

Comparison of Simulation Programs for Supercapacitor Modelling Model Creation and Verification

Master of Science Thesis

PATRIK JOHANSSON
BJÖRN ANDERSSON

180p Electrical Engineering
Department of Energy and Environment
Division of Electric Power Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2008

VOLVO
Volvo Technology Corporation

Abstract

To meet increasing demands for fuel efficiency, alternative drivetrains are being developed. Electric hybrid vehicles are one such alternative and an important part of them is their energy storage. To provide this storage the supercapacitor is an interesting component because of its high power density compared to batteries. The increasing use of this component provides the base for this project.

This report consists mainly of three parts. The first part contains information about seven different simulation programs, which has been gathered through brief evaluations of the respective programs. In this part general information about the programs and their capabilities is presented. The analysed programs are Simulink, SimPowerSystems, OrCAD Capture, PSCAD, Saber, PLECS and Dymola. Advantages and disadvantages are summarised for the program alternatives.

The second part of the report presents the supercapacitor modelling. During the initial tests of all simulation programs a basic model is developed. After initial program tests using the basic model, the search for a more advanced model is started. A part of this effort is to conduct several measurements on various supercapacitors in varying temperatures. The measurement results are used to develop an advanced model which includes temperature dependency and is able to accurately model the behaviour of different supercapacitors. After the advanced model development is finished, it is implemented in the five programs which seemed most interesting during the initial round of modelling. The advanced model is given the name DTDSM which stands for Dynamic Temperature Dependent Supercapacitor Model. These programs are Simulink, SimPowerSystems, Saber, PLECS and Dymola.

The third part of the report covers the simulation results. It is found that it is possible to do multi-domain modelling in the programs Saber and Dymola using predefined components. Furthermore it is observed that the electrical modelling available in all program alternatives except for Simulink, can improve the model overview and reduce problems with causality. Regarding simulation performance in the program alternatives it is found that for simulations resulting in comparable accuracy, Saber performs fastest with a time of 5 seconds and Dymola has the longest time with 41 seconds. Moreover the model error of the DTDSM Simulink model seems to be relatively stable during simulation and the magnitude of the error towards the measurement is also low, about 2.1 % of the rated voltage. For the basic model the same error is at about 5 %.

Keywords: DTDSM, supercapacitor, modelling, simulation, Simulink, SimPowerSystems, PSpice, PSCAD, Saber, PLECS, Dymola, ultracapacitor, electrochemical double layer capacitor.

Preface

This Master of Science thesis project is conducted for the Alternative Drivetrains department at Volvo Technology Corporation. The project supervisor is Jens Groot. Some of the work is done at the department of Electrical Engineering at Chalmers. The thesis examiner at Chalmers is Torbjörn Thiringer.

Acknowledgements

We would like to thank our project supervisor Jens Groot for all the help he has provided us with during our work. We would also like to thank Torbjörn Thiringer for valuable comments on the report.

List of Symbols

P	Power	[W]
U, u	Voltage	[V]
I, i	Current	[A]
t	Time	[s]
T	Temperature	[°C]
R	Resistance	[Ω]
L	Inductance	[H]
C	Capacitance	[F]
E	Energy	[J]
Q	Charge	[C]
ΔQ	Charge difference	[C]
ΔV	Voltage difference	[V]
\dot{u}	Voltage derivative	[V/s]
R _s	Series Resistance	[Ω]
R _l	Leakage Resistance	[Ω]

Abbreviations

AC	Alternating Current
ACN	Acetonitrile (CH ₃ CN)
CBR	City Bus Route
CIS	Content Information System
CNLS	Complex Nonlinear Least Squares
CS	Current Source
DAE	Differential Algebraic Equations
DC	Direct Current
DTDSM	Dynamic Temperature Dependent Supercapacitor Model
FACTS	Flexible AC Transmission Systems
GUI	Graphical User Interface
HDL	Hardware Description Language
HEV	Hybrid Electric Vehicle
HIL	Hardware In the Loop
Li-Ion	Lithium Ion
LM	Linear Model
MAST	HDL used by Saber
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NEDC	New European Driving Cycle
NiCd	Nickel Cadmium
NiMH	Nickel Metal Hydride
NLC	Nonlinear Capacitance
ODE	Ordinary Differential Equation
PC	Propylene Carbonate
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
SLPS	Simulink-PSpice
SOC	State Of Charge
VHDL	VHSIC hardware description language
VHDL - AMS	VHDL - Analog and Mixed-Signal
VHSIC	Very-High-Speed Integrated Circuits
VS	Voltage Source
VTEC	Volvo Technology Corporation

Contents

1	Introduction	1
2	The Supercapacitor	3
2.1	Electrodes	3
2.2	Electrolyte	4
2.3	The Dual Layer Technique.....	4
2.4	Additional Information.....	6
3	Program Overview	7
3.1	Simulink	7
3.1.1	Parameter Estimation	8
3.1.2	User Interface	11
3.1.3	Summary	12
3.2	SimPowerSystems	13
3.2.1	Analysis Tools.....	14
3.2.2	Program Help	14
3.2.3	User Interface	15
3.2.4	Calculation Method	15
3.2.5	Summary	16
3.3	OrCAD Capture.....	16
3.3.1	User Interface	17
3.3.2	Different Types of Simulations	18
3.3.3	Calculation Method	18
3.3.4	Issues with Help Function	19
3.3.5	Summary	19
3.4	PSCAD	19
3.4.1	User Interface	19
3.4.2	Calculation Method	21
3.4.3	Summary	22
3.5	Saber.....	22
3.5.1	Supercapacitor Component	23
3.5.2	Parts Gallery	23
3.5.3	Additional Tools.....	24
3.5.4	Co-Simulation	24
3.5.5	User Interface	25
3.5.6	Import and Export	26
3.5.7	Summary	27
3.6	PLECS.....	27
3.6.1	Semiconductor Components	28
3.6.2	Nonlinear Components.....	28
3.6.3	Thermal Modelling.....	29
3.6.4	User Interface	30
3.6.5	Calculation Method	31
3.6.6	Summary	31
3.7	Dymola	32
3.7.1	Performance	33
3.7.2	Thermal Modelling.....	33

3.7.3	Component Library	33
3.7.4	Extended Functionality	34
3.7.5	Steady-State Calculations.....	34
3.7.6	3D Animation.....	35
3.7.7	User Interface	35
3.7.8	Installation Problems.....	37
3.7.9	Summary	38
4	Measurements.....	39
4.1	Screening Test.....	41
4.2	Two Branches Test.....	41
4.3	Varying Voltage Test	42
4.4	Series Resistance Test	42
4.5	Temperature Measurements	43
4.5.1	Test Procedure.....	43
4.6	Further Measurement Analysis	44
4.6.1	Series Resistance Measurements.....	44
4.6.2	Capacitance Temperature Dependency	45
4.6.3	Capacitance Voltage Dependency.....	46
5	Modelling.....	49
5.1	Basic Model.....	49
5.1.1	Simulink Model.....	50
5.1.2	Component Value Tuning	51
5.1.3	Component Value Determination.....	51
5.2	Program Alternatives.....	58
5.2.1	Nonlinear Capacitance	58
5.2.2	SimPowerSystems	59
5.2.3	OrCAD Capture.....	62
5.2.4	PSCAD	64
5.2.5	Saber.....	66
5.2.6	PLECS.....	68
5.2.7	Dymola	70
5.3	Search for a New Model Structure.....	71
5.3.1	Two Branches Model	72
5.3.2	DTDSM.....	75
5.4	Program Tests using DTDSM.....	81
5.4.1	SimPowerSystems	81
5.4.2	Saber.....	83
5.4.3	PLECS.....	83
5.4.4	Dymola	84
6	Simulation Results.....	87
6.1	Basic Model.....	87
6.1.1	Program Alternatives.....	89
6.2	Two Branches Model.	89
6.3	DTDSM.....	91
6.4	Estimation of Model Errors.....	94
6.4.1	Analysis of Errors for Basic Model.....	94
6.4.2	Analysis of Errors for DTDSM.....	99

7	Performance and Cost	103
7.1	Simulation Performance	103
7.2	Program Cost.....	104
8	Conclusions	107
9	Discussion	108
10	Appendix A – Controlled current source.....	109
10.1	Simulink	109
10.2	OrCAD Capture.....	109
10.3	Saber.....	110
10.4	PLECS	111
10.5	Controlled Current Source	112
11	Appendix B – DTDSM Figures	113
11.1	Simulink	113
11.2	SimPowerSystems	117
12	References	120

1 Introduction

To improve fuel efficiency of the vehicles available today, one area which is being investigated is that of alternative drivetrains. When designing a drivetrain for an HEV, an important part of the system is the electric energy storage. One alternative used to provide this storage is the supercapacitor. Because of its advantage in power density compared to batteries, it is starting to become more frequently used as energy storage for certain hybrid systems. When designing a new system it is necessary to create a model of that system in order to test if it will work using computer simulations. Since the supercapacitor is still a rather new component, the development of appropriate models is still a subject which is being investigated. Another concern when designing a hybrid system or creating a component model is which program is supposed to be used for this model development. By using different programs, the development process is affected and both advantages and disadvantages can be found for the different alternatives.

Regarding the development of supercapacitor models, some work has already been done. Some have tried to create simple, yet reasonably accurate models [1]. Others have designed more complicated models, which better resemble the actual supercapacitor behaviour [3]. However those models are also more difficult to tune when they are supposed to be used for different supercapacitors. When searching for previous work in the field of supercapacitor modelling, something that appears to be less frequently implemented in models is the temperature dependency of the supercapacitor.

When previous program comparisons are investigated it is found that no study has been made on the combination of programs investigated in this project. Some comparisons have been made between a few of the programs but not with the same objective as this project. One example of such a study is [2], however there the purpose of the comparison is to find a replacement for a specific program and not to test the general abilities of the programs. Since program development is a field with rapid changes, the validity of previous program comparisons quickly reduces with time.

The rapid program development is one reason that the main objective of this project is to analyse several different simulation programs. The secondary goal is to create an accurate model of a supercapacitor. One purpose of this model is to be able to use it as a test model for the various programs. Before the program alternatives are analysed, a basic model of a supercapacitor is to be developed in Simulink. Moreover an objective is to implement this model in each of the analysed programs. The analysed programs are Simulink, SimPowerSystems, OrCAD Capture, PSCAD, Saber, PLECS and Dymola. To reach a more advanced supercapacitor model, measurements are to be conducted and a new model structure is to be developed. An aim for this advanced model is to accurately model the supercapacitor behaviour in varying temperatures. In addition to the creation of the more advanced model, one aim is to use this model in further program analysis of the most interesting program alternatives. A goal with the program analysis is to find advantages and disadvantages regarding their capabilities as modelling and simulation programs.

An overview of the contents of the various chapters is presented as an introduction to the contents of the report.

Chapter 2 introduces the supercapacitor by providing a theoretical background to its functionality.

Chapter 3 contains the program presentations for all the tested programs. The features of all the alternatives as well as their advantages and disadvantages are mentioned. A figure displaying the user interface together with notes about the respective environments is included.

Chapter 4 describes the various measurements that are conducted in order to discover the component properties. Both a previously conducted measurement and the measurements done during this project are covered.

Chapter 5 deals with the modelling of a supercapacitor. The process of designing first a basic model and then redesigning it to become more advanced is described. The behavioural properties of the model are explained. Model implementation in the program alternatives is covered in separate sections which include figures of the resulting models and notes concerning differences in the modelling processes.

Chapter 6 displays results acquired during simulations in the program alternatives. Results are presented and analysed for both the basic model and the advanced model. Since the amount of simulation results is large only the most interesting results are displayed. This chapter contains information concerning the differences in simulation results where such can be found.

Chapter 7 presents some results regarding the performance of the programs, accuracy and cost.

Chapter 8 and 9 cover the project conclusions and a discussion on the completed work along with considerations for the future.

2 The Supercapacitor

The supercapacitor is an energy storage device that has one specific property that makes it an interesting component in some applications. This property is its high power density that enables it to handle fast fluctuations in energy level. When comparing supercapacitors with batteries the main difference is in energy and power density. Supercapacitors have a significantly lower energy density than the batteries but on the other hand they have a higher power density compared with batteries. This can be seen in Figure 1 which is a Ragone plot of different electric energy storages. The area which represents fuel cells is valid for systems including the hydrogen storage. The battery area is an average of the most common battery types such as Li-Ion and NiMH. Another difference is the principles upon which they are built. Batteries use chemical processes to store energy which can then be released as electricity. Supercapacitors on the other hand store energy through charge separation. This means that the need for chemicals is reduced, which enables a longer life length of supercapacitors.

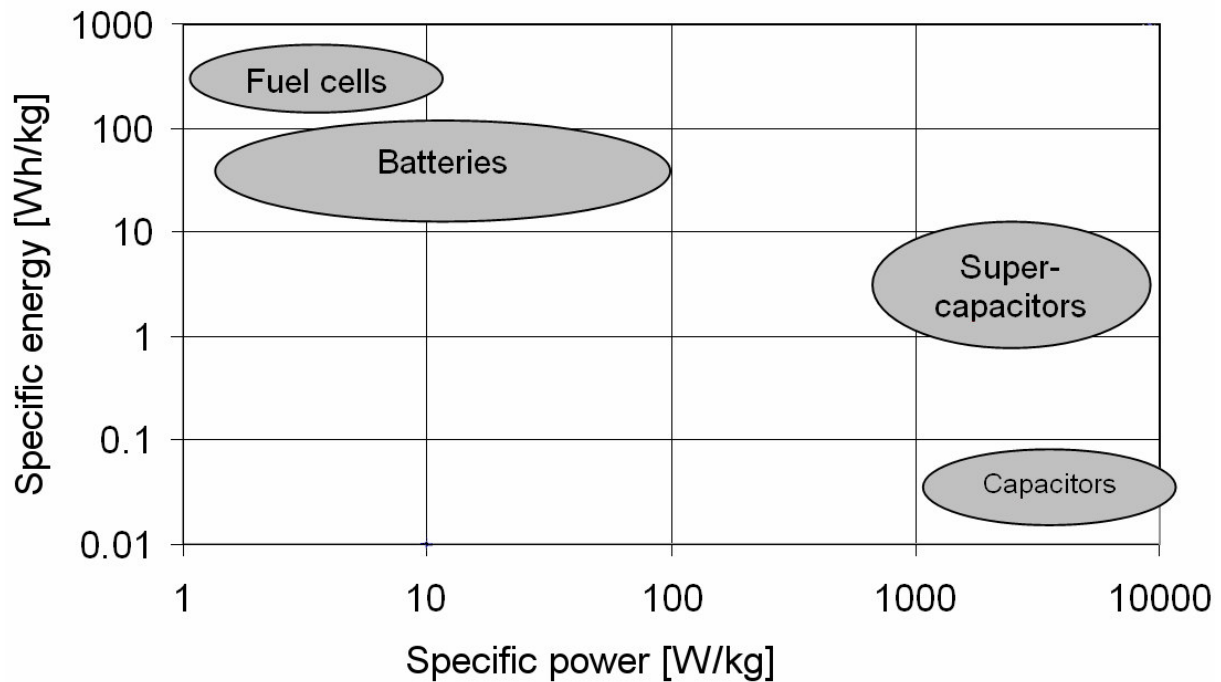


Figure 1: Ragone plot displaying the energy and power properties of various technologies.

The physics behind the electrolytic capacitors were first described by Hermann von Helmholtz back in 1853. He concluded that the interaction between a conductor and the electrolyte inside a capacitor is determined by electrostatic relations and that there are no chemical reactions involved in the process [3]. The supercapacitor is based on this theory and it is an improved version of the original capacitor design.

2.1 Electrodes

In a traditional electrostatic capacitor the electrodes consist of a thin surface on which the charges are gathered. In a supercapacitor the electrodes consist of a porous micro structure

that can be made of carbon material and around it the electrolyte is deposited. This structure gives a significantly larger specific surface area than the electrostatic capacitors have. To separate anode and cathode a thin isolation membrane is located between the electrodes of the supercapacitor. Even though the membrane separates the electrode surface it still enables ions to move freely.

2.2 Electrolyte

The main difference between the supercapacitor and the ordinary electrostatic capacitor is that the electrolyte in the supercapacitor contains free charges in the form of ions. The ordinary capacitor does not have this type of free charges. There are two main electrolytes today on the market, ACN and PC [4]. The properties of the electrolyte set the rated voltage for the capacitors. The rated voltage must be lower than the oxidation voltage for the electrolyte. If the oxidation voltage is reached a chemical process is started in the electrolyte that creates gases from the electrolyte. Today electrolytes are available that give a rated voltage of up to 2.8 V. One important property of the electrolyte is that it must be able to dissolve some types of salt which provide the free ions in the capacitor. Another desired property is that the ions have a high mobility in the electrolyte. This is because the mobility mostly determines the series resistance of the supercapacitor. High mobility gives low series resistance and vice versa. The other contribution to the series resistance comes from the resistance in the conductors but that is relatively low compared to the contribution from the electrolyte. In some applications a third important property can be added to the electrolyte. This property is the temperature stability of the ion mobility. This property differs between the two electrolytes that are used today. The ACN electrolyte gives a more stable series resistance during temperature changes compared to PC, especially at low temperatures. The drawback of using ACN is that this substance has an environmental disadvantage. If the capacitor is over charged to the oxidation voltage, the resulting gases contain cyanide which is a toxic substance and for this reason it is not allowed to be used in some nations. [5]

Inside the supercapacitor there are mainly two physical laws that determine the behaviour of the ions, diffusion and electrostatic relation. When the supercapacitor has been completely discharged the ions in the electrolyte become evenly distributed due to the diffusion. As soon as the supercapacitor is charged the ions are attracted by the electric field which is formed between the electrodes. Because of this field a separation of ions is started. The self discharge of the supercapacitor is mainly caused by the diffusion. [6]

2.3 The Dual Layer Technique

When the supercapacitor is charged, the electrons at the cathode attract positive ions and on the anode the vacancies of electrons attract negative ions in order to locally obtain a charged balance. This attraction of ions leads to a capacitance being formed between the ions and the surface of the electrode. The name dual layer comes from the two layers of ions at each electrode. The layer closest to the electrode acts as a dielectric and the layer outside the first layer holds the charges [7]. This occurs at both electrodes in the supercapacitor and the total capacitance consists of these two capacitances connected in series. When charges attract ions, they are gathered at the electrode surface. This is shown in Figure 2 which is an ideal case. In the picture that describes the charged state, all the ions are at the respective electrodes. In reality the diffusion causes some ions to be located at varying distances around the electrodes

[5]. The intensity of the electric field determines the concentration of ions at the electrodes, which means that an increased voltage results in an increased capacitance.

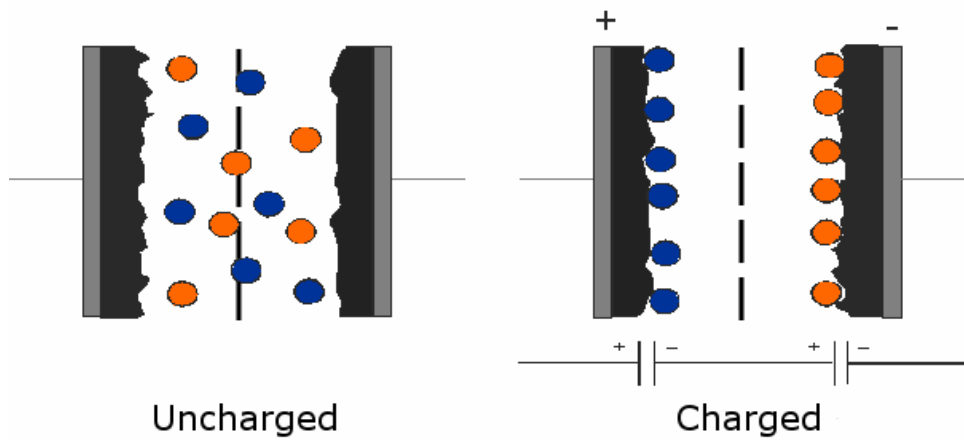


Figure 2: Supercapacitor structure, red circles describe positive ions and blue describes negative ions.

Outside the carbon surface there are two layers, the inner layer consists of non-conducting molecules from the electrolyte. The outer layer consists of ions surrounded by electrolyte molecules. The distance between the carbon surface and the ions is similar to the distance between the electrodes that exist in an ordinary capacitor. Figure 3 shows an illustration of this process near the cathode side of the supercapacitor, a similar process occurs at the anode. This means that the capacitance is dependent on the ion concentration at the electrodes and the carbon surface area. Since the surface area is extremely large and the distance between the surface and ions is very short, this combination results in a large capacitance [5]. This relation is explained by

$$C = \frac{\epsilon A}{d} \tag{1}$$

where A stands for surface area, d stands for the distance between the charges and ϵ is the dielectricity constant.

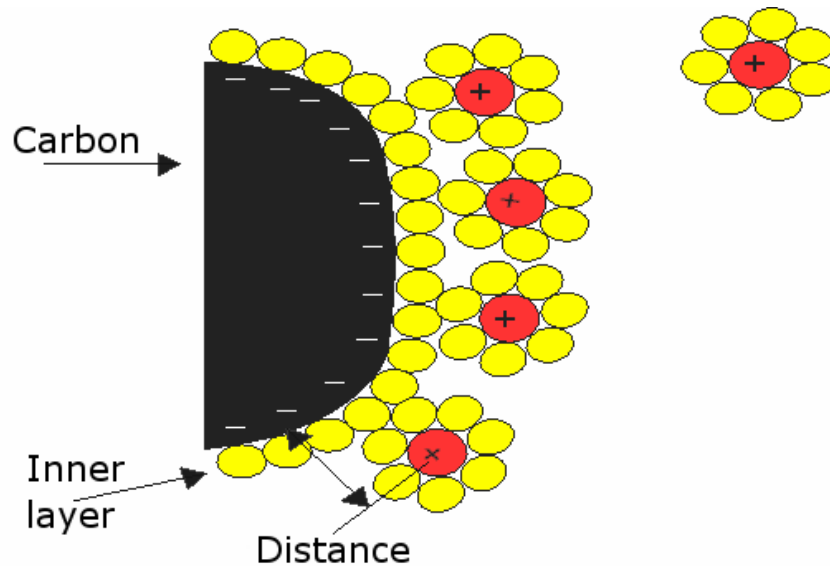


Figure 3: Process that takes place near the carbon surface of a supercapacitor. The yellow circles describe the electrolyte molecules and the red describe the positive ions.

2.4 Additional Information

A benefit of using supercapacitors is that they have a long lifetime, about 1 000 000 charging cycles compared to some battery types that last for about 1 000 cycles [5]. The number of cycles for batteries is valid for variations of SOC of about 80 %. Another benefit of the supercapacitor is that the number of charging cycles is not greatly affected by the variation of SOC compared to batteries. Batteries have a significantly shorter lifetime when they are used in cycles with large SOC variation compared with low SOC variation cycles [8].

Today the differences in energy and power density between these components can be used as an advantage by combining them in a power system where they can complement each other. One interesting application is the area of hybrid vehicles, where the regenerative braking creates high transients in currents and power demand during short time periods. Other applications that supercapacitors are used for are different types of power boost applications such as start of electrical drives and pulsating lasers. They can also be used in UPS systems for shorter time periods. The main reason why the supercapacitors are not used in more applications today is the cost. The price for a supercapacitor from Maxwell is today around 1 cent per farad or \$ 2.85 per kJ and their price will probably drop more in the future. [9]

Supercapacitors are still relatively new on the market and there appears to still be room for improvements in their capability. One example of possible future improvements are supercapacitors based on new nanotube structures, which today are on an experimental stage. They are made using carbon that is arranged in a symmetric way that creates a significantly larger surface area. The carbon material that is used today has an irregular size and shape of the structure which reduces the effective area [8]. With the new technology the supercapacitors can reach a higher energy density compared to the supercapacitors that are on the market today and they might even reach a level comparable with the energy density of some batteries.

3 Program Overview

Since the target of this study is to compare how well the tested programs perform when simulating a supercapacitor, all comparisons are made from that perspective. In this chapter all the different programs are presented and analysed. Each program is given a short introduction. The usability is discussed, including how easy it is to understand, how fast the development process is and how the interface works. How the programs make calculations is covered. The mechanisms to import and export data as well as various ways to present results are investigated. For each program the most important advantages and disadvantages are summarized.

Some properties are equal in all the programs, so to avoid unnecessary repetition these properties are presented here. The first property is about the process of model creation. All programs use a graphical interface to build models in which different components are connected together using wires. In Simulink the models are created using only mathematical components and the wires only carry signal values. All the other programs have the ability to use electrical components to create a model structure and there the wires can carry both electrical signals and ordinary signal values. All programs have different kinds of libraries which hold the components available for model design. The organization and function of these libraries is further described in each program section.

Much of the information regarding the programs has been found in the Help functions of the respective programs. Other information sources are the web pages for the programs, books and scientific reports.

3.1 Simulink

Simulink is a software package that is a part of MATLAB and the tested version of Simulink is 6.5. The version of MATLAB that has been tested is 7.3.0 (R2006b). Simulink is supposed to simplify the modelling and analysis of dynamic systems compared to performing the same operations using only code in MATLAB programming. When studying other simulation program alternatives it becomes clear that the MATLAB and Simulink environment seem to be somewhat of a standard when conducting simulations. Many of the other programs actually have interfaces to Simulink which make integration possible. Some programs enable co-simulation with Simulink while others only have specific interfaces for import and export to Simulink. This integration possibility is one positive property of Simulink and it also shows that the program is widely spread. The close integration with MATLAB makes it possible to use its script language for advanced calculations as a part of the Simulink models.

The basic building blocks available in Simulink are mostly various mathematical blocks and signal handling blocks. This means that more complex domain specific components must be modelled using these basic blocks, since no such components are available in the standard library. There is however a possibility to increase the number of building blocks available by adding new toolboxes to Simulink. There are toolboxes available that enable modelling in both the electrical domain and the mechanical domain. Some examples of toolboxes are SimPowerSystems, SimDriveline and SimHydraulics. The last two of these toolboxes require

the toolbox Simscape to function. Simscape provides basic abilities to model multi-domain physical systems and can receive extended functionality through combination with more specialised toolboxes. If Simscape is installed on a computer it is possible to simulate models created using SimDriveline, SimHydraulics or SimMechanics without having licenses for those products installed. By combining toolboxes, modelling speed can be enhanced and entire systems can be simulated more easily and with better overview.

The MathWorks also offers other types of toolboxes and applications that increase the functionality of MATLAB and Simulink. Examples of such components are Symbolic Math Toolbox, Statistics Toolbox, Robust Control Toolbox, Signal Processing Toolbox, Real-Time Workshop and Control System Toolbox.

One advantage with the signal flow modelling used in Simulink is that it can be applied on many systems from different domains, as long as the equations describing the system behaviour are known and they can be expressed by state space equations. By using fundamental equations to describe the system behaviour it is possible to be certain of how all the parts function. That can be an advantage over other simulation programs, where the theoretical backgrounds to some of the modelling components are not known. But since most other programs include the ability to create mathematical models as well, this can not be seen as a significant advantage.

The fact that all the blocks used in Simulink are mathematical can also be a disadvantage. When creating an electrical circuit or some other type of model, the causality of the system has to be decided. It is not possible to build a circuit model that is designed to have a current as input and then change the input to a voltage. To be able to use a different kind of input, the entire model of the circuit has to be modified so the electrical equations are correct for that implementation. This means that two separate models must be created in order to conduct tests using first current and then voltage as input. With large systems containing many complicated mathematical relations the process of inverting the model is difficult. This is a significant disadvantage with Simulink which results in problems during model modifications, increased modelling time and lack of overview. The mathematical modelling used in Simulink is its most important property, since it is what separates it from the other tested program alternatives which are more targeted at direct electrical modelling.

3.1.1 Parameter Estimation

One additional component which can be added to Simulink is called Parameter Estimation 1.2.1. This tool can be used to find parameter values for model components that result in a model behaviour that resembles measurement results better. Complete models as well as specific sections or subsystems can be used for estimations. The tool is easy to use and has a separate interface that is used for test setup. Different input and output signals can be chosen, in order to find parameter values using several sets of measurement data. The parameters that should be estimated are also chosen and initial values as well as min and max values can be set. If the model has several outputs it is possible to set different weights for them. If an output is noisy the weight can be set low and if an output is important for the model accuracy the weight can be set higher. The results of the estimations can be illustrated using different types of plots such as cost function, parameter sensitivity and measurement compared with

simulation which are all predefined in the tool. In Figure 4 the program interface while conducting settings for the estimated parameters is shown.

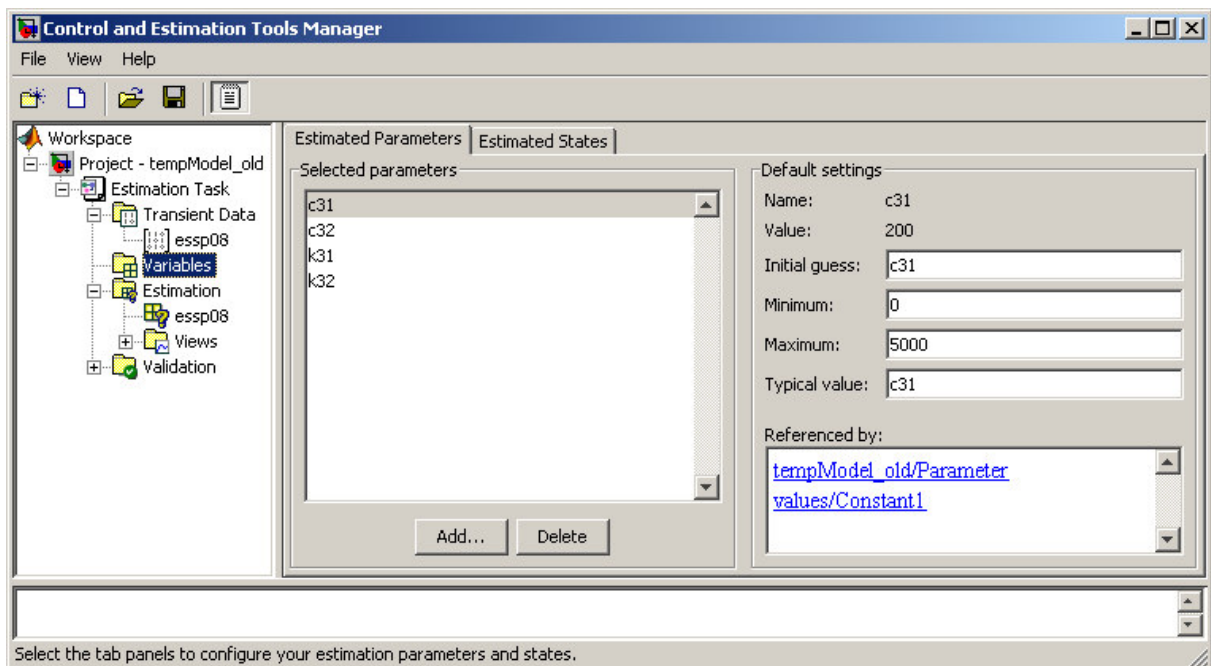


Figure 4: Program interface for Parameter Estimation.

When some iteration is completed the new parameter values can be shown as well as the plots that are selected during the setup of the test. This provides a good overview over how the model accuracy has improved and in what way the parameters are tested. These plots can either be shown during the estimation, to show its progress or only after completion if the end results are most interesting. To further analyse the behaviour of the model using the new estimated parameters it is possible to validate it with a different set of data than that used in the estimation process.

In Parameter Estimation it is possible to pre-process the input data before using it in the estimations. This enables elimination of outliers, de-trending and filtering of data. If the initial conditions of a model are unknown they can be estimated in the same way as the parameters. This can be useful when several parameter estimations using different data sets are run, because then the initial conditions might vary.

To be able to use the Parameter Estimation program with Simulink, the Optimization Toolbox must be installed. This is because the program uses optimization methods which are included in that toolbox. The Optimization Toolbox contains tools and several algorithms for use in optimization problems. The algorithms can solve constrained and unconstrained continuous and discrete problems. It is possible to use both the command line and a separate GUI for the optimization problems. Using the GUI it is possible to automatically generate M-code which can be useful for automating tasks. Since all the functions are written with the MATLAB language it is possible to inspect the algorithms and modify them. Constrained and unconstrained nonlinear as well as multi-objective optimization can be handled. Nonlinear

least-squares problems can be solved which can be used for data fitting.

Adaptive Lookup Table

An adaptive lookup table builder can be used to create tables for use in models which have a time-varying behaviour. This is done by taking the input and output measurements of a plant's behaviour and then using them to update the content of the lookup table used to describe the plant behaviour. The ability to model time-varying behaviour can be used for systems that are affected by wear, environmental conditions and manufacturing tolerances.

Optimizations

During tests with Parameter Estimation, it is found that the process can be especially time consuming for certain models. In order to reduce the required time, the parameter boundaries and the model composition can be altered. The tool has a separate section called Optimization Options which include several setting alternatives. Four different optimization algorithms are available which result in varying computational speed and accuracy. The optimization termination can also be determined by specifying for example maximum number of iterations or lowest allowed change in parameter value. The standard simulation options are also available in the tool which enables setting of solver type, maximum step size and similar properties. Another way to improve the estimation performance is to use the Accelerator simulation mode instead of the Normal mode in the model. The Accelerator mode uses compiled C code to run the simulations instead of interpreted code which is the case in Normal mode. However the Accelerator mode can not be used if the model contains algebraic loops. If the model contains MATLAB embedded function blocks they must either be removed or replaced with Fcn blocks. The Fcn block only allows direct mathematical expressions and not any other programming such as for-loops which can be used in the embedded function block.

Test of Function

To test the function of Parameter Estimation it is used to estimate the parameters in the temperature modelling part of the final supercapacitor model which is described in Section 5.3.2. These estimations improve the model accuracy significantly. The test setup is easy and can be done fast. By running an initial estimation with less precise optimization settings and large parameter intervals it is possible to find good approximations in a short time. Using the parameter values found in another test with better settings enables more accurate estimation of the values. The various settings affect the estimation performance greatly and to find appropriate values requires some trials, in order to reach a good compromise with speed and accuracy.

The temperature model improvement as a result of the parameter estimation can be seen in Figure 5 where the measured temperature is plotted together with the model result using both the old parameter values and the new estimated values.

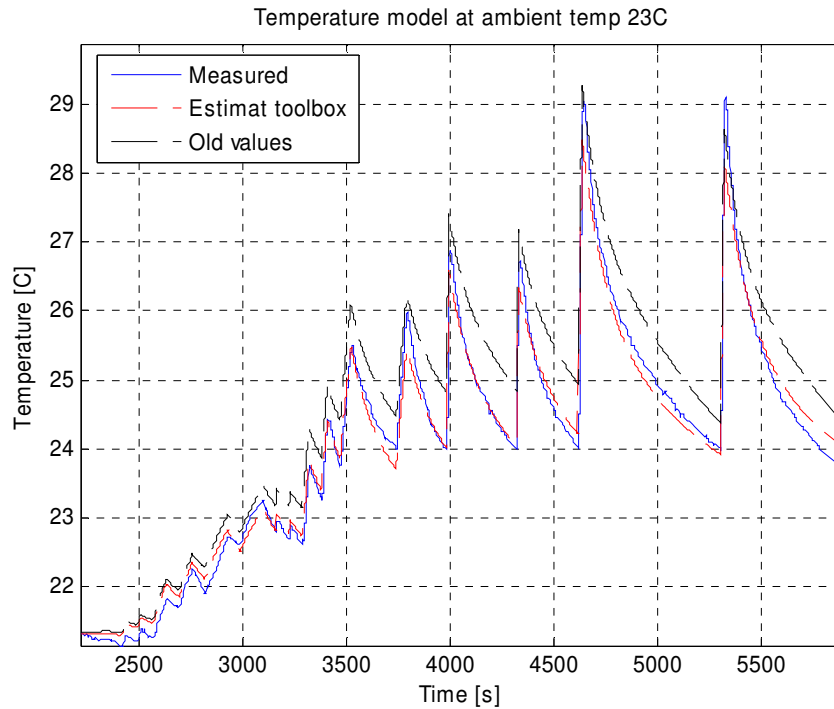


Figure 5: Results of parameter estimation.

The improvement of the model accuracy can be measured using an error value which computes the difference between the measured curve and the model curve. The expression used to do this calculation will be shown in (22). The error value for the old parameters is 0.3578 and the error value for the new parameters is 0.054 which indicates a significant improvement. The used parameter values are presented in Table 1 where the parameters c and k are constants that are used to tune the dynamic behaviour of the temperature variations. The part of the model that uses these parameters results in a time constant for the temperature variations. The numbers 1 and 2 indicate that the model contains two time constants and therefore two parts that use the parameters c and k . The model is described further in Section 5.3.2 under the heading Thermal Modelling.

Table 1: Parameter values before and after estimation process.

Parameters	$c1$	$c2$	$k1$	$k2$
Old value	300	400	0.3	5
New value	500	273.81	0.359	2.2372

3.1.2 User Interface

All components in Simulink are gathered in a large library which is arranged in sub groups according to function. To mention some examples there are groups for input, output, logic and math. Finding the needed part is mostly easily accomplished, especially after some initial searches through the contents of the library. The base in all models is that one or several signals are used as input. They are then manipulated using the different blocks, to finally

result in one or several output signals. This type of development results in reduced time usage when creating complex models and also enables better overview and an easier understanding of large systems, compared to creating models using only code.

One feature that can be useful while modelling in Simulink is the ability to combine several signals into one signal line. This saves space and can improve the overview of the model as well as simplify different connection points. After being combined the signals can then be separated where the individual signals are needed again. This can also be used when several input signals are supposed to be sent into a system to view how the results vary with the input. By multiplexing the signals to one signal line it is not necessary to have several copies of the system to be able to test all the inputs simultaneously. Instead they can all be sent to one system model, each input then generates a separate output signal. The user interface of Simulink is shown in Figure 6.

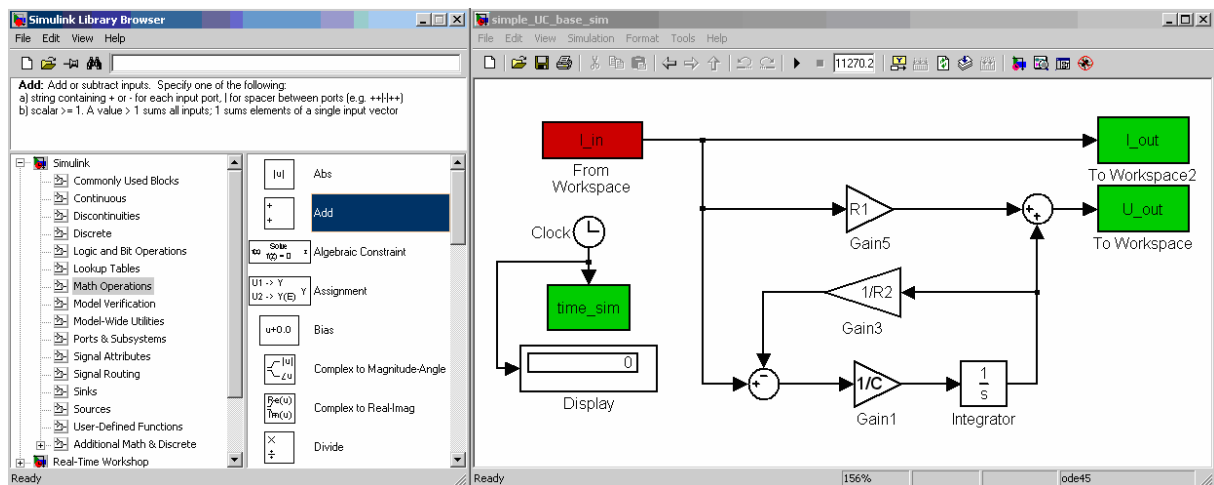


Figure 6: User interface for Simulink.

When developing a model some parts of it are often recycled several times. The ability to create subsystems that contain several different components can be helpful in these situations. Subsystems can be masked so that the important parameter settings can be made in a dialog box instead of in the various components in the subsystem. Some advantages of masking subsystems are that unintended modification can be avoided, the standard icon can be replaced with an icon specific to the subsystem function and block parameter dialogs can be gathered to one single interface. After creating a masked subsystem it is also possible to save the custom block in a separate library for use in future model creation.

3.1.3 Summary

Here some of the advantages and disadvantages of Simulink are summarised.

- + Large user base.
- + Direct access to MATLAB environment.
- + Well structured Help with relevant information.
- + Extensive control library allows easy implementation of any control algorithm.

- + Possibility to add functionality using extra components.
- + Ability to use multiplexed signals.
- + Model calibration can be added, which enables parameter estimation.

- Reduced simulation performance compared to models designed using only code.
- Only mathematical modelling.
- Problems with causality.
- Model overview is not as good as in programs that enable electrical modelling.

3.2 SimPowerSystems

The version of SimPowerSystems that has been analysed is 4.5. SimPowerSystems is a toolbox that can be added to the component library in Simulink to model electrical circuits using standard symbols. The block libraries and simulation methods in SimPowerSystems were developed by TransÉnergie Technologies Inc. of Montreal [10]. The toolbox contains standard circuit components such as resistors, capacitors, transformers and power electronics components. The library also contains more advanced components such as electric drives and FACTS components. In total there are more than 150 blocks in the library distributed in the eight libraries. The electric drives include both DC and AC drives such as four-quadrant chopper DC drive and brushless DC motor drive. To be able to interface the electrical systems mechanically, SimPowerSystems includes a simple mechanical shaft component. A component that is new in the 4.5 version of SimPowerSystems is the battery block. It can model most popular battery types, such as NiMH, Lead-Acid, Li-Ion, and NiCd. This is done using a generic model in which the parameters are altered to represent the differences in discharge characteristics. The model also allows modification of these parameters in order to create a user defined battery model. Examples of other included components are PWM generator, fourier analysis and different electrical machines.

In the program description for Simulink the problem concerning the causality of an implemented model is mentioned. In SimPowerSystems that problem does not exist when creating electrical circuits and that is a major advantage. The input to the modelled circuit can be switched from voltage source to current source without having to alter the modelled circuit, which saves time compared to modelling in Simulink. The ability to combine the electrical circuits built using the components in the SimPowerSystems library, with the MATLAB & Simulink environment, is a considerable advantage. This is especially advantageous at VTEC where a majority of the models are developed in Simulink. The combination enables a better overview of the electrical part of the model while still being able to control the circuit using Simulink components. The direct access to all data in the MATLAB workspace saves time compared to working in programs where all such data must be imported from external data files. Access to the data processing capabilities in MATLAB is useful especially when analysing simulation results.

In general when looking closer at the more advanced components in the different libraries it is noted that they model a great deal of the behavioural properties of the real components. One example of a component where there are some properties that are not included in the model is the MOSFET transistor. The gate is controlled by a switch signal instead of a specific voltage level. This means that if the transistor switching should be controlled by a voltage level a

separate measurement circuit must be created and connected. One thing not included in the model is the effect of the capacitances between gate and source and between gate and drain. This means that the dynamic behaviour during the switching is not included. However in many circuits this behaviour is not important for the overall function, which means that the available model is sufficient. The accuracy of the included components is something that must be considered during all model development work, to make sure none of the important behaviours are neglected.

