Gerhard Benedikt Weiß, BSc

Improvements to the battery charging process during start-stop events of passenger cars

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Electrical Engineering

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn., Roland Seebacher

Electrical Drivers and Machines Institute

Dipl.-Ing., Alexander Mori, Infineon Technologies Austria AG
Dipl.-Ing. Dr.techn., Christoph Seidl, Infineon Technologies Austria AG

Graz, January 2016

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis dissertation.

Date

Signature

Acknowledgements

First of all, I would like to express my deepest sense of gratitude to my supervisor Professor Seebacher from Graz University of Technology Electrical Drives and Machines Institute. I thank him for the systematic guidance, continuous advice and encouragement throughout the course of my thesis.
My sincere thanks also go to DI Mori Alexander and Dr. Seidl Christoph from Infineon Development Center Graz, who made this thesis possible and supported me as well.
I am grateful to Ludovica Cintio and Victoria Payne for correcting my broken English sentences.
Last but not least, I would like to thank my parents, who gave me the opportunity to study and finally write this thesis.

Danksagung

Zunächst möchte ich mich bei Professor Seebacher von der Technische Universität Graz Institut für Elektrische Antriebstechnik und Maschinen für die wissenschaftliche Betreuung meiner Masterarbeit bedanken. Vor allem für seine kontinuierliche Unterstützung, Engagement und Expertise bin ich sehr dankbar.
Großer Dank gilt auch DI Mori Alexander und Dr. Seidl Christoph von Infineon Development Center Graz, welche diese Arbeit ermöglicht haben. Ihre Unterstüzung habe ich auch sehr geschätzt.
Herzlich bedanken möchte ich mich auch bei meinen Englisch Korrekturleserinnen Ludovica Cintio und Victoria Payne.
Zu guter Letzt bedanke ich mich bei meinen Eltern, die mir die Möglichkeit gegeben haben, zu studieren und schlussendlich diese Arbeit zu verfassen.

Abstract

Title: Improvements to the battery charging process during start-stop events of passenger cars

Keywords: Alternator, Vehicle Electrical System, Load Response Control, Bang-Bang Control, Delay Time Minimization, Enhanced Charge Control

Based on a patent idea from Bayrische Motoren Werke (BMW), an algorithm is developed to enhance the battery charging process in automobiles. The aim is to minimize the delay time of the charging process under certain constraints. Neither analytical calculations, measurements nor simulations have been done so far.
In order to ascertain the potential of the patent, application measurements with an *Infineon* alternator integrated circuit (IC) are conducted in the first stage. The main part deals with the algorithm development and implementation in Matlab/Simulink. Possibilities and constraints of the IC are considered. For this purpose, a simplified behaviour model of the current alternator controller is designed, as well as an additional bang-bang controller. The latter and the manipulation of the alternator controller are the main focus of this work. Analytical and numerical calculations in respect of alternator speed and load are done. These calculations are used to determine the control parameters. A dynamic model of the alternator is used for verification simulations. The final outcome is a configurable simulation test bench. A comparison using the conventional controller and the enhanced algorithm is given at the end.

Kurzfassung

Titel: Untersuchungen zur Verbesserung des Batterie-Ladebetriebs bei Start-Stopp Vorgängen in Kraftfahrzeugen

Schlagwörter: Lichtmaschine, Personenkraftwagen (Pkw)-Bordnetz, Momentenrückwirkung, Zweitpunktregler, Totzeit Minimierung, Verbesserte Ladekontrolle

Basierend auf einer Patentidee von BMW wurde ein Algorithmus zur Verbesserung des Batterie-Ladevorgangs in Pkws entwickelt. Das Ziel ist die Minimierung der Totzeit, bis die Batterie von der Lichtmaschine geladen wird. Gleichzeitig wird die Momentenrückwirkung der Lichtmaschine auf den Verbrennungsmotor begrenzt. Bisher wurden dazu weder analytische Berechnungen noch Messungen oder Simulationen durchgeführt.
Um das Potential des Patentes zu ermitteln, wurden als Erstes Applikationsmessungen mit einem *Infineon* Lichtmaschinenregler durchgeführt. Der Hauptteil dieser Arbeit beschäftigt sich mit der Algorithmus-Entwicklung und -Implementierung in Matlab/Simulink. Dabei werden die Möglichkeiten und Beschränkungen des *Infineon* IC's berücksichtigt. Zu diesem Zweck wurden ein vereinfachtes Verhaltensmodell des aktuellen Lichtmaschinenreglers und ein zusätzlicher Zweipunktregler entworfen. Letzterer und die Manipulation des bisherigen Reglers sind Schwerpunkte dieser Arbeit. Mithilfe von analytischen und numerischen Berechnungen wurden die Parameter des Zweipunktreglers bestimmt. Rotorgeschwindigkeit und elektrische Last wurden dabei miteinbezogen. Für Verifikationssimulationen wurde ein dynamisches Modell der Lichtmaschine verwendet. Das Endergebnis ist eine "simulation test bench", welche von AnwenderInnen konfiguriert werden kann. Ein Vergleich zwischen dem bisherigen Regler und dem verbesserten Algorithmus wird am Ende präsentiert.

Contents

Acronyms

BLZ	Blind-Zone Value
BMW	Bayrische Motoren Werke
DC	Duty Cycle
ECU	Engine Control Unit
ECC	Enhanced Charge Control
EMF	Electromotive Force
IC	Integrated Circuit
ADC	Analog Digital Converter
LIN	Local Interconnect Network
LRC	Load Response Control
NVM	Non-Volatile Memory
PI	Proportional Integral controller
Pkw	Personenkraftwagen
PSB	Phase Signal Boost
PWM	Pulse-Width Modulation
FT	Fall Time
RT	Rise Time

1 Preface

1.1 Motivation

Nowadays (modern) passenger cars with a combustion engine are often equipped with an intelligent Start-Stop system. As described in [14], the basic idea is to save fuel and reduce emissions by turning off the engine automatically if the car stands still. More specifically, the Engine Control Unit (ECU) controls the start and stop system. If the engine runs in idle for a certain time and some other conditions are fulfilled (standstill, engine temperature,...), the ECU shuts the engine down. Usually it will be started again by pushing the clutch pedal. It is shown that the fuel saving rate is up to 8.31% in urban driving condition by using an intelligent Start-Stop system.
On the other hand, the electrical grid is powered by the automotive battery if the engine is off. There is no way out, hence the alternator can not produce energy without rotation. Moreover, each engine start heavily stresses the battery hence a high current (up to $500A$) is needed for the starter system and the ignition. From this point of view, the battery should get charged as fast as possible by the alternator after the engine starts. The point is, if the alternator produces energy to charge the battery, the engine load torque is increased. In short, charging the battery is on torque. Figure 1.1 gives an overview of the power flow. Especially during the start process, engine torque is needed for the acceleration. For this reason, most alternator control ICs like the Infineon TLE8880 have a special function called Load Response Control (LRC). This function prevents abrupt torque loading in case of low revolutions. However, it has the disadvantage of delaying the charging process. Measurements conducted by BMW have shown that it takes several seconds for the battery to charge. During the delay time the battery gets more and more discharged by the electrical loads.
For this purpose, BMW patented an idea to minimize the delay time and still assure a gently raising torque of the alternator (see [12]). The overall goal is to increase the number of the battery charge cycles (life-time), hence the automatic Start-Stop system entails an increase of engine starts. However, there are no measurements, simulations or analytical investigations done to proof the patent idea, nor an approach as to how to implement it.

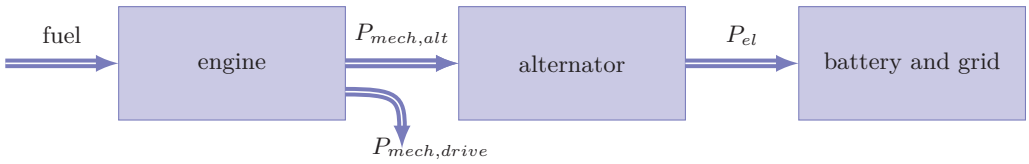


Figure 1.1: Power flow $P_{mech} = \tau_{eng}\omega_{eng}$

This thesis is based on close cooperation between Infineon and BMW. The aim is to prove the patent idea and to develop a feasible IC concept. It was supervised by Infineon and Graz University of Technology Electrical Drives and Machines Institute.

1.2 Aims of this work

- Investigations of the patent potential (in best case) by application measurements using the current alternator IC
- Development of a concept that meets the aim of the patent idea, considering restraints of the IC
- Implementation of the algorithm in Matlab/Simulink
- Analytical and numerical calculations in order to determine parameter of the algorithm
- Verification simulations with an existing alternator model
- Further alternator measurements in order to verify model simplifications regarding the algorithm
- Development of a voltage controller based on the alternator IC (including LRC)
- Simulations of the complete system: alternator, developed algorithm and manipulation of the voltage controller
- Comparison between closed loop simulations using the IC based controller and using the enhanced version

1.3 Chapter structure

Chapter 2 gives an overview of the electrical grid in the auto mobile. The main components are briefly explained. Furthermore, the BMW patent idea is presented including some simulations from chapter 4.
Investigations concerning the phase signal and the charging condition are presented in chapter 3. It contains also application measurements with the current IC.

The main part is described in chapter 4. It explains the concept of the patent implementation and all development steps of the final algorithm. Each step contains analytical and numerical calculation. Also test and verification simulations are given.

Chapter 5 presents further alternator measurements, which are relevant for the developed algorithm. Simulation results of the complete system are shown and discussed in chapter 6. Finally, a conclusion of the whole thesis is given in chapter 7.

The appendix shows the implementation of the complete system in Matlab/Simulink.

2 Introduction

2.1 Electrical System

A brief overview of the electro-mechanical system is shown in figure 2.1. The alternator is driven by the combustion engine via a V-belt. As expressed in [8] pages 384-385, it is used to convert mechanical energy into electrical energy. Usually, the speed ratio a_{mech} ranges from 1:2 to 1:3. The battery is an energy accumulator and the electrical loads are consumer. There are systems with two different batteries (a starter battery and a general-purpose battery). Nonetheless, for this thesis, systems with one battery are considered. Hence, it has to supply both the starter motor and other electrical components they are combined to one load in figure 2.1.

$$\omega_{mech,alt} = a_{mech}\omega_{eng} \quad \forall\ a_{mech} \in [2\ 3] \tag{2.1}$$

In general, a stable grid voltage V_{ba} is demanded. However, the induced voltage, also called Electromotive Force (EMF) depends among others on the rotational velocity $\omega_{alternator}$. For that reason only, it is obvious that a voltage controller is needed to keep V_{ba} constant. In addition, depending on load and battery state, the needed output power varies.
As seen in figure 2.1, the brush-holder contains the alternator IC (indicated by a switch). Carbon-brushes are used to apply current to the slip-rings of the rotor.

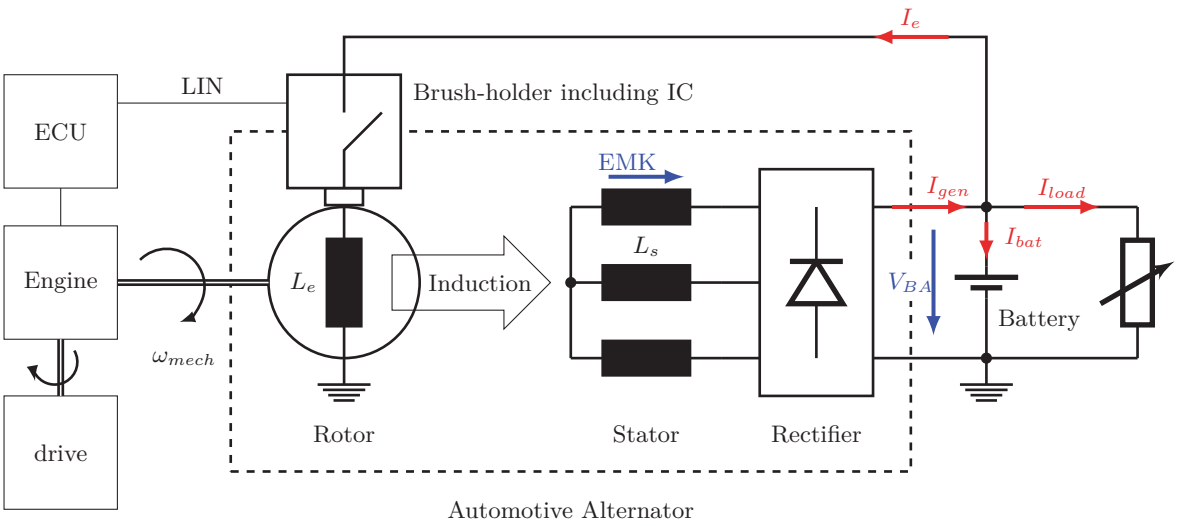


Figure 2.1: Overview electro-mechanical system

2.1.1 Alternator

Most manufacturers publish little information about the alternator. However, many investigations have been done on automotive alternators in the past. From those, it is known that a claw-pole synchronous generator,also called "Lundell" alternator, is coupled to a passive rectifier(see [15] page 1).
Figure 2.2 shows the principle electrical circuit. R_c is the contact resistor to the slip rings (carbon brushes). The filter capacitor C_s is usually located in the brush-holder.

Construction: "The excitation coil is wound axially on the rotor. This coil is surrounded by two solid iron pole pieces, or claw poles, and is fed from the stator via a pair of slip rings. The stator is composed by a slotted laminated iron core and a three-phase winding, wye or delta connected. Usually six diodes in a full-bridge configuration are used to rectify the output current."([11] page 1-2) It is important to point out that the rectifier and the generator are in the same cage and it is not designed to unsolder the connection. The brush-holder is exchangeable.

Operation: Basically, a current is applied to the excitation coil which produces a magnetic field. By rotation this field induces an AC voltage in the stator winding(EMF). As mentioned above, the three-phase current is rectified by the diodes if a load is connected. Nevertheless, there is a ripple in the output voltage V_{ba} and current I_{gen}.

Characteristics: "The output power is controlled by regulating the field current. Generally, Lundell alternators are characterized by low efficiency and low manufacturing cost."([11] page 2) The typical output power capability is about 3kW (see [15] page 1) and the operational speed ranges from 1600rpm to 15 000rpm(see [11] page 2).

Further information concerning the alternator can be found in [8] pages 434-461.

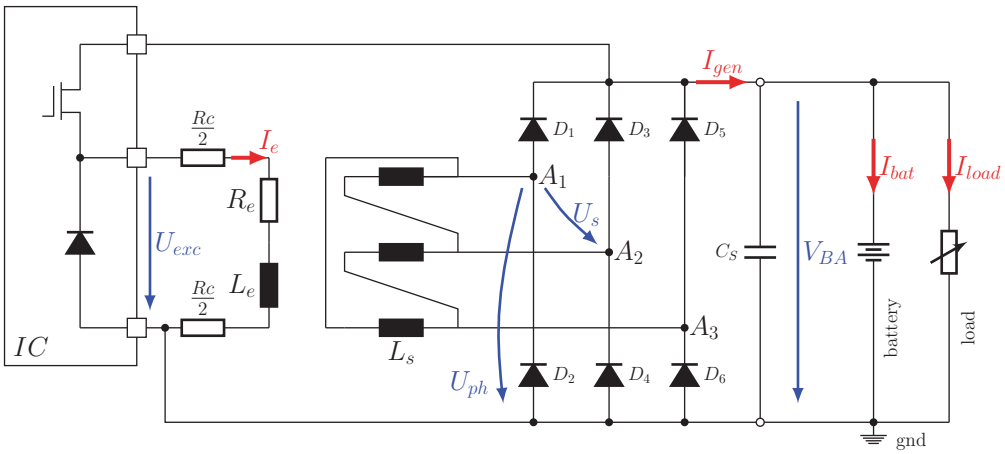


Figure 2.2: Principle electrical circuit

2.1.2 Battery

As explained in [5] page 275, the automotive battery is a rechargeable electrochemical unit. It converts electric current into a modified chemical compound in order to store energy. This chemical reaction can be reversed to release current. The tasks of the battery are (see [8] page 400) to:

1. provide energy for the engine start,
2. provide energy while the engine is running if the output power of alternator is too low,
3. save energy if the consumption of the load is lower than the produced energy,
4. power the electrical components if the engine is off
5. dampen voltage peaks in the grid

For the engine start, a high current (300A to 500A see [8] page 390) is produced by the battery. When the engine is running, the alternator normally produces enough electricity to power the loads and to charge the battery. That means the generator output current I_{gen} is greater than the load current I_{load}, otherwise the battery gets discharged.

Many automotive batteries are "using lead plates in a dilute sulphuric acid electrolyte".([5] page 275). For this reason, it is common to say 'lead-acid' batteries. They consist of six cells in a series. Each has a nominal voltage of $2.1V$, which gives a nominal battery voltage of $U_N = 12.6V$. In [8] pages 418-425 the battery characteristics are well described, the following explains the most important parameters:

Open-circuit voltage U_0: In contrast to the nominal value, U_0 is the actual voltage across the unloaded battery in steady-state. It is dependent on the state of charge and the electrolyte temperature.

Internal resistance R_i: is the sum of the cell resistances and internal connecting elements. Similar like the U_0 it depends on the state of charge and other conditions.

Terminal voltage U_k: is the measured voltage at the terminals of the battery under load. It depends on the previous parameters and discharge/load current (equation 2.3).

$$I_{bat} = -I_{load} \quad \forall \quad I_{gen} = 0A \tag{2.2}$$

$$U_k = U_0 - I_{load} R_i \tag{2.3}$$

Available capacity K: is "the quantity of electricity which the battery can deliver under specified conditions"([8], page 420). It depends on the state of charge and temperature ϑ. The recent charge $Q(t)$ is given by:

$$Q(t) = \int_{t_0}^{t} I_{bat}(\tau)\, d\tau + K(t_0, \vartheta) \tag{2.4}$$

2.1.3 Load

A wide range of electrical devices are installed in a conventional passenger car but not all of them are switched on at the same time. Generally the switch-on durations vary. There are a few continuous consumers like, the electric fuel pump, some long-time consumers for example the car sound system, and many short-term consumers like the turn-signal lamps or the electric-power window. Figure 34 on page 461 in [8] gives an overview grouped by the switch-on duration. On page 385 a table is shown with the power input values and the average electrical load requirements of each component. According to this, there is a total amount of $1145W$ which is installed but the average value is $600W$.

2.1.4 Alternator IC

As a constant grid voltage is required, the main function is to control the alternator output voltage V_{BA}. However there are several other tasks, like the LRC, which are covered by the IC. The whole thesis refers to the **Infineon TLE8880**, basic information can be found in [4]. Relevant functions regarding the topic of this thesis are explained briefly in this section. A simplified block diagram is shown in figure 2.4.

Measured Quantities

- V_{BA} ... rectified output voltage, measured from V_{BA} to gnd, analog low pass filtered and converted to a digital signal.
- U_{ph} ... induced voltage referred to gnd, it is not converted to digital signal but compared with internal reference values by (three) comparators (no Analog Digital Converter (ADC) available).
- I_e ... excitation/field current, measurement is implemented in the free wheeling path (conducting during pulse-off time) and converted to a digital signal.

Closed-Loop Voltage Control

Since the induced stator voltage depends on the field strength and rotation, the output voltage is controlled by regulating the field current. For this purpose, a fixed frequency Pulse-width Modulation (PWM) voltage is used to adjust the average current. The duty cycle (DC) of the PWM is calculated by a digital Proportional Integral (PI)-controller which compares V_{BA} with the desired set voltage V_{set}. Via serial network protocol (LIN) interface V_{set} is adjustable from 10.6V to 16V. As seen in figure 2.4, the output stage is a high-side switch. The amplitude of the PWM is V_{BA} itself. From a control systems view, V_{BA} is the controlled variable and I_e is the manipulated variable. However, the controller output is the DC of the PWM voltage.

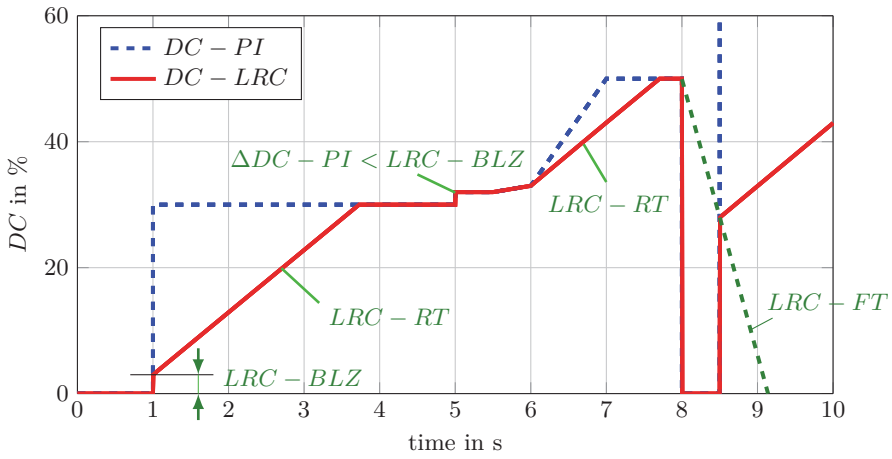


Figure 2.3: Simulation of the LRC function with $LRC-BLZ = 3\%$, $LRC-RT = 5s$, $LRC-FT = 1s$

Load Response Control

As already mentioned in section 1.1, the LRC prevents engine speed hunting and vibration due to sudden electrical loads which cause abrupt torque loading of the engine at lower speeds. The alternator torque is proportional to the output current I_{gen}. The idea is to limit the rise gradient of I_{gen} which in turn is dependent on the field current. From this it follows that limiting the excitation DC 's rate of rise leads to gradient limitation of I_{gen}. In other words, the LRC is a limitation of the PI-controller output. Comparing figure 3.9 (measurement LRC deactivated) and 3.8 (measurement $LRC - RT = 10s$ activated) in section 3.2 the effect of the LRC is illustrated. Figure 2.3 shows a simulink-simulation of LRC-function based on LRC in the IC. In this case the input $DC - PI$ is an arbitrary trace just to demonstrate how the LRC works.

LRC-BLZ: If the DC change of the PI-controller is smaller than the LRC-Blind-Zone Value (BLZ), the LRC gradient limitation is not active. In case of an increase greater than the $LRC - BLZ$, the output value is the sum of the current DC and the blind-zone value, further on the LRC gradient is applied. This can be seen at the beginning in figure 2.3. The aim of the $LRC - BLZ$ is to improve the dynamic in case of small DC changes. The customer has the possibility to choose 3%, 6% or 12% as a blind-zone value by programming the Non-Volatile Memory (NVM) of the IC.

LRC-RT: The LRC-Rise Time (RT) is the applied gradient limitation in case of DC increase greater than the $LRC - BLZ$. It is defined as the ramp-up time and adjustable from $0s$ till $15s$ via LIN. For example, $LRC - RT = 5s$ means to go from $DC = 0\%$ to $DC = 100\%$ in $5s$.

LRC-FT: On the contrary, the LRC-Fall Time (FT) is defined as the ramp-down time to go from $DC = 100\%$ to $DC = 0\%$. It is calculated internally and not seen at the output stage. Nevertheless, if there is a sudden DC change from low to high, the LRC-ramp is allowed to start at the last internal calculated LRC fall value. This is shown at the end in figure 2.3. The aim is to prevent a restart from low DC in case of a short time load throw off. The $LRC - FT$ value is configurable $1s$ or $2s$ by programming the NVM.

LRC-DIS: The LRC-function is disabled if the measured rotor speed is above the $LRC - DIS$ (disable speed) threshold. In case of no communication, the threshold value is the NVM-value (programmable $3000rpm$ or $4000rpm$), otherwise adjustable between $2400rpm$ and $8000rpm$ via LIN.

Phase Signal Boost

The phase voltage is used for the determination of the rotor speed. If LRC is enabled, it can happen that the amplitude of U_{ph} is very low at the regulation start. To assure a proper signal, Phase Signal Boost (PSB) is activated if the amplitude is lower than a dedicated threshold value. While it is active, the following actions are repeated (until U_{ph} exceeds the threshold value):

1. Exciation PWM DC is set to 100% for $t_{PSB,ON}$
2. Exciation PWM DC is set to 0% for $t_{PSB,OFF}$

The values of $t_{PSB,ON}$ and $t_{PSB,OFF}$ can be found in the data-sheet. Obviously, PSB has a higher priority than LRC and therefore overrules the output stage. It is important to point out that the LRC-ramp itself is not changed.

Speed measurement

By measuring the (electrical) frequency f_{ph} of the phase signal U_{ph} and using the configurable pole pair number p, the rotor speed is determined. The end-user has to program the NVM value according the alternator. The measurable speed ranges from $500rpm$ to $24000rpm$.

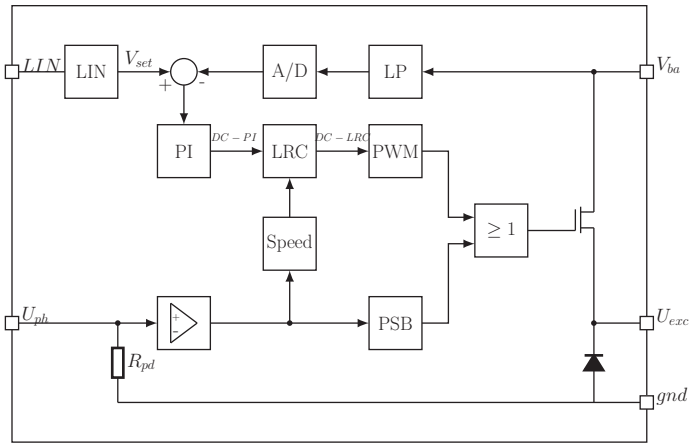


Figure 2.4: Block diagram

2.2 Patent BMW

The patent [12] is published by BMW. As mentioned at the very beginning, it is an idea without any proof. However, for the explanation here, simulations are used which have been developed within this thesis. Later on it will be shown how they work. At the moment only the functionality is focused (as it is in the original patent). Therefore, three simulations are presented. Since the patent contains no timing values or other data, they are omitted here as well.

Figure 2.5 shows a simulation with disabled LRC. Hence, there is no gradient limitation of the controller output, the current I_{gen} increases rapidly and would cause a sudden torque load. Nonetheless, there is a (relative small) delay time between start of regulation and $I_{gen} > 0A$. This is because of the rotor time constant.

On the contrary, figure 2.6 presents a simulation with enabled LRC as it is state of the art. Obviously the delay time has increased significantly and the steady-state value is reached much later. Likewise, I_{gen} is rising gently, which is a better treatment for the engine. It can also be seen that it is not necessary to increase I_e in a slow manner until $I_{gen} > 0A$. This is the crucial point of the patent idea. While $I_{gen} = 0A$, I_e can increase fast without any load response effect. As soon as $I_{gen} > 0A$, the further increase of I_e must follow the LRC gradient.

Figure 2.7 shows exactly the described behaviour of the patent. The LRC is set to 100% until $I_{gen} > 0A$. Subsequently, I_e increases as fast as possible. Once $I_{gen} > 0A$, the LRC gradient is applied and I_{gen} rises gently. This procedure combines both advantages, a minimal delay time and prevention of sudden torque loading. It is in nature of the LRC that the steady-state value of I_{gen} is reached later. Through the patent improvement, the time span is reduced, hence the delay time is minimized.

However, the last simulation (patent) is contrived in order to demonstrate the idea. It

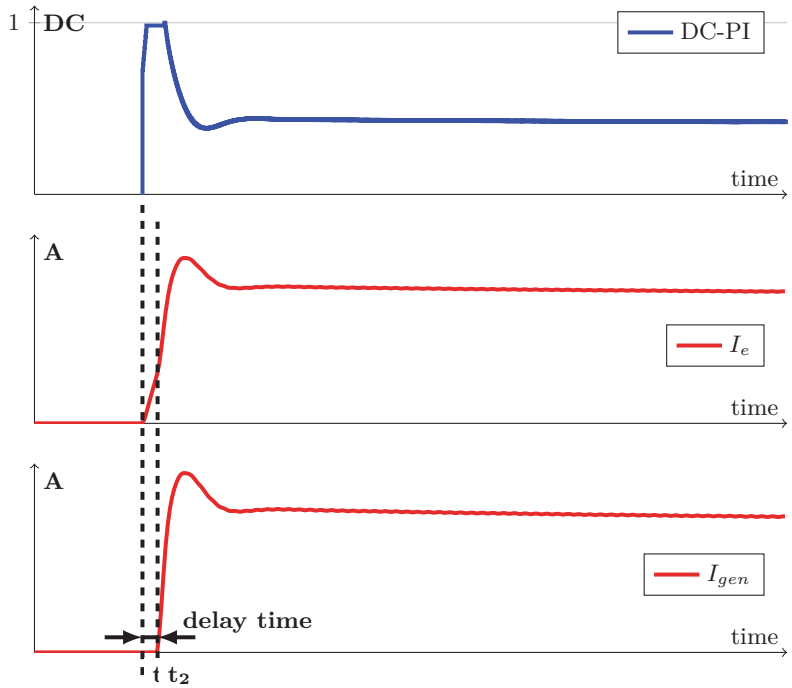


Figure 2.5: simulation LRC disabled

uses the condition $I_{gen} > 0A$ which is not possible with IC. Also the start value of the LRC-ramp at t_2 is a hard coded value. It had been determined with help of the previous simulation. Notwithstanding, the simulation shows that the idea works. The final algorithm of this thesis does not use theses simplifications. As a consequence it works less perfect as it is presented here.

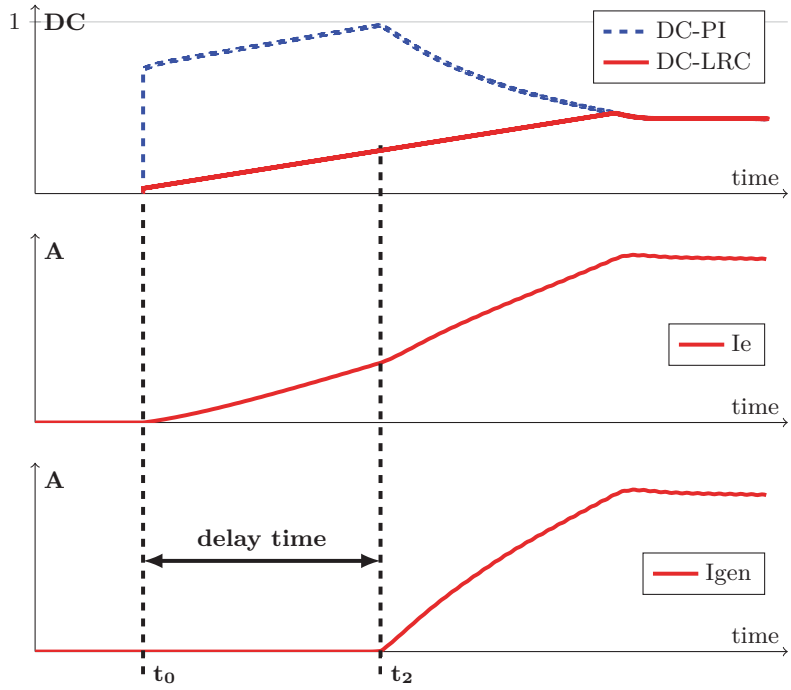


Figure 2.6: simulation LRC enabled

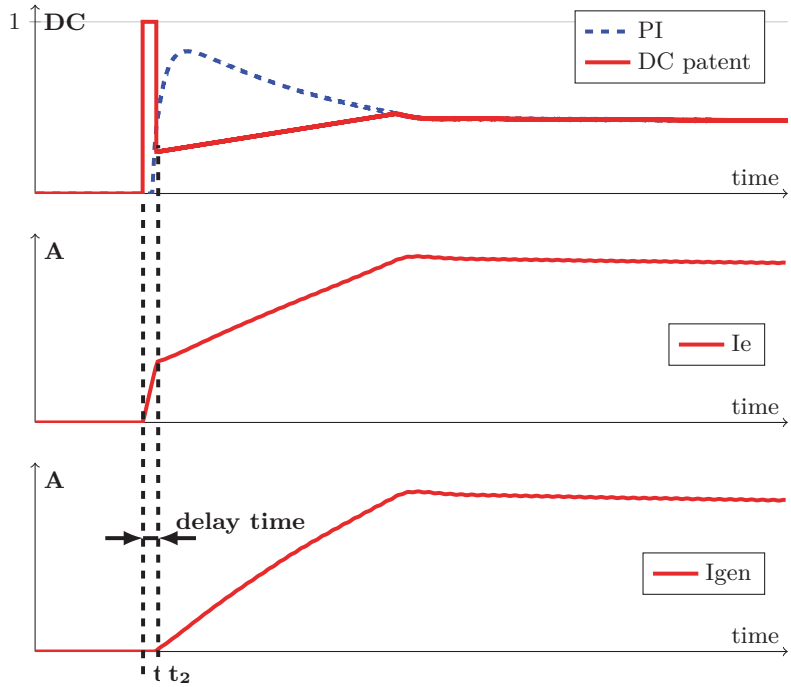


Figure 2.7: principle simulation patent

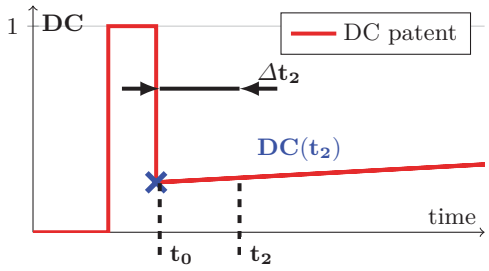


Figure 2.8: principle simulation LRC disabled

Open Questions

Figure 2.8 shows again the $DC - trace$ of the patent idea. The improvement looks quite promising and actually pretty simple. On closer consideration, it turns out that it is more complicated than it looks.

Issues not covered or briefly mentioned by the patent:

1. Time span Δt_2?
   The duration of fast increasing I_e ($DC = 100\%$) is clear, until $I_{gen} > 0$. As explained above, I_{gen} is not measured by the IC, so that criteria can not be used. Even it had been, it might have problems because of the reaction time. Obviously, Δt_2 is the delay time, which depends on rotor time constant. However, it is not known by the IC since each alternator has a different one. Moreover the rotor time "constant" depends on the excitation current because of the iron saturation.
   A hint is given by the patent. It suggests to increase i_e fast as long as the induced stator voltage is smaller then V_{bat} or some value around. This is reasonable, hence the output current will only flow if the induced voltage is greater than the battery voltage. Though, it is not explained how this could be implemented and under which circumstances (speed, load,...).
2. DC-value at t_2?
   This value is very important because it is the transition value from $I_{gen} = 0$ to $I_{gen} > 0A$. After the time span Δt_2 the DC-ramp has to start with this one. If the DC at t_2 is too low, the delay time will be increased. In contrast, if it is too high, I_{gen} will not increase according to the desired LRC gradient at the first moment (and may cause an abrupt torque load). From the simulation with enabled LRC (figure 2.6) the corresponding value is known, but how is it calculated analytically. Moreover, how can this value be determined by the IC? Unfortunately, no advice is given by the patent.
3. Speed influence?
   In the patent the effect of changing the rotor speed is neglected. The lower the speed, the higher the necessary excitation current, subsequently the longer the delay

time. Since the rotor time "constant" is dependent on I_e (saturation) the correlation between speed and the necessary field current is non linear.
Moreover, the rotor speed trace during the engine start sequence is a dynamic trace. As a matter of fact, the induced voltage is directly influenced by acceleration and deceleration.

4. Electrical load?
   As long as the output current is zero, the electrical loads are powered by the battery. Depending on the load current the terminal voltage V_{bat} is higher or lower. As mentioned in item 1 the idea is to increase the excitation current fast till V_{bat} is reached. At the first view it seems like the load has impact to this time span, hence $V_{bat}(I_{load})$. In further consequence the value $DC(t_2)$ would be influenced too.
5. Core losses
   In the patent it is not mentioned that the excitation field causes core losses. These losses consist of two parts: hysteresis losses and eddy current losses. The domains in the stator iron are aligning to the rotating excitation field. Eddy currents are induced because the field is changing for the stator. The needed energy for turning the domains and the energy loss caused by eddy currents must be covered by the engine. In other words, additional torque is produced by the rotating excitation field even if the output current is zero.
6. Alternator variation
   It is not investigated which parameters are important for the application and how to deal with different alternators.
7. Temperature influence?
   Temperature has an effect on the alternator parameters. For example, the forward voltage of the rectifier diodes strongly depend on temperature.
8. Start of regulation?
   There are also no definitions who or which event will trigger the controller start. The engine start is a dynamic speed trace. Thus, the moment of starting the regulation is essential. Notwithstanding, some measured traces of the start process are provided by BMW.
9. Other interaction?
   It is also not noticed that there are other IC fucntions like PSB which could interact with the patent idea.

Most of these topics will be clarified by this thesis and the developed algorithms will deal with some of these issues.

2.3 Demands and requirements

2.3.1 Demands BMW

1. Investigation of patents capabilities by real measurements (how much time can be saved)
2. Figure out possible applications
3. Clarify open questions
4. Analytical investigations
5. Development of an algorithm which meets the patent idea with the following conditions:
   - robust approach hence it should be applicable for several alternators
   - the end user (BMW) needs to adjust as little parameters as possible
   - a few options may be chosen
6. Verification of the algorithm by simulations

2.3.2 Requirements Infineon

1. Algorithm development on the basis of an existing alternator model in Matlab/Simulink
2. Simple concept which considers the IC boundaries:
   - no ADC for the phase voltage U_{ph}
   - no measurement of the output current I_{gen}
   - the resolution of the measured excitation current is low
   - if possible use of already existing components
   - relative low computing power
   - relative small memory (in case of lookup table, constants,...)
3. Design of a voltage controller based on the *Infineon* IC (behaviour model)
4. Simulation of the whole algorithm: implemented concept and manipulation of voltage controller (= enhanced version)
5. Set up a configurable simulation test bench: different speed traces, electrical loads (static, switching), controller selection,....
6. Comparison between closed loop simulations using the IC based controller and using the enhanced version

3 Preliminary Analysis

3.1 Phase Signal and Battery

The comparison of the induced stator voltage with the battery voltage is essential for the application. The grid voltage depends on the battery properties and the load, as long as the induced voltage is too low to charge the battery. Connecting a load will cause an immediate voltage drop because of the internal battery resistor (equation 2.3) and a slow drift depending on capacity. Nevertheless, for a limited time span (up to $30s$), the drift is negligible and the battery voltage is assumed as a constant value (depending on the load). The voltage drop is seen in figure 3.8 and 3.9. The following definitions refer to the electrical circuit in figure 2.2:

$$gnd\text{...negative pole of the battery} \tag{3.1}$$
$$V_{bat}\text{...terminal voltage battery} \tag{3.2}$$
$$V_{BA}\text{...terminal voltage alternator} \tag{3.3}$$
$$u_{12} := u_{A1\rightarrow A2} = \hat{\mathrm{U}}_s \; sin(\omega t) \tag{3.4}$$
$$u_{23} := u_{A2\rightarrow A3} = \hat{\mathrm{U}}_s sin(\omega t + {}^{2\pi}/_{3}) \tag{3.5}$$
$$u_{31} := u_{A3\rightarrow A1} = \hat{\mathrm{U}}_s sin(\omega t - {}^{2\pi}/_{3}) \tag{3.6}$$
$$u_{ph1} := u_{A1\rightarrow gnd} \tag{3.7}$$
$$u_{ph2} := u_{A2\rightarrow gnd} \tag{3.8}$$
$$u_{ph3} := u_{A3\rightarrow gnd} \tag{3.9}$$

The battery will be charged if current flows from the generator into the battery. That implies, that one upper diode and one lower diode are conducting. Since the potential between V_{BA} and gnd is fixed with V_{bat}, the amplitude $\hat{\mathrm{U}}_s$ of the delta voltages (u_{12}, u_{23}, u_{31}) must exceed V_{bat} plus two times the forward voltage of the diodes.

charge condition:

$$\hat{\mathrm{U}}_s > V_{bat} + 2U_f \tag{3.10}$$
$$\hat{\mathrm{U}}_{s_{crit}} := V_{bat} + 2U_f \tag{3.11}$$

Condition 3.10 shows, that the peak value of the induced voltage must be compared with V_{bat}. Equation 3.11 is the definition of the maximum value, which still does not charge the battery (critical value). Clearly it depends on the semiconductor properties. It may happen that the diodes partly conduct, when the induced voltage comes close to the critical

value. However, the resulting current should be minor in this case. With definition 3.11 the following can be expressed:

$$V_{BA} = V_{bat}(I_{load}) \quad \forall \quad \hat{\mathrm{U}}_s < \hat{\mathrm{U}}_{s_{crit}} \tag{3.12}$$

$$V_{bat} = V_{BA} = \bar{U}_s \quad \forall \quad \hat{\mathrm{U}}_s \geq \hat{\mathrm{U}}_{s_{crit}} \tag{3.13}$$

where $\bar{U}_s$ is the rectified stator voltage.
The point is what is measured by the IC. As explained in section 2.1.4, the induced voltage is measured against *gnd*. That means the delta voltage is not available. Moreover, a $100k\Omega$ pull down resistor is connected internally between phase pin and *gnd* (see figure 2.4). The question is how to deduce from u_{ph} the charging condition 3.10 and what is the impact of the pull down resistor. To make it clear, spice simulations were done and compared with real measurements.

3.1.1 LTspice simulation without pull down resistor

Figure 3.1a shows the simulation circuit in LTspice. The battery is simplified as a large capacity and an internal resistor. The diode model is MUR460. AC-voltage sources are used to emulate the generator (without harmonics).

As long as the diodes are not conducting, each half bridge is a symmetric voltage divider with $A_1 = A_2 = A_3 = \frac{V_{bat}}{2}$. This can be seen in figure 3.1c. The peak amplitude of the phase signals can be calculated by equation 3.14.

$$\hat{\mathrm{U}}_{ph1} = \hat{\mathrm{U}}_{ph2} = \hat{\mathrm{U}}_{ph3} = \frac{\hat{\mathrm{U}}_s}{\sqrt{3}} + \frac{V_{bat}}{2} \quad \forall \quad \frac{\hat{\mathrm{U}}_s}{\sqrt{3}} \leq \frac{V_{bat}}{2} \tag{3.14}$$

If condition 3.10 is fulfilled, current is flowing and the following equations are valid (Kirchhoff's voltage low):

$$\hat{\mathrm{U}}_{ph1,1} = -\hat{\mathrm{U}}_{31-} - U_{D6} \qquad \hat{\mathrm{U}}_{ph1,2} = \hat{\mathrm{U}}_{12+} - U_{D4} \tag{3.15}$$

$$\hat{\mathrm{U}}_{ph2,1} = -\hat{\mathrm{U}}_{U23-} - U_{D4} \qquad \hat{\mathrm{U}}_{ph2,2} = -\hat{\mathrm{U}}_{23-} - U_{D2} \tag{3.16}$$

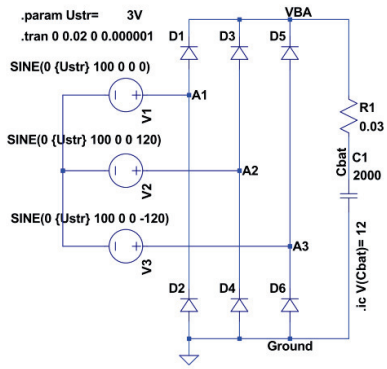
$$\hat{\mathrm{U}}_{ph3,1} = -\hat{\mathrm{U}}_{U12-} - U_{D2}, \qquad \hat{\mathrm{U}}_{ph3,2} = \hat{\mathrm{U}}_{23+} - U_{D6} \tag{3.17}$$

where $\hat{\mathrm{U}}_{ph1,1}$ refers to the first maxima of $u_{ph1}(t)$, $\hat{\mathrm{U}}_{ph1,2}$ the second maxima. U_{12+} maxima and U_{12-} minima of $u_{12}(t)$.
Figure 3.1e shows the simulation of this case. The values of U_{D2}, U_{D4}, U_{D6} are depending on the current-voltage characteristic of the diodes. For the critical point is

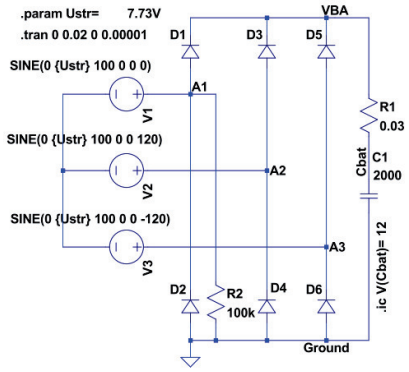
$$U_d = U_f \quad \forall \quad \hat{\mathrm{U}}_s = \hat{\mathrm{U}}_{s_{crit}} \tag{3.18}$$

The critical phase voltage is calculated by inserting $\hat{U}_s = \hat{U}_{s,crit}$ in any equations of 3.15-3.17. With equation 3.11 and 3.18 it follows:

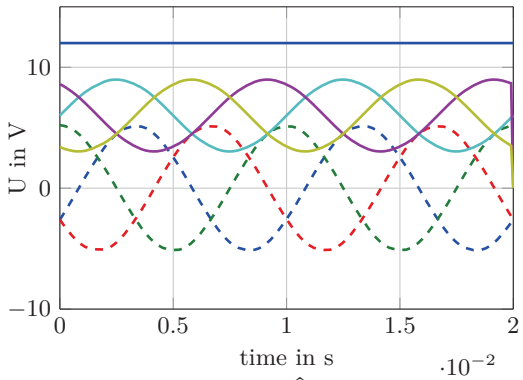
$$\hat{\mathrm{U}}_{ph_{crit}} = \hat{U}_{s,crit} - U_f = V_{bat} + 2U_f - U_f = V_{bat} + U_f \tag{3.19}$$



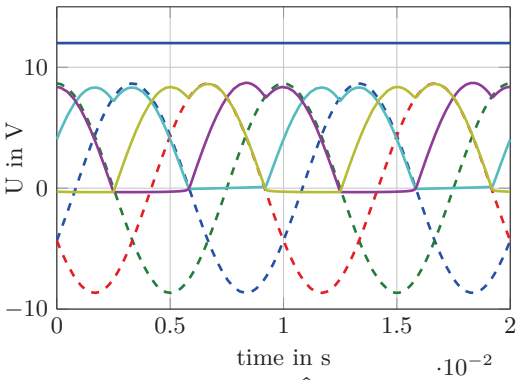
(a)No R_{pd}



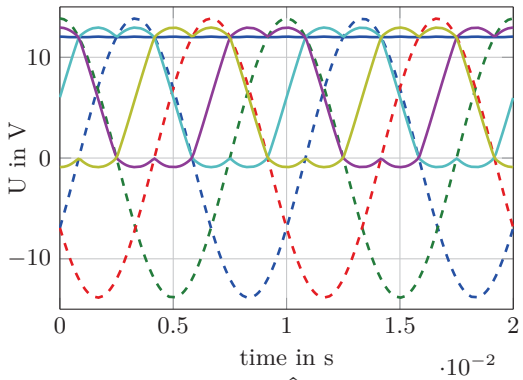
(b)$R_{pd} = 100k\Omega$



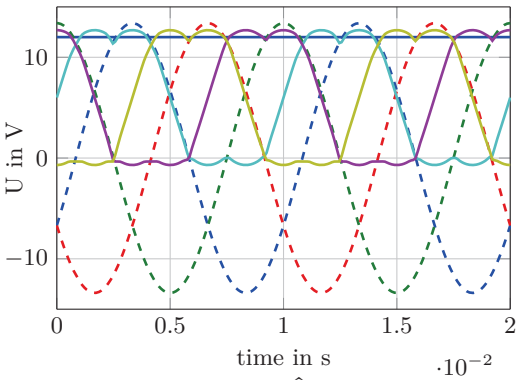
(c)No R_{pd}, $\hat{\mathrm{U}}_s = 5.2V$



(d)$R_{pd} = 100k\Omega$, $\hat{\mathrm{U}}_s = 8.66V$



(e)No R_{pd}, $\hat{\mathrm{U}}_s = 13.8V$



(f)$R_{pd} = 100k\Omega, \hat{\mathrm{U}}_s = 13.4V$

Figure 3.1: simulation LTspice, initial battery voltage $V_{bat_0} = 12V$,
u_{12} - - -, u_{23} - - -, u_{31} - - -, u_{ph1} ——, u_{ph2} ——, u_{ph3} ——, V_{BA} ——

3.1.2 Alternator measurement without pull down resistor

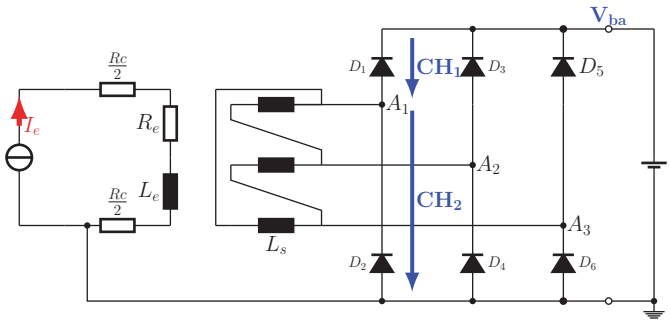


Figure 3.2: electric circuit without $R_{pd} = 100k\Omega$

A brush-holder without IC is mounted on the alternator. The rotor is driven with a constant speed and $0.5A$ (direct current) are impressed to the excitation winding. The battery is connected to the terminals of the rectifier but there is no resistor load. To each diode of the half bridge a differential voltage probe (Tektronix TDP0500) is connected in parallel. The impedance of each probe is $1M\Omega$. The traces are recorded with an oscilloscope (Tektronix MSO4000)
Figure 3.3 shows a result of the measurements. $CH1 = V_{BA} - U_{ph1}$, $CH2 = U_{ph1}$, $CH1 + CH2 = V_{BA}$. Due to "nonsinusoidal flux density distribution" (see [3] on page 679) harmonics are caused. Apart from that the measurement and the simulation show similar behaviour. In both cases, the phase signal has an offset of $V_{bat}/2$.

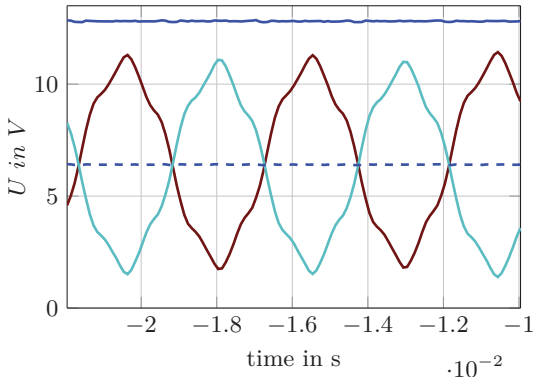


Figure 3.3: alternator measurement $I_e = 0.5A$, initial battery voltage $V_{bat_0} \approx 12.5V$, no R_{pd}
— $CH1 : V_{BA} \rightarrow A_1$, — $CH2 : A_1 \rightarrow gnd$, — $CH1 + CH2$, - - - $\frac{1}{2}(CH1 + CH2)$

3.1.3 LTspice simulation with pull down resistor

The IC uses a pull-down resistor R_{pd} between A_1 and gnd (parallel to Diode D_2). This unbalances the voltage divider. In figure 3.1d a simulation is shown with this pull-down. As

long as the amplitude of the delta voltage does not fulfil the charging condition 3.10, A_1 is pulled down. The delta voltage u_{31} is split to R_{pd} and D_6. During the negative half-wave, D_6 gets conducting, hence A_1 is pulled down to ground and A_3 is negative. The current through D_6 is limited by the R_{pd} and the diode characteristics. In this particular case, the voltage drop of D_6 it is about $0.32V$. The same is valid for D_4. Consequently, the peak value of $U_{ph1,1}$ and $U_{ph1,2}$ is $\hat{U}_s - 0.32V$. On the other hand, $\hat{U}_{ph2,2}$ and $\hat{U}_{ph3,1}$ are $\hat{U}_s$, hence $U_{D2} = 0V$ (equations 3.20 - 3.23).

$$U_{D2} = 0V \tag{3.20}$$

$$U_D = U_{D6} = U_{D4} \tag{3.21}$$

$$\hat{U}_{ph1,1} = \hat{U}_{ph1,2} = \hat{U}_{ph2,1} = \hat{U}_{ph3,2} = \hat{U}_s - U_D \tag{3.22}$$

$$\hat{U}_{ph2,2} = \hat{U}_{ph3,1} = \hat{U}_s \tag{3.23}$$

Note, equations 3.20 - 3.23 are valid if $\hat{U}_s < \hat{U}_{s_{crit}}$. In case of $\hat{U}_s \geq \hat{U}_{s_{crit}}$ the diodes are conducting and the situation is the same as in section 3.1.1. Equations 3.15 - 3.17 are valid. This can be seen in figure 3.1f. With 3.22 and 3.18 the critical value of the phase voltage referred to gnd is:

$$\hat{U}_{ph1,1_{crit}} = \hat{U}_{s_{crit}} - U_f \tag{3.24}$$

Equation 3.11 leads to:

$$\hat{U}_{ph1,1_{crit}} = V_{bat} + U_F \tag{3.25}$$

3.1.4 Alternator measurement with pull down Resistor

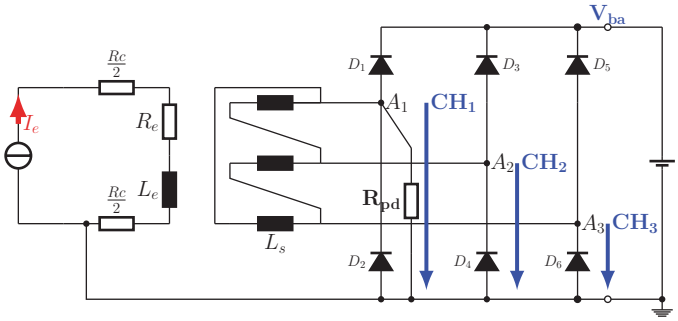


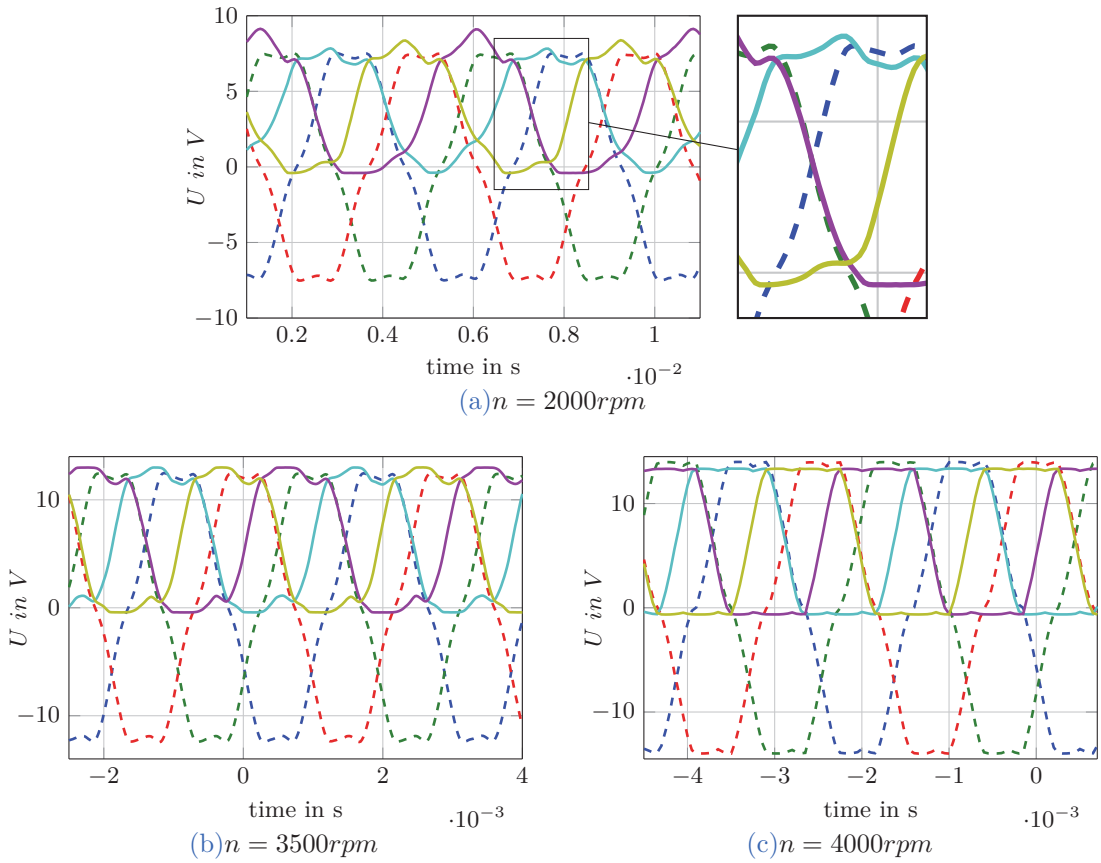
Figure 3.4: electric circuit with $R_{pd} = 100k\Omega$

Same set-up as in section 3.1.2 but a $100k\Omega$ resistor is connected between A_1 and gnd. Three passive voltages probes (Tektronix TPP0500) are used to measure $A_1 -> gnd = u_{ph1}$, $A_2 -> gnd = u_{ph2}$ and $A_3 -> gnd = u_{ph3}$. Each has an impedance of $10M\Omega$. The trace of the delta voltages is calculated by:

$$u_{12} = u_{ph1} - u_{ph2} \tag{3.26}$$

$$u_{23} = u_{ph2} - u_{ph3} \tag{3.27}$$

$$u_{31} = u_{ph3} - u_{ph1} \tag{3.28}$$



(a)$n = 2000rpm$ (b)$n = 3500rpm$ (c)$n = 4000rpm$

Figure 3.5: alternator measurement $I_e = 0.5A$, initial battery voltage $V_{bat_0} \approx 12.5V$, $R_{pd} = 100k\Omega$ u_{12} - - -, u_{23} - - -, u_{31} - - -, u_{ph1} ——, u_{ph2} ——, u_{ph3} ——, V_{BA} ——

Figure 3.5a shows a measurement, where the amplitude of the delta voltage is lower than the battery voltage. Contrary to the simulation in figure 3.1d, u_{ph_2} and u_{ph_3} are positive during the maxima of u_{ph1}. Somehow there is a positive voltage shift of $\approx +0.3V$. Therefore, the maxima of u_{ph1} referred to gnd is $\hat{U}_{ph1} \approx 7.8V$ while the amplitude of the delta voltage is lower $\hat{U}_{12} \approx 7.5V$. In case of the maxima of u_{ph2} and u_{ph3} the shift is even greater, although the amplitude of delta voltage stays the same.
Another measurement is done where the amplitude of the delta voltage is more or less the battery voltage. The maximum of $u_{ph_1}(t)$ referred to gnd is $\hat{U}_{ph1} \approx 13V$ while $\hat{U}_{12} \approx 12.5V$. This can be seen in figure 3.5b.

