FACULTY OF ENGINEERING AND ARCHITECTURE
ELECTRICAL AND ELECTRONICS ENGINEERING

HIGH POWER EFFICIENCY DESIGN APPROACH OF AN LLC RESONANT CONVERTER FOR UPS BATTERY CHARGER APPLICATION AND BATTERY CHARGE - DISCHARGE REGRESSION MODEL

GRADUATION PROJECT

in partial fulfillment of the requirements for the degree of
BACHELOR OF SCIENCE

TURHAN CAN KARGIN
150403005

SUPERVISOR: ASSOC. PROF. SAVAŞ ŞAHİN

June 2020

HIGH POWER EFFICIENCY DESIGN APPROACH OF AN LLC RESONANT CONVERTER FOR UPS BATTERY CHARGER APPLICATION AND BATTERY CHARGE - DISCHARGE REGRESSION MODEL

A GRADUATION PROJECT

by

TURHAN CAN KARGIN

submitted to the ELECTRICAL AND ELECTRONICS ENGINEERING of

İZMİR KÂTİP ÇELEBİ UNIVERSITY

Approved by:

PROF. DR. ADNAN KAYA

(Signature)

Member Name (Signature)	Member Name (Signature)
Member Name (Signature)	Member Name (Signature)

JUNE 2020

ABSTRACT

In this thesis, an optimal design procedure of inductor-inductor-capacitor (LLC) resonant DC-DC converter is developed for uninterruptible power supply (UPS) battery charge applications based on high power efficiency. The LLC resonant converters have many advantages such as high-power efficiency and less switching losses when compared with other converters features. It is also capable of operating in narrow switching frequency where zero voltage switching can be provided. The DC-DC converter with 400V input and 48V/3.1A output has been selected as an experimental setup. In order to reach optimal design of LLC resonant converter and required output values, switching frequency might be determined as above of resonance frequency, based on theoretical calculations and Power Electronics Simulation package program. The obtained maximum power efficiency with the proposed method was measured as 95.22%. Besides, charge-discharge models of the battery were obtained from the battery data obtained via deriving regression models with machine learning algorithms where battery electrical energy consumptions, battery status, and temperature data can be analyzed. R^2 score and root mean square error tests are performed for ten different regression models. Random forest regression is determined as the best model among regression models for the obtained data set.

Keywords: Energy; UPS battery charge; LLC resonant converter; regression models

ÖZET

Bu tezde, yüksek güç verimliliğine dayanan kesintisiz güç kaynağı (KGK) akü şarj uygulamaları için indüktör-indüktör-kapasitör (LLC) rezonans DA-DA dönüştürücünün optimal tasarım prosedürü geliştirilmiştir. LLC rezonans dönüştürücülerinin yüksek güç verimliliği ve diğer dönüştürücülerin özellikleriyle karşılaştırıldığında daha az anahtarlama kaybı gibi birçok avantajı vardır. Ayrıca sıfır voltaj anahtarlamanın sağlanabildiği dar anahtarlama frekansında da çalışabilir. 400V giriş ve 48V / 3.1A çıkışlı DA-DA dönüştürücü, deneysel kurulum olarak seçilmiştir. LLC rezonans dönüştürücüsünün optimum tasarımına ve gerekli çıkış değerlerine ulaşmak için, anahtarlama frekansı teorik hesaplamalara ve Güç elektroniği benzetim çalışmaları paket programına dayanarak rezonans frekansının yukarısında olduğu belirlenmiştir. Önerilen yöntemle elde edilen maksimum güç verimliliği 95.22% olarak ölçülmüştür. Ayrıca, akü şarj-deşarj modelleri, akü elektrik enerjisi tüketiminin, akü durumunun ve sıcaklık verilerinin analiz edilebildiği makine öğrenme algoritmaları ile regresyon modelleri türetilerek elde edilen akü verilerinden elde edilmiştir. On farklı regresyon modeli için R^2 ve hata kareler ortalamasının karesi testi yapılmıştır. Random Forest regresyonu, elde edilen veri seti için regresyon modelleri arasında en iyi model olarak belirlenmiştir.

Anahtar Kelimeler: Enerji; KGK akü şarjı; LLC rezonans dönüştürücü; regresyon modelleri

ACKNOWLEDGMENTS

This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under 2209B-Bachelor Final Thesis Focused on Industry Program with Project number 1139B411900874.

This work is presented in 2nd International Conference of Applied Sciences, Engineering and Mathematics (ICASEM 2020).

I would like to express my special thanks to my advisor Assoc. Prof. Savaş ŞAHİN for his patience, motivation and continuous support. My sincere thanks also goes to Mr. Fırat DEVECI for his guidance and helping me with every step of the project.

Finally, I would like to thank my family and friends for their endless support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

LLC Inductor-Inductor-Capacitor

SRC Series Resonant Converter

DC Direct Current

AC Alternating Current

PFC Power Factor Correction

UPS Uninterruptible Power Supply

IC Integrated Circuit

ZVS Zero Voltage Switching

PCB Printed Circuit Board

SMPS Switch Mode Power Supply

CV Constant Voltage

CC Constant Current

EMI Electromagnetic Interference

LGBM Light Gradient Boosting Machine

XGB eXtreme Gradient Boosting

MLP Multilayer Perceptron

ANN Artificial Neural Network

RMSE Root Mean Square Error

1. INTRODUCTION

The growing demand for higher power density and low profile in power converter designs has forced designers to increase switching frequencies. Operation at higher frequencies considerably reduces the size of passive components such as transformers and filters. However, switching losses have been an obstacle to high frequency operation. In order to reduce switching losses, allowing high frequency operation, resonant switching techniques have been developed. These techniques process power in a sinusoidal manner and the switching devices are softly commutated. Therefore, the switching losses and noise can be dramatically reduced. Conventional resonant converters use an inductor in series with a capacitor as a resonant network. Two basic configurations are possible for the load connection: series connection and parallel connections [1]. See Figure 1.1.

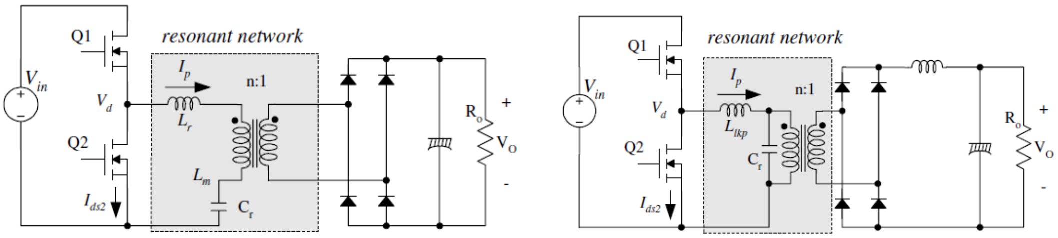


Figure 1-1 Half-bridge series and parallel resonant converters [1]

These two different resonant converters have some limitations. For SRC, the rectifier-load network is placed in series with the L-C resonant network [2-4]. From this configuration, the resonant network and the load act as a voltage divider. By changing the frequency of driving voltage V_d, the impedance of the resonant network changes. The input voltage will be split between this impedance and the reflected load. Since it is a voltage divider, the DC gain of an SRC is always lower than 1. At light load condition, the impedance of the load will be very large compared to the impedance of the resonant network; all the input voltage will be imposed on the load. This makes it difficult to regulate the output at light load. Theoretically, frequency should be infinite to regulate the output at no load. For parallel resonant converter, the rectifier-load network is placed in parallel with the resonant capacitor as depicted [5-7]. Since the load is connected in parallel with the resonant network, there inevitably exists large amount of circulating current. This makes it difficult to apply parallel resonant topologies in high power applications. Therefore, in order to solve the limitations of the conventional resonant converters, the LLC resonant converter has been proposed. The LLC-type resonant converter has many advantages over conventional resonant converters. First, it can regulate the output over wide line and load variations with a relatively small variation of switching frequency. Second, it can achieve zero voltage switching (ZVS) over the entire operating range. Finally, all essential parasitic elements, including junction capacitances of all semiconductor devices and the leakage inductance and magnetizing inductance of the transformer, are utilized to achieve ZVS. See Figure 1.2.

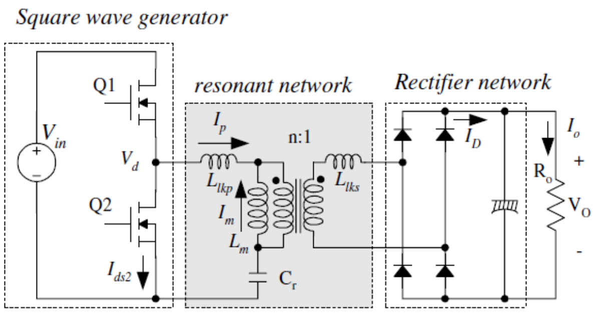


Figure 1-2 A schematic of half-bridge LLC resonant converter [1]

Recently, the LLC resonant converter has drawn a lot of attention due to its advantages over the conventional series resonant converter and parallel resonant converter: narrow frequency variation over wide load and input variation and Zero Voltage Switching (ZVS) of the switches for entire load range. The LLC-type resonant converter has many advantages over conventional resonant converters. First, it can regulate the output over wide line and load variations with a relatively small variation of switching frequency. Second, it can achieve zero voltage switching (ZVS) over the entire operating range. Finally, all essential parasitic elements, including junction capacitances of all semiconductor devices and the leakage inductance and magnetizing inductance of the transformer, are utilized to achieve ZVS. The battery used in Uninterruptible Power Supply should be in such a way that it should satisfy the features such as smooth and quick charging, high power density, high efficiency. Furthermore, battery technology is improving and as this transition occurs, the charging of these batteries becomes very complicated due to the high voltages and currents involved in the system and the sophisticated charging algorithms [8]. The most commonly used battery charging architecture is shown in figure 3. It consists of mainly two stages, namely power factor correction PFC stage and DC-DC converter stage. The power factor correction stage is a continuous conduction mode of boost topology [9]. In this thesis, the main focus is the DC-

DC converter which plays an important role in battery charger by regulating the output current and voltage. See Figure 1.3.

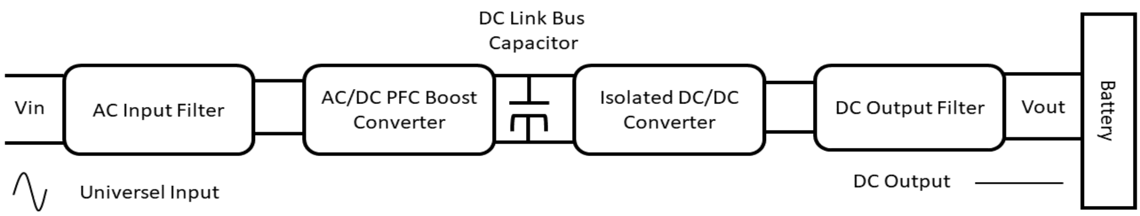


Figure 1-3 Block diagram of a universal battery charger

In this thesis, an optimal design procedure of inductor-inductor-capacitor (LLC) resonant converter for UPS battery charger applications based on high efficiency and Battery Charge-Discharge regression model is proposed. In the design procedure, 4x12V UPS battery is used. Thus, LLC resonant converter should be regulated the output voltage in a wide voltage range with different load conditions according to typical charging profile of battery. For the design procedure, basic operation characteristics of LLC resonant converter is defined, and operation regions are discussed in terms of high efficiency. The operation regions of LLC resonant converter are discussed to regulate wide output voltage range. Therefore, the purpose of the thesis is designing and producing an LLC resonant converter that will have 48V/3.1A output values to charge 4 x 12V / 30Ah batteries. The circuit is simulated using PSIM software. PCB design was performed by Eagle Autodesk. In addition, it is presented as secondary output to find the charge-discharge models under varying conditions by deriving the regression models with machine learning algorithms where the battery electricity energy consumption, battery status and temperature data can be analyzed.

2. MATERIALS

2.1. PSIM Simulation Software

PSIM [10] is an Electronic circuit simulation software package, designed specifically for use in power electronics and motor drive simulations but can be used to simulate any electronic circuit. Developed by Powersim, PSIM uses nodal analysis and the trapezoidal rule integration as the basis of its simulation algorithm. PSIM provides a schematic capture interface and a waveform viewer Simview. PSIM has several modules that extend its functionality into specific areas of circuit simulation and design including control theory, electric motors, photovoltaics and wind turbines. PSIM is used by industry for research and product development and it is used by educational institutions for research and teaching. There are modules that enable motor drive simulation, digital control, and the calculation of thermal losses due to switching and conduction. There is a renewable energy module which allows for the simulation of photovoltaics (including temperature effects), batteries, supercapacitor, and wind turbines. See Figure 2.1.

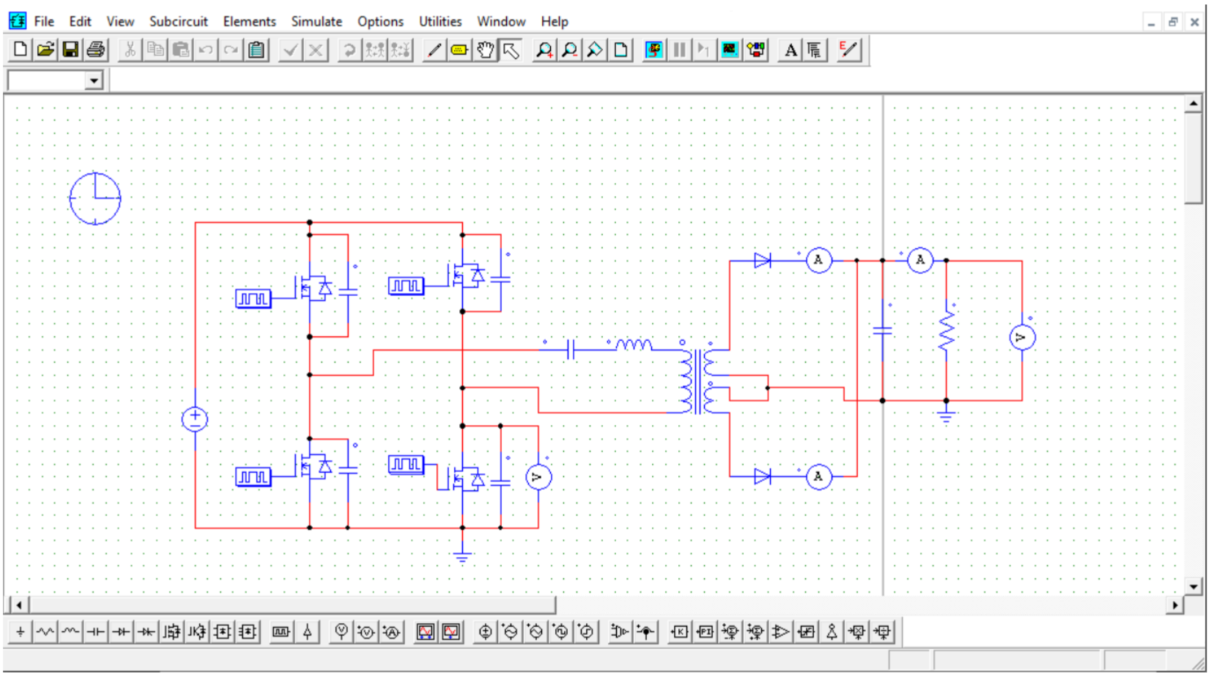


Figure 2-1 PSIM Software interface [10]

2.2. MATLAB Simulink

Simulink [11] is a MATLAB-based graphical programming environment for modeling, simulating, and analyzing multidomain dynamical systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in automatic control and digital signal processing for multidomain simulation and model-based design. See Figure 2.2.

