



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC

CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGEEL

DISSERTAÇÃO DE MESTRADO

MODELAGEM, SIMULAÇÃO E ESTIMAÇÃO DE CARGA DE BATERIAS DE LÍTIO COM IMPLEMENTAÇÃO DE UM CARREGADOR DE BATERIAS

CHRYSTIAN LENON REMES

JOINVILLE, 2016

CHRYSTIAN LENON REMES

**MODELAGEM, SIMULAÇÃO E ESTIMAÇÃO DE
CARGA DE BATERIAS DE LÍTIO COM
IMPLEMENTAÇÃO DE UM CARREGADOR DE
BATERIAS**

Dissertação de Mestrado, apresentada como requisito para obtenção do Grau de Mestre em Engenharia Elétrica pela Universidade do Estado de Santa Catarina, no Centro de Ciências Tecnológicas.

Orientador: Sérgio Vidal Garcia Oliveira

**JOINVILLE
2016**

R386m Remes, Chrystian Lenon
Modelagem, simulação e estimação de carga de baterias de lítio com implementação de um carregador de baterias / Chrystian Lenon Remes . – 2016.
355p. : il. ; 21 cm

Orientador: Sérgio Vidal Garcia Oliveira

Bibliografia: p. 218-228

Dissertação (mestrado) – Universidade do Estado Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Engenharia Elétrica, Joinville, 2016.

1. Duplo-Forward. 2. Controle ótimo. 3. Modelagem de Baterias. 4. Estimação do SOC. I . Oliveira, Sérgio Vidal Garcia. II. Universidade do Estado de Catarina. Programa de Pós- Graduação em Engenharia Elétrica. III. Título.

CDD 621.31242 – 23.ed.

CHRYSTIAN LENON REMES

MODELAGEM, SIMULAÇÃO E ESTIMAÇÃO DE CARGA DE

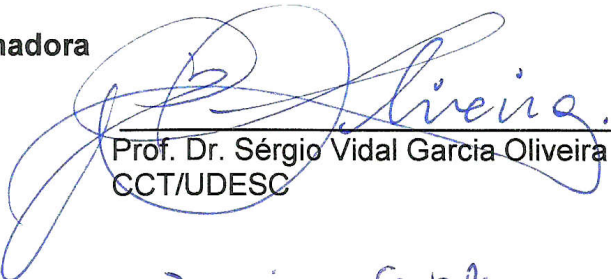
BATERIAS DE LÍTIO COM IMPLEMENTAÇÃO DE UM

CARREGADOR DE BATERIAS

Dissertação apresentada ao Curso de Programa de Pós-Graduação em Engenharia Elétrica como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração “Sistemas Eletroeletrônicos”.

Banca Examinadora

Orientador:



Prof. Dr. Sérgio Vidal Garcia Oliveira
CCT/UDESC

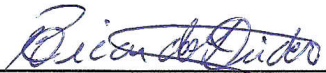
Membros



Profa. Dra. Mariana Santos Matos Cavalcante
CCT/UDESC



Prof. Dr. Joselito Anastácio Heerd
CCT/UDESC



Prof. Dr. Ricardo Lüders
UTFPR

Joinville, SC, 24 de fevereiro de 2016.

Dedico esta dissertação à minha esposa, Jéssica, aos meus pais, Paulo e Cida, e à minha irmã Paula, pelo suporte e amor incondicional.

AGRADECIMENTOS

Primeiramente, agradeço igualmente à minha esposa Jéssica, aos meus pais Paulo e Cida, e à minha irmã Paula pelo carinho, amor, companheirismo e também por todo o suporte ao longo de toda esta etapa e também das anteriores, apoiando minhas escolhas e me ajudando sempre que possível, nunca deixando de acreditar em mim. Com certeza, este trabalho não existiria sem vocês.

Aos meus familiares por todo o carinho e por entenderem o motivo de não poder estar mais tão próximo de todos, em especial aos meus queridos avós, André e Eva (*in memoriam*), ao meu avô Jorge e à minha avó Natalina, e também ao meu tio Amarildo, que sempre acreditou no meu trabalho e ajudou no meu desenvolvimento pessoal e profissional.

Ao meu orientador, Sérgio V. G. Oliveira, que sempre confiou no meu trabalho e nas minhas decisões, me auxiliando sempre que possível com suas orientações, na área técnica, e também, pessoalmente, não medindo esforços para estar presente em grandes momentos da minha vida.

Ao professor Alessandro L. Batschauer, por estar disponível a ceder um pouco do seu disputado tempo para me ajudar em algumas dúvidas de controle, e também, por todos os ensinamentos valiosos tanto na graduação, quanto no mestrado, além das boas histórias e do companheirismo.

Aos professores Joselito A. Heerd, Marcello Mezaroba e Yales R. de Novaes por também estarem sempre dispostos a ajudar e a compartilhar as suas experiências.

À professora Mariana S. M. Cavalca, por me apresentar, me guiar e me fazer gostar da área de controle digital e de controle ótimo, e também por todos os trabalhos desenvolvidos, além da sua disposição em ajudar sempre.

Aos meus amigos Gustavo Lambert, Rubens T. Hock Jr. e Felipe Stein por todos os cafezinhos e *pão de queijos* inspiradores e engraçados, além de toda a ajuda técnica que foi fundamental neste desenvolvimento, na eletrônica, placas de circuito impresso e *snubbers*, respectivamente.

Ao Felipe J. Zimann por todos os cafezinhos e *breaks* engraçados, mas também pelas boas discussões de controle, pelos trabalhos desenvolvidos juntos e pelas boas risadas e causos.

A todos os bolsistas do nPEE pelas ajudas e pelos momentos de descontração, em especial ao Murilo B. Rosa e ao Henrique F. Souza, que acompanharam mais de perto e ajudaram o meu trabalho.

A todos os demais professores da graduação e do mestrado pelas ricas experiências.

Aos meus amigos e agora mestres Heric D. Farias e Renan Sebem, por todos os almoços e jantares, pelos trabalhos juntos, viagens, risadas, indignações, ideias, pensamentos, enfim, por compartilharem as tantas experiências boas e ruins desta época, sempre com grande companheirismo aliado à contribuição pessoal.

A todos os meus amigos de longa data, em especial ao Alexandre Coccia dos Santos e ao Maycon Senna, pela amizade e apoio ininterruptos mesmo com a longa distância.

*“...a new day will dawn for those
who stand long and the forest will
echo with laughter”*

Jimmy Page e Robert Plant

RESUMO

As baterias, em especial as de Lítio, vêm se tornando cada vez mais integradas ao cotidiano das pessoas através de inúmeras aplicações, sejam estas celulares, notebooks ou mesmo os promissores Veículos Elétricos. Diante deste cenário, técnicas de estimação do Estado de Carga (SOC) das baterias têm chamado atenção dos pesquisadores, que buscam o aprimoramento dos métodos de estimação já existentes. Este trabalho traz um estudo geral sobre as técnicas de estimação do SOC já disponíveis na literatura sintetizando os principais pontos de vista apontados por diversos autores, além da implementação da estimação do SOC através do Filtro de Kalman Estendido, método já conhecido na literatura, e de duas metodologias aqui propostas baseadas no Filtro de Kalman Adaptativo, juntamente com os resultados e análises de cada uma das metodologias de estimação. Além da estimação do SOC, esta dissertação também aborda a modelagem e simulação de baterias, necessárias ao estudo de estimação do SOC, e a construção de um conversor Duplo-Forward com um controlador LQG, utilizado como carregador de baterias.

Palavras-chave: Duplo-Forward, Controle Ótimo, Modelagem de Baterias, Estimação do SOC.

ABSTRACT

Batteries, especially Lithium ones, have been become more integrate to the daily life of people through a huge amount of applications, like cellphones, laptops or even the promising Electric Vehicles. In this scenario, techniques to estimate the State Of Charge (SOC) of batteries have gained attention of researchers, which aim to improve the existent estimation methods. This work brings a general study about the SOC estimation techniques available in the literature summarizing the main points of view of different authors, beyond the implementation of SOC estimation through Extended Kalman Filter, a known method in the literature, and through two methodologies proposed based on Adaptive Kalman Filter, along with the results and analyzes of each one of the estimation techniques. Beyond of the SOC estimation, this master thesis also discusses modeling and simulation of batteries, which are needed to the SOC estimation study, and the development of a Two-Switch Forward Converter (2SFC) with a LQG controller, used as a battery charger.

Keywords: 2SFC, Optimal Control, Battery Modeling, SOC Estimation.

LISTA DE FIGURAS

FIGURA 1	– Curva Característica de v_{oc} por SOC de uma bateria Li-ion do tipo NCA.	49
FIGURA 2	– Esquemático de um carregador de baterias comercial conectado à rede	55
FIGURA 3	– Esquemático do conversor Duplo-Forward conectado a um arranjo de baterias	57
FIGURA 4	– Formas de onda simuladas da corrente no indutor (azul) e da tensão no capacitor de saída (vermelho)	60
FIGURA 5	– Imagem do protótipo do conversor Duplo-Forward projetado	61
FIGURA 6	– Formas de onda experimentais de corrente no indutor (azul - CH2) e tensão no capacitor de saída (amarelo - CH1)	61
FIGURA 7	– Formas de onda de ondulação de corrente no indutor (azul claro - CH2) e de tensão no capacitor de saída (azul escuro - CH1)	62
FIGURA 8	– Representação da carga de uma célula NCR 18650 com perfil CC-CV	66
FIGURA 9	– Esquema genérico do circuito de condicionamento de sinais	68
FIGURA 10	– Esquemático do arranjo de baterias.	69
FIGURA 11	– Esquemático completo do sistema de carga de baterias.	72
FIGURA 12	– Circuito do conversor Buck.	74
FIGURA 13	– Etapa 1 de funcionamento do conversor Buck.	75
FIGURA 14	– Etapa 2 de funcionamento do conversor Buck.	75
FIGURA 15	– Diagrama de blocos do LQG com configuração em cascata.	82
FIGURA 16	– Ilustração do deslocamento de uma partícula $x(i)$ pelo espaço de busca	92
FIGURA 17	– Diagrama de Bode da malha de interna de corrente: Planta sem compensador ($G_{id}(z)$ em azul), planta com compensador em malha aberta ($FTMA_i$ em verde) e planta compensada em malha fechada ($FTMF_i$ em vermelho).	100

FIGURA 18	– Diagrama de Bode da malha de externa de tensão: Planta sem compensador ($G_{vi}(z)$ em azul), planta com compensador em malha aberta ($FTMA_v$ em verde) e planta compensada em malha fechada ($FTMF_v$ em vermelho).	101
FIGURA 19	– Diagrama de Bode do KF (corrente): Planta em malha aberta ($G_{id}(z)$ em azul), planta com KF em malha aberta ($FTMA_{iKF}$ em verde) e planta com KF em malha fechada ($FTMF_{iKF}$ em vermelho).	103
FIGURA 20	– Diagrama de Bode do KF (tensão): Planta em malha aberta ($G_{vi}(z)$ em azul), planta com KF em malha aberta ($FTMA_{vKF}$ em verde) e planta com KF em malha fechada ($FTMF_{vKF}$ em vermelho).	104
FIGURA 21	– Resposta simulada do sistema a um degrau de referência de tensão de 0 a 4,2 V sem limitação de corrente (controle de tensão com carga de 0,8 Ω).	107
FIGURA 22	– Resposta simulada do sistema a um degrau de referência de tensão de 0 a 4,2 V com limitação de corrente em 3 A (controle de corrente com carga de 0,8 Ω).	108
FIGURA 23	– Resposta simulada do sistema a um degrau de carga -75% (de 0,8 Ω para 3,2 Ω)	109
FIGURA 24	– Resposta simulada do sistema a um degrau de carga +300% (de 3,2 Ω para 0,8 Ω)	111
FIGURA 25	– Resposta experimental do sistema a um degrau de referência de tensão de 0 a 4,2 V sem limitação da corrente e carga resistiva de 0,8 Ω CH1 (tensão, azul) = 1V/div CH2 (corrente, verde) = 1A/div tempo = 2ms/div .	113
FIGURA 26	– Resposta do sistema a um degrau de referência de tensão de 0 a 4,2 V com limitação da corrente em 3 A e carga resistiva de 0,8 Ω CH1 (tensão, azul) = 500mV/div CH2 (corrente, verde) = 500mA/div tempo = 2ms/div	114
FIGURA 27	– Carga de uma única bateria: tensão do conversor (v_o vermelho), tensão da bateria (v_b azul) e corrente de carga (i_b verde). Condição inicial de carga $SOC_o \approx 0$	116

FIGURA 28	– Carga de duas baterias em série: tensão do conversor (v_o vermelho), tensão das baterias em série (v_b azul) e corrente de carga (i_b verde). Condição inicial de carga $SOC_o \approx 0,2$ para ambas as baterias	117
FIGURA 29	– Carga de um arranjo com 6 baterias (3 em paralelo em série com mais 3 em paralelo): tensão do conversor (v_o vermelho), tensão das baterias em série (v_b azul) e corrente de carga (i_b verde). Condição inicial de carga $SOC_o \approx 0,35$ para todas as baterias	118
FIGURA 30	– Modelo genérico de uma bateria representado por elementos de circuitos elétricos (ECM) . .	121
FIGURA 31	– Representação de um ECM com capacitância variável no lugar de uma fonte de tensão . . .	122
FIGURA 32	– Analogia física entre um capacitor comum e um capacitor dependente da própria tensão .	123
FIGURA 33	– Representação do conversor Duplo-Forward conectado à uma bateria	124
FIGURA 34	– ECM de uma bateria considerando parâmetros variáveis no tempo	127
FIGURA 35	– Curva característica genérica de uma bateria Li-ion do tipo NCA e seu ajuste polinomial de 7ª ordem	128
FIGURA 36	– Curva da capacidade máxima normalizada da bateria em função da sua corrente normalizada	129
FIGURA 37	– Diagrama de blocos do simulador de baterias utilizando o modelo de corrente	135
FIGURA 38	– Resultados da estimação dos parâmetros $\hat{R}_o$ (azul), $\hat{R}$ (vermelho), $\hat{C}$ (roxo), juntamente com a corrente i_b (verde) da bateria	137
FIGURA 39	– Resultados da constante de tempo $\hat{R}\hat{C}$ estimada (vermelho) e da tensão v_{rc} estimada (azul), juntamente com a corrente da bateria (verde)	138
FIGURA 40	– Formas de onda das tensões v_t (azul) e v_{oc} (vermelho) da bateria, juntamente com seu SOC (preto) e sua corrente i_b (verde)	140

FIGURA 41	– Erros de estimação utilizando-se os métodos de <i>Coulomb Counting</i> (verde e azul) e Estimação pela Tensão Terminal (vermelho) . . .	146
FIGURA 42	– Diagrama de blocos do estimador com EKF .	160
FIGURA 43	– Diagrama de blocos do estimador com KF adaptativo	163
FIGURA 44	– Diagrama de blocos do estimador com KF adaptativo e costimação da curva $v_{oc} \times SOC$. . .	170
FIGURA 45	– Parâmetros R_o (azul), R (vermelho) e C (roxo) estimados mantidos constantes para estimação do SOC com EKF ($SOC_o = 0$), juntamente com a corrente i_b (verde)	175
FIGURA 46	– Formas de onda da tensão $\hat{v}_{rc}$ (azul) e de $\hat{RC}$ (vermelho), juntamente da corrente i_b (verde), para o caso do EKF ($SOC_o = 0$)	176
FIGURA 47	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo EKF, juntamente com a corrente i_b (verde) ($SOC_o = 0$)	177
FIGURA 48	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) para o caso do EKF ($SOC_o = 0$)	178
FIGURA 49	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo EKF, com a corrente i_b (verde) ($SOC_o = 0, 35$)	179
FIGURA 50	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) para o caso do EKF ($SOC_o = 0, 35$)	180
FIGURA 51	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo EKF, juntamente com a corrente i_b (verde) ($SOC_o = 0, 7$)	181
FIGURA 52	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) para o caso do EKF ($SOC_o = 0, 7$)	182
FIGURA 53	– Formas de onda experimentais do SOC (gráfico superior) e de v_t (gráfico inferior) estimados pelo EKF ($SOC \approx 0$)	185
FIGURA 54	– Formas de onda experimentais do SOC (gráfico superior) e de v_t (gráfico inferior) estimados pelo EKF ($SOC = 0, 35$)	186

FIGURA 55	– Formas de onda simuladas do SOC (azul) e de v_t (vermelho) estimados por um EKF, reproduzindo-se as mesmas condições do teste experimental de estimação com EKF, juntamente com o erro de estimação do SOC (verde) ($SOC = 0,35$)	187
FIGURA 56	– Formas de onda experimentais do SOC (gráfico superior) e de v_t (gráfico inferior) estimados pelo EKF ($SOC = 0,7$)	188
FIGURA 57	– Parâmetros R_o (azul), R (vermelho) e C (roxo) mantidos constantes para estimação com KF adaptativo, juntamente com a corrente i_b (verde) ($SOC_o = 0$)	192
FIGURA 58	– Formas de onda da tensão $\hat{v}_{rc}$ (azul) e de $\hat{RC}$ (vermelho), juntamente com a corrente i_b (verde) para o caso do KF adaptativo ($SOC_o = 0$)	193
FIGURA 59	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) para o caso do KF adaptativo, juntamente com a corrente i_b (verde) ($SOC_o = 0$)	194
FIGURA 60	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) pelo KF adaptativo ($SOC_o = 0$)	195
FIGURA 61	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo KF adaptativo, juntamente com a corrente i_b (verde) ($SOC_o = 0,35$)	196
FIGURA 62	– Forma de onda do SOC estimado (vermelho) pelo KF adaptativo e do seu erro de estimação (verde) ($SOC_o = 0,35$)	197
FIGURA 63	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho), juntamente com a corrente i_b (verde), para o caso do KF adaptativo ($SOC_o = 0,7$)	198
FIGURA 64	– Forma de onda do SOC estimado (vermelho) pelo KF adaptativo e do seu erro de estimação (verde) ($SOC_o = 0,7$)	199
FIGURA 65	– Formas de onda estimadas dos parâmetros $\hat{R}_o$ (azul), $\hat{R}$ (vermelho) e $\hat{C}$ (roxo) no caso do KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)	203

FIGURA 66	– Formas de onda da tensão $\hat{v}_{rc}$ (azul) e de $\hat{RC}$ (vermelho), juntamente com a corrente i_b (verde) no caso do KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)	204
FIGURA 67	– Formas de onda de $\hat{\beta}_1$ (roxo), $\hat{\beta}_o$ (verde) e $\hat{v}_{oc}$ (vermelho) estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)	205
FIGURA 68	– Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$, juntamente com a corrente i_b (verde) ($SOC_o = 0$)	206
FIGURA 69	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) com o KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)	207
FIGURA 70	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) com o KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0,35$)	208
FIGURA 71	– Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) utilizando o KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$) ($SOC_o = 0,7$)	209
FIGURA 72	– Formas de onda de $\hat{\beta}_1$ (roxo) e de $\hat{\beta}_o$ (verde), além da tensão $\hat{v}_{oc}$ (vermelho), estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0,35$)	210
FIGURA 73	– Formas de onda de $\hat{\beta}_1$ (roxo) e de $\hat{\beta}_o$ (verde), além da tensão $\hat{v}_{oc}$ (vermelho), estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0,7$)	211
FIGURA 74	– Imagens do indutor produzido	237
FIGURA 75	– Imagens do transformador produzido	241
FIGURA 76	– Imagem do dissipador dos interruptores	246
FIGURA 77	– Imagem do dissipador dos diodos de saída	249
FIGURA 78	– Esquemático do LPF na entrada do optoacoplador. Os componentes utilizados encontram-se destacados em preto.	254