Nevertheless, current flow to battery will start when both the upper and the lower diode of the half-bridges, are conducting. Therefore, charging condition 3.10 is still valid. This can be seen in figure 3.5c. The peak value of the phase signal referred to gnd is $\hat{U}_{ph1} \approx 13.3V$, while the amplitude of delta voltage is $U_{12} \approx 14V$. The measurement shows the same behaviour like the simulation in figure 3.1f.
The causes of the positive ground shift, if $U_s \leq U_{s,crit}$, are not clear. Different LTspice

simulations with harmonics and parasitic capacitors to the diodes were done, but they did not show this behaviour.

3.1.5 Conclusion

The battery will be not charged until condition 3.10 is fulfilled. Subsequently the critical value is given by equation 3.11: $U_{s,crit} = V_{bat} + 2U_f$. The critical value referred to gnd is given by Kirchhoff's voltage low: $U_{ph,crit} = U_{s,crit} - U_f = V_{bat} + U_f$, see equation 3.19 and 3.25. This is always true independent of any pull down resistor.
If a pull down resistor is connected as shown as in 3.1b or 3.4, the half-bridge is unbalanced. In that case the phase voltage is given by:

$$\hat{U}_{ph} = \hat{U}_s + U_{shift} \tag{3.29}$$

The LTspice simulation has shown as expected:

$$U_{shift} = -U_D \tag{3.30}$$

and

$$0 < U_D < U_f \quad \forall \quad U_f \leq \hat{U}_s < \hat{U}_{s,crit} \tag{3.31}$$

$$U_D = U_f \quad \forall \quad \hat{U}_s = \hat{U}_{s,crit} \tag{3.32}$$

$\hat{U}_{ph}$ is always lower than $\hat{U}_s$, since U_{shift} is in both cases a negative value. The charging condition is examined by comparing the recent value $\hat{U}_{ph}(t)$ with the critical value $\hat{U}_{ph,crit}$:

$$\hat{U}_{ph}(t) \overset{!}{\geq} V_{bat} + U_f \tag{3.33}$$

Since, the critical value is used in condition 3.33 it refers to $U_{shift} = -U_f$, thus it is assumed:

$$\hat{U}_{s,estimated} = \hat{U}_{ph}(t) + U_f \tag{3.34}$$

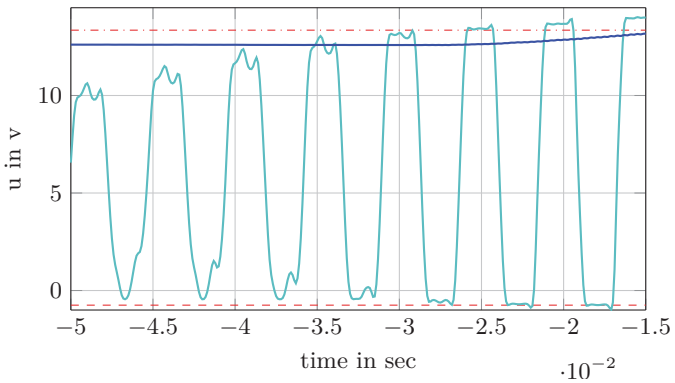


Figure 3.6: zoom start-up measurement $n = 2100rpm$, $LRC = 0s$, $V_{set} = 14V$, u_{ph1} ——, V_{BA} ——, $V_{bat_0} \approx 12.6V$, $U_f = 0.75V$ (estimated), $-U_f$(estimated) - - -, $V_{bat_0} + U_f$ -·-·-

However, the real value of U_s is given by rearranging equation 3.29:

$$\hat{U}_{s,real} = \hat{U}_{ph}(t) - U_{shift}(t) \tag{3.35}$$

With equation 3.31 and 3.32 it can be seen that $\hat{U}_{s,estimated}$ is $\hat{U}_{s,real}$ only if the charging condition 3.10 is fulfilled, otherwise $\hat{U}_{s,estimated}$ is greater than $\hat{U}_{s,real}$. This is described by the following:

$$\hat{U}_{s,estimated} = \hat{U}_{s,real} \quad \forall \quad \hat{U}_{s,real} \geq \hat{U}_{s,crit} \tag{3.36}$$

$$\hat{U}_{s,crit} > \hat{U}_{s,estimated} > \hat{U}_{s,real} \quad \forall \quad \hat{U}_{s,real} < \hat{U}_{s,crit} \tag{3.37}$$

For example, the simulated case in figure gives, $U_{shift} = -U_d \approx -0.32V$ and $\hat{U}_{ph} \approx 8.33V$. This leads to $U_{s,real} = 8.33 + 0.32V = 8.65V$. From the data sheet of the diodes ($MUR460$) it is known that $U_f = 0.65V (@I_f = 0.1A)$. Using that value gives, $\hat{U}_{s,crit} = 12V + 2 \cdot 0.65V = 13.3V$ and $\hat{U}_{ph,crit} = 12.65V$. The calculated value by equation 3.34 gives $U_{s,estimated} = 8.33V + 0.65V = 8.98V$. Although $\hat{U}_s$ is estimated too large, it is correctly reconsidered that the charging condition is not fulfilled $U_{s,estimated} < U_{s,crit}$. On the contrary, the alternator measurements have shown that there is an positive shift if $\hat{U}_{s,real} < \hat{U}_{s,crit}$, see figure 3.5a and 3.5b. Thus, the peak of the phase voltage referred to gnd is greater than $\hat{U}_{s,real}$. As consequence, the estimation of $\hat{U}_s$ by equation 3.34 is even worse. U_f is approximately $0.75V$ see figure 3.6. In case of $n = 3500rpm$, $\hat{U}_{s,estimated}$ is given by $13.75V$ while $\hat{U}_{s,real} = 12.5V$. Since $\hat{U}_{s,crit} = 14V$ it is correctly reconsidered that the charging condition is not fulfilled. The measurement with $n = 4000rpm$ gives $\hat{U}_{s,estimated} = 14.05V$. It is correctly reconsidered that the charging condition is fulfilled.

Notwithstanding in terms of robustness, implementation, parameter drift and other effects, the target value of the algorithm during the first part (fast increase) will be considered to be $\hat{U}_{ph}(t) \stackrel{!}{=} \hat{U}_{ph,target} = V_{bat} + \Delta U$, where ΔU is adjustable by the end-user. If ΔU is chosen $0.5V$, and the battery voltage is given by $V_{bat} = 12.5V$, then the target value is considered as $\hat{U}_{ph,target} = 13V$ while the critical value for the phase voltage is still $\hat{U}_{ph,crit} = 13.25V$. The difference is only $0.25V$. Reviewing the measurement $n = 3500rpm$ shows that $U_{ph}(t) = U_{ph,target}$. Consequently, the first part would be done. The fast increase of i_e will be stopped. As before $\hat{U}_{s,estimated} = 13.75V$, which is also $0.25V$ lower than $\hat{U}_{s,crit}$. The point is that $U_{s,real} = 12.5V$, which gives a real difference to the critical value of $1.5V$. In other words, the desired distance to critical value is $0.25V$. The algorithm assumes this is fulfilled hence $\hat{U}_{s,estimated} = 13.75V$, but in reality is the distance $1.5V$. The fast increase of i_e is stopped (way) too early. The further increase will be done according to the LRC gradient. The outcome is, it will take more than expected till the battery will be charged. Notwithstanding, this is still better than increasing the current slowly from the beginning on. Besides, measurements with increasing excitation current and constant speed have shown: the closer U_{ph} is to $U_{ph,crit}$, the better $\hat{U}_{s,estimated}$ matches with $\hat{U}_{s,real}$, see figure 3.6.

3.2 Start Up Measurements

In order to determine the potential of the patent idea, measurements with different LRC settings were conducted.

General setup

- Valeo FG18 alternator, mounted brush-holder including the IC
- the rotor is driven at $2100rpm$ (typical idle speed)
- mostly with connected load ($I_{load} \approx 35A$)
- the electrical circuit matches with figure 2.2
- $V_{set} = 14V$ (via LIN), $LRC - BLZ = 3\%$, $LRC - RT$ varies (via LIN)
- Tektronix MSO4000 is used to record the trace
- U_{ph}, V_{ba}, and U_{exc} are measured with passive voltage probes (TPP0500)
- I_{gen} is measured using a current transducer (ratio $^{10mV}/_{A}$) and a differential voltage probe (TDP0500)

Figure 3.7 shows in detail the results of the measurement with $LRC - RT = 1s$. All measured quantities are (digitally) low pass filtered in the post process because of noise. The cut-off frequency is $2kHz$. Since the fundamental frequency of the induced voltage is about $200Hz$, the third and the fifth harmonic are not affected by the filter. The time stamps are defined as:

- t_0 start point of the voltage control (execution of the LIN start command)
- t_1 first intersection U_{ph} and V_{bat}, $\Delta t_1 = t_1 - t_0$
- t_2 first time $I_{gen} \geq 1A$, battery gets charged $\Delta t_2 = t_2 - t_0$
- t_3 first intersection V_{ba} and V_{set}, the target value is reached $\Delta t_3 = t_3 - t_0$

The DC in the figures is calculated from rising edge to rising edge of the measured excitation PWM. At the beginning, PSB is dominating, because the DC of the LRC-ramp is too low for a proper phase signal. This can be seen by the two outliers in the calculated DC-trace and by the two long pulses in the measured PWM trace.
Furthermore, as long as U_{ph} is smaller than the battery voltage V_{ba} (timespan Δt_1) or rather $V_{bat} + U_f$ (referred to as gnd) no current is produced by the alternator, then I_{gen} is increasing according to the LRC-gradient. That matches with the simulation in section 2.2: increasing the excitation current slowly until t_2 is reached is not necessary in respect of load response.
The same principle is seen in figure 3.8 but with $LRC - RT = 10s$. Hence, the rate of the LRC-ramp is lower as in figure 3.7, the PSB interaction is lasting longer. Nevertheless, PSB does not reduce the time span Δt_1 or rather Δt_2, because the LRC gradient is independent of PSB. In other words, it takes the same time to reach the $DC(t_2)$-value. In terms of clarity, the envelope of $U_{ph}(t)$ is plotted and the PWM-trace is omitted in figure 3.8. The offset at the very beginning of the DC-trace is the $LRC - BLZ$ value.

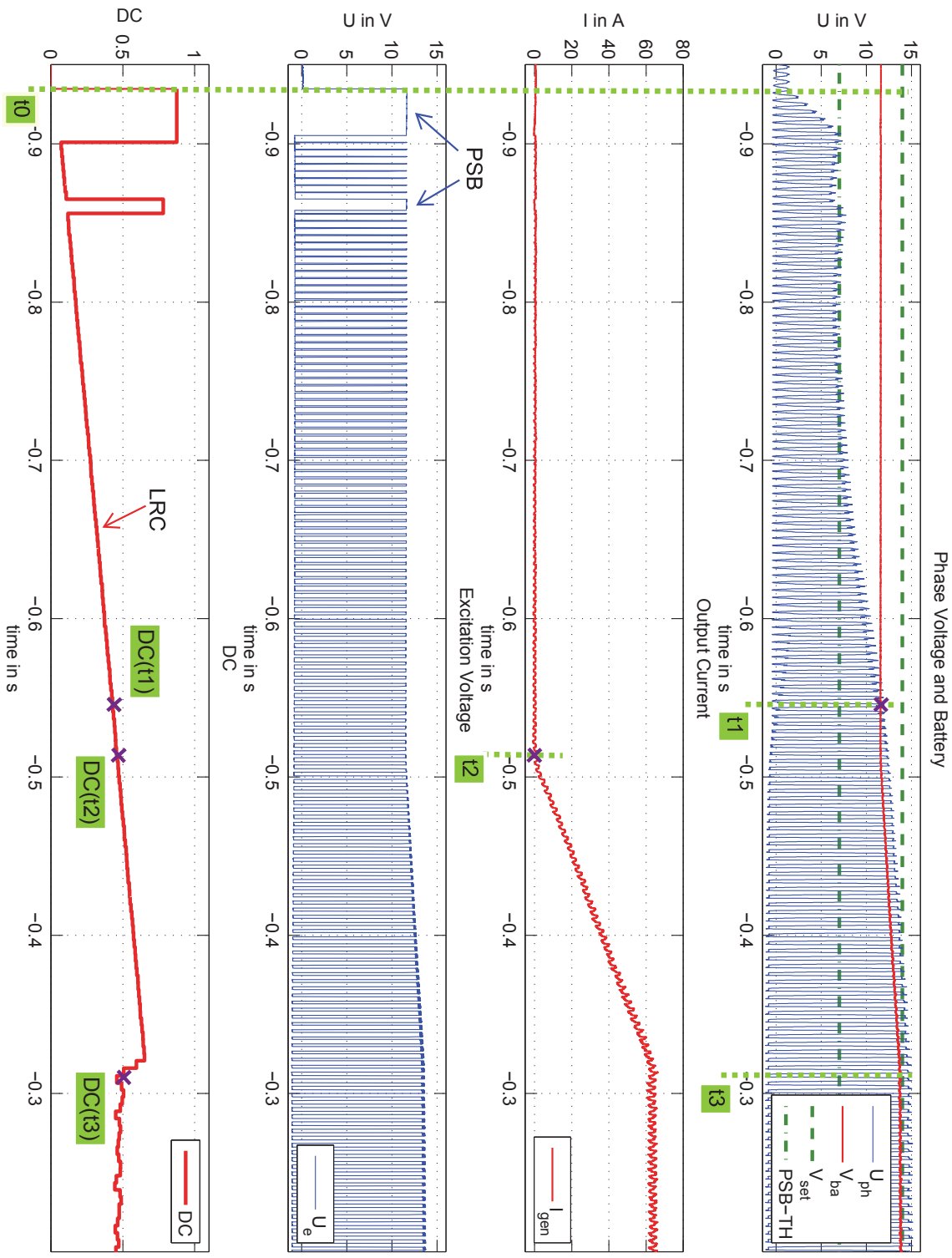


Figure 3.7: Startup $n = 2100rpm,\ LRC = 1s,\ V_{set} = 14V, I_{load} \approx 35A$

On the contrary, figure 3.9 shows a measurement without LRC. As in the simulation with disabled LRC, the voltage controller uses the full actuating variable till V_{ba} reaches V_{set}. As a result the time span Δt_3 is short, but the generator output current I_{gen} increases rapidly. Note: figure 3.7 and 3.9 have different time axes.

Further measurements were conducted. Table 3.1 presents results for different $LRC - RT$ values. In contrast, table 3.2 shows the results of measurements with same $LRC - RT$ value but different loads. The column I_{gen} refers to the steady-state value of the generator output current (cursor value in figure 3.8).

$LRC - RT$ s	Δt_1 s	Δt_2 s	Δt_3 s	$DC(t_1)$ %	$DC(t_2)$ %	$DC(t_3)$ %	$V_{ba}(t_0)$
0	0.06	0.07	0.18	99.8	99.8	46.3	11.47
1	0.38	0.42	0.86	41.11	45.56	43.23	11.60
5	1.28	1.45	2.70	29.65	32.89	42.29	11.43
10	2.36	2.64	4.72	27.88	30.22	42.80	11.46
15	3.31	3.75	6.67	25.78	29.02	43.42	11.46

Table 3.1: Overview $n = 2100rpm,\ I_{gen} \approx 35A$ (steady-state)

I_{gen} A	Δt_1 s	Δt_2 s	Δt_3 s	$DC(t_1)$ %	$DC(t_2)$ %	$DC(t_3)$ %	$V_{ba}(t_0)$ V
0	2.33	2.61	3.0	27.27	30.22	30.31	12.64
35	2.36	2.64	4.72	27.88	30.22	42.80	11.46
75	2.27	2.62	6.33	26.67	30.22	60.71	10.68

Table 3.2: Overview $n = 2100rpm,\ LRC - RT = 10s$

The time span Δt_1 or rather Δt_2 could be minimized by the patent idea. As a consequence Δt_3 would also be shortened. In case of slow LRC-ramps ($LRC - RT > 5s$) there is a potential of seconds that could be saved, see table 3.1.
As mentioned in section 3.1, connecting a load to the battery causes an immediate voltage drop. That is clear to see in figure 3.8 and 3.9 at the beginning. In both cases, the load is connected one second before the voltage controller starts. The voltage drop depends on the load current. If the $LRC - RT$ is not changed Δt_1, Δt_2 are almost the same values, although different loads were connected to the battery, see table 3.2.
The situation is clear: if V_{bat} is higher, the excitation current increases faster (if the same DC-value is applied). For this reason, different initial values $V_{ba}(t_0)$ result in the same time span. From that it can be concluded, that the time, which could be saved, is independent of the load. It is important to point out that this is true only for constant loads.

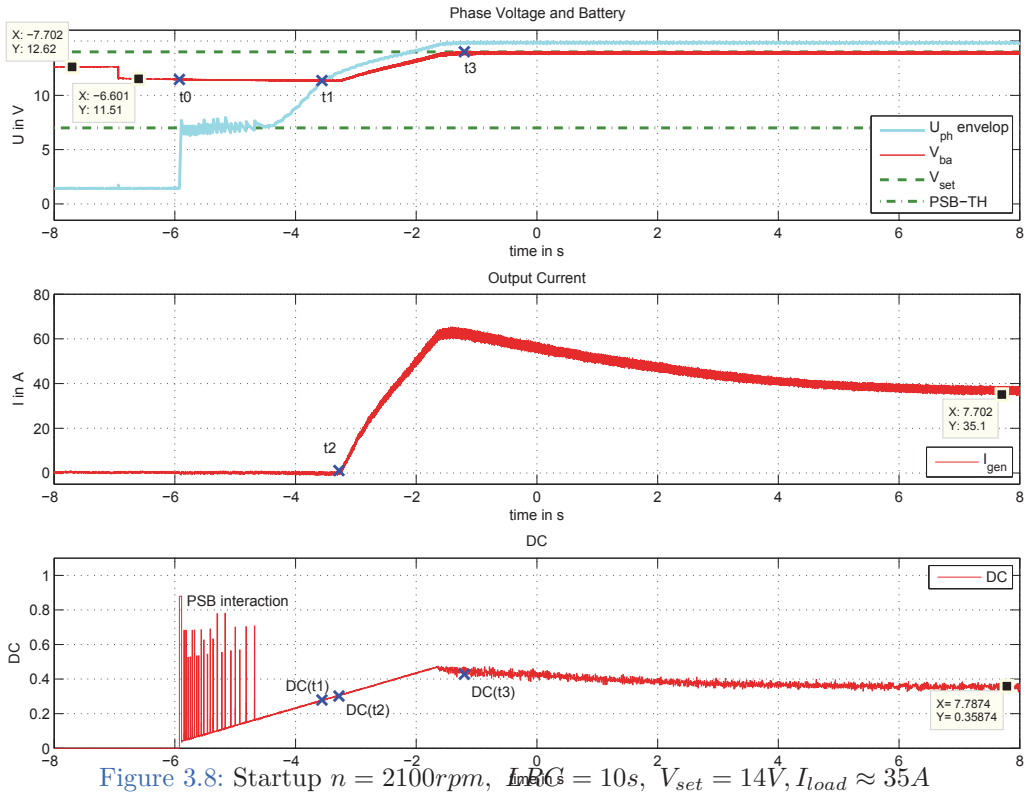


Figure 3.8: Startup $n = 2100rpm,\ t_{BC} = 10s,\ V_{set} = 14V, I_{load} \approx 35A$

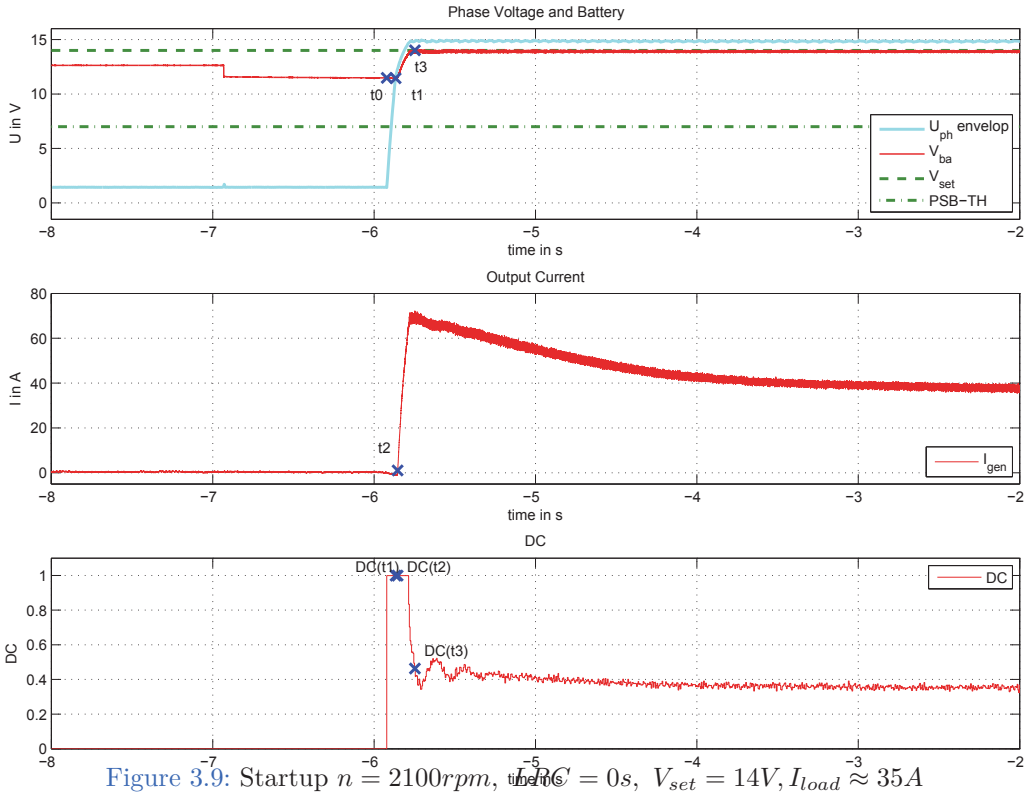


Figure 3.9: Startup $n = 2100rpm,\ t_{BC} = 0s,\ V_{set} = 14V, I_{load} \approx 35A$

3.3 Load throw off

Turning off electrical components can cause an overshoot of V_{ba}, hence the load is changed abruptly. As a consequence, the excitation voltage is switched off by the IC as long as $V_{ba} > V_{set}$. Consequently the phase voltage U_{ph} goes down. A possible application for the patent idea could be to keep the phase voltage close to the battery instead of switching off the excitation voltage completely. Measurements with the following set-up were conducted:

- mounted brush-holder including Infineon alternator IC
- the rotor is driven at a constant speed
- $V_{set} = 14.5V$ (via LIN)
- $LRC - BLZ = 3\%$
- $LRC - RT = 0s$
- sudden load throw off I_{load} $90A \rightarrow 10A,\ n = 1700rpm$ (steady-state values)
- the traces of U_{ph},V_{bat},I_{gen} and U_{exc} are measured

Figure 3.10 shows a typical situation of a battery overshoot caused by the load change. The time stamps are defined as:

- t_0 load throw off (overshoot V_{ba})
- t_1 first intersection $U_{ph} < V_{bat}$, $\Delta t_1 = t_1 - t_0$
- t_2 IC starts voltage control (after overshoot), $\Delta t_2 = t_2 - t_0$

It is obvious, that a high excitation current is needed if the alternator is heavily loaded, especially in case of low speed. Since the excitation voltage can not change the polarity ($U_e = -V_{bat}$), the current decreases slowly if the excitation is switched off $U_e = 0V$. Thus, the overshoot lasts relatively long and U_{ph} stays above V_{ba}. The idea to keep the phase voltage close to V_{ba} could be done for the time span $t_2 - t_1$ which is relatively short. With increasing alternator speed, the time span increases but it is still low. An overview is shown in figure 3.11. For this reason, the case load throw off is not considered as a possible application.
It is important to point out that, different $LRC - RT$ would not change the time span $t_2 - t_1$, because the point in time when the voltage controller starts regulating again depends only on the trace of V_{ba}. However, the patent idea could be used to boost the restart, it is a similar case as in section 3.2.

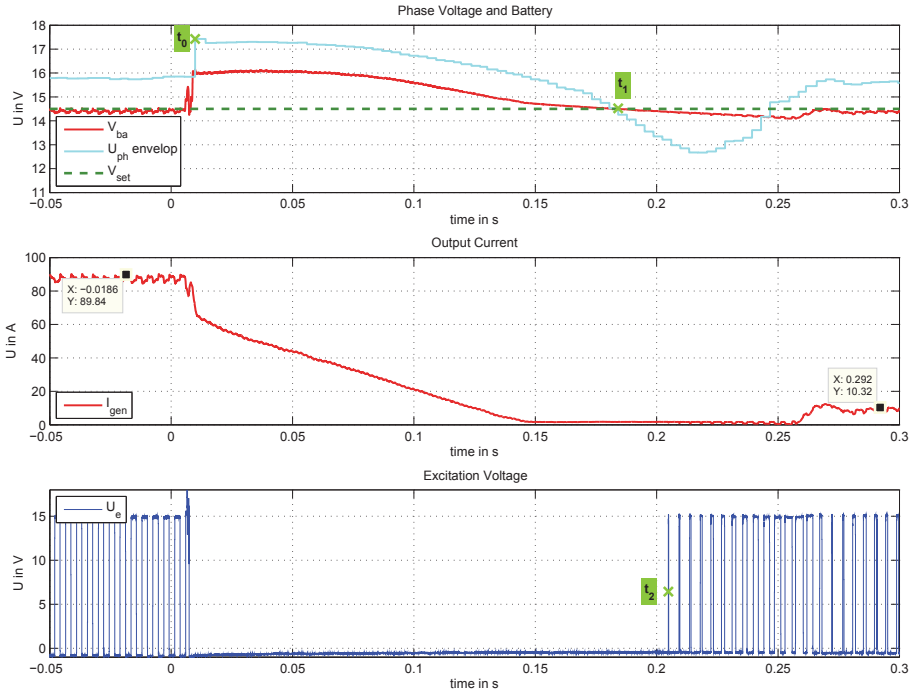


Figure 3.10: load throw off I_{load} $90A \rightarrow 10A,\ n = 1700rpm,\ Vset = 14V$

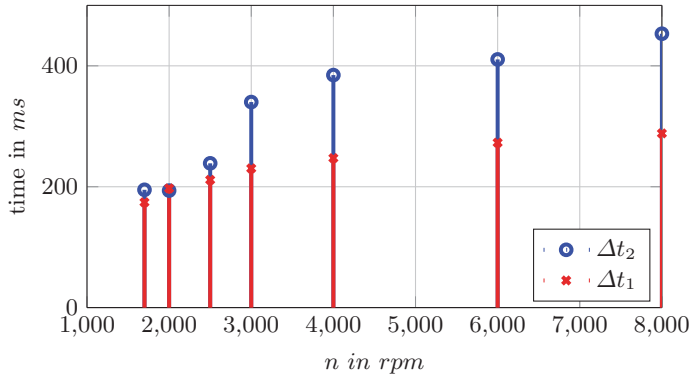


Figure 3.11: The difference between Δt_2 and Δt_1 is minimal.

3.4 Switching load

A switching load (on/off/on,..) can also cause a battery overshoot. If LRC is activated, it could take a relatively long time till the battery gets loaded again. The reason is the slow DC-ramp when the controller restarts regulating. Measurements with the following settings were conducted:

- mounted brush-holder including Infineon alternator IC
- the rotor is driven at a constant speed ($LRC_{OFF} = 4000rpm$)
- $V_{set} = 14.5V$ (via LIN)
- $LRC - BLZ = 3\%$
- $LRC - RT = 15s$
- $LRC - FT = 1s$
- pulsed load I_{load} $115A \rightarrow 50A$ (steady-state values)
- pulse frequency f_{load} varies
- the traces of U_{ph},V_{bat},I_{gen} and U_{exc} are measured

Figure 3.12 shows a measurement with a rejection frequency $f_{load} = 5Hz$. As explained before, the excitation current decreases slowly. For that reason, U_{ph} stays above during the first phase with a low load. Since the LRC fall timer allows restart regulation with a relatively large DC, U_{ph} is always above V_{ba}. Obviously there is no use case. Also other measurements have shown a similar behaviour.

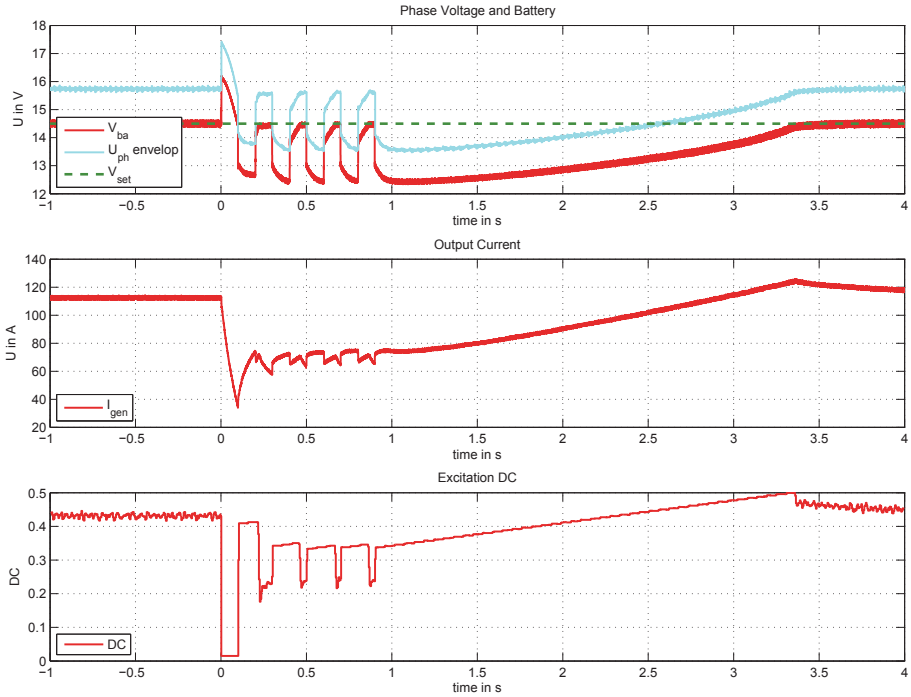


Figure 3.12: Pulsed load $f_{load} = 5Hz,\ I_{load}:\ 115A \rightarrow 50A,\ n = 3000rpm$

3.5 Conclusion

Measurements have shown that an appropriate application for the patent idea is the start-up situation if $LRC - RT > 5s$. In case of a single load throw off the phase voltage U_{ph} does not break down much, before the controller starts regulating again. However the patent could be used to boost the restart process (a similar situation to the start-up situation). Switching loads are handled by the LRC-fall timer. For that reason, this thesis focuses only on the start-up situation.

4 Development

4.1 Concept Phase Control

Basic consideration

An output current $I_{gen} > 0A$ causes load torque. As long as the rectifier diodes are not conducting, the output current is zero. With equation 3.25 the following can be expressed:

$$I_{gen} \approx 0A \quad \forall \quad \hat{U}_{ph} < V_{bat} + U_f \tag{4.1}$$

In case of rotation, there is always a load torque because of mechanical friction τ_f. If the excitation winding is current-carrying ($i_e \neq 0A$) as well, core losses are produced in stator. These cause a further load torque (see open questions in section 2.2). The stator slots and the rotor poles produce a torque ripple (cogging torque), but this is zero-mean. The engine has to overcome all, even if $I_{gen} = 0A$. Nevertheless, they are neglected in this thesis.
The original patent idea is to bring $\hat{U}_{ph}$ close to the critical value (equation 3.25) as fast as possible. Priority is to assure $I_{gen} \approx 0A$ before the LRC-gradient is applied. For that reason, the target value is considered as:

$$\hat{U}_{ph,target} = V_{BA} \pm \Delta U. \tag{4.2}$$

V_{BA}= V_{bat} and ΔU is an adjustable margin. Details follow in section 4.4.1.

Objectives:

The application has to deal with three tasks:

1. increase field current as fast as possible until $\hat{U}_{ph} \leq \hat{U}_{ph,target}$
2. determine the DC which keeps $\hat{U}_{ph} \approx \hat{U}_{ph,target}$
3. use the determined DC as an offset value for the LRC-ramp

The determined DC in step 2 is from now on called handover value DC_{HO}.

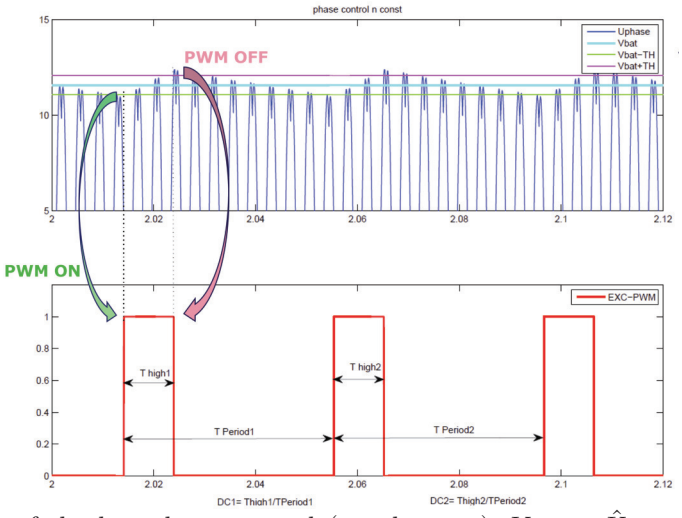


Figure 4.1: Concept of the bang-bang control (steady-state), $V_{ref} = \hat{U}_{ph,target} = V_{BA}$, $TH = TH_1 = TH_2$

Approach:

A phase controller is needed, because the aim of step 1 and 2 is $\hat{U}_{ph} \stackrel{!}{=} \hat{U}_{ph,target}$. Since a simple implementation is wanted, the idea is to use a bang-bang controller with the following principle:

$$V_{ref} = \hat{U}_{ph,target} \tag{4.3}$$

$$PWM = 1 \ \ \forall \ \ \hat{U}_{ph} < (V_{ref} - TH_1) \tag{4.4}$$

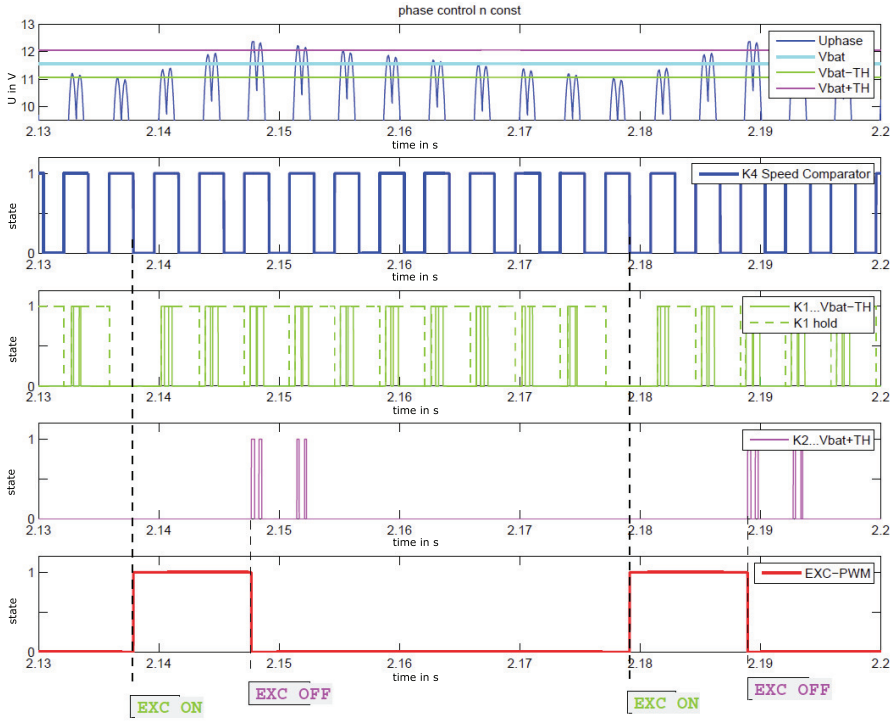
$$PWM = 0 \ \ \forall \ \ \hat{U}_{ph} \geq (V_{ref} + TH_2) \tag{4.5}$$

Figure 4.1 shows a simulation of the controller and the alternator for a constant rotor speed. With the values TH_1 and TH_2 a hysteresis is adjustable. In the steady-state, the DC of the PWM is the handover value DC_{HO}:

$$DC_{HO} = \frac{T_{high}}{T_{Period}}\bigg|_{steady-steate} \tag{4.6}$$

As mentioned before, no ADC is available, but with one additional comparator this bang bang controller is feasible to implement. Also for the PSB function comparators are used. However, there is an essential difference. For the PSB $\hat{U}_{ph}$ is compared with a fixed reference voltage. Now the reference value of the bang-bang controller is the analogue value of V_{BA} or rather $V_{BA} \pm \Delta U$. Furthermore, the hysteresis should be adjustable by at least a few discrete values. Nevertheless, the estimated effort by the Infineon designers to implement such a comparator in the IC is relatively low. The determination of the DC can be done with a resettable counter.

For this thesis, the bang-bang comparator is considered to be two comparators without hysteresis, but adjustable thresholds (simplification). One is used for condition 4.4 and the other one for 4.5. Notwithstanding, with one or two comparators, condition 4.4 is always fulfilled during the zero crossing of U_{ph}. However, the determination of the speed is based on detecting zero crossings. This information can be used to activate or deactivate condition 4.4.
Figure 4.2 shows in detail the simulation and the logic equations of the bang bang controller. The PWM is set ON at the falling edge of the speed comparator (K_4) if the phase voltage has not exceeded the lower threshold (K_1). It is set OFF immediately if the phase voltage exceeds the upper threshold (K_2). The Simulink implementation is given in the appendix, see figure .3.



component	output	condition
speed-comparator:	$K_4 = 0$ $K_4 = 1$	$\hat{U}_{ph} < V_{min}$ $\hat{U}_{ph} \geq V_{min}$
on-comparator:	$K_1 = 0$ $K_1 = 1$	$\hat{U}_{ph} < (V_{ref} - TH_1)$ $\hat{U}_{ph} \geq (V_{ref} - TH_1)$
off-comparator:	$K_2 = 0$ $K_2 = 1$	$\hat{U}_{ph} < (V_{ref} + TH_2)$ $\hat{U}_{ph} \geq (V_{ref} + TH_2)$
RS-flipflop K_1:	$K_{1,hold} = 0$ $K_{1,hold} = 1$	$\urcorner\llcorner K_4$ (reset) $K_1 = 1$ (set)
RS-flipflop K_2: (redundant)	$K_{2,hold} = 0$ $K_{2,hold} = 1$	$\lrcorner\ulcorner K_4$ (reset) $K_2 = 1$ (set)
RS-flipflop PWM_{bb}:	$PWM_{bb} = 0$ $PWM_{bb} = 1$	$K_2 = 1$ (reset) $\urcorner\llcorner K_4 \wedge K_{1,hold} = 0 \wedge K_{2,hold} = 0$ (set)

Figure 4.2: Bang bang control detail

The bang-bang controller works with the assumption that $\hat{U}_{ph}$ is greater than a dedicated minimum value V_{min}. In reality there is always a remanence field. By rotation it should be enough to detect the induced voltage, even if the field current is zero. To assure a proper phase signal PSB gets active.
However, for the simulation, the alternator model is initialized without any remanence. Therefore, an extra start-boost mechanism is needed. The idea is to increase the excitation

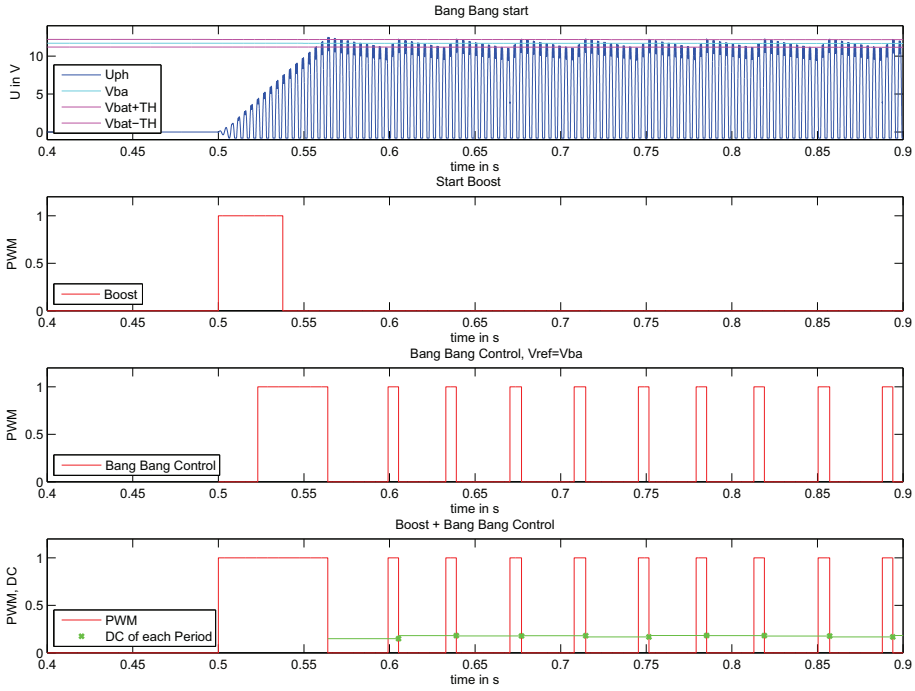


Figure 4.3: Start boost and bang-bang control, $V_{ref} = V_{ba}$

current with $DC = 100\%$ till the phase voltage reaches a dedicated threshold V_{PSB} (similar to the PSB-function without pulse-off time). The system will be triggered by a start signal (assumption from ECU). With the following condition, there is an overlapping of start-boost and the bang bang controller:

$$V_{min} < V_{PSB} < V_{ref} \tag{4.7}$$

The simulation is shown in figure 4.3. The bottom plot shows the overlapping PWM of the start boost and the bang bang controller. The long pulse at the beginning matches with the first part of the patent idea (step 1). The following periods keep $\hat{U}_{ph} \approx V_{ref}$ (step 2). The frequency of the PWM is not fixed, it depends on the following parameters:

- rotor speed
- alternator parameters
- Bang Bang hysteresis TH and V_{ref}

The lower the speed, the more field current is needed. The more current, the lower the gradient of the flux $\Psi_e(i_e)$. For that reason, the PWM frequency is lower at a low speed. The smaller the hysteresis, the greater the frequency. First simulations have shown, that the frequency of the bang-bang PWM is relatively low in general (between $30 - 50Hz$). On the contrary, the output of the V_{ba}-controller is a fixed frequency PWM ($220Hz$). To sum up, the DC handover value is determined by a system with a relatively low frequency,

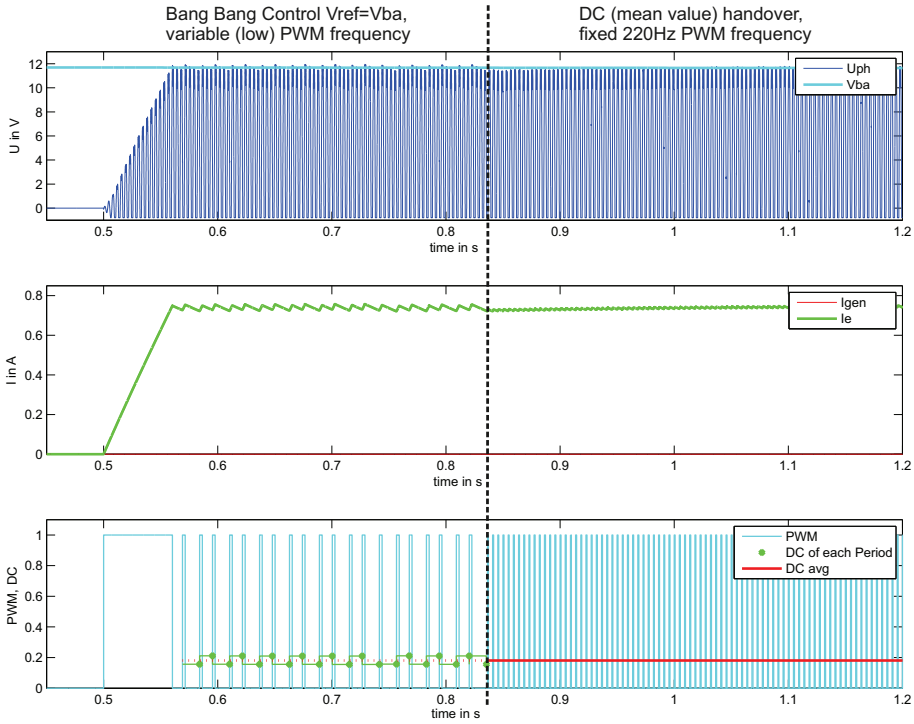


Figure 4.4: Start boost and bang bang control, $V_{ref} = V_{ba}$

but it will be handed to a system with a higher frequency.
Figure 4.4 shows a simulation with the handover. Since the DC is a relative value, the PWM frequency does not matter. The point is, that the average value of the excitation current remains the same. At least this is true in the simulation, hence the alternator model does not consider eddy current or other frequency dependencies. Also measurements with a real alternator have shown, that the average current stays the same. In section 5.2.1, measurements with PWM frequency-sweeps (fixed DC) are presented.

In figure 4.3 and 4.4 it can be seen that there is jitter of DC. The reason will be focused in section 4.5. In order to provide, a reasonable DC value some averaging is needed. The crucial point is, how many PWM periods are used for the determination. The more periods, the more time is lost. Obviously, there is a conflict with the goal: minimization of the delay time. Section 4.5 deals with a trade-off between handover accuracy and number of periods. The result is that, about five periods should be enough in most of the cases.

The DC trace of the concept is presented in figure 4.5. Contrary to the original patent, an intermediate step has been added in order to determine DC_{HO}. The draft is overdrawn, the intermediate step takes about the half time of the long pulse. The reference value of the phase controller is $V_{ba} + \Delta U$. With the variation of ΔU, the end user can choose the distance to the critical value. It is assumed, that a start signal from the ECU triggers the

start-up. There are two possibilities for the handover. It can be chosen whether it will be done as fast as possible or at any time later. As fast as possible, means when a dedicated number of PWM periods N_{min} is reached. The other option is, that the handover can be delayed by the ECU to an advantageous moment in respect of torque loading. This concept is from now on called Enhanced Charge Control (ECC). A schematic overview of the ECC is given in figure 4.6.

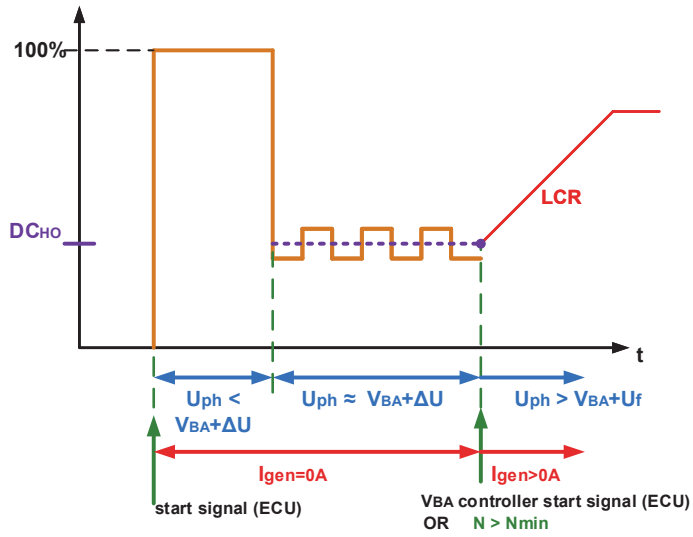
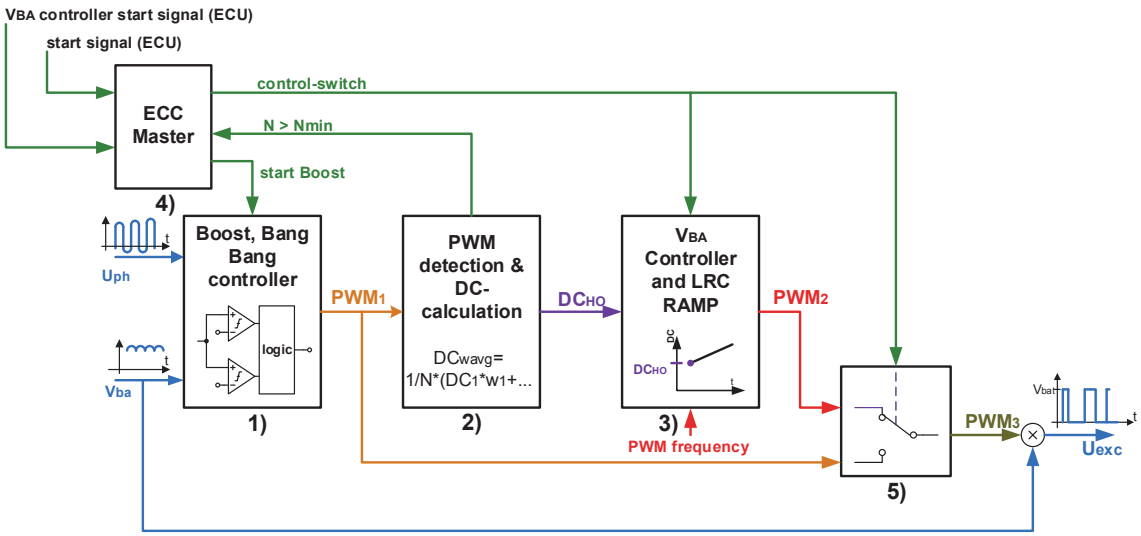


Figure 4.5: *DC*-trace ECC



PWM_1: PWM of the start boost and bang bang controller, variable frequency
PWM_2: PWM of the V_{ba} controller and LRC, fixed frequency
PWM_3: $[PWM_1\ PWM_2]$
N: current number of PWM_1 periods
N_{min}: minimum Number of PWM_1 periods

Figure 4.6: Block digram ECC

Block 1) in figure 4.6 is the phase controller, Block 2) determines the DC of the PWM_1 and calculates the handover value. The V_{ba}-controller inclusive LRC can be seen in Block 3). The actions of these are controlled by Block 4), the ECC-Master. The idea of the master is that options like the handover criteria and other parameters can be adjusted (by programming the NVM or via LIN). It also controls the PWM-switch in Block 5). PWM_3 is fed to the output stage. Table 4.1 presents a comparison of the principle implementation in Simulink and a possible IC. The final Simulink implementation is given in the appendix 7.

To sum up, the following parameters need to be adjusted by the end user:

- the safety margin ΔU
- the relative thresholds TH_1 and TH_2
- the minimum number of averaging periods N_{min}

In order to figure out which values are reasonable, this thesis covers:

- analytical/numerical calculation of the ideal DC
- analytical/numerical calculation of TH_1 and TH_2
- clarification of the DC-jitter and development of a filter algorithm
- investigations into rotor speed influence static and dynamic
- investigations into load influence

	Simulink	IC
Phase controller: (time continuous)	relational operations, logic operations	comparators, logic elements
DC-determination: (time discrete)	MATLAB function block (triggered)	counter, VHDL
V_{BA} Controller/LRC: (time discrete)	MATLAB function block (triggered)	VHDL
ECC Master (time discrete)	MATLAB function block (triggered)	VHDL

Table 4.1: comparison implementation

4.2 Development Environment and Simplifications

All considerations and simulations in this chapter are based on a model of the Valeo A14 alternator ($150A$). This model was developed by the Graz University of Technology Electrical Drives and Machines Institute. It has been in use by Infineon since 2002. This section gives a brief overview. Informations about the parameter identification and modelling can be found in [17] and [18].

4.2.1 Claw-pole Generator

Table 4.2 shows the used model parameter. The column property refers to the model properties.

parameter	property
excitation winding resistor R_e	constant
excitation main inductance $L_{he}(i_\mu)$	saturable
exictation leakage inductance $L_{\sigma e}$	constant
winding ratio a	constant
mutual inductance $M(i_\mu)$	saturable
stator main inductance $L_{hs}(i_\mu)$	saturable
stator leakage inductance $L_{\sigma s}$	constant

Basic facts for the generator model:

- based on the fundamental frequency
- dynamic model of a non-salient pole synchronous machine
- magnetic saturation considered
- iron losses not considered (eddy currents)
- temperature not considered
- implemented in matlab/simulink

Table 4.2: Model parameters Valeo A14

The stator is assumed to be a three phase system (abc-system). The Clarke transformation [6] gives the space phasor equation in the stator reference frame ($\alpha\beta$-system):

$$\underline{u}_s^s = R_s \underline{i}_s^s + \dot{\underline{\Psi}}_s^s \tag{4.8}$$

The excitation winding on the rotor is a one-phase winding. Since the d-axis of the rotor reference frame is aligned with the winding, the q components are zero. Thus, $u_e = u_{ed} = \underline{u}_e^r$ is given by:

$$u_e = R_e i_e + \dot{\Psi}_e \tag{4.9}$$

The stator voltage phasor is transformed into the rotor reference by multiplication of equation 4.8 with $e^{-j\varphi}$ ($\varphi = p\omega_{mech}t$). The result is:

$$\underline{u}_s^r = \underline{i}_s^r + j\dot{\varphi}\dot{\underline{\Psi}}_s^r + j\dot{\varphi}\underline{\Psi}_s^r \tag{4.10}$$

Equation 4.10 and 4.9 rearranged and written in components gives:

$$\dot{\Psi}_{sd} = -R_s i_{sd} + \dot{\varphi}\Psi_{sq} + u_{sd} \tag{4.11}$$

$$\dot{\Psi}_{sq} = -R_s i_{sq} - \dot{\varphi}\Psi_{sd} + u_{sq} \tag{4.12}$$

$$\dot{\Psi}_e = -R_e i_e + u_e \tag{4.13}$$

The flux linkages are expressed by:

$$\Psi_{sd} = L_{\sigma s} i_{sd} + L_h i_{sd} + M i_e \tag{4.14}$$

$$\Psi_{sq} = L_{\sigma s} i_{sq} + L_h i_{sq} \tag{4.15}$$

$$\Psi_e = L_{\sigma e} i_e + L_{he} i_e + M \frac{3}{2} i_{sd} \tag{4.16}$$

The voltages u_e and $u_{sd},\ u_{sq}$ are the input variables of the model. The fluxes Ψ_e and $\Psi_{sd},\ \Psi_{sq}$ are used as state variables. By rearranging equation 4.14-4.16, the output variables i_e and $i_{sd},\ i_{sq}$ are determined. As mentioned in table 4.2, the mutual inductance depends on the magnetization. In the simulation a 2D look up table is used to determine the value of M depending on the magnetization state:

$$M = M(\Psi_\Sigma, \Psi_{sq}) \qquad \Psi_\Sigma = \Psi_{sd} + \Psi_e \frac{L_{\sigma s}}{L_{\sigma e} a} \tag{4.17}$$

The turn ratio a is used to calculate L_{he} and L_h:

$$L_h = M a \qquad L_{he} = \frac{M}{\frac{2}{3} a} \tag{4.18}$$

An overview of the Simulink model is given in the appendix 7.