One disadvantage with the model design process in SimPowerSystems is that special attention is needed when using capacitor or inductor elements in the circuits. Connecting a capacitor in parallel with a voltage source or an inductance in series with a current source creates errors when simulation is attempted. To fix the problem with the capacitor a small resistance should be connected in series between the voltage source and the capacitors. To solve the problem with the inductors a large resistance should be connected in parallel with the inductors. A similar problem affects diodes as well. When they are connected to a current or voltage source the snubber resistance must be used for the component to function.

When connecting a load to circuits it is possible to use a few different predefined components. One is a circuit consisting of a resistance, an inductance and a capacitance in series and the other has the three parts connected in parallel. For these two components it is possible to configure the properties in two ways. The individual values can be set in Ohm, Henry, and Farad or the load can be characterized by its power usage.

3.2.1 Analysis Tools

In SimPowerSystems there is a specific GUI for analysis of circuits and systems called powergui. It enables display and modification of initial state values, impedance spectroscopy measurements and display of steady-state voltage and current as well as state variables. It can display impedance versus frequency plots and perform FFT analysis of the simulation results. It can generate a report containing steady-state values of the model parameters. The setting of initial values includes for example the voltage over a capacitance and the current through an inductance. If the Control System Toolbox is installed, powergui can generate a state-space model of the system.

3.2.2 Program Help

The help sections for the respective components contain well structured and detailed information. What aspects are modelled and how this modelling is done is described using text and some other informative addition. Sometimes this is provided in the form of equations and sometimes a functional schematic, Simulink schematic or SimPowerSystems schematic is used. For some components, additional illustrations describing the model behaviour compared to real components are included. Almost all components have some sort of example that illustrates the model function and how it can be used. Associated assumptions and limitations are stated for some models where this is of interest. This can be useful information if there are some model aspects that are especially important for a specific simulation. If some behavioural property is not included in a model it is always possible to either create an entirely new model using the available components or add to an existing component. The modified model can then be saved in a user library as a new component. All components that have a dialog box include a list in the help section about the available parameters and various

possible settings. Some of the more advanced components have references to scientific publications describing the model function or some subject of interest to its creation.

3.2.3 User Interface

When creating a model using SimPowerSystems the same workspace that is used for model creation in Simulink is used. The main difference is that there now exist many more components to choose from in the library. All the new components are found in the sub library called SimPowerSystems. Most of these components are different from the ordinary Simulink components in that they model real electrical components instead of just being mathematical components. The availability of these new components enables more direct circuit design which improves the overview of the model; because it actually looks similar to the circuit it is modelling.

When building a model using SimPowerSystems some attention is needed when interfacing the system with ordinary Simulink components. Signals generated by blocks in the SimPowerSystems library can have two types of connection ports, electrical or normal signal ports. Specific measurement blocks can be used as conversion from the electrical system to Simulink and inversely a controlled voltage source is an example of a block that converts from Simulink to SimPowerSystems. By using these conversion blocks it is possible to connect electrical models with other models in Simulink. The user interface of SimPowerSystems is shown in Figure 7.

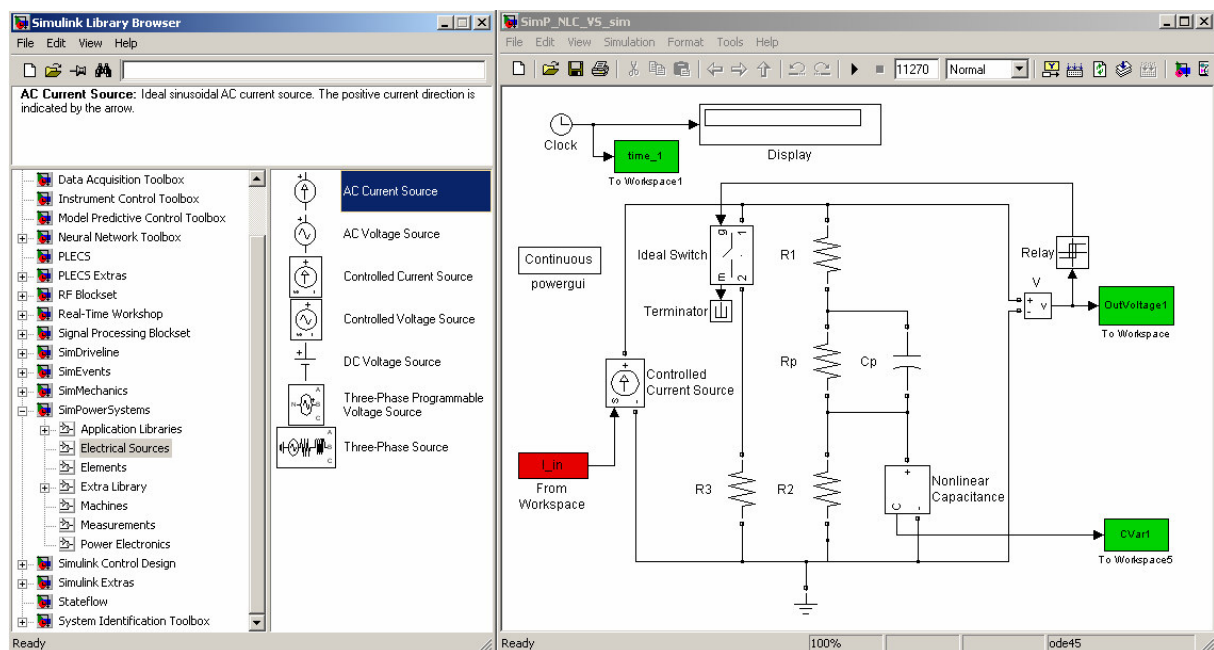


Figure 7: User interface for SimPowerSystems.

3.2.4 Calculation Method

When a design is made some regular Simulink components are linked together with the electrical components available in SimPowerSystems. To get a mathematical description of the modelled system, the electrical part of the system is first described by a state space model

that contains the component values of the circuit. This state space model can be interpreted by Simulink so an equivalent model can be created. This model can then be calculated by MATLAB using different kinds of ODE solvers such as ODE 45 which is the standard method [11]. The continuous simulation uses variable-step integration which results in a smaller number of steps than a fixed-step method of comparable accuracy. Although the accurate simulation results are desirable, larger systems can result in too long simulation times if a continuous simulation is conducted.

In addition to using continuous simulation of the power system it is also possible to use discretization and phasor simulation. Discretization uses fixed time-step integration which can increase computational performance for larger systems if an appropriate step size is chosen. Phasor simulation uses algebraic equations of fixed frequency instead of differential equations, which enables transient stability studies. Since the phasor simulation only studies changes in magnitude and phase of the voltages and currents it is not necessary to solve all differential equations that are usually solved during continuous simulation. The phasor simulation only requires solving simpler algebraic equations for the phasors. This enables a significant increase in simulation performance however the results are only valid for the chosen frequency.

3.2.5 Summary

Here some of the advantages and disadvantages of SimPowerSystems are summarised.

- + Electrical modelling.
- + Reduces causality problems.
- + Comprehensive component documentation.
- + Direct access to MATLAB & Simulink environment.

- Many models only include part of the behaviour of actual components.
- Only electrical modelling (additional toolboxes are available).
- Complications with capacitors and inductors regarding how they are connected with different sources.
- Snubbers are needed and setting their values can complicate simulations.

3.3 OrCAD Capture

The program suite version that has been analysed is OrCAD Unison EE 16 which comes from the company Cadence. The programs in the suite being investigated are Capture and PSpice. Circuit design is done in OrCAD Capture. This program is intended mainly for electrical circuit design and simulation. This means that this program is not made to create models for other domains. However there exists a possibility for co-simulation with Simulink if a program called SLPS is added to the OrCAD program suite. This makes it possible to use models created in OrCAD Capture together with model systems in the Simulink environment. Many different libraries are available for inclusion when creating circuits. Some contain standard electrical components while others hold more advanced components such as microprocessors. It is also possible to add libraries from component manufacturers or your own special libraries. Simulation calculations and result presentation are both done by the PSpice program. The programs from Cadence are intended to be used to aid in the design and

simulation of electrical circuits. This means they are not as well suited for more general model creation using mathematical operations, although it is possible to use mathematical functions and expressions. Instead an advantage of the programs is the amount of accurate electric components available for complete circuit design, which can be used for direct PCB design.

3.3.1 User Interface

The circuit design in Capture is done in a schematic window. Components can be found in the parts library, where they are arranged in several smaller libraries and the components in one library often have similar function or application areas. It is important to remember placing a ground symbol in the circuit, otherwise the simulation fails. The interface of the program is shown in Figure 8. In the left part of the window an overview of the project is shown. This overview enables easy access to different designs in one project by arranging them as separate pages in the schematic hierarchy.

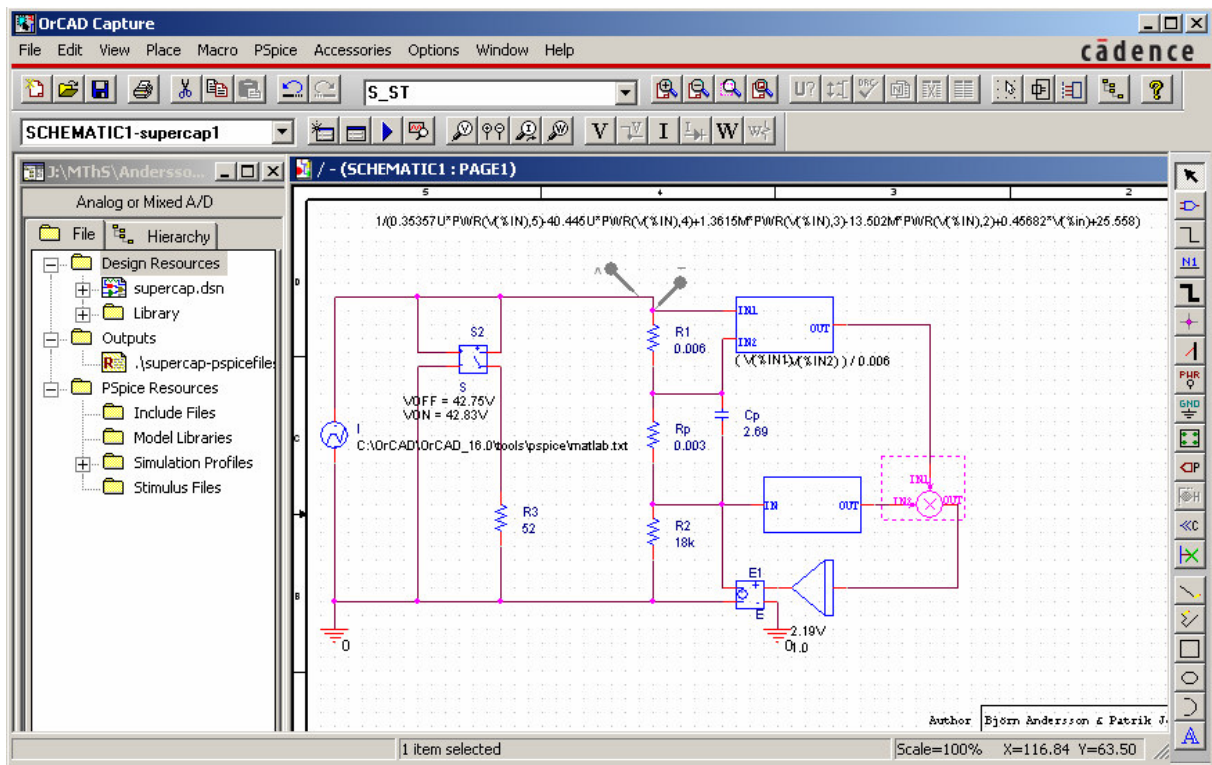


Figure 8: User interface for schematic creation in Capture.

When a simulation is started the Probe window automatically appears on the screen. Here the results of the simulation can be studied if voltage and current probes have been placed somewhere in the schematic. Fourier analysis can be used to evaluate the results of the data in Probe. This analysis is done on the data stored from the simulation and that data can also be manipulated using other mathematical expressions. In Figure 9 the Probe window is shown displaying the results of a simulation.

Program Overview

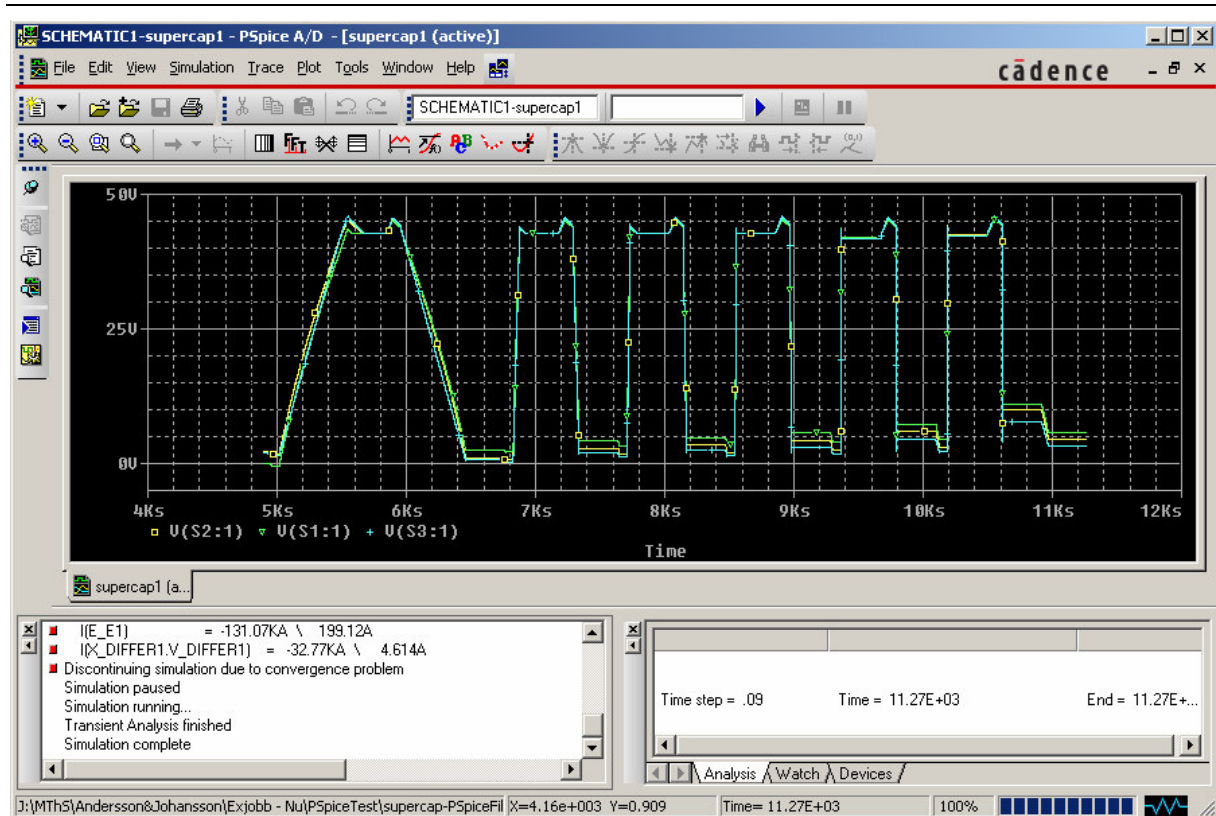


Figure 9: User interface for viewing simulation results in PSpice.

3.3.2 Different Types of Simulations

Different types of simulations can be done, each presenting different types of results. In these simulations different parameters, like the supply voltage or resistor value, can be swept to find a good value on a component. The available simulation types are bias point, AC sweep, DC sweep and transient simulation. In an extended program version called OrCAD EE Designer Plus it is also possible to receive more information. There it is possible to conduct sensitivity analysis, optimization, stress analysis and Monte Carlo analysis. One advantage of simulating in PSpice is that if it stops for some reason, it can be started at the same spot again after the error has been corrected. A complete restart is avoided and time is saved. PSpice includes the ability to conduct analog and mixed-signal simulations.

3.3.3 Calculation Method

PSpice uses node analysis as calculation method. In node analysis the different currents in to a node are summed up. This method leads to one equation for each node and for all nodes an equation system is generated on matrix form. To solve the matrix equation a numeric solver using the Gauss method is used. To be able to model a nonlinear component, a linearization is done in each point of the solution. To adapt a linear function in each point PSpice uses the Newton-Raphsons method to calculate the derivative for each point. PSpice uses a variable time step when it solves the equation system. This algorithm can optimize the time step to the appropriate conditions and therefore avoid large output data files. A shorter computation time is also achieved. [12]

3.3.4 Issues with Help Function

When using the help function for OrCAD some observations can be made. Initial observations conclude that since the help system covers all programs in the entire program suite it is sometimes hard to find the correct information. Many of the subjects that are searched after do not have any information in the help function. One particular subject that is missing in the help is information about the basic libraries and their contents. This lack of information makes modelling unnecessarily difficult. Since the library components are not described in the help function no information is to be found on how to import data into Capture circuits. In order to find the needed part to provide a controlled current source, several different libraries must be searched. This lack of assistance from the program help function motivates the use of some sort of literature describing these subjects. An available addition to the tested Capture program is called CIS (Content Information System). This addition has not been tested but it is mentioned since it is said to provide an improved ability to find information about the various available parts.

3.3.5 Summary

Here some of the advantages and disadvantages of OrCAD Capture are summarised.

- + Provides fast, intuitive schematic editing.
- + Boosts schematic editing efficiency by design reuse.
- + Includes several accurate electric parts.
- + Characterized components from manufacturers.
- + Large library of electrical components.

- Bad help function.
- Incomplete information on parts.
- Made for circuit design, not general model creation.
- Not designed to use control elements and functions.

3.4 PSCAD

The version of PSCAD that is analysed is 4.2.1 (Aug 11 2006). PSCAD is a simulation program that is used for AC and DC power simulations. PSCAD is only the graphical layer in which circuits are drawn and results can be presented, the actual calculations are all carried out by the EMTDC (Electromagnetic Transients including DC) solution engine. As can be understood from the program name, its main purpose is to create electrical circuits and analyse electromagnetic transients using simulations. However it can be used for many types of electrical simulations besides those investigating transients. To use MATLAB or Simulink together with PSCAD is possible through a specific interface. This interface uses a subroutine created using FORTRAN code to use the MATLAB functionality in a PSCAD model.

3.4.1 User Interface

In the program it is possible to schematically construct a circuit, using various electrical components. After the construction of the circuit is completed, it is simulated and results can be presented directly in the workspace area using special graph frames. These graphs enable a fast and simple presentation of simulation results. One important property of the modelling

environment in PSCAD is that it is possible to alter the model during simulations and view the results directly. There are even special blocks included in the component library that are designed for this purpose, such as a switch and a slider control. The component library contains many different blocks. Simple passive elements and control functions are included as well as complex models like electric machines and FACTS devices. It is also possible to build custom models either graphically or using the included Design Editor. Even though EMTDC calculates its solutions in the time domain, the results can still be converted to phasor magnitudes and angles via the built-in transducer and measurement functions in PSCAD.

One advantage over some other simulation programs (SimPowerSystems) is that EMTDC does not restrict how circuit elements can be combined. Any number of switching elements, sources, etc. can be placed in series or in parallel. To improve the ability to simulate highly nonlinear systems a system initialization technique can be used. Something called a Snapshot File can be created during any part of a simulation. The file contains all states and variables frozen from that time. Later a new simulation can be started using this Snapshot File and thereby unnecessary simulation time can be avoided. This can be especially useful when a simulation needs to run for some time before reaching steady-state. A Snapshot can be taken when steady-state has been achieved and all subsequent testing can be done using the Snapshot as starting point. [13]

One disadvantage when using the library to find a block is that they are not organised in a suitable way inside the various sub groups. The components are not displayed in order to provide a complete overview of all the blocks in the sub group, unless a very large monitor is used. This lack of overview and organization results in unnecessary time consumption. Even though the organization is not as good as it could be, it is still relatively easy to find the desired component. The components are also well documented in the program help which makes it easy to find the appropriate component and learn how it works. The user interface of PSCAD is shown in Figure 10.

Program Overview

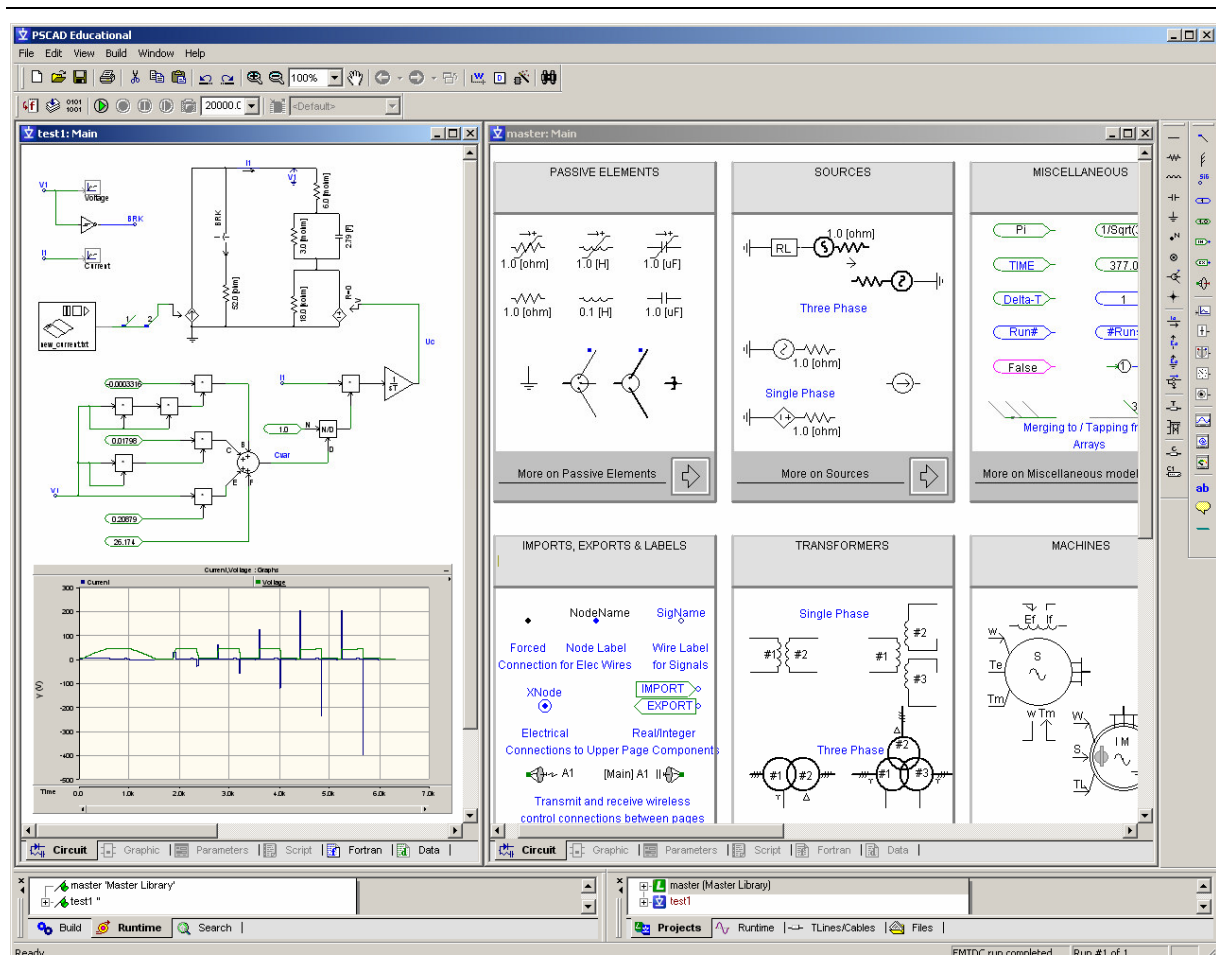


Figure 10: User interface for PSCAD.

3.4.2 Calculation Method

In the first section about PSCAD it is mentioned that the EMTDC calculation engine is used to solve the circuits that have been created. EMTDC is used to represent and solve differential equations in the time domain. The solution is based on the application of the trapezoidal rule to convert the differential equations of the network components to algebraic equations [14]. A fixed time step is used when a system is simulated. This can cause problems with large file sizes or simulation accuracy in some simulations. If a too large solution time step is set on a test that has fast variations in its value, the result might not be correct. Conversely if a small solution time step is chosen on a test that has a long duration, the file size becomes very large. This means that it is hard to get reliable results for long lasting tests without getting unnecessarily large data file sizes. If variable time step is used, the amount of data can be substantially reduced while still receiving good results. Since transients mostly have a short duration they are a good example of something that is well suited to simulate using PSCAD, that is also the claim of the company producing PSCAD [15].

3.4.3 Summary

Here some of the advantages and disadvantages of PSCAD are summarised.

- + Well structured Help with relevant information.
- + Includes several electric parts.
- + Large library of electrical components.
- + Ability to alter model during simulation.
- + Results can be presented directly next to the model.

- Lines are not connected when parts are moved during design.
- Only fixed step simulation is available

3.5 Saber

The company Synopsis produces a program that is called Saber. The part of Saber that is used to implement the model of the supercapacitor is called Saber Sketch and the analysed version is 4.0 (release Z-2007.03-SP2). Saber is a simulation program that simulates physical effects in several engineering domains such as mechanical, hydraulic, thermal and electric. Signal-flow algorithms and software control can also be used in the simulations. Some of the important features of the program are that it has a large model library, ability to do advanced analysis and support for standard modelling languages including MAST and VHDL-AMS. MAST is the analog, mixed-signal and mixed-technology HDL used in Saber. VHDL-AMS is an industry standard modelling language for mixed signal circuits which enables both continuous-time and event-driven modelling. Examples of the available analysis are Monte Carlo and Sensitivity. Through the use of the various analysis tools in Saber, Robust Design methodologies are supported which can be important when modelling large systems such as entire vehicles. Other than Sketch, Saber also includes a product called Harness which is used to design wire harnesses. It is not tested in this project since it is not relevant for the simulations being conducted.

Different types of simulations can be done in Saber. The small-signal frequency analysis is used to find the frequency response of a system. The DC transfer analysis performs a sweep over an independent variable and performs an operating point analysis at each value. The operating point analysis can also be conducted separately. The time-domain transient analysis simulation is probably the most important type since it often produces interesting results for many types of systems modelled in Saber. When starting a simulation some setting alternatives are presented. Some of the settings available to alter are the simulation time, integration method and algorithm used during simulation.

The program is mostly used in automotive, aerospace, power and IC industries. All these fields each have their own categories with parts in Saber to provide access to the most frequently used parts in the respective field. Some of the main design areas directed to the automotive industry are power window, wire harness, brake systems, hydraulic ABS, electrical power and power roof. The main areas mostly involve different parts of the electrical system and that is also the main focus of Saber, even though it includes the capabilities to model other domains as well. Examples of components available in the

automotive library are sparkplugs, electrical motors, batteries, fuel cell stacks, hydraulic pumps and valves.

Sketch makes it possible to create electrical circuits using standard electrical components and connect those using wires. This means that in its basic function it is very similar to programs such as PSCAD, PLECS and OrCAD Capture. There are however some properties of the program that separate it from the others. Saber allows simulations in several physical domains, models of mechanical and hydraulic components are available for simulation together with the electrical system. The multi-domain simulations are possible because Saber uses the MAST modelling language to create the behavioural models of all the parts in the library. The language is powerful and can model a vast number of components. For many of the available parts it is possible to view their MAST template which describes their behaviour, both using an introductory text and through actual MAST code. The complete MAST templates are only available for the basic parts such as resistors and capacitors. For more advanced components such as batteries and motors the main part of the MAST code has been encrypted and can therefore not be viewed. However the major effects in those models are still presented in text and the included model parameters are described. Some parts also have more detailed information about their functionality gathered in individual PDF-files.

3.5.1 Supercapacitor Component

In Saber there already exists a model of a supercapacitor in the Parts Gallery. In the program the model is called ultracapacitor which is an alternative name for supercapacitor. The implemented model uses a structure similar to the one used in the Two Branches model which is described further in Section 5.3.1. An additional aspect of the model in Saber is that it has a series resistance which is temperature dependent. Saber is the only program among the seven tested which includes a supercapacitor component that is this advanced. The component is tested and is found to function well and the related information enables parameters to be set easily.

3.5.2 Parts Gallery

A number of components in the Parts Gallery are modified versions of the general models in Saber. The modifications are made in order to characterize commercially available electrical components. According to the Parts Gallery there are about 28 000 characterized components available. Since the information regarding this characterization is proprietary, most of these component templates are encrypted. There are two classes of characterized components. One class has more detailed models that include model features such as static and/or dynamic thermal effects, operating area limits, statistical variations, aging and failure effects. The other model class contains models with more basic functionality and they are derived from data sheet specifications.

If there is a need for a model that is not available in the parts gallery there are some alternative ways to acquire such a model. It may be available by Synopsys in an optional library or it may have been created after the program was bought. The manufacturers of the parts, universities or user groups are other sources where it might be possible to find a completed model. A new model can be created by combining existing models and giving it a new symbol. The most powerful way to create a new model is using the MAST language. The disadvantage with this method is that it complicated and takes longer time than for example

creating a new model using the graphical interface in Saber.

To aid in the development of new models, Saber has a collection of tools that are gathered into a toolbox called Model Architect. Some of the tools available there are StateAMS, Table Lookup, Spice to Saber Translator, Battery Tool, Drive Cycle Editor, Load Profile Editor and Scanned Data Utility.

StateAMS is a tool that can create models mixed-signal finite state models using state charts and equations in a graphical process. Table Lookup can be used to create models from measurements, data sheets or other data sources. The Spice to Saber Translator can create MAST templates from several available Spice models. The Battery Tool can be used to characterize batteries using measurement data and characteristic parameters. The behavioural models in Saber that can be used for this are two lead-acid batteries, one with a thermal dependency and one without. The specifications used include for example DC values and self-discharge characteristics. The Drive Cycle Editor is used to characterize vehicle speed as a function of time using a model available in Saber. The editor contains several pre-defined drive cycles that can be used directly such as the NEDC cycle. It is also possible to use other sources of data. The Load Profile Editor can characterize resistance, power and current load models using data describing their variation in time. The Scanned Data Utility can be used as a utility to import data to several of the tools in the Model Architect collection. This is done using some type of image containing measurement data which can be displayed in the tool window. Using the imported image, curves can be defined by clicking on selected until the entire curve shape is captured. This can be useful if a curve is only available in a graphical format and not as data points.

3.5.3 Additional Tools

SaberRT is a tool that allows real-time simulation of Saber designs and also creation of C interfaces for HIL or other real-time applications. Using SaberRT it is possible to alter a model during simulation resulting in immediate effect in the simulation. The simulation results can be observed using gauges, walking waveforms or custom animations. A tool called Macro Recorder can be used to record actions performed in the different parts of Saber. The actions can then be played back as a sort of script which means that repetitive tasks can be automated. The Report Tool can be used to create a text report which can contain results from simulations, stress and sensitivity tests as well as operating point data.

3.5.4 Co-Simulation

Co-simulation is possible together with other programs such as Simulink, ModelSim and Verilog through the use of special tools. Simulink co-simulation can be done by taking the design done in Saber and then inserting into a Simulink model as a separate block. Then Simulink controls the simulation. Another possibility is to import a Simulink model to Saber, which means that Saber controls the simulation process. The simulations are done using a synchronous communication method between the simulation engines. This means that both engines progress independently in time and exchange data at determined time points. A disadvantage with this approach is that when continuous systems are simulated, the step-size must be small enough to produce accurate results, which can produce long simulation times.

3.5.5 User Interface

The graphical interface in Sketch can be seen in Figure 11. The parts gallery on the left is a large library which includes several tens of thousands of parts. As can be seen in Figure 11 it includes many different types of parts. The automotive library alone contains 1238 parts in different categories such as Fuel Injection, Ignition, Starting & Charging and Wiring & Lighting. The interface appears to be somewhat dated and could use an update in order to enhance the usability of the program. However even though it is not easy to see at a first glance, the program is full of useful features. Many of these features can be found in the icon-based menu at the lowest part of the program. From here it is possible to reach the parts gallery, a drawing tool and some other useful tools.

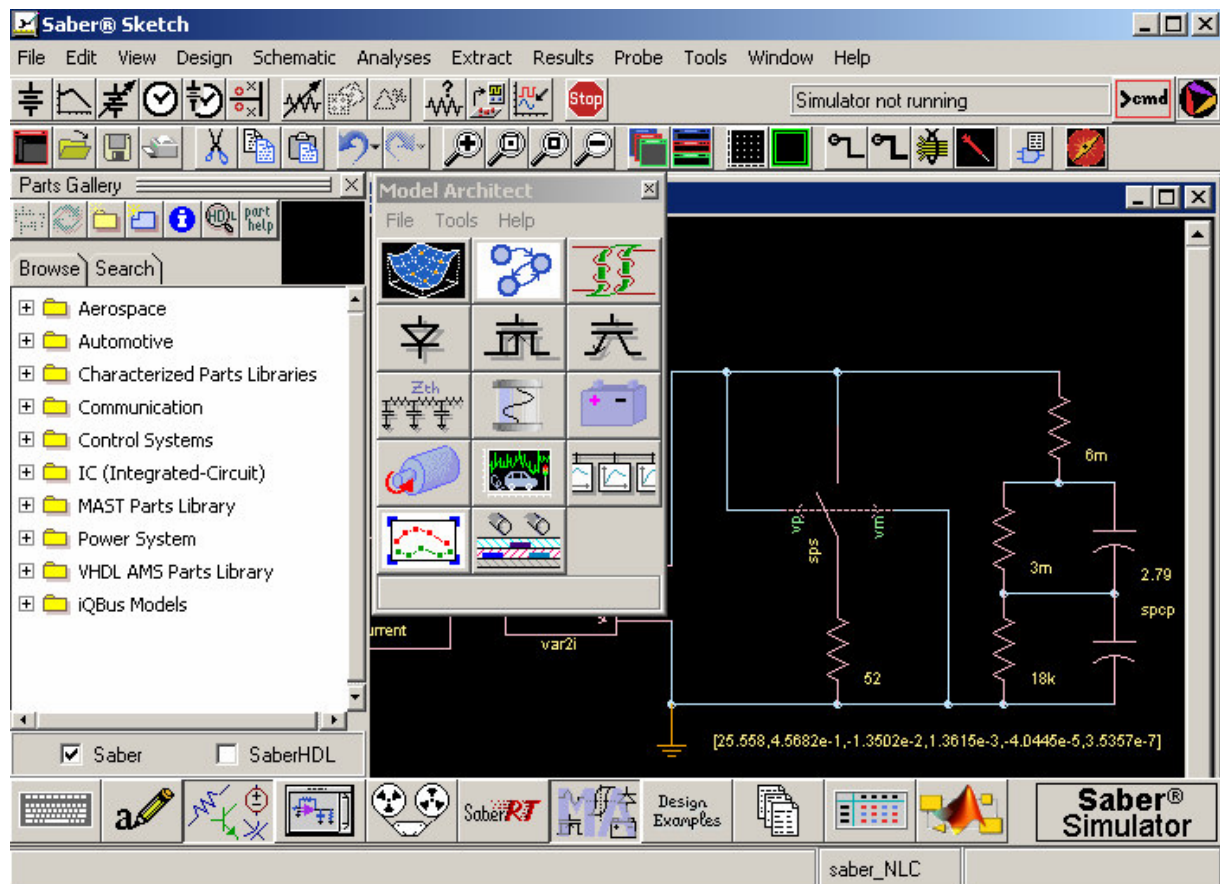


Figure 11: User interface for Saber.

Even though the parts library contains such a vast number of parts, it is often still possible to locate specific parts rather easily. The parts can be found either by searching for them using different search terms or by browsing through the various categories that the parts are organised in. In these categories some parts are duplicated, in order to avoid unnecessary browsing through several libraries for parts that are used when creating models of different categories.

To keep an overview of the parts used in a model, the Design Tool can be used. It displays an alphabetical list of the parts which can be useful when trying to find a specific component in a large schematic. The Design Tool is especially useful for larger designs which contain several

schematics that are arranged in a hierarchical order. The overview is improved and it is possible to archive the entire model in a selected directory. The tool can also create a report containing the elements of the entire design including their symbol names and their location in the file structure.

When the Parts Library and the Design Tool are opened they are positioned to the left of the Sketch window, taking up some of the available area for schematic design. If additional space is needed both windows can either be easily minimized using their respective buttons in the icon menu or they can be moved to a position that is more appropriate, by holding the title list and dragging the window to the new location. This ability to redesign the interface can be useful to accommodate user preferences.

Schematic Design

When several hierarchical schematics are used in a design, the Symbol Editor in Sketch can be used to create a symbol for it, in order to simplify its use in the design hierarchy. The symbol can be created automatically by the program if the appropriate input and output ports are available in the schematic. To make the symbols more easily identifiable it is possible to use the included drawing tool to create a suitable graphical representation of its function. Navigating around in a schematic can be done using the zooming buttons in the toolbar and if several schematics are used they can be arranged as tiles or cascaded depending on what is most appropriate.

Simulation

When simulations are supposed to be done the top row of the toolbar is useful since it provides access to the various simulation types as well as some analysis tools such as Monte Carlo analysis, parameter sweeps, sensitivity analysis and Fourier analysis.

The simulation results can be viewed directly in Sketch where a graph window can be placed out and the probe can easily be moved around the circuit to show different results. If the results need to be studied closer, a program called CosmosScope can be used. This program is more like a digital oscilloscope where for example Fourier transforms and max, min values can be calculated. It provides additional abilities when it comes to analysing results in the form of waveforms. CosmosScope has a measurement tool which includes over 50 different types of measurements. Some examples are delta y, max, min, rise time, fall time, period, RMS, peak-to-peak, histogram and standard deviation. Using these pre-defined measurement types directly on the results in a graphical interface is convenient for analysing results compared to writing code to do the measurements. The program includes a Waveform Calculator which can be used to conduct calculations on the waveforms in CosmosScope. The calculator can also be used for ordinary algebra and can be programmed to perform user defined calculations. Except for the normal x-y plotting CosmosScope provides the ability to plot results using a Smith or a polar chart.

3.5.6 Import and Export

To import data into Saber the Table Lookup tool which lies in the Model Architect menu must be opened. In this tool it is possible to import data from various file formats such as .dat and .txt. The .dat file format can be used by MATLAB data-files. When importing data using this

tool it must be checked so that two consecutive data points do not contain the same value in the time vector. If the values in the data that should be imported do not meet this requirement the data file has to be modified. Otherwise a message is displayed that warns about that the data is not monotonically increasing. The data file containing current measurements did not meet the demand of value separation in the time vector. To solve this problem, a MATLAB program containing for-loops is used in which the values are separated in time. When the data has been imported into the tool it is possible to view a plot of the values in a separate part of the tool window. The imported data can then be placed in the circuit as a lookup table where the current is the output if time is connected to the input.

The problem with monotonically increasing values in the time vector can also occur when exporting the data from Saber to a data file. Using the measurement data available, for simulations in Saber with the various models of the supercapacitor results in that problem. The output data has several consecutive data points at different parts of its time vector that are not separated in value. Two or more values in a row in the time vector have the same time value and this creates problems when attempts are made to interpolate the data in MATLAB. To fix the problem the data points are separated by looping through the entire vector and changing the time values so that no consecutive values are equal. When the issues concerning import to and output from Saber are solved, the simulations can be completed. In the previously analysed programs, the supercapacitor is mostly modelled in three different ways. In Saber a fourth approach to the modelling is also investigated. This approach is special because it uses a specific predefined component to model the nonlinear capacitance. This component uses a polynomial to describe the relation between voltage and capacitance. This modelling alternative can be seen in Section 5.2.5 about the polynomial model

3.5.7 Summary

Here some of the advantages and disadvantages of Saber are summarised.

- + Multi-domain simulation program.
- + The largest component gallery.
- + Good functionality to analyse simulation results.
- + Characterized components from manufacturers.
- + Good program Help.
- + Co-simulation with Simulink is possible.
- + Some components include aging effects.
- + Includes an advanced supercapacitor component.

- Graphical interface is somewhat dated.
- Co-simulation with Simulink requires that Saber is running.

3.6 PLECS

The version of PLECS that has been analysed is 1.1. PLECS is a toolbox for implementation of electric circuit design in Simulink. The main difference between PLECS and other toolboxes in Simulink is that PLECS is developed by Plexim GmbH and is not a product from The MathWorks like SimPowerSystems is. PLECS uses the calculation methods included in MATLAB to solve the mathematical equations that arise from the electrical circuits. PLECS

was from the beginning developed for simulation of power electronics but it can be used to simulate many electrical components or different phenomenon. One important benefit of using PLECS instead of using other circuit simulation programs on the market except for SimPowerSystems, is that PLECS easily can be integrated with Simulink. Simulink is good at simulating closed loop systems, which typically contain different kind of control loops. This can be difficult to implement in other programs. In Simulink it takes a long time to implement electrical circuits because they must be modelled using mathematical expressions, which make them hard to redesign. PLECS is an interface that simplifies the circuit design compared to Simulink while still enabling access to the ordinary Simulink development environment.

3.6.1 Semiconductor Components

The semiconductor models and especially the transistor models have the same disadvantage as the similar models in SimPowerSystems. The models only include the drain–source relation. The transistor switches state between on and off depending on a logical gate signal instead of a threshold voltage which is the case in programs such as OrCAD Capture. The dynamics during shifts in signal level on the gate related to the gate–source and gate–drain capacitances are not included either. This leads to the conclusion that the models do not include all of the physical behaviour of the component, which is needed for example in a design of the drive circuit to a power converter. In order to use PLECS for such simulations the model must be modified by the user. The existing models are still useful when larger models are studied, such as a complete electric drive system. This is because the losses in the converter mainly appear between drain and source in the switching transistors. The models can also be used to study the output voltage from the converters without a loss of accuracy.

3.6.2 Nonlinear Components

A library of nonlinear components is also included. The values of these components can change during the simulation using control signals. The voltage over the components or the currents through them must be measured, in order to be passed through some calculation steps to find the component values. The component models are created using controlled voltage or controlled current sources. This means that the models can cause problems in designs which use current or voltage sources if they are combined inappropriately. Current sources can not be connected in series with each other and voltage sources can not be connected in parallel. The advantage of using the available nonlinear components is that the overview of the design is improved. One major disadvantage is that a nonlinear resistor does not exist as a separate component. Instead it exists with a capacitance in parallel or an inductance in series. If a nonlinear resistor is needed the capacitance value needs to be set to a low value but larger than zero. This can introduce a new time constant that is significantly smaller than the other time constants in the system. This causes the solver in MATLAB to use a smaller step size which results in a greatly increased simulation time. In this case a nonlinear model using a controlled source is to prefer. Figure 12 shows the nonlinear resistor in the model circuit of the new supercapacitor model.