FIGURA 79	– Esquemático da alimentação isolada do optoacoplador, com os componentes utilizados destacados em preto.	255
FIGURA 80	– Representação do transformador não ideal . .	258
FIGURA 81	– Imagem do ensaio do dissipador dos interruptores	262
FIGURA 82	– Imagem do ensaio do dissipador dos diodos .	263
FIGURA 83	– Esquemático da PCI do Retificador e do Conversor Duplo-Forward	268
FIGURA 84	– Projeto da PCI do Retificador e do Conversor Duplo-Forward	269
FIGURA 85	– Formas de onda da tensão V_{gs} (rosa e amarelo) e da corrente de <i>gate</i> (azul e vermelho) dos interruptores	271
FIGURA 86	– Forma de onda da tensão V_{gs} experimental (amarelo - CH1)	272
FIGURA 87	– Formas de onda simuladas das correntes nos semicondutores: I_{sw1} (rosa), I_{sw2} (amarelo), I_{din1} (azul claro), I_{din2} (preto), I_{do1} (azul) e I_{do2} (vermelho)	273
FIGURA 88	– Formas de onda simuladas das tensões nos semicondutores: V_{sw1} (rosa), V_{sw2} (amarelo), V_{din1} (azul claro), V_{din2} (preto), V_{do1} (azul) e V_{do2} (vermelho)	274
FIGURA 89	– Formas de onda experimentais da tensão sobre os interruptores S_{w1} (esquerda) e S_{w2} (direita) .	275
FIGURA 90	– Formas de onda experimentais da tensão sobre os diodos D_{o1} (esquerda) e D_{o2} (direita) . . .	275
FIGURA 91	– Formas de onda simuladas das tensões e correntes do transformador: V_p (rosa), V_s (amarelo), I_p (azul), I_s (vermelho)	276
FIGURA 92	– Forma de onda experimental demonstrando o efeito do <i>Ground Bounce</i> (azul claro - CH2) .	277
FIGURA 93	– Potências médias instantâneas simuladas nos semicondutores: P_{sw1} (rosa), P_{sw2} (amarelo), P_{din1} (azul claro), P_{din2} (preto), P_{do1} (azul) e P_{do2} (vermelho)	278
FIGURA 94	– Esquema de medição do sinal v_{in} para condicionamento	279

FIGURA 95	– Esquema de medição do sinal v_o para condicionamento.	280
FIGURA 96	– Esquema de medição do sinal i_L para condicionamento.	282
FIGURA 97	– Esquema de medição do sinal v_b para condicionamento.	283
FIGURA 98	– Esquema de condicionamento do sinal de PWM.	285
FIGURA 99	– Esquema de medição do sinal i_b para condicionamento.	286
FIGURA 100	– Esquema de condicionamento do sinal de temperatura do LM35.	288

LISTA DE TABELAS

TABELA 1	– Principais características de alguns tipos de baterias	44
TABELA 2	– Tecnologias de baterias utilizadas pelas montadoras de automóveis	45
TABELA 3	– Especificações do conversor Duplo-Forward	58
TABELA 4	– Parâmetros utilizados no algoritmo PSO	94
TABELA 5	– Parâmetros utilizados no algoritmo PSO do simulador de baterias	134
TABELA 6	– Parâmetros de simulação do EKF	173
TABELA 7	– Parâmetros de simulação do KF adaptativo	189
TABELA 8	– Parâmetros de simulação do KF adaptativo com coestimação da curva $v_{oc} \times SOC$	200
TABELA 9	– Definições para o projeto do indutor	234
TABELA 10	– Dados experimentais do indutor manufaturado	238
TABELA 11	– Definições para o projeto do transformador	238
TABELA 12	– Cálculo das perdas do transformador	240
TABELA 13	– Parâmetros do transformador executado e perdas totais	241
TABELA 14	– Dados de catálogo do MOSFET IRF740	244
TABELA 15	– Dados de catálogo do diodo MBR1545CT	247
TABELA 16	– Definições e dados para projeto do <i>snubber</i>	250
TABELA 17	– Dados de catálogo do diodo MBR1545CT	258
TABELA 18	– Medições de indutância do transformador	259
TABELA 19	– Medições do ensaio do dissipador dos interruptores	262
TABELA 20	– Medições do ensaio do dissipador dos diodos	263
TABELA 21	– Dados do circuito de condicionamento de v_{in}	280
TABELA 22	– Dados do circuito de condicionamento de v_o	281
TABELA 23	– Dados do circuito de condicionamento de i_L	283
TABELA 24	– Dados do circuito de condicionamento de v_b	284
TABELA 25	– Dados do circuito de condicionamento de i_b	286
TABELA 26	– Dados do circuito de condicionamento de v_b	287

LISTA DE ABREVIATURAS E SIGLAS

2SFC	Two-Switch Forward Converter
ADC	Analog to Digital Converter
AMPOP	Amplificador Operacional
ANN	Artificial Neural Network
BEV	Battery Electric Vehicle
BMS	Battery Management System
CA	Corrente Alternada
CA-CC	Corrente Alternada para Corrente Contínua
CC	Corrente Contínua
CC-CC	Corrente Contínua para Corrente Contínua
CC-CV	Constant Current Constant Voltage
CCM	Continuous Conduction Mode
CKF	Collective Kalman Filter
DCM	Discontinuous Conduction Mode
DSC	Digital Signal Controller
DUKFST	Dual Spherical Unscented Kalman Filter
ECM	Electric Circuit Model
EKF	Extended Kalman Filter
EMS	Energy Management System
EV	Electric Vehicle
FNN	Fuzzy Neural Network

GA Genetic Algorithm

HEV Hybrid Electric Vehicle

HIF H-Infinity Filter

KF Kalman Filter

Li Lítio

LPF Low-Pass Filter

LQ Linear Quadratic

LQE Linear Quadratic Estimator

LQG Linear Quadratic Gaussian

LQR Linear Quadratic Regulator

LS Least Squares

MOSFET Metal-Oxide Semiconductor Field Effect Transistor

MWLS Moving Window Least Squares

NiCd Níquel-Cádmio

NiMH Níquel-Hidreto Metálico

PCI Placa de Circuito Impresso

PFC Power Factor Correction

PHEV Plug-in Hybrid Electric Vehicle

PID Proporcional Integral Derivativo

PSO Particle Swarm Optimization

PWM Pulse-Width Modulation

RLS Recursive Least Squares

SOC State Of Charge

SOH State Of Health

SPKF Sigma Point Kalman Filter

SVM Support Vector Machines

UKF Unscented Kalman Filter

USABC United States Advanced Battery Consortium

V2G Vehicle to Grid

ZOH Zero-Order Holder

ZVS Zero Voltage Switching

LISTA DE SÍMBOLOS

β_o Coeficiente linear da curva $v_{oc} \times SOC$. $[V]$

β_1 Coeficiente angular (derivada) da curva $v_{oc} \times SOC$. $[V]$

B_{max} Máxima densidade de fluxo magnético. $[T]$

B_{max_t} Máxima densidade de fluxo magnético no transformador.
 $[T]$

C Capacitância do efeito dinâmico da bateria. $[F]$

C_o Capacitor do filtro de saída. $[\mu F]$

C_{fc} Capacitância do filtro capacitivo do retificador. $[\mu F]$

Δi_L Ondulação de corrente no indutor. $[A]$

D_{in1} Diodo da entrada 1

D_{in2} Diodo da entrada 2

D_{max} Razão cíclica máxima do conversor

D_{min} Razão cíclica mínima do conversor

D_{o1} Diodo de condução direta

D_{o2} Diodo de roda-livre

ΔT_L Variação de temperatura do indutor. $[K]$

ΔT_t Variação de temperatura do transformador. $[K]$

Δv_o Ondulação de tensão no capacitor de saída. $[V]$

ESR_{fc} Capacitância do filtro capacitivo do retificador. $[m\Omega]$

ESR_{max} Resistência série máxima do capacitor do filtro de saída.
 $[m\Omega]$

f_a Frequência de amostragem. $[kHz]$

f_s Frequência de chaveamento. $[kHz]$
 i_b Corrente da bateria. $[A]$
 $I_{L_{ef}}$ Corrente eficaz no indutor. $[A]$
 $I_{L_{pk}}$ Corrente de pico no indutor. $[A]$
 $I_{o_{med}}$ Corrente média de saída. $[A]$
 J_{max} Máxima densidade corrente. $[A/cm^2]$
 J_{max_t} Máxima densidade corrente do transformador. $[A/cm^2]$
 k_{pt} Fator de ocupação do primário
 k_w Fator de ocupação da janela
 k_{wt} Fator de ocupação da janela do transformador
 l_{gap} Entreferro do indutor. $[mm]$
 L_m Indutância magnetizante do transformador. $[mH]$
 L_o Indutor do filtro de saída. $[\mu H]$
 L_{lk_1} Indutância de dispersão do primário. $[\mu H]$
 L_{lk_2} Indutância de dispersão do secundário. $[\mu H]$
 μ_o Permeabilidade magnética no vácuo. $[H/m]$
 n_{cond_L} Número de condutores do indutor
 N_L Número de espiras do indutor
 N_p Número de espiras do primário do transformador
 N_s Número de espiras do secundário do transformador
 N_t Relação de transformação
 $P_{kf}(k|k)$ Representação genérica da matriz de covariância do vetor de erros de estimação no instante k dadas as observações até o instante k

P_{LQ} Matriz solução da equação algébrica de Ricatti para os problemas do tipo LQ

$P_{d.L}$ Potência dissipada no enrolamento do indutor. [W]

$P_{d.t}$ Potência dissipada no enrolamento do transformador. [W]

$P_{in_{max}}$ Potência máxima na entrada do conversor

$P_{nucleo.L}$ Potência dissipada no núcleo magnético do indutor. [W]

$P_{nucleo.t}$ Potência dissipada no núcleo magnético do transformador. [W]

$P_{o_{max}}$ Potência máxima de saída do conversor

$Q_{kf}(k)$ Representação genérica da matriz de covariância dos estados do processo no instante k

Q_{LQ} Representação genérica da matriz de ponderação dos estados dos elementos do LQG

$Q_{LQ.a}$ Representação genérica da matriz de ponderação dos estados dos elementos do LQG estendida

Q_{max} Capacidade máxima de carga da bateria. [mAh]

R Resistência do efeito dinâmico da bateria. [Ω]

$R_{kf}(k)$ Representação genérica da matriz de covariância da saída do sistema no instante k

R_{LQ} Representação genérica da matriz de ponderação da ação de controle dos elementos do LQG

$R_{LQ.a}$ Representação genérica da matriz de ponderação da ação de controle dos elementos do LQG estendida

η Rendimento do conversor

η_t Rendimento esperado do transformador

η_v Rendimento da tensão do conversor

R_L Resistência do enrolamento do indutor. $[m\Omega]$
 R_o Resistência Série da Bateria. $[\Omega]$
 R_1 Resistência do enrolamento primário do transformador. $[\Omega]$
 R_{fc} Impedância equivalente do conversor. $[\Omega]$
 R_2 Resistência do enrolamento secundário do transformador. $[m\Omega]$
 $R_{th_{nucleo.L}}$ Resistência térmica do núcleo de ferrite. $[K/W]$
 $R_{th_{nucleo.t}}$ Resistência térmica do núcleo do transformador. $[K/W]$
 $S_{total_{cond}}$ Seção total do condutor do indutor. $[mm^2]$
 Sw_1 Interruptor 1
 Sw_2 Interruptor 2
 T_a Período de amostragem. $[\mu s]$
 $u(k)$ Representação genérica do sinal da ação de controle na amostra k
 $V_{f_{do}}$ Queda de tensão de condução direta nos diodos de saída. $[V]$
 V_{in} Tensão de entrada. $[V]$
 $V_{in_{max}}$ Tensão de entrada máxima. $[V]$
 $V_{in_{min}}$ Tensão de entrada mínima. $[V]$
 v_o Tensão de saída. $[V]$
 v_{oc} Tensão de circuito aberto da bateria. $[V]$
 $V_{o_{max}}$ Tensão de saída máxima. $[V]$
 $V_{o_{min}}$ Tensão de saída mínima. $[V]$
 v_{rc} Tensão dinâmica da bateria. $[V]$
 v_t Tensão terminal da bateria. $[V]$

$X(k)$ Representação genérica do vetor de estados na amostra k

$\hat{X}(k)$ Representação genérica da estimação do vetor de estados na amostra k

$\hat{X}(k|k)$ Representação genérica da estimação do vetor de estados na amostra k dadas as informações até a amostra k

$x(i)$ Representação genérica da partícula i do algoritmo PSO

z_d Constante de tempo discreta do modelo da bateria

SUMÁRIO

1	INTRODUÇÃO	42
1.1	SISTEMAS DE GERENCIAMENTO DE BATERIAS	45
1.2	ALGORITMOS DE ESTIMAÇÃO DO SOC E SOH DE BATERIAS	47
1.2.1	Estimação do Estado de Carga SOC	47
1.2.2	O Estado de Saúde (SOH)	50
1.3	OBJETIVOS	50
1.4	JUSTIFICATIVA	51
1.5	HIPÓTESES	52
1.6	ORGANIZAÇÃO DO TRABALHO	53
2	O CARREGADOR DE BATERIAS	54
2.1	DUPLO-FORWARD COMO CARREGADOR DE BA- TERIAS	56
2.1.1	Especificações do Conversor Duplo-Forward . .	57
2.2	RESULTADOS DO CONVERSOR EM MALHA ABERTA	58
2.3	CONSIDERAÇÕES SOBRE O CONVERSOR DUPLO- FORWARD	63
3	CONTROLE, PROTEÇÃO E CONDICIONA- MENTO DE SINAIS	65
3.1	CIRCUITOS DE CONDICIONAMENTO DE SINAIS	67
3.2	O ARRANJO DE BATERIAS E SUAS PROTEÇÕES	69
3.3	CONFIGURAÇÃO DO DSC	70
3.4	PROTEÇÕES POR SOFTWARE	71
3.5	ESQUEMÁTICO COMPLETO	71
3.6	PROJETO DE CONTROLE	72
3.6.1	Modelagem do Conversor Duplo-Forward . . .	74
3.6.2	Estratégias de Controle Ótimo	77
3.6.3	Representação incremental em espaço de estados	79
3.6.4	Projeto do LQG	81
3.6.4.1	Análise de controlabilidade, observabilidade e estabi- lidade	87
3.6.5	Encontrando as matrizes de ponderação	88

3.6.6	Particle Swarm Optimization	89
3.6.7	Resumo das etapas de projeto do controlador	95
3.7	RESULTADOS DO CONTROLE DO CARREGADOR DE BATERIAS	97
3.7.1	Resposta em frequência dos controladores	97
3.7.2	Análise do controlador por simulação	105
3.7.3	Resultados experimentais do controlador	112
3.8	CONSIDERAÇÕES SOBRE O CONTROLE DO CONVERSOR	119
4	MODELAGEM E SIMULAÇÃO DE BATERIAS	120
4.1	MODELAGEM DA BATERIA PARA SIMULAÇÃO DO PROJETO DO CONTROLADOR	123
4.2	MODELAGEM POR CIRCUITOS ELÉTRICOS COM PARÂMETROS VARIÁVEIS	126
4.2.1	Modelo de corrente da bateria	130
4.2.2	Modelo de tensão da bateria	131
4.3	SIMULAÇÃO DO MODELO DINÂMICO DA BATERIA	132
4.4	RESULTADOS DE SIMULAÇÃO DO MODELO DINÂMICO DA BATERIA	135
4.5	CONSIDERAÇÕES FINAIS SOBRE A MODELAGEM E SIMULAÇÃO DE BATERIAS	141
5	ESTIMAÇÃO DE CARGA DE BATERIAS	143
5.1	ESTIMAÇÃO DE CARGA POR COULOMB COUNTING E PELA TENSÃO TERMINAL	143
5.2	CONTEXTUALIZAÇÃO DE MÉTODOS DE ESTIMAÇÃO DO SOC PRESENTES NA LITERATURA	147
5.3	O FILTRO DE KALMAN E O FILTRO DE KALMAN ESTENDIDO	153
5.3.1	O Filtro de Kalman	154
5.3.2	O Filtro de Kalman Estendido	157
5.4	ESTIMAÇÃO DO SOC COM FILTRO DE KALMAN ESTENDIDO	158

5.4.1	Análise de Observabilidade do Sistema Linearizado	161
5.5	ESTIMAÇÃO DO SOC COM FILTRO DE KALMAN A PARTIR DE UM MODELO ADAPTATIVO . . .	161
5.5.1	Identificação dos parâmetros resistivos e capacitivos da bateria	163
5.6	ESTIMAÇÃO DO SOC COM FILTRO DE KALMAN A PARTIR DE UM MODELO ADAPTATIVO DE ORDEM 3	167
5.6.1	A obtenção da nova variável de estado da bateria	167
5.6.2	Análise de Observabilidade do Sistema Linearizado de ordem 3	170
5.7	RESULTADOS DA ESTIMAÇÃO DO SOC DE BATERIAS	171
5.7.1	Simulação de Estimação do SOC com EKF . .	172
5.7.2	Resultados experimentais de estimação do SOC com EKF	183
5.7.3	Simulação de Estimação do SOC com KF Adaptativo	189
5.7.4	Simulação de Estimação do SOC com KF adaptativo com coestimação da curva voc pela carga	200
5.8	CONSIDERAÇÕES FINAIS SOBRE A ESTIMAÇÃO DE CARGA DE BATERIAS	212
6	CONCLUSÕES E TRABALHOS FUTUROS .	214
	REFERÊNCIAS	218
	APÊNDICES	229
A	DIMENSIONAMENTO DO CONVERSOR DUPLO-FORWARD	229
A.1	DIMENSIONAMENTO DO RETIFICADOR	229
A.1.1	Dimensionamento do filtro capacitivo do retificador	229
A.1.2	Dimensionamento dos diodos do retificador . .	230