4.2.2 Rectifier

It is assumed that a three-phase bridge rectifier with passive diodes is coupled to the generator. As explained in [18] page 4, the model of the rectifier consists of three identical half-bridges with a filter capacitor Cs on the DC-side. "The diodes are implemented with linear characteristic curves. Auxiliary capacitors C_p and damping resistors R_p are used for calculation of the diode potential." Figure 4.7 shows a half-bridge of the model. If the voltage difference at the diode is lower than U_f, it is not conducting. Equations 4.19

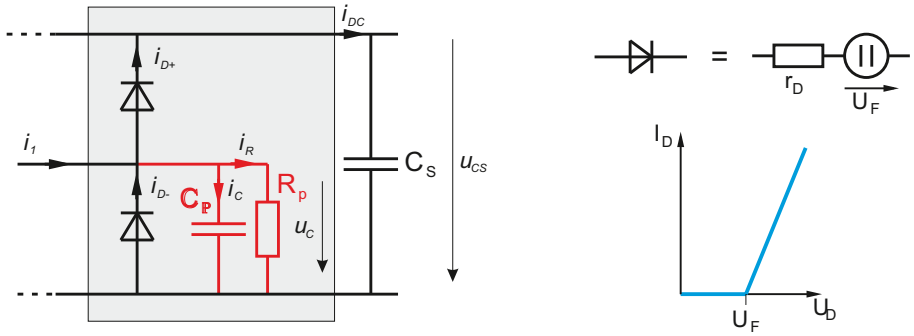


Figure 4.7: Implementation of the half bridge; R_p and C_p are used to determine the diode potential. Source: [18] page 4

and 4.20 describe the upper diode:

$$i_{D+help} = -\frac{u_{cs} + U_F - u_c}{r_D} \tag{4.19}$$

$$i_{D+} = \begin{cases} i_{D+help}, & \text{if } i_{D+help} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

The lower diode is described by equation 4.21 and 4.22:

$$i_{D-help} = -\frac{u_c + U_F}{r_D} \tag{4.21}$$

$$i_{D-} = \begin{cases} i_{D-help}, & \text{if } i_{D+help} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.22}$$

A simulation of the generator and connected rectifier is shown in figure 4.8a. On the contrary to section 3.1.3, each half bridge is unbalanced by $R_p = 1k\Omega$. Thus, the lower diodes ($D_2,\ D_4,\ D_6$) are alternately conducting (according the negative half-wave of the delta voltage). The voltage drop of each is U_f even if $U_s \leq U_{s,crit}$. In other words, the dump resistors cause a shift of $A_1 = A_2 = A3 = -U_f$ in any case. For that reason, the peak value of phase voltage referred to ground is given by:

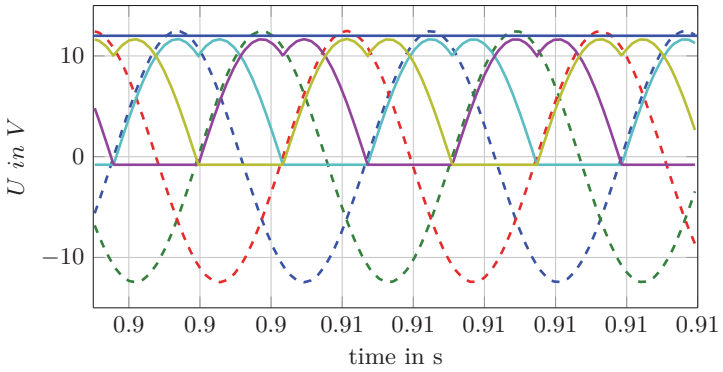
$$\hat{U}_{ph} = \hat{U}_{ph1,1} = \hat{U}_{ph1,2} = \hat{U}_{ph2,1} = \hat{U}_{ph2,2} = \hat{U}_{ph3,1} = \hat{U}_{ph3,2} \tag{4.23}$$

$$\hat{U}_{ph} = \hat{U}_s - U_f \quad \forall \quad \hat{U}_s > U_f \tag{4.24}$$

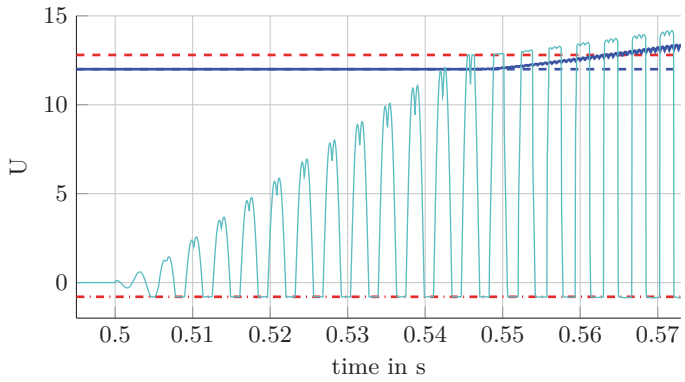
For the simulation, the charge condition is always given by:

$$\hat{U}_{ph,crit} = \hat{U}_{s,crit} - U_f = V_{ba}(t_0) + U_f \tag{4.25}$$

where t_0 is the start point of regulation. Figure 4.8b shows a simulation with constant speed and increasing excitation current. The rising of the phase voltage and the transition not charging/charging the battery can be seen.



(a)$I_e = 0.7A,\ n = 2100rpm,\ u_{12}$ - - -, u_{23} - - -, u_{31} - - -, u_{ph1} ——, u_{ph2} ——, u_{ph3} ——, V_{BA} ——



(b)i_e increasing, $n = 2100rpm$, —— V_{BA}, - - - $V_{ba}(t_0)$, - - - $V_{ba}(t_0)$+U_f, -·-·- $-U_F$, —— u_{ph}

Figure 4.8: Simulation generator and rectifier, $V_{bat}(t_0) = 12V$

4.2.3 Battery and electrical load

The battery model is simplified to a large capacitor and an internal resistor (see [19]). The electrical load is considered as ohmic resistance. Figure 4.9 shows the electrical circuit. The calculation of the circuit is given by equations 4.26-4.29.

$$V_{bat} = V_{BA} = U_{CS} \tag{4.26}$$

$$\frac{dU_i}{dt} = \frac{1}{C_{bat}} I_B = \frac{1}{C_{bat}} \left(\frac{V_{bat} - U_i}{R_i} \right) \tag{4.27}$$

$$I_L = \frac{V_{BA}}{R_{load}} \tag{4.28}$$

$$\frac{dU_{CS}}{dt} = \frac{1}{C_S} I_{CS} = \frac{1}{C_S} \left(I_{GEN} - I_L - I_B \right) \tag{4.29}$$

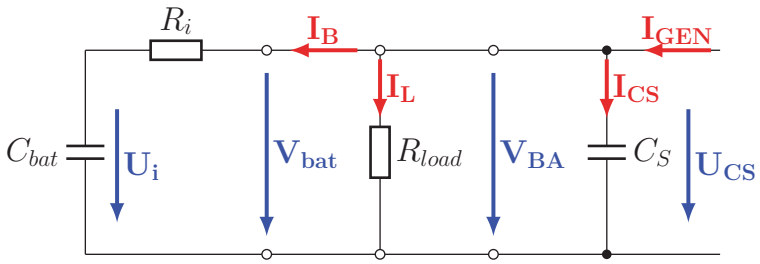


Figure 4.9: Electrical circuit, battery, load resistor and filter capacitor

4.2.4 Simplifications

- the combustion engine is not simulated,
  - the rotor speed trace is an input variable, given by the simulation end user
  - there is no feedback
- the rotor speed measurement is neglected, the speed information is taken directly from the input variable
- V_{BA} AD conversion is simplified
- IC issues like bit precision are neglected
- temperature and other influences are not simulated
- generator simplifications see basic facts in table 4.2

4.3 Alternator Theory

Reasonable criteria for the values of the ECC parameters are required. In order to determine them, a closer look at the alternator behaviour is necessary. Equation 4.30 shows the link between rotor speed n in rpm and the electrical angular frequency ω.

$$\dot{\varphi} = \omega = 2\pi f_{el} = 2\pi p \frac{n}{60} \tag{4.30}$$

4.3.1 Torque

So far it had been claimed that the alternator torque is proportional to the output current. The reason is the determination of the air gap torque. It is given by (see [16] equation 5.52 on page 312):

$$\tau_{rotor} = -\tau_{stator} = -\frac{3}{2} p \Im\{\underline{\Psi}_s^s \; \underline{i}_s^{*s}\} \tag{4.31}$$

Note the output current I_{gen} is the rectified stator current $\underline{i}_s$.

4.3.2 Flux calculations

The aim of this section is to figure out (simple) equations to describe the alternator behaviour for the ECC. It is assumed, that the ECC works as defined. Thus, the output current is zero, which means idle run:

$$\underline{i}_s^s = 0 \tag{4.32}$$

Assumption 4.32 simplifies equation 4.8 to:

$$\underline{u}_s^s = \dot{\underline{\Psi}}_s^s \tag{4.33}$$

Equation 4.33 transformed into the rotor reference frame gives:

$$\underline{u}_s^r = \dot{\underline{\Psi}}_s^r + j\dot{\varphi}\underline{\Psi}_s^r \tag{4.34}$$

The stator flux linkage can be written as:

$$\underline{\Psi}_s^r = (L_{\sigma s} + L_h)\underline{i}_s^r + Mi_e \tag{4.35}$$

$$\underline{\Psi}_s^r = \Psi_{sd} + j\Psi_{sq} \tag{4.36}$$

Note $i_e = i_{ed} = \underline{i}_e^r$. Since $\underline{i}_s = 0$, equation 4.35 is simplified to

$$\underline{\Psi}_s^r = Mi_e \tag{4.37}$$

Obviously, the q-component is zero:

$$\underline{\Psi}_s^r = \Psi_{sd} = Mi_e \tag{4.38}$$

$$\Psi_{sq} = 0 \tag{4.39}$$

Since $\Psi_{sq} = 0$ the derivative of the stator flux is:

$$\dot{\underline{\Psi}}_s^r = \dot{\Psi}_{sd} \tag{4.40}$$

Equations 4.38 and 4.40 in equation 4.34 lead to:

$$\underline{u}_s^r = \dot{\Psi}_{sd} + j\dot{\varphi}\Psi_{sd} \tag{4.41}$$

component by component:

$$\underline{u}_s^r = u_{sd} + ju_{sq} \tag{4.42}$$

absolute value:

$$|\underline{u}_s^r|^2 = u_{sd}^2 + u_{sq}^2 = \dot{\Psi}_{sd}^2 + (\dot{\varphi}\Psi_{sd})^2 \tag{4.43}$$

The aim of the phase controller is to maintain the induced voltage.

Requirement: keep $|\underline{u}_s^r|^2$ constant

$$|\underline{u}_s^r|^2 \stackrel{!}{=} C \tag{4.44}$$

Equation 4.44 leads to a differential equation for Ψ_{sd}:

$$C = \dot{\Psi}_{sd}^2 + (\dot{\varphi}\Psi_{sd})^2 \tag{4.45}$$

$$\dot{\Psi}_{sd} = \sqrt{C - (\dot{\varphi}\Psi_{sd})^2} \tag{4.46}$$

Problem: It cannot be ruled out that $C = (\dot{\varphi}\Psi_{sd})^2$. In case it does $\dot{\Psi}_{sd} = 0$. An increase of the speed $\dot{\varphi}$ causes a negative Term in the root (equation 4.46). The result is a complex solution, which is infeasible. For that reason, requirement 4.44 cannot be fulfilled in general. In other words, any change of flux leads to an increase of voltage $\underline{u}_s$ for the first moment.

New approch Maintain $u_{sq} = \dot{\varphi}\Psi_{sd}$ constant instead of $|\underline{u}_s^r|$ and keep in mind $|\underline{u}_s^r|$ is the geometrical of sum u_{sq} and u_{sd}.

$$\dot{\varphi}\Psi_{sd} = U_{sq} \tag{4.47}$$

$$U_{sq} = const \tag{4.48}$$

The derivative of equation 4.47 gives a differential equation for Ψ_{sd}:

$$\frac{d}{dt}(\dot{\varphi}\Psi_{sd}) = 0 \tag{4.49}$$

$$\ddot{\varphi}\Psi_{sd} + \dot{\varphi}\dot{\Psi}_{sd} = 0 \tag{4.50}$$

Differential equation 4.50 describes the dynamic and equation 4.47 the static behaviour:

$$\dot{\Psi}_{sd} = -\frac{\ddot{\varphi}}{\dot{\varphi}}\Psi_{sd} \tag{4.51}$$

$$\Psi_{sd} = \frac{U_{sq}}{\dot{\varphi}} \tag{4.52}$$

Equation 4.47 in equation 4.43 gives:

$$|\underline{u}_s^r| = \sqrt{\dot{\Psi}_{sd}^2 + U_{sq}^2} \tag{4.53}$$

For a given value of U_{sq}, Ψ_{sd} and $\dot{\Psi}_{sd}$ can be calculated (equations 4.51, 4.52) . With $\dot{\Psi}_{sd}$ and equation 4.53 $|\underline{u}_s|$ is determined. The aim is to find the maximum value $U_{sq_{max}}$ which still assure the following condition:

$$|\underline{u}_s^r| \leq U_{s_{max}} \tag{4.54}$$

With the definition of the critical value in section 4.3.2, the upper limit of $U_{s_{max}}$ is given by:

$$U_{s_{max}} \leq \hat{\mathrm{U}}_{s_{crit}} \tag{4.55}$$

It is possible to determine $U_{sq_{max}}$ using an iterative process ($\dot{\varphi}, \ddot{\varphi}$ must be known):

1. choose a limit value for $U_{s_{max}}$, but take 4.55 into account
2. start with $U_{sq_{max}} = U_{s_{max}}$
3. calculate Ψ_{sd}, $\dot{\Psi}_{sd}$ and $|\underline{u}_s|$ using $U_{sq} = U_{sq_{max}}$
4. check if condition 4.56 is fulfilled.
5. if not, decrease $U_{sq_{max}}$ till 4.56 is fulfilled

$$\sqrt{U_{sq_{max}}^2 + \dot{\Psi}_{sd}^2} \overset{!}{\leq} U_{s_{max}} \tag{4.56}$$

Another possibility is to combine equations 4.51 and 4.52 to:

$$\dot{\Psi}_{sd} = -\frac{\ddot{\varphi}}{\dot{\varphi}^2} U_{sq} \tag{4.57}$$

Putting 4.57 in equation 4.53 gives:

$$|\underline{u}_s^r| = \sqrt{U_{sq}^2 + \left(-\frac{\ddot{\varphi}}{\dot{\varphi}^2} U_{sq}\right)^2} \tag{4.58}$$

Factoring out U_{sq}^2 leads to:

$$|\underline{u}_s^r| = U_{sq}\sqrt{1 + \left(\frac{\ddot{\varphi}}{\dot{\varphi}^2}\right)^2} \tag{4.59}$$

if the minimum alternator speed is considerd to be $n = 800rpm$ and the maximum acceleration to be $\dot{n} = 6000^{rpm}/_{s}$ it can be shown that the root term in equation 4.59 approximately equals one. Using equation 4.30 and a pole pair number $p = 8$ gives:

$$\left(\frac{\ddot{\varphi}}{\dot{\varphi}^2}\right)^2 = \left(\frac{\ddot{n}}{\dot{n}^2 \frac{2\pi}{60} 8}\right)^2 \tag{4.60}$$

$$\sqrt{1 + \left(\frac{6000}{800^2 \frac{2\pi}{60} 8}\right)^2} \approx 1.000031 \tag{4.61}$$

The consequence is:

$$|\underline{u}_s^r| \approx U_{sq} \tag{4.62}$$

Conclusion: If $\dot{\varphi}\Psi_{sd}$ is kept constant, the influence of the d component is negligible. The stator voltage $\underline{u}_s^r$ is mainly affected by its q-component $u_{sq} = \dot{\varphi}\Psi_{sd} = U_{sq}$.
To ensure condition 4.54, a safety factor can be introduced:

$$f = \sqrt{1 + \left(\frac{\dot{n}_{max}}{n_{min}^2 \frac{2\pi}{60} p}\right)^2} \tag{4.63}$$

$$U_{sq_{max}} = \frac{U s_{max}}{f} \tag{4.64}$$

$$U_{sq_{max}} \leq |\underline{u}_s^r| \leq U_{s_{max}} \ \forall n \geq n_{min} \mid |\dot{n}| \leq \dot{n}_{max}, u_{sq} = U_{sq_{max}} \tag{4.65}$$

Consequently, the equation for the stator voltage 4.41 can be reduced to:

$$\underline{u}_s^r = j\dot{\varphi}\Psi_s^r \tag{4.66}$$

The stator voltage is simplified to one component:

$$|\underline{u}_s^r| = \dot{\varphi}\Psi_s^r \tag{4.67}$$

using 4.37 leads to:

$$|\underline{u}_s^r| = \dot{\varphi}Mi_e \tag{4.68}$$

Note, these equations are correct as long as the stator current is zero.

The new approach shows that $|\underline{u}_s^r| \approx U_{sq}$, assuming that the product $\dot{\varphi}\Psi_{sd}$ is constant. It is clear that Ψ_{sd} must decrease if $\dot{\varphi}$ increases and vice versa. The differential equation for $\dot{\Psi}_{sd}$ is given by 4.57. To sum up:

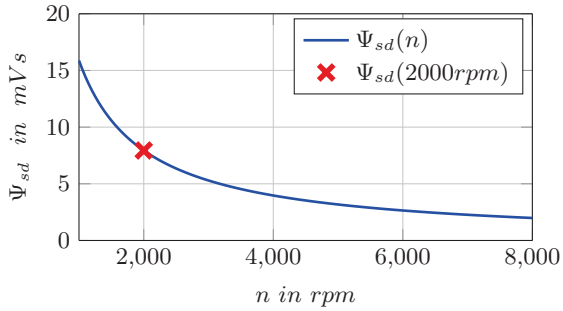
$$|\underline{u}_s^r| \approx U_{sq} \quad if \quad u_{sq} = const \tag{4.69}$$

$$u_{sq} = const \quad if \quad \dot{\Psi}_{sd} = -\frac{\ddot{\varphi}}{\dot{\varphi}^2}U_{sq} \tag{4.70}$$

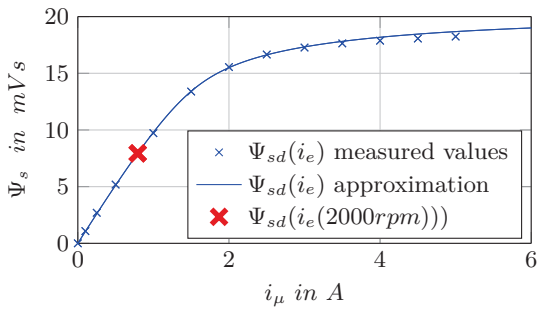
In the application, it is the other way around. The comparators control the stator voltage $|\underline{u}_s^r|$. By comparing the peak value of U_{ph} with a reference value ($V_{bat} + \Delta U$). Equation 4.46 has shown that it is complicated to find a differential equation for $\dot{\Psi}_{sd}$ to keep $|\underline{u}_s^r|$ constant. In case of a bang-bang controller this is impossible anyway.
With the reduced requirement of maintaining $\hat{U}_{ph}$ between a upper and a lower threshold, the negligence of $\dot{\Psi}_{sd}$ in equation 4.41 is satisfied. In figure 4.11 it can be seen that $\dot{\Psi}_{sd}$ is very low, even if $|\dot{n}| = 6000\ ^{rpm}/_{s}$. Equation 4.52 describes the steady state and 4.51 the dynamic behaviour good enough for this application.
Although $\dot{\Psi}_{sd}$ is negligible in the geometrical sum, it is necessary that the value of equation 4.51 is feasible by the bang-bang controller. Otherwise, the q-component $\dot{\varphi}\Psi_{sd}$ can not remain constant.
Figure 4.10a shows the needed stator flux over speed to keep the phase voltage around V_{bat}. It is calculated by the following steps:

1. $\hat{\mathrm{U}}_{ph,target} = V_{bat} = e.g. : 12.5V\ (\Delta U = 0V)$
2. equation 4.24: $\hat{\mathrm{U}}_s = \hat{\mathrm{U}}_{ph,target} + U_f = 13.3V$
3. $U_{sq} = \hat{\mathrm{U}}_s$
4. equation 4.52: $\Psi_{sd}(\omega) = \frac{U_{sq}}{\omega}$

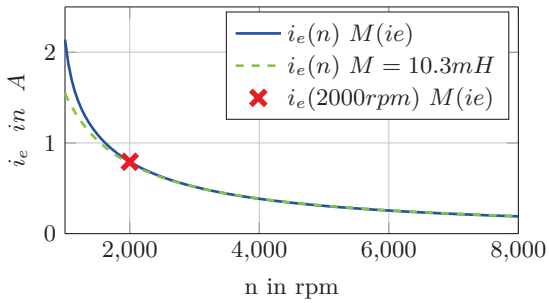
The needed stator flux $\Psi_{sd}(\omega)$ is independent of any alternator parameter. Only the pole pair number p is used to plot Ψ_{sd} over n in rpm, see figure 4.10a. However, depending on the alternator characteristics more or less excitation current is needed to produce Ψ_{sd}. The magnetization curve of the alternator is presented in figure 4.10b.



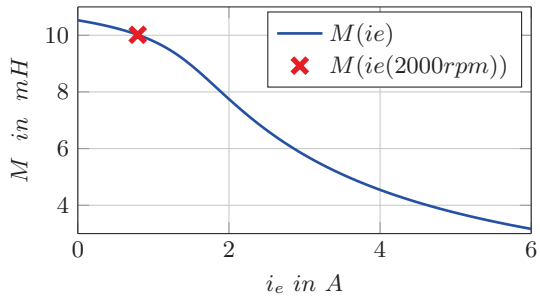
(a)$\Psi_{sd}(n)$ determined by equation 4.52



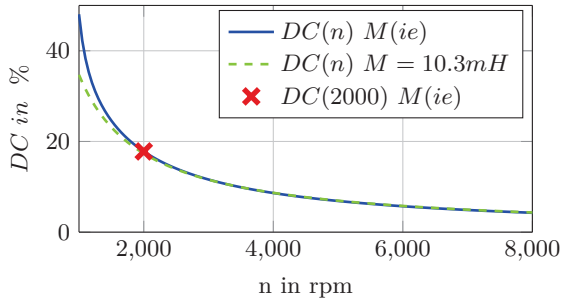
(b)Magnetization curve $\Psi_{sd}(i_e)$



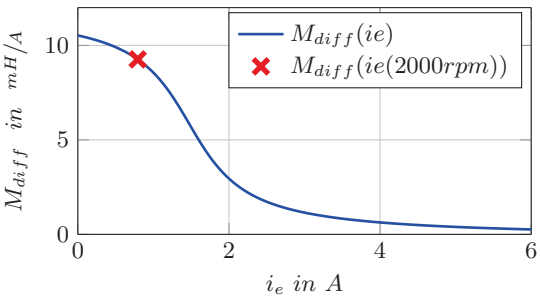
(c)i_e calculated by equation 4.77



(d)$M(i_e) = \frac{\Psi_{sd}(i_e)}{i_e}$



(e)Duty cycle given by equation 4.81



(f)$M_{diff}(i_e)$, see equation 4.97

Figure 4.10: Alternator Valeo A14

4.3.3 Feasibility of Ψ_{sd}, i_e and determination of the DC

The magnetization curve is a function of i_μ, which is given by:

$$\underline{i}_\mu = (\underline{i}_s + i'_e) \tag{4.71}$$

where i'_e is the excitation current referred to the stator winding. In case of idle run where $\underline{i}_s = 0$, the stator flux can be written as (see equation 4.37):

$$\Psi_s = \Psi_{sd,feasible} = L_h i'_{e,feasible} = M i_{e,feasible} \tag{4.72}$$

That means, the horizontal axis of figure 4.10b is equivalent to i_e and the vertical axis to Ψ_{sd} under the condition $\underline{i}_s = 0$. The average value of i_e is adjustable via the PWM DC:

$$i_{e,feasible} = DC\frac{V_{bat}}{R_e} \tag{4.73}$$

maximum value $DC = 1$:

$$\Psi_{sd,feasible_{max}} = M\frac{V_{bat}}{R_e} \tag{4.74}$$

minimum value $DC = 0$:

$$\Psi_{sd,feasible_{min}} = 0 \tag{4.75}$$

Depending on the speed ω and for a desired U_{sq} the following needs to be fulfilled:

$$0 \leq \frac{U_{sq}}{\omega} \leq M\frac{V_{bat}}{R_e} \tag{4.76}$$

Or seen from the current (equations4.37 and 4.52 ie):

$$i_e(\omega) = \frac{\Psi_{sd}}{M} = \frac{1}{M}\frac{U_{sq}}{\omega} \tag{4.77}$$

this condition must be fulfilled:

$$0 \leq i_e(\omega) \leq \frac{V_{bat}}{R_e} \tag{4.78}$$

If condition 4.76 or rather 4.78 is not fulfilled, the target value can not be reached $\hat{\mathrm{U}}_{ph} \neq \hat{\mathrm{U}}_{ph,target}$. That could be a problem for very low revolutions. The minimum speed for a desired voltage $U_{sq} = \hat{\mathrm{U}}_{ph,target}$ can be calculated by:

$$\omega_{min} = \frac{U_{sq}}{M\frac{V_{bat}}{R_e}} \tag{4.79}$$

Furthermore the DC is defined by:

$$DC = \frac{T_{High}}{T_{Period}} = \frac{u_e}{V_{bat}} \tag{4.80}$$

u_e is given by equation 4.9. At constant speed and steady state $\dot{i_e} = 0$. Consequently $\dot{\Psi}_e = 0$. Thus, this simplifies equation 4.9 to $u_e = R_e i_e$ and the static DC is given by:

$$DC(\omega, \dot{\omega} = 0) = \frac{i_e(\omega)R_e}{V_{bat}} \tag{4.81}$$

It can be shown that the DC is almost independent of V_{bat}. Replacing i_e by equation 4.77 gives:

$$DC = \frac{1}{M}\frac{U_{sq}}{\omega}\frac{R_e}{V_{bat}} \tag{4.82}$$

Hence, $U_{sq} = \hat{U}_{ph} + 2U_f = V_{bat} + U_f$ and $2U_f < V_{bat}$, 4.82 can be simplified to:

$$DC = \frac{1}{M}\frac{R_e}{\omega}\frac{V_{bat} + 2U_f}{V_{bat}} \tag{4.83}$$

$$DC \approx \frac{1}{M}\frac{R_e}{\omega} \tag{4.84}$$

The analytical calculation of i_e works with the simplification $M = const.$ Actually, M is a function of i_μ or rather i_e. In fact, Ψ_{sd} is calculated using equation 4.52 and with the magnetization curve of the alternator i_e is obtained by interpolation. An example is given in figure 4.10a and 4.10b for $n = 2000rpm$. The calculated flux by equation 4.52 is $\Psi_{sd}(2000rpm) = 7.934mVs$. By interpolating $i_e = f(\Psi_{sd})$ gives for $\Psi_{sd}(2000rpm)$ a current of $i_e(\Psi_{sd}(2000rpm)) = 0.8A$. In figure 4.10c and 4.10e it can be seen that the lower the speed, the more field current is needed, the greater the difference of the calculation method.
Table 4.3 shows the analytical calculated values of i_e and DC (by equation 4.83) for low revolutions. Conversely table 4.4 shows the numerically calculated values (by interpolation).

	$V_{bat} = 10V$		$V_{bat} = 12.5V$	
n	i_e	DC	i_e	DC
rpm	A	%	A	%
$n = 1000rpm$	1.25	35.2	1.55	34.7
$n = 1500rpm$	0.84	23.5	1.03	23.1
$n = 2100rpm$	0.59	16.8	0.73	16.5
$n = 3000rpm$	0.41	11.7	0.51	11.2

Table 4.3: Analytical calculation $M = 10.3mH$

	$V_{bat} = 10V$		$V_{bat} = 12.5V$	
n	i_e	DC	i_e	DC
rpm	A	%	A	%
$n = 1000rpm$	1.413	39.7	2.14	48.1
$n = 1500rpm$	0.86	24.3	1.10	24.6
$n = 2100rpm$	0.60	16.9	0.75	16.9
$n = 3000rpm$	0.41	11.7	0.52	11.6

Table 4.4: Numerical calculation $M(i_e)$

In case of the analytical calculation, $DC(n)$ is almost the same value for different values of V_{bat}. In the numerical case it is widely the similar except for very low revolutions. In other words, the needed DC to keep $\hat{U}_{ph} \approx V_{bat}$ depends only on the speed, if $n \geq 1500rpm$. The minimum speed is determined with $M = M(i_e = \frac{V_{bat}}{R_e})$, which gives a value of $n_{min} = 860rpm$.

4.3.4 Feasibility of $\dot{\Psi}_{sd}$

As long as the rotor speed is constant, $\dot{\Psi}_{sd} = 0$. In case of a dynamic speed, Ψ_{sd} must increase or decrease in order to keep $|\underline{u}_s|$ constant. The corresponding $\dot{\Psi}_{sd}$ is calculated by equation 4.40. Notwithstanding, the feasible $\dot{\Psi}_{sd}$ depends on the alternator characteristic. Obviously, on the ability of raising or decreasing the flux, which is done by regulating the field current. It can be concluded that, the rotor parameters are essential.
The rotor voltage u_e is given by equation 4.9 and the excitation flux by 4.16. With the

assumption $i_s = 0$, equation 4.16 is simplified to:

$$\Psi_e = L_{\sigma_e} i_e + L_{he} i_e = L_{\sigma_e} i_e + \Psi_{he} \tag{4.85}$$

L_{he} is replaced by equation 4.18:

$$\Psi_e = L_{\sigma_e} i_e + \frac{3}{2}\frac{1}{a}\Psi_{sd} \tag{4.86}$$

The derivative leads to:

$$\dot{\Psi}_e = L_{\sigma_e} \dot{i_e} + \frac{3}{2}\frac{1}{a}\dot{\Psi}_{sd} \tag{4.87}$$

For a better understanding, the calculation of $\Psi_{sd,feasbile}$ is done first analytical with the simplification $M = const.$ Afterwards, the numerical calculation is shown with $M = f(i_e)$.

Analytical

The derivative of equation 4.37 gives,

$$\dot{\Psi}_{sd} = (\dot{M} i_e) = M \dot{i_e} \tag{4.88}$$

equation 4.88 rearranged leads to:

$$\dot{i_e} = \frac{\dot{\Psi}_{sd}}{M} \tag{4.89}$$

equation 4.89 inserted in 4.87:

$$\dot{\Psi}_e = \left(\frac{L_{\sigma_e}}{M} + \frac{3}{2}\frac{1}{a}\right)\dot{\Psi}_{sd} \tag{4.90}$$

$\dot{\Psi}_e$ inserted in equation 4.9 gives:

$$u_e = R_e i_e + \left(\frac{L_{\sigma_e}}{L_g} + \frac{3}{2}\frac{1}{a}\right)\dot{\Psi}_{sd} \tag{4.91}$$

by rearranging equation 4.91 $\dot{\Psi}_{sd}$ is calculated to:

$$\dot{\Psi}_{sd} = \frac{u_e - R_e i_e}{(\frac{L_{\sigma_e}}{M} + \frac{3}{2}\frac{1}{a})} \tag{4.92}$$

With

$$u_e = DCV_{bat} \tag{4.93}$$

the feasible flux is given by:

$$\dot{\Psi}_{sd_{feasible}} = \frac{DCV_{bat} - R_e i_e}{\left(\frac{L_{\sigma_e}}{M} + \frac{3}{2}\frac{1}{a}\right)} \tag{4.94}$$

i_e is determined by 4.77.

Numerical

$$\Psi_{sd} = \Psi_{sd}(i_e(t)) \tag{4.95}$$

On the contrary to the previous calculation, $\dot{\Psi}_{sd}$ is a partial derivative now:

$$\dot{\Psi}_{sd} = \frac{\partial \Psi_{sd}}{\partial i_e} \dot{i_e} \tag{4.96}$$

Definition of the differential mutual inductance:

$$M_{diff}(i_e) := \left. \frac{\partial \Psi_{sd}}{\partial i_e} \right|_{i_e} \tag{4.97}$$

With definition 4.97, it follows:

$$\dot{i_e} = \frac{\dot{\Psi}_{sd}}{M_{diff}(i_e)} \tag{4.98}$$

The same steps as before lead to:

$$\dot{\Psi}_{sd} = \frac{u_e - R_e i_e}{(\frac{L_{\sigma_e}}{M_{diff}(i_e)} + \frac{3}{2}\frac{1}{a})} \tag{4.99}$$

i_e is determined by interpolation $i_e = f(\Psi_{sd})$, the same is done for $M_{diff} = f(i_e)$

$$\dot{\Psi}_{sd_{feasible}} = \frac{DCV_{bat} - R_e i_e}{\left(\frac{L_{\sigma_e}}{M_{diff}(i_e)} + \frac{3}{2}\frac{1}{a}\right)} \tag{4.100}$$

negative maximum value $DC = 0$:

$$\dot{\Psi}_{sd_{feasible,min}} = -\frac{R_e i_e}{\left(\frac{L_{\sigma_e}}{M_{diff}(i_e)} + \frac{3}{2}\frac{1}{a}\right)} \tag{4.101}$$

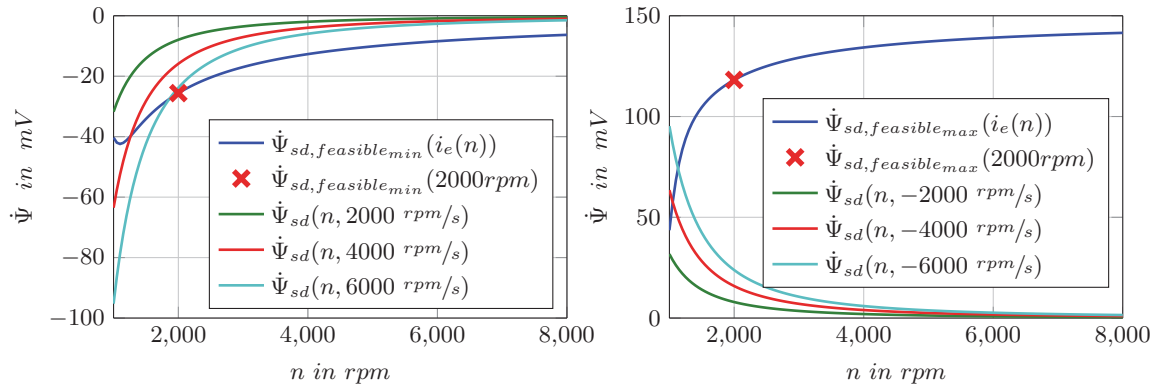
positive maximum value $DC = 1$:

$$\dot{\Psi}_{sd_{feasible,max}} = \frac{V_{bat} - R_e i_e}{\left(\frac{L_{\sigma_e}}{M_{diff}(i_e)} + \frac{3}{2}\frac{1}{a}\right)} \tag{4.102}$$

With equation 4.57 and 4.101, 4.102 the following condition can be expressed:

$$\dot{\Psi}_{sd_{feasible,min}} \leq -\frac{\ddot{\varphi}}{\dot{\varphi}^2} U_{sq} \leq \dot{\Psi}_{sd_{feasible,max}} \tag{4.103}$$

The calculated $\dot{\Psi}_{sd}$ by equation 4.57 is negative in case of an increasing speed. The reduction of the flux is too slow, if $\dot{\Psi}_{sd}$ is lower than $\dot{\Psi}_{sd_{feasible,min}}(i_e(n))$. As a result, $\hat{\mathrm{U}}_{ph}$ will exceed $\hat{\mathrm{U}}_{ph,target}$. If $\dot{\Psi}_{sd}$ exceeds $\dot{\Psi}_{sd_{feasible,max}}(i_e(n))$, then the increase of the field is too slow. $\hat{\mathrm{U}}_{ph}$ will be lower than $\hat{\mathrm{U}}_{ph,target}$. Figure 4.11a shows the $\dot{\Psi}_{sd,feasible}$ and $\dot{\Psi}_{sd}$ for different positive speed gradients. The same is shown in figure 4.11b for negative gradients. It can be seen that $|\dot{\Psi}_{sd_{feasible,min}}|(n)$ is lower than $\dot{\Psi}_{sd_{feasible,max}}(n)$. That means, increasing the flux can be done faster than decreasing. The reason is clear: the field current rises by applying $U_{exc} = V_{bat}$, while reducing is achieved passively by $U_{exc} = 0V$.



(a)Increasing speed (b)Decreasing speed

Figure 4.11: Calculation of $\dot{\Psi}_{sd,feasible}(n)$ and $\dot{\Psi}_{sd}(n,\dot{n})$

4.3.5 Summary

desired voltage:	$\lvert\underline{u}_s^r\rvert \approx U_{sq} \approx \hat{U}_{ph,target} + U_f$
stator flux by equation 4.52:	$\Psi_{sd}(\omega) = \frac{U_{sq}}{\omega}$
feasible stator flux by magnetization curve:	$\Psi_{sd}(i_e)$
field current i_e by interpolation:	$i_e = f(\Psi_{sd})$
feasible current i_e by equation 4.73:	$i_{e,feasible} = DC\frac{V_{bat}}{R_e}$
dynamic stator flux by equation 4.51:	$\dot{\Psi}_{sd}(\omega,\dot{\omega}) = -\frac{\omega}{\dot{\omega}}\Psi_{sd}(\omega)$
feasible dynamic by equation 4.100:	$\dot{\Psi}_{sd_{feasible}} = \frac{DCV_{bat} - R_e i_e}{\left(\frac{L_{\sigma_e}}{M_{diff}(i_e)} + \frac{3}{2}\frac{1}{a}\right)}$

4.4 Phase Controller

The phase controller is a simple bang bang controller (ON/OFF). The main task is to maintain $V_{ref} - TH_2 \leq \hat{U}_{ph} \leq V_{ref} + TH_1$. It consists of comparators and the logic elements. The basic function is explained in section 4.1. The reference value V_{ref} and the relative thresholds TH_1, TH_2 are adjustable by the end user.

4.4.1 Comparator Reference Voltage V_{ref}

V_{ref} is given by: $V_{ref} = \hat{U}_{ph,target} = V_{ba} + \Delta U$. The idea is that ΔU is adjustable. The problematic of implementing an adjustable analogue voltage in the IC is not considered by this thesis (accuracy, drift,...).

Constant rotor speed

The critical state of charging/ not charging is reached if $\hat{U}_s = V_{bat} + 2U_f$ (see section 3.1). From this point of view, the most efficient value of ΔU is the forward voltage U_f. $\Delta U = U_f$ gives $\hat{U}_{ph,target} = V_{bat} + U_f$, which in turn leads to $U_s = V_{bat} + 2U_f = U_{s,crit}$ (see section 4.2.2 or rather 3.1). Hence the bang bang controller works with a hysteresis, $\hat{U}_{ph}$ exceeds $\hat{U}_{ph,target}$ temporary and may cause $I_{gen} > 0A$.
Figure 4.12 shows a simulation with $\Delta U = 0.7V$ and $TH_1 = TH_2 = 0.1V$. The forward voltage of the diodes is $U_f = 0.8V$. It can be seen, that there are short moments with $I_{gen} \geq 0A$. The first peak is about $0.2A$ and the rest is about $0.05A$. These small values are no issue at all, especially if rotational inertia is considered. Nevertheless, to operate close to the critical value is risky in terms of robustness. For example U_f strongly depends on temperature. Since the thesis proceeds in general on a worst case scenario, most simulations are conducted with $\Delta U = 0V$, $V_{ref} = V_{bat}$ for constant speeds (see figure 4.4).
The lower ΔU, the lower the handover value, the greater the delay time Δt_2. Table 4.5 shows the calculated DC (see section 4.3.3) for different ΔU values. The greater the speed, the lower the difference. For speeds above the typical idle speed ($2100rpm$) the difference is negligible low. Note, that the flux is calculated by $\hat{U}_{ph,target} = V_{bat} + \Delta U$, $\hat{U}_s = \hat{U}_{ph,target} + U_f$.

	$\Delta U = -2V$	$\Delta U = -1V$	$\Delta U = 0V$	$\Delta U = 0.8V$
$n = 1000rpm$	$DC = 34\%$	$DC = 39.6\%$	$DC = 48\%$	$DC = 59\%$
$n = 1500rpm$	$DC = 20.4\%$	$DC = 22.5\%$	$DC = 24.6\%$	$DC = 26.4\%$
$n = 2100rpm$	$DC = 14.2\%$	$DC = 15.5\%$	$DC = 16.9\%$	$DC = 18\%$
$n = 3000rpm$	$DC = 9.6\%$	$DC = 10.7\%$	$DC = 11.6\%$	$DC = 12.3\%$

Table 4.5: Calculated (static) *DC*

Consequently i_e is determined for $V_{bat} + \Delta U$. However, the DC refers always to V_{bat} (see equation 4.81). This, makes a difference to table 4.4 where i_e is determined for different values of V_{bat}.

Dynamic rotor speed

Figure 4.13 shows a simulation with $\Delta U = 0V$, $V_{ref} = V_{BA}$ and an increase of the rotor speed from $1000rpm$ to $3000rpm$ within $1/3s$ ($6000^{rpm}/_{s}$). Obviously, the output current rises, although the bang bang switches off. This is exactly the problem which had been explained in section 4.3.4. In figure 4.11a it can be seen that until $2000rpm$, $\dot{\Psi}_{sd}(n, 6000\ ^{rpm}/_{s})$ is lower than the $\dot{\Psi}_{sd,feasible}$. In other words, the speed increases faster than the field can be reduced. It is debatable that $10A$ ($120W \sim 0.1Nm$)output current matter, when the speed increases with $6000^{rpm}/_{s}$. Notwithstanding, a method is developed how to deal with this kind of situations. Actually, the only way out is, to make ΔU negative for low revolutions

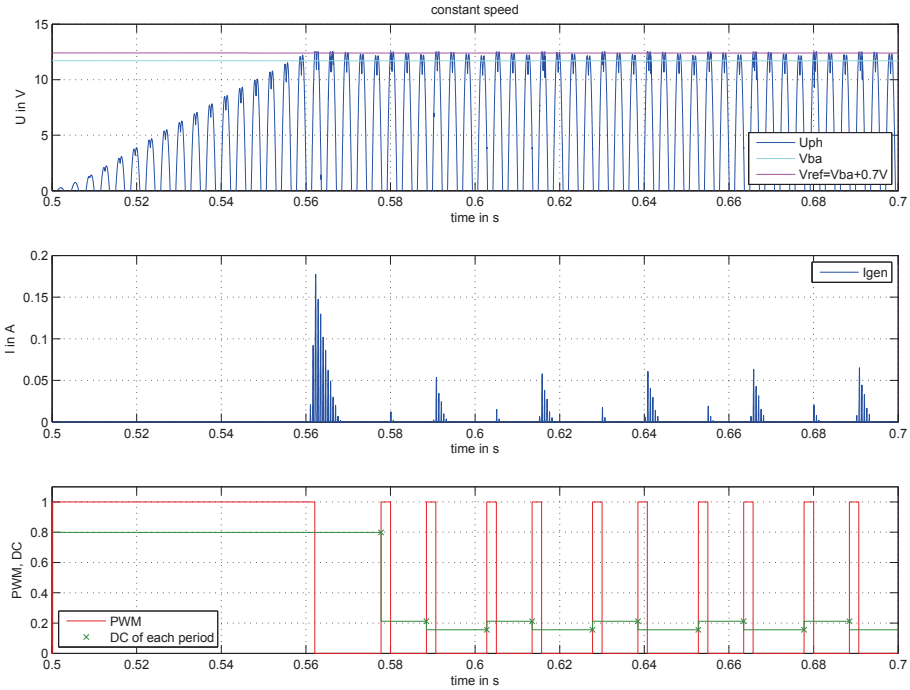


Figure 4.12: Simulation with $\Delta U \approx U_f$

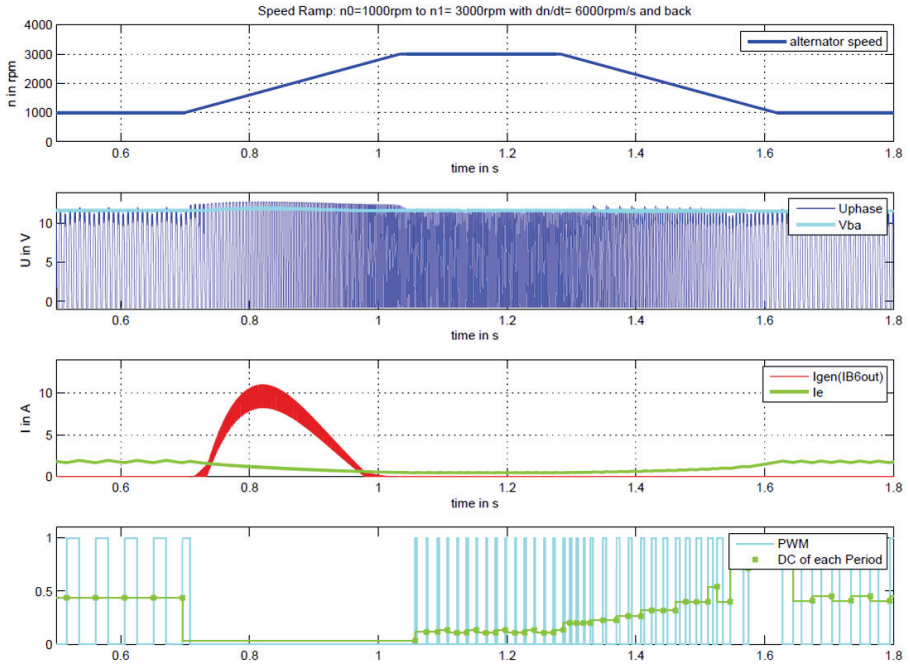


Figure 4.13: Simulation with $\Delta U = 0,\ V_{ref} = V_{BA}$

$\Rightarrow V_{ref} < V_{bat}$. This raises the question how much V_{ref} should be reduced. Of course, this causes additional delay time. The idea is to assume a maximum speed gradient, for example $7500^{rpm}/_{s}$. For this gradient the value of V_{ref} or rather ΔU is calculated, so that $\hat{U}_{ph}$ will not exceed the critical value. At first, the calculation is shown analytically with the assumption of constant parameters (no saturation). Afterwards, the numerical way, which considers saturation, is briefly given.

Analyctical:
Basis is equation 4.66, hence the value of $\dot{\Psi}_{sd}$ is negligible in equation 4.41.

$$|u_s| = u_{sd} = \omega(t)\Psi_{sd}(t) \tag{4.104}$$

It is assumed that the increase of the speed starts at t_0. At the same time, the bang-bang controller switches off. It may happen that the switch off is delayed one electrical period (depending on detection of $\hat{U}_{ph}$). From that point onward, the field current is given by:

$$i_{e0} := i_e(t = t_0) \tag{4.105}$$

$$i(t) = i_{e0}e^{-\frac{t}{\tau_e}} \quad \forall \quad t \geq t_0 \tag{4.106}$$

The flux is given by equation 4.38:

$$\Psi_{sd}(t) = Mi_e(t) = Mi_{e0}e^{-\frac{t}{\tau_e}} = \Psi_{sd0}e^{-\frac{t}{\tau_e}} \tag{4.107}$$

Note that the trace of the stator flux depends on the rotor time constant τ_e ($i_s = 0$). Ψ_{sd0} is calculated by equation 4.52:

$$\Psi_{sd0} = \frac{U_{sq0}}{\omega_0} \approx \frac{\hat{U}_{s0}}{\omega_0} \tag{4.108}$$

Using equations 4.107 and 4.108, $|u_s|$ can be rewritten as:

$$|u_s| = (\omega_0 + \dot{\omega}t)\frac{\hat{U}_{s0}}{\omega_0}e^{-\frac{t}{\tau_e}} \tag{4.109}$$

The idea is to calculate the maxima of the stator voltage:

$$\frac{d|u_s|}{dt} \stackrel{!}{=} 0 \tag{4.110}$$

The derivation leads to:

$$\frac{d|u_s|}{dt} = \hat{U}_{s0}e^{-\frac{t}{\tau_e}}\left(-\frac{1}{\tau_e}\frac{\dot{\omega}}{\omega_0}t - \frac{1}{\tau_e} + \frac{\dot{\omega}}{\omega_0}\right) \stackrel{!}{=} 0 \tag{4.111}$$

Equation 4.111 rearranged, gives the point in time of the maxima:

$$t_{max} = \tau_e - \frac{\omega_0}{\dot{\omega}} \tag{4.112}$$

That implies:

$$\tau_e \geq \frac{\omega_0}{\dot{\omega}} \tag{4.113}$$

The stator flux declines faster than the speed increases if condition 4.113 is not fulfilled. $t = t_{max}$ in equation 4.109 gives:

$$|u_{smax}| = |u_s(t = t_{max})| = \hat{U}_{s0} e^{-\left(1-\frac{\omega_0}{\tau_e \dot{\omega}}\right)\frac{\dot{\omega}\tau_e}{\omega_0}} \tag{4.114}$$

$$K(\omega, \dot{\omega}) := e^{-\left(1-\frac{\omega_0}{\tau_e \dot{\omega}}\right)\frac{\dot{\omega}\tau_e}{\omega_0}} \tag{4.115}$$

$$|u_{smax}| = \hat{U}_{s0} K \tag{4.116}$$

The idea is, to set $|u_{smax}| = \hat{U}_{s,crit}$ and calculate $\hat{U}_{s,0}$:

$$\hat{U}_{s0}(\omega, \dot{\omega}) = \frac{\hat{U}_{s,crit}}{K(\omega, \dot{\omega})} = \frac{V_{bat} + 2U_f}{K(\omega, \dot{\omega})} \tag{4.117}$$

The phase value, referred to *gnd*, is determined by equation 4.24:

$$\hat{U}_{ph0}(\omega, \dot{\omega}) = \hat{U}_{s0}(\omega, \dot{\omega}) - U_f \tag{4.118}$$

Finally, $\Delta U(\omega, \dot{\omega})$ is given by:

$$\Delta U = \hat{U}_{ph0} - \hat{U}_{ph,crit} = \frac{\hat{U}_{s,crit}}{K} - U_f - (V_{bat} + U_f) = \frac{V_{bat} + 2U_f}{K} - (V_{bat} + 2U_f) \tag{4.119}$$

Numerical:
The numerical calculation is done in the same way. The point is, that the rotor time "constant" τ_e is a function of i_e, because of saturation. With equation 4.18 the rotor flux is calculated by:

$$\Psi_e = (L_{\sigma e} + L_{he}(i_e))i_e = (L_{\sigma e} + \frac{3}{2}\frac{1}{a}M(i_e))i_e \tag{4.120}$$

The idea is to simulate the trace $\Psi_e(i_e(t))$ by implementing the rotor equation 4.9 $\dot{\Psi}_e = u_e - i_e R_e$. The simulink model is presented in figure 4.14. Since the magnetization curve is given, $M(i_e)$ is given. $\Psi_e(i_e)$ is determined for $0A < i_e < \frac{V_{bat}}{R_e}$ (equation 4.120). The inverse $i_e(\Psi_e)$ is implemented as a lookup table. Also $\Psi_{sd}(i_e)$ is implemented as lookup table. The trace of the stator flux is needed for further calculations in section 4.4.2. The initial condition of the integrator is set to the maximum value $\Psi_{e0} = \Psi_e(\frac{V_{bat}}{R_e})$ and $u_e = 0$ (decreasing field current).
Figure 4.15a shows the simulated trace of $i_e(t)$ and figure 4.15b $\Psi_e(i_e(t))$. By using the Matlab curve-fitting toolbox, an exponential function with constant parameters $\Psi_{e0,fit} e^{-\frac{t}{\tau_{fit}}}$ is fitted in the trace of $\Psi_e(i_e(t))$.

Equations 4.114-4.119 are calculated with $\tau_e = \tau_{fit}$. The trace of ΔU over n for different speed gradients $\dot{n}$ is shown in figure 4.15c. The same calculation is repeated with different values of V_{bat} and $\dot{n} = 7500^{rpm}/_s$, see figure 4.15d. Also, two possible IC implementations $\Delta U(n)$ are plotted. $\Delta U_{IC1}(n)$ is more conservative than $\Delta U_{IC2}(n)$.

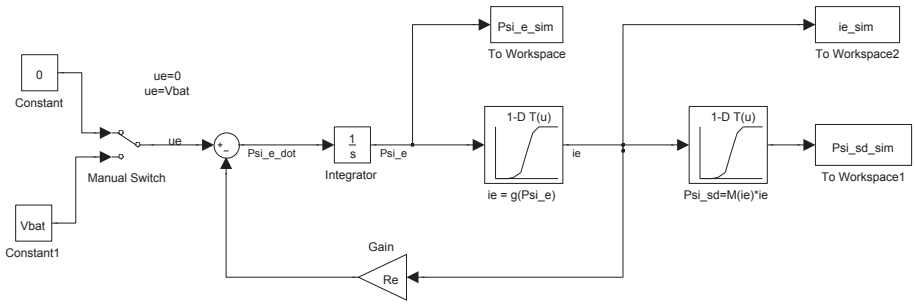
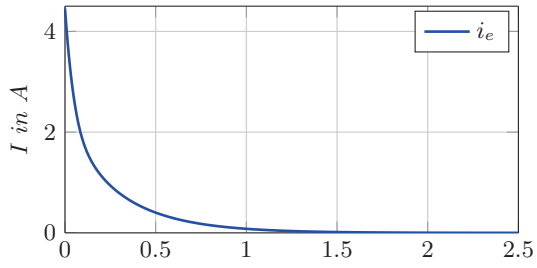
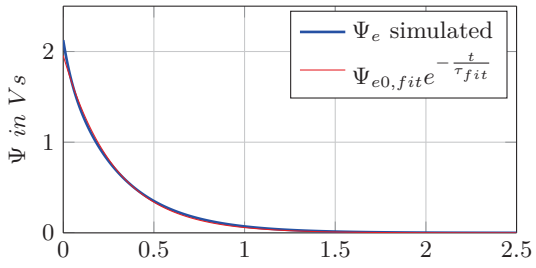


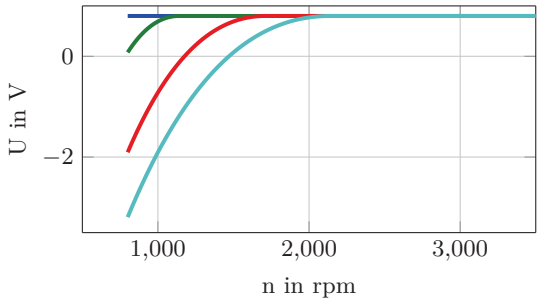
Figure 4.14: Simulation with $\Delta U = 0,\ V_{ref} = V_{BA}$



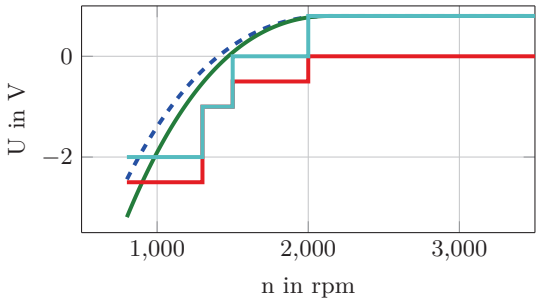
(a)Simulate field current



(b)Simulated stator flux and fitted curve



(c)Calculated $\Delta U(n, \dot{n}, , V_{bat} = 12.5V)$
- $\Delta U(n, \dot{n} = 2000^{rpm}/_{s})$
- $\Delta U(n, \dot{n} = 4000^{rpm}/_{s})$
- $\Delta U(n, \dot{v} = 6000^{rpm}/_{s})$
- $\Delta U(n, \dot{n} = 7500^{rpm}/_{s})$



(d)Calculated $\Delta U(n, \dot{n} = 7500^{rpm}/_{s}, V_{bat})$
- $\Delta U(n, V_{bat} = 10V)$
- $\Delta U(n, V_{bat} = 12.5V)$
- $\Delta U(n)_{IC1}$
- $\Delta U(n)_{IC2}$

Figure 4.15: Numerical calculation $\Delta U(n, \dot{n})$

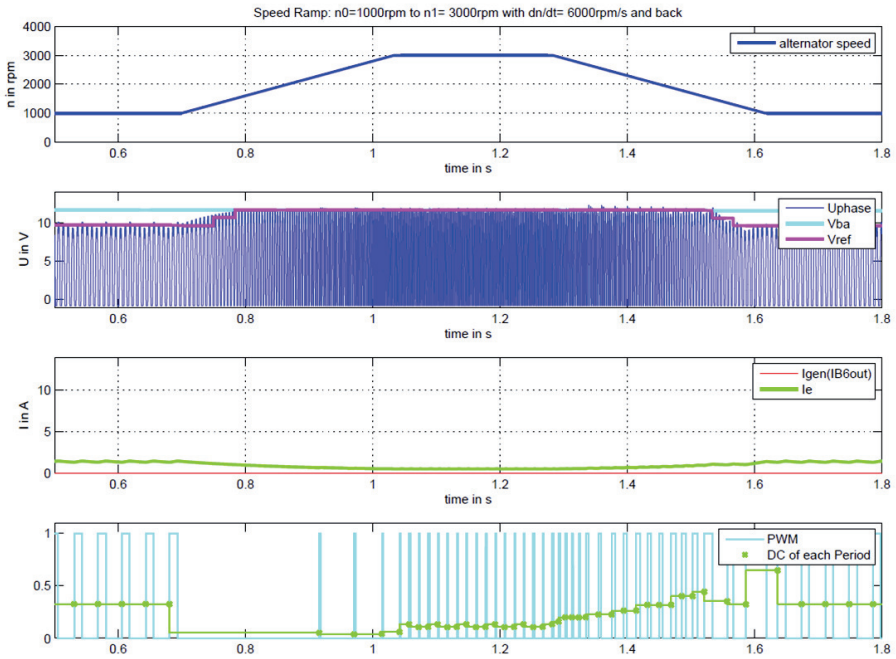


Figure 4.16: Simulation with $\Delta U_{IC1}(n)$

The simulation from the beginning (see figure 4.13) is repeated, but this time $V_{ref} = V_{ba} + \Delta U(n)$. The result is given in figure 4.16. It can be seen that a critical overshoot of the U_{ph} is avoided. On the other hand, declining V_{ref} results in a lower handover value (see table 4.5). Finally, the end user has to decide if this action is necessary or not. Moreover, $\Delta U(n)$ depends on the rotor parameter $\tau_e(i_e)$, which differs from alternator to alternator.

Temperature influences

As mentioned before, U_f decreases with rising temperature. As a result the critical value $U_{s,crit}$ declines. Consequently, ΔU depends on temperature. However, $\Delta U = 0V$ is more than enough margin for constant speed traces.

Eddy current influences

The influence is not relevant for a constant speed, because the induced voltage depends on the average value of the field current. Caution is needed for the dynamic speed traces. Investigations in section 5.1.2 have shown, that the real reduction of the field is slower than simulated. The reason is, that eddy currents oppose field changes. Subsequently, the calculated reduction by equation 4.101 is too fast. If it comes to a phase voltage overshoot

in consequence of a rapid speed increase, it lasts longer or rather the maximum is higher. The dynamic declining of V_{ref} or rather $\Delta U(n)$ is calculated too neglectful in terms of eddy currents. Nevertheless, the supposed IC implementation $\Delta U(n)_{IC1}$ is lower than $\Delta U(n)$ (see figure 4.15d), therefore the consequences should be minor.

