad (2.3)$$

$$\begin{bmatrix} y_{2,1} \\ y_{2,2} \\ \vdots \\ y_{2,t} \end{bmatrix} = \tanh \left(\begin{bmatrix} w_{2,11} & w_{2,12} & \dots & w_{2,1s} \\ w_{2,21} & w_{2,22} & \dots & w_{2,2s} \\ \dots & \dots & \ddots & \vdots \\ w_{2,t1} & w_{2,t2} & \dots & w_{2,ts} \end{bmatrix} \begin{bmatrix} y_{1,1} \\ y_{1,2} \\ \vdots \\ y_{1,s} \end{bmatrix} - \begin{bmatrix} b_{2,1} \\ b_{2,2} \\ \vdots \\ b_{2,t} \end{bmatrix} \right) \quad (2.4)$$

There is no rule for how many neurons that is appropriate. This depends on the application. The number of hidden layers on the other hand is easier. For most practical applications one hidden layer is enough, two on the limit. To get the right number of neurons many techniques can be applied. One is testing different structures to see what structure that gives the best result. Another more methodical way to do it is to use genetic algorithms. In [6] a method for deciding the number of neurons in a network to perform time series forecasting is proposed, using genetic algorithms. A presentation of genetic algorithms is given in Section 2.4.

2.1.4 Training of MLPs

Training of ANNs is done like training of a single perception, only with a more generalized algorithm. The idea behind ANNs is to learn from previous measurements. An input vector is applied to the network and an output vector of the network is calculated. The calculated output vector is compared with measurements, and weight corrections are calculated on the base of the output error. The bigger and more complex the neural network is the more training sets are needed. As a rule of thumb one claims that to train a network with W weights to estimate within an error ϵ , W/ϵ training sets are needed [3]. The threshold b is also associated with a weight that is updated.

A network with many neurons take long time to train. With too many neurons in the network a phenomena called overtraining can also occur. This means that the network learns all the details in the training sets, but has little capability of generalizing.

2.1.5 Other networks

There are many different types of artificial neural networks, each suitable for different purposes [3][12]. In [13] a study of the use of the different neural network types within industrial applications was made. The distribution is 81.2 % for the MLP, 5.4 % for Hopfield networks, 8.3 % for Kohonen networks and 5.1 % for others. The MLP presented in this chapter is therefore without doubt the most used network.

2.2 Fuzzy logic sets

Fuzzy sets were first proposed in 1965 and is a modification of the traditional thinking in programming. Often variables are classified with hard limiters [4]. For example a man is said to be high if he is higher than 175 cm and not high if he is below. In fuzzy sets the fuzzy variables (for example the height of a man) and different properties (like high, low and normal) are related with membership functions. For example a man is said to be 0 % high if he is 175 cm or lower and 100 % high if he is 185 cm or taller. With a linear relationship in between the man is 50 % high if he is 180 cm. In this chapter the different parts of a fuzzy set is explained and an example from the Matlab Fuzzy Toolbox is presented.

2.2.1 Membership functions

In an example from the Matlab Fuzzy Logic Toolbox the tip to give in a restaurant was calculated on the base of the service and how good the food is. The Figures 2.3 - 2.5 show the membership functions for this example. Membership functions relate fuzzy variables (service, food quality and tip) and linguistic variables (good, poor, high, low, etc.). The food was said to be 100 % delicious if it got the grade 9 (from 0 to 10) and 0 % delicious if it got the grade 7. The service was described with three linguistic variables. The output of the example with the restaurant was the tip given in percent. The fuzzy variable tip had three linguistic variables: low, average and generous.

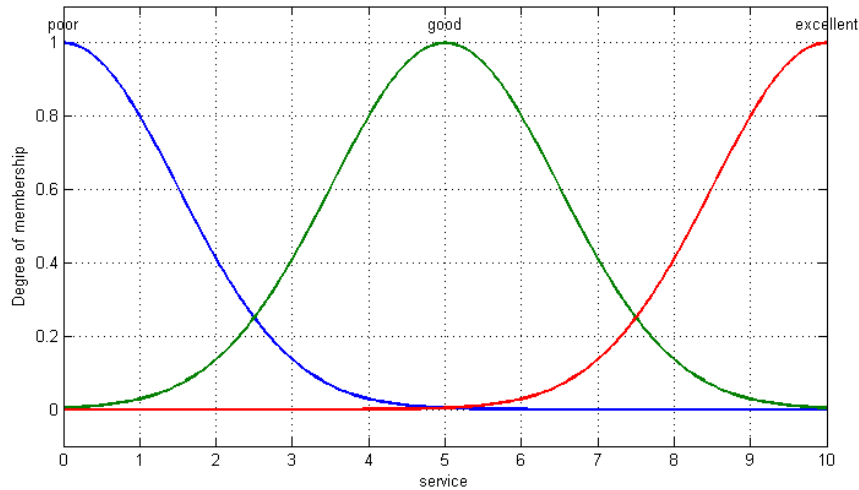


Fig. 2.3: Membership function service

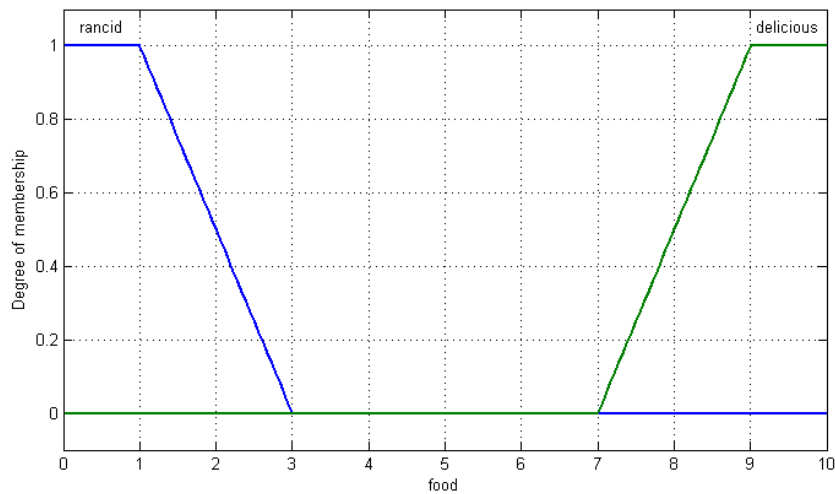


Fig. 2.4: Membership function food

Different curves can be chosen for the membership functions. In Figure 2.3 gaussian membership functions are chosen, while the membership functions in the Figures 2.4 and 2.5 are so called trapezoidal and triangular membership functions respectively.

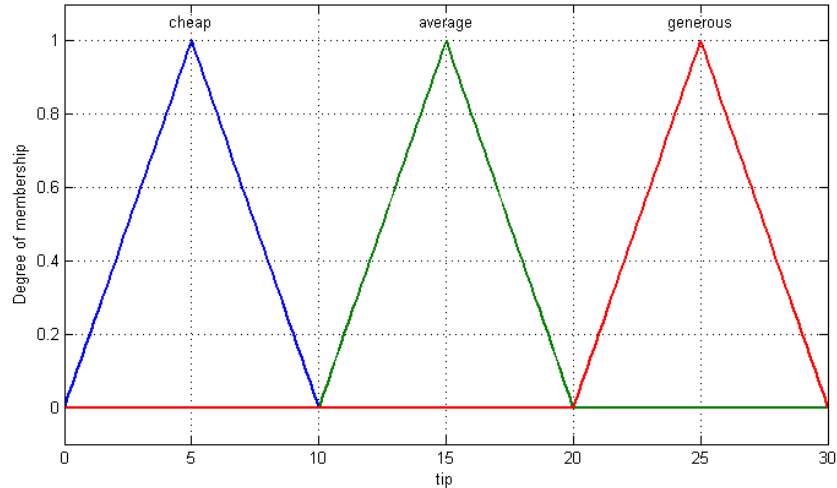


Fig. 2.5: Membership function output tip

AI techniques have no general design procedures and there is no rule for which type of membership functions to choose or how many linguistic variables that are appropriate for each fuzzy variable. There are examples of that gaussian membership functions give better fit to testing data, but are less robust and less capable of generalizing than triangular and trapezoidal functions [2][8]. Regarding the spacing of the membership functions, an overlap of 20 % to 50 % between the linguistic variables are often suggested as a good approach [3].

2.2.2 Rules

The inputs and the output in a fuzzy set is related with *if then* statements. The statements use the linguistic variables to compare the input and output variables instead of using numbers that is the traditional approach in computer programming [3]. There are two types of rules; *and* and *or*. The number and type of rules to use needs to be decided by the person implementing the fuzzy model. In controllers the rules usually cover all possible combinations of linguistic variables. In the tip example presented above three rules were used.

Rule 1: if service is poor or food is rancid then tip is cheap

Rule 2: if service is good then tip is average

Rule 3: if service is excellent or food is delicious then tip is generous

2.2.3 Mamdani and Sugeno

There are two main types of fuzzy controllers, Mamdani and Sugeno [3]. In Mamdani fuzzy controllers the properties are not different for the input and output variables. In Sugeno controllers the output is a function of the input variables. This function can of course also be a constant. The defuzzification technique used in the Mamdani method requires a powerful computing unit. It is the most suitable technique for capturing expert knowledge. Sugeno is more suited for optimization and control [3].

2.2.4 Aggregation and defuzzification

Based on the fuzzy input variables the membership of each of the output membership functions is decided. There are different ways of doing this. If the degree of membership of a trapezoidal membership function is 0.2 the top of the membership function can simply be cut at 0.2. An other approach is to scale the membership function so that the width of the top of the trapezoidal shape is equal to the one with 1 as maximum. The weighting of the different inputs and rules is called aggregation. The modified output membership functions are put together to a new shape. The center of gravity (COG) of this figure (Equations 2.5 and 2.6) gives the output value. This is called defuzzification.

$$COG = \frac{\int_a^b \mu_A(x)x dx}{\int_a^b \mu_A(x) dx} \quad (2.5)$$

$$COG = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{k=a}^b \mu_A(x)} \quad (2.6)$$

2.2.5 Adaptive fuzzy logic

For time variant processes and non linear processes with a wide range of operating points it is desirable to change the parameter of the controller. For classical controllers look up tables are frequently used. Fuzzy controllers can be tuned on line when controlling a process. The rule base is usually kept constant, but the membership functions can be adjusted. In [15] a method for drawing trapezoidal membership functions was proposed. Although there are examples of fuzzy control tuning by changing the shapes of the membership functions adaption by tuning input and output gain is more frequently used. Such controller setups are presented in Chapter 4.

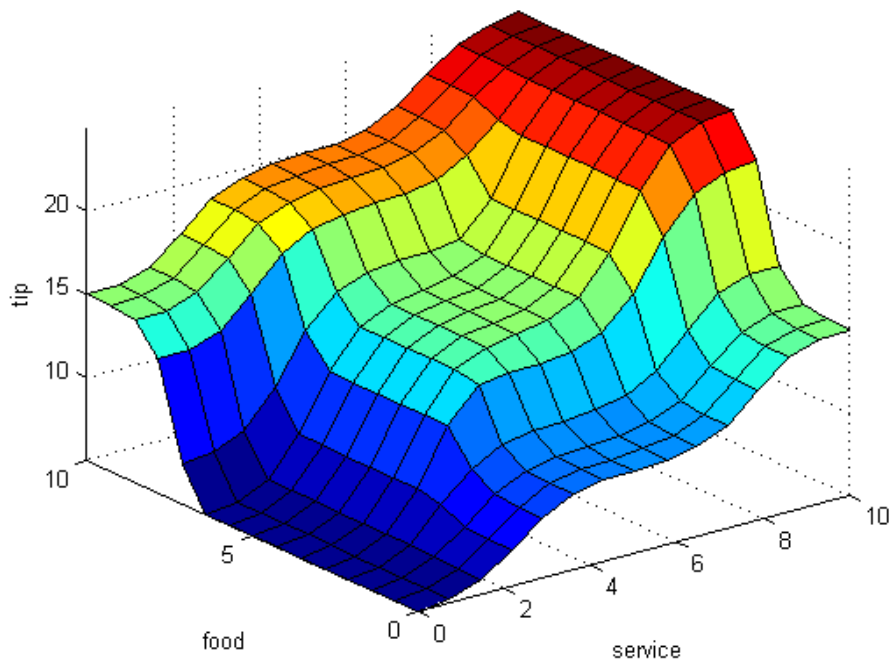


Fig. 2.6: Command gensurf in Matlab

2.2.6 Graphical presentation

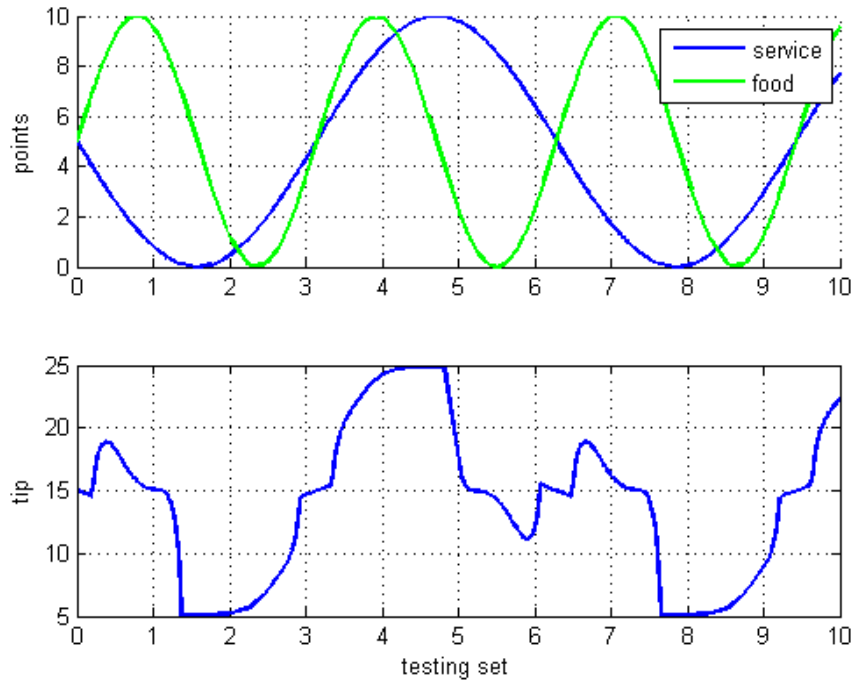


Fig. 2.7: Simulation results for fuzzy block

In the tipper example how much tip to give was a result of two variables, namely the service and the food quality. In Figure 2.6 the tip is presented as a function of the two input variables. The system can also be simulated in Matlab Simulink. In Figure 2.7 the tip example has been simulated with two sinus signal inputs. The two signal representing the service and the food quality have the same amplitude but a different frequency and phase angle. The result is shown in the lower part of the figure.

