**Cuprins**

1 Rezumat în limba Engleză………………………………………………………………………

1.1 Theoretical Fundamentals……………………………………………………………..

1.1.1 Sources……………………………………………………………………….

1.1.1.1 Three-Phase Voltage Source……………………………………….

1.1.1.2 Mono-Phase Voltage Source……………………………………….

1.1.2 AC – DC Rectifier……………………………………………………………

1.1.3 DC-DC Convertor……………………………………………………………

1.1.3.1 DC – DC Boost…………………………………………………….

1.1.4 Battery………………………………………………………………………..

1.1.4.1 BMS………………………………………………………………..

1.1.4.2 Lithium-Ion………………………………………………………...

1.1.5 Invertor……………………………………………………………………..

1.1.6 CI/CD………………………………………………………………………...

1.1.7 GIT…………………………………………………………………………...

1.1.6 F.O.C…………………………………………………………………………

1.1.7 Safety System………………………………………………………………...

1.1.8 A.I & Reinforcement Learning………………………………………………

1.1.9 Porsche Taycan Turbo S Specs………………………………………………

1.2 Implementation………………………………………………………………………...

1.2.1 Implementation Three – Phase Voltage Source……………………………...

1.2.2 Implementation AC-DC Rectifier……………………………………………

1.2.3 Implementation DC-DC Boost……………………………………………….

1.2.5 Implementation 3L Invertor………………………………………………….

1.2.4 Implementation Battery………………………………………………………

1.2.6 Implementation F.O.C………………………………………………………..

1.2.7 Implementation A.I & Reinforcement Learning……………………………..

1.2.8 Implementation Safety Systems……………………………………………...

1.2.9 Implementation Git…………………………………………………………..

1.2.10 Implementation GitHub Actions……………………………………………

1.3 Experimental Results…………………………………………………………………..

1.3.1 Results for Voltage Current and Power……………………………………...

1.3.2 Results for signals……………………………………………………………

1.3.3 Results for CI/CD Implementation…………………………………………..

1.3.4 Results for GitHub Implementation………………………………………….

2 Planificarea activității…………………………………………………………………………….

3 Stadiul actual……………………………………………………………………………………..

4 Fundamentare teoretică…………………………………………………………………………..

4.1 Sursa de alimentare…………………………………………………………………….

4.1.1 Trifazic……………………………………………………………………….

4.1.2 Monofazic……………………………………………………………………

4.2 AC – DC Convertor……………………………………………………………………

4.3 DC-DC Convertor……………………………………………………………………...

4.3.1 DC – DC Boost………………………………………………………………

4.4 Baterie………………………………………………………………………………….

4.4.1 BMS………………………………………………………………………….

4.4.2 Litiu-Ion……………………………………………………………………...

4.5 Invertor…………………………………………………………………………………

4.6 CI/CD…………………………………………………………………………………..

4.7 GIT……………………………………………………………………………………..

4.6 F.O.C…………………………………………………………………………………...

4.7 Sisteme de siguranță……………………………………………………………………

4.8 A.I & Reinforcement Learning………………………………………………………...

5 Implementarea soluției adaptate………………………………………………………………….

5.1 Implementare Trifazic………………………………………………………………….

5.2 Implementare AC-DC Convertor………………………………………………………

5.3 Implementare DC-DC Boost…………………………………………………………...

5.4 Implementare Baterie…………………………………………………………………..

5.5 Implementare Invertor 3L……………………………………………………………...

5.6 Implementare F.O.C……………………………………………………………………

5.7 Implementare Reinforcement Learning………………………………………………..

5.8 Implementare Sisteme de siguranță……………………………………………………

5.9 Implementare Git………………………………………………………………………

5.10 Implementare GitHub Actions………………………………………………………..

6 Rezultate experimentale………………………………………………………………………….

6.1 Rezultate Tensiune Curenți si Putere..............................................................................

6.2 Rezultate Semnale...........................................................................................................

6.3 Rezultate Implementare CI/CD………………………………………………………..

7 Concluzii…………………………………………………………………………………………

8 Bibliografie………………………………………………………………………………………

9 Anexe…………………………………………………………………………………………….

10 CV………………………………………………………………………………………………

**Listă de figuri**

Figura 1. Primul Model Porsche 356

Figura 2. Porsche 911 Le Mans

Figura 3. Porsche Taycan Turbo S

Figura 4. Arhitectura proiect

Figura 5. Semnale sinusoidale Tri-Fazic

Figura 6. Convertor AC-DC

Figura 7. Convertor Boost

Figura 8. Convertor Boost circuit închis

Figura 9. Convertor Boost circuit deschis

Figura 10. Baterie Litiu-Ion

Figura 11. Invertorul 3L

Figura 12. CI/CD

Figura 13. Stator si Rotor in componenta unui motor electric

Figura 14. Curenții A B C

Figura 15. Vectorii A B C pe graf

Figura 16. Lungimea și direcția de deplasare a vectorilor A B C

Figura 17. Vectorii Alfa și Beta pe graf

Figura 18. Transformata Clark directă și inversă

Figura 19. Vectorii dq pe graf

Figura 20. Transformata Park directă și inversă

Figura 21. Reinforcement Learning

Figura 22. Circuitul proiectului

Figura 23. Solver

Figura 24. Generare Cod

Figura 25. Implementare sursă Tri-Fazic

Figura 26. Implementare AC-DC Convertor Full-Wave

Figura 27. Calculul valorii medii

Figura 28. Implementare D.C-D.C Boost

Figura 29. Implementare baterie

Figura 30. Implementare invertor 3L

Figura 31. Implementare F.O.C

Figura 32. Implementare Transformata Clark

Figura 33. Implementare transformata Park

Figura 34. Implementare Transformata inversă Park

Figura 35. Implementare Reinforcement Learning

Figura 36. Implementare Reinforcement Learning Submodul

Figura 37. Implementare modul Erori

Figura 38. Implementare modul clasificare Erori

Figura 39. Git add .

Figura 40. Git commit -m „Mesaj”

Figura 41. Git push

Figura 42. Cod C.I/C.D

Figura 43. Valoarea medie de la ieșirea convertorului A.C-D.C

Figura 44. Tensiunea de la ieșirea D.C-D.C Boost

Figura 45. Tensiune, Curent, S.O.C și Putere pentru baterie

Figura 46. Culori pentru timpi

Figura 47. Culori pentru timp circuit mare

Figura 48. Culori pentru timp circuit mic

Figura 49. Semnal Trifazat

Figura 50. Semnal după conversie A.C-D.C

Figura 51. Semnal după conversie D.C-D.C Boost, la aproximativ 800V

Figura 52. Semnalul ieșire inversor 3L cu rotații la 3000

Figura 53. Semnalul ieșire inversor 3L cu rotații la 6000

Figura 54. Viteza de feedback, cosinus și sinus

Figura 55. Curenții de pe stator

Figura 56. Semnalele Alfa și Beta

Figura 57. Semnalele d și q

Figura 58. Semnalele ds și qs

Figura 59. Transformata inversă Park

Figura 60. Transformata inversă Clark

Figura 61. Semnale pentru Invertor de control

Figura 62. Rulare terminată pentru compilare proiect principal

Figura 63. Terminal Agent

**Listă de abrevieri**

A.I: Artificial Intelligence.

M.L: Machine Learning.

FOC: Field Oriented Control.

PMSM: Permanent Magnet Synchronous Machines.

PSM: PWM methods of electrical machines.

PWM: Pulse with Modulation.

EV: Electrical Vehicles.

SVM: Space Vector Modulation.

SOC: State of Charge.

SOH: State of Health.

SOP: State of Power.

BMS: Battery Management System.

P: Proportional.

I: Integral.

D: Derivative.

PI: Proportional - Integral.

PD: Proportional - Derivative Controller.

PID: Proportional - Integral - Derivative Controller.

NPC: Neutral Point Clamped.

MOSFET: Metal - Oxid - Semiconductor Tranzistor.

IGBT: Tranzistor Bipolar cu Poarta Izolata.

SVM: Space Vector Modulation.

ACIM: AC Induction Machine

CI/CD: Continua Integrare Continua Dezvoltare

SMPS: Switch Mode Power Supply

R: Rezistenta

L: Bobina

C: Capacitor

RLC: Rezistenta – Bobina - Capacitor

DC: Curent Direct

AC: Curent Alternativ

P: Putere

NEDC: New European Driving Cycle

WLTP: Worldwide Harmonized Light Vehicles Test Procedure

D: Direct

Q: Quadrature

MPC: Model Predictive Control

Rload: Rezistanta load

SVPWM: Space Vector Pulse With Modulation

T.V.S: Transient Voltage Suppression)

**1.Rezumat în limba Engleză**

The idea behind this project was to create a fully functional and working system related to EV’s and to use A.I and M.L to improve the concepts that are very well known in the automotive industry. For that it was necessary to construct a system that takes the energy from the power grid, we selected the three-phase system, to simulate the real time application, then to convert the AC into DC signal, using a rectifier. The values are chosen for a specific car, I choose a very beautiful hyper electrical car, Porsche Taycan Turbo S that has an 800V battery packet, so we needed an DC-DC Boost to be able to achieve that value and to be able to store this value into the battery system. From these we went into the Current control system to be able to control the 3L inverter. In all these aspects, we can use ML and AI to improve some of the parts of the control look for the current and to improve the DC-DC system functionality.

**1.1 Theoretical Fundamentals**

**1.1.1 Sources**

For the project we need to implement a voltage source. To be as close as possible to reality I choose to use a Three-Phase Voltage Source, because this is how most of the people that uses EV’s are charging their cars. But I also wanted to cover a few aspects of the Mono-Phase Voltage Source, because this is another solution that can be used to charge the electrical vehicle.

* + - 1. **Three-Phase Voltage Source**

A Three- Phase Voltage source uses three live or hot wires, one neutral and one ground wire and the Three – Phase terminology comes from three distinct currents, one for each live wire from the circuit.

The main benefit of the Three – Phase voltage source it is how it is managing power, instead of simply multiplying the voltage with amps, the power equation looks more like this:

(1)

With this equation, the three phase voltage source can deliver 1.73 more power than a single-phase system, making this system more efficient for charging, especially for charging a high-power electrical vehicle.

* + - 1. **Mono-Phase Voltage Source**

To be able to understand the difference between Three – Phase and Mono –Phase Voltage source, we need to explain also how the Mono-Source Voltage is working. This Mono-Phase Voltage source only works with one or two wires, instead of three live wires that the three phase it is providing to us. That means we will get less power, and the final equation will look like this:

(2)

**1.1.2 AC – DC Rectifier**

A rectifier or an AD-DC Convertor is an electrical device that is used to convert AC current into DC current. The AC current periodically reverses directions and the DC current it flows in one direction. These electrical circuits have many uses, usually into high power or power electronics schemes and they have a big and key role in charging the EVs.

There are many types of Rectifiers, we have Single Phase Rectifier’s: Half-wave rectifiers which are used for a single-phase power supply, where either the positive or the negative signal of the power source is going to pass, and the other half is going to be blocked. Because of that the mean voltage it’s going to be lower. Another example will be Full-Wave rectifier which converts the whole input from the power source into DC, so the output voltage will be higher.

Then we have Three-Phase rectifiers, the Single-Phase rectifiers are more used for low power systems, for the high-power systems we are going to use Three-Phase Rectifiers, this will also be used for our project. We are using a Three-Phase full Wave Controlled Rectifier, which consists of three single phase full wave-controlled rectifiers connected in such a way that all the three phase voltages are used to create the DC output power.

* + 1. **DC-DC Convertor**

To move forward with the project, the next step it is to have a DC-DC Converter that can generate the desired voltage that we need. For our selected car, which is Porche Taycan Turbo S has an 800 V battery pack declared, so the development will be in that way that we need to achieve that voltage value and to achieve the desired Power which is 560kW. The company declared that the voltage pack it is around 630-825V, based on the configuration system, so that will also our range, with a power between 400-560kW. There are two types of Converters that can be used, at first, we will go with the DC-DC Boost implementation.

**1.1.3.1 DC – DC Boost**

The Boost Converter or the step-up converter it is used to increase the value of voltage while decreasing the current, in analogy to this we have the Buck converter that is used to lower the voltage value. The Boost converter it is an electrical circuit that comes from the group of switch mode power supply (SMPS), this electrical circuit contains a diode, a switch, an RL or just an R system, one resistor and one capacitor, of course that the circuit can be improved or remodeled by desire. The role of the capacitor at the end of the system is to reduce the voltage ripple. The Voltage Ripple it is the residual periodic variation of the DC Voltage within the power supply. The principle of the Boost Converter is the tendency of an inductor to resist changes in current by either increase or decrease the energy stored in the inductors magnetic field. So, we have two possible ways, one when the switch is closed so the current it’s going to flow into a clockwise direction and some energy will be stored in the magnetic field of the inductor, results that the left part of the inductor will be positive, and the second way it’s when the switch is open, and the current will go through the diode and will face the capacitor and the R load. For the second way the magnetic field energy that was stored in the previous method will be decreased and will be used to maintain the current throw the inductor, as a result, the left part of the inductor will be negative and the current from the voltage source and from the inductor witch it’s in series with the voltage source will be added together and will be redirect to the forward-biased Diode and to the load.

* + 1. **Battery**

Next, we need to discuss about the Batteries, the most used battery types in the automotive industry are Lithium-Ion. The Porche Taycan comes in a variety of packages, from the Performance Battery package up to the Performance Battery plus. The Porche Taycan Turbo S contains two deck Performance Battery Plus, with thirty-three cell modules consisting of 12 individual cells each, 396 in total, with a total capacity of 93.4kWh. It is the first production car with a system voltage of eight hundred volts (voltage range 610 to 835 volts) instead of the usual 400 voltage range for electric cars. Taycan Turbo S: electricity consumption combined 26.0 kWh/100 km (NEDC); 23.4 – 21.9 kWh/100 km (WLTP); Combined CO2 emissions 0 g/km.

**1.1.4.1 BMS**

The Battery Management System of the any electronic circuit that uses a battery or a battery pack it’s a very important part of the system, this is used to monitor key values of the battery, for example the voltage and current of the battery, the S.O.C ( State of charge), the S.O.H ( State of Health) and the S.O.P (State of Power). Another important aspect of the Battery Management System is temperature monitoring, which keeps the battery pack out of any danger, we already saw enough videos and events when electric cars has catch on fire because of the poor Battery Management System that was implemented, this needs to be changed.