4.4.2 Comparator Thresholds (relative)

The thresholds TH_1 and TH_2 are relative values to V_{ref}. Since $U_{exc} = V_{ba}$ as long as $\hat{U}_{ph} < V_{ref} + TH_1$, it is important to choose TH_1 carefully. The average DC of the bang-bang controller PWM correlates with V_{ref} if the thresholds are chosen symmetric $TH_1 = TH_2$

Constant rotor speed

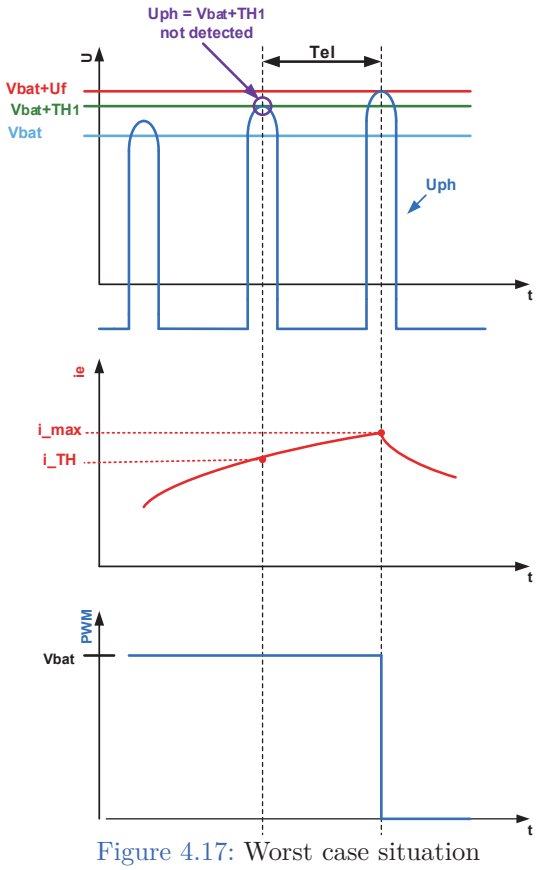


Figure 4.17: Worst case situation

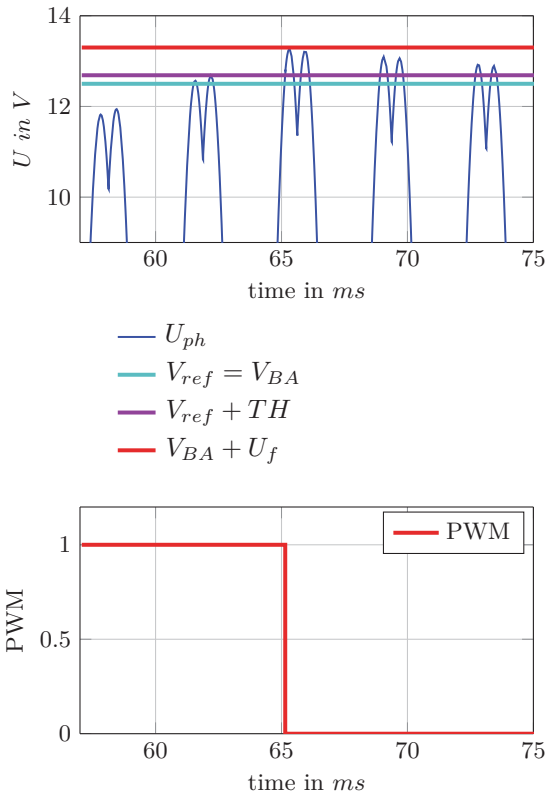


Figure 4.18: Worst case simulation

To figure out a reasonable value of TH_1, a worst case situation is considered, see figure 4.17. It is assumed that $\hat{U}_{ph}$ is slightly lower than $V_{ref} + TH_1$, or the detection $\hat{U}_{ph} = V_{ref} + TH_1$ has not worked. As a result, the bang bang controller does not switch off. The field current

is further increased for one electrical period. After that period the bang bang controller switches off surely, because $\hat{U}_{ph}$ is definitely greater than $V_{ref} + TH_1$. The idea is to calculate the maximum value of TH_1, where $\hat{U}_{ph}$ is still less or equal $\hat{U}_{ph,crit}$. Again the calculation is done analytically with the assumption of constant alternator parameters and afterwards numerical with saturable parameters. The electrical period T_{EL} depends on the rotor speed and the pole pair number (see equation 4.30).

Analytical:
Basically, the field current $i_{e,crit}$ for the critical value $\hat{U}_s = \hat{U}_{s,crit}$ is determined, and the current value one period before $\Rightarrow i_{e,TH}$. The excitation circuit is in principle an RL circuit:

$$u_{R_e}(t) + u_{L_e} = U_{exc} \tag{4.121}$$

$$i_e(t)R_e + L_e\frac{di}{dt} = V_{BA} = V_{bat} \tag{4.122}$$

$$\frac{di}{dt} + \frac{R_e}{L_e}i_e(t) = \frac{V_{bat}}{L_e} \tag{4.123}$$

The homogeneous solution of equation 4.123 can be found by separation of the variables, the particular by using a constant approach $i_{e,p} = C$. The final solution is given by:

$$i_e(t) = \frac{V_{bat}}{R_e}(1 - e^{-\frac{t}{\tau_e}}) + i_e(t_0)e^{-\frac{t}{\tau_e}} \tag{4.124}$$

The excitation current is also determined by equation 4.77. With the assumption $U_{sq} \approx \hat{U}_{sq}$ the critical current is given by:

$$i_{e,crit}(\omega) = \frac{\hat{U}_{s,crit}}{\omega M} = \frac{V_{bat} + 2U_f}{\omega M} \tag{4.125}$$

With the following definition and rearranging equation 4.124, $i_{e,TH}$ is calculated by:

$$t = T_{EL} \qquad i_e(t=0) = i_{e,crit} \qquad i_e(t_0) = i_e(t - T_{EL}) = i_{e,TH} \tag{4.126}$$

$$i_{e,TH}(\omega) = -\frac{V_{bat}}{R_e}(e^{\frac{T_{EL}}{\tau_e}} - 1) + i_{e,crit}e^{\frac{T_{EL}}{\tau_e}} \tag{4.127}$$

Using equation 4.68 and 4.24 leads to:

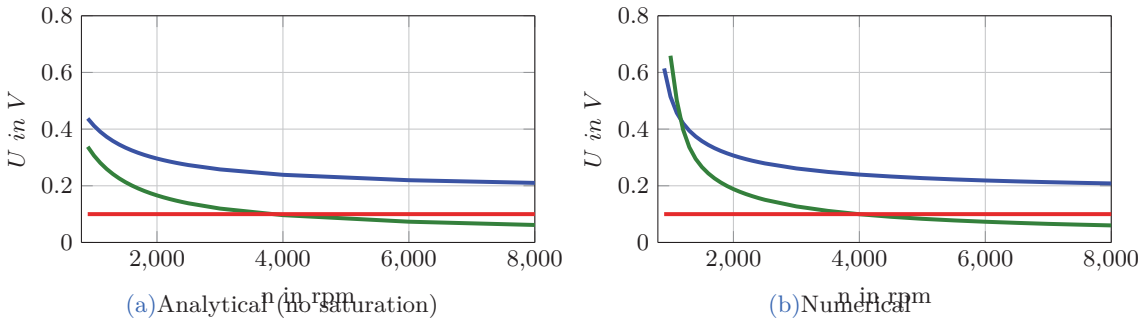
$$\hat{U}_{ph,TH}(\omega) = \hat{U}_{s,TH}(\omega) - U_f = \omega M i_{e,TH}(\omega) - U_f \tag{4.128}$$

Finally, the relative threshold is calculated by:

$$TH_1(\omega) = \hat{U}_{ph,TH} - V_{ref} \tag{4.129}$$

It is assumed that:

$$V_{ref} = V_{bat} + \Delta U < \hat{U}_{ph,TH}(\omega) \tag{4.130}$$



(a)Analytical (no saturation) (b)Numerical

Figure 4.19: Calculated threshold $TH_1(n, \dot{n} = 0)$,
$TH_1(n, V_{bat} = 10V)$, $TH_1(n, V_{bat} = 12.5V)$, $\Delta U(n)_{IC1}$

If condition 4.130 is not fulfilled, then ΔU must be declined. Figure 4.19a shows the trace of TH_1 over n for two different values of V_{bat} and $\Delta U = 0V$. Surprisingly, TH_1 declines with increasing speed. The reason is that for a low speed, a higher field current is needed. The higher i_e, the lower the gradient $\dot{i_e}$. As a consequence, i_e can not increase much during one electrical period. For that reason, the worst case situation is not as bad for low speeds as for high speeds. On the contrary, if it comes to an overshoot of $\hat{U}_{ph}$ the reduction of i_e is slower for low speeds than for higher speeds. As result the overshoot lasts longer.
Furthermore, it can be seen that the analytical calculation gives $TH_1(V_{bat} = 12.5V) < TH_1(V_{bat} = 10V)$. The reason is that τ_e is in both cases the same. For the same time span, the current increases faster if the amplitude of the applied voltage is higher. Consequently, TH_1 has to be lower. The traces in figure 4.19a are calculated for $V_{ref} = V_{bat}$.

Numerical:
The numerical calculation considers the flux. Nevertheless, the steps are the same as for the analytical calculation. At first, the stator flux $\Psi_{sd,crit}$ for the critical value $\hat{U}_s = \hat{U}_{s,crit}$ is determined and the flux value one period before $\Rightarrow \Psi_{sd,TH}$. For this purpose, the trace of $\Psi_{sd}(i_e(t))$ is simulated, see figure 4.14. On the contrary to the simulation in section 4.4.1, the initial value of the integrator is set to zero $\Psi_{e0} = 0$ and $u_e = V_{bat}$ (rising field current). Figure 4.20a shows the simulated trace of $i_e(t)$ and 4.20b $\Psi_{sd}(i_e(t))$. The critical flux is calculated by equations 4.52 and 4.69:

$$\Psi_{sd,crit}(\omega) = \frac{\hat{U}_{s,crit}}{\omega} = \frac{V_{bat} + 2U_f}{\omega} \tag{4.131}$$

As an example, the value of $\Psi_{sd,crit}(n = 2000rpm)$ is marked with a red cross in figure 4.20b. That point in time is defined as t_1. The value at $t_0 = t_1 - T_{EL}$ gives the threshold value $\Psi_{sd,TH}(n = 2000rpm)$ (marked with a green circle). With the interpolated value $\Psi_{sd,TH}(n)$ or rather $\Psi_{sd,TH}(\omega)$ at $t_0(\omega)$, the absolute threshold $U_{ph,TH}(\omega)$=$U_{s,TH}(\omega) - U_f$ is calculated by:

$$U_{ph,TH} = \omega \Psi_{sd,TH}(\omega) - U_f \tag{4.132}$$

Figure 4.19b shows the relative threshold $TH_1(\omega)$ determined by equation 4.129. Due to saturation effects $TH_1(V_{bat} = 12.5V) > TH_1(V_{bat} = 10V)$ at the very beginning. A worst-case simulation with $n = 2000rpm$ and the determined $TH_1(n = 2000rpm) = 0.187V$ is presented in figure 4.18. The point in time for the activation of the phase controller was extra determined to enforce the worst-case situation. It can be seen that $\hat{U}_{ph}$ does not exceed $V_{bat} + U_f$, although $\hat{U}_{ph} \approx V_{bat} + TH_1$ is not detected in the first place. For any TH_1 values lower than $TH_1(n = 2000rpm)$, the bang bang controller would have switched off immediately. In general, a constant value for TH_1 over speed is desired. As shown in figure 4.19b, $TH_1(n) = 0.1V$ covers the worst case situation from low revolutions up to $4000rpm$, in case of $V_{bat} = V_{ba} = 12.5V$ (no load connected to the battery load). As mentioned before, a critical overshoot of $\hat{U}_{ph}$ at higher speeds is less significant than at lower speeds. In case of a connected load to the battery, V_{bat} declines and the choice $TH_1 = 0.1V$ is less problematic.

The calculations show that the range of ΔU and TH_1 is very limited. For example, $TH_1 = 0.1V$ refers to $\Delta U = 0V$. Probably, it is hard to implement a comparator with such a small hysteresis $\pm 0.1V$. The point is, that these calculations are done for a worst case situation and the condition that $\hat{U}_{ph}$ never exceeds $\hat{U}_{ph,crit}$. Perhaps, these constraints are too hard. The central statement is: the closer TH_1 to V_{ref}, the better.

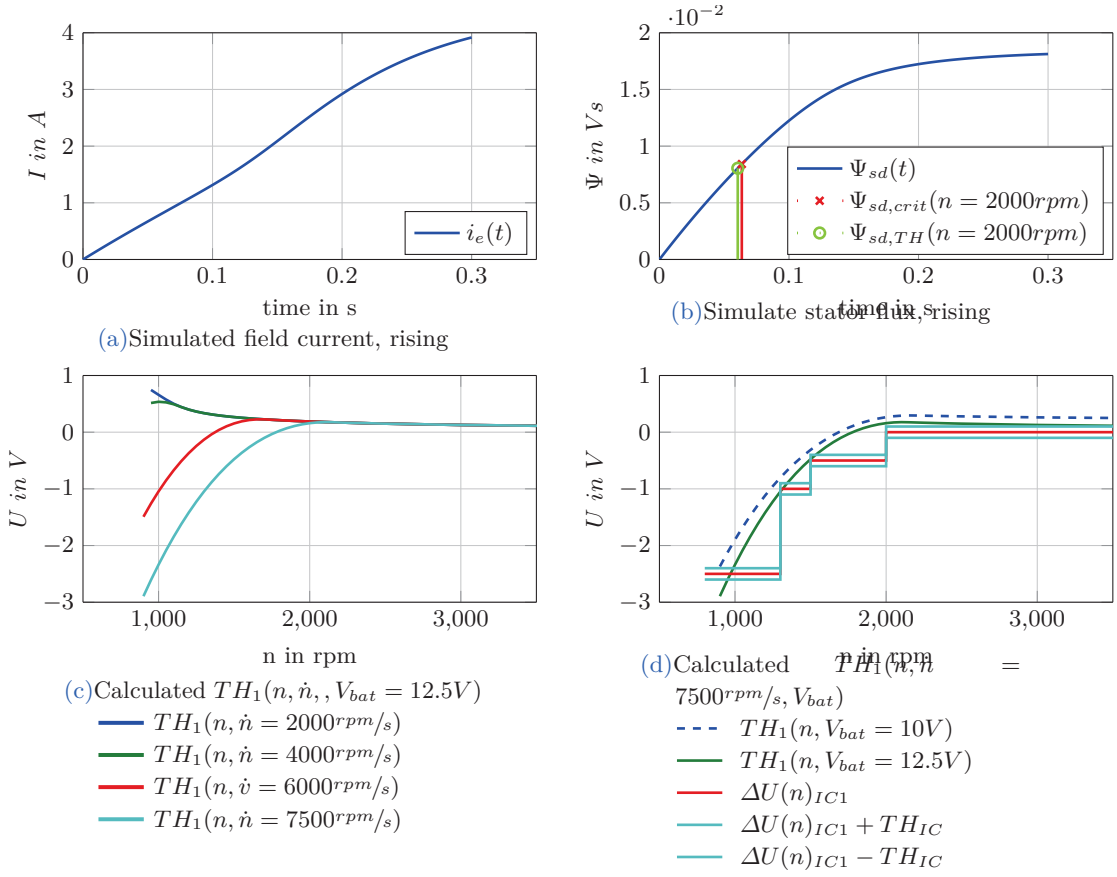
Dynamic rotor speed

The combination of the previous calculation and the dynamic $V_{ref}(\omega, \hat{\omega})$ calculation (see section 4.4.1) gives the dynamic threshold $TH_1(\omega, \dot{\omega})$. The idea is to replace the fix value $\hat{U}_{s,crit}$ in equation 4.131 with $\hat{U}_{s,0}(\omega, \dot{\omega})$ (equation 4.117):

$$\Psi_{sd,crit}(\omega, \dot{\omega}) = \frac{\hat{U}_{s,0}(\omega, \dot{\omega})}{\omega} \tag{4.133}$$

The rest of the calculation is done in the same way as before. Figure 4.20c shows the trace of $TH_1(n)$ for different speed gradients (see also figure 4.15c) and $V_{bat} = 12.5V$. The values are referred to as $\Delta U = 0V$, $V_{ref} = V_{ba}$. The same is given in figure 4.20d with $\dot{n} = 7500rpm$ and two different values of V_{ba}. Also a possible IC implementation is plotted. The thresholds relative to $V_{ref} = V_{bat} + \Delta U_{IC1}$ are fixed with $TH_{IC} = TH_1 = TH_2 = 0.1V$. ΔU_{IC1} is given in section 4.4.1, see figure 4.15d. The value of TH_{IC} is taken from the calculations with constant rotor speed ($\Delta U = 0V, TH_1 = 0.1V$). To ensure $\hat{U}_{ph} \leq \hat{U}_{ph,crit}$, the following condition has to be fulfilled:

$$\cancel{V_{bat}} + \Delta U_{IC1}(n) + TH_{IC} \leq \cancel{V_{bat}} + TH_1(n, \dot{n}) \tag{4.134}$$



(a)Simulated field current, rising

(b)Simulate stator flux, rising

(c)Calculated $TH_1(n, \dot{n}, , V_{bat} = 12.5V)$

(d)Calculated $TH_1(n, \dot{n} = 7500^{rpm}/_s, V_{bat})$

Figure 4.20: Numerical calculation $TH_1(n, \dot{n})$

Temperature influences

$\hat{U}_{s,crit}$ is defined as maximum allowed value. Since U_f decreases with rising temperature, $\hat{U}_{s,crit}$ decreases as well. The calculation of threshold values should be done with the critical value at hot temperature. As a consequence, the trace of $TH(n)$ in figure 4.19 would be lower. Nevertheless, a fixed threshold value of $0.1V$ seems still reasonable.

Eddy current influences

Real alternator measurements have shown the flux increases, slower than simulated (see sections 5.1.2 and 5.1.3). As a consequence U_{ph} increases not as fast as assumed. The threshold values are calculated too conservative. Probably TH_1 can be chosen greater than $0.1V$ in terms of eddy currents.

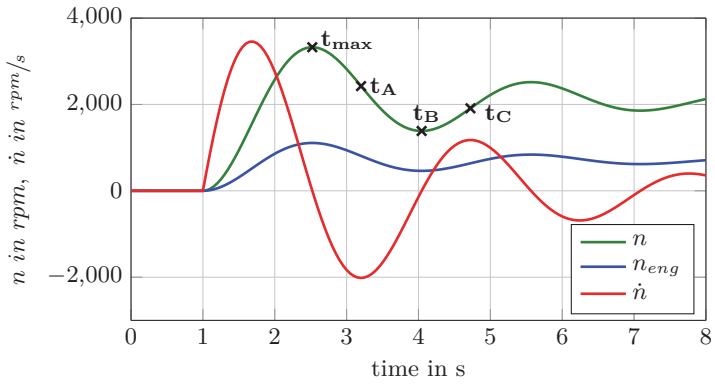


Figure 4.21: standard (dynamic) speed trace

4.4.3 Dynamic Speed trace

Based on measurements from BMW, the engine speed trace during the start process is approximated by a second order system:

$$G(s)_n = \frac{K_p}{T^2 s^2 + 2dTs^2 + 1} \tag{4.135}$$

$$K_p = n_{eng,ss} \qquad m = \frac{n_{eng,max}}{n_{eng,ss}} \qquad d = \frac{1}{\sqrt{1 + \left(\frac{\pi}{ln(m)}\right)^2}} \qquad T = \frac{\Delta t_{max}}{\pi} \tag{4.136}$$

$$n = a_{mech} n_{eng} \tag{4.137}$$

t_{max} is the point in time, when the speed trace reaches the peak value $n_{eng,max}$. $n_{eng,ss}$ is the steady-state value, which is the idle run speed. K_p is set $n_{eng,ss}$, because a step with amplitude one is applied to the transfer function $G(s)$. The following values are adopted from the measurements and considered as standard/default values:

$$n_{eng,ss} = 700rpm \qquad n_{eng,max} = 1200rpm \qquad \Delta t_{max} = 1.5s \qquad a_{mech} = 3 \tag{4.138}$$

Figure 4.21 shows the standard speed trace. Hence the speed trace is started at $t = 1$, the (first) maximum is reached at $t = 2.5s$. The maximum gradient is $\dot{n} = 3455^{rpm}/_{s}$. From now on $\Delta U = 0V$ and $TH_1 = TH_2 = 0.1V$ are defined as default values for the phase controller. A simulation of the phase controller configured with the default values and the standard speed trace is shown in figure 4.22. The start condition for the phase control is $n > 800rpm$. For reasons of clarity the envelope of $U_{ph}(t)$ is plotted (in red). It can be seen that $\hat{U}_{ph} \approx V_{BA}$ and the output current $I_{gen} = 0A$ for the whole trace. Obviously, reducing $V_{ref} < V_{ba}$ is not necessary for the standard speed trace. In section 4.4.3 simulations with increased speed gradient are presented.

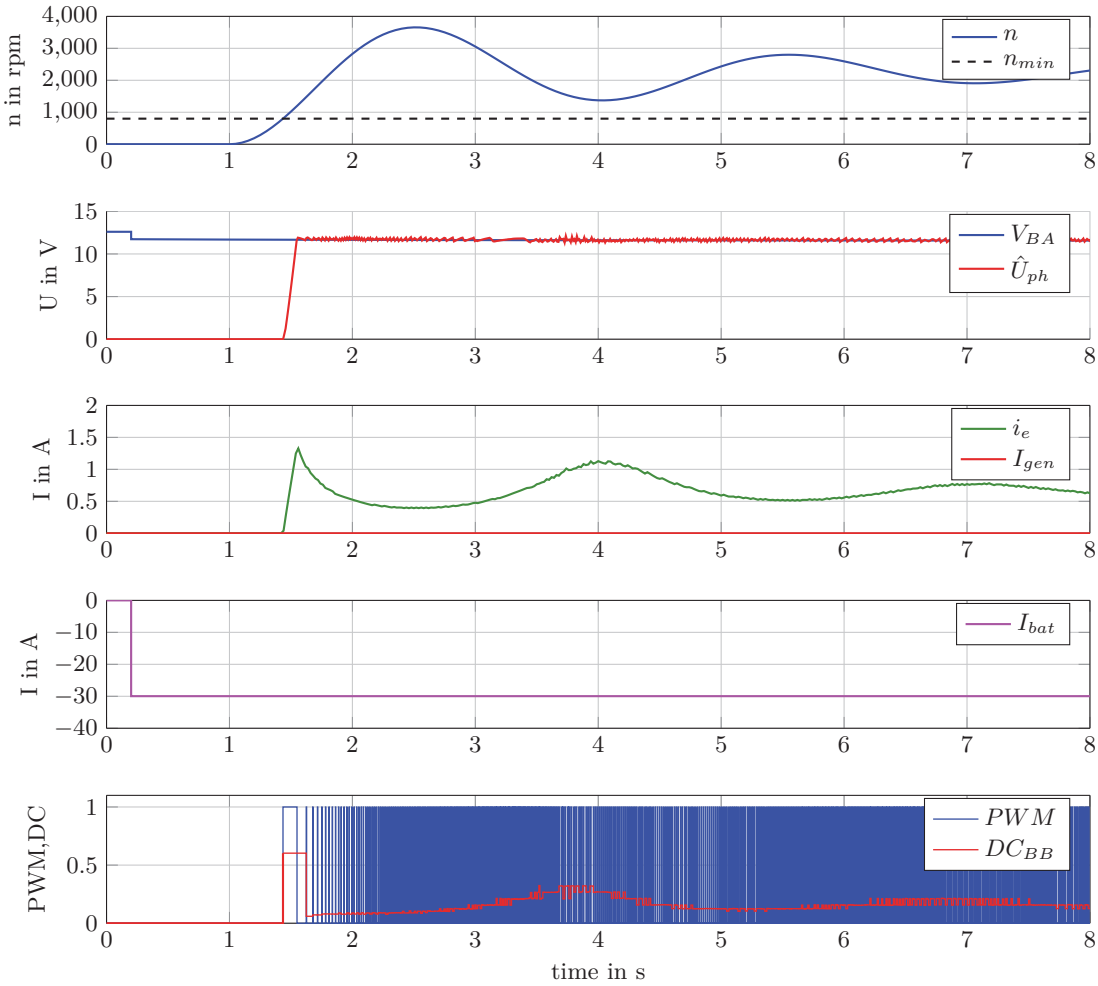


Figure 4.22: Simulation standard speed trace, $\Delta U = 0V,\ TH = 0.1V$

4.5 Duty Cycle Determination and Handover

The DC of the bang-bang PWM is determined by function-block 2) (see figure 4.6). Two different algorithms are implemented to determine the recent value. The final Simulink implementation is given in the appendix 7.

Counter implementation

The rising edge and the falling edge of the PWM are detected. Actually, the output of the comparators is a boolean signal. The edge detection is done by observing the transition "0" $\rightarrow$ "1" or rather "1" $\rightarrow$ "0". At the falling edge the current counter value cnt is read out and saved. At the rising edge the DC is calculated and the counter is reset. This is described by equations 4.139-4.142:

$$cnt = cnt + 1 \ @\_f_{CLK} \tag{4.139}$$

$$cnt_H = cnt \ @\_PWM \tag{4.140}$$

$$DC = \frac{cnt_H}{cnt} \ @\_PWM \tag{4.141}$$

$$cnt = 0 \ @\_PWM \tag{4.142}$$

Hence the PWM frequency is relatively low, a clock frequency of $f_{CLK} = 22kHz$ is sufficient (see table 4.7). Also, two DC averaging methods are implemented. One is a simple moving average:

$$\begin{aligned} DC_{MAVG}(i) &= \frac{1}{i}\sum_{1}^{i} DC(i) \quad \forall \quad 1 \leq i < N_{avg} \\ DC_{MAVG}(i) &= \frac{1}{N_{avg}}\sum_{i-N_{avg}+1}^{i} DC(i) \quad \forall \quad i \geq N_{avg} \end{aligned} \tag{4.143}$$

i is the index of the current bang-bang period, N_{avg} is the number of points, which are used to compute the average value. The other filter is an exponential weighted moving average:

$$\begin{aligned} DC_{EWMA}(i) &= DC_{MAVG}(i) \quad \forall \quad 1 \leq i < 2 \\ DC_{EWMA}(i) &= \alpha DC(i) + (1-\alpha) DC_{EWMA}(i-1) \quad \forall \quad i \geq 2 \end{aligned} \tag{4.144}$$

α is the so called smoothing factor $\alpha \in [0\ 1]$. The higher α, the higher is the impact of the current sample. Figure 4.23 shows a simulation of the implemented DC detection and the averaging algorithms (Block 2) in figure 4.6). As test input, two alternating PWM signals are used. Note, both algorithms are sample based. Since the bang-bang PWM frequency varies, the DC samples are non equidistant.

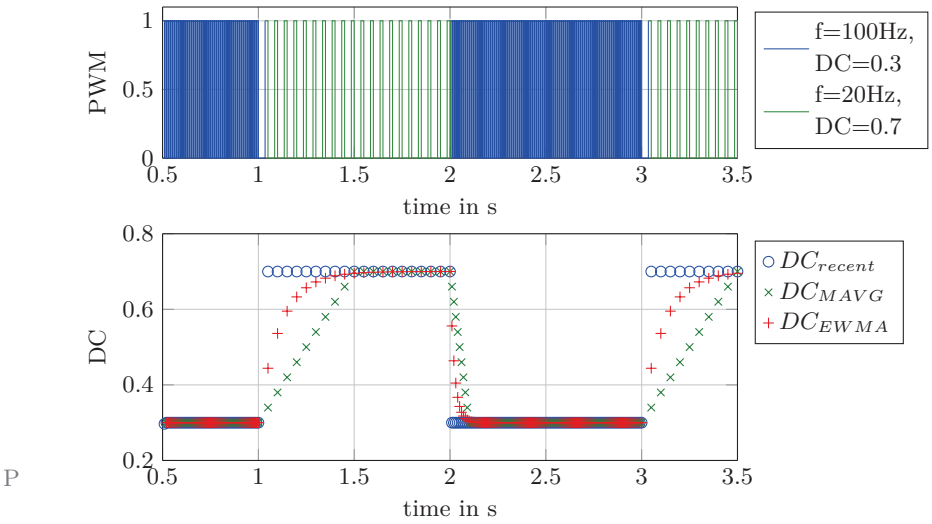


Figure 4.23: Test and comparison of the DC determination, $N_{avg} = 10, \alpha = 0.36$

Excitation current measurement

Instead of determining the DC of the PWM, the (static) DC is calculated by (see equation 4.81): .

$$DC_{ie} = \frac{R_e \bar{i}_e}{V_{bat}} \tag{4.145}$$

The excitation current measurement is used to determine the current of each PWM period $\bar{i}_e$. Equation 4.145 works if the excitation resistor is known. In case it is not, R_e can be (online) estimated by the current measurement. Figure 4.24 shows the principle field current trace. During the PWM off-time the excitation current is measured by the IC (free wheeling path). It is assumed that a current sample at the beginning of the off-time t_{FE} and at the end t_{RE1} or rather t_{RE2} is given $\Rightarrow$ $i_e(t_{RE1})$, $i_e(t_{FE})$, $i_e(t_{RE2})$ measured. The current mean value of a period is defined by:

$$\bar{i}_e := \frac{i_e(t_{RE1}) + i_e(t_{FE}) + i_e(t_{RE2})}{3} \tag{4.146}$$

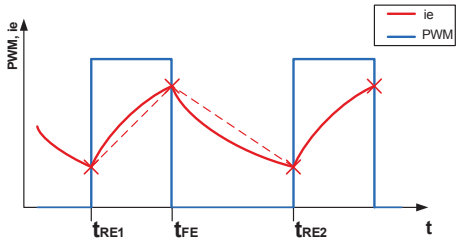


Figure 4.24: Principle excitation current trace

During the on-time the current trace is expressed by equation 4.124. With the boundary values $i_e(t_{FE})$ and $i_e(t_{RE1})$ it follows:

$$i_e(t_{FE}) = \frac{V_{bat}}{R_e}\left(1 - e^{-\frac{t_{ON}}{\tau_e}}\right) + i_e(t_{RE1})e^{-\frac{t_{ON}}{\tau_e}} \tag{4.147}$$

$$t_{ON} = t_{FE} - t_{RE1} = cnt_H T_{CLK} \tag{4.148}$$

During the off-time $i_e(t)$ can be expressed with the last term of equation 4.124. With the boundary values $i_e(t_{FE})$ and $i_e(t_{RE2})$ it follows:

$$i_e(t_{RE2}) = i_e(t_{FE})e^{-\frac{t_{OFF}}{\tau_e}} \tag{4.149}$$

$$t_{OFF} = t_{RE2} - t_{FE} = (cnt@\llcorner PWM - cnt_H)T_{CLK} \tag{4.150}$$

Equations 4.147 and 4.149 give:

$$\tau_e = -\frac{t_{OFF}}{ln\left(\frac{i_e(t_{RE2})}{i_e(t_{FE})}\right)} \tag{4.151}$$

$$R_e = V_{bat}\frac{1 - e^{-\frac{t_{ON}}{\tau_e}}}{i_e(t_{FE}) - i_e(t_{RE1})e^{-\frac{t_{ON}}{\tau_e}}} \tag{4.152}$$

Exponential and logarithms operations are hard to implement in the IC. A Taylor series can be used to approximate the exponential terms in equations 4.147 and 4.149 (see [2] on page 601):

$$e^{-\frac{\Delta t}{\tau_e}} = 1 - \frac{\Delta t}{\tau_e} + \frac{1}{2!}\left(\frac{\Delta t}{\tau_e}\right)^2 + ... \tag{4.153}$$

Neglecting terms with a degree higher than one (linearisation), gives:

$$i_e(t_{FE}) = \frac{V_{bat}}{\tilde{R}_e}\left(1 - 1 + \frac{t_{ON}}{\tilde{\tau}_e}\right) + i_e(t_{RE})\left(1 - \frac{t_{ON}}{\tau_e}\right) \tag{4.154}$$

$$i_e(t_{RE2}) = i_e(t_{FE})\left(1 - \frac{t_{OFF}}{\tilde{\tau}_e}\right) \tag{4.155}$$

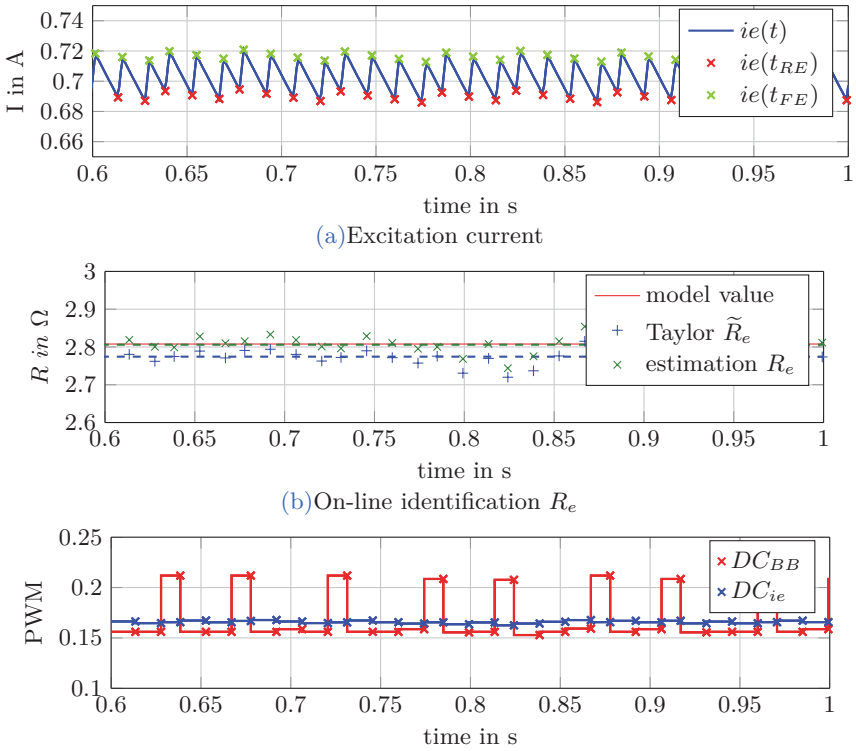
Replacing $\tilde{\tau}_e = \frac{\tilde{L}_e}{\tilde{R}_e}$ and rearranging leads to:

$$\widetilde{R}_e = V_{bat}\frac{\frac{\Delta i_{e_F}}{t_{OFF}}}{\frac{\Delta i_{e_F}}{t_{OFF}} i_e(t_{RE1}) - \frac{\Delta i_{e_R}}{t_{ON}} i_e(t_{FE})} \tag{4.156}$$

$$\Delta i_{e_F} = i_e(t_{RE2}) - i_e(t_{FE}) \tag{4.157}$$

$$\Delta i_{e_R} = i_e(t_{FE}) - i_e(t_{RE1}) \tag{4.158}$$

Figure 4.25 shows a simulation at constant speed. Both the determined DC_{BB} of the PWM and the calculated DC_{ie} using equations 4.146 and 4.145 are plotted, as well the field current $i_e(t)$ and the on-line identification of R_e. The mean value of the estimated R_e by



(a)Excitation current

(b)On-line identification R_e

(c)Unfiltered DC trace

Figure 4.25: Static DC determination by current measurement and R_e identification

Taylor $\widetilde{\bar{R}}_e = 2.77\Omega$ (equation 4.156) is slightly lower than the mean value by equation 4.152 $\bar{R}_e = 2.806\Omega$. Nevertheless, the relative error referred to the real value (model value) 2.8Ω is less than one percent.
DC_{ie} is calculated with the current values of $\widetilde{R}_e$ (equation 4.145). It can be seen that the trace of DC_{ie} is much smoother than DC_{BB}. The jitter of DC_{BB} is naturally low pass filtered by the excitation RL circuit. Notwithstanding, some averaging of the estimated $\tilde{R}_e$ could be done.

On the other hand, the value of Δi_{e_F} or rather Δi_{eR} is low, about $30mA$. The accuracy of the excitation current measurement is given with $250mA$, see [4]. As a consequence, the on-line identification of R_e is not feasible with the current IC. Table 4.9 shows the minimum Δi_{e_F} or rather Δi_{eR} at several speed values. Nevertheless, the current mean value of a period $\bar{i}_e$ can be roughly calculated by equation 4.146 since the absolute values are used. If R_e is known, the approach calculating DC_{ie} by measuring the excitation current still works. In that case, the end-user or the alternator manufacturer needs to choose a value for R_e (maybe via programming the NVM). Another possibility could be that the current measurement of the IC will be improved. Moreover, real measurements at low PWM frequencies have shown, that Δi_e is larger than simulated (see figure 5.10). For that reasons, this approach is still considered as an option in this thesis.

4.5.1 Constant rotor speed

Determination of the handover value by filtering DC_{BB}

The theoretical calculation of the static DC $DC(n) = DC(n, \dot{n} = 0)$ is given in section 4.3.3.

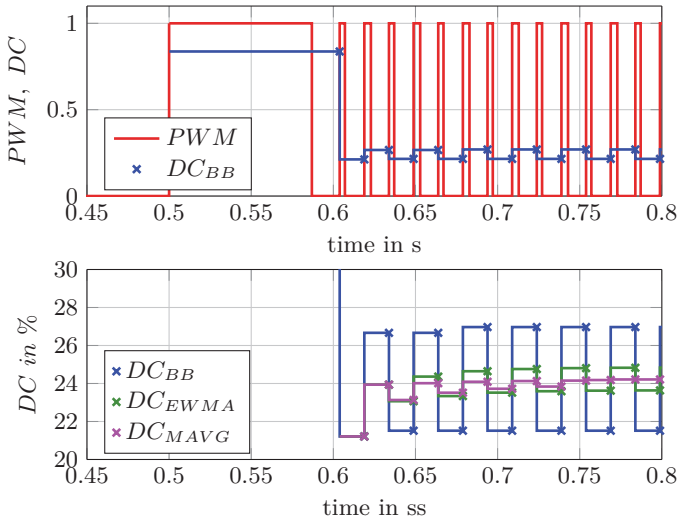


Figure 4.26: Simulation $n = 2100rpm, \Delta U = 0V, TH = 0.1V$. DC jitter caused by mistimed sampling

As mentioned before, the DC-trace of the bang bang controller DC_{BB}(i) jitters, see figure 4.26. Several simulations with different speed values have been made. Minimum, maximum, expected value (mean value) and standard deviation s_{BB} are given in table 4.6. As well $DC(n)$ for $V_{ref} = V_{bat}$ (based on the numerical calculation). The very first period (long pulse at the beginning, index i=0) is ignored for all filtering calculations (see figure 4.27).

The cause of the jitter is mistimed sampling of $\hat{U}_{ph}$. With the comparators $\hat{U}_{ph} < V_{ref} - TH_2$ and $U_{ph} > V_{ref} + TH_1$ is detected. The sampling time is the electrical period T_{EL}, which is given by the rotor speed. In case of a constant speed $\hat{U}_{ph}$ appears periodic, but the amplitude value depends on the field current. The time span till $\hat{U}_{ph}$ reaches exactly the

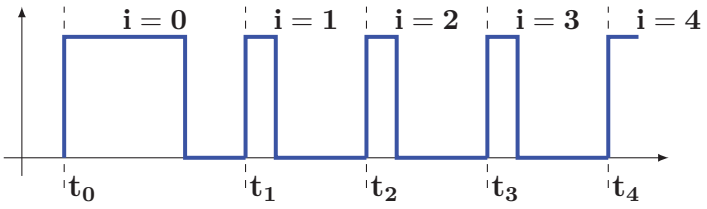


Figure 4.27: Index definition of the PWM periods

n rpm	DC_n %	$\bar{DC}_{BB}(i)$ %	s_{BB} %	$max(DC_{BB})$ %	$min(DC_{BB})$ %
1000	43.8	44.1	6.3	54.4	40.3
1500	24.4	24.1	2.8	27.0	21.2
2100	16.9	17.4	2.7	21.1	15.6
2500	14.1	14.1	1.8	15.9	12.2
3000	11.6	11.7	1.0	12.4	10.3
4000	8.7	8.8	0.3	9.0	7.9
6000	5.7	5.8	0.3	6.2	5.6
8000	4.3	4.3	0.3	4.5	3.9

Table 4.6: Analysis of the DC jitter, $V_{ba} = V_{bat} = 11.7V$

upper threshold, or rather the time span till $\hat{U}_{ph}$ is lower than $\hat{U}_{ph} < V_{ref} - TH_2$, depends on the rotor parameters. Mistimed sampling occurs if the needed time span is not a multiple integer of the electrical period. As a consequence, the number of electrical periods during a PWM period varies. This is illustrated in figure 4.28. Table 4.7 gives an overview of the PWM frequency variation at each rotor speed. As mentioned in section 4.1, the PWM frequency also depends on the rotor speed.

Due to the jitter, averaging is necessary in order to determine a proper handover value. However, each PWM period is an additional delay time. A trade off between accuracy and averaging time is needed. For moving average as well as exponential smoothing the averaging time depends on the used number of PWM periods. Table 4.7 shows also the time span from the second till the fourth $t_4 - t_1$, sixth $t_6 - t_1$and eleventh $t_{11} - t_1$ period. The theoretical $DC(n)$ and the (handover) value of each filtering algorithms at t_4 (three samples), t_6 (five samples) and t_{11} (ten samples) is given in table 4.8. The absolute error is defined as:

$$e\,(n)|_{t_i} := DC\,(n) - DC_{HO}|_{t_i} \tag{4.159}$$

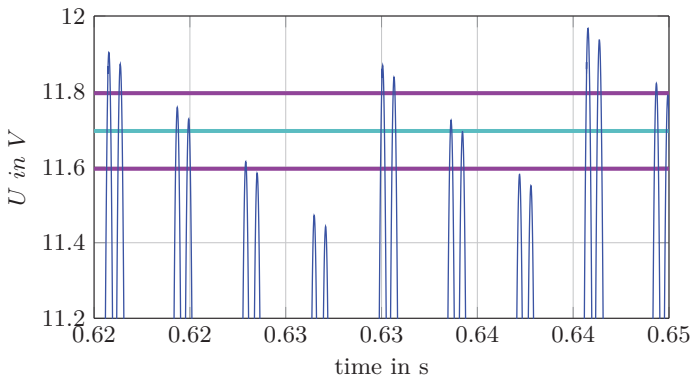


Figure 4.28: Zoom simulation $n = 2100rpm$, mistimed sampling

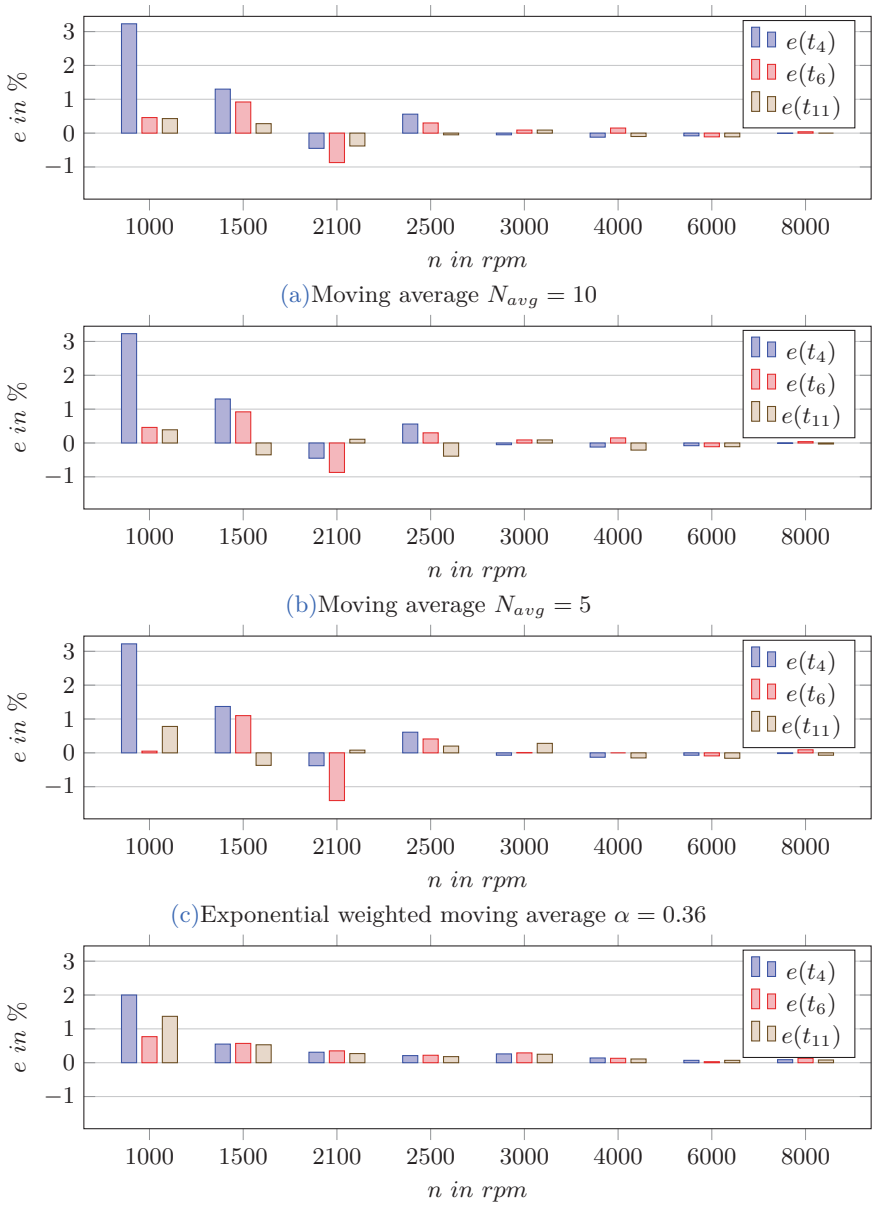
n rpm	f_{EL} Hz	$\bar{f}_{PWM}$ Hz	$max(f_{PWM})$ Hz	$min(f_{PWM})$ Hz	s_f Hz	$t_4 - t_1$ ms	$t_6 - t_1$ ms	$t_{11} - t_1$ ms
1000	133.3	36.4	44.4	33.3	5.1	60	112.5	255
1500	200	66.7	66.7	66.7	0	30	60	135
2100	280	76.3	93.22	69.8	10.8	28.6	53.6	121.5
2500	333.3	75.7	83.33	66.9	8.5	27	54	120
3000	400	76.0	80	66.7	6.2	25	52.5	120
4000	533.3	75.3	76.4	66.7	2.9	26.2	54.4	120
6000	800	74.7	80	72.6	3.4	27.5	53.7	120
8000	1066.6	75.3	82.1	71.2	3.2	26.2	52.5	120

Table 4.7: Analysis of the PWM frequency, $V_{ba} = V_{bat} = 11.7V$

		DC_{MAVG} $N_{avg} = 10$			DC_{MAVG} $N_{avg} = 5$			DC_{EWMA} $\alpha = 0.36$		
n rpm	$DC(n)$ %	$@t_4$ %	$@t_6$ %	$@t_{11}$ %	$@t_4$ %	$@t_6$ %	$@t_{11}$ %	$@t_4$ %	$@t_6$ %	$@t_{11}$ %
1000	43.7	40.5	43.3	43.3	40.5	43.3	43.4	40.5	43.7	43.0
1500	24.4	23.1	23.5	24.1	23.1	23.5	24.8	23.1	23.3	24.8
2100	16.9	17.4	17.8	17.3	17.3	17.8	16.8	17.3	18.3	16.8
2500	14.1	13.5	13.8	14.1	13.5	13.8	14.4	13.4	13.6	13.9
3000	11.6	11.7	11.5	11.5	11.7	11.5	11.5	11.7	11.6	11.3
4000	8.7	8.8	8.6	8.6	8.8	8.6	8.7	8.8	8.7	8.8
6000	5.7	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.8	5.9
8000	4.3	4.3	4.2	4.3	4.3	4.3	4.3	4.3	4.2	4.2

Table 4.8: Comparison of the filtering algorithms

and plotted in figure 4.29 for $t_4,\ t_6,\ t_{11}$. Note that the unit of the absolute error e is %, because $DC(n)$ and DC_{HO} are given in %. It can be seen, that all averaging algorithms deliver a proper handover value. Only a very low speed (1000rpm) and using just three samples gives a relatively worse result (blue). The absolute error is about 3%. The point is that the very first sample $DC_{BB}(i = 1)$ is in general too low. That can be seen clearly in figure 4.26. The outcome is that the five sample method (handover at t_6) is the best trade-off. In average, it takes $60ms$ (see table 4.6 column $t_6 - t_1$) and the error of the handover values is about $\pm 1\%$ or less (see figure 4.29 red bar).
As long as the current number of PWM periods is lower then five, both moving average algorithms are identical. However, even at t_{11} (ten samples) there is almost no difference, see figure 4.29a and 4.29b. The exponential smoothing uses a value of $\alpha = 0.36$. Actually, this value was found by optimization for dynamic speed traces. Nevertheless, the results for constant speed are almost as good as the (normal) moving average algorithm (see figure 4.29c).



(a)Moving average $N_{avg} = 10$

(b)Moving average $N_{avg} = 5$

(c)Exponential weighted moving average $\alpha = 0.36$

(d)Approach excitation current measurement

Figure 4.29: Absolute error n constant: $e = DC(n, \dot{n} = 0) - DC_{HO}(i)$ in %

Determination of the handover by measuring i_e

The same simulations are done (actually at once). The field current of each period $\bar{i}_e(i)$ is on-line calculated by equation 4.146. The Taylor approximation is used to estimate $\widetilde{R}_e(i)$, see equation 4.156 .The same moving average algorithms as for the DC_{BB} filtering is used to smooth $\widetilde{R}_e(i)$, $N_{avg} = 10$. $\bar{DC}_{ie}(i)$ is calculated by equation 4.145, using the averaged $\bar{\widetilde{R}}_e(i)$ and $\bar{i}_e(i)$. Table 4.9 gives the theoretical $DC(n)$ and $\bar{DC}_{ie}$ at $t_4,\ t_6,\ t_{11}$, as well as the minimum and average of Δi_{e_F} or rather Δi_{e_R}. The absolute error e at $t_4,\ t_6,\ t_{11}$ is plotted in figure 4.29d.

		$\bar{DC}_{ie}(i)$						
n	$DC(n)$	@t_4	@t_6	@t_{11}	$min(\lvert\Delta i_{e_F}\rvert)$	$mean(\lvert\Delta i_{e_F}\rvert)$	$min(\Delta i_{e_R})$	$mean(\Delta i_{e_R})$
rpm	%	%	%	%	mA	mA	mA	mA
1000	43.7	41.8	43.0	42.4	100	175.9	176.1	233.5
1500	24.4	23.9	23.9	23.9	40.8	42.7	37.2	59.1
2100	16.9	16.6	16.5	16.6	20	26.6	26.2	36.5
2500	14.1	13.8	13.8	13.9	13.8	22.3	22.2	30.3
3000	11.6	11.4	11.3	11.4	12.7	18.8	18.7	25.4
4000	8.7	8.5	8.5	8.5	14.0	14.4	14.7	19.4
6000	5.7	5.7	5.6	5.7	9.0	9.7	9.63	13.0
8000	4.3	4.2	4.2	4.2	5.0	7.3	7.3	9.7

Table 4.9: DC determination by current measurement

Temperature influences

With rising temperature, the excitation resistor increases. Since the flux is independent of temperature the needed field current stays the same. As a consequence, the DC must increase because of R_e (see equation 4.81). Notwithstanding, it is no issue because the phase controller is a closed loop controller. Furthermore, there is no effect to DC detection or rather to the averaging algorithm. Also the on-line identification gives the recent value of R_e. Subsequently, the calculated DC by equation 4.145 is correct. In case that a programmed NVM value is used for R_e, a temperature compensation may be needed.

Eddy current influences

The direct component of the field current is essential for the induced voltage. The phase controller adjusts it by regulating the DC of the PWM. At steady-state, eddy currents do not influence the DC, because they act only on the alternating field component $\dot{\Psi}_s,\ \dot{\Psi}_e$. It is assumed that the magnetization of the iron is linear. In fact, eddy current influence the

shape of the excitation current. The Δi_{e_R} or rather the Δi_{e_F} is increased. If saturation is considered it might be possible that a higher (average) current is needed. As result the DC increases.
Notwithstanding, for DC detection and for the handover, it has no effect. Measurements with a fixed DC value but variable PWM frequency have shown that the average excitation current stays the same, see section 5.2.1. On the other hand, at the beginning of the ECC the excitation current needs to be raised (from zero). Eddy currents oppose the rising field. As a result the duration of the first pulse will be increased. To sum up, the time span till the handover can be done increases, but there seems to be no issue to handover the determined value into the $220Hz$ system.

Another point is the on-line identification of R_e. The calculation assumes a single RL-circuit. As mentioned before, the shape of i_e is influenced by eddy currents. In figure 5.10 it can be seen that the trace can not be described with single time-constant. As a result the on-line estimation may give improper values. Nevertheless, the approach of measuring the excitation current and calculating the static DC still works if R_e is known.

4.5.2 Dynamic rotor speed

The calculation of the static DC is based on $\dot{i_e} = \dot{\Psi}_e = 0$, which is true for a constant rotor speed. However, in case of a dynamic speed trace i_e must be raised or decreased depending on the current speed. This is seen at the i_e-trace in figure 4.22. Consequently, $\dot{\Psi}_e \neq 0$ and the dynamic DC is given by:

$$DC(\omega, \dot{\omega}) = \frac{u_e}{V_{bat}} = \frac{R_e i_e}{V_{bat}} + \frac{\dot{\Psi}_e}{V_{bat}} \tag{4.160}$$

With equation 4.81 it follows:

$$DC(\omega, \dot{\omega}) = DC(\omega, \dot{\omega} = 0) + \frac{\dot{\Psi}_e}{V_{bat}} \tag{4.161}$$

The dynamic DC is the static DC plus the term $\frac{\dot{\Psi}_e}{V_{bat}}$. In case of $\dot{\omega} = 0$ (constant speed) the dynamic DC is the static DC. Ψ_e is given by equation 4.16, replacing the term $Mi_e = \Psi_{sd}$ (equation 4.38) leads to:

$$\Psi_e = L_{\sigma_e} i_e + \frac{3}{2}\frac{1}{a}\Psi_{sd} \tag{4.162}$$

The derivative gives:

$$\dot{\Psi}_e = L_{\sigma_e} \dot{i_e} + \frac{3}{2}\frac{1}{a}\dot{\Psi}_{sd} \tag{4.163}$$

At first, the calculation of $\dot{\Psi}_e$ is done analytically $M = const$ and afterwards numerically $M = M(i_e)$

Analytical calculation:
$\dot{i_e}$ replaced by equation 4.89 leads to:

$$\dot{\Psi}_e = L_{\sigma_e} \frac{\dot{\Psi}_{sd}}{M} + \frac{3}{2}\frac{1}{a}\dot{\Psi}_{sd} = \dot{\Psi}_{sd}\left(\frac{L\sigma_e}{M} + \frac{3}{2}\frac{1}{a}\right) \tag{4.164}$$

Numerical calculation:
$\dot{i_e}$ replaced by equation 4.98, gives:

$$\dot{\Psi}_e = \dot{\Psi}_{sd}\left(\frac{L_{\sigma_e}}{M_{diff}(i_e)} + \frac{3}{2}\frac{1}{a}\right) \tag{4.165}$$

The upper plot of figure 4.30 shows the $DC(n, \dot{n})$ for different speed gradients. The lower plot shows $\dot{\Psi}_{sd}(n, \dot{n})$ and the feasible flux change (dashed). It can be seen that the calculated DC is negative at low revolutions, if the speed gradient is above $4000^{rpm}/_s$. The problem is the feasibility of decreasing the stator flux. $\dot{\Psi}_{sd}(\dot{\omega} = 4000^{rpm}/_s)$ is lower than the feasible minimum. The same problem occurs for the speed gradient of $-6000^{rpm}/_s$. The feasible flux increase is too low at $1000rpm$. Thus, a non feasible DC over 100% is determined. However, for the standard dynamic speed trace it is no problem. A simulation (default configuration) is shown in figure 4.31. The red crosses are the unfiltered $DC_{BB}(i)$ values of the bang-bang controller. Also the calculated static $DC(n)$ and the dynamic $DC(n, \dot{n})$ are plotted. It can be seen that $DC_{BB}(i)$ follows the dynamic $DC(n, \dot{n})$. Notwithstanding, both strategies for determination of the static DC and of the dynamic DC were developed. In section 6.2.2 it will be discussed which handover value gives better results.