2.3 ANFIS

An adaptive neuro-fuzzy inference system (ANFIS) is a hybrid between neural networks and fuzzy sets. Neural networks must often be regarded as black

boxes. The outputs from a neural network are based on the training sets and there are no logic rules inside the model. In a fuzzy system the relationships between the input and the output fuzzy variables are given as *if then* statements. This is more transparent and easier to understand. On the other hand there are few rules for defining the different membership functions and rules in a fuzzy system. An ANFIS has layers with neural networks and at least one rule layer. In this way it combines the best of two worlds, namely the transparency of the fuzzy sets and the ability neural networks have to learn [3].

2.4 Genetic algorithms

Genetic algorithm are used for optimization later in this report and the theory behind will therefore be briefly presented. Like every other optimization algorithm genetic algorithms work by changing some parameters in a system to obtain an optimal solution. These parameters can for instance be different gains in a controller. Optimization algorithms also have an objective function that shall be minimized or maximized. This function can be the absolute value of the output error of a controlled system. In genetic algorithms each set of variable parameters are regarded as a DNA, and can be called an individual. Initially a population of certain (often 10-30) individuals are randomly generated. The population undergoes processes like in the nature. The crossover probability is quite high. In the crossover process two new DNA strings are made by cutting and pasting together two old DNA strings. Other processes from nature such as migration and mutation are also taken into account. Migration means that an individual leaves the population and that a new one comes in, whereas mutation means that a random change in the DNA occurs. The migration and mutation probabilities are low, but insures that the optimization does not stop in a local minimum of maximum.

The different populations that are generated are called generations. If the optimization converges the average error in every generation gets smaller and smaller. Genetic algorithms is regarded as an artificial intelligence technique and is a very powerful way of optimizing.

Chapter 3

Energy management

Artificial intelligence techniques have a variety of applications within the field of renewable energy production. Possible applications are production and load forecasting, design, modeling, energy management and control [1]. Design and modeling will not be discussed in this project. However, the other above mentioned topics will be investigated. The basis of a good energy management is to have good prediction of the production and the load. Secondly the energy management algorithm must be a good one. This chapter presents examples of doing this with artificial intelligent techniques. The chapter is based on literature studies only. It is desirable to compare the AI techniques with conventional methods. Such studies, however, have not been found for all the applications.

3.1 Power flow

As mentioned above one part of operating a stand-alone power plant is to decide the power flow, in other words when to start the diesel engine, when to charge the battery etc. In [5] an algorithm for deciding when to start or stop the diesel engine was proposed using artificial neural networks. In addition to on/off the algorithm also gave the diesel power production reference. The stand-alone plant is shown in Figure 3.1. The AC load consisted of some houses and a water destilation plant cleaning water through osmosis. Access of fresh water is a big problem many places in the world. Often it is

transported a long way, causing high CO_2 emissions from the trucks transporting the water. Therefore it is better to have a distributed production of fresh water. From an electric point of view the osmosis unit is an induction machine (a pump).

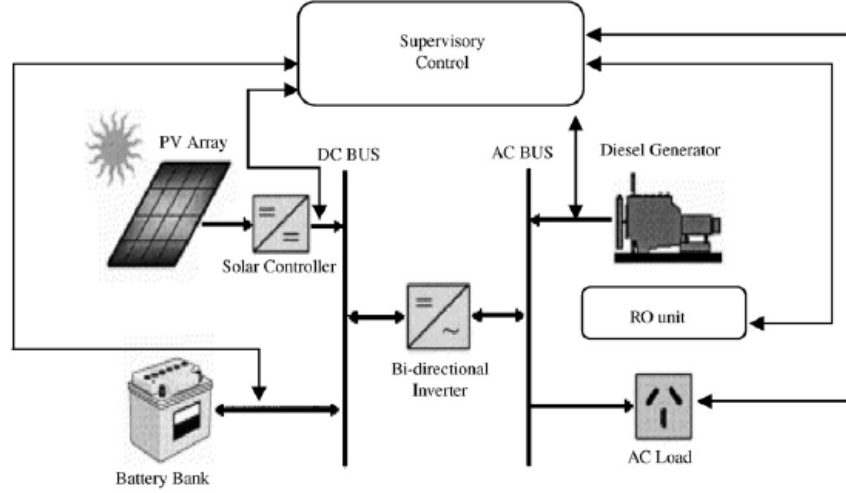


Fig. 3.1: Scheme stand-alone plant in [5]

All the components in Figure 3.1 were modeled and implemented in a simulation program. The objective of the neural network model was to decide the power flow. Therefore the energy was the main focus of the modeling. The model of the solar panel was given by Equation 3.1, where G_N is the normalized irradiance $G_N = G/(1000W/m^2)$, I_{lg} is the light generated current, I_{SCR} is the short circuit current, I_t is the short circuit current temperature current, T_c is the cell temperature and T_r is the cell temperature reference.

$$I_{lg} = I_{SCR}G_N + I_t(T_c - T_r) \quad (3.1)$$

The fuel consumption FC of the diesel engine was given by Equation 3.2 where P_{DG} is the diesel operating power, P_{GD} is the rated power of the diesel generator, α is the fuel consumption efficiency and FC_{rated} is the fuel consumption at rated power. An interesting point is that the diesel engine has a high consumption at no load. It is therefore desirable to run the diesel generator at either full load or to turn it off completely.

Several models of the battery were given modeling different phenomena. These models will not be presented here. The bidirectional inverter was modeled with Equation 3.3. $P_{INV,LOSS}$ was modeled as IR^2 , where R is the AC resistance.

$$FC = \alpha(P_{DG} - P_{DG,rated}) + FC_{rated} \quad (3.2)$$

$$P_{INV,DC} = P_{INV,AC} + P_{INV,LOSS} \quad (3.3)$$

The reverse osmosis unit was operated in steady state and the energy consumption is linear to the water that was produced.

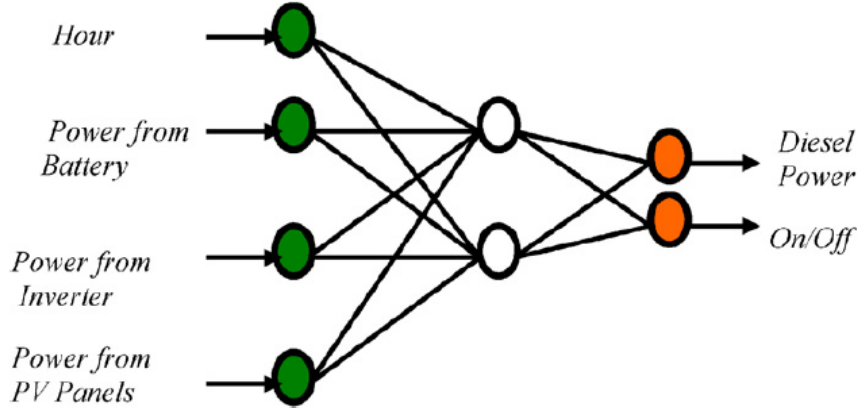


Fig. 3.2: ANN model from [5]

The artificial neural network was trained with simulation results and tested with a real model. The rating of the system was as follows. The solar panels produced 1.2 kW under standard test conditions with rated voltage at 110 V. A 5 kW Honda diesel engine was used connected to a 6 kVA generator. Nine 12 V 125 Ah SunGel200 batteries were used. The bi-directional inverter had a rating of 5 kVA. The reverse osmosis unit had a capacity of 1 m³/day, with a power demand of 0.78 kW.

The system was tested through 186 cases that covered all the different operating conditions (different production from the solar panels, different load

etc.). The ANN model decided when to turn on and off the diesel engine as well as it estimated the power needed from the diesel engine. The needed power was estimated with an average accuracy of 97 %.

The model did not include predictions and was only for made for secure operation in every operating point. It illustrates the ability neural networks have to store and use measurements in an efficient way. To cover the small mistakes in estimation error a standard PI-controller is used. With this controller the stability in the AC grid was maintained.

3.2 Hybrid energy storage

Energy storage systems are often designed using a hybrid power supply. Then the qualities of different energy storage mediums can be combined to satisfy the demand of the load and to prolong the lifetime of the energy supply. The combination of a fuel cell and a battery makes it possible to downsize the fuel cell to only cover average power. A battery has a much faster response than a fuel cell and can be used to cover the peaks. The same combination can be done with high energy density/low power density batteries and super-capacitors. It is essential to share the power flow between the different sources at all time.

In [18] a three layer control strategy was proposed for performing power management in a hybrid power supply consisting of a fuel cell and a battery. On the top a pure supervision system chosed between three operating modes: Only battery, hybrid power supply or battery charge. This system was operated based on certain conditions and was implemented in the stateflow toolbox in Matlab. For example one condition was that the minimum fuel cell off time was 3 min. In a wind diesel system such a condition should also be applied to the on/off control of the diesel generator set. It is not good for the diesel engine to be turned on and off too often. The article took non linearities and losses into account when modeling of the components in the system. For example the battery was modeled as the circuit in Figure 3.3, where V_{oc} and R_a are functions of the state of charge (SOC).

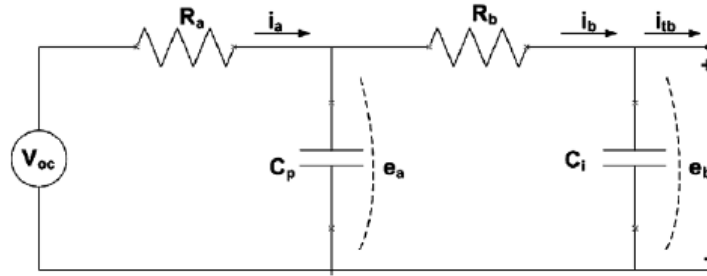


Fig. 3.3: Battery model used in [18]

In the second layer of the control the power reference of the fuel cell was decided based on demanded power, SOC of the battery, the battery power from the previous sampling, the temperature of the fuel cell and the fuel cell power itself from the previous sampling. The schematic view of the fuzzy controller that was used is given in Figure 3.4.

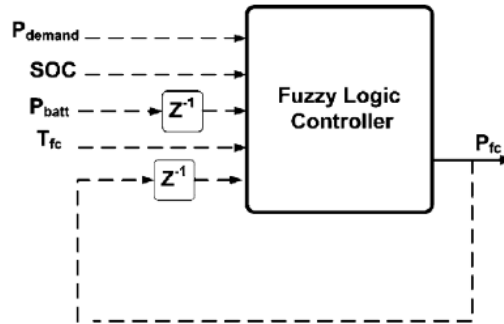


Fig. 3.4: Fuel cell controller in [18]

In the third layer of the controller the duty cycles for the PWM command was calculated based on voltage and power reference values. For the inverter traditional PI-controllers were chosen. However, for the DC/DC converter connecting the fuel cell to the DC-bus a neuro-fuzzy controller was used. The controller manages to share the fluctuations in load power between the fuel cell and the battery in a satisfactory way. When connecting both a fuel cell and a battery to a DC-bus the power sharing can also be done by using a fast PI controller for the battery converter and a slow PI controller for the fuel cell. This, however, can lead to stability problems if the battery runs empty,

because the SOC of the battery is not a part of the control strategy.

3.3 Prediction

To perform an optimal power flow it is often useful to know what one can expect. Artificial intelligence can be used to predict based on previous measurements.

3.3.1 Wind power prediction

In [2] a method for forecasting wind speed production using an adaptive neuro-fuzzy inference system (ANFIS) was presented. A classification of power forecasting depending on how far ahead the power should be predicted was defined. This classification is shown in Table 3.1 and will be used throughout the report.

Power forecast type	Time span
Intermediate term	6 hours - 2 days
Short term	30 minutes - 6 hours
Very short term	less than 30 minutes

Table 3.1: Power forecast classification

For intermediate term wind speed forecasting numeric weather prediction (NWP) techniques are frequently used. These techniques are based on flow equations. NWPs proves to be less suitable on a short time interval because it takes too long time to do the calculations. For doing short term prediction the NWPs have often been supplemented with so called persistence models. They are based on correlation between the present speed and the speed some time ahead. The ANFIS model in [2] was an alternative to the persistence models. The big advantages of the ANFIS model were that it is fast and that it does not include modeling of the wind farm topography. The same ANFIS model can be applied to different wind farms as long as it is trained with the measurements from each wind farm. In the article the wind speed was predicted 2.5 s ahead in time based on 4 previous measurements. More input variables would probably have given a more accurate forecast but a too long

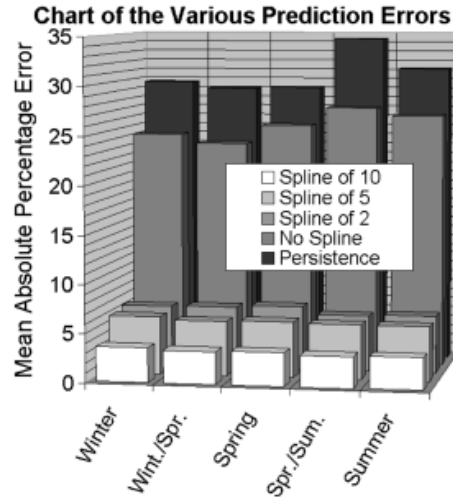


Fig. 3.5: Results of different prediction methods in [2]

training time. Several simulations were done with the ANFIS model using different raw data. The results were compared with a persistence model. The input data to the ANFIS model was splined. This is an interpolating technique. The ANFIS model always took in 4 inputs but the spline interpolated different number of points between the measurements. Spline is not useful for persistence models because the correlation is already known for interpolated points. As seen in Figure 3.5 the output of the model depends heavily on the spline. The prediction tested over 8 months gave an error of 30 % for the persistence model. The best ANFIS model had a prediction error of about 3 %. Without splining the improvement was of about 5 %.