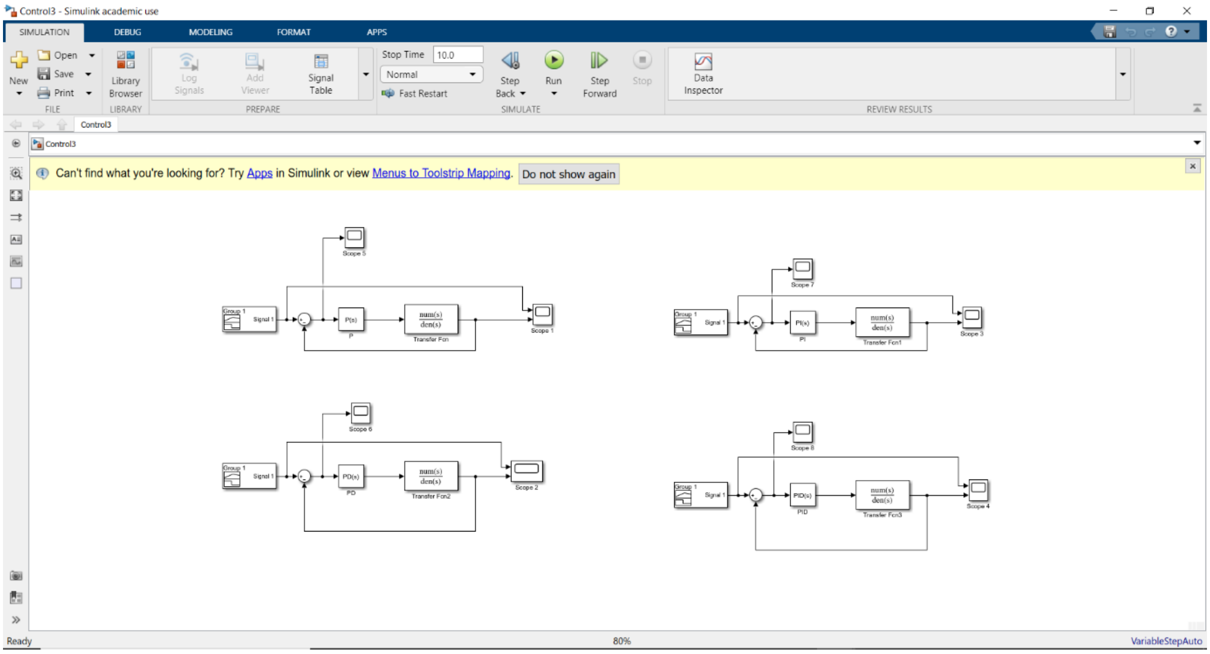


Figure 2-2 MATLAB Simulink interface [11]

2.3. EAGLE Autodesk

EAGLE [12] is a scriptable electronic design automation (EDA) application with schematic capture, printed circuit board (PCB) layout, auto-router and computer-aided manufacturing (CAM) features. EAGLE stands for Easily Applicable Graphical Layout and is developed by CadSoft Computer GmbH. EAGLE contains a schematic editor, for designing circuit diagrams. Schematics are stored in files with .SCH extension, parts are defined in device libraries with .LBR extension. Parts can be placed on many sheets and connected together through ports. The PCB layout editor stores board files with the extension .BRD. It allows back-annotation to the schematic and auto-routing to automatically connect traces based on the connections defined in the schematic. In this thesis, circuit diagram and PCB designs implemented on this program. See Figure 2.3.

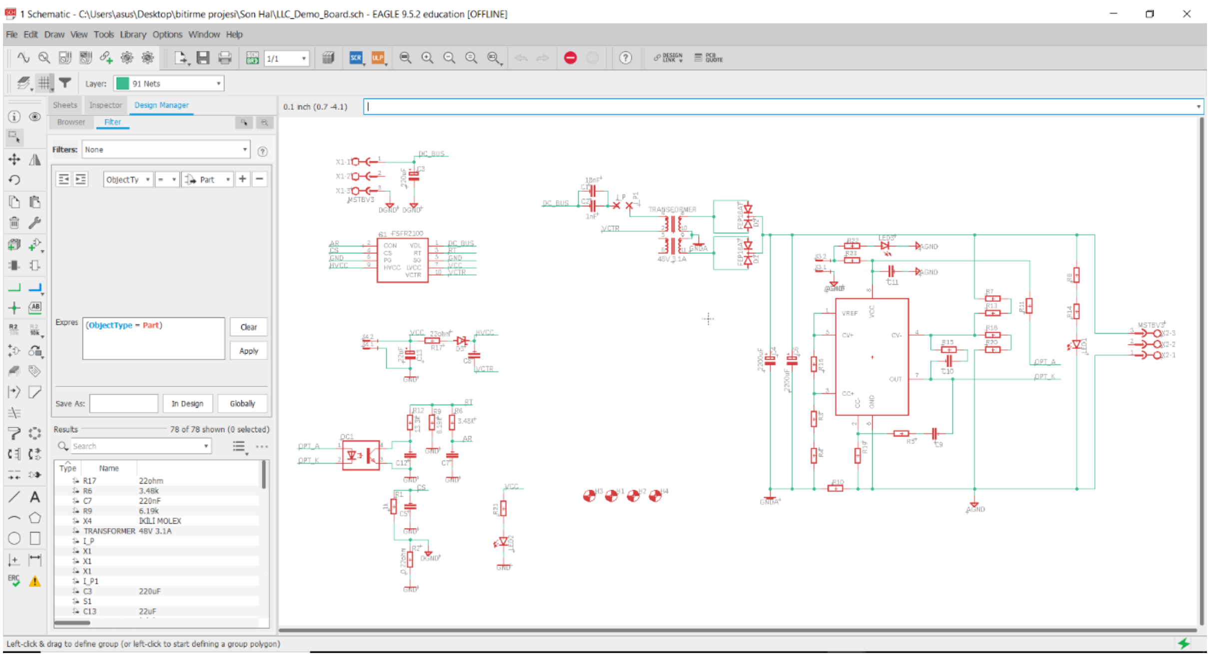


Figure 2-3 EAGLE Autodesk interface [12]

2.4. Python and Spyder

Python [13] is an interpreted, high-level, general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Spyder [14] is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers, and data analysts. It features a unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. Furthermore, Spyder offers built-in integration with many popular scientific packages, including NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy, and more. Beyond its many built-in features, Spyder can be extended even further via third-party plugins. Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console or advanced editor, in your own software. Python programming language is used for the analysis of the consumption data. See Figure 2.4.

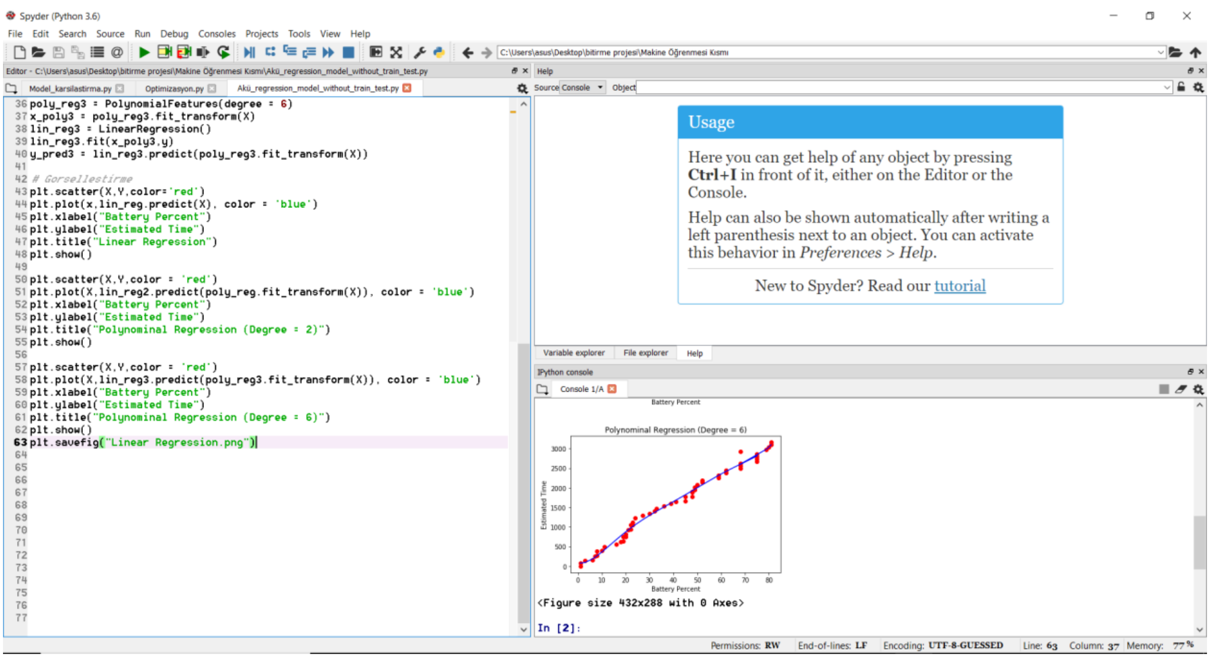


Figure 2-4 Spyder interface [14]

3. METHODS

3.1. Circuit Design

A dc/dc converter with 148.8W/48V output has been selected as a design for UPS battery charger. The design specifications are as follows:

- Input voltage: 400Vdc (output of PFC stage)

- Output: 48V/3.1A (148.8W)

- Holdup time requirement: 20ms (50Hz line freq.)

- DC link capacitor of PFC output: 220 $\boldsymbol{\mu F}$

[STEP-1] Define the system specifications

-Estimated efficiency (Eff): The power conversion efficiency must be estimated to calculate the maximum input power with a given maximum output power. If no reference data is available, use Eff = 0.88~0.92 for low voltage output applications and Eff = 0.92~0.96 for high voltage output applications. See Equation 3.1

$$P_{in} = \frac{P_o}{E_{ff}} \tag{3.1}$$

-Input voltage range ($\boldsymbol{V_{in}^{min}}$and $\boldsymbol{V_{in}^{max}}$) : Typically, it is assumed that the input voltage is provided from Power Factor Correction (PFC) pre-regulator output. When the input voltage is supplied from PFC output, the minimum input voltage considering the hold-up time requirement is given. See Equation 3.2

$$V_{in}^{min} = \sqrt{V_{O.PFC}^{\ 2} - \frac{2P_{in}T_{HU}}{C_{DL}}} \tag{3.2}$$

where VO.PFC is the nominal PFC output voltage, THU is a hold up time and CDL is the DC link bulk capacitor. The maximum input voltage is given. See Equation 3.3

$$V_{in}^{max} = V_{O.PFC} \tag{3.3}$$

[STEP-2] Determine the maximum and minimum voltage gains of the resonant network

It is typical to operate the LLC resonant converter around the resonant frequency (f_o) in normal operation to minimize switching frequency variation. When the input voltage is supplied from the PFC output, the input voltage has the maximum value (nominal PFC output voltage) in normal operation. Designing the converter to operate at f_o for the maximum input voltage condition, the minimum gain should occur at the resonant frequency (f_o). The gain at f_o is a function of the ratio (k=L_m/L_{lkp}) between the magnetizing inductance and primary side leakage inductance. Thus, the value of k should be chosen to obtain the minimum gain. While a higher peak gain can be obtained with a small k value, too small k value results in poor coupling of the transformer and deteriorates the efficiency. It is typical to set k to be 5~10, which results in a gain of 1.1~1.2 at the resonant frequency (f_o). With the chosen k value, the minimum voltage gain for maximum input voltage (V_{in}^{max}) is obtained. See Equation 3.4

$$M^{min} = \frac{V_{RO}}{\frac{V_{in}^{max}}{2}} = \frac{L_m+L_{lkp}}{L_m} = \frac{k+1}{k} \tag{3.4}$$

Then, the maximum voltage gain is given. See Equation 3.5

$$M^{max} = \frac{V_{in}^{max}}{V_{in}^{min}} . M^{min} \tag{3.5}$$

The ratio (k) between L_m and L_{lkp} is chosen as 9.4.

[STEP-3] Determine the transformer turns ratio (n=Np/Ns)

Since the full-wave bridge rectifier is used for the rectifier network, the transformer turns ratio is given. See Equation 3.6

$$n = \frac{N_p}{N_s} = \frac{V_{in}^{Max}}{2(V_o+2V_f)} M^{min} \tag{3.6}$$

[STEP-4] Calculate the equivalent load resistance (R_{ac})

With the transformer turns ratio obtained, the equivalent load resistance is obtained.
See Equation 3.7

$$R_{ac} = \frac{8n^2}{\pi^2}\frac{V_o^2}{P_o}E_{ff} \tag{3.7}$$

[STEP-5] Design the resonant network

With k chosen in STEP-2, read proper Q value from the peak gain curves in Figure 8 that results in enough peak gain. 10~15% margin on the peak gain is typical. Then, the resonant parameters are obtained. See Equation 3.8

$$L_p = L_m + L_{lkp} = (k+1)L_{lkp}$$

$$L_r = L_{lkp} + L_m//L_{lkp} = L_{lkp}\left(1 + \frac{k}{k+1}\right)$$

$$C_r = \frac{1}{2\pi Q \cdot f_o \cdot R_{ac}} \tag{3.8}$$

$$L_r = \frac{1}{(2\pi f_o)^2 C_r}$$

$$L_p = \frac{(k+1)^2}{(2k+1)}L_r$$

As calculated in STEP-2, the maximum voltage gain (M_{max}) for the minimum input voltage (V_{in}^{min}) is 1.22. With 10% margin, a peak gain of 1.5 is required. k has been chosen as 9.4 in STEP-2 and Q is obtained as 0.302 from the peak gain curves. By selecting the resonant frequency as 110kHz.

[STEP-6] Design the transformer

The worst case for the transformer design is the minimum switching frequency condition, which occurs at the minimum input voltage and full load condition. To obtain the minimum switching frequency, plot the gain curve using the gain equation and read the minimum switching frequency. Then, the minimum number of turns for the transformer primary side is obtained. See Equation 3.9

$$N_p^{min} = \frac{n(V_0 + 2V_f)}{2f_s^{min}\Delta B A_e} \tag{3.9}$$

where Ae is the cross-sectional area of the transformer core in m2 and ΔB is the maximum flux density swing in Tesla. If there is no reference data, use ΔB =0.25~0.30 T. Then, choose the proper number of turns for the secondary side that results in primary side turns larger than N_p^{min} as given. See Equation 3.10

$$N_p = n \,.\, N_s > N_p{}^{min} \tag{3.10}$$

Ae = 97.1 mm2 is selected for the transformer. From the gain curve in Figure 9 of the minimum switching frequency is obtained as 85kHz.