**1.1.4.2 Lithium-Ion**

The Lithium-Ion Battery or Li-Ion Battery it is a rechargeable battery that uses the reversible intercalation of Li ions into electronically conducting solids to store energy. These are the most used and the best types of energy storage on the market at this point, they are heavily used in the automotive industry.

* + 1. **Invertor**

The Three-Level Inverter (3L), that can be also named Neutral-Point Clamped Inverter (NPC), it generates three output voltages. The electrical system it’s made from switches (IGBT’s or MOSFET’s) and diodes, that can will be used to create the Positive Half Cycle were S1 and S2 are ON, the button S3 and S4 are off. Then the Zero level (0) were the mid switches S2 and S3 are ON, the top and button ones are off, and the load is connected to the mid-point. And the last configuration it’s Negative Half-Cycle, the button two switches S3 and S4 are ON, the top ones S1 and S2 are off.

**1.1.6 CI/CD**

CI/CD or the Continuous Integration and Continuous Development it is a method of development that helps the developers to be able to move faster in the progress. It is used to run the jobs for the software engineer, which accelerates the development of the project. This is working by creating a pipeline that runs, simulates, and saves the results of a compiled project by connecting this with GitHub and using Git, this process accelerates the development. There are two big competitors, one it is Jenkins, which is more used in big teams and productions and the other one is GitHub Actions, new on the market, much simple and easy to use, this is supposed to replace Jenkins soon.

**1.1.7 GIT**

Git it is a distributed version control system that allows the users to keep a track on their work. With some amazingly simple commands we can store, evaluate, and keep track of our work. The projects can be stored on GitHub, in a repository where they can be used and seen by anyone, or they can be set as private.

**1.1.6 F.O.C**

To understand the Field Oriented Control method, we need to understand Clark and Park transformations which are the keys of this method. At first, we need to have an A.C Current Focus, that means we need to work with A.C current, that will lead us to the Three Phase Current control and to the Three Phase machines. In these machines we have three coils, mounted fixed on the stator with a shift of 120 degree between them. On the rotor we have a permanent magnet, this setup will generate three current I1, I2 and I3 and will generate the rotating magnetic field.

If we are projecting these currents on a graft, we will be able to find the rotation of the total magnetic field by using the right-hand rule. To understand why this method is so important we need to understand why we need it, at first, we will see the difference between D.C machines and A.C machines. In a D.C machine we need to calculate one D.C Output Voltage and to control one D.C Current in order to provide us the desired Torque, in an A.C machine we need to calculate Three A.C output voltages, to control Three AC Currents according to rotor position in order to find the desired Torque, for that it was necessary to come up with a simplifier solution.

Then we need to move the three-phase coordinate system, we will call this one, abc- coordinate system, must be moved into alpha and beta coordinate system, this will be called Clark Transformation, and then moved into dq coordinate system, this will be called Park Transformation. After that we need to control d and q currents with an PI controller or another method, for this project we used Reinforcement Learning that will help us to gain the desired control. Then we have the Inverse Clark and Inverse Park transformation, this means we will go back with the transformations to generate the PWM signals that will be used to control the 3L Inverter.

**1.1.7 Safety System**

This will be used to protect the electrical circuit from any potential issues that can cause extreme damage to the system. These issues can come from different areas, they can be from a way to high temperature or can come from a damage component.

**1.1.8 A.I & Reinforcement Learning**

. Reinforcement Learning it’s a type of learning witch it’s focused on making decisions focused on reward in each situation. Reinforcement Learning involves learning through experience, an agent learns to achieve results by performing different actions and receiving feedback. The Reinforcement Learning use the principle of error and trail, the agent takes actions withing the environment, receives rewards or penalties and adjust its behavior to maximize the cumulative reward.

**1.1.9 Porsche Taycan Turbo S Specs**

The Porche Taycan it is the most powerful EV car from Porche, it is a luxury sedan that has multiple packages, power packages and battery packages. The power range it is between 240kw/300kW (The normal driving mode and sport mode) and 460kW/560kW for the Taycan Turbo S. The one that will use for this project.

**1.2 Implementation**

**1.2.1 Implementation Three – Phase Voltage Source**

For the implementation of Three – Phase Voltage Source will use the block that Simulink can give us, will be set to 400V and 50 Hz, this is the standard Three Phase system in Romania.

**1.2.2 Implementation AC-DC Rectifier**

The implementation of the AC-DC Rectifier we must follow the theoretical part, where we can see the calculations and how it’s functioning. We modelate the system to receive three inputs signals from the Three Phase Voltage Source, we have diodes with a low internal resistance and a Forward Voltage of 0.8V. Snubber resistance of 500Ohms and Snubber capacitor of 250e-9. Output resistance will be 1 Ohm.

**1.2.3 Implementation DC-DC Boost**

For the implementation of DC-DC Boost I used the information’s from the theoretical part, as a bonus for a better working I added an MPC component. The MPC component it’s a Model Prediction Control which controls the current from inductance to regulate the output voltage. The MPC component receives feedback from the output voltage of the system and the input voltage that is showing the system in real time. The primary role of the MPC component it’s to keep the output voltage to a specific reference Voltage. It calculates the optimal control actions by minimizing a cost function that includes the error between the desired and actual outputs. Based on its prediction and optimization, MPC Boost controls the inputs of the DC-DC Boost, this includes manipulating quantities such as the duty cycle of the switching element in the converter, witch directly impacts the output voltage. The output from the MPC is indicated by S signal (the control signal) witch it’s used to regulate the boost converter’s operation effectively. The model predictive control approach that allows the system to adapt to dynamic changes, such as varying load conditions or input voltage fluctuations.

**1.2.4 Implementation Battery**

For the Battery implementation I used a simple block from Simulink library, that block replicates a Lithium-Ion Battery that it’s very used in the automotive industry. This block will store the voltage given by the D.C-D.C Boost and will output the Battery Voltage, Power, Current and S.O.C that will give us how much the battery has charged.

**1.2.5 Implementation 3L Invertor**

The 3L invertor is a type of power electronics converter that turns DC power into AC power. It makes this achievement by using three levels of voltage, negative, positive and the zero one. In the model that was implemented we are using six IGBT transistors which are two by two connected to one of the three lines, this is how we can generate these voltages signals. These transistors are controlled by a PWM signal that is generated using the F.O.C control algorithm. The output of these transistors is then filtered to be sure we have a smooth signal.

**1.2.6 Implementation F.O.C**

To implement the F.O.C (Field Oriented Control) algorithm we need to follow very carefully every step in the theoretical part. We will use the stator currents A, B and C and the Rotor speed and angle to implement the Clarke and Park transformation. The first one will be the Clark transformation that takes our three-phase system into a two-phase system, then we move the signals into the Park transformation, we are resulting the D and Q signal that will be controlled into the Reinforcement Learning. The output of the Reinforcement Learning are the actions, the Ds and Qs actions, that are used in the next step to create the PWM signals which are necessary for the 3L inverter to work. More about the Reinforcement Learning are presented in the next section.

**1.2.7 Implementation A.I & Reinforcement Learning**

This is main object in the F.O.C implementations for this project. I used Twin Delayed Deep Deterministic Policy Gradient, this type of Machine Learning uses an Actor-Critic topology. Where the Actor it’s that part of the Agent that takes decisions based on the current observations and the Critic evaluates the actions taken by the Actor, based on rewards. The Agent (created by the Actor and the Critic) starts with random actions, to explore the place and to gain experience. In the meantime, the data it’s store in a buffer to be able to be used later for the training process. TD3 uses two Critic networks for finding the Q value (Expected Return based on his actions). TD3 also uses target networks, which are slower versions of the actor and critic network, which are used for the stability and to find the Q value without using the big architecture. For the code implementation, we start by defining the key values, the PWM frequency, sampling times and data type. Then we will need to define the number of Observations and the number of Actors and then we need to define the environment. Next step it’s to create the Critic and Actor network, and in here we need to define three paths, the first one it’s State Path witch it’s responsible for the observations processing, then we have the Action path which process the similar actions and then we have the Common Path that combine these two paths to find the Q values. Then we will need to define the Actor network which uses a tanh function that keeps the values in a certain range. The next step it’s to define the TD3 agent which needs the next values, the Discount Factor that represents how much the Agent is going to care about the rewards, 1 means the most and 0 the lest, Experience Buffer it’s a long term memory that it’s used to store the previous data, Mini Batch size that’s the number of experiences, Smoothing factor that controls how the environment it’s updating. In the last part we define how the agent it’s going to train.

**1.2.8 Implementation Safety Systems**

The Safety System it’s very important to keep the system in the normal limits of functionality. In here I took all the signals from the system, and I verify each one of them if they work in normal way between a maximum and a minimum value. If any signal will show a drop in the value or a sudden increase, this one will trigger an error, then based on the error type they are filtered between High Severity Error, Medium Severity Error and Low Severity Error. For the High Severity Errors, I created a system that will shut down the system so the car will not be damaged.

**1.2.9 Implementation Git**

We need to install the Git setup from the Git Web page, then we will use standard Git commands to add the documents, to commit them and to push them into the remote branch, where it will be stored, updated and can be taken at any time we want. We can use different branches for different implementations and then to merge them with the main branch.

**1.2.10 Implementation GitHub Actions**

GitHub Actions it’s implemented on GitHub and requires an agent to run the build. This is used to compile and simulate the entire toolchain, and it’s using his own programming language. This code it’s write in a yaml file.

**1.3 Experimental Results**

**1.3.1 Results for Voltage Current and Power**

The first value will be the output voltage from the A.C-D.C converter, that should be 1.73 times bigger than the input voltage from the Tri-Phase Voltage Source, so the output voltage it's now 537.5V. We can see that in the image 33. Then after the DC-DC Converter we need to have an approximative 800 voltage, we manage to get 810V as the output of the D.C-D.C Boost converter, that also uses an M.P.C controller, and the Power it's 559.4kWz with it's very close to the 560kWz declared by the manufacturer. In the end our Lithium-Ion Battery can store up to 809V and the S.O.C it's 98.3 precent.

**1.3.2 Results for signals**

The signals can be seen on the images 36 to 47. The first image represents the Tri-Phase signals from our voltage source, this source was set up to 400V and 50 Hz as it is presented in Romania. Next, we have the A.C-D.C convertor. Next, we will have the D.C-D.C Boost signal that was created with the help of the M.P.C Boost component. There we can see a D.C signal that has a value of approximative 810 V, this is in the range that was presented by the manufacturer. Then we will see the signals after the 3L inverter was activated, these are A.C like signals, for a mechanical speed of 3000 rad/s we will have an 200V signals. Then we can increase the mechanical speed, and the signals will increase in amplitude. Then we need to see the cosine and sine signals that were made from the angle of the rotor, the Speed Feed Back that will be given to us by the speed of the rotor. Next, we need to see the signals from the stator, the three a, b and c signals, that will be used in the F.O.C control. After that we will have the first transformation, the Clark Transformation and we will have the alpha and beta signal, then the Park transformation that will result the dq signals. Next point it's the Reinforcement Learning block that will output the dq signals, then we are going backward using inverse Park and inverse Clarke transformation, the final signal will be send to the 3L to control the transistors.

**1.3.3 Results for CI/CD Implementation**

For the CI/CD Implementation we used GitHub Actions as the toolchain that does the job, he is used to run the entire project and to check is everything it’s fine, if the result is not good or the project is not compiling, I will receive an email with details about the project.

**1.3.4 Results for GitHub Implementation**

Using Git and GitHub to save the Project, I used multiple branches or directly the main branch for the implementation. With some simple commands, using the bash window, I can store and see the updates, and I can revisit some old parts of the project.

**2 Planificarea activității**

|  |  |  |  |
| --- | --- | --- | --- |
| Număr curent | Activitate | Data de început | Data de sfârșit |
| 1. | Discuție cu profesorul de disertație despre posibile teme de disertație. | 10.10.2024 | 1.11.2024 |
| 2. | Încărcarea primului demo si întocmirea actelor | 1.11.2024 | 05.01.2025 |
| 3. | Încărcarea celui de al doilea demo | 05.01.2025 | 09.02.2025 |
| 4. | Pregatirea si implementarea Reinforcement Learning | 09.02.2025 | 03.03.2025 |
| 5. | Pregatire Sisteme de siguranță | 03.03.2025 | 10.03.2025 |
| 6. | Finalizare documentație | 10.03.2025 | 15.03.2025 |
| 7. | Revizuire proiect | 15.03.2025 | 20.04.2025 |

**3 Stadiul actual**

În momentul de față, putem spune că proiectul a ajuns la un punct de succes. Prin această remarcă îmi doresc să precizez că au fost valorificate principalele teme abordate. Scopul acestui proiect a fost să aducă beneficiile date de către M.L și A.I, respectiv procesare de date, care sunt tot mai răspândite în ziua de azi și vor deveni tot mai des utilizate, asupra unei mici parți care este utilizată de către marile companii auto în ziua de azi. Acest sistem replică modul de gândire și de utilizare a componentelor de electronică de putere în crearea de noi tehnologii folosite pentru dezvoltarea sistemelor de alimentare și de control în lumea mașinilor electrice.

După cum bine știm, toate aceste tehnologii care se află deja de o bună perioada de timp pe piața și în industrie, au reușit să prindă amploare în diverse domenii de activitate, precum în fizica prin dezvoltarea de software folosind tehnici de rețele neuronale pentru depistarea de celule cancerigene la diferite organe ale corpului uman sau pentru depistarea unor noi domenii în fizică. Pentru mai multe informații se recomandă cartea scrisă de către domnul profesor doctor matematician Roger Penrose – Mintea noastră cea de toate zilele. Această carte prezintă o examinare extrem de profundă a minții, a inteligenței și a limitelor de calcul. Domnul profesor începe cu ideea următoare, dacă un computer poate să fie vreodată la același nivel cu oamenii prin creativitate și prin conștiință, aducând în prim plan ideea că inteligența artificială (A.I) poate să replice gândirea umana pe deplin. Argumentul sau este că capabilitățile mintii, în special conștiința, nu pot să fie replicate în totalitate de către algoritmi matematici.