3.3.2 Load prediction

There are many reasons for predicting the power load in an electricity grid. On a long term basis load prediction is used for maintenance scheduling and for taking decisions on whether or not to buy new equipment. If it is a marked with coal or gas fired powerplants it is important to buy the right amounts of fuel. Short term prediction of the load is important to insure that the grid is in operation at any time. In the nordic countries the electricity is traded in a marked. Every day before 12 o'clock the electric power producers have to report to the stock marked how much energy they are going to produce the following 24 hours [7]. Knowing what is coming is essential to gain as much money as possible. In [8] a method for predicting the spot price directly was proposed. This is not relevant for stand-alone power supply where the price only is relevant when planning.

A more interesting approach can be found in [9] where a method for predicting the power consumption one hour ahead using a fuzzy logic system was presented. The measured data were from peninsular Malaysia and the software tool FuzzyTECH 5.52 was used to implement the fuzzy logic model. The inputs of the model was previous load demand, temperature and time. The output was the predicted load at the indicated time. The model did not take into account weekly or monthly variations, nor special events such as festivals or other events when the electricity demand was expected to be abnormal. The previous load demand and the load estimate was given by the linguistic variables *very small*, *small medium*, *large* and *very large*. The temperature was given in *low temp*, *medium temp* and *high temp*. All the fuzzy variables were normalized between 0 and 1. Triangular and trapezoidal shapes were chosen for the membership functions. This is often more robust than bell functions [2][8]. An example of a rule can be *If* (previous load demand is small and temperature is low temp and time is 00) *then* (load demand is small). With the given membership functions there were many rule combinations. In the end 210 rules were used. The center of gravity method was used for defuzzification. The fuzzy model was tested on measured data from Wednesday 26th of Feb 2003. The average error was of less than 1 %. The biggest error was 2.26 % and the smallest 0.03 %. Both these estimates were too low.

1 % error is quite good. The prediction could most likely be improved by

using an ANFIS model. An ANFIS model is easier to train from measurements. The simple fuzzy logic model can be tuned by modifying the membership functions. As for the structure in artificial intelligence models in general there is no algorithm for deciding what membership functions to use. Different variations must be tested. In the fuzzy model a overlap of 25 %-50 % is often regarded as appropriate [3].

Estimation one day ahead based on AI perform as good as statistic techniques. Often, however, the focus on special events such as Christmas, 17. of May etc. is more important than changing the calculating algorithm to one based on AI [26].

Chapter 4

Control

In a stand-alone power supply the dynamic behavior given by for example change in load, diesel generator set on/off and change in production of renewable energy must be handled at all time. If a AC-grid is being used a frequency of 50 Hz along with a given voltage must be kept within acceptable limits. For doing this controllers are installed both on the diesel generator set and on the inverter connecting the battery to the AC-grid. The voltage can be controlled with the excitation of a synchronous generator. The generator can also turn even if the diesel motor is stopped (using a clutch). In this case the generator will work as a synchronous condenser delivering reactive power to the grid and stabilizing the grid with its inertia. The control of the diesel generator set will not be further discussed here.

There have been presented a lot of articles about the use of ANNs in control [12]. Examples are current control of inverter drives, prediction of trajectories in robot control, control of turbo generators and regulation of temperature. In this chapter mainly the control of DC/DC converters is presented. A brief presentation of the inverter is also given. The information is based on literature study only.

4.1 Inverter

The inverter is an important component in a stand-alone plant. AC grids are normally used and the inverter is needed to interface batteries and solar panels with the grid.

4.1.1 Voltage source inverter

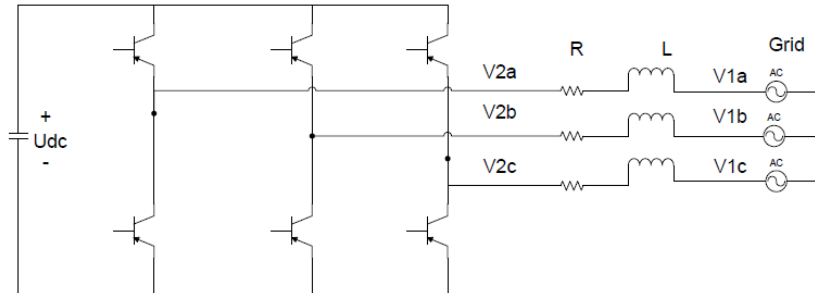


Fig. 4.1: Voltage source inverter, source [25]

In the latest years the voltage source inverter (VSI) has become the preferred converter for medium power and medium voltage applications. A circuit diagram of a VSI is shown in Figure 4.1. Here the diodes in anti parallel with the transistors are not drawn. As switches IGBTs are often used because of their high efficiency and their fairly high maximum voltages. PWM control is preferred because it reduces the size of the filters needed between the converter and the grid.

4.1.2 Phase locked loop

To synchronize the converter with the frequency in the grid a phase locked loop (PLL) is used. The schematic view of a PLL is shown in Figure 4.2. The PLL gives the θ to the vector control. θ_{ref} is usually zero, f_n is 50 Hz and T_{dq} are the transformations given in the next section. The controller and filter parameters must be set to suitable values.

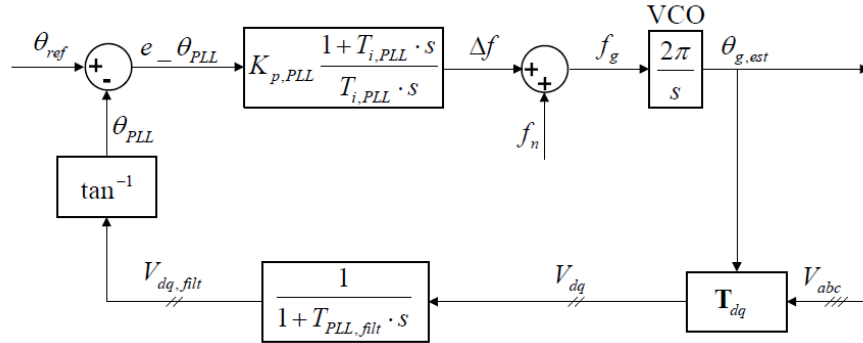


Fig. 4.2: Phase locked loop source [14]

4.1.3 Vector control

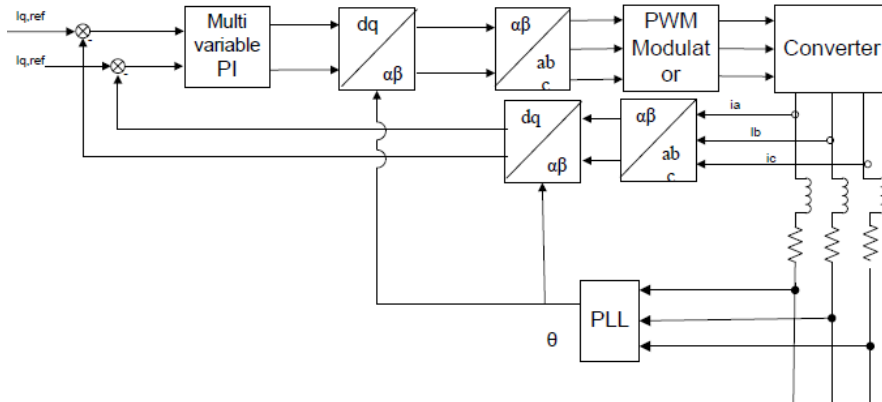


Fig. 4.3: Vector control schematic, source [25]

Vector control is often used in electric drives and is also suited for the control of the voltage and the frequency in a AC minigrid. The idea behind is to transform the sinusoidal voltages and currents into constant values. The transformations are given in the Equations 4.1 - 4.2 [27]. In these equations a symmetrical three phase system is assumed. A can be either voltage or current. $\omega t = \theta$ is given by the phase locked loop. When changing to control based on artificial intelligence it is most likely the best to keep the transformations that are used in classical control and only replace the controller (the block multivariable PI in Figure 4.3). The use of an intelligent controller

could for example decrease the derivative of the current. Less variations in current can increase battery lifetime. A disadvantage with such a controller is that it most likely pushes the system closer to the limit of instability, as it does not respond so fast. The use of AI controllers for inverters has not been further investigated.

$$\begin{bmatrix} A_d \\ A_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \sin(\omega t) & \sin(\omega t - \frac{2\pi}{3}) & \sin(\omega t + \frac{2\pi}{3}) \\ \cos(\omega t) & \cos(\omega t - \frac{2\pi}{3}) & \cos(\omega t + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} A_a \\ A_b \\ A_c \end{bmatrix} \quad (4.1)$$

$$\begin{bmatrix} A_a \\ A_b \\ A_c \end{bmatrix} = \begin{bmatrix} \sin(\omega t) & \cos(\omega t) \\ \sin(\omega t - \frac{2\pi}{3}) & \cos(\omega t - \frac{2\pi}{3}) \\ \sin(\omega t + \frac{2\pi}{3}) & \cos(\omega t + \frac{2\pi}{3}) \end{bmatrix} \begin{bmatrix} A_d \\ A_q \end{bmatrix} \quad (4.2)$$

4.2 DC/DC converter

The most important use of DC/DC converters in stand-alone power plants is to connect solar panels to the inverter. This is important to operate the solar panels at a voltage that gives the maximum power. If a DC grid is being used the DC/DC converter becomes more important. DC/DC converters are nonlinear. Because of their non linear behavior DC/DC converters and control based on artificial intelligent techniques go well together. In this section control of the boost (step-up) converter is treated. The circuit diagram for such a converter is shown in Figure 4.4.

4.2.1 Transfer function model

A state space model of a boost converter is given in Equation 4.3 [19]. x_1 is capacitor voltage and x_2 inductor current. This equation is highly non-linear because D is a function of V_{in} and the capacitor voltage. In fact D is to be controlled by the PI or fuzzy controller.

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -(1-D)\frac{1}{L} \\ (1-D)\frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_{in} \quad (4.3)$$

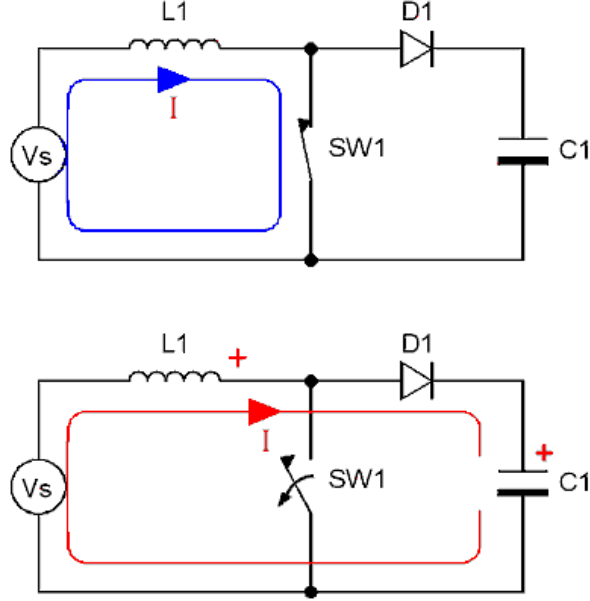


Fig. 4.4: Schematic of boost converter

The jacobian method can be used to linearize Equation 4.3 [20]. The linearized equation is given in Equation 4.4. Here Δx_1 , Δx_2 and ΔD are the state space variables. The equation is valid for small deviations from the operating point (x_1, x_2, D)

$$\begin{bmatrix} \frac{d\Delta x_1}{dt} \\ \frac{d\Delta x_2}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -(1-D)\frac{1}{L} \\ (1-D)\frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L}x_2 \\ -\frac{1}{C}x_1 \end{bmatrix} \Delta D \quad (4.4)$$

The transfer function from ΔD to ΔV_{out} can be found based on the state space equation using the formula in Equation 4.5. C is in this case the measurement matrix.

$$H(s) = C(sI - A)^{-1}B \quad (4.5)$$

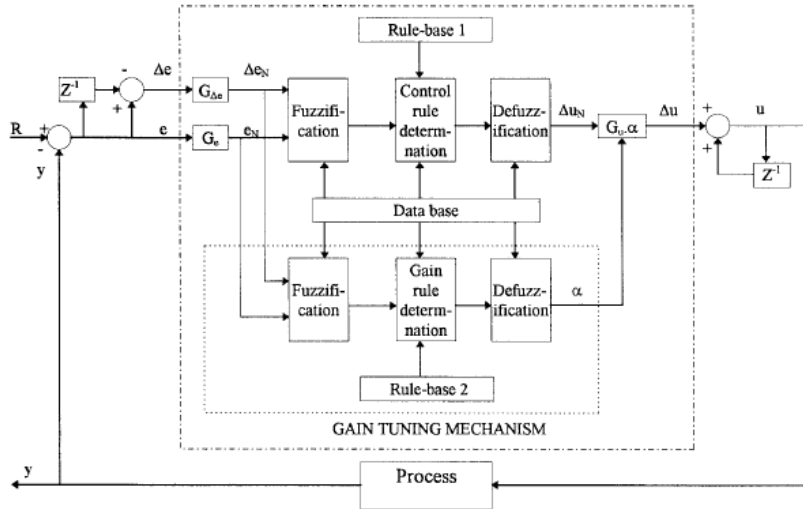


Fig. 4.5: Fuzzy controller with output gain tuning, source [23]

4.2.2 Fuzzy control of DC/DC converters

There are many different ways of implementing a fuzzy controller. One thing that has to be designed is the logic within the controller like the number of membership functions and rules. In the literature for fuzzy control of converters it seems like the standard is to have the error and the derivative of error as input and change in duty cycle as output. Usually the input and output values are between -1 and 1 or 0 and 1. Then the inputs and outputs must be scaled to fit in with the process. Adjusting the output gain on line with an other fuzzy controller in parallel can give better response in some cases [22]. This is shown schematically in Figure 4.5.