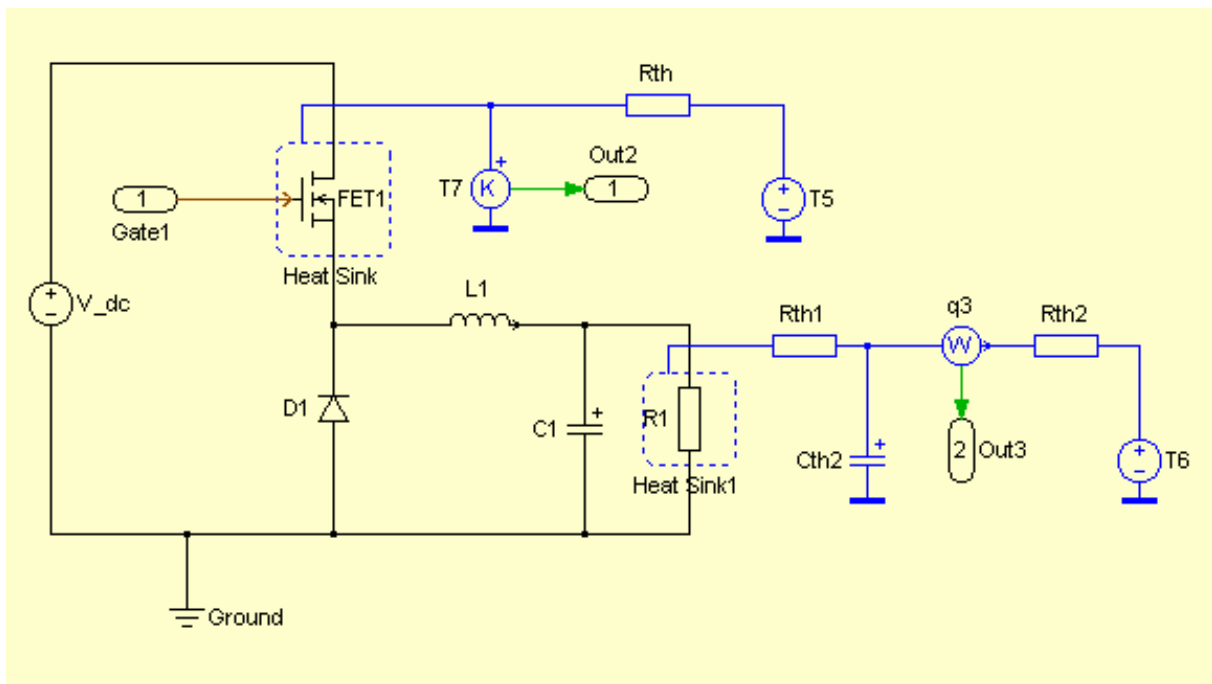


Figure 13: An example of a model in PLECS that uses the thermal components.

3.6.4 User Interface

The modelling in PLECS is very similar to model creation in SimPowerSystems. Both have a workspace in which electrical components can be placed and connected using wires and both are used together with the Simulink modelling structure. To build a model in PLECS a circuit block must be placed in Simulink and inside this block the PLECS schematic can be placed. In Figure 14 the user interface is presented and there both the schematic area and the component library can be seen.

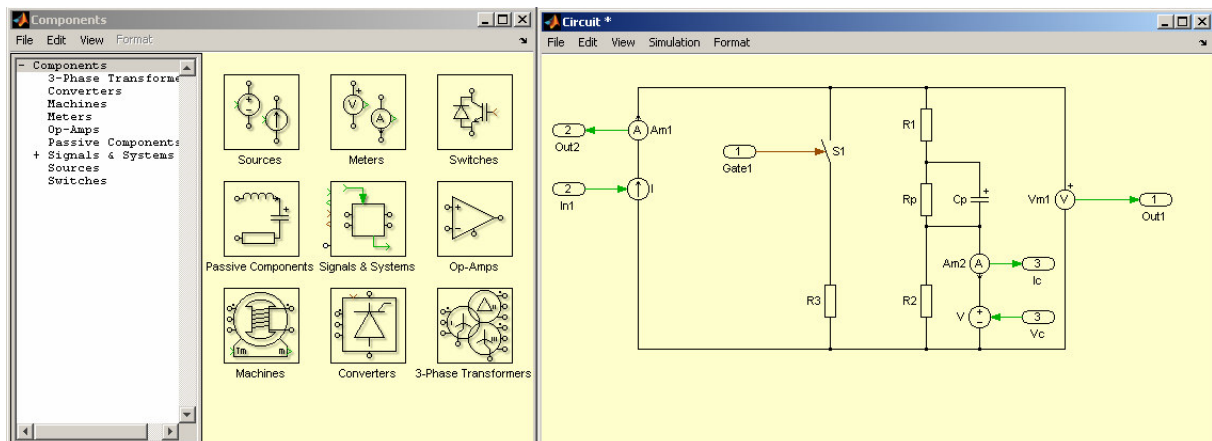


Figure 14: User interface in PLECS.

The components in PLECS are divided into different libraries which are well structured by applications areas. There are libraries for ordinary components such as resistors, capacitors, operation-amplifiers and power electronics but also more advanced libraries such as electric drives, converters and transformers.

In earlier versions of PLECS all the calculations had to be done outside the circuit window. This is because PLECS and Simulink do not use the same graphical interface, which means that PLECS and Simulink components can not be used in the same window. In the tested version some calculation can be done in the circuit window using the math operators that are located inside the Signal & Systems library in PLECS. Some mathematical blocks that exist in PLECS are comparator, function block, integrator, Fourier series calculation and lookup tables. Still the preferred method is to send the signals to Simulink so the mathematical and circuit window remain separated. However sometimes it can be useful if the calculation needs to be done inside a subsystem and many ports need to be added to reach the Simulink level. This is illustrated in Figure 15, where the calculations for the model of the nonlinear capacitance are done outside the circuit block.

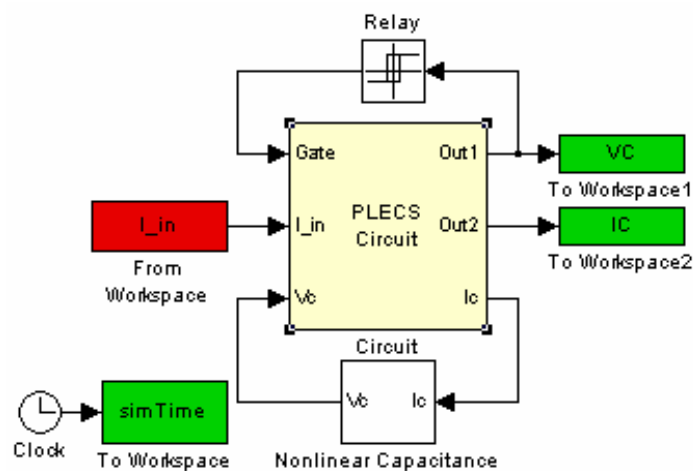


Figure 15: PLECS interfaced with Simulink modelling.

3.6.5 Calculation Method

When a PLECS circuit design is made the program transforms it to a state space model. There the energy storing components generate different states from the voltage and current variables using Kirchhoff voltage and current law. After applying these laws on the meshes and nodes in the circuit, the system can be reduced after some calculation. PLECS automatically transfers the resulting state space model into an S-function which MATLAB solver uses to run the simulation. [16]

3.6.6 Summary

Here some of the advantages and disadvantages of PLECS are summarised.

- + Direct access to MATLAB & Simulink environment.
 - + Well organised components.
 - + Thermal modelling library is included.
 - + Circuits are displayed clearly.
 - + Nonlinear component library is included.
-

- Semiconductor models are not including all effects.
- Does not have the same function as SimPowerSystems to analyse the results.
- Only electrical domain simulation (some thermal domain components exist).
- Limited to place electric components in restricted modelling area.

3.7 Dymola

In this report Dymola version 6.1 has been used to model the supercapacitor. Dymola is a graphical interface to the Modelica language which is a language structure that is used to model physical phenomena. The main benefit of using this program is that several domains easily can be integrated in the same model. Most of the other programs in this report are made for simulation in one domain, for example OrCAD Capture, PLECS and PSCAD are made for electrical simulations. In Dymola mechanical, electrical, thermodynamic, hydraulic, pneumatic, thermal and control systems can be combined in one model. The modelling in Dymola can be done both using the graphical interface and by writing code directly in the Modelica language. It is also possible to simulate models made in Dymola with Simulink models, using a specific Dymola block that can be placed inside a Simulink model. That block takes a Dymola model and translates it to an S-function that can be used in Simulink. The block can then be connected to other Simulink blocks as well as other Dymola blocks. A special Simulink export functionality can be added, which enables creation of Simulink models which include Dymola blocks to be simulated on computers which do not have Dymola installed.

One important advantage with the modelling structure in Dymola is that it is not necessary to consider the causality of the model. This is achieved by representing all the model parameters using differential algebraic equations (DAE). While modelling it is not necessary to consider the contents of these equations since the program takes care of the organization of all the model equations. When it is time for simulation they are all arranged so that the system behaviour can be calculated. This simplifies the modelling procedure compared to mathematically based models such as those used in Simulink. When the causality of the model does not have to be considered it is possible to replace parts in the model and still be able to simulate the system. To control the time it takes to simulate as well as the simulation accuracy settings such as solver type and error tolerance can be altered.

If several simulations are supposed to be conducted using varying parameter values, this can be accomplished through the use of a scripting language. The language includes abilities to load model libraries, set parameters and start values as well as to simulate and plot variables.

A disadvantage with Dymola is the limited ability to conduct post-processing, data manipulation and control design. This can limit its abilities to act as a stand alone simulation tool. However if it is combined with software that is more suited for such applications such as MATLAB & Simulink, it can still provide significant benefits to the modelling process compared to modelling using only Simulink.

3.7.1 Performance

In order to increase the computational performance of Dymola several features are available. The built-in symbolic translator processes Modelica equations and simplifies them in order to increase simulation speed. To do this all the equations used to describe the behaviour of the components used in a model are gathered. The equations are then sorted so they can be calculated in the correct order. Finally the states and variables are identified and the equations are modified so they are all written in state-space form. By doing these symbolic manipulations unnecessary states can be avoided which improves simulation performance. The model translation also handles algebraic loops by analysing the system equations.

By using the performance increasing features, the computational speed can be improved which can enable real-time simulation for various models. The translator can generate C-code for the simulation which can be exported to Simulink or HIL platforms such as dSPACE or xPC. Inline integration is another feature which is supposed to increase simulation speed. Inline integration uses a symbolic and numeric approach to solve differential-algebraic equations systems [17]. By tuning the step size used while solving the system an appropriate balance between simulation performance and accuracy can be obtained.

3.7.2 Thermal Modelling

When designing models in Dymola it is possible to use the thermal components in the Modelica library. The use of those components can improve the overview of the temperature calculations compared to making the temperature calculations using mathematical blocks. The reason for this is that the temperature components are adapted for thermal calculations and can therefore be more appropriate for such calculations.

3.7.3 Component Library

All of the components available in the different libraries for Dymola are created using the Modelica language. The help is well structured and contains relevant information about the program parts. The components included in the Modelica standard library each have an information page where the used Modelica code is shown together with an explanation of how the component works. Even the components in the additional commercial libraries are created using that language; however the help sections for those components do not display their Modelica definitions. The component parameters and connectors are defined and explained in the help section. A few of the more complex components in the Modelica standard library also have references to scientific reports containing more detailed information.

The standard Modelica library is always included in Dymola and it contains a great deal of components needed for model creation. It is provided by the Modelica Association which is a non-profit, non-governmental organization with the aim of developing and promoting the Modelica modelling language. However there are also additional libraries available that can be used in Dymola. Some are free and others are commercial and are provided by different companies. There are free libraries created by parties other than the Modelica association available for download at modelica.org. Examples of such libraries are Fuzzy Control, SPICELib, NeuralNetwork and FuelCellLib. Some examples of commercial libraries are PowerTrain, SmartElectricDrives and VehicleDynamics.

When creating models in Dymola it is possible to create new libraries which can contain user defined models. This enables re-use of already created models which prevents unnecessary modelling. A created model can be used in other models and to simplify recognition of the various models they can each be given a specific icon. The icons are created using simple drawing tools such as rectangle, line and polygon and when completed they appear in the Packages window instead of the standard icon.

When building a new library it is possible to create packages and models that are based on previously available components. If a new model is taken as an example it can either be a copy of an available model or it can be extended from that model. In both cases the new model contains the same components as the original model. The difference is in how they are linked to the original model. The copied model is not linked, which means that if any alterations are made to the original, the new copy is not affected. However the changes in the original are visible in the new extended model, since it is linked to the original.

3.7.4 Extended Functionality

To extend the functionality of the Modelica program other components can be added. Examples of such components are real time simulation, 3D-animation and Simulink Interface. There also exists components which are part of a Design package and they can add abilities such as calibrating and optimising models. Another example of the capabilities that are part of the Design package is the Monte Carlo analysis tool which enables model testing.

The Model Calibration tool can be especially useful when adapting a model to several different actual components, since it is used to tune parameter values of a created model using measurement data. This can save time when adapting models to different components such as various supercapacitors from different manufacturers.

The company Claytex offers a specific tool called Simelica, which can be used to translate a Simulink model into an equivalent Modelica representation [18]. The Modelica model can then be used inside Dymola in the same way as a model created directly in the program. This can be valuable if many models have already been created in Simulink and a switch to the Dymola modelling environment is made. Time can be saved in the translation of these models since a great deal of manual model creation can be avoided. However, one advantage of implementing models using Dymola is that they can be modelled using other blocks than the mathematical ones in Simulink. This means that many models can benefit from being redesigned using the components available in Dymola instead of translating a Simulink model directly. Another advantage of translating the models from Simulink to Modelica code can be that the simulation speed can be improved because of the symbolic capabilities of Dymola.

3.7.5 Steady-State Calculations

If steady-state calculations are to be conducted, this is possible in many cases by setting the initial values for the model components and then running a test with short time duration. The main application of Dymola is to simulate dynamics over longer times and not to do steady-state calculations. Because of this it is sometimes not possible to set all the initial conditions so that steady-state calculations can be made right away. In those cases it is possible to do an initial run with the model so that it reaches steady-state and then save the results at that time,

so additional tests can be started using that result file.

3.7.6 3D Animation

To simplify visualisation of a system it is possible to build 3D models which can be animated and controlled using models created in the program. This is done using a specific library called Multibody, which is a part of the Modelica standard library and lies in the sub group Mechanics. The library can be used to for mechanical systems such as robots, mechanisms and vehicles. Several different predefined parts are available to use when building 3D models. By connecting the parts using components such as springs or joints many different systems can be modelled. The library also contains various forces that can be included in the models to affect their behaviour. This animation can be a good way to visualize the model function in order to verify its function.

3.7.7 User Interface

The models are built in a graphical workspace where different components can be placed out to represent different parts of the model, Figure 16 shows the workspace. The different components are easily found in a well structured library that can be seen in the window called Packages on the top left part of the figure. The Components window at the bottom left contains a list of all the components in the active model. The list can be used to easily access the different components and alter their properties.

When building models it is important to consider which connection points on the blocks belong to which signals. This is especially important when rotating and flipping blocks, because then the signal positions are repositioned. The name of each connection point can be checked by holding the mouse pointer over the connection.

Program Overview

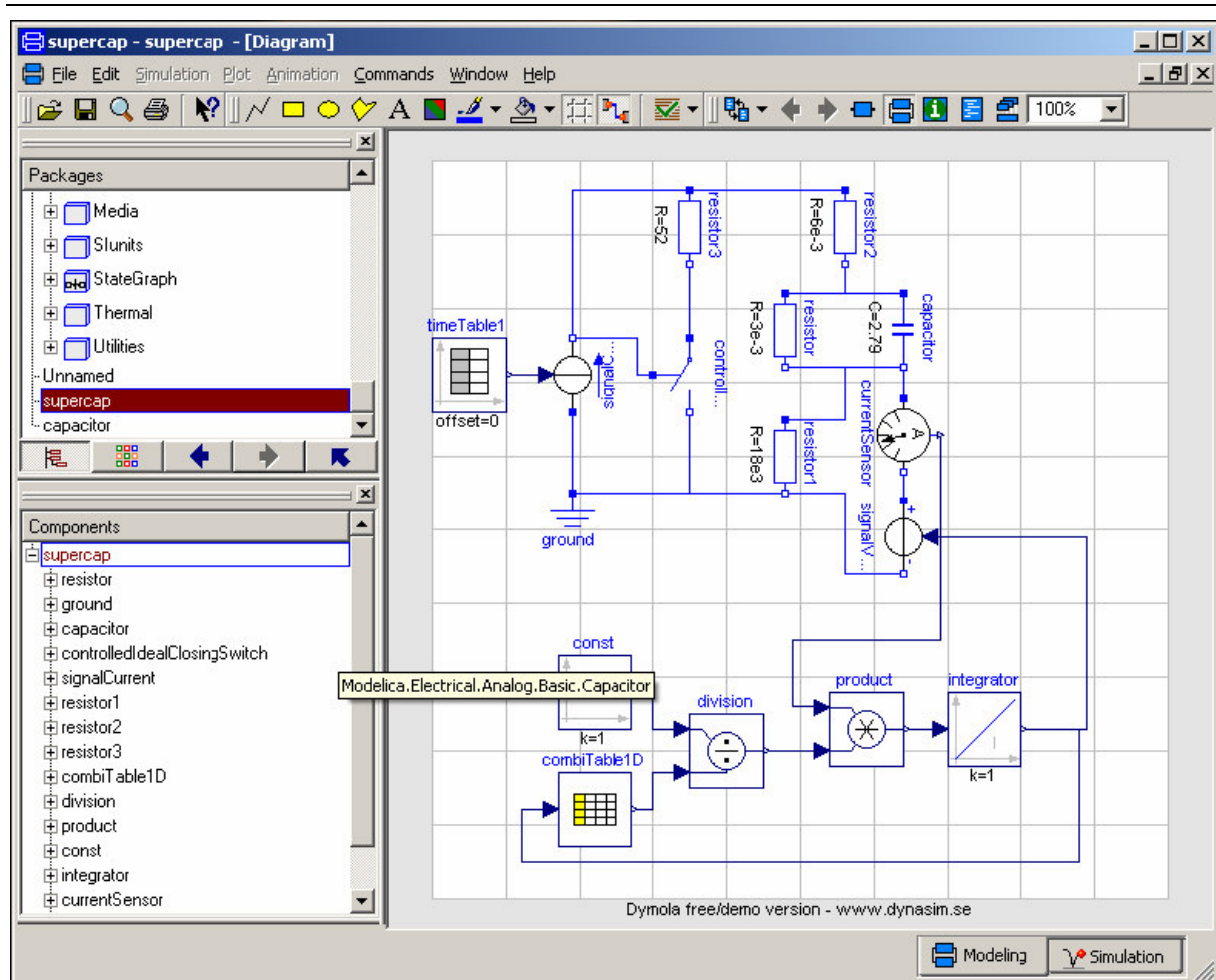


Figure 16: User interface for modelling in Dymola.

To start a simulation the simulation window must be used. To be able to see the results the signals of interest must be chosen. This is done in the left part of the window where all of the components in the model are listed. One advantage with this list is that all the values are already calculated so the user only needs to choose the different values that should be represented in the graph window. This means that no probes or measurements blocks need to be placed out to view the results. By using the functions available in the “Advanced” menu below the list with variables additional analysis is possible. The difference between two signals can be plotted and results from tests using different parameter values can be compared. The standard representation of all signals is done using time on the x-axis, however it is also possible to plot one signal against another. Beyond these alternatives it is not possible to do any additional post-processing of the signals that result from the simulations. So if more advanced analysis should be done, the data must be exported to another program such as MATLAB. Exporting data can be done to the file formats .mat, .txt or .csv by right-clicking on the model name with the correct test number and selecting Save As.

The presented results in the respective plot windows can be exported to .png files that can be used when creating reports. No additional file formats such as .emf, .tiff or .jpg are available for saving images of the plot windows. There are some alternatives available when customizing the appearance of a plot. Plotting range, axis labels, legends, heading and line styles can be edited using some pre-defined alternatives. Some plotting alternatives such as

changing the size of the text in the plots or controlling the details of the appearance of plotted values are not available. To have more control of how the results are presented the data must be exported to another program that has these capabilities. The program interface when viewing simulation results can be seen in Figure 17.

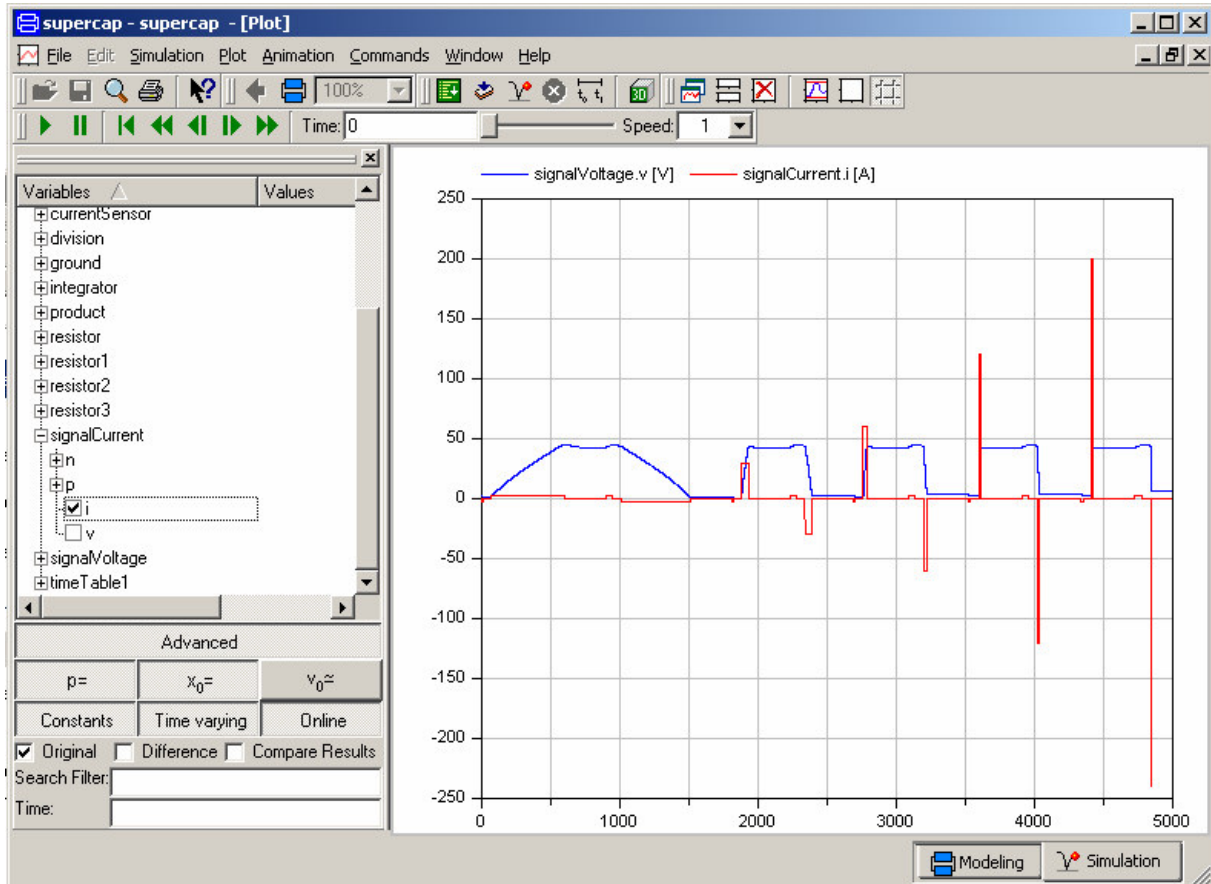


Figure 17: User interface for simulation results in Dymola.

3.7.8 Installation Problems

During attempts to install the demo version of Dymola on two different computers, an error related to the GCC compiler occurs, which makes simulation impossible. To get some help about this problem Dynasim was contacted and they provided solution suggestions which alleviated the problem. A third computer displayed some performance problems that could not be explained by its hardware configuration. This problem appears to be computer specific since it does not appear on any other computer. When testing an evaluation version of Dymola on a new computer none of these problems occurred.

3.7.9 Summary

Here some of the advantages and disadvantages of Dymola are summarised.

- + Multi-domain simulation program.
- + Equation based modelling.
- + DAE solver.
- + Symbolic manipulations in solver.
- + Several additional libraries available.
- + Simulation with Simulink is possible without Dymola installed.
- + Model calibration can be added, which enables parameter estimation.

- Limited ability to analyse results.
- Limited export capabilities.

4 Measurements

This chapter presents information about different measurements conducted on supercapacitors. At VTEC measurement data for several supercapacitors is available. The data from one such test is used in the creation of the first model version. That test is called screening test because it is supposed to provide a good overview of the behavioural aspects of the supercapacitor. To further analyse the behaviour of supercapacitors additional measurements are made on a few different supercapacitors. The names and properties of the tested supercapacitors are gathered in Table 2. A first test round using three different test procedures is conducted with these four supercapacitors in a room temperature of approximately 23 degrees Celsius. One of the three test procedures is chosen to be used for additional testing at different temperatures to be able to see what effect temperature has on the supercapacitor behaviour.

Table 2: Properties of tested supercapacitors.

Name	Voltage [V]	Capacitance [F]	Supplier	Type	Nom. Current [A]	Cells
ESSP08	2.3	2700	Nippon-Chemi-Con	PC	500	1
ESSP18	2.7	5000	NessCap	ACN	1000	1
ESSP48	42	33	EPCOS	ACN	500	18
ESSP16	2.7	2600	Maxwell	ACN	-	1

To perform tests on the supercapacitors the test equipment available at VTEC is used. This test equipment included a Digatron Battery Tester. The program used to create programs and control the test circuits is called BTS-600. The Digatron has a maximal sample speed of 100 Hz and a value insecurity of 0.5 % of full scale for the voltage and current measurements. When conducting measurements on individual supercapacitors it is necessary to connect a support battery in the measurement circuit to be able to reach voltages below 3 V. This is because the measurement equipment can only reach that level on its own. The battery is connected in series with the measurement equipment according to Figure 18 where the measurement points are also specified.

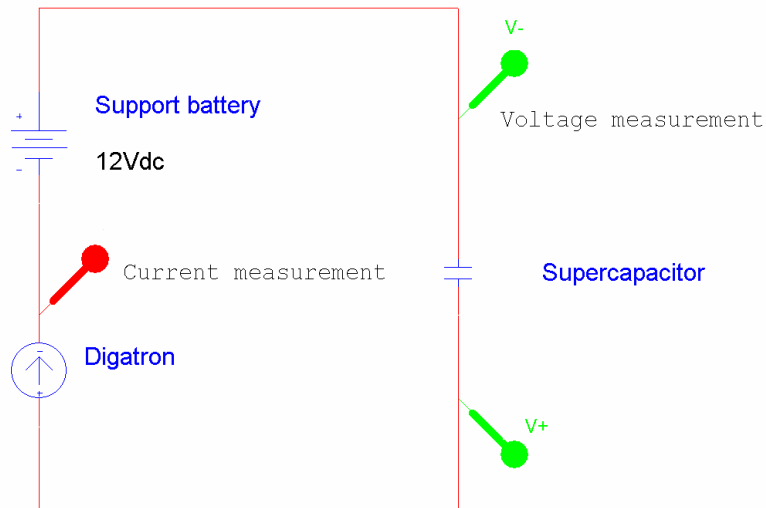


Figure 18: Measurement circuit.

The battery being used for this purpose previous to the start of this project had deteriorated so much that the construction of a new support battery was necessary. The new battery is created by connecting 12 battery packs of 12 V each together in parallel using copper plates of varying sizes. The copper area is selected relatively large in order to minimize the resistance losses. The batteries are stacked together using pieces of chipboard as can be seen in Figure 19. Using a special programming language the tests can be written using various commands for actions such as charge and discharge. Four different test procedures are used and they are all described in the following sections.



Figure 19: Photo of the support battery used in measurements with single supercapacitors.

4.1 Screening Test

The test described here that had already been conducted before the start of this project. The measurement data belongs to ESSP48 which is why that supercapacitor is analysed and used to create an initial model in the various program alternatives. The test consists of some charge and discharge cycles using increasing current levels under decreasing time intervals. The resulting voltage over the supercapacitor is shown in Figure 20.

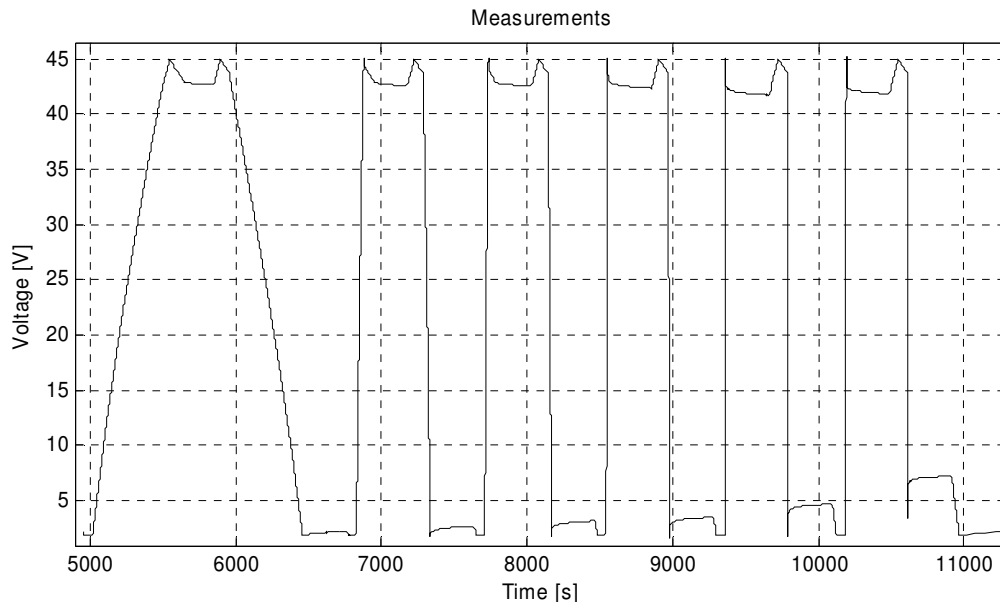


Figure 20: Voltage from screening test using ESSP48.

As seen in Figure 20, the charge/discharge current is initially very small which results in a slow increase in voltage level. As the current level increases, the voltage derivative also increases. The final discharge is done using a current of 400 A, which results in a discharge time of approximately 3 seconds.

4.2 Two Branches Test

Since the screening test was conducted before the start of this project work, the two branches test is the first measurement conducted in this project. The purpose of the measurement is to find the component values according to the two branches model which is described in Section 5.3.1. The test consists of charging the supercapacitor from being completely discharged to being fully charged using a constant current. Then the charging current is turned off and the supercapacitor is left for 20 minutes to see how the voltage varies over time. One advantage with the two branches model is that it brings a simple and documented approach to parameter determination [1]. The relatively simple structure of the model circuit also makes tuning of the model parameters easy since the effect of altering a parameter value is seen directly in the resulting model voltage. When creating the test program the appropriate data gathering intervals and charge current levels are found using some trial runs. The test program is used to test all the four supercapacitors. The resulting voltage curve from the test using ESSP48 is shown in Figure 21.

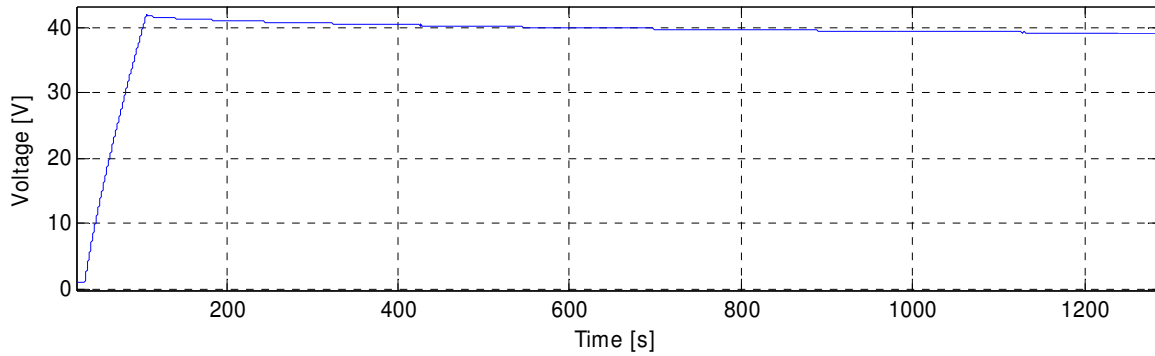


Figure 21: Voltage curve for the charge test used for the 2 branches model on ESSP48.

4.3 Varying Voltage Test

To better analyse the behaviour of the different supercapacitors a new test is designed based on the procedure used in the screening test. Instead of always charging to the rated voltage, this test uses charges that reach different voltage levels. During the cycle the supercapacitor is charged to different increasing voltage levels. The reason for charging to different voltages is to be able to see how the supercapacitor behaviour is affected by varying voltage levels. Between each charge a discharge is done so that charging always starts from the same level. This charging cycle is then repeated using different current levels. During the first cycle the supercapacitor is charged to three different voltage levels but when the current increases the lowest voltage level only consists of losses in the series resistance. Because of this the low voltage level is excluded from the high current test. The resulting voltage curve using ESSP48 is shown in Figure 22.

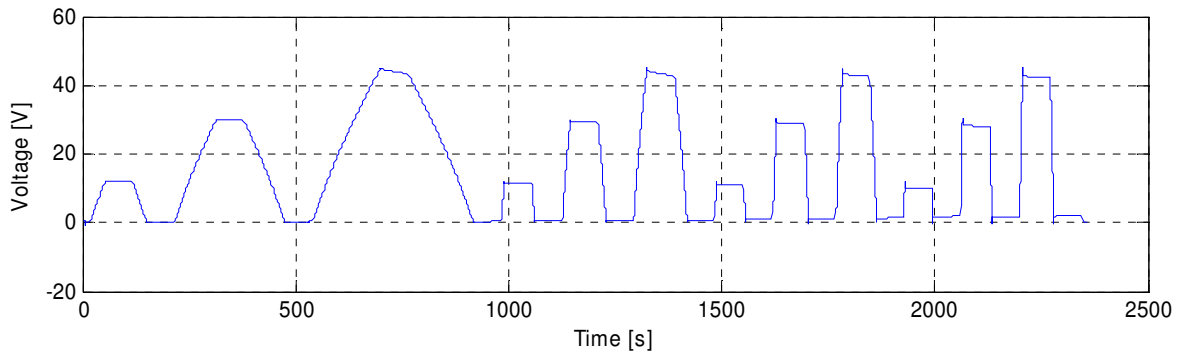


Figure 22: Voltage curve for the test using multiple voltages and currents on ESSP48.

4.4 Series Resistance Test

This test is appropriate to use when estimating a value on the series resistance. To do this several different voltage drops are analysed in order to find an average value. The voltage drops occur because of the series resistance in the supercapacitors. These voltage drops occur after either charges or discharges at different voltage levels which also makes it possible to see how the series resistance of the supercapacitor varies with different voltage levels. A cycle is created which consists of several charge steps from zero voltage up to the rated

working voltage level, and then discharge steps from that level back to zero voltage using the same steps as in the charge mode. All charges and discharges use the same current which is chosen so the voltage change becomes reasonably fast but still does not create too large temperature increases.

The test described in Section 4.3 contains several charges and discharges at different voltage levels as well as at different current levels. Since that test also displays other behavioural properties of the supercapacitors it is therefore determined to be more appropriate for determination of the series resistance value as well. However the series resistance test described here can still be interesting to use as a way to verify the function of the improved model. One test using this test cycle is shown in Figure 23 in order to clarify its function.

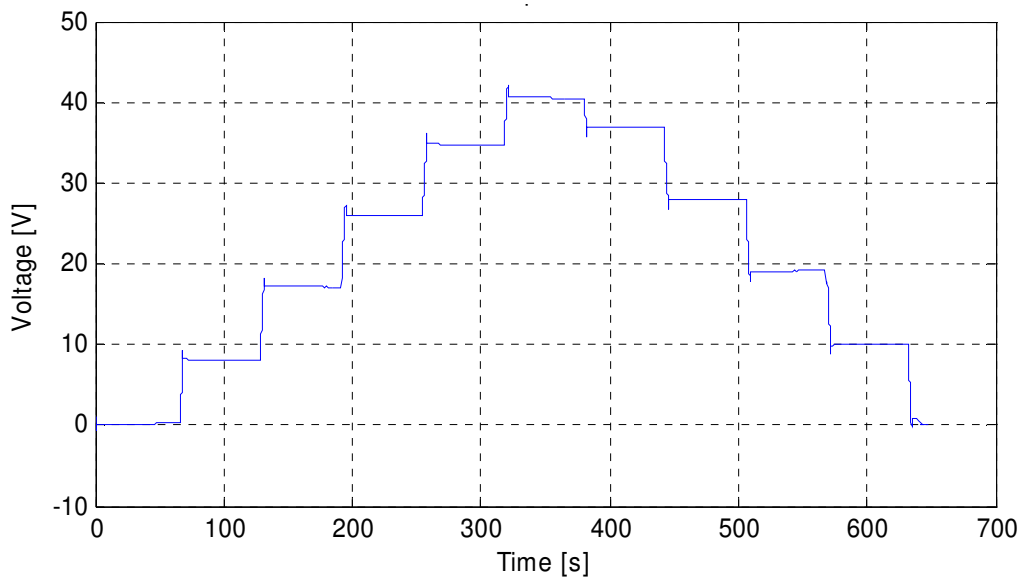


Figure 23: Voltage curve for the test used to find the series resistance value on ESSP48.

4.5 Temperature Measurements

All test cycles are tested in room temperature for all the different supercapacitors and the voltage variation test described in Section 4.3 is selected to be performed at different temperatures. It is chosen because it includes most of the interesting characteristics of the supercapacitor. The purpose of these tests is to see how the supercapacitor behaviour changes with the temperature. To be able to do the tests at certain temperatures a climate chamber is used. All the four supercapacitors are connected to a cable each which means that additional connections can be made outside the climate chamber. Thus unnecessary access inside the chamber is avoided which helps keep the temperature stable.

4.5.1 Test Procedure

The four different temperatures chosen to be included in the test procedure are -30°C , -15°C , 0°C and 40°C . Tests at room temperature (23°C) had already been completed so no new measurements are done using that temperature. The resulting voltage curves for the different temperatures using ESSP48 are shown in Figure 24. The most visible difference between the

measurement results is that low temperatures mean longer simulation time. This is because the measurement program must wait longer for the temperature to drop to an appropriate level. In low temperatures the series resistance losses are higher which leads to a larger temperature increase during these measurements.

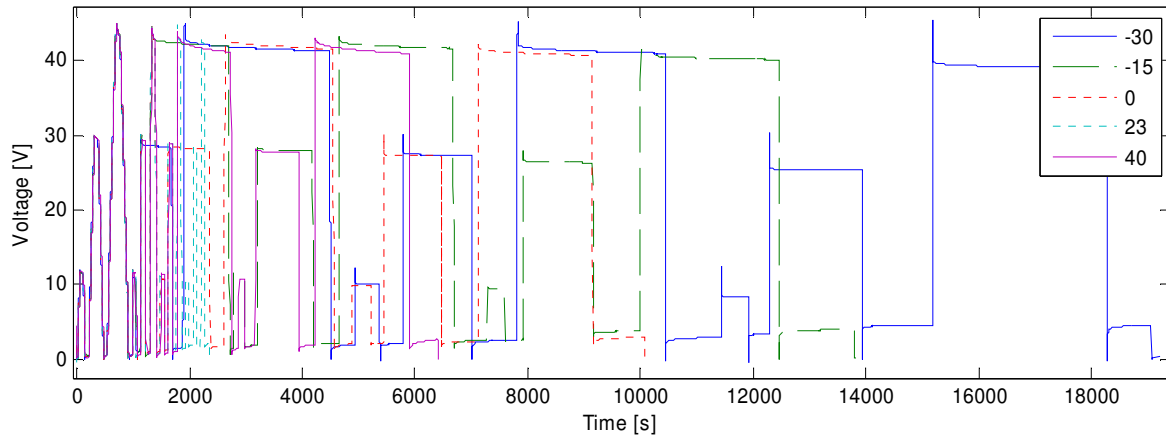


Figure 24: Resulting voltages using ESSP48 at different temperatures.

4.6 Further Measurement Analysis

From the measurements conducted using the various supercapacitors at different temperatures some additional analysis is conducted. Conclusions drawn from these analyses are presented and discussed.

4.6.1 Series Resistance Measurements

One observation made from analysing the results is that the series resistance increases with decreasing temperature. This effect is mainly caused by the decreased mobility of the ions in the electrolyte during low temperatures. An explanation on how the calculation of the series resistance is done can be found in Section 5.1.3 under the heading Resistance R1. When the results are studied, it is observed that the series resistance dependence on temperature changes for the different supercapacitors. This change in behaviour can be explained by the different types of electrolytes used by the supercapacitors, where the two types are ACN (acetonitrile) and PC (propylene carbonate). In the temperature interval that is used during measurements, the ESSP08 series resistance increases almost 3.7 times at $-30\text{ }^{\circ}\text{C}$ compared with the value at $40\text{ }^{\circ}\text{C}$, which can be seen in Figure 25. For the supercapacitors ESSP16 and ESSP18 the temperature effects have a much smaller impact on the series resistance changes compared to ESSP08. The resistance for ESSP18 increases about 1.35 times between the same temperatures. The results of the measurements for ESSP18 can be seen in Figure 25. The ESSP48 capacitor also has a resistance trend similar to ESSP08 but there the resistance varies with a factor of two. This larger variation in series resistance for ESSP48 can be explained by the module being built up from 16 series connected supercapacitors rather than electrolyte properties since ESSP48 uses the same electrolyte as ESSP18 and ESSP16. The large deviations in series resistance for different measurement points can be because of heating effects caused by large currents.