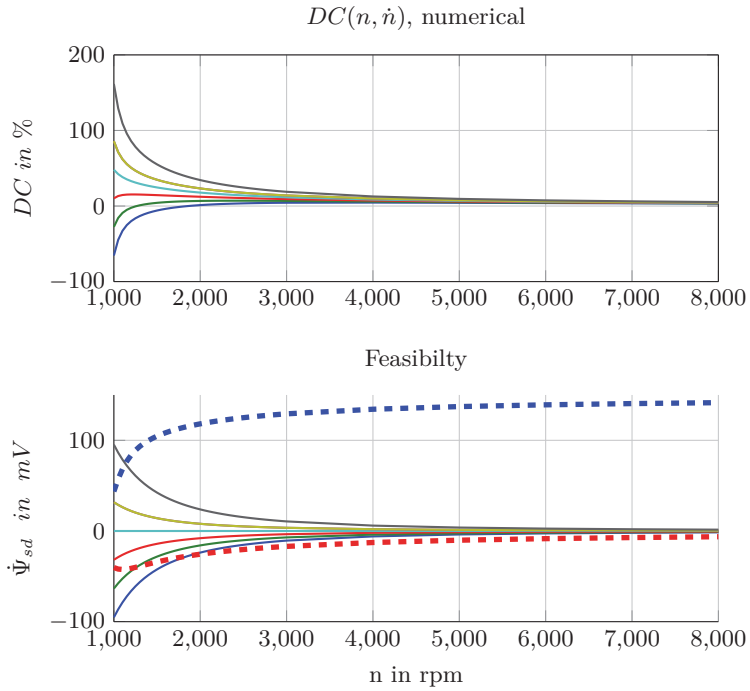


Figure 4.30: Dynamic $DC(n, \dot{n})$ and $\dot{\Psi}(n, \dot{n})$ — $\dot{n} = 6000^{rpms}/_s$, — $\dot{n} = 4000^{rpms}/_s$, — $\dot{n} = 2000^{rpms}/_s$, — $\dot{n} = -6000^{rpms}/_s$, — $\dot{n} = -4000^{rpms}/_s$, — $\dot{n} = -2000^{rpms}/_s$, — $\dot{n} = 0^{rpms}/_s$, ▪▪▪ $\dot{\Psi}_{feasible,max}$, ▪▪▪ $\dot{\Psi}_{feasible,min}$

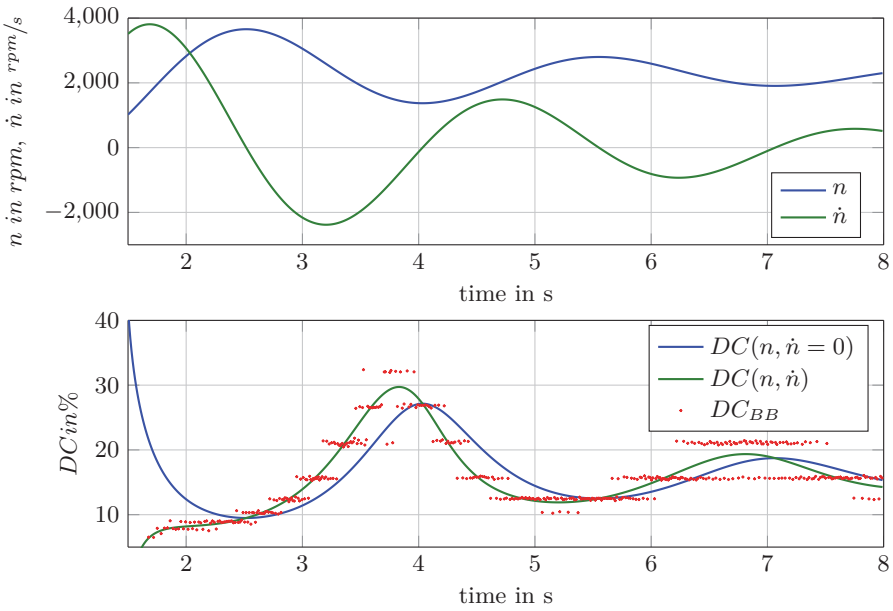


Figure 4.31: DC traces standard dynamic speed trace

Strategy I: determine static DC

Basically, it is uncertain how the speed trace continues after the handover to the V_{ba}-controller. Thus, the DC value according to the current speed is transferred:

$$DC_{HO} \stackrel{!}{=} DC(n, \dot{n} = 0) \tag{4.166}$$

Two options are implemented. The first one is an intuitive approach: weight the $DC_{BB}(i)$ samples with a factor proportional to the current speed gradient. The other option is to calculate the static DC by measuring the excitation current as it is done for the constant speed.

Weighted sample:

$$\widetilde{DC}_{BB}(i) = w(i) DC_{BB}(i) \tag{4.167}$$

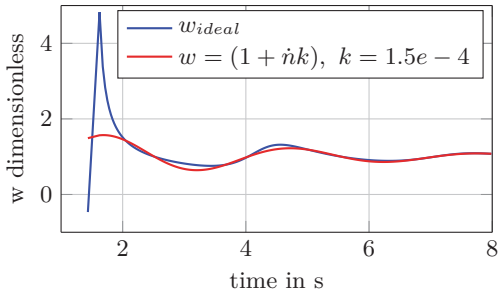
$$w(i) \sim \left.\frac{dn}{dt}\right|_i \tag{4.168}$$

For example $w(i)$ could be formulated by:

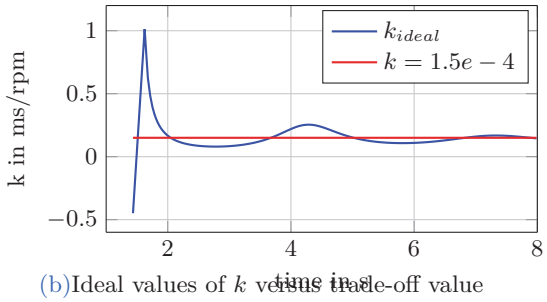
$$w(i) = 1 + k \frac{n(i) - n(i-1)}{T_{PWM}(i)} \tag{4.169}$$

where k is an adjustable value. To determine a proper value for k, the ideal weight factor $w_{ideal}(i)$ is calculated by replacing $DC_{BB}(i)$ with the theoretical value $DC(n|_i, \dot{n}|_i)$ and $\widetilde{DC}_{BB}(i)$ by $DC(n|_i)$:

$$w_{ideal}(i) = \frac{DC(n|_i)}{DC(n|_i, \dot{n}|_i)} \tag{4.170}$$



(a)Ideal weight factors versus trade off-factors



(b)Ideal values of k versus trade-off value

Figure 4.32: Approach weighted samples

Figure 4.32a shows the ratio ($=w_{ideal}$) static DC to dynamic DC over time for the standard speed trace (in blue). As the next step, the ideal values $k_{ideal}(i)$ are calculated by rearranging equation 4.169:

$$k_{ideal}(i) = \frac{w_{ideal}(i) - 1}{\dot{n}|_i} s/rpm \tag{4.171}$$

The unit of k is s/rpm, because the weight factor w is dimensionless. Since a single value is required the mean value of $k_{ideal}(i)$ could be determined:

$$k = \bar{k}_{ideal} = 1.44 \cdot 10^{-4} s/rpm \tag{4.172}$$

Another possibility is to use the least-square method to solve the overdetermined equation:

$$k = \dot{n}^{+}(w_{ideal} - 1) = 1.62 \cdot 10^{-4} \tag{4.173}$$

$\dot{n}^{+}$ is the pseudoinverse of $\dot{n}$. In the end, $k = 1.5 \cdot 10^{-4}$ is used. The ideal values $k_{iedal}(i)$ (blue) and the estimated k (red) are shown in figure 4.32b. The weight factors are calculated by equation 4.169 using the estimated k value, and plotted in figure 4.32a (red). Finally, the weighted DC samples $\widetilde{DC}_{BB}(i)$ are filtered using a moving average:

$$\begin{aligned} DC_{WMAVG}(i) = & \quad \frac{1}{i}\sum_{1}^{i} w(i)DC_{BB}(i) \quad \forall \quad 1 < i < N_{avg} \\ DC_{WMAVG}(i) = & \quad \frac{1}{N_{avg}}\sum_{i-N_{avg}+1}^{i} w(i)DC(i) \quad \forall \quad i \geq N_{avg} \end{aligned} \tag{4.174}$$

Hence the static DC is determined as handover value, the absolute error is defined by equation 4.159 (section 4.5.1 constant speed). Figure 4.34c shows the error using a moving average with weighted sample $WMAVG$ and the non weighted samples $MAVG$. In both cases an average length of five samples is used. In order to evaluate each algorithm and to

compare the results with different speed traces, the following quality criteria are defined:

$$J_1 := \frac{1}{i|_{t_{sim}} - i|_{t_{max}}} \sum_{m=i|_{t_{max}}}^{i|_{t_{sim}}} |e_m| \tag{4.175}$$

$$J_2 := \underset{m}{max} \left(|e_i| \right) \tag{4.176}$$

$$J_3 := \underset{m}{min} \left(|e_i| \right) \tag{4.177}$$

The first one gives the average error per sample, the other two the greatest positive and negative error of the trace. In figure 4.34c it can be seen that the error of the $MAVG$ or rather $WMAVG$ is very high (almost 20%) at the beginning of the speed trace. The reason is that the speed increases fast, thus the DC_{BB} samples are much lower than the static DC. Even weighting gives just a little improvement, because the factor k is not determined for such gradients.
However, the situation is not as critical as it looks. Start-up measurements from BMW have shown that the voltage regulation is always started after the first speed maxima. Therefore, the error at the beginning of the speed trace is not relevant at all. Since it is immaterial the calculation of J_1, J_2 and J_3 starts at t_{max}. Actually, this is also the reason why k is designed as trade off factor, see figure 4.32b.
Several simulations are conducted. The speed trace parameter Δt_{max} is varied. For each iteration, the simulation time t_{sim} is adapted in order to always compare the same section of the trace. Table 4.10 gives an overview of the variation. Figure 4.35a shows J_1 for each speed trace. J_2 and J_3 are plotted in figure 4.35b or rather 4.35c.

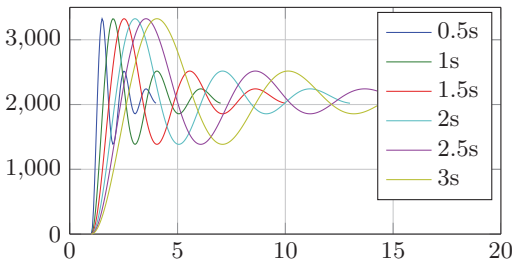
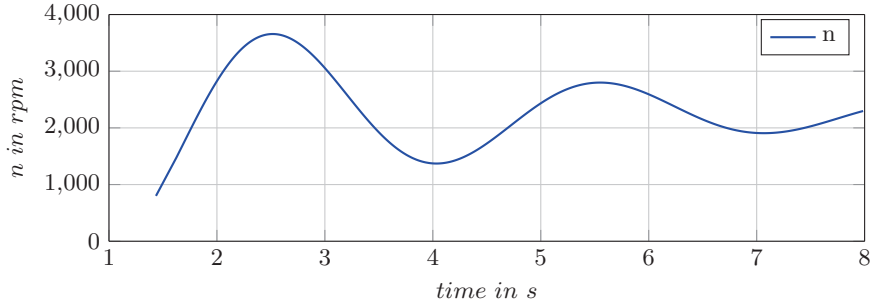


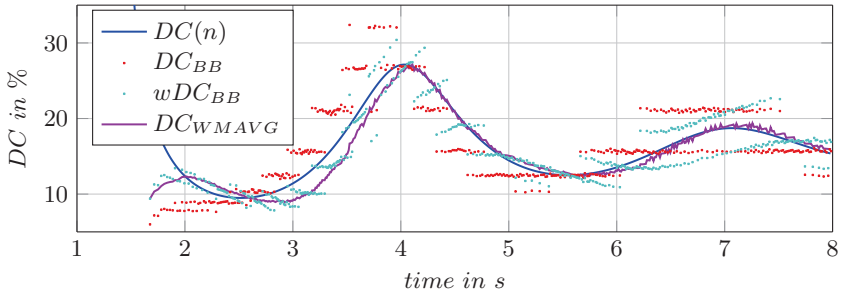
Figure 4.33: Speed traces

Δt_{max} *in s* :	0.5	1	1.5	2	2.5	3
t_{sim} *in s* :	3	6	9	12	15	18

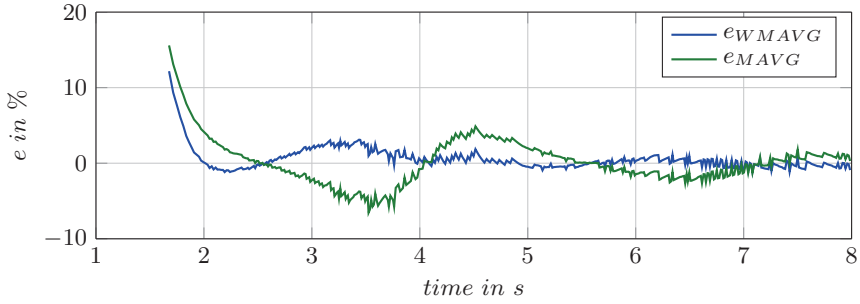
Table 4.10: Variation of Δt_{max}



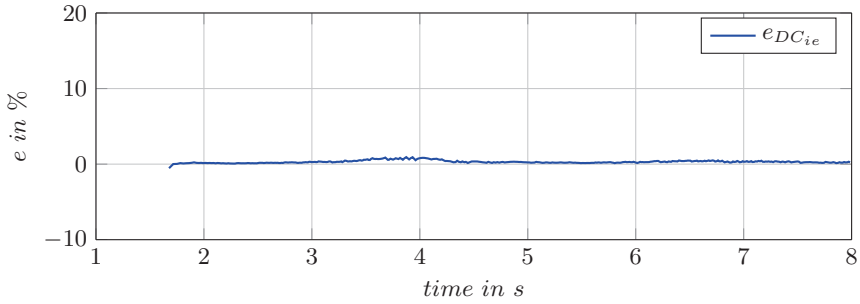
(a)Standard dynamic speed trace $\Delta t_{max} = 1.5s$



(b)Simulated DC trace

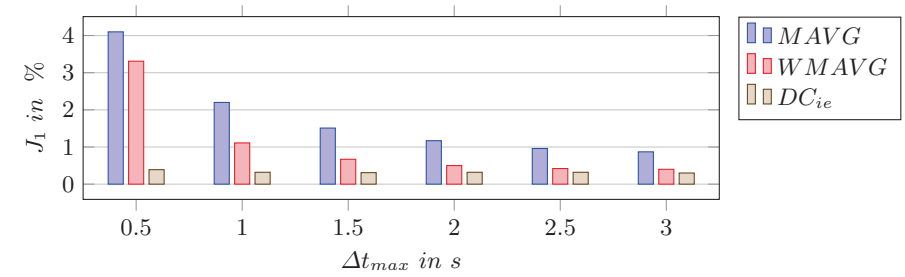


(c)Absolute error weighted and non weighted moving average $N_{avg} = 5$

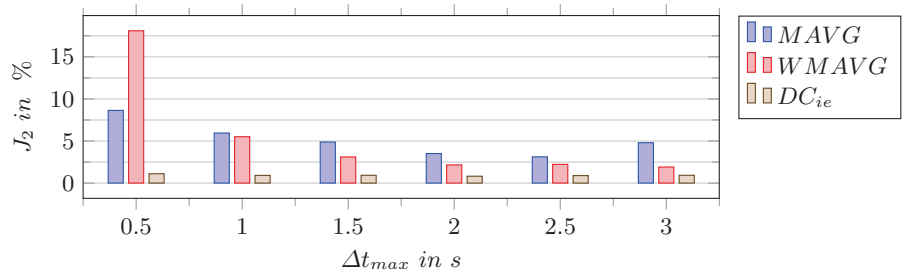


(d)Absolute error approach excitation current measured

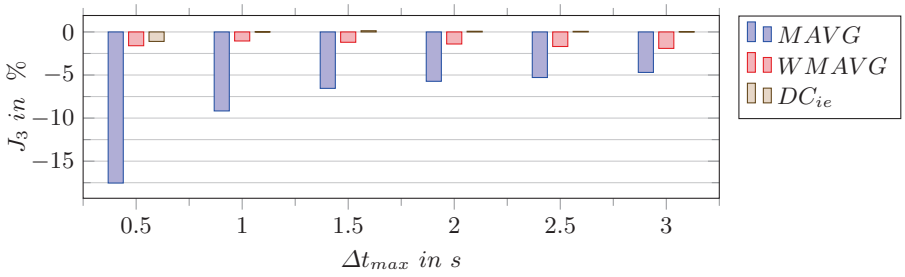
Figure 4.34: Absolute error Strategy I: $e = DC(n, \dot{n} = 0) - DC_{HO}(i)$ in %



(a)Average error per sample



(b)Greatest positive error

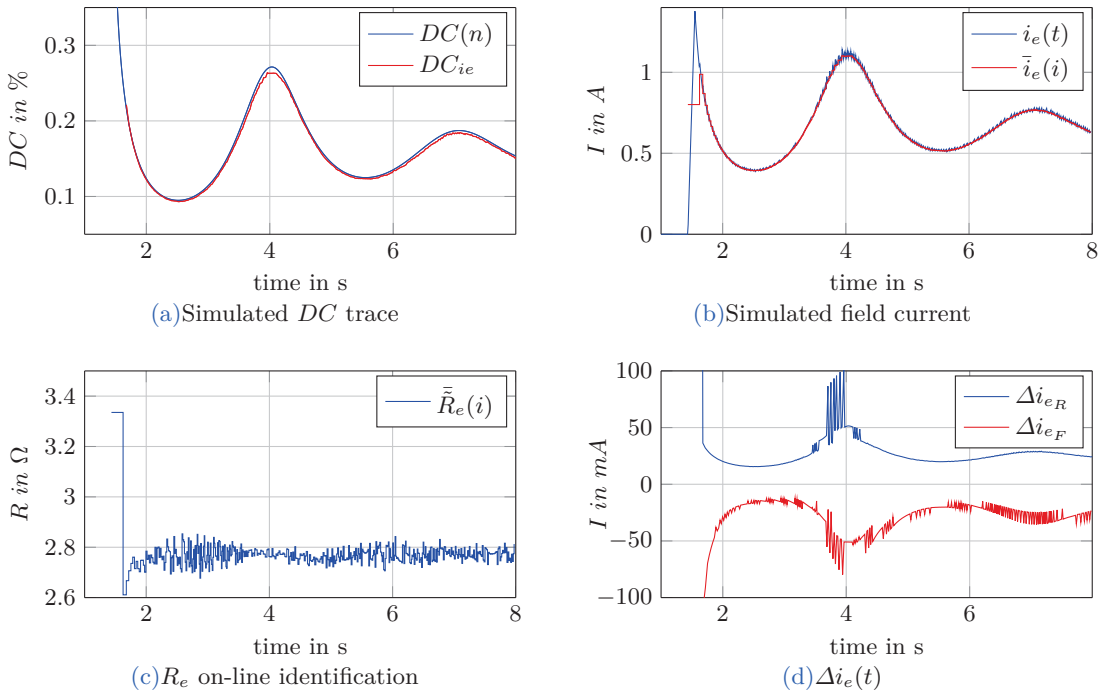


(c)Greatest negative error

Figure 4.35: Strategy I: static DC-error, dynamic speed trace

Excitation current measurment:
Same proceeding as in section 4.5.1. Figure 4.36b shows the average excitation current of each PWM period $\bar{i}_e(i)$ (see equation 4.146). The delta current values are plotted in figure 4.36d. Figure 4.36c shows the on-line identified and averaged excitation resistor $\hat{\bar{R}}_e(i)$, see equation 4.156. The theoretical $DC(n, \dot{n} = 0)$ and the on-line calculated $DC(\bar{i}e)$ by equation 4.145 are given in figure 4.36a. The absolute error is calculated by equation 4.159 and plotted in figure 4.34d. As already done before, the speed parameter Δt_{max} is varied. J_1, J_2 and J_3 of each trace are shown in figures 4.35.

Conclusion:
Obviously, the approach excitation current measurement gives, in general, much better results than the averaging methods. J_1 is less than a half percent for all speed traces. Even the greatest error is about one percent for all. Accentually, this is not a surprise. The static DC is directly calculated by measuring the needed field current. On the contrary,



(a)Simulated DC trace (b)Simulated field current

(c)R_e on-line identification (d)$\Delta i_e(t)$

Figure 4.36: Dynamic speed trace, static DC determination by field current measurement

the averaging methods use the DC_{BB} samples, which basically give the dynamic DC. Nevertheless, the average error per sample (J_1) is relatively low also for the averaging methods, except for the fastest speed trace $\Delta t_{max} = 0.5s$. Weighting the samples improves J_1 for all traces significantly, compared to the normal moving average. The greatest negative error is surprisingly low.

Strategy II: determine the dynamic DC

The basic idea is that the V_{BA} controller has to deal with the current speed gradient as well. Thus, the DC in the current situation is handed over:

$$DC_{HO} \stackrel{!}{=} DC(n, \dot{n}) \tag{4.178}$$

Indeed, it is more likely that the speed trace continues with the current speed gradient $\dot{n}$, then suddenly $\frac{dn}{dt}$. The error of strategy II is defined straight forwardly by:

$$e\,(n)|_{t_i} := DC\,(n, \dot{n}) - DC_{HO}|_{t_i} \tag{4.179}$$

Hence the phase controller deals with the dynamic speed trace, the DC_{BB} samples are the dynamic DC. Nevertheless, filtering is necessary because of the jitter (see figure 4.31). Qualified algorithms are the exponential smoothing or a moving average with a short

sample length. In fact, the exponential smoothing is meant for this purpose. The α value is found by optimization. For this purpose, the simulation data t_{max} variation is used (see 4.10). For each speed trace, the following objective function is calculated by:

$$J_{t_{max}} = \frac{1}{N} e_{t_{tmax}} e_{t_{tmax}}^T \tag{4.180}$$

where $e_{t_{tmax}}$ is the error vector of the trace Δt_{max}. The Nelder-Mead algorithm (fminsearch) is used to minimize the weighted sum of the single objectives functions:

$$J(\alpha) = \frac{1}{10} J_{0.5s} + \frac{2}{10} J_{1s} + \frac{5}{10} J_{1.5s} + \frac{2}{10} J_{2s} + \frac{1}{10} J_{2.5s} + 0 J_{3s} \tag{4.181}$$

The standard dynamic speed trace is weighted most strongly ($J_{1.5s}$), the other factors are chosen arbitrarily. It gives $\alpha = 0.36$. Figure 4.37d shows the error over time for standard trace. Also a moving average with a (fixed) sample length of $N_{avg} = 5$ is used, see figure 4.37c. Furthermore, a modified moving average algorithm is tested. The idea is to change the sample length depending on the current speed gradient:

$$N_{dyn} = \begin{cases} 3, & \text{if } \left| \frac{n(i)-n(i-1)}{T_{PWM}(i)} \right| \geq 2000^{rpm}/_{s} \\ 5, & \text{if } \left| \frac{n(i)-n(i-1)}{T_{PWM}(i)} \right| > 100^{rpm}/_{s} \\ 10, & \text{otherwise} \end{cases} \tag{4.182}$$

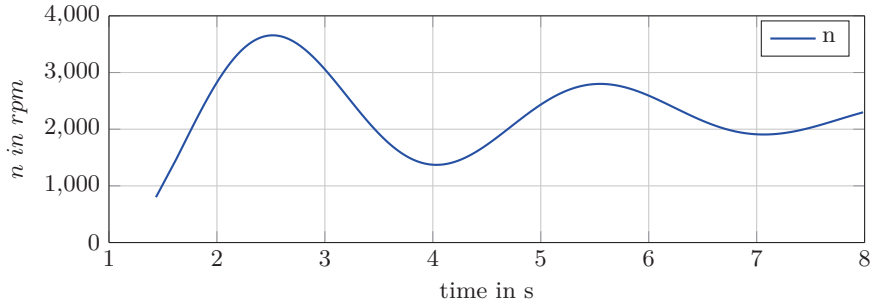
The values of N_{dyn} and the gradient regions are chosen intuitively. Hence $e_{N_{dyn}}(t)$ for the standard speed trace looks pretty much the same as $e_{N_{avag}}(t)$, it is omitted in figure 4.37.

The same criteria as for the strategy I are used to evaluate different speed traces (see section 4.5.2). The average error per sample (J_1) is shown in figure 4.38a. The greatest positive error (J_2) and negative error (J_3) of each trace is given in figure 4.38b or rather 4.38c.

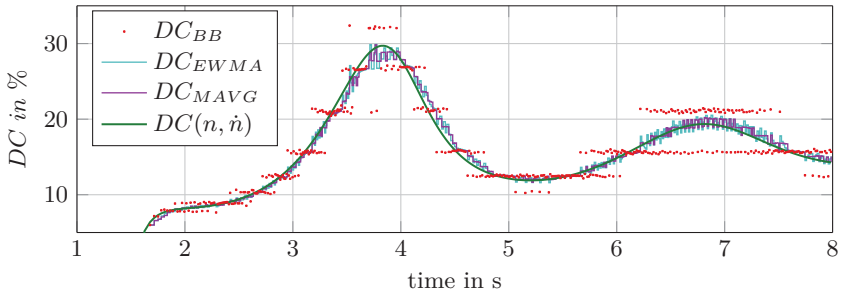
Conclusion:
The average error per sample is quite low for all algorithms and speed traces. For the fastest speed trace it is about one and a half percent, and for the rest, a half percent. Also the greatest error is kept within reasonable limits, except for $\Delta t_{max} = 0.5s$. It turned out the moving average with a fixed sample length of five, gives slightly better results than the exponential smoothing. This is kind of a surprise, considering the value of α was found by optimization while $N_{avg} = 5$ was chosen intuitively. It can be seen that the modification of a dynamic sample length gives better results for fast speed traces.

Eddy current influences

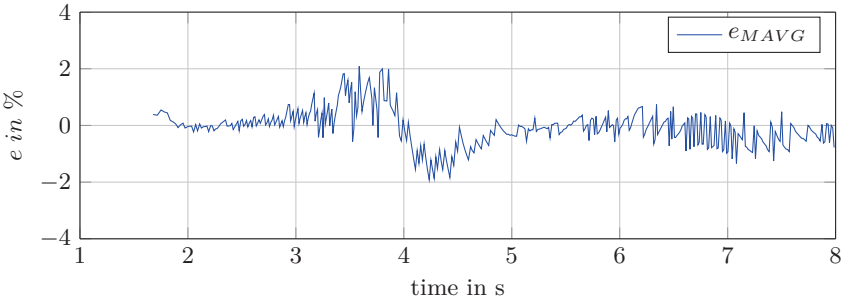
In section 5.2.2 *DC*-sweep measurements are conducted with different PWM frequencies. It can be seen that the excitation current traces are more or less the same (figure 5.12 and 5.13). Thus, it seems to be no problem to hand over the dynamic *DC* value.



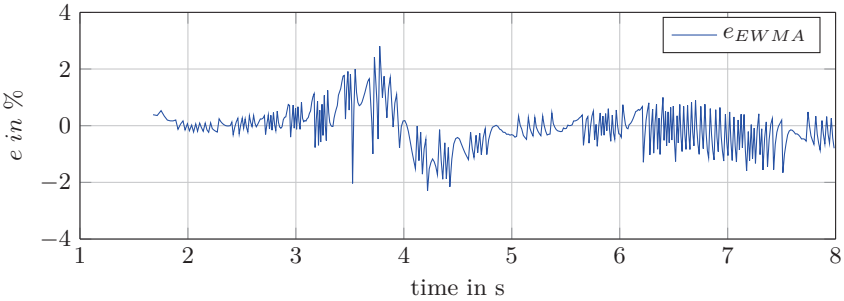
(a)Standard dynamic speed trace



(b)Simulated DC trace

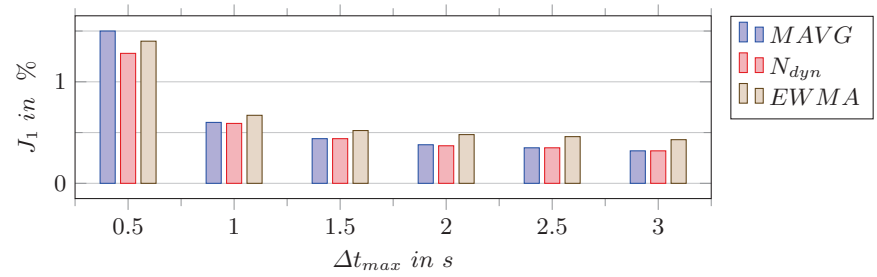


(c)Absolute error moving average $N_{avg} = 5$

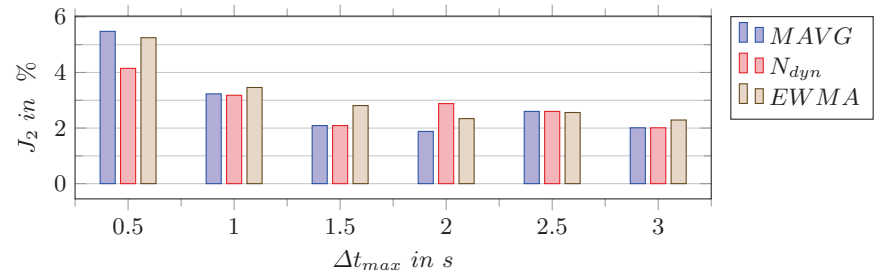


(d)Absolute error exponential weighted moving average $\alpha = 0.36$

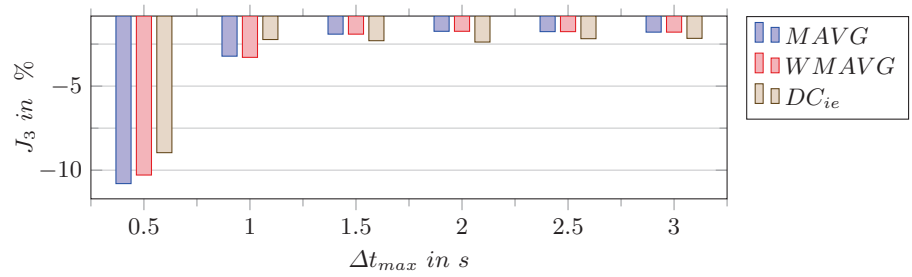
Figure 4.37: Absolute error Strategy II: $e = DC(n,\dot{n}) - DC_{HO}(i)$ in %



(a)Average error per sample



(b)Greatest positive error



(c)Greatest negative error

Figure 4.38: Strategy II: dynamic DC-error, dynamic speed trace

4.6 Load Influence

4.6.1 static load

As mentioned in section 2.2, the value of V_{bat} depends on the load or rather on the load current. In section 4.3.3 it is shown by calculation, the lower V_{bat}, the lower i_e. However, the DC value stays the same if the alternator parameters are constant, see table 4.3. If saturation is considered, it changes in case of very low revolutions ($1000rpm$), see table 4.4. The same is shown now by simulation. $\overline{DC}_{BB}$ and $\bar{i}_e$ are mean values of the whole simulation.

	$R_l = 0.195\Omega,\ V_{bat} = 9.6V$			$R_l = 0.39\Omega,\ V_{bat} = 11.6V$			No load, $V_{bat} = 12.6V$		
n	$\bar{i}_e$	$\overline{DC}_{BB}$	$t_6 - t_1$	$\bar{i}_e$	$\overline{DC}_{BB}$	$t_6 - t_1$	$\bar{i}_e$	$\overline{DC}_{BB}$	$t_6 - t_1$
rpm	A	%	ms	A	%	ms	A	%	ms
1000	1.12	40.63	120.0	1.26	44.12	120.0	1.49	48.1	112.5
1500	0.70	23.99	60.0	0.75	24.34	60.0	0.82	24.5	55.0
2100	0.51	17.13	53.6	0.55	17.46	53.5	0.59	17.33	50
3000	0.37	11.68	52.5	0.39	11.68	55.0	0.43	11.85	52.5

Table 4.11: Simulation results, static load variation, generator in no load operation

4.6.2 switching loads

If a load is switched off during operation, the value of V_{bat} or rather V_{ref} increases. As a consequence, the phase controller has to raise the field current, which may take a short moment. On the contrary, if a load is switched on abruptly, V_{ba} decreases promptly. The field current must decrease. Since reducing i_e is slow (passive by $U_{exc} = 0V$), it may happen that the phase voltage is higher than the recent critical value $\hat{U}_{ph,crit}(V_{bat})$. Subsequently, output current is produced for a short time span. However, it depends on how fast and how much it changes.
Figure 4.39 shows a simulation with an abrupt changing load from $30A$ to $50A$ and back. In terms of clarity the envelop of $U_{ph}(t)$ is plotted. Obviously, the phase controller can handle the situation without problems. Another simulation with an abrupt change from $0A$ to 80 is shown in figure 4.40. Since, the reduction of i_e is too slow, there are output current peaks with an amplitude value about $10A$. The peaks could be prevented by declining $V_{ref} \rightarrow \Delta U < 0V$. In contrast to the phase voltage overshoot caused by a strong speed gradient, it lasts only a short time. Thus, declining V_{ref} seems not be worth it. Moreover, the mechanical inertia probably dampens the consequences of the output peaks.
A simulation with a switching load and dynamic speed trace is shown in section 6.1.2.

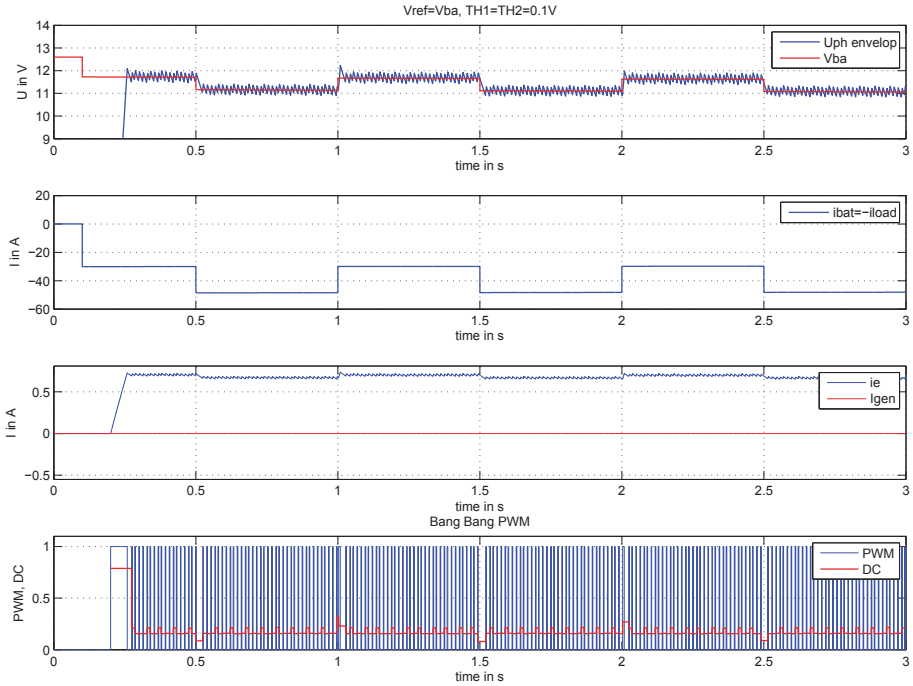


Figure 4.39: Rotor speed $n = 2100rpm$ and switching load $0A \rightarrow 50A \rightarrow 30 \rightarrow 50$...

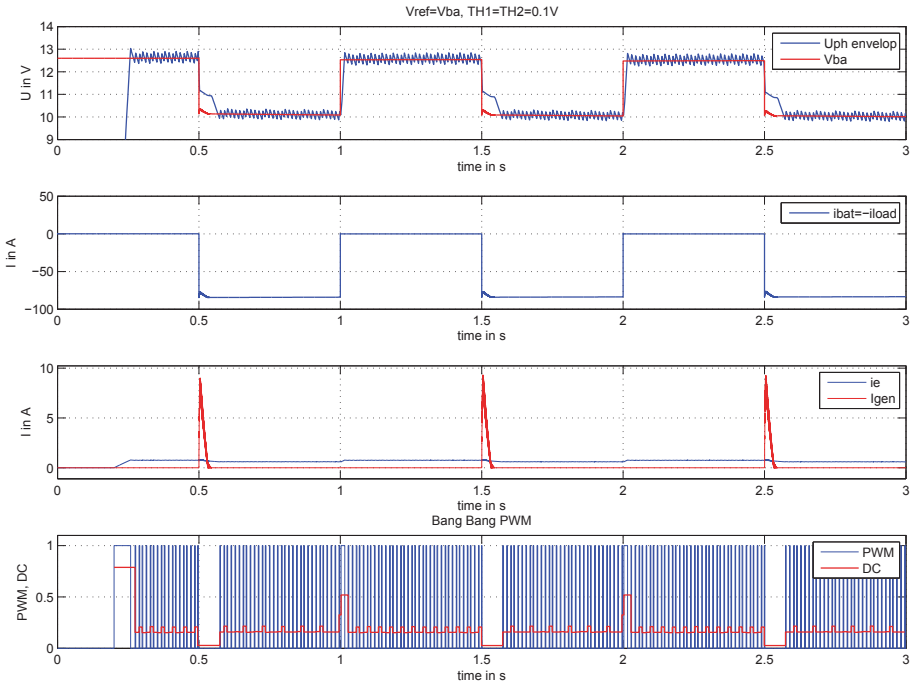


Figure 4.40: Rotor speed $n = 2100rpm$ and switching load $30A \rightarrow 50A \rightarrow 30A \rightarrow 50A$...

4.7 V_{BA} Controller and LRC

In order to run a complete simulation of the system the V_{BA} controller is designed as well. Hence, the IC uses a PI-controller (see [4]) the same type is used here. The following is a simplified behaviour model of the real IC. The final Simulink implementation is given in the appendix 7.

The aim is to control the direct component of V_{BA}. As a matter of fact, there is always a ripple in the voltage trace because of the rectifier properties. Due to this ripple and other effects (noise,...) it is necessary to filter $V_{BA}(t)$. In [19] on page 57 it is described that the IC uses a second order analogue filter. The cut-off frequency is $f_c = 2kHz$ (anti aliasing). Moreover, the control deviation of the PI controller is determined and sampled. The sampled error is low pass filtered again by a discrete filter, which has a cut-off frequency of $f_c = 190Hz$.
In this thesis it is simplified. V_{BA} is low pass filtered by a second order continuous filter, which has a cut-off frequency of $f_c = 160Hz \Rightarrow \bar{V}_{BA}(t)$. The filtered voltage is sampled and the controller error calculated without further filtering. Since discrete time steps are used it follows $t = kT_d$:

$$e_{PI_k} = e_{PI}(kT_d) = V_{set} - \bar{V}_{BA}(kT_d) \tag{4.183}$$

T_d is the sample time and k the index of the current iteration.

4.7.1 Controller Design

A simple approach is conducted, hence the V_{BA} controller is needed only to demonstrate how the ECC works. The idea is to design it, for an certain operation point (linearisation). The proceeding is the following:

1. Identification of the control process $G(s) = \frac{\bar{V}_{BA}(s)}{U_e(s)}$ by simulation
2. Design of a discrete PI controller using the frequency response characteristic
3. Output conversion, PWM generation and Anti-Windup design

Process Identification and sampling rate

The process parameters are estimated by simulating the step response. The inner core of the controller, considers the excitation voltage u_e as output variable. For that reason, a voltage step is applied. A constant rotor speed of $n = 2100rpm$ and $i_e = 2A$ are chosen as operating point. This, gives a step amplitude of $u_e = R_e i_e = 5.62V$. Figure 4.41 shows the simulated response of the alternator with a connected load $R_{load} = 0.39\Omega$ (blue). The

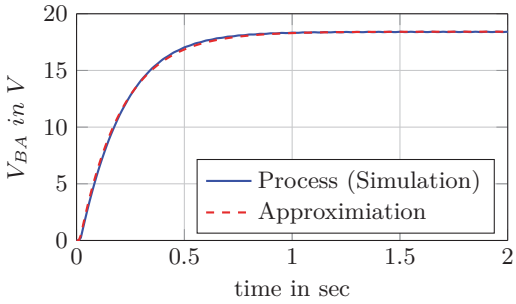


Figure 4.41: Step response of the control process

battery is disconnected for the design. According to Ziegler-Nichols [20](see also [1] pages 75/76) the process system can be approximated by:

$$G(s) \approx K_S \frac{1}{\tau s + 1} e^{-Ls} = 3.28 \frac{1}{0.2s + 1} e^{-0.015s} \tag{4.184}$$

The delay time L is determined by the first point in time $V_{ba}(t) \geq 1e^{-4}V$. The gain is given by $K_S = \frac{V_{BA}(t=2s)}{u_e}$. By fitting an exponential function into the trace of $\frac{V_{BA}(t>L)}{K_S}$, τ is identified. The trace of the approximated process is plotted in red in figure 4.41. Assuming a linear system behaviour (constant inductance) the generator can be described by a transfer function first order (PT_1). The delay time is probably caused by the diodes of the rectifier $\Rightarrow PT_1T_L$.

Since a discrete design will be conducted, the time continuous function $G(s)$ is transformed to a discrete transfer function $G^*(z)$. As explained in [7] on page 46, $G^*(z)$ is calculated by:

$$G^*(z) = \frac{z-1}{z} \boldsymbol{Z} \left\{ \frac{G(s)}{s} \right\} \tag{4.185}$$

This poses the question of which sample time T_d should be used. The Nyquist-Shannon sampling theorem claims that the sample rate has to be two times higher than the fastest system frequency. The delay time, can be approximated by an all-pass filter. This is called *Padé* approximation. The proceeding can be found in [10] on pages 196/197:

$$e^{-sL} \approx \frac{1 - L\frac{s}{2}}{1 + L\frac{s}{2}} \tag{4.186}$$

Since, $\frac{L}{2} < \tau$ a minimum rate could be calculated by:

$$fs \geq 2\frac{1}{\frac{L}{2}} \qquad\qquad fs \geq 266Hz \tag{4.187}$$

The point is, that the DC value for the PWM generation will be updated with $440Hz$. Moreover, the final ECC algorithm uses a system clock of $22kHz$. From the same clock,

$2200Hz$ and $220Hz$ are deduced. The V_{ba} controller is triggered by $2200Hz$, due to synchronisation issues.

$$T_d = \frac{1}{2200Hz} \tag{4.188}$$

To sum up, the calculation rate of the PI controller is fixed with $2200Hz$. Every fifth value is fed to the output stage, hence the update rate of the PWM is $440Hz$. Applying equation 4.185 on 4.184 and $T_d = {}^1/_{2200Hz}$ gives:

$$G^*(z) = \frac{0.00745}{z - 0.9977} z^{-33} \tag{4.189}$$

In order to use the frequency response characteristic in the same way as for continuous systems, the bilinear q-transformation is applied (see [7] page 76):

$$G^*(q) = G(z)|_{z=\frac{1+q\frac{T_d}{2}}{1-q\frac{T_d}{2}}} \tag{4.190}$$

$$G^*(q) = \frac{-0.0037273(q - 4400)}{(q + 5)} e^{-0.015q} \tag{4.191}$$

Discrete controller

The time continuous PI-controller is given by:

$$u_e(t) = K_{PI}\left(e_{PI}(t) + \frac{1}{T_N}\int e_{PI}(t)dt\right) = K_{PI}e_{PI}(t) + \frac{K_{PI}}{T_N}\int e_{PI}(t)dt \tag{4.192}$$

Using the *Laplace*, z and q-transformation gives:

$$U_e(s) = \mathcal{L}\{u_e(t)\} = K_{PI}E_{PI}(s) + \frac{K_{PI}}{T_N}E_{PI}\frac{1}{s} \tag{4.193}$$

$$R(s) = \frac{U_e(s)}{E(s)} = K_{PI} + \frac{K_{PI}}{T_N}\frac{1}{s} = K_{PI}\frac{\left(s + \frac{1}{T_N}\right)}{s} \tag{4.194}$$

$$R(z) = \frac{U_e(z)}{E(z)} = K_{PI} + \frac{K_{PI}}{T_N}T_d\frac{1}{z-1} = \breve{K}_{PI}\frac{z + \frac{1}{\breve{T}_N}}{z-1} \tag{4.195}$$

$$R(q) = \frac{U_e(q)}{E(q)} = K_{PI} + \frac{K_{PI}}{T_N}\frac{1 - q\frac{T_d}{2}}{q} = \tilde{K}_{PI}\frac{\left(q + \frac{1}{\widetilde{T}_N}\right)}{q} \tag{4.196}$$

There are no specifications given regarding the rise time and the overshoot of the closed loop. The zero of $R(q)$ is chosen in such a way, that it cancels the pole of $G^*(q) \rightarrow \widetilde{T}_N = 0.2$. The gain $\tilde{K}_{PI}$ is adjusted for a phase margin of $\phi_r = 60\% \rightarrow \tilde{K}_{PI} \approx 2.09$, which gives a 10% overshoot. For a phase margin of $\phi_r = 50\%$ (overshoot 20%) a gain of $\tilde{K}_{PI} \approx 2.63$ is determined.

$$R_{10}(q) = 2.09\frac{q+5}{q} \qquad R_{20}(q) = 2.63\frac{q+5}{q} \tag{4.197}$$

Figure 4.42 shows the design steps using the bode diagram. Note, that the x-axes show the transformed frequency $\Omega = \frac{2}{T_d} tan\left(\frac{\omega}{2}\right)$. The inverse q-transformation applied on $R_{10}(q)$ and $R_{20}(q)$ gives:

$$R_{10}(z) = 2.1\frac{z - 0.9977}{z - 1} = \frac{2.1z - 2.097}{z - 1} \tag{4.198}$$

$$R_{20}(z) = 2.82\frac{z - 0.9977}{z - 1} = \frac{2.82z - 2.815}{z - 1} \tag{4.199}$$

The step response of the closed loop system $T(z) = \frac{G^*(z)R(z)}{1+G^*(z)R(z)}$ is presented in figure 4.43.

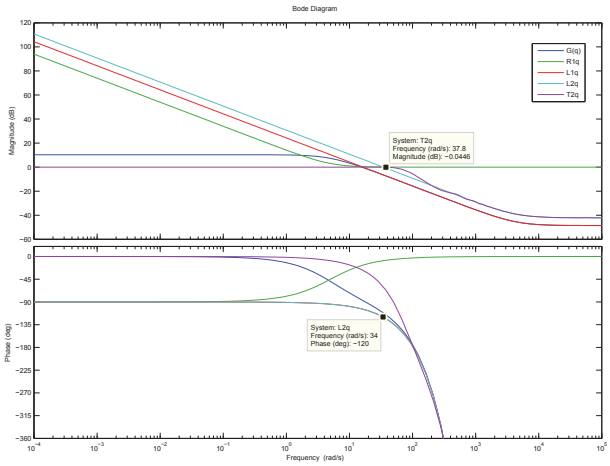


Figure 4.42: Bode diagram controller design

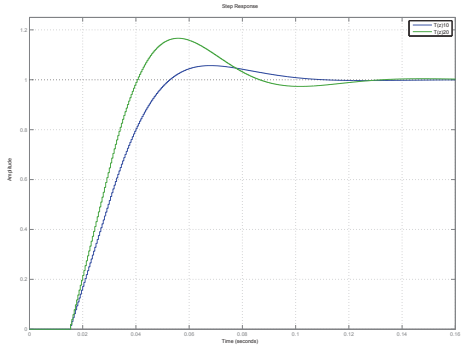


Figure 4.43: Step response $T(z)$

Rearranging 4.195 leads to:

$$R(z) = \frac{K_{PI}z + \left(\frac{K_{PI}}{T_N}T_d - K_{PI}\right)}{(z - 1)} = \frac{b_1 z + b_0}{z - 1} \tag{4.200}$$

From equation 4.200 it can be seen:

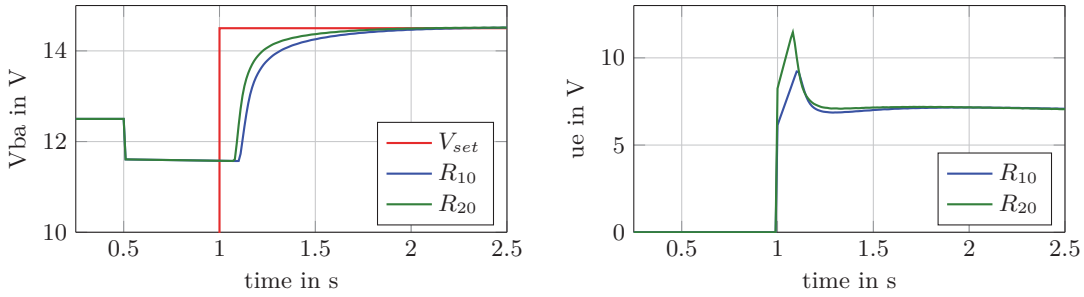
$$b_1 = K_{PI} \qquad b_0 = \left(\frac{K_{PI}}{T_N}T_d - K_{PI}\right) \tag{4.201}$$

Obviously b_1 and b_0 are the coefficients of the transfer function 4.198 or rather 4.199. The values of K_{PI} and T_N can be calculated, since T_d is known, $R_{10}(z): \ K_{PI} = 2.1, T_N = 2.002$, $R_{20}(z): \ K_{PI} = 2.63, T_N = 2.002$.
Further rearranging of equation 4.195 gives:

$$U(z) = K_{PI}E(z) + \left(\frac{K_{PI}}{T_N}T_d - K_{PI}\right)E(z)z^{-1} + U(z)z^{-1} \tag{4.202}$$

As described in [7] a recurrence relation can be written as z equation by:

$$\mathcal{Z}\{f_k\} = \sum_{i=0}^{\infty} f_i z^{-1} \tag{4.203}$$



(a)$\bar{V}_{BA}(t)$ (b)$u_e(kt)$

Figure 4.44: Closed loop simulation controller, alternator, battery $V_{bat_0} = 12.5V$ and load $R_{load} = 0.39\Omega$

Applying $\mathcal{Z}^{-1}$ on equation 4.202 results in:

$$u_{e_k} = K_{PI} e_{PI_k} + \left(\frac{K_{PI}}{T_{PI}} T_d - K_{PI} \right) e_{PI_{k-1}} + u_{e_{k-1}} = K_{PI} e_{PI_k} + i_k \tag{4.204}$$

i_k is the discrete integrator. During the iteration k, i_{k+1} is calculated, which is used as current integrator content in the next iteration (equation 4.204):

$$i_{k+1} = \frac{K_{PI}}{T_{PI}} T_d e_{PI_k} - K_{PI} e_{PI_k} + u_{e_k} = \frac{K_{PI}}{T_{PI}} T_d e_{PI_k} + i_k \tag{4.205}$$

Equation 4.204 and 4.205 are implemented in block 3) of figure 4.6. A closed loop simulation of the controllers, alternator model, battery and load is presented in figure 4.44. The settings are $n = 2100rpm$ and $V_{set} = 14.5V$ (activated at $t = 1s$).

It can be seen, that there is no overshooting in both cases (R_{10} and R_{20}). The final algorithm uses the R_{20}, hence the results are slightly better. The principle patent simulations in section 2.2 are done with a more agile PI-controller $K_{PI} = 3.7$, $T_{PI} = 0.05$ in order to illustrate the load response effect. In fact, these values were found by using the empirical formula of Ziegler-Nichols [20] (see also [1] pages 75/76).

Output conversion, PWM and Anti-Windup

Output conversion: The final output is a DC value. Thus, the internal calculated u_e is divided by the current V_{BA} value:

$$DC_{PI_k} = \frac{u_{e_k}}{\bar{V}_{BA_k}} \tag{4.206}$$

Of course DC_{PI_k} is limited within $[0, 1]$. Moreover, the LRC limits the rise gradient of the DC trace. That means, the saturation value DC_{sat} is the current LRC value: $DC_{sat_k} = DC_{LRC_k}$. In case it is not active $DC_{sat_k} = sat(DC_{PI_k})$.
Anti wind-up: Hence the controller has an integrating component, an anti wind-up

mechanism is needed. Otherwise, overshoots can be caused. Moreover, u_e (in V) is considered as output value by the internal controller. Thus, DC_{sat} must be converted to $u_{e,sat}$:

$$u_{e,sat_k} = DC_{sat_k} V_{BA_k} \tag{4.207}$$

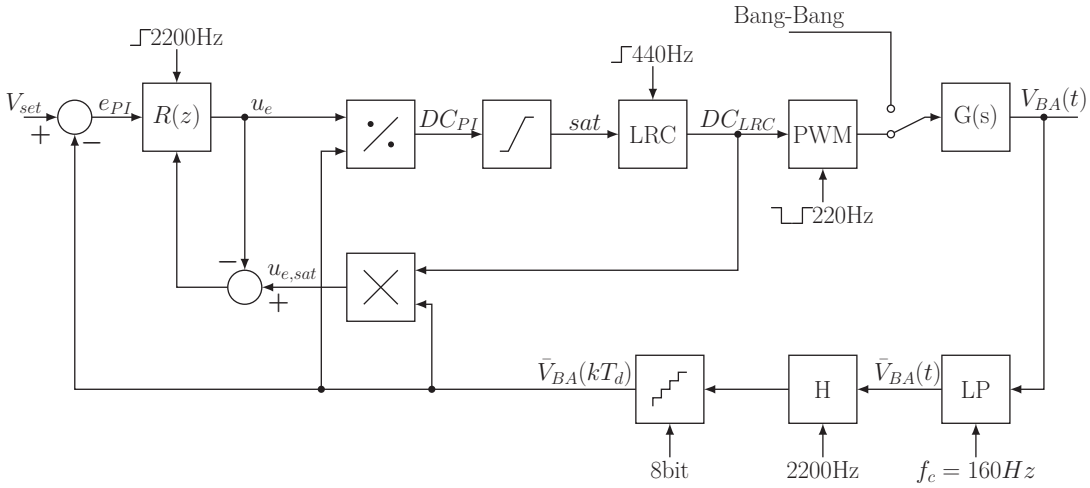


Figure 4.45: Schematic overview V_{BA} controller

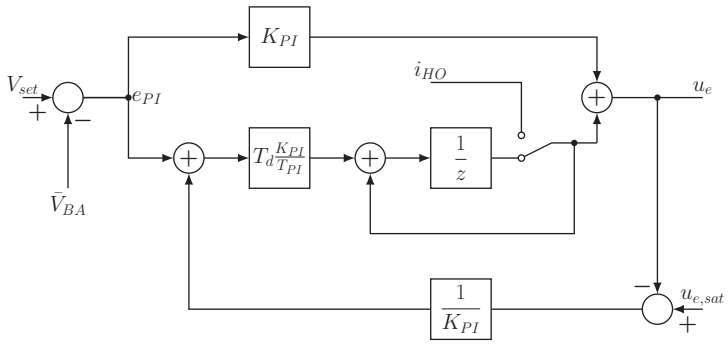


Figure 4.46: PI-controller with anti wind-up mechanism

A schematic overview of the closed loop is given figure 4.45. The conditioning technique of *Hanus* [9] is used as anti wind-up. Origin are equations 4.204 and 4.205. The main idea is to determine a fictive set-point V^*_{set} which leads to $u^*_{e_k}$=u_{e,sat_k} if the limitation is active. However, this is equivalent to calculate a fictive error $e^*_{PI_k}$ which leads to $u^*_{e_k}$=u_{e,sat_k}:

$$u_{e,sat_k} = K_{PI} e^*_{PI_k} + i_k \tag{4.208}$$

$$i_{k+1} = i_k + \frac{K_{PI}}{T_{PI}} T_d e^*_{PI_k} \tag{4.209}$$

Rearranging equation 4.204 to i_k and inserting into equation 4.208 gives:

$$e^*_{PI_k} = \frac{u_{e,sat_k} - u_{e_k}}{K_{PI}} + e_{PI_k} \tag{4.210}$$

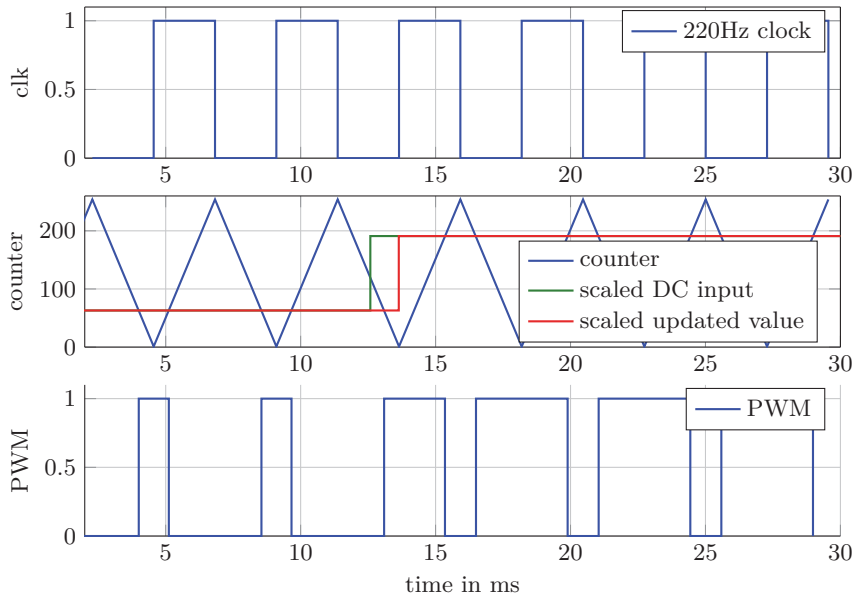


Figure 4.47: Phase synchronous PWM

Finally $e^*_{PI_k}$ inserted in equation in 4.209 gives:

$$i_{k+1} = i_k + \frac{K_{PI}}{T_{PI}} T_d \left(\frac{u_{e,sat_k} - u_{e_k}}{K_{PI}} + e_{PI_k} \right) \tag{4.211}$$

Figure 4.46 shows the structure of the anti wind-up mechanism.