A.2	PROJETO DO CONVERSOR DUPLO-FORWARD	231
A.2.1	Relação de transformação	231
A.2.2	Projeto do filtro de saída	232
A.2.3	Projeto dos elementos magnéticos	233
A.2.3.1	Projeto e Execução do Indutor de Saída	234
A.2.3.2	Projeto e Execução do Transformador	237
A.2.4	Dimensionamento dos semicondutores	242
A.2.4.1	Dimensionamento dos interruptores	242
A.2.4.2	Dimensionamento dos diodos do estágio de entrada	243
A.2.4.3	Dimensionamento dos diodos do estágio de saída	243
A.2.4.4	Dimensionamento Térmico dos Semicondutores	244
A.2.5	Projeto dos circuitos de grampeamento	249
A.2.6	Dimensionamento do circuito de comando	251
A.2.6.1	Potência do circuito de comando e obtenção da resistência de <i>gate</i>	251
A.2.6.2	Escolha dos optoacopladores	252
A.2.6.3	Filtro Passa Baixas para condicionamento dos optoacopladores	253
A.2.6.4	Fontes auxiliares isoladas para o circuito de comando	254
A.2.7	Dimensionamento das proteções e circuitos auxiliares	255
B	INFORMAÇÕES COMPLEMENTARES DO ESTÁGIO DE POTÊNCIA DO CONVERSOR	257
B.1	OBTENÇÃO DOS PARÂMETROS DO TRANSFORMADOR DE ALTA FREQUÊNCIA	257
B.2	OBTENÇÃO DA RESISTÊNCIA TÉRMICA DOS DISSIPADORES	260
B.2.1	Resultados do dissipador dos interruptores	261
B.2.2	Resultados do dissipador dos diodos de saída	263
B.3	LEDS INDICADORES DE OPERAÇÃO	264
B.4	DEMAIS RESULTADOS DO CONVERSOR DUPLO-FORWARD	265
B.4.1	Resultados do projeto da placa de circuito impresso	265
B.4.2	Resultados complementares do conversor	270

C	INFORMAÇÕES COMPLEMENTARES SOBRE OS CIRCUITOS DE CONDICINAMENTO	279
C.1	CIRCUITOS DE CONDICIONAMENTO UTILIZADOS	279
C.1.1	Condicionamento do sinal da tensão de entrada do conversor	279
C.1.2	Condicionamento do sinal da tensão de saída do conversor	280
C.1.3	Condicionamento do sinal da corrente no indutor	281
C.1.4	Condicionamento dos sinais de tensão das baterias	283
C.1.5	Condicionamento do sinal de PWM (saída) . .	284
C.2	CIRCUITOS DE CONDICIONAMENTO NÃO UTILIZADOS	285
C.2.1	Condicionamento dos sinais de corrente das baterias	285
C.2.2	Condicionamento dos sinais de temperatura das baterias	287
	ANEXOS	289
A	ANEXOS DO DESENVOLVIMENTO DO CON- VERSOR DUPLO-FORWARD	289
A.1	CÓDIGOS PARA OBTENÇÃO DAS INDUTÂNCIA DO TRANSFORMADOR	289
B	ANEXOS DO DESENVOLVIMENTO DO CON- TROLADOR	290
B.1	CÓDIGOS PARA CONFIGURAÇÃO DO DSC . . .	290
B.1.1	Configuração do ADC	290
B.1.2	Configuração do PWM	292
B.1.3	Configuração das GPIOs	302
B.1.4	Configuração das Interrupções	303
B.1.5	Configuração de sistema	304
B.1.6	Configuração do Watchdog	307

B.2	CÓDIGOS PARA OBTENÇÃO DAS MATRIZES Q E R E DO LQG	307
B.2.1	Obtenção das matrizes de peso	307
B.2.2	Obtenção dos ganhos do LQG	310
B.3	FUNÇÕES DO CONTROLADOR LQG IMPLEMEN- TADOS EM C	314
B.4	CÓDIGO MAIN.C DO CONTROLADOR E SUAS PROTEÇÕES SEM ESTIMAÇÃO DO SOC	316
C	CÓDIGOS PARA SIMULAÇÃO DA BATERIA	321
C.1	CÓDIGO PRINCIPAL DO SIMULADOR DE BATE- RIAS	321
C.2	FUNÇÕES DE INICIALIZAÇÃO DO ALGORITMO DE SIMULAÇÃO DE BATERIAS	324
C.2.1	Inicialização do perfil de corrente da bateria .	324
C.2.2	Inicialização da curva de capacidade da bateria em função da corrente	324
C.2.3	Inicialização do campo de busca do algoritmo PSO	324
C.2.4	Inicialização de ruídos para análise de medições de grandezas da bateria	325
C.3	FUNÇÕES PARA CÁLCULO DO SOC E DA TEN- SÃO DE CIRCUITO ABERTO	325
C.3.1	Cálculo do SOC através da corrente	325
C.3.2	Cálculo do SOC através da tensão	325
C.3.3	Cálculo da tensão de circuito aberto	325
C.4	FUNÇÕES DE EXECUÇÃO DO ALGORITMO PSO PARA BATERIAS	325
C.4.1	Algoritmo PSO para encontrar parâmetros no modelo de corrente	325
C.5	FUNÇÕES PARA CÁLCULO DO FITNESS DOS ALGORITMOS PSO	327
C.5.1	Cálculo da função fitness do PSO para o mo- delo de corrente	327
D	CÓDIGOS PARA ESTIMAÇÃO DO SOC . .	328

D.1	CÓDIGO PRINCIPAL DE SIMULAÇÃO DE ESTIMAÇÃO DO SOC COM EKF	328
D.2	CÓDIGO PRINCIPAL DE SIMULAÇÃO DE ESTIMAÇÃO DO SOC COM KF ADAPTATIVO	332
D.2.1	Função de inicialização do campo de busca dos parâmetros do estimador do SOC	338
D.2.2	Função do estimador PSO de parâmetros do modelo	338
D.2.3	Função de cálculo da função de custo da estimação de parâmetros com PSO	339
D.3	CÓDIGO PRINCIPAL DE SIMULAÇÃO DE ESTIMAÇÃO DO SOC COM KF ADAPTATIVO E COESTIMAÇÃO DA CURVA VOC	340
D.3.1	Função de inicialização do campo de busca dos parâmetros do estimador do SOC com coestimação da curva voc	346
D.3.2	Função do estimador PSO de parâmetros do modelo com coestimação da curva voc	347
D.3.3	Função de cálculo da função de custo da estimação de parâmetros com PSO com coestimação da curva voc	348
D.4	CÓDIGO PRINCIPAL ESTIMAÇÃO DO SOC COM EKF UTILIZADO NO DSC	349
D.4.1	Função para estimação de carga utilizando Coulomb Counting	353
D.4.2	Função para estimação de carga utilizando EKF	354

1 INTRODUÇÃO

“Lithium-ion batteries have received an increasing interest from the scientific community. (...) they offer higher energy efficiency and power density. Moreover, several other advantages, (...) make them good candidates for many applications such as laptops, mobile phones, and electric vehicles.”

Hicham Chaoui et al. [1]

O armazenamento de energia tem ocupado um papel de destaque tanto na indústria quanto no cotidiano da população. É cada vez mais comum a utilização de tecnologias de armazenamento de energia, dentre as quais as baterias possuem um papel privilegiado. Existem ainda outras tecnologias de armazenamento menos difundidas, como os supercapacitores, células a combustível, volantes de inércia (comumente conhecidos como *flywheels*) e o armazenamento de energia através do ar comprimido. Porém, especialmente quando se trata de mobilidade, as baterias ainda assumem grande vantagem por possuírem grande densidade de energia por massa e por volume, o que garante que sejam opções mais viáveis quando relacionadas à grandes massas girantes, por exemplo. Sua desvantagem está atrelada ao fato de possuir baixa durabilidade para aplicações que necessitam de grande quantidade de ciclos [2,3].

Dentre muitos fatores, é principalmente devido a sua alta capacidade de armazenamento que as baterias são largamente utilizadas em equipamentos eletrônicos portáteis como laptops e telefones celulares, *Battery Electric Vehicles* (BEV - Veículos Elétricos a Bateria), *Hybrid Electric Vehicles* (HEV - Veículos Híbridos Elétricos), além dos sistemas de geração distribuída e *Smart Grids*.

Diante da grande quantidade de aplicações que utilizam baterias, os BEVs e HEVs merecem atenção, uma vez que o setor de transportes atualmente é extremamente dependente de recursos petrolíferos no Brasil e no mundo, e espera-se que, no atual ritmo de consumo do petróleo, haja uma grande escassez deste recurso em apenas algumas décadas [4,5]. É importante ainda frisar que a queima de combustíveis fósseis é tida como grande responsável pelo acúmulo de gases de efeito

estufa na atmosfera terrestre, fato que é apontado por muitos cientistas como o principal fator desencadeante do aquecimento global [5–7]. Ainda, diante de uma possível escassez do petróleo e do desenvolvimento de tecnologias sustentáveis, o mercado e os pesquisadores no mundo todo tem depositado grandes esforços em veículos menos dependentes de petróleo, como os BEVs e HEVs [8,9].

Os BEVs têm como vantagens a reduzida emissão de gases de efeito estufa, menor emissão de ruídos audíveis e maior eficiência energética [10,11]. Porém, as desvantagens inerentes das tecnologias atuais de baterias, como o custo elevado, baixa tempo de vida útil, além de autonomia reduzida e elevado tempo de recarga em relação aos veículos a combustão, quando comparado com seu tempo de abastecimento, são motivações para se investir no seu aprimoramento. Ainda, fatores políticos, a exemplo da Alemanha e Estados Unidos que vêm estabelecendo metas futuras que impõe uma maior utilização de energias renováveis e que consequentemente elevam a participação das gerações de energia solares e eólicas, alavancam o uso de baterias e juntamente o interesse no seu desenvolvimento. O órgão americano United States Advanced Battery Consortium (USABC) estabeleceu diversas metas de médio e longo prazo para baterias, tais como a redução dos custos (\$/kWh), aumento das densidade de energia (Wh/L) e de potência (W/L), aumento no tempo de vida útil (em anos) e etc. [8]. Com todos estes fatores é notório que há um grande clamor pelo avanço das baterias em geral.

Dentre as tecnologias de baterias utilizadas, destacam-se as baterias de Chumbo-Ácido (Pb-Acid), Níquel-Cádmio (NiCd), Níquel-Hidreto Metálico (NiMH) e as baterias de Lítio (Li) [9]. A Tabela 1 mostra as principais características destes tipos de baterias.

É possível perceber que as tecnologias NiMH e Lítio são superiores às demais, especialmente com relação a densidade energética e ao número de ciclos máximos disponíveis, e por isso estas são as tecnologias mais utilizadas atualmente em diversas aplicações.

A Tabela 2 mostra as tecnologias de baterias utilizadas em BEVs e HEVs de diversas montadoras de automóveis. É possível notar que as baterias do tipo NiMH e Lítio são as tecnologias utilizadas no setor, com predominância das baterias de Lítio. Além disso, nos trabalhos mais recentes, pesquisadores tem focado maior atenção às baterias

Tabela 1 – Principais características de alguns tipos de baterias

Índice	Pb-Acid	NiCd	NiMH	Li
Tensão da célula (V)	2	1.2	1.2	3.6
Energia específica (Wh/kg)	30-35	30-35	35-65	100-160
Densidade energética (Wh/L)	130	150	200	270-360
Número de ciclos máximo (-20% de Q_n)	200	500	500-1000	500-2000
Efeito memória	Não	Sim	Sim (fraco)	Não

Fonte: Adaptado de Zhou, Y. e Li, X. [12]

de Lítio [10, 13, 14]. Tendo por base estes fatos, esta pesquisa irá se focar apenas em baterias de Lítio.

Neste trabalho, os termos e abreviaturas serão utilizados na língua inglesa, após sua apresentação na língua portuguesa, visando uma maior facilidade ao leitor em se familiarizar com as literaturas da área que são predominantemente publicadas em inglês.

A seguir, serão discutidos brevemente os *Battery Management Systems* (BMS - Sistemas de Gerenciamento de Bateria), que são peça fundamental em diversas aplicações que fazem uso de bateria, e os indicadores *State Of Charge* (SOC - Estado de Carga) e *State Of Health* (SOH - Estado de Saúde) de baterias, os quais em geral são rotinas contidas nos BMSs. Na sequência, serão apresentados os objetivos deste trabalho, bem como sua justificativa e hipóteses.

Tabela 2 – Tecnologias de baterias utilizadas pelas montadoras de automóveis

Compania	País	Modelo do Veículo	Tecnologia de Bateria
GM	EUA	Chevy-Volt Saturn HEV	Li NiMH
Toyota	Japão	Prius, Lexus	NiMH
Ford	EUA	Fusion, MKZ Escape PHEV	NiMH Li
Honda	Japão	Civic, Insight	NiMH
Hyundai	Coréia do Sul	Sonata	Li
Chrysler	EUA	Chrysler EV	Li
BMW	Alemanha	X6 Mini E (2012)	NiMH Li
BYD	China	E6	Li
Daimler Benz	Alemanha	ML450, S400 Smart EV (2010)	NiMH Li
Mitsubishi	Japão	iMiEV (2010)	Li
Nissan	Japão	Altima Leaf EV (2010)	NiMH Li
Tesla	EUA	Roadster (2009)	Li
Think	Noruega	Think EV	Li, Na

Fonte: Adaptado de Young et. al. [8]

1.1 SISTEMAS DE GERENCIAMENTO DE BATERIAS

O Sistema de Gerenciamento de Baterias, ou BMS, é um sistema que monitora e controla o funcionamento de baterias visando aumentar sua vida útil, eficiência de operação, segurança e promover a

estimação de indicadores da bateria [9].

Em *Smart Grids*, os BMS e/ou os *Energy Management Systems* (EMS - Sistemas de Gerenciamento de Energia) fazem a estimação de indicadores como o SOC e o SOH, que estão intimamente ligados ao desempenho da rede. Isto porque em um *Smart Grid*, por exemplo, deve-se conhecer qual o nível de carga presente em uma dada bateria para que então seja definido se esta poderá ou não ser utilizada para injetar energia na rede, além de informar qual deverá ser a corrente de carga ou descarga de tal bateria.

Para os EVs, existe uma preocupação especial com a segurança dos bancos de baterias contra eventuais choques mecânicos, riscos de incêndio e na estimação precisa do SOC da bateria. Afinal, a bateria do veículo não pode explodir com uma simples colisão em baixas velocidades, e também um usuário não pode ficar parado à beira da rodovia simplesmente devido à uma estimação errônea do SOC da bateria de seu BEV. Tais eventos são gerenciados também pelo BMS. Além destes fatores, deve-se atentar ao fato de que muitas das baterias, especialmente as de íons de Lítio (Li-ion), não podem sofrer cargas ou descargas muito profundas, o que facilmente poderia ocorrer em viagens muito longas. Desta forma, circuitos de proteção contra sobrecargas e descargas profundas são de suma importância.

Nota-se que o BMS é uma figura muito importante para que a operação de baterias seja segura e eficiente. Dentre as principais funções de um BMS estão [9]:

- Monitoramento de Células: as baterias necessitam de uma alta precisão nas leituras de tensão e corrente para uma boa estimação do SOC, além de tais leituras servirem de base para os sistemas de proteção. Além disso, a temperatura também é um fator importante, muitas vezes servindo de base para sistemas de controle térmico da bateria em BMSs mais complexos.
- Proteção e Segurança: o BMS é o encarregado de verificar se a bateria está operando de forma segura e de atuar os sistemas de proteção caso seja necessário.

- Estimaco de Indicadores: a estimaco do SOC e do SOH tambm  uma funo executada pelo BMS.
- Balanceamento de clulas: em sistemas de maior complexidade, o BMS tambm agrega a funo de coordenar o balanceamento de carga entre clulas.

Estando ento a estimaco dos indicadores SOC e SOH dentre as responsabilidade dos BMS, estas sero descritas a seguir.

1.2 ALGORITMOS DE ESTIMACO DO SOC E SOH DE BATERIAS

A estimaco de indicadores como o SOC e o SOH  um importante recurso presente nos BMS. A utilizaco do SOC e do SOH, por exemplo,  vital tanto para um melhor gerenciamento da taxa de carga/descarga visando o aumento de vida til da bateria, quanto para um melhor controle dos EMS presentes em *Smart Grids*, principalmente nos casos em que se utilizam tecnologias como a *Vehicle-to-Grid* (V2G - Veculo para Rede), onde os EVs tambm podem fornecer energia  rede em momentos de pico de demanda. Alm disso, o SOC  uma informaco que deve estar sempre acessvel ao usurio do BEV, que se baseia no SOC para determinar quando  o melhor momento para uma recarga. Assim como o SOC, o SOH tambm deve ser uma informaco acessvel ao usurio, que poder determinar qual o tempo de vida til restante de sua bateria e quando esta dever ser trocada com base neste indicador [9].

1.2.1 Estimaco do Estado de Carga SOC

O SOC representa a quantidade de carga til de uma bateria em um determinado instante de tempo, podendo ser medido apenas de maneira indireta. Pode-se definir o SOC conforme a expresso (1.1).

$$SOC(t) = SOC_o - \frac{1}{Q_{max}(t)} \int i(t) dt \quad (1.1)$$

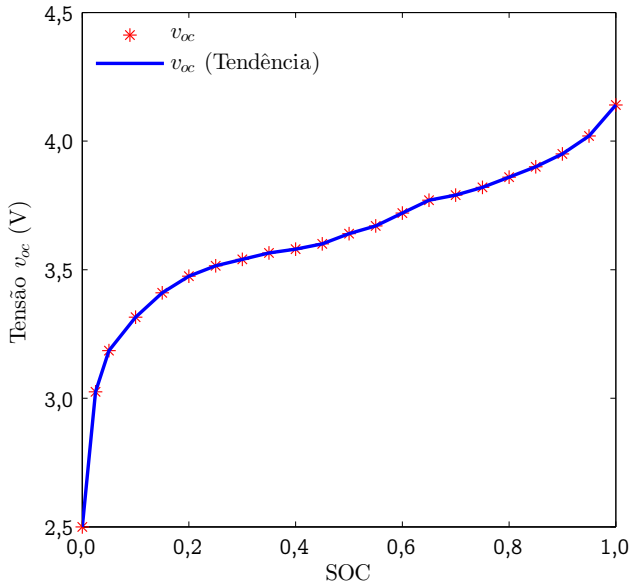
Onde SOC_o representa o estado inicial do SOC no momento t_o , $i(t)$ representa a corrente da bateria (positiva durante a descarga e negativa durante a carga) e $Q_{max}(t)$ representa a capacidade máxima de carga da bateria no instante t [15]. Nota-se, portanto, que o $SOC(t)$ é uma grandeza adimensional.

Na estimação do SOC, baseando-se na expressão (1.1), deriva-se um dos métodos mais utilizados e conhecidos da literatura, que é chamado de *Coulomb Counting* (Contagem de Coulombs, em português) ou *Ah Counting* (Contagem de Ah, em português). Este método simples realiza a integração da corrente para obter a carga da bateria e consequentemente, seu SOC. Logo, além da simplicidade, este método tem como vantagem a necessidade apenas da medição de corrente. Sua grande desvantagem está atrelada à dificuldade de se ter a condição inicial SOC_o , além da acumulação de possíveis erros de medição polarizados, que fazem com que o algoritmo de estimação passe a divergir do valor real ao longo do tempo [16–18].