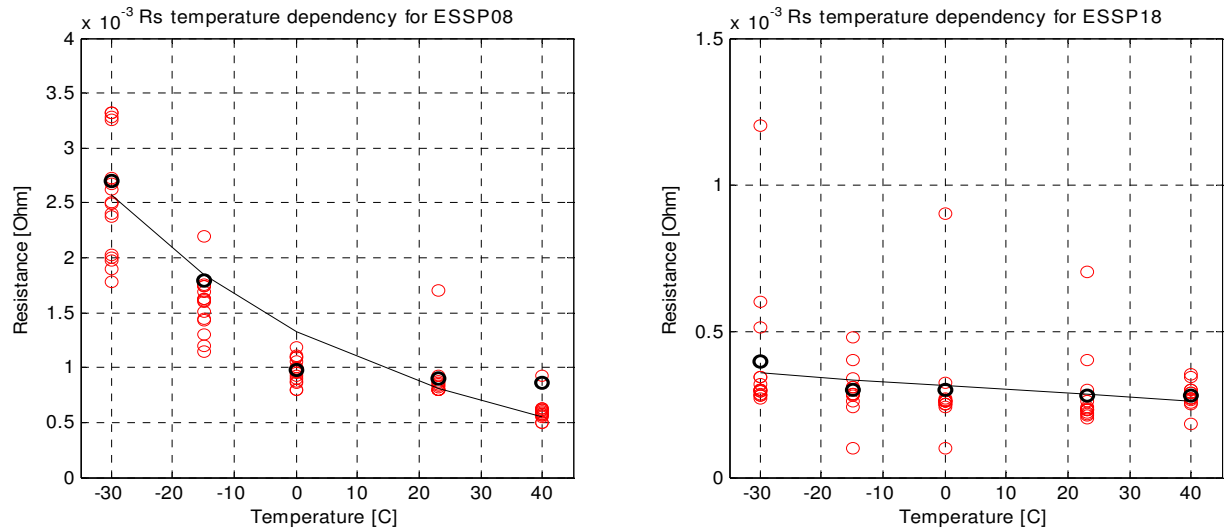


Figure 25: Series resistance temperature dependency, the left part is for ESSP08 and the right represents ESSP18.

4.6.2 Capacitance Temperature Dependency

The capacitance depends mostly on physical values such as the effective area of the electrodes and ion concentration in the electrolyte. These values are not supposed to be affected by the small temperature interval that is used for the measurements. An explanation on how the calculation of the capacitance is done can be found in Section 5.1.3 under the heading Main Capacitance. When the measurements of the capacitors are studied their relation between capacitance and temperature seem to be similar for all four of them. At low and medium charging current, up to 100 A, some of the capacitors seem to have a temperature dependency that can be established but at larger currents the measured voltage often fluctuates too much to be useful. This relation can be explained by the ions in the electrolyte having more difficulties to arrange themselves near the carbon surface during low temperatures. The results of the capacitance temperature dependency can be seen in Figure 26 for ESSP08.

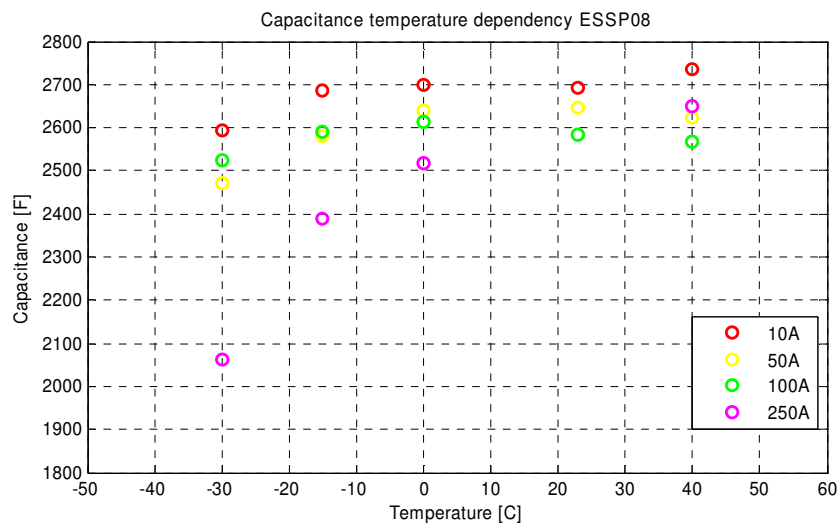


Figure 26: Capacitance variation with temperature.

4.6.3 Capacitance Voltage Dependency

The capacitance also has a voltage dependency which can be seen during charges or discharges with constant current. This introduces another nonlinear behaviour in the system. After calculating the capacitance with the method described in Section 5.1.3 under the heading “Lookup Table for Nonlinear Capacitance”, the result is that the capacitance increases almost linearly with the voltage.

The capacitance is related to the concentration of ions near the electrodes. When the voltage increases the electric field attracts more ions and the concentration of ions increases near the electrodes in the capacitor. This leads to an increased capacitance [5].

When analysing the results a behavioural property is observed. In the results from measurements using ESSP18 the capacitance relation to voltage seems to be different for charge and discharge. This relation can be seen in Figure 27 where the capacitance during charges increases almost linearly but during discharges the capacitance seems to be constant.

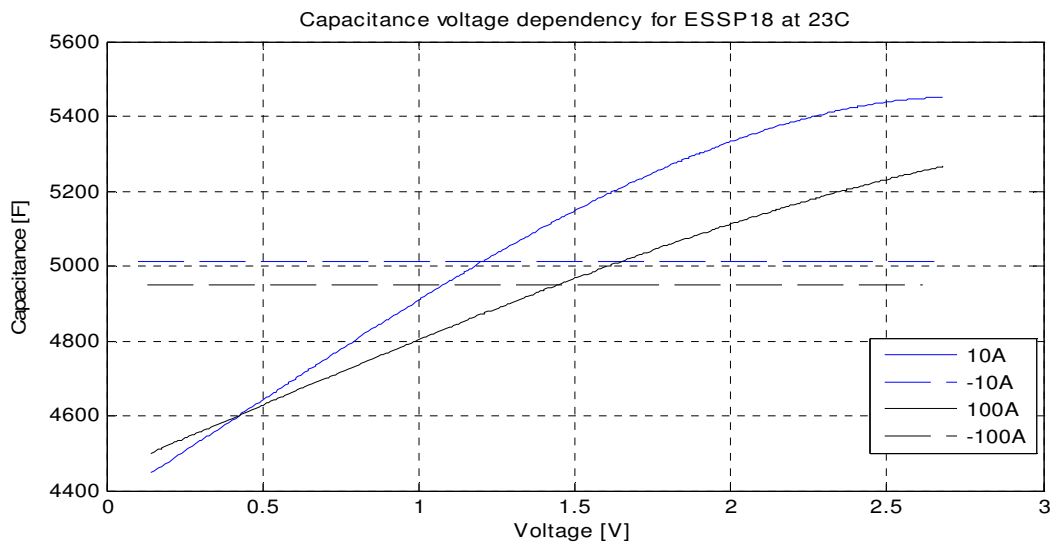


Figure 27: Capacitance voltage dependency for ESSP18, calculated with constant charging current.

This trend can also be observed for ESSP08 but here the differences are smaller. For ESSP08 the curves that describe the voltage dependency during charges are more separated than the curves describing the relation during discharges. One possible explanation to the differences between charges and discharges is that, during charges diffusion and electric forces obstruct each other but during discharges the forces are cooperating. The differences for ESSP08 can be seen in Figure 28.

Measurements

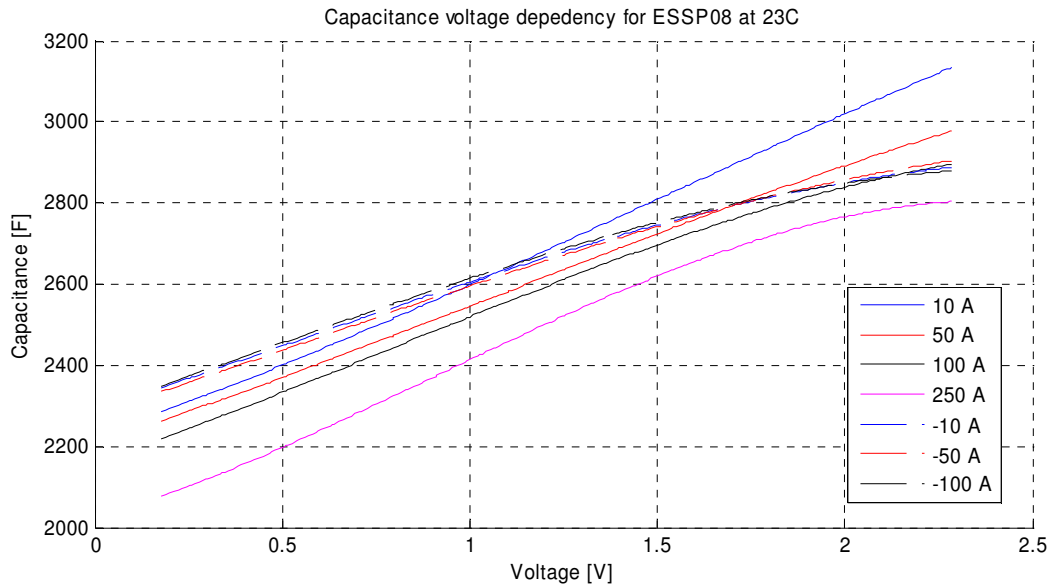


Figure 28: Capacitance dependency with voltage for ESSP08, calculated with constant charging current.

If a capacitance value is calculated using the calculation which is described in (4), a difference can be observed for high and low currents. The highest capacitance is always found when the low currents are used and the lowest capacitance when the high currents are used. The reason for this can be that a current offset is present in the measurement equipment during charges. This offset affects the results with low currents more than the high currents since it is a larger part of the total current at those times.

5 Modelling

Before any of the programs are tested, an initial model has to be constructed. This model is first described as an electric circuit in the Section 5.1. In order to create this circuit, the measurement data described in Section 4.1 is used. Since Simulink is the development environment mostly used at VTEC, the basic model is created in that program. The model is developed so that it results in reasonably accurate following of the available measurement curves.

5.1 Basic Model

A supercapacitor can be modelled using some standard circuit components as shown in Figure 29. This circuit design is used because a similar circuit is presented in the data sheet for the supercapacitor from EPCOS [19] and because of recommendations from the project supervisor. The circuit includes all the parts that are used in the basic model. Since it is a conceptual circuit more than an actual functioning circuit, it does not show how the switch or the capacitors are controlled. The two variable capacitances are nonlinearly varying with the voltage that is applied across the circuit, according to the relation found in the measurement data. This relation is described more in detail in Section 5.1.3 under the heading “Lookup Table for Nonlinear Capacitance” in the part about the capacitance.

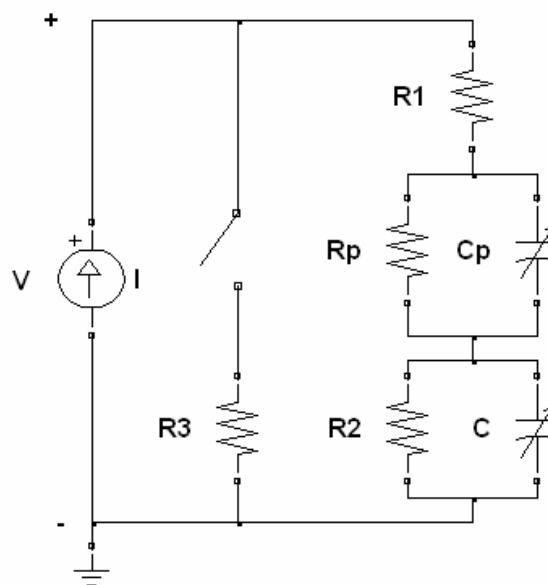


Figure 29: The basic circuit model of the supercapacitor.

The capacitance C is responsible for the most important phenomenon in the model. It determines how charge is handled in the circuit. The amount of energy stored and the rate of energy level variations are both determined mainly by the capacitance value. The resistance R_2 that is connected in parallel with the capacitor is meant to represent the self discharge effect. The series resistance R_1 represents the losses during charge and discharge. These losses occur because the conducting element in the supercapacitor has a resistance, so the connection is not ideal. The over voltage protection provided by R_3 and the switch controlling its connection to the circuit is necessary to prevent damage to the capacitor elements by

balancing the voltage level. The voltage balancing is needed because otherwise the voltage in one separate cell can increase higher than the others resulting in gassing or explosion. This voltage difference can occur if one cell has a lower capacitance than the others, since that results in more energy being stored. The resistance R_p and the capacitance C_p are included in the circuit to model some of the fast dynamics in the behaviour of the supercapacitor.

5.1.1 Simulink Model

Simulink is used to create a first model of the supercapacitor according to the basic circuit described above. Initial model testing is done with a simple circuit consisting of a resistance in series with a capacitance and resistance in parallel. This base circuit manages to show the basic function of the supercapacitor. By adding more components until the circuit described in Figure 29 is achieved, the accuracy of the model is improved. The Simulink model that is used as the basic model of the supercapacitor is shown in Figure 30. The relay block controls the switch that connects the balancing resistance R_3 to the circuit. Since that part of the circuit is supposed to keep the voltage over the supercapacitor from rising too high, it is set to be connected when the voltage goes above 42.84 V. It is disconnected when the voltage goes below 42.83 V. These voltage values are selected after studying the measured voltage curve where it can be seen that a significant derivative change occurs after a voltage peak at 42.83 V. This derivative change can be interpreted to occur because of the connection of the extra resistance since the increased load means that the voltage increases slower. The relay block in Simulink can only be set to have a turn on value that is larger than or equal to the turn off value which is why turn on is set to 42.84 V. The protection provided by this circuit is important because supercapacitors are sensitive to over voltage.

After studying the measured voltage curve it is found that it starts at an initial voltage level of 2.19 V. This initial value can be included in the model by setting the initial condition in one of the integrator blocks to 2.19. The abs-block that is in front of the lookup table is used to prevent negative voltages from being sent in. This is necessary because a negative voltage should correspond to the same capacitance value as the corresponding positive voltage. Otherwise the lookup table uses extrapolation to find a capacitance value that matches the negative voltage which is not correct.

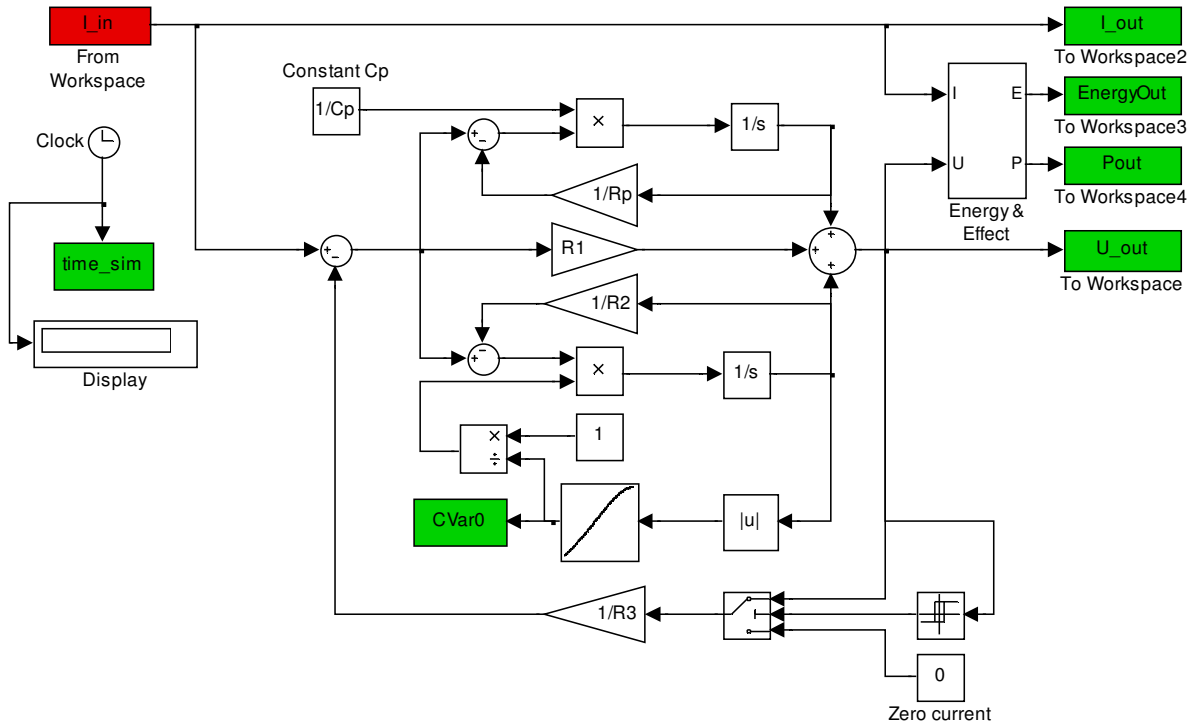


Figure 30: Basic circuit with variable capacitance value.

5.1.2 Component Value Tuning

When all the blocks of the basic model are connected correctly, they are assigned values according to what is found in the measurement data. The process of gathering these values is described in Section 5.1.3. By looking at how well the curve shape of the voltage received from the model matches the voltage from measurement data, the accuracy of the component values can be investigated. By altering the component values one at a time, the model is adapted to resemble the measurement curve as much as possible. When the best curve fit is achieved, the component values are recorded and they are all gathered in Table 3. No mathematical methods are used to find the most appropriate values, only qualitative testing by individual parameter alterations.

Table 3: Component values for basic model in Simulink. The capacitance value is included because it is used for initial tests, before the variable capacitance is implemented.

Component	R1 [mΩ]	R2 [kΩ]	R3 [mΩ]	Rp [mΩ]	C [F]	Cp [F]
Value	6	18	52	3	35	C/13

5.1.3 Component Value Determination

The component values are found by analysing measurement data from a test that had already been made on the supercapacitor. This data is displayed in Figure 20.

Main Capacitance

The capacitance value can be calculated in two different ways. The first method is to look at

the voltage derivative during charging of the supercapacitor. The relation between voltage derivative and the capacitance is

$$i(t) = C \frac{d}{dt} u(t) \quad (2)$$

where C is the capacitance [20]. Using this relation the capacitance can be calculated for different parts of the voltage curve. When high currents are used, other effects than the capacitance can affect the voltage level. These effects can cause the calculated capacitance value to be incorrect.

According to

$$Q = \int i(t) dt \quad (3)$$

where Q stands for charge, the charge in a capacitor can be calculated using the integral of the current during one charging cycle [21]. The capacitance value can then be calculated using

$$C = \frac{\Delta Q}{\Delta u} \quad (4)$$

where ΔQ and Δu represent differences in charge and voltage [22]. Since the energy level can be calculated using only one value on the voltage, only two points on the voltage curve are needed to be able to calculate the energy storage capability. This means that the voltage variation during the charge is not important to be able to determine the energy content in the supercapacitor.

The second method is to look at the energy stored in the supercapacitor. The main advantage of using this method is to avoid the effects of nonlinear capacitance during different charging levels. Then the expression

$$W = \frac{1}{2} C \cdot (U_2 - U_1)^2 \quad (5)$$

can be used to calculate the capacitance.

In the initial part of the test cycle, the supercapacitor is charged from almost empty to full charge. This part of the test cycle is shown in Figure 31, where the important points on the two curves have been marked.

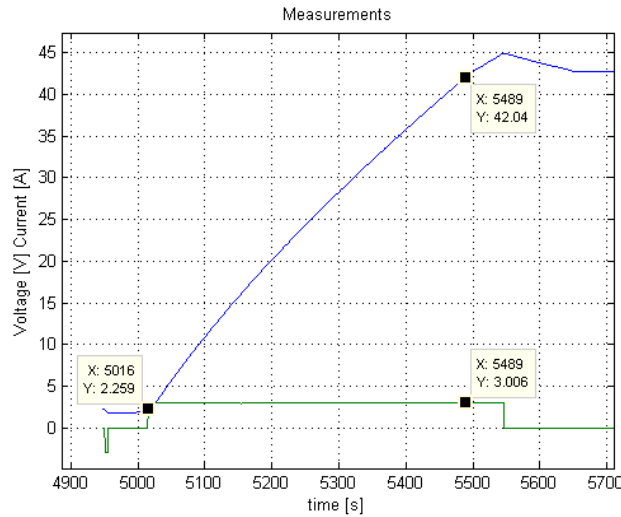


Figure 31: First measurement points on test cycle curves.

To obtain a good value on the capacitance, measurement points are taken at several places during the test cycle. All these measurement points are gathered in Table 4. Only the first test is shown with a figure, the other two tests are very similar and therefore only the values are presented in Table 4. When taking these measurement points it is important to note that the voltage drop occurring over the series resistance is constant during charging. This means that it does not affect the voltage difference measurement.

Table 4: Three tests to calculate the capacitance value in the circuit model.

Test nr	I [A]	T0 [s]	t1 [s]	u0 [V]	u1 [V]	Δt [s]	Δu [V]	Q [C]	C [F]
1	3.0	5016	5489	2.259	42.04	473	39.781	1419	35.6
2	60	7708	7733	1.88	45.07	25	43.19	1500	34.7
3	200	9360	9367	1.9	45.16	7	43.26	1400	32.4

Lookup Table for Nonlinear Capacitance

Since the capacitance varies with the voltage it is better to include this relation into the model than to simply have a constant C-value all the time. To do this the part of the measurement curve that is displayed in Figure 31 is analysed. This is the part of the measurement that reaches from 5018.13 to 5449.23 seconds. New vectors are created for time, current and voltage that only contain values from the analysed part. Using the new vectors the voltage vector can be plotted and the cubic fitting tool in MATLAB can find an equation describing the curve. The equation of the voltage curve is

$$u = 7.1138 \cdot 10^{-8} t^3 - 1.1657 \cdot 10^{-3} t^2 + 6.4391t - 11945 \quad (6)$$

and the derivative of that expression is then calculated by hand which results in

$$\dot{u} = 2.13414 \cdot 10^{-7} t^2 - 2.3314 \cdot 10^{-3} t + 6.4391 \quad (7)$$

where \dot{u} stands for the voltage derivative. Using (2) the capacitance value can be calculated for the different voltage levels. The voltage vector and the capacitance vector are connected to the same time vector, so any given element in the voltage vector corresponds to the calculated capacitance value for that voltage level. In the Simulink model described in Section 5.1.1 a lookup table is added, so the correct capacitance value can be used in the simulations. The capacitance curve that is the result of the input voltage to the lookup table is shown in Figure 32.

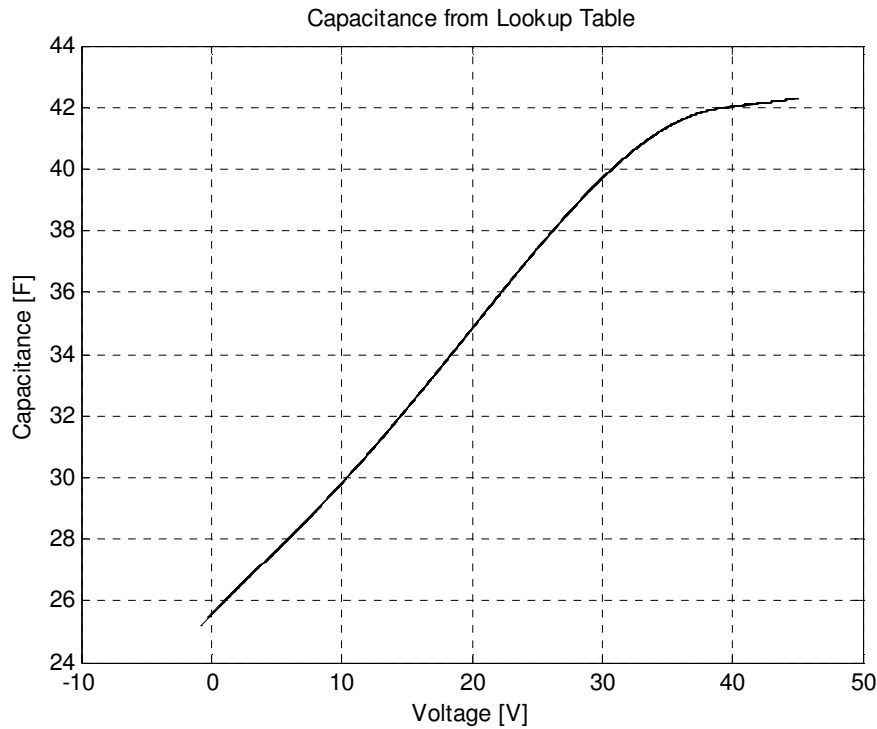


Figure 32: Capacitance curve received from the Lookup Table in Simulink.

Capacitance C_p

The value of the capacitance C_p is chosen to be one thirteenth of C . This value is chosen because the datasheet describing the supercapacitor that is being investigated suggested such a relation [19]. The impact of the C_p component in the model is very small, which means that a change in its value does not impact model accuracy much.

Resistance R_1

The value of the series resistance R_1 can be found by observing a rapid change in charging current and measuring the initial voltage variation. This voltage variation has a steeper inclination than the variation that occurs right after because of the supercapacitor discharge. The variation corresponds to the voltage drop over the resistance R_1 . The first value on R_1 is calculated using the voltage drop that occurs at the beginning of the final discharge. This is the first method used to determine a value on R_1 . To show which measurement points are selected, that part of the test cycle is shown in Figure 33. R_1 is also calculated using values from other parts of the test cycle. However, since these are similar to the first test their values are only displayed in Table 5. Using

$$R1 = \frac{\Delta u}{\Delta i} \tag{8}$$

R1 can be determined, if the influence of the resistance R2 is neglected. The variations in the acquired R1 values using method one can be because of the effect of the balancing resistor, which is engaged at the high voltage levels.

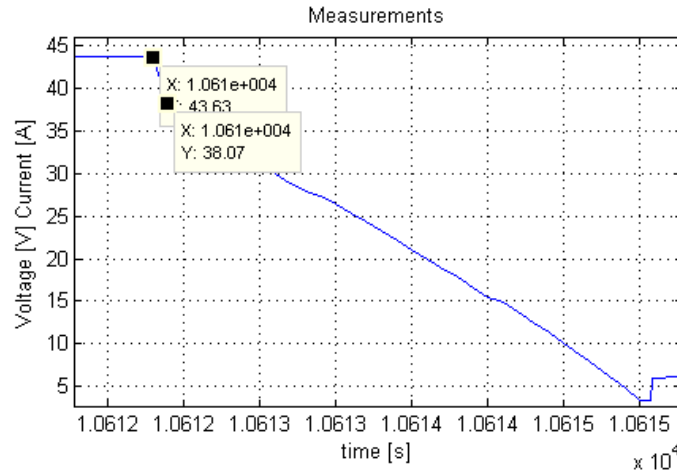


Figure 33: First method used to acquire a value on the R1 component.

Table 5: Values gathered from calculation of R1 value using method one.

Test nr	I [A]	t0 [s]	u0 [V]	u1 [V]	Δu [V]	R1 [m Ω]
1	-400	10610	43.63	38.07	5.56	13.9
2	-120	8968	43.71	41.76	1.95	16.2
3	-240	9788	43.71	40.41	3.3	13.7

Values on R1 can also be acquired by looking at the voltage variation that occurs after a discharge has ended. This second method to find a value on R1 uses data points like those shown in Figure 34. Values gathered from those tests are shown in Table 6.

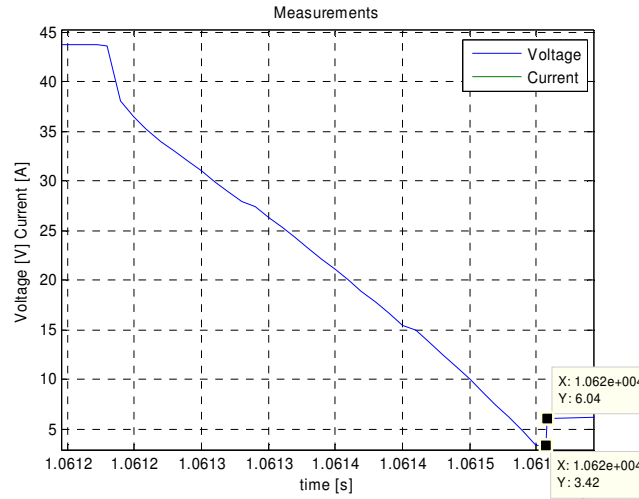


Figure 34: Second method used to acquire a value on the R1 component.

Table 6: Values gathered from calculation of R1 value using method two.

Test nr	I [A]	t0 [s]	u0 [V]	u1 [V]	Δu [V]	R1 [m Ω]
1	-400	10620	3.42	6.04	2.62	6.55
2	-240	9794	1.65	3.54	1.89	7.875
3	-120	8980	1.59	2.51	0.92	7.67
4	-60	8167	1.76	2.29	0.53	8.83

According to the data sheet [19] the value on the ESR should be 24 m Ω . When observing the measured curve, this value does not appear from the calculations. The value rather seems to be around 14 m Ω or 7 m Ω depending on where the values are taken from. One reason for this deviation can be that the value on the data sheet states the DC-resistance and not the ohmic part, which is what is found in the performed calculations. The deviation can also be because the resistance could be varying with voltage, because of sampling errors resulting in inaccurate value selection or because of the non ideal current step. One important source of errors for the measurements made using method two is the self heating that occurs when high currents are used.

Resistance R2

Since the self discharge is very low, the value of the R2 resistance is hard to acquire from measurement analysis. A slow discharge means that the resistance value must be high. Discussions with the project supervisor concluded that a good estimate value on the resistance is 18 k Ω .

Resistance R3

R3 is the name of the resistance used in the balancing circuit. The value on the resistance R3 can be calculated by measuring the voltage drop after the charging current has been turned off. In Figure 35 the first test is shown. The resistance value can be calculated using

$$R3 = \frac{\Delta t}{-\ln\left(\frac{v_1}{v_0}\right)C} \quad (9)$$

where v_0 and v_1 are the first and second voltage point. The test results from some voltage drops like these at different times of the test are shown in Table 3.

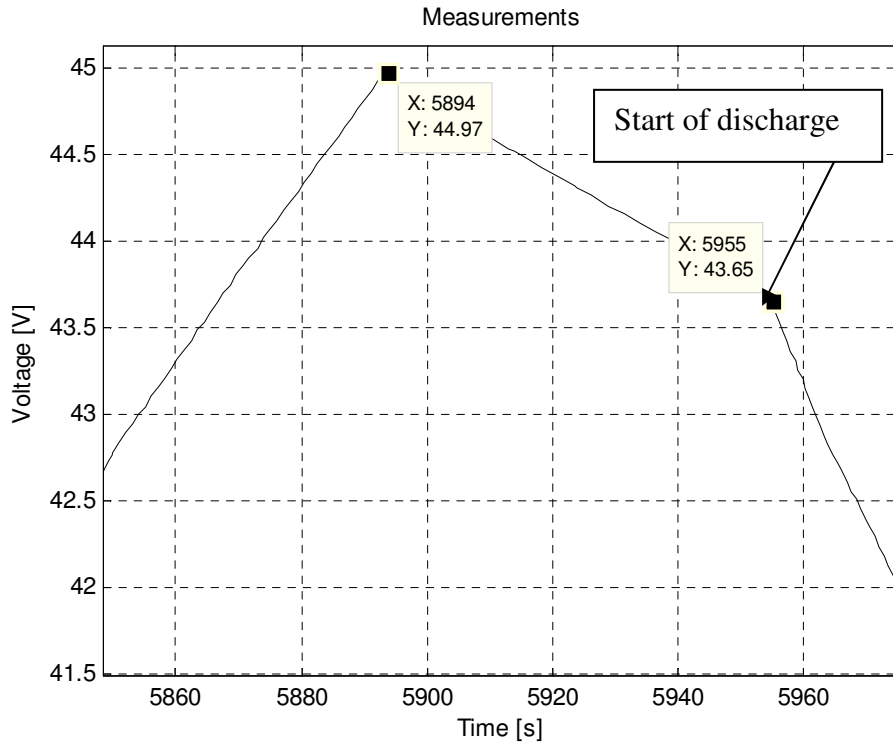


Figure 35: Test to acquire value on R3 component.

Table 7: Data from tests made to acquire a value on R3.

Test nr	t0 [s]	t1 [s]	u0 [V]	u1 [V]	u1/u0 [V]	Δt [s]	R3 [Ω]
1	5894	5955	44.97	43.65	0.97	61	57.22
2	7226	7286	44.96	43.71	0.972	60	60.36
3	8085	8143	44.91	43.71	0.973	58	60.54
4	8908	8968	44.97	43.71	0.971	60	58.25

Resistance Rp

The value of resistance Rp can not be determined by looking at measurement data because its effect on the curve shape is not visible enough. As described in Section 5.1.2 the value is instead found by testing different values to find the best match with measurement data.

5.2 Program Alternatives

After the basic model of the supercapacitor had been completed in Simulink, modelling could be initiated in the other program alternatives. In the following sections the modelling process and the finished models for these programs are presented. When modelling the supercapacitor some modelling alternatives are tested in the different programs. Two of them are included to enable comparison of their respective properties.

5.2.1 Nonlinear Capacitance

When modelling the supercapacitor there is one component that can be modelled in several different ways, the capacitance C . If the nonlinear behaviour of the capacitance is ignored it is sufficient to use a standard capacitance with a constant value in the circuit. If the nonlinear behaviour of the capacitance should be modelled this can be done in at least two different ways that both are based on the same idea. The idea is to use a controlled current source or a controlled voltage source to represent the nonlinear capacitance in the circuit. Before these simulation alternatives are selected, several other alternatives are tested and information regarding nonlinear modelling is consulted. The process of modelling using a controlled current source is explained since some parts of that modelling approach are used in a few of the program alternatives. However the approach with the controlled voltage source is chosen as main modelling type because it can be implemented in all the program alternatives and because it provides the best results. Modelling was also conducted using the controlled current source in some of the programs but because of its similarity to the controlled voltage source model these results are only presented in Appendix A – Controlled current source.

If a modelling alternative using only circuit elements is desired it is possible to use two capacitances connected in parallel with each other. One of the two circuit elements must model a capacitance with a value proportional to the applied voltage. If the other capacitance has a constant value the combination of the two can result in a nonlinear capacitance. A disadvantage with this method is that the exact nonlinear behaviour can not be modelled for more complex nonlinearities. This is the reason that this method is not used in the developed models in this project. An advantage is that it is easy to tune its behaviour.

Controlled Voltage Source

To model the nonlinear capacitance using a controlled voltage source the expression in (2) can be reversed which results in

$$u(t) = \int \frac{i(t)}{C} dt \quad (10)$$

which is the integral form of the expression. The controlled voltage source needs a controlling signal that represents the voltage over the nonlinear capacitance. To determine this voltage the current through the component must be known. Therefore the current must be measured when creating a model. The capacitance is determined using the voltage over the component according to the text under the subject “Lookup Table for Nonlinear Capacitance” in Section 5.1.3. Integration is then performed to reach the correct voltage value which is connected to the controlled voltage source.

The controlled voltage source is to prefer when the nonlinear capacitance is modelled. This is because the voltage represents a state variable in the model and in the solution of the equations the signal must pass an integrator. This leads to that an initial value of the voltage over the capacitance can be specified at the beginning of the simulation. Another advantage of using this method can be that some numerical errors, that can occur when a derivative is used, are avoided. These numerical errors can occur when fast current steps are used because they result in large derivative values. This does not create a problem when integration is used. The calculation of the integral only uses past data but the derivative requires future data which is not available during the simulation. This means that the derivative must be approximated which can create errors in the resulting values. Because of these advantages the controlled voltage source model appears to be the preferred alternative when modelling the supercapacitor.

Controlled Current Source

Using (2) a controlled current source can be created. To calculate the current through the nonlinear capacitance the voltage must first be measured. This voltage is used both to calculate the capacitance value using (10) and to calculate the current through the capacitance. According to (2) the voltage derivative must be calculated before it can be multiplied with the capacitance. The capacitance value must also be found but this can not be done using the method described for the controlled voltage source. This is because of some calculation problems that occurred in some of the programs. To solve this problem a polynomial had to be adapted to the original function that describes the relation between voltage and capacitance. The resulting polynomial is

$$C = p1 \cdot u^5 + p2 \cdot u^4 + p3 \cdot u^3 + p4 \cdot u^2 + p5 \cdot u + p6 \quad (11)$$

and the values of the coefficients are gathered in Table 8.

Table 8: Polynomial coefficients for capacitance expression.

Coefficient	p1	p2	p3	p4	p5	p6
Value	3.5357e-7	-4.0445e-5	1.3615e-3	-1.3502e-2	4.5682e-1	25.558

One thing to keep in mind when creating this kind of circuit is the direction of the current. If this is inverted, an error in the resulting voltage is introduced. One disadvantage of using a controlled current source is that the derivative part can introduce numerical errors. Another is that the initial voltage level cannot be set as easily as with the controlled voltage source. This is a serious disadvantage, since it means that it is not possible to set a start SOC for electric energy storage.

5.2.2 SimPowerSystems

In SimPowerSystems the supercapacitor can be modelled in some different ways. The alternative models are presented together with an explanation on how they are created.

Constant Capacitance

A basic circuit model for the supercapacitor made in SimPowerSystems is shown in Figure 36. The component values used in the simulation are those used in the initial test in Simulink that are found in Table 3. This is not the final basic circuit since the effect of a variable capacitance value is not included in this model. A constant capacitance value is used instead which also is a good approximation. By excluding the nonlinear components when modelling the supercapacitor the model is greatly simplified, since all the needed components are already available in the component libraries.

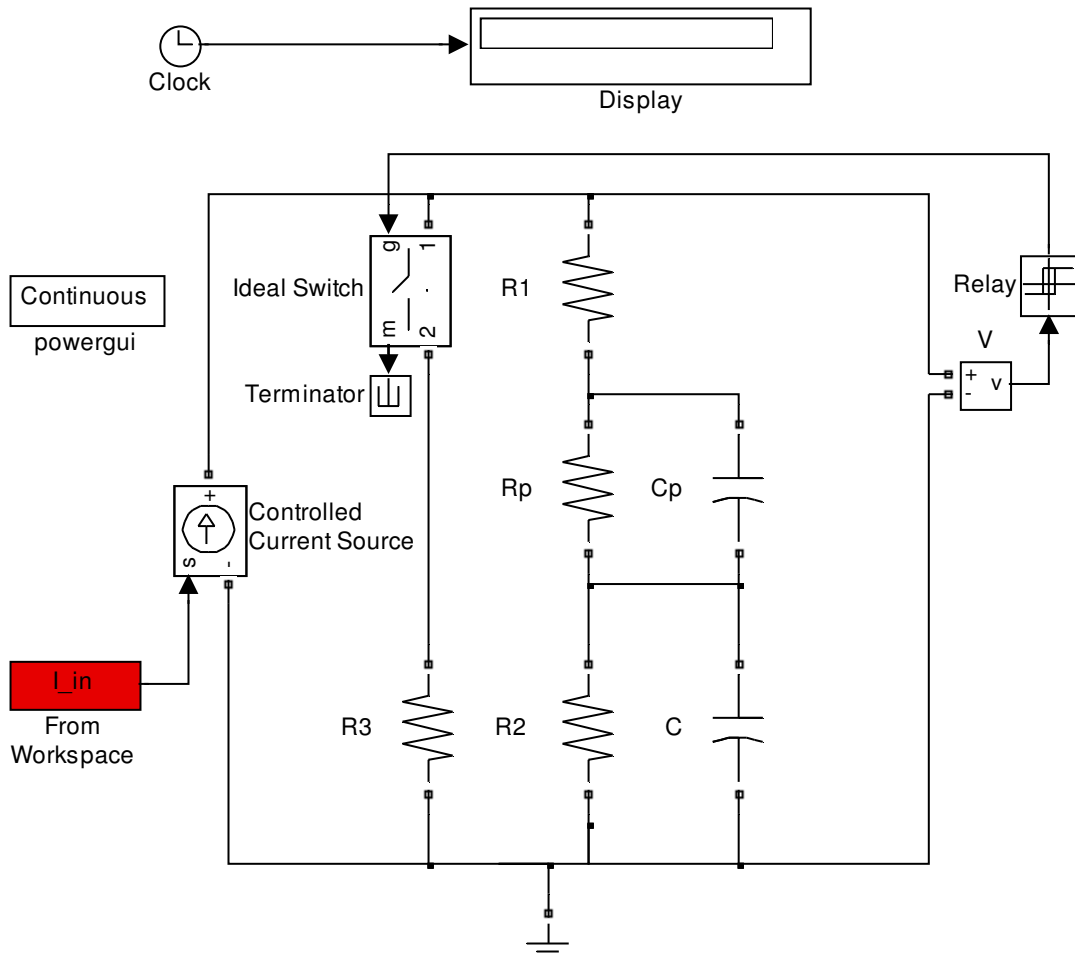


Figure 36: The basic model circuit for the supercapacitor made in SimPowerSystems.

Nonlinear capacitance

The nonlinear capacitance should vary with voltage. Several ideas on how to accomplish this are tested. In the MATLAB Help for SimPowerSystems there is a guide on how to implement a nonlinear inductance and also a nonlinear resistance. However there is no help on implementing a nonlinear capacitance.

In order to model a nonlinear capacitance, the current or voltage must first be measured using the appropriate measurement block. Using Simulink blocks various calculations can be done that give the current or voltage through the capacitance. The calculated value is then connected to either a controlled voltage source or a controlled current source, to enable the

reconnection to the electrical circuit in SimPowerSystems. In Figure 37 the model circuit that uses variable capacitance is shown. The modelling of the variable capacitance is gathered in a sub block where all the necessary components are found. For the three different modelling implementations described in the sections below, the contents of the sub block is shown in the respective section.

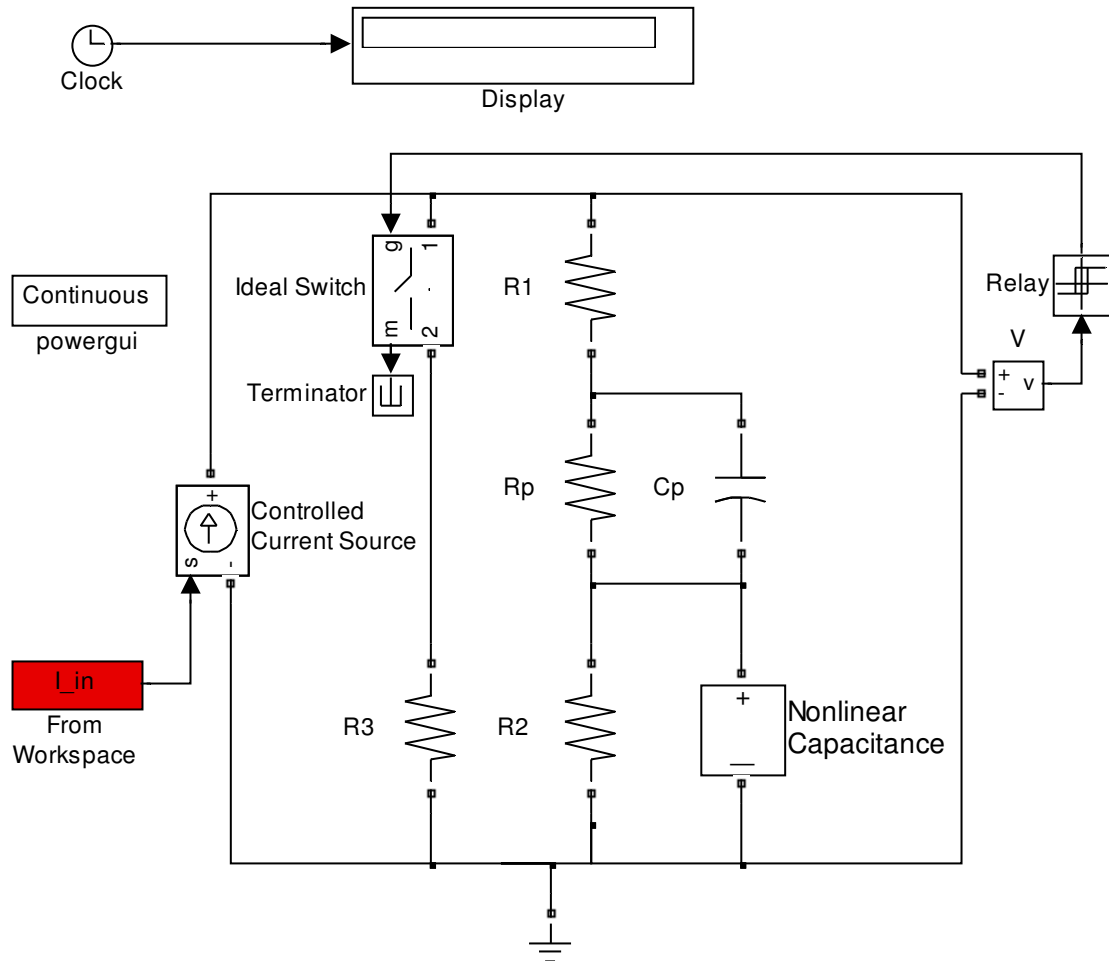


Figure 37: Model circuit in SimPowerSystems, using nonlinear capacitance.