In [15] an adaptive fuzzy logic controller (AFLC) for controlling DC/DC converters was presented. The rules were decided in forehand but the properties of the trapezoidal membership functions were adjusted to suite the converter and the load. The controller was applied to three converters: Buck, boost and buck-boost. Simulations were used for tuning the controller and the converters were tested with hardware. The criteria for the adaption algorithm was the least error. For the practical tests the AFLC was implemented into a *ST52E420* micro controller. The goal was to control the output voltage. The

inputs of the fuzzy controller were output voltage error and change in error. The output of the fuzzy controller was the change in duty cycle. The adaptive fuzzy controller however could control all the three converters within acceptable deviations of voltage.

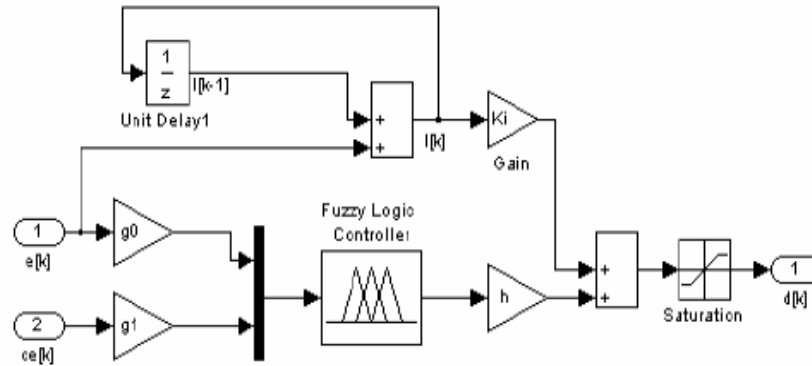


Fig. 4.6: Fuzzy controller with integrator in parallel, source [22]

For DC/DC boost converters good results have been found by using one fuzzy controller with a simple integral part in parallel [23][24]. In Figure 4.6 this structure is shown. This structure is not adaptive, however the controller parameters must be tuned. If one has a good simulation model of the physical system this model can be used to tune the controller. This controller is further investigated in the next chapter.

Another strategy can be to tune K_p and K_i in a PI controller with a fuzzy controller. The idea is to have larger K_p when the error or change in error is big. There are several different strategies for developing such a controller [28][29][30]. One fuzzy tuned PI controller tuned is developed in the next chapter.

4.2.3 Maximum power point tracking

Due to the physics of the semiconductors solar panels have a certain i-v characteristic. This means that for each and every solar radiation and tempera-

ture (in the semiconductors) it exists an operating voltage on the terminals of the solar panel that gives a maximum power output. The search for the maximum power is called maximum power point tracking (MPPT) and different algorithms are proposed for finding the ideal operating point [11].

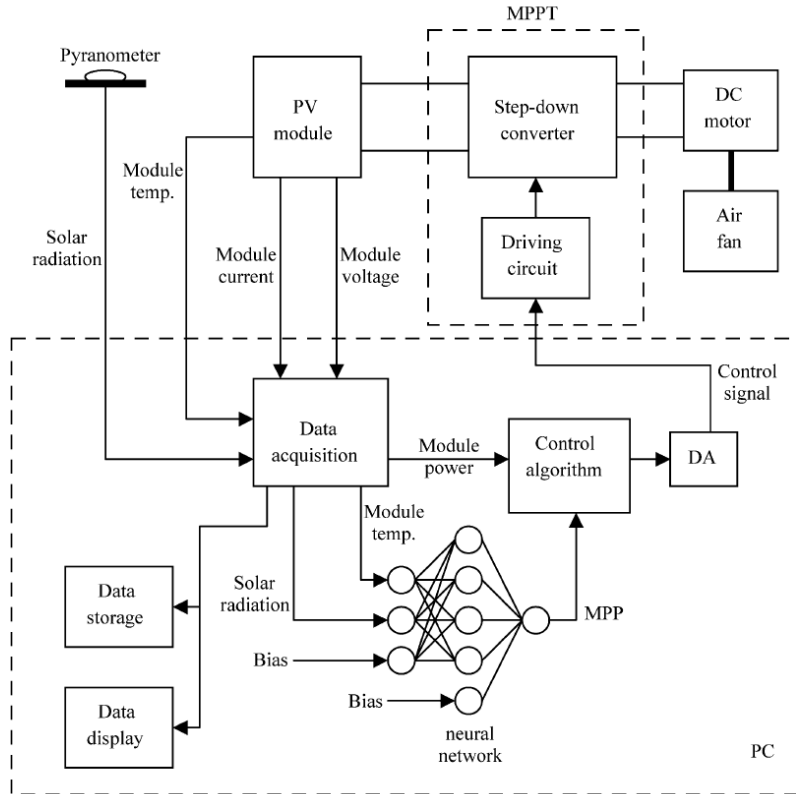


Fig. 4.7: Schematic diagram for MPPT, source [10]

In [10] a method for doing maximum power point tracking (MPPT) of solar panels using artificial neural networks was proposed. The high power system to control consisted of a solar panel, a step-down converter and a DC-motor as load. See Figures 4.7 and 4.8. The control system was implemented in a PC. A data acquisition system converted the analog measurements of solar radiation, panel temperature as well as current and voltage measurements into digital signals that the PC could read.

In the controller the actual power drawn from the solar panel was calculated

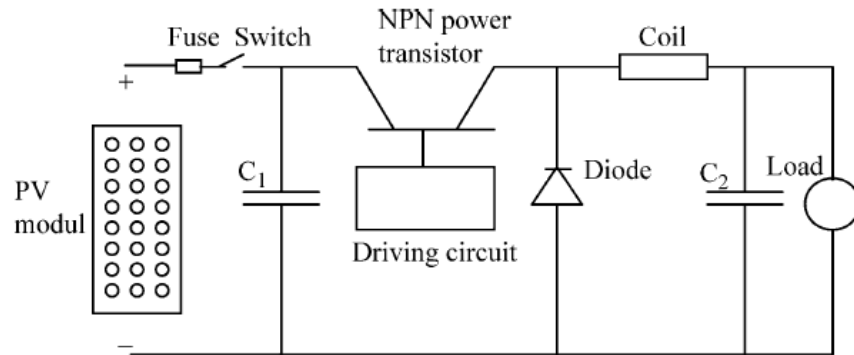


Fig. 4.8: Step-down converter, source [10]

using the voltage and current measurements. The maximum power that was possible to achieve was given by an artificial neural network. The inputs to the ANN model were the solar radiation and the temperature of the solar panel. If the difference between the actual power and the achievable power was bigger than a certain limit, ΔP_{max} , the perturb and observe technique was activated. With the perturb and observe technique the control signal was changed in one direction. Then the power output was measured again. If the power output was higher after the perturbation than before the control signal was changed more in the same direction. If not, the direction of the perturbation of the control signal was changed. If the measured power arrived at a maximum the control signal remained unchanged and a comparison was made with the neural network again. A flow chart of the algorithm is shown in Figure 4.9.

The power output of the system was of about 97 % of the theoretical achievable power. The article said nothing about how the neural network was trained. This can be done on site, which demands a powerful digital controller. This gives a result that is optimal for each and every site and panel. The neural network can also be trained with simulations or one panel of each type can be tested and one can assume that the rest of the panels behave the same way. A problem with some MPPT techniques is that they perturb all the time. As a result of that the drawn power is most of the time less than the maximum power. This was not the case with the control system treated in the article referred to in this paragraph, because the perturb and observe

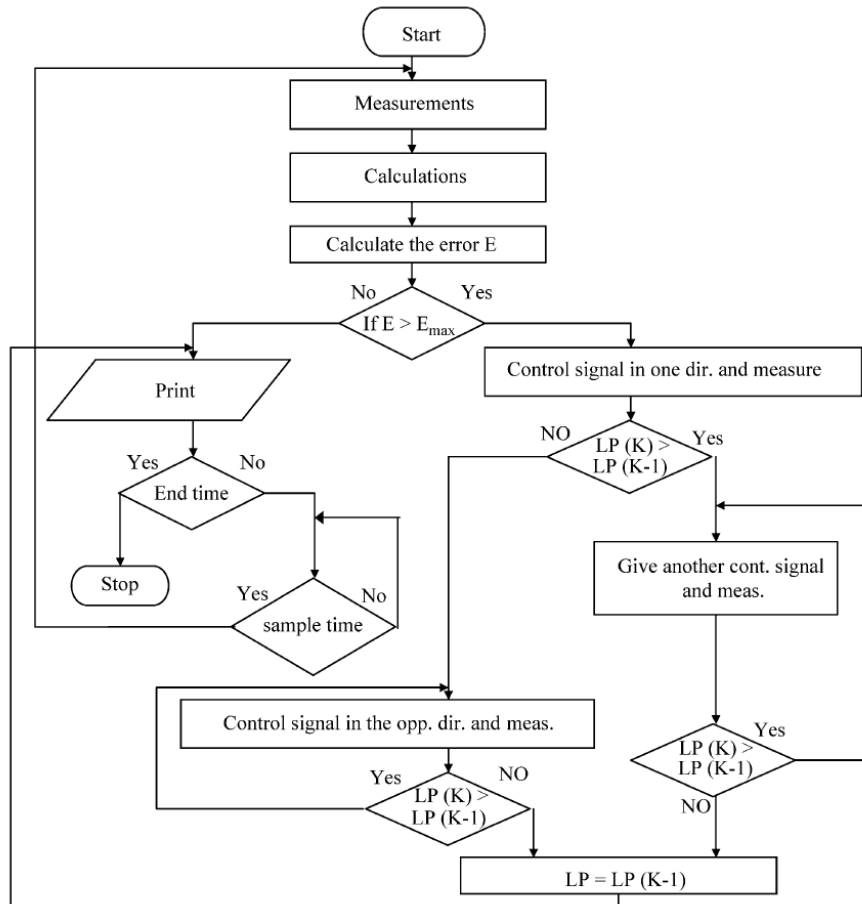


Fig. 4.9: Flow chart, source [10]

algorithm was only activated when the difference between the actual power and the theoretical maximum power was bigger than a certain limit. Another problem with many MPPT techniques is that they can stop the iterations at a local minimum. This was not mentioned in the article [10]. A way of solving this could be to have the voltage corresponding to the maximum power as a output of the neural network model instead of just having the maximum power as the output. Then the perturb and observe algorithms starts to search near the right voltage immediately.

4.2.4 Comparison of different controllers

In [16] a FLC, a PI-controller and a fixed frequency sliding mode controller were compared for the control of a step-down (buck) converter. Sliding mode control is also known as hysteresis control and is a very robust non linear control strategy. The PI-controller was of pure classical type. It was designed using bode plots. Suitable phase and gain margin were chosen. In PI controllers there is a trade off between speed and robustness. A smaller gain margin makes the controller faster but closer to the limit of instability. Transfer function theory is not developed for fuzzy controllers. They have to be designed on the bases of experts or by adaption algorithms. The input of the fuzzy logic controller in [16] was output voltage error and change in error. The output was change in duty cycle. The fixed frequency sliding mode controller (FFSMC) will not be discussed in detail here. For the comparison two sliding mode controllers were developed, *SMC1* and *SMC2*.

For the simulations three phenomena were investigated, start up, change in load and change in input voltage. For start up the FLC had the longest response time. FLC and PI had the same overshoot and SMC2 had the biggest overshoot. SMC1 was the controller with best response for start up. For change in load SMC1 had the biggest overshoot. The overshoot was almost the same for PI, FLC and SMC2, but the response time for the FLC was better than the others. The behavior of the different controller was very different for the change in input voltage. FLC and SMC2 proved to be the best controllers for handling the change in input voltage. FLC was claimed to be easier to implement than SMC2. Both these controllers were regarded to have the best over all dynamic performance.

4.3 Commercial products

Producers might not always write in the data sheet that they have used AI-techniques. There are however some industrial products that we know use these techniques. The J300 Series IGBT inverters from Hitachi use fuzzy logic control. It calculates optimal acceleration and decelerations times. The control is sensorless and is based on rotor resistance estimation. The VS-615G5 from Yaskawa Drive uses a neural network as flux observer.

Chapter 5

Simulations on Boost converter

In this chapter simulations in a boost converter are performed. Different control strategies are tested. A classical PI approach is compared with two controllers using fuzzy logic. This chapter is based on the authors work using the theory presented in Chapter 4.

5.1 Converter setup

5.1.1 Parameters

A boost converter was to be controlled. It had the parameters given in Table 5.1. The parameters were chosen based on an experimental setup treated in the next chapter. The circuit diagram for the converter is shown in Figure 5.1.

5.1.2 Transfer function model

The boost converter is a highly non linear system. The parameters of the PI controller must therefore be decided based on a certain operating point.