Outro método bastante simples na estimação do SOC é a sua obtenção através da curva característica de Tensão de Circuito Aberto (v_{oc}) pelo SOC. A Figura 1 mostra um exemplo de curva $v_{oc} \times SOC$, que possui características extremamente não lineares. Outro comentário importante é que a tensão v_{oc} só pode ser obtida após longos períodos de descanso da bateria, isto porque a bateria possui uma tensão associada ao seu comportamento dinâmico, conhecido como efeito de relaxação. Desta forma, a tensão terminal da bateria (v_t) diverge do valor da tensão v_{oc} durante sua operação ou mesmo algumas horas depois de sua operação, acarretando em erros de estimação do SOC. Assim, este método não é recomendado para a estimação online do SOC, além de demandar o conhecimento da curva característica v_{oc} por SOC [16, 18, 19].

Métodos baseados em Inteligência computacional, como *Artificial Neural Networks* (ANN - Redes Neurais Artificiais), *Lógica Fuzzy* e *Fuzzy Neural Networks* (FNN - Redes Neurais Fuzzy) vem sendo utilizados também na estimação do SOC [18, 20–24]. Apesar de conseguirem resultados muito precisos na estimação, tais métodos necessitam de grandes volumes de dados para o treinamento das redes. Além disso, o envelhecimento da bateria traz modificações internas nos seus parâme-

Figura 1 – Curva Característica de v_{oc} por SOC de uma bateria Li-ion do tipo NCA.



Fonte: produção do próprio autor, 2016

tros, acarretando em perda de precisão dos modelos de estimação das ANNs e FNNs, que precisariam de novos conjuntos de dados e de uma nova etapa de treinamento para adaptação à tais efeitos.

Uma das soluções que vem sendo extremamente abordadas recentemente são estimadores baseados no Filtro de Kalman (KF) e suas variantes, como o *Extended Kalman Filter* (EKF - Filtro de Kalman Estendido) [25–30], o *Unscented Kalman Filter* (UKF - Filtro de Kalman “Unscented”) [18, 30–33] e o *H-Infinity Filter* (HIF - Filtro H-infinito) [16, 30, 34].

Nota-se que são diversas as possibilidades e abordagens utilizadas na estimação do SOC, mostrando o interesse da comunidade científica no tema. Outro tema que também é bastante abordado é a estimação do SOH, descrita a seguir.

1.2.2 O Estado de Saúde (SOH)

O SOH é um indicador que monitora qual é a capacidade máxima de carga de uma data bateria ao longo do tempo, em relação a sua capacidade máxima nominal. Este indicador é definido na maioria das literaturas pela equação (1.2):

$$SOH(t) = \frac{Q_{max}(t)}{Q_n} \quad (1.2)$$

Onde o $Q_{max}(t)$ é a capacidade máxima da bateria em um determinado instante de tempo t e Q_n é a capacidade de carga máxima nominal especificada pelo fabricante. Apesar desta ser a definição mais aceita recentemente pela literatura, outros autores defendem que a estimação do SOH deva ser feita utilizando-se em conjunto outras grandezas, como por exemplo o perfil de corrente de carga e descarga utilizado ao longo do tempo, sua temperatura de operação ao longo do tempo, entre outros. Isto porque, analogamente com os seres humanos, a saúde de uma bateria também depende do seu histórico de vida, não sendo possível assim estabelecer o nível de saúde de uma pessoa sem conhecer fatores como a sua idade, seus comportamentos e suas atividades [9]. Percebe-se que há ainda muito a ser explorado no que diz respeito à estimação do SOH.

Tendo sido comentados sobre os principais aspectos que dizem respeito às baterias, segue-se então com os objetivos deste trabalho.

1.3 OBJETIVOS

Serão descritos aqui os objetivos do presente trabalho, divididos entre objetivo geral e objetivos específicos.

Objetivo Geral: realizar um estudo aprofundado sobre as técnicas de estimação do SOC com a execução métodos de estimação do SOC em baterias de Lítio e de um carregador de baterias controlado.

Objetivos Específicos:

- Desenvolver um conversor Duplo-Forward como carregador de baterias, visando manipular o SOC das mesmas.

- Utilizar técnicas de controle ótimo no projeto de controle do conversor.
- Abordar as diferentes técnicas de estimação do SOC existentes na literatura.
- Realizar a estimação do SOC através do Filtro de Kalman Estendido, o qual é um dos métodos já conhecidos na literatura.
- Propor, uma nova estratégia de estimação do SOC de baterias de Lítio, dispondo do mínimo de informações sobre a mesma, utilizando técnicas de Inteligência Computacional, a qual é descrita na seção 5.6.

Para que tais objetivos fossem atingidos, foi necessária uma ampla revisão da literatura existente sobre a estimação do SOC em baterias, além de conhecimentos em eletrônica de potência e controle para o desenvolvimento do carregador de baterias. Também foram necessários conhecimentos sobre o projeto e funcionamento de algoritmos de estimação baseados KF, além do domínio de técnicas de inteligência computacional visando o desenvolvimento de uma nova estrutura de estimação do SOC.

Destacados os objetivos do trabalho, mostra-se a seguir a justificativa que motiva o desenvolvimento do mesmo. As contribuições do trabalho serão detalhadas no capítulo 6.

1.4 JUSTIFICATIVA

Conforme foi abordado no início desta seção, a estimação do SOC é de extrema necessidade para qualquer aplicação que utilize baterias, sendo um fator crítico para BEVs, HEVs e *Smart Grids*. Além disso, existem diversas propostas para a estimação do SOC, sendo que muitas delas não são capazes de fornecer uma estimativa precisa ou necessitam de uma vasta gama de parâmetros e comportamentos de uma dada bateria para a realização de uma boa estimação, o que muitas vezes leva a necessidade de testes e ensaios com longos tempos de duração. Tendo em mente tais problemas, é notória a necessidade de

se conhecer mais sobre as técnicas existentes de estimação do SOC, buscando o aprimoramento das mesmas. Tal aprimoramento seria de grande valia para indústrias, consumidores e para a academia em geral, o que justifica o desenvolvimento deste trabalho.

Do ponto de vista acadêmico, a revisão de literaturas que tratam a estimação do SOC será importante para que o autor amplie seus conhecimentos sobre o tema, e que tais conhecimentos também sejam encontrados mais facilmente por outros pesquisadores, principalmente os nativos em língua portuguesa, que também estejam começando sua pesquisa em estimação do SOC. Além disso, o desenvolvimento de um carregador de baterias e de ao menos um método de estimação do SOC trazem ao autor mais desenvoltura com seu tema de pesquisa e trazem aos demais pesquisadores um guia para que também possam realizar tais desenvolvimentos. Destaca-se ainda que o projeto de controle do conversor através do controle ótimo serve de auxílio tanto para o autor, na sua empreitada de estimação do SOC, a qual se utiliza de técnicas similares, quanto para o leitor se familiarizar com a aplicação desta metodologia dentro da área de eletrônica de potência.

A seguir, são tratadas as hipóteses que cercam esta dissertação.

1.5 HIPÓTESES

As hipóteses deste trabalho foram fundamentadas basicamente em informações disponíveis na literatura sobre baterias e com base em algumas observações vistas nos experimentos. Estas foram adotadas visando restringir o foco do trabalho, evitando levar em consideração efeitos secundários. As hipóteses utilizadas nesta dissertação são:

- O ambiente possui pouca variação de temperatura, de forma que esta possa ser desprezada. Não serão levados em consideração quaisquer efeitos devidos a variação de temperatura na estimação do SOC ou dos parâmetros da bateria.
- Como a capacidade máxima da bateria e consequentemente seu SOH variam muito lentamente no tempo (na ordem de meses ou anos), sua influência não será considerada. A capacidade

máxima de carga da bateria será assumida constante e igual a sua capacidade nominal.

- Os efeitos do envelhecimento da bateria sobre a estimação dos parâmetros da mesma são ignorados.
- Assume-se que o sentido da corrente (carga ou descarga) não altera quaisquer características da bateria, ou seja, os efeitos de histerese são completamente desprezados.
- As variações paramétricas entre diferentes baterias são desprezadas. Esta afirmação diz respeito a qualquer parâmetro da bateria, como suas resistências e capacitâncias internas, curva característica de Tensão v_{oc} por SOC e sua quantidade de carga máxima nominal Q_n .
- Os efeitos de desbalanceamento de carga em arranjos com mais de uma bateria são desconsiderados.

Sendo feitas tais hipóteses, na sequência é apresentada a organização do trabalho, que encerra este capítulo introdutório.

1.6 ORGANIZAÇÃO DO TRABALHO

Dada a contextualização acerca da importância das baterias e dados os objetivos, justificativa e hipóteses deste trabalho, são mostrados a seguir o seu desenvolvimento, acompanhados dos seus resultados e da conclusão. No capítulo 2, é apresentado o conversor Forward, utilizado como carregador de baterias, para manipulação do SOC das mesmas, enquanto que no capítulo 3 encontra-se o seu projeto de controle. No capítulo 4 é discutida a modelagem de baterias juntamente com a simulação de tal modelo, o qual será utilizado posteriormente no trabalho. No capítulo 5, é feita uma abordagem sobre os diferentes métodos de estimação do SOC disponíveis na literatura, além de trazer o desenvolvimento e os resultados dos métodos de estimação implementados, seguidos pela conclusão do trabalho, apresentada no capítulo 6. Após a conclusão, estão disponíveis os Apêndices e Anexos que compõem a presente dissertação.

2 O CARREGADOR DE BATERIAS

“...os VEH [Veículos Elétricos Híbridos] podem desempenhar um papel tecnológico importante auxiliando a trazer maturidade de tecnologias para motores, para eletrônica de potência e para baterias, podendo reduzir os custos dessas tecnologias.”

Schmitt, F. W. [35]

Em veículos que utilizam a eletricidade, tanto em *Plug-in Hybrid Electric Vehicles* (PHEV - Veículos Elétricos *Plug-in*), HEVs e BEVs, existe a necessidade do processamento de energia elétrica. No caso dos HEVs, o motor a combustão pode funcionar como gerador que alimenta as baterias, havendo a necessidade de adequação nos níveis de tensão e corrente fornecidos pelo gerador. A mesma ideia vale para a frenagem regenerativa, que converte a energia cinética das rodas em energia elétrica, que também deve ser armazenada em baterias e também necessita de processamento. Em BEVs em geral, o processamento de energia também é necessário, seja para adequar a energia de Corrente Alternada (CA) fornecida pela rede elétrica para Corrente Contínua (CC) e/ou também para adequar os níveis de tensão e corrente CC para a bateria. Desta forma, a eletrônica de potência possui um papel fundamental tanto para as baterias quanto para os BEVs e HEVs, adequando os níveis de tensão e corrente necessários para cada aplicação.

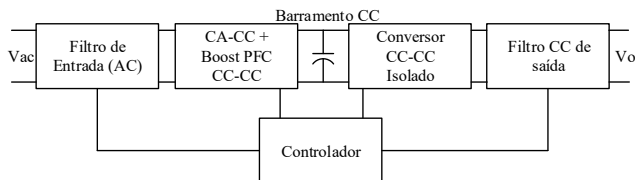
O principal papel da eletrônica de potência é realizar o processamento de energia com a maior eficiência e segurança possível. Para tal, pesquisadores da área de eletrônica de potência têm pesquisado novas tecnologias de semicondutores, novas topologias de conversores estáticos, além do aprimoramento de tecnologias já existentes e das estratégias de controle, otimizando eficiência, segurança e também o tempo de vida útil de componentes, aliando tudo isto à redução de custo. E isto também se aplica aos conversores estáticos utilizados como carregadores de baterias.

Existem diversos tipos de carregadores de baterias, que podem ser divididos entre carregadores *on-board*, embutidos nos próprios carros, e *off-board*, comumente utilizados em estações de carga maio-

res. Ainda, os carregadores podem ser divididos entre carregadores do tipo CA e CC. Os carregadores CA, apesar de poderem ser conectados diretamente a rede, possuem a necessidade de um conversor CA-CC, preferivelmente com *Power Factor Correction* (PFC - Correção de Fator de Potência). Já os carregadores CC são montados em locais fixos, necessitando de estações dedicadas, mas podendo fornecer maior potência, diminuindo o tempo de recarga das baterias. Em [36] é possível ver a distribuição de tais estações de carga nos Estados Unidos e que estas já apresentam números significativos, como é o caso do estado da Califórnia, sendo que muitas destas estações são pertencentes às próprias montadoras, a exemplo da Nissan e da Tesla. É importante também destacar que tanto os carregadores CA quanto os carregadores CC são divididos em níveis de acordo com a sua potência [5, 37, 38].

Os carregadores *on-board* em geral são constituídos de dois estágios, um CA-CC e outro CC-CC, conforme mostra a Figura 2. Dentre as topologias de conversores usados no estágio CA-CC em BEVs, são destacadas por [5] o Boost PFC Interleaved, o Bridgeless Boost PFC e o Bridgeless Interleaved Boost PFC. Com relação aos conversores CC-CC utilizados no segundo estágio, [5] cita o conversor Ponte Completa ZVS (*Zero Voltage Switching*) com Filtro de Saída Capacitivo, o conversor Ponte Completa ZVS Interleaved com Dobrador de Tensão e o conversor Ponte Completa com LLC Ressonante.

Figura 2 – Esquemático de um carregador de baterias comercial conectado à rede



Fonte: Adaptado de Williamson, S. S., Rathore, A. K. e Musavi, F. [5]

Nesta pesquisa, será desenvolvido um carregador de baterias de pequeno porte, visando carregar pequenos arranjos de baterias. O obje-

tivo deste desenvolvimento é que o carregador possa ser utilizado como ferramenta no estudo de arranjos com uma ou mais células, dispostas em série e/ou paralelo, mas sem que haja dependência de grandes bancos de baterias. Assim, reduz-se drasticamente o custo de compra de um banco de baterias para o custo de apenas algumas células, além de reduzir a complexidade de projeto, o custo do carregador de baterias e de facilitar o manuseio operacional e envolver menores riscos de segurança. É importante salientar que estudos sobre o rendimento do conversor e de seus componentes, bem como ensaios térmicos do conversor em operação, características de comutação e outras análises características da eletrônica de potência estão fora do escopo deste trabalho, que se foca apenas no funcionamento do conversor para sua utilização como ferramenta. Os resultados deste carregador de baterias encontram-se publicados em [39].

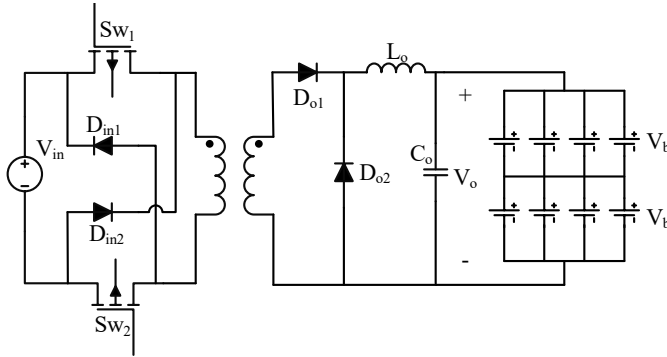
A seguir, será feita uma descrição do conversor Duplo-Forward utilizado neste trabalho, bem como a descrição detalhada das suas etapas de projeto. Serão expostos ainda os passos de projeto e dimensionamento de componentes os quais foram de difícil acesso ao autor, como a escolha de optoacopladores e de seu condicionamento, o dimensionamento de elementos de proteção, dentre outros. As estratégias de controle do conversor e as lógicas de proteção por software serão descritas no Capítulo 3. Outras informações sobre o conversor Duplo-Forward podem ser encontradas em [40–43], que apresentam versões mais elaboradas do conversor Duplo-Forward.

2.1 DUPLO-FORWARD COMO CARREGADOR DE BATERIAS

O *Two-Switch Forward Converter* (2SFC - Conversor Duplo-Forward), também conhecido como Forward a Duas Chaves, é um conversor isolado baseado no conversor Forward, um derivado do conversor Buck, possuindo assim característica abaixadora de tensão e sendo um conversor adequado para potências menores que 500 W [39, 44]. Como esta dissertação irá tratar de arranjos de baterias com poucas células (até oito), este se mostrou adequado para operação como carregador, além de ser um conversor robusto, apresentando boa controlabilidade e simplicidade, e dispensando a necessidade de um enrolamento de des-

magnetização, como no caso do conversor Forward [39,45]. Ainda, por ser isolado, este conversor agrega maior segurança ao usuário. A Figura 3 mostra o conversor 2SFC.

Figura 3 – Esquemático do conversor Duplo-Forward conectado a um arranjo de baterias



Fonte: Remes, C. L., Rosa, M. B. e Oliveira, S. V. G [39]

O funcionamento do conversor 2SFC é bem simples, e as etapas de operação deste conversor podem ser divididas em apenas três, uma representando o período de tempo em que ambos interruptores Sw_1 e Sw_2 estão ligados, outras duas quando ambos interruptores estão desligados e a carga é alimentada pelo filtro de saída.

Tendo-se um conhecimento básico acerca do conversor 2SFC, passa-se a seguir às especificações de projeto do mesmo, que irão delinear suas etapas de desenvolvimento. O projeto do estágio de potência do conversor, em geral, segue a metodologia proposta por [46].

2.1.1 Especificações do Conversor Duplo-Forward

As baterias aqui utilizadas são baterias de Lítio Panasonic NCR 18650 3,6 V e 2,75 Ah. Conforme a Figura 3, as baterias podem ser organizadas em até dois conjuntos em série de até quatro baterias em paralelo, totalizando um máximo oito células. Uma única célula descarregada possui aproximadamente 2,5 V de tensão terminal, podendo

atingir até $4,2\text{ V}$ no fim do processo de carga. Abrangendo desde a situação de uma única célula descarregada até duas células completamente carregadas, o conversor deve ser capaz de impor uma tensão de saída que varie entre $2,5\text{ V}$ e $8,4\text{ V}$. Com relação a capacidade de corrente do conversor, foi adotado que o mesmo deve ser capaz de impor uma corrente de $1C$ (sendo que $1C$ é uma nomenclatura em baterias que define a corrente de carga capaz de carregar completamente uma dada bateria em uma hora, o que neste caso representa $2,75\text{ A}$, já que a bateria possui capacidade de $2,75\text{ Ah}$) [47] para cada bateria. Considerando até quatro baterias em paralelo drenando $1C$ cada, tem-se que a corrente de saída do conversor deve ser de pelo menos 11 A . Tendo estas características em mente, foram definidas as especificações do conversor conforme a Tabela 3.