Controlled Voltage Source

When implementing this kind of calculation it is important to consider the order in which the integration is performed. If the current is integrated first and then multiplied with the factor $1/C$ the resulting voltage is not correct. This error can be caused by the algebraic loop that is introduced when calculating in the described order. If the integration is moved so that it occurs after the multiplication with $1/C$ the result is correct. This placement of the integrator block also makes it possible to introduce an initial voltage over the capacitance. This is because the output of the integrator block represents the voltage over the capacitance. The contents of the sub block that models the nonlinear capacitance as a controlled voltage source is shown in Figure 38.

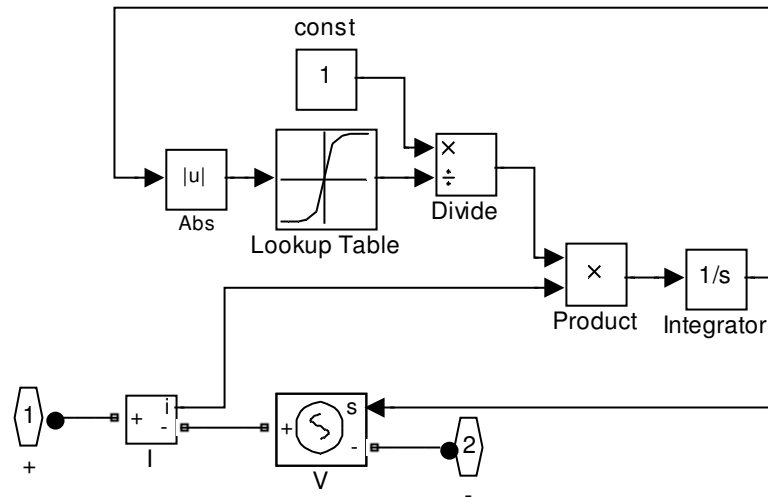


Figure 38: Sub block for nonlinear capacitance, using controlled voltage source.

5.2.3 OrCAD Capture

Modelling in OrCAD Capture is done using mainly electrical components. There are also some mathematical components available that can be used for various calculations. When modelling the supercapacitor both types of components are used. All of the solution alternatives use the same component to switch the current to the balancing resistance R3. As seen in the figures, turn off voltage is 42.75 V and turn on voltage is 42.83 V. These values are chosen after the measured voltage curve is studied. If there are many fast variations in a simulation one setting that might need to be altered is the one called transient time point iteration limit. If this parameter is increased the fast variations are handled more efficiently.

Constant Capacitance

The first modelling alternative is the one with constant capacitance values. That circuit is shown in Figure 39. Here the circuit consists entirely of regular electrical components. The connected current source is controlled by a data file in which the current from the previously conducted measurement had been stored.

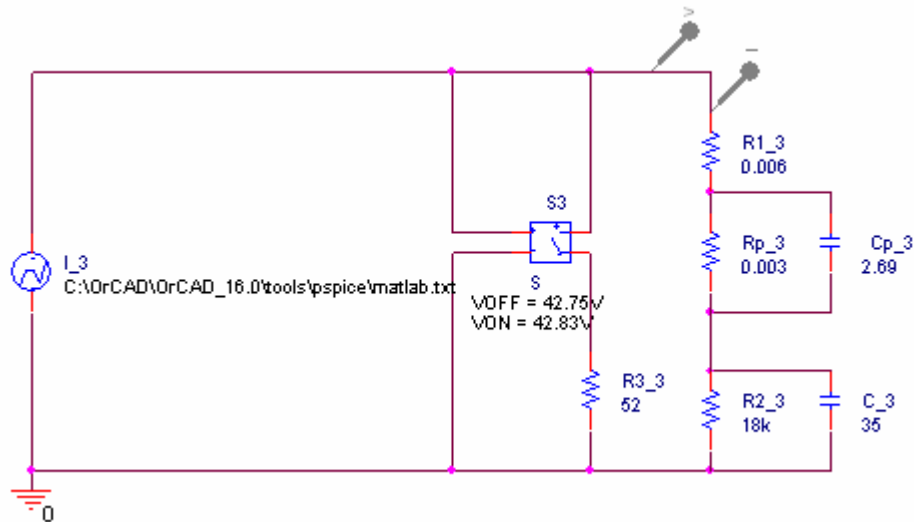


Figure 39: Supercapacitor model in Capture, using constant capacitance.

Nonlinear Capacitance

Just as in SimPowerSystems, there are some difficulties when trying to model the nonlinear capacitance. Attempts are made to find a suitable component to model the capacitance. Various random libraries are searched, hoping to find appropriate parts, however the search is not successful. A guide describing how to create a nonlinear capacitor model was found [23]. The ABM-library contains blocks such as expressions and mathematical operators which are needed to create the model of the nonlinear capacitance.

The polynomial described in (11) is used to get the capacitance value from the voltage. The expression in that equation is shown at the top of Figure 40. Information on how to write these expressions can be found in a reference guide for PSpice that is available online [24]. The information found there is necessary in order to be able to write the expression using no more than 132 characters, which is the upper limit for the block that is used. According to the guide referenced above, it is supposed to exist a component in which you are able to use a lookup table. That component was however not found which is why the approach with the polynomial is used instead.

Controlled Voltage Source

In Figure 40 the large capacitance in Figure 39 is replaced with a model of a nonlinear capacitance. In Figure 40 the voltage required for the controlled voltage source block is calculated using (10). This means that the current and capacitance are needed before their product can be integrated. The current is found using the block located at the top in Figure 40 which uses Ohms law to calculate the voltage through the R1 resistance. The reason that a polynomial is used to calculate the capacitance instead of using a lookup table is that the component designated for this purpose is difficult to implement which leads to its exclusion. A clear disadvantage with using a polynomial to accomplish this task is that the expression becomes long and complicated. This can be a source of errors when conducting simulations, especially if alterations are to be made frequently in the expression. The integrator block is to the right of the controlled voltage source and in it the initial voltage over the capacitance is set to 2.19 V. The component used for the integration is called INTEG which is available in the

ABM library. This component enables setting of the initial value which is a desired feature.

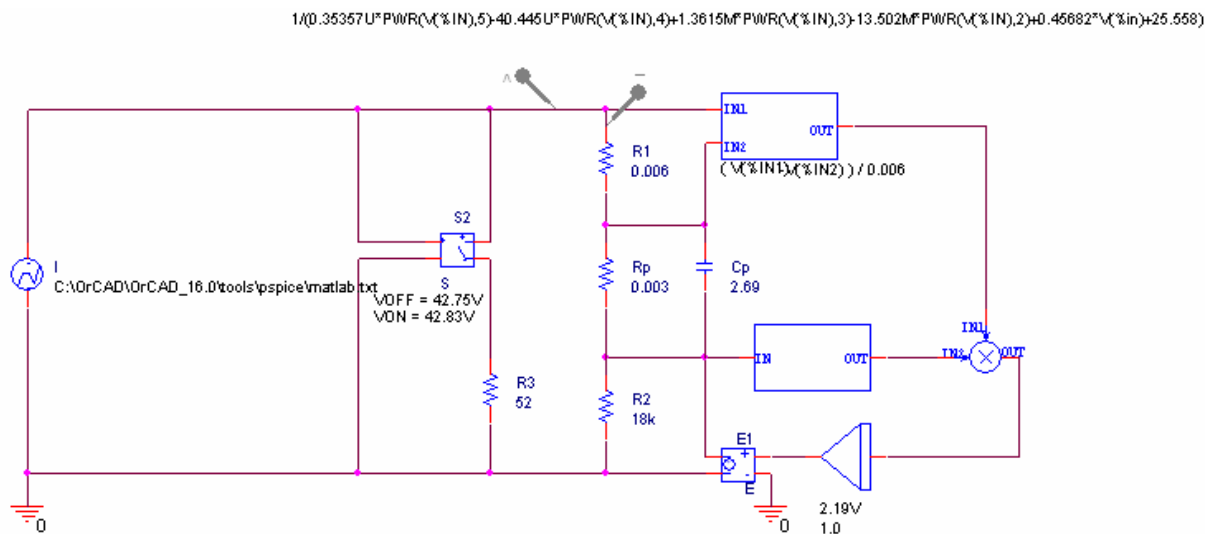


Figure 40: Model with nonlinear capacitance, using controlled voltage source.

5.2.4 PSCAD

Modelling in PSCAD is easy to get started with because of the well written help function which introduces the basic function of the program in a comprehensive way. The various components used in the model design are explained and some examples of circuits are presented. A useful feature in PSCAD is the ability to use labels to connect different parts of the model. An example of this in the model in Figure 41 is the label “I1” which is used both to connect the measured current to a graph function and to insert it in the voltage calculation. Another label is called “BRK” which is used inside of the switching component to control opening and closing times. An advantage with this feature is that unnecessary wiring can be avoided. The feature is also available in the other tested programs except for PLECS and Dymola. One thing that needs to be considered when modelling in PSCAD is that the wires that connect the different components do not always stay connected if they are moved. This means that if some components must be moved, rewiring is sometimes necessary.

Constant capacitance

When creating a model using constant capacitance only standard electrical components are needed. All the component values are set directly in the circuit for each component and the values can be seen in Figure 41. The switching voltage levels are set in the hysteresis block that is located to the left of the BRK label. The switch is closed when the voltage reaches 42.84 V and it is opened when the voltage falls below 42.83 V. When measuring the voltage and current in the model circuit two special measurement components are used. The voltage measurement component is connected to any point in the circuit and the current measurement component needs to be included in the circuit so the current flows through it. Both these components use labels to define the recorded values. These can then be used in other parts of the circuit or become connected to a graph block that can be used to present the values in a plot.

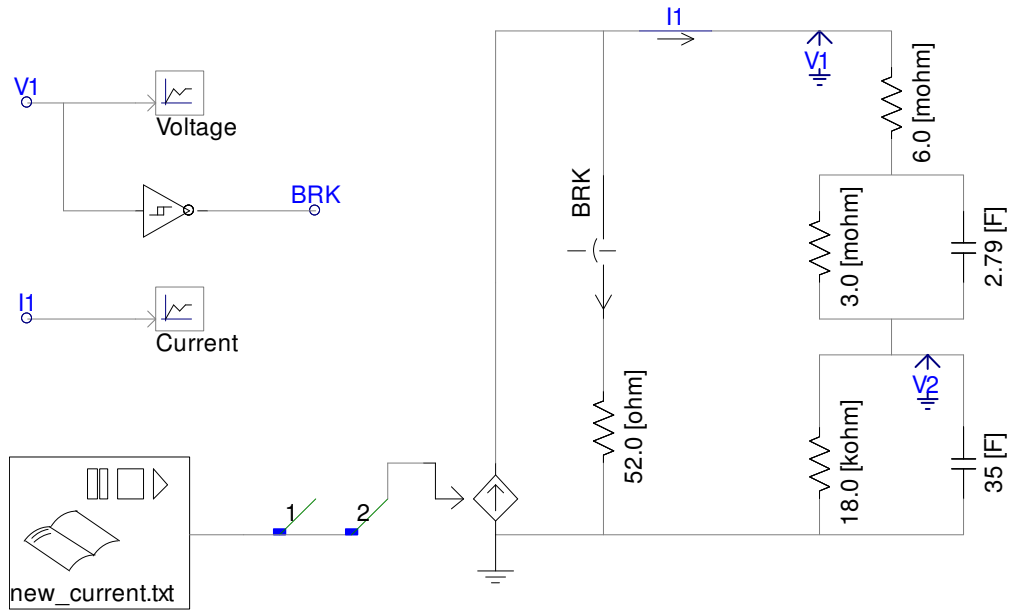


Figure 41: Model made in PSCAD, using constant capacitance.

Nonlinear Capacitance

In PSCAD the nonlinear capacitance is only modelled using the controlled voltage source. The controlled current source is omitted because of problems with the derivative calculation that is necessary to perform to receive the current level.

Controlled Voltage Source

The model using a nonlinear capacitance that is modelled using a controlled voltage source is shown in Figure 42. The capacitance value is calculated using the expression from (11). Then the voltage is calculated in the same way as described for the model in SimPowerSystems. An attempt is made to use a component that function as a lookup table to find the capacitance value which is not successful. That is why the alternative with the polynomial is chosen instead. Since no component is found that can calculate this function value directly it is implemented using several separate mathematical blocks. The integrator component also includes a possibility to set the initial voltage which improves the model accuracy.

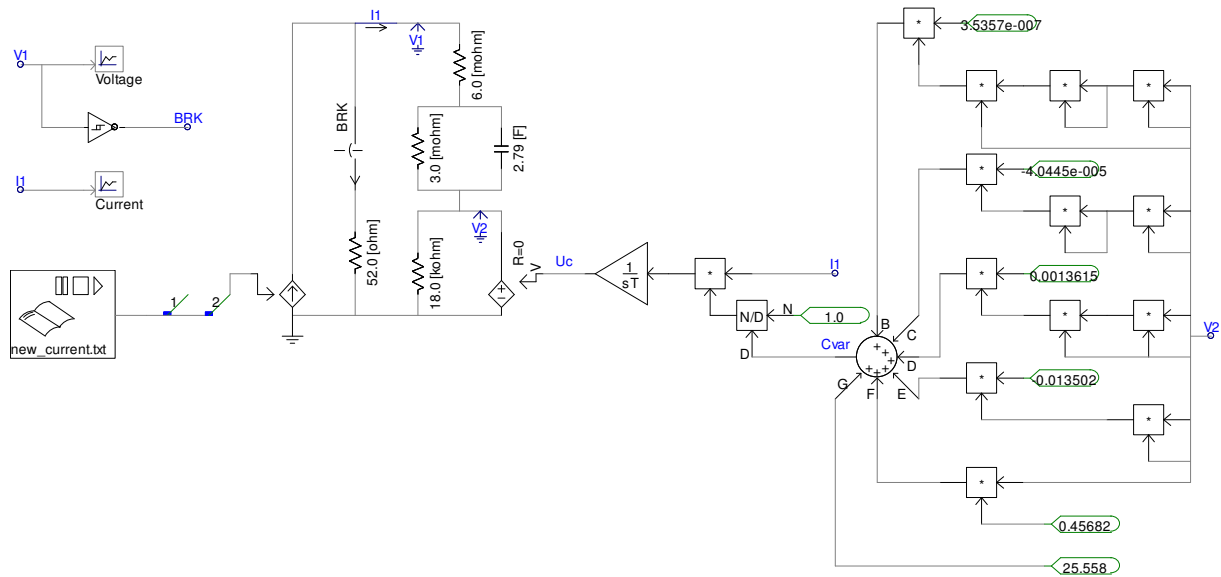


Figure 42: Model made in PSCAD, using controlled voltage source.

5.2.5 Saber

The process of importing the current from an external file into Saber is done in a somewhat different way than in most other programs. Here a lookup table that has the simulation time as input is used to create a signal that can be used by the controlled current source that is connected to the main circuit. The type of block that is used to convert the signal from the lookup table to a current that can be used in the circuit is a type of component that is available in many different variations in Saber. These conversion blocks can convert between ordinary control values and electrical, mechanical and hydraulic systems.

Constant Capacitance

The electric circuit that represents the linear model is shown in Figure 43. The values used on the passive elements are the same as those that have been used previously in all the other programs. The input current is connected to the circuit using a specific interface block that converts the data output from the lookup table to a current that can be directly connected to the circuit. The switch is configured to turn on when the voltage reaches 42.84 V and then turn off when it falls below 42.82 V. To set this value a limit voltage of 42.83 and a hysteresis of 0.1 is selected.

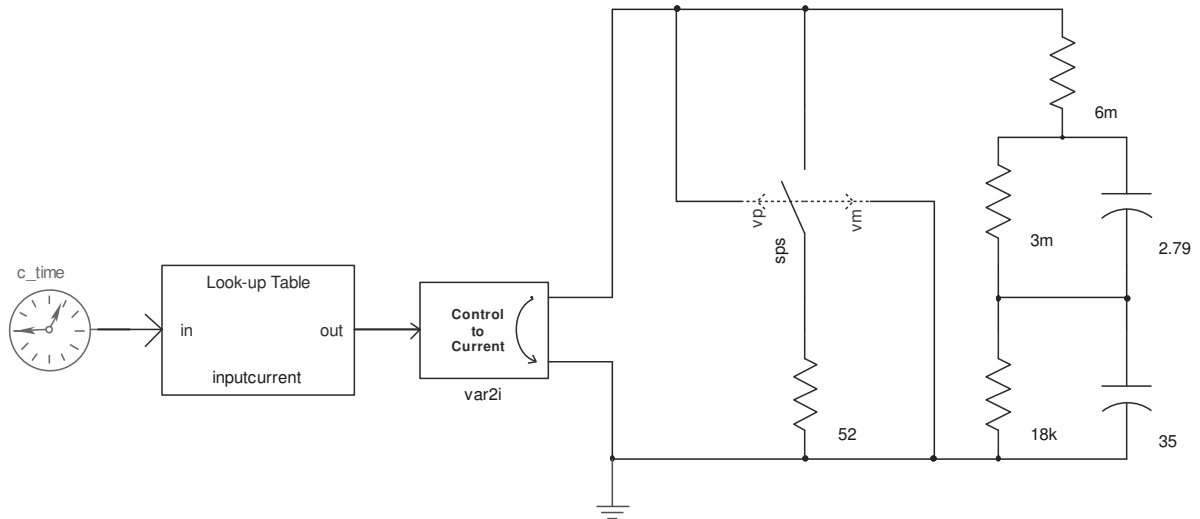


Figure 43: Linear model of supercapacitor in Saber.

Nonlinear Capacitance

Modelling the nonlinear capacitance in Saber can be done in three different ways. The first two are the same that are used in the other programs but the third is special in that it uses only one component to model the nonlinear capacitance instead of several that is the case for the two other models. This extra model is included in the testing procedure in order to compare the differences between the different model types.

Polynomial Model

Figure 44 shows the model that has a nonlinear capacitance part that is directly defined using a polynomial that describes the relation between voltage and capacitance. The values presented below the main electrical circuit are the coefficients of the polynomial. They are arranged so the coefficient that belongs to the lowest order is presented first and the following values are of increasing order. The polynomial that is used is the one that is defined in (11). Advantages of using this type of component are that it simplifies the circuit design and provides a better overview of the entire model circuit. A possible disadvantage is that it might be less accurate than the other modelling alternatives.

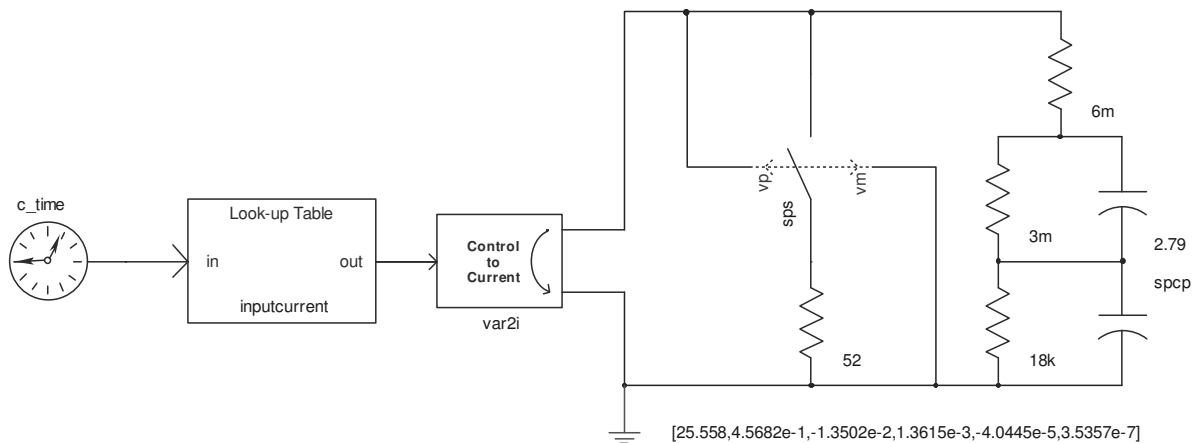


Figure 44: Model containing the nonlinear capacitance defined by a polynomial.

Controlled Voltage Source

To be able to compare the results in Saber with those received using the other program alternatives the nonlinear capacitance is also modelled using both a controlled voltage and a controlled current source. When creating models using these two approaches, the same modelling process as described previously is used. The model for the controlled voltage source is shown in Figure 45. Instead of the more common measurement and source components used in other programs, Saber uses interface blocks to measure currents and voltages as well as connecting signal values to the circuit. The other components used in the model of the nonlinear capacitance are similar to those used in other programs.

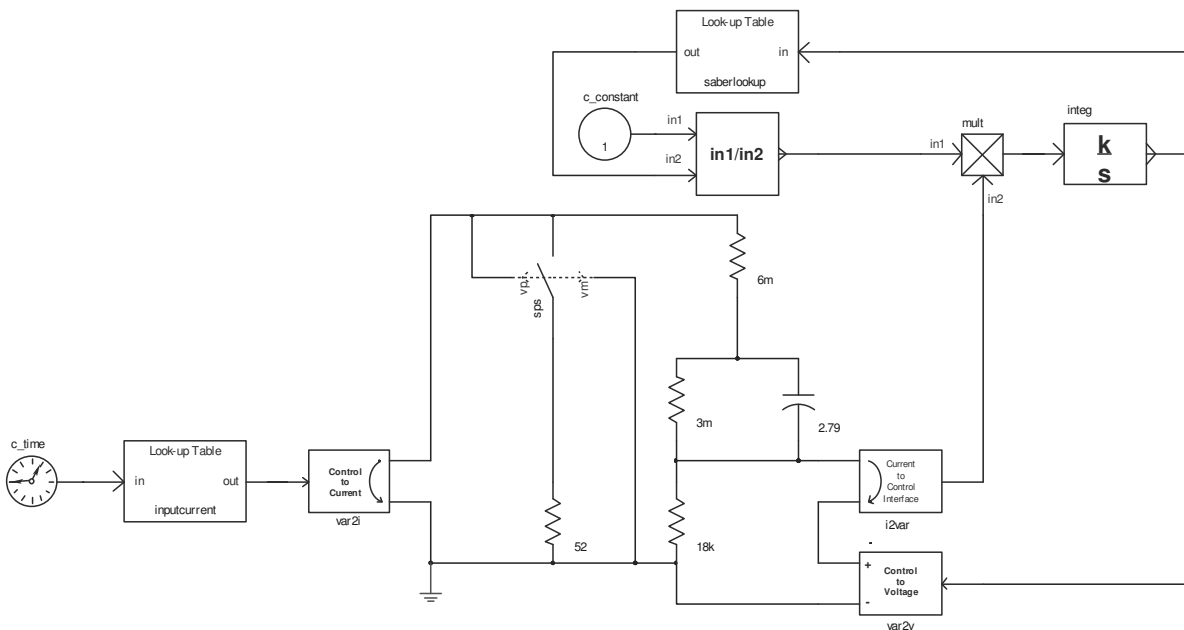


Figure 45: Nonlinear capacitance model using a controlled voltage source.

5.2.6 PLECS

When creating a model in PLECS one thing that needs to be considered is how the signals should be connected. Since calculations are needed to create a complete model of the supercapacitor some signals must be sent out so they can be used in calculations and then the results must be sent back in to the model in PLECS.

Constant Capacitance

Figure 46 shows the linear model of the supercapacitor. There the input and output signals are clearly visible. The only signal processing that is needed in this model is the control of the switch. To do this the voltage is sent out to Simulink where it goes through a relay block that results in a control signal which can be sent back in to PLECS.

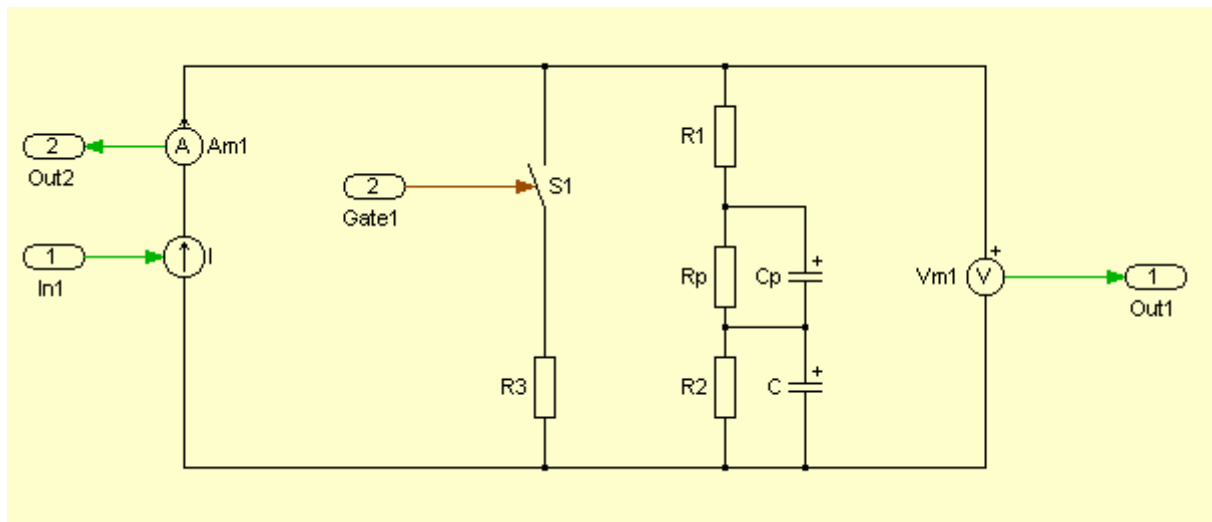


Figure 46: Linear model of the supercapacitor.

Nonlinear Capacitance

The nonlinear capacitance requires more calculations than the model with a constant capacitance. These must also be done in Simulink outside of the circuit model in PLECS.

Controlled Voltage Source

The nonlinear capacitance uses a controlled voltage source in the same way as described for SimPowerSystems. The calculation of the voltage for this source requires a current. This current is measured in the circuit and sent out to Simulink where the necessary calculations are done and then the voltage is sent in to the controlled voltage source. The calculation of this voltage is not displayed because it is done just like the one shown in Figure 38. The only difference is that the measurement block and the controlled voltage source in that figure is not used for the calculation in PLECS.

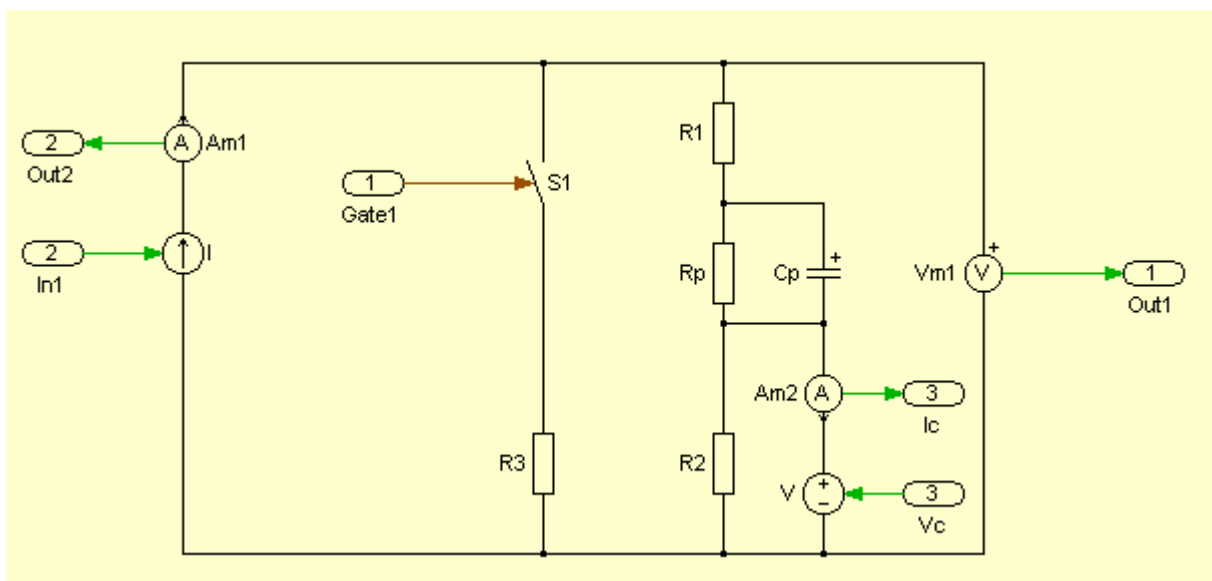


Figure 47: Nonlinear model using controlled voltage source.

5.2.7 Dymola

To build the supercapacitor model standard electrical components are connected to each other. To import the current signal a controlled current source is used which receives its data from a TimeTable component. The input to the component should be a matrix stored in the .txt or .mat format. The data stored in .mat format needs to be stored using version 4 of the MATLAB file type to enable import into Dymola. Data files that are not stored in the correct version are tested and they do not function. However there are also problems when trying to import data saved in the .txt-format so only .mat version 4 files are used. To import data also takes a long time when the data file size is large. The data file that is used as input to control the current source has 5048 rows. That takes about five minutes to import to the TimeTable component which is too long. To avoid the problems associated with import of data to table components it is possible to link the components directly to data files of .mat version 4 format. Then the data is accessed directly in the respective files instead of importing the data to the model file. The simulation performance is improved when using this approach and the model is simplified. It is also possible to use .csv-files for this purpose according to the Modelica documentation. The ability to enter data points manually using the TimeTable component can still be useful for smaller data sets when lookup tables are to be created.

Constant Capacitance

When the current is connected to the model circuit the remaining components are placed out in the same way used in the other programs. The components are easy to find and connect to each other. One detail when connecting the components is important to keep in mind during circuit design. The wire connections can only be made from one node to another node, they can not be made from one node to some point on a wire. If that is attempted the components do not get an adequate connection resulting in model errors. The switch has a threshold voltage of 42.83 V. The supercapacitor model using constant capacitance is shown in Figure 48.

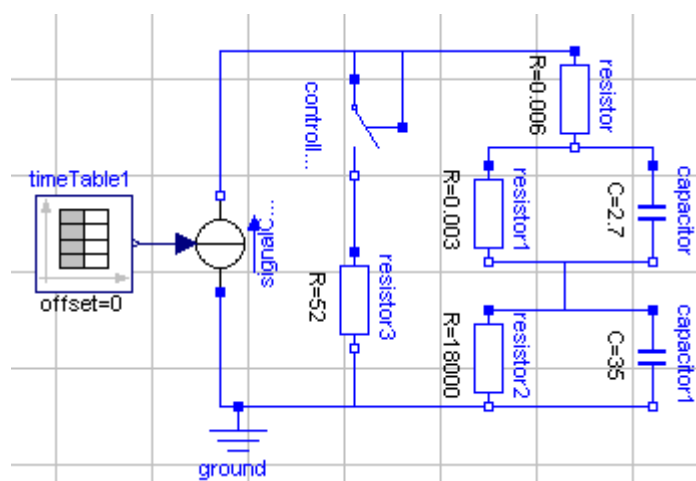


Figure 48: Supercapacitor model using constant capacitance.

Nonlinear Capacitance

The two standard approaches with the controlled sources are used to model the nonlinear capacitance. The components used for this part of the model are similar to those used by other programs even if they look slightly different in this program.

Controlled Voltage Source

The modelled supercapacitor using controlled voltage source in Dymola is shown in Figure 49. The main circuit components are still the same as for the constant capacitance model but below those are the extra components that are used for the calculation of the voltage over the nonlinear capacitance. The component `combiTable1D` works as a lookup table in Simulink when finding the correct capacitance for a given voltage level. The integrator gain is set to one which can be seen below the component where `k` is set to one.

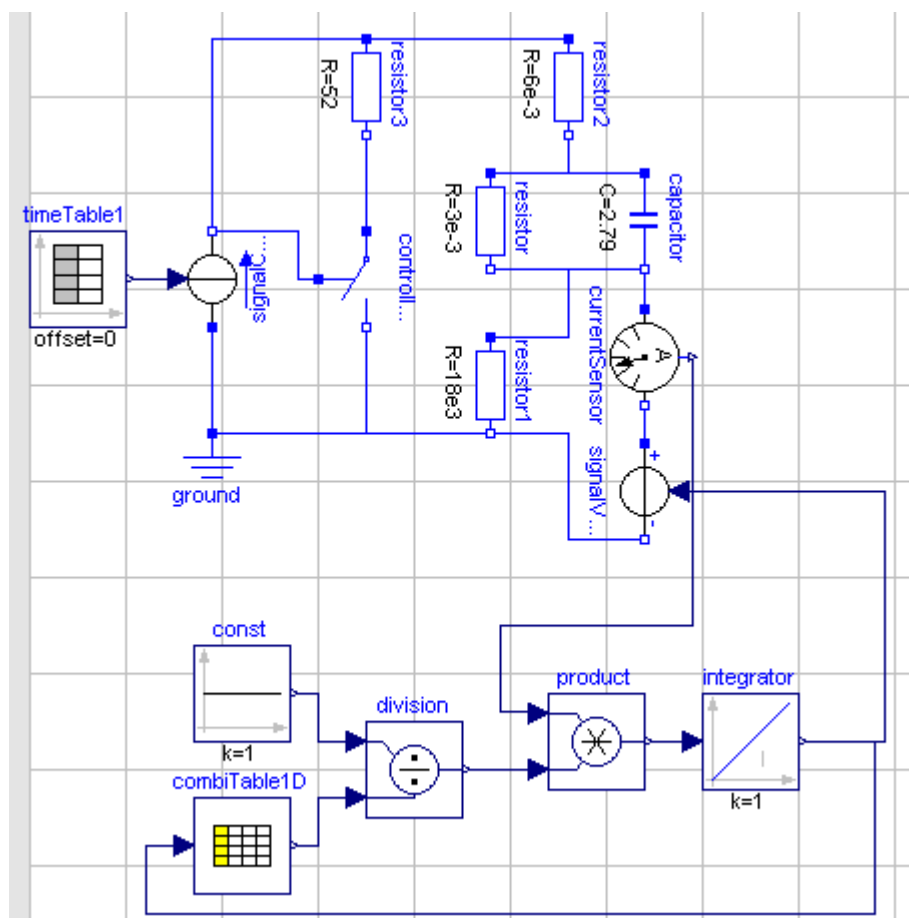


Figure 49: Supercapacitor model using controlled voltage source.

5.3 Search for a New Model Structure

The supercapacitor model used in the modelling during the initial program tests is based on information in a data sheet from EPCOS. For the creation of a more advanced model, it is decided to leave the old model structure. The main reason for this is that the old version does not model the supercapacitor behaviour well enough but also because it is hard to determine

the model parameters. The old version needs improvements when it comes to modelling the supercapacitor dynamics, which can be seen after charge- and discharge pulses. Another large flaw in the old model is that it does not include any temperature effects. This means that it can not be used to model the supercapacitor behaviour in an environment where the surrounding temperature changes over time.

After some research, a model called Two Branches model is found and it is described in more detail in Section 5.3.1. The two branches model is a simple model that is not able to reach the desired accuracy requirements but can still be interesting because of its simplicity. After some tests with this model it is decided to develop a more complex model which includes the ability to model both the supercapacitor dynamics and its thermal behaviour. Because of its functionality it is given the name “Dynamic Temperature Dependent Supercapacitor Model” (DTDSM) and it is described in detail in Section 5.3.2.

5.3.1 Two Branches Model

This model is proposed in a report by R. Faranda, M. Gallina and D.T. Son [1]. The model claims to be easier to use compared to other model types available for supercapacitor modelling, while still reaching good model accuracy. The simplicity of the model lies in its small amount of components and the documented method used to find parameter values. All that is supposed to be needed to determine all the model parameters is one measurement from which four data points should be recorded. This measurement consists of a single charge of the supercapacitor from being completely discharged to being fully charged. When full charge has been reached, the charging current is turned off and the voltage level is measured during 20 minutes. The circuit design of this model is shown in Figure 50.

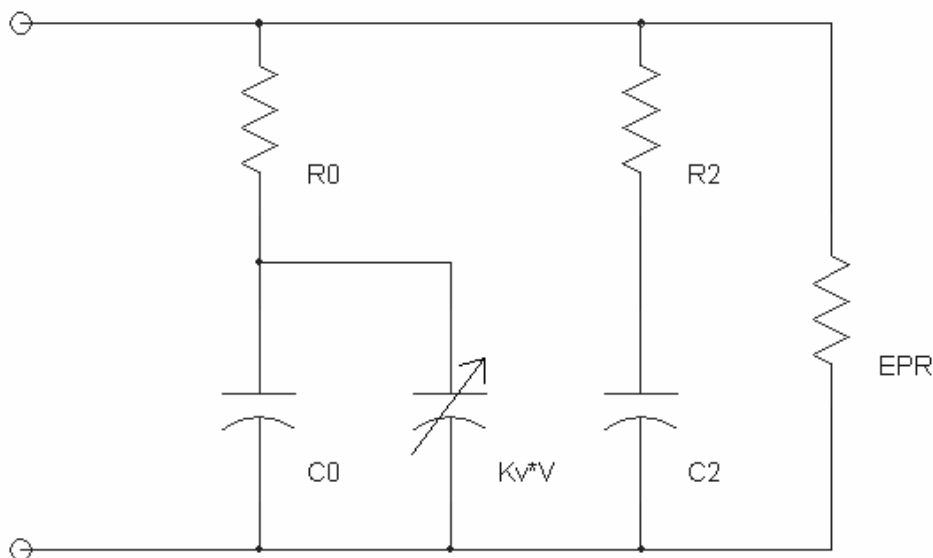


Figure 50: Two Branches model.

The process of finding the parameter values is described in the report that introduces the concept of the Two Branches model. The most important steps of this process are presented in the following text. The value of the series resistance R_0 can be found by looking at the initial

voltage rise at the start of the charging. At this time the capacitance is assumed to be discharged so all the voltage rise is over R0. Then it is possible to use

$$R0 = \frac{\Delta V}{I} \quad (12)$$

to calculate the resistance value. Using equations describing the variable capacitance and thus creating a link between voltage and current, two expressions are found which are used to find the values C0 and Kv. C0 is calculated using

$$C0 = \left[\frac{t_1}{V_1} - \frac{V_1 t_2 - t_1 V_2}{V_2^2 - V_1 V_2} \right] I \quad (13)$$

and Kv is calculated using

$$Kv = 2 \left[\frac{V_1 t_2 - t_1 V_2}{V_1 V_2^2 - V_1^2 V_2} \right] I \quad (14)$$

where the two times and voltages are from two points of the voltage curve during the charge. The two points should be spaced apart so they cover a large part of the rising curve shape but not so far that they are located on either the voltage rise in the beginning or the voltage drop in the end.

The next parameters to be determined are R2 and C2. Since these components are used to model the dynamic behaviour of the supercapacitor, a time constant which describes that behaviour must be chosen. In the report which introduces the Two Branch model a time constant of 240 seconds is chosen after some tests with different values, so that is used in the initial tests here as well. The choice of time constant determines how the model behaves during the slow self discharge after the initial charge. This means that by changing the value on the time constant that behaviour can be altered. To determine the value on C2 an expression describing the charge level is used which is

$$Q_{tot} = I_c T_c = C_2 V_{2f} + \left(C_0 + \frac{k_v}{2} V_{2f} \right) V_{2f} \quad (15)$$

and using this expression, C2 can be calculated since all the other parameter values are known, if V2f is the voltage at the time that is three times the time constant (720 seconds).

Since the time constant is calculated by multiplying R2 with C2, the value on R2 can be found after the value of C2 has been calculated. Then all the parameter values used in the Two Branches model are known and it can be used for simulations. The locations of the

measurement points used for the model are pointed out in Figure 51 which displays the measurement curve for the supercapacitor ESSP08.

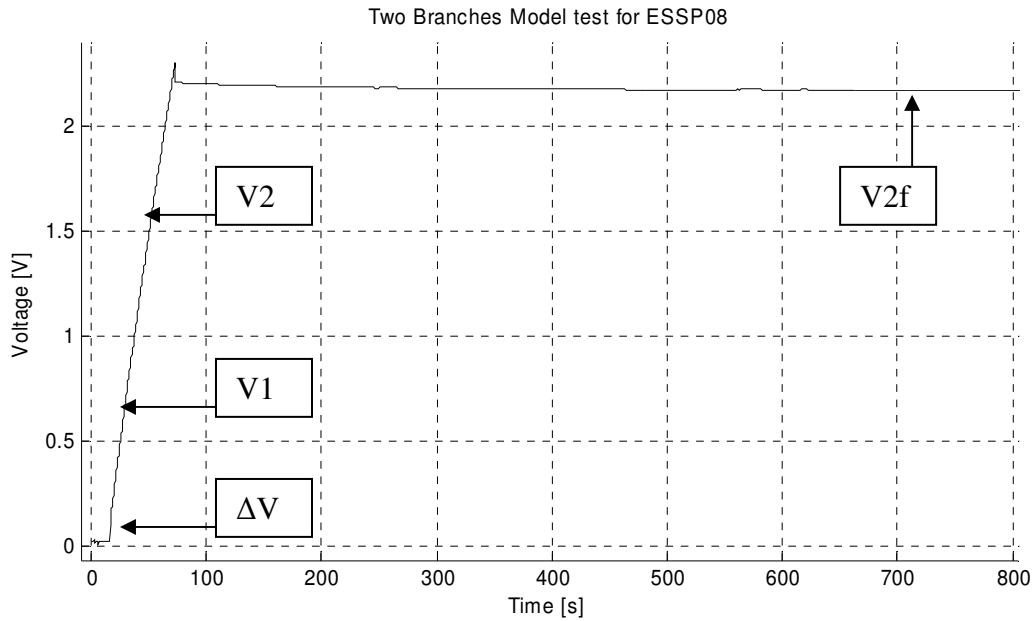


Figure 51: Measurement points used to characterize the Two Branches model.