PWM generation: The DC value is converted to a PWM, using an eight bit counter. As stated in section 4.1 the PWM frequency is $220Hz$. Actually, this is only true if the DC value is not changing. The implementation allows an update with $440Hz$. Figure 4.47 shows how it works. An input change from $DC = 0.25$ to $DC = 0.75$ is simulated. At each edge of the $220Hz - PWM$ clock, the counter direction is changed ($440Hz$). At the same time the counter comparison value is updated. The comparison value is the input DC multiplied by MAX $= 2^8 - 1$:

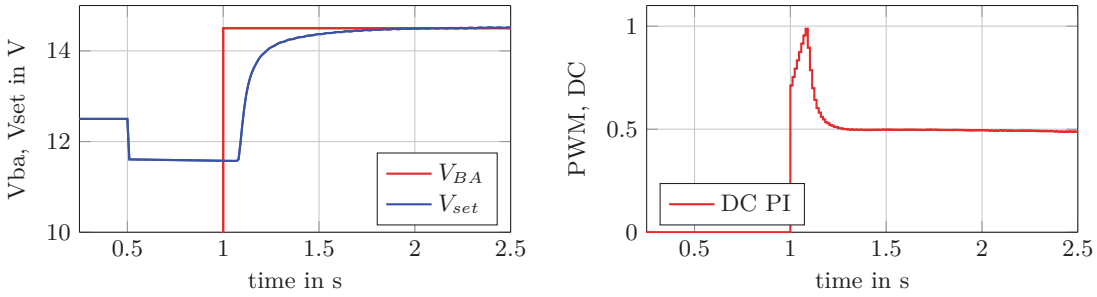
$$\text{comparison value} = DC \cdot MAX \tag{4.212}$$

where $DC \in [0, 1]$. The counter frequency is calculated by:

$$T_{cnt} = \frac{\frac{1}{2f_{PWM}}}{\text{MAX}} \qquad f_{cnt} = \frac{1}{T_{cnt}} \tag{4.213}$$

Finally the PWM is given by:

$$U_{exc} = \begin{cases} V_{BA}, & \text{if counter value} \leq \text{compare value} \\ 0V, & \text{otherwise} \end{cases} \tag{4.214}$$



(a)$\bar{V}_{BA}(t)$ (b)$DC(kt)$

Figure 4.48: Closed loop simulation controller (R_{20}) with output conversion, alternator, battery $V_{bat_0} = 12.5V$ and load $R_{load} = 0.39\Omega$

The PWM function block is named $DC2PWM$, the recent DC of the PWM is called DC_{2PWM}. DC_{2PWM}(which is either $DC_{PI,440Hz}$ or DC_{LRC}) is fed back to the anti wind-up mechanism:

$$DC_{sat_k} = DC_{2PWM_j} \tag{4.215}$$

where j is the index of the $440Hz$ iteration (see next section 4.7.2).
The V_{BA} controller and the $DC2PWM$ block work synchronously, hence the PWM clock ($220Hz$) is deduced from the system clock. Figure 4.48 shows the same simulation as before including the output conversion, anti wind-up mechanism and $DC2PWM$. Actually, the anti wind-up mechanism is not active, because the internally calculated DC is not greater than one. It will be active in the next section, because of the LRC limitation.

4.7.2 Load Response Control

The task of the LRC has been described in section 1.1 and 2.1.4. Basically, the following calculation is done every $T_{440Hz} = 1/440Hz$ if it is not disabled by the end-user or a speed condition:

$$DC_{PI,440Hz_j} = DC_{PI,2200Hz_k} \tag{4.216}$$

$$PI_{grad_j} = \frac{DC_{PI,440Hz_j} - DC_{2PWM_{j-1}}}{T_{440Hz}} \tag{4.217}$$

$$LRC_{grad} = \frac{1}{LRC - RT} \tag{4.218}$$

$$DC_{LRC_j} = DC_{2PWM_{j-1}} + LRC_{grad} T_{440Hz} \tag{4.219}$$

$$DC_{2PWM_j} = \begin{cases} DC_{LRC_j}, & \text{if } PI_{grad_j} \geq LRC_{grad} \\ DC_{PI,440Hz_j}, & \text{otherwise} \end{cases} \tag{4.220}$$

Note, k is fives times incremented, when j increments one. Also the $LRC - BLZ$ and $LRC-RT$ have been implemented. A test of the LRC function itself is shown in section 2.1.4,

figure 2.3. The result of the previous simulation with enabled LRC is shown in section 6.2.1 figure 6.4a.

4.7.3 DC Handover

The idea of the ECC is that the LRC ramp starts with an offset, which is the determined handover value DC_{HO} (see concept 4.1, task three). To ensure a smooth transition from $\hat{U}_{ph}$ control (bang-bang controller) to V_{BA} control, the integrator content of the PI-controller is manipulated. The aim is to ensure that the internally calculated DC_{PI} value matches with the handover value DC_{HO}:

$$DC_{PI} \stackrel{!}{=} DC_{HO} \tag{4.221}$$

At the same time, the LRC-function is forced to pass through DC_{PI} for one iteration j.

The handover command HO_{cmd} is controlled by the ECC-Master. If it is set true, the integrator content is overwritten by:

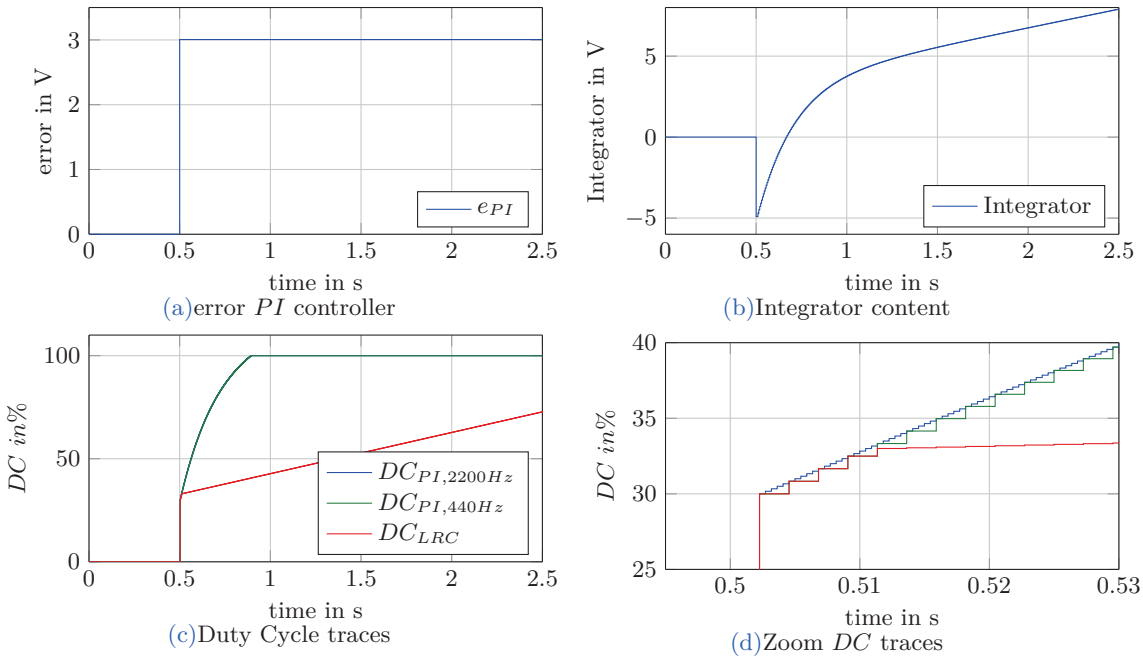
$$u_{HO} = DC_{HO} V_{BA_k} \tag{4.222}$$

$$i_k = i_{HO} = u_{HO} - K_{PI} e_{PI_k} \quad \forall \quad HO_{cmd} == true \tag{4.223}$$

Equation 4.223 is the result of rearranging equation 4.204 and replacing u_{e_k} by u_{HO}. Note, e_{PI_k} is the current error of the V_{BA} controller ($V_{set} - V_{BA_k}$). Of course, this is done before, equation 4.204 is executed. In figure 4.46 this indicated by a switch. For one iteration k, the upper path with i_{HO} is put through. The LRC-function is manipulated by setting the current gradient zero:

$$PI_{grad_j} = 0 \quad \forall \quad HO_{cmd} == true \tag{4.224}$$

Figure 4.49 shows an open-loop simulation of the handover proceeding. The settings are: $V_{set} = 14.5V$, $V_{BA} = 11.5V$, $LRC - RT = 5s$, $LRC - BLZ = 3\%$, $DC_{HO} = 30\%$ and activation at $= 0.5s$. Due to an error of $3V$ and $DC_{HO} = 0.3$, the integrator is set to minus $5V$(plot 4.49b). In figure 4.49c it can be seen that the PI-controller gives as output value the desired 30% and the LRC-ramp starts with this value. Moreover, the zoom of the DC traces (plot 4.49d)shows that the LRC trace follows the PI-controller till $DC = 33\%$. The further increase of the LRC ramp is according to the RT gradient. This is by purpose, because the $LRC - BLZ = 3\%$ is enabled. The end-user can configure, weather the BLZ will be used during the handover or not. It also can be seen that DC traces are a bit delayed. Due to synchronisation issues, the first DC calculation is executed at the next falling or rising edge of the $220Hz$ clock. Actually, it is implemented by examining if the current $2200Hz$ period k is a multiple integer of $5 \cdot 220Hz$.



(a)error PI controller (b)Integrator content (c)Duty Cycle traces (d)Zoom DC traces

Figure 4.49: Open loop handover proceeding

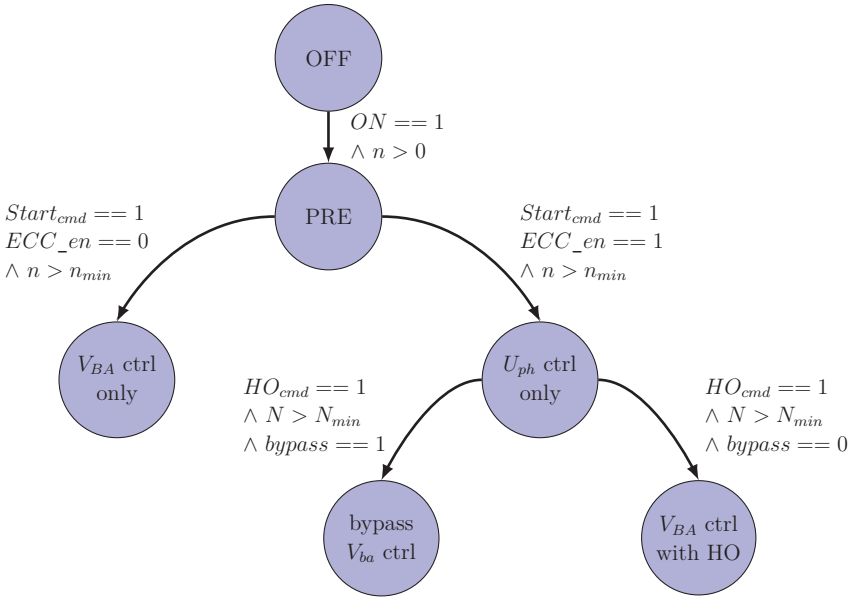
4.8 Complete Simulation

The complete simulation is defined in this thesis as simulation of all blocks together, see figure 4.5. The overall goal is to demonstrate how the ECC works and how much time can be saved with it.The complete simulation can be considered to be a test-bench for the ECC algorithm. An overview of the final Simulink implementation is given in the appendix 7. However, two more functions are missing:

4.8.1 Deep Thought

This function-block is not shown in figure 4.5. It is an additional element, which is used for test and examination purposes. Equations 4.52, 4.81, 4.51 as well as 4.165 and 4.161 are implemented. Moreover, all alternator model parameters are given and the current speed $n(t)$, $\dot{n}(t)$ as well $V_{BA}(t)$. Thus, the theoretical $DC(n(t), \dot{n} = 0, V_{BA}(t))$ and $DC(n(t), \dot{n}(t), V_{BA}(t))$ are on-line calculated. For example, it has been used for error calculations of the handover value in section 4.5.1 and 4.5.2. Also the closed loop handover process under ideal conditions can be simulated with it. This block is named after the supercomputer from the novel *"The Hitchhiker's Guide to the Galaxy"*(Douglas Adams). The implementation is given in the appendix 7.

4.8.2 ECC Master



states	description
OFF:	nothing active
PRE:	V_{BA} and U_{ph} controller off, PSB [a] is available (depends on initialization)
V_{BA} ctrl only:	PSB is available, V_{BA} controller enabled, LRC depends on initialization
U_{ph} ctrl only:	bang-bang controller enabled, as well DC-detection and handover calculation
V_{BA} ctrl with HO:	V_{BA} controller enabled and manipulation of the PI-controller possible,
by pass V_{BA} ctrl:	handover of the calculated DC-value directly to $DC2PWM$ function block

variables	description
ON:	general on, point in time adjustable in the initialization file
$Start_{cmd}$:	starts regulation, point in time adjustable in the initialization file
n_{min}:	minimum rotor speed in rpm, value adjustable in the initialization file
ECC_en:	ECC enabled, bit adjustable in the initialization file
HO_{cmd}:	handover command, point in time adjustable in the initialization file
N_{min}:	minimum Number of bang-bang PWM periods, value adjustable in the initialization file
$bypass$:	bypass V_{BA} controller, bit adjustable in the initialization file

Figure 4.50: finite state machine ECC-Master

[a] Actually, the start-boost mechanism of the bang-bang controller is used as PSB in this thesis: $PWM = 1\ \forall\ \hat{U}_{ph} < V_{PSB},\ PWM = 0\ \forall\ \hat{U}_{ph} \geq V_{PSB}$

As already mentioned, the ECC Master controls all other elements. It also takes care of the handover process and gives the reference value $V_{ref} = V_{BA} + \Delta U$ for the phase controller. The master is initialised via a script file. Depending on the settings the following simulations are possible:

- V_{BA} control only: closed loop simulation alternator, PI-controller and enabled or disabled LRC, see section 4.7

- U_{ph} control only: closed loop simulation alternator, phase controller with or without DC handover to $DC2PWM$(by pass V_{BA} controller, see figure 4.4)
- V_{BA} control with (test) handover: closed loop simulation alternator, manipulatable PI-controller and LRC with any DC_{HO} value
- Complete simulation: closed loop simulation, alternator, phase controller, DC_{HO} handover to V_{BA} controller, manipulatable PI-controller and LRC

The ECC Master is implemented as finite-state machine. Figure 4.50 shows the state diagram. As explained in section 4.5, different filtering methods are implemented in the DC determination block (function-block no 2) (DC_{MAVG}, DC_{EMA}, DC_{ie},..). The end-user selects which DC value will be used as handover value in the script file. Also, the handover options, minimum number of PWM-periods or certain point in time, can be chosen[1]. General simulations settings like, speed trace, load, parameter of the controllers, etc are configurable as well. The source code of the initialization file and the Simulink implemenation of the ECC Master is given in the appendix 7

4.8.3 Timing concept

The V_{BA} controller, the master, and the DC determination block are implemented in Matlab functions. The master is triggered by the system clock $22kHz$. Since, the eight bit counter, which is used for the determination of the recent DC_{BB} value, is implemented in the DC determination block, it is also triggered by the $22kHz$. Nevertheless, the averaging algorithms are executed only if the rising edge of the PWM is detected (asynchronous). As mentioned before, the V_{BA} controller is triggered by $2200Hz$ and the PWM generation block uses a $220Hz$ signal to control another internal counter (see section 4.7.1).

[1] If the minimum number of PWM-periods is desired, the end-user has to configure the same point in time for $Start_{cmd} = 1$ and $HO_{cmd} = 1$

5 Alternator Measurements

Several alternator measurements have been conducted in order to develop a new model as part of further work. Notwithstanding, a few measurements are presented here, which are relevant for the ECC algorithm. The following measurements refer to the *Denso SC6* alternator.

5.1 Standstill test

The ECC parameters are determined without considering eddy currents. In order to figure out the influence of them, excitation voltage steps are applied to a real alternator. The goal is to identify the excitation inductance $L_e(I_\mu)$, the excitation resistor Re and the mutual inductance $M(I_\mu)$. With these alternator parameters, a standstill model is developed. Since these parameters are determined at steady-state, eddy currents are not considered by the model. The same voltage steps as for the real measurements are simulated. The comparison of the measured traces and the simulated trace will show the influence of eddy currents.

5.1.1 Measurement

General setup:

- The rotor is not driven (standstill)
- There is no load connected to the rectifier (open load)
- The brush-holder (without alternator IC but connections for external supply) is mounted and connected to a dc-link converter and a power amplifier
- The trace of the excitation current $i_e(t)$ and the excitation voltage $u_e(t)$ are measured
- The trace of two stator voltages (delta voltages) is measured

Figure 5.1 shows the principle set-up. The excitation voltage u_e is tapped directly at the slip rings. u_{step} is the applied voltage and R_{carbon} the resistance of the brush-holder.

Basically, a voltage step is applied to the field winding. From the resulting i_e, the flux-linkage Ψ_e is calculated and finally the self inductance at the steady state. This is described

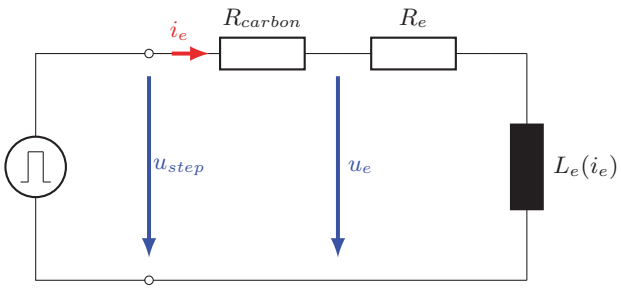


Figure 5.1: Excitation circuit with brushholder and pulse source

by the following equations:

$$u_{e_{ss}} := u_e(t = t_{steady-state}) \tag{5.1}$$

$$i_{e_{ss}} := i_e(t = t_{steady-state}) \tag{5.2}$$

$$R_e = \frac{u_{e_{ss}}}{i_{e_{ss}}} \tag{5.3}$$

$$\Psi_{e,step}(t) = \int_0^t (u_e(\tau) - R_e i_e(\tau))\mathrm{d}\tau \tag{5.4}$$

$$\Psi_{e,step_{ss}} := \Psi_{e,step}(t = t_{steady-state}) \tag{5.5}$$

$$L_e(i_{e_{ss}}) = \frac{\Psi_{e,step_{ss}}}{i_{e_{ss}}} \tag{5.6}$$

$$\Psi_{e_0} := \Psi_e(t = 0) \tag{5.7}$$

$$\Psi_{e_{ss}} = \Psi_{e,step_{ss}} + \Psi_{e_0} \tag{5.8}$$

If the trace of two stator voltages is also measured, $\underline{u}_s(t)$ can be calculated. Furthermore, the stator flux and the mutual inductance are determined by:

$$\Psi_s(t) = \Psi_{sd}(t) = \int_0^t |\underline{u}_s(\tau)| sign(u_e(\tau) - R_e i_e(\tau))\mathrm{d}\tau \tag{5.9}$$

$$\Psi_{s_{ss}} := \Psi_s(t = t_{steady-state}) \tag{5.10}$$

$$M(i_{e_{ss}}) = \frac{\Psi_{s_{ss}}}{i_{e_{ss}}} \tag{5.11}$$

$$\tag{5.12}$$

Considerations:

1. As excitation current increases, eddy currents are induced in the iron. According to Lenz's law, they oppose a field to the source field (see [3] page 30). More details follow in section 5.2. By calculating the self inductance at the steady-state, eddy current influences are eliminated (see equation 5.6).
2. As mentioned before, R_e is needed to calculate the flux (equation 5.4). Thus, the determination of R_e has direct impact on the steps afterwards and must be done

carefully for each measurement. In view of this, R_e should be ascertained in the steady state with some averaging of u_e and i_e. The crucial point is to figure out when the system is at steady state and what a proper step width is.
First measurements showed that the rotor has a time constant about $\tau \approx 200ms$. According to this, a step width of a few seconds should be good enough. The problem with it is the measurement and equation 5.4. Depending on the equipment, each measured quantity has an offset. In equation 5.4 the offset error will be integrated over time. In order to keep the influence low, the integration time should be as short as possible. However, looking at 5.8 shows that the lapse of time till $t = t_{steady}$ is the integration time. To sum up, the conflict is the duration of the step width. On the one hand, the step width must be long enough to reach a steady state. On the other hand, it has to be short enough because of the offset error.

3. The saturation of the inductances depends on the magnetization current due to the ferromagnetic material. In this case, the magnetization current is the excitation current because of the open load $\underline{i_s} = 0$ (see equation 4.71). In order to obtain the full magnetization curve $\Psi_e(i_e)$ or rather $\Psi_{sd}(i_e)$, several voltage steps with different amplitudes are applied. It is obvious, that the main rotor time *constant* $\tau_{e,main} = \frac{L_e(i_e)}{Re}$ is a function of current as well [1]. The consequence is that $t_{steady-state}$ changes. Taking point 2 into account, it is clear that for each step height a proper step width must be chosen.
4. The total excitation flux consists of two parts: the flux caused by $i_e(t)$ (equation 5.4) and Ψ_{e_0}, which is the remanence of the rotor claws. A problem is that the remanence is influenced by the previous i_e. Applying voltage steps with different step heights will change Ψ_{e_0} each time. This leads to distorted results of the measured magnetization curve $\Psi_{e,step_{ss}}(i_{e_{ss}})$. To avoid this problem, the rotor is degaussed before a measurement is conducted.

Degaussing Process

Degaussing can be achieved by applying a sinus signal with a continuous decreasing amplitude (slowly). This was done with a power inverter that produces a pulse pattern of a sinus signal with the following values:

- Frequency $f = 0.5Hz$
- Start amplitude $U_0 = 14V$
- Gradient $\frac{\mathrm{d}U}{\mathrm{d}t} = -0.1V/s$

Figure 5.2 shows the beginning of the degaussing process. The upper plot views the pulse pattern of the inverter and the measured current. The lower plot views the fundamental

[1] As explained in point 1, eddy currents influence the excitation current trace. Thus, and because of saturation, $i_e(t)$ can not be described with a single time constant.

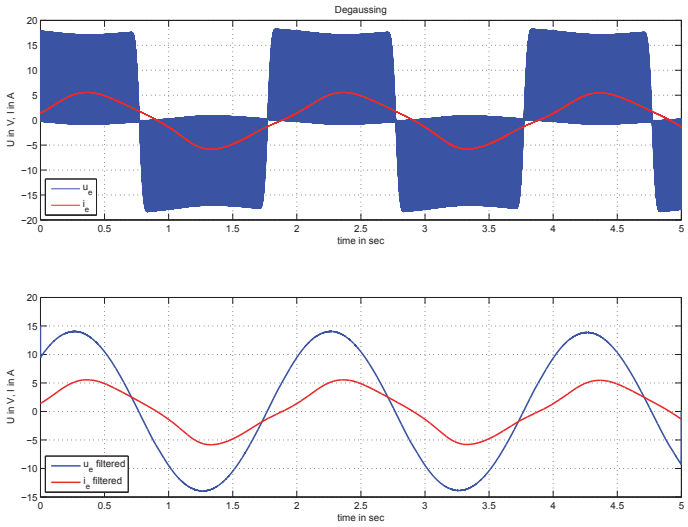


Figure 5.2: Begin of the degaussing process

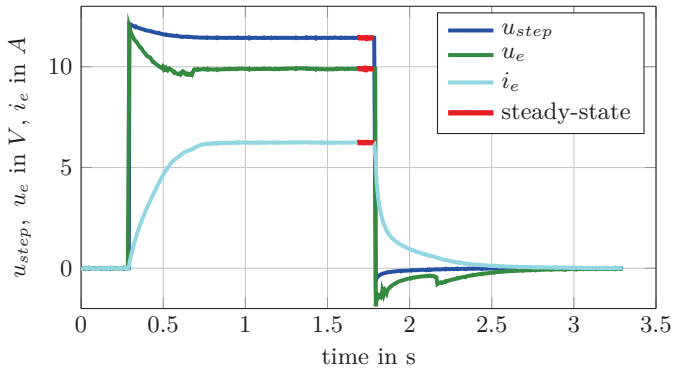
wave of the pattern and the filtered current. The amplitude of both is slowly decreasing.

Measurement Procedure

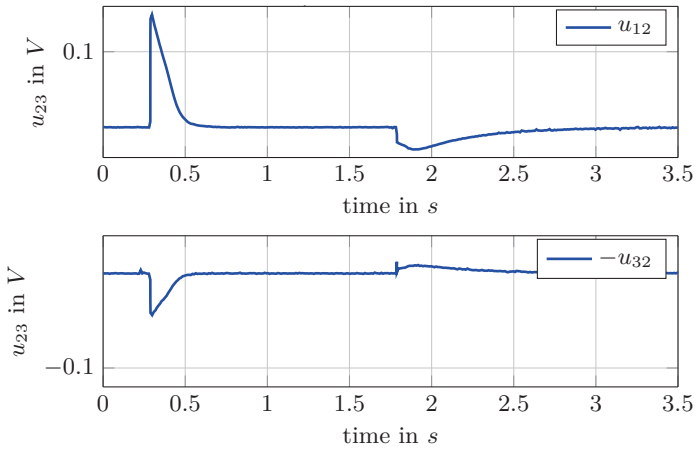
1. Degauss the rotor
2. Apply voltage step and measure the traces
3. Degauss again the rotor
4. Decrease step height, apply the voltage step and measure the traces.
5. Repeat 3. and 4. till a step height of $\sim 1V$ is reached

Results

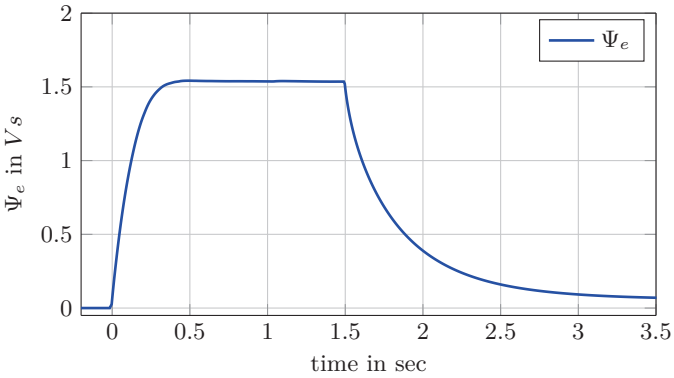
Figure 5.3 shows the traces of the first voltage step $u_{step} = 12V$. The steady-state region is considered as the part at the end of the step (marked in red), see figure 5.3a. The flux $\Psi_{e,step}(i_e(t))$ is calculated by equation 5.4 and plotted over time in figure 5.3c. The results of all steps are shown in figure 5.4. With decreasing step height, the length of the step has to be extended (see figure 5.4a). The calculated flux $\Psi_{e,step}(i_e(t))$ is plotted over $i_e(t)$ in figure 5.4b. Same for the stator flux, $\Psi_s(i_e(t))$ is determined by equation 5.9 and plotted in figure 5.4c. A special interpolation function on the basis of [13] is utilized through the steady-state values. Also the determined rotor resistor and the sum of rotor and carbon (brushholder) resistor are shown in figure 5.4d. The magnetization curve (steady-state values of the rotor flux and the stator flux) are shown in figure 5.5a. Finally $L_e(i_e)$ and $M(i_e)$, are calculated by equation 5.6 or rather 5.11 and plotted in figure 5.5b.



(a)The voltage drop of the source's internal resistor and of the supply line decreases u_{step}

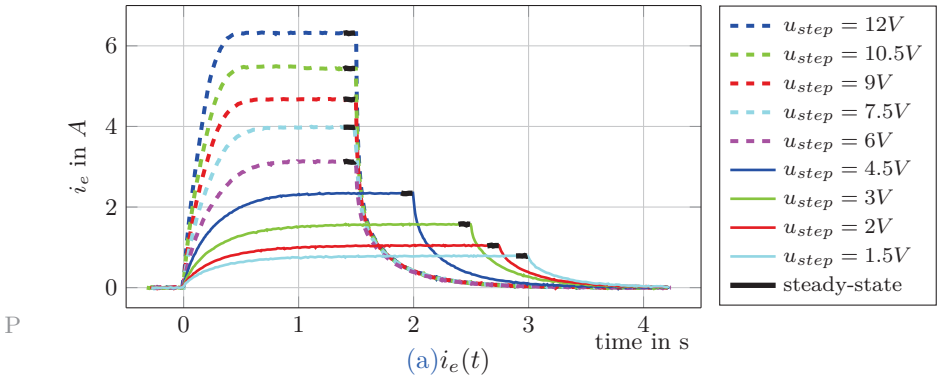


(b)measured trace of delta voltages

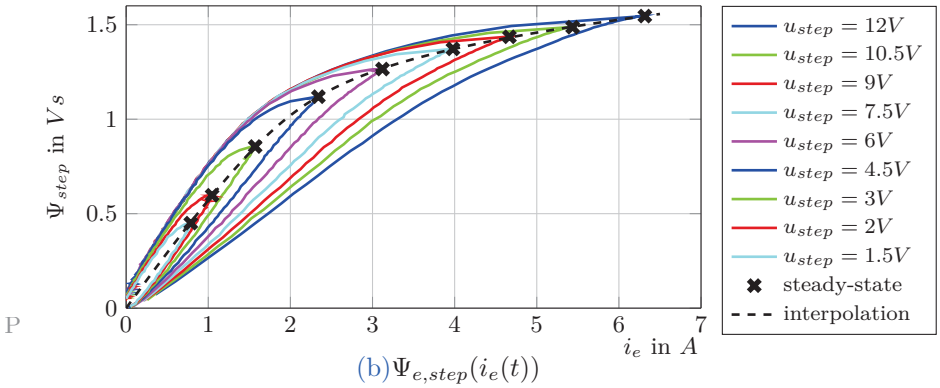


(c)calculated $\Psi_{e,step}(t)$

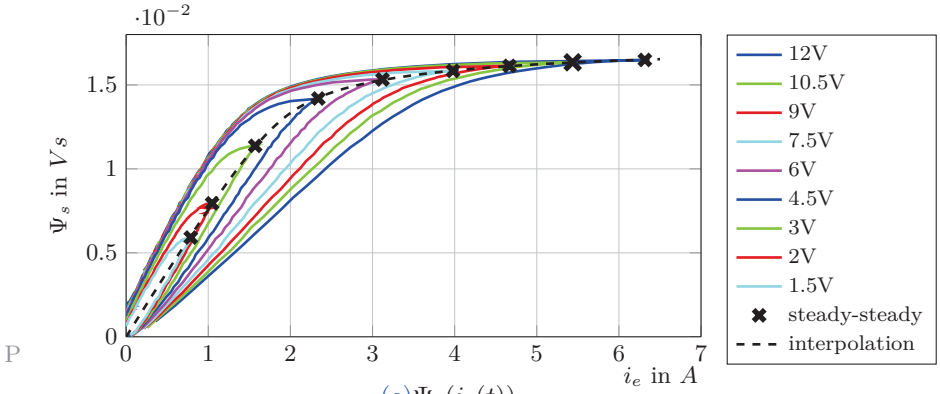
Figure 5.3: Voltage step with an amplitude of 12V



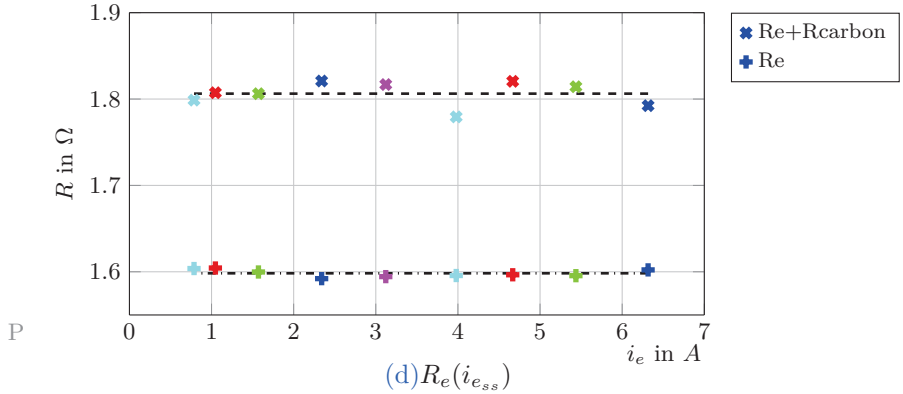
(a)$i_e(t)$



(b)$\Psi_{e,step}(i_e(t))$

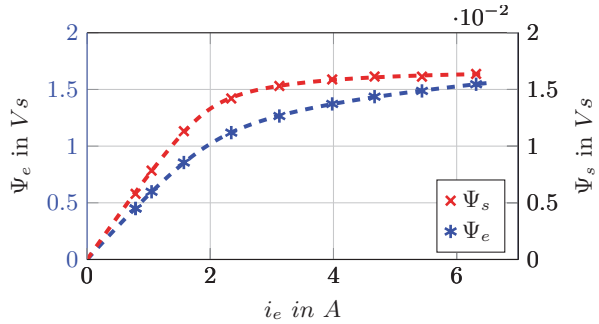


(c)$\Psi_s(i_e(t))$

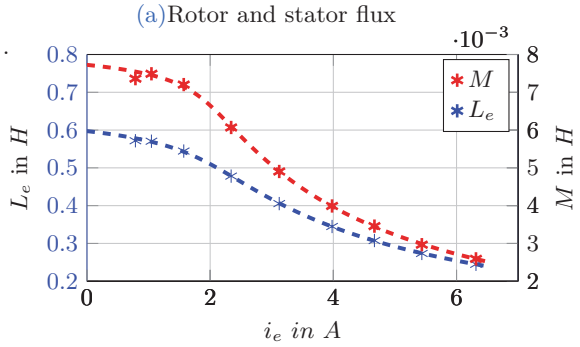


(d)$R_e(i_{e_{ss}})$

Figure 5.4: Measurements and determined results of the Standstill Test Denso SC6



(a)Rotor and stator flux



(b)Excitation and mutual inductance

Figure 5.5: Magnetization curve and inductances *Denso SC6*

5.1.2 Simulation

In section 4.4.1, a model was used to simulate the trace of $\Psi_e(i_e(t))$ (see figure 4.14). For the simulation of the standstill test, this model is modified. The point is that remanence is not considered. In figure 5.3c it can be seen that the flux does not return to zero after the voltage step. The remaining flux is the remanence $\Psi_{e,0}$. In order to simulate the standstill test more correctly, two look up tables of $\Psi_e(i_e)$ are used. One is for the rise of $\Psi_e(t)$ the other one is for the decrease. The rising look up table contains $\Psi_e(i_e)$ as seen in figure 5.5a. The falling one contains scaled data:

$$\Psi_{e,rise}(i_e) = \Psi_e(i_e) \tag{5.13}$$

$$scale = \frac{\Psi_{e,step_{ss}}(i_{ess}) - \Psi_{e_0}}{\Psi_{e,step_{ss}}} \tag{5.14}$$

$$\Psi_{e,fall}(i_e) = \Psi_{e,step_{ss}}(1 - scale) + \Psi_e(i_e) \cdot scale \tag{5.15}$$

At the falling edge of u_e, it is switched from the look up table $\Psi_{e,rise}(i_e)$ to $\Psi_{e,fall}(i_e)$. The same is done for the stator flux $\Psi_s(i_e)$. The model is presented in figure 5.6.
Simulated and measured traces are shown in figure 5.7 and 5.8. A comparison is given in table 5.1. Column t_{90} is the time span from applying the voltage step till 90% of $\Psi_{s_{ss}}$ are reached. Column t_{10} is the time span from the falling edge of u_e till 10% of the delta $\Psi_{e_{ss}} - \Psi_{e_{end}}$ are reached. Δt_{90} and Δt_{10} is the difference between measurement and simulation.

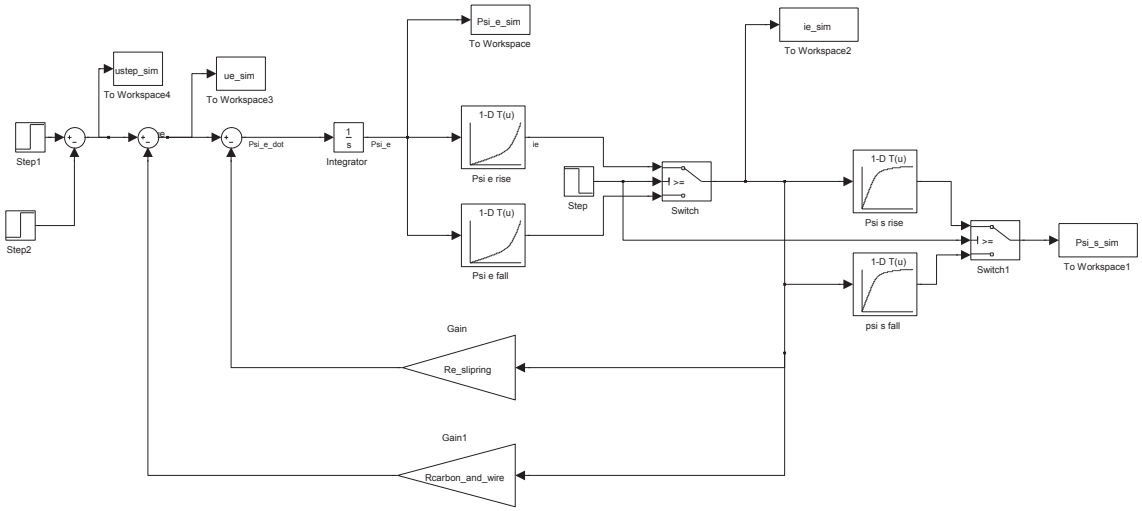


Figure 5.6: Simulink model standstill test, Denso SC6

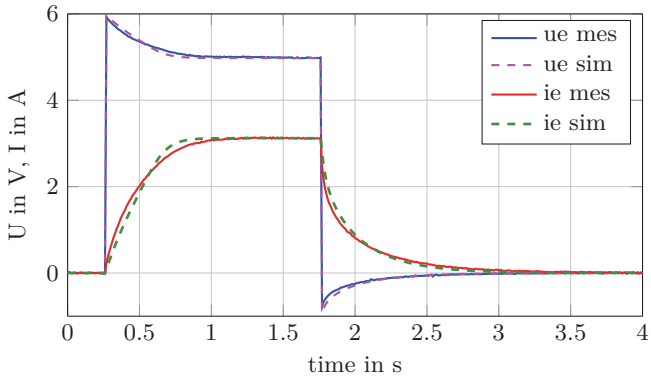


Figure 5.7: Simulate and measured traces of $i_e(t)$ and $u_e(t)$

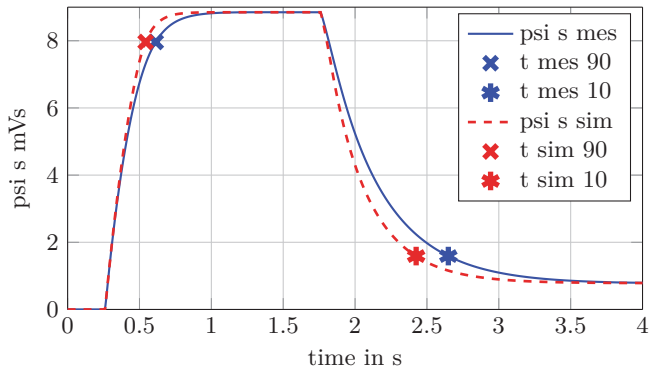


Figure 5.8: Simulate and measured traces of the stator flux $\Psi_s(t)$

$u_{e_{ss}}$ V	$t_{90,meas}$ ms	$t_{90,sim}$ ms	Δt_{90} ms	$t_{10,meas}$ ms	$t_{10,sim}$ ms	Δt_{10} ms
12	155	126	29	849	687	162
9	215	174	41	895	673	221
6	352	282	70	887	663	224
3	692	604	88	834	657	177

Table 5.1: Comparison standstill test

5.1.3 Conclusion

In figure 5.8 it can be seen that the measured as well as the simulated traces of Ψ_e increase and decrease with different gradients. Thus, the values in column t_{90} and t_{10} are different. The cause is saturation of the iron. Considering the simulated data, it can be seen that this effect is declining with decreasing amplitude values of $u_{e_{ss}}$. This is expected, since saturation is declining.

The simulated trace of Ψ_e increases faster than the measured trace. This is expected as well, because eddy currents oppose field changes. Consequently, the resistive component is dominating at the beginning, thus the measured $i_e(t)$ rises faster (see figure 5.7). As a result, the term $u_e(\tau) - R_e i_e(\tau)$ in equation 5.4 is lower. The integral is growing slower, which means Ψ_e and Ψ_s are growing slower. This effect has a positive consequence on the threshold calculations. This is explained in section 4.4.2.
On the other hand, the decrease is slower than the simulated. This has a negative effect on the calulation of $\Delta U(n, \dot{n})$ (see section 4.4.1).

5.2 PWM Measurements

In order to examine if eddy currents have an effect on the handover value of the ECC, different PWM experiments are conducted. Basically, the same set-up as in section 5.1.1 is used. Only the control of the power inverter is changed.

5.2.1 Frequency Sweep

For a fixed DC value, a frequency sweep is conducted at stand still. The frequency is stepwise changed and ranges from $20Hz$ to $200Hz$ (amplitude $14V$). There are always five periods of the same frequency. An overview of a measurement with $DC = 30\%$ is presented in figure 5.9. For each PWM period, the average current is calculated $\bar{i}_{e,k}$, as well the mean value over all $\bar{i}_{e,all}$. Furthermore, the minimum, maximum and the standard deviation s

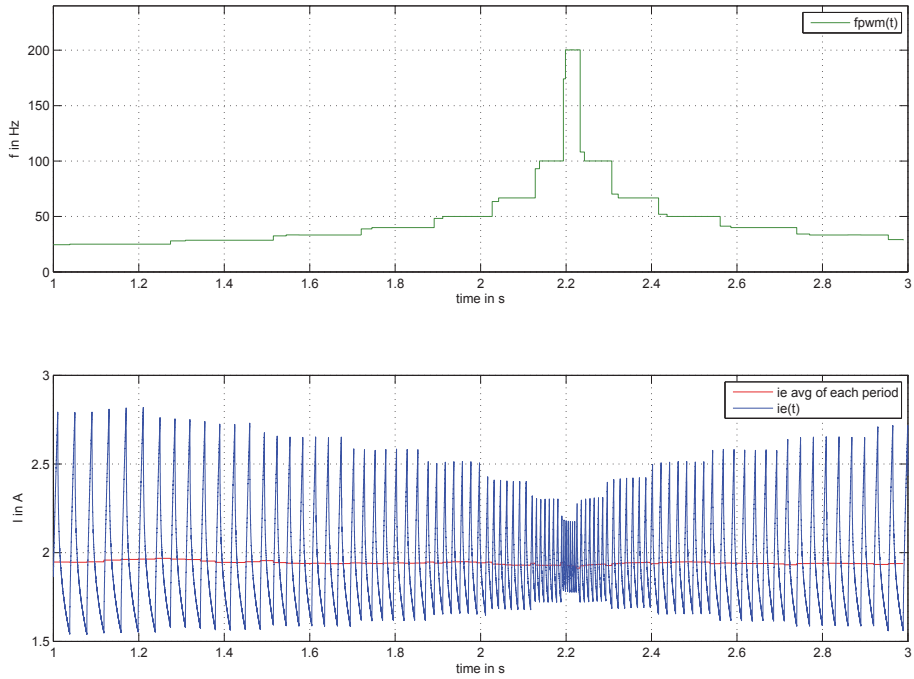


Figure 5.9: Frequency sweep $20Hz - 200Hz$, $DC = 30$

are determined. The same frequency sweep is applied for different DC values. Table 5.2 gives an overview of the results. Obviously, the fluctuation of $\bar{i}_{e,k}$ is very low. However, due to eddy currents, the trace of $i_e(t)$ can not be described with single time constant. This can be seen in figure 5.10 and 5.9.

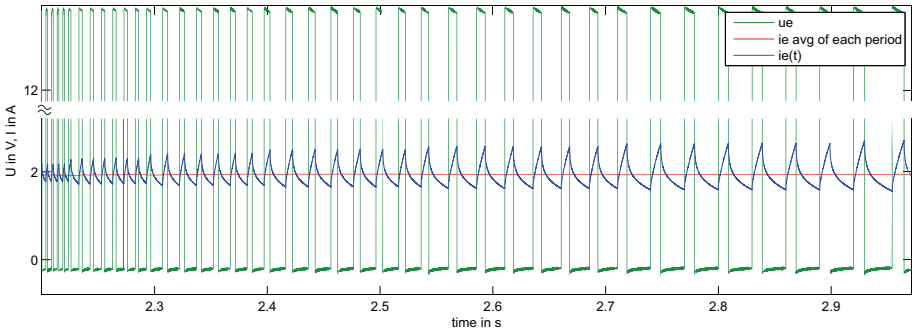


Figure 5.10: Zoom frequency sweep $20Hz - 200Hz$, $DC = 30$

DC	$\bar{i}_{e,all}$	$min(\bar{i}_{e,k})$	$max(\bar{i}_{e,k})$	$s(\bar{i}_{e,k})$
%	A	A	A	mA
15	0.83	0.81	0.84	4.86
30	1.94	1.91	1.97	9.50
43	2.83	2.80	2.85	8.03
57	3.80	3.77	3.85	13.23
70	4.78	4.75	4.82	14.53
85	5.98	5.96	6.02	12.92

Table 5.2: Current fluctuation frequency sweep

5.2.2 Duty Cycle Sweep

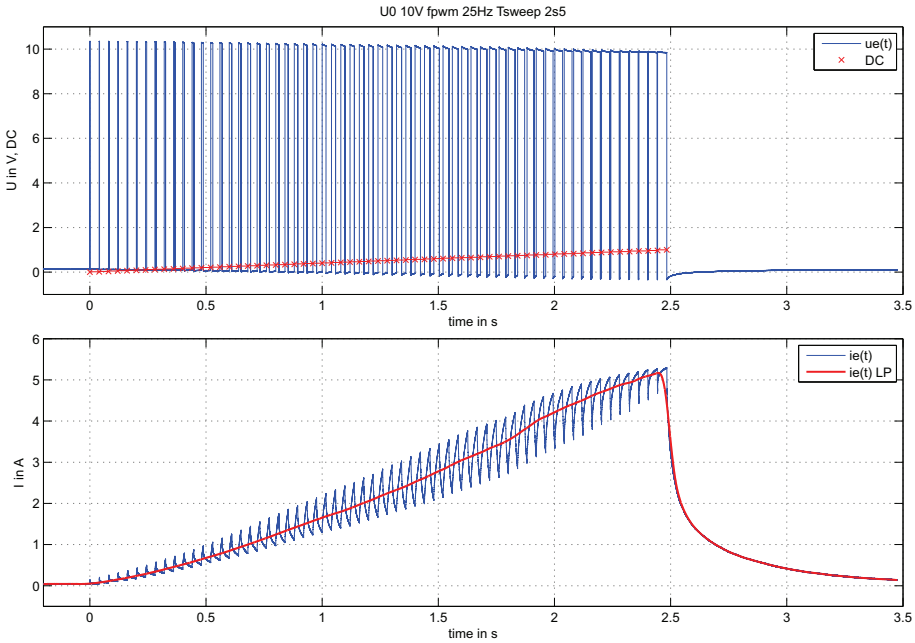


Figure 5.11: PWM duty cycle sweep

The same measurements as before are done, but the other way around. For a fixed frequency, the DC value sweeps. Two different experiments are conducted. The first one increases the DC-value from 0% to 100% within $1s$, the other one within $2.5s$. Each is done for a fixed PWM frequency of $25Hz$, $55Hz$, $110Hz$, and $220Hz$. The upper plot of figure 5.11 shows the PWM sequence of a single sweep. The lower plot shows the measured $i_e(t)$ (blue), the red trace is i_e(t) low pass filtered. The filter is adopted for each PWM frequency. The results of all measurements are given in figure 5.12 and 5.13. It can be seen that the current traces of the same DC sweep are similar, although the PWM frequency varies.

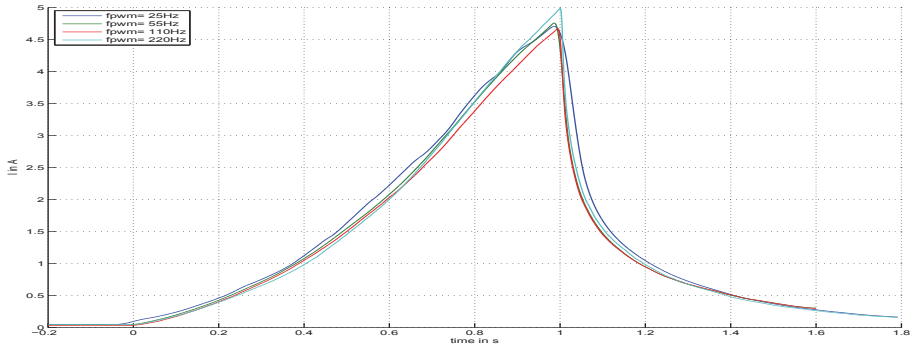


Figure 5.12: Low pass filtered current traces, sweep time $1s$

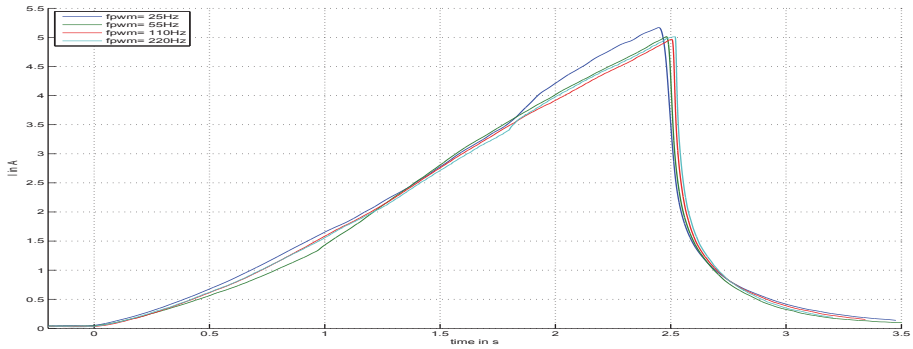


Figure 5.13: Low pass filtered current traces, sweep time $2s5$

5.2.3 Conclusion

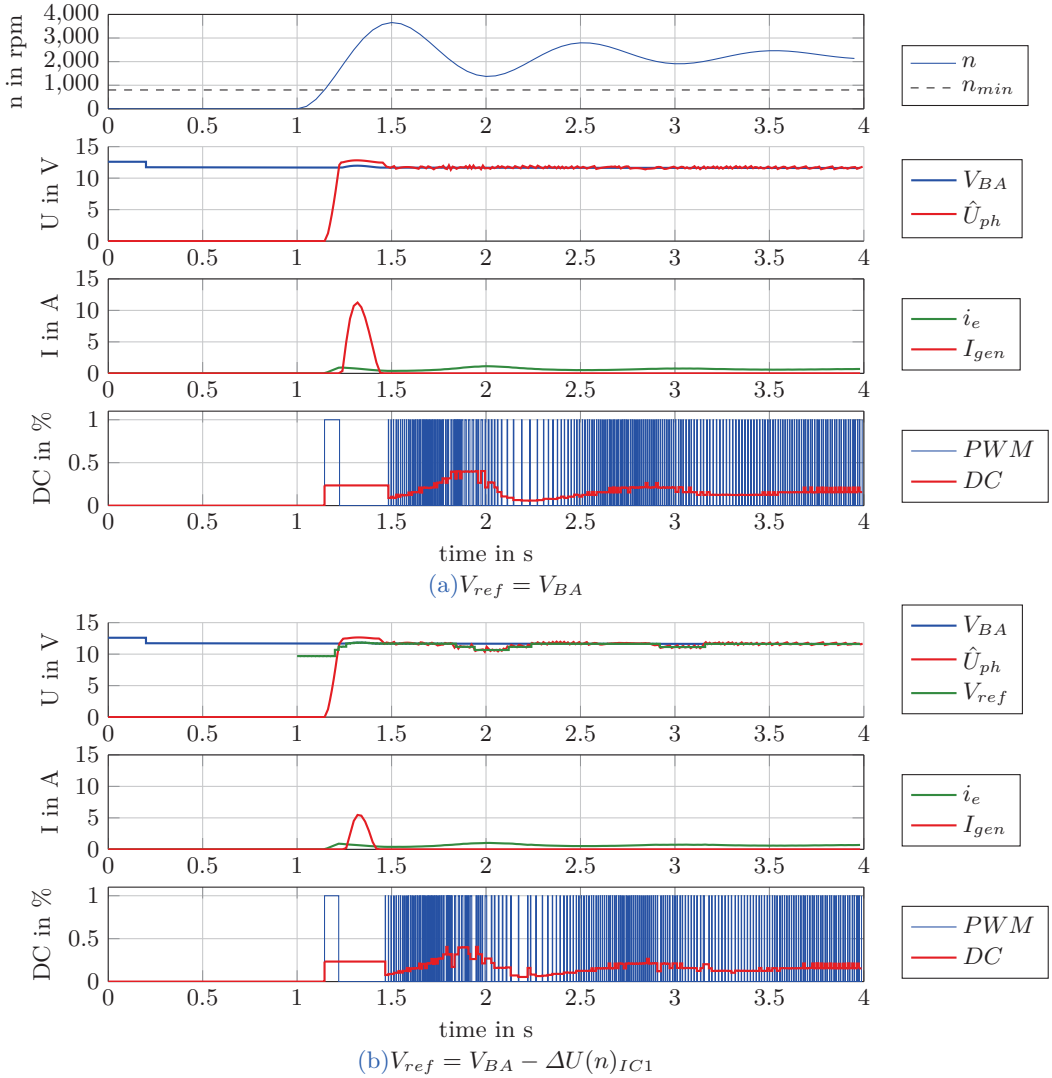
Neither the experiments in section 5.2.1, nor the duty cycle sweep have shown a significant dependency of the PWM frequency. The first case was done in order to examine the handover process at a constant rotor speed. Since the average current stays at the same value, it seems to be unproblematic.
In case of a dynamic speed trace, the DC is changing all the time. However, the duty cycle sweep experiments have shown that current traces are more or less the same. Thus, it is assumed that the handover works correctly for dynamic cases as well.

6 Simulation Results

6.1 Phase Controller Only

6.1.1 Fast dynamic speed trace

In section 4.4.3 a simulation with the standard (dynamic) speed trace is shown. In this section, two simulations are given with an increased speed gradient: $\Delta t_{max} = 0.5s \Rightarrow$



(a)$V_{ref} = V_{BA}$

(b)$V_{ref} = V_{BA} - \Delta U(n)_{IC1}$

Figure 6.1: $\hat{U}_{ph}$ control, dynamic speed trace $\Delta t_{max} = 0.5s$

$\dot{n}_{max} = 11428^{rpm}/_{s}$. At first, the phase controller is configured with the default values, note $\Delta U(n) = 0V$. The simulation result is plotted in figure 6.1a. It can be seen that there is an output current overshoot with an amplitude of $12A$. The same simulation is repeated, but the phase controller is configured with $\Delta U(n)_{IC1}$ from section 4.4.2, see figure 6.1b. Since $\Delta U(n)_{IC1}$ is calculated for a maximum gradient of $7500^{rpm}/_{s}$, the overshoot can not be prevented, but the amplitude is reduced to $5A$. However, after the first speed maxima $t \geq t_{max}$ no further output current is produced, even if $\Delta U(n) = 0V$ for the rest of the trace. In other words, if the phase control starts at $t = t_{max}$, the default configuration $V_{ref} = V_{BA}$ will be sufficient.

6.1.2 Standard dynamic speed trace and switching load

Figure 6.2 shows a simulation with the standard speed trace and a switching load between $30A$ and $85A$. It can be seen that there are short output current spikes. The reason is explained in section 4.6.2. DC_{BB} and the filtered DC traces are plotted in figure 6.3. Due to the abrupt changes of V_{BA}, DC_{BB} has to change promptly as well. Especially, when the load switches back to $30A$, the phase controller has to rise i_e because V_{BA} is increased. Thus, the DC value in the first moment is higher. However, the moving average and the exponential smoothing algorithm roughly filter these spikes. Also the error $e = DC(n(t), \dot{n}(t), V_{BA}(t) - DC_{HO}(t)$ is given in figure 6.3. $DC(n(t), \dot{n}(t), V_{BA}(t))$ is calculated on-line by the function block *Deep Thought*, which has as inputs the current speed, the current speed gradient and the current V_{BA}.

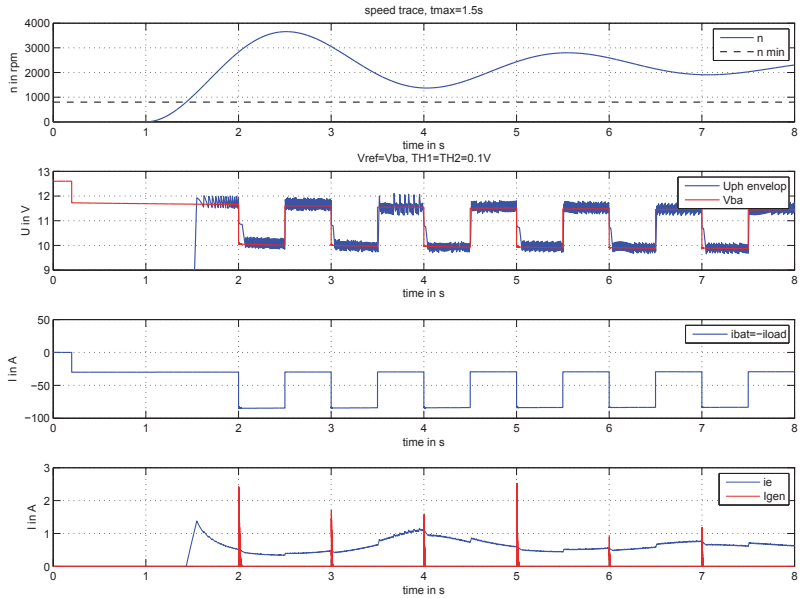


Figure 6.2: Simulation standard speed trace and switching load $85A \rightarrow 30A \rightarrow 85A \rightarrow 30A$...