Pentru crearea acestui proiect am dorit să utilizez tehnologii care sunt tot mai des răspândite în lumea automotive, acestea fiind pachetul Matlab - Simulink care a fost studiat dea lungul anilor de licență în cadrul facultății, cat și la acest master. În principal Matlab a fost folosit pentru scrierea de cod, pentru implementarea M.L, definirea de date, iar partea de Simulink este folosită pentru crearea schemei electronice și pentru implementarea codului în modul M.B.D (Model Based Development). Acesta este modul de implementare și în companiile mari care dezvoltă software pentru domeniul automotive. Este folosit deoarece viteza de dezvoltare a unui proiect a crescut considerabil față de abordarea clasica de scriere de cod în limbajul de programare C. Astfel pachetul Matlab - Simulink conține și uneltele necesare pentru convertirea codului M.B.D în cod C sau C++.

Ideea proiectului a venit de la implementarea unei scheme deja existente pe piață, a tuturor componentelor necesare pentru încărcarea și funcționarea unui motor electric pentru mașinile electrice. Astfel am dorit să vin cat mai aproape de realitate și am început prin setarea puterii de la sursa de tip tri-fazic care exportă 400V la o frecvență de 50Hz.

Acest curent de tip AC trebuie să fie convertită în curent de tip DC, pentru acest lucru avem nevoie de un convertor de tip AC – DC sau cum este mai răspândit după denumirea de Rectifier. Aceste convertoare pot să fie de mai multe tipuri, eu am ales de tip Full-Wave Rectifier care folosește mai multe diode pentru a converti întreg semnalul AC în semnal de ieșire DC. De precizat că diodele sunt componente semiconductoare care permit trecerea curentul electric într-o singură direcție. Mai multe detalii despre funcționare o să fie prezentate în paginile ce urmează.

Următorul pas este să folosim un DC-DC Boost pentru a putea obține valoare dorită a tensiunii. În cazul de față s-a dorit ca sistemul să fie realizat pentru o mașina de putere foarte mare și anume Porsche Taycan Turbo S. Acesta este versiune de top a celor de la grupul Porsche, o mașina full electrică care întrece orice rival în funcție de performanțe. Totuși, toate acestea vin cu un cost și anume o putere de aproximativ 560kW ( în sarcină maximă ) și aproximativ 800V pentru alimentarea. Pentru acest lucru DC-DC Boostul nostru își face treaba într-o foarte bună măsură. Pentru această parte am dorit implementarea unei funcții pe baza M.L care să controleze comutatorul și să ajute sistemul la valoarea tensiunii dorite.

Mai departe este nevoie de un sistem de stocare a energiei, Porsche folosește un număr impresionant de celule Litiu-Ion, aproximativ 396 de celule individuale, distribuite atât în paralel cat și în serie. Aceste celule pot să fie direct importate din librăria pusă la dispoziție de către Simulink. Pentru o compilare mai rapidă se vor folosi un număr mai mic de celule, dar o reprezentare vizuala a celulelor poate să fie realizată. Pentru o funcționare bună a acestora și în siguranță, este necesară implementarea unui sistem de management al acestora. Principalele puncte ale acestui sistem sunt de a ține în siguranță sistemul de baterii, de a monitoriza parametrii cheie ( tensiune, curent, S.O.C), de a ține în balans valoarea tensiunii pe de baterii, deoarece nu ne dorim să avem tensiune mai mare pe o baterie față de altă baterie, calcularea S.O.C, S.O.H și S.O.P, comunicarea cu alte componente din sistem și protecția bateriei în partea de încărcare și în partea de descărcare.

Mai departe sistemul încarcă bucla noastră închisă de control care folosești metode de F.O.C (Field-Oriented Control) și metode avansate de control folosind PWM ( Puls cu modulație).

Controlul scalar, precum strategia de tip „V/Hz” scade mult în performanțe, deoarece avem oscilații ale Cuplului în motoarele pe inducție. Pentru a avea performanțe optime, o metodă superioară de control precum F.O.C este necesară.

F.O.C sau după ceea de a doua denumire, Vector Control, este o tehnică folosita pentru controlul P.M.S.M ( Permanent Magnet Synchronous Machine) și A.C.I.M ( A.C induction motors). F.O.C ne oferă o bună controlare asupra cuplului și a vitezei. Implementarea necesită folosirea de algoritmi matematici care transformă curenții de pe stator din cadrul de referință staționar într-un cadru de referință sincronizat cu fluxul rotorului, care mai are denumirea de cadru de referință d-q. Cu această implementare se obține un control mai bun al fluxului magnetic și al cuplului motorului, ceea ce duce la o mai bună funcționare a motorului și la rezultate mai optime. Într-un scurt rezumat putem spune că F.O.C permite o reglare mai ușoară a cuplului și a vitezei motorului ceea ce duce la performanțe mai bune și la un control mai eficient. Mai multe informații despre modul de implementare și despre partea teoretică, o să fie prezentate în capitolele ce urmează.

Pentru versionarea proiectului și pentru menținerea lui, am folosit Git precum și GitHub, acesta are rolul de a menține un istoric al proiectului și de a ține fiecare versiune salvata pe server. Rolul său este foarte important, fiind utilizat în toate companiile de dezvoltare de software. Proiectul o să fie pus pe privat din motive de siguranță pe aceasta platformă.

O altă unealtă foarte puternică care ajută la dezvoltarea mai rapidă a software lui și care începe să fie tot mai utilizată în lumea automotive este Jenkins sau noul său competitor GitHub Actions. Acestea au rolul de a grăbi dezvoltarea prin crearea așa numitor run-uri sau pipeline-uri care au rolul de a compila și executa comenzi care pregătesc și testează software-ul sau hardware-ul pentru utilizator. Implementarea unei metode de CI/CD (Continuă Integrare Continuă Dezvoltare) este o unealtă foarte importantă și puternică pentru dezvoltarea de proiecte. În cazul de față cele două metode menționate mai sus, sunt ambele optime și bune de implementat pentru proiectul nostru, dar Jenkins este deseori folosit în producție, în momentul în care avem o echipa mare de dezvoltatori iar întreg procesul de compilare și de setare al mediului de rulare al software-ului devine destul de mare, așa că, pentru proiectul nostru este mai bine să alegem GitHub Actions, deoarece acesta necesită doar contul de GitHub asociat proiectului, se implementează și se menține mai ușor decât Jenkins, este de mai actualitate și începe să acopere o mai mare parte din piață.

Ca și viitoare dezvoltări, proiectul de față este conceput ca o baza din care alte proiecte de o amploare mai mare pot să fie dezvoltate, nenumărate modele pot să fie adăugate în acest sistem pentru al face mai bun pe toate planurile. De exemplu putem discuta despre înlocuirea DC-DC Boost-ului cu un DC-DC Bi-direcțional boost care să permită și controlul curentului pentru descărcarea și încărcarea bateriilor. Înlocuirea sistemei actual de baterii cu un sistem mai complex și mai aproape de realitate, dar acest lucru are nevoie si de o putere de compilare mai mare. Introducerea de noi metode de control al curenților sau de optimizarea metodei deja implementate. Optimizarea sau refacerea modelelor de A.I și M.L care ajută la optimizarea sistemului nostru. Implementarea unui sistem de management al bateriilor mult mai complex, care să acopere toate ariile de funcționare și de siguranță, precum și implementarea unui sistem mai complex de siguranță, care să acopere o gama mult mai largă de posibile erori.

Unele erori care au aparat pe parcurs sunt cele de conectare, în Matlab sunt mai multe tipuri de conectori, avem cel clasic din librăria Simulink care este virtual dar avem și un alt tip de conectorii de tip real, acesta se afla în librăria Simscape, mai multe componente din librăria Simscape au fost utilizate în acest proiect, deci un punct important și un topic necesar este acela de a înțelege cum cele două tipuri de conectori pot să fie folosite în același proiect. Alte probleme sunt cele de compilare și de rutare, Matlab și Simulink sunt foarte sensibile la tot ce ține de compilare și rutare a proiectelor, de exemplu, tot ce ține de proiect trebuie să fie într-o cale bine stabilită și nicidecum să fie schimbată. Versiunea de Matlab folosită este R2022.b pentru a folosi o altă versiune este necesară adaptarea proiectului la versiunea dorită, acest lucru poate produce diferite erori.

Daca se dorește rularea proiectului, este necesar sa avem aceasta versiune de Matlab împreună cu pachetul Simulink instalată și cu licența. Avem nevoie de git sa fie instalat, precum și de formarea unui agent pentru rularea C.I/C.D, sau de folosirea resurselor date de către Github, lucru ce trebuie să fie modificat în fișierul din componenta C.I/C.D.

**4 Fundamentare teoretică**

În capitolul de Fundamentare Teoretică, o să trecem în scurt prin toată materia de bază, care trebuie înțeleasă pentru ca proiectul să poată să fie dus la final. Unele aspecte o să fie explicate în mare cu accent doar pe ceea ce avem nevoie pentru dezvoltarea proiectul. Pentru mai multe informații se recomandă să se acceseze sursele din capitolul Bibliografie.

Înainte de a intra în topic uri legate strict de funcționarea componentelor care o să fie utilizare în acest proiect, îmi doresc să aloc câteva rânduri pentru a prezenta compania Porsche și pentru a face cunoscuta istoria lor, precum și cum au ajuns la modelul Taycan.

Istoria companiei Porsche a început în perioada celui de al 20 secol de către Ferdinand Porsche, un pioner in lumea automotive, în anul 1931 a fost fondat D. Ing. h. c. F. Porsche GmbH în Stuttgard, Germania. Inițial compania oferea servicii de consultanță pentru developarea mașinilor și nu produceau ei mașini sub numele lor, unul dintre misiunile lor date de către cei de la Volkwagen a fost să facă designul pentru celebrul Volkwagen Beetle, care a devenit unul dintre cele mai celebre mașini ale tuturor timpurilor.

În anul 1948, Porsche introduce prima lor mașină, și anume Porsche 356, așa cum se poate vedea în imaginea de mai jos. Acest model a fost proiectat de către Ferdinand Ferry Porsche, fiul lui Ferdinand Porsche, și a fost creat în Austria. Modelul 356 a fost ușor și impresionant pe drum, a fost o mașina sport, elegantă, care a setat standardele pentru compania Porsche folosind un motor termic și o structură aerodinamică. Și-a câștigat un nume în lumea automotive și a sporturilor cu mașini prin performanța și ingeniozitatea de care au dat dovadă inginerii.



Figura 1. Primul Model Porsche 356

De-a lungul timpului, compania Porsche și-a perfecționat și a explorat prin crearea de noi modele. Modelul foarte apreciat precum Porsche 911, introdus prima dată în anul 1964, a devenit foarte cunoscut. Acesta are un design unic în lume și cu o dinamică de neegalat, a rămas un emblem pentru lumea sportului după câștigarea cursei de 24 de ore de la Le Mans.



Figura 2. Porsche 911 Le Mans

Catalogul celor de la Porsche a crescut prin adăugarea de noi modele pentru toate categoriile de cumpărători. Avem modele mai mari precum Porsche Cayenne sau SUV de statura mai mică precum Macan, sau sedan din gama de lux precum Porsche Panamera care a fost introdus în anul 2009.

Premiile din lumea sportului au început să apară și să crească numele companiei, cele mai semnificative fiind câștigarea cursei de 24 de ore de la Le Mans și câștigarea cursei de Anduranță.

Introducerea modului Taycan. În secolul 21, compania Porsche începe să facă trecerea de la producerea de mașini termice spre cele electrice. Introducerea modelului Taycan complet electric face startul unei noi ere pentru companie. Original a fost introdus ca fiind un concept cu numele Misiune E în anul 2015, modelul Taycan își face apariția în anul 2019, combinând istoria Porsche cu tehnologie electrică de ultimă oră.



Figura 3. Porsche Taycan Turbo S

Taycan a fost introdus împreună cu modelul de top Taycan Turbo S, acesta a fost răspunsul dat de către compania Porsche către cererea tot mai mare de mașini electrice de performanță mare. Modelul oferă performanță de ultimă oră, lux și o durată mare de viată, toate acestea fără a scoate gaze nocive pentru mediul înconjurător. Performanțele acestui model sunt bine cunoscute, acesta are două motoare electrice care acționează sistemul all-wheel drive și poate produce pană la 750 de cai putere folosind overboost și launch control. Acest lucru îl face capabil să accelereze de la 0 la 100 de km pe ora în numai 2.6 secunde. Viteza maximă a acestui model este de 161 de mile pe oră, ceea ce este foarte impresionant pentru o mașină electrică. Arhitectura avansată de 800 de volți ajută la încărcarea rapidă a modelului, reducând foarte mult timpul de așteptare. Acest model este echipat cu un sistem adaptiv al suspensiei pe aer și virarea roților din spate, asigurând un standard foarte înalt pentru controlul mașinilor pe drum din această gamă. Interior acestei mașini arată atenția la detalii a companiei, având un aspect extrem de luxos și de elegant, dar care este combinat cu sportivitate.

Porsche își stabilește obiective ambițioase pentru sustenabilitate și electrificare, ca parte a strategiei sale pe termen lung. Până în 2030, Porsche își propune ca peste 80% dintre vehiculele sale noi să fie electrificate. Taycan este doar începutul călătoriei electrificate a Porsche, cu modele viitoare care vor extinde și rafina ofertele electrice ale brandului.

În concluzie, Porsche Taycan Turbo S este un vehicul semnificativ care evidențiază capacitatea Porsche de a inova și de a conduce în segmentul mașinilor sport de lux. Acesta face legătura între istoria renumită a unui brand cunoscut pentru excelența sa inginerească și cerințele viitorului pentru performanță auto sustenabilă și captivantă. Prin angajamentul său față de inovație, Porsche rămâne o forță de pionierat în lumea auto, modelând viitorul vehiculelor de înaltă performanță, eco-friendly. În continuare o să trecem pe prezentarea parții tehnice, în imaginea de mai jos se poate observa arhitectura proiectului.

A diagram of a computer

AI-generated content may be incorrect.

Figura 4. Arhitectura proiect

**4.1 Sursa de alimentare**

În cadrul proiectului se poate alege o sursă de alimentare Mono-Fazic sau Tri-Fazic, diferențele dintre cele două o să fie explicate în capitolele ce urmează, se va pune accent pe Tri-Fazic, deoarece aceasta este implementată în proiect dar și Mono-Fazic este o variantă bună.