[STEP-7] Select the resonant capacitor

When choosing the resonant capacitor, the current rating should be considered since a considerable amount of current flows through the capacitor. The RMS current through the resonant capacitor is given. See Equation 3.11

$$I_{cr}^{\mathrm{Rms}} \cong \sqrt{\left[\frac{\pi I_O}{2\sqrt{2}n}\right]^2 + \left[\frac{n(V_O+2\cdot V_r)}{4\sqrt{2}f_O L_m}\right]^2} \tag{3.11}$$

Then, the maximum voltage of the resonant capacitor in normal operation is given. See Equation 3.12. See Figure 3.1.

$$V_{C_r}^{max} \equiv \frac{V_{in}^{max}}{2} + \frac{\sqrt{2}\cdot I_{cr}^{RMS}}{2\cdot\pi\cdot f_0\cdot C_r} \tag{3.12}$$

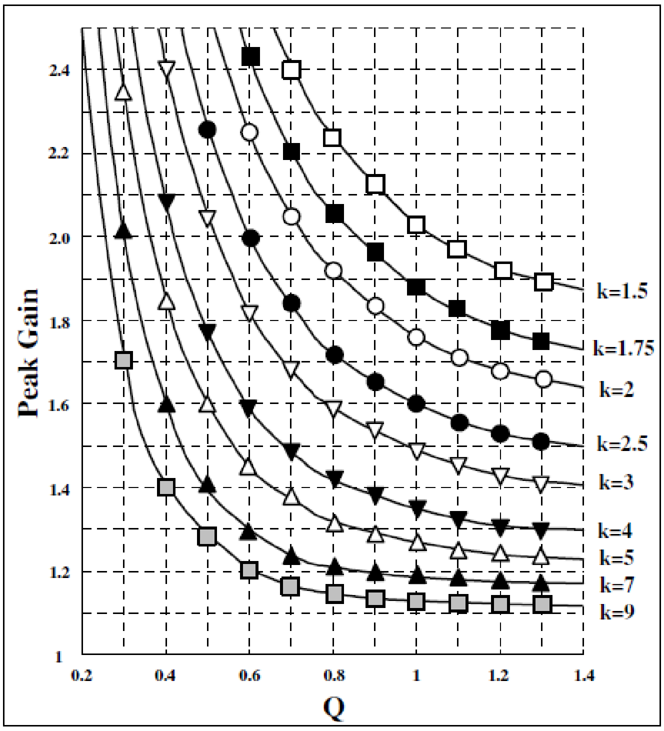


Figure 3-1 Peak Gain (Attainable Maximum Gain) vs. Q for Different k Values

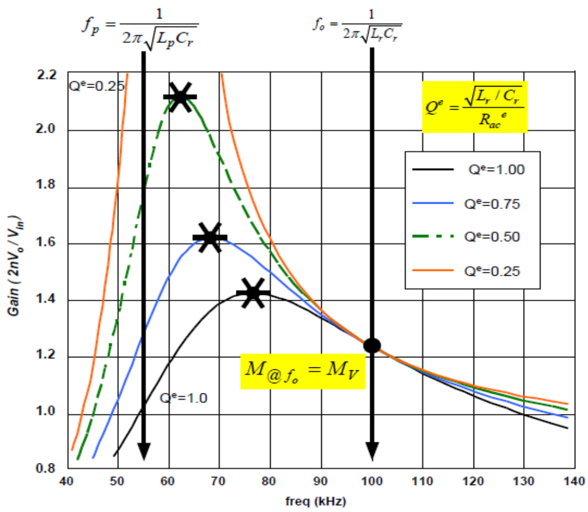


Figure 3-2 Typical Gain Curves of LLC Resonant Converter

Following flowchart diagram was created for the design procedure and values in were obtained as the design results. See Figure 3.2, 3.3 and 3.4.

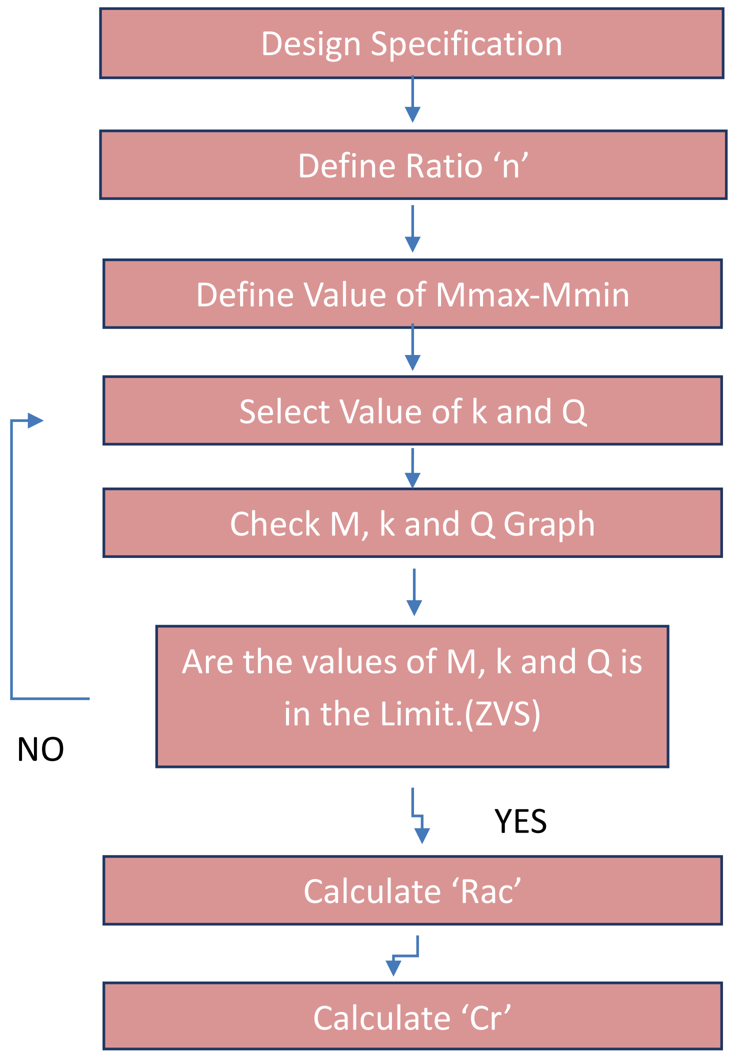


Figure 3-3 Flowchart diagram for design procedure

Design results are calculated as;

E_{ff} = %96	V_{in}^{min}= 363.276V	k = 9.4
P_{in} = 150 W	V_{in}^{max}= 400 V	
P_{out} = 148.8W	M^{min} = 1.1066	
$n = 4.375$	M^{max} = 1.22	
R_{ac} = 251.9 Ω	C_r = 19.03 nF	
L_r = 110 uH	L_p = 600 uH	
L_{lkp} = 57.78 uH	L_{lks} = 2.879 uF	
L_m = 542.2177 uH	N_p= 36	
N_s = 8	I_{Cr}^{RMS} = 1.044 A	
V_{Cr}^{Max} = 495 V	Q = 0.302	

Figure 3-4 Circuit design results

3.2. Simulation

After finding the theoretical results, the simulation of the converter to compare theoretical results and simulation results was built. Typical waveforms which it has to be seen on the simulation program. I_p is primary side leakage inductance current, I_m is magnetizing current I_{ds2} and $V_{ds2}$are second MOSFET current and voltage and I_d is rectifier side current. As you see, the square wave generator produces a square wave voltage, Vd by driving switches, Q1 and Q2 with alternating 50% duty cycle for each switch. The square wave generator stage can be built as a full-bridge or half bridge type. In this thesis it was used as half bridge type. See Figure 3.5.

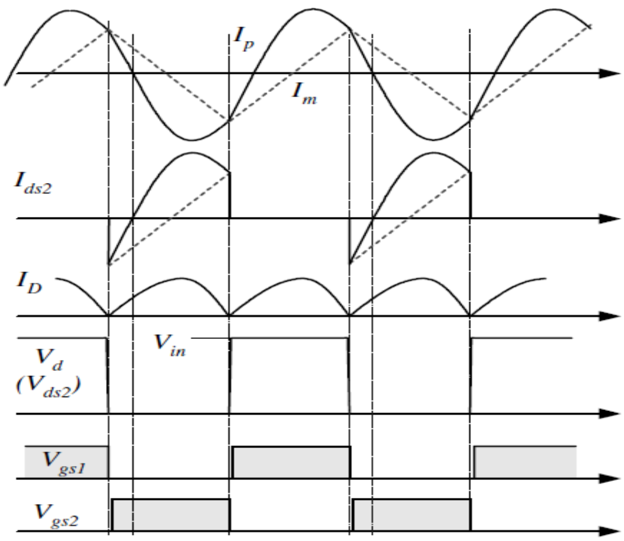


Figure 3-5 Typical voltage and current waveforms of half-bridge LLC resonant converter

Firstly, LLC Resonant topology was built in PSIM. The duty cycle is 0.5 and resonant frequency is 110kHz. In order to reach optimal design of LLC resonant converter and required output values, switching frequency was determined as above of resonance frequency, based on theoretical calculations and PSIM Simulation software. Therefore, the switching frequency is 115kHz. Obtained efficiency is 95.22% and that soft switching has been achieved. The intersection area between voltage and current of MOSFET is very less. See Figure 3.6 and 3.7

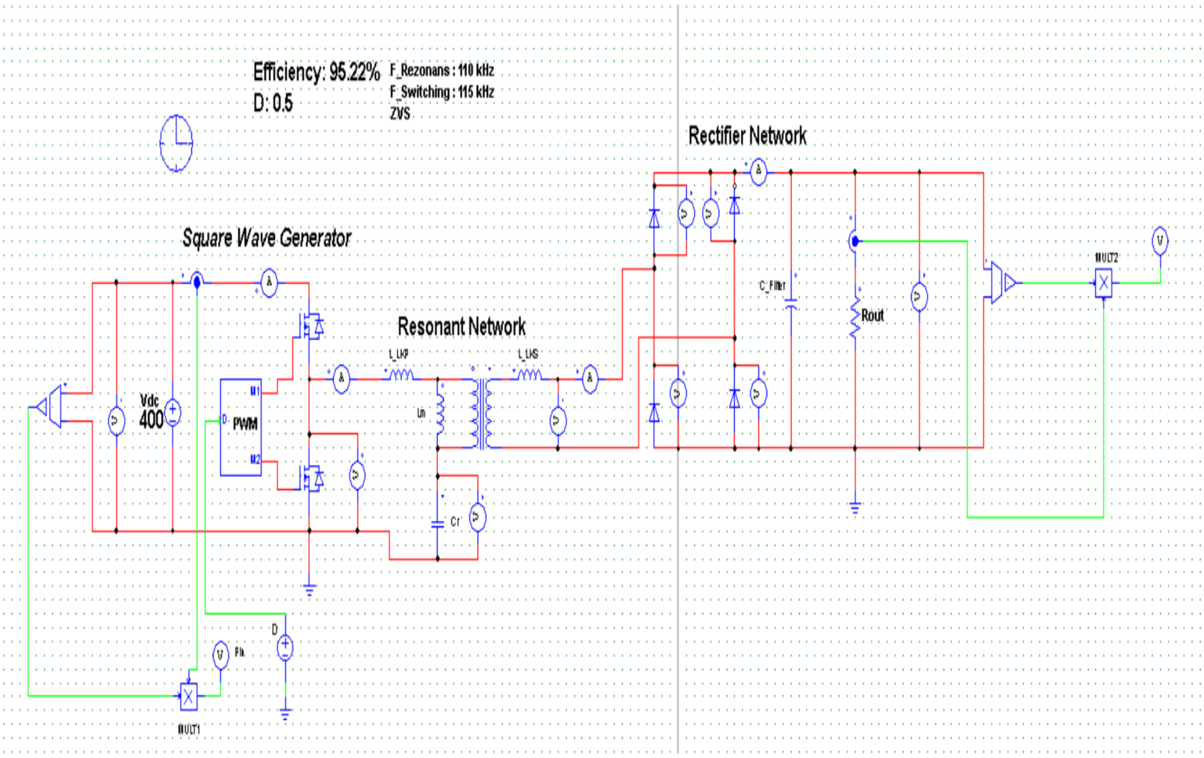


Figure 3-6 PSIM simulation design of LLC resonant DC-DC converter

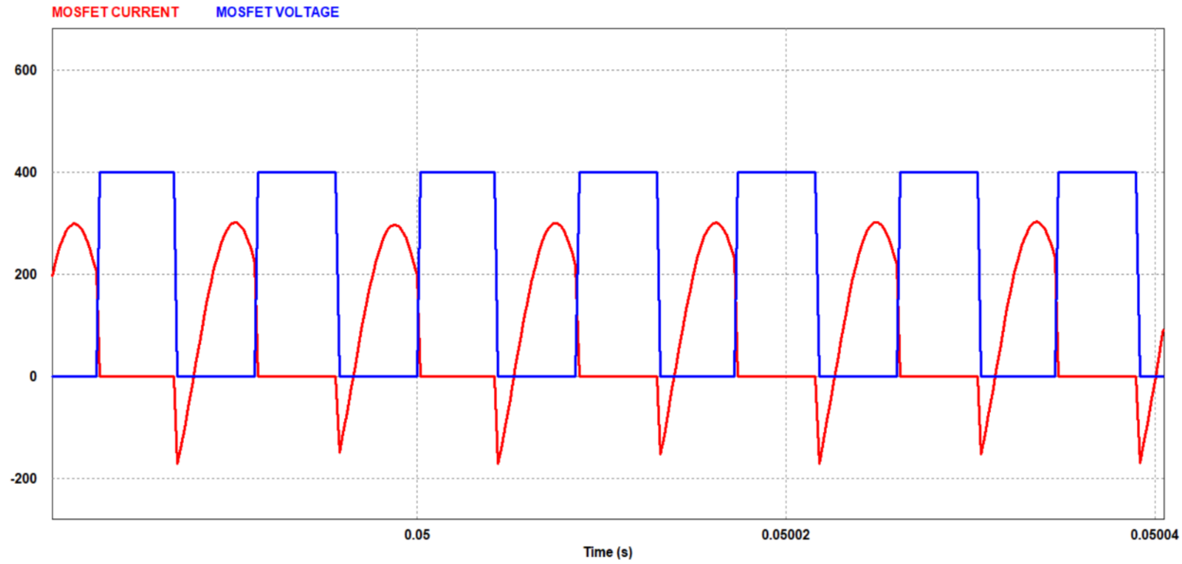


Figure 3-7 MOSFET voltage (blue) and current (red) achieving soft switching

The output result of the PSIM Simulation were obtained. The output voltage is 47.67V and the output current is 3.07A which are very close to required values. Besides, primary side leakage inductance current, magnetizing current, rectifier side current, and MOSFETs' voltage waveforms were obtained. Simulation results are like these typical waveforms. See Figure 3.8, 3.9, 3.10 and 3.11.