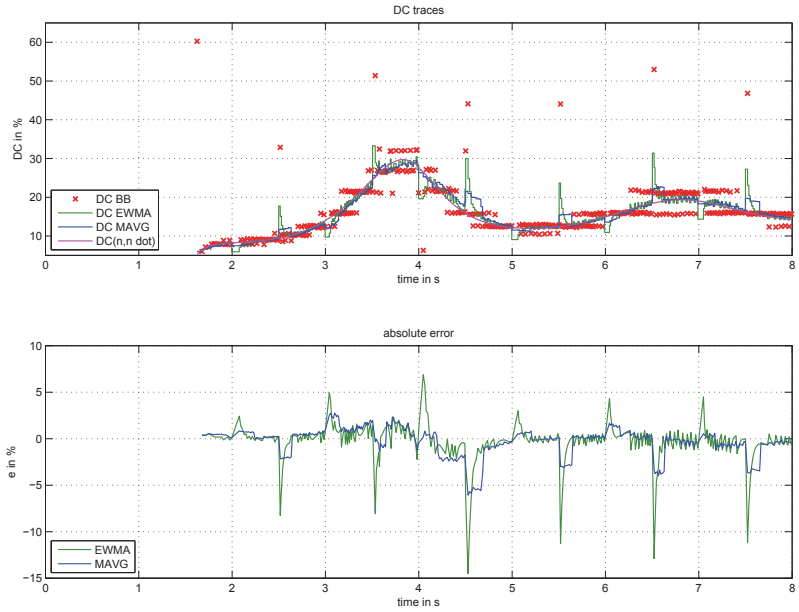


Figure 6.3: DC traces, standard speed trace and switching load $85A \rightarrow 30A \rightarrow 85A \rightarrow 30A$...

6.2 Start Up Simulations

6.2.1 Constant rotor speed

In order to evaluate the final ECC algorithm, start up simulations similar to the start up measurements in section 3.2 are conducted. Each simulation is done twice: at first ECC is disabled and then enabled.
Figure 6.4a shows a simulation with disabled ECC at $n = 2100rpm$. The LRC settings are $LRC - RT = 10s$ and $LRCT - BZ = 3\%$, also PSB is enabled. Thus, the phase voltage is kept around V_{PSB} by comparator till the DC of the LRC is large enough to keep $\hat{U}_{ph}$ above V_{PSB}. Since PSB does not influence the LRC ramp, the time span $t_0 - t_1$ or rather $t_0 - t_2$ is the same, whether PSB is enabled or not. Table 6.1 gives an overview of the timings for several $LRC - RT$ settings. Note that Δt refers always to t_0.
The same simulations are repeated with enabled ECC. The phase controller is initialized with the default parameters. As transfer options are chosen: handover as fast as possible $N > N_{min}$ with $N_{min} = 5$ and $DC_{HO} = DC_{MAVG}$ with $N_{avg} = 5$. Figure 6.4b shows the $LRC - RT = 10s$ simulations with enabled ECC and table 6.2 gives an overview of the new timings.

A comparison of table 6.1 and 6.2 shows that up to $2.4s$ can be saved by the ECC. However, the timings alone are not a sufficient evaluation. Any handover value higher than the current DC of the LRC ramp shortens the time span $t_0 - t_2$. Another criteria is the increase of I_{gen}. Actually, the aim of the LRC is a gently rising output current. In best case, the

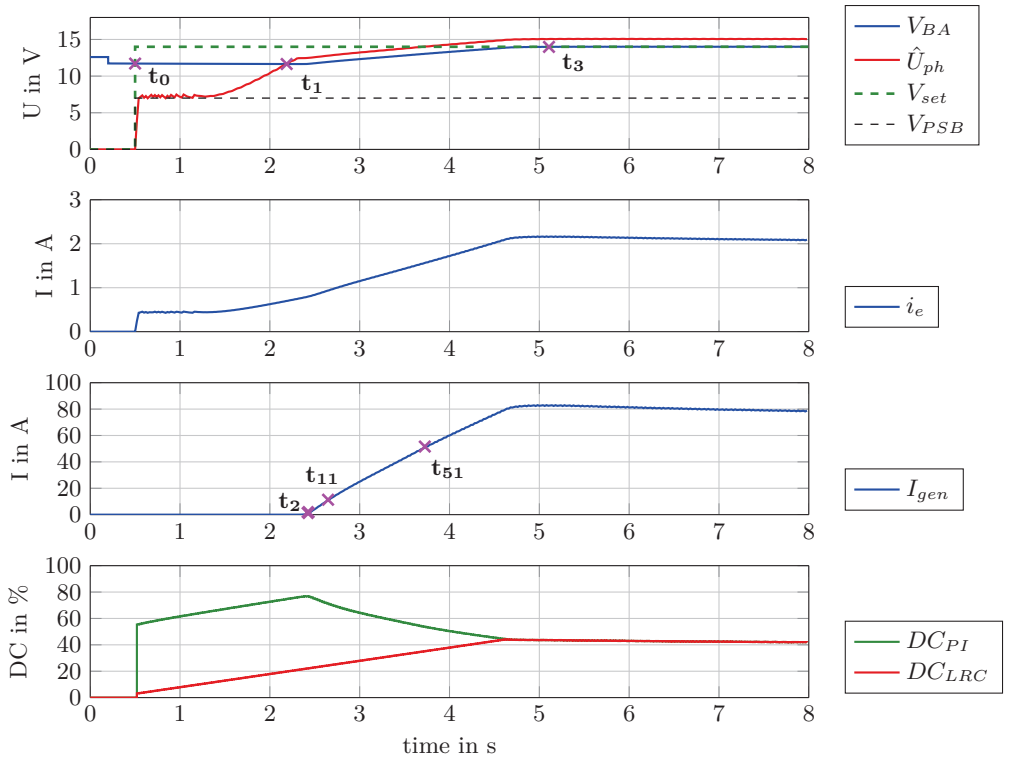
ECC minimizes the delay time $t_0 - t_2$ and the gradient of I_{gen} is the same gradient as in case of the disabled ECC. Thus, figure 6.5 gives a direct comparison of both, the delay time Δt_2 and the increase of I_{gen}. While figure 6.5a shows Δt_2 for different $LRC - RT$ and $n = 2100rpm$, Δt_2 for different speed values and fixed $LRC - RT = 10s$ is given in 6.5b. The average slope from $I_{gen} = 1A$ to $I_{gen} = 11A$ is plotted in figure 6.5c or rather 6.5d. Also, the average slope from $I_{gen} = 1A$ to $I_{gen} = 51A$ is given in figure 6.5e and 6.5f. It can be seen that $\frac{10A}{t_{11}-t_2}$ is increased for $LRC - RT = 10s$ and $LRC - RT = 15s$ if the ECC is enabled. The BLZ value (3%) boost the handover value $\Rightarrow DC_{HO} + LRC - BLZ$. The lower the LRC gradient, the more impact has the BLZ at the beginning.

$LRC - RT$ s	Δt_1 s	Δt_{HO} s	Δt_2 s	Δt_3 s	$DC_{LRC}(t_1)$ %	$DC_{LRC}(t_2)$ %	$DC_{LRC}(t_3)$ %
5	1.00	-	1.12	2.72	23.0	25.3	43.5
10	1.70	-	1.93	4.61	20.0	22.3	43.5
15	2.40	-	2.75	6.50	19.0	21.4	43.3

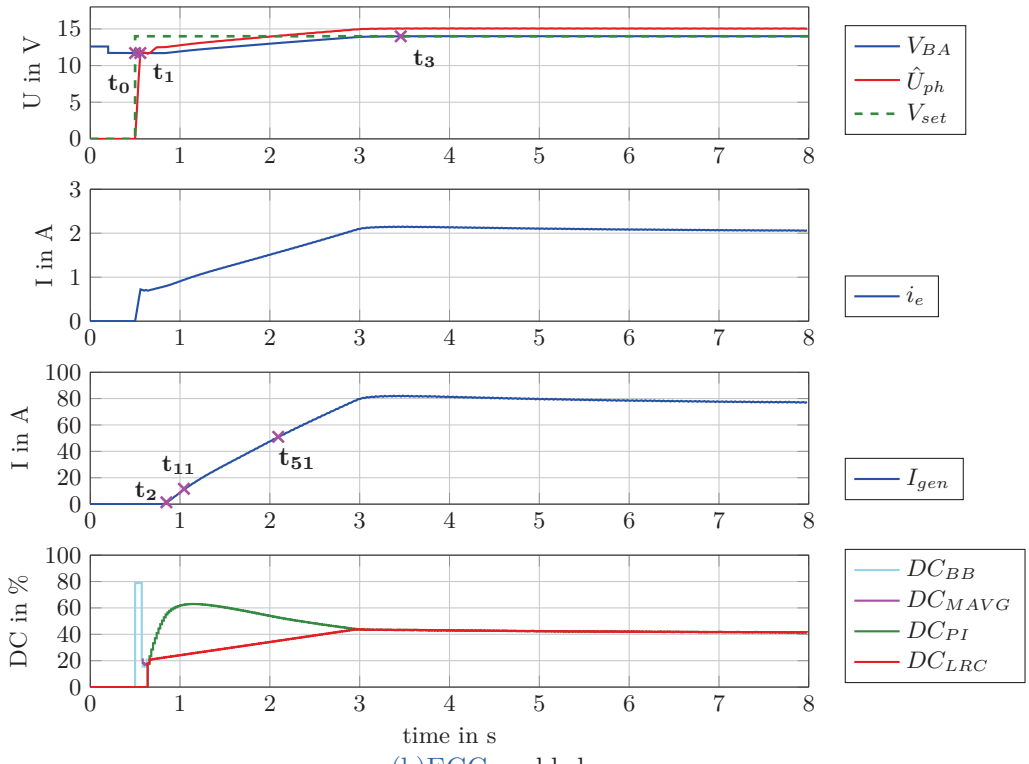
Table 6.1: Overview start up simulation $n = 2100rpm$, ECC disabled

$LRC - RT$ s	Δt_1 s	Δt_{HO} s	Δt_2 s	Δt_3 s	DC_{HO} %	$DC_{LRC}(t_2)$ %	$DC_{LRC}(t_3)$ %
5	0.06	0.14	0.32	2.01	17.8	24.1	43.3
10	0.06	0.14	0.35	2.96	17.8	22.7	43.1
15	0.06	0.14	0.37	3.95	17.8	22.2	42.9

Table 6.2: Overview start up simulation $n = 2100rpm$, ECC enabled



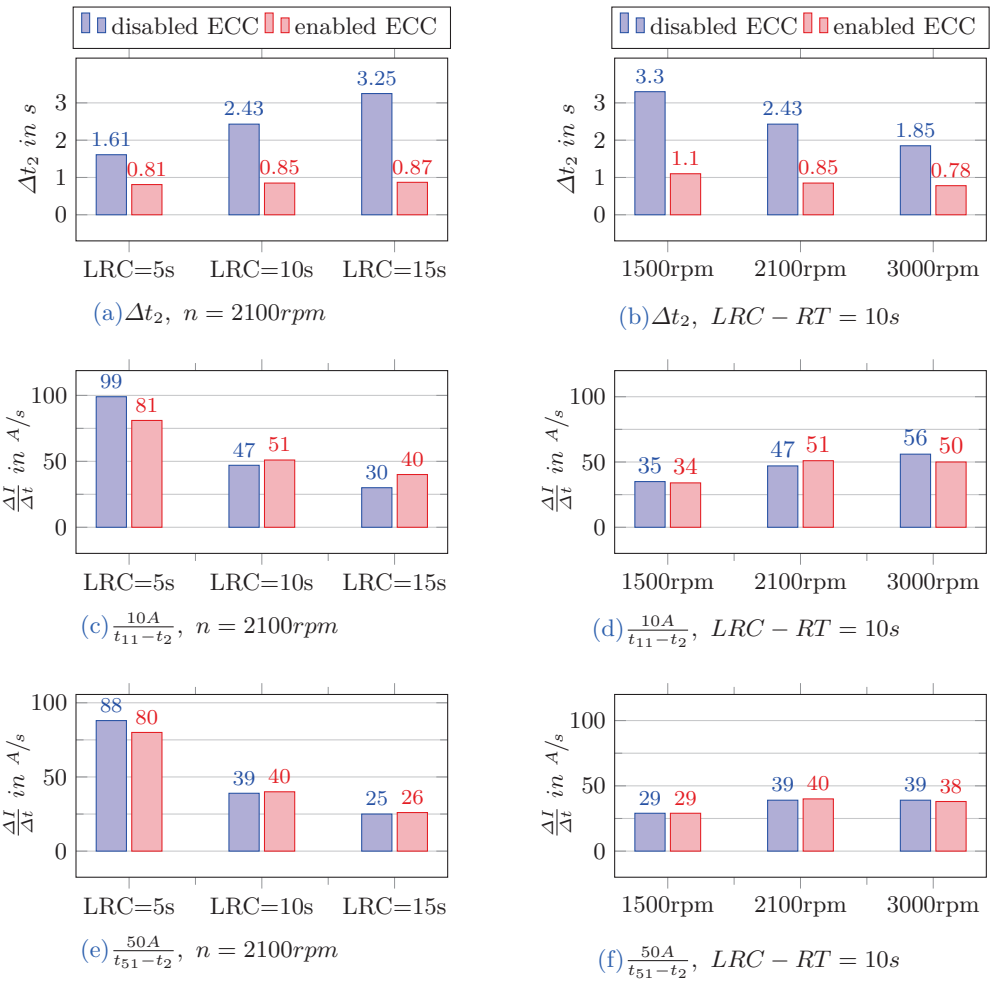
(a)ECC disabled



(b)ECC enabled

t_0 ... start of regulation
t_1 ... $\hat{U}_{ph} \geq V_{BA}$
t_3 ... $V_{BA} \geq V_{set}$
t_2 ... $I_{gen} \geq 1A$
t_{11} ... $I_{gen} \geq 11A$
t_{51} ... $I_{gen} \geq 51A$

Figure 6.4: Start up simulation $n = 2100rpm,\ LRC - RT = 10s, LRC - BLZ = 3\%$



(a) Δt_2, $n = 2100rpm$

(b) Δt_2, $LRC - RT = 10s$

(c) $\frac{10A}{t_{11}-t_2}$, $n = 2100rpm$

(d) $\frac{10A}{t_{11}-t_2}$, $LRC - RT = 10s$

(e) $\frac{50A}{t_{51}-t_2}$, $n = 2100rpm$

(f) $\frac{50A}{t_{51}-t_2}$, $LRC - RT = 10s$

Figure 6.5: Comparison start up simulation, n constant

6.2.2 Dynamic rotor speed

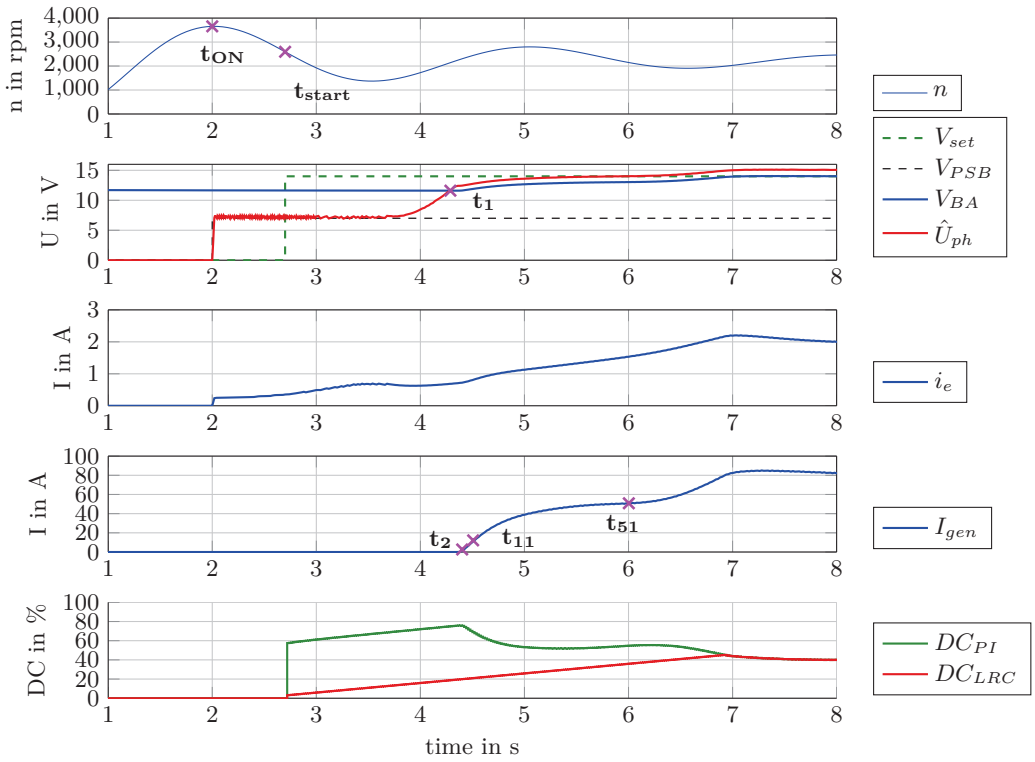
In case of a dynamic speed trace, the start point of the regulation is essential. Measurements from BMW show that the V_{BA} control is usually triggered more or less at the middle of the speed trace's first maxima and minima. In this thesis, three different start points are considered. The first one (t_A) is defined at the first negative maxima of $\dot{n}$, which matches with the BMW start point. The second one (t_B) is defined at the first minima of n ($\dot{n} = 0$) and the last one at the second positive maxima of $\dot{n}$. In figure 4.21 all three start points are marked.
Actually, it is desired that the output current begins to rise gently at these points in time. Since the normal V_{BA} controller starts the LRC ramp with relatively low BLZ-value, the point in time when I_{gen} rises will be delayed. However, the trigger command for the V_{BA} controller cannot be brought forward, because the delay time is unknown. Moreover, in case of a dynamic speed trace, the delay time is not constant. It could happen that I_{gen} rises too early (before t_A or rather $t_B,\ t_C$) if the start command is moved forward. An advantage of the ECC is that the $\hat{U}_{ph}$ control can be started at any point. The rise of I_{gen} depends on the handover command. Thus, the phase control is always triggered at the first maxima of the speed trace (t_{max}) and the handover command is set at t_A or rather $t_B,\ t_C$. However, t_0 refers always to the desired point in time for $I_{gen} > 0A$. t_2 is always the point in time when I_{gen} really begins to rise. $\Delta t_2 = t_2 - t_0$ is the unwanted delay time.

Simulations using the standard speed trace $\Delta t_{max} = 1.5s$ are presented in figure 6.6 and 6.7. The speed trace starts at $t = 0.5s$, thus gives the values $t_{max} = 2s,\ t_A = 2.7s,\ t_B = 3.54s$ and $t_C = 4.22s$. The settings are $LRC - RT = 10s,\ LRC - BLZ = 3\%$ and the $DC_{HO} = DC_{MAVG},\ N_{avg} = 5$ samples. Since the unweighed moving average is used, it is intended to handover the dynamic DC.
Figure 6.6a shows a simulation with disabled ECC and $t_0 = t_A = 2.7s$. It can be seen that the output current begins to increase at $t_2 = 4.4s$. With enabled ECC, t_2 is reduced to $3.77s$. The saved time is about $0.6s$. In case of $t_0 = t_B = 3.54s$ and disabled ECC, t_2 is given by $t_2 = 4.94s$. If the ECC is enabled, t_2 is moved forward to $t_2 = 3.72s$. The difference is $1.22s$. Furthermore, $t_2(t_0 = t_B)$ is a bit lower than $t_2(t_0 = t_A)$ if ECC is enabled. Indeed, the later handover command at $t_B = 3.54s$ results in an earlier rise of the output current. The point is that the ECC transfers the current DC value to keep $\hat{U}_{ph}$ around V_{bat}. In case of $t_0 = t_{HO} = t_A$ the handover value is $DC_{HO} = 16.7\%$. The theoretical value is $DC(n, \dot{n}) = 17.4\%$, which means the ECC works as defined. In figure 6.7b it can be seen that $\hat{U}_{ph}$ actually increases immediately after the handover, because the $LRC - BLZ$ boosts DC_{HO} a little bit ($DC_{HO} + BLZ \approx 20\%$). Nevertheless, $\hat{U}_{ph}$ is declining a short moment later. The problem is that the speed is still decreasing. Low revolutions require by nature a higher DC to keep $\hat{U}_{ph}$ and the speed gradient is negative, too. However, after the handover, the DC can only increase with the LRC gradient, which is obviously too low to compensate the falling speed trace. Thus, $\hat{U}_{ph}$ sinks till the speed minima (=t_B) is reached.

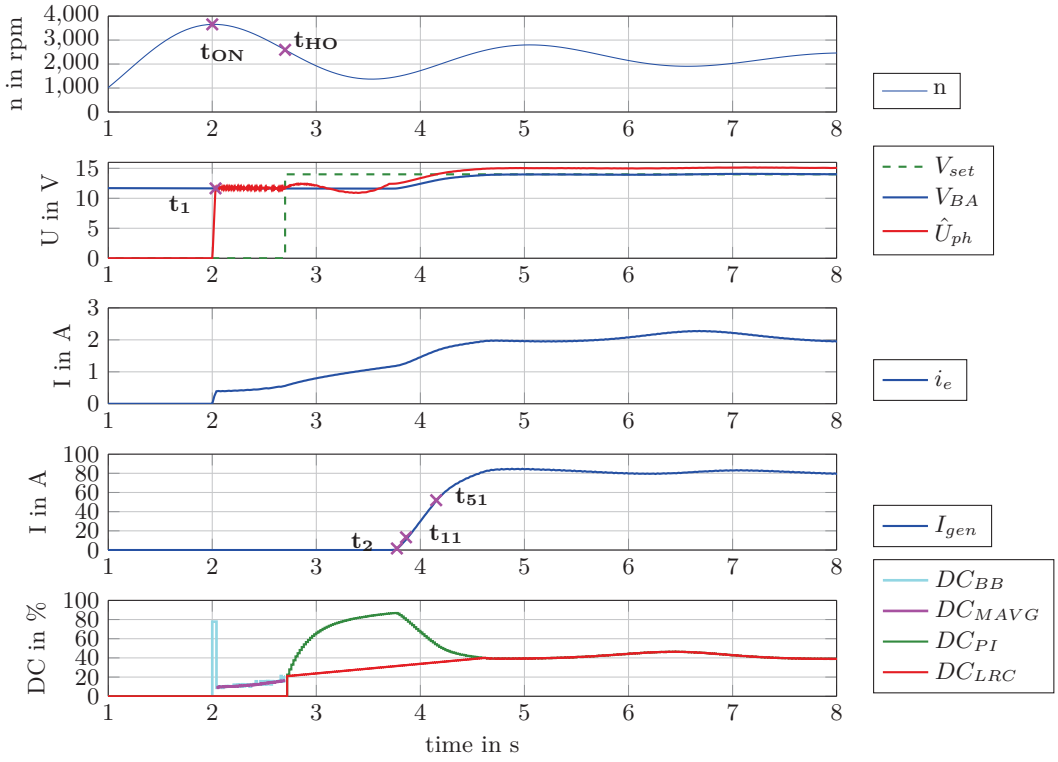
Consequently, the time span Δt_2 increases. The static DC at t_A is $DC(n, \dot{n} = 0) = 13.4\%$, which would result in an even larger time span.
On the contrary, if $t_0 = t_{HO} = t_B$ like in figure 6.7b, the handover value is relative high $DC_{HO} = 28.4\%$ because of the speed minima. I_{gen} starts to rise $0.14s$ after the handover command. The saved time compared to disabled ECC (see figure 6.7a) is about $1.3s$. The simulation with $t_0 = t_C$ looks basically similar to figure 6.7, therefore it is not illustrated here. A comparison of the delay time Δt_2 between simulations with disabled and enabled ECC for different $LRC - RT$ and t_0 is given in figure 6.8a, 6.8c and 6.8e. The rise gradient of I_{gen} is not constant due to the dynamic speed trace, see figure 6.6 and 6.7. Since I_{gen} starts to rise at different points in time, a comparison of the gradient is not conclusive.

Note, in figure 6.6a it can be seen that i_e stops to rise at $t \approx 3.8s$, although the DC of the LRC ramp is increasing. The reason is that PSB was active before. Thus, the recent excitation current is higher than the excitation current which would be produced by the recent DC value of the LRC ramp. At $t = 3.8s$ PSB tries to reduce i_e because the speed is increasing. Consequently, PSB wants to set $PWM = 0$, but the excitation output stage is logical linked with DC of the V_{BA} controller (see 2.4). However, the DC of the LRC ramp is too low to increase i_e, but prevents declining. At $t \approx 4.2$ the DC is large enough and i_e is rising again.

Further simulations with a speed trace of $\Delta t_{max} = 0.5s$ are conducted. A comparison using the theoretical static DC and the theoretical dynamic DC (determined by *Deep Thought*) as handover value are given in the right column of figure 6.8. Since, it is supposed that the rise of I_{gen} begins more or less at the same time, also the average current slope $\frac{10A}{t_{11}-t_2}$ is plotted (see figure 6.8f). As explained before, the static DC is lower than the dynamic DC at t_A. Since the speed at t_A, t_B and t_C is always the same value, the static DC stays the same for any variation of Δt_{max}. The dynamic DC changes according to the gradient. The lower Δt_{max}, the greater is the difference of DC_{stat} and DC_{dyn}. Notwithstanding, $DC_{HO}(t_B) = DC_{dyn} = DC_{stat}$ because the speed gradient is zero at t_B. Since the speed is increasing at t_C, the dynamic DC is lower than the static DC. Consequently, the time span Δt_2 is greater if $DC_{HO}(t_C) = DC_{dyn}$. In case of $DC_{HO}(t_C) = DC_{stat}$, the increase of I_{gen} starts $0.05s$ after the handover command, but it can also be seen that the rise gradient is very high. For these reasons, it is recommended to handover the dynamic DC in general. The moving average with a short sample length (e.g.: $N_{avg} = 5$) or the exponential smoothing are qualified to determine the handover value.

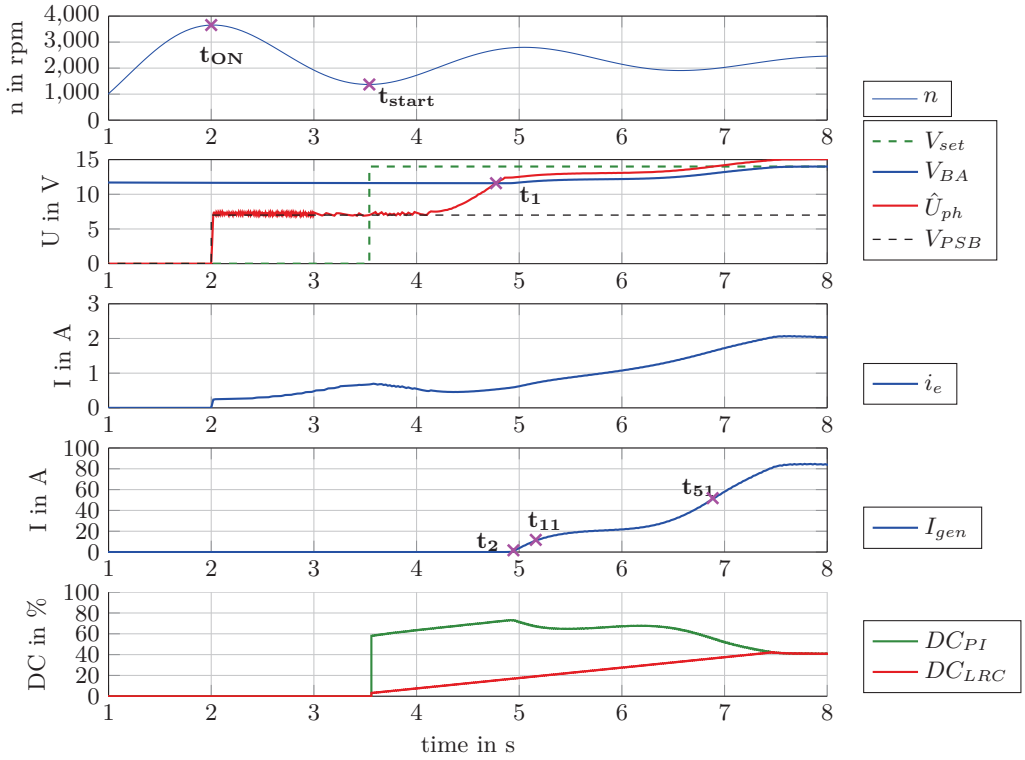


(a)ECC disabled, $t_{ON} = t_{max},\ t_0 = t_{start} = t_A$

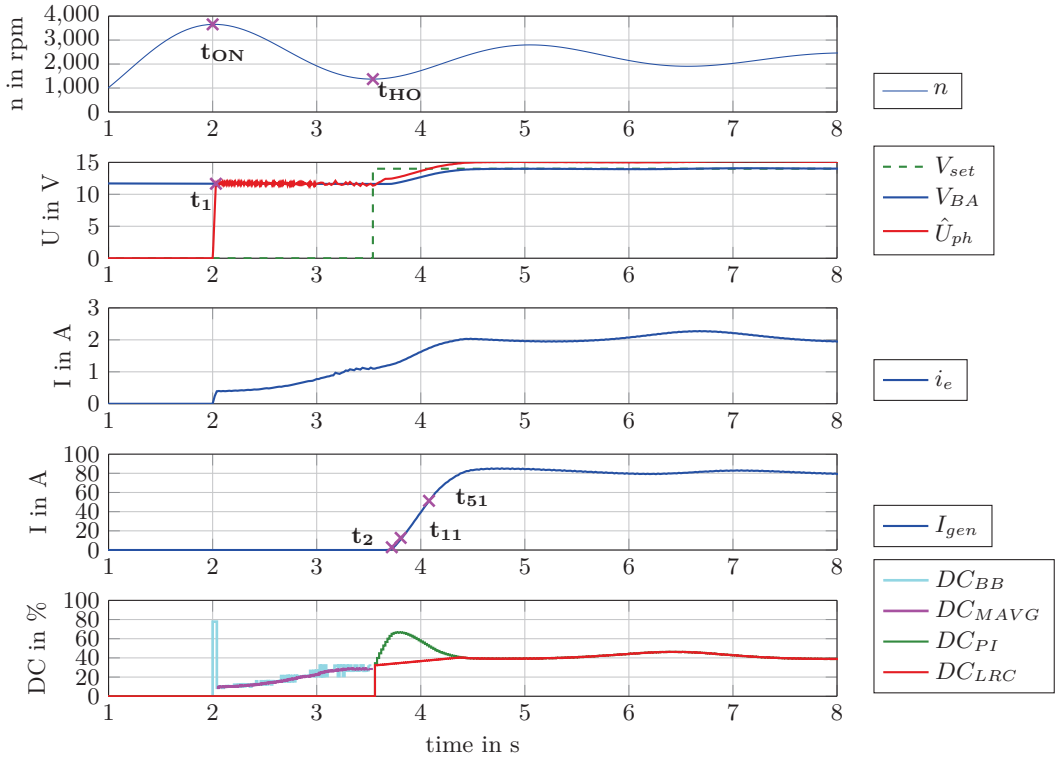


(b)ECC enabled, $t_{start} = t_{ON} = t_{max},\ t_0 = t_{HO} = t_A$

Figure 6.6: Start up simulation dynamic speed trace $\Delta t_{max} = 1.5s,\ t_0 = t_A = 2.7s$ engine start at $t = 0.5s,\ LRC - RT = 10s,\ LRC - BLZ = 3\%,\ DC_HO = DC_{MAVG},\ N_{avg} = 5$

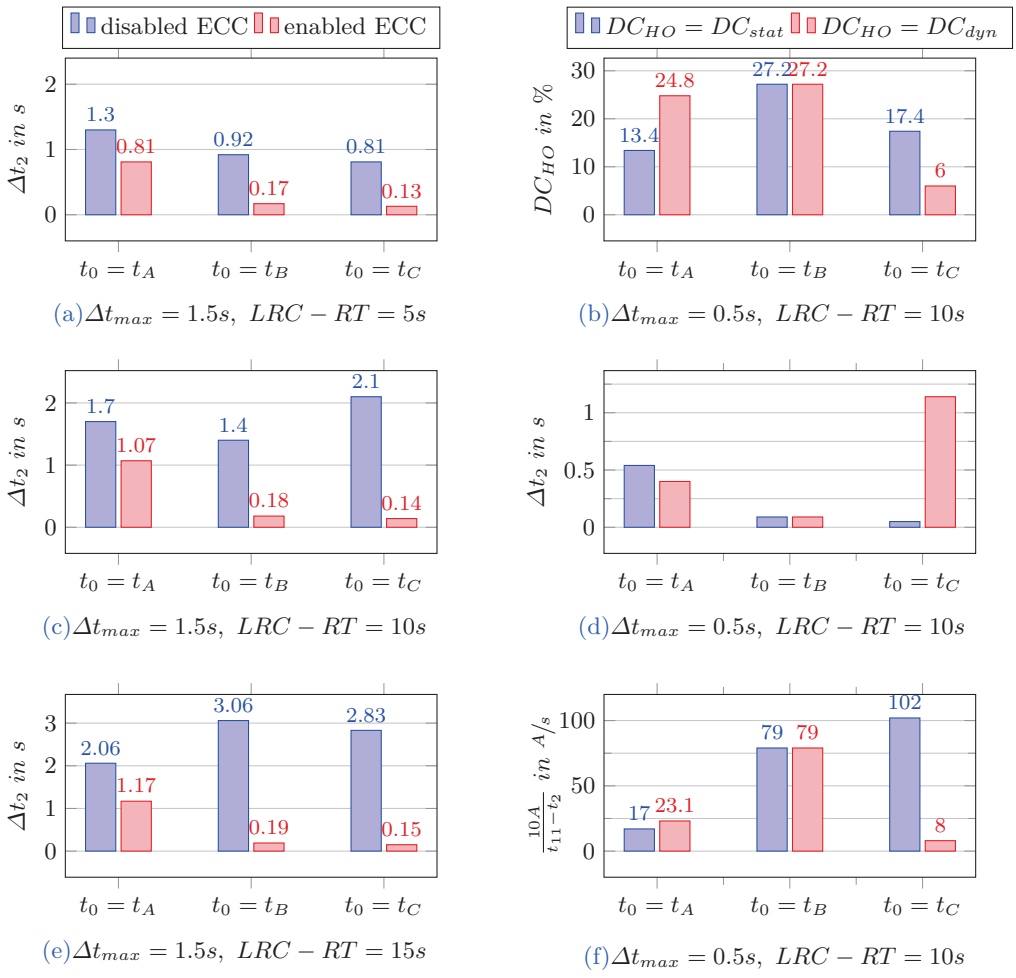


(a)ECC disabled, $t_{ON} = t_{max},\ t_0 = t_{start} = t_A$



(b)ECC enabled, $t_{start} = t_{ON} = t_{max},\ t_0 = t_{HO} = t_B$

Figure 6.7: Start up simulation dynamic speed trace $\Delta t_{max} = 1.5s,\ t_0 = t_B = 3.54s$ engine start at $t = 0.5s,\ LRC - RT = 10s,\ LRC - BLZ = 3\%$



(a) $\Delta t_{max} = 1.5s,\ LRC - RT = 5s$

(b) $\Delta t_{max} = 0.5s,\ LRC - RT = 10s$

(c) $\Delta t_{max} = 1.5s,\ LRC - RT = 10s$

(d) $\Delta t_{max} = 0.5s,\ LRC - RT = 10s$

(e) $\Delta t_{max} = 1.5s,\ LRC - RT = 15s$

(f) $\Delta t_{max} = 0.5s,\ LRC - RT = 10s$

Figure 6.8: Comparison start up simulation, dynamic speed trace

7 Final Conclusion

Investigations in section 3.1 have shown that the battery will be charged if the induced delta voltage $\hat{U}_s \geq U_{s,crit}$, with $U_{s,crit} = V_{BA_0} + 2U_f$. Straight forward the critical value referred to ground is defined by $U_{ph,crit} = \hat{U}_{s,crit} - U_f$. The IC uses a pull down resistor to measure $U_{ph} \Rightarrow \hat{U}_{ph,measured} = \hat{U}_s + U_{shift}$. Alternator measurements have shown that there is a positive ground shift $U_{shift} > 0V$ if the charging condition is not fulfilled. Thus, the estimation of U_s by $U_s(t) = \hat{U}_{ph,measured}(t) + U_f$ is too conservative in general. However, the model implementation of the rectifier causes a shift of $U_{shift} = -U_f$ even if the charging condition is not fulfilled. Thus, $\hat{U}_{ph,sim}(t) = \hat{U}_{ph,crit} - x$ results in $\hat{U}_{s,sim}(t) = \hat{U}_{s,crit} - x$, where x is an arbitrary value.
Further application measurements have shown that the start up situation is an appropriate use case for the patent idea, see section 3.2. A three-step-concept is developed in order to minimize the delay time Δt_2: 1) fast increase of the excitation field 2) determination of DC handover value 3) transfer and manipulation of V_{BA} controller (section 4.1). For this purpose, a simple bang-bang controller is designed: $V_{ref} - TH_1 < \hat{U}_{ph} < V_{ref} + TH_2$, $\bar{\hat{U}}_{ph} = \hat{U}_{ph,target} = V_{ref} = V_{BA}(t) + \Delta U$. ΔU defines the distance to the critical value and is adjustable by the end-user. Note $x = -U_f + \Delta U$. At low speeds and strong positive speed gradients, unwanted output current could be produced if the safety distance x is too low, see figure 4.13 in section 4.4. However, simulations using a dynamic speed trace based on measurements from BMW ($\Delta t_{max} = 1.5s$) have shown that $\Delta U = 0V \Leftrightarrow x = -U_f$ is sufficient (figure 4.22). In case of more dynamic traces ($\Delta t_{max} < 1.5s$) it is recommended to trigger the phase control at or after the first speed maxima, see section 6.1. The relative threshold values $TH_1 = TH_2 = 0.1V$ are calculated assuming a worst case situation (section 4.4.2).
For the determination of the handover value two different possibilities are developed (section 4.5). The first one is based on filtering the DC values of the bang-bang PWM (DC_{BB}), the other one uses the excitation current measurement and calculates the DC value. Both have in common that several PWM periods are needed, which takes some time. In section 4.5.1 it is figured out that five periods is a good trade-off between time and accuracy. For the DC filtering two methods are implemented: a moving average and an exponential smoothing. In case of constant speed the absolute error of the handover value ($e = DC(n) - DC_{HO}$) is within 3% for both methods. If the current measurement option is used, it is less than 2% and for most of the speeds even less than 0.5% (figure 4.29). In case of a dynamic speed trace the error depends on the desired handover value. In order to transfer the static DC ($DC(n, \dot{n} = 0) = DC(n)$) the moving average is modified. The DC_{BB} samples are weighted with a factor proportional to the speed gradient. The average error per sample (J_1) is less than 2% for dynamic speed traces $\Delta t_{max} > 1s$, for $\Delta t_{max} = 0.5s$ it is about 4%. The current measurement option delivers much better results.

For all simulated speed traces the average error per sample is less than 0.5% (figure 4.35). If the dynamic DC ($DC(n, \dot{n})$) is desired as handover value, the unweighed DC_{BB} samples must be filtered. The normal moving average or the exponential smoothing algorithm can be used for that. Both give good results, the average error per sample is about 1% or less for all of the traces (figure 4.37).
In general it is reasonable to handover the dynamic DC since the V_{BA} controller has to deal with current speed gradient as well. The V_{BA} controller is a simplified behaviour model of the alternator IC, see section 4.7. It consists of a discrete PI controller and the LRC function. The conditioning technique of *Hanus* is used as anti wind-up mechanism. The PI controller is designed for a certain operation point (linearisation of the control process at $n = 2100rpm$). Finally, the handover is implemented by manipulating the integrator content according to the transferred DC value and the current error of the V_{BA} controller ($e_{PI} = V_{set} - V_{BA}$). At the same the LRC is forced to pass-through the DC of the V_{BA} controller for one iteration. A finite-state machine (ECC-Master, section 4.8.2) is used to trigger the bang-bang controller, the handover process and the V_{BA} controller. Via initialization script all relevant settings are adjustable.

Final simulations at constant speed have shown, that up to three seconds can be saved by the ECC, while the average rise gradient of I_{gen} stays more or less the same (see figure 6.5 in section 6.2.1). Certainly, the rotor speed and LRC settings are crucial factors. At the typical idle speed $n = 2100rpm$ and a moderate LRC gradient $LRC - RT = 10s$, the saved time is about $1.6s$. The rise gradient of the output current is a bit increased at the beginning ($51^{A}/_{s}$ compared to $47^{A}/_{s}$), but the average slope remains the same ($40^{A}/_{s}$ compared to $39^{A}/_{s}$). In case of dynamic speed trace the point in time of the handover is essential. In general, it is recommended to trigger the phase control $\hat{U}_{ph}$ at the first speed maxima ($t = t_{max}$). The most efficient point in time for the DC transfer (to the V_{BA} controller) is when the handover value has its maximum value. In case of the dynamic DC this is not necessarily at the speed minima, see figure 4.31. Nevertheless, the speed minima is a good transfer point, because the handover value is relative high and the speed will increase. The saved time for different transfer points and $LRC - RT$ can be seen in figure 6.8.

Appendix

Overview complete simulation and initialization file

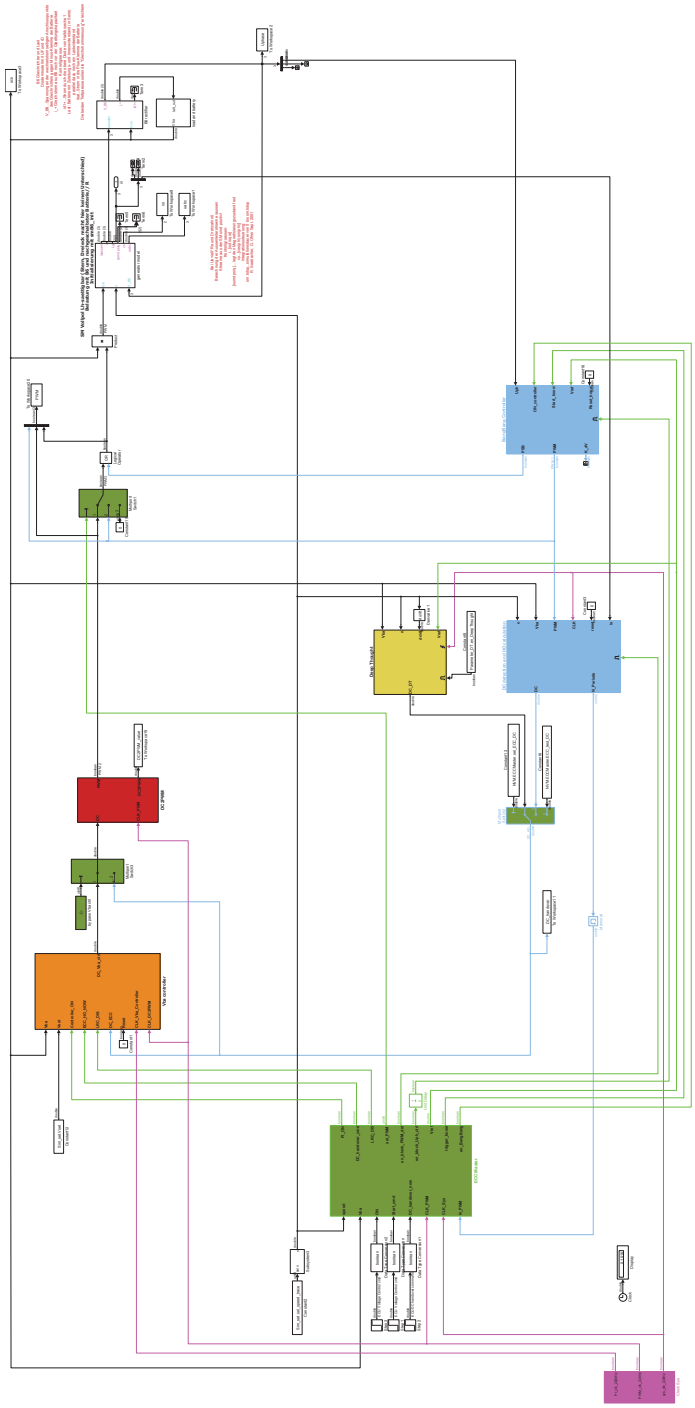


Figure .1: Overview Complete Simulation

The code of the initialization file is given at the following pages:

```
close all;
clear all;
clc;

%%

% Simulation Settings
% ------------------------------------------------------------------------
    Sim_set.t_sim=                  8;      % Simulation Time
    Sim_set.Vbat=                   12.6;   % Initial Battery Value
    Sim_set.Rbasic=                 0.39;   % value of the basic load
    Sim_set.t_Rbasic=               0.2;
    % time step when basic load will be connected
    Sim_set.Rpulsed=                0.12;
    % pulsed load (parallel to the basic load)
    Sim_set.t_Rpulsed=              50;
    % time step when pulsed load will be connected
    Sim_set.frequency_Rpulsed=      10;     % puls frequency
    Sim_set.sel_speed_trace=        2;      % select speed trace
                                            % 1...Constant speed or step
                                            % 2...Engine trace
                                            % 3...Ramp
                                            % 4...Sawtooth
                                            % 5...Sequence
    Sim_set.Vset=                   14;     % Vba setvalue in volt
    Sim_set.ON=                     2;      % time in sec
    Sim_set.Start_cmd=              Sim_set.ON;
    % starts Vba ctrl or rather Uph ctrl
    %Sim_set.handover_cmd=          Sim_set.Start_cmd;
    %DC handover as soon as N_PWM >= Nmin is satisfied (Nmin Master config)
    Sim_set.handover_cmd=           3.54;   % point in time of the handover


% Model Initialition Values
% ------------------------------------------------------------------------
    load('A14_biased.mat');         %load alternator data

    %sm=smB6_init_ue131115(M,Rlast,Rz,Rb,Ub,ie0,n0);
    sm= smB6_init_ue131115(M,0.39,0.0,0.03,Sim_set.Vbat,2,2000);

    %Integrator init value
    sm.psisRd0= 0;
    sm.psisRq0= 0;
    sm.psieRd0= 0;

    %(new) battery parameters
    load('x_opt_Bat4');
    x(4)=   x(4)*2;
    x(2)=   x(2)*1.3;
    x(1)=   x(1)*1.02;
    sm.Cs=  1.5e-6;
    sm.U0=  Sim_set.Vbat;
    sm.R1=  x(1);
    sm.C1=  x(2);
    sm.R2=  x(3);
```

```
    sm.C2=  x(4);
    sm.U02= x(5);
    sm.U01= sm.U0-x(5);

%current filter (Igen and Ie)
%-----------------------------------------------------------------------
fc=150;
wc=2*pi*fc;

% Clock System
% ----------------------------------------------------------------------
    f_PWM=220;
    CLK.f_PWM=  f_PWM;
    CLK.f_PI=   10*f_PWM;
    CLK.f_SYS=  100*f_PWM;

    CLK.Td_SYS= (1/CLK.f_SYS)/2;        %(1/(220*100))/2
    CLK.Td_PI=  (1/CLK.f_PI)/2;
    CLK.Td_PWM= (1/CLK.f_PWM)/2;


% ECC Master
% ----------------------------------------------------------------------
    NVM.ECCMaster.en_ECC=          true;
    % true... if ECU Start command == true   -> ECC starts Uph ctrl
    % false...if ECU Start command ==true    -> Vba ctrl starts immeditaly
    NVM.ECCMaster.en_PSB=           true;    % enable/disable PSB
    NVM.ECCMaster.deltaU=           0;
    % Vref= Vba+deltaU (deltaU independent of n), default value 0
    NVM.ECCMaster.n_min=            800;     % minimum alternator speed
    NVM.ECCMaster.Nmin=             5;
    %Nmin minimum number of PWM periods for handover
    NVM.ECCMaster.en_dynVref=       false;
    % false...Vref= Vba,
    % true...Vref= Vba - deltaU(n) for Uph ctrl
    NVM.ECCMaster.by_pass_Vba_ctrl= uint8(1);
    % 1... DC of the Vba controller is feed to DC2PWM (default)
    % 2... ECC DC is directly feed to DC2PWM (bypass Vba controller)

    NVM.ECCMaster.sel_ECC_DC=       2;
    % select DC handover value:
    % 1...calculated DC by DeepThought
    % 2...estimatet Comparator DC (default)
    % 3...test DC (ECC_test_DC)
    NVM.ECCMaster.ECC_test_DC=    0.4;       % any test DC


% Vba Controller
% ----------------------------------------------------------------------
    NVM.Vba_ctrl.dis_Vba_ctrl=          false;
    % enablbe/disable Vba Controller in general
    NVM.Vba_ctrl.LRC_en=                true;
    % LRC in general on/off (only possible if dis_Vba_ctrl= false)
    NVM.Vba_ctrl.sel_Output=            uint8(3);
    % select Output value
```

```
    % 1... DC_2200Hz (limited)
    % 2... DC_440Hz
    % 3... DC_LRC_440Hz (default)

    % PI Parameter
    % ------------------------------------------------
    %Ziegler:
    %Kp= 3.7;
    %T_I= 0.05;
    %Ts_PI= 1/(10*220);

    %FKL:
    %load('Matlab_files_only\010_Main\04_Vba_Controller\05_FKL\Rz_ue10')
    load('Matlab_files_only\010_Main\04_Vba_Controller\05_FKL\Rz_ue20.mat')
    [numR2z,denR2z,Td]=tfdata(R2z,'v');

    Kp=numR2z(1);               %R(z)=(b1*z+b0)/(z-1)
                                %b1=Kp
                                %b0=(KiTd-Kp)
    Ki=(numR2z(2)+Kp)/Td;
    T_I=Kp/Ki;
    Ts_PI=Td;

    NVM.Vba_ctrl.PI.Kp=                         Kp;
    NVM.Vba_ctrl.PI.T_I=                        T_I;
    NVM.Vba_ctrl.PI.Ts=                         1/CLK.f_PI;
    NVM.Vba_ctrl.PI.AW_sel=                     3;
    % select feedback for AntiWindup
    % 1...DC_PI_2200Hz
    % 2...DC_PI_440Hz
    % 3...DC_LRC_440Hz (should be standard, default)
    % any other value...DC_unlimited (not satured), no anti windup

    % LRC Parameter
    % ------------------------------------------------
        NVM.Vba_ctrl.LRC.Ts_LRC=        1/(2*CLK.f_PWM);
        %LRC is executed at rising AND falling edge of the PWM-Clock
        NVM.Vba_ctrl.LRC.BLZ=           3;   %in %
        NVM.Vba_ctrl.LRC.RT=            10;  %in sec
        NVM.Vba_ctrl.LRC.FT=            1;
        NVM.Vba_ctrl.LRC.en_BLZ_ECC=    true;
        %true..., in the next steps the LRC_DC is allowed
        %to jump up to DC_ECC+BLZ (if needed)
        %false... in the next step LRC_DC=DC_ECC+LRC_gradient*time, no jump

% Uphase Controller (Bang Bang)
% ------------------------------------------------------------------------
    TH= 0.1;
    NVM.Uph_ctrl.rel_TH_OFF=    TH;
    NVM.Uph_ctrl.rel_TH_ON=     TH;
    NVM.Uph_ctrl.dis_ON_Comp=   false;

% DC detection and handover calculation:
% ------------------------------------------------------------------------
    NVM.DC_cal_Parameter.CLK=           1/CLK.f_SYS;
```

```
    NVM.DC_cal_Parameter.N_mavg=        5;
    NVM.DC_cal_Parameter.alpha=         0.36;
    NVM.DC_cal_Parameter.k=             1.5e-4;
    NVM.DC_cal_Parameter.N_mavg_Re=     10;
    NVM.DC_cal_Parameter.sel_DC_alg =   uint8(3);
    % select DC Output
    % 1...DC_recent:    DC_BB, current DC, no filtering, no avering
    % 2...DCmavg:       (normal)moving Average with sample lenght = N_mavg
    % 3...DC_EMA:       exponential weight moving Average (alpha)
    % 4...DC_ie_Taylor: using ie measurement, Re est by taylor-> static DC
    % 5...DC_ie:        using ie measurement, Re estimation   -> static DC
    % 6...DC_recent_w:  weighted DC_BB samples, no filtering
    % 7...DC_wmag:      weighted moving average -> static DC
    % 8...DC_Ndyn:      unweighted mavg with dynamic length N_mavg

% Vba Input (LP second order)
%--------------------------------------------------------------------------
    Vba_Input.wc=2*pi*160;
    [Vba_Input.num_LP Vba_Input.den_LP]=butter(2,Vba_Input.wc,'s');
    Vba_Input.Ts=           1/2*NVM.Vba_ctrl.PI.Ts;
    Vba_Input.Resolution=   1/2^8;


% DC2PWM
% -------------------------------------------------------------------------
    DC2PWM.Resolution=  (2^8-1);
    DC2PWM.T_cnt=       (1/(2*f_PWM))/DC2PWM.Resolution;
    DC2PWM.CLK_delay=   1/CLK.f_SYS;


% Parameter DC Calculation ideal (Deep Thought)
% -------------------------------------------------------------------------
    Parameter_DT.en_DeepThought=   true;      % enable/disable Deep Thought
    Parameter_DT.sel_DC_alg_DT=    uint8(3);
    % select calculated DC of DeepThought:
    % 1...DCdyn_theo_Vref
    % 2...DCstat_theo_Vref
    % 3...DCdyn_theo_Vbat,
    % 4...DCstat_theo_Vbat
    Parameter_DT.p=                 M.nenn.p;
    Parameter_DT.Lg_vec=            M.leer.Lgerw;
    Parameter_DT.imuee_vec=         M.leer.ieerw;
    Parameter_DT.Uf=                M.B6.UF;
    Parameter_DT.Re=                M.parameter.Re;
    Parameter_DT.Ldiff_vec=         M.leer.Ldifferw;
    Parameter_DT.Lsig_e=            M.parameter.Lsige;
    Parameter_DT.ueV=               M.parameter.ue;


% Speed Traces
% -------------------------------------------------------------------------

    % Constant Speed (or speed step if wanted)
    Speed.Const.n0=              0;         %start value
    Speed.Const.n1=           2100;         %end values
```

```
    Speed.Const.tstep=          0;          %if tstep=0 ->starts immedatly

    % Engine trace:
    Speed.Engine.n_max =  1200;             %engine speed max
    Speed.Engine.n_stat=  750;              %Alternator : n_stat = 2100 rpm
    Speed.Engine.Ts=      1.5;              %speed max is reached @Ts
    Speed.Engine.ratio=     3;              %n_altnator = ratio * n_engine
    Speed.Engine.t_start=   0.5;            %Engine start time
    [Speed.Engine.num Speed.Engine.den ]= ...
    n_trace_TF( Speed.Engine.n_max,Speed.Engine.n_stat, Speed.Engine.Ts);

    % Speed Ramp(s):
    Speed.Ramp.Init_Ramp1=  1000;           %starts with 1000rpms
    Speed.Ramp.t_Ramp1=     1;
    Speed.Ramp.slope_Ramp1= 6000;           %after 1s dn/dt 6000rpms/s
    Speed.Ramp.slope_Ramp2= -6000;
    Speed.Ramp.t_Ramp2=     1+2/6;          %after 2/6s constant speed
    Speed.Ramp.Init_Ramp2=  0;
    Speed.Ramp.t_Ramp3=     1+2/6+1;
    Speed.Ramp.slope_Ramp3= -6000;          %after 1s dn/dt -6000rms/s
    Speed.Ramp.Init_Ramp3=  0;
    Speed.Ramp.t_Ramp4=     1+2/6+1+2/6;
    Speed.Ramp.slope_Ramp4= 6000;           %after 2/6s constant speed
    Speed.Ramp.Init_Ramp4=  0;

    % Repeating Sawthooth
    Speed.Sawthooth.Period= 1;
    Speed.Sawthooth.n_min=  2000-100;
    Speed.Sawthooth.n_max=  2000+100;

    % Repeating Sequence
    Speed.Sequence.Values= [3 1 4 2 1].'*1000;
    Speed.Sequence.Timing= [0 0.1 0.5 0.6 1].';
    Speed.Sequence.Ts=      0.01;



%%  start simulation
    sim('Complete_Simulation_final.mdl')


%% figures:
```

