



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
ENGENHARIA ELÉTRICA - ELETRÔNICA

Hardware e Software para monitoramento e operação de BESS

Gabriel de Aquino Figueira

MANAUS-AM

2023

Gabriel de Aquino Figueira

Hardware e Software para monitoramento e operação de BESS

Monografia apresentada à Coordenação do
Curso de Engenharia Elétrica - Eletrônica
da Universidade Federal do Amazonas, como
parte dos requisitos necessários à obtenção
do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Renan Landau Paiva de Medeiros

MANAUS-AM

2023

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

F475h Figueira, Gabriel de Aquino
Hardware e Software para monitoramento e operação de BESS /
Gabriel de Aquino Figueira . 2023
54 f.: il. color; 31 cm.

Orientador: Renan Landau Paiva de Medeiros
TCC de Graduação (Engenharia Elétrica - Eletrônica) -
Universidade Federal do Amazonas.

1. Bateria. 2. Lítio-íon. 3. Contage de Coulomb. 4. Curva de carga.
I. Medeiros, Renan Landau Paiva de. II. Universidade Federal do
Amazonas III. Título

Gabriel de Aquino Figueira

HARDWARE E SOFTWARE PARA MONITORAMENTO E OPERAÇÃO DE BESS

Monografia apresentada à
Coordenação do Curso de Engenharia
Elétrica-Eletrônica da Universidade
Federal do Amazonas, como parte dos
requisitos necessários à obtenção do
título de Engenheiro Eletricista.

Aprovado em 28 de fevereiro de 2023.

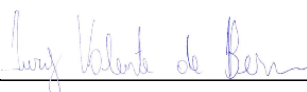
BANCA EXAMINADORA



Prof. Dr. Renan Landau Paiva de Medeiros
Universidade Federal do Amazonas – UFAM



Prof. Dr. Florindo Antonio de Carvalho Ayres Júnior
Universidade Federal do Amazonas – UFAM



Prof. Dr. Iury Valente de Bessa

Universidade Federal do Amazonas – UFAM

Dedico este trabalho aos meus pais, Lis e Marcelo e irmãos, Gabrielle e Moisés por me darem amor e suporte durante toda a minha vida.

Resumo

A perspectiva de crescimento da demanda por sistemas fotovoltaicos e veículos elétricos nos últimos anos criou a necessidade de melhoria em sistemas de gerenciamento de baterias para aplicações comerciais e industriais. O BESS é um sistema que fornece uma solução que possibilita o controle e o uso inteligente dos recursos energéticos. Neste trabalho de conclusão de curso é desenvolvido um sistema para monitorar a carga e a descarga de uma bateria de lítio-íon. É projetado um circuito capaz de carregar e descarregar a bateria, sendo que a carga é realizada utilizando o método CCCV. Um microcontrolador e alguns sensores foram utilizados para capturar a tensão e corrente da bateria, e realizar a estimativa de SOC através da contagem de Coulomb. Os dados destas medições são enviados via *websocket* para um servidor, o qual armazena as medições em um banco de dados relacional. Todos os dados podem ser visualizados em gráficos atualizados em tempo real em um *frontend* que é executado em um navegador web. Através dos experimentos, foi possível validar o funcionamento do sistema através da observação dos gráficos de carga e descarga da bateria.

Palavras-chave: bateria de lítio-íon, contagem de Coulomb, carga CCCV, curva de carga e descarga, *websocket*.

Abstract

The prospect of growing demand for photovoltaic systems and electric vehicles in recent years has created the need for improvement in battery management systems for commercial and industrial applications. BESS is a system that provides a solution that enables the control and intelligent use of energy resources. In this course conclusion work, a system is developed to monitor the charge and discharge of a lithium-ion battery. A circuit capable of charging and discharging a battery is designed, and the charge is performed using the CCCV method. A microcontroller and some sensors were used to capture the voltage and current of the battery, and perform the SOC estimation through the Coulomb count. Data from these measurements are sent via websocket to a server, which stores the measurements in a relational database. All data can be visualized in graphs updated in real time in a frontend that runs in a web browser. Through the experiments, it was possible to validate the functioning of the system through the observation of the graphs of charge and discharge of the battery.

Keywords: lithium-ion battery, Coulomb count, CCCV charging, charge and discharge curve, websocket.

Lista de Figuras

2.1	Retenção de carga de uma bateria de níquel-metal hidreto em várias temperaturas. Adaptado de (Linden and Reddy, 2002)	5
2.2	Curvas de descarga de uma bateria de níquel-metal hidreto a 20°C. (a) - taxa 0.2C. (b) - taxa 1C. (c) - taxa 2C. (d) - taxa 3C. Adaptado de (Linden and Reddy, 2002)	6
2.3	Curvas de descarga de uma bateria de lítio-íon comercial tipo 18650 em regime de corrente constante (CC). Adaptado de (Linden and Reddy, 2002)	7
2.4	Topologia de um BESS. Adaptado de (Hesse et al., 2017)	8
2.5	Topologia de um BESS e BMS. Adaptado de (Lawder et al., 2014)	9
2.6	(a) ECM com circuito RC de primeira ordem. (b) ECM com circuito RC de segunda ordem. Adaptado de (Zhang et al., 2017)	10
2.7	Visão geral da simulação realizada no Simulink. Elaborado pelo autor . . .	11
2.8	Estimação de SOC com a contagem de Coulomb no Simulink. Elaborado pelo autor	11
2.9	Modelo elétrico da bateria em blocos do Simulink. Elaborado pelo autor .	12
2.10	Curva de carga e SOC mostradas no dashboard. Elaborado pelo autor . . .	13
3.1	Curvas de carregamento de uma bateria de lítio-íon. Adaptado de (Jiang and Zhang, 2015)	15
3.2	Curvas de descarga de uma bateria de lítio-íon após vários ciclos. Adaptado de (Jiang and Zhang, 2015)	16
3.3	Curvas de OCV para diferentes valores de corrente na bateria. Adaptado de (Rivera-Barrera et al., 2017)	17
3.4	Modelo de predição de SOC baseado em uma rede neural do tipo BP. Adaptado de (Yan and Wang, 2017)	19

4.1	Visão geral do sistema. Elaborado pelo autor.	23
4.2	Circuito elétrico desenvolvido. Elaborado pelo autor.	24
4.3	Diagrama de estados da estimação de SOC. Elaborado pelo autor.	26
4.4	Curvas de SOC-OCV de acordo com temperatura. Adaptado de Zhang et al. (2018).	28
4.5	Aproximação linear para curva de SOC-OCV. Adaptado de Baccouche et al. (2017).	29
4.6	Faixa de SOC ideal para operação de uma bateria. Adaptado de Aoxia and K. (2016)	30
4.7	Fluxo de dados do sistema. Elaborado pelo autor.	31
4.8	Tabela do banco de dados visualizada no MySQL Workbench. Elaborado pelo autor.	31
4.9	Frontend do projeto rodando em um navegador web. Elaborado pelo autor.	33
5.1	Circuito montado em protoboard. Elaborado pelo autor.	34
5.2	Tensão durante o teste de descarga da bateria. Elaborado pelo autor. . . .	35
5.3	Corrente o teste de descarga da bateria. Elaborado pelo autor.	35
5.4	SOC o teste de descarga da bateria. Elaborado pelo autor.	36
5.5	Gráfico da tensão na bateria durante o carregamento CC-CV. Elaborado pelo autor.	37
5.6	Gráfico da corrente na bateria durante o carregamento CC-CV. Elaborado pelo autor.	37
5.7	Gráfico do SOC durante o carregamento CC-CV. Elaborado pelo autor. . .	38

Lista de Tabelas

4.1	Relação de SOC-OCV a 25°C. Adaptado de Baccouche et al. (2017)	29
-----	--	----

Lista de Abreviaturas e Siglas

BESS	Battery Energy Storage System
BMS	Battery Management System
CC	Coulomb Counting
CCCV	Constant Current Constant Voltage
CV	Constant Voltage
EKF	Extend Kalman filter
OCV	Open Circuit Voltage
SOC	State of Charge
SOH	State of Health

Sumário

1	Introdução	1
1.1	Objetivo Geral	1
1.2	Objetivos Específicos	1
1.3	Metodologia	2
1.4	Organização do trabalho	2
2	Conceitos básicos sobre sistemas de armazenamento de energia elétrica	3
2.1	Dispositivos	3
2.1.1	Baterias	3
2.1.1.1	Classificação de baterias entre primárias e secundárias . .	4
2.1.1.2	Classificação de baterias secundárias acordo com o material	4
2.2	Topologias sobre sistemas de gerenciamento de baterias	7
2.3	Modelagem	9
2.3.1	Modelo no Simulink	10
3	Estratégias clássicas para monitoramento e supervisão de sistemas de armazenamento de energia elétrica	14
3.1	Estado de Carga (SOC)	14
3.1.1	Tensão de circuito aberto (OCV)	16
3.1.2	Contagem de Coulomb	17
3.1.3	Redes Neurais	18
3.1.4	Filtro de Kalman Estendido	19
3.2	Estado de Saúde (SOH)	20
3.3	Carregamento da bateria	21

4	Desenvolvimento de um sistema de monitoramento e supervisão para sistemas de armazenamento de energia elétrica	22
4.1	Visão Geral	22
4.2	Hardware	23
4.2.1	Multiplexador analógico	24
4.2.2	Medição de tensão	25
4.2.3	Medição de corrente	25
4.2.4	Estimação de SOC	26
4.2.5	Modos de operação do circuito	26
4.3	Software	30
4.3.1	Backend	30
4.3.2	Frontend	32
5	Análise de Resultados	34
5.1	Teste de descarga	35
5.2	Teste de carga	36
6	Conclusão	39
6.1	Trabalhos futuros	40
	Referências Bibliográficas	41
A	Códigos utilizados	44

Capítulo 1

Introdução

Neste trabalho será apresentado um circuito elétrico para realizar um experimento de carga (CCCV) e descarga de uma bateria de lítio-íon. Será embarcado um algoritmo em um microcontrolador que realiza a estimação de SOC através da contagem de Coulomb, e também a operação do circuito para decidir se ele trabalha em modo de carga ou descarga. Além disso, será mostrado o desenvolvimento de um *backend* capaz de receber, armazenar em banco de dados, e enviar informações através de *websocket*. Há também um *frontend* que se comunica com o *backend* e exibe em tempo real os dados obtidos no circuito elétrico. Serão mostrados experimentos de carga e descarga da bateria para demonstrar o funcionamento do sistema.

1.1 Objetivo Geral

Desenvolver um sistema capaz de monitorar e supervisionar o processo de carga e descarga de uma bateria de lítio-íon.

1.2 Objetivos Específicos

- Implementar um circuito para a realização de experimentos de carga e descarga em uma bateria.
- Aplicar a contagem de Coulomb para a estimação de SOC.
- Desenvolver uma aplicação web capaz de exibir gráficos em tempo real com dados extraídos de uma bateria.

- Desenvolver um servidor para armazenar e fornecer dados de medições da bateria.

1.3 Metodologia

Este trabalho consiste no desenvolvimento de um sistema que possui três frentes: o circuito elétrico, *frontend*, *backend*. Para validação do sistema, serão realizados testes práticos de carga e descarga de uma bateria. Os resultados serão interpretados através de comparações com outros trabalhos e com modelos matemáticos simulados.

1.4 Organização do trabalho

Este trabalho está organizado da seguinte maneira:

- No capítulo 2 será feita uma introdução teórica que apresenta conceitos sobre sistemas de armazenamento de energia elétrica, bem como alguns tipos de baterias e uma modelagem matemática de uma bateria de lítio-íon;
- No capítulo 3 será feita uma continuação da fundamentação teórica iniciada no capítulo 2, porém, com ênfase em técnicas para o monitoramento e supervisão de baterias;
- No capítulo 4 serão apresentados os detalhes da implementação do sistema proposto;
- No capítulo 5 serão mostrados os resultados e análise dos teste práticos;
- No capítulo 6 serão feitas as conclusões e sugestões para futuros trabalhos.

Capítulo 2

Conceitos básicos sobre sistemas de armazenamento de energia elétrica

Este capítulo apresenta brevemente conceitos acerca de dispositivos fundamentais para um sistema de armazenamento de energia. Serão mostradas uma breve revisão sobre os tipos de bateria mais utilizados, uma base teórica para a modelagem de baterias de Lítio-íon, e uma visão geral do funcionamento de um BESS.

2.1 Dispositivos

2.1.1 Baterias

Baterias são dispositivos que convertem energia química contida em seus materiais em energia elétrica por meio de uma reação eletroquímica. (Linden and Reddy, 2002) Uma bateria é formada por uma ou mais células eletroquímicas, conectadas eletricamente em série ou paralelo. A célula é caracterizada por ser a unidade eletroquímica mais básica em um sistema. (Linden and Reddy, 2002)

Em aplicações de BESS, há sete características principais para decidir qual a composição da bateria mais adequada, são eles: longevidade, segurança, densidade de potência (W/kg), preço, densidade de energia (Wh/kg), capacidade de energia (Ah), tensão por célula. (Aoxia and K., 2016)

2.1.1.1 Classificação de baterias entre primárias e secundárias

2.1.1.1.1 Células e Baterias Primárias São células e baterias que não podem ser recarregadas de forma fácil ou efetiva. Este tipo de bateria tem como vantagem um custo baixo, boa vida útil, alta densidade de energia, taxa de descarga moderada, e pouca ou nenhuma manutenção necessária. (Linden and Reddy, 2002)

2.1.1.1.2 Células e Baterias Secundárias São aquelas capazes de serem recarregadas através da passagem de corrente elétrica pelo sentido oposto ao sentido da corrente de descarga. (Linden and Reddy, 2002) As baterias secundárias possuem como característica a possibilidade de serem recarregadas, alta densidade de energia (embora geralmente menor que nas baterias primárias), alta taxa de descarga e boa performance em baixa temperatura. (Linden and Reddy, 2002)

2.1.1.2 Classificação de baterias secundárias acordo com o material

2.1.1.2.1 Baterias de Chumbo-ácido São baterias secundárias constituídas majoritariamente por chumbo e componentes de chumbo, e são amplamente utilizadas em sistemas de partida de veículos. (Linden and Reddy, 2002) A tensão nominal de uma célula de chumbo-ácido é de 2V. A tensão de circuito aberto depende diretamente da concentração de eletrólito na célula, variando de 2,125V a 2,05V. (Linden and Reddy, 2002) A taxa de autodescarga é baixa, ficando geralmente em menos de 0,5% ao dia em uma temperatura de 25°C. (Linden and Reddy, 2002)

2.1.1.2.2 Bateria de Níquel-metal Hidreto As baterias de níquel-metal hidreto (NiMH) são baterias recarregáveis contruídas geralmente em configurações cilíndricas, de botão, ou prismáticas. (Linden and Reddy, 2002) As principais vantagens de uma célula deste tipo são o seu baixo custo e a sua segurança. A alta taxa de autodescarga é uma de suas principais desvantagens (Aoxia and K., 2016). Observa-se através da Figura 2.1 que a taxa de autodescarga é acentuada com o aumento da temperatura.