Parameter	Value
L	37 mH
C	3300 μF
R_{load}	331 Ω

Table 5.1: Parameters boost converter

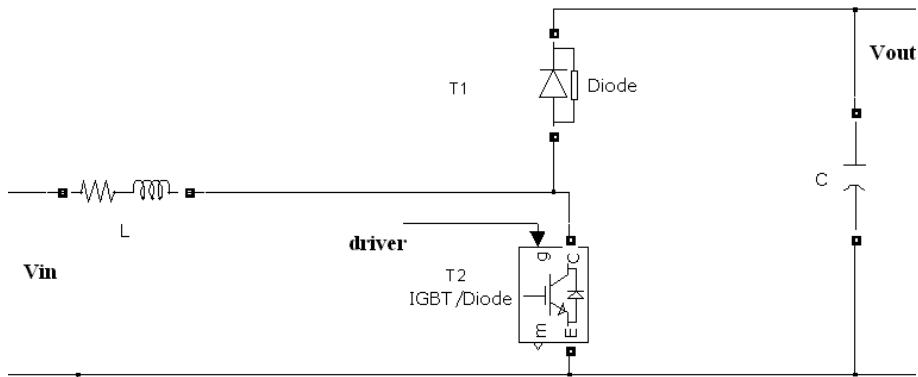


Fig. 5.1: Circuit diagram boost converter

Non-linear controllers like fuzzy controllers often have a wider operating area. A bode plot for the transfer function presented in Section 4.2 with the parameters for the converter in the simulation model is shown in Figure 5.2. One can read from the figure that the bandwidth is smallest when the input voltage is low.

5.2 Controller design

Description	Abbreviation
PI controller	PI
PI controller with fuzzy tuning	PI_{tuned}
Fuzzy logic controller with integrator in parallel	$FLC_{para-int}$

Table 5.2: Abbreviations of controllers

In this section three different controllers are designed. The abbreviations are

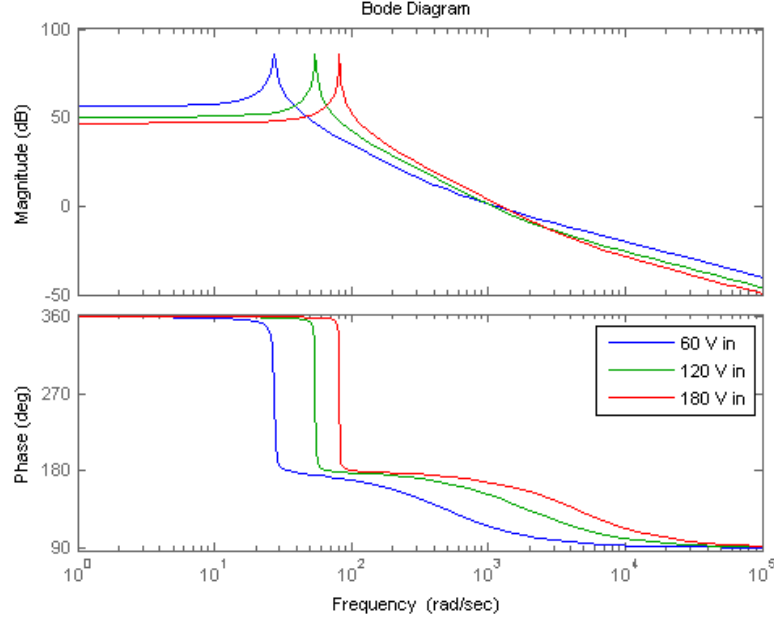


Fig. 5.2: Bode plot of transfer function for boost converter

given in Table 5.2. Both of the controllers using fuzzy logic had triangular membership functions and 7×7 rules. All controllers had a saturation block that limited the output of the controller between -0.9 and 0.9. Anti wind-up was also used for all the controllers.

5.2.1 PI controller

To design the parameters of *PI* Ziegler-Nichols method was used. This is an experimental method for finding suitable controller parameters [20]. A proportional controller was used at first. The gain was increased until a standing oscillation was observed at the exit. The critical gain K_{pc} and the period T_{ic} of the oscillation was noted. Suitable controller parameters are found by multiplying the values of K_{pc} and T_{ic} by 0.45 and 0.85 respectively. These constants are empiric constants. The transfer function for *PI* is given in Equation 5.1. The values of K_p and K_i are given in Table 5.3. It is seen from Figure 5.2 that the bandwidth is smallest at minimum input voltage.

The controller parameters must therefore be designed based on this operating point.

$$h_{PI}(s) = \frac{K_p s + K_i}{s} \quad (5.1)$$

Input voltage	K_p	K_i
60 V	8.1	27.36
180 V	17.1	81.97

Table 5.3: Parameters for PI

5.2.2 PI controller tuned with fuzzy controller

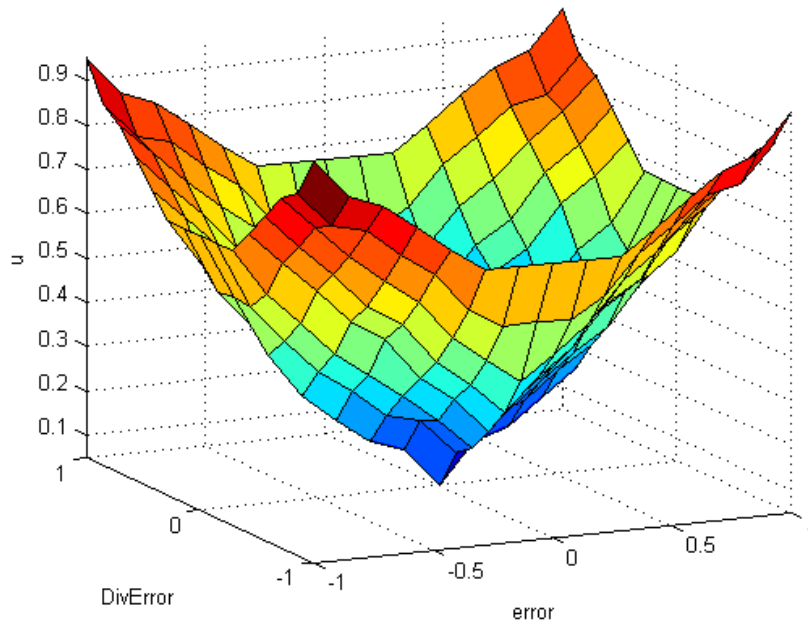
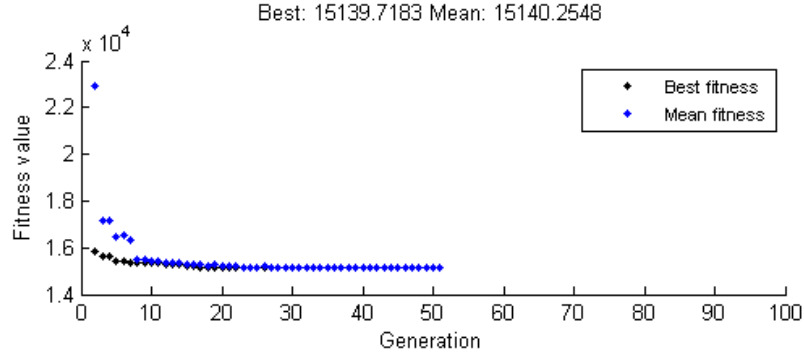


Fig. 5.3: Surface plot PI_{tuned}

The parameters of a PI controller can be tuned using a fuzzy controller. This can be done by having a forward coupling from the disturbance or it can be

Fig. 5.4: Optimization PI_{tuned}

done more generally by only tuning the parameters based on the output error. In the simulations a fuzzy controller with error and derivative of error was used to tune the PI controller. The surface plot of PI_{tuned} is shown in Figure 5.3. It shows the output of the fuzzy controller as a function of the two input parameters error and change in error. As seen in the figure the rules were made for making K_p larger when the error or the change in error was big.

Since there is no transfer function theory for fuzzy controllers other methods must be used to decide the parameters. It was chosen to use genetic algorithms for doing this. All the sets of parameters were tested on the simulation file of the boost converter. The average and least error in every generation is shown in Figure 5.4. It is clear that the optimization converges. The error is the absolute value of the output voltage error in pu for a 1 s simulation with step time of 1 μ s. For a brief introduction to genetic algorithms see Section 2.4.

The optimal parameters are shown in Table 5.4. k_1 and k_2 are input gains of the fuzzy controller for error and difference in error respectively. k_3 is the offset of K_p . k_4 is the output gain of the fuzzy controller. The formula for K_p is given in Equation 5.2 where f_{FL} is the output of the fuzzy controller. K_i is set to $3.4K_p$ since this is the relation between the two constant in Table 5.3 (60 V input voltage). It is seen from Table 5.4 that the K_p is significantly higher for this converter than for the one designed with Ziegler

Nichols method.

$$K_p = k_3 + k_4 f_{FL} \tag{5.2}$$

Parameter	Value
k_1	73.2
k_2	9598
k_3	20
k_4	20

Table 5.4: Parameters for PI_{tuned}

5.2.3 Fuzzy controller with integrator

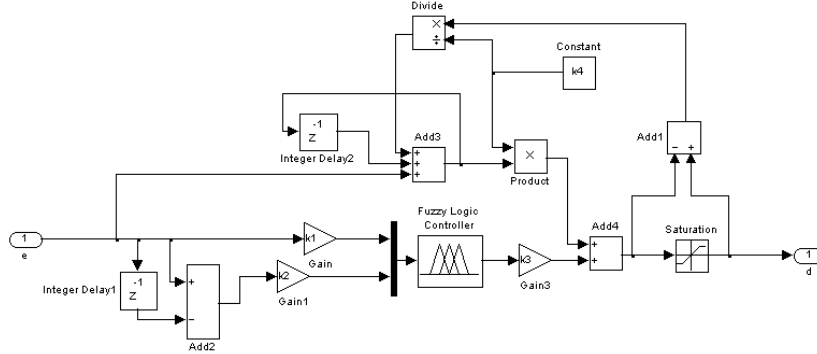


Fig. 5.5: Schematics $FLC_{para-int}$

A fuzzy controller with an integrator, $FLC_{para-int}$, was designed. This structure has been found in [23] and [24], however a anti wind-up was added. This gave faster response. The schematics of the controller is shown in Figure 5.5. The surface plot for $FLC_{para-int}$ is shown in Figure 5.6. The rules for this controller is taken from [23]. The genetic algorithm toolbox in Matlab was used to tune the four parameters of $FLC_{para-int}$; gain for error, gain for derivative of error, output gain of fuzzy controller and integrator gain. The

fitness of the different generations in the optimization is shown in Figure 5.7. The blue dots are the average in every generation, whereas the blue ones are the best individuals. It is seen that the error is much less than it was for PI_{tuned} . The result of the optimization is given in Table 5.5.

Parameter	Value
k_1	100
k_2	6315
k_3	3.21
k_4	0.023

Table 5.5: Parameters for $FLC_{para-int}$

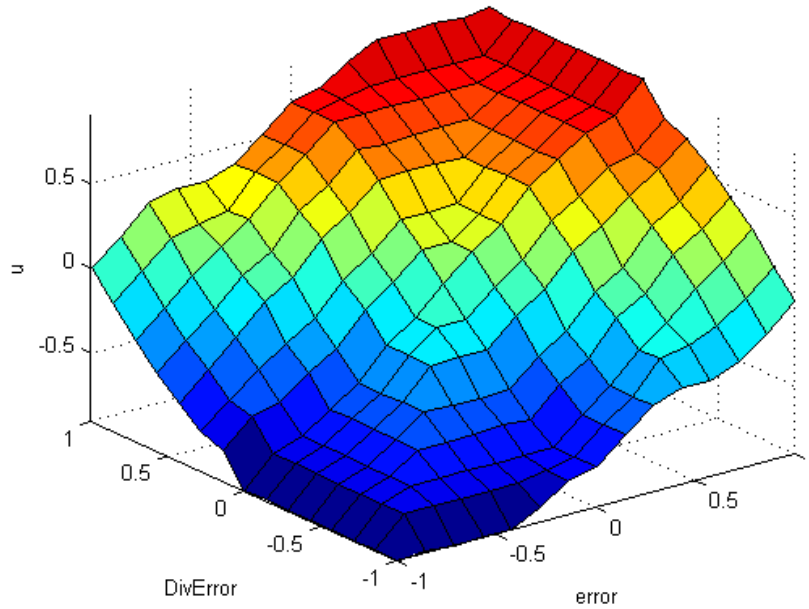
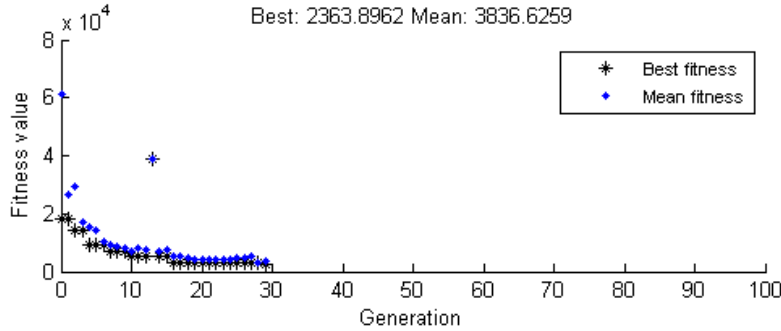


Fig. 5.6: Surface plot $FLC_{para-int}$

Fig. 5.7: Optimization $FLC_{para-int}$

5.3 Description of simulation case

The response of the output voltage of the boost converter given a perturbation was investigated. The perturbation in this case was a change in the input voltage. The voltage steps down from 180 V to 60 V after 0.5 s. The simulation lasts for 1 s. The output capacitor had an initial voltage of 200 V. The robustness of the fuzzy controller with integrator in parallel was also tested. This was done by parametric variations.