Zoom $\hat{U}_{ph}$ controller, DC detection and *DeepThought*

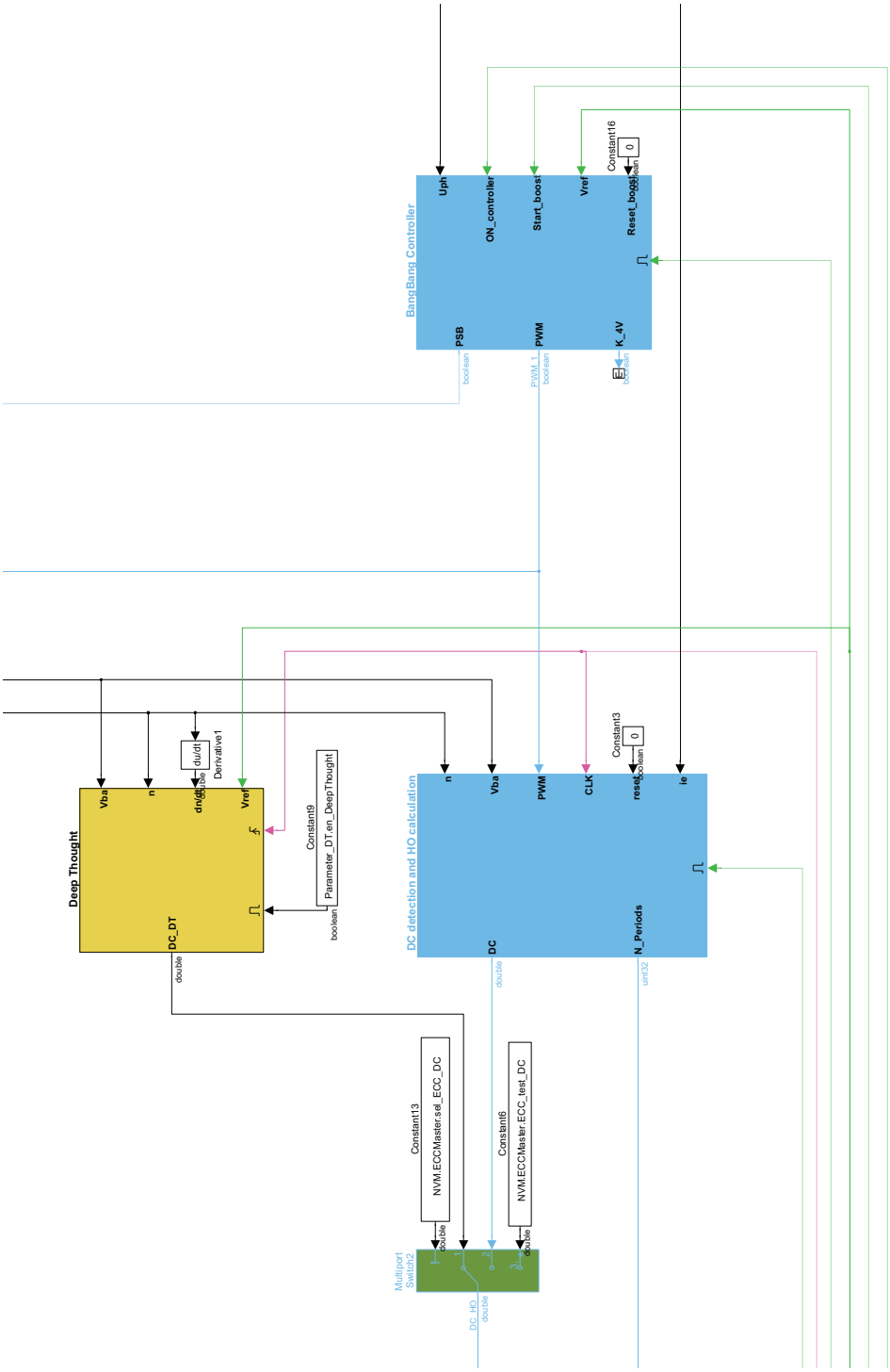
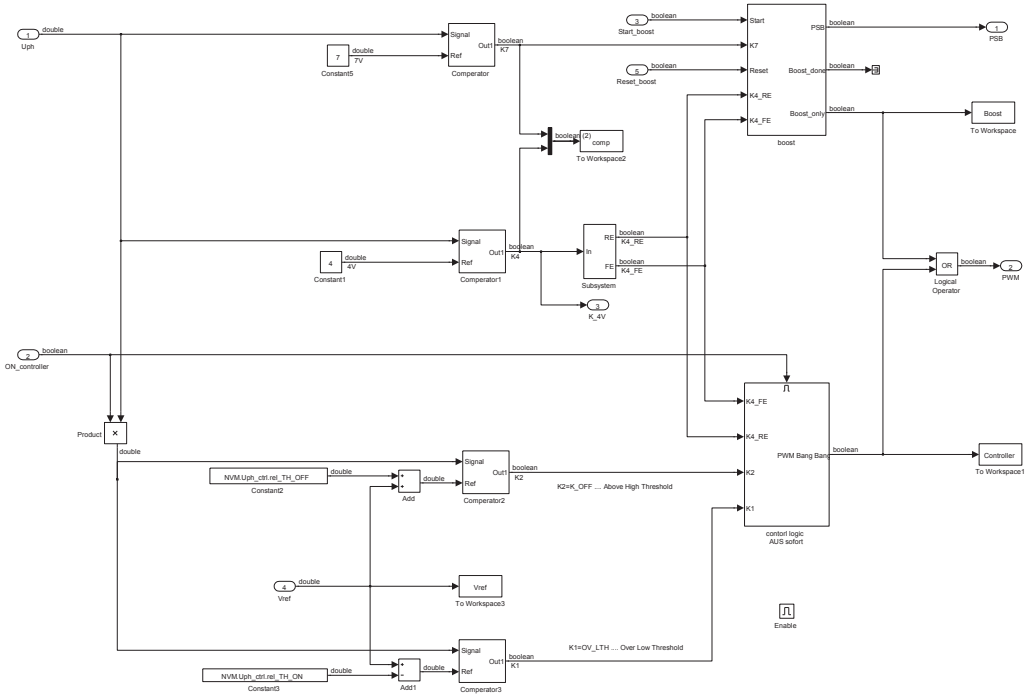
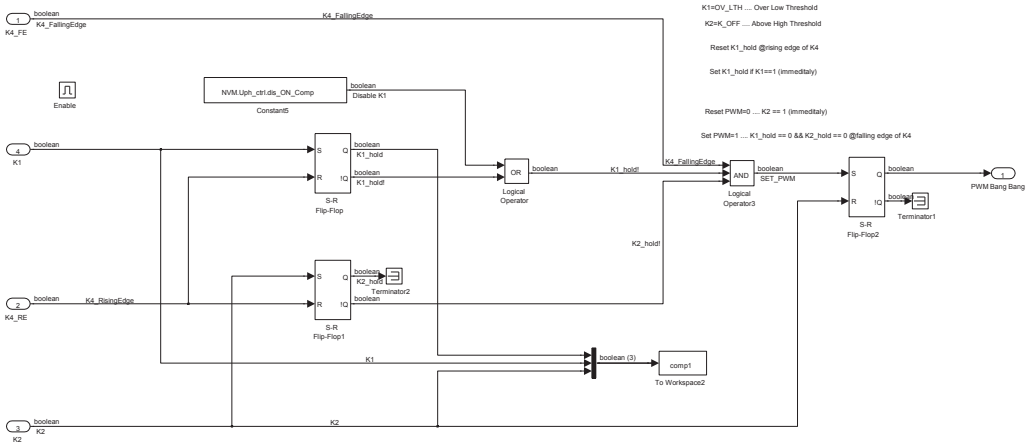


Figure .2: Bang-bang controller, DC Detection and Deep Thought

$\hat{U}_{ph}$ controller



(a)Comparator part



(b)Logic part

Figure .3: Bang-bang controller

DC determination and calculation of the handover value

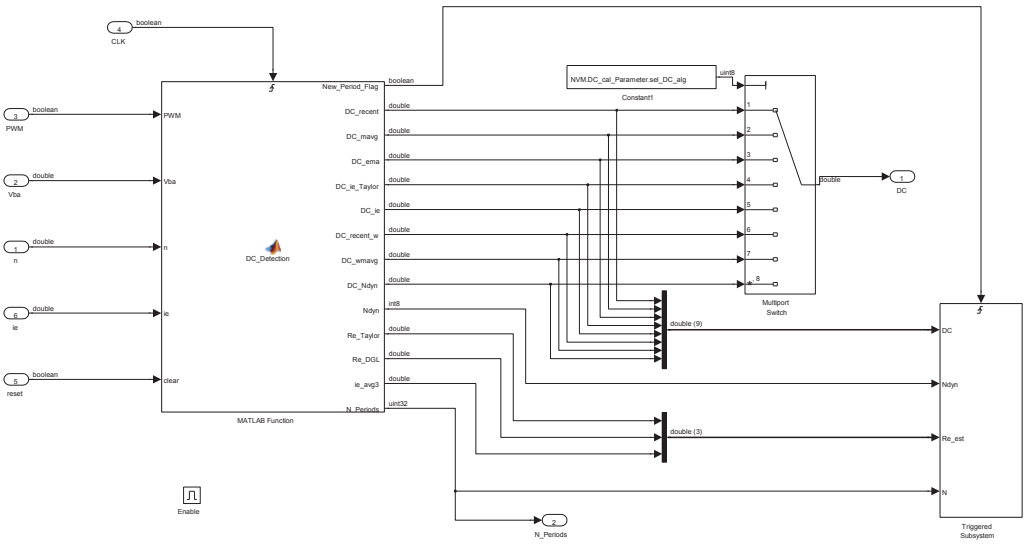


Figure .4: DC determination and DC_{HO} calculation

The code of DC determination and the DC filtering algorithms is given at the following pages:

```
function [New_Period_Flag, DC_recent, DC_mavg, DC_ema,DC_ie_Taylor,...
    DC_ie, DC_recent_w, DC_wmavg, DC_Ndyn,  Ndyn, Re_Taylor, Re_DGL, ...
    ie_avg3, N_Periods]= DC_Detection(NVM, PWM, Vba, n, ie ,clear)

    N_max= uint8(20); %N_max...Number of max samples which are are saved

    alpha=      NVM.DC_cal_Parameter.alpha;
    N_mavg=     NVM.DC_cal_Parameter.N_mavg;
    N_mavg_Re=  NVM.DC_cal_Parameter.N_mavg_Re;
    Tclk=       NVM.DC_cal_Parameter.CLK;
    k=          NVM.DC_cal_Parameter.k;         %weight constant

    persistent counter;
    persistent cnt_value_FE;
    persistent PWM_old;
    persistent cnt_PWM_RE;
    persistent Buffer_cnt_value_RE;
    persistent Buffer_cnt_value_FE;
    persistent Buffer_weights;
    persistent DC_recent_Upd;
    persistent DC_recent_w_Upd;
    persistent DC_EMA_Upd;
    persistent DC_mavg_Upd;
    persistent DC_wmavg_Upd;
    persistent DC_Ndyn_Upd;
    persistent n_old;
    persistent Ndyn_Upd;


    %ie measured:
    persistent ie_RE_old;
    persistent ie_FE;
    persistent Re_Taylor_Upd;
    persistent Re_DGL_Upd;
    persistent Buffer_Re_Taylor;
    persistent Buffer_Re_DGL;
    persistent ie_avg3_Upd;
    persistent DC_Re_Taylor_Upd;
    persistent DC_Re_mavg_Taylor_Upd;
    persistent DC_Re_DGL_mavg_Upd;

    if(isempty(counter) || clear == true)
        PWM_old=                false;
        cnt_PWM_RE=             uint32(0);
        counter=                uint16(0);
        cnt_value_FE=           uint16(0);

        Buffer_cnt_value_RE=    eps*ones(1,N_max); %avoid division by zero
        Buffer_cnt_value_FE=    zeros(1,N_max);
        Buffer_weights=         zeros(1,N_max);
        n_old=                  0;
        Ndyn_Upd=               int8(5);

        DC_recent_Upd=          NaN;
        DC_recent_w_Upd=        NaN;
```

```
    DC_EMA_Upd=             NaN;
    DC_mavg_Upd=            NaN;
    DC_wmavg_Upd=           NaN;
    DC_Ndyn_Upd=            NaN;

    ie_RE_old=              0;
    ie_FE=                  0;
    Buffer_Re_Taylor=       zeros(1,N_max);
    Buffer_Re_DGL=          zeros(1,N_max);
    ie_avg3_Upd=            NaN;
    Re_Taylor_Upd=          NaN;
    Re_DGL_Upd=             NaN;
    DC_Re_Taylor_Upd=       NaN;
    DC_Re_mavg_Taylor_Upd=  NaN;
    DC_Re_DGL_mavg_Upd=     NaN;

end

%1) counter increment 22kHz
% ------------------------------------------------------------
counter=    counter+1;

 %2)Sample and hold @PWM rising edge, DC calculations and counter reset
 % ------------------------------------------------------------------
 if(PWM_old == false && PWM == true)
    New_Period_Flag=    true;
    cnt_PWM_RE=    cnt_PWM_RE+1;                % count PWM RisingEdge
    ie_RE=  ie;

    if(cnt_PWM_RE > 1)
        DC_recent_Upd=      double(cnt_value_FE)/double(counter);
        % current/recent PWM-DC

        dn= (n-n_old)/(double(counter)*Tclk); %n/dt= (n(i)-n(i-1))/Tpwm
        w=  1+k*dn;                           % weight factor
        DC_recent_w_Upd=    DC_recent_Upd*w;

        [Re_Taylor_Upd, Re_DGL_Upd, ie_avg3_Upd]= ...
        Re_calculation(ie_FE,ie_RE,ie_RE_old,Tclk,...
        double(cnt_value_FE),double(counter),Vba);
        DC_Re_Taylor_Upd=   (Re_Taylor_Upd*ie_avg3_Upd)/Vba;


        if(cnt_PWM_RE >2)
            if(abs(dn)>2000)
                Ndyn_Upd=   int8(3);
            elseif(abs(dn)>100)
                Ndyn_Upd=   int8(5);
            else
                Ndyn_Upd=   int8(8);
            end

            Buffer_index= uint8(mod(cnt_PWM_RE-2,cast(N_max,'uint32')));
            % circular Index of N_max
            if(Buffer_index == 0)
```

```
            Buffer_index= N_max;
        end

        Buffer_cnt_value_RE(Buffer_index)=  counter;    %cnt Period
        Buffer_cnt_value_FE(Buffer_index)=  cnt_value_FE; %cnt High
        Buffer_Re_Taylor(Buffer_index)=     Re_Taylor_Upd;
        Buffer_Re_DGL(Buffer_index)=        Re_DGL_Upd;
        Buffer_weights(Buffer_index)=       w;

        DC_mavg_Upd= MAVG_calculation(int8(N_mavg),...
        (cnt_PWM_RE-2), double(Buffer_cnt_value_FE), ...
        double(Buffer_cnt_value_RE), int8(Buffer_index), int8(N_max));

        DC_wmavg_Upd= MAVG_calculation(int8(N_mavg), (cnt_PWM_RE-2), ...
        double(Buffer_cnt_value_FE).*Buffer_weights, ...
        double(Buffer_cnt_value_RE), int8(Buffer_index), ...
        int8(N_max));

        DC_Ndyn_Upd= MAVG_calculation(Ndyn_Upd, (cnt_PWM_RE-2), ...
         double(Buffer_cnt_value_FE), double(Buffer_cnt_value_RE),...
         int8(Buffer_index), int8(N_max));

        [Re_Taylor_mavg_Upd DC_Re_mavg_Taylor_Upd]=...
        Re_MAVG(int8(N_mavg_Re),(cnt_PWM_RE-2),...
         Buffer_Re_Taylor,int8(Buffer_index), int8(N_max),...
         ie_avg3_Upd,Vba);

        [Re_DGL_mavg_Upd    DC_Re_DGL_mavg_Upd]= ....
        Re_MAVG(int8(N_mavg_Re),(cnt_PWM_RE-2),Buffer_Re_DGL,...
        int8(Buffer_index), int8(N_max),ie_avg3_Upd,Vba);

    end
    n_old=              n;
 else
    DC_recent_Upd=      NaN;
 end

ie_RE_old= ie_RE;
counter=       uint16(0); % Reset counter @PWM RE -> New Periord

 % DC Exponetnial Moving Average
 % -------------------------------------------------------------
 if(cnt_PWM_RE > 4)
    DC_EMA_new= alpha*DC_recent_Upd + (1-alpha)*DC_EMA_Upd;
 elseif(cnt_PWM_RE > 3)
    DC_EMA_new= 1/2*(DC_recent_Upd + DC_EMA_Upd);
 elseif(cnt_PWM_RE > 2)
    DC_EMA_new= DC_recent_Upd;
 else
    DC_EMA_new= NaN;
 end

 DC_EMA_Upd= DC_EMA_new;
```

```
    else
        New_Period_Flag= false;
    end

    % 3)Sample and hold @PWM falling Edge
    % ----------------------------------------
    if(PWM_old == true && PWM == false)
        cnt_value_FE= counter;      % Remember cnt value @PWM FE -> High
        ie_FE= ie;
    end


    % 4) Outputs (22kHz)
    % ------------------------------------------------------------------
    DC_recent=              DC_recent_Upd;
    DC_recent_w=            DC_recent_w_Upd;
    N_Periods=              cnt_PWM_RE-2;
    DC_ema=                 DC_EMA_Upd;
    DC_mavg=                DC_mavg_Upd;
    DC_wmavg=               DC_wmavg_Upd;
    Re_Taylor=              Re_Taylor_Upd;
    Re_DGL=                 Re_DGL_Upd;
    DC_Re_Taylor=           DC_Re_Taylor_Upd;
    DC_ie_Taylor=     DC_Re_mavg_Taylor_Upd;
    DC_ie=         DC_Re_DGL_mavg_Upd;
    ie_avg3=                ie_avg3_Upd;
    Ndyn=                   Ndyn_Upd;
    DC_Ndyn=                DC_Ndyn_Upd;

    PWM_old=                PWM;

end


%-------------------------------------------------------------------------
function [DC_mavg]  = MAVG_calculation(N_Samples,cnt_value_Periods,...
    High_Array,Period_Array,Array_index,Array_length)
    %calculates moving Average with N_avg values or less

    if(cnt_value_Periods < N_Samples)
        N= double(cnt_value_Periods);  %Navg >  then counted DC Periods
        DC_mavg=  1/N * sum( High_Array(Array_index:-1:1)./...
                  Period_Array(Array_index:-1:1) );
    else
        N= double(N_Samples);     %more counted DC Periods than Navg

        if(Array_index >= N_Samples)
            index_N_avg = (Array_index - N_Samples)+1;
            DC_mavg= 1/N * sum( High_Array(Array_index:-1:index_N_avg)./...
                                Period_Array(Array_index:-1:index_N_avg) );

        else      %circular

            sum1= sum(  High_Array(Array_index:-1:1)./...
                        Period_Array(Array_index:-1:1) );
```

```
            index_sum2 = (Array_length - (N_Samples-Array_index))+1;
            sum2= sum(  High_Array(Array_length:-1:index_sum2)./...
                        Period_Array(Array_length:-1:index_sum2) );
            DC_mavg=     1/N * (sum1 + sum2);
        end

    end

end

function [Re_mavg DC_Re_mavg]  = Re_MAVG(N_Samples,cnt_value_Periods,...
    Re_Array,Array_index,Array_length,ie_period,Vbat)

    if(cnt_value_Periods < N_Samples)
        N= double(cnt_value_Periods);
        Re_mavg=     1/N * sum( Re_Array(Array_index:-1:1) );
    else
        N= double(N_Samples);
        if(Array_index >= N_Samples)

            index_N_avg = (Array_index - N_Samples)+1;
            Re_mavg=     1/N * sum( Re_Array(Array_index:-1:index_N_avg) );
        else
            sum1=       sum( Re_Array(Array_index:-1:1) );
            index_sum2 = (Array_length - (N_Samples-Array_index))+1;
            sum2=       sum( Re_Array(Array_length:-1:index_sum2) );
            Re_mavg=     1/N * (sum1 + sum2);
        end
    end
    ue=          Re_mavg*ie_period;
    DC_Re_mavg=  ue/Vbat;
end

function [Re_Taylor, Re_DGL, ie_avg3]= ...
    Re_calculation(ie_fall,ie_rise,ie_rise_old,Tclk,High,Period,Vbat)

    delta_ie_rise=  ie_fall - ie_rise_old;
    delta_t_rise=   High*Tclk;

    delta_ie_fall=  ie_rise - ie_fall;
    delta_t_fall=   (Period-High)*Tclk;

    %ie_avg2=        (ie_fall + ie_rise)/2;
    ie_avg3=        (ie_rise_old + ie_fall + ie_rise)/3; % Current Period

    %-------------------------------------------------------------
    %DGL Berechnung
    tau_DGL=    - delta_t_fall / log(ie_rise/ie_fall);
    Re_DGL=     Vbat * (1-exp(-delta_t_rise/tau_DGL)) / ...
        (ie_fall - ie_rise_old*exp(-delta_t_rise/tau_DGL));
    Le_DGL=     tau_DGL*Re_DGL;

    ue_stat_DGL=    Re_DGL*ie_avg3;
    DCstat_DGL=     ue_stat_DGL/Vbat;
```

```
    %-------------------------------------------------------
    %DGL durch Taylor

    Num=    (delta_ie_fall/delta_t_fall);
    Den=    (delta_ie_fall/delta_t_fall)*ie_rise_old - ...
        (delta_ie_rise/delta_t_rise)*ie_fall;

    Re_Taylor=  Vbat * Num/Den;
    Le_Taylor=  - ie_fall/(delta_ie_fall/delta_t_fall) * Re_Taylor;

    ue_stat_DGL=    Re_Taylor*ie_avg3;
    DCstat_Taylor=  ue_stat_DGL/Vbat;

end
```

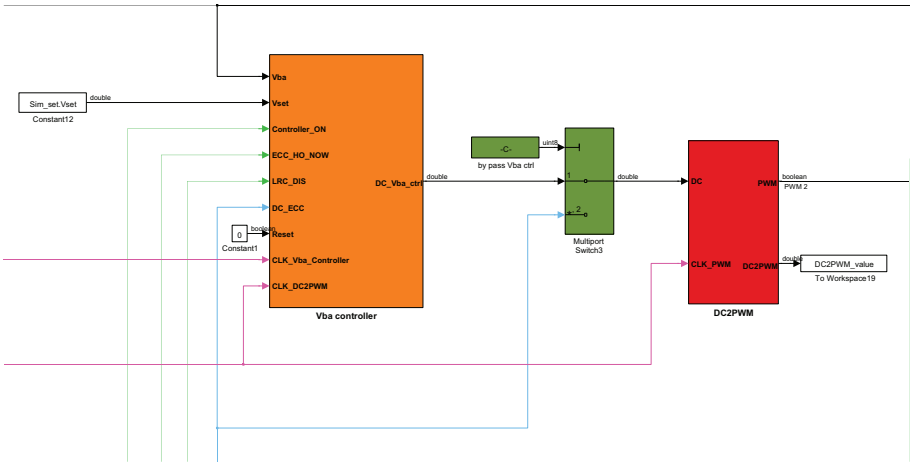
Deep Thought

The function code of *Deep Thought* is given here:

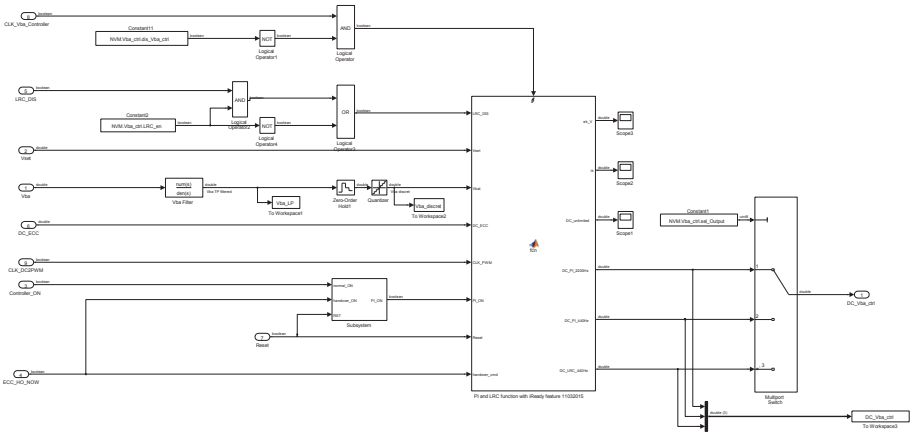
10.01.16 18:05 Block: Complete Simul.../Deep Thought Cal 1 of 1

```
function [  ie_theo_Vba,DCstat_theo_Vba,DCdyn_theo_Vba, ie_theo_Vref,...
            DCstat_theo_Vref,DCdyn_theo_Vref] = ...
            Algorithmus_DC_cal(Parameter_DT,Vba_Sim,n_Sim,dn_Sim,Vref)

    %referd to Vba
    [ie_theo_Vba,DCstat_theo_Vba,DCdyn_theo_Vba]= ...
    Cal_ie_DCstat_DCdyn(Parameter_DT,Vba_Sim,Vba_Sim,n_Sim,dn_Sim );

    %refered to Vref (from ECC Master)
    [ie_theo_Vref,DCstat_theo_Vref,DCdyn_theo_Vref]= ...
    Cal_ie_DCstat_DCdyn(Parameter_DT,Vref,Vba_Sim,n_Sim,dn_Sim );
end


function [ie_theo,DCstat_theo,DCdyn_theo] = ...
         Cal_ie_DCstat_DCdyn(P,Vsim_desired,Vba,n_Sim,dn_Sim )

    %DC stationary
    %-------------------------------------------------------------------------
        %Psi_s= Us/w:
        Psi_s= (Vsim_desired+P.Uf)/(2*pi*P.p*n_Sim/60);
        %Psi_s= Lg(ie)*ie ->ie:
        ie_theo= interp1(P.Lg_vec.*P.imuee_vec,P.imuee_vec, Psi_s);
        DCstat_theo= (ie_theo*P.Re)/Vba;

    %DC dynamic
    %-------------------------------------------------------------------------
      Psi_s_dot= -(dn_Sim/n_Sim) * Psi_s;
      Ldiff= interp1(P.imuee_vec,P.Ldiff_vec, ie_theo);
      Psi_e_dot= Psi_s_dot * (P.Lsig_e/Ldiff + 3/(2*P.ueV));
      DCdyn_theo= DCstat_theo + Psi_e_dot/Vba;
end
```

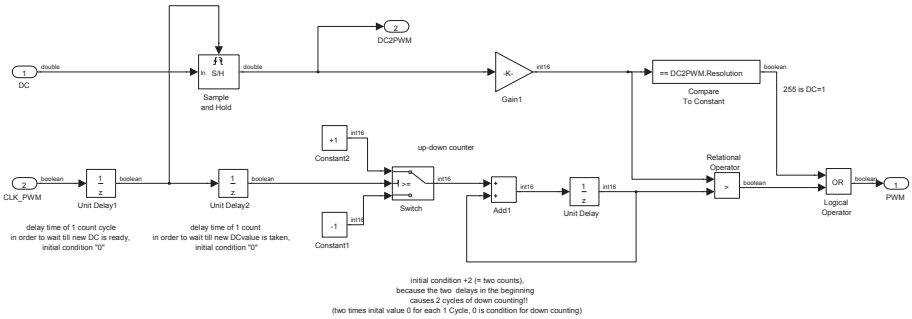
V_{BA} controller and $DC2PWM$



(a)Overview



(b)V_{BA} controller



(c)$DC2PWM$

Figure .5: V_{BA} controller and $DC2PWM$

The function code of V_{BA} controller is given at the following pages:

```
function [  ek_V, ik, DC_unlimited, DC_PI_2200Hz, DC_PI_440Hz, ...
            DC_LRC_440Hz] = fcn(LRC_DIS, NVM, Vset, Vbat, DC_ECC,...
                                CLK_PWM, PI_ON, Reset, handover_cmd)

  PI_Parameter=     NVM.Vba_ctrl.PI;
  LRC_Parameter=    NVM.Vba_ctrl.LRC;

  % peristent variables
  % ------------------------------------
    persistent CLK_PWM_old;             % edge detection PWM-Clock (220Hz)
    persistent run_PI;
    persistent handover_cmd_old;
    persistent handover_NOW;

  % persistent variables PI-controller
  % ------------------------------------
    persistent i_new;                       % -> i[k+1]
    persistent DC_PI_440Hz_old;

  % persistent variables LRC
  % ------------------------------------
    persistent DC_LRC_old;
    persistent LRC_state;
    persistent Rise_counter;
    persistent Fall_counter;
    persistent Fall_Value;
    persistent submitted_DC_value;          % DC handover value

    if (isempty(CLK_PWM_old) || Reset == true )
      CLK_PWM_old=                  false;
      handover_cmd_old=             false;
      handover_NOW=                 false;

      run_PI=                       false;
      i_new=                        0;
      DC_PI_440Hz_old=              0;

      DC_LRC_old=                   0;
      LRC_state=                    0;
      Rise_counter=                 0;
      Fall_counter=                 0;
      Fall_Value=                   0;
      submitted_DC_value=           0;
    end


    if(PI_ON == false)
        run_PI= false;
    elseif(CLK_PWM ~= CLK_PWM_old) %start synchron to the PWM Clock (RE or FE)
        run_PI= true;
    end

    if(handover_cmd_old == false && handover_cmd == true) % RE handover cmd
        handover_NOW=   true;
    end
```

```
% Call PI_Controller each trigger cycle (if PI_ON= 1),f_PI= 10*f_PWM
% ------------------------------------------------------------------
ek_V= Vset -Vbat; %do always

if(run_PI == true)

    if(handover_NOW== false)
      ik=               i_new;
      %i_new updated in the pervious iteration (first run: initial value),
      %new update after DC saturation and feedback choice
    else
       u_handover_V=    DC_ECC * Vbat;
       ik=              u_handover_V - PI_Parameter.Kp*ek_V;
    end

    uk_V=             PI_Parameter.Kp*ek_V + ik;
    DC_unlimited=     uk_V/Vbat;


    DC_PI_2200Hz=   max(0, min(1, DC_unlimited));   %saturation

    % 440Hz - Part
    %-----------------------------------------------------------------
    if(CLK_PWM ~= CLK_PWM_old)             % PWM-Clock edge (440Hz)!!!
        DC_PI_440Hz=       DC_PI_2200Hz;
        DC_PI_440Hz_old=   DC_PI_440Hz;

        if(LRC_DIS == false)
            %Call LRC function:
            %-------------------------------------------------
            [DC_LRC, LRC_state_next, Rise_counter_upd, ...
            Fall_counter_upd, Fall_Value_upd, DCECC_upd]= ...
            LRC(DC_PI_440Hz, DC_LRC_old,LRC_state, Rise_counter, ...
            Fall_counter, Fall_Value,submitted_DC_value ,...
            handover_NOW, LRC_Parameter);

            DC_LRC_440Hz=               max(0, min(1, DC_LRC));
            DC_LRC_old=                 DC_LRC_440Hz;
            LRC_state=                  LRC_state_next;
            Rise_counter=               Rise_counter_upd;
            Fall_counter=               Fall_counter_upd;
            Fall_Value=                 Fall_Value_upd;
            submitted_DC_value=         DCECC_upd;

            handover_NOW=               false;      % handover done
        else
            DC_LRC_440Hz=               DC_PI_440Hz;
            DC_LRC_old=                 DC_LRC_440Hz;
        end
    else
        DC_PI_440Hz=        DC_PI_440Hz_old;
        DC_LRC_440Hz=       DC_LRC_old;
    end
```

```
        %-------------------------------------------------------------

        %choose AntiWind Up Feedback (DC_FB)
        switch(PI_Parameter.AW_sel)
            case 1
                DC_FB=  DC_PI_2200Hz;
            case 2
                DC_FB=  DC_PI_440Hz;
            case 3
                DC_FB=  DC_LRC_440Hz;
            otherwise
                DC_FB=  DC_unlimited;   % no saturation (should be avoided)
        end

        uk_V_sat=   DC_FB*Vbat;
        i_new=      ik + PI_Parameter.Ts*...
                        (PI_Parameter.Kp/PI_Parameter.T_I)*...
                        (ek_V + (uk_V_sat- uk_V)/PI_Parameter.Kp);
        % i_new update: condition technique et Hanus (Anti-Windup)

    else

        DC_PI_2200Hz=       DC_PI_440Hz_old;
        DC_unlimited=       DC_PI_2200Hz;           % not important
        DC_PI_440Hz=        DC_PI_440Hz_old;
        DC_LRC_440Hz=       DC_LRC_old;

        ik=                 0;   % PI controller switched off
        i_new=              ik;

    end
    %-------------------------------------------------------------------

    CLK_PWM_old=        CLK_PWM;
    handover_cmd_old=   handover_cmd;
end


function [  DC_new, LRC_state_next, Rise_counter_upd, Fall_counter_upd,...
            Fall_Value_upd, DCECC_upd]  = LRC(DC_PI, DC_old, LRC_state,...
            Rise_counter, Fall_counter, Fall_Value,DCECC ,ECC_NOW, LRC)

    delta_T=            LRC.Ts_LRC;

    BLZ=                LRC.BLZ;  %in %
    LRC_RiseTime=       LRC.RT;   %in sec
    LRC_FallTime=       LRC.FT;

    en_BLZ_ECC=      LRC.en_BLZ_ECC;

    LRC_gradient=       (100/LRC_RiseTime);
    LRC_gradient_Fall=  (100/LRC_FallTime);
    LRC_gradient =      LRC_gradient/100;
    LRC_gradient_Fall=  LRC_gradient_Fall/100;
    BLZ=                BLZ/100;
```

```
N_Rise=             LRC_RiseTime/delta_T;
N_Fall=             LRC_FallTime/delta_T;


%-----------------------------------------------------------------------

%States
LRC_NOP=        0;
LRC_Rising=     1;
LRC_Falling=    2;
ECC_active=  3;
ECC_BLZ=     4;

if(ECC_NOW == true)
    LRC_state= ECC_active;
    PI_gradient=0;
else
    PI_gradient= (DC_PI-DC_old)/delta_T;
end


switch(LRC_state)

  %-------------------------------------------------------------------
  case LRC_NOP
      if PI_gradient <= LRC_gradient
          DC_new=           DC_PI;
          Rise_counter_upd= Rise_counter;    %no change of the Rise Timer

          if(PI_gradient > - LRC_gradient_Fall || DC_PI >= DC_old - BLZ)
              LRC_state_next=   LRC_NOP;
              Fall_counter_upd= Fall_counter; %no change of the Fall Timer
              Fall_Value_upd=   Fall_Value;
          else
              LRC_state_next=   LRC_Falling;
              Fall_counter_upd= 0;
              Fall_Value_upd=   DC_old;
          end
      else
          DC_new_max=           DC_old + LRC_gradient*delta_T + BLZ;
          Fall_counter_upd=     Fall_counter; %no change of the Fall Timer
          Fall_Value_upd=       Fall_Value;
          if(DC_PI < DC_new_max)
               if(PI_gradient > BLZ/delta_T)
                    LRC_state_next=     LRC_NOP;
                    Rise_counter_upd=   Rise_counter; %no change
               else
                    LRC_state_next=     LRC_Rising;
                    Rise_counter_upd=   0;       %reset Rise Timer
               end
          else
              DC_new=           DC_new_max;
              LRC_state_next=   LRC_Rising;
              Rise_counter_upd= 0;               %reset Rise Timer
```

```
        end
    end
DCECC_upd= DCECC;                   %no change of DC handover value!!

%-----------------------------------------------------------------
case LRC_Rising

  if(PI_gradient <= LRC_gradient)

      DC_new = DC_PI;
      Rise_counter_upd= Rise_counter;   %no change of the Rise Timer

      if(DC_PI < DC_old)
        LRC_state_next = LRC_Falling;
        Fall_counter_upd = 0;                  %Reset Fall Timer
        Fall_Value_upd = DC_old;
      else
        LRC_state_next = LRC_NOP;
        Fall_counter_upd = Fall_counter; %no change of the Fall Timer
        Fall_Value_upd= Fall_Value;
      end

  else
      %LRC-active
      DC_new = DC_old + LRC_gradient*delta_T;
      Rise_counter_upd= Rise_counter + 1;   %increment Rise Timer
      Fall_counter_upd = Fall_counter;  %no change of the Fall Timer
      Fall_Value_upd= Fall_Value;

      if(Rise_counter_upd < N_Rise)
        LRC_state_next=LRC_Rising;
      else
        LRC_state_next=LRC_NOP;
      end

  end
DCECC_upd= DCECC;          %no change of DC handover value!!

%-----------------------------------------------------------------
case LRC_Falling

  Fall_Value_upd=     (Fall_Value - LRC_gradient_Fall*delta_T);
  Fall_counter_upd=   Fall_counter+1;       %increment Fall Timer

  if(DC_PI <= DC_old || DC_PI < Fall_Value_upd)
      DC_new=             DC_PI;
      Rise_counter_upd=   Rise_counter;  %no change of the Rise Timer

      if(Fall_counter <= N_Fall)
        LRC_state_next= LRC_Falling;
      else
        LRC_state_next= LRC_NOP;
      end
  else
```

```
    if(Fall_Value_upd <= DC_old)
      DC_new_max=   DC_old + BLZ;
    else
       DC_new_max=  Fall_Value_upd + BLZ;
    end

    if(DC_PI <= DC_new_max)
          DC_new= DC_PI;
    else
          DC_new= DC_new_max;
    end
    LRC_state_next=     LRC_Rising;
    Rise_counter_upd=   0;           %Reset Rise Timer
end
DCECC_upd= DCECC;                   %no change of DC handover value!

%-----------------------------------------------------------------

DC_new=           DC_PI;
DCECC_upd=     DC_PI;               %Remember handover value

Fall_counter_upd= Fall_counter;
Fall_Value_upd=   Fall_Value;

if(en_BLZ_ECC == false)
  LRC_state_next=   LRC_Rising;     %if no BLZ is allowed
  Rise_counter_upd= 0;
else
    LRC_state_next=     ECC_BLZ;
    Rise_counter_upd=   Rise_counter;
end


%-----------------------------------------------------------------
case ECC_BLZ
      if(PI_gradient < - LRC_gradient_Fall)
          DC_new=           DC_PI;
          LRC_state_next=   LRC_Falling;
          Fall_counter_upd= 0;
          Fall_Value_upd =  DC_old;
          Rise_counter_upd= Rise_counter;
      else
          if(DC_PI < DCECC+BLZ)
              DC_new=           DC_PI;
              LRC_state_next=   ECC_BLZ;
              Rise_counter_upd= Rise_counter;
          else
              DC_new=           DCECC+BLZ;
              LRC_state_next=   LRC_Rising;
              Rise_counter_upd= 0;
          end

          Fall_counter_upd= Fall_counter;
          Fall_Value_upd=   Fall_Value;
      end
```

```
            DCECC_upd=      DCECC;      %no change of DC handover value!!

      otherwise
            LRC_state_next=   LRC_NOP;
            DC_new=           DC_old;
            Rise_counter_upd= Rise_counter;   %no change of the Rise Timer
            Fall_counter_upd= Fall_counter;   %no change of the Fall Timer
            Fall_Value_upd=   Fall_Value;
            DCECC_upd=      DCECC;
    end

end
```

ECC Master

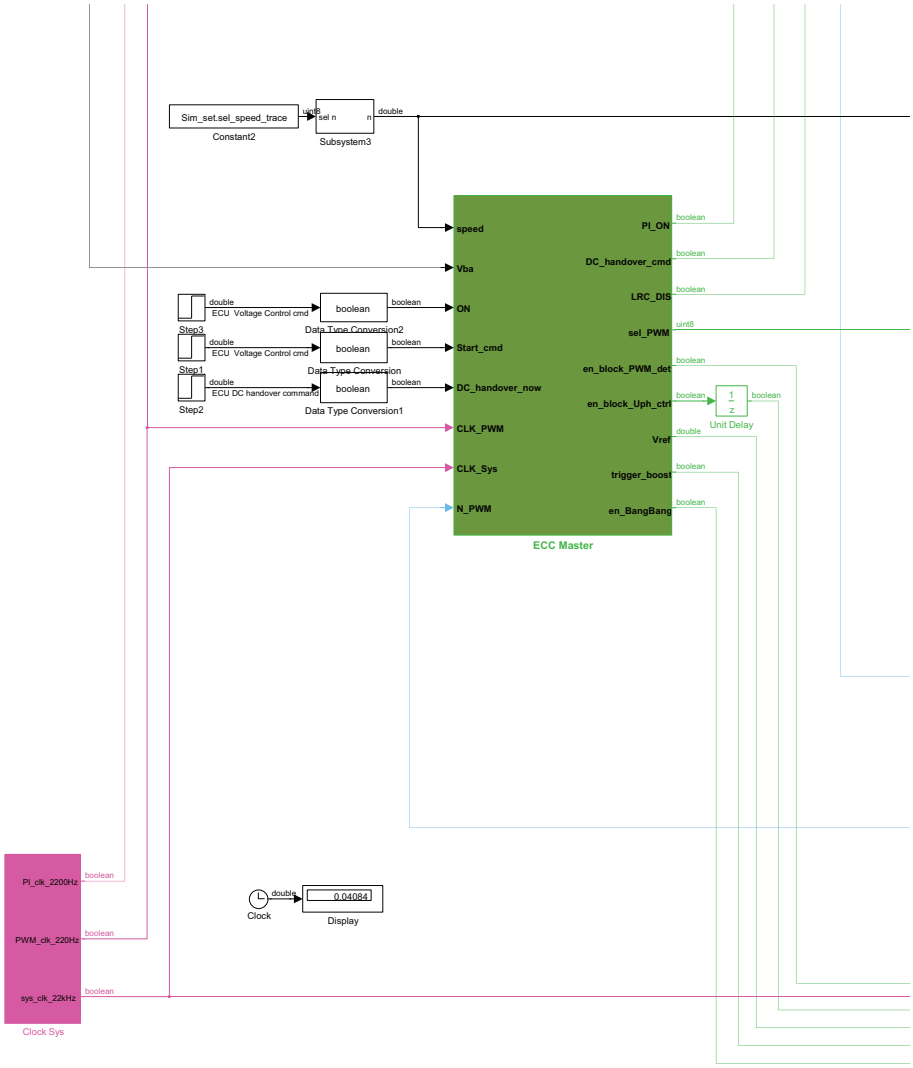


Figure .6: ECC Master and clock system

The function code of ECC Master is given at the following pages:

```
function [  PI_ON, DC_handover,LRC_DIS, selected_PWM, en_block_PWM_det, ...
            en_block_ctrl, Vref, trigger_boost,en_BangBang,state]  = ...
            fcn(NVM, speed, Vba, ON, Start, DC_transfer_now ,N_PWM, CLK_PWM)

persistent CLK_PWM_old;
persistent sel_PWM_old;
persistent handover_done;
persistent state_old;

if (isempty(sel_PWM_old))
  sel_PWM_old=      uint8(3);
  CLK_PWM_old=      false;
  handover_done=    false;
  state_old=        0;
end

%states
state_OFF=          0;
state_Pre=          1;
state_Vba_only=     2;
state_BangBang=     3;
state_byPass=       4;
state_handover=     5;


switch(state_old)
    case state_OFF
        if(speed > 0 && ON == true)
              state=    state_Pre;
        else
              state=    state_OFF;
        end

    case state_Pre
        if( speed >= NVM.ECCMaster.n_min ...
            && NVM.ECCMaster.en_ECC == true && Start == true)
            state=  state_BangBang;
        elseif( speed >= NVM.ECCMaster.n_min &&...
                NVM.ECCMaster.en_ECC == false && Start == true)
            state=  state_Vba_only;
        else
            state=  state_Pre;
        end

    case state_BangBang
        if( N_PWM >= NVM.ECCMaster.Nmin && DC_transfer_now == true ...
            && NVM.ECCMaster.by_pass_Vba_ctrl == 1)
            state=  state_handover;
        elseif( N_PWM >= NVM.ECCMaster.Nmin && DC_transfer_now == true ...
                && NVM.ECCMaster.by_pass_Vba_ctrl == 2)
            state=  state_byPass;
        else
            state=  state_BangBang;
        end
```

```
    case state_handover
        state=  state_handover;

    case state_byPass
        state=  state_byPass;

    case state_Vba_only
        state=  state_Vba_only;

    otherwise
        state= state_old;
end


state_old= state;

switch(state)
    case state_OFF
      en_block_ctrl=    false;
      en_block_PWM_det= false;

      trigger_boost=    false;
      en_BangBang=      false;

      sel_PWM=          uint8(3);       % 3... DC= 0;
      PI_ON=            false;
      DC_handover=      false;

    case state_Pre
      en_block_PWM_det= false;

      trigger_boost=    true;
      en_BangBang=      false;

      sel_PWM=          uint8(3);       % 3... DC= 0;
      PI_ON=            false;
      DC_handover=      false;


      if(NVM.ECCMaster.en_PSB == true)
        en_block_ctrl=  true;
      else
        en_block_ctrl=  false;
      end

    case state_BangBang
        %Boost and Bang Bang Control (ECC phase)

        en_block_ctrl=      true;
        % this signal is DELAYED OUTSIDE (one Sys Period)!,
        %enables block phase controller,
        en_block_PWM_det=   true;
        %enables block DC Detection, should be enabled BEFORE block
        %phase controller is enabled
```

```
    trigger_boost=      true;           % starts Boost
    en_BangBang=        true;           % enables Bang Bang controller

    sel_PWM=            uint8(2);       % selects Bang Bang PWM

    PI_ON=              false;
    DC_handover=        false;

case state_handover
    trigger_boost=      false;          % disables Boost
    en_block_PWM_det=   false;


    if(handover_done == false)

        if(CLK_PWM ~= CLK_PWM_old)
           PI_ON=               true;
           DC_handover=         true;
           sel_PWM=             uint8(1); % PWM= DC2PWM
           en_block_ctrl=       true;
           en_BangBang=         true; % Bang Bang controller still on

           handover_done=       true;
        else
           en_block_ctrl=       true;
           en_BangBang=         true;
           sel_PWM=             uint8(2);        % PWM= Bang Bang PWM
           PI_ON=               false;           % delay handover
           DC_handover=         false;           % delay handover
        end
    else
           PI_ON=               true;
           DC_handover=         true;
           sel_PWM=             uint8(1);        % PWM= DC2PWM
           en_BangBang=         false;
           en_block_ctrl=       false;
    end

case state_byPass
    trigger_boost=      false;          % disables Boost
    en_block_PWM_det=   false;
    PI_ON=              false;
    DC_handover=        false;

    if(handover_done == false)
        if(CLK_PWM ~= CLK_PWM_old)
           sel_PWM=             uint8(1);  % PWM= DC2PWM
           en_BangBang=         true;   % Bang Bang controller still on
           en_block_ctrl=       true;

           handover_done=       true;
        else
           en_block_ctrl=       true;
           en_BangBang=         true;
           sel_PWM=             uint8(2);        % PWM= Bang Bang PWM
```

```
            end
        else
                sel_PWM=            uint8(1);        % PWM= DC2PWM
                en_block_ctrl=      false;
                en_BangBang=        false;
        end

    case state_Vba_only
        en_block_PWM_det=   false;
        en_BangBang=        false;

        PI_ON=              true;
        DC_handover=        false;
        sel_PWM=            uint8(1);

        trigger_boost=      true;

      if(NVM.ECCMaster.en_PSB == true)
        en_block_ctrl=  true;
      else
        en_block_ctrl=  false;
      end

    otherwise
        sel_PWM=            uint8(3);
        PI_ON=              false;
        DC_handover=        false;
        en_block_PWM_det=   false;
        en_block_ctrl=      false;
        trigger_boost=      false;
        en_BangBang=        false;
end


if(CLK_PWM ~= CLK_PWM_old && sel_PWM ~= sel_PWM_old)
    selected_PWM=   sel_PWM;
    sel_PWM_old=    selected_PWM;
else
    selected_PWM=    sel_PWM_old;
end

CLK_PWM_old= CLK_PWM;

if(NVM.ECCMaster.en_dynVref == true)
  if(speed > 2000)
    deltaU= 0;
    %Vref= Vba + deltaU;
  elseif(speed > 1500)
    deltaU= -0.5;
    %Vref= Vba + deltaU;
  elseif(speed > 1300)
    deltaU= -1;
    %Vref= Vba + deltaU;
  else
    deltaU=-2;
```

```
    %Vref= Vba + deltaU;
  end
  Vref=     Vba + deltaU;
else
  Vref=     Vba+NVM.ECCMaster.deltaU;
end

LRC_DIS= false;
```

Alternator Model, Rectifier and load

For completeness also an overview of the alternator model, the rectifier, and load is given in figure .7.

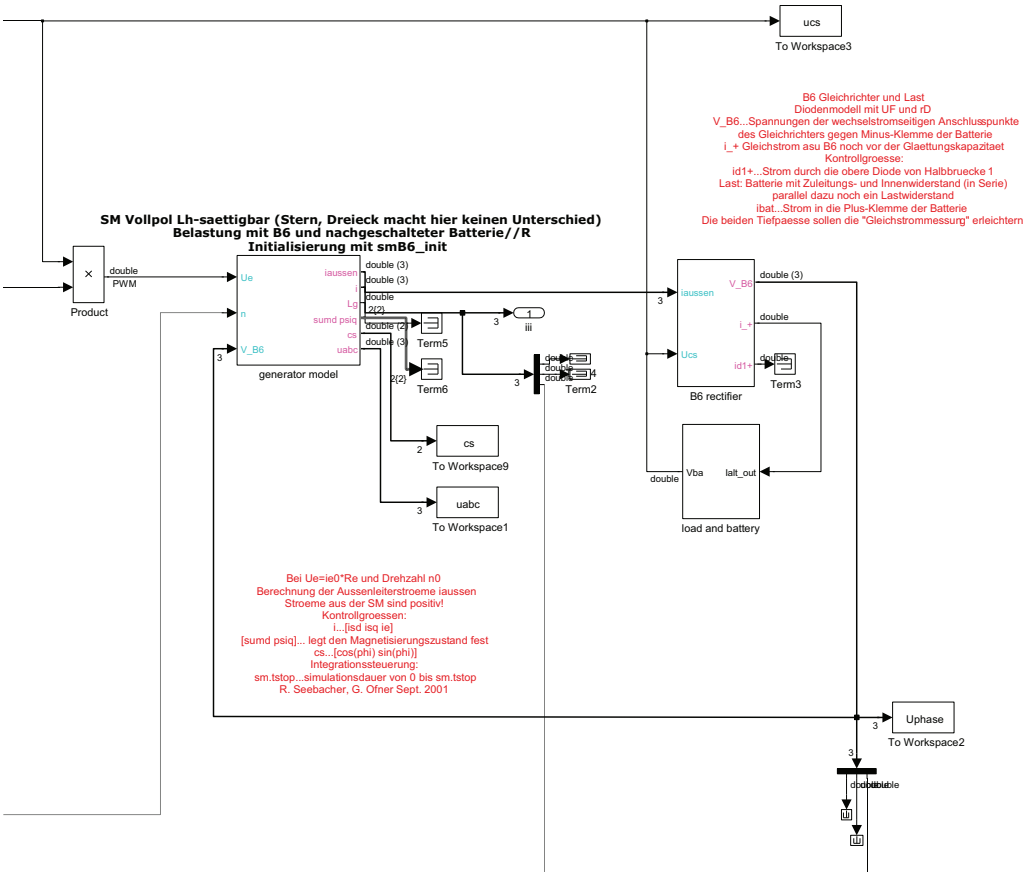


Figure .7: Overview alternator model, rectifier and load

Bibliography

[1] Frank Allgöwer. *Vorlesung Regelungstechnik 1*. University Lecture Notes. 2009. URL: http://www.ist.uni-stuttgart.de/education/courses/RTI/download/Skript/script.pdf (cit. on pp. 90, 93).

[2] Hans-Jochen Bartsch and Michael Sachs. *Taschenbuch mathematischer Formeln für Ingenieure und Naturwissenschaftler: für Studium und Beruf*. German. 23., überarb. Aufl. München: Fachbuchverl. Leipzig im Carl-Hanser-Verl, 2014. ISBN: 9783446437357;3446438009;3446437355;9783446438002; URL: http://castor.tugraz.at/F?func=direct&doc_number=000469154 (cit. on p. 69).

[3] Stephen J. Chapman. *Electric machinery fundamentals*. English. 3.;3; Boston [u.a.]: WCB McGraw-Hill, 1999. ISBN: 0070119503;9780070119505; URL: http://castor.tugraz.at/F/5HEQM88G9MC6HCDF1CSIMYYNXCHHFTN3PTSPRVJQ8BAQNPBTL7-75003?func=find-b&find_code=WRD&adjacent=N&request=Electric+machinery+fundamentals&x=38&y=16 (cit. on pp. 18, 102).

[4] *Datasheet TLE8880*. Infineon. URL: http://www.infineon.com/cms/de/product/automotive-system-ic/alternator-regulator/TLE8880TN/productType.html?productType=5546d462455feb53014597ed096139dc (cit. on pp. 6, 70, 89).

[5] Tom Denton. *Automobile mechanical and electronic systems: automotive technology - vehicle maintenance and repair*. English. 1. publ. Amsterdam [u.a.]: Elsevier [u.a.], 2011. ISBN: 0080969453;9780080969459; URL: http://castor.tugraz.at/F?func=direct&doc_number=000430710 (cit. on p. 5).

[6] W.C. Duesterhoeft, Max W. Schulz, and Edith Clarke. "Determination of Instantaneous Currents and Voltages by Means of Alpha, Beta, and Zero Components." In: *American Institute of Electrical Engineers, Transactions of the* 70.2 (July 1951), pp. 1248–1255. ISSN: 0096-3860. DOI: 10.1109/T-AIEE.1951.5060554. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5060554 (cit. on p. 39).

[7] Felix Gausch, Anton Hofer, and Kurt Schlacher. *Digitale Regelkreise: ein einfacher Einstieg mit dem Programm µLINSY*. German. 2., durchges. u. korr. Aufl. München ; Wien: Oldenbourg, 1993. ISBN: 3486227343;9783486227345; URL: http://castor.tugraz.at/F?func=direct&doc_number=000209530 (cit. on pp. 90–92).

[8] Robert Bosch GmbH. *Bosch automotive electrics and automotive electronics: systems and components, networking and hybrid drive*. English. 5. Wiesbaden: Springer Vieweg, 2014. ISBN: 9783658017835;365801783X; DOI: 10.1007/978-3-658-01784-2. URL: http://castor.tugraz.at/F?func=direct&doc_number=000463394 (cit. on pp. 3–6).

[9] R. Hanus, M. Kinnaert, and J. L. Henrotte. "Conditioning Technique, a General Anti-windup and Bumpless Transfer Method." In: *Automatica* 23.6 (Nov. 1987), pp. 729–739. ISSN: 0005-1098. DOI: 10.1016/0005-1098(87)90029-X. URL: http://dx.doi.org/10.1016/0005-1098(87)90029-X (cit. on p. 94).

[10] Martin Horn and Nicolaos Dourdoumas. *Regelungstechnik: rechnerunterstützter Entwurf zeitkontinuierlicher und zeitdiskreter Regelkreise*. German. München [u.a.]: Pearson Studium, 2004. ISBN: 3827372607;9783827372604;9783827370594;3827370590; URL: http://castor.tugraz.at/F?func=direct&doc_number=000282737 (cit. on p. 90).

[11] R. Ivankovic, J. Cros, and P. Viarouge. "Experimental comparison of rectifiers for Lundell automotive alternators." In: *Power Electronics and Applications, 2009. EPE '09. 13th European Conference on*. Sept. 2009, pp. 1–10. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5279029&newsearch=true&queryText=Experimental%20Comparison%20of%20Rectifiers%20for%20Lundell%20Automotive%20Alternators (cit. on p. 4).

[12] Witting Volkmar (AT) Kadric Almas (DE) Sudan Christoph (DE). "Hochfahren des Erregerstroms eines an ein Bordnetz eines Kraftfahrzeugs angeschlossenen Mehrphasenwechselstrom Generators beim Aktivieren des Generators." German. DE 10 2013 207 135 A1. Oct. 2014. URL: https://register.dpma.de/DPMAregister/pat/PatSchrifteneinsicht?docId=DE102013207135A1&page=1&dpi=150&lang=de (cit. on pp. 1, 9).

[13] P. Krause, O. Wasynczuk, and S. Sudhoff. "Basic Principles for Electric Machine Analysis." In: *Analysis of Electric Machinery and Drive Systems*. Wiley-IEEE Press, 2002, pp. 1–65. ISBN: 9780470544167. DOI: 10.1109/9780470544167.ch1. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5265720 (cit. on p. 104).

[14] Sheng Qiao et al. "Application of Engine Intelligent Start-Stop System in Technology of Vehicle Fuel Saving." In: *Measuring Technology and Mechatronics Automation (ICMTMA), 2014 Sixth International Conference on*. Jan. 2014, pp. 128–131. DOI: 10.1109/ICMTMA.2014.35. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6802651&queryText=start-stop&newsearch=true (cit. on p. 1).

[15] Dimitrios Sarafianos et al. "Characterisation and modelling of automotive Lundell alternators." In: *Power Electronics and Drive Systems (PEDS), 2015 IEEE 11th International Conference on*. June 2015, pp. 928–933. DOI: 10.1109/PEDS.2015.7203470. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7203470&queryText=Characterisation%20and%20modelling%20of%20automotive%20Lundell%20alternators&newsearch=true (cit. on p. 4).

[16] Dierk Schröder. *Elektrische Antriebe - Grundlagen: Mit durchgerechneten Übungs- und Prüfungsaufgaben*. German. 2013. ISBN: 3642304710;9783642304712; DOI: 10.1007/978-3-642-30471-2. URL: http://castor.tugraz.at/F?func=direct&doc_number=000453785 (cit. on p. 43).

[17] R. Seebacher and G. Dannerer. "Measurement Based Parameter Estimation for Claw Pole Generators." In: *Proceedings of the 15th International Conference on Electrical Machines ,ICEM 2002*. Aug. 2002. URL: https://online.tugraz.at/tug_online/fdb_bearb_2.editVeroeff?cfanr=F12109&pPersonNr=0&corg=2318&pVeroeffNr=F91049&vEditable=N (cit. on p. 39).

[18] R. Seebacher and G. Ofner. "Modelling of a Claw Pole Generators." In: *Proceedings of the 15th International Conference on Electrical Machines ,ICEM 2002*. Aug. 2002. URL: https://online.tugraz.at/tug_online/fdb_bearb_2.editVeroeff?cfanr=F12110&pPersonNr=0&corg=2318&pVeroeffNr=F91052&vEditable=N (cit. on pp. 39, 40).

[19] Daniel Watzenig. "Reglerentwurf für einen Klauenpolsynchrongenerator." German. MA thesis. 2002. URL: http://castor.tugraz.at/F?func=direct&doc_number=000257337 (cit. on pp. 42, 89).

[20] J. G. Ziegler and N. B. Nichols. "Optimum Settings for Automatic Controllers." In: *Trans. ASME* 64 (1942), pp. 759–768. URL: http://chem.engr.utc.edu/Student-files/x2008-Fa/435-Blue/1942-paper.pdf (cit. on pp. 90, 93).