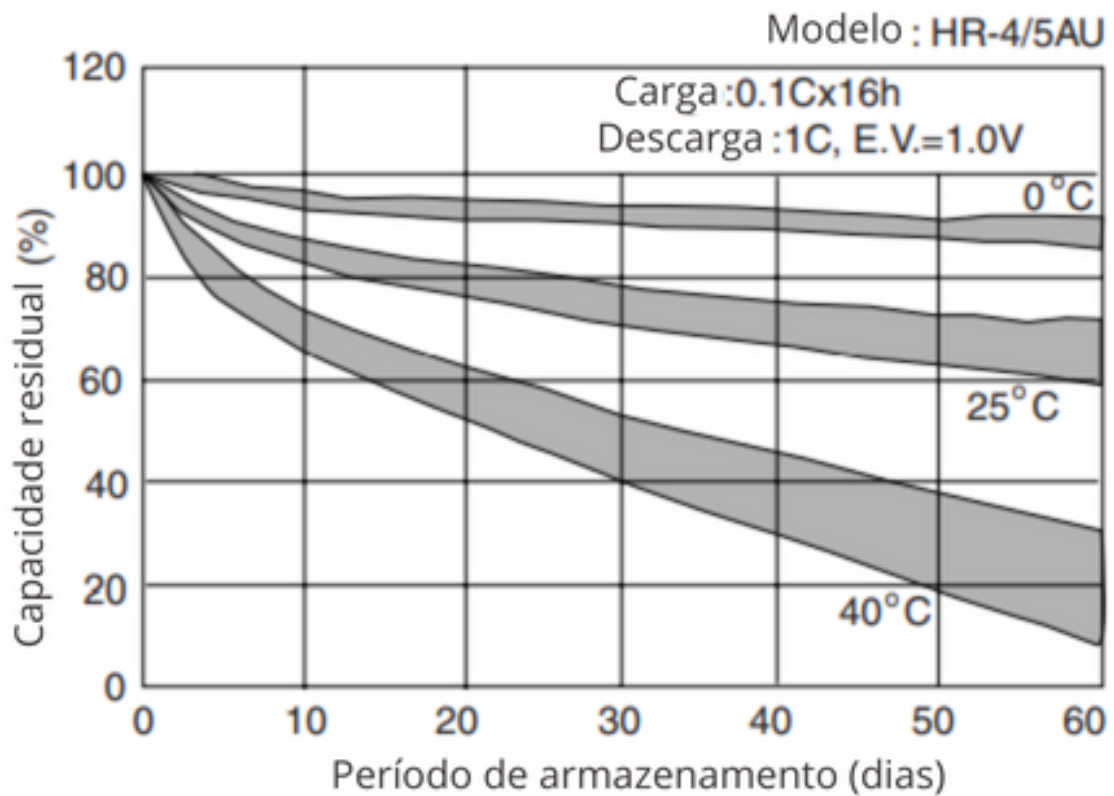


Figura 2.1: Retenção de carga de uma bateria de níquel-metal hidreto em várias temperaturas. Adaptado de (Linden and Reddy, 2002)

Outra desvantagem é a baixa tensão de cada célula. (Aoxia and K., 2016) Segundo (Linden and Reddy, 2002), a tensão nominal de uma bateria deste tipo é de 1,2V, a tensão de circuito aberto com a bateria completamente carregada varia entre 1,35V e 1,25V, e a tensão em que se considera a bateria totalmente descarregada é 1,0V. Através da Figura 2.2 observa-se a característica da curva de descarga desta bateria.

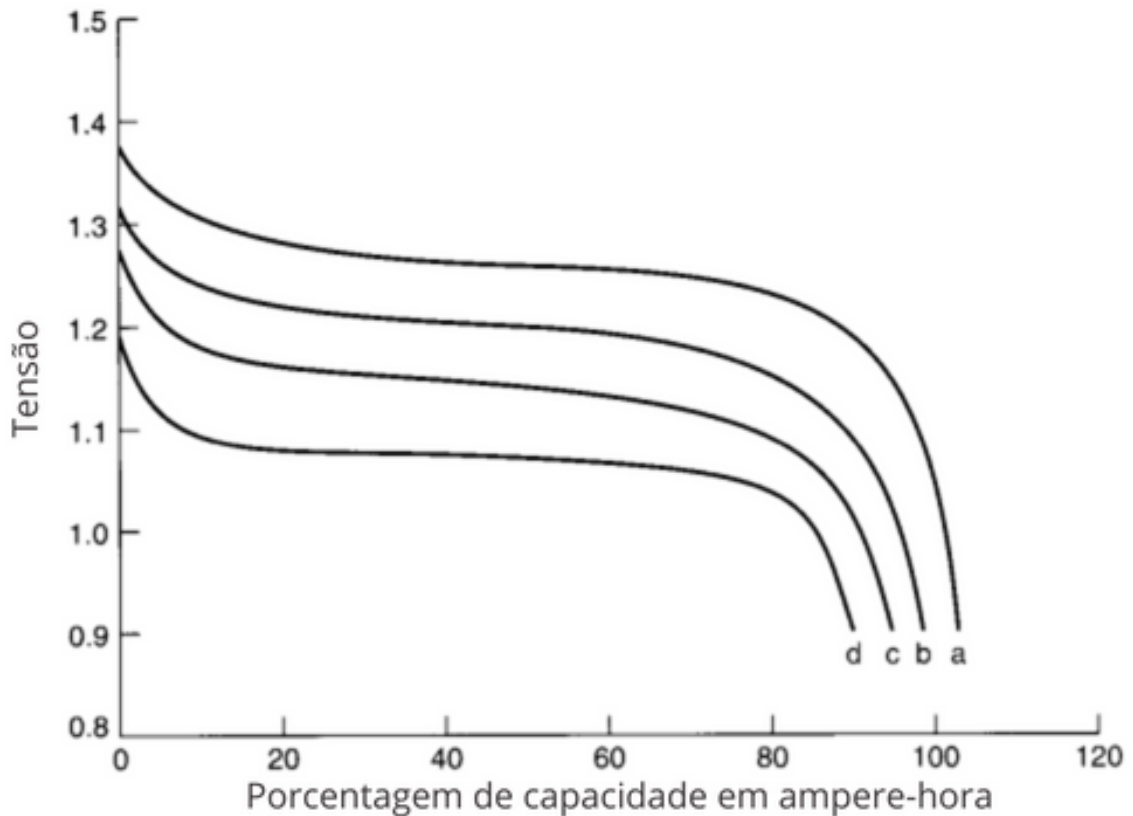


Figura 2.2: Curvas de descarga de uma bateria de níquel-metal hidreto a 20°C. (a) - taxa 0.2C. (b) - taxa 1C. (c) - taxa 2C. (d) - taxa 3C. Adaptado de (Linden and Reddy, 2002)

Este tipo de bateria também sofre com efeito de memória, isto é, pode ocorrer uma queda na tensão e uma perda na capacidade máxima quando são realizadas cargas e descargas parciais em vez de ciclos completos. (Linden and Reddy, 2002)

2.1.1.2.3 Bateria de Lítio-íon Amplamente utilizadas em BESS e em diversas outras aplicações, as baterias de lítio-íon (*Li-ion*) são feitas com compostos de lítio que servem de material positivo e negativo.

As células de lítio-íon possuem uma tensão nominal de 3,6V e geralmente são escolhidas para aplicações nas quais a densidade de energia e o peso da bateria são características primordiais. (Aoxia and K., 2016)

Entre as vantagens deste tipo de bateria estão a sua alta densidade de energia, baixa taxa de autodescarga (2% a 8% por mês), alto ciclo de vida (maior que 100 ciclos) e uma ampla faixa de temperatura de operação (carga de 20°C a 60°C e descarga 40°C a 65°C). As células de lítio-íon usualmente operam entre 2,5V e 4,2V. (Linden and Reddy, 2002) Uma desvantagem destas baterias é que, diferente de alguns outros tipos de baterias como as células aquosas, elas não possuem mecanismos químicos para prevenir sobrecarga. Por

isso, estas células são dependentes de circuitos elétricos e peças mecânicas para prevenir problemas como sobrecarga, sobredescarga e superaquecimento. Além disso, estas baterias perdem capacidade permanentemente quando operam em temperaturas maiores que 65°C. (Linden and Reddy, 2002)

A curva de descarga deste tipo de bateria é mostrada na Figura 2.3.

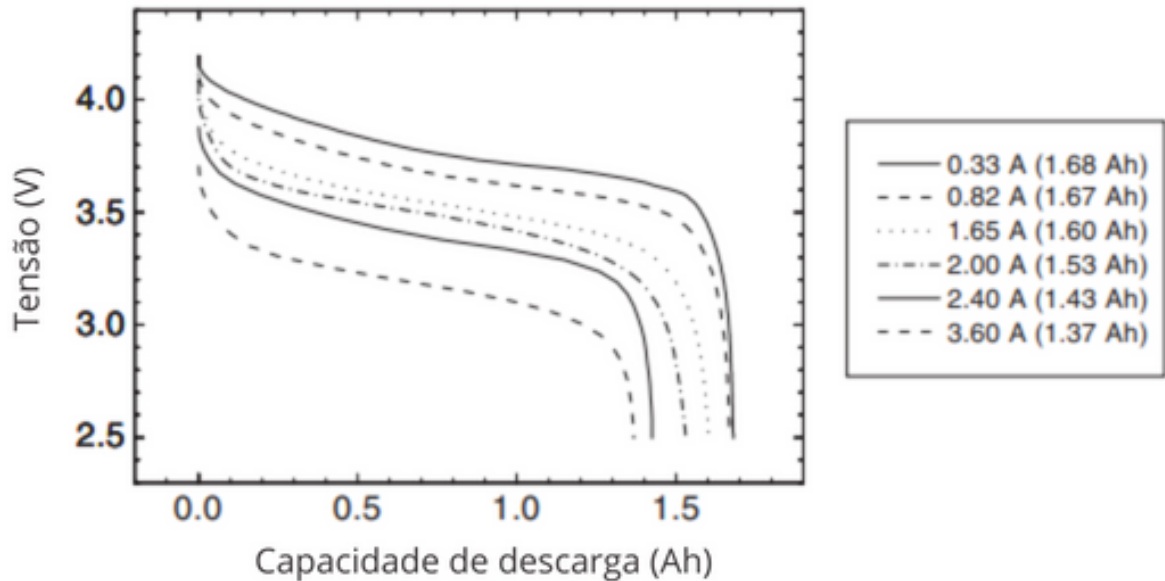


Figura 2.3: Curvas de descarga de uma bateria de lítio-íon comercial tipo 18650 em regime de corrente constante (CC). Adaptado de (Linden and Reddy, 2002)

2.2 Topologias sobre sistemas de gerenciamento de baterias

A Figura 2.4 mostra um esquemático de um BESS com os componentes agrupados por função. Os blocos dentro do conjunto Operação do Sistema (*System Operation*) representam os sistemas de supervisão e aquisição de dados e de controle de temperatura; dentro de Bateria (*Battery*) há módulos de controle e monitoramento que atuam sobre múltiplas baterias ligadas em conjunto; e o bloco Eletrônica de Potência (*Power Electronics*) são circuitos para realização da conexão do sistema com uma rede de energia elétrica.

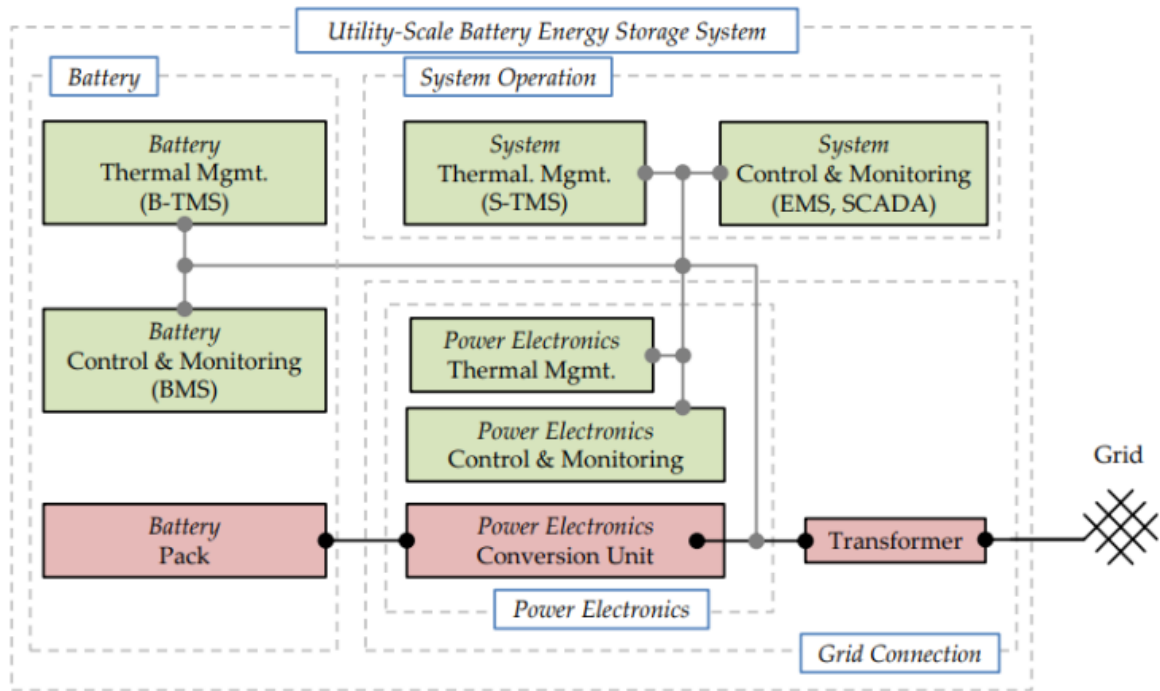


Figura 2.4: Topologia de um BESS. Adaptado de (Hesse et al., 2017)

Segundo Lawder et al. (2014), o BESS é um componente fundamental para melhorar a eficiência de redes elétrica. O sistema de gerenciamento de baterias (BMS, ou, em inglês, *battery management system*) é uma parte integral de um BESS e possui a função de monitorar, supervisionar e dar segurança às baterias, medindo parâmetros como o estado de carga, estado de saúde, temperatura da bateria, entre outros. (Lawder et al., 2014)

A Figura 2.5 mostra a implementação da relação entre o BESS e o BMS. Lawder et al. (2014) diz que o BESS da Figura 2.5 pode possuir várias objetivos, como balanceamento de carga em baterias, controle de temperatura de células, monitoramento de tensão e corrente, e vários outros objetivos que dependem da aplicação prática do sistema.

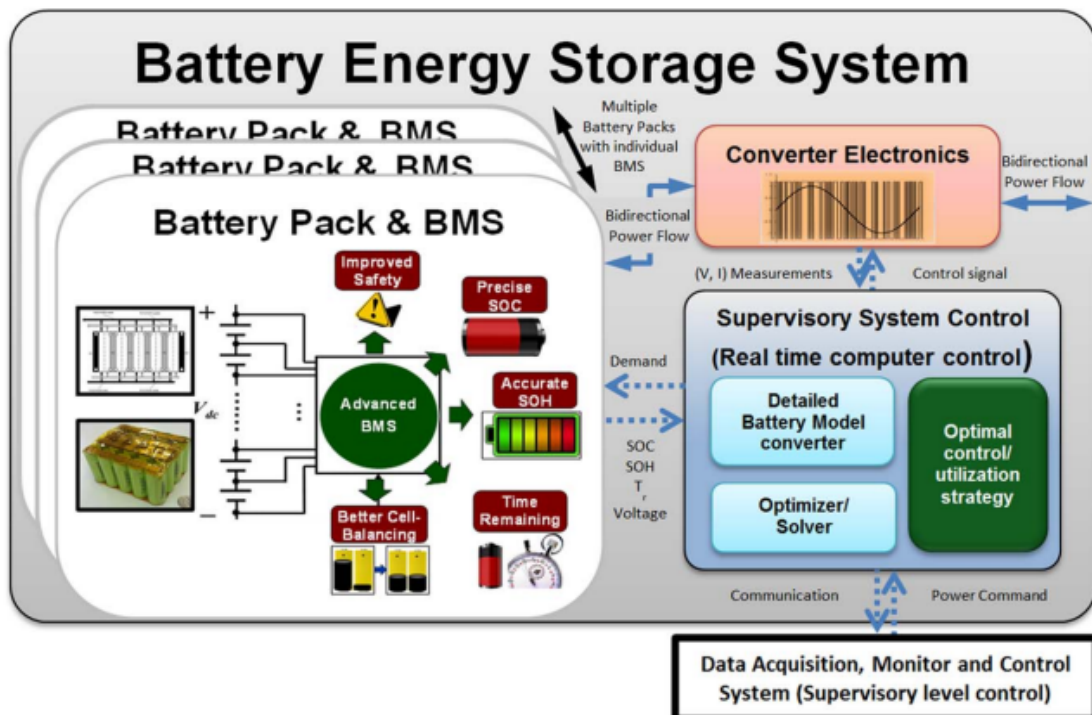


Figura 2.5: Topologia de um BESS e BMS. Adaptado de (Lawder et al., 2014)

2.3 Modelagem

O modelo matemático de uma bateria é útil para prever o comportamento dela. De forma generalizada, os modelos podem ser divididos entre eletroquímicos e elétricos. Os modelos eletroquímicos são tipicamente utilizados para compreender os processos químicos dentro da bateria, também servem de suporte para a concepção e manufatura de novos modelos de baterias. Já os modelos elétricos (ECM, *electrical circuit model* em inglês) descrevem comportamentos como a curva carga e descarga, e são amplamente utilizados para estimação de parâmetros como SOC e SOH. (Zhang et al., 2017)

A Figura 2.6 mostra circuitos que descrevem o ECM de uma bateria de lítio-íon. Voc é a OCV e varia não linearmente com o SOC, R_0 é a resistência ôhmica da bateria, R_1 é a resistência de polarização, C_1 é a capacitância de polarização. Dos parâmetros exclusivos do sistema de segunda ordem, C_2 e R_2 são a resposta não linear da polarização da bateria. A resposta de polarização da bateria é a resposta transiente quando os terminais sofrem uma mudança abrupta na corrente. (Zhang et al., 2017) Este circuito é conhecido como modelo de bateria de Thévenin.