Some details are worth noting when gathering data from the various points in the curve. The two data points that are recorded on the voltage rise during charging, have a voltage level that contains the voltage over the series resistance in the supercapacitor. To compensate for this the voltage recorded for the two points must be subtracted by the value of the voltage drop. Another important observation concerning the recorded points is that in order for the time values to be correct they must be adjusted so the start time is when the initial voltage rise has been completed and the voltage over the supercapacitor starts to increase.

When this approach is attempted using a measurement conducted on one of the supercapacitors the model results with a rather bad accuracy. However after some tuning of the parameter values a good model behaviour is acquired. The changes in parameter values are mostly done on R2 and C2 since it appeared that the time constant had been chosen too small. The Two Branches model implementation in Simulink can be seen in Figure 52.

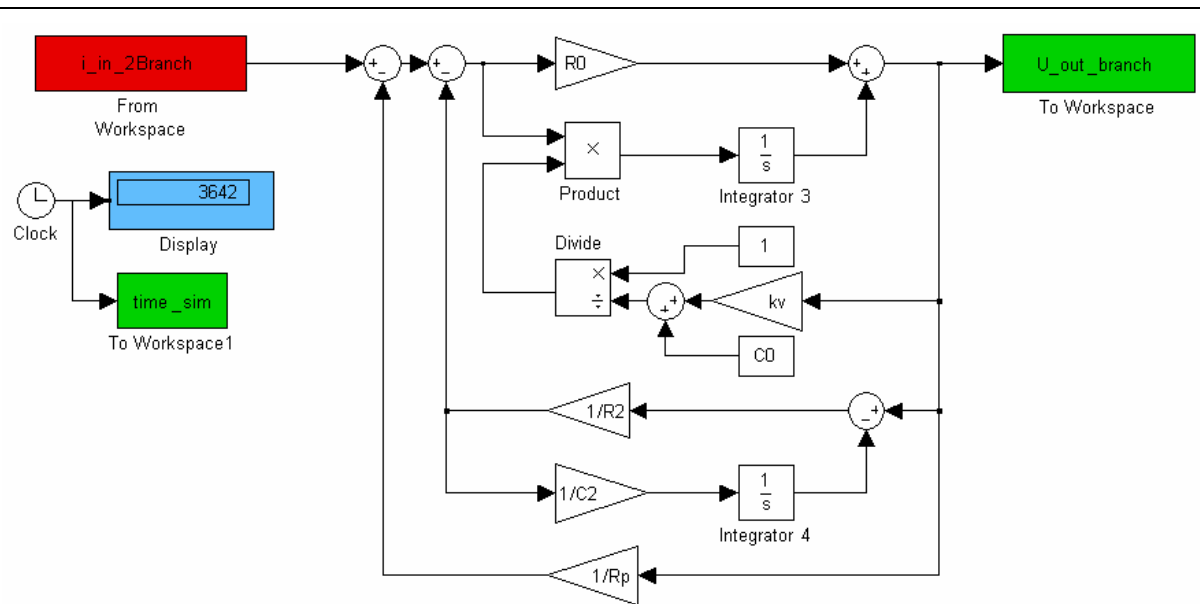


Figure 52: The Two Branches model implemented in Simulink.

5.3.2 DTDSM

The final model includes temperature dependent components and is given the name DTDSM which stands for Dynamic Temperature Dependent Supercapacitor Model. The model is based on the model structure in Figure 53. The capacitance named C is the main energy storage component and the two parallel branches ($C2$ and $C3$) are supposed to model the dynamic behaviour. By altering the component values in these two parts their time constants change which affects how fast they charge and discharge.

One disadvantage with the new model is that it is not able to model the dynamics with time constants that are too long. To be able to do this yet another branch could be added but this would also increase the model complexity which means increased computation time. In order to model slow dynamics a component that releases a small amount of energy under a long time is needed. This can be accomplished by setting for example $R3$ in the model to a large value and also use a large capacitance value on $C3$. However, then some errors might occur in the modelling of some part of the faster dynamics. This means that when creating a model with any number of branches the main concern is how to choose the model parameters so it models the most important behavioural features.

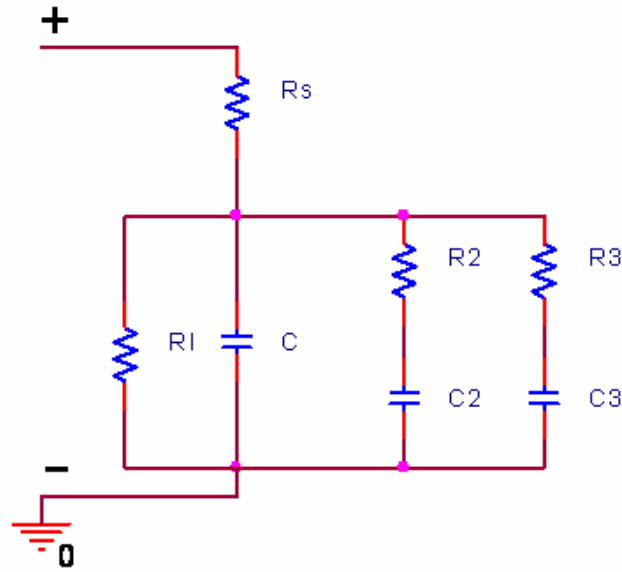


Figure 53: DTDSM electric circuit.

The subsystems describe different properties of the supercapacitor. The model includes a temperature dependency which introduces an increased level of complexity compared to the simpler model used in the initial tests. The DTDSM contains several subsystems and can be seen in Figure 54 where it is modelled in Simulink. The various parts of the circuit modelled are placed in three separate sub blocks in order to improve the overview and simplify the signal connections needed in the model. The contents of the blocks are displayed in Appendix B – DTDSM Figures.

The capacitance curve used to create the lookup table in the model is the curve for discharge from the top voltage level using 10 A current for the ESSP08. Values on the constant parameters in the DTDSM model are gathered in Table 9. The remaining components in the model (C and R_s) are nonlinear and therefore do not have simple parameter values. Their nonlinear behaviours are described further in their respective sections of the model description.

Table 9: Parameter values for the DTDSM adapted to ESSP08.

Parameter	$R1$ [k Ω]	$C2$ [F]	$C3$ [F]	$R2$ [Ω]	$R3$ [Ω]
Value	15	80	145	0.5	10

The values stated are only valid for ESSP08. Values for the other supercapacitors are not presented since no simulations are conducted using the measurement results gathered from these supercapacitors.

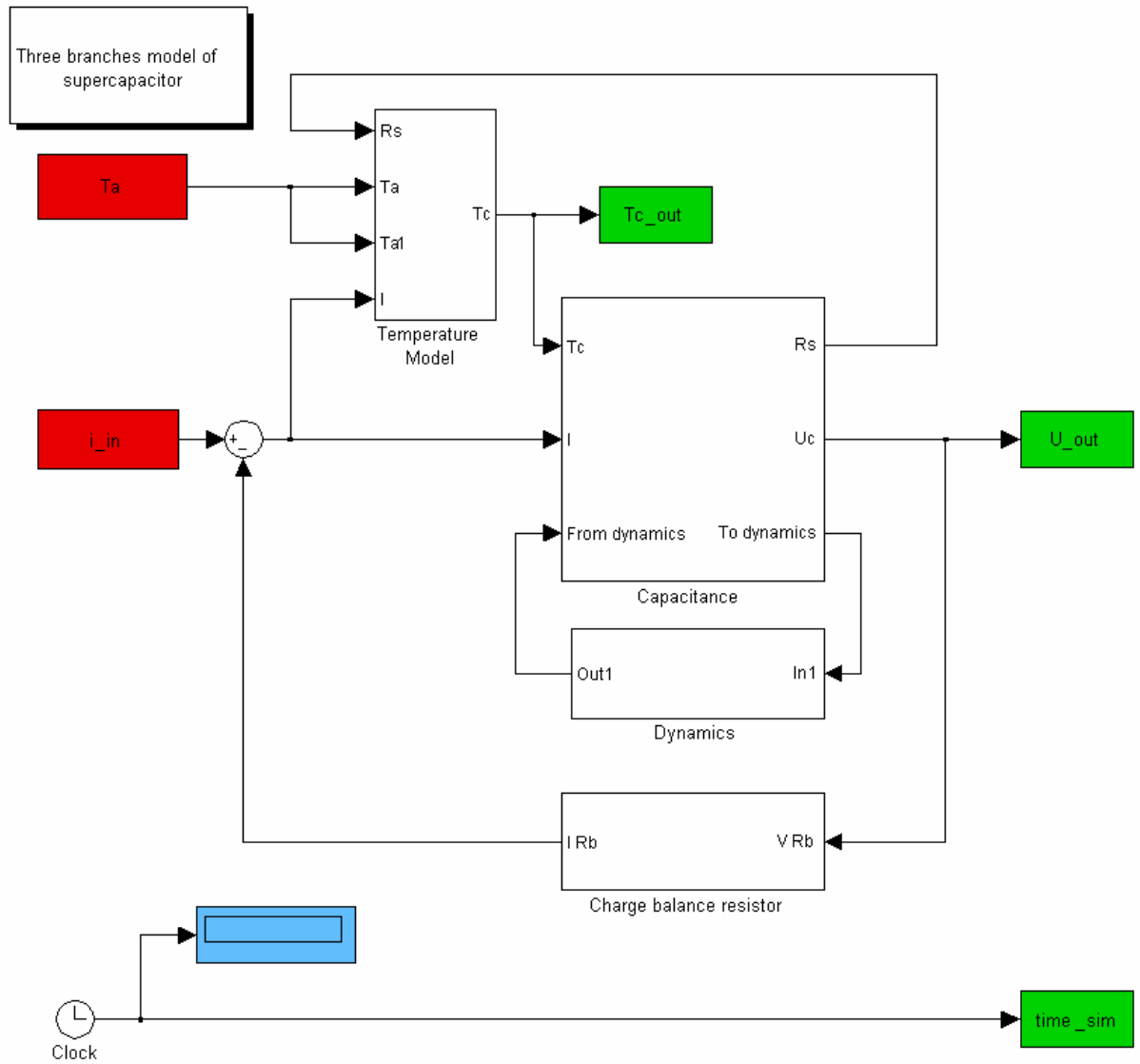


Figure 54: DTDSM implementation in Simulink.

Capacitance

The main capacitance is described by the capacitance block named C and it is modelled in a way that is similar to the basic model. This means that it uses a lookup table that describes the capacitance dependency with voltage is used. The values in the lookup table are taken from measurements done at 23 °C on the various supercapacitors, which means that each supercapacitor uses a separate lookup table. An addition in this model is the correction factor for the temperature variations. The correction factor is multiplied with the capacitance value from the lookup table in order to adjust the capacitance curve to different temperatures. The correction factor is calculated inside a MATLAB embedded function where the series resistance is also calculated. The correction factor is calculated by dividing the capacitance value for the temperature at a certain time point, with the nominal capacitance value according to

$$corr = \frac{p1 \cdot Tc^3 + p2 \cdot Tc^2 + p3 \cdot Tc + p4}{2700} \tag{16}$$

where the coefficients used to define the curve, which describes the capacitance variation with temperature for ESSP08, are gathered in Table 10.

Table 10: Coefficients in expression for capacitance calculation.

Coefficients	p1	p2	p3	p4
Value	0.002161	-0.06140	-0.18243	2701.990

This has the effect that the capacitance curve is shifted up or down depending on the temperature calculated in the model. The curve which relates the capacitance value for a supercapacitor to the temperature is created using the measurement described in Section 4.6.2. From this measurement a polynomial expression is created. In order to find an appropriate expression, the model error is calculated and the expression which results in an acceptable error value is chosen for further use. The calculation of the model error is described in more detail in Section 6.4. The curve relating the capacitance to temperature changes for different supercapacitors, even though the main curve shape remains similar.

Series Resistance

The R_s value also changes with temperature which can be seen in Section 4.6.1 about the measurements. To model the behaviour an exponential function is adapted to the measured values. This function is implemented in the same embedded function as the capacitance temperature dependency. The form of the expression is

$$R_s = Ae^{-\lambda T} \quad (17)$$

where A is the initial value, T stands for temperature and λ is the growth constant. By altering the values on A and λ , the behaviour for the various supercapacitors can be modelled. The series resistance is the component that generates most of the losses in the capacitor during charges and discharges. This means that it generates the heat that causes the temperature to rise in the capacitor during the cycles. The parameter values used in the DTDSM adapted to ESSP08 in the calculation of R_s are $A = 0.0013$ and $\lambda = 0.0219$.

Dynamics

The two branches connected in parallel with the main capacitance, model the slow dynamics of the supercapacitor. The names of the components in these parts are R_2 , C_2 , R_3 and C_3 . These dynamics of the supercapacitor can be seen after charging and discharging. The first voltage drop which is almost an ideal step is caused by the series resistance. After this voltage drop, the voltage of the supercapacitor appears to have an exponential behaviour. Figure 55 shows this dynamic behaviour after a charge of a supercapacitor. In that figure the three points used to determine the exponential behaviour are pointed out using the names p1, p2 and p3. In the actual supercapacitor this dynamic comes from the uneven charge distribution in the carbon layer [3]. To include more effects of dynamics in the model, several branches can be added in parallel. According to tests with such models, more than two branches seem to result in a relatively small increase of the accuracy. Therefore they are not used when

developing the advanced model further.

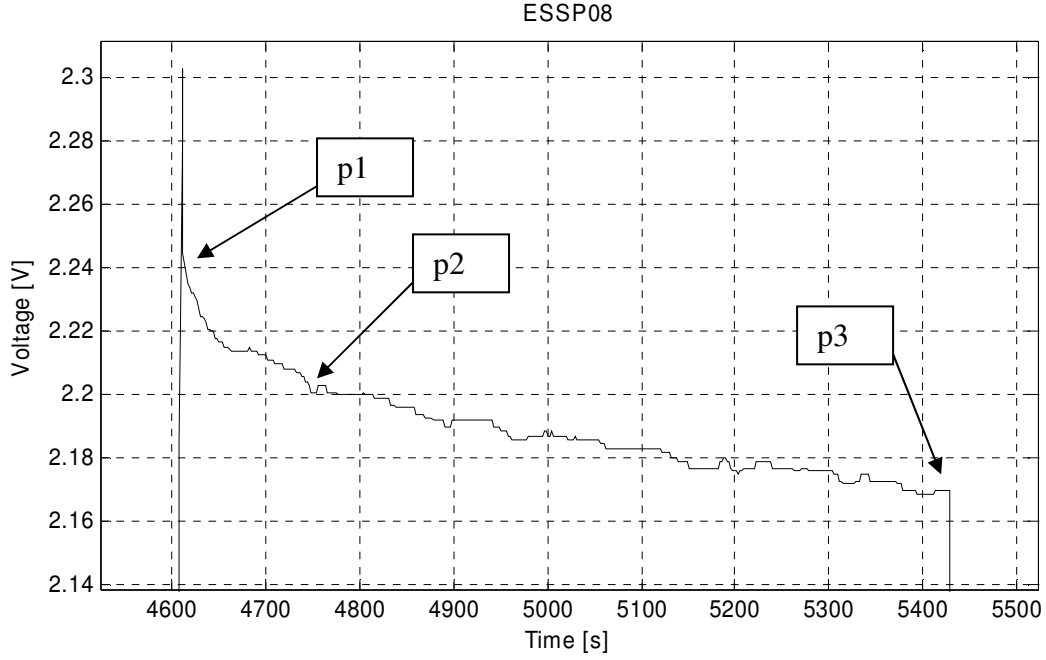


Figure 55: Exponential behaviour of voltage drop.

To calculate the value of the dynamic components, an assumption is made. This assumption is that no energy disappears after charging the capacitor. This means that the amount of energy stored in the supercapacitor is the same during this period. When the charging ends the voltage starts to decrease. Then the capacitance must increase if the energy level should remain constant according to

$$E = \frac{1}{2}CV^2 \Rightarrow \frac{1}{2}C_1V_1^2 = \frac{1}{2}C'V_2^2 \Rightarrow \{C' = C_1 + C_2\} \Rightarrow C_2 = C_1\left(\frac{V_1^2}{V_2^2} - 1\right). \quad (18)$$

This increase in capacitance can occur because of redistribution of charges. The redistribution can be modelled using a parallel RC branch. The capacitance is calculated using (18) where C is the rated capacitance and C' consists of the sum of capacitance from C and C_2 . The values are taken from measurements after charging the supercapacitor which can be seen in Figure 55. The resistance in the branch is calculated by looking at the time constant of the voltage level variation. The time constant is then used to calculate the R_2 value using

$$R_p = \frac{\tau}{C_p}. \quad (19)$$

The values on R_2 and C_2 are then implemented in the model providing acceptable results. These values are used as starting point in an attempt to find optimum values. The final values are found by studying the error value which will be described in (22). The value on the main capacitance must be altered when the additional component C_p is added to the model. This

must be done in order to keep the total model capacitance at the correct level.

Thermal Modelling

When the thermal properties of the capacitor are modelled some assumptions are made to simplify the model. The heat resistance of the material in the capacitor is neglected. This assumption leads to that when the heat is created by losses in the series resistance the whole capacitor receives the same temperature. It is also assumed that the ambient temperature is not affected by the heat that dissipates from the capacitor. From the measurement data gathered during the test for component values, the temperature of the capacitor and the ambient temperature can be found.

During the charge pulses with low currents the temperature effects are small but with higher currents the thermal dynamics can be seen more clearly in the temperature measurements. The temperature is measured at the connecting electrodes at the supercapacitor. This connection is chosen in order to get the best thermal contact with the supercapacitor. However it can lead to some part of the measured temperature appearing because of contact resistance at the supercapacitor. These effects are neglected when the model is created. When the measured data is studied the first approach is to create a linear model using

$$c \frac{\partial T}{\partial t} = P - k(T - T_a). \quad (20)$$

The result is a model that follows the temperature curve periodically but misses the temperature peaks or a model that follows the peak temperature but has a too fast temperature decrease, especially during small temperature differences between the capacitor and its surroundings. By observing measurement data, it is possible to conclude that the temperature variations include two time constants. To model this behaviour a second order model is created by using two first order models.

The model is described by the state space model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{k_1}{c_1} & 0 \\ 0 & -\frac{k_2}{c_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{c_1} & \frac{k_1}{c_1} \\ \frac{1}{c_2} & \frac{k_2}{c_2} \end{bmatrix} \begin{bmatrix} P \\ T_a \end{bmatrix} \quad (21)$$

$$T = [1 \quad 1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - T_a$$

where the constant c is the thermal capacitance for the material and the constant k represents the thermal resistance that describe how fast the heat dissipates from the capacitor. P is the effect generated in the resistance and T_a is the ambient temperature. The two states in the model are x_1 and x_2 . The result is promising and after some parameter tuning of the model, it describes the temperature dynamics satisfactorily. To be able to describe the temperature

dependency in the entire available temperature range, the model is extended. The extension consists of a nonlinearity which is caused by adding temperature dependency to the constants in the matrix. The new constants are introduced in the Simulink model as lookup tables for the different constants. The basic principle is still the same as for the first equation. The derivative of the temperature depends on the generated heat subtracted with the dissipating heat. The generated heat comes from the losses in the series resistance.

5.4 Program Tests using DTDSM

The new model is implemented in some of the different program alternatives to see how well it works and if there are any difficulties in its implementation. Two programs are omitted from this evaluation round because they are not as interesting as the other alternatives. The omitted programs are PSCAD and OrCAD Capture. The main cause of their exclusion is that they are too specialized at creating electrical models. This limitation is present in SimPowerSystems and PLECS as well but they have the advantage of being closely related to MATLAB and Simulink.

5.4.1 SimPowerSystems

The new model is also implemented in SimPowerSystems. To be able to introduce the new circuit some modification is done from the earlier model. The nonlinear capacitance model is expanded with an embedded function that calculates the thermal correction factor for the capacitance curve. Other than that the modelling of the nonlinearity of the capacitance uses the same structure as for the simpler model that is described in Section 5.1. The entire model can be seen in Figure 56.

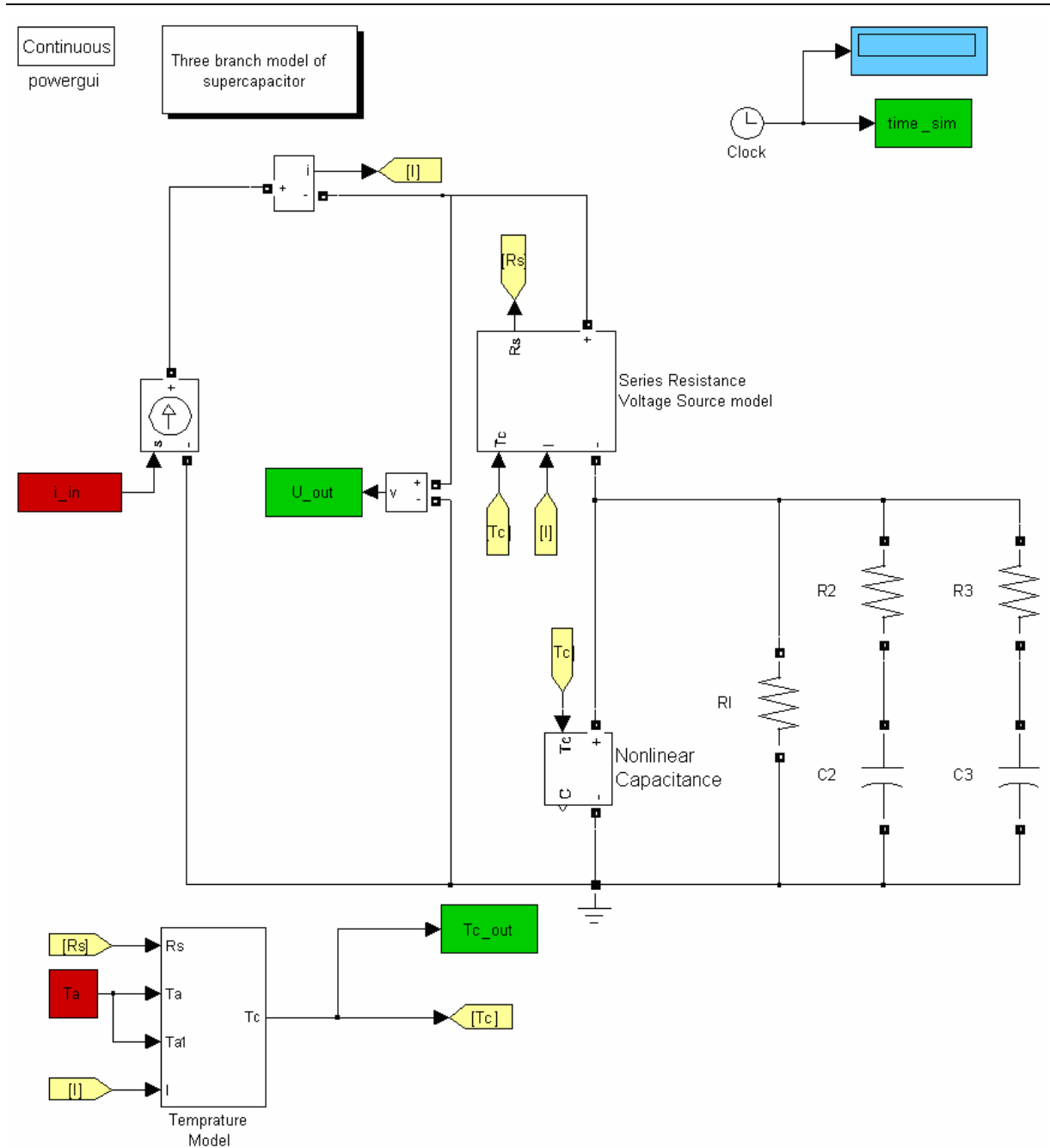


Figure 56: The complete SimPowerSystems model of the DTDSM.

The series resistance is also nonlinear in this model and the value changes with the temperature. To model this relation between temperature and resistance an embedded function is implemented in the series resistance block, where the expression is the same that is used in the Simulink model described in Section 5.3.2. The nonlinear resistance can be modelled either with a controlled voltage or a controlled current source, without any problems with its initial condition. This is because the resistance has an algebraic relation between current and voltage that does not include any dynamics. In this model the controlled voltage source is chosen because of the problems that occur when two current sources are connected in series.

The linear branches are implemented as R2, R3, C2 and C3 in parallel with the main

capacitance. The thermal model is also the same as in the Simulink model.

5.4.2 Saber

The DTDSM is implemented in Saber using an approach similar to the SimPowerSystems implementation. In Saber the nonlinear resistance and capacitance components in the circuit use symbols that are designed using the simple drawing tools in the program. The Temperature calculations are also represented by such a symbol. Hiding behind these symbols, are the actual calculations being conducted to model the nonlinear behaviour. Some thing that is special for the implementation in Saber is that a lookup table is used to produce the input signal. Another thing is the conversion from signal value to electrical value, such as voltage or current, which is done using interface blocks instead of controlled sources and measurement blocks. The main part of the Saber model is shown in Figure 57.

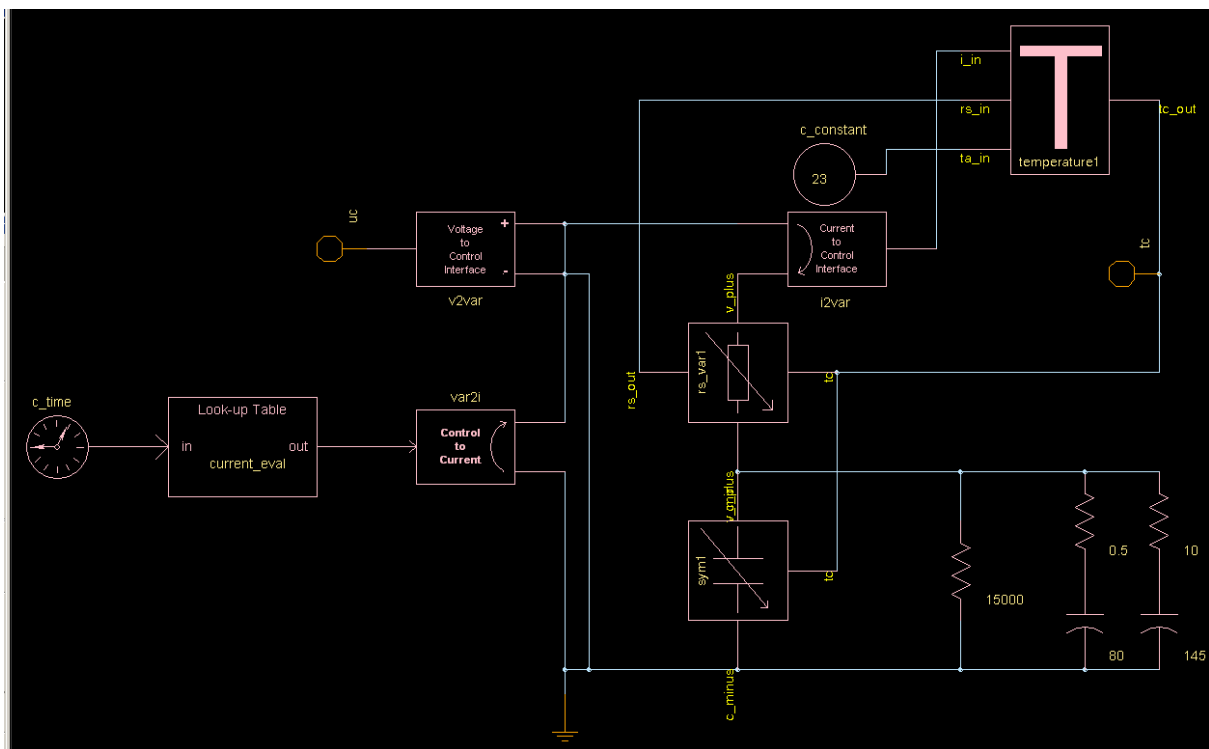


Figure 57: DTDSM implemented in Saber.

5.4.3 PLECS

Figure 58 presents the PLECS implementation of DTDSM which has an appearance that is similar to the implementation in Simulink. However the main difference is that the circuit components are all gathered in a separate sub block. Instead of modelling the circuit components using mathematical expressions, actual circuit components are used as can be seen in Figure 59. The nonlinear components in the circuit are modelled using controlled voltage sources. All calculations needed to control these components are done outside the PLECS circuit model in the Simulink environment. An advantage of the separation of electrical components and mathematical calculations can be that the overview of the model function is improved. However since it is possible to accomplish this separation in other programs as well it can not be seen as an important advantage. Being forced to separate the

electrical part of the model can actually be a disadvantage, since it limits the ability to design specialized electrical components which can be used in circuits as regular components. In PLECS the calculations required for such special components must be conducted separately.

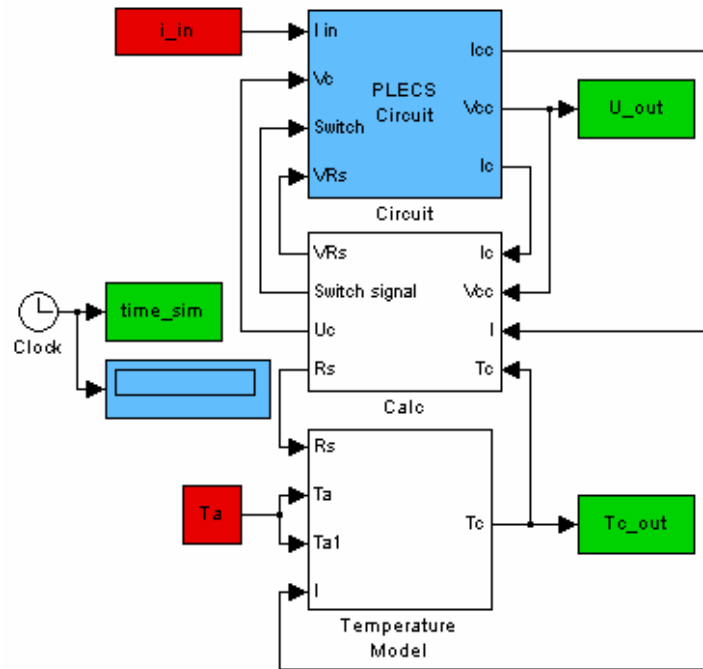


Figure 58: Simulink view of the DTDSM implementation using PLECS.

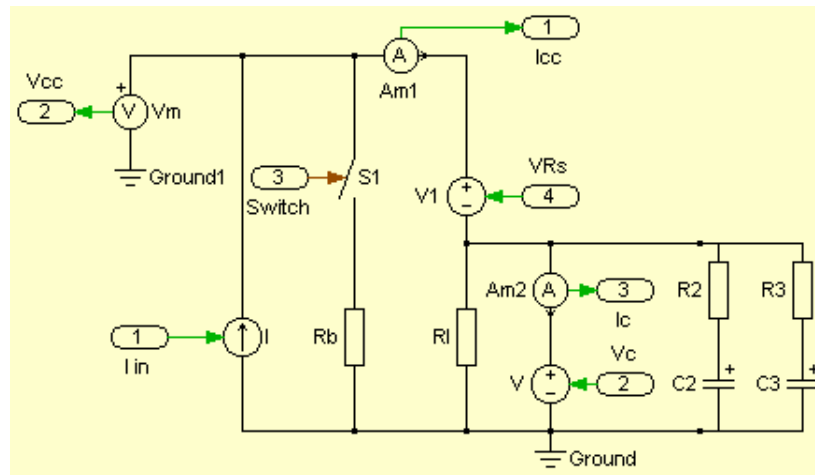


Figure 59: Electric circuit part of the PLECS model.

5.4.4 Dymola

The DTDSM implementation in Dymola is highly similar to that in SimPowerSystems. However a few differences can be noted when studying the model, which is displayed in Figure 60. In SimPowerSystems labels are used to separate the temperature calculation from the main electrical circuit. No such capability is found in Dymola during component searches. In some cases this separation can prove advantageous, while it might introduce complications in other situations. In the Dymola model custom icons are used to represent the nonlinear

components and the temperature calculation block. The icons are easily created using a drawing tool. Similar functionality is available when creating components in Saber, Simulink and SimPowerSystems.

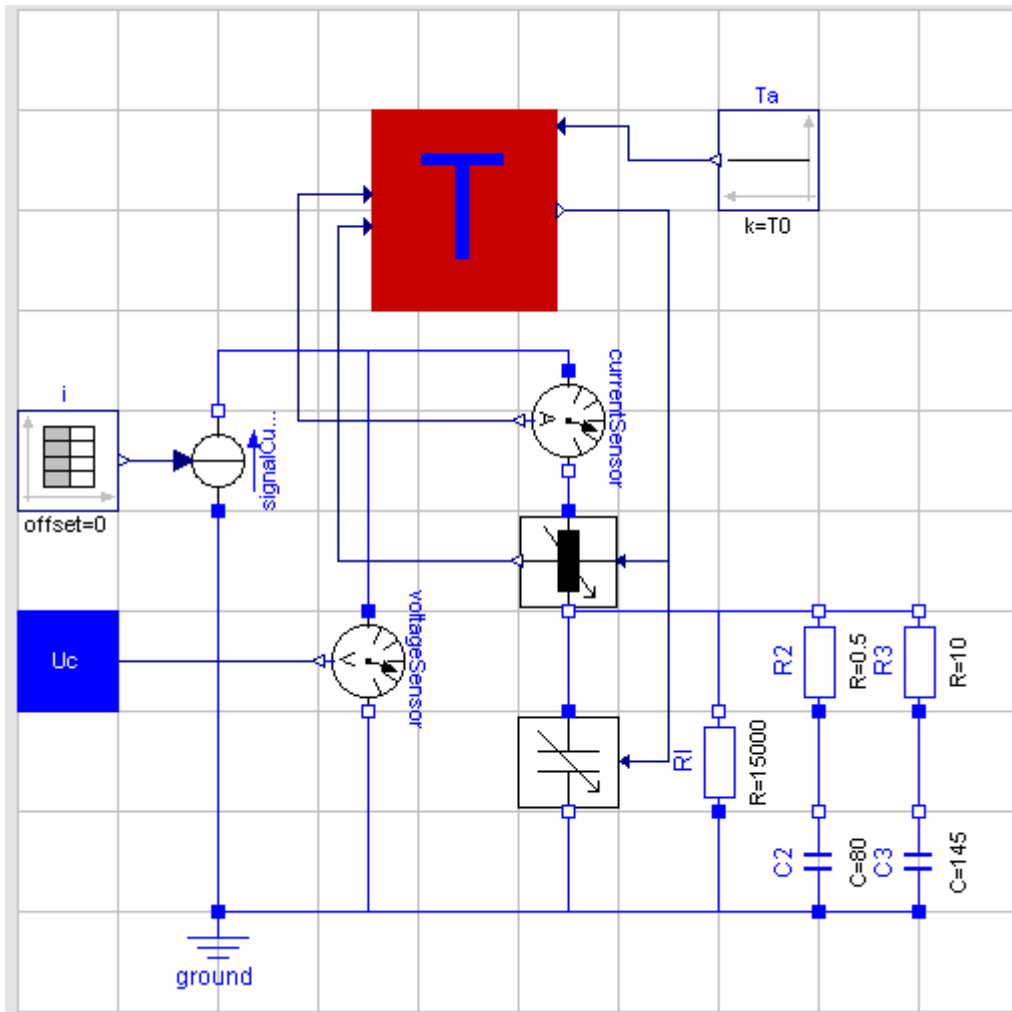


Figure 60: DTDSM implemented in Dymola.

6 Simulation Results

This section presents the results of the simulations in the different programs. Only the most interesting results are presented since the total amount of results is too large. Using the presented results, observations are made and results are explained.

6.1 Basic Model

To see how model improvements affect the results, the final basic model is compared with the first simple model. The first simple model consists of a series resistance connected in series with a parallel connected leakage resistance and capacitance and it is shown in Figure 61.

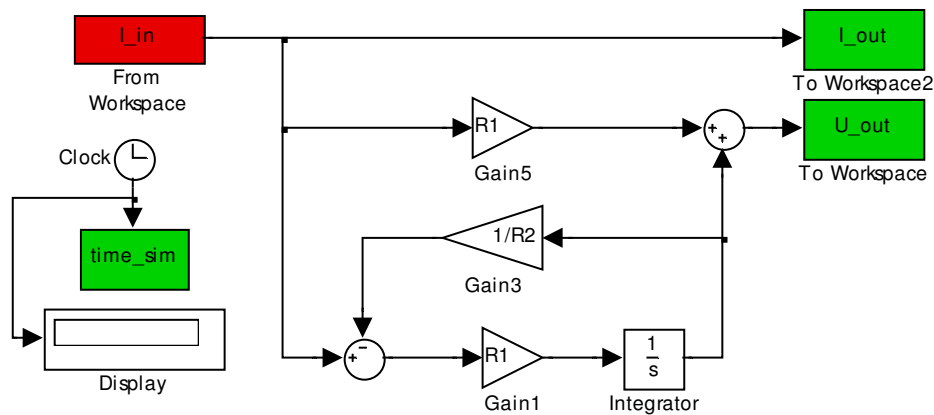


Figure 61: Simple model of supercapacitor for comparison with the final basic model shown in Figure 30.

When using the model in Figure 61, the resistance R1 had to be increased to 24 m Ω in order to provide better similarity with measurements. With the altered R1 value, the simulation resulted in the voltage curve shown in Figure 62. This model gives the main shape of the real behaviour of the component but it is clearly not good enough, although it does show the basic characteristics of the supercapacitor. Since the model does not include the voltage balance resistance, which is connected in parallel with the circuit, the voltage gets to high. So the voltage increases during the cycle compared to the measurements.

Simulation Results

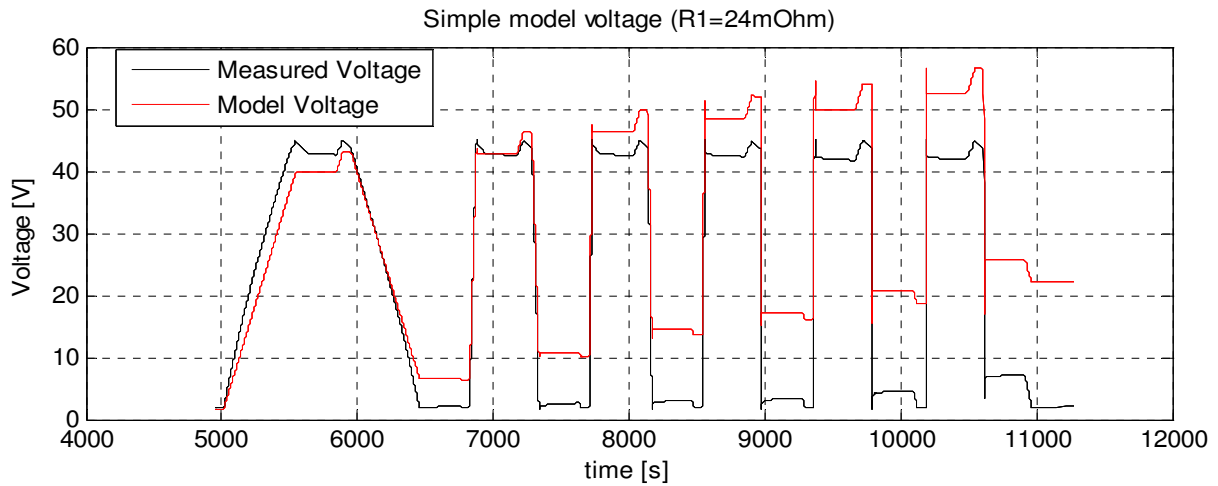


Figure 62: Resulting voltage curve from simple model.

The first model version which includes the effect of nonlinear capacitance which can be seen in Figure 30 contains some improvements over the simple model. The resulting voltage is shown in Figure 63.

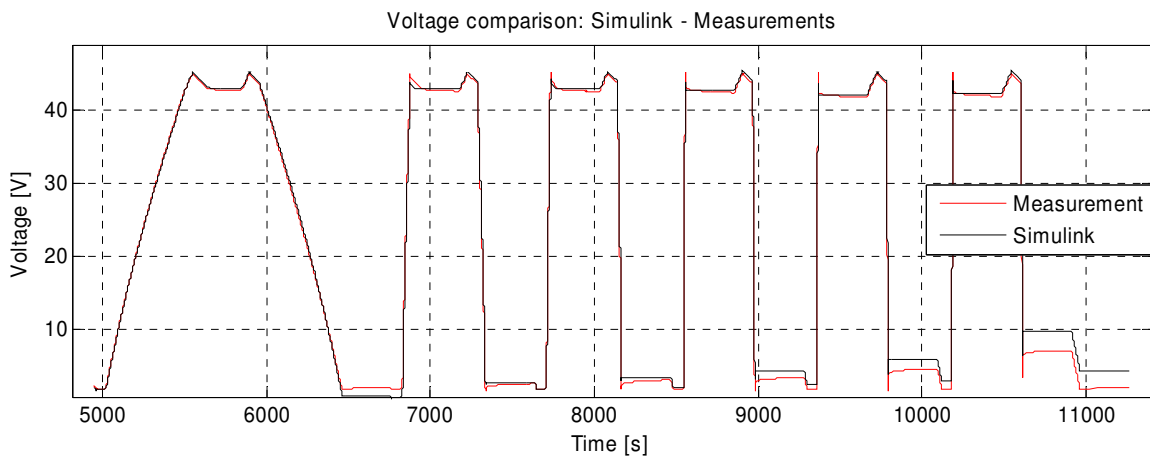


Figure 63: Voltage curve using first model version including nonlinear capacitance.

Here the effect of the over voltage balance resistance is clearly visible. The resistance is connected when the voltage reaches a voltage level that is over 42.8V. A result of this is that the model voltage does not overshoot the measured voltage. The effect of the nonlinear capacitance is most visible at the first rise of the voltage. There the model voltage variation clearly is nonlinear, as opposed to the simple model where it is linear. The effects of the dynamic behaviour during the times of zero charging current should be represented by R_p and C_p . Unfortunately the dynamics are not represented well by these two components as can be seen most clearly after the last discharge when the voltage is at its low level. There the measurement curve shows a small increase in voltage level. The model voltage does not display this dynamic behaviour; instead the voltage level is constant.

Several component values are tested for R_p and C_p to try to increase their effect on the voltage level. When reasonable component values are used the effect is not altered much. However some tests are done using very large values that show some effect but these are

discarded because they are believed to affect the charge level in a negative way. The deviation between the voltage level of the model and the measurement is larger at the end of the test cycle than it is at the beginning of the test.

This increased deviation can be caused by an offset in the current supplied by the measurement equipment. This offset causes a deviation because it creates a difference between the input energy in the model and the actual input energy supplied during measurements.

6.1.1 Program Alternatives

The results of the various program alternatives are highly similar to each other. However it is still possible to do some general observations regarding the measurement results. The current source model displays an error in the initial value. This is because it is not possible to set an initial value in this model, which means that it starts from zero. The effect of the nonlinear capacitance can be seen if the first charge is studied, where the linear model does not follow the measurement curve well. However the linear model appears to have the advantage of reaching an end value that is closer to the measurement. In Saber it is possible to use an alternative component to model the nonlinear capacitance. In that model a specific nonlinear capacitance component that used a polynomial to determine its capacitance is used. The results from all the programs are similar to the results from Simulink and they are therefore not presented here.