Tabela 3 – Especificações do conversor Duplo-Forward

Descrição	Variável	Valor
Tensão de Entrada	V_{in}	295 a 326 V
Frequência de chaveamento	f_s	100 kHz
Ondulação de corrente no indutor	Δi_L	1.5 A
Ondulação de tensão no capacitor	Δv_o	3 mV
Potência máxima de saída	$P_{o_{max}}$	120 W
Corrente máxima de saída	$I_{o_{med}}$	12 A
Tensão máxima de saída	$V_{o_{max}}$	10 V

Fonte: produção do próprio autor, 2016

A partir das especificações de projeto, é realizado o desenvolvimento do carregador de baterias, o qual é detalhado no Apêndice A. Na sequência, serão mostrados os resultados do conversor Duplo-Forward projetado em malha aberta.

2.2 RESULTADOS DO CONVERSOR EM MALHA ABERTA

Esta seção traz os resultados obtidos do conversor Duplo-Forward, publicados em [39], demonstrando sua operação em malha aberta con-

forme os requisitos de projetos delimitados na Tabela 3 da seção 2.1.1. Assim, serão mostrados os resultados de simulação e experimentais de corrente no indutor e tensão no capacitor de saída, visando mostrar seus valores médios, ondulação e potência de saída, validando assim o funcionamento do conversor. Resultados adicionais podem ser encontrados na seção B.4 do Apêndice B.

Na Figura 4 são mostrados os resultados de simulação da corrente no indutor e da tensão no capacitor de saída, em um momento arbitrário.

Através das formas de onda mostradas na Figura 4, nota-se que a corrente no indutor possui um valor médio aproximado de $11,7\text{ A}$, enquanto a tensão do capacitor possui um valor médio de aproximadamente $9,875\text{ V}$, o que resulta numa potência de saída de $115,54\text{ W}$. Além disso, os valores de ondulação de corrente e de tensão ficaram próximos de $1,5\text{ A}$ e 3 mV .

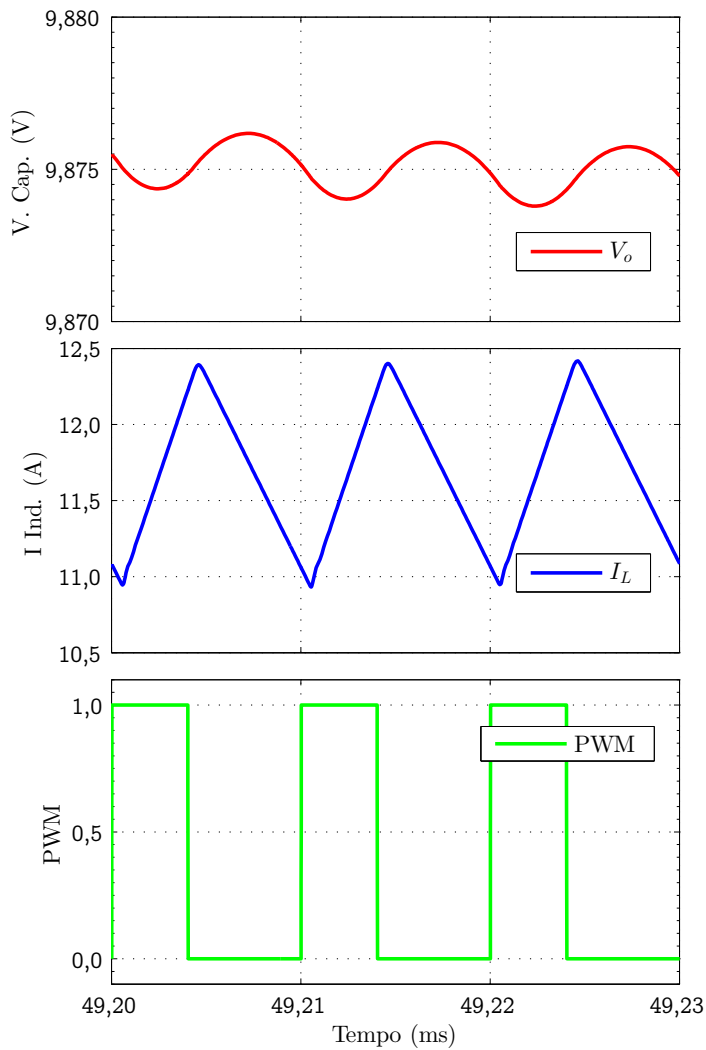
Os resultados obtidos em simulação comprovam o funcionamento do conversor. As ondulações de tensão e corrente ficaram dentro dos valores especificados. Já os valores de tensão e corrente médios obtidos ligeiramente abaixo do especificado podem ser explicados pela ondulação de tensão do capacitor de filtro do retificador de entrada, que levou a uma potência de saída ligeiramente abaixo do valor nominal. Tal ondulação é inerente de um conversor conectado a um retificador com filtro capacitivo e pode ser corrigida com o controle do conversor.

Tendo sido o conversor validado malha em aberta via simulação, será mostrada agora sua validação através de resultados experimentais. A Figura 5 apresenta uma foto do conversor Duplo-Forward montado juntamente com seu retificador ponte completa. As formas de onda de corrente no indutor e tensão no capacitor são mostradas na Figura 6.

Nestes resultados da Figura 6, os valores de tensão e de corrente médios observados foram de $9,99\text{ V}$ e $11,9\text{ A}$, o que resulta em uma potência de aproximadamente 119 W .

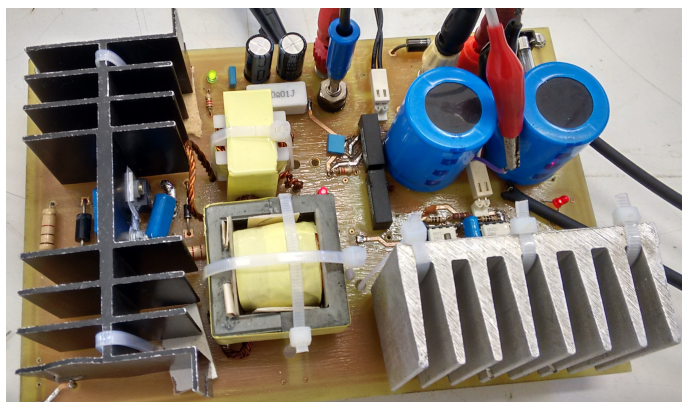
Para a análise das ondulações, foram obtidas medidas com o osciloscópio Tektronix MDO4054B-3 e ponteiras de alta frequência, visando aprimorar a acuracidade das formas de onda, que são mostradas na Figura 7.

Figura 4 – Formas de onda simuladas da corrente no indutor (azul) e da tensão no capacitor de saída (vermelho)



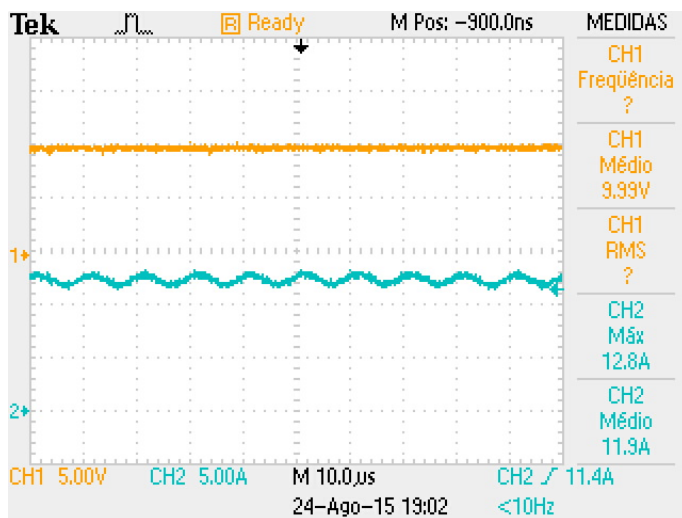
Fonte: produção do próprio autor, 2016

Figura 5 – Imagem do protótipo do conversor Duplo-Forward projetado



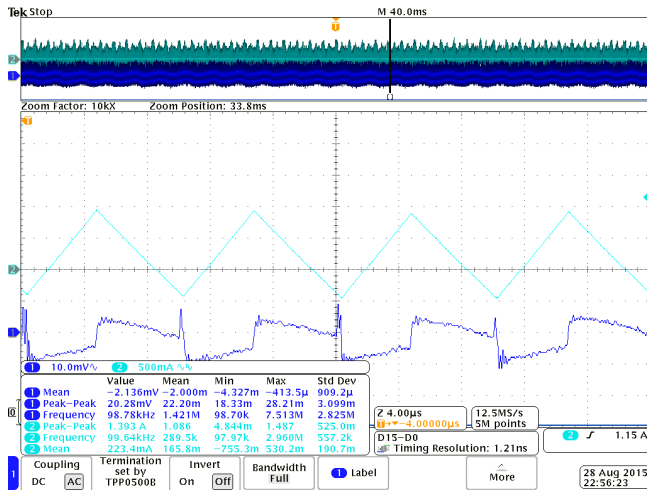
Fonte: produção do próprio autor, 2016

Figura 6 – Formas de onda experimentais de corrente no indutor (azul - CH2) e tensão no capacitor de saída (amarelo - CH1)



Fonte: produção do próprio autor, 2016

Figura 7 – Formas de onda de ondulação de corrente no indutor (azul claro - CH2) e de tensão no capacitor de saída (azul escuro - CH1)



Fonte: produção do próprio autor, 2016

Os resultados obtidos através de medidas com acoplamento CA no osciloscópio mostram claramente uma ondulação de corrente no indutor de aproximadamente 1,4 A. Já a forma de onda de tensão no capacitor de saída apresenta um comportamento inesperado, apresentando uma forma de onda com componente quadrada somada a uma forma de onda com componente aproximadamente triangular. Isto pode ser justificado pelo fato de que o objetivo de medição contempla valores muito pequenos (na ordem de poucos *mV*), tornando as medições extremamente suscetíveis a ruídos. Somado a isto, há ainda o fato de o indutor estar muito próximo do capacitor de saída no conversor físico, o que pode gerar tensões induzidas indesejadas, e que possivelmente são a causa da componente quadrada medida, já que a tensão no indutor é uma forma de onda do tipo quadrada. Desta forma, a ondulação de tensão no capacitor não pôde ser comprovada experimentalmente, onde pôde-se afirmar apenas que a ondulação de tensão é menor que 20 *mV*.

2.3 CONSIDERAÇÕES SOBRE O CONVERSOR DUPLO-FORWARD

Este capítulo apresentou o conversor Duplo-Forward aplicado na carga de baterias de Lítio. Apesar do desenvolvimento do conversor não estar atrelado ao objetivo geral deste trabalho, o mesmo foi incluído entre os objetivos específicos por estar diretamente ligado às baterias e à manipulação do SOC das mesmas, além de trazer um forte amadurecimento em aspectos da engenharia voltada à implementação de *hardware*, uma vez que a eletrônica de potência exige conhecimentos sobre elementos magnéticos, dispositivos de acionamento (amplificadores, optoacopladores, etc.), dispositivos semicondutores e seu dimensionamento térmico, circuitos de proteção, filtros, ruídos, estratégias de medição (instrumentação eletrônica, dentre outras, e todos estes conhecimentos acabam sendo trabalhados de maneira integrada, o que garante um amplo leque de aprendizado.

Após todo o dimensionamento de componentes e da execução da montagem do conversor, os quais tiveram suas etapas descritas nos Apêndices A e B, foram obtidos os resultados do funcionamento do conversor em malha aberta. Em tais resultados, observou-se que o conversor atinge seus valores de tensão e corrente máximos de saída especificados, sendo capaz de trabalhar em sua potência nominal. Com relação às ondulações de corrente no indutor e de tensão no capacitor, apenas a primeira foi validada experimentalmente ficando bem próxima do valor especificado de projeto, enquanto a ondulação de tensão só pode ser validada através de simulações.

Ainda que a ondulação de tensão não pôde ser validada experimentalmente, o conversor pode ser utilizado sem maiores problemas para a carga de baterias, uma vez que mesmo uma ondulação de 20 mV não é suficiente para comprometer as mesmas, aliado ao fato de o conversor ser capaz de atingir sua potência de saída requerida. Ademais, deve-se lembrar que a ondulação de tensão experimental possivelmente está muito abaixo dos 20 mV , uma vez que há fortes indícios de que a medição desta tensão está poluída com ruídos e que os valores de ondulação foram validados via simulação, conforme abordado na seção 2.2. Sendo assim, é viável a utilização do conversor, comprovada posteriormente com a carga de baterias.

No capítulo seguinte, serão tratados os desenvolvimentos do controle do conversor, das suas proteções, além dos circuitos utilizados no condicionamento dos sinais medidos.

3 CONTROLE, PROTEÇÃO E CONDICIONAMENTO DE SINAIS

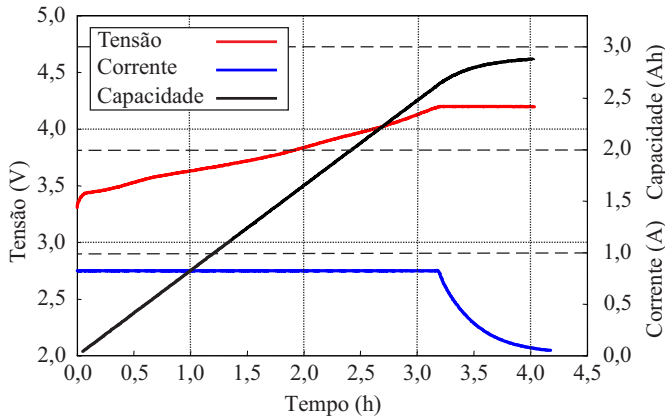
Aplicações que exigem um alto grau de confiabilidade na operação das baterias, como é o caso dos BEVs e *Smart Grids*, requerem um BMS com capacidade de monitorar os sinais da bateria com boa precisão, protegê-la de situações que ofereçam risco e ainda de gerenciar o carregador de baterias. Para uma boa estimação do SOC, por exemplo, que é uma das tarefas a serem executadas pelo BMS, necessita-se de medições de tensão com tolerância na casa dos mV , enquanto a medição de corrente, principal grandeza utilizada em diversos algoritmos de estimação, requer uma precisão de cerca de 0,5 a 1% do valor de corrente nominal da aplicação. Tais medições frequentemente levam à necessidade de circuitos de condicionamento de sinais [9, 48].

A proteção das baterias também é outro fator relevante, uma vez que existem várias situações que podem oferecer risco à operação das mesmas, em especial às baterias de Lítio. Fatores como a operação com elevadas correntes (maior que $1C$) de carga ou descarga, descargas profundas, sobrecargas, temperaturas extremas, entre outros, podem acarretar em incidentes graves com as baterias de lítio [9, 48]. Não é difícil de se encontrar vídeos que mostram explosões de baterias de Lítio pelos mais diversos motivos [49, 50]. Logo, baterias de lítio exigem estratégias que garantam que sua operação não excederá os limites seguros.

A carga das baterias também deve ser feita de forma cuidadosa. Existem métodos simplificados, como é o caso da carga em corrente constante, porém a escolha da corrente de carga não se mostra uma tarefa fácil, já que correntes elevadas podem sobreaquecer as baterias ao ponto de explodi-las, enquanto correntes muito baixas requerem longos tempos de recarga [51]. Desta forma, um dos métodos de carga mais difundidos tanto na academia quanto na indústria, é a carga com Corrente Constante e Tensão Constante (CC-CV), que agrega uma boa relação entre o tempo de recarga e confiabilidade de operação [48, 51–56]. A Figura 8 mostra as curvas de tensão, corrente e SOC da bateria NCR 18650 Panasonic, utilizada neste trabalho, durante a sua carga com o protocolo CC-CV, obtida através do catálogo do fabricante.

O perfil de carga CC-CV é dividido fundamentalmente em duas

Figura 8 – Representação da carga de uma célula NCR 18650 com perfil CC-CV



Fonte: Adaptado de Panasonic Corporation [56]

fases, sendo a primeira fase a carga em corrente constante e a segunda fase a carga em tensão constante. Conforme mostra a Figura 8, quando descarregada, a bateria possui uma tensão relativamente baixa, próxima de 2,5 V, que se eleva juntamente com o aumento de carga da bateria, até atingir seu valor máximo de tensão, que é de 4,2 V. Dentro destes limites de tensão, a bateria é carregada com corrente constante. Tendo atingido seu valor máximo de 4,2 V, a carga da bateria passa então à fase de tensão constante, controlando-se a tensão da bateria ao redor do seu valor máximo, o que por consequência leva à diminuição gradativa da corrente de carga até que esta fique próxima de 50 mA, quando se considera que a bateria está completamente carregada.

Ao utilizar o perfil de carga CC-CV, o conversor Duplo-Forward deverá ser controlado tanto em tensão quanto em corrente, sendo que a transição entre as duas fases do protocolo de carga CC-CV deve ocorrer de forma suave. Ainda, deve-se destacar que a ondulação de corrente no indutor é $\Delta I_L = 1,4 \text{ A}$, levando o conversor a trabalhar no modo de condução descontínua DCM quando for exigida uma corrente com valor médio de aproximadamente 700 mA ou menor, o que acarreta em

mudanças na planta a ser controlada. Logo, o controle do carregador de baterias deve ser capaz de lidar com tais mudanças, controlando o conversor em ambos os modos de condução, contínuo (*Continuous Conduction Mode* - CCM) ou descontínuo (*Discontinuous Conduction Mode* - DCM) [39].

De forma a realizar a carga das baterias com segurança e confiabilidade e ainda monitorar os seus valores de tensão e corrente, os quais são necessários para a estimação do SOC e para a proteção das mesmas, são mostrados neste capítulo o desenvolvimento de circuitos de condicionamento de sinais, bem como o projeto de controle do conversor Duplo-Forward e os resultados obtidos durante a carga de diferentes arranjos de baterias. Na sequência, será mostrado o desenvolvimento dos circuitos de condicionamento de sinais.

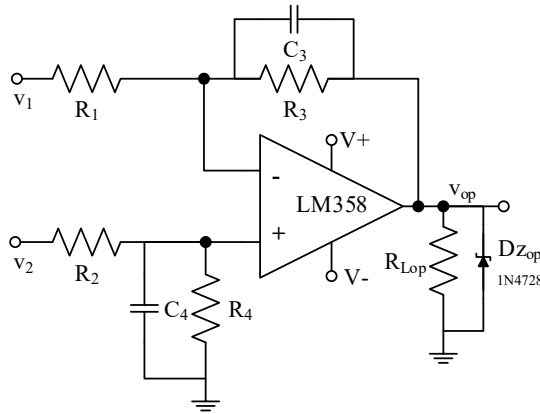
3.1 CIRCUITOS DE CONDICIONAMENTO DE SINAIS

O condicionamento dos sinais a serem utilizados tanto nas estratégias de controle quanto de proteção é realizado com o objetivo de adequar os níveis do sinal medido, tanto em magnitude quanto em aspectos frequenciais. Desta forma, foram utilizados em todos os pontos de medição desejados amplificadores diferenciais visando amplificar o sinal de tensão entre dois terminais de medição, agregando facilidade na sua realização, redução de ruído de modo comum e adequação da magnitude da tensão medida. Foram ainda adicionados dois capacitores à configuração original do amplificador diferencial para que sejam filtradas as componentes de alta frequência, além da adição de um diodo zener para limitação da tensão de saída do Amplificador Operacional (AMPOP).