Para este trabalho foi escolhido o desenvolvimento considerando o circuito RC de segunda ordem, por causa de sua simplicidade e acurácia em relação ao de primeira ordem.

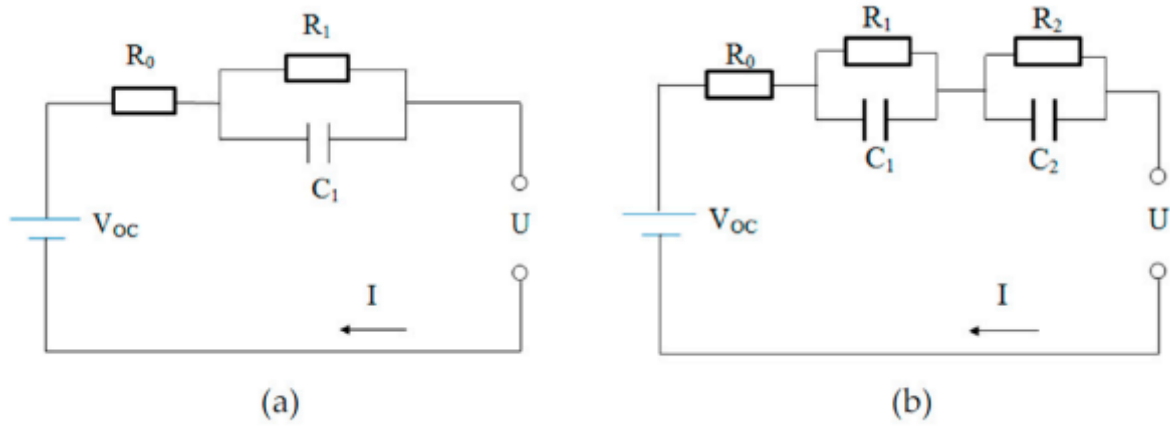


Figura 2.6: (a) ECM com circuito RC de primeira ordem. (b) ECM com circuito RC de segunda ordem. Adaptado de (Zhang et al., 2017)

Os estudos de Cacciato et al. (2015) e Zhang et al. (2017) desenvolvem os cálculos ECM através da lei de Kirchhoff. Com base nisso, observa-se que a Equação 2.1 mostra a somatória das tensões no circuito.

$$U(t) = VOC - IR_0 - IR_1 e^{-\frac{t}{R_1 C_1}} - IR_2 e^{-\frac{t}{R_2 C_2}} \quad (2.1)$$

Os valores dos elementos do circuito e da curva OCV-SOC foram retirados de Zhang et al. (2017), que faz experimentos de *Hybrid Pulse Power Characterization* (HPPC) para os estimar. Como os valores dos elementos possuem não linearidades que dependem do SOC, os dados foram inseridos em lookup tables dentro do Simulink para realizar as simulações do modelo.

2.3.1 Modelo no Simulink

A Figura 2.7 mostra os principais blocos desenvolvidos para a simulação. O bloco "Switch" possui uma lógica para alternar entre a simulação de carga e descarga facilmente. O bloco "SoC" implementa a contagem de Coulomb para estimação de SOC. O bloco "Cálculo da OCV" é feito através da Equação 2.1, e a relação não linear do SOC com VOC é colocada na *lookup table* que tem como entrada o resultado da contagem de Coulomb.

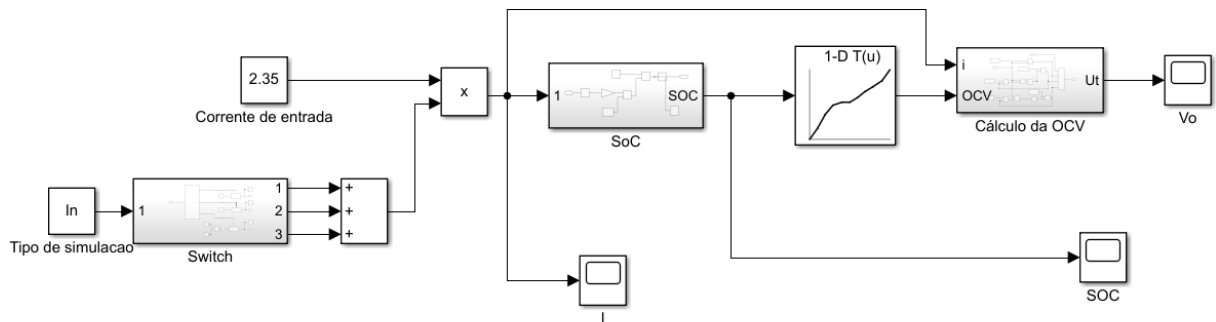


Figura 2.7: Visão geral da simulação realizada no Simulink. Elaborado pelo autor

A Figura 2.8 mostra a implementação da contagem de Coulomb, que será descrita no capítulo seguinte.

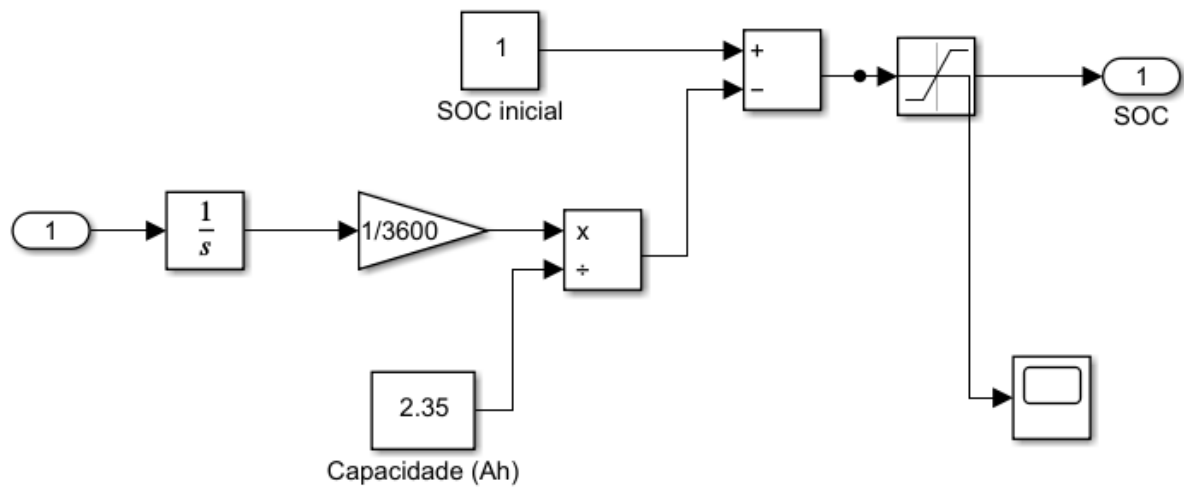


Figura 2.8: Estimação de SOC com a contagem de Coulomb no Simulink. Elaborado pelo autor

A Figura 2.9 tem o cálculo da malha descrita pela 2.1. Este bloco soma a contribuição de cada elemento do circuito.

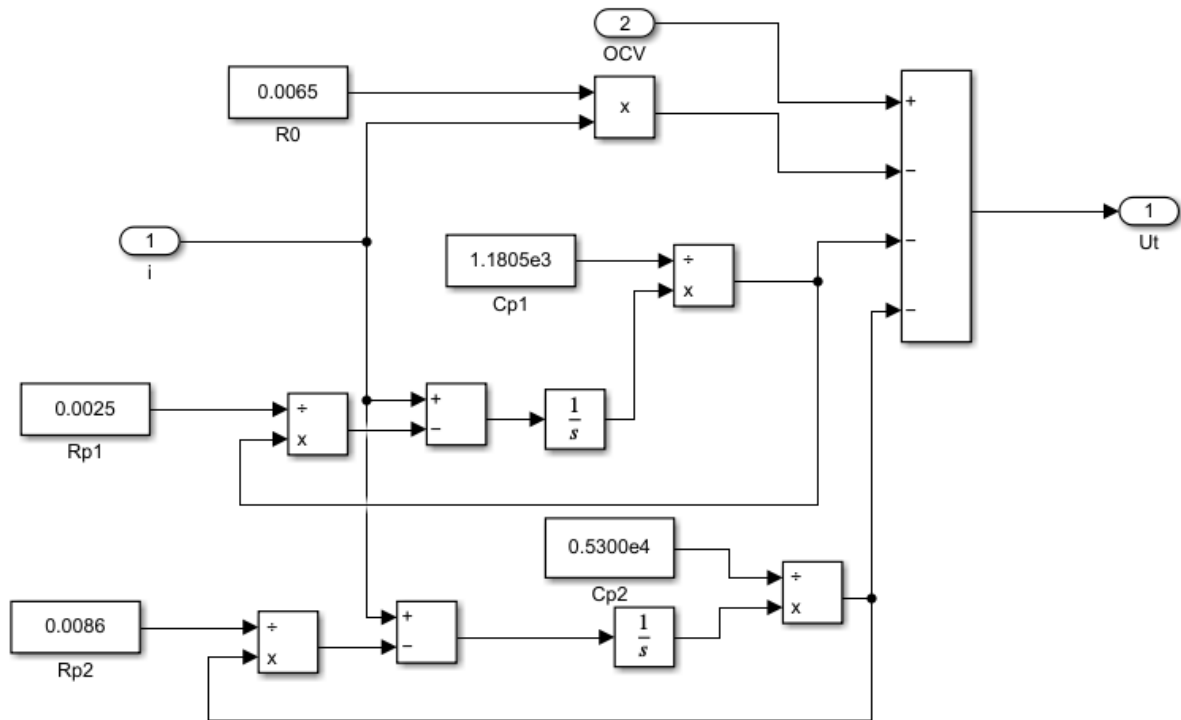


Figura 2.9: Modelo elétrico da bateria em blocos do Simulink. Elaborado pelo autor

O resultado da simulação é mostrado na Figura 2.10 através de uma interface criada com o bloco "dashboard".

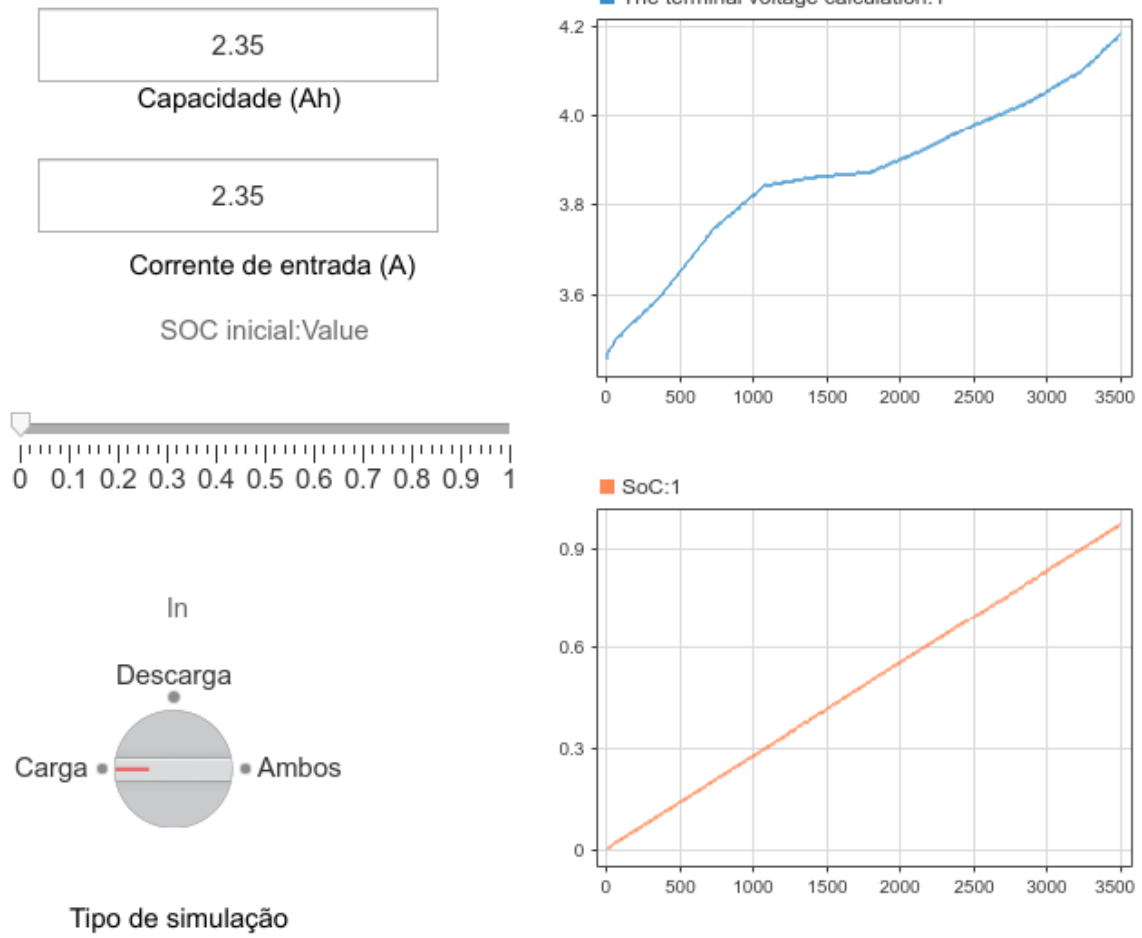


Figura 2.10: Curva de carga e SOC mostradas no dashboard. Elaborado pelo autor

Capítulo 3

Estratégias clássicas para monitoramento e supervisão de sistemas de armazenamento de energia elétrica

Neste capítulo serão apresentados conceitos e técnicas para o monitoramento de características que são relevantes durante o uso de um sistema de baterias. O objetivo é realizar uma revisão teórica sobre as técnicas mais utilizadas para medição do estado de carga e estado de saúde de baterias. Também serão apresentados de forma breve alguns métodos de carregamento de bateria.

3.1 Estado de Carga (SOC)

O SOC é principal propriedade útil para descrever quanta energia uma bateria ainda possui. (Park et al., 2020) A sobrecarga ou a sobredescarga resultam em danos permanentes em uma bateria, diminuindo a sua vida útil. Este problema pode ser evitado se a quantidade de energia remanescente em uma célula for conhecida. Por isso, um parâmetro importante a ser estimado por um sistema de baterias é o estado de carga (state of charge, ou SOC). (Saji et al., 2019) Uma bateria em sua capacidade máxima de carga possui 100% de SOC, e 0% de SOC em sua capacidade mínima. (Wu et al., 2016)

De acordo com a Equação 3.1 , o SOC se refere a porcentagem de energia que uma

bateria possui em relação a sua capacidade máxima.

$$SOC = \frac{Q_{rem}}{Q_{max}} 100\% \quad (3.1)$$

Em 3.1, Q_{rem} é a energia remanescente e Q_{max} é a capacidade máxima da bateria.

Pode se observar na Figura 3.1 a curva de carregamento de uma bateria de lítio-íon. As características da curva mais importantes a se ressaltar são as dinâmicas transientes que acontecem quando o SOC oscila entre valores muito baixos(em torno de 0% a 20%) ou muito altos(em torno de 80% a 100%), e a aproximação linear para valores intermediários(em torno de 20% a 80%).