**4.1.1 Trifazic**

**A diagram of a graph

Description automatically generated**

Figura 5. Semnale sinusoidale Tri-Fazic

Sursa de alimentare aleasă este Tri-Fazic, des utilizată de către posesorii de mașini electrice pentru încărcarea acestora. Această sursa de alimentare poate distribuii mai multă putere, aproximativ 1.73 mai mult decât o simplă sursă de alimentare mono-fazica. Așa cum ne spune și numele, Tri-Fazic, înseamnă că avem 3 faze (3 surse de alimentare A.C), semnale sub formă sinusoidală prin care circula curent alternativ, acestea fiind defazate unul față de celălalt. Pe lângă acestea, mai avem un fir de împământare și un fir neutru. Ecuația care caracterizează acest tip de sursă de alimentare este următoarea:

(3)

**4.1.2 Monofazic**

Sursa de alimentare Mono-Fazic, spre deosebire de cea Tri-Fazic, are rezultate mai slabe și nu se recomandă pentru încărcarea mașinilor electrice, în special pentru cele de putere mare. După cum ne sugerează denumirea, Mono-Fazic, vom avea o singură fază, o singură sursă A.C care o să ne dea o putere considerabil mai mică. Ecuația matematică este descrisă astfel:

(4)

**4.2 AC – DC Convertor**

Pentru a putea încărca bateria mașinii electrice, este nevoie să trecem semnalul nostru din curent alternativ în curent direct, din A.C in D.C. Pentru a putea realiza acest lucru avem nevoie de un Rectifier sau cum mai este numit, convertor A.C-D.C.

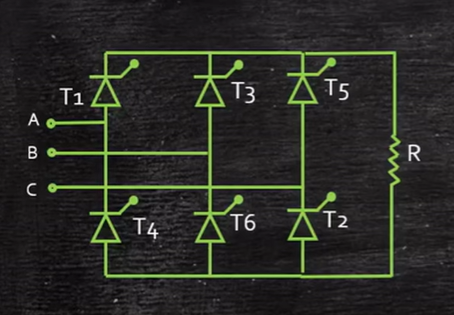


Figura 6. Convertor AC-DC

Pentru implementarea proiectului se folosește un convertor A.C-D.C de trei faze cu întreg spectrul ( Full-Wave), acesta combină două convertoare A.C-D.C jumătate de spectru, pentru a avea o ondulație de tensiune mai mică. Ondulația de tensiune este o fluctuație a tensiunii de ieșire, care se poate observa la ieșirea convertorului sau la ieșirea sursei de alimentare, aceasta apare din cauza procesului de conversie a energiei. Circuitul nostru conține șase diode. A, B și C reprezintă cele trei semnale sinusoidale, date de către sursa de alimentare, circuitul are șase diode notate de la T1 la T6, acestea funcționează după următoarea metodă (6,1), (1,2), (2,3), (3,4), (4,5), (5,6) iar apoi ciclul se repetă. Astfel se formează tensiunile VAB, VAC, VBC, VBA, VCA și VCB, fiecare corespunzătoare cu diodele sale active, vezi tabelul de conducție.

Tabelul 1. Funcționare A.C-D.C

|  |  |
| --- | --- |
| Tensiunea de ieșire | Diodele în conducție |
| AB | 6,1 |
| AC | 1,2 |
| BC | 2,3 |
| BA | 3,4 |
| CA | 4,5 |
| CB | 5,6 |

Mai departe definim tensiunea VA, VB și VC, care reprezintă tensiunile semnalelor sinusoidale de la intrare, după următoarele formule:

*VA= Vm sin (), VB= Vm\*sin (-120)*, *VC = Vm\*sin (-240),*

*(5)*

unde Vm reprezintă amplitudinea maximă a semnalului de intrare, iar semnalele sunt defazate cu 120 de grade, unul față de celălalt. Astfel putem deduce că

*Vo = VAB = VA-VB = Vm\*sin () – Vm\*sin (-120) =*

*(6)*

Din această ecuație, deducem că tensiunea de ieșire o să fie întotdeauna mai mare și defazată cu 30 de grade.

**4.3 DC-DC Convertor**

În cadrul convertoarelor D.C-D.C, am dorit să prezint două tipuri de convertoare care pot să fie folosite pentru a ne realiza obiectivul. Primul tip fiind cel D.C-D.C Boost sau Step Up Convertor, care a fost implementat în proiect, iar cel de al doilea fiind D.C-D.C Bi-directional, care este mai avansat, folosește și partea de D.C-D.C Bock.

**4.3.1 DC – DC Boost**

Convertorul D.C-D.C Boost, este folosit pentru a creste valoarea tensiunii de la intrare, pană la o valoare dorită de către noi. În imaginile de mai jos, putem vedea schema electrică a acestui tip de convertor. Unde Vin reprezintă sursa de intrare, sau sursa de alimentare. Inductanța L înmagazinează energie în câmpul magnetic creat când comutatorul S este în conducție și eliberează această energie când comutatorul S este blocat, acest lucru ajută la creșterea tensiunii. Comutatorul S, de obicei de tip MOSFET (Metal Oxid Semi-Conductor Tranzistor) sau IGBT ( Tranzistor Bipolar cu poartă Izolată), acestea sunt controlați de către un semnal P.W.M. Mutarea lor regulată din starea de ON sau OFF, pentru a modifica tensiunea de ieșire. A se vedea în pozele de mai jos, când un comutator este OFF, și când un comutator este ON. Vedem că, curenții circulă diferit. Rolul diodei D este de a asigura că, curentul circula de la inductor spre ieșire, când S este OFF, astfel blocând curentul de întoarcere. Capacitatea C are rolul de a filtra tensiunea de ieșire, funcționează ca o rezervă de energie pentru ieșire. Iar Rload reprezintă rezistența internă, care cauzează pierdere de putere și afectează eficiența.

**A diagram of a circuit

Description automatically generated**

Figura 7. Convertor Boost

**A diagram of a rectangle with a blue line and a red line

Description automatically generated**

Figura 8. Convertor Boost circuit închis

**A diagram of a rectangle with a line and a red line

Description automatically generated**

Figura 9. Convertor Boost circuit deschis

**4.4 Baterie**

Pentru partea de baterii o să acoperim două porțiuni importante, și anume B.M.S ( battery management system) și tipul de baterie folosit de către mașinile electrice, Lithiu-Ion.

**4.4.1 BMS**

Sistemul de funcționare al bateriilor este foarte important pentru orice circuit electric care conține baterii. Prin acesta se monitorizează componente cheie ale bateriei, de exemplu curentul și tensiunea de pe baterie, S.O.C care provine de la sistemul de încărcare, S.O.H care provine de la sistemul de viață al bateriei, sau durata de viată a bateriei și S.O.P, care provine de la starea de putere. Un alt aspect important al sistemului de funcționare al bateriei este acela de monitorizare al temperaturii în baterie.

Acestea sunt doar câteva componente de bază ale unui sistem de monitorizare a bateriei, alte componente care trebuie să fie luate în considerare sunt cele de monitorizare, precum monitorizarea tensiunii pentru a ne asigura că bateriile funcționează în parametrii optimi și siguri pentru buna rulare a mașinii, tot în această categorie avem și sisteme de măsurare a curentului care monitorizează curentul pe durata încărcării și descărcării bateriei.

Tot pentru protecția bateriei trebuie să avem sisteme de protecție la supra încărcare, de a prevenii celulele de baterie să fie încărcate mai mult decât maxim lor, protecție la descărcare, care previn descărcarea celulelor sub un anumit prag care poate duce la deteriorarea acestora, avem protecția la scurt circuit și respectiv protecția față de temperaturi ridicate sau temperaturi scăzute.

Comunicarea sistemului de management al bateriei se realizează de cele mai multe ori prin diferite protocoale de comunicare precum SPI, I2C, CAN Bus, SMBus și așa mai departe, pentru a oferii în timp real informații și sesizări despre statusul și viața bateriei.

Un alt aspect foarte important este salvarea datelor pe care le furnizează toate aceste sisteme de securitate și de management, astfel sunt folosite soft-uri de salvare de date (Logging) care o să ne dea performanțele bateriei.

Un ultim aspect care trebuie să fie luat în considerare este integrarea unui alt tip de sistem și anume de management al energiei, pentru a optimiza folosirea energiei.

**4.4.2 Litiu-Ion**

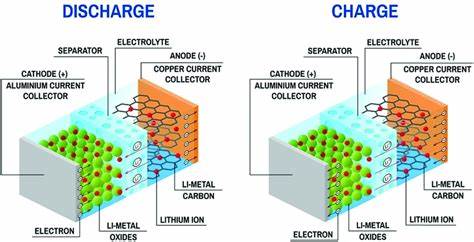


Figura 10. Baterie Litiu-Ion

Cea mai răspândită baterie pentru mașinile electrice este acea cu Litiu-Ion. Studiul nostru de caz se face pe Porsche Taycan Turbo S, așa că o să ne referim la câteva date despre sistemul de înmagazinare a energiei a acestuia. Folosește un număr impresionant de 396 de celule, grupate cate 12 în 33 de module, acestea folosesc o combinație chimică de Nickel-Manganese-Cobalt pentru a balansa densitatea energiei, performanța și longevitatea. Acestea au o greutate de 630kg ( 1389lbs). Bateriile funcționează la o tensiune de 800V, ceea ce este dublu față de marea majoritate a mașinilor electrice care funcționează la 400V. Capacitatea lor este de 93.4kWh. Garanția lor este de 8 ani sau 100.000 de mile (160934.4km).

Aceste baterii sunt compuse din Anod, Catod, Electrolit și un separator. Anodul este compus din grafit, care înmagazinează Litiu - ioni în momentul încărcării, catodul este format din diferite materiale, precum Oxid de Litiu Cobalt, Fosfat de Litiu Fier sau Oxid de Litiu Nichel Mangan Cobalt. Electrolitul este sub forma lichidă și acesta facilitează transportul Litiu Ion dinspre anod spre catod. Separatorul împiedică contactul fizic între anod și catod, în momentul transferului de energie. Metoda de încărcare, Ionii de litiu se deplasează dinspre catod spre anod prin electrolit, integrându-se în grafitul din structura anodului. Iar metoda de descărcare, ionii de litiu se deplasează dinspre anod spre catod, astfel eliberând energia înmagazinată prin electronii ce se deplasează spre surse externe.

**4.5 Invertor**

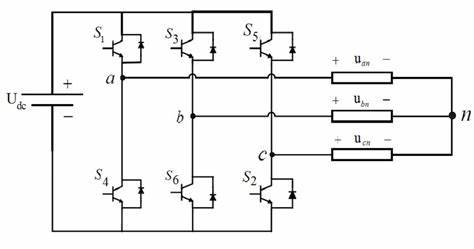


Figura 11. Invertorul 3L

Invertorul 3L este des folosit în electronică de putere pentru a controla motoare trifazice prin metode de control precum F.O.C. Acesta este format din șase tranzistoare IGBT sau MOSFET, sunt comutatoare de putere responsabile pentru controlul tensiunii aplicate motorului electric. Fiecare tranzistor cuprinde și cate o diodă de liberă trecere ( Freewheeling Diode ) care permit trecerea curentului prin stadiul de OFF al tranzistoarelor. Ieșirea circuitului este formată din trei curenți, necesari pentru controlul motorului electric. La ieșirea circuitului se adaugă un circuit format din condensatoare și inductanțe pentru a filtra semnalele sinusoidale de la ieșire. Comutatoarele S1, S3 și S5 reprezintă partea superioară a invertorului, S2, S4 și S5 reprezintă partea inferioară. Aceste comutatoare sunt controlate de un S.V.P.W.M ( Space Vector Pulse With Modulation) sau de un semnal clasic P.W.M prin metoda F.O.C. Rolul acestui tip de invertor este de a genera tensiune trifazică A.C dintr-o sursă de alimentare D.C prin comutarea tranzistoarelor. F.O.C este folosit pentru a controla viteza și cuplul motoarelor A.C prin controlul amplitudinii și a fazei curenților statorului. Invertorul preia semnal P.W.M modulat de la controlerul F.O.C, pentru a produce semnale de tensiunea A.C dorită. Aceste semnale controlează câmpul magnetic și curentul motorului electric pe a obține un cuplu precis și control al vitezei. Controlul P.W.M (Pulse Width Modulation), Invertorul folosește semnalul PWM de pentru a controla deschiderea și închiderea tranzistoarelor, determinând astfel forma și frecvența ieșirii curentului alternativ. Prin ajustarea lățimii impulsurilor de comutare, se obține controlul precis al tensiunii și frecvenței ieșirii. Comutarea tranzistoarelor trebuie să fie bine sincronizată pentru a asigura o ieșire A.C echilibrată și pentru a preveni scurtcircuitul direct între sursa D.C și ieșire. Invertorul trifazat cu sursă de tensiune este utilizat într-o gamă largă de aplicații, precum in industria auto folosit pentru alimentarea motoarelor trifazate care necesită controle precise ale vitezei și cuplului sau in energie regenerabilă prin convertește energia captată de panouri solare sau turbine eoliene, facilitând trimiterea acesteia în rețeaua electrică.

**4.6 CI/CD**

**A diagram of a company

AI-generated content may be incorrect.**

Figura 12. CI/CD

C.I/C.D vine de la Continuă Integrare și Continuă Dezvoltare, scopul acestei metode este de a accelera procesul de dezvoltare al proiectelor de software. Prin crearea de așa zice pipeline-uri, prin care se explică fiecare pas ce trebuie urmărit de către aplicația aleasă, pentru efectuarea acestor pași, unealta folosită în cazul de față, este Jenkins sau GitHub Actions.

Se poate folosi atât partea de interfață, ce ne este dată de către Jenkins, scrierea de cod Groovy pentru Jenkins sau Cod GitHub pentru GitHub Actions. Beneficiile acestor metode sunt depistarea de erori încă din faza de implementare și trimiterea de rezultate direct către dezvoltator, tester sau client. Deoarece putem configura ca, în momentul în care încărcam în repozitoriu proiectul nostru, să pornim șirul de acțiuni prestabilit, care o să ruleze în Cloud, prin intermediul resurselor puse la dispoziție de către GitHub sau de către o entitate pentru Jenkins, sau local, prin intermediul unui agent. Astfel putem depista erorile și să fie rezolvate în timp.

**4.7 GIT**

Git este un mod de versionare a proiectelor, acesta este folosit pentru a menține un istoric al proiectelor și de a avea posibilitatea de a reveni la o variantă mai veche a proiectului, dacă acest lucru este dorit. Prin posibilitatea de a crea o copie a proiectului, se poate realiza munca în echipă. Deoarece un participant la proiect poate să preia o versiune a proiectului, să lucreze pe o parte din proiect, oricât de mult este nevoie, un alt participant poate să ia altă parte din proiect și să lucreze pe partea respectivă, se pot crea ramuri diferite, astfel proiectul să continue pe ramura principală doar când acesta este testat și a ajuns la maturitatea necesară. Astfel ramurile pe care au lucrat dezvoltatorii pot să fie adăugate ramurii principale.