Na Figura 9 tem-se o circuito de condicionamento apresentado de forma genérica. As tensões v_1 e v_2 representam os sinais de entrada do circuito de condicionamento, enquanto v_{op} representa a tensão de saída já condicionada. O AMPOP utilizado é um LM358, alimentado por uma fonte externa de 5 V não simétrica. O resistor na saída do AMPOP é apenas um resistor de carga com alta impedância, possuindo uma resistência $R_{Lop} = 10 \text{ k}\Omega$. O diodo zener Dz_{op} (1N4728) limita a tensão de saída em 3,3 V. Esta limitação da tensão de saída

condicionada serve como proteção ao *Digital Signal Controller* (DSC - Controlador Digital de Sinais) TMS320F28335, o qual possui vários *Analog-to-Digital Converters* (ADC - Conversores Analógico-Digitais) que permitem uma tensão de entrada na faixa de 0 a 3 V. Todos os sinais utilizados passam por estes ADCs do DSC, os quais possuem um ganho de 4096/3.

Figura 9 – Esquema genérico do circuito de condicionamento de sinais



Fonte: produção do próprio autor, 2016

Considerando que $R_1 = R_2$, $R_3 = R_4$, $C_3 = C_4$, o ganho desta estrutura pode ser descrito pela equação (3.1).

$$G_{op}(s) = \frac{v_{op}(s)}{v_2(s) - v_1(s)} = \frac{R_3}{R_1} \frac{1}{sR_3C_3 + 1} \quad (3.1)$$

A partir de (3.1), tem-se então que o amplificador tem característica de um *Low-Pass Filter* (LPF - Filtro Passa-Baixas) de primeira ordem, com ganho CC $G_{op}(0) = R_3/R_1$ e frequência de corte $f_{c.op} = 1/(2\pi R_3 C_3)$. Foi definida uma frequência de corte de aproximadamente $f_{c.op} = 10 \text{ kHz}$ para todos os circuitos de condicionamento, de forma a atenuar sinais na ordem da frequência de chaveamento (100 kHz) mas ainda tendo boa representatividade dos sinais

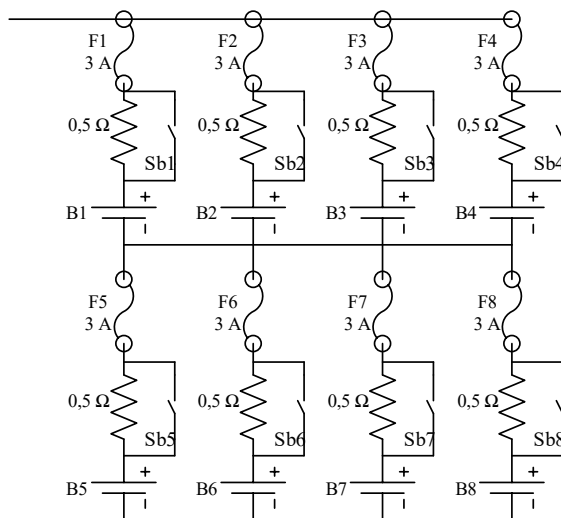
em frequências na ordem de poucos kHz , os quais serão utilizados no controlador.

Tendo sido definidos os aspectos mais gerais do condicionamento dos sinais, como a limitação da tensão condicionada e frequência de corte do LPF, passa-se então ao circuito do arranjo das baterias. Detalhes sobre cada um dos circuitos de condicionamento implementados podem ser encontrados no Apêndice C.

3.2 O ARRANJO DE BATERIAS E SUAS PROTEÇÕES

Esta seção busca mostrar o desenvolvimento do circuito das baterias, compondo além destas, seus elementos de proteção. A Figura 10 mostra o esquemático geral do circuito das baterias.

Figura 10 – Esquemático do arranjo de baterias.



Fonte: produção do próprio autor, 2016

A partir da Figura 10, nota-se que cada bateria possui um fusível individual de 3 A, visando proteger individualmente cada bateria contra correntes maiores que $1C$. Ainda, cada bateria conta com um

resistor de $0,5 \Omega$ em paralelo com um interruptor Sb . A função do resistor de $0,5 \Omega$ é limitar a corrente entre baterias durante a fase de balanceamento de carga e consequentemente evitar o curto-circuito entre baterias conectadas em paralelo, sendo que este balanceamento antecede a fase de recarga das baterias. É importante notar que durante o balanceamento de cargas, todos os interruptores Sb devem estar abertos. Tais interruptores são fechados quando deseja-se realizar o *bypass* dos resistores, ou seja, durante a etapa de recarga de baterias, após um balanceamento prévio de carga.

Existem pesquisas que tem por objetivo realizar o balanceamento de carga entre células de forma automatizada e com a maior eficiência possível, o que não estava entre os objetivos deste trabalho, o que justifica o uso de resistores e interruptores manuais para realizar tal tarefa.

3.3 CONFIGURAÇÃO DO DSC

O DSC TMS320F28335 da Texas Instruments é uma espécie de microcontrolador dedicado à tarefas de controle, possuindo módulos dedicados de ADC, PWM, interrupções, dentre outras funções que em geral são necessárias em estratégias de controle.

Ao se utilizar um microcontrolador, deve-se primeiramente realizar a configuração dos seus registradores buscando adaptar suas funcionalidades às necessidades da aplicação que se deseja controlar. Estes registradores são os responsáveis por configurar as portas digitais como entrada ou saída, definir quais as portas lógicas que podem ser utilizadas para a saída do PWM, definir quantos e quais canais do ADC serão utilizados, quais interrupções serão utilizadas e como elas serão disparadas, entre outras funcionalidades.

Informações mais detalhadas sobre o funcionamento e configuração do DSC podem ser encontradas nos manuais do fabricante ou também através da análise dos códigos dispostos na seção B.1 do Anexo B desta dissertação.

3.4 PROTEÇÕES POR SOFTWARE

Além das proteções realizadas por *hardware*, foram estruturadas diversas lógicas de proteção, que são disparadas por lógicas do tipo *if – else*.

A lista abaixo contempla todas as proteções que foram implementadas por *software*:

- Proteção contra sobretensão do conversor ($V_o > 10 \text{ V}$).
- Proteção contra sobretensão das células ($V_b > 4,2 \text{ V}$).
- Proteção contra subtensão das células ($V_b < 2,5 \text{ V}$).
- Proteção contra sobrecarga do conversor ($I_L > 12 \text{ A}$).
- Proteção contra curto-circuito na saída do conversor ($V_o < 0,1 \text{ V}$ e $I_L > 1 \text{ A}$).

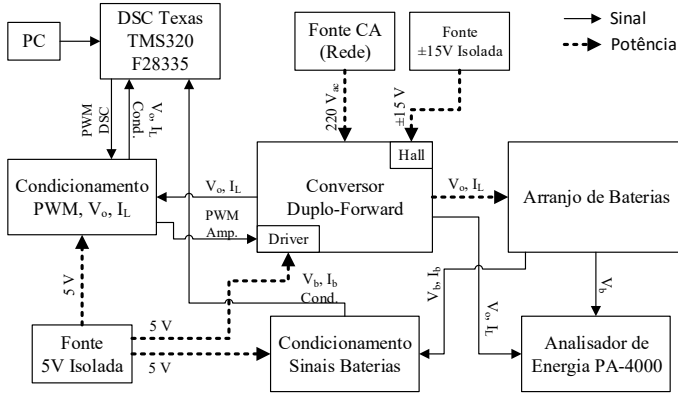
Estas situações são verificadas durante a interrupção de *timer*, que ocorre a cada 1 s . Se dentro deste período alguma destas situações perdurar por mais de 200 ms , o algoritmo de proteção é disparado, o qual força a saída do PWM do DSC para nível baixo e posteriormente, leva todos os sinais de referência para zero. A partir deste ponto, são contadas 10 interrupções de *timer*. Se a anormalidade não voltar a ocorrer, a saída do PWM volta a funcionar e as referências de controle são reestabelecidas. Caso a anormalidade persista ou volte a ocorrer, mantém-se a saída do PWM desligada e volta-se a contar 10 interrupções de *timer* do zero, até que o sistema não apresente mais quaisquer problemas de operação.

3.5 ESQUEMÁTICO COMPLETO

Enquanto no capítulo 2 foram abordados os aspectos de projeto referentes ao estágio de potência, em especial do conversor 2SFC, neste capítulo foram apresentados os circuitos que realizam o tratamento de sinais necessários às estratégias de controle do conversor e consequentemente ao seu funcionamento.

Para finalizar a discussão sobre os elementos relacionados ao *hardware* deste trabalho, é apresentado na Figura 11 um diagrama de blocos completo contendo todos os elementos principais que compõe o carregador de baterias e uma representação de como estes estão interligados.

Figura 11 – Esquemático completo do sistema de carga de baterias.



Fonte: produção do próprio autor, 2016

É importante comentar que, todos os sinais enviados ao DSC foram digitalmente calibrados e multiplicados pelos inversos dos ganhos dos seus circuitos de condicionamento, retornando ao valor real (ou muito próximo) das medidas. Assim, uma corrente de 12 A no conversor, por exemplo, deve ser lida no DSC com um valor próximo de 12. Isto permite a utilização de realimentação unitária em todas as etapas de projeto do controlador.

Tendo-se em mãos os circuitos de *hardware* necessários para o carregador de baterias e para as estratégias de controle, além da configuração do DSC, passa-se então ao projeto de controle do conversor.

3.6 PROJETO DE CONTROLE

O controle do conversor Duplo-Forward é fundamental para a operação de carga das baterias, a qual em um perfil de carga CC-

CV exige que a saída do carregador seja controlada tanto em corrente quanto em tensão.

Para a realização do controle do carregador de baterias, foi utilizada a abordagem de controle ótimo (ou Linear Quadrático – LQ) em espaço de estados, com o objetivo de explorar a utilização desta técnica no controle de conversores, e também para que esta dissertação sirva como um guia prático para futuros trabalhos que desejem utilizar tal abordagem (que pode ser estendida a outros conversores além do Duplo-Forward).

O DSC TMS320F28335 da Texas Instruments será utilizado para a realização da técnica de controle escolhida, o que implica na implementação de um controlador digital, resultando assim em uma maior facilidade em mitigar variações paramétricas [52] e em realizar a calibração das medidas, além de possibilitar a implementação de lógicas de proteção por software, trazendo grande flexibilidade ao projeto.

Durante todo o projeto de controle, são utilizadas as seguintes suposições:

- O modelo é linear e invariante no tempo.
- A ondulação da tensão de entrada do conversor é desconsiderada, assumindo-se uma tensão de entrada constante V_{in} .
- A bateria é considerada uma carga resistiva de valor constante. A variação de tal resistência de carga é tratada como uma perturbação do sistema.
- Todos os componentes são considerados ideais.

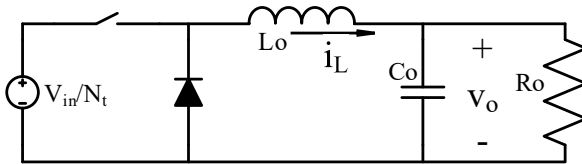
Para que o controle do conversor seja realizado tanto em corrente quanto em tensão, será utilizada uma malha em cascata, que será abordada em detalhes adiante. Na sequência será abordada a modelagem do conversor, que posteriormente será utilizada no seu projeto de controle. Toda a abordagem de controle e os resultados descritos aqui podem ser encontrados em [39].

3.6.1 Modelagem do Conversor Duplo-Forward

Para realizar a modelagem do conversor Duplo-Forward, será utilizado o conceito de valor médio instantâneo [57], no qual considera-se o valor médio de uma grandeza dentro do seu período de comutação, o que implica em desprezar dinâmicas com frequências na ordem da frequência de chaveamento (ou maiores) e levar em conta apenas as dinâmicas mais lentas (em geral até 10 vezes menores que f_s).

Uma vez que os componentes do conversor são considerados ideais, a tensão aplicada no enrolamento primário do conversor Duplo-Forward é diretamente refletida para o enrolamento secundário, que poderá fazer com que o diodo de condução direta D_{o1} entre ou não em condução. Assim, o enrolamento secundário pode ser visto como uma fonte de tensão e o diodo D_{o1} pode ser interpretado como um interruptor, caracterizando assim o conversor Buck, e por este motivo, este pode ser utilizado para representar o conversor Duplo-Forward durante a etapa de modelagem. A Figura 12 mostra o circuito do conversor Buck.

Figura 12 – Circuito do conversor Buck.

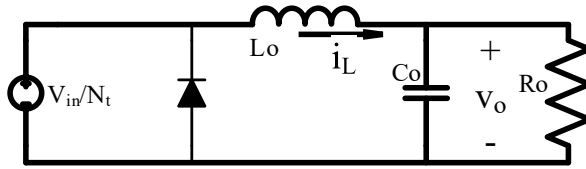


Fonte: produção do próprio autor, 2016

O conversor Buck operando no CCM possui duas etapas de operação. Uma delas é quando o interruptor está fechado e a energia da fonte é transferida para o indutor e para a carga, conforme a Figura 13.

A partir da Figura 13, pode-se equacionar a corrente no indutor $i_L(t)$ e a tensão no capacitor $v_o(t)$, que de forma genérica, são descritos

Figura 13 – Etapa 1 de funcionamento do conversor Buck.



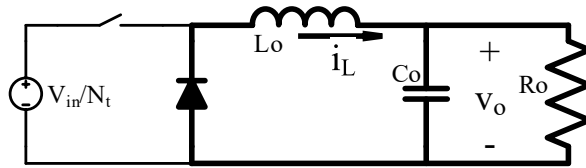
Fonte: produção do próprio autor, 2016

conforme a equação (3.2).

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \end{bmatrix} = [A_{c.on}] \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix} + [B_{c.on}] \quad (3.2)$$

A outra etapa de operação consiste no tempo em que o interruptor está aberto, forçando com que a corrente do indutor circule pelo diodo de roda-livre e fazendo com que a carga seja alimentada pela energia armazenada no filtro de saída, conforme Figura 14.

Figura 14 – Etapa 2 de funcionamento do conversor Buck.



Fonte: produção do próprio autor, 2016

Na etapa de operação descrita pela Figura 14, $i_L(t)$ e $v_o(t)$ podem ser descritos de forma genérica pela equação (3.3).

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \end{bmatrix} = [A_{c.off}] \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix} + [B_{c.off}] \quad (3.3)$$

Tendo-se os equacionamentos de ambas as etapas de operação e utilizando-se do conceito de valor médio instantâneo, pode-se então

agregar os dois modelos em um único que represente a média de tais grandezas dentro de um período, conforme apresentado em (3.4) [58].

$$\begin{cases} A_c = A_{c.on}d + A_{c.off}(1-d) \\ B_c = B_{c.on}d + B_{c.off}(1-d) \end{cases} \quad (3.4)$$

Como $A_{c.on} = A_{c.off}$ e $B_{c.off} = \mathbf{0}$, tem-se então as matrizes A_c e B_c que representam o modelo dinâmico do sistema conforme o conceito de valor médio instantâneo, apresentadas em (3.5).

$$\begin{cases} A_c = A_{c.on} = A_{c.off} \\ B_c = B_{c.on} \end{cases} \quad (3.5)$$

Sendo assim, as equações dinâmicas de $i_L(t)$ e $v_o(t)$ podem ser escritas conforme equação (3.6), que é a representação em espaços de estados do conversor Buck no domínio contínuo pelo conceito de valor médio instantâneo.

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \end{bmatrix} = \begin{bmatrix} 0 & -1/L_o \\ 1/C_o & -1/R_o C_o \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix} + \begin{bmatrix} \frac{V_{in}}{N \cdot L_o} \\ 0 \end{bmatrix} d(t) \quad (3.6)$$

De forma mais genérica, o modelo é dado pela relação (3.7).

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \end{bmatrix} = [A_c] \begin{bmatrix} i_L(t) \\ v_o(t) \end{bmatrix} + [B_c] d(t) \quad (3.7)$$

Tal modelo deve então ser discretizado para a implementação do controlador no domínio digital. Utilizando o conceito do *Zero-Order Holder* (ZOH - Segurador de Ordem Zero), tem-se que as matrizes de um sistema discreto em espaço de estados podem ser obtidas através de um modelo em espaços de estados contínuo, utilizando-se a equação descrita em (3.7), resultando no sistema discreto da equação (3.8). A frequência de amostragem utilizada foi $f_a = 20 \text{ kHz}$ (5 vezes menor que f_s), com o objetivo de dar ao DSC mais tempo para realização dos

cálculos do controlador.

$$(A_c, B_c) \rightarrow ZOH(f_a) \rightarrow (A, B)$$

$$\begin{bmatrix} i_L(k) \\ v_o(k) \end{bmatrix} = [A] \begin{bmatrix} i_L(k-1) \\ v_o(k-1) \end{bmatrix} + [B] d(k-1) \quad (3.8)$$

O vetor de estados deste sistema é $X(k) = [i_L(k) \ v_o(k)]^T$, enquanto a variável de controle é $u(k) = d(k)$. A partir do vetor de estados $X(k)$, pode-se escrever a saída do sistema como sendo a corrente $i_L(k)$ ou como sendo a tensão $v_o(k)$, segundo as equações (3.9) e (3.10), respectivamente, onde $C_i = [1 \ 0]$ e $C_v = [0 \ 1]$. Uma vez que não há influência instantânea da ação de controle $u(k)$ sobre $i_L(k)$ ou $v_o(k)$, tem-se que $D = 0$ para ambos os casos.

$$y_i(k) = C_i X(k) = i_L(k) \quad (3.9)$$

$$y_v(k) = C_v X(k) = v_o(k) \quad (3.10)$$

Assim, tem-se o modelo discreto do conversor em espaço de estados descrito pelas matrizes A , B , C e D , onde C pode ser C_i ou C_v , dependendo da saída que se deseja utilizar. É a partir destes modelos que serão obtidos os controladores e observadores do conversor.

3.6.2 Estratégias de Controle Ótimo

O projeto de controladores baseados nas técnicas de controle ótimo já é bem comum na literatura em geral. O *Linear-Quadratic Regulator* (LQR - Regulador Linear Quadrático) e o *Kalman Filter* (KF - Filtro de Kalman) (também conhecido como *Linear-Quadratic Estimator* - LQE) são estratégias de regulação bem difundidas baseadas em controle ótimo (ou *Linear-Quadratic* - LQ).

Ao passo que estratégias de controle clássicas como o Proporcional Integral Derivativo (PID) se baseiam em polos e zeros do sistema para obtenção do controlador, estratégias de controle ótimo se emba-

sam na otimização de funções que de certa forma caracterizem o sistema e suas variáveis. Essas funções são comumente chamadas de funções de custo, as quais devem ser minimizadas para que se encontre o controlador ótimo relacionado a tais funções. Uma das funções de custo amplamente descritas na literatura é descrita na equação (3.11) [59].

$$J = \sum_{k=0}^{\infty} (X^T(k)Q_{LQ}X(k) + u^T(k)R_{LQ}u(k)) \quad (3.11)$$

Esta função de custo realiza a minimização do vetor de estados do sistema $X(k)$ e da ação de controle $u(k)$, os quais são ponderados pelas matrizes de peso Q_{LQ} e R_{LQ} , respectivamente.