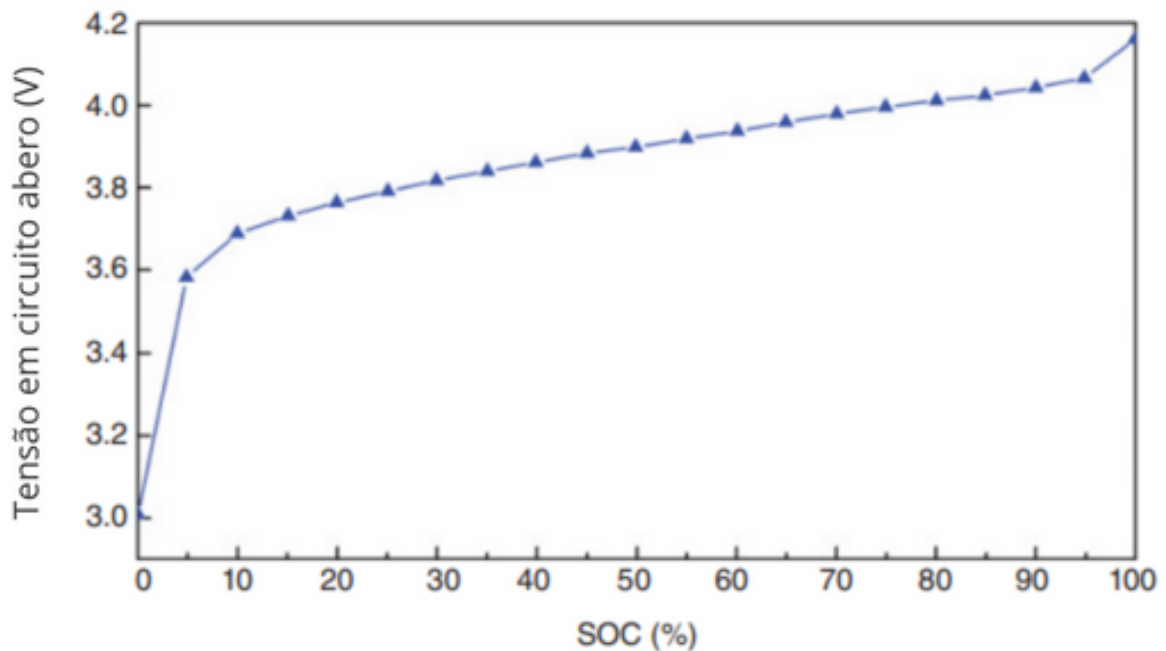


Figura 3.1: Curvas de carregamento de uma bateria de lítio-íon. Adaptado de (Jiang and Zhang, 2015)

Conforme Jiang and Zhang (2015) mostra, o SOC é afetado pela desgaste da bateria durante ciclos de carga e descarga. A Figura 3.2 que a capacidade máxima de descarga é reduzido à medida que os ciclos são realizados.

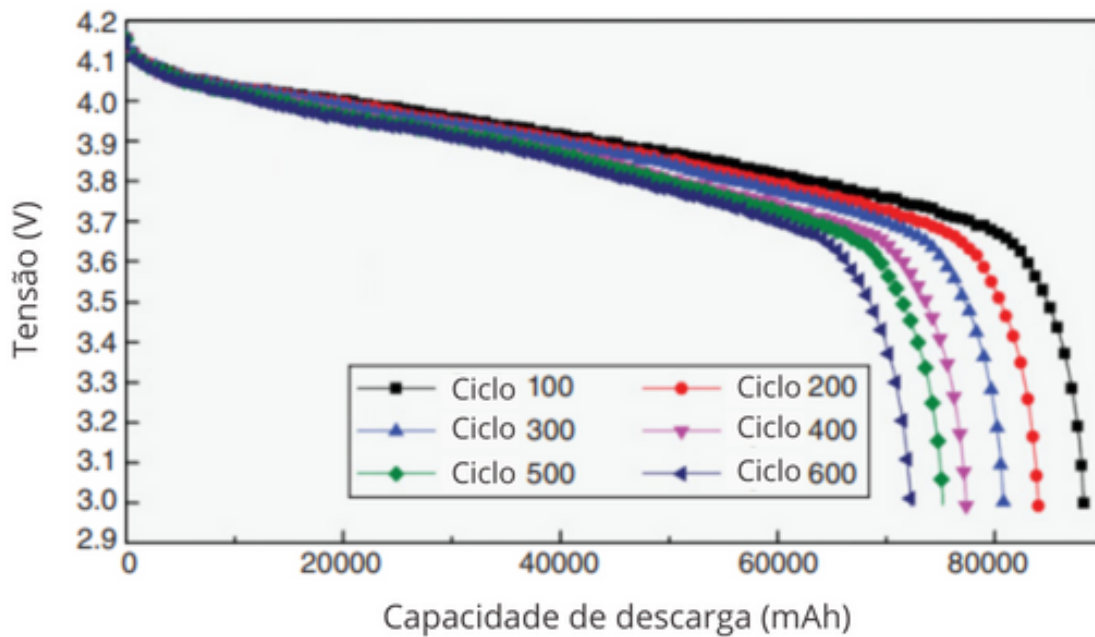


Figura 3.2: Curvas de descarga de uma bateria de lítio-íon após vários ciclos. Adaptado de (Jiang and Zhang, 2015)

Há diversas metodologias para a estimação do SOC que serão brevemente percorridas, e cada uma possui suas vantagens e desvantagens. Neste trabalho a técnica escolhida para ser aplicada foi a contagem de Coulomb.

3.1.1 Tensão de circuito aberto (OCV)

Este método funciona através da caracterização de uma curva de tensão de circuito aberto (OCV), com a caracterização feita através de medição direta da curva de OCV ou então através de modelos matemáticos de células. (Lavigne et al., 2016) Neste método, a tensão na célula é medida continuamente e o SOC equivalente é obtido de uma tabela. (Rivera-Barrera et al., 2017) A relação OCV-SOC muda dependendo da célula e possui muita imprecisão. (Rivera-Barrera et al., 2017) A Figura 3.3 demonstra a diferença na curva OCV-SOC quando a bateria é alimentada com valores diferentes de corrente.

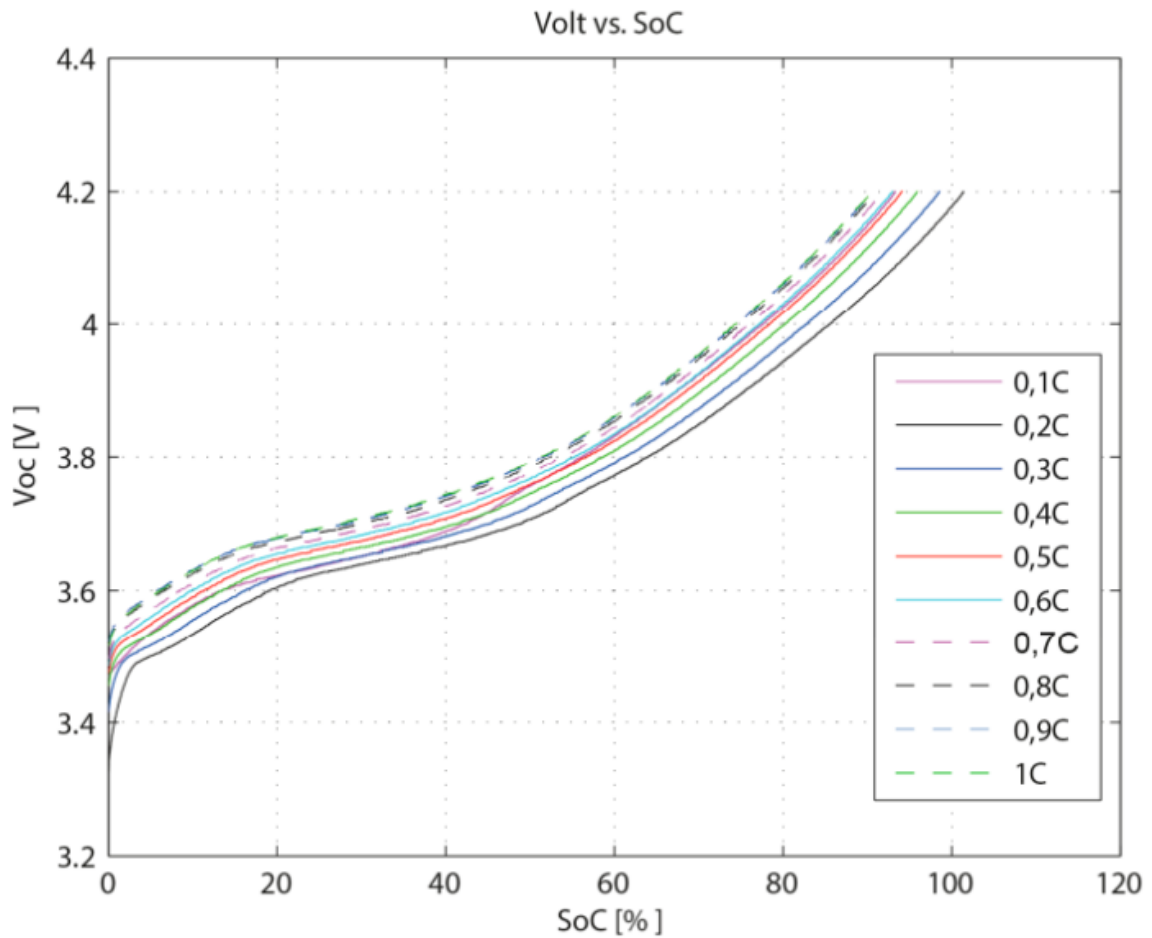


Figura 3.3: Curvas de OCV para diferentes valores de corrente na bateria. Adaptado de (Rivera-Barrera et al., 2017)

Ainda que tenha muitos problemas práticos, este método é popularmente utilizado como suporte para outros métodos. (Rivera-Barrera et al., 2017)

3.1.2 Contagem de Coulomb

Através deste método, o SOC pode ser calculado através da comparação da quantidade de Coulombs remanescentes na bateria e a capacidade total dela. (Movassagh et al., 2021) Conforme a 3.2, a aplicação da contagem de Coulomb funciona essencialmente integrando a corrente de entrada ou de saída da bateria:

$$SOC(t) = SOC(t - 1) + \int \frac{I(t)}{C_{bat}} dt \quad (3.2)$$

Em 3.2, SOC(t) representa o SOC em um instante t, SOC(t - 1) representa o último SOC calculado, e I(t) é a corrente entrando ou saindo da bateria, Q é a capacidade da

bateria e ΔT é a diferença entre os instante t e $t-1$.

A contagem de Coulomb tem como vantagem ser simples e demandar menos poder computacional do que outros métodos, e tem como desvantagens o fato de precisar conhecer previamente o estado de SOC inicial da bateria e ser um sistema de malha aberta, o que propaga erros. (Saji et al., 2019) Um erro na medição de SOC pode resultar em sobrecarga ou sobredescarga da bateria. (Xing et al., 2011) A precisão da contagem de Coulomb depende primeiramente da precisão da medição de corrente da bateria e da estimação do SOC inicial. O processo de integração de corrente da bateria possui erros acumulados durante a carga e descarga. O processo de autodescarga também acumula erros na integração. (Ng et al., 2009)

3.1.3 Redes Neurais

O SOC pode ser estimado através de redes neurais treinando uma rede através de dados de um modelo matemático de uma bateria de lítio-íon. (S. Hou and Han, 2021) De acordo com Hu and Wang (2019), os parâmetros de entrada para um sistema de estimação de SOC usando uma rede neural baseada no algoritmo *back propagation* (BP) são primeiramente a corrente e a tensão.

A rede neural é capaz de encontrar a relação entre parâmetros de dados complexos. O modelo de predição de SOC não é influenciado por características não-lineares ou mecanismos internos da bateria e não possui erro acumulado. (Yan and Wang, 2017)

A Figura 3.4 mostra um exemplo de estrutura modelo de predição que usa a corrente e a tensão da bateria:

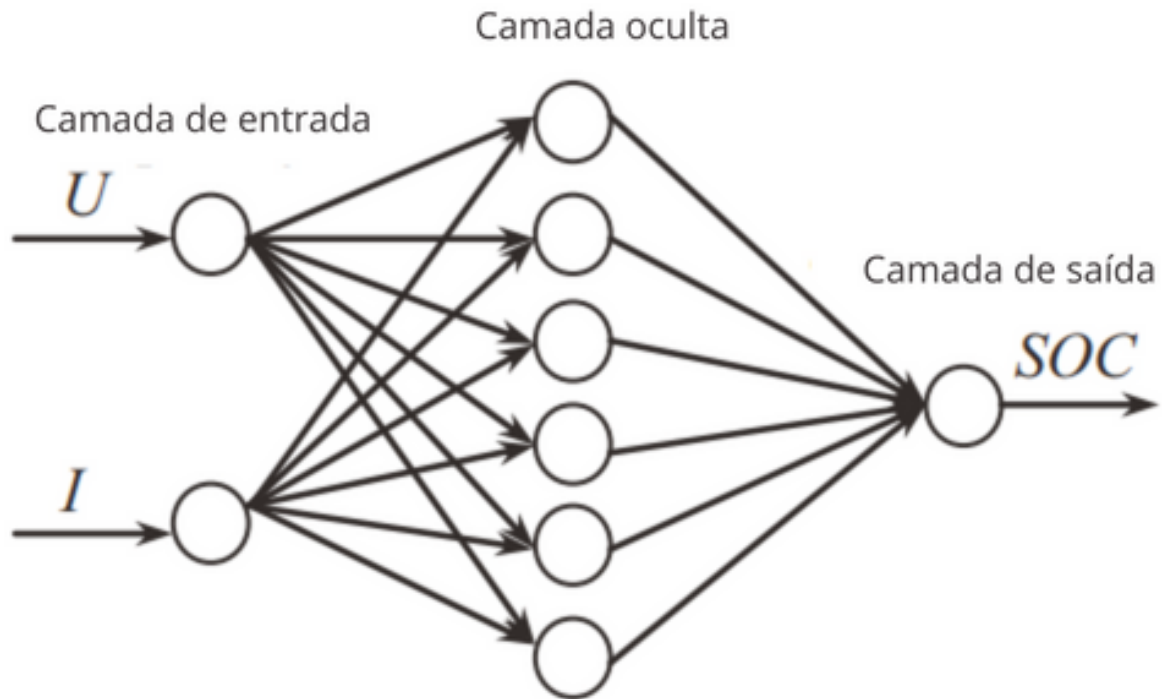


Figura 3.4: Modelo de predição de SOC baseado em uma rede neural do tipo BP. Adaptado de (Yan and Wang, 2017)

3.1.4 Filtro de Kalman Estendido

O filtro de Kalman é um método recursivo que usa um sistema descrito em espaço de estados para obter uma estimativa ótima de uma variável de interesse que não pode ser medida diretamente, é calculado de forma recursiva e tem a característica de adaptabilidade. A técnica conhecida como filtro de Kalman estendido (EKF) é uma combinação da linearização das características não lineares da bateria seguido do cálculo recursivo do filtro de Kalman. (Jiang and Zhang, 2015) A descrição matemática do filtro de Kalman pode ser encontrada em (Simon, 2006), e a descrição procedual da aplicação teórica do EKF pode ser lida em (Jiang and Zhang, 2015)

O EKF é capaz de estimar o SOC mesmo com um valor inicial inacurado de SOC. A precisão do método é reduzida quando há erros nos parâmetros do modelo da bateria.

Em relação a outros métodos de estimação de SOC, o EFK possui uma ótima precisão da estimativa e a vantagem de não depender do conhecimento do SOC inicial da bateria. Entretanto, a precisão deste método depende fortemente da acurácia do modelo do circuito

equivalente da bateria, e o custo computacional é elevado. (Liu et al., 2017)

3.2 Estado de Saúde (SOH)

O estado de saúde (SOH, em inglês, *state of health*) é um dos parâmetros mais importantes para a operação de um BMS. (Jiang and Zhang, 2015) As baterias em geral sofrem desgaste ao longo dos ciclos de operação, resultando em perda de capacidade máxima. (Linden and Reddy, 2002) Essa degradação provoca a necessidade de estudo sobre a característica que relaciona capacidade máxima com a capacidade atual.

O SOH é definido pela Equação 3.3 como a relação percentual entre a capacidade de carga média (Q_r) e capacidade máxima nominal (Q_{max}). (Jain et al., 2021)

$$SOH = \frac{Q_r}{Q_{max}} 100\% \quad (3.3)$$

Pode se dizer que o SOH é a característica que permite medir o desempenho de uma bateria em uma condição atual de uso em relação a uma nova. Ele também é um indicador de quanto da vida útil da bateria ainda resta antes da necessidade de substituição. (Jain et al., 2021)

O SOH também tem influência nos métodos de estimação de SOC, sendo fundamental a sua calibração para maximizar a precisão do SOC. (Ng et al., 2009)

As principais características que influenciam no SOC são a temperatura de operação, ciclos de carga e descarga, profundidade de descarga, sobrecarga, taxas de carga e descarga e envelhecimento da célula. (Balagopal and Chow, 2015)

Uma forma simples de medição de SOH é armazenar os dados de OCV e SOC de uma bateria nova, que supostamente possui 100% de SOH, e então fazer uma comparação do SOC e OCV atuais com os valores armazenados utilizando a Equação 3.3. Conforme mostra a Figura 3.2, a redução da capacidade da bateria faz com ela precise de uma quantidade de carga diferente para atingir as OCVs de corte, ou seja, para um mesmo valor de OCV espera-se que uma bateria usada tenha um SOC diferente de uma bateria nova. Segundo Jain et al. (2021), a desvantagem de medir o SOH através deste método é que ele depende da realização de ciclos completos de carga e descarga para atualizar a estimativa do SOH.

3.3 Carregamento da bateria

De acordo com Linden and Reddy (2002), as baterias podem ser danificadas permanentemente ao sofrerem sobrecarga, sobrecorrente ou superaquecimento. Para evitar estes danos, usualmente são adotadas algumas estratégias de carregamento.