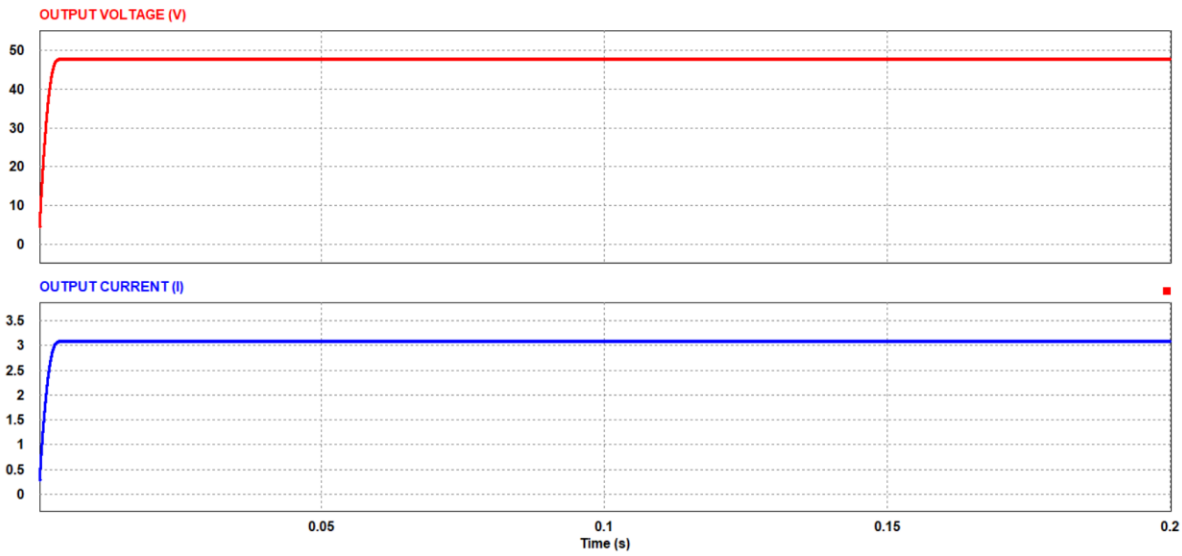


Figure 3-8 Output Voltage (red) and output current (blue) of the LLC Resonant Converter

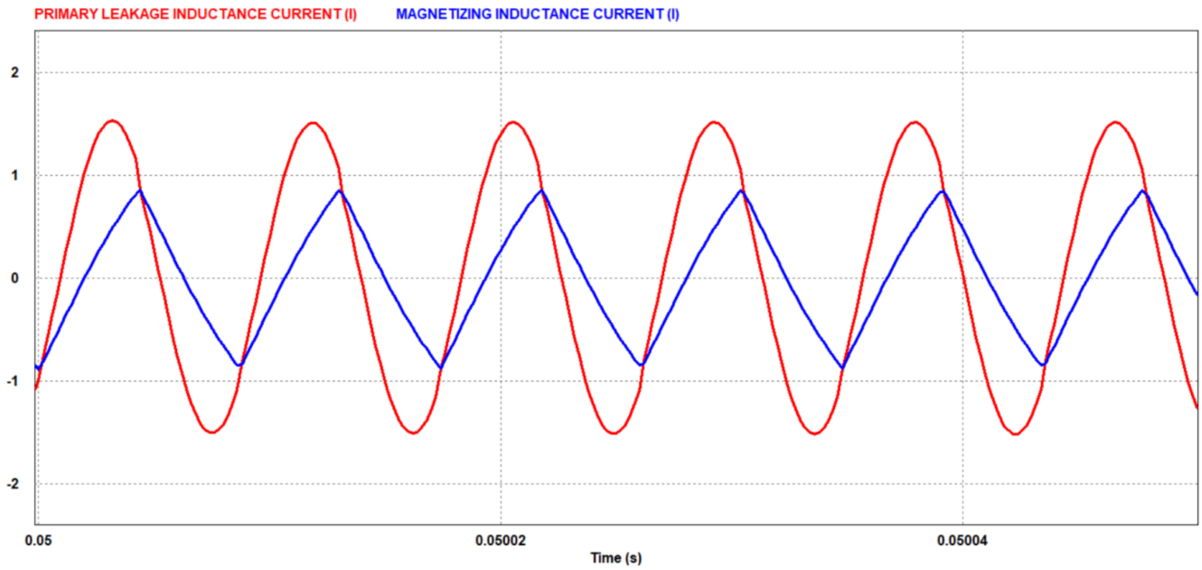


Figure 3-9 Magnetizing (blue) and primary side leakage inductor (red) current

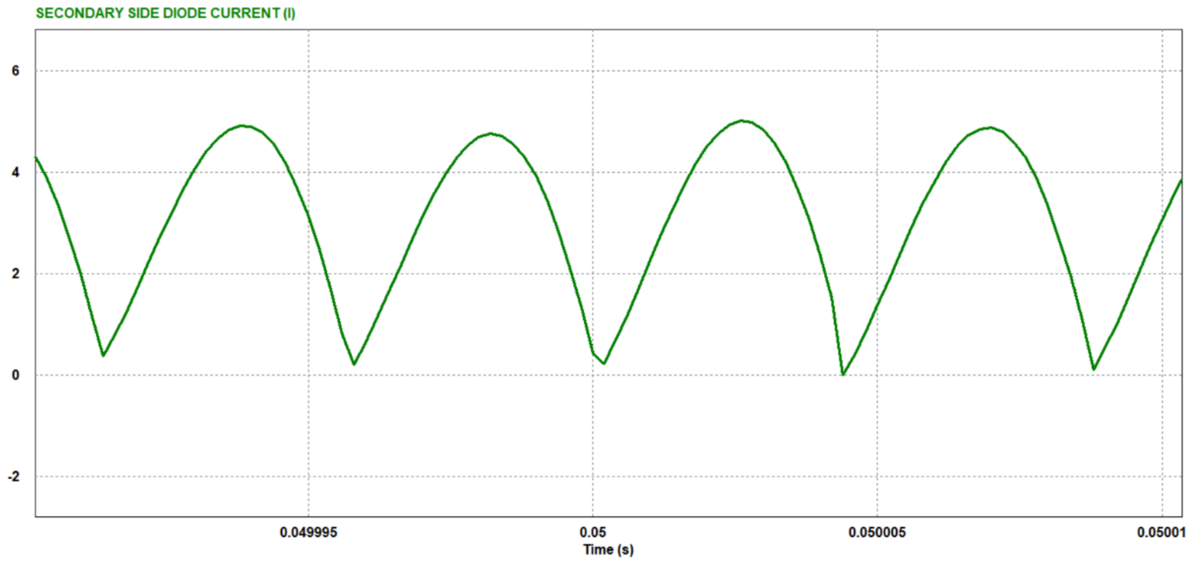


Figure 3-10 Secondary side diode current (A)

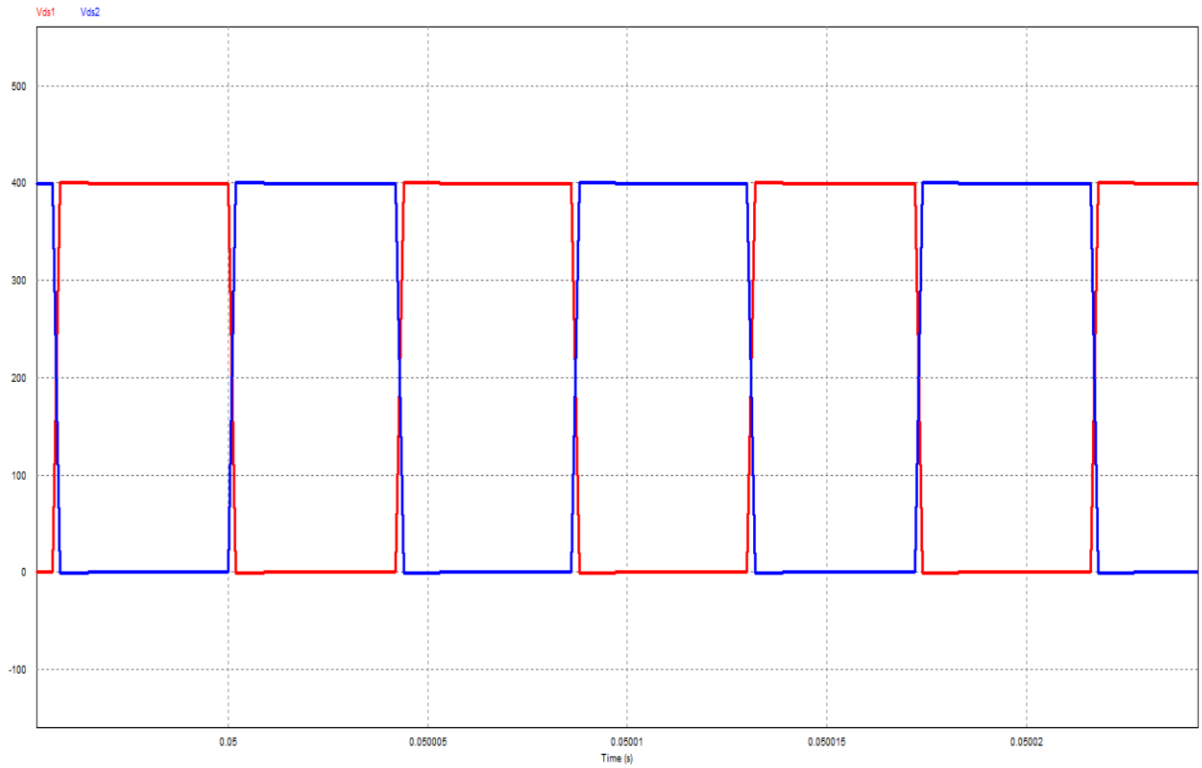


Figure 3-11 First and second MOSFET voltage (V)

The RMS current through the resonant capacitor was measured as 1.072A and the maximum voltage of the resonant capacitor in normal operation was measured as 311.5V. See Figure 3.12.

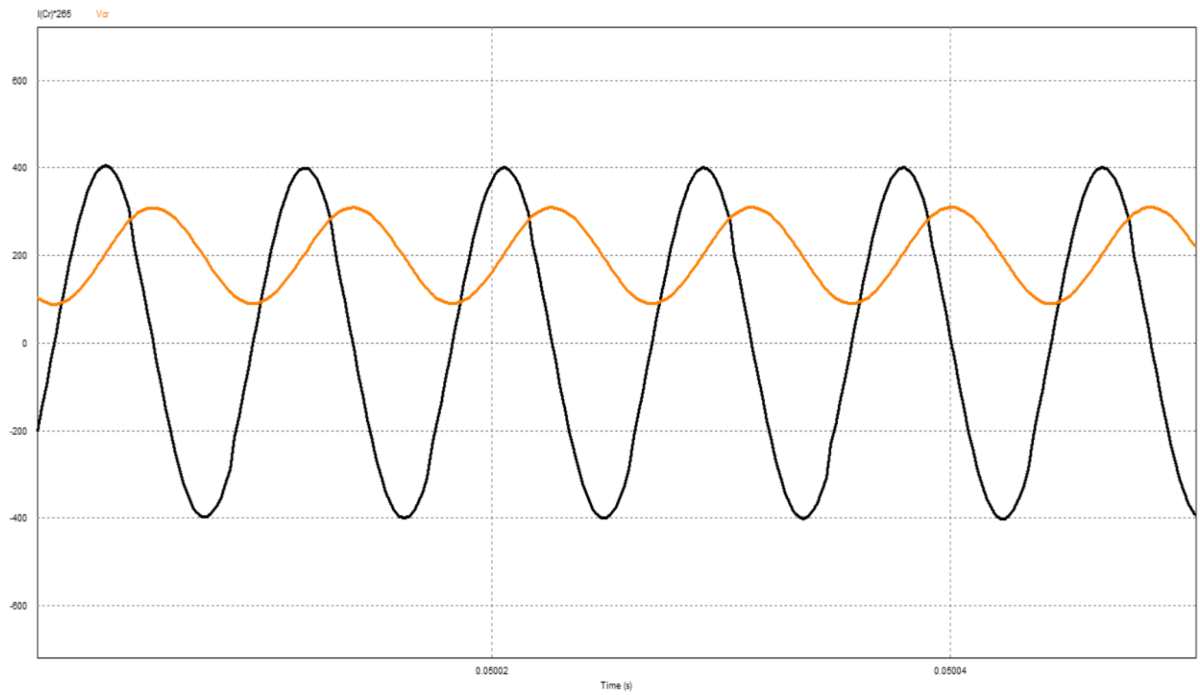


Figure 3-12 Current (black) and voltage (orange) waveforms of resonant capacitor

After that, the system was created on MATLAB Simulink. The required waveforms are created on MATLAB Simulink too. See Figure 3.13 and 3.14.

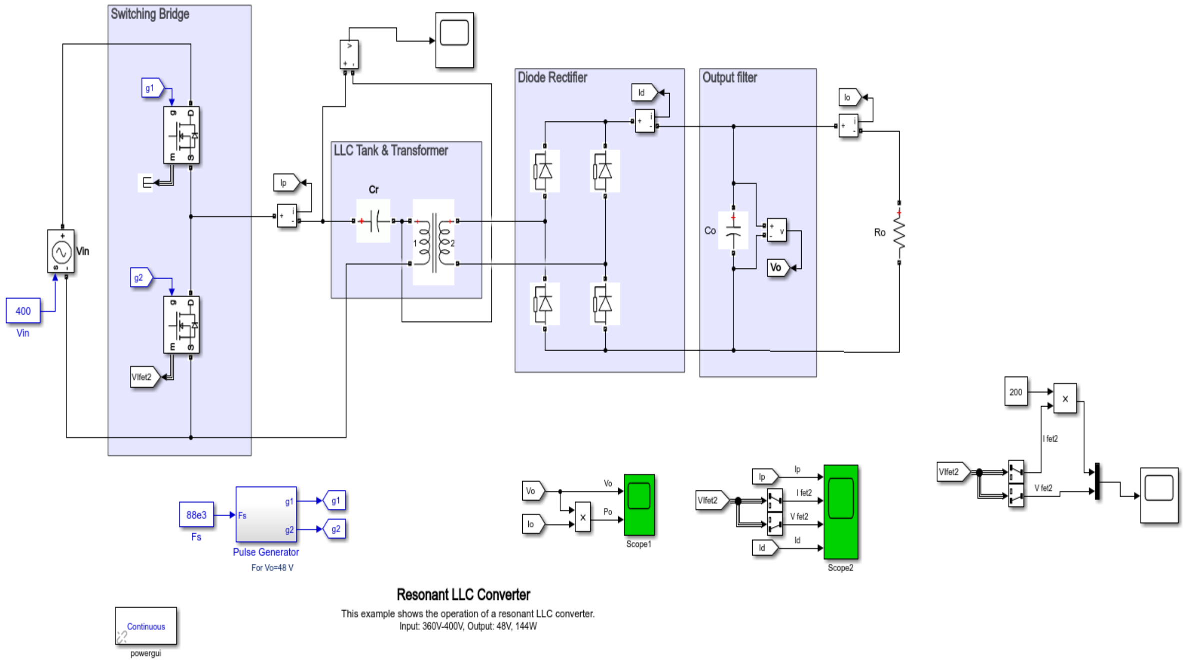


Figure 3-13 MATLAB Simulink design of LLC resonant DC-DC converter

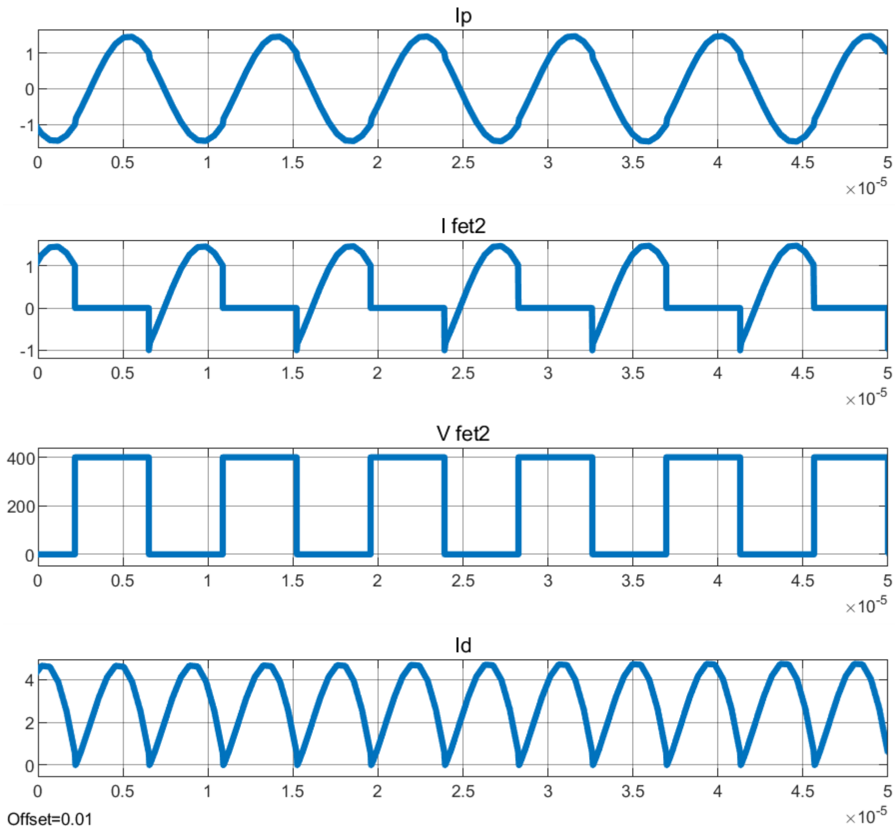


Figure 3-14 MATLAB Simulink design waveforms

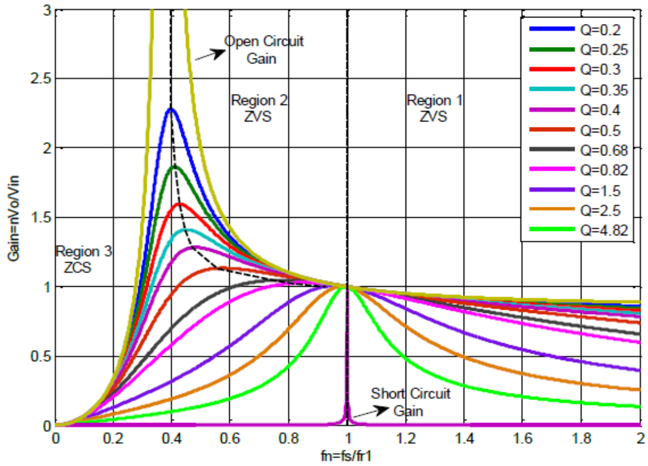


Figure 3-15 Typical dc voltage gain characteristic of LLC resonant converter as function of load and normalized switching frequency variation [15].