Acest tip de implementare este folosit în proiectele de la firmele mari, cat și cele mici de dezvoltare software, este o unealtă puternică care trebuie să fie stăpânită de către orice inginer software. Prin folosirea unor comenzi simple, se poate versiona și menține sigur proiectul, iar dezvoltarea sa, poate continua armonios. Totodată, acesta ne facilitează funcționarea cu metoda C.I/C.D, prin pornirea acțiunilor noastre, stocarea proiectului sau chiar pentru rularea acestuia și trimiterea de rezultate.

**4.6 F.O.C**

O mare parte a proiectului, este crearea acestui modul de control al curentului. Pentru a putea înțelege de ce ne dorim acest lucru este necesar să explicam, în mare, câteva idei care stau în spatele acestui tip de control. Vom trece prin prezentarea transformării Clark și a transformării Park.

În primul rând trebuie să luam la cunoștință că în sistemul nostru folosim un motor electric trifazic, înseamnă că trebuie să putem control cei trei curenți care intră în acest motor. În motorul electric avem un stator, acesta este partea fixă a motorului, care conține înfășurarea statorică, care este compusă din multiple straturi de fir de cupru, acesta generează câmpul magnetic, este amplasat în partea exterioară. În partea interioară, se află rotorul care este partea mobilă formată din magneți.

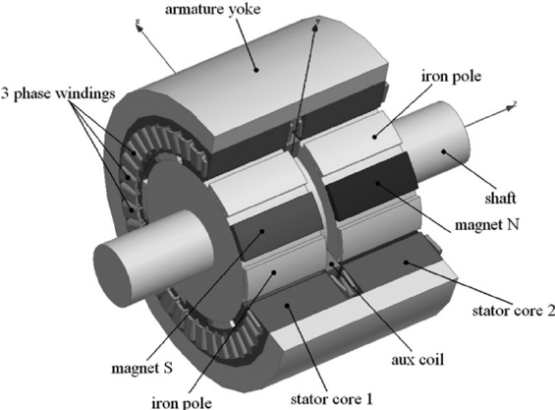


Figura 13. Stator si Rotor in componenta unui motor electric

Aceste componente generează trei curenți, o să îi notăm cu I1, I2 și I3, vor fi reprezentați pe o axa, sub formă sinusoidală defazați cu 120 de grade. Totodată, prin acești curenți se poate determina direcția fluxului magnetic total, prin măsurarea acestuia în diferite puncte, pentru fiecare semnal sinusoidal în parte.

A diagram of a graph

Description automatically generated

Figura 14. Curenții A B C

Mai departe, când vine vorba de mașini electrice de curent continuu, avem nevoie să calculăm doar o singură tensiune de ieșire și să controlăm un singur curent D.C, pentru a avea ca și rezultat cuplul dorit, iar pentru mașinile de curent alternativ, avem nevoie să calculăm trei tensiuni de ieșire și să controlăm trei curenți, în funcție de poziția statorului, pentru a calcula cuplul dorit, astfel avem nevoie de o metoda mult mai rapida și mai simplificată. Pentru acest caz, s-a format metoda F.O.C. Mai departe se face translarea grafului nostru, cu semnale sinusoidale, într-un graf 2D, reprezentat pe trei axe, cate o axă pentru fiecare semnal, acesta se va numi graful a.b.c. Întotdeauna axa a, va fi la 0 grade, b o să fie la 120 de grade și c o să fie la 240 de grade. Pentru fiecare axa se definește un vector Va, Vb și Vc iar suma lor v-a defini vectorul V. Va o să fie lungimea vectorului, valoare reala iar sunt valorile funcției complex exponențială, fiecare reprezentant 0, 120 și 240 de grade. Graful nostru, are atât parte pozitivă cat și parte negativă.

(7)

Putem numi Va, Vb și Vc coordonatele lui V și să notăm:

(8)

A diagram of a circle with arrows and a circle with arrows

Description automatically generated with medium confidence

Figura 15. Vectorii A B C pe graf

Prin acest graf 2D, putem reprezenta orice valoare din sistemul de coordonare tri-fazic printr-un vector cu lungime de 3/2.

A diagram of a graph

Description automatically generated

Figura 16. Lungimea și direcția de deplasare a vectorilor A B C

Transformarea Clark este definită prin trecerea din graful a.b.c în graful alfa-beta. Acest graf este reprezentat prin axa alfa și prin axa beta, care formează un unghi de 90 de grade, și care este suprapus pe axa abc, prin proiecția vectorilor Va, Vb și Vc pe axele alfa și beta, putem defini vectorii Valfa și Vbeta, folosim rația trigonometrica pentru a defini lungimea noilor vectori.

(9)

(10)

A diagram of a circle with arrows

Description automatically generated

Figura 17. Vectorii Alfa și Beta pe graf

(11)

(12)

Pentru realizarea Transformării Clark Inverse, se fac calcule în sens invers, prin proiecția vectorilor Valfa și Vbeta, pe axele Va, Vb și Vc. Astfel, ne vor rezulta următoarele ecuații:

(13)

(14)

(15)

După aceste ecuații, putem vizualiza Transformata Clarke, respectiv semnalele rezultate din acestea. În ecuațiile noastre, atât pentru transformata directă cat și pentru transformata inversă, f poate să fie înlocuit cu tensiune, curent sau flux.

A diagram of mathematical equations

Description automatically generated

Figura 18. Transformata Clark directă și inversă

În continuare, o să vorbim despre Transformarea Park, aceasta se realizează cu ajutorul grafului dq, în care q înseamnă cuadratură iar d direct. Axele dq sunt formate prin mutarea axelor alfa-beta cu un unghi teta. Aceste axe se rotesc, la viteza rotorului, astfel se face tranziția de la axe fixe la axe în mișcare, dar le putem vizualiza ca și axe fixe, deoarece vectorii se rotesc și ei împreună cu axele la aceeași viteză, astfel poziția rămâne la fel. Pentru a putea defini Transformarea Park directă și inversă, trebuie să proiectăm pe axele de coordonate dq pentru transformarea directă și pe axele alfa-beta pentru transformarea indirectă. Același principul folosit și la axele de mai sus.

(16)

(17)

(18)

(19)

A circle with arrows and a red arrow

Description automatically generated

Figura 19. Vectorii dq pe graf

După efectuarea calculelor, ajungem la câteva analize care descriu această transformare, putem înlocui f cu tensiune, curent sau flux.

A diagram of a solar system

Description automatically generated with medium confidence

Figura 20. Transformata Park directă și inversă

**4.7 Sisteme de siguranță**

Sistemele de siguranță au rolul de a menține circuitul nostru electric în parametrii optimi de funcționare, iar in cazul oricărei abateri de la limitele impuse să se ia masurile prevăzute. Astfel în cazul unei erori, de la o simplă eroare pană la o eroare majoră care ar putea duce la distrugerea completă sau parțială a sistemului, circuitul nostru de siguranță trebuie să se asigure că această eroare este capturată și că ține circuitul sub control, tot în cadrul sistemului de siguranță intră și componenta de ținere a unui istoric pentru ceea ce am realizat, astfel putând vedea ceea ce a cauzat eroarea respectivă.

Componentele care fac parte din gama electronicelor de putere, precum este și schema noastră electrică sunt componente cruciale pentru conversia electricității D.C în electricitate A.C, foarte folosite în domeniul energiei reutilizabile, sursele de putere care nu pot să fie întrerupte și motoarelor electrice. Pentru aceste domenii trebuie realizate sisteme de siguranță specifice, o să enumeram câteva categorii care trebuie respectate și luate în considerare.

O să începem prin protecția la supra curent, pentru a prevenii acest efect putem folosii puncte de închidere a circuitului care să prevină deteriorarea componentelor sau chiar incendierea acestora. Tot în această categorie intră și senzorii și limitatorii care monitorizează trecerea curentului prin invertor sau prin alte componente cheie are schemei electrice.

Următoarea categorie o să fie protecție la supra voltaj, pentru acest punct se folosesc varistor sau diode T.V.S ( Transient Voltage Suppression), acestea absorb creșterile bruște de tensiune, protejând astfel invertorul de aceste creșteri bruște. Un alt mod care a fost implementat în proiect este dat de către restricțiile de tensiune, acestea verifică dacă tensiune dintr-o anumită zonă respectă valorile normale, daca nu este conformă acesta v-a ridica o eroare.

Mai departe o să putem vorbi de protecția la temperaturi mari care pot dăuna grav sistemul. Cele mai des utilizate componente sunt senzorii de temperatură care monitorizează în continuu temperatura în puncte cheie ale sistemului precum comutatoarele. În caz de temperaturi ridicate, acestea semnalează o eroare care duce la oprirea totală a sistemului sau reducerea tensiunii de alimentare pană în parametrii optimi. Mai există o metoda numita deconectarea directă a componentei care produce temperatura ridicată.

Următoarea metodă de protecție este protecția față de scurt ( Short Circuit Protection) care presupune folosirea de mecanisme de comutare rapidă care deconectează rapid invertorul dacă un scurt este detectat. Se mai folosesc protecții diferențiale care compară ieșirea cu intrarea pentru a detecta valori neconforme.

Următoarea categorie este protecția la izolare și defecte la împământare. Pentru această categorie se folosesc detectoare de defecte de împământare, acestea detectează defectele de împământare și declanșează acțiuni de protecție pentru a evita orice fel de pericol de electrocutare ( folosite pentru sistemele realizate fizic). O altă componentă folosită sunt transformatoarele de izolare, acestea asigura izolare galvanică pentru a separa circuitele cu valori diferite și pentru a garanta siguranța tuturor celor implicați.

**4.8 A.I & Reinforcement Learning**

**A diagram of a company

AI-generated content may be incorrect.**

Figura 21. Reinforcement Learning

Unul dintre cele mai răspândite domenii din ziua de azi este cel al Inteligenței Artificiale. Tehnologia A.I datează din anii 1950, când pionieri precum Alan Turing și John McCarthy au început să exploreze posibilitatea creării de mașini inteligente. Turing a propus celebrul test Turing pentru a evalua inteligența unei mașini, iar McCarthy a fost cel care a inventat termenul de "inteligență artificială" în 1956. În anii 1980, sistemele de inteligenta artificiala au cunoscut un prag înalt datorită capacității lor de a simula cunoștințele umane în domenii specifice si vaste, cum ar fi tehnologie sau biologie. Cu toate acestea, limitările tehnologice ale vremii au împiedicat crearea unor sisteme cu adevărat "inteligente".

Testul Turing se face cu ajutorul unui evaluator uman, care pune aceleași întrebări atât unui calculator cat și unui computer, dar identitatea lor este ascunsa, Evaluatorul va încerca să decidă care dintre participanți este uman și care este mașină bazându-se doar pe răspunsurile primite. Dacă evaluatorul nu poate determina în mod constant care dintre participanți este computerul, atunci se consideră că mașina a trecut testul, demonstrând o formă de inteligență similară cu cea umană.

În prezent A.I a început să fie folosit în toate ariile, atât cele tehnice cat și medicale sau școlare, fiind o bază a unui nou viitor spre care lumea se dezvoltă. Acesta a fost conceput prin ideea de a ajuta și de a simplifica viața utilizatorilor, dar și pentru dezvoltarea și cercetarea diferitelor teme precum descoperirea unui remediu pentru cancer. Inteligența Artificială vine sub multe forme, aceasta are la bază un cumul foarte mare de date, unele specializate pentru un anumit set de date iar altele cuprind date universale, acestea sunt filtrare prin intermediul algoritmilor matematici și formează modul de gândire al Inteligenței Artificiale. Inteligența Artificială poate să fie împărțită în trei mari categorii, Rule-Based A.I, care conține un set bine definit de reguli și logică, Machine Learging (M.L) care învață asemănările din cumuli mari de date pentru a putea face alegerile corecte, iar ultima este Deep Learning, acesta este asemănător cu Machine Learging doar că folosește rețele neuronale pentru a învăța.

Reinforcement Learning este implementat în acest proiect, în sistemul de F.O.C. Este o subdiviziune al Machine Learning, împreună cu Sepervised Learning. Aceasta funcționează printr-un agent care învață din greșeli și câștiguri, învață să facă mișcări corecte prin interacțiunea cu mediul înconjurător, după mișcările corecte, primește un câștig, iar după mișcările greșite, primește o penalitate. Scopul este de a învăța o strategie care să maximizeze câștigurile în timp.

Componentele pentru Reinforcement Learning sunt următoarele:

Agent: Acesta reprezintă cel ce învață sau ia deciziile, în cazul nostru, acesta o să controleze motorul electric.

Mediul: Lumea în care agentul interacționează, în cazul de față acesta este mediul de dezvoltare Simulink.

Observații: Informații din mediu pe care agentul le folosește pentru a-si crea drumul.

Acțiunile: Alegerile pe care agentul le poate face, precum creșterea sau scăderea vitezei motorului.

Câștig: Informațiile pe care le primește agentul bazate pe acțiunile lui.

**5. Implementarea soluției adaptate**

**A diagram of a computer

AI-generated content may be incorrect.**

Figura 22. Circuitul proiectului

În capitolul ce urmează, o să discutăm despre implementarea proiectului pas cu pas. Vom vedea cum au fost gândite și cum au fost create modulele pentru fiecare parte din proiect.

Pentru simularea codului se foloseste combinatia de taste CTRL+S, pentru a compila proiectul se foloseste combinatia de taste CTRL+D, iar pentru a contrui proiectul se foloseste combinatia de taste CTRL+B.

Înainte de a intra în partea de implementare a proiectului trebuie să facem setările de bază și anume setarea timpului de rulare. Se începe prin accesarea panoului de configurare. Aici setam timpul de terminare la 10.0 secunde, se poate pune și un timp mai lung, vizionam pagina Solver.