De acordo com Ming et al. (2018) e Etacheri et al. (2011), algumas das metodologias mais utilizadas para carregamento são: Corrente Constante (*Constant Current*, ou CC), que é um o método em que a bateria é alimentada por uma corrente constante; Tensão Constante (*Constant Voltage*, ou CV), no qual a bateria é carregada por uma tensão constante; Temperatura Constante (*Constant Temperature*, ou CT), a temperatura da bateria é mantida constante; e Corrente Constante Tensão Constante (*Constant Current Constant Voltage*, ou CC-CV), que é um método que carrega a bateria em dois estágios, um com corrente constante e outro com tensão constante.

O método CC-CV tem algumas vantagens em relação aos outros apresentados. Nesta metodologia, uma bateria descarregada é inicialmente alimentada por uma corrente constante, o que previne correntes muito elevadas. Quando a bateria atinge um nível de SOC mais elevado, o carregamento é chaveado para o modo CV, que evita sobrecarga e faz com a corrente seja gradualmente reduzida enquanto o SOC é aumentado.

Capítulo 4

Desenvolvimento de um sistema de monitoramento e supervisão para sistemas de armazenamento de energia elétrica

Neste capítulo serão mostrados os detalhes do sistema desenvolvido para este trabalho. Dentre os principais componentes e tecnologias usados estão: no *hardware* foram utilizados o NodeMCU ESP8266, sensor de corrente ACS712, multiplexador analógico, resistores e transistores; no software foi feito o uso do *Node.js*, *javascript* no navegador e *MySQL*.

Os códigos utilizados neste trabalho podem ser encontrados no Apêndice A.

4.1 Visão Geral

A Figura 4.1 mostra a visão geral do sistema. Primeiramente, a finalidade do *hardware* é fazer a estimação do SOC através de algoritmos embarcados no NodeMCU, tais algoritmos utilizam dados das medições dos sensores ligados na bateria e no próprio microcontrolador, e então estes dados são enviados para o servidor utilizando *websocket*. Ao receber uma mensagem via *websocket*, o servidor faz o tratamento do pacote de dados e os envia para o banco de dados relacional MySQL. O servidor também foi programado para suportar requisições HTTP.

O *frontend*, que é uma aplicação web, inicia fazendo uma requisição HTTP para

o servidor que o responde com os dados armazenados no banco. Com estes dados, a aplicação plota gráficos de tensão, corrente e SOC. Além disso, é feito um *handshake* com o servidor *websocket*, para que os dados coletados em tempo real pelo *hardware* sejam disponibilizados imediatamente para o *frontend*. Então, a aplicação atualiza os gráficos plotados.

O NodeMCU controla o estado do sistema, isto é, ele decide se a bateria será carregada ou descarregada, usando o SOC estimado como referência para a escolha do estado.

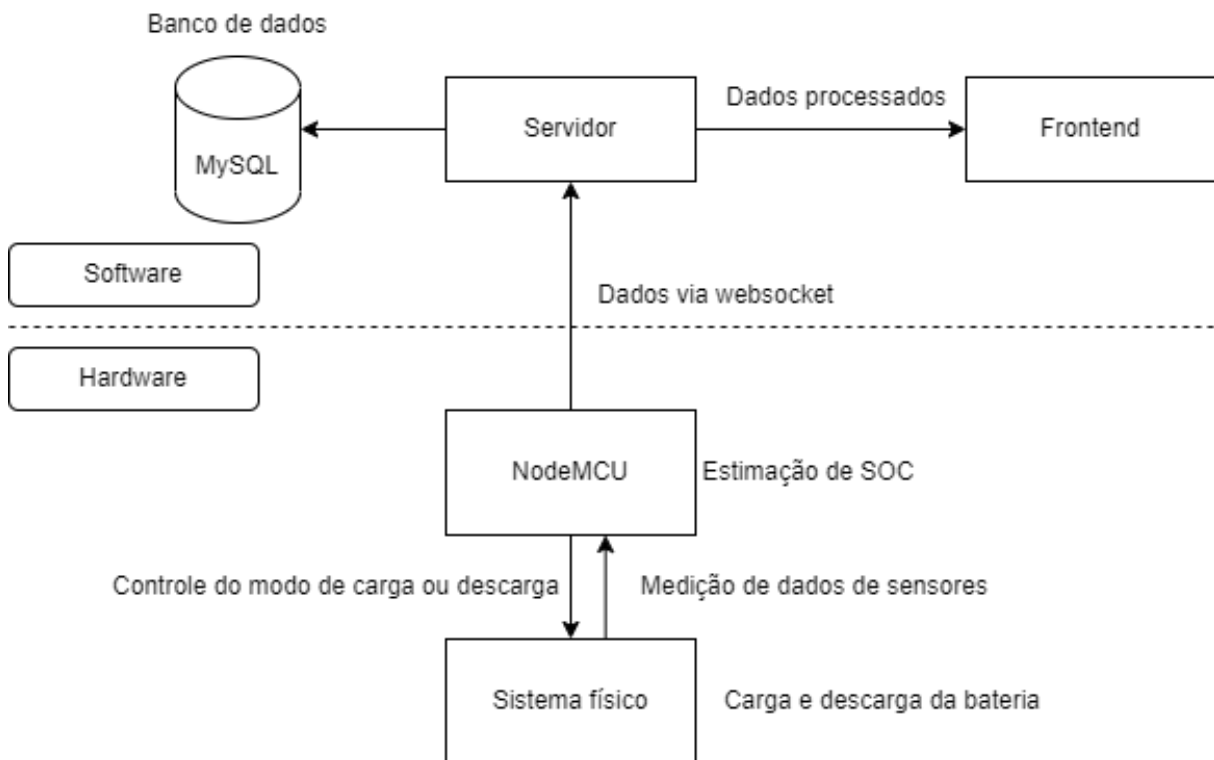


Figura 4.1: Visão geral do sistema. Elaborado pelo autor.

A técnica utilizada para estimação de SOC foi uma combinação da estimação via OCV e contagem de Coulomb.

Eventualmente neste trabalho serão mostradas algumas figuras contendo dados relativos ao SOH. Apesar disso, nenhuma técnica de estimação de SOH foi implementada durante o desenvolvimento.

4.2 Hardware

O NodeMCU ESP8266 foi o microcontrolador escolhido por causa de sua fácil e ampla utilização, além de já ter o módulo ESP8266 que possibilita a conexão à internet sem fio.

O microcontrolador é conectado à internet com o uso das bibliotecas "ESP8266WiFi.h" e "ESP8266WiFiMulti.h". E ele se conecta através de *websocket* com o uso da biblioteca "WebSocketsClient.h".

A Figura 4.2 mostra o circuito elétrico desenvolvido. Esse circuito é capaz de medir a corrente e tensão da bateria, chavear entre os modos de carga e descarga, e alternar entre os modos de carga CC e CV.

As baterias utilizadas neste trabalho foram todas de lítio-íon com capacidade de 3800mAh, modelo FX-L18650.

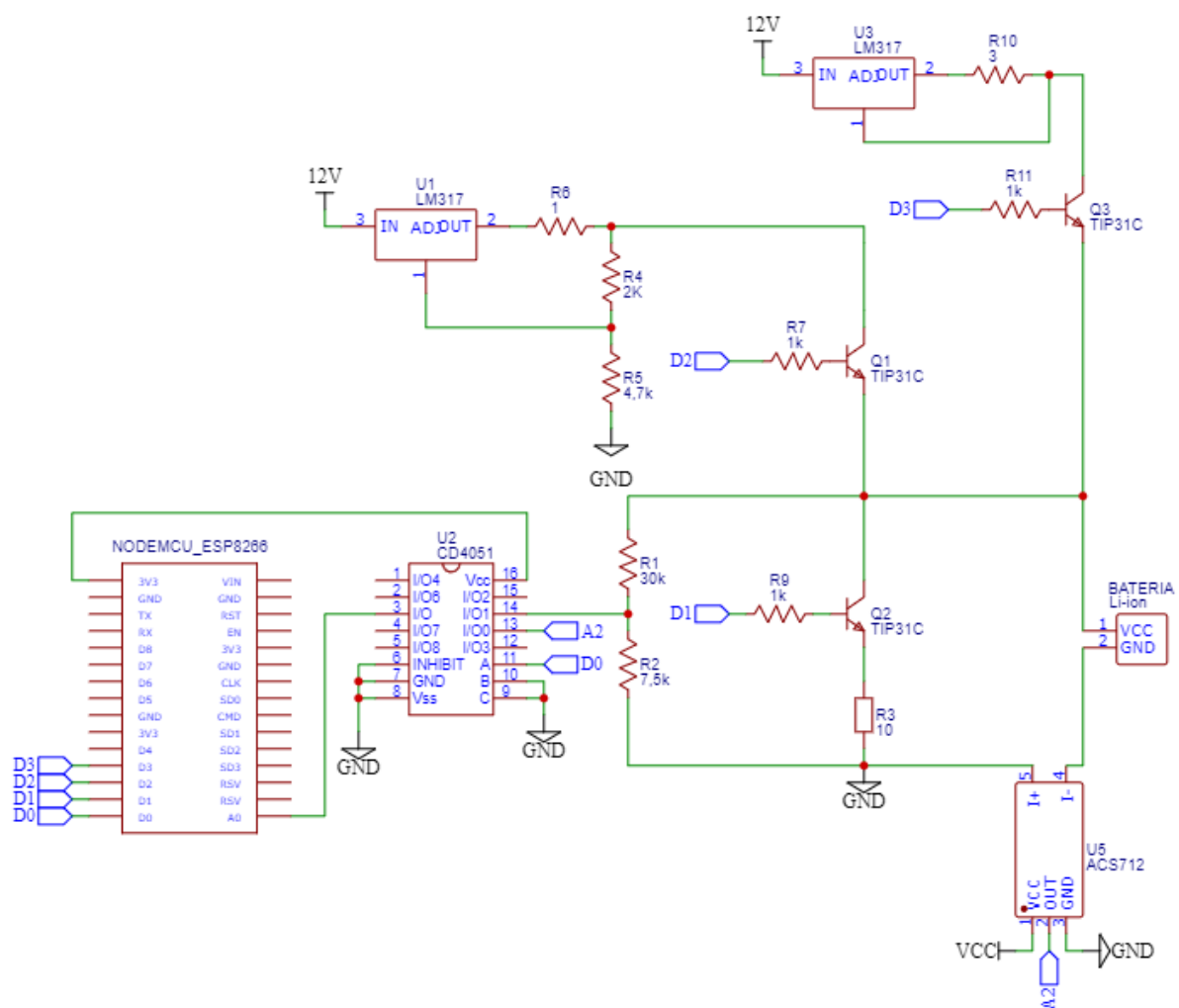


Figura 4.2: Circuito elétrico desenvolvido. Elaborado pelo autor.

4.2.1 Multiplexador analógico

Para o projeto, é necessária a medição de corrente e tensão. Portanto, são necessários dois pinos analógicos para realizar a leitura dos sensores. Como o NodeMCU ESP8266

possui apenas um pino desse tipo, foi utilizado o multiplexador analógico CD4051 para possibilitar a leitura de múltiplas entradas analógicas.

Na Figura 4.2, a saída digita D0 serve para multiplexar entre as portas analógicas I/O0 e I/O1. Sendo I/O0 utilizada para medição de corrente e I/O1 para a medição de tensão.

4.2.2 Medição de tensão

A medição de tensão é realizada pelo conversor A/D do microcontrolador. Como a tensão máxima suportada pelo conversor é de 3,3V, e uma bateria de lítio-íon completamente carregada possui uma tensão que pode ultrapassar 4V, foi utilizado um módulo comercial de divisor de tensão que possui os resistores R1 e R2 na 4.2. Com o divisor de tensão, pode ser calculada a nova tensão suportada, conforme calculado com a Equação 4.1.

$$3,3 = \frac{7,5}{7,5 + 30} V_{max} = 16,5V \quad (4.1)$$

$$V_{max} = 16,5V \quad (4.2)$$

Com esse novo valor de tensão máxima de operação, a resolução do ADC se torna 16,11mV, demonstrado pela Equação 4.3.

$$resolucao = \frac{16,5}{1024} = 16,11mV \quad (4.3)$$

4.2.3 Medição de corrente

Foi utilizado o sensor de corrente AC/DC ACS712 5A conectado em série à bateria para realizar a medição.

O ACS712 é capaz de medir corrente contínua e alternada na faixa de -5A a 5A através do efeito Hall. O sensor possui três terminais, sendo eles o Vcc, GND, e uma saída analógica. Além disso, há um borne no qual deve se conectar o circuito que se deseja medir a corrente. A alimentação nominal é de 5V. O módulo possui uma sensibilidade de 185mV/A.

Durante a maioria dos testes, este sensor apresentou bastante ruído, o que chegou até a inviabilizar a validação de alguns resultados de experimentos. Para mitigação dos ruído, foi utilizada a biblioteca "ACS712.h" que realiza uma calibração que calcula o *offset* do ruído, e utiliza a média de várias medições, fazendo com que a saída seja mais estável.

4.2.4 Estimação de SOC

Segundo Saji et al. (2019), a CC precisa saber o estado de carga inicial da bateria. Por isso, foi utilizado a medição de OCV para estimar o estado de carga inicial. Após isso, o algoritmo da contagem de Coulomb foi implementado para realizar a estimação de SOC. A Figura 4.3 mostra o diagrama de estados da estimação de SOC.



Figura 4.3: Diagrama de estados da estimação de SOC. Elaborado pelo autor.

4.2.5 Modos de operação do circuito

O circuito da Figura 4.2 é capaz de alternar entre quatro estados: medição de OCV, descarga, carga com corrente constante e carga CV. A alteração dos modos é feita usando o microcontrolador para alterar o estado dos transistores entre ativos e em corte.

No modo de medição de OCV, todos os transistores são colocados em corte para que a bateria opere com uma corrente bem baixa, ocasionada pela medição de tensão. Este modo é utilizado para determinar o estado de SOC inicial da bateria e é o estado inicial do sistema.

No modo de descarga, apenas Q2 é ativo. Neste modo a bateria descarrega dissipando a potência em R3. Quando ela atinge 30% de SOC, o circuito muda para o modo carregamento CC

No modo carga com corrente contínua, apenas Q3 é ativo. A bateria carrega com corrente constante até atingir um SOC de 60%, para então mudar o estado para carrega-

mento CV. Para obter uma corrente constante de 417mA, foi utilizada uma configuração com do LM317 com o resistor R10, O cálculo é mostrado na Equação 4.4. O LM317 possui esta e várias outras configurações em seu datasheet.

$$I_{cc} = \frac{1,25}{R10} = 417mA \quad (4.4)$$

No modo carga CV, apenas Q1 é ativo. A bateria carrega com tensão constante até atingir 80% de SOC, e então passa para o modo de descarga. Com essa configuração de resistores, o LM317 fornece 4,19V para a bateria. O cálculo é mostrado na Equação 4.5

$$V_{cv} = 1,25 * \left(1 + \frac{R5}{R4}\right) = 4,19V \quad (4.5)$$

A combinação dos métodos de OCV e CC foi baseada no estudo de Baccouche et al. (2017).

A estimação do SOC inicial usando a curva SOC-OCV utiliza a curva para temperatura de 25°C dada por Zhang et al. (2018) mostrada na Figura 4.4.