6.2 Two Branches Model.

The results from simulations with the Two Branches model for the supercapacitor ESSP08 are presented in Figure 64, which shows the model ability to follow the measurement cycle which is used to identify the parameter values. Figure 65 shows the model response using the evaluation CBR cycle. The cycle is a simulated HEV battery load cycle and it is described further in Section 6.3.

Simulation Results

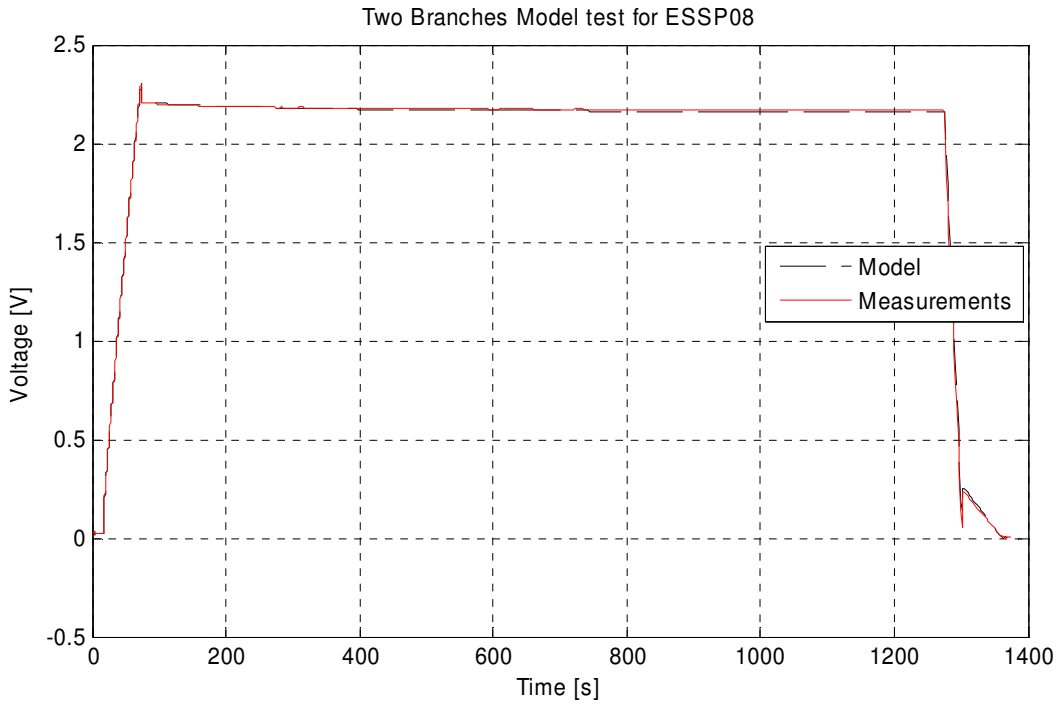


Figure 64: The Two Branches model simulated using a test cycle.

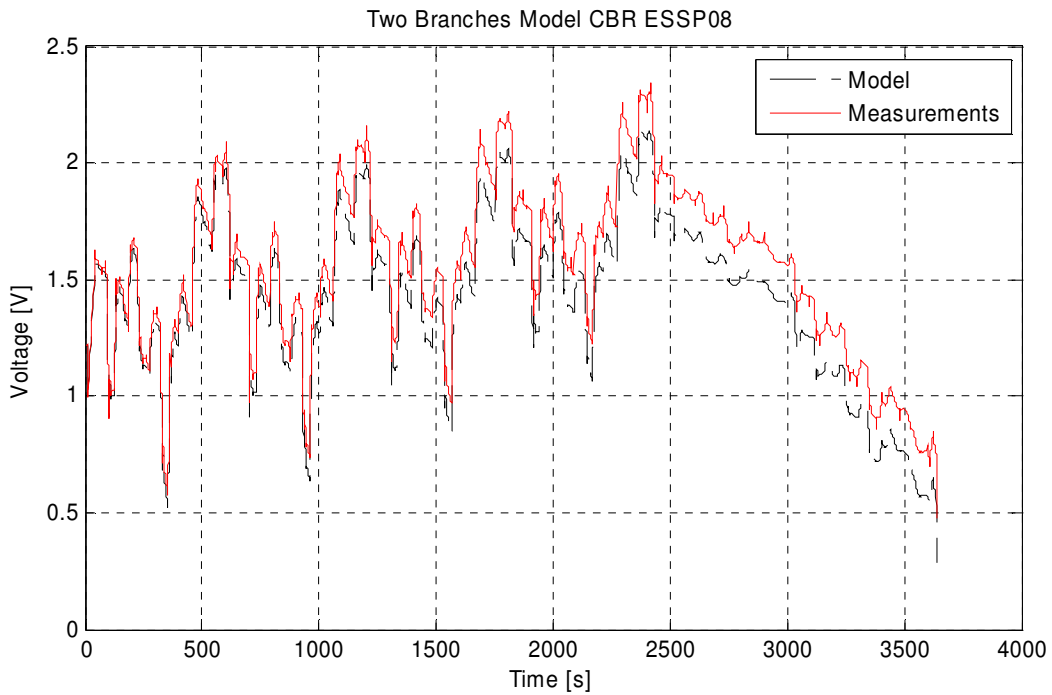


Figure 65: Simulation results for Two Branches model using CBR cycle.

The model seems to describe the main dynamic behaviour rather well when the CBR cycle is simulated. However, if the model error is analysed during the simulation time it can be observed that the deviation increases with time. This can be seen in Figure 66. The increasing model error seems to reduce when the supercapacitor reaches the rated voltage. This increased model deviation over time can be caused by a current offset in the measurement equipment.

However the other results from simulation with DTDSM model do not display the same trend. Most of these errors are assumed to be caused by errors in the created model. Probably some of the parameter values are not optimal. The model appears to be too simple to be able to model the behaviour of the supercapacitor adequately. However even though it results in some errors there are still applications where this model can be useful, mostly because of its simplicity.

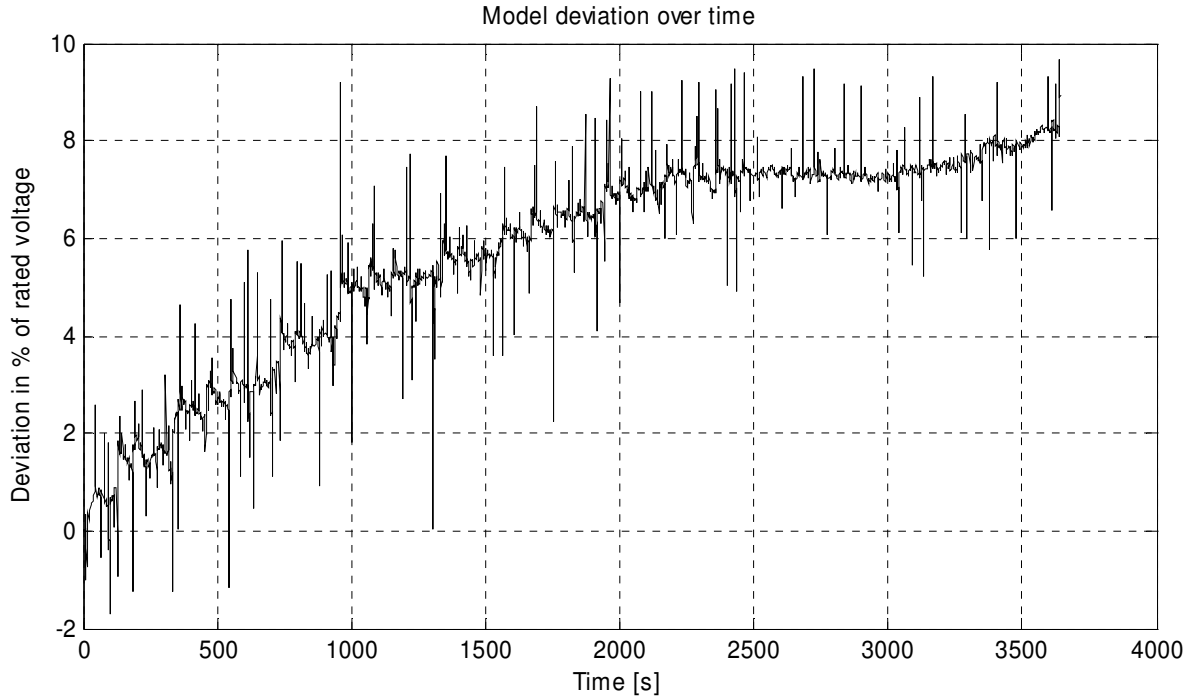


Figure 66: The Two Branches model deviation from measurements when the CBR cycle is simulated.

6.3 DTDSM

The results of simulations with the new model in Simulink are plotted together with the measurements in Figure 67 which shows the capacitor voltage and temperature at 23 °C. Figure 68 shows the capacitor at -30 °C. The model seems to follow the actual behaviour of the supercapacitor. The change in temperature results in a variation in the series resistance of the supercapacitor and the model is able to follow these changes accurately. The temperature model also seems to model the rise in temperature well.

Simulation Results

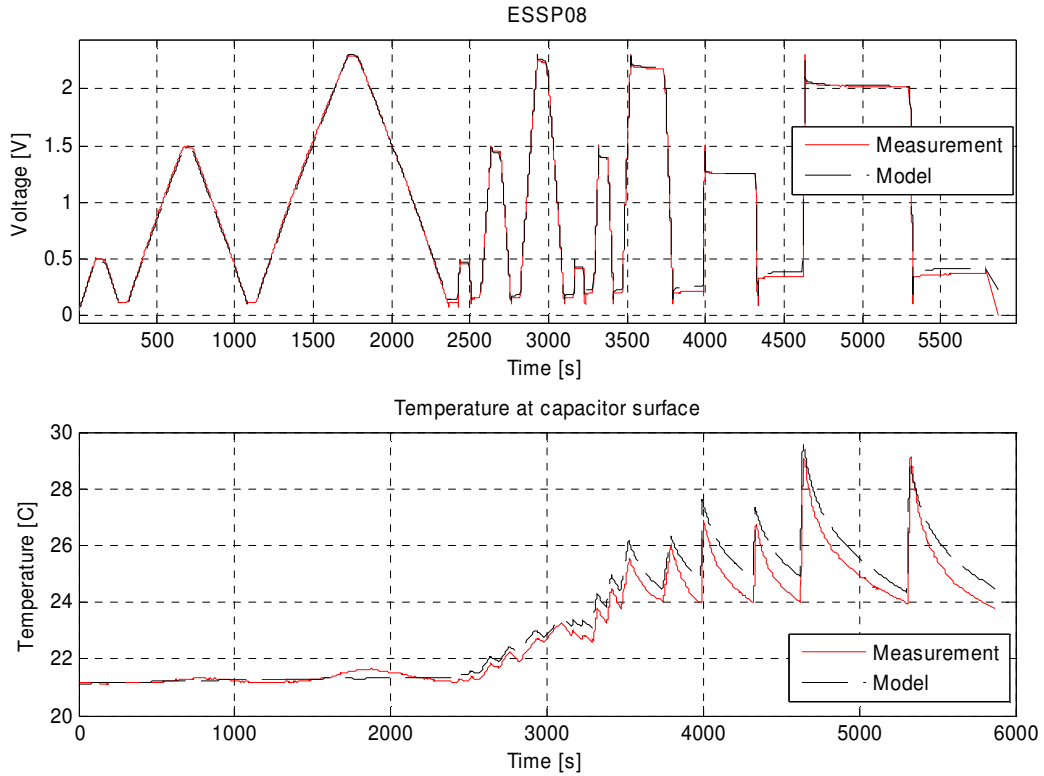


Figure 67: Results with ESSP08 at 23°C, Varying voltage test.

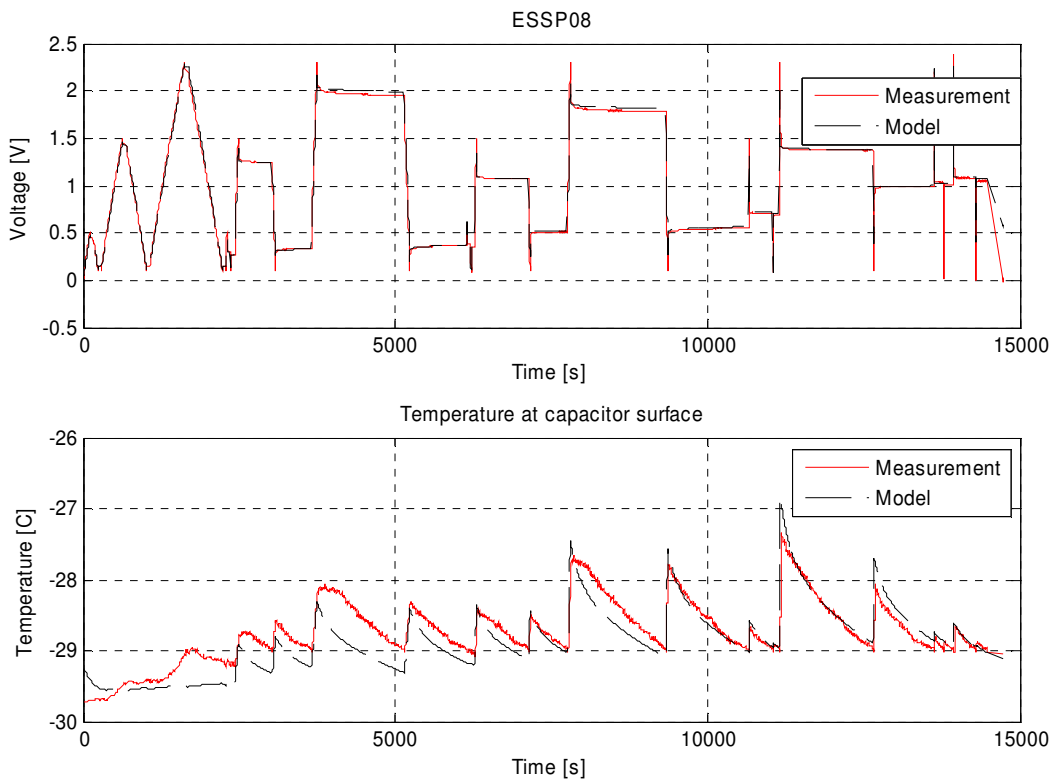


Figure 68: Results with ESSP08 at -30°C, Varying voltage test.

For evaluation of the model behaviour a different test cycle is used. The new test cycle is

Simulation Results

based on measurements from a test drive with a city bus in Gothenburg. The cycle contains different events like acceleration, braking and gear shifts, it also contains different topologies such as ascents and descents. The test cycle is used on ESSP08 and the reason for choosing this supercapacitor is that it has the largest variations in model parameters during different temperatures, especially for the series resistance. The results from this test can be seen in Figure 69.

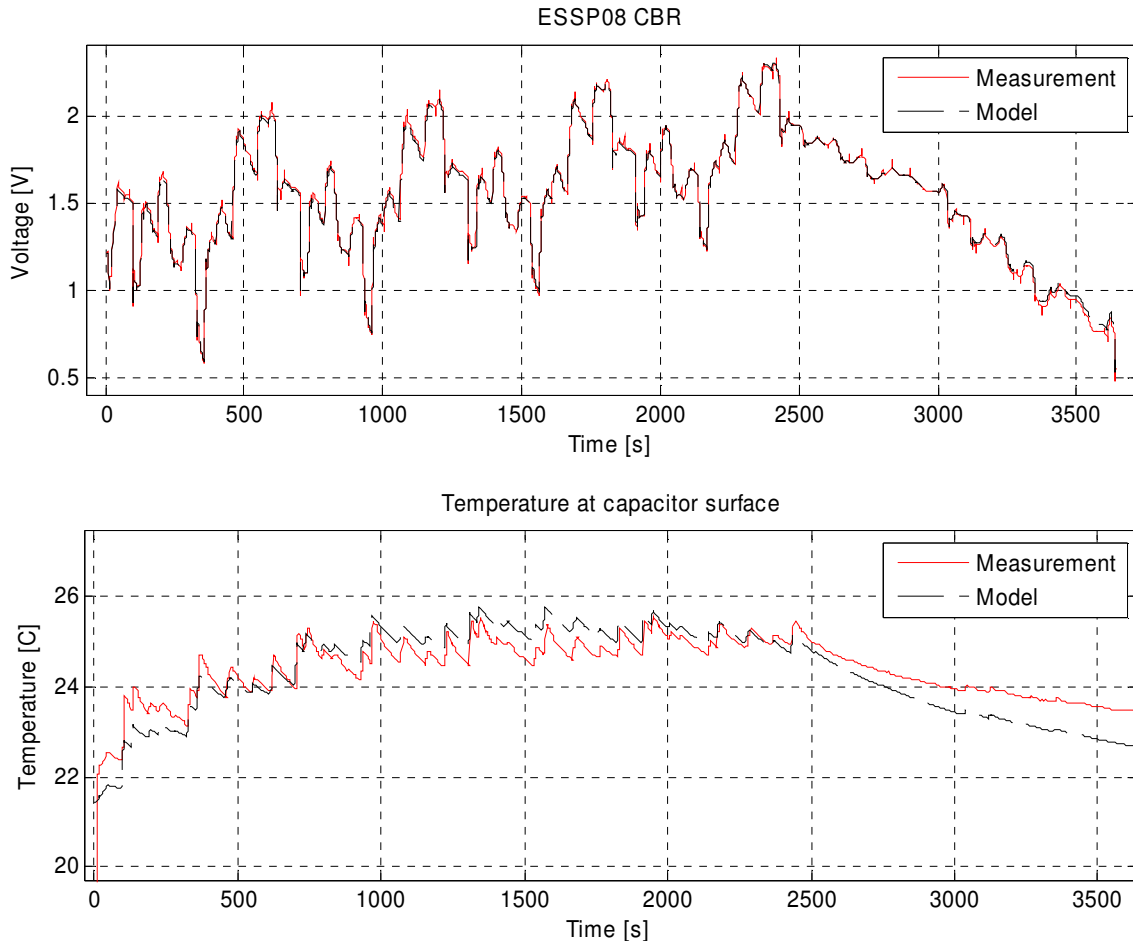


Figure 69: The results from simulation of CBR cycle.

In Figure 70 the results from the different programs are shown. The results are similar in all programs and only small variations between the programs can be observed. There is however a small offset between the measurements and results for Dymola.

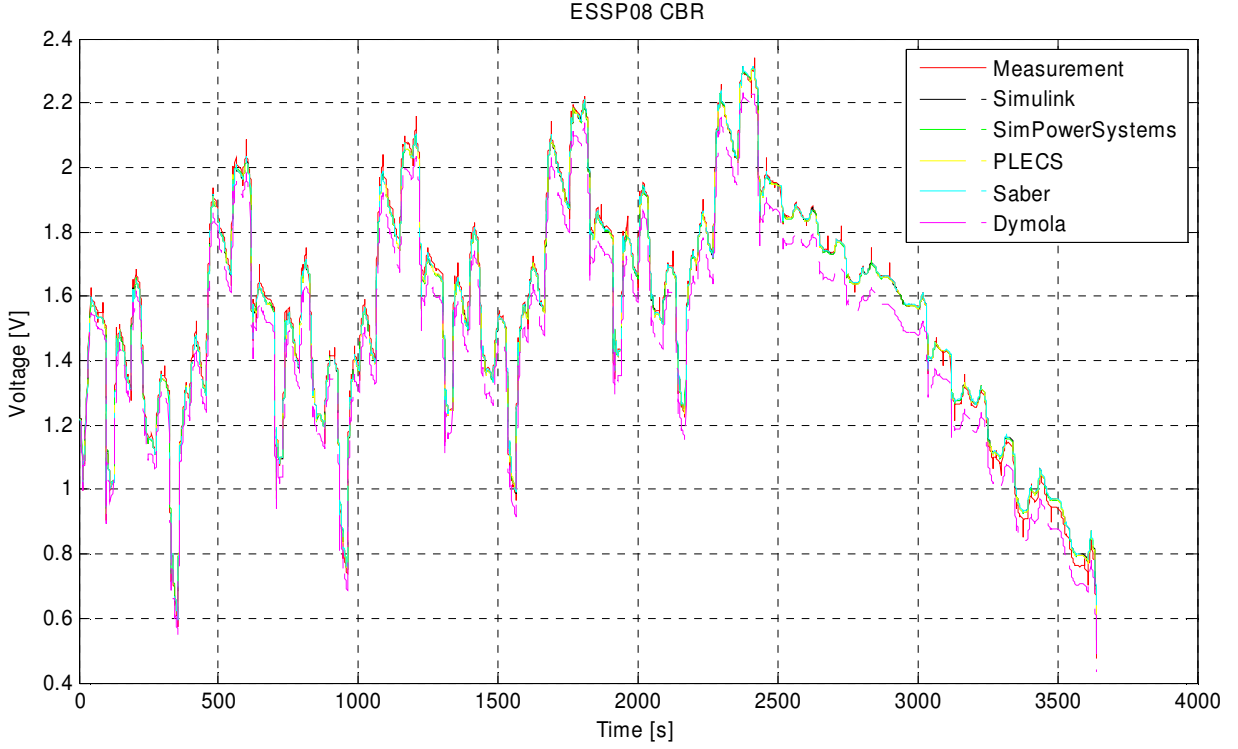


Figure 70: Model results in program alternatives compared to measurement results.

6.4 Estimation of Model Errors

To form a mathematical expression for the deviance between the models and the measurements a penalty function is used. This function is called V_n and is defined as

$$V_n = \frac{1}{N} \sum_{t=1}^N \varepsilon^2(t, \theta) \quad (22)$$

where the function calculates the model error in each time point and the calculation of this error is represented by ε in

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t | \theta) \quad (23)$$

where $\varepsilon(t, \theta)$ is the error function, $y(t)$ is the measurement value and $\hat{y}(t | \theta)$ represents the model with the vector θ which is the vector that contains the model parameters [25]. The error is calculated for all values and normalized with the number of data points. The function V_n gives a measure on how well the model describes the real system given in measurement data. In the expression for the error value N is the number of data points.

6.4.1 Analysis of Errors for Basic Model

The best model is the model that gives the minimum value of the penalty function V_n . The

Simulation Results

results of calculating the error function for the models in the different programs can be seen in Table 11. All the tests are made with the model of the supercapacitor ESSP48, using the test cycle from the Screening Test.

A second measurement value which can be used to judge the accuracy of the simulation results for the various models is the Final Value Error. This value is found by taking the final value of a model simulation and subtracting the final value from the measurement data. This means that the model ability to reach the correct final value is analysed. The Final Value Errors are all recorded together with the V_n value in Table 11.

Table 11: Error function comparison between different models in the analysed programs.

Model	Value on V_n	Final Value Error [V]
Simulink NLC	0.8832	1.9617
SimPowerSystems LM	3.2501	-0.9654
SimPowerSystems NLC VS	0.8823	-2.1809
SimPowerSystems NLC CS	1.3469	-1.9836
OrCAD Capture LM	3.2662	-0.7985
OrCAD Capture NLC VS	0.9446	-1.8803
OrCAD Capture NLC CS	2.6791	-3.1681
PSCAD LM	4.6907	-0.9768
PSCAD NLC VS	0.8951	-2.3184
Saber LM	3.2290	-1.021
Saber NLC	0.7793	-1.684
Saber NLC VS	0.6231	-1.705
Saber NLC CS	1.1363	-0.693
PLECS LM	4.2585	-0.8074
PLECS NLC VS	0.9532	-1.9154
PLECS NLC CS	1.4328	-1.6958
Dymola LM	4.8300	-0.982
Dymola NLC VS	0.8757	-2.17
Dymola NLS CS	1.3353	-1.978

Since the model development is finalised using Simulink, the implementations in the other program alternatives are supposed to follow the behaviour of the Simulink model. In order to demonstrate their abilities to follow the Simulink model, error values are recorded for their deviation from the resulting Simulink curve. The presented results are found using the NLC VS models since they produce the best results. The values are all gathered in Table 12. According to these values, the SimPowerSystems implementation follows the Simulink model best.

Simulation Results

Table 12: Error values for the various program alternatives.

Model	Value on V_n	Final Value Error
SimPowerSystems	0.00003204	-0.00189
OrCAD Capture	0.011100	0.3191
PSCAD	0.826600	-0.1190
Saber	0.489200	0.4946
PLECS	0.228500	0.2839
Dymola	0.000968	0.0290

From the results of the calculation of V_n for the models, the best results are given by the NLC VS models. The presented results in this section come from tests using the Screening Test on ESSP48. Figure 71 shows a plot of model deviation over the simulation interval compared with the measured voltage. In this plot, it can be observed that a trend exists in the error behaviour. The error seems to have an increasing trend during the simulation interval. This error can be caused by an existing offset current in the measurement equipment. A negative error value corresponds to a model voltage which is higher than the actual voltage. Using the first and last value to calculate the offset current with (2), the result is an offset current of 5.83 mA for the linear model. The model error seems to increase during time, but during the period between charge and discharge the error decreases. This appears because the model does not describe the slow dynamics in the system in an accurate way. The offset current that causes the voltage level at the uncharged state to increase should also affect the top voltage after the charges, however it does not. The reason for this is that the charge balance resistor limits the voltage at the top level.

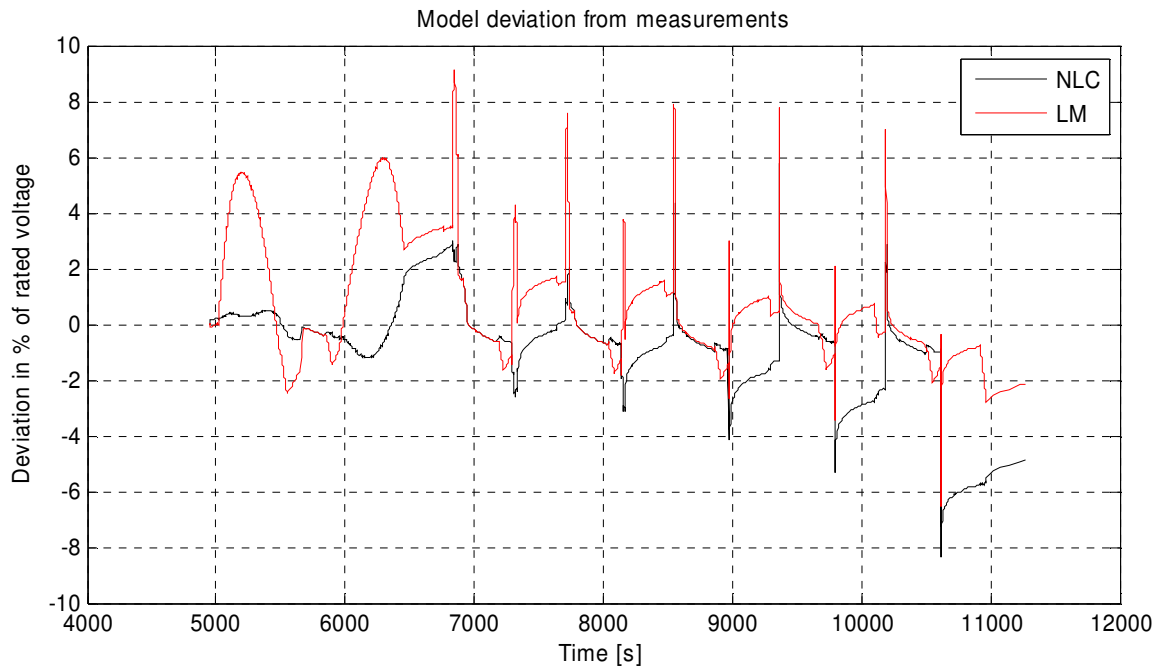


Figure 71: Model deviation from measurements for the basic Simulink model.

The difference between the linear and the nonlinear model is highlighted by Figure 71. This enables better comparison between the two model types. The linear model can not include the effects of a voltage dependent capacitance, the effect of this can most easily be observed during the first charge and discharge cycle. The result of the model deviations at this point is that the linear model has a significantly larger error than the nonlinear model which includes the mentioned effects.

Another observation is that the nonlinear model seems to have a faster increase of the deviation than the linear model. The nonlinearity in the model can cause an imbalance between the charge and discharge steps. This result is that the linear model comes closer to the final value than the nonlinear model. This means that the linear model is a good approximation for the supercapacitor. However whether this error is caused by the nonlinearity or the offset current, it is still small (The voltage error is about 5% of the rated value of the supercapacitor).

If the model deviation is plotted with the charging current as an x-axis, an increasing trend is observed. Both the nonlinear model and the linear model demonstrate similar results which can be seen in Figure 72. During discharges the model can not reach the measured voltage which also can be seen in Figure 63. During charges the model generates a lower voltage than the measurements. This is probably caused by an incorrect series resistance value. The largest error occurs when the largest currents are used. This is because the large currents produce a voltage drop over the series resistance, which affects the voltage over the capacitance in the model. This leads to an incorrect capacitance value in the lookup table.

Simulation Results

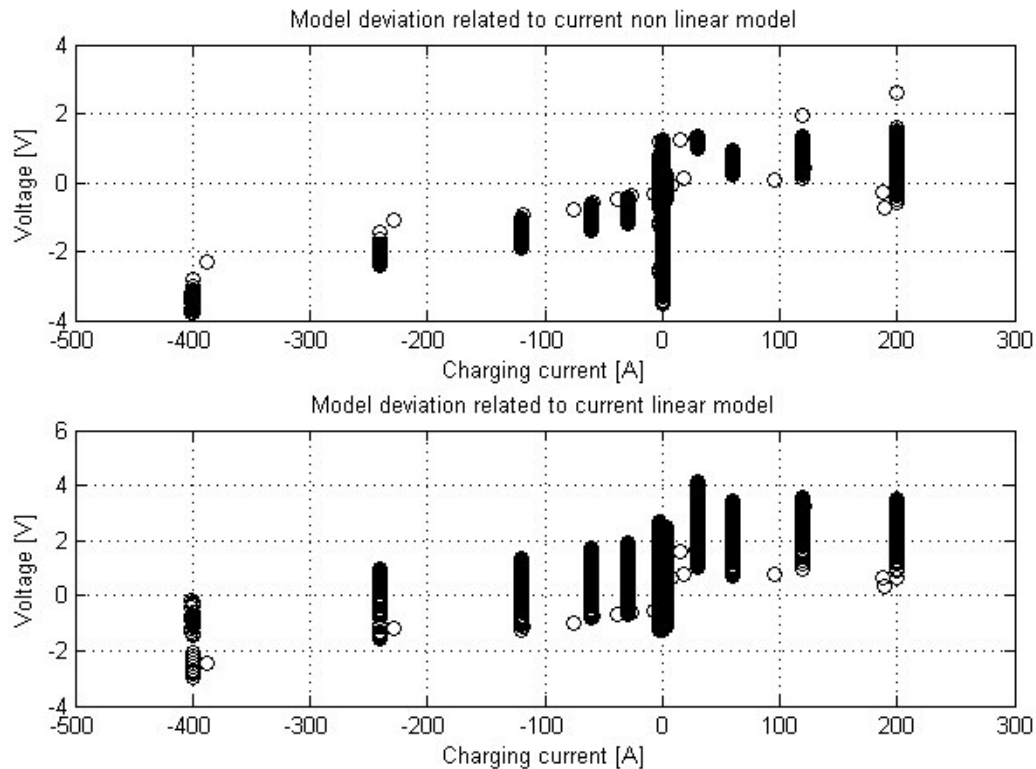


Figure 72: Model deviation from measurement related to charging current.

If the deviation from Simulink results is studied for the different simulation programs, the first observation is that most programs have only small deviations which can be seen in Figure 73. PSCAD, PLECS and Saber have the most significant deviations. The PSCAD model has a large error during charges and discharges. The error also increases for larger currents. The PLECS model deviation is largest in the beginning and decreases over time. The Saber model produces an error after about half the measurement cycle which results in the largest deviation from the Simulink model behaviour. Possible reasons for these errors are incorrect model settings, solver settings or differences in model implementation.

Simulation Results

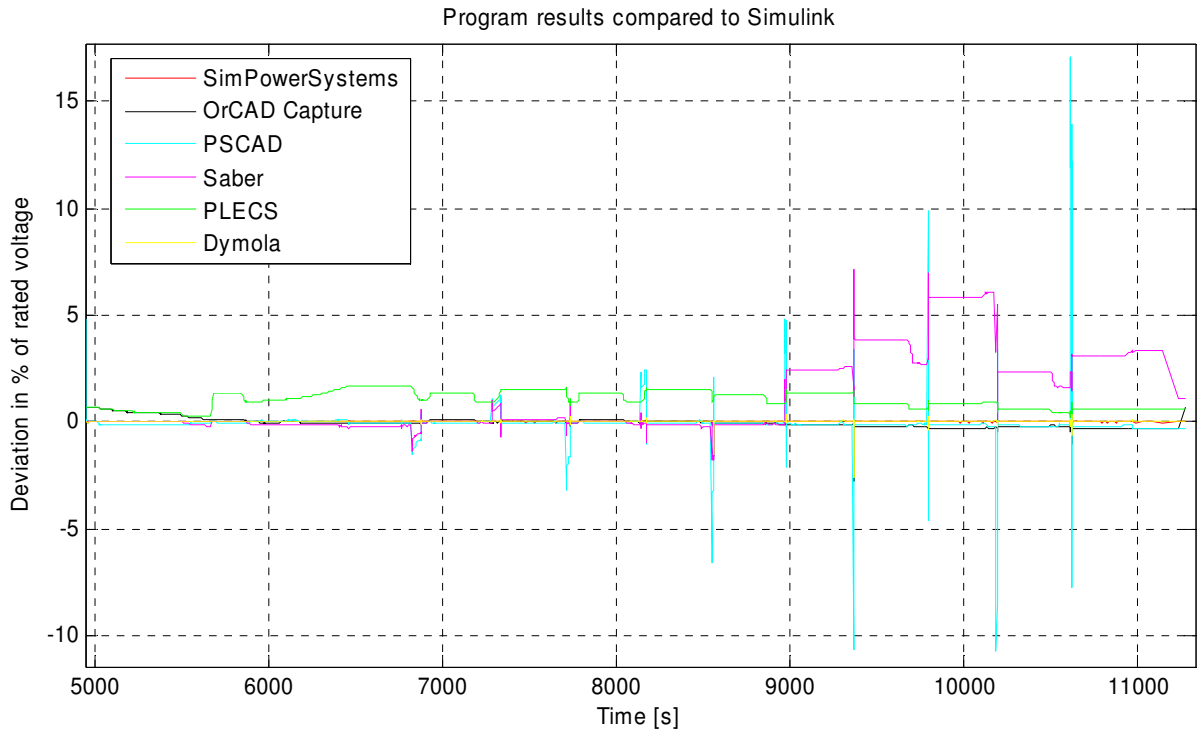


Figure 73: Model deviation from Simulink result for different programs.

6.4.2 Analysis of Errors for DTDSM

The model deviation from the measurement is calculated over time and the result is shown in Figure 74. The model error seems to be relatively stable during the simulation and the magnitude of the error is also low, about 2.1 % of the rated voltage.

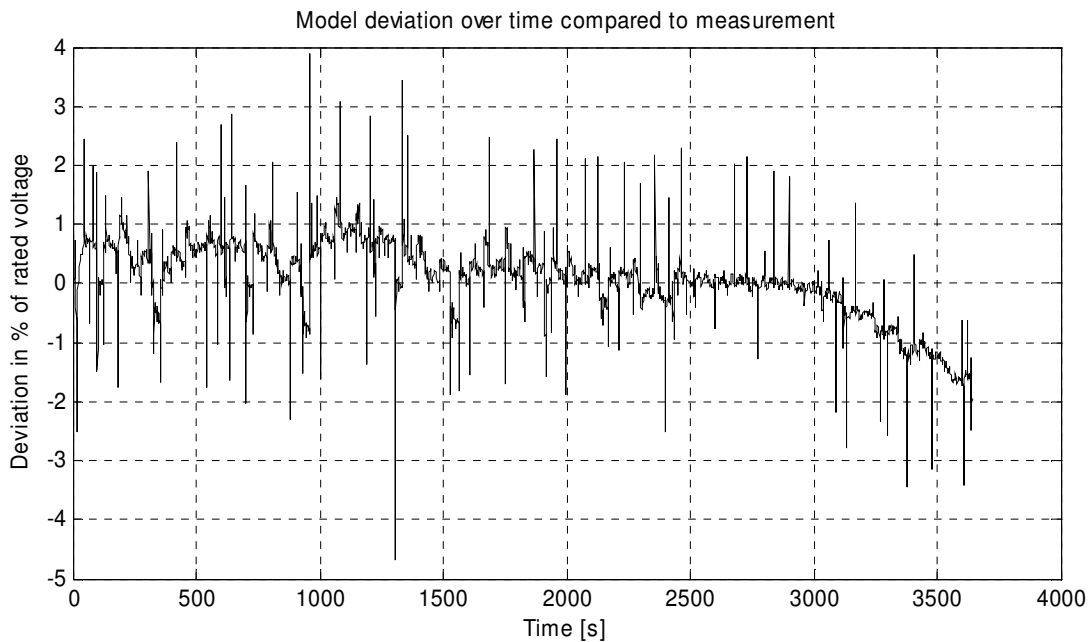


Figure 74: Model deviation over time for the DTDSM.

Simulation Results

One event that the model can not describe is the short pike voltages that occur during the simulation, some of these are shown in Figure 75. If the current is studied together with voltage, the current peaks that should correspond to the voltage pikes do not exist. This conflicting fact leads to the conclusion that a measurement error exists. The error can have many sources, the voltage or current measurements are two possible sources. It is also possible that the connection between the measurement equipment and the supercapacitor includes some dynamics or that the equipment alone causes the error. In this case it is assumed that the voltage measurement is the cause to the error and that these pikes do not exist. In a real application one cause of the fast pikes can be a fast change in load and one example could be gear shifts. Since these fast transient pikes have a low energy level and the current that causes them is small, the assumption that they do not exist does not introduce any larger problems in the model accuracy.

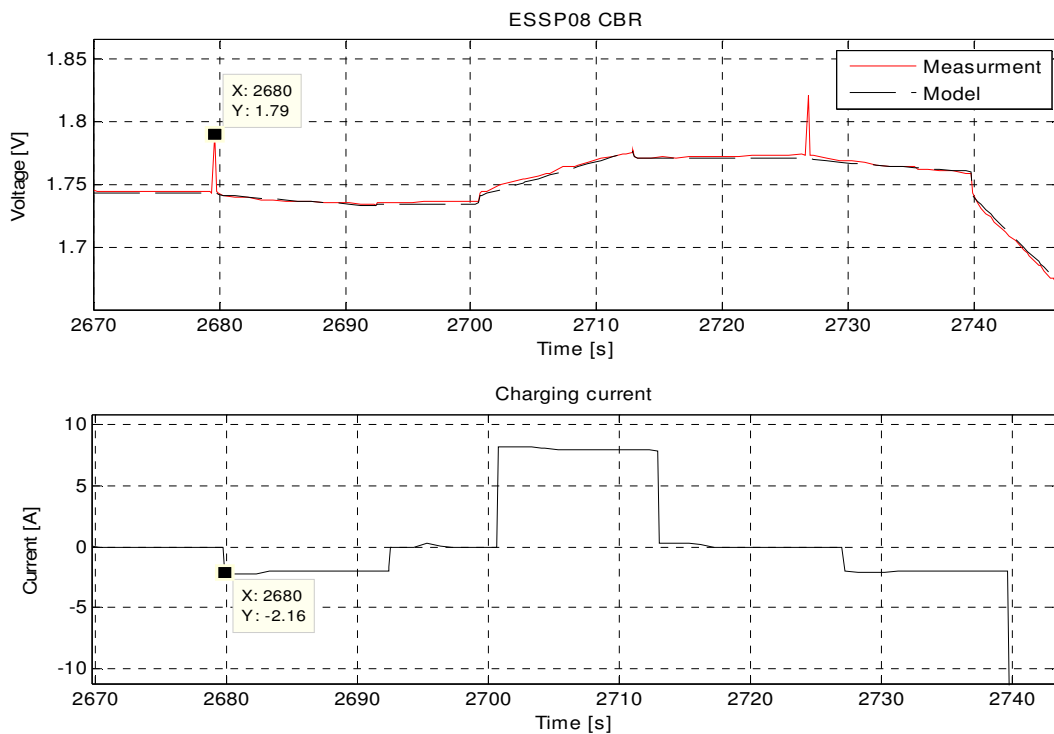


Figure 75: The figure shows that there does not exist a corresponding current to the voltage pikes.

The DTDSM seems to follow the measurements rather well but to check the model functionality the energy in the system is studied. If the energy in the capacitor is calculated by integrating the product of current and voltage, the energy that is used during charges is larger than during discharges. This statement is correct if the voltage level is set to be the same before and after the cycle. The reason for the extra energy requirement is that the energy disappears in the series resistance. This calculation is done for ESSP08 using the test cycle called varying voltage test. The results confirmed the assumptions and the resulting sum of positive and negative energy is larger than zero. By calculating the energy that disappears in the series resistance and subtracting it from the energy already calculated, the results should be zero since the capacitor can not create energy. This energy is calculated by integrating the developed power in the series resistance. When this calculation is done for ESSP08 model

Simulation Results

there is still a positive amount of energy remaining. From this energy the corresponding voltage is calculated by using

$$E = \frac{1}{2} CV^2 \quad (24)$$

where the rated capacitance value is used.

The result is that the voltage level increases 0.98 V by the end of the simulation. This is a significantly higher voltage level than the resulting voltage from the simulation, which is 0.24 V. If the model is reviewed it can be observed that some energy also disappears in the two branch resistors. The calculations are redone and this time the energy consumption from R2 and R3 is included in the model. This calculation gives an energy level at 146.9 J at the end of simulation which corresponds to a voltage of 0.32V. The final value is not the same as the results from the model but the difference is acceptable. The error most likely arises from the nonlinearity in the capacitance which is ignored in this calculation. Figure 76 shows the energy in the system during the test cycle.