É possível notar que a equação (3.11) possui um termo dado por $X^T(k)Q_{LQ}X(k)$ e um outro termo dado pela relação $u^T(k)R_{LQ}u(k)$. Uma vez que as matrizes Q_{LQ} e R_{LQ} são quadradas e constantes, estas tem como única função ponderar os termos $X^T(k)X(k)$ e $u^T(k)u(k)$.

No caso do conversor Duplo-Forward, os estados do vetor $X(k)$ são a corrente no indutor $i_L(k)$ e a tensão no capacitor de saída $v_o(k)$. Quando Q_{LQ} é diagonal, o termo $X^T(k)Q_{LQ}X(k)$ resulta em uma ponderação de $i_L^2(k)$ e $v_o^2(k)$, que são a energia do sistema. Já a variável de controle $u(k)$ é a própria razão cíclica do sistema $d(k)$, e o seu termo quadrático está ligado a energia do sinal de controle. Desta forma, a equação (3.11) é uma função de custo que minimiza a energia do sistema juntamente com a energia do seu sinal de controle.

Ao escolher um valor de Q_{LQ} maior em (3.11), haverá um compromisso maior de se minimizar os estados $X(k)$ o mais rápido possível tendo uma menor preocupação com o sinal de controle $u(k)$, levando a uma ação de controle mais agressiva, enquanto que ao escolher um valor maior de R_{LQ} , tem-se um compromisso maior em reduzir a energia do sinal de controle do que de minimizar os estados, o que leva a ações de controle mais suaves, ou seja, mais conservativas. Um desafio no projeto de reguladores e compensadores LQ é como escolher os valores de Q_{LQ} e R_{LQ} que trarão o melhor compromisso entre a minimização da energia do sistema e da sua ação de controle. Na seção 3.6.5 será discutido o algoritmo aqui utilizado para auxiliar em

tal escolha.

Outro fator importante a ser destacado é que tanto o LQR quanto o KF são reguladores, o que implica que estes não procuram seguir uma referência específica. Em outras palavras, ambos buscam levar os estados $X(k)$ para zero, que é o papel executado por um regulador. Para que o LQR e o KF possam atuar como controladores, ou seja, seguir referências não nulas, deve-se utilizar um estado adicional que realizará o papel de integrador do sistema. A obtenção desta representação com estados adicionais será descrita na seção 3.6.3.

3.6.3 Representação incremental em espaço de estados

Há diversas formas de se aumentar os estados do sistema para a inclusão do integrador, onde uma delas é a representação incremental em espaço de estados. Esta forma incremental também é conhecida como forma estendida por Δ , que é o operador de diferenças finitas. Esta representação será aqui utilizada porque a adição do estado adicional é observável desde que o sistema em questão também seja observável, enquanto que outras representações podem acarretar em estados adicionais não observáveis. Considerando um sistema genérico conforme a equação (3.8) e multiplicando tal equação pelo operador Δ , tem-se uma nova representação do sistema, mostrada na equação (3.12).

$$\begin{cases} \Delta \times [X(k) = AX(k-1) + Bu(k-1)] \\ \Delta X(k) = A\Delta X(k-1) + B\Delta u(k-1) \end{cases} \quad (3.12)$$

Sabe-se que a saída do sistema pode ser descrita pela equação (3.13).

$$y(k) = y(k-1) + \Delta y(k) \quad (3.13)$$

Ainda, tem-se a seguinte relação entre a saída do sistema e os seus estados, conforme (3.14).

$$y(k) = CX(k) \quad (3.14)$$

Multiplicando-se (3.14) por $\Delta = 1 - z^{-1}$, tem-se a equação

(3.15).

$$\Delta y(k) = C\Delta X(k) \quad (3.15)$$

Manipulando (3.15), tem-se a seguinte relação mostrada em (3.16).

$$y(k) - y(k-1) = C\Delta X(k) \quad (3.16)$$

Aplicando a Transformada Z em (3.16), tem-se sua representação no domínio Z conforme (3.17).

$$Y(z)(1 - z^{-1}) = C\Delta X(z) \quad (3.17)$$

Representando-se (3.17) como uma Função de Transferência, tem-se a equação (3.18), que nada mais é que um integrador puro no domínio discreto.

$$\frac{Y(z)}{\Delta X(z)} = \frac{C}{1 - z^{-1}} \quad (3.18)$$

Juntando-se as equações (3.12) e (3.16) tem-se a relação (3.19).

$$\begin{cases} \Delta X(k) = A\Delta X(k-1) + B\Delta u(k-1) \\ y(k) = y(k-1) + C\Delta X(k) \end{cases} \quad (3.19)$$

A equação (3.19) pode então ser escrita na forma matricial (para o caso SISO - *Single-Input Single-Output*, conforme equação (3.20).

$$\begin{bmatrix} \Delta X(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & \mathbf{0} \\ CA & 1 \end{bmatrix} \begin{bmatrix} \Delta X(k-1) \\ y(k-1) \end{bmatrix} + \begin{bmatrix} B \\ CB \end{bmatrix} \Delta u(k-1) \quad (3.20)$$

Fazendo $X_a(k) = [\Delta X(k) \ y(k)]^T$, tem-se então o novo sistema em espaço de estados em (3.21).

$$\begin{cases} X_a(k) = A_a X_a(k-1) + B_a \Delta u(k-1) \\ y(k) = C_a X_a(k) \end{cases} \quad (3.21)$$

Utilizando-se do sistema dado por (3.8) e de sua função de custo descrita em (3.11), tem-se a nova função de custo a ser minimizada,

definida conforme a equação (3.22).

$$J_a = \sum_{k=0}^{\infty} (X_a^T(k) Q_{LQ.a} \Delta X_a(k) + \Delta u^T(k) R_{LQ.a} \Delta u(k)) \quad (3.22)$$

A partir da representação do sistema na forma incremental em espaço de estados dada pela equação (3.21), é realizado o projeto do controlador LQG .

3.6.4 Projeto do LQG

O compensador *Linear-Quadratic-Gaussian* (LQG - Linear-Quadrático-Gaussiano) é a união entre o controlador (LQR) e o estimador de estados (KF), os quais são projetados seguindo a metodologia de controle ótimo. Esta metodologia de projeto é a metodologia empregada em [39], estando alguns dos dados aqui comentados também publicados no artigo mencionado.

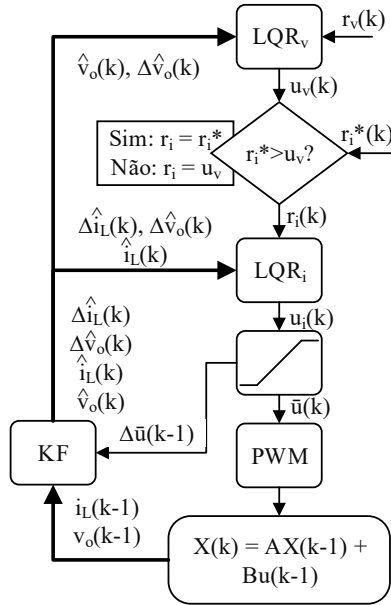
Conforme dito na seção 3.1, ambos os sinais de tensão de saída v_o e corrente no indutor i_L são condicionados e fornecidos ao DSC, ou seja, os dois estados necessários estão disponíveis para realimentação, o que em princípio dispensaria a necessidade de estimação dos mesmos. Porém, para que haja menor influência do ruído de comutação, há um apelo de se atenuar fortemente as frequências mais altas. Assim, existe a necessidade de se utilizar o KF, o qual irá atuar apenas como um filtro, e não como estimador de estados propriamente, uma vez que o LQR sozinho garante apenas uma atenuação de -20 dB/dec para altas frequências, valor este muitas vezes não suficiente para um conversor chaveado.

De forma a controlar tanto a corrente no indutor i_L como a tensão no capacitor v_o , foi utilizada uma malha com configuração em cascata [60,61] com saturadores, a qual possui boa facilidade de lidar com o perfil de carga CC-CV. A Figura 15 apresenta o diagrama de blocos desta malha de controle.

Esta estratégia de controle pode ser descrita pelos seguintes passos:

- (i) O KF utiliza o valor da variação da ação de controle anterior

Figura 15 – Diagrama de blocos do LQG com configuração em cascata.



Fonte: produção do próprio autor, 2016

$\Delta \bar{u}(k-1)$ calculada e as leituras anteriores de $i_L(k-1)$ e $v_o(k-1)$ para realizar a filtragem de tais sinais, o que resulta nos sinais estimados $\hat{i}_L(k)$ e $\hat{v}_o(k)$ e nas suas respectivas diferenças $\Delta \hat{i}_L(k)$ e $\Delta \hat{v}_o(k)$.

- (ii) O valor de tensão estimada $\hat{v}_o(k)$, da sua variação $\Delta \hat{v}_o(k)$ e o valor da tensão de referência $r_v(k)$ são passados ao controlador de tensão LQR_v , que gera a partir de tais sinais uma referência de corrente $u_v(k)$ como saída, que tem por objetivo fazer a tensão $\hat{v}_o$ seguir seu valor de referência r_v .
- (iii) A referência de corrente $u_v(k)$ gerada pelo LQR_v é comparada com um valor de referência de corrente $r_i^*(k)$ estipulado. Se o valor $u_v(k)$ é maior que $r_i^*(k)$, então limita-se o valor da

referência r_i ao valor previamente estipulado, ou seja, $r_i(k) = r_i * (k)$. Caso o valor $u_v(k)$ seja menor que $r_i * (k)$, este então é passado como valor de referência de corrente, ou seja, $r_i(k) = u_v(k)$.

- (iv) A referência de corrente $r_i(k)$ é passada ao controlador de corrente LQR_i , juntamente com as estimações de $\hat{i}_L(k)$, $\hat{v}_o(k)$ e suas diferenças $\Delta\hat{i}_L(k)$ e $\Delta\hat{v}_o(k)$. A partir de tais sinais, é calculada a ação de controle $\Delta u(k) = \Delta d(k)$ de forma a seguir a referência de corrente $r_i(k)$.
- (v) Tendo-se $\Delta u(k)$, calcula-se o valor de razão cíclica $u(k)$, o qual é passado por um saturador resultando em uma ação de controle $\bar{u}(k)$ que é entregue ao modulador PWM e consequentemente ao conversor.
- (vi) A partir do sinal $\bar{u}(k)$ e do seu valor anterior $\bar{u}(k-1)$, tem-se o valor real de $\Delta\bar{u}(k)$ aplicado (o qual já leva em conta o efeito da saturação), que será utilizado na próxima iteração do KF juntamente com os valores de $i_L(k)$ e $v_o(k)$ lidos a partir do conversor.

Conforme exemplificado, esta estratégia de controle em cascata atende aos requisitos do protocolo de carga CC-CV. A transição entre as fases de carga em corrente constante e tensão constante ocorre de forma suave e natural, já que não há trocas bruscas nos controladores utilizados. O controle atuante é sempre o controlador de corrente LQR_i , o qual pode ter sua referência de corrente definida pelo controlador de tensão LQR_v ou não, no caso do valor de referência calculado ser menor que o previamente estipulado. Além disso, a corrente de saída sempre será limitada a um valor máximo predefinido (neste exemplo, 0,825 A), o que também pode ser de grande utilidade em casos de curto-circuito na saída, sob os quais o controlador não deixará a corrente ultrapassar o seu valor limite, reduzindo a razão cíclica o quanto for necessário para que isso aconteça. Desta forma, esta estratégia de controle agrega uma proteção a mais às baterias de forma inerente, resultando em uma operação com maior confiabilidade.

Conforme a Figura 15, esta estrutura de controle em cascata se utiliza de dois controladores, o de tensão LQR_v e o de corrente LQR_i , além do KF no estágio de realimentação. Como o modelo do conversor Duplo-Forward não possui integrador na origem, é necessária a utilização de uma ação de controle integral para os dois LQRs e para o KF. Desta forma, devem ser obtidas as matrizes que modelam o sistema em tal representação.

No caso do controlador de corrente, deseja-se controlar a corrente do indutor $i_L(k)$ a partir da razão cíclica $d(k)$. Partindo-se do modelo discreto do conversor Duplo-Forward descrito pela equação (3.8) e considerando a saída do sistema como sendo a corrente $i_L(k)$, conforme equação (3.9), ao utilizar o conceito de estado estendido descrito na seção 3.6.3, chega-se no seguinte modelo incremental em espaço de estados do conversor, conforme equação (3.23).

$$\begin{bmatrix} \Delta i_L(k) \\ \Delta v_o(k) \\ i_L(k) \end{bmatrix} = \begin{bmatrix} A & \mathbf{0} \\ C_i A & 1 \end{bmatrix} \begin{bmatrix} \Delta i_L(k-1) \\ \Delta v_o(k-1) \\ i_L(k-1) \end{bmatrix} + \begin{bmatrix} B \\ C_i B \end{bmatrix} \Delta d(k-1) \quad (3.23)$$

Reescrevendo-se a equação (3.23), tem-se o sistema descrito por (3.24).

$$\begin{bmatrix} \Delta i_L(k) \\ \Delta v_o(k) \\ i_L(k) \end{bmatrix} = [A_{ia}] \begin{bmatrix} \Delta i_L(k-1) \\ \Delta v_o(k-1) \\ i_L(k-1) \end{bmatrix} + [B_{ia}] \Delta d(k-1) \quad (3.24)$$

Para o controlador de tensão, deseja-se controlar a tensão $v_o(k)$ a partir da corrente $i_L(k)$, sendo que tal relação pode ser obtida através da própria equação (3.6), e encontra-se disposta em (3.25).

$$[\dot{v}_o(t)] = [-1/R_o C_o][v_o(t)] + [1/C_o][i_L(t)] \quad (3.25)$$

Discretizando via ZOH e utilizando o conceito da representação incremental em espaço de estados sobre (3.25), e ainda considerando a saída do conversor como sendo a tensão $v_o(k)$, conforme equação

(3.10), chega-se na expressão (3.26).

$$\begin{bmatrix} \Delta v_o(k) \\ v_o(k) \end{bmatrix} = \begin{bmatrix} A_v & 0 \\ C_v A_v & 1 \end{bmatrix} \begin{bmatrix} \Delta v_o(k-1) \\ v_o(k-1) \end{bmatrix} + \begin{bmatrix} B_v \\ C_v B_v \end{bmatrix} \Delta i_L(k-1) \quad (3.26)$$

Reescrevendo a expressão (3.26), tem-se a equação (3.27).

$$\begin{bmatrix} \Delta v_o(k) \\ v_o(k) \end{bmatrix} = [A_{va}] \begin{bmatrix} \Delta v_o(k-1) \\ v_o(k-1) \end{bmatrix} + [B_{va}] \Delta i_L(k-1) \quad (3.27)$$

De forma genérica, tendo-se as matrizes A e B do sistema e sendo tal sistema controlável, e ainda supondo que as matrizes de ponderação Q_{LQ} e R_{LQ} também são conhecidas, o ganho do controlador LQR em sua estrutura não recursiva pode ser encontrado através de (3.28).

$$K = R_{LQ}^{-1} B^T P_{LQ} \quad (3.28)$$

Onde P é a solução da Equação Algébrica de Riccati:

$$A^T P_{LQ} + P_{LQ} A - (P_{LQ} B) R_{LQ}^{-1} (B^T P_{LQ}) + Q_{LQ} = 0 \quad (3.29)$$

A escolha pela estrutura não-recursiva do LQR foi feita com o objetivo de calcular o ganho K *off-line* e assim poupar esforços computacionais inerentes do cálculo recursivo do ganho do compensador.

A partir dos modelos obtidos nas equações (3.24) e (3.27), tem-se que as matrizes A e B são A_{ia} e B_{ia} para o controlador de corrente e A_{va} e B_{va} para o controlador de tensão. Os passos para a obtenção das matrizes de ponderação do LQR_i (Q_{ia} e R_{ia}) e do LQR_v (Q_{va} e R_{va}) serão comentados na seção 3.6.5.

Agora, com relação ao Filtro de Kalman, a estimação dos estados pode ser escrita conforme equação (3.30).

$$\hat{X}_a(k) = A \hat{X}_a(k-1) + B \Delta u(k-1) + L(Y(k-1) - \hat{Y}(k-1)) \quad (3.30)$$

Esta forma de estimação também é conhecida como Preditor de Kalman, por utilizar os valores passados da saída para a estimação atual do vetor de estados, os quais são multiplicados pelo ganho de

estimação L . Existe a necessidade de “filtragem” (ou predição) de ambas as variáveis $i_L(k)$ e $v_o(k)$. Logo, para que seja obtido o modelo do Duplo-Forward na forma incremental levando-se em conta ambas as variáveis, tanto $i_L(k)$ quanto $v_o(k)$ serão utilizados como saídas do sistema, o que implica em (3.31).

$$Y(k) = C_{kf} [i_L(k) \ v_o(k)]^T = [i_L(k) \ v_o(k)]^T \quad (3.31)$$

Em suma, tem-se $C_{kf} = I$, onde I é a Matriz Identidade. Utilizando-se o conceito de estado estendido da seção 3.6.3, tem-se então um modelo de quarta ordem, conforme descrito em (3.32).

$$\begin{bmatrix} \Delta i_L(k) \\ \Delta v_o(k) \\ i_L(k) \\ v_o(k) \end{bmatrix} = \begin{bmatrix} A & \mathbf{0} \\ C_{kf} A & I \end{bmatrix} \begin{bmatrix} \Delta i_L(k-1) \\ \Delta v_o(k-1) \\ i_L(k-1) \\ v_o(k-1) \end{bmatrix} + \begin{bmatrix} B \\ C_{kf} B \end{bmatrix} \Delta d(k-1) \quad (3.32)$$

Reescrevendo (3.32), tem-se a expressão (3.33).

$$\begin{bmatrix} \Delta i_L(k) \\ \Delta v_o(k) \\ i_L(k) \\ v_o(k) \end{bmatrix} = [A_{kf}] \begin{bmatrix} \Delta i_L(k-1) \\ \Delta v_o(k-1) \\ i_L(k-1) \\ v_o(k-1) \end{bmatrix} + [B_{kf}] \Delta d(k-1) \quad (3.33)$$

Novamente, de forma genérica, sabendo-se as matrizes A e C de um sistema que é observável, e supondo que as matrizes Q_{LQ} e R_{LQ} são conhecidas, o ganho do KF em sua forma não recursiva pode ser obtido através da relação dada por (3.34).

$$L = R_{LQ}^{-1} C_{kf} P_{LQ} \quad (3.34)$$

Onde P_{LQ} é a solução da Equação Algébrica de Riccati dada por (3.35).

$$A_{kf} P_{LQ} + P_{LQ} A_{kf}^T - (P_{LQ} C_{kf}^T) R_{LQ}^{-1} (C_{kf} P_{LQ}) + Q_{LQ} = 0 \quad (3.35)$$

No caso do KF, as matrizes A e C estão representadas pelas

matrizes A_{kf} e C_{kf} , respectivamente, conforme equação (3.33). Novamente, as matrizes de ponderação do KF, aqui representadas por Q_{LQ} e R_{LQ} , também serão encontradas na seção 3.6.5. A seguir, será feita a análise de controlabilidade, observabilidade e estabilidade dos modelos em malha aberta utilizados nos LQR_s e no KF.