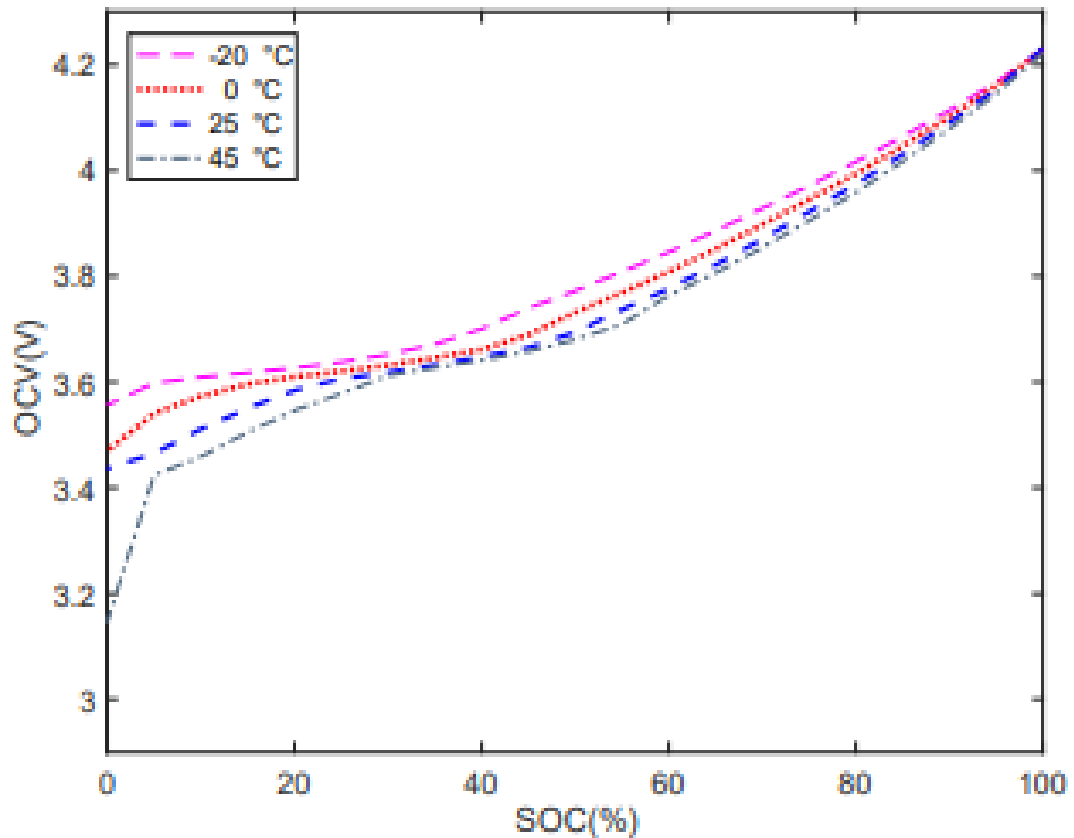


Figura 4.4: Curvas de SOC-OCV de acordo com temperatura. Adaptado de Zhang et al. (2018).

Para ser embarcada, são feitas aproximações lineares da curva dividindo-a em oito segmentos usando o estudo de Baccouche et al. (2017), conforme mostra a Figura 4.5. A curva passa a ser descrita pela equação 4.6.

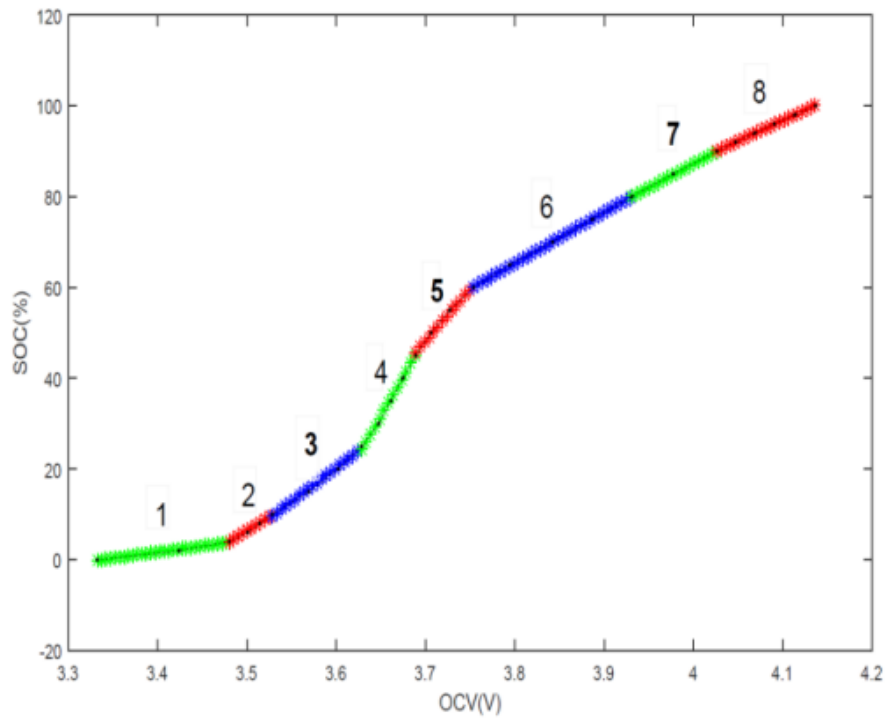


Figura 4.5: Aproximação linear para curva de SOC-OCV. Adaptado de Baccouche et al. (2017).

$$SOC(OCV) = a * OCV - b \quad (4.6)$$

Os valores de da Equação a e b 4.6 dependem do valor de OCV e estão descritos na Tabela 4.1

Tabela 4.1: Relação de SOC-OCV a 25°C. Adaptado de Baccouche et al. (2017)

Segmento	1	2	3	4	5	6	7	8
Faixa de Tensão (V)	[3.3; 3.452]	[3.452; 3.508]	[3.508; 3.595]	[3.595; 3.676]	[3.676; 3.739]	[3.739; 3.967]	[3.967; 4.039]	[4.039; 4.132]
A	26.55	125	149	344	229.5	111.9	104.8	90.61
B	88.6	431.1	516.1	1225	800.9	359.9	332	274.7

Com o estado inicial de SOC na memória, o programa passa então a estimar o SOC através da CC aplicando a Equação 3.2. A CC é calculada a cada 500 milisegundos.

Um observação importante é que o método de aproximação da curva de SOC-OCV mostrado acima utilizou dados de outros estudos e usam modelos de baterias diferentes das usadas neste trabalho. Por isso, a validação prática da metodologia de estimação de

SOC via OCV implementada no sistema está altamente suscetível a imprecisões.

A Figura 4.6 mostra uma aproximação de valores ideais para o SOC de uma bateria em operação. Segundo Linden and Reddy (2002), a sobrecarga ou sobredescarga podem causar danos permanentes na bateria. Por isso, foi determinada a operação do sistema desenvolvido com valores de SOC entre 30% e 80%.

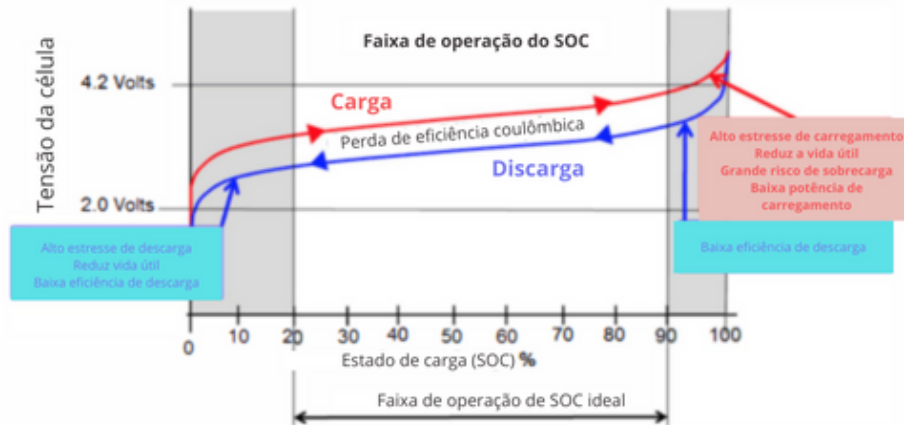


Figura 4.6: Faixa de SOC ideal para operação de uma bateria. Adaptado de Aoxia and K. (2016)

Ao iniciar o sistema, ele começa no estado de estimação de SOC via OCV, então se o SOC for maior que 30%, o estado é alterado para o modo de descarga da bateria. E caso o SOC for menor que 30%, o sistema entra no modo de carregamento por corrente constante. Neste modo, ao atingir 60% de SOC, o sistema passa para o modo CV. No modo CV, ao atingir 80% o estado é alterado para o modo de descarga.

4.3 Software

Esta sessão descreve a implementação das camadas de software utilizadas neste trabalho. O objetivo dos softwares desenvolvidos é fornecer um serviço capaz de armazenar e exibir dados em tempo real, que são coletados ou calculados pela camada de hardware.

4.3.1 Backend

O *backend* do projeto foi desenvolvido utilizando o Nodejs, que é um ambiente que permite a execução de javascript fora do navegador. O intuito do *backend* é executar um servidor *websocket* com a funcionalidade de diferenciar os usuários - o NodeMCU e

o frontend -, mandando e recebendo mensagens adequadas sobre os estados do sistema para cada um deles.

Todo instante que o NodeMCU captura dados do circuito, ele os envia para o servidor através de *websocket*. E então, imediatamente, o servidor armazena estes dados no banco e os envia para o *frontend*, caso ele esteja conectado.

A Figura 4.7 mostra o fluxo de dados do sistema.

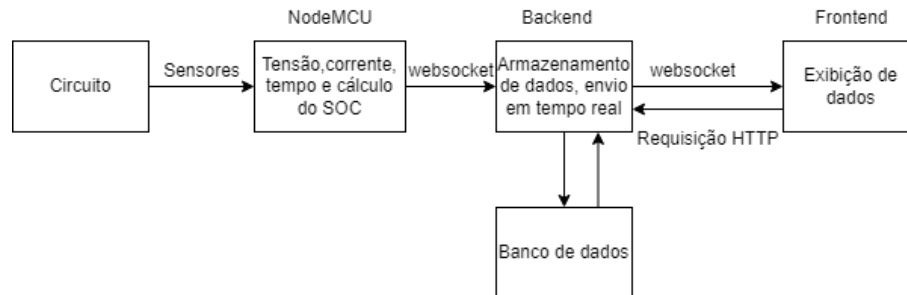


Figura 4.7: Fluxo de dados do sistema. Elaborado pelo autor.

O *backend* também é responsável por receber do estado de operação do circuito e o mandar para o *frontend*.

A outra funcionalidade do *backend* é fazer o armazenamento em um banco de dados de todas as medições enviadas pelo NodeMCU. Foi utilizado o banco de dados MySQL e a comunicação entre o ele o Nodejs é feita usando o Knex, que é um *query builder* feito em javascript.

O banco de dados tem um formato bem simples, ele possui apenas uma tabela, mostrada na Figura 4.8. A coluna "ocv" refere-se à tensão nos terminais da bateria, "corrente" é a medição da corrente que passa pela bateria, "soc" é estimado pelo microcontrolador, e "timestamp" é o tempo em milisegundos desde o início da execução do sistema. "soh" não é estimado neste trabalho, mas possui uma coluna na tabela.

id	ocv	corrente	soc	soh	timestamp
10702	3.8379	0.0044	67.9184	100.0000	29941
10703	3.8135	-0.0290	67.9183	100.0000	30448
10704	3.8232	-0.0150	67.9182	100.0000	31005
10705	3.8184	-0.0200	67.9181	100.0000	31513
10706	3.8281	0.0093	67.9182	100.0000	32020
10707	3.8232	-0.0240	67.9181	100.0000	32529
10708	3.8281	-0.0150	67.9180	100.0000	33040
10709	3.8379	-0.0150	67.9180	100.0000	33550

Figura 4.8: Tabela do banco de dados visualizada no MySQL Workbench. Elaborado pelo autor.

4.3.2 Frontend

O frontend foi todo desenvolvido em javascript para ser uma página web para ter como objetivo a exibição gráfica em tempo real dos dados enviados pelo servidor. Os gráficos exibidos foram feitos utilizando a biblioteca *Chartjs*.

Ao iniciar a página web, ela faz uma requisição HTTP para o servidor que responde com os dados armazenados no banco de dados. Então os gráficos são plotados e o *frontend* já se conecta ao servidor *websocket*. Daí em diante o servidor, assim que recebe novos dados do microcontrolador, envia mensagens contendo novos dados para o frontend, que por sua vez atualiza os gráficos em tempo real.

Tanto as requisições HTTP quanto as mensagens via websocket utilizam o formato JSON para compatibilidade em todos os ambientes.

A Figura 4.9 mostra o *frontend* desenvolvido com alguns dados fictícios, sem o backend funcionando. Na parte superior esquerda são mostrados os dados de SOC e SOH, na parte superior direita é exibido o modo de operação do circuito. Os gráficos de tensão, corrente e SOC ficam abaixo.

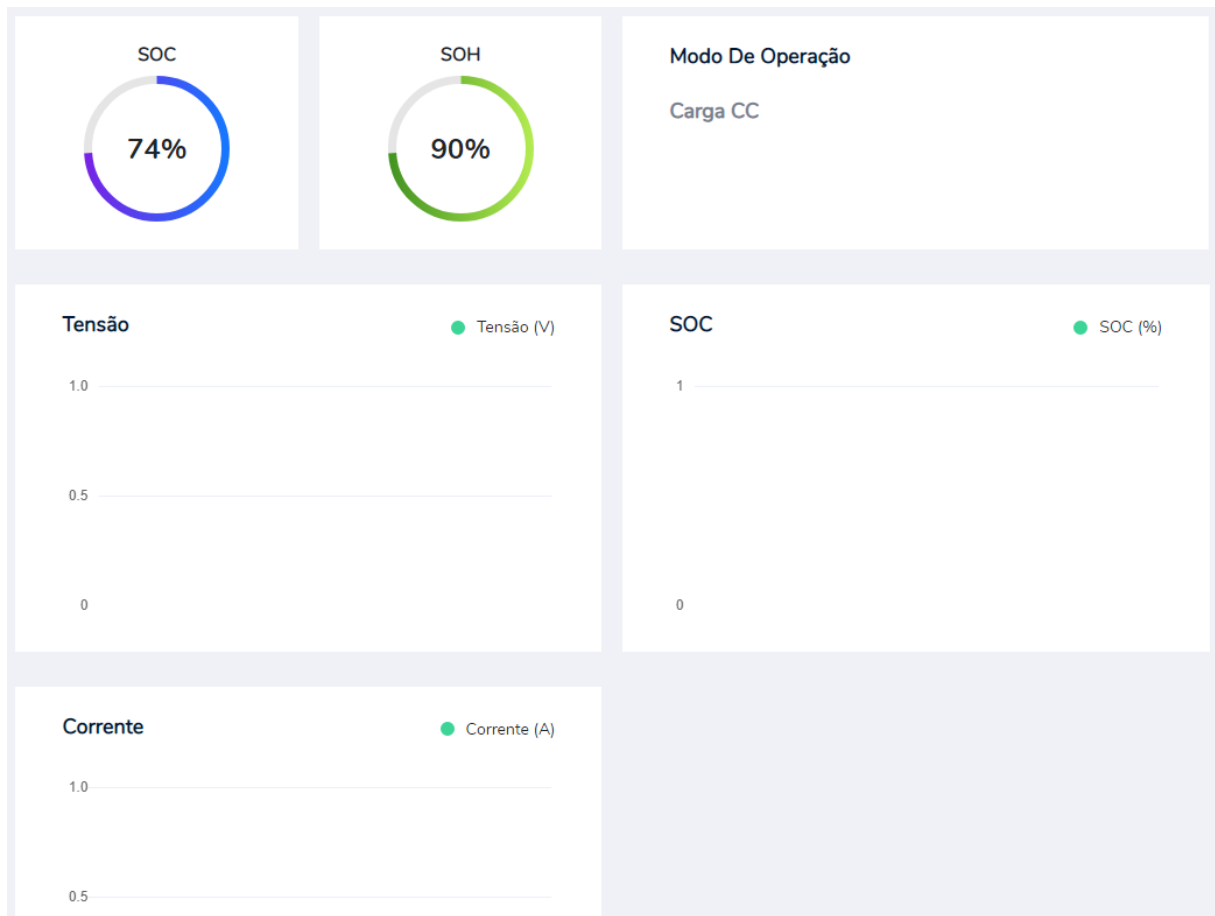


Figura 4.9: Frontend do projeto rodando em um navegador web. Elaborado pelo autor.

Capítulo 5

Análise de Resultados

Neste capítulo serão analisados os resultados de alguns testes realizados para validar as funcionalidades desenvolvidas neste projeto.

Para o sistema todo funcionar é necessário ter o *backend* sendo executado, um processo do MySQL aberto, o microcontrolador deve estar em execução e é necessário abrir ou atualizar pelo menos uma vez a página do *frontend* assim que o *backend* esteja inicializado.

A Figura 5.1 mostra o circuito montado para realização dos experimentos a seguir. O esquemático deste circuito é o da Figura 4.2.