In this simulations, Region 1 ZVS was decided to operate which means switching frequency is bigger than resonant frequency. Therefore, switching frequency is selected the as 115kHz (calculated resonant frequency is 110kHz and Q = 0.302). Typical half-bridge LLC resonant converter waveforms should is obtained in the simulation program and the simulation results that are similar to these typical waveforms. Besides, the simulation results of the resonant capacitor maximum voltage and RMS current to select appropriate resonant capacitor and according to theoretical calculation, results are very similar which means theoretical calculations correspond to simulation results. Soft switching has been obtained succesfully. Finally, output voltage and output current in figure 6 are 47.67V and 3.07A in simulation which is very similar to theoretical values (48V/3.1A). In addition, simulation of LLC resonant converter has been obtained by MATLAB Simulink to be sure that calculation is correct, and results are almost the same. See Figure 3.15.

3.3. Hardware Design Part

FSFR2100XS was chosen as power unit for LLC resonant DC-DC converter and TSM1011 was chosen as control unit. See Figure 3.16 and 3.17.

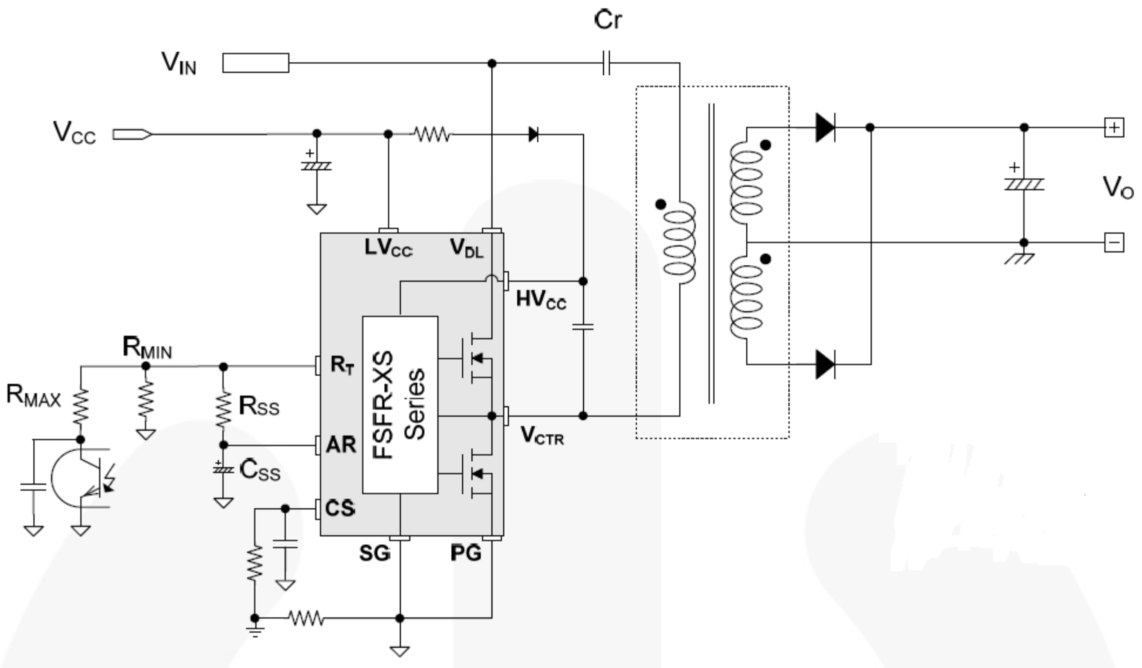


Figure 3-16 Final version of the circuit without secondary side control IC

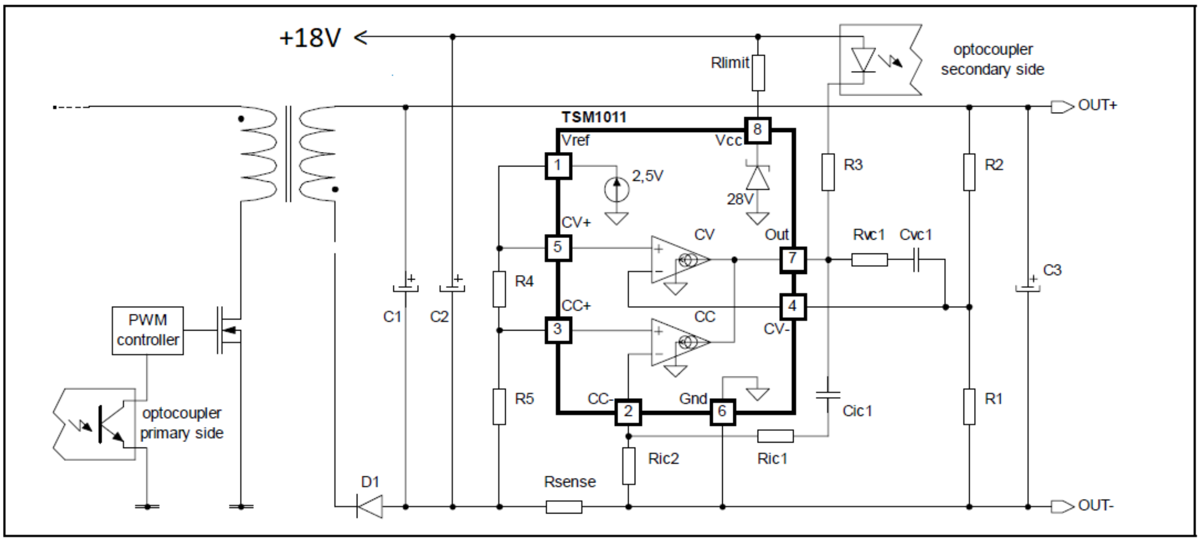


Figure 3-17 TSM1011 IC (Secondary side control unit)

The TSM1011 is a highly integrated solution for SMPS applications requiring CV (constant voltage) and CC (constant current) modes. The voltage reference combined with one operational amplifier makes it an ideal voltage controller. The other operational amplifier, combined with few external resistors and the voltage reference, can be used as a current limiter [16]. For Transformer, Würt Elektronik component transformer was used. It has turn ratio 35:8 and Input voltage is 360V to 400V, output voltage is 48V [17]. See Figure 3.18 and 3.19.

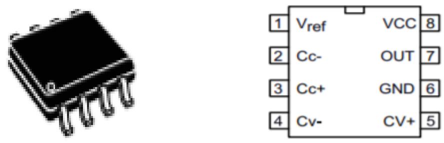


Figure 3-18 TSM1011 IC

D Electrical Properties:

Properties	Test conditions		Value	Unit	Tol.
Inductance	100 kHz/ 100 mV	L	600	µH	±10%
Turns ratio		n	35 : 8 : 8 : 3		±3%
Saturation current	I∆L/LI < 20%	I_{sat}	2.0	A	typ.
DC Resistance 1	@ 20°C	R_{DC1}	260	mΩ	max.
DC Resistance 2	@ 20°C	R_{DC2}	26.5	mΩ	max.
DC Resistance 3	@ 20°C	R_{DC3}	26.5	mΩ	max.
DC Resistance 4	@ 20°C	R_{DC4}	145	mΩ	max.
Leakage inductance	100 kHz/ 100 mV	L_S	100	µH	±10%
Insulation test voltage	W1,4 => W2,3	U_T	4000	V (AC)	

D2 Application Properties:

Properties		Value	Unit
Input voltage	U_I	360-400	V (DC)
Output voltage	U_O	48	V
Output current	I_O	3.1	A
Auxiliary voltage	U_{aux}	18.0	V
Switching frequency	f_{switch}	70-120	kHz

Figure 3-19 Transformer 760895451 and properties

The FSFR-XS series includes highly integrated power switches designed for high-efficiency half-bridge resonant converters. Offering everything necessary to build a reliable and robust resonant converter, the FSFR- XS series simplifies designs while improving productivity and performance. The FSFR-XS series combines power MOSFETs with fast-recovery type body diodes, a high- side gate-drive circuit, an accurate current controlled oscillator, frequency limit circuit, soft-start, and built-in protection functions. The high-side gate-drive circuit has common-mode noise cancellation capability, which guarantees stable operation with excellent noise immunity. The fast-recovery body diode of the MOSFETs improves reliability against abnormal operation conditions, while minimizing the effect of reverse recovery. using the zero-voltage-switching (ZVS) technique dramatically reduces the switching losses and significantly improves efficiency. The ZVS also reduces the switching noise noticeably, which allows a small- sized Electromagnetic Interference (EMI) filter [18]. See Figure 3.20.

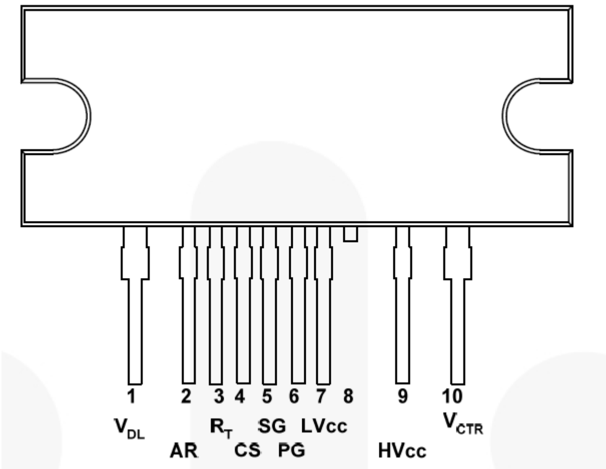


Figure 3-20 FSFR-XS package diagram

The design stages of the LLC resonant DC-DC converter card is made by using the Eagle software program. Necessary electronic components were researched according to the supplier websites (OZDISAN and DIGIKEY) provided by Tescom A.Ş. for drawing schematic and PCB board of the circuit on Eagle. According to the components found, it was checked whether the required footprint libraries were available in Eagle. If no suitable library was found, appropriate footprints were drawn. There were few appropriate footprints which must be drawn for the circuit. These footprints were drawn. For instance, the footprint of TSM1011 control IC should have been drawn because there was no appropriate footprint library on EAGLE. Later, the schematic of the circuit was created on Eagle Autodesk. See Figure 3.21 and 3.22.

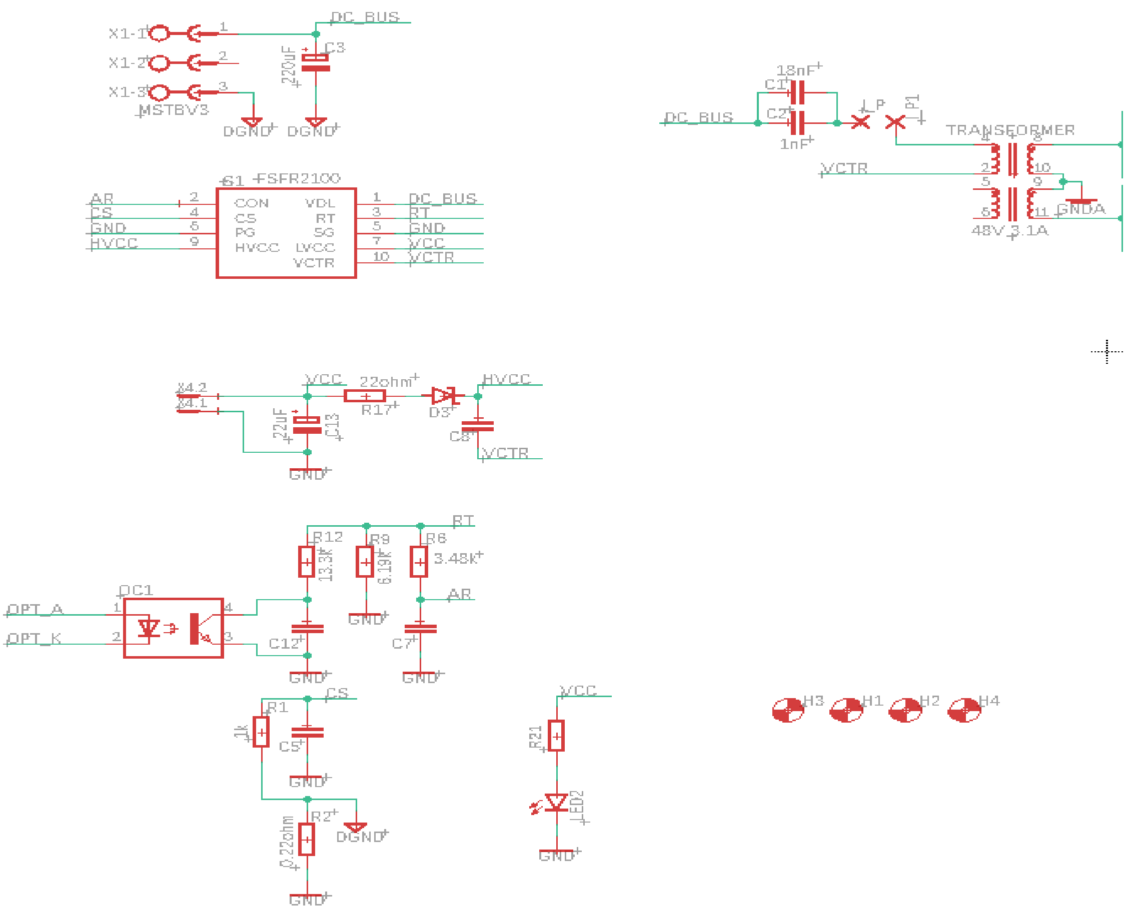


Figure 3-21 Primary side schematic of LLC Resonant Converter

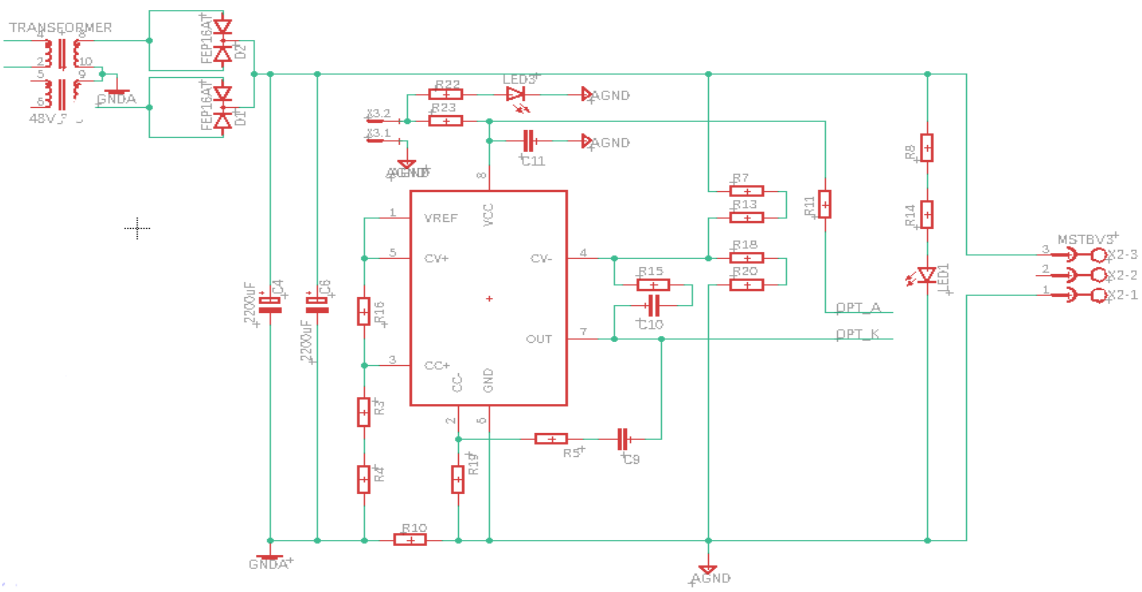


Figure 3-22 Secondary side schematic of LLC Resonant Converter

After the schematic parts is finished, PCB design of the converter is made. The area of the PCB has been determined as 10 cm x 10 cm. In addition, current values of all routes were measured in the simulation program. Because the routes of the card will be made thick or thin according to the current values. PCB composition is all about layering one material over another. The thickest, the middle part of the board is an insulating substrate. On either side of that is a thin layer of copper, where electric signals pass through. To insulate and protect the copper layers, they are covered with a thin layer of lacquer-like solder mask, which is what gives the PCB color. Finally, to top it all off, layer of ink-like silkscreen is added which can add text and logos to the PCB. In Eagle software has function as a switch from the schematic editor to the PCB board design. All the parts it was added from the schematic should be there, stacked on top of each other, ready to be placed and routed. See Figure 3.23 and 3.24.