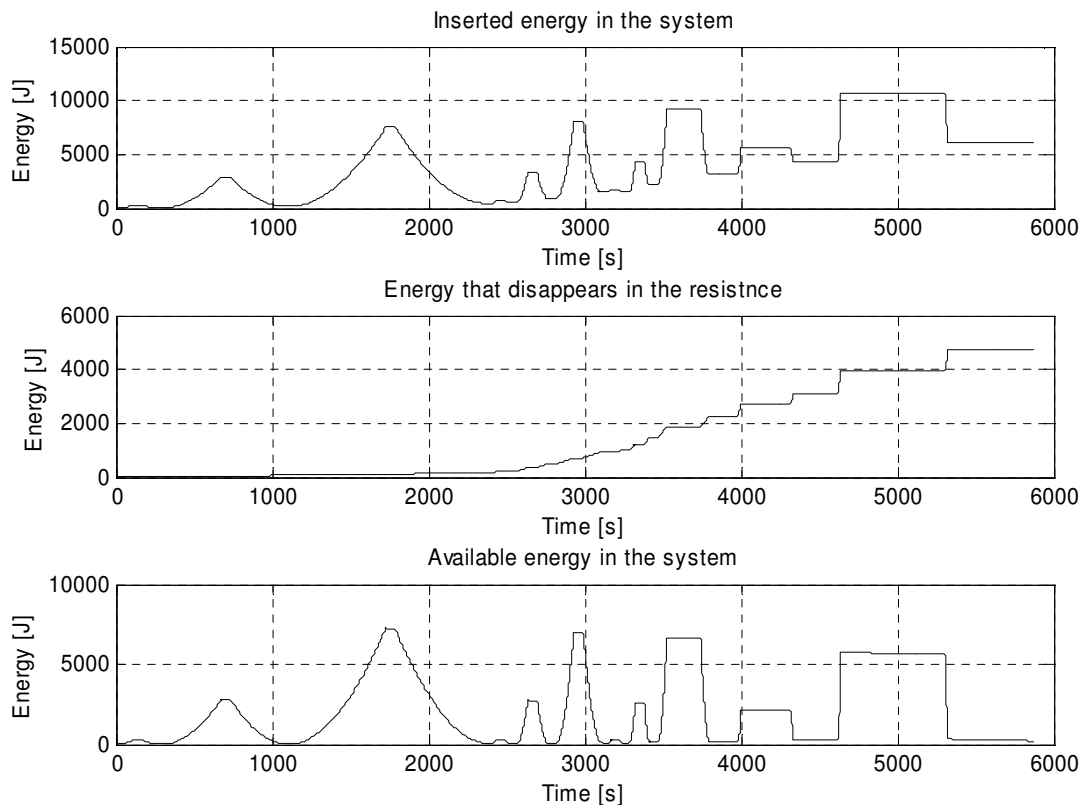


Figure 76: The energy in the system. The first one shows the energy which is calculated by measuring voltage and current at the electrodes of the capacitor. The second shows the energy that disappears in the resistive parts of the circuit and the third one shows the available energy that is stored in the supercapacitor.

7 Performance and Cost

In this chapter the performance of the programs is presented. A short discussion regarding program pricing is also presented.

7.1 Simulation Performance

One important measurement of program performance is how long it takes to complete a simulation. This can be important especially when large complex models are simulated but even for simpler models unnecessary simulation time may lead to increased costs if they are used as subcomponents in larger systems. The accuracy of simulation results is often tightly linked with the simulation time which is why it is necessary to record a value of the accuracy for resulting simulation times in order to reach any conclusions on which alternatives produce the desired results.

There are many factors that affect simulation performance. The hardware in the computer being used for simulations is one important factor. This is an issue for the program comparisons in this report because some programs are only available on one computer, while other programs must be run on a different computer. This problem occurred because of limited program licences. In each program there are often many alternatives to choose from in the simulation settings. The chosen alternatives affect the performance and accuracy of the simulation.

To measure the simulation time the DTDSM, presented in Section 5.3.2, is modelled in the five programs chosen for further evaluation. Simulations are only conducted with this model in order to enable simulation conditions that are as similar as possible. In order to find the most appropriate solver settings in the various programs several tests with different settings are conducted. The most interesting results from these tests are gathered in Table 13 where the setting alternatives, simulation time and result accuracies are shown. The model parameters used in all the tests are adapted to model the behaviour of the supercapacitor ESSP08.

Because the simulations have been conducted on two different computers, the resulting times from one of the computers are taken as the standard values. The times resulting from tests using the other computer are scaled in order to reach comparable values. To find an appropriate scaling value a simulation in Simulink was run on both computers and the time used was noted. On the first computer the simulation took 28.89 seconds to complete and on the second computer it took 12.0059 seconds. All simulations except those done using SimPowerSystems have been done on the faster second computer. So the values presented for that program are scaled by multiplying their time values with 0.416 (12.0059/28.89).

Table 13: Performance test results for the program alternatives.

Program	Solver	Time Step	Tolerance	Sim. Time [s]	Error value
Simulink	ODE45	15	1e-7	15.942	0.000251
Simulink	ODE45	15	1e-5	9.446	0.000284
Simulink	ODE45	15	1	6.825	3.900900
Simulink	ODE23t	15	1e-5	11.65	0.012700
SimPowerSystems	ODE45	15	1e-7	14.41	2.54e-4
SimPowerSystems	ODE45	15	1e-5	10.42	3.23e-4
SimPowerSystems	ODE45	15	1	8.70	3.9009
Saber	Gear	1	-	0.19	0.8532
Saber	Gear	0.1	-	5	0.0017
Saber	Gear	1e-4	-	313	0.0017
Saber	Gear	1e-5	-	40	0.0017
PLECS	ODE45	15	1e-9	23.38	2.52e-4
PLECS	ODE45	15	1e-7	10.651	2.62e-4
PLECS	ODE23	15	1e-9	31.822	2.52e-4
PLECS	ODE45	1	1e-3	8.546	3.36e-4
Dymola	Dassl	-	1e-6	41	0.0057
Dymola	Euler	-	1	20.4	4.7899
Dymola	Runge-kutta	7.28	1	20.1	14.55
Dymola	Runge-kutta	1e-4	1e-6	260	0.833

The properties of the different solvers used are not discussed since that is not a focus of the conducted comparison. They are only stated as reference to the chosen alternatives for the various simulations.

Each program contains several setting alternatives which affect these results, which makes it difficult to find the optimal settings for the programs. Therefore the values gathered from the simulations can only be used to provide an indication on program performance and accuracy. More optimal settings and model optimizations can improve results a great deal in all of the tested program alternatives. When the gathered error values are compared, it is found that Simulink reaches the lowest error value (0.000251). However, SimPowerSystems and PLECS produce similar error values. Regarding performance, it is important to study the combination of error value and simulation time in order to find the best result. This is because the fastest results often produce low accuracy. Depending on how important the accuracy is, the best performing program changes. Because of this no particular program can be stated as the top performer.

The main reason that the results for the programs are relatively similar to each other is probably that the model that is used to conduct comparisons is not complex enough to emphasize the differences between the programs.

7.2 Program Cost

When a new program is taken into consideration for use in simulations the cost of that program is one important factor. Depending on several factors the costs can vary

significantly. Several types of licenses are available depending on how the program is supposed to be used and how many users there are. Because of this it is necessary to contact the respective companies for the correct price information. Depending on the amount of licenses and additional options selected a price can then be offered.

8 Conclusions

Since the basis for this program evaluation has not been to compare programs from any specific requirements, no conclusion is drawn regarding which programs are the best alternatives. The aim of the evaluation is to find out more about the properties of the programs and state their advantages and disadvantages. With this background the most important conclusion drawn regarding program functionality is that all program alternatives, except for Simulink, are able to do electrical modelling. This ability simplifies electrical model creation since it removes some concerns of causality and also improves the circuit overview. The two programs Saber and Dymola also include the ability to create multi-domain models including areas such as electrical, hydraulic, thermal and mechanical. The programs SimPowerSystems, PLECS, PSCAD and OrCAD are all designed for electrical modelling. It can be noted that while SimPowerSystems is a toolbox for MATLAB & Simulink which enables electrical modelling, there also exist other toolboxes that extend the modelling ability to other domains such as hydraulics and mechanics.

A specific advantage of the programs Simulink, SimPowerSystems and PLECS is that they all have direct access to the MATLAB environment. The two programs Simulink and OrCAD Capture have the advantage of large user bases. OrCAD Capture and Saber are the only two programs which include characterised components. Dymola is the only program where it is possible to add a function, which can simulate Simulink models with Dymola models inside, on a different computer which does not have Dymola installed.

When comparing measurement results for the program alternatives one conclusion is that they all perform relatively similar to each other. Regarding simulation performance it is found that for simulations resulting in comparable accuracy, Saber performs fastest with a time of 5 seconds and Dymola has the longest time with 41 seconds. These times are recorded from simulations using the DTDSM adapted to ESSP08.

Regarding the created model which is named DTDSM the conducted simulations show that it is capable of accurately modelling the supercapacitor behaviour well, including its temperature dependency. The model can be adapted to different types of supercapacitors by altering its parameter values and some additional adaptations. If the accuracy of the DTDSM and the basic model are compared, it is noted that magnitude of the error is about 2.1 % of the rated voltage for DTDSM and about 5 % for the basic model.

9 Discussion

From the program evaluation no distinct conclusion is presented regarding which program that is the best choice. As mentioned in the conclusion this has not been the purpose of the conducted evaluation. However if any of the program should be taken into use, new studies must be conducted where it is investigated if it meets a set of specified demands. The evaluation conducted in this project can then be used as a base in determining which program alternatives seem interesting. The presented information about the programs is only valid for the program versions which have been investigated. This means that since all programs release new versions continuously the information about their properties will become outdated eventually.

One aspect of the modelling of a component is aging and effects of different types of usage. Depending on component age and how it is used, its behaviour changes. Such effects are not included in the model created in this project and can be important for certain applications. In most programs effects like these are not included in the various component models.

Frequency can have a large effect on the capacitance of a supercapacitor for higher frequencies, in the form of lowered capacitance. Those effects are not included in the DTDSM since it is not intended for such usage scenarios. Therefore it needs to be altered if such effects are to be studied.

A disadvantage of the DTDSM is that since it includes so many behavioural properties it also contains many parameters that must be adjusted in order to model different supercapacitors. However when the model adjustments are completed no further adjustments are needed when simulations are done using different temperatures. For supercapacitors which are not strongly affected by changes in temperature this feature is not as important and then it is possible to use a simpler model without receiving any major errors in the simulation results.

The created supercapacitor model is designed to model the measured behaviour of the actual component. It is not created using the basic physical equations, which control the various aspects of the energy storing process, as base. Such a model may be able to model the behaviour better. However since the simulation results using DTDSM are as good as they are it is hard to see any real potential for large improvements.

One opportunity for additional characterisation of the supercapacitors can be to use impedance spectroscopy measurements. Such measurement results can then be used to find parameter values for certain model structures using some fitting procedure. An example of a freely available program which uses CNLS frequency-response fitting to do this is called LEVMW.

10 Appendix A – Controlled current source

During modelling using the first model version an additional approach is evaluated. It is similar to the modelling done using a controlled voltage source but lacks some of the advantages that model has. Because of the similarities between the two model types the work done using this modelling approach is gathered here instead of being presented in the main report.

10.1 Simulink

Using (11) the capacitance can be determined using an Embedded MATLAB Function. The current can also be calculated directly in the function when the capacitance is known, if the voltage is sent through a derivative block before entering the function. This means that there are two signals going into the function block, the voltage and the voltage derivative. By multiplying the voltage derivative with the capacitance value a current value is acquired.

One thing remaining in the simulation is figuring out how to set an initial value on the voltage over the capacitance. The contents of the sub block that models the nonlinear capacitance as a controlled current source is shown in Figure 77.

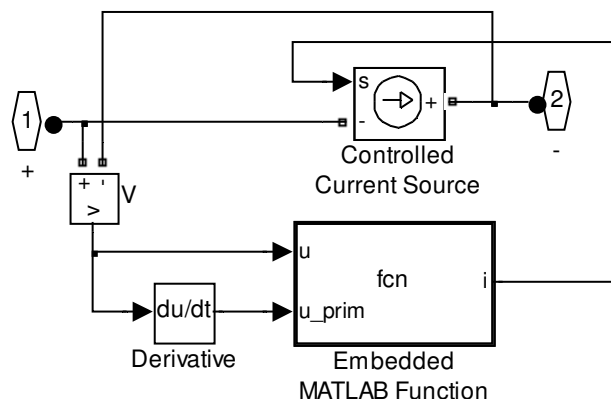


Figure 77: Sub block for nonlinear capacitance, using controlled current source.

10.2 OrCAD Capture

The main part in the model is the controlled current source, which is the part next to the resistance R2_2. It generates the current that would go through the capacitance and supplies it to the main circuit. (2) is used to calculate the current and for that the voltage derivative and the capacitance are needed. The voltage derivative is found using a separate derivative block.

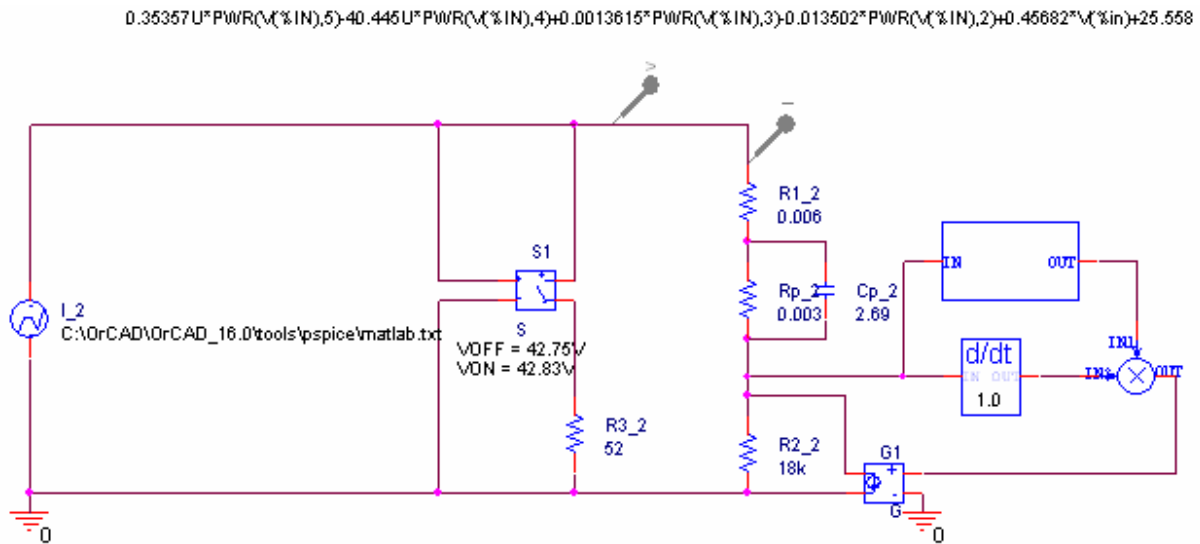


Figure 78: Model with nonlinear capacitance, using controlled current source.

10.3Saber

The model for the controlled current source is shown in Figure 79. In this model the lookup table that is used for the controlled voltage source model can not be used because of problems with mathematical loops. Instead the polynomial approach is used. To realise this polynomial standard mathematical functions such as multiplications and summations are used. This can be done more efficiently using a single expression but such a component was not found when designing the model. The method with mathematical blocks is more space consuming as well as time consuming without any real advantages.

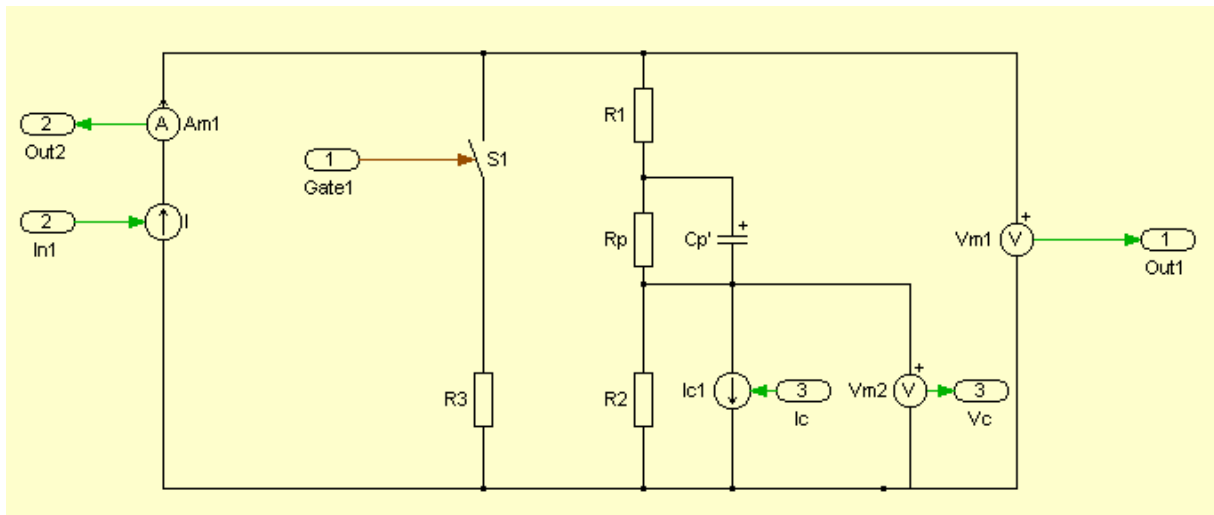


Figure 80: Nonlinear model using controlled current source.

10.5 Controlled Current Source

The components used for calculation of the nonlinear capacitance are now replaced to calculate the correct current that goes through the capacitance instead of the voltage over the capacitance. The controlled current source model in Dymola is shown in Figure 81.

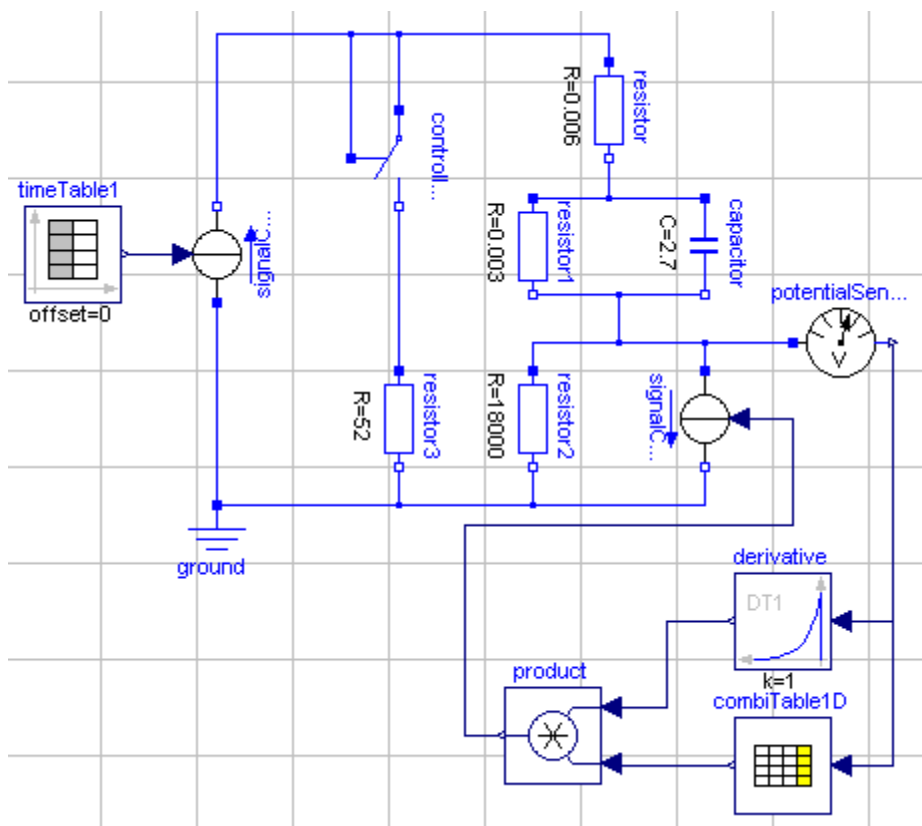


Figure 81: Controlled current source model made in Dymola.

11 Appendix B – DTDSM Figures

To improve the understanding of the DTDSM, all the included sub blocks are presented together with the main model structure for both the Simulink model and the SimPowerSystems model. These model implementations are selected since they are representative of the two alternative ways of modelling, electrical and mathematical.

11.1 Simulink

The main model structure in Simulink is seen in Figure 82.

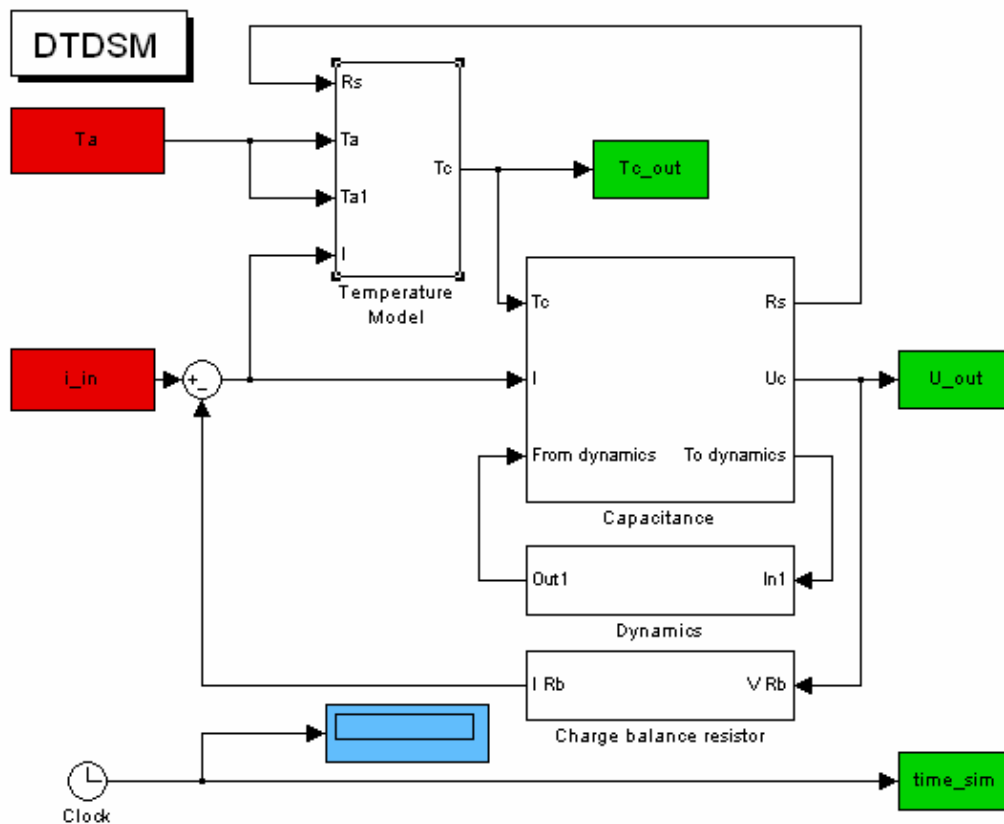


Figure 82: Main model structure in Simulink.

The Temperature model sub block in Simulink is seen in Figure 83.

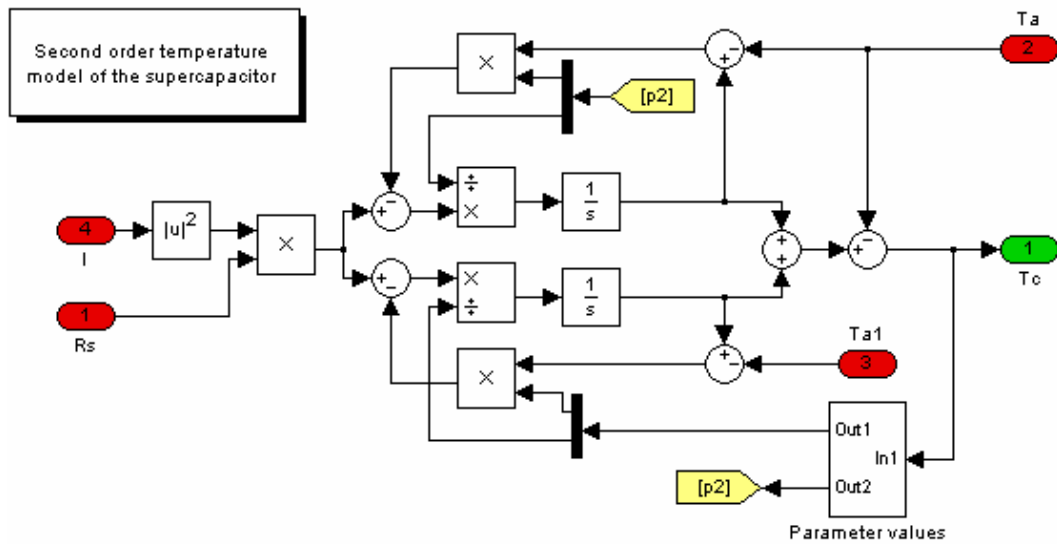


Figure 83: Temperature model sub block in Simulink.

The Parameter calculations in the Temperature model sub block in Simulink are seen in Figure 84.

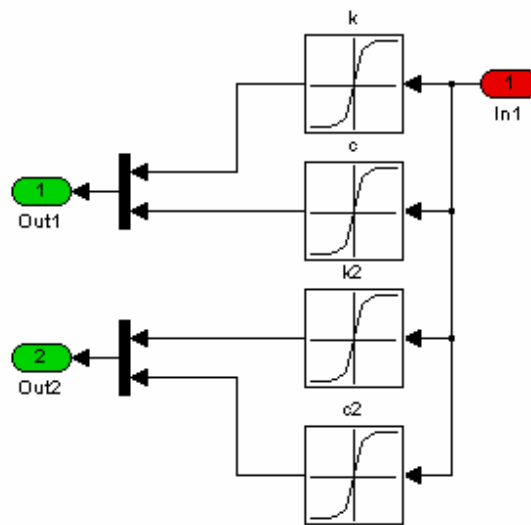


Figure 84: Parameter calculations in the Temperature model sub block in Simulink.

The Capacitance sub block in Simulink is seen in Figure 85.

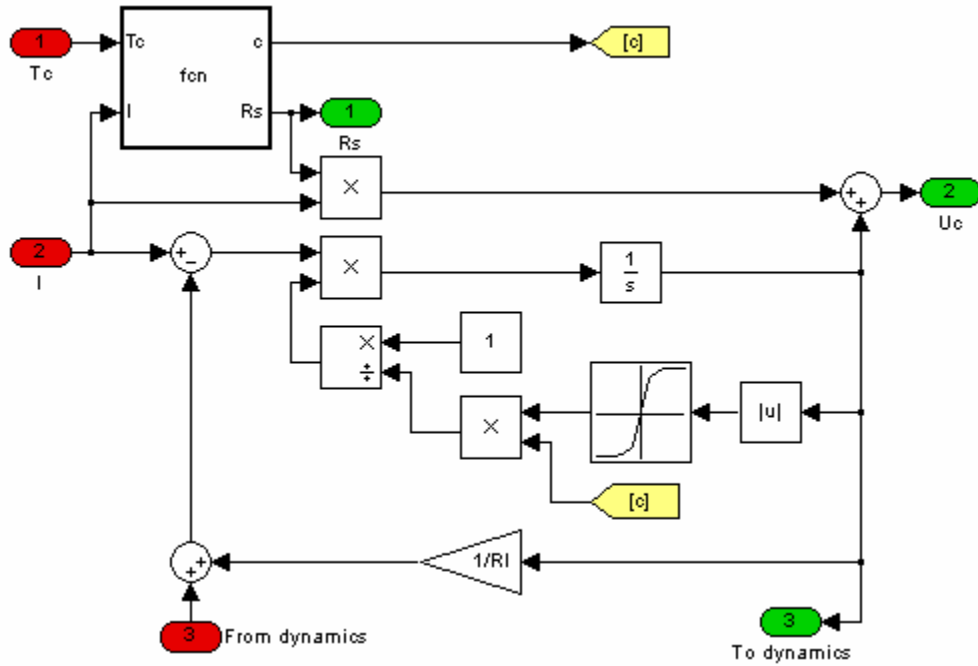


Figure 85: Capacitance sub block in Simulink.

The Dynamics sub block in Simulink is seen in Figure 86.

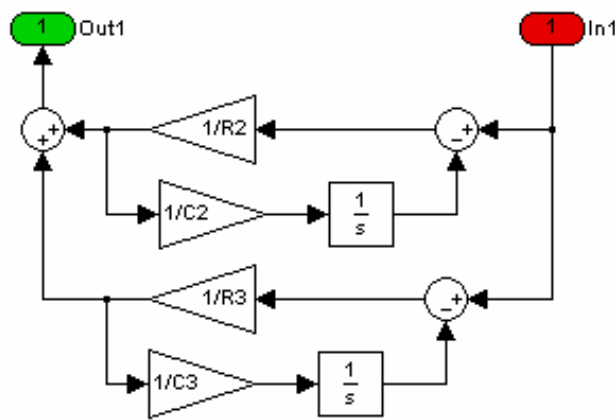


Figure 86: Dynamics sub block in Simulink.

The Charge balance resistor sub block in Simulink is seen in Figure 87.

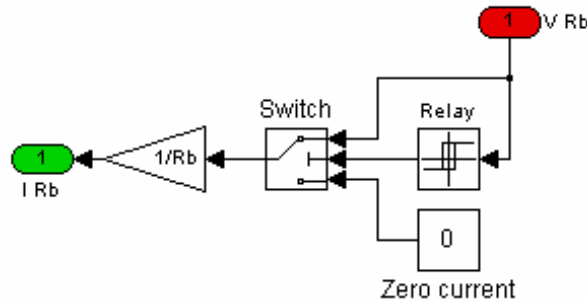


Figure 87: Charge balance resistor sub block in Simulink.

The code used in the MATLAB embedded function is presented below. The code is used to conduct calculations in the Capacitance sub block.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This file calculates the series resistance
% of the supercapacitor. The model describes
% how the temperature of the capacitor are related
% to the series resistance.
% The file also calculates a correction factor
% for the capacitance value when the temperature
% is changed.
% Input parameters:  Tc: Temperature
%                   I: Input current.
% Output parameters: c: Correction factor for capacitance
%                   Rs: Series resistance.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [c, Rs] = fcn(Tc, I)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  ESSP08  %%%%%%%%%
%Rs
A08 = 0.0013;
lamda08 = 0.0219;
Rs = A08*exp(-lamda08*Tc);

%Capacitance
p1 = 0.002161;
p2 = -0.06140;
p3 = -0.18243;
p4 = 2701.990;

cor = p1*Tc^3+p2*Tc^2+p3*Tc+p4;
c = cor/2700;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  ESSP16  %%%%%%%%%
% Rs
% A08 = 6.9236e-004;
% lamda08 = 0.0019;
% Rs = A08*exp(-lamda08*Tc);
% Rs = 1.5e-3;
%
% p1 = 0.000893779704902236;
% p2 = -0.0156784257320508;
% p3 = -0.448472202140165;
% p4 = 2745.08406007808;
%
% cor = p1*Tc^3+p2*Tc^2+p3*Tc+p4;
%

```

```
%      c = cor/2600;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ESPP48  %%%%%%%%%%
% c=1;
% %% Rs determination
%
% A48 = 0.0132558775268682;
% lamda48 = 0.0114837292314154;
% Rs = A48*exp(-lamda48*Tc);
% %% Temperature correction factor for C value
% p1 = 5.22252863079299e-06;
% p2 = 7.56398145320391e-05;
% p3 = -0.00805279160400273;
% p4 = 35.5795106609314;
% cor = p1*Tc^3+p2*Tc^2+p3*Tc+p4;
% c = cor/35.6;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

11.2 SimPowerSystems

The main model structure in SimPowerSystems is seen in Figure 88.

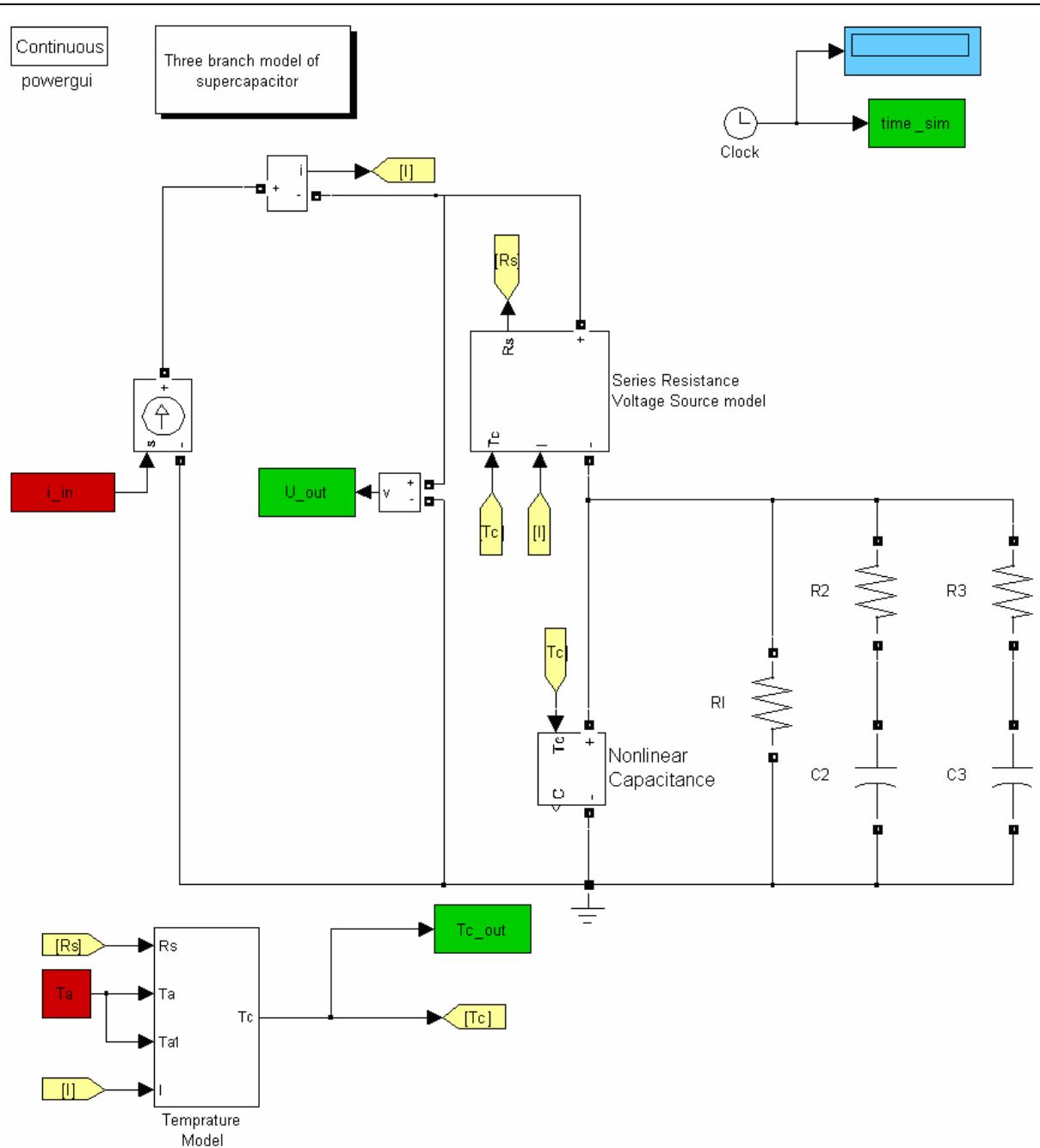


Figure 88: Main model structure in SimPowerSystems.

The Temperature model sub block is the same that is used in Simulink.

The Nonlinear capacitance in SimPowerSystems is seen in Figure 89.

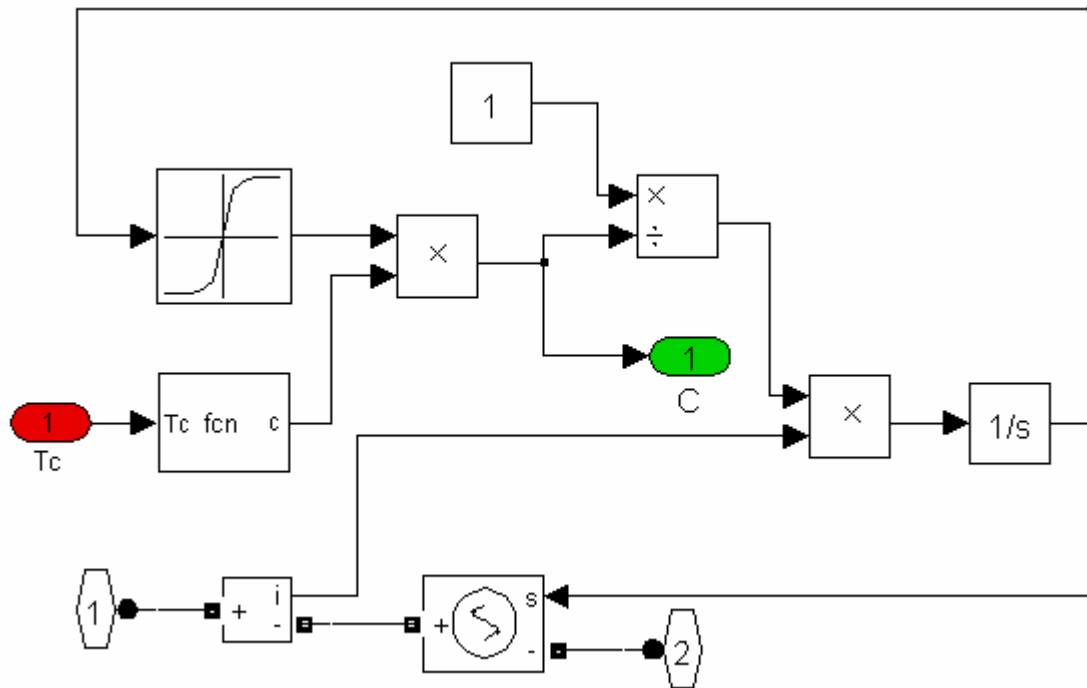


Figure 89: Nonlinear capacitance sub block in SimPowerSystems.

The Nonlinear resistance in SimPowerSystems is seen in Figure 90.

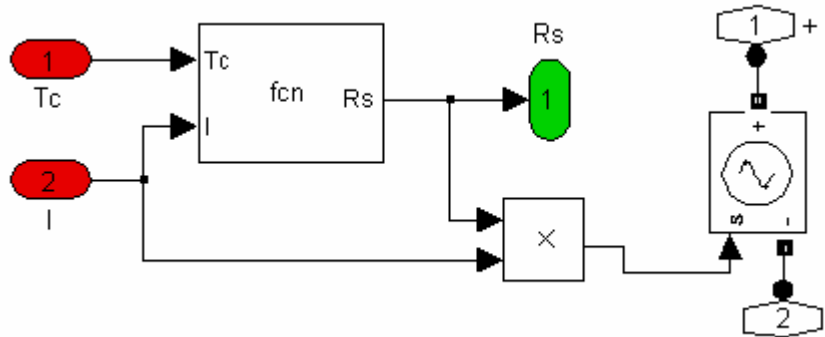


Figure 90: Nonlinear resistance sub block in SimPowerSystems.

12 References

- [1] Faranda, R.; Gallina, M.; Son, D.T.; A new simplified model of Double-Layer Capacitors 2007; ICCEP '07. International Conference on Clean Electrical Power 21-23 May 2007 Page(s):706 – 710; ISBN: 1-4244-0632-3
 - [2] Karlsson, A, 2005, Evaluation of Simulink/SimPowerSystems and other Commercial Simulation Tools for the Simulation of Machine System Transients, 2008-01-22, <http://www.ee.kth.se/php/modules/publications/reports/2005/IR-RT-EX-0505.pdf>
 - [3] F.BELHACHEM, S.RAL, B.DAVAT, 2000, A physical based model of power electric double-layer supercapacitors.pdf, 2008-10-09, http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=882604&isnumber=19086
 - [4] Wikipedia, 2007-12-12, Wikipedia, <http://en.wikipedia.org/wiki/Acetonitrile>, 2008-01-07
 - [5] Stanley Atcitty, 2006, Electrochemical Capacitor Characterization for Electric Utility Applications, 2008-01-07, http://scholar.lib.vt.edu/theses/available/etd-11162006-151143/unrestricted/AtcittyDissertation_11-29.pdf
 - [6] David A. New, John G. Kassakian, Joel E. Shinddall, Double Layer Capacitors: Automotive Applications and Modelling, 2008-01-08, http://lees.mit.edu/lees/posters/double_layer_capacitors_poster.ppt
 - [7] Hamann, C, Hamnett, A, Vielstich, W 1998, Electrochemistry, Wiley-VCH, Weinheim Germany, ISBN: 3-527-29095-8
 - [8] Riccardo Signorelli, Joel Schindall, John Kassakian, 2005 ,Carbon Nanotube Enhanced Double Layer Capacitor, 2008-01-06, http://lees.mit.edu/lees/posters/RU13_Oct05.pdf
 - [9] Richard Ball, 2006, Electronicsweekly.com, 2008-01-06, <http://www.electronicsweekly.com/Articles/2006/03/01/37810/Supercapacitors+see+growth+as+costs+fall.htm>
 - [10] The MathWorks, 2007, SimPowerSystems Introduction, 2008-01-10, <http://www.mathworks.com/products/simpower/description1.html>
 - [11] van der Sluis, L. 2001; Transients in Power Systems; John Wiley & Sons, Ltd; West Sussex, England; ISBN: 0-471-48639-6; page 153
 - [12] Molin, B. 2003; Simulering med Pspice; Studentlitteratur; Lund, Sweden; ISBN: 91-44-03150-5; page 60
 - [13] EMTDC Help in PSCAD, section: “EMTDC vs. Other EMTP-Type Programs”
 - [14] van der Sluis, L. 2001; Transients in Power Systems; John Wiley & Sons, Ltd; West Sussex, England; ISBN: 0-471-48639-6; page 137
 - [15] EMTDC Help in PSCAD, section: “Introduction for EMTDC “
 - [16] Alimelng, J.H.; Hammer, W.P. 1999; PLECS - Piece-wise Linear Electrical Circuit Simulation for Simulink; 2007-10-24; <http://ieeexplore.ieee.org/iel5/6444/17234/00794588.pdf>; pages 2-3
 - [17] Elmqvist H., Otter M., and Cellier F.E. 1995, Inline Integration: A New Mixed Symbolic/Numeric Approach for Solving Differential-Algebraic Equation Systems, Keynote Address, Proceedings ESM'95, European Simulation Multiconference, Prague, Czech Republic, 5. - 8. Juni, pages. 23-34.
 - [18] Claytex 2008, Simelica - the Simulink to Modelica Translator, 2008-01-10, <http://www.claytex.com/Default.aspx?tabid=51>
-

References

- [19] Data sheet for supercapacitor from EPCOS with Part No.: B48621-S0203-Q288
- [20] Dorf Richard C, Svoboda James A, 2004, Introduction to electric circuits, John Wiley & Sons, ISBN 0-471-44795-1, page 242
- [21] Dorf Richard C, Svoboda James A, 2004, Introduction to electric circuits, John Wiley & Sons, ISBN 0-471-44795-1, page 245
- [22] Wikipedia, 2008-01-06, Capacitance, 2008-01-10, <http://en.wikipedia.org/wiki/Capacitance>
- [23] Coby Bassett 1999; A Nonlinear Capacitor Model for Use in PSpice; 2007-10-24; <http://www.cadence.com/appnotes/ANonlinearCapacitorModelForUseInPSpice.pdf>
- [24] Cadence 2000; Reference Guide for PSpice; 2007-10-03; <http://www-classes.usc.edu/engr/ee-ep/326/pspceref.pdf>; page 18
- [25] Ljung, L. & Glad, T. 2004; Modellbygge och Simulering, Second Edition; Studentlitteratur; ISBN 91-44-02443-6. page 287