5.4 Results and discussion

5.4.1 Voltage response

The voltage response is shown in Figure 5.8. The input voltage is the graph at the top, with the output voltage using the three different converters below. $FLC_{para-int}$ had the least deviation from the reference voltage and the shortest settling time. The maximum error in the output voltage is only 3 % for $FLC_{para-int}$ compared to 20 % for PI . The details for all the three controllers are given in Appendix B. PI_{tuned} is quite aggressive because of the large K_p . Oscillations are expected because of this. However there are oscillations in the controller loop for both the controllers with fuzzy logic.

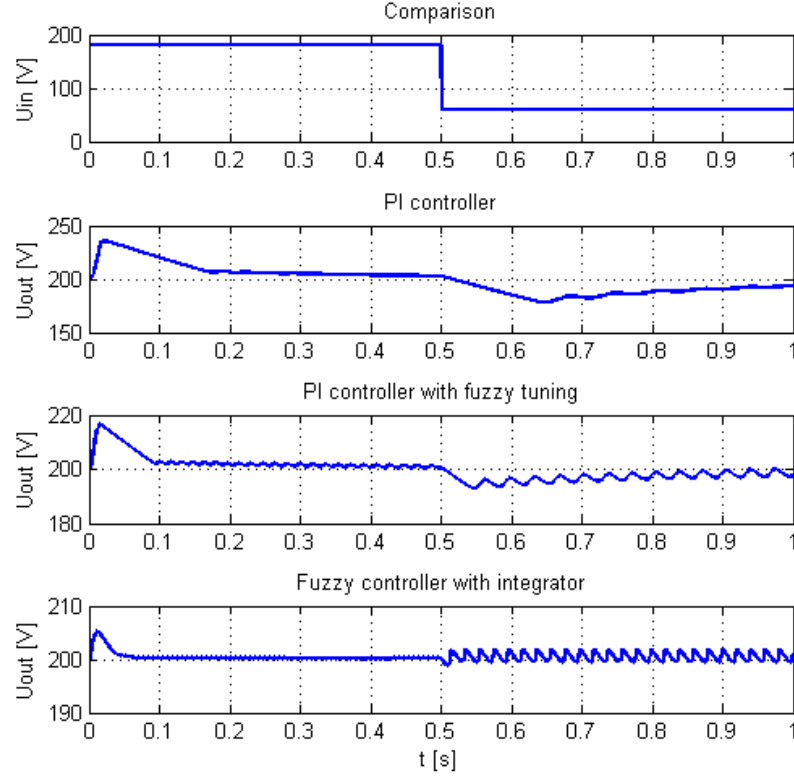


Fig. 5.8: U_{in} , PI , PI_{tuned} and $FLC_{para-int}$

5.4.2 Parametric variations

$FLC_{para-int}$ was tested for 10 %, 20 % and 30 % increase and decrease in L and C , and R_{load} . At 10 % variation no difference was observed. With 20 % variation the response of the output voltage was still the same, except from when R_{load} was decreased. Then large oscillations occurred in the output voltage (200 – 250 V). At 30 % variations the simulations stopped because the anti wind-up became an algebraic loop. It is assumed that this error is related to the simulation program and that it is not a problem for the implementation of the controller.

5.4.3 Discussion

The good performance of $FLC_{para-int}$ represents a major amelioration for the control of boost converters. But although settling time was shorter and error was smaller for $FLC_{para-int}$ compared to PI there are concerns. As previously mentioned there is no transfer function theory for fuzzy controllers. Genetic algorithms proved to be an effective way of optimizing the controller parameters, however the stability can never be guaranteed as it is impossible to test the system for an infinite number of disturbances. In this chapter an experimental method was used to find the parameters of the PI . Ziegler Nichols is however well documented in theory and transfer functions can always be used for PI controllers to check the phase and gain margins.

An other problem is the oscillations in the controller loop for PI_{tuned} and $FLC_{para-int}$. The reason for these oscillations were not understood. However, the frequency of the oscillations seems to correspond with the resonance frequency of the converter. The resonance frequency for different operating points can be seen in Figure 5.2. The reason for the big oscillations that was experienced when R_{load} was decreased could be that the resonance frequency of the converter was slightly changed.

Based on the experience from the simulations it is the author's opinion that AI is more suitable for supervisory control, i.e. setting the reference values for classical controllers controlling the converters. If this had been found out earlier the focus of this report would have been shifted.

Chapter 6

Practical work



Fig. 6.1: Experimental setup

In this chapter experimental work is carried out on a boost converter.

6.1 dSPACE

The laboratory experiments were performed using a so called dSPACE. It is a tool that makes it possible to control hardware applications in real time using controllers implemented in Matlab Simulink. The dSPACE is suitable for prototype testing. It has both analogue and discrete inputs and outputs. In this setup the analogue ports were used. The analogue inputs were used for measuring voltage and current. They took in a -10 to 10 V signal. The 5 V PWM signal had its own output port. A more detailed presentation of the dSPACE is given in Appendix A.

6.2 Power circuit

6.2.1 Power supply

In order to do a step in input voltage two variacs connected to a relay were used as power supply. After the relay the voltage was rectified with a diode rectifier and smoothen with a capacitor bank of $6600 \mu\text{F}$. The use of this fairly big capacitor bank resulted in a slow voltage source, unable to step the voltage. An attempt of using a smaller capacitor bank resulted in oscillations.

6.2.2 Converter

The converter that was used was one leg of a 20 kW three phase inverter designed by Kjell Ljøkelsøy at SINTEF. The converter had a $3300 \mu\text{F}$ capacitor bank. The PWM signal was provided by the dSPACE. All other logic such as for example dead time was handled by the converter. The IGBTs could handle a switching frequency of 25 kHz. In these experiments 10 kHz was used. Between the power supply and the converter a 37 mH iron core inductor was placed. This inductor should have been a bit bigger and it should have had a ferrite core. This however is not important for the results. The circuit of the converter as well as a picture of it is shown in Figure 6.2 and 6.3 respectively.

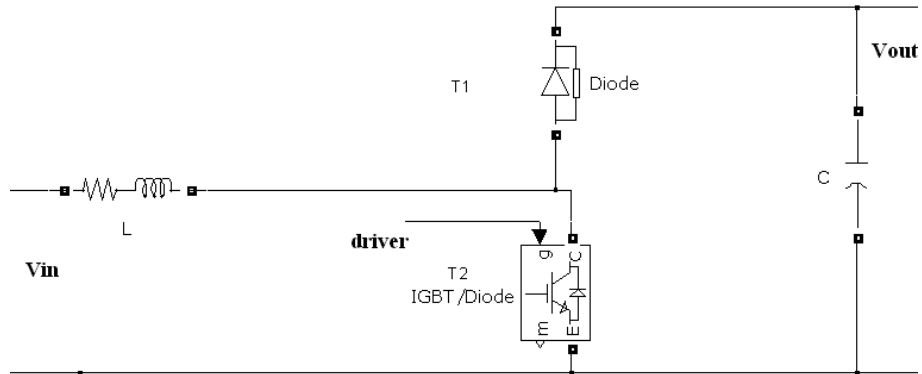


Fig. 6.2: Circuit diagram boost converter

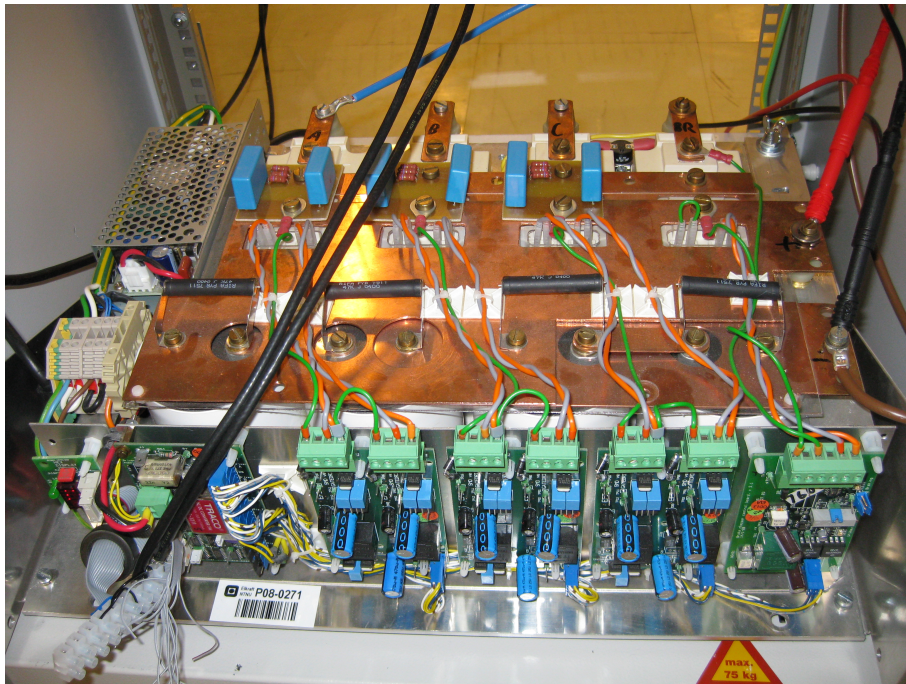


Fig. 6.3: Converter

6.2.3 Load

The load was a water cooled resistance of 331Ω . It was supposed to model an inverter delivering constant power.

6.3 Controller design

6.3.1 PI controller

The PI controller worked with a sampling time of $30 \mu s$. It was the same as the one used in the simulations. See Section 5.2.1.

6.3.2 PI with fuzzy tuning

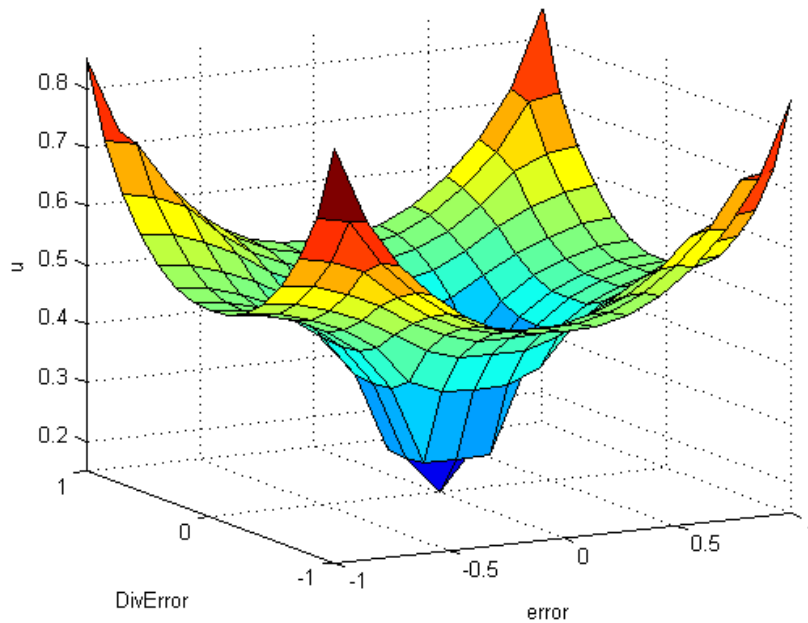


Fig. 6.4: Surface plot of fuzzy controller with 9 rules

The dSPACE could not handle a pure fuzzy controller because of lack of computing power. Therefore only a PI controller with fuzzy tuning was implemented as an alternative to the pure PI. The controller designed in this section has the same structure of the fuzzy part as the one in Section 5.2.2, however with fewer rules. A simple fuzzy controller with 9 rules for tuning a PI controller was tested. This controller had a sampling time of $400 \mu\text{s}$, i.e. four times the period of the PWM signal. A surface plot of the controller is shown in Figure 6.4.

The relationship between K_i and K_p was set to 3.4, as this is the relationship found by Ziegler Nichols method in Section 5.2.1. K_p is given by two parameters: The offset k_3 and the output gain of the fuzzy controller k_4 . $K_p = k_3 + k_4 f_{FL}$, where f_{FL} is the output of the fuzzy controller. Two different cases of k_3 and k_4 were measured.

6.4 Description of experiments

The response of a decrease in input voltage for the power circuit described in the previous section was examined. The voltage dropped from approximately 180 V to approximately 80 V within 0.5 s. The input voltage as well as the output voltage, input current, duty cycle and the output of the fuzzy controllers are shown in the figures in the following sections.

6.5 Results

6.5.1 PI controller

The measurements with the PI controller is shown in Figure 6.5. The output voltage of the converter was at 200 V as it was supposed to be. The measurements are quite noisy. It is more noise for the PI controller than for the PI controller with fuzzy tuning. This is most likely due to the higher sampling frequency of the PI controller. High sampling frequency is not always good. The output of the controller is shown in the figure as *contr*. The duty cycle increased as the input voltage went down, as it is supposed to do.