Mai departe o să prezentam câteva dintre setările ce trebuie făcute pentru a putea construi proiectului.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 23. Solver

Următorul punct este pentru generarea de cod dacă este necesar, vom selecta grt.tlc, limbajul dorit și trebuie să avem un compailer C/C++ instalat. Dar se pot alege și alte variante, de exemplu, putem alege grt.tlc care este Generic Real Time Target, acesta este utilizat pentru generarea de cod C/C++ pentru aplicații de simulare care nu necesită capacitați de timp real specifice hardware lui utilizat. Următorul este realtime.tlc, Visual C/C++ Solution for Simulink Coder, acesta este folosit pentru a genera soluții C/C++ care pot să fie folosite pentru dezvoltarea de aplicații în mediul de dezvoltare Visual Studio. Rsim.tlc, provine de la Rapid Simulation Target, ideale pentru simulări rapide și teste. Rtwsfcn.tlc, S-Function Target, utilizat pentru generarea de funcții S-Function care sunt module de Simulink utilizate pentru implementării algoritmilor personalizați în cadrul Simulink. Slrt.tlc și slrtert.tlc, Simulink Desktop Real-Time, avem rularea simulărilor în timp real direct pe desktop. Slrealtime.tlc, Simulink Real-Time, acesta este destinat aplicaților care necesită simulare în timp real pe hardware dedicat, cum sunt sistemele de control dedicat. Systemverilog\_dpi\_ert.tlc și systemverilog\_dpi\_grt.tlc, System Verilog DPI Component Generator, acestea sunt folosite pentru integrarea componentelor generate din Simulink în fluxuri de simulare HDL, prin intermediul interfeței de programare directă System Verilog DPI.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 24. Generare Cod

**5.1 Implementare Trifazic**

Pentru implementarea sursei de alimentare Tri-Fazic, o să folosim blocul pus la dispoziție de către Simulink. Acesta este setat la 400V la o frecvență de 50Hz, așa cum ne indică norma de putere pentru Tri-Fazic pentru Romania. Acest modul are trei ieșiri, fiecare ieșire corespunzătoare pentru o fază dată de către una din cele trei surse de curent alternativ. Iar pentru vizionarea lor, sunt conectate la un modul de măsuratură pentru curent și tensiune.

A diagram of a circuit

Description automatically generated

Figura 25. Implementare sursă Tri-Fazic

**5.2 Implementare AC-DC Convertor**

Pentru implementarea Convertorului nostru A.C-D.C, trebuie să urmăm modelul prezentat în capitolul teoretic de mai sus. În acel capitol am vorbit despre calculele și schema electrică care trebuie realizată pentru a putea dezvolta această parte din proiect. Astfel, se realizează schema electrica în Simulink, cu, pinii de conectare pentru cele trei semnale de fază de la sursa de alimentare, la intrarea circuitului. Se aleg diode cu o rezistentă internă mică de 0.001 Ohms și cu un Forward Voltage implicit la 0.8 V. Rezistența Snubber de 500 Ohms și capacitate Snubber 250e-9. Rezistența de ieșire o să fie de 1 Ohm iar pentru a putea vizualiza semnalul de ieșire, avem nevoie de blocuri de măsurare a tensiunii și blocuri de afișare. Mai departe se face calculul valorii medii a semnalului de ieșire pentru a putea măsura corect valoarea tensiunii de ieșire.

Ecuația pentru calculul valorii medii este următoarea:

(20)

A screenshot of a computer

Description automatically generated

Figura 26. Implementare AC-DC Convertor Full-Wave

A diagram of a function

Description automatically generated

Figura 27. Calculul valorii medii

**5.3 Implementare DC-DC Boost**

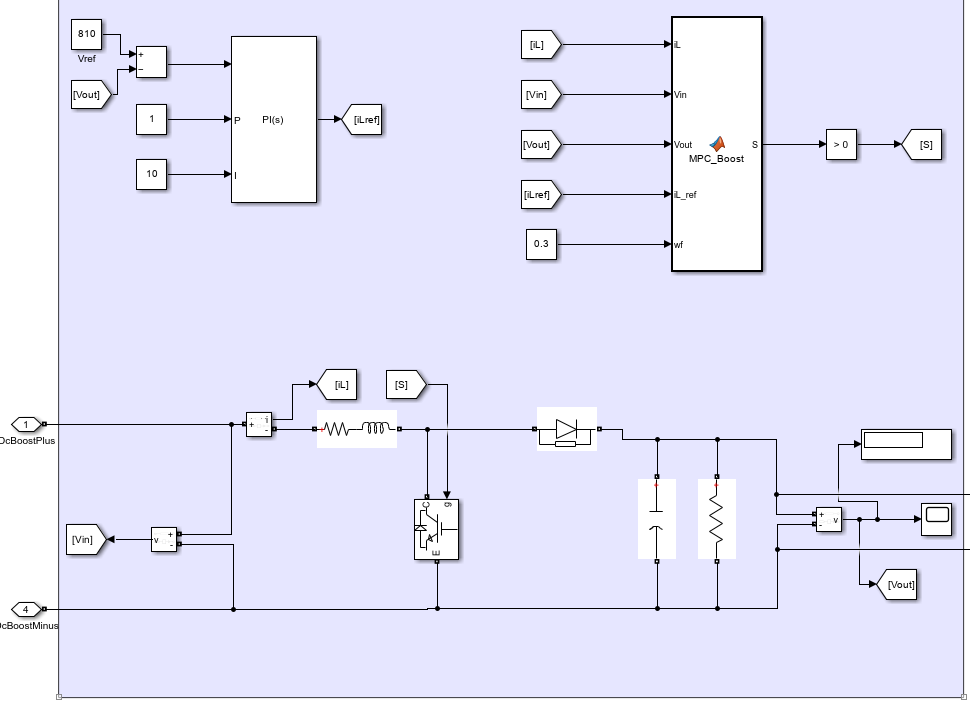
****

Figura 28. Implementare D.C-D.C Boost

Pentru implementarea D.C-D.C Boost, am folosit schema prezentată în partea de teorie și principiile prezentate. Astfel ceea ce am dorit să adaug în această schemă deja cunoscută, a fost componenta denumită M.P.C\_Boost. Rolul acestei componente M.P.C este de a asigura valoarea tensiunii de ieșire în funcție de multiplii parametrii. Componenta primește informații de la tensiunea de ieșire și de la alți parametrii precum tensiunea de intrare și curentul pe bobină.

Funcția Matlab M.P.C\_Boost (Model Predictive Control) este concepută în felul următor, începem prin definirea componentei de control iL, care este curentul prin bobină, apoi se definește funcția cost, care este o expresie matematică ce evaluează performanțele sistemului de control, arătând cat de bine își face acesta lucrul. Aceasta este formata din curentul estimat la un pas k minus valoarea curentului la pasul k+1.

(21)

Trebuie să definim formula lui Euler explicită pentru aplicația noastră, unde Ts este timpul de eșantionare sau perioada de timp dintre două măsurători consecutive, iL curentul din inductor la pasul de timp următor iar iL este curentul din inductor la pasul de timp actual. Aceasta formulă este folosită pentru a prezice dinamica curentului prin inductor.

(22)

Astfel prin rearanjarea componentelor ajungem la următoarea formulă:

(23)

Se iau cele două cazuri posibile S=1 și S=0, pentru S=1 formula este următoarea:

(24)

Iar pentru S=0 avem următoarea formulă:

(25)

Prin combinarea celor două ecuații ajungem la următoarea ecuație:

(26)

Și astfel definim funcția de predicție pentru controlul curentului de inducție:

(27)

**5.4 Implementare Baterie**

**A screenshot of a computer

AI-generated content may be incorrect.**

Figura 29. Implementare baterie

Pentru implementarea bateriei am folosit un bloc specific din librăria Simulink. Acest bloc replică o baterie de tip Litiu-Ion, foarte des folosită în industria auto. Ieșirile acestui bloc sunt Curentul, Tensiunea și S.O.C care ne indică cat la sută este încărcata bateria. Prin produsul dintre curent și tensiune ne rezultă Puterea. Sub această implementare am introdus și un bloc care măsoară S.O.H, care ne arată starea de sănătate a bateriei, fiind un circuit fără erori și destul de simplu făcut, starea bateriei o să rămână perfectă.

**5.5 Implementare Invertor 3L**

**A diagram of a computer program

Description automatically generated**

Figura 30. Implementare invertor 3L

Invertorul 3L este un tip de convertor din gama electronicelor de putere care convertește putere de curentul direct D.C în putere de curent alternativ A.C prin trei nivele de putere, pozitiv, negativ și zero. Acest design permite o limitare a zgomotului și un stres mai mic asupra tranzistoarelor.

În modelul de față, care este și cea mai răspândita metodă, se folosesc tranzistoare I.G.B.T (Insulated Gate Bipolar Transistor) ca și elemente de comutare, în figura de față avem șase astfel de tranzistoare, trei pentru ramura de sus și trei pentru ramura de jos. Acestea sunt configurate cu o rezistență internă mică de 0.001Ohm, rezistența Snubber de 1e5. Aceste tranzistoare au nevoie de semnale de control de tip P.W.M, care este oferit de către implementarea F.O.C, are rolul de a determina care tranzistoare sunt deschise și care sunt închise. Schema de la ieșirea ramurilor noastre o să filtreze semnalul de ieșire. Fiecare două tranzistoare sunt conectate la cate o fază de ieșire, acesta aranjare ne permite să avem cele trei nivele de tensiune. Porțile logice S1, S2 și restul sunt folosite pentru a sincroniza și pentru a da semnalele necesare pentru deschiderea și închiderea tranzistoarelor.

**5.6 Implementare F.O.C**

**A diagram of a computer

AI-generated content may be incorrect.**

Figura 31. Implementare F.O.C

Implementarea F.O.C se realizează prin trecerea semnalelor de la ieșirea motorului nostru electric P.M.S.M prin mai multe faze matematice. Semnalele de ieșire necesare sunt următoarele, viteza și unchiul rotorului, măsurate în radiani și radiani pe secundă. Respectiv curenții de pe Stator măsurați în Amperi. Să trecem prin acești curenți:

Viteza Rotorului: acest semnal reprezintă viteaza rotorului în radiani pe secundă, este o componentă necesară care ajută la determinarea intrărilor necesare pentru a menține o performanță necesară a motorului.

Unchiul Rotorului: acest semnal ne indică poziția rotorului în radiani. Poziția acestuia este esențială pentru transformarea Park, care transformă curenții din axa Alpha-Beta în axa dq.

Curenții statorului și anume curenții de pe faza A, B și C, acești curenți sunt esențiali pentru a determina cuplul instant și sunt folosiți pentru a transforma curenții de trei faze în curenții de două faze prin transformarea Clarke.

După cum putem observa, semnalele sinus și cosinus sunt derivate din semnalul ce ne dă poziția rotorului, acestea sunt folosite atât pentru transformarea Park cat și pentru a genera semnalele PWM necesare pentru controlul inventorului 3L. În transformarea Park, semnalul cosinus este folosit împreună cu semnalul direct (Id) al transformatei, iar semnalul sinus este folosit împreună cu semnalul de quadratura (Iq).

În continuare, cele trei semnale is\_a, is\_b si is\_c date de către Stator sunt folosite pentru a crea semnalele Alpha și Beta prin transformarea Clarke. Rolul acestei transformate este de a trece din sistemul de coordonate trifazic în sistemul de coordonate bi-fazic. Conform calcului prezentat în partea de teorie, semnale Alpha și Beta sunt calculate cu ajutorul semnalelor de intrare.

**A diagram of triangles and lines

AI-generated content may be incorrect.**

Figura 32. Implementare Transformata Clark

Următorul pas este transformarea Clarke, în care trecem din sistemul de coordonate bi-fazic Alpha-Beta în sistemul de coordonate bi-fazic DQ. După implementarea formelor descrise în partea de teorie ne rezultă semnalele D și Q. Iar in următorul pas, aceste semnale sunt folosite pentru partea de Reinforcement Learging care este explicată în următorul capitol.

**A diagram of a diagram

AI-generated content may be incorrect.**

Figura 33. Implementare transformata Park

Îeșirea blocului de Reinforcement Learning este folosită pentru a genera semnalele Ds și Qs. Care la rândul lor sunt trecute prin transformarea Park Inversă și în cele din urmă folosite pentru a genera semnalele P.W.M necesare pentru controlul invertorului 3L

**A diagram of a power generator

AI-generated content may be incorrect.**

Figura 34. Implementare Transformata inversă Park

**5.7 Implementare Reinforcement Learning**

În implementarea Reinforcement Learning, o să trecem prin două aspecte, primul o să fie implementarea în cod a acestui tip de Machine Learning folosind Matlab, iar cea dea a doua parte o să fie implementarea în sistemul mare din Simulink și rolul său în toata topologia F.O.C.

Tipul nostru folosit se numește Twin Delayed Deep Determinustic Policy Gradient, acesta funcționează în felul următor. Se începe prin partea de arhitectură Actor-Critic, unde actorul este acea parte a agentului care decide ce acțiune să ia în funcție de observația curentă. Iar Critic, această parte evaluează acțiunea luată de către agent prin preconizarea potențialului câștig (Valoarea Q). Următoarea parte este cea de învățare a procesului. Agentul începe prin acțiuni aleatorii și explorează mediul pentru a colecta experiență ( Observații, Acțiunii și Câștiguri). În timp ce agentul acționează, acesta înregistrează observațiile și câștigurile într-un buffer de memorie. Agentul se antrenează periodic cu aceste date. TD3, modulul folosit de către noi folosește două rețele critice pentru a prezice valoarea Q. Acesta ajuta la minimizarea valorilor mai mari decât valorilor posibile. Când se face calcularea valorii dorite, algoritmii TD3 ia minimum dintre două valori Q de la critic. Acesta asigură un proces de învățare mai curat. TD3 folosește target networks, care sunt versiuni mai lente ale actorului și a rețelei critice, acestea sunt folosite pentru stabilizare, ieșirile acestor rețele sunt folosite pentru compunerea valorii Q în locul folosirii rețelei direct. TD3 adaugă și zgomot acțiunilor luate de către agent pentru a încuraja explorarea, acestea ajută agentul să găsească noi strategii în timp și să se perfecționeze. Învățarea continuă pentru multe episoade, un episod este un drum complet prin care acesta încearcă să ajungă la destinație.