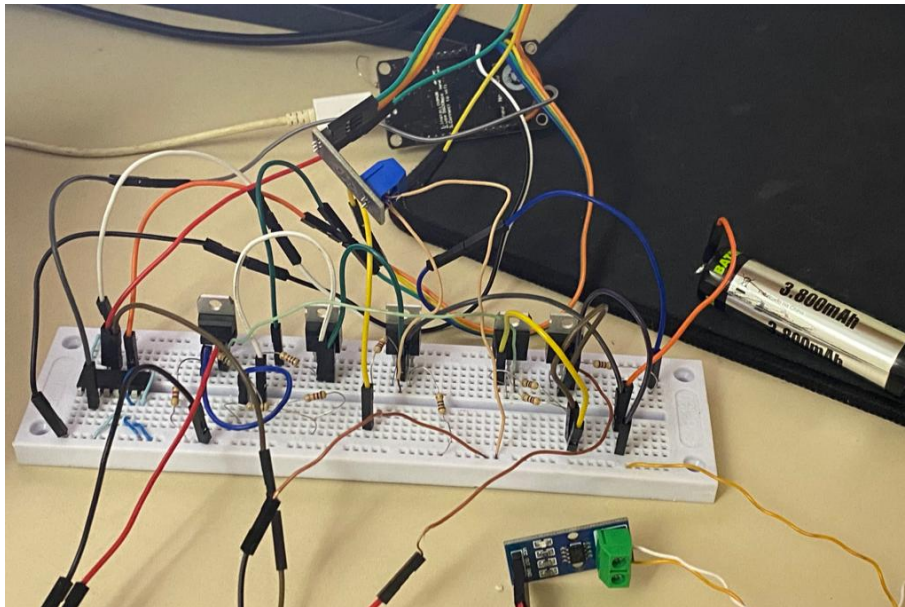


Figura 5.1: Circuito montado em protoboard. Elaborado pelo autor.

5.1 Teste de descarga

Neste teste foi utilizada uma bateria carregada que descarrega dissipando a energia em uma carga resistiva. Os resultados são mostrados na Figuras 5.2, 5.3 e 5.4.

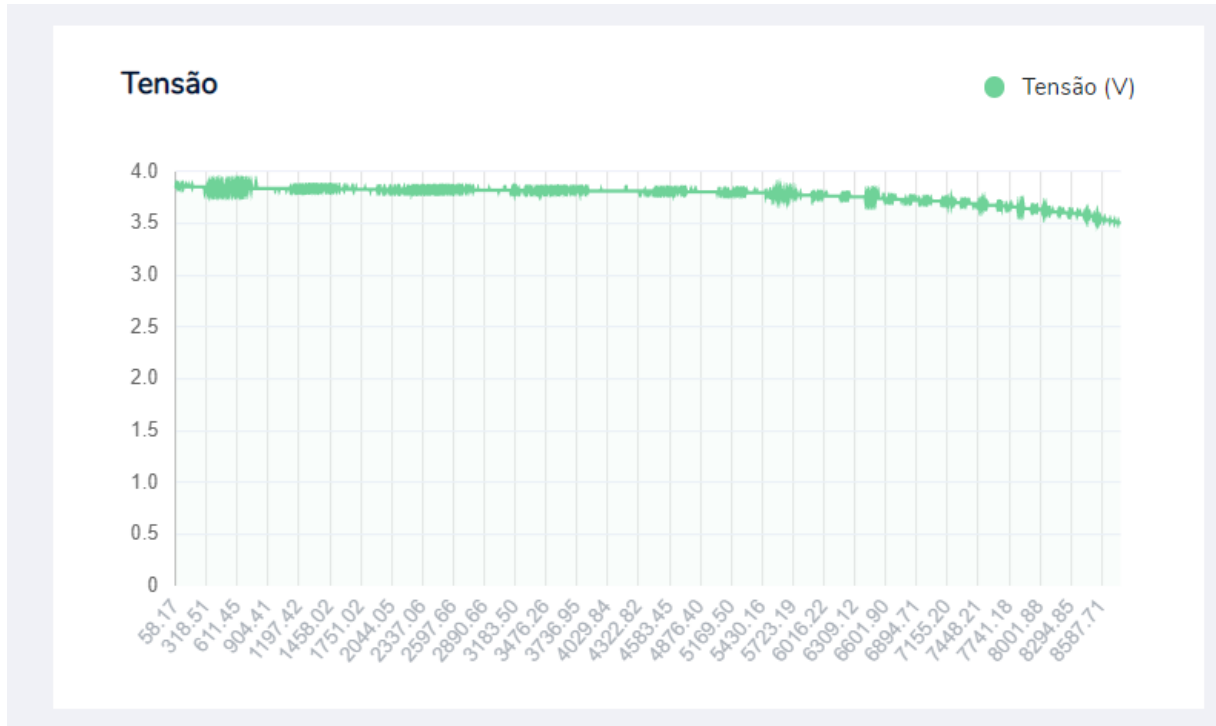


Figura 5.2: Tensão durante o teste de descarga da bateria. Elaborado pelo autor.

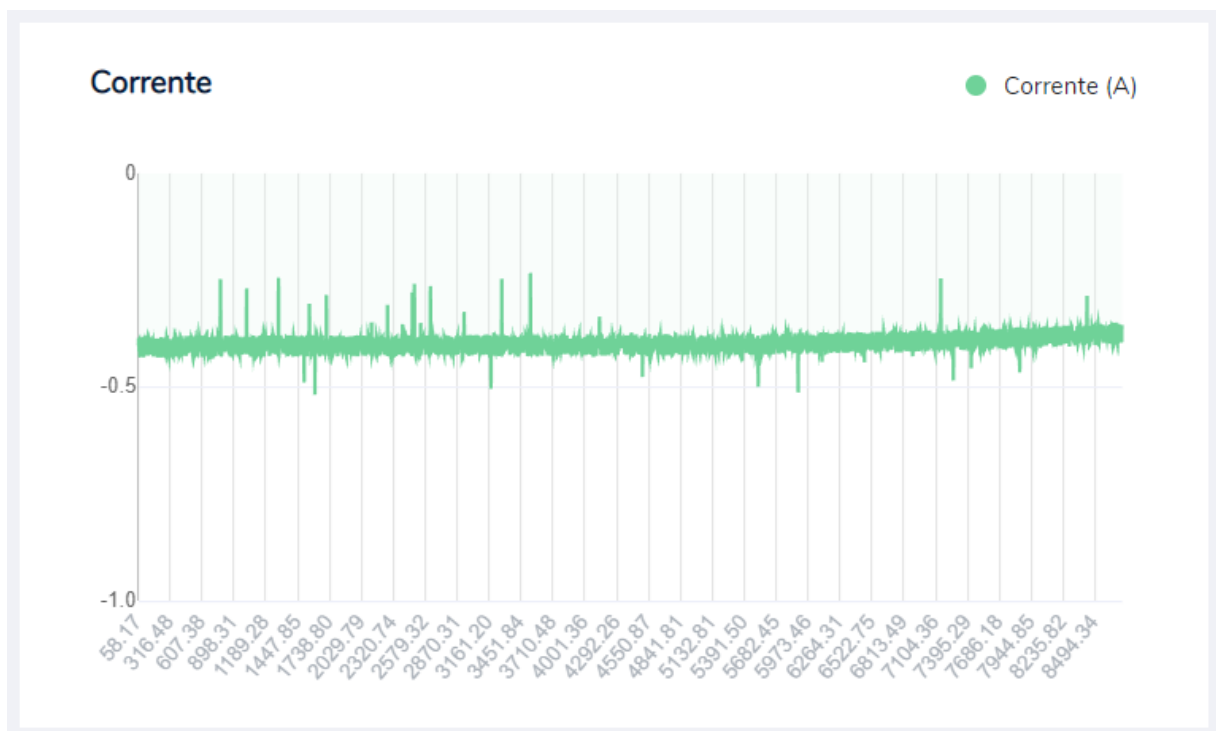


Figura 5.3: Corrente o teste de descarga da bateria. Elaborado pelo autor.

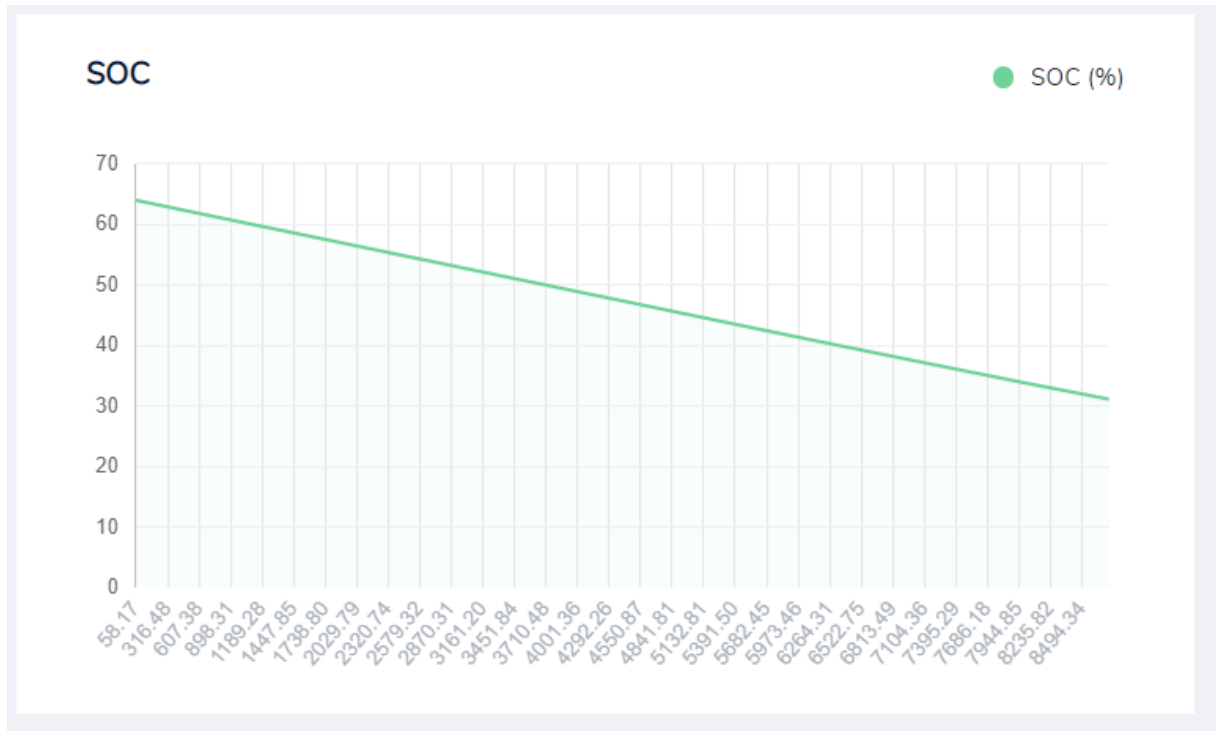


Figura 5.4: SOC o teste de descarga da bateria. Elaborado pelo autor.

Através de inspeção visual, é nota-se que a tensão é diminuída conforme o SOC é reduzido, de forma similar a o que ocorre na Figura 3.2.

5.2 Teste de carga

Neste teste, uma bateria descarregada foi conectada ao circuito, ela inicialmente carrega no modo *Constant Current* e posteriormente, ao atingir 60% de SOC, o carregamento muda para o modo *Constant Voltage*.



Figura 5.5: Gráfico da tensão na bateria durante o carregamento CC-CV. Elaborado pelo autor.

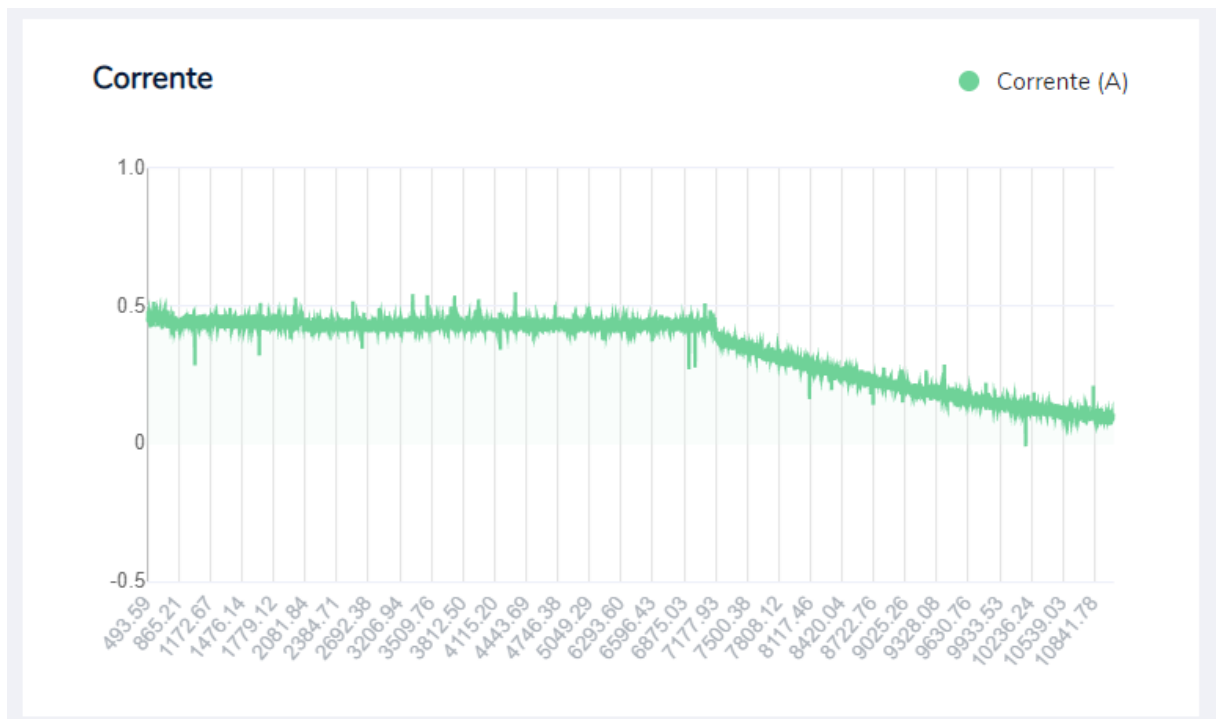


Figura 5.6: Gráfico da corrente na bateria durante o carregamento CC-CV. Elaborado pelo autor.

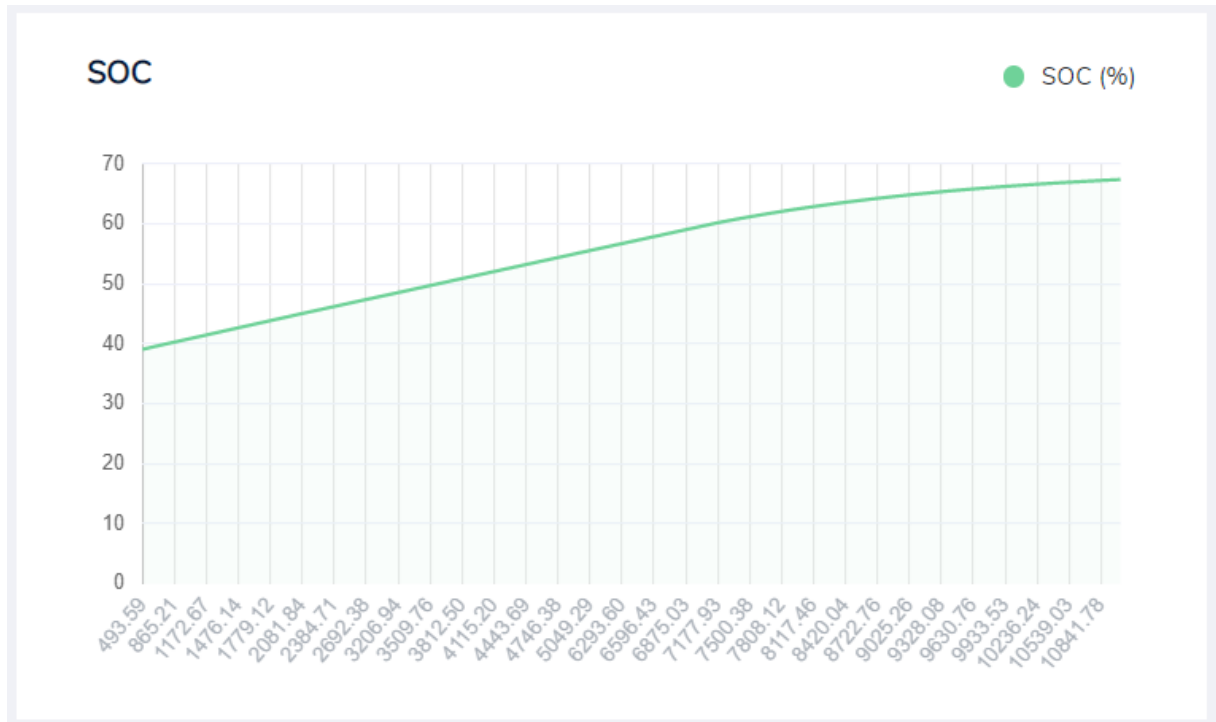


Figura 5.7: Gráfico do SOC durante o carregamento CC-CV. Elaborado pelo autor.