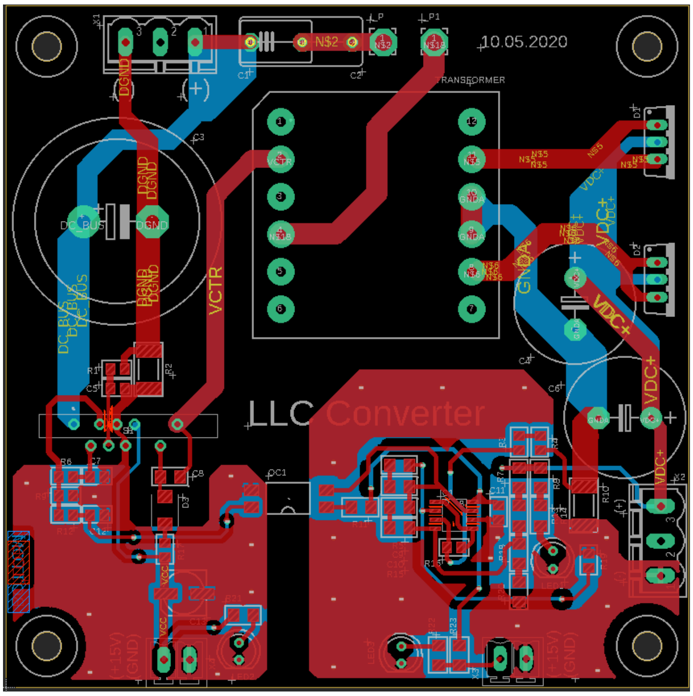


Figure 3-23 Bottom and top layer connections

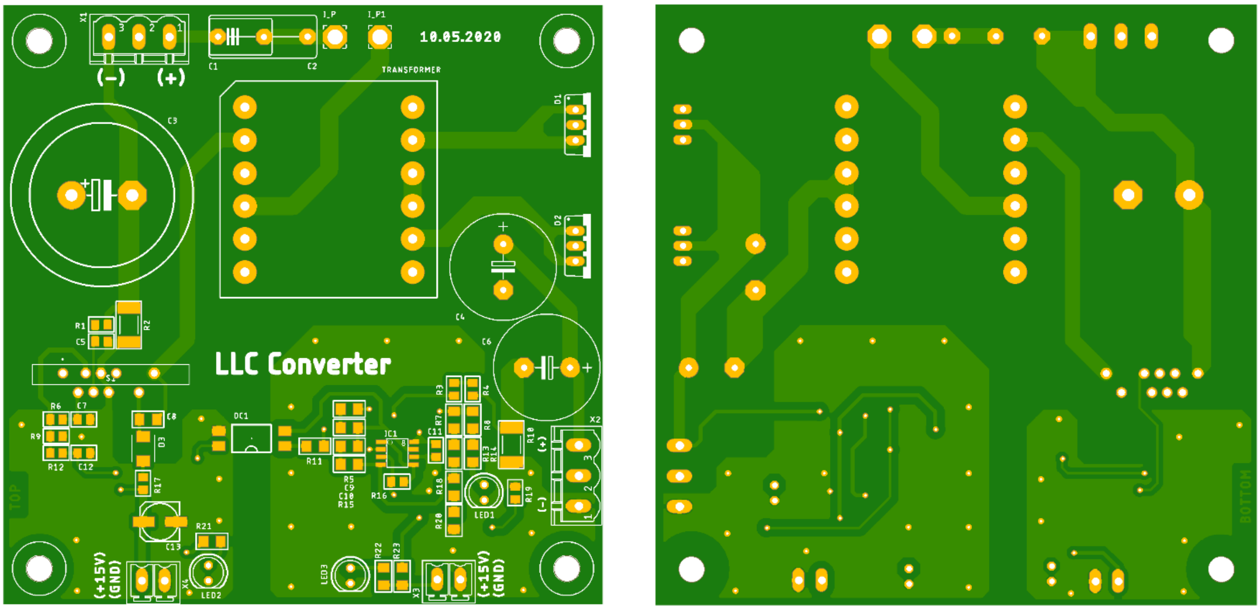


Figure 3-24 PCB Board of the LLC resonant converter

3.4. Battery Charge – Discharge Regression Model

3.4.1 Data Preprocessing

Data preprocessing is the initial step to be able to apply supervised machine learning algorithms successfully and to have better understanding of the data used. Completion of missing data, removal of unnecessary features, data normalization, merging or removing features that are correlated are the few examples which are commonly used for data preprocessing [19]. Min-max normalization is one of the most common ways to normalize data [20]. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1. See Equation 3.13.

$$z = \frac{x - min(x)}{max(x) - min(x)} \tag{3.13}$$

3.4.2 Support Vector Regression

Support Vector Machines in their current form were developed at the AT&T Bell Laboratories and gained momentum with the paper by Cortes and Vapnik (1995). Initial applications focused on binary classification of test instances and pattern recognition. Support Vector Machine Regression aims at finding a linear hyperplane, which fits the multidimensional input vectors to output values. The outcome is then used to predict future output values that are contained in a test set. Let us define a set of data points Let us define a set of data points $\mathrm{P} = (\mathrm{x_i}, \mathrm{a_i}), i = 1, \ldots\ldots.. n \ with \ x_i$ the input vector of data point i, $\mathrm{a_i}$ the actual value and n the number of data points. For linear functions f, the hyperplane that is constructed by the SVR is determined as follows. See Equation 3.14.

$$f(x) = wx + b \tag{3.14}$$

Notation-wise, equation (3.14) displays similarities to a linear regression model [21].

3.4.3 Linear Regression

Linear regression is a statistical approach for modeling the relationship between a dependent variable with a given set of independent variables [22]. The regression is used to identify the strength of the effect that the independent variable(s) have on a dependent variable. Also, it provides to understand how much the dependent variable changes with a change in one or more independent variables. In a simple linear regression model, a single response measurement Y is related to a single regressor X for each observation. The assumption of the model is that the conditional mean function is linear. See Equation 3.15.

$$Y = \alpha_0 + \alpha_1 x_1 \tag{3.15}$$

In most problems, more than one predictor variable is available [23]. This leads to the following multiple regression mean function. See Equation 3.16.

$$Y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots\ldots + \alpha_i x_i \tag{3.16}$$

where Y is the predicted output, xn (n=1,2,3,…,i) are the features, βn (n=0,1,2,…,i) are the constant parameters determined in the training phase of the model. βn coefficients are most important parameter for predict output variable accurately.

3.4.4 LightGBM Regression

LightGBM [24] is a gradient boosting framework that uses tree-based learning algorithms. There are several advantages. These are better accuracy, lower memory usage and faster training speed and higher efficiency etc.

3.4.5 XGB Regression

XGBoost [25] is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.

3.4.6 Gradient Boosting Regression

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees [26]. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

3.4.7 Random Forest Regression

Random forest is similar to decision tree in that samples are drawn to construct multiple trees; the difference is that the each tree is grown with a randomized subset of predictors, hence the name ‘‘random’’ forests. A large number of trees (500 to 2,000) are grown, hence a ‘‘forest’’ of trees. The number of predictors used to find the best split at each node is a randomly chosen subset of the total number of predictors. The number of predictors used to find the best split at each node is a randomly chosen subset of the total number of predictors. The trees are grown to maximum size without pruning, and aggregation is by averaging the trees [27]. Out of-bag samples can be used to calculate an unbiased error rate and variable importance, eliminating the need for a test set or cross-validation. Because a large number of trees are grown, there is limited generalization error which means that no overfitting is possible, a very useful feature for prediction.

3.4.8 Decision Tree Regression

Decision Trees are a non-parametric supervised learning method used for classification and regression [28]. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

3.4.9 MLP Regressor

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptron [29]. An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer [30]. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training.

3.4.10 K-Neighbors Regression

In pattern recognition, the k-nearest neighbors algorithm (k-NN) [31] is a non-parametric method used for classification and regression In both cases, the input consists of the k closest training examples in the feature space. K-NN is a type of instance-based learning or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

3.4.11 Error Metrics

In an accurate and reliable regression model, the aim is that there is no difference between the actual observation value and the estimated value or the difference is minimum. Various numerical measure methods are developed for this. Coefficient of determination (R2) is a numerical measure of some type of correlation, meaning a statistical relationship between two variables [32]. See Equation 3.17

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{m}(y_i - \grave{y}_i)^2}{\sum_{i=1}^{m}(y_i - \tilde{y}_i)^2}\right] \tag{3.17}$$

Root Mean Square Error (RMSE) is the standard deviation of the residuals prediction errors [33]. Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results. See Equation 3.18.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(y_i - \grave{y}_i)^2}{m}} \tag{3.18}$$

3.4.12 Dataset and Regression Models Results

Firstly, the dataset was obtained from TESCOM A.Ş. batteries. There are 4 different variables in the dataset which are discharge current, battery percentage, indoor temperature and Time. See Figure 3.25.

Index	Discharge_Current	Battery(+)	Battery(-)	Batt	Estimate_time	Temp
0	83	375	376	81	3161	24
1	83	374	375	81	3096	24
2	83	372	373	80	3037	24
3	83	369	370	79	2981	24
4	84	366	368	68	2921	24
5	84	367	368	75	2860	24
6	84	367	368	75	2797	24
7	84	367	368	75	2736	24
8	84	367	368	75	2677	24
9	84	366	368	68	2621	24
10	84	366	367	68	2556	24
11	84	366	367	68	2496	24
12	84	365	367	62	2441	24
13	84	365	366	62	2377	24
14	85	364	366	59	2316	24
15	85	364	365	59	2256	24
16	85	363	365	52	2197	24
17	85	363	364	52	2137	24
18	85	362	364	50	2077	24
19	85	361	363	49	2017	24
20	85	361	362	49	1957	24
21	85	361	362	48	1897	24
22	85	360	362	48	1893	24
23	85	360	361	48	1776	24

Figure 3-25 The dataset obtained from TESCOM A.Ş.

Next, the data set was tried with 10 different regression models in Python. Root Mean Square error and R2 Score comparison of the models and result were obtained. See Figure 3.26, See Table 3.1.

Table 3-1: Root Mean Square Error and R^2 Score Values of Regression Models

Results / Regression Models	Root Mean Square Error	R^2 Score
SVR	1057.0299	0.0008
Linear Regression	77.2032	0.9946
LGBM Regressor	465.0893	0.8065
XGB Regressor	93.0894	0.9922
Polynomial Regressor	70.5056	0.9956
Gradient Boosting Regressor	70.8098	0.9955
Random Forest Regression	67.843	0.9959
Decision Tree Regression	128.161	0.9853
MLP Regression	994.0306	0.1164
K-Neighbours Regressor	104.8204	0.9901

Finally, Random Forest has been decided as best model among 10 different regression models. After that, optimization and importance of variables algorithm was built. Originally, Random Forest regression model had 67.843 Root mean square error and 0.9959 R^2 score. After applying optimization algorithm, it has 55.4131 Root mean square error and 0.9973 R^2score. See Figure 3.27, 3.28 and 3.29.

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
from sklearn.preprocessing import scale
from sklearn.preprocessing import StandardScaler
from sklearn import model_selection
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import neighbors
from sklearn.svm import SVR
import xgboost
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from catboost import CatBoostRegressor


df = pd.read_csv("Battery.csv")


def compML(df,y,alg):
    # Train-Test Ayrımı
    y = df[y]
    X = df[["Batt","Discharge_Current","Temp","Battery(+)"]]
    X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=42)

    # Modelleme
    model = alg().fit(X_train,y_train)
    y_pred = model.predict(X_test)
    RMSE = np.sqrt(mean_squared_error(y_test,y_pred))
    r2score = r2_score(y_test,y_pred)
    model_ismi = alg.__name__
    print(model_ismi,'Modeli Test Hatası:', RMSE)
    print(model_ismi,'Modelin R2 Skoru:', r2score)
    return RMSE

compML(df,"Estimate_time",SVR)
```

```python
models = [LinearRegression,
          LGBMRegressor,
          XGBRegressor,
          GradientBoostingRegressor,
          RandomForestRegressor,
          DecisionTreeRegressor,
          MLPRegressor,
          KNeighborsRegressor,
          SVR]

for i in models:
    print(compML(df,"Estimate_time",i),"\n")
```

Figure 3-26 Regression models comparison code

```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split,GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.ensemble import RandomForestRegressor

df = pd.read_csv("Battery.csv")
y = df["Estimate_time"]
X = df[["Batt","Discharge_Current","Temp","Battery(+)"]]

# Train-Test Ayrımı
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=42)

# Modelleme
rf_model = RandomForestRegressor(random_state=42).fit(X_train,y_train)
y_pred = rf_model.predict(X_test)
RMSE = np.sqrt(mean_squared_error(y_test,y_pred))
r2score=r2_score(y_test,y_pred)

#Optimizasyon
rf_params = {"max_depth": [5,3],"max_features": [2,4],
             "n_estimators":[200,1000],"min_samples_split":[2,4]}
rf_cv_model = GridSearchCV(rf_model,rf_params,cv=10,n_jobs=-1,verbose=2).fit(X_train,y_train)
rf_cv_model.best_params_

#Tuned Model
rf_model = RandomForestRegressor(random_state=42,
                                 max_depth=5,
                                 max_features=4,
                                 min_samples_split=2,
                                 n_estimators=500)
rf_tuned=rf_model.fit(X_train,y_train)


y_pred = rf_tuned.predict(X_test)
RMSE_tuned = np.sqrt(mean_squared_error(y_test,y_pred))
r2_tuned = r2_score(y_test,y_pred)
```

Figure 3-27 Optimization code