Codul începe prin definirea valorilor pentru frecvența P.W.M, perioada acesteia care este unu supra P.W.M, sampling time și tipul de date. Următorul pas este de a definii Observațiile și Acțiunile. Avem opt date de intrare, astfel agentul v-a observa aceste opt date de intrare despre motor, aceste sunt: Viteza de referința ( Speed Ref), reprezintă viteza dorită de către motor, aceasta valoare este folosita pentru ca R.L să determine cat de departe este de valoarea dorită a vitezei. Viteza răspunsului (Speed Feedback), este viteza actuală a motorului, agentul compară această valoare cu valoarea vitezei de referință pentru a calcula eroarea și a ajusta semnalul de control. Id, curentul direct și Iq curentul de cuadratura, care sunt curenții din sistemul F.O.C și respectiv curenții de referință a acestora.

PWM\_frequency = 1e3;%PWM motor frequency set to 1000Hz

T\_pwm = 1/PWM\_frequency;%Period of the Frequency

Ts = T\_pwm; %Sampling time

Ts\_simulink = T\_pwm/2;%Sampling time

Ts\_speed = 10\*Ts; %Sampling time

dataType = 'single';

Curentul Iq este curentul de cuadratura, folosit pentru a înțelege cat de eficient sistemul nostru produce cuplu. Curentul Id este esențial pentru a demonstra cum câmpul magnetic afectează performanța.

numObservations = 8;%Dimension of the observation space

observationInfo = rlNumericSpec([numObservations 1]);%Observations are numeric with additonal name and descriptions

observationInfo.Name = 'observations';

observationInfo.Description = 'Information on error and reference signal';

numActions = 2;%defines the dimension of the action space

actionInfo = rlNumericSpec([numActions 1]);

actionInfo.Name = 'vqdRef';

agentblk = 'DisertatieVanca/Reinforcement Learning/RL Agent';%Specifies the block in Simulink

env = rlSimulinkEnv(mdl,agentblk,observationInfo,actionInfo);%Creates an reinforced learning environment that connects Simulink with actions and observations

rng(0) % fix the random seed

Următorul punct este crearea rețelei actor și critic, rețeaua Critic preia observațiile și acțiunile ca și intrări, și avem definirea celor trei căi din rețeaua Critic, State Path care este responsabilă de procesarea observațiilor folosind o cale conectată, Action Path care procesează acțiunile similare și Common Path care combină cele două căi menționate mai sus pentru a produce valoarea Q. Acest cod TD3 folosește două astfel de rețele Critic.

statePath = [featureInputLayer(numObservations,'Normalization','none','Name','State')

fullyConnectedLayer(64,'Name','fc1')];

actionPath = [featureInputLayer(numActions, 'Normalization', 'none', 'Name','Action')

fullyConnectedLayer(64, 'Name','fc2')];

commonPath = [additionLayer(2,'Name','add')

reluLayer('Name','relu2')

fullyConnectedLayer(32, 'Name','fc3')

reluLayer('Name','relu3')

fullyConnectedLayer(16, 'Name','fc4')

fullyConnectedLayer(1, 'Name','CriticOutput')];

criticNetwork = layerGraph();

criticNetwork = addLayers(criticNetwork,statePath);

criticNetwork = addLayers(criticNetwork,actionPath);

criticNetwork = addLayers(criticNetwork,commonPath);

criticNetwork = connectLayers(criticNetwork,'fc1','add/in1');

criticNetwork = connectLayers(criticNetwork,'fc2','add/in2');

criticOptions = rlRepresentationOptions('LearnRate',1e-4,'GradientThreshold',1);

critic1 = rlQValueRepresentation(criticNetwork,observationInfo,actionInfo,...

'Observation',{'State'},'Action',{'Action'},criticOptions);

critic2 = rlQValueRepresentation(criticNetwork,observationInfo,actionInfo,...

'Observation',{'State'},'Action',{'Action'},criticOptions);

Următoarea este definirea rețelei Actor, care decide ce acțiune să ia în funcție de observațiile curente. La fel ca și rețeaua critic, rețeaua Actor procesează statusul actual. Folosește o funcție tanh de activare care ajută acțiunile să rămână într-un anumit domeniu.

actorNetwork = [featureInputLayer(numObservations,'Normalization','none','Name','State')

fullyConnectedLayer(64, 'Name','actorFC1')

reluLayer('Name','relu1')

fullyConnectedLayer(32, 'Name','actorFC2')

reluLayer('Name','relu2')

fullyConnectedLayer(numActions,'Name','Action')

tanhLayer('Name','tanh1')];

actorOptions = rlRepresentationOptions('LearnRate',1e-3,'GradientThreshold',1,'L2RegularizationFactor',0.001);

actor = rlDeterministicActorRepresentation(actorNetwork,observationInfo,actionInfo,...

'Observation',{'State'},'Action',{'tanh1'},actorOptions);

Următorul pas este crearea agentului TD3, acesta are nevoie de următoarele puncte, Sample Time, este intervalul la care agentul face acțiuni, Discount Factor, reprezintă cat de mult la agent ii pasă de câștigurile următoare, unu fiind valoarea cea mai mare, Experience Buffer, este o memorie pe termen lung care stochează acțiunile anterioare pentru a învăța, Mini-Batch Size, numărul de experiențe de învățat, Smoothing Factors, controlează evoluția rețelei pentru a avea o stabilitate liniară.

Ts\_agent = Ts;

agentOptions = rlTD3AgentOptions("SampleTime",Ts\_agent, ...

"DiscountFactor", 0.995, ...

"ExperienceBufferLength",2e6, ...

"MiniBatchSize",512, ...

"NumStepsToLookAhead",1, ...

"TargetSmoothFactor",0.005, ...

"TargetUpdateFrequency",10);

agentOptions.ExplorationModel.Variance = 0.05;

agentOptions.ExplorationModel.VarianceDecayRate = 2e-4;

agentOptions.ExplorationModel.VarianceMin = 0.001;

agentOptions.TargetPolicySmoothModel.Variance = 0.1;

agentOptions.TargetPolicySmoothModel.VarianceDecayRate = 1e-4;

agent = rlTD3Agent(actor,[critic1,critic2],agentOptions);

Ultima parte este antrenarea agentului, trebuie să definim timpul total de antrenare în secunde, numărul maxim de episoade și numărul maxim de pași pe episod. Mai avem nevoie de un criteriu de oprire în momentul în care câștigul mediu cade sub un anumit prag.

T = 5.0;%Total training durations

maxepisodes = 1000;%Limit how many training episodes can occur

maxsteps = ceil(T/Ts\_agent); %Defines the maximum steps per episode based on sampling time

trainingOpts = rlTrainingOptions(...

'MaxEpisodes',maxepisodes, ...

'MaxStepsPerEpisode',maxsteps, ...

'StopTrainingCriteria','AverageReward',...

'StopTrainingValue',-190,...

'ScoreAveragingWindowLength',100);

**A diagram of a computer

Description automatically generated**

Figura 35. Implementare Reinforcement Learning

A diagram of a computer program

Description automatically generated

Figura 36. Implementare Reinforcement Learning Submodul

Blocul de Observare primește ca și intrări semnalele menționate mai sus, din acestea rezultă și semnalele Iq eroare și Id eroare folosite pentru blocul de castig. Blocul castig calculează câștigul bazat pe erori metrice și penalizări, erori mari pătratice primesc un câștig de -5, micile penalități primesc un câștig de -0.1 și o penalitate foarte mare de -100 dacă procesul este gata dar nu este funcțional. Blocul Este Gata, oprește sistemul bazat pe un algoritm, valoarea curentului Iq ref, dacă este mai mare sau cel puțin egală cu unu, iar valoarea Sample time pentru agent să fie mai mare sau egală cu trei. Acest bloc de Reinforcement Learning este folosit pentru a calibra parametrii Ds și Dq, care sunt prezentați în partea de implementare F.O.C.

**5.8 Implementare Sisteme de siguranță**

**A white paper with yellow arrows

AI-generated content may be incorrect.**

Figura 37. Implementare modul Erori

**A diagram of a graph

AI-generated content may be incorrect.**

Figura 38. Implementare modul clasificare Erori

Pentru implementarea sistemului de siguranță o să luăm toate semnale din sistem. Aceste semnale vor fi duse într-un sub sistem separat unde o să fie verificate dacă funcționeze între anumite limite. Dacă apare o creștere mare sau o scădere bruscă a valorii unui semnal, acesta o să formeze un semnal de eroare care mai departe o să fie catalogat ca semnal de eroare mare, semnal de eroare mică sau semnal de eroare medie. În funcție de tipul de semnal sistemul trebuie să răspundă corespunzător.

**5.9 Implementare Git**

Pentru implementarea Git, nu o să intrăm prea mult în amănunte, aceasta se realizează prin instalarea pachetului Git de pe pagina cu același domeniu, apoi crearea unui cont pe platforma GitHub, crearea unui repozitoriu nou și apoi prin folosirea Git Bash și a acțiunilor deja cunoscute, putem versiona proiectul după bunul plac. După fiecare modificare făcută asupra proiectului, se recomandă să intrăm în git bash, și să adăugam următoarele comenzi, git status, pentru a putea vedea branch-ul pe care lucram și fișierele care au fost modificate. Apoi git add . pentru a adăuga tot în stadiul de Index, apoi git commit -m „ adăugare comentariu” pentru a duce fișierele modificate în stadiul de commit. Apoi git push, pentru a le încarcă în cloud pe GitHub în repozitoriu nostru. Se recomandă adăugarea de comentariu scur. Se pot folosi comenzi precum gitk –all& pentru vizionarea întregii ramuri git.

A screenshot of a computer screen

Description automatically generated

Figura 39. Git add .

A screen shot of a computer program

Description automatically generated

Figura 40. Git commit -m „Mesaj”

A computer screen shot of a program

Description automatically generated

Figura 41. Git push

**5.10 Implementare GitHub Actions**

Modul de implementare C.I/C.D ales este GitHub Actions, deoarece este mai nou, mai actual și mai bine utilizat pentru proiecte mai mici. Primul prag este alegerea repezitorului dorit, apoi mergem pe actions, trecem peste tutorial și ne duce pe o pagină unde vom scrie codul nostru într-un fișier yaml. Pentru această parte din proiect, ne dorim să avem un astfel de fișiere, unul care să ruleze proiectul principal.

Un pas important este crearea unui runner, care să compileze proiectul nostru și toți pașii de care avem nevoie, acesta o să fie activat prin simpla comandă de git push, în git bash, în momentul în care ne dorim să încărcăm proiectul în cloud. Următorul pas este de a scrie codul pentru fișierul yaml, acest fișier o să fie interpretat de către GitHub Actions, și va fi rulat.

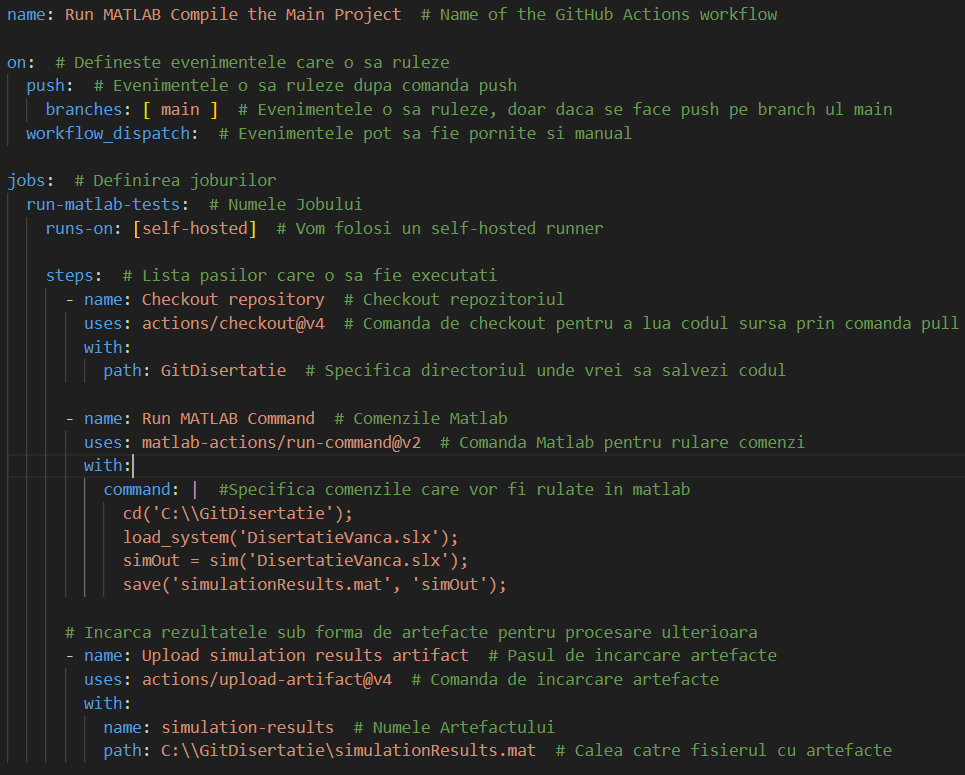


Figura 42. Cod C.I/C.D

**6 Rezultate experimentale**

**6.1 Rezultate Tensiune Curenți si Putere**

Pentru partea de rezultate experimentale, o să specificam curenții, tensiunile și puterile, respectiv alți parametrii care sunt importanți pentru realizarea proiectului.

Vom începe prin prezentarea tensiunii de ieșire de la circuitul A.C-D.C, după cum bine am explicat în partea de teorie, aceasta trebuie să fie de 1.73 ori mai mare decât tensiunea de intrare, la intrare vom avea 400 de volți dați de către sursa de alimentare continuu, iar la ieșirea circuitului de conversie A.C-D.C vom aveam în jur de 538.2 V, după calculul tensiunii după formula:

(28)

Unde VLL o să fie tensiune de intrare, în cazul nostru 400V. După calcul valorii medii, valoarea de ieșire este de 537.5 V, considerată destul de bună, aproape de valoarea ideală. Se poate observa în figura 33.

A screenshot of a computer

Description automatically generated

Figura 43. Valoarea medie de la ieșirea convertorului A.C-D.C

Producătorul declară că valoarea tensiunii pentru Porsche Taycan Turbo S este de aproximativ 800V ( 615-835V, cuprinzând atât modul normal de condus, cat și modul sport), valorile pe care le-am țintit după aplicarea D.C-D.C Boost-ului fiind astfel de 800V ieșire circuit, și o putere de 560kW, valori date de producător. În final, așa cum se observă în figura 34, am reușit să scoatem 809V la ieșire și 559.4kWz, ceea ce este o valoare foarte apropiată de ceea ce ne dorim și o valoare a tensiunii care se află în parametrii.