Na Figura 5.5 observa-se que a tensão na bateria é aumentada gradativamente até que o SOC, na Figura 5.7, atinge 60%. Daí em diante o sistema entra no modo CV, sendo observado que a tensão se estabilizou e a corrente, Figura 5.6, começou a decair gradativamente. Também observa-se que a curva do SOC passou a crescer mais lentamente a partir deste ponto, o que é esperado, já que no modo CV a corrente é mais baixa.

Capítulo 6

Conclusão

Neste trabalho foi desenvolvido um sistema para monitoramento e supervisão de carga e descarga de uma bateria de lítio-íon. O trabalho foi desenvolvido em três blocos diferentes, sendo eles o circuito elétrico, *frontend* e *backend*. No circuito elétrico foram aplicados métodos para a aquisição de dados de sensores, acionamento de circuitos de carga e descarga através de algoritmo embarcado, conexão do sistema com um servidor *websocket* e estimação de SOC. No *frontend* foi realizado a plotagem em tempo real de dados obtidos através das medições de sensores e estimação de SOC. E no *backend* foram implementados um servidor *websocket*, um HTTP, e armazenamento em bancos de dados.

A principal dificuldade deste trabalho foi realizar uma medição precisa da corrente elétrica na bateria. Muito se deve ao sensor usado para a medição não ser o mais adequado para a medição de corrente na faixa em que os testes foram realizados. Observa-se na Figura 5.6 que a corrente medida possui bastante ruído.

Foi realizada a montagem do circuito elétrico para a realização de testes. Os resultados foram satisfatórios. A comparação das curvas de carga e descarga obtidas com outras curvas retiradas de outros estudos mostraram que o sistema desempenhou conforme o esperado, tanto no modo de carga, como no de descarga. Por fim, a aplicação da contagem de Coulomb foi bem sucedida, os valores de SOC estavam compatíveis com a corrente medida e com a curva OCV-SOC. E o *backend* e o *frontend* funcionaram de acordo com o que foram propostos.

É importante ressaltar que a validade do funcionamento do sistema é altamente comprometida por causa da imprecisão na estimação de SOC através do OCV por conta da natureza dos dados utilizados no trabalho. Outro problema que afeta a precisão do traba-

lho é o fato da contagem de Coulomb ser um método de estimação que integra erro, que associado ao problema do ruído no sensor de corrente cria um erro relativamente elevado no resultado do sistema.

6.1 Trabalhos futuros

Para futuros trabalhos são feitas as sugestões de usar um sensor de corrente mais preciso, implementar um método de estimação de SOH, incluir comandos no *frontend* para permitir o controle manual de variáveis do circuito em tempo real - por exemplo, o controle manual do modo de operação e a faixa de corte para alteração entre cada estado -, e a implementação do circuito em placa de circuito impresso para minimizar ruídos e agentes externos sob a operação do sistema.

Uma grande melhoria no trabalho seria a implementação de uma metodologia mais sofisticada da estimação de SOC. A implementação atual é completamente comprometida pela natureza dos dados utilizados, pelo ruído nas medições e pela metodologia em si. Uma metodologia de estimação de SOC como o filtro de Kalman estendido poderia ajudar no tratamento das entradas ruidosas e das incertezas das entradas do sistema.

Outro aprimoramento que pode ser realizado é a mudança na taxa de envio e atualização das informações via *websocket*. No sistema atual, os dados são enviados em pacotes individuais a cada período de amostragem do sistema (500 milissegundos). O aumento na frequência de envio acarreta em uma sobrecarga de processamento nos sistemas. Uma possível solução é realizar as medições com uma taxa de amostragem maior, enviando-as em pacotes em vez de individualmente.

Referências Bibliográficas

- D. Linden and T. B. Reddy, *Handbook of Batteries*, 3rd ed. McGraw-Hill, 2002.
- H. C. Hesse, M. Schimpe, D. Kucevic, and A. Jossen, “Lithium-ion battery storage for the grid—a review of stationary battery storage system design tailored for applications in modern power grids,” *Energies*, vol. 10, 12 2017.
- M. T. Lawder, B. Suthar, P. W. C. Northrop, S. De, C. M. Hoff, O. Leitermann, and V. R. Subramanian, “Battery energy storage system (bess) and battery management system (bms) for grid-scale applications,” *Proceedings of the IEEE*, 102(6), p. 1014–1030, 2014.
- L. Zhang, H. Peng, Z. Ning, Z. Mu, and C. Sun, “Comparative research on rc equivalent circuit models for lithium-ion batteries of electric vehicles,” *Applied Sciences*, 2017.
- J. Jiang and C. Zhang, *Fundamentals and Application of Lithium-ion Batteries in Electric Drive Vehicles*. Willey, 2015.
- J. Rivera-Barrera, N. Muñoz-Galeano, and H. Sarmiento-Maldonado, “Soc estimation for lithium-ion batteries: Review and future challenges,” *Electronics*, vol. 6, 2017.
- Q. Yan and Y. Wang, “Predicting for power battery soc based on neural network,” *2017 36th Chinese Control Conference (CCC)*, vol. 86, pp. 4140–4143, 07 2017.
- R. Zhang, B. Xia, B. Li, L. Cao, Y. Lai, and W. Zheng, “A study on the open circuit voltage and state of charge characterization of high capacity lithium-ion battery under different temperature,” *Energies*, 2018.
- I. Baccouche, S. Jemmali, A. Mlayah, B. Manai, and N. E. B. Amara, “Implementation of an improved coulomb-counting algorithm based on a piecewise soc-ocv relationship for soc estimation of li-ion battery,” *International Journal of Renewable Energy Research*, vol. 8, pp. 178–187, 03 2017.

- C. Aoxia and S. P. K., “Advancement in battery technology: A state-of-the-art review,” *2016 IEEE Industry Applications Society Annual Meeting*, pp. 1–10, 2016.
- M. Cacciato, G. Nobile, G. Scarcella, and G. Scelba, “Real-time model-based estimation of soc and soh for energy storage systems,” *2015 IEEE 6th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, pp. 1–8, 2015.
- S. Park, T. Kang, J. Ahn, and S. Park, “Review of state-of-the-art battery state estimation technologies for battery management systems of stationary energy storage systems,” *Journal of Power Electronics*. 2020, 2020.
- D. Saji, P. S. Babu, and K. Ilango, “Soc estimation of lithium ion battery using combined coulomb counting and fuzzy logic method,” *2019 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)*, pp. 948–952, 2019.
- T. H. Wu, J. K. Wang, C. S. Moo, and A. Kawamura, “State-of-charge and state-of-health estimating method for lithium-ion batteries,” *2016 IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL)*, pp. 1–6, 2016.
- L. Lavigne, J. Sabatier, J. M. Francisco, F. Guillemard, and A. Noury, “Lithium-ion open circuit voltage (ocv) curve modelling and its ageing adjustment,” *Journal of Power Sources*, 324, p. 694–703, 2016.
- K. Movassagh, A. Raihan, B. Balasingam, and K. Pattipati, “A critical look at coulomb counting approach for state of charge estimation in batteries,” *Energies* 2021, 14, 4074, 2021.
- Y. Xing, E. W. M. Ma, K. L. Tsui, and M. Pecht, “Battery management systems in electric and hybrid vehicles,” *Energies*, 4(11), p. 1840–1857, 2011.
- K. S. Ng, C.-S. Moo, and Y.-C. H. Yi-Ping Chen and, “Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries,” *Applied Energy*, vol. 86, pp. 1506–1511, 2009.
- Z. Z. S. Hou, Y. Qian and J. Han, “Neural networks for state-of-charge estimation of li-ion batteries,” *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 729–732, 2021.

- Y. Hu and Z. Wang, “Study on soc estimation of lithium battery based on improved bp neural network,” *2019 8th International Symposium on Next Generation Electronics (ISNE)*, pp. 1–3, 2019.
- D. Simon, *Optimal state estimation : Kalman, H [infinity] and nonlinear approaches*. Wiley-Interscience, Hoboken, 2006.
- M. Liu, N. Cui, S. Liu, C. Wang, C. Zhang, and S. Gong, “Adaptive strong tracking unscented kalman filter based soc estimation for lithium-ion battery,” *2017 Chinese Automation Congress (CAC)*, pp. 1437–1441, 2017.
- P. Jain, S. Saha, and V. Sankaranarayanan, “Novel method to estimate soh of lithium-ion batteries,” *2021 Innovations in Energy Management and Renewable Resources(52042)*, pp. 1–5, 2021.
- B. Balagopal and M.-Y. Chow, “The state of the art approaches to estimate the state of health (soh) and state of function (sof) of lithium ion batteries,” *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference*, pp. 1302–1307, 2015.
- L. Ming, Z. Guo-Rong, L. Jiang-Hua, W.-J. Li, B. Li, and B.-Y. Luo, “A three-bridge topology for constant current and constant voltage output in ipt system,” *IEEE 2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*, pp. 1–5, 11 2018.
- V. Etacheri, R. Marom, R. Elazari, G. Salitra, and D. Aurbach, “Challenges in the development of advanced li-ion batteries: a review.” 2011.

Apêndice A

Códigos utilizados

O código do *frontend* pode ser encontrado em <https://github.com/GabrielFigueira1/bess-frontend>

Para executar-lo, abrir o arquivo "template/index.html".

O código do *backend* pode ser encontrado em <https://github.com/GabrielFigueira1/bess-backend> Para executar-lo, usar o comando "\$node index.js".

O código embarcado no NodeMCU é este:

```
#include <Arduino.h>
#include <stdlib.h>

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#include "ACS712.h"

#include <Wire.h>

#include <WebSocketsClient.h>

#include <Hash.h>

#define DEBUG
```

```
ACS712 sensor(ACS712_20A, A0);

// Battery data
static const float MAX_CAPACITY = 3800; //mAh
static const float NOMINAL_VOLTAGE = 3.7; //V

static float soc;
static float soh = 100;

static const float CC_TIMESTEP = 500; //ms
static float next_timestep;
static float last_timestep;

enum m_state
{
    discharge = 0,
    cc,
    cv
};

enum m_state state = discharge;

ESP8266WiFiMulti WiFiMulti;
WebSocketsClient websocket;

#define USE_SERIAL Serial

void websocketEvent(WStype_t type, uint8_t * payload, size_t length) {

    switch(type) {
        case WStype_DISCONNECTED:
            USE_SERIAL.printf("[WSc] Disconnected!\n");
```

```
        break;
    case WStype_CONNECTED: {
        USE_SERIAL.printf("[WSc] Connected to url: %s\n", payload);

        // send message to server when Connected
        websocket.sendTXT("Connected");
    }
    break;
    case WStype_TEXT:
        USE_SERIAL.printf("[WSc] get text: %s\n", payload);

        // send message to server
        // websocket.sendTXT("message here");
        break;
    case WStype_BIN:
        USE_SERIAL.printf("[WSc] get binary length: %u\n", length);
        hexdump(payload, length);

        // send data to server
        // websocket.sendBIN(payload, length);
        break;
        case WStype_PING:
            // pong will be send automatically
            USE_SERIAL.printf("[WSc] get ping\n");
            break;
        case WStype_PONG:
            // answer to a ping we send
            USE_SERIAL.printf("[WSc] get pong\n");
            break;
    }
}
```

```
//#####BMS Methods#####
```

```
void SendPayload(float ocv, float current, float _soc, float _soh, float timestamp)
{
    USE_SERIAL.println("Sending payload");
    String payload = "{\"ocv\":\"" + String(ocv, 6)+ "\",\"corrente\":\"" + String(current, 6)+ "\"}";
    #ifdef DEBUG
    USE_SERIAL.printf("Payload data: %s\n", payload.c_str());
    #endif
    websocket.sendTXT(payload);
}
```

```
float GetInitialSOC(float vlt)
```

```
{
    float voltage = vlt;
    float initialSOC;
    float a;
    float b;

    if (voltage < 3.452)
    {
        a = 26.55;
        b = 88.6;
    }
    else if (voltage < 3.508)
    {
        a = 125;
        b = 431.1;
    }
    else if (voltage < 3.595)
    {
```

```
    a = 149;
    b = 516.1;
}
else if (voltage < 3.676)
{
    a = 344;
    b = 1225;
}
else if (voltage < 3.739)
{
    a = 229.5;
    b = 800.9;
}
else if (voltage < 3.967)
{
    a = 111.9;
    b = 359.9;
}
else if (voltage < 4.039)
{
    a = 104.8;
    b = 332;
}
else
{
    a = 90.61;
    b = 274.7;
}

initialSOC = (a * voltage) - b;
USE_SERIAL.printf("Initial SOC is: %f\n", initialSOC);
return initialSOC;
```

```
}
/*
float GetSOH(float actualSOC, float voltage)
{
    float expectedSOC = GetInitialSOC(voltage); //Gets soc with ocv curve
    float actualSOH = (actualSOC/expectedSOC) * 100;
    return actualSOH;
}*/

void CoulombCounting(float current, float &soc, float t)
{
    soc = soc + (current/(MAX_CAPACITY*3600))*(t) * 100;
#ifdef DEBUG
    USE_SERIAL.printf("CC data: %f mA, %f , %f ms\n", current, soc, t);
#endif
}

void GetSensorsData(float &voltage, float &current)
{
    digitalWrite(D0, HIGH);
    int adcValue = analogRead(A0);
    delay(50);
    voltage = adcValue * 0.003222656 * 5;
    digitalWrite(D0, LOW);
    current = sensor.getCurrentDC();
    delay(50);
    USE_SERIAL.printf("current: %f, voltage: %f", current, voltage);
}

void SetDischargeMode()
{
    state = discharge;
}
```

```
    digitalWrite(D1, HIGH);
    digitalWrite(D2, LOW);
    digitalWrite(D3, LOW);
    USE_SERIAL.printf("Discharge Mode enabled\n");
}

void SetCVMode()
{
    state = cv;
    digitalWrite(D1, LOW);
    digitalWrite(D2, HIGH);
    digitalWrite(D3, LOW);
    USE_SERIAL.printf("CV Mode enabled\n");
}

void SetCCMode()
{
    state = cc;
    digitalWrite(D1, LOW);
    digitalWrite(D3, LOW);
    digitalWrite(D3, HIGH);
    USE_SERIAL.printf("CC Mode enabled\n");
}

void setup() {
    USE_SERIAL.begin(115200);
    USE_SERIAL.setDebugOutput(true);

    USE_SERIAL.println();
    USE_SERIAL.println();
    USE_SERIAL.println();

    pinMode(D0, OUTPUT);
```

```
pinMode(D1, OUTPUT);
pinMode(D2, OUTPUT);
pinMode(D3, OUTPUT);

digitalWrite(D0, LOW);
digitalWrite(D1, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, LOW);

sensor.calibrate();

for(uint8_t t = 4; t > 0; t--) {
  USE_SERIAL.printf("[SETUP] BOOT WAIT %d...\n", t);
  USE_SERIAL.flush();
  delay(1000);
}

WiFiMulti.addAP("WIFI_SSID", "WIFI_PASSWORD");

while(WiFiMulti.run() != WL_CONNECTED) {
  delay(100);
}

// server address, port and URL
WebSocket.begin("192.168.1.8", 3000, "/");

// event handler
WebSocket.onEvent(WebSocketEvent);

// use HTTP Basic Authorization this is optional remove if not needed
WebSocket.setAuthorization("user", "Password");
```



```
// try ever 5000 again if connection has failed
webSocket.setReconnectInterval(5000);

webSocket.enableHeartbeat(15000, 3000, 2);

delay(1000);

float m_current;
float m_voltage;
GetSensorsData(m_voltage, m_current);
soc = GetInitialSOC(m_voltage);
next_timestep = CC_TIMESTEP + millis();

if (soc < 30)
{
    SetCCMode();
}
else
{
    SetDischargeMode();
}
}

void loop() {
    float m_current;
    float m_voltage;
    GetSensorsData(m_voltage, m_current);

    Serial.println();
    webSocket.loop();
    delay(400);
}
```

```
//Serial.print("loop completed");
if (millis() >= next_timestep)
{
    float delta_t;
    float current_now = m_current;
    float ocv_now = m_voltage;
    float time_now = millis();
    delta_t = time_now - last_timestep;

    CoulombCounting(current_now, soc, delta_t);

    switch(state)
    {
        case discharge:
            if (soc < 30)
                SetCCMode();
            break;
        case cc:
            if (soc > 60)
                SetCVMode();
            break;
        case cv:
            if (soc > 80)
                SetDischargeMode();
            break;
    }

    SendPayload(ocv_now, current_now, soc, soh, time_now);
    last_timestep = time_now;

    next_timestep = CC_TIMESTEP + millis();
}
```

}