```python
# Importances of Variables
Importance = pd.DataFrame({'Importance':rf_tuned.feature_importances_*100},
                          index=X_train.columns)

Importance.sort_values(by='Importance',
                       axis=0,
                       ascending = True).plot(kind = 'barh',
                                              color = 'b',)

plt.xlabel('Variable Importance')
plt.gca().legend_=None
```

Figure 3-28 Importance of variables code

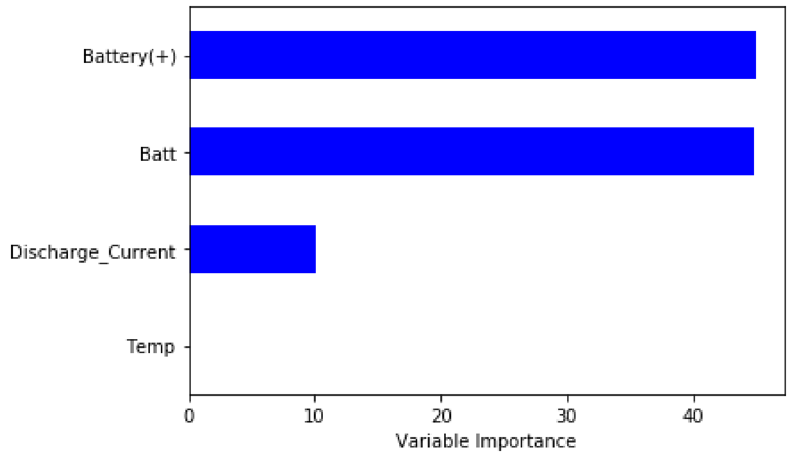
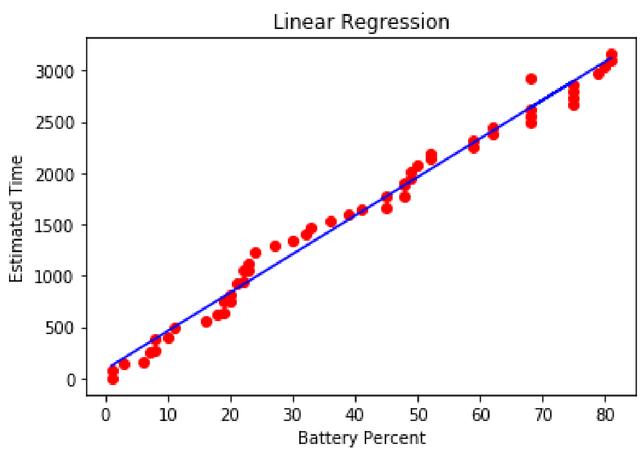
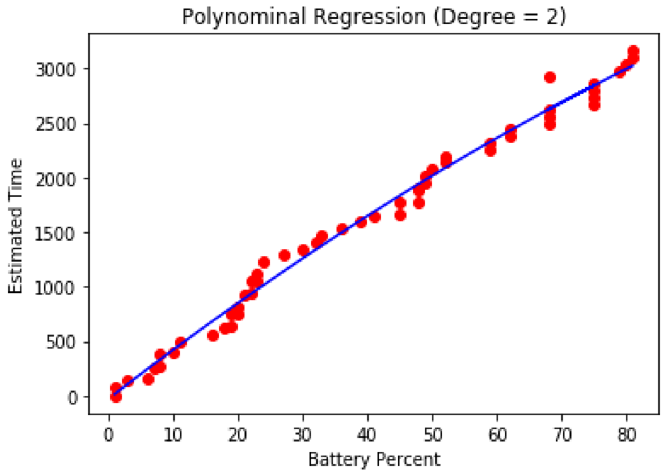


Figure 3-29 Importance of variables result

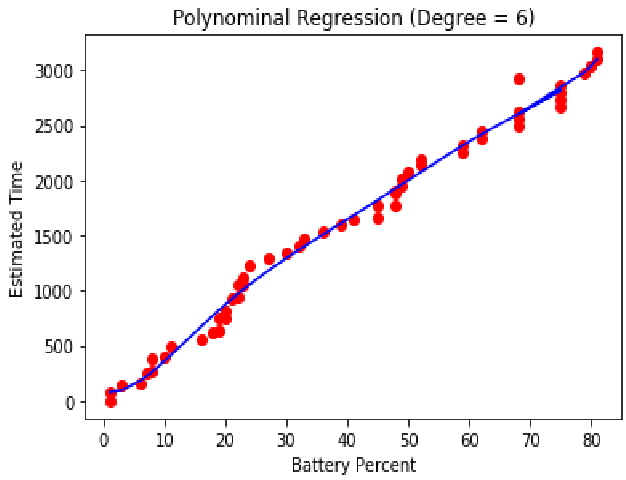
Later, model visualization code was built. As shown in the images, we have proved that Linear Regression, Polynomial Regression, Gradient Boosting Regression, Random Forest Regression, and XGB Regression are more successful models. Models were modeled to show how long the battery can be discharged according to the battery's charge rate. The conclusion we will draw from these images gives polynomial regression closer estimates than linear regression. When we compare polynomial degrees, six predicts better results than two degrees. However, this does not mean that as the degree increases, it always gives better results. It will be overfitting. According to the algorithm, 6 degrees gives the most ideal result. Finally, the reason why these models were created is to be able to predict how long a battery with a known charge rate can be discharged according to the load and its medium temperature. See Figure 3.30(a), 3.30(b), 3.30(c), 3.30(d), 3.31, 3.32, 3.33, 3.34, 3.35, 3.36, 3.347, 3.38, 3.39, and 3.40.



(a)



(b)



(c)

Figure 3-30 (a) Graph of Linear Regression Model, (b) Graph of Polynomial Regression Model (Degree = 2), (c) Graph of Polynomial Regression Model (Degree = 6)

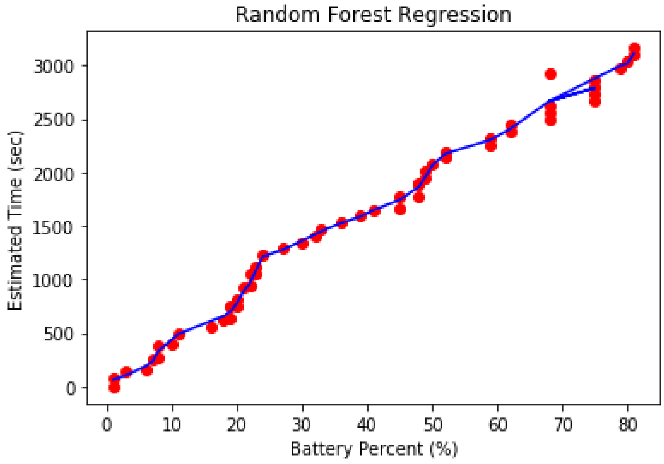


Figure 3-31 Graph of Random Forest Regression Model

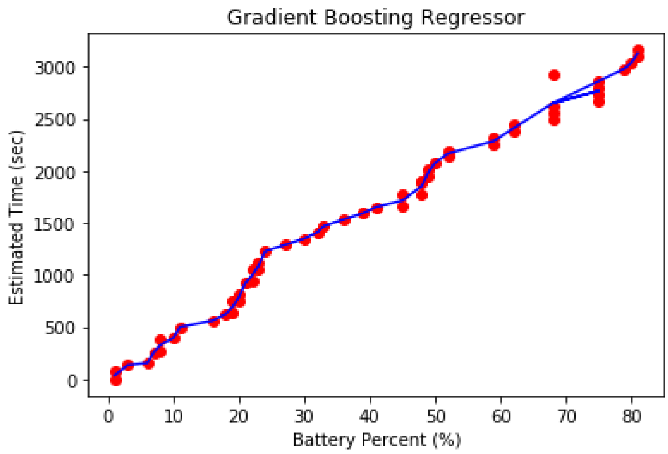


Figure 3-32 Graph of Gradient Boosting Regression Model

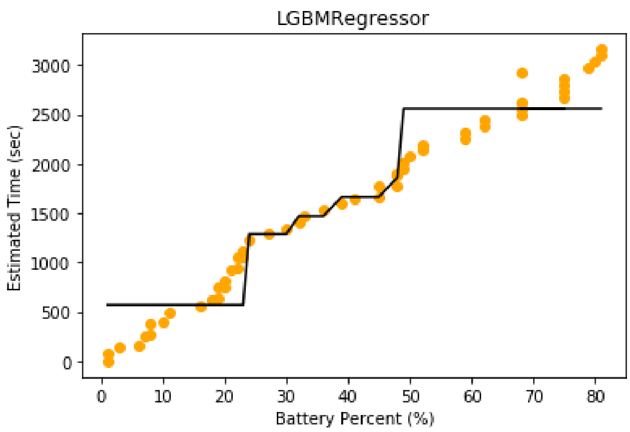


Figure 3-33 Graph of LightGBM Regression Model

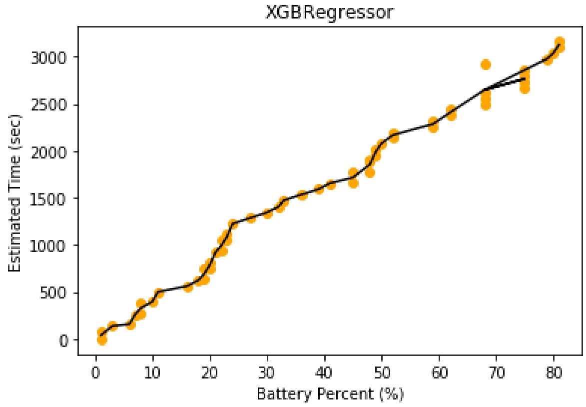


Figure 3-34 Graph of XGB Regression Model

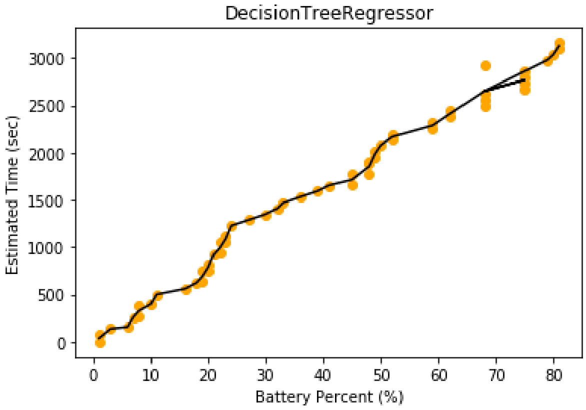


Figure 3-35 Graph of Decision Tree Regression Model

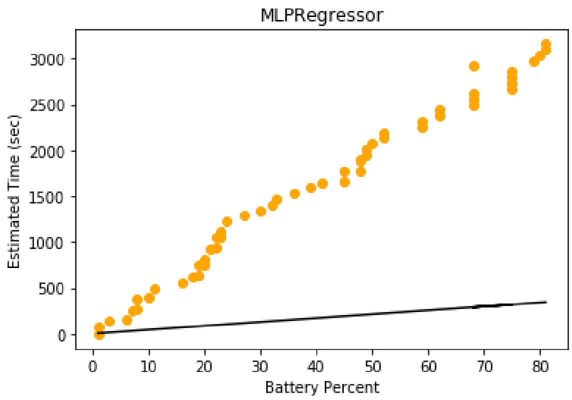


Figure 3-36 Graph of MLP Regression Model

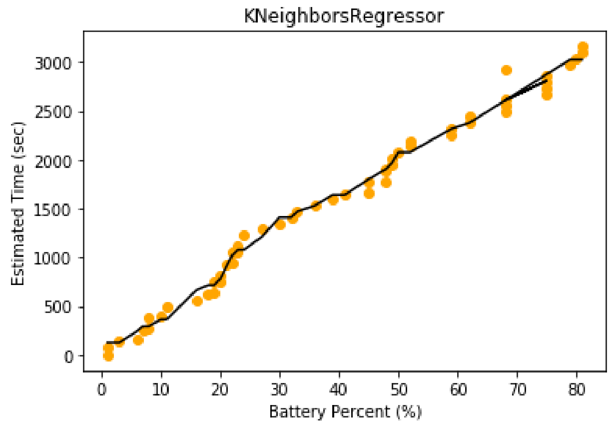


Figure 3-37 Graph of K-Neighbors Regression Model

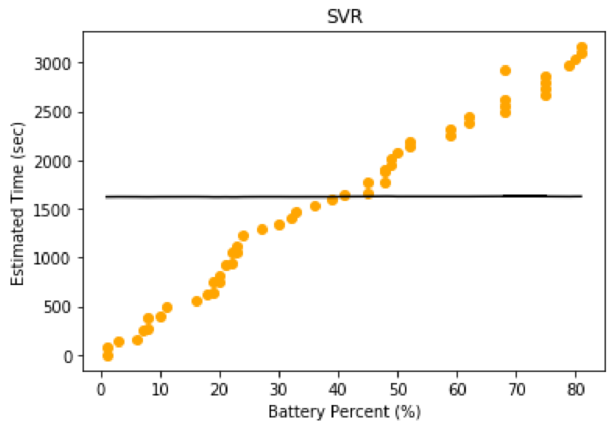


Figure 3-38 Graph of Support Vector Regression Model

```python
#1. kutuphaneler
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# veri yukleme
veriler = pd.read_csv('Battery.csv')

#data frame dilimleme (slice)
y = veriler.iloc[:,4:5]
x = veriler.iloc[:,3:4]

#NumPY dizi (array) dönüşümü
X = x
Y = y

#linear regression
#doğrusal model oluşturma
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X,Y)
y_pred = lin_reg.predict(X)

#polynomial regression
#doğrusal olmayan (nonlinear model) oluşturma
#2. dereceden polinom
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 2)
x_poly = poly_reg.fit_transform(X)
lin_reg2 = LinearRegression()
lin_reg2.fit(x_poly,y)
y_pred2 = lin_reg2.predict(poly_reg.fit_transform(X))

# 6. dereceden polinom
poly_reg3 = PolynomialFeatures(degree = 6)
x_poly3 = poly_reg3.fit_transform(X)
lin_reg3 = LinearRegression()
lin_reg3.fit(x_poly3,y)
y_pred3 = lin_reg3.predict(poly_reg3.fit_transform(X))
```