A diagram of a computer

Description automatically generated

Figura 44. Tensiunea de la ieșirea D.C-D.C Boost

A diagram of a power plant

Description automatically generated

Figura 45. Tensiune, Curent, S.O.C și Putere pentru baterie

În Figura 35 putem observa că tensiunea pe baterie se încadrează între parametrii pe care noi îi dorim. Bateria este încercată la 98.3 la sută iar produsul dintre curent si tensiune, fiind puterea este cea dorită și anume cea de 560kw, așa cum ne precizează producătorul.

Tot pentru partea de simulare trebuie să verificăm timpii de procesare, pentru acest aspect Simulink ne ajută cu un sistem de culori. Prin rularea comenzilor de rulare cu vederea culorilor, putem observa că o mare parte a sistemului rulează cu un timp de 5e-5, acestea fiind toate semnalele pentru curenți și tensiuni, reprezentate cu, culoarea roșie în imaginea de mai jos, valorile modificate cu ajutorul blocurilor de gain au culoarea verde, componentele rezultate în urma aplicării Reinforcement Learning și anume acțiunile au culoarea albastră și timpul aferent. Valorile constante o să fie roz iar blocurile care au ca semnale ce se transmit la diferite valori o să fie de culoarea galben.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 46. Culori pentru timpi

A diagram of a computer

AI-generated content may be incorrect.

Figura 47. Culori pentru timp circuit mare

A diagram of a machine

AI-generated content may be incorrect.

Figura 48. Culori pentru timp circuit mic

**6.2 Rezultate Semnale**

În acest sub-capitol, o să vedem semnalele din proiectul nostru, primul și cel de al doilea semnal prezentat o să fie cel realizat după conversia din A.C în D.C, respectiv semnalul trifazat de la intrarea circuitului nostru:

A screen shot of a graph

Description automatically generated

Figura 49. Semnal Trifazat

A screen shot of a graph

AI-generated content may be incorrect.

Figura 50. Semnal după conversie A.C-D.C

Mai departe o să vizualizăm semnalul procesat după aplicarea D.C-D.C Boost-ului, acesta trebuie să ajungă la o valoare de aproximativ 800 V ( Producătorul a declarat că valorile tensiunii pentru Porsche Taycan Turbo S se învârt între 615-835V, având modul normal de condus și modul sport, totuși declarat pe pagina este de 800V). Semnalul nostru ajunge la o valoare de 809V, care este acceptabilă și se află în parametrii menționați mai sus.

A screenshot of a computer

Description automatically generated

Figura 51. Semnal după conversie D.C-D.C Boost, la aproximativ 800V

În figura 39 putem vedea semnalele după activarea invertorului 3L. Pentru o viteza mecanica de 3000 de rad/s, care este o valoare mică, avem o tensiune de ieșire sinusoidală de 200V.

A graph of a graph

AI-generated content may be incorrect.

Figura 52. Semnalul ieșire inversor 3L cu rotații la 3000

O mașină performantă cum este Porche Taycan Turbo S poate funcționa și la 20.000 de rotații, în următorul exemplu o să stabilim un număr de 6000 de rotații, așa cum putea vedea în figura 40.

A graph of a wave

AI-generated content may be incorrect.

Figura 53. Semnalul ieșire inversor 3L cu rotații la 6000

În următoarea imagine putem vedea Viteza de feedback care o să fie folosită ulterior pentru partea de Reinforcement Learning, aceasta este dată de către viteza rotorului setata momentan la 3000 de rotații, iar următoarele sunt semnalele sinus și cosinus formate din unghiul rotorului.

A screen shot of a graph

AI-generated content may be incorrect.

Figura 54. Viteza de feedback, cosinus și sinus

Tot din componenta motorului electric trebuie să extragem curenții de pe stator și anume curenții a, b și c. Acestea pot să fie văzuți în imaginea 42. Avem curenții de pe statorul unei mașini electrice, care sunt curenți trifazici, variind sinusoidal în timp. Acești curenți sunt specifici mașinilor electrice trifazice, cum ar fi motoarele asincrone sau sincrone, și sunt necesari pentru generarea unui câmp magnetic rotativ în stator.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figura 55. Curenții de pe stator

Prima transformată realizată este transformata Clark, aceasta rezultă semnalele alfa și beta care pot să fie văzute în figura 43. Curenții au fost transformați folosind transformarea Clark. Transformarea Clarke este o metodă utilizată pentru a converti curenții trifazici în doi curenți ortogonali (i\_α și i\_β) într-un sistem de referință 2D. Această transformare este utilă pentru analizarea și controlul mașinilor electrice, deoarece simplifică modelul matematic transformându-l din sistemul trifazic original într-un sistem de coordonate ortogonale.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figura 56. Semnalele Alfa și Beta

Următoarea transformată este transformata Park, se pot observa semnalele dq în imaginea 44. Transformarea Park este o extensie a transformării Clarke, având ca scop să transforme curenții trifazici în două componente într-un sistem de coordonate rotative (d-q), care se rotește sincron cu câmpul magnetic al mașinii electrice. În sistemul dq, componentele „d” și „q” reprezintă direct și în cuadratură față de vectorul câmpului magnetic la care se raportează transformarea. Această metodă este foarte utilă în controlul vectorial al mașinilor electrice, deoarece permite munca cu valori constante sau aproape constante în loc de valori sinusoidale, simplificând analiza și controlul. Componenta d, este aliniată cu axa de referință roșie (directă) și, într-un motor sincron, corespunde magnetizării. Componenta q este ortogonală față de axa de referință (cuadratură), fiind responsabilă în principal de generarea cuplului motor.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figura 57. Semnalele d și q

Mai departe se pot observa semnalele ds și qs care se pot vedea în figura 45, acestea sunt rezultate după controlul dat de către Reinforcement Learning. În aceste imagini putem vedea aceasta curbă de învățare, la început avem un semnal mai haotic care indică învățarea realizată de către componenta de Reinforcement Learning. Putem observa că își îmbunătățește performantele în timp, ajungând stabil, ceea ce indică că agentul a ajuns la o strategie relativ optimă pentru problema noastră.

A black grid with yellow lines

AI-generated content may be incorrect.

Figura 58. Semnalele ds și qs

După controlul dat de către componenta de Reinforcement Learning, urmează să mergem înapoi prin transformata inversă Park și inversă Clark, semnalele pot să fie văzute în figurile ce urmează. Putem oberva ca ne intoarcem la semnalele sinusoidale care o sa fie folosite pentru controlul Invertorului.

Ca și rezultate putem vedea semnale mult mai curate, cu mult mai puțin zgomot și distorsiuni, acestea au nivele de amplitudine consistente și faze stabile. Acest lucru ne indică că blocul de Reinforcement Learning și-a efectuat treaba cu succes. Dacă sinusoidele sunt stabile și fără distorsiuni semnificative, indică faptul că Reinforcement Learning a determinat strategii eficiente de control, care conduc la operațiuni stabile ale motorului sau sistemului electric.

A screenshot of a computer screen

AI-generated content may be incorrect.

Figura 59. Transformata inversă Park

A screen shot of a screen

AI-generated content may be incorrect.

Figura 60. Transformata inversă Clark

A screenshot of a screen

AI-generated content may be incorrect.

Figura 61. Semnale pentru Invertor de control

**6.3 Rezultate Implementare CI/CD**

În acest sub-capitol, o să prezentam rolul pe care îl are implementarea C.I/C.D în proiectul nostru. După cum bine am discutat, implementarea C.I/C.D ajută dezvoltatorul Software să progreseze mai repede prin proiect. Acesta va putea să verifice funcționarea întregului proiect chiar după ce a încărcat în cload proiectul după ultima salvare. Prin folosirea comenzilor de bază git.

Scopul acestei implementări a fost de a rula în primul rând întreg proiectul principal.. În următoarele poze, o să vedem rezultate implementării C.I/C.D și timpul economisit de către dezvoltator.

În cele doua imagini de mai jos putem observa că rularea noastră a avut succes, acesta a compilat proiectul și nu a găsit probleme în compilarea lui. Întreg procesul poate să fie setat să ruleze, să simuleze sau să construiască întreg proiectul, în funcție de necesitatea fiecăruia.

În prima imagine se pot vedea și pașii pe care agentul i-a urmărit, și anume pregătirea rulării, preluarea repozitorului de pe pagina de GitHub, rularea comenzilor din Matlab, de precizat că în fișierul în care se face această rulare se folosesc comenzile pe care le folosim în terminalul din Matlab, pentru rulare proiect. Apoi salvăm rezultatele sub formă de artefacte într-un fișier și în final terminăm rularea. În ultima imagine putem vedea agentul activ.

A screenshot of a computer

Description automatically generated

Figura 62. Rulare terminată pentru compilare proiect principal

A screenshot of a computer

AI-generated content may be incorrect.

Figura 63. Terminal Agent

**7 Concluzii**

În concluzie, proiectul reprezintă o bază pentru înțelegerea modului de implementare și de funcționare a controlului curentului și încărcare pentru mașinile electrice. Mașina aleasă pentru acest proiect, nu a fost aleasă aleatoriu, am dorit să aduc în prin proiect o mașină iubită de pasionații auto, o mașină îndrăgita atât pentru performantele sale extraordinare cat și pentru aspectul ei deosebit. Astfel Porsche Taycan Turbo S, este modelul perfect care îmbină atât performanța cat și eleganța de care dă dovadă, aducându-i acest proiect dinamică.

Prin implementarea tuturor acestor tehnologii, am introdus astfel o mare parte din tehnologiile care sunt folosite în ziua de azi în industria Automotive. După cum putem vedea, lumea automotive este acaparată de către mașinile electrice, mașinile hibrid sau alte tipuri de configurații deoarece acestea au un avantaj asupra fiabilității, economiei și asupra poluării.

Putem spune că proiectul de față este o bază foarte bună pentru viitoarele dezvoltări, acesta conține toate punctele necesare pentru implementare mai mare și pentru un proiect mai amplu. Prin implementarea fiecărei componente în parte putem vedea foarte bine modul de funcționare și rolul fiecărei componente în schema mare. Acest proiect poate să fie folosit și pentru scopuri educaționale, astfel se poate oferi o mai bună înțelegere a componentelor electronice și a rolului lor în piață și în dezvoltare.

Proiectul m-a ajutat foarte mult să înțeleg personal mai bine cum funcționează o astfel de implementare, multa muncă s-a depus în înțelegerea funcționalității fiecărei componente și modulului de integrare a unor componente moderne precum Reinforcement Learning și Model Prediction Control, care nu sunt des utilizate în industrie în momentul de față, dar pe viitor cu siguranța o sa fie introduse. Nu toate componentele funcționează în mod ideal, acest proiect este la scară mică, chiar foarte mică față de un proiect realizat în cadrul unei corporații pentru doar una din multele componente alese de către mine. Dezvoltarea controlului unui invertor de tip 3L durează pană la doi, trei ani de implementare și de testare pană la realizarea unui demo, de către un grup de ingineri software, hardware, de sistem și mecanici. Ulterior proiectul demo este dus mai departe către echipa de business care realizează produsul final și apoi este introdus pe piață.

În final putem spune că proiectul a fost un succes, componentele noi introduse funcționează în conformitate, s-au adus și idei de viitoare dezvoltări, captează toate punctele principale de care avem nevoie pentru a înțelege această arhitectura și funcționează în parametrii normali dați de către producător pentru modelul de mașina ales. Valorile au fost formate pentru modelul de top a celor de la grupul Porsche. Dea lungul dezvoltării proiectului am dat de diferite probleme, de funcționare, de conectare, valori neconforme și alte probleme de compilare specifice acestui tip de proiect, dar în final proiectul a fost dus la bun sfârșit.

**8 Bibliografie**

[1] [What Is Machine Learning (ML)? | IBM](https://www.ibm.com/topics/machine-learning)

[2] [Taycan, Taycan 4S, Turbo and Turbo S – The battery - Porsche Taycan](https://media.porsche.com/mediakit/taycan/en/porsche-taycan/die-batterie)

[3] [Porsche Taycan | Porsche Eastern Europe](https://www.porsche.com/central-eastern-europe/en/models/taycan/taycan-models/taycan/)

[4] [Materii prime pentru bateriile mașinilor electrice: care sunt, de unde provin și cum reducem dependența de ele - Autocritica](https://www.autocritica.ro/green-zone/materii-prime-pentru-bateriile-masinilor-electrice-care-sunt-de-unde-provin-si-cum-reducem-dependenta-de-ele/)

[5] [Reinforcement learning - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-reinforcement-learning/)

[6] [Boost converter - Wikipedia](https://en.wikipedia.org/wiki/Boost_converter)

[7] [AC-DC Converters including Buck, Boost and Flyback](https://www.electronics-tutorials.ws/connectivity/ac-dc-converters.html)

[8] [AC to DC Converters: Features, Design & Applications](https://how2electronics.com/ac-to-dc-converters-features-design-applications/)

[9] How AC To DC Converter Works || How Inverter Works || How Rectifier Works || Part – 1

[10] [Reinforcement Learning - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-reinforcement-learning/)

[11] [Reinforcement learning - Wikipedia](https://en.wikipedia.org/wiki/Reinforcement_learning)

[12] [PMSM Field-Oriented Control](https://www.mathworks.com/help/sps/ref/pmsmfieldorientedcontrol.html)

[13] [Field-Oriented Control (FOC)](https://www.mathworks.com/help/mcb/gs/implement-motor-speed-control-by-using-field-oriented-control-foc.html)

[14] [Permanent Magnet Synchronous Motors (PMSM)](https://www.mathworks.com/help/mcb/pmsm.html)

[15] [Get Started with Motor Control Blockset - Help Center Answers - MATLAB & Simulink](https://www.mathworks.com/support/search.html?fq%5B%5D=asset_type_name:answer&fq%5B%5D=category:mcb/getting-started-with-mcb&page=1&s_tid=CRUX_topnav)

[16] <https://www.mathworks.com/matlabcentral/fileexchange/159051-simulation-of-ac-dc-converter-fed-dc-motor-drive?s_tid=srchtitle_support_results_5_AC-DC>

[17] [Reinforcement Learning for Developing Field-Oriented Control - Help Center Videos - MATLAB & Simulink](https://www.mathworks.com/support/search.html/videos/reinforcement-learning-for-field-oriented-control-of-a-permanent-magnet-synchronous-motor-1587727861081.html?c%5B%5D=support&fq%5B%5D=asset_type_name:video&fq%5B%5D=category:simulink/index&page=1)

**9 Anexe**

**10 CV**