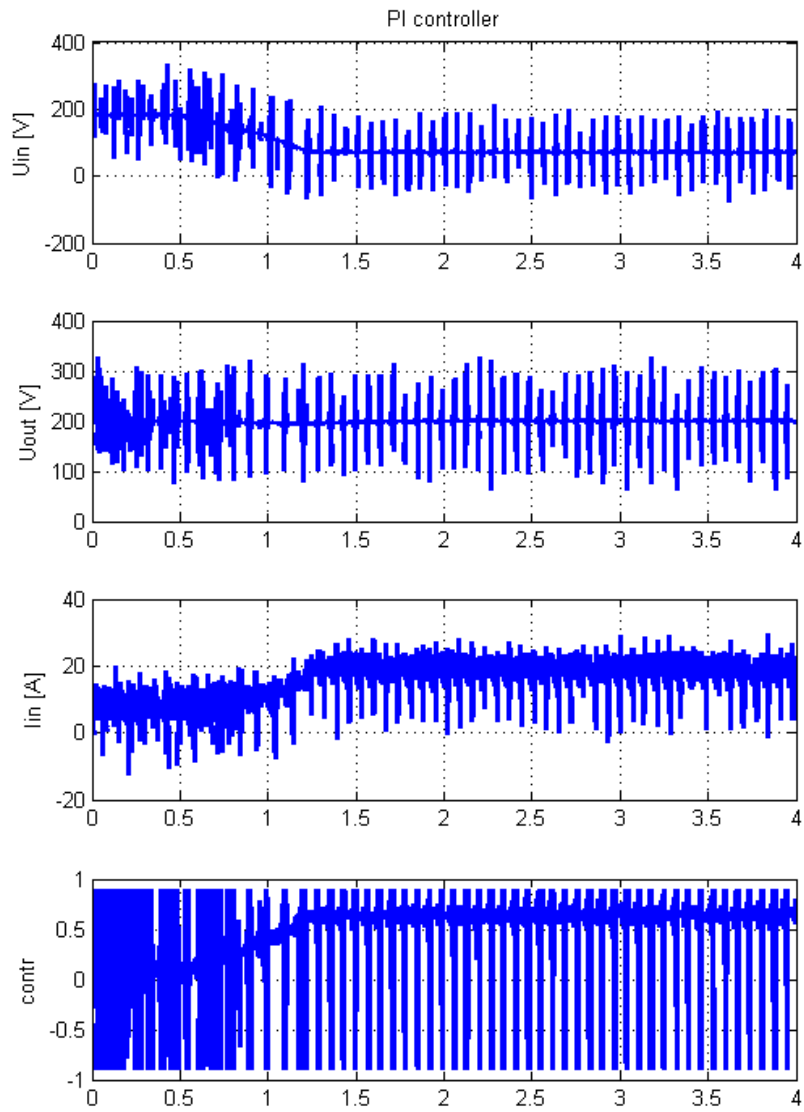


Fig. 6.5: Response of PI controller

6.5.2 PI with fuzzy tuning

In this section the results of the PI controller with fuzzy tuning are presented. In Figure 6.6 the response is shown with $k_3 = 2$ and $k_4 = 8$. This gives a big influence of the fuzzy controller. In Figure 6.7 $k_3 = 8$ and $k_4 = 2$. The constants k_3 and k_4 are explained in Section 5.2.2. The differences between the two graphs are not significantly. The fuzzy controller is not so well tuned. This can be seen in Figures 6.6 and 6.7 where the output of the fuzzy controller is plotted on the bottom. *contr* is the output of the controller.

It was desirable to do more experiments, however problems occurred with the control of the converter. It is seen from the figures that there is a perturbation that shakes the converter and the controller. The responses of this unknown perturbation occurs with a fairly constant time interval, however the length of this interval is different for the two controllers. The reason for this perturbation was not understood.

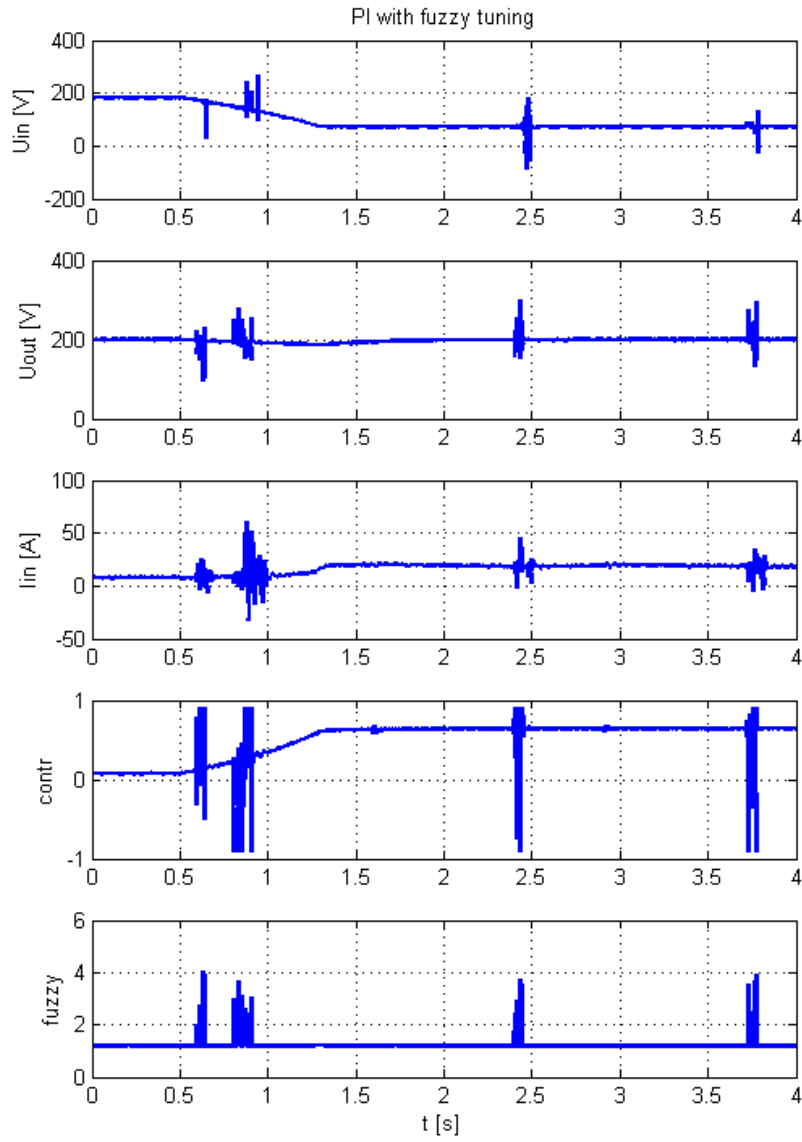


Fig. 6.6: Response of fuzzy tuned controller, $k_3 = 2, k_4 = 8$

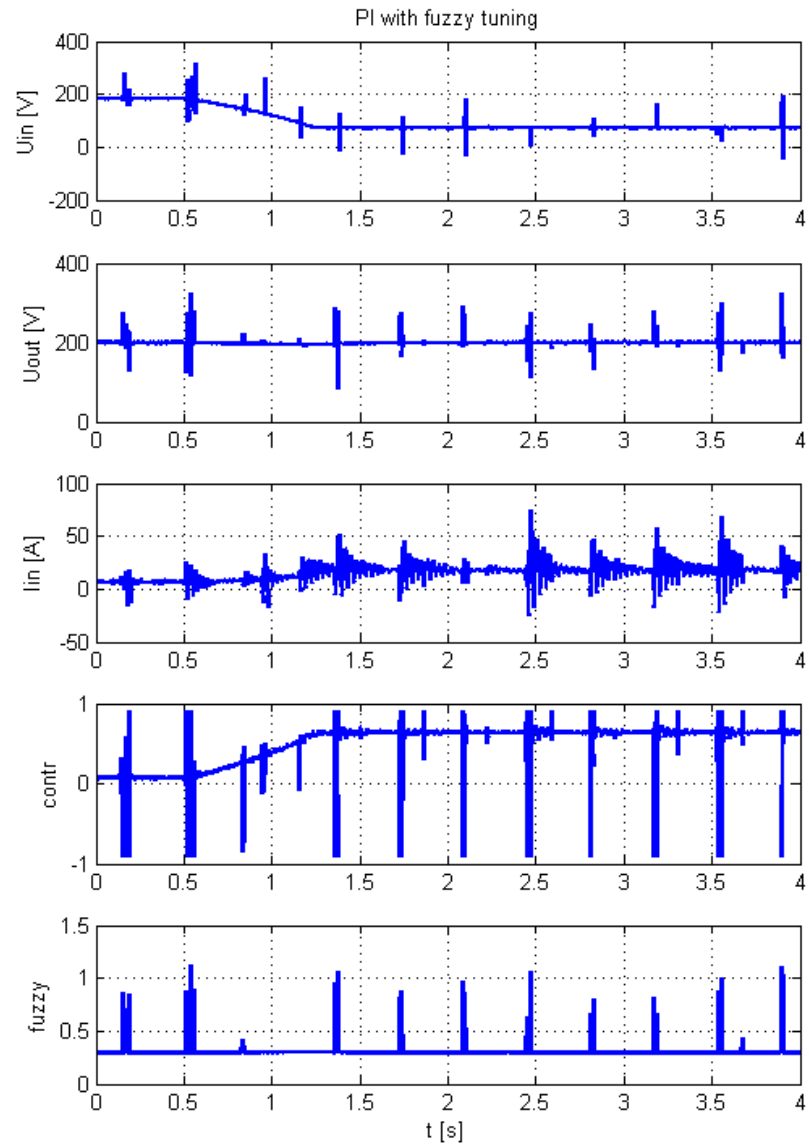


Fig. 6.7: Response of fuzzy tuned controller, $k_3 = 8, k_4 = 2$

6.6 Discussion

Big problems were experienced when controlling the converter with the dSPACE and with finding a suitable power supply. A good voltage source was used at first. This source could step the voltage very fast. The components in the converter were chosen based on having such a quick voltage source. However, the voltage source got broken, and the replacement was a much slower voltage source. Because of the slow dynamics of the input voltage source in combination with the big output capacitor of the converter it was hard to see any difference at all between the controllers. The experiments were however useful for the author and led to an increased understanding of digital control, for example that higher sampling frequency gives more noise.

6.7 Implementation of AI in a DSP

Controllers and estimators based on AI often demand a DSP with higher performance than PI controllers. A 49 rules fuzzy controller with seven membership functions for the two inputs and the output was implemented on the above mentioned dSPACE. Then only a sampling time of 100 ms was achieved, which must be regarded as useless for applications within power electronics. In a dSPACE the matlab code is being translated and downloaded into the DSP. When written directly in C++ the code can be more efficient. In [24] a computing time of 3.6 μ s was achieved when implementing a fuzzy controller with a 33x33 rulebase on a Texas Instruments DSP. The reference PI controller in the same paper had a computing time of 1 μ s. With modern DSPs it is not a problem to implement controllers based on AI, however the price of the DSPs will always be higher than the ones needed to implement a PI controller.

Chapter 7

Conclusions and further work

7.1 Conclusions

For energy management artificial intelligence techniques are especially suited for sharing power within hybrid energy sources. It is also suited for prediction, in particular short term prediction. Wind power 2.5 s ahead in time has been predicted using an adaptive neuro inference system (ANFIS) with an average error of 3 %. This prediction algorithm also had the advantage of being fast.

The most suitable controller designed of the ones tested is a fuzzy controller with an integrator in parallel. The maximal deviation from the reference in output voltage for a boost converter with fuzzy control was 3 % compared to 20 % for a classical PI controller. This must be regarded as a very good result. A disadvantage with fuzzy controllers is the lack of designing procedures. No transfer function theory is developed for fuzzy controllers. As an alternative genetic algorithms proved to be suitable for deciding optimal controller parameters, however the stability can never be guaranteed. Despite the good results obtained when controlling the boost converter in Chapter 5 it is the authors opinion that controllers based on AI are most suitable for supervisory control, i.e. setting the reference values for PI controllers controlling the converters. An exception may be when the converter has a need

for a very accurate response and that it undergoes only a limited number of perturbations.

Fuzzy controllers can easily be implemented in modern DSPs. However, they demand longer computing time than PI controllers and may, with cheap DSPs or inefficient code, be the limitation on the bandwidth of the controller.

7.2 Proposed further work

Artificial intelligence can be useful for many applications within the field of electrical engineering, like for example estimation, prediction, control and even modeling. Within distributed generation the applications for inverters can be further investigated. For example can a non linear control give less variations in the current and thus increase battery lifetime.

It would have been interesting to look at the application within electrical drives. One could have tried to look at alternatives to classical observers and estimators. In that case the controller should be implemented in a DSP directly to get more efficient code than what is achievable with a dSPACE.

Artificial intelligence is still on a stage where different applications must be tested and comparisons must be made to classical approaches. The goal must be to get an overview of the different applications where AI is suited. Based on the research that has been done up until now it seems like this is within supervisory control, non-linear systems, estimation and prediction.