```python
# Gorsellestirme
plt.scatter(X,Y,color='red')
plt.plot(x,lin_reg.predict(X), color = 'blue')
plt.xlabel("Battery Percent")
plt.ylabel("Estimated Time")
plt.title("Linear Regression")
plt.show()

plt.scatter(X,Y,color = 'red')
plt.plot(X,lin_reg2.predict(poly_reg.fit_transform(X)), color = 'blue')
plt.xlabel("Battery Percent")
plt.ylabel("Estimated Time")
plt.title("Polynominal Regression (Degree = 2)")
plt.show()

plt.scatter(X,Y,color = 'red')
plt.plot(X,lin_reg3.predict(poly_reg3.fit_transform(X)), color = 'blue')
plt.xlabel("Battery Percent")
plt.ylabel("Estimated Time")
plt.title("Polynominal Regression (Degree = 6)")
plt.show()
plt.savefig("Linear Regression.png")
```

Figure 3-39 Model visualization code for linear and polynomial regression models

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn import neighbors
from sklearn.svm import SVR
import xgboost
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
# veri yukleme
veriler = pd.read_csv('Battery.csv')
#data frame dilimleme (slice)
y = veriler.iloc[:,4:5]
x = veriler.iloc[:,3:4]
#NumPY dizi (array) dönüşümü
X = x
Y = y
```

```python
# Random Forest
modelRF = RandomForestRegressor().fit(X,Y)
y_predRF = modelRF.predict(X)
# GradientBoostingRegressor
modelGBR = GradientBoostingRegressor().fit(X,Y)
y_predGBR = modelGBR.predict(X)
# LGBMRegressor
modelLGBR = LGBMRegressor().fit(X,Y)
y_predLGBR = modelLGBR.predict(X)
# XGBRegressor
modelXGBR = XGBRegressor().fit(X,Y)
y_predXGBR = modelXGBR.predict(X)
# DecisionTreeRegressor
modelDCR = DecisionTreeRegressor().fit(X,Y)
y_predDCR = modelDCR.predict(X)
# MLPRegressor
modelMLP = MLPRegressor().fit(X,Y)
y_predMLP = modelMLP.predict(X)
# KNeighborsRegressor
modelKn = KNeighborsRegressor().fit(X,Y)
y_predKn = modelKn.predict(X)
# SVR
modelSVR = SVR().fit(X,Y)
y_predSVR = modelSVR.predict(X)
```

```
# Gorsellestirme Random Forest
plt.scatter(X,Y, color = 'red')
plt.plot(x,y_predRF, color ='blue')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("Random Forest Regression")
plt.show()
# Gorsellestirme GradientBoostingRegressor
plt.scatter(X,Y,color='red')
plt.plot(x,y_predGBR, color = 'blue')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("Gradient Boosting Regressor")
plt.show()
# Gorsellestirme LGBMRegressor
plt.scatter(X,Y,color='orange')
plt.plot(x,y_predLGBR, color = 'black')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("LGBMRegressor")
plt.show()
# Gorsellestirme XGBRegressor
plt.scatter(X,Y,color='orange')
plt.plot(x,y_predXGBR, color = 'black')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("XGBRegressor")
plt.show()
# Gorsellestirme DecisionTreeRegressor
plt.scatter(X,Y,color='orange')
plt.plot(x,y_predDCR, color = 'black')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("DecisionTreeRegressor")
plt.show()
```

```
# Gorsellestirme MLPRegressor
plt.scatter(X,Y,color='orange')
plt.plot(x,y_predMLP, color = 'black')
plt.xlabel("Battery Percent")
plt.ylabel("Estimated Time (sec)")
plt.title("MLPRegressor")
plt.show()
# Gorsellestirme KNeighborsRegressor
plt.scatter(X,Y,color='orange')
plt.plot(x,y_predKn, color = 'black')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("KNeighborsRegressor")
plt.show()
# Gorsellestirme SUR
plt.scatter(X,Y,color='orange')
plt.plot(x,y_predSUR, color = 'black')
plt.xlabel("Battery Percent (%)")
plt.ylabel("Estimated Time (sec)")
plt.title("SUR")
plt.show()
```

Figure 3-40 Model visualization code for other regression models

4. SUMMARY AND CONCLUSION

In the proposed thesis, LLC resonant DC-DC converter was designed to achieve high efficiency in UPS battery charging applications. LLC resonant DC-DC converter was chosen because the LLC resonant converters have many advantages such as high-power efficiency, less switching losses, and operating in narrow switching frequency where zero voltage switching can be provided when compared with other converters features. The 48V-30Ah battery is charged with a prototype that operates with 48 V constant output voltage and 3.1A constant current output parameters with the design approach to be obtained. Besides, battery charge-discharge regression model is made by using machine learning algorithms, where battery status, battery electrical energy consumption and temperature data were analyzed. Root mean square error and R^2 score tests were performed for different regression models are generated in Python and the results of them are compared to each other. Random forest regression has been decided as the best regression among regression models for the obtained data set.

As a result, 95.22% power efficiency is achieved by the simulation program PSIM and battery unit of UPS in Tescom A.Ş. can be charged with high efficiency. Besides, it can be predicted how long a battery with a known charge rate can be discharged according to the load and its medium temperature.

REFERENCES

[1] Design Considerations for an LLC Resonant Converter Hangseok Choi Fairchild Semiconductor 82-3, Dodang dong, Wonmi-gu Bucheon-si, Gyeonggi-do, Korea

[2] A. F. Witulski and R. W. Erickson, “Design of the series resonant converter for minimum stress,”

[3] R. Oruganti, J. Yang, and F.C. Lee, “Implementation of Optimal Trajectory Control of Series Resonant Converters,”.

[4] V. Vorperian and S. Cuk, “A Complete DC Analysis of the Series Resonant Converter,”

[5] Y. G. Kang, A. K. Upadhyay, D. L. Stephens, “Analysis and design of a half-bridge parallel resonant converter operating above resonance,”

[6] R. Oruganti, J. Yang, and F.C. Lee, “State Plane Analysis of Parallel Resonant Converters,”

[7] M. Emsermann, “An Approximate Steady State and Small Signal Analysis of the Parallel Resonant Converter Running Above Resonance,”

[8] An LLC Resonant DC–DC Converter for Wide Output Voltage Range Battery Charging Applications Fariborz Musavi, Senior Member, IEEE, Marian Craciun, Member, IEEE, Deepak S. Gautam, Student Member, IEEE, Wilson Eberle, Member, IEEE, and William G. Dunford, Senior Member, IEEE

[9] LLC Resonant Converter for Battery Charging Application G. Subitha Sri and Dr. D. Subbulekshmi School of Electrical Engineering, VIT University, Chennai Associate professor, School of Electrical Engineering, VIT University, Chennai

[10] PSIM (Online) https://powersimtech.com/products/psim/

[11] MATLAB Simulink (Online) https://www.mathworks.com/products/simulink.html

[12] Eagle (Online) https://www.autodesk.com/products/eagle/overview

[13] Python (Online) https://www.python.org

[14] Spyder (Online) https://anaconda.org/anaconda/spyder

[15] High efficiency design approach of a LLC resonant converter for on-board electrical vehicle battery charge applications Sevilay ÇETİN

[16] TSM1011 https://www.st.com/en/power-management/tsm1011.html

[17] Transformer https://www.we-online.com/

[18] FSFR2100XS https://www.onsemi.com/products/power-management/ac-dc-controllers-regulators/offline-regulators/fsfr2100xs

[19] Elmaz, F., Yücel, Ö., & Mutlu, Evaluating the Effect of Blending Ratio on the Co-Gasification of High Ash Coal and Biomass in a Fluidized Bed Gasifier Using Machine Learning. Mugla Journal of Science and Technology, 5(1), 1-12.

[20] Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462.

[21] Support Vector Machine Regression for project control forecasting Mathieu Wautersa , Mario Vanhoucke

[22] Seber, George AF, and Alan J. Lee (2012). Linear regression analysis. Vol. 329. John Wiley & Sons

[23] Myers, R. H., & Myers, R. H. (1990). Classical and modern regression with applications (Vol. 2). Belmont, CA: Duxbury press.

[24] LGBM Regression (Online) https://lightgbm.readthedocs.io/en/latest/

[25] XGB Regression (Online) https://machinelearningmastery.com

[26] Breiman, L. (June 1997). "Arcing The Edge" . Technical Report 486. Statistics Department, University of California, Berkeley.

[27] Pal, M. (2005). Random forest classifier for remote sensing classification. International Journal of Remote Sensing, 26(1), 217-222

[28] Kamiński, B.; Jakubczyk, M.; Szufel, P. (2017). "A framework for sensitivity analysis of decision trees". Central European Journal of Operations Research. 26 (1): 135–159.

[29] Hastie, Trevor. Tibshirani, Robert. Friedman, Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009.

[30] Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986

[31] Altman, Naomi S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression" The American Statistician. 46 (3): 175–185.

[32] Nagelkerke, Nico JD. "A note on a general definition of the coefficient of determination." Biometrika 78.3 (1991): 691-692.

[33] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research, 30(1), 79-82.

PERSONAL INFORMATION

Turhan Can Kargın

Abdulhamid Yavuz Caddesi - Atifbey Mahallesi - No:31 Daire:3 Kat :3 - Gaziemir, 35410 Izmir (Turkey)

+905455291939

turhancan.kargin@gmail.com

https://www.linkedin.com/in/turhancankargin/ https://turhancankargin.wordpress.com/

Skype live:turhancans97

Sex Male | Date of birth 11/11/1997 | Nationality Turkish

EDUCATION AND TRAINING

09/2015–Present

Bachelor of Science (BSc), Electrical and Electronics Engineering

Izmir Katip Celebi University, Izmir (Turkey)

http://www.ikc.edu.tr/

- Intensive Intermediate English Coursework (2015-2016)
- Cumulative GPA: 3.6/4

Elective Courses

1) Power Electronics

2) Industrial Automation

3) Process Control and Instrumentation

4) Introduction to Machine Learning

5) Presentation Techniques

6) FPGA

09/2016–Present

Business Administration

Anadolu University, Eskisehir (Turkey)

www.anadolu.edu.tr/acikogretim

Faculty of Open University

Cumulative GPA: 2.71/4

17/02/2019–06/07/2019 Erasmus+ (Exchange) Program

Poznan University of Technology, Poznan (Poland)
https://www.put.poznan.pl/en/erasmus/erasmus-incoming-students

- I have attended the Erasmus+ Exchange Program, which allows the undergraduate students to take varied courses about their majors at different universities for a half year in Europe.

Courses:

1) Computer Networks

2) Optical Fiber Comm. Systems

3) Telecommunication Networks

4) Digital Signal Processing

5) Polish

- Semester Average Grade : 4.53/5

WORK EXPERIENCE

18/06/2018–13/07/2018 Production Intern

Tescom Elektronik San. ve Tic. A.Ş., İzmir (Turkey)
http://www.tescom-ups.com/

▪ I was a production intern at Tescom UPS. I have worked on different sections of the manufacturing: PCB Typesetting, Card Test, Cabling and Device Test

- Power Electronics
- Uninterruptible Power Supplies

Business or sector Uninterruptible Power Supply

05/08/2019–05/09/2019 Office Intern

ELPRO Enerji Sanayi ve Ticaret Limited Şirketi, İzmir (Turkey)
http://www.elpro.com.tr/

- Structure Electrical Installation Design
- AutoCad

Business or sector Electrical

03/07/2017–28/07/2017 Optional Intern

Maselmak Mühendislik, Ankara (Türkiye)

http://maselmak.com.tr/

- Electrical Panel Design (See Electrical)
- Tia Portal
- PLC

Business or sector Automation

16/12/2019–Present Part-Time Working Student Engineer

Grup ASM Endüstriyel Motor, Izmir (Turkey)

http://grup-ex.com/

- Ex-Proof Products

Business or sector Industrial Motor, Automotive, Machinery

PERSONAL SKILLS

Mother tongue(s) Turkish

Foreign language(s)

	UNDERSTANDING		SPEAKING		WRITING
	Listening	Reading	Spoken interaction	Spoken production	
English	C1	C1	C1	C1	C1
	Certificate of Attendance from Turkish-American Association				
German	B1	B1	B1	B1	B1
	Das Akademie				
Polish	A1	A1	A1	A1	
	Polish Language Course for Foreigners				

Communication skills

- Strong communication skills.
- Open-minded,
- Tolerant
- Good Listener
- Open to Criticism

Organisational / managerial skills

- IEEE IKCU Board Member (Treasurer) (400+ Members)
- IEEE IKCU Power and Energy Society Vice Chairman (Community with 50 active members)

Job-related skills

- To be able to produce solutions quickly
- Team work predisposition

Digital skills

SELF-ASSESSMENT				
Information processing	Communication	Content creation	Safety	Problem-solving
Proficient user	Independent user	Basic user	Basic user	Proficient user

Digital skills - Self-assessment grid

- Proteus Isis/Ares
- NI Multisim
- MATLAB
- Google SketchUp
- Arduino
- C Programming Language
- Microsoft Office Powerpoint/Excel/Word
- AutoCAD
- WireShark
- SoMachine
- ABB Automation Builder

- Power Electronics (DC-DC Converters)
- Eagle Autodesk
- PSIM
- Labview
- Python
- Machine Learning
- PIC Programming
- Raspberry Pi
- FPGA (VHDL)

Other skills

- Guitar
- Sport
- Football
- Fishing
- Basketball
- Photography
- Barista
- Travel Lover: Greece,Macedonia,Serbia,Hungary,Austria,Slovakia,Czechia,Germany,Holland,Belgium,France,Spain,Italy,Slovenia,Croatia,Bulgaria,Ukraine,Georgia,Bosnia Herzegovina,Montenegro,Albania,Kosovo.Poland,Sweden,Denmark,Portugal,Morocco,Norway

Driving licence B

ADDITIONAL INFORMATION

Projects

- Basic Uninterruptible Power Supply (2018)

We have designed basic UPS for Electronics-2 Lecture with my group.

- Smart Traffic Light (2017)

Smart Traffic Light with logic gates has been designed for Logic Design Lecture with my group.

- High Power Efficiency Design Approach of an LLC Resonant Converter for UPS Battery Charger Application and Battery Charge-Discharge Regression Model (2020- Graduation Project)

The purpose of the project is to design and produce an LLC resonant converter that will have 48V/3.1A output values to charge 4 x 12 V / 30 Ah batteries and obtaining a charge-discharge regression model of battery with machine learning algorithms.

- Climate Change: Prediction of Future Temperature (2019)

In this project, we used machine learning algorithms to see the rising of the average temperature in the Earth. We can predict the future average temperature for countries and besides, we can see the decline of the ice extent of north and south poles.

- OPC Server Based DC Motor Control with PLC Communication System (2019)

In this project, dc motor control was performed between NI Labview 2019 and ABB PM564 type PLC by using the OPC server.

- Controlling AC Motor Speed and Direction by Using PLC and STM32F4 Microcontroller with Matlab Graphical User Interface (2019)

MATLAB GUI is designed to make serial bi-directional communication with STM32 microcontroller and a uni-directional communication is made between PLC and STM32 microcontroller. After the connections, information from external sensors from other devices is used to drive AC Motor with the help of PLC.

- Erbium-Doped Fiber Amplifier (2019)

Optical Fiber Comm. Systems Lecture project.

Certifications

- Machine Learning with Python Udemy Certificate of Completion
- PCB Design with Autodesk Eagle Udemy Certificate of Completion
- The 3D Printing Revolution by University of Illinois (Coursera)
- 3D Printing Applications by University of Illinois (Coursera)
- Turkcell Arduino 101 Certificate of Achievement
- Turkcell Python 101 - 201 - 301 - 401 Certificate of Achievement
- Turkcell Data Visualization Certificate of Achievement
- Turkcell Data Manipulation 101 - 201 Certificate of Achievement
- Turkcell Statistics for Data Science 101 - 201 Certificate of Achievement
- Turkcell Data Pre-Processing Certificate of Achievement
- Turkcell Introduction to Data Science and Artificial Intelligence
- Turkcell Data Literacy Certificate of Achievement
- Turkcell Machine Learning 101 - 201 - 301 - 401 - 501 Certificate of Achievement
- Turkcell Big Data with Python Certificate of Achievement
- Turkcell Data Science Project Cycle Certificate of Achievement
- IZMIR-TAA C2 Level English Certificate of Attendance
- Polish Language Course Certificate for Foreigners
- Das Akademie B1.1 Level German Certificate
- KOSGEB Entrepreneurship Training Certificate of Participation
- ICASEM 2020 CERTIFICATE OF PARTICIPATION

Memberships

- The Chamber of Electrical Engineers (EMO) for Students
- IEEE Global Member
- IEEE PES Global Member
- IEEE Izmir Katip Celebi University

References

- Assoc. Prof. Savas Sahin

Department of Electrical And Electronics Engineering at Izmir Katip Celebi University / savas.sahin@ikcu.edu.tr