Bibliography

- [1] Artificial intelligence techniques for photovoltaic applications: A review, Adel Mellita, Soteris A. Kalogirou
- [2] Short term wind power forecasting using hybrid intelligent systems, M. Negnevitsky, Member, IEEE, P. Johnson, Student Member, IEE and S. Santoso, Senior Member, IEEE
- [3] Michael Negnevitsky, Artificial intelligence, second edition, ISBN 0-321-20466-2
- [4] Modern power electronics and AC Drives, Bimal K. Bose, ISBN 0-13-016743-6
- [5] Predictive control of an integrated PV-diesel water and power supply system using an artificial neural network, Ali Al-Alawi, Saleh M Al-Alawi, Syed M Islam
- [6] An Evolutionary Artificial Neural Network Time Series Forecasting System, Paulo Cortez, José Machado, José Neves, Universidade do Minho, Portugal
- [7] The nordic stock marked for electrical power
<http://www.nordpool.no>
- [8] Short term power system forecasting using an adaptive neural-fuzzy interference system, Cameron Potter, Michal Negnevitsky
- [9] A fuzzy logic technique for short term load forecasting, Baharudin, Saad, Ibrahim
- [10] Maximum power point tracking controller for PV systems using neural networks, Bahgat, Helwa, Ahmad, Shenawy

- [11] Fundamentals of Grid Connected Photo-Voltaic Power Electronic Converter Design, Master of Science in Energy and Environment NTNU, Svein Erik Evju, June 2007
- [12] A Comprehensive Review for Industrial Applicability of Artificial Neural Networks, June 2003, Magali Meireles, Paulo Almeida, Marcelo Simoes
- [13] Neural Networks: A Comprehensive foundation, 2nd edition New York Perentice-Hall, 1995, S. Haykin
- [14] Control of variable speed pumped storage hydro power plant for increased utilization of wind energy in an isolated grid, Master theses, Jon Are Suul, June 2006
- [15] Adaptive fuzzy logic controller for DC/DC converters, Cetin Elmas, Omer Deperlioglu, Hasan Huseyin Sayan
- [16] Comparison of Fuzzy, PI and Fixed Frequency Sliding Mode Controller For DC-DC Converters, Hamit Erdem, The Baskent university, Turkey
- [17] A more accurate current mode control model, Dr. Ray Ridley, Ridley Engineering
- [18] Intelligent power management strategy of hybrid distributed generation system, Amin Hajizadeh, Masoud Aliakbar Golkar
- [19] Sliding mode control of dc-dc boost converter, Hanifi Guldemir
- [20] Reguleringsteknikk, Trond Andresen, Bjarne Foss, Jens Balchen
- [21] Fundamentals of Power Electronics, Second Edition, Robert Erickson, Dragan Maksimovic
- [22] A robust self-tuning scheme for PI- and PD-type fuzzy controllers, Rajani K. Mudi, Nikhil R. Pal
- [23] Experimental evaluation of a fuzzy controller using a parallel integrator structure for DC-DC converters, Liping Guo, J. Y. Hung, R. M. Nelms
- [24] Comparative evaluation of linear PID and fuzzy control for a boost converter, Liping Guo, John Y. Hung, R. M Nelms
- [25] Statcom and energy storage in grid integration of wind farms, specialization project, Sverre Skalleberg Gjerde

- [26] Petter Støa, SINTEF Energy research, personal communication.
- [27] Sim power systems toolbox in Matlab
- [28] Fuzzy-logic-based self-tuning PI controller for speed control of indirect field-oriented induction motor drive, Mannan, Murata, Tamura, Tsuchiya.
- [29] Self-tuning of PI controllers using fuzzy techniques, Barbu, Ceanga, Caraman
- [30] Fuzzy-logic-based self-tuning PI controller for speed control of indirect field-oriented induction motor drive, Mannan, Murata, Tamura, Tsuchiya.

Appendix A

dSPACE

In this appendix the dSPACE 1104 at ENO is explained.

A.1 Getting started

- Start the computer and login. Login studadm can be used.
- Start Matlab. Choose the right platform RTI1104.
- When connected open a new Simulink file. The dSPACE communication blocks can be found in the Simulink Library. The next section treats the most useful dSPACE blocks. Most of the other blocks can be used like constant, step, ramp, sum, gain, integrator, mux, demux, saturation etc. Scopes and to file/to workspace can not be used. Logging must be done with Control Desk. Remember to use terminators at all loose ends.
- Open configuration parameters and choose a suitable sampling time (simulation time step). The analogue inputs and outputs can handle down to approximately $6 \mu s$. There is a trade off between the complexity of the controller and the achievable sampling time. $20 - 50 \mu s$ is a suitable sampling time.
- Press ctrl+B. The model is now being build and downloaded to the dSPACE. This may take some time. After it is finished the Control

Desk can be opened. Here the simulation can be stopped and started again.

A.2 Input and output blocks

Useful blocks are the analogue inputs (*DS1104MUX_ADC*) and outputs (*DS1104DAC_C*) and the PWM block (*DS1104SL_DSP_PWM3*). The analogue outputs are useful for controlling voltage sources or electronic loads. The analogue inputs can be used to take measurements from voltage and current probes. A ± 10 V signal in the dSPACE board correspond to a ± 1 signal in the Simulink file. The PWM block *DS1104SL_DSP_PWM3* generates PWM signals for three legs. The inputs are the duty cycles and a boolean value for start/stop of PWM signal generation. The PWM signals can be found on the dSPACE board on the Slave I/O PWM contact. PWM signals for A, B and C are found at pin number 7, 8 and 9 respectively. The complementary signals are found at pin number 26, 27 and 28. Many converters demands a constant 5 V on signal. This can be generated with the *DS1104SL_DSP_BIT_OUT_C0*. Choose suitable pin number. The slave PWM contact can also take in signals with the block *DS1104SL_DSP_BIT_IN_C0*. This can for instance be an error signal from the converter supervisory control.

A.3 Logging

The logging of input data can be done using the Control Desk. Execute the following steps.

- Open a new layout.
- Find CaptureSettings in the right menu (under DataAcquisition) and paste it into the layout. Do the same with PlotterArray (also under DataAcquisition).
- Find the .sdf file in the File Selector (at the bottom of the screen). Drag the .sdf file to the ds1104 in the left menu (Platform). A new flag with the .sdf filename now appears next to the File Selector.

- Find the variables that shall be logged in this new flag and drag them into the PlotterArray.
- Double click on the CaptureSettings window and choose the only Capture available. Also choose the variables. For acquisition Simple can be used. Then one has to save manually. Autosave saves the file automatically (overwrites the file). Autaname makes a new file with a unique name for each capture.
- If wanted a trigger can be used to start the capture of data. This can be for example be a step function in the Simulink file.
- If a real time error appears the sampling time might be too small. Increase the sampling time in Configuration parameters in Simulink.

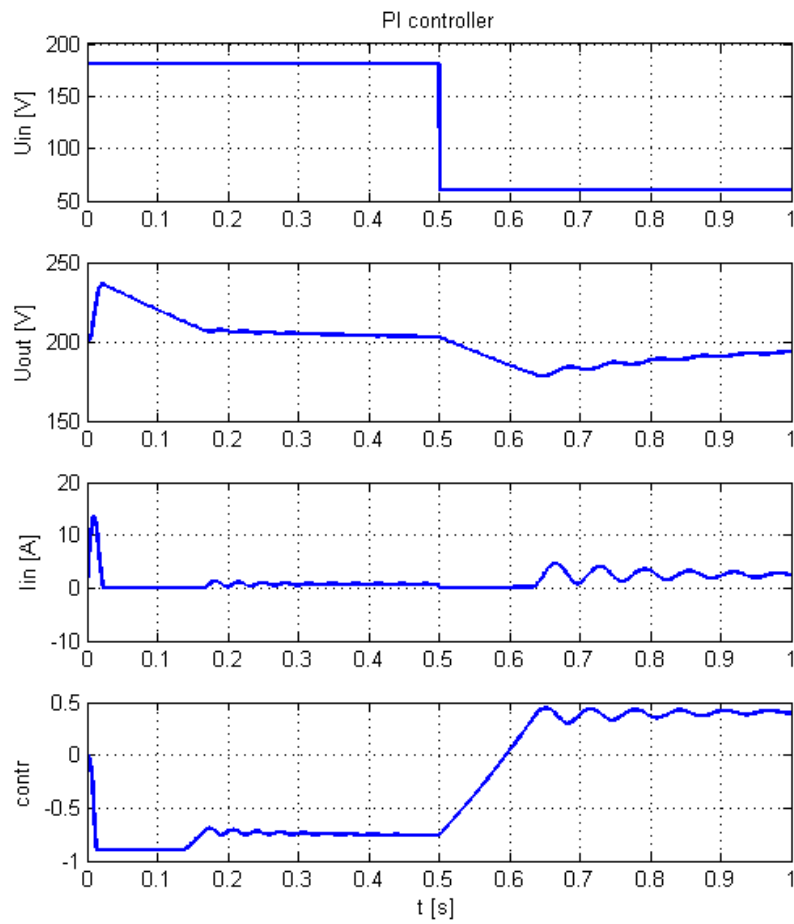
Appendix B

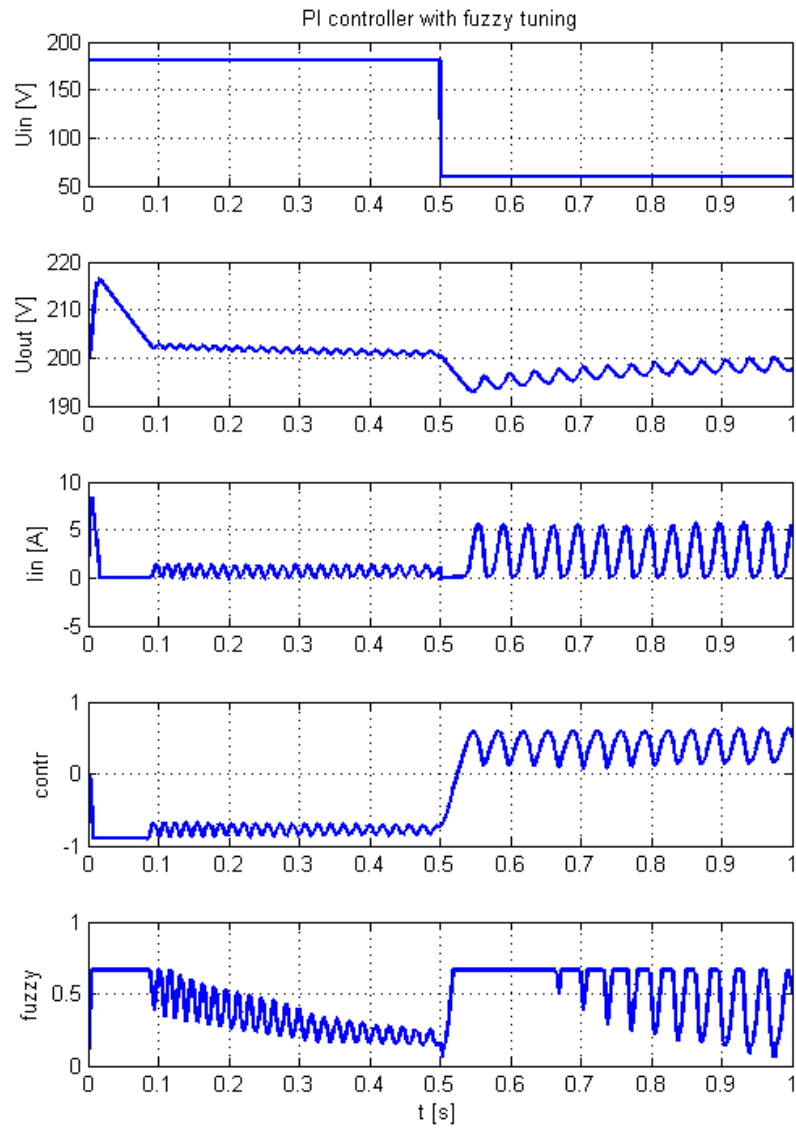
Details of simulations

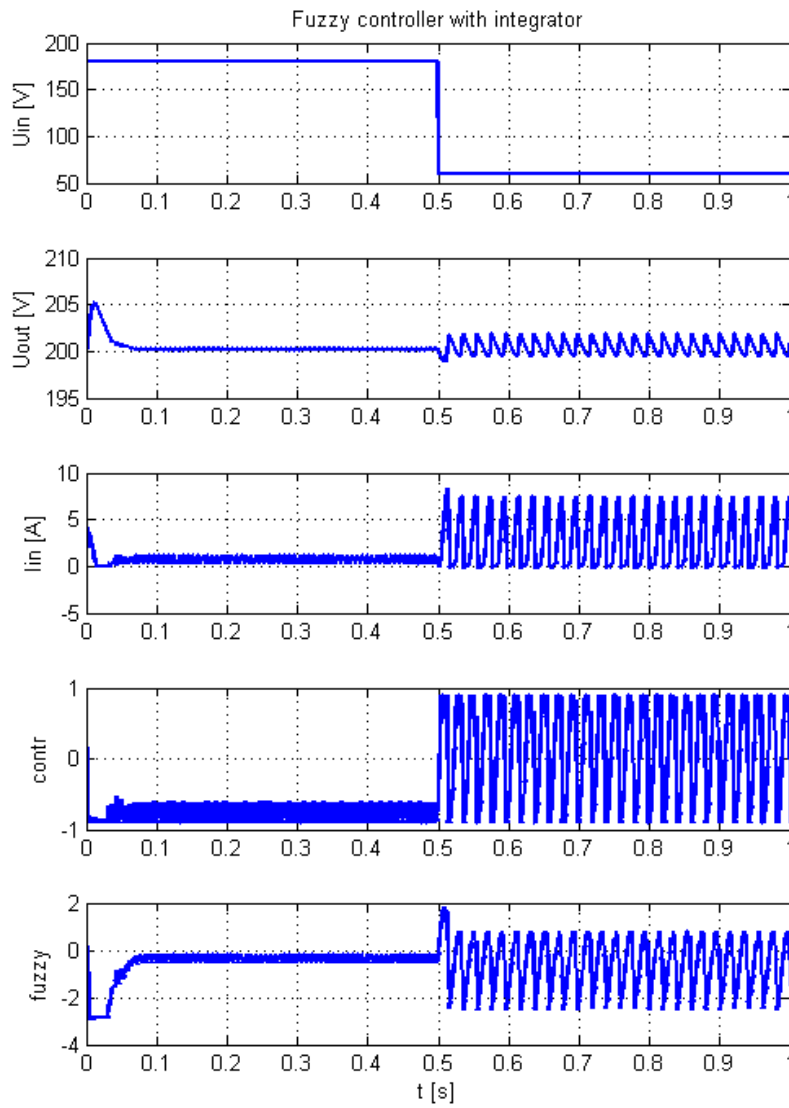
Controller	Abbreviation
PI	Figure B.1
PI_{tuned}	Figure B.2
$FLC_{para-int}$	Figure B.3

Table B.1: Overview of the figures

Table B.1 gives the overview of the plots that are shown in this appendix. In the plots the input voltage, output voltage and input current are shown as well as the output of the controller (input to the PWM block) and the output of the fuzzy controller. For $FLC_{para-int}$ the plot fuzzy is the output of the fuzzy controller after the gain. The oscillations in the output voltage for PI_{tuned} and $FLC_{para-int}$ can be found again in the output of the fuzzy controller. This is a disadvantage with the fuzzy controllers.

Fig. B.1: Details of *PI*

Fig. B.2: Details of PI_{tuned}

Fig. B.3: Details of $FLC_{para-int}$