3.6.4.1 Análise de controlabilidade, observabilidade e estabilidade

Serão tratados aqui os conceitos de controlabilidade, observabilidade e estabilidade em malha aberta do sistema a ser trabalhado. Porém, como a bateria se comporta como uma carga de resistência variável perante o conversor, será admitido aqui o pior caso sob a perspectiva de controle, que é quando a resistência de carga atinge seu maior valor, diminuindo o amortecimento do sistema na frequência de ressonância. Tal valor é encontrado através da equação (3.36) e será utilizado durante todas as análises.

$$Ro = \frac{V_{o_{max}}}{I_{o_{min}}} = \frac{10 \text{ V}}{50 \text{ mA}} = 200 \text{ } \Omega \quad (3.36)$$

Para a realização do projeto dos controladores de corrente LQR_i e de tensão LQR_v , deve-se garantir que tais sistemas são controláveis, pois só assim pode-se chegar a uma solução factível e estável do sistema realimentado.

A partir da modelagem do sistema no tempo contínuo mostrada na equação (3.6) e da sua discretização a partir do ZOH ($f_a = 20 \text{ kHz}$), foram montadas as respectivas representações incrementais do modelo para cada situação, conforme abordado na seção 3.6.3.

No caso do LQR_i , necessita-se do modelo apresentado na equação (3.23), o qual resulta nos seguintes valores:

$$\begin{bmatrix} \Delta i_L(k) \\ \Delta v_o(k) \\ i_L(k) \end{bmatrix} = \begin{bmatrix} 0,97 & -1,16 & 0 \\ 0,05 & 0,97 & 0 \\ 0,97 & -1,16 & 1 \end{bmatrix} \begin{bmatrix} \Delta i_L(k-1) \\ \Delta v_o(k-1) \\ i_L(k-1) \end{bmatrix} + \begin{bmatrix} 30,17 \\ 0,81 \\ 30,17 \end{bmatrix} \Delta d(k-1) \quad (3.37)$$

Já para o LQR_v , o qual utiliza o modelo apresentado pela

relação (3.26), tem-se os seguintes valores numéricos.

$$\begin{bmatrix} \Delta v_o(k) \\ v_o(k) \end{bmatrix} = \begin{bmatrix} 0,99 & 0 \\ 0,99 & 1 \end{bmatrix} \begin{bmatrix} \Delta v_o(k-1) \\ v_o(k-1) \end{bmatrix} + \begin{bmatrix} 0,05 \\ 0,05 \end{bmatrix} \Delta i_L(k-1) \quad (3.38)$$

A análise de controlabilidade do sistema é feita através da matriz de controlabilidade do modelo, a qual deve possuir um *rank* igual a ordem do sistema. Com relação ao Filtro de Kalman, uma análise semelhante deve ser realizada, buscando saber se o sistema é observável, através da matriz de observabilidade [62].

Através do modelo incremental mostrado em (3.32), chega-se nos valores mostrados em (3.39) para o modelo incremental a ser utilizado pelo Filtro de Kalman.

$$\begin{bmatrix} \Delta i_L(k) \\ \Delta v_o(k) \\ i_L(k) \\ v_o(k) \end{bmatrix} = \begin{bmatrix} 0,969 & -1,164 & 0 & 0 \\ 0,053 & 0,969 & 0 & 0 \\ 0,969 & -1,164 & 1 & 0 \\ 0,053 & 0,969 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta i_L(k-1) \\ \Delta v_o(k-1) \\ i_L(k-1) \\ v_o(k-1) \end{bmatrix} + \begin{bmatrix} 30,17 \\ 0,81 \\ 30,17 \\ 0,81 \end{bmatrix} \Delta d(k-1) \quad (3.39)$$

Analisando-se tais modelos dados por (3.37), (3.38) e (3.39), tem-se que suas respectivas matrizes de controlabilidade e observabilidade possuem rank igual à ordem do modelo, atestando a condição de controlabilidade e observabilidade necessárias. Além disso, também é possível determinar que os autovalores dos modelos estão contidos dentro do círculo unitário, garantindo também a estabilidade em malha aberta dos mesmos.

3.6.5 Encontrando as matrizes de ponderação

Estratégias de controle ótimo possuem uma metodologia bem definida para a obtenção dos ganhos do compensador de um sistema realimentado [62], exceto pela definição das matrizes de ponderação Q_{LQ} e R_{LQ} , que geralmente são escolhidas com base no conhecimento prévio da planta ou pela utilização de dados empíricos do sistema, o que resulta em uma fase não sistematizada de projeto do compensador e sem garantias de que o sistema controlado terá um bom comportamento dinâmico [63].

No caso desta dissertação, deve-se realizar a sintonia dos controladores LQR_i e LQR_v e do Filtro de Kalman, que caracterizam o seguinte cenário:

- A malha interna de corrente é um sistema de ordem 3 com uma saída, resultando em matrizes de ponderação $Q_{LQ.ia(3 \times 3)}$ e $R_{LQ.ia(1 \times 1)}$.
- A malha externa de tensão é um sistema de ordem 2 com uma saída, resultando em matrizes de ponderação $Q_{LQ.va(2 \times 2)}$ e $R_{LQ.va(1 \times 1)}$.
- O KF será baseado em um sistema de 4ª ordem com 2 saídas, resultando em matrizes de ponderação $Q_{LQ.kf(4 \times 4)}$ e $R_{LQ.kf(2 \times 2)}$.

Considerando-se que todas as matrizes de ponderação são diagonais, existem 13 parâmetros a serem definidos para a realização do LQG , o que dificulta a obtenção dos ganhos dos controladores e do KF a partir de métodos não sistematizados como o de tentativa e erro, devido à grande quantidade de parâmetros.

Visando utilizar um método sistemático e que garanta bons resultados dinâmicos ao sistema de controle como um todo, foi utilizado o *Particle Swarm Optimization* (PSO - Método de Enxame de Partículas), que é um algoritmo de otimização de funções contínuas e não lineares [64], e que será comentado em detalhes na seção 3.6.6.

3.6.6 Particle Swarm Optimization

A utilização de algoritmos de otimização como o PSO ou *Genetic Algorithm* (GA - Algoritmos Genéticos) na determinação das matrizes de ponderação em controladores LQ é um assunto já conhecido na literatura. Em [65] o GA é utilizado na obtenção das matrizes de ponderação de um LQR utilizado para controlar um pêndulo invertido. Já [66] faz o uso do PSO na determinação das matrizes Q_{LQ} e R_{LQ} de um LQR utilizado na suspensão ativa de um veículo guiado por trilhos. Em [63] foi feita a utilização de ambos PSO e GA

na determinação das matrizes de ponderação de um LQR aplicado no controle de sistema de pressão. As matrizes de peso sintonizadas pelos algoritmos PSO e GA foram comparadas com tentativas de sintonia manuais, apresentando resultados mais satisfatórios que tais tentativas manuais, e se mostraram como alternativas sistematizadas na determinação das matrizes de ponderação de estratégias de controle ótimo. Além dos métodos supracitados, há outras alternativas sistematizadas para a obtenção das matrizes Q_{LQ} e R_{LQ} , como é o caso da Regra de Bryson [67], que não se utiliza de Inteligência Computacional, mas da normalização das variáveis de estado $X(k)$ e da ação de controle $u(k)$.

O algoritmo PSO empregado neste trabalho pode ser descrito pelos seguintes passos:

- inicialização do algoritmo PSO.
 - (i) criar um espaço de busca, o qual limita o domínio das partículas $x(i)$.
 - (ii) criar n (tamanho do enxame) candidatos $x(i)$ aleatórios (cada candidato contendo uma possível solução, ou seja, 13 parâmetros) dentro do espaço de busca.
 - (iii) determinar quem é o melhor candidato (x_{global})
 - (iv) definir cada candidato atual $x(i)$ como sendo o melhor valor já encontrado x_{local} pela partícula i , ou seja, $x_{local}(i) = x(i)$.
 - (v) inicializar a inércia $\Delta x(i)$ de cada partícula i (zero ou valores aleatórios).
- processo iterativo (até atingir o valor de mínimo da função *fitness* ou o limite máximo de iterações).
 - (i) calcular o vetor de deslocamento $\Delta x(i)$ da partícula $x(i)$, conforme equação (3.41), e atualizar a posição da partícula $x(i)$.
 - (ii) avaliar os candidatos $x(i)$ na função de *fitness* J_{PSO} , dada pela equação (3.42).

- (iii) comparar o valor da função J_{PSO} obtido por cada i -ésimo candidato com o melhor valor já obtido anteriormente (melhor global) e com o melhor valor que a i -ésima partícula já encontrou (melhor local).
 - (iv) atualizar dos valores x_{global} e $x_{local}(i)$, caso seja pertinente.
 - (v) verificar requisitos de parada do algoritmo.
- retorna os valores encontrados.

A codificação da partícula $x(i)$ é composta da seguinte forma:

$$x = [\text{diag}(Q_{LQ.ia}) \quad \text{diag}(Q_{LQ.va}) \quad \text{diag}(Q_{LQ.kf}) \quad R_{LQ.ia} \quad R_{LQ.va} \quad \text{diag}(R_{LQ.kf})] \quad (3.40)$$

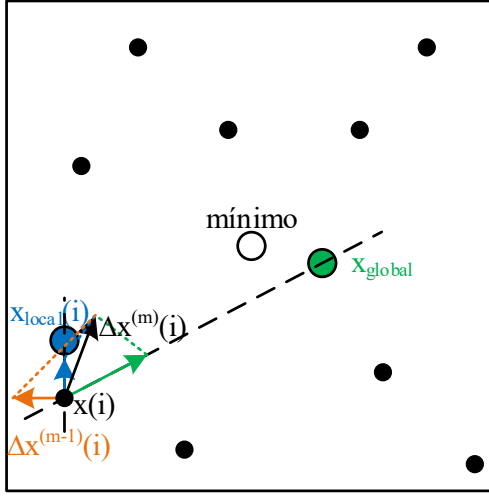
Já o vetor de deslocamento de uma partícula qualquer no espaço de busca é obtido através de uma combinação linear de três vetores (ver Figura 16): i) um vetor que aponta na direção de x_{global} (verde), ii) um vetor que aponta da direção de $x_{local}(i)$ (azul) e iii) um vetor que aponta para a direção do deslocamento da partícula feito na iteração anterior, representando a inércia da mesma (laranja). Este vetor de deslocamento pode ser escrito conforme equação (3.41).

$$\begin{cases} \Delta x^{(m)}(i) = \alpha_1 [x_{local}^{(m)}(i) - x(i)] + \alpha_2 [x_{global}^{(m)} - x(i)] + \alpha_3 \Delta x^{(m-1)}(i) \\ x(i)^{(m)} = x(i)^{(m-1)} + \Delta x(i)^{(m)} \end{cases} \quad (3.41)$$

Os valores α_1 (cognição), α_2 (relacionamento social) e α_3 (inércia) são utilizados para ponderar os vetores da combinação linear podendo assim incorporar características de buscas mais locais ou globais ao algoritmo, dependendo das suas escolhas. Foram adotados aqui valores aleatórios que mudam a cada iteração, podendo variar de 0 até um valor de amplitude pré-determinado (ver Tabela 4). A Figura 16 ilustra a forma como o vetor de deslocamento é obtido dentro de um espaço de busca bidimensional.

A função *fitness* definida, apresentada em (3.42), almeja minimizar o somatório da variação dos estados $\Delta X(k)$, da variação da ação de controle $\Delta u(k)$ e do erro $r(k)-y(k)$, sendo este erro de

Figura 16 – Ilustração do deslocamento de uma partícula $x(i)$ pelo espaço de busca



Fonte: produção do próprio autor, 2016

corrente ou o erro de tensão, para que ambos sejam avaliados.

$$J_{PSO} = \sum_{k=0}^N \frac{1}{2} T_a \left[\Delta X(k) Q_{PSO} \Delta X^T(k) + \Delta u(k) R_{PSO} \Delta u^T(k) + \left(\frac{r(k) - y(k)}{r(k)} \right)^2 \right] \quad (3.42)$$

O algoritmo PSO é empregado de forma que, para cada partícula $x(i)$, seja executada uma simulação da carga de uma bateria por um tempo suficiente (de N iterações) para que ocorra tanto a etapa de carga de corrente constante do perfil CC-CV quanto a etapa de carga de tensão constante. Estas necessitam também da modelagem da bateria, que é apresentada na seção 4.1 e será melhor discutida no capítulo 4.

Para diminuir o número de iterações necessárias para a simulação da bateria para cada uma das partículas, fixou-se a condição inicial de carga da bateria em $SOC_o = 0,73$, fazendo com que a simulação da bateria inicie na fase de carga de corrente constante, porém já estando muito próxima de trocar para a fase de carga em tensão

constante. Com tal condição inicial, é possível obter uma simulação da carga da bateria operando nestas duas fases do protocolo CC-CV com $N = 600$ amostras, o que representa um tempo de simulação de $0,03 \text{ s}$ (considerando $f_a = 20 \text{ kHz} = 1/T_a$). Ao final de cada simulação de carga da bateria executada, o valor da função J_{PSO} é obtido e avaliado, perante os valores de Q_{LQ} e R_{LQ} escolhidos para os LQR s e para o KF (os quais são os valores da partícula $x(i)$). O algoritmo PSO então se utiliza dos valores J_{PSO} de cada uma das partículas para que, recursivamente, se alcance o menor valor possível da função *fitness* J_{PSO} . A execução do algoritmo PSO, bem como a simulação da carga das baterias para cada partícula do PSO, é realizada *offline*.

Na função *fitness* J_{PSO} , as matrizes de peso Q_{PSO} e R_{PSO} foram adotadas de forma a se balancear todos os termos da equação, para que ambos tenham o mesmo nível de importância, seguindo-se o raciocínio análogo ao da Regra de Bryson [67]. O termo referente ao erro já é normalizado pelo próprio valor de referência $r(k)$. Na equação 3.42, $r(k)$ representa a referência de corrente, enquanto $y(k)$ a corrente $i_L(k)$ como saída. Já as matrizes de peso foram definidas conforme as relações dadas em (3.43).

$$\begin{cases} Q_{PSO} = \text{diag}(1/\Delta i_{L_{max}}^2 & 1/\Delta v_{o_{max}}^2) \\ R_{PSO} = \text{diag}(1/\Delta u_{i_{max}}^2 & 1/\Delta u_{v_{max}}^2) \end{cases} \quad (3.43)$$

Os valores de $\Delta i_{L_{max}}$ e $\Delta v_{o_{max}}$ podem ser encontrados a partir das equações (3.44) e (3.45) que foram baseadas nas equações a diferenças da corrente no indutor e da tensão no capacitor, enquanto os valores $\Delta u_{i_{max}}$ $\Delta u_{v_{max}}$ são definidos a partir das equações (3.46) e (3.47), onde foram considerados a razão cíclica máxima e a corrente máxima que passa por uma única célula de bateria.

$$\Delta i_{L_{max}} = V_{in_{max}} T_a / L_o = 31,96 \text{ A} \quad (3.44)$$

$$\Delta v_{o_{max}} = \Delta i_{c_{max}} T_a / C_o = 0,085 \text{ V} \quad (3.45)$$

$$\Delta u_{i_{max}} = D_{max} = 0,45 \quad (3.46)$$

$$\Delta u_{v_{max}} = I_{o_{max}}/4 = 3 \text{ A} \quad (3.47)$$

Os parâmetros utilizados no algoritmo PSO estão dispostos na Tabela 4.

Tabela 4 – Parâmetros utilizados no algoritmo PSO

Parâmetro	Valor
x_{min}	$10^{-3} \cdot [1 \ 1 \ 1 \ 1 \ 1 \ 0,1 \ 0,1 \ 0,1 \ 0,1 \ 1 \ 1 \ 1 \ 0,1 \ 0,1]$
x_{max}	$10^5 \cdot [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 100 \ 100 \ 100 \ 100]$
tamanho do enxame	15
parâmetros	13
limite iterações	50
α_1	[0 0,45]
α_2	[0 0,3]
α_3	[0 0,25]

Fonte: produção do próprio autor, 2016

Nota-se que os limites de x foram estabelecidos em um espaço de busca amplo (pouco restritivo), assumindo-se pouco conhecimento do sistema.

Como a amplitude de α_1 (cognição) é maior que a de α_2 (relacionamento social), há uma tendência maior de se realizar buscas mais locais (apontando para $x_{local}(i)$, o que aumenta o tempo de convergência do algoritmo mas diminui as chances de que o mesmo convirja para mínimos locais).

Após a execução do algoritmo PSO para encontrar as matrizes de ponderação dos controladores e do KF, foram obtidos os seguintes

resultados:

$$\left\{ \begin{array}{l} Q_{LQ.ia} = 10^4 \text{diag}(3, 3170 \ 2, 1201 \ 1, 217) \\ R_{LQ.ia} = 10^6 1, 428829 \\ Q_{LQ.va} = 10^4 \text{diag}(1, 6109 \ 4, 6040) \\ R_{LQ.va} = 10^6 1, 482734 \\ Q_{LQ.kf} = 10^4 \text{diag}(3, 2055 \ 2, 3081 \ 3, 5287 \ 2, 7179) \\ R_{LQ.kf} = 10^6 \text{diag}(1, 821372 \ 3, 973014) \end{array} \right. \quad (3.48)$$

Tais matrizes de ponderação resultaram em estruturas realimentadas estáveis e foram utilizadas na implementação do carregador de baterias. Salienta-se que não foram realizados estudos estatísticos ou mais aprofundados sobre os resultados dos controladores obtidos através da utilização do PSO na determinação das matrizes de peso, uma vez que o principal foco era na utilização de uma metodologia sistematizada de obtenção de tais matrizes, o que foi atingido. Ademais, a escolha dos parâmetros do PSO foi pouco explorada nesta pesquisa, havendo ainda possibilidade de ajustes melhores dos parâmetros do algoritmo PSO dos que os mostrados na Tabela 4, com a chance de obtenção de melhores resultados, já que o foco deste trabalho é a utilização do PSO como uma simples ferramenta visando sistematização. Assim, um estudo mais rigoroso sobre este método utilizado e sobre a sintonia dos seus parâmetros está fora do escopo aqui delineado.

Todos os códigos utilizados nas etapas de projeto encontram-se disponibilizados na seção B.2 do Anexo B. A seguir, é mostrado um resumo da metodologia de projeto utilizada para obtenção do controlador.

3.6.7 Resumo das etapas de projeto do controlador

As etapas de projeto descritas anteriormente mostram a obtenção dos controladores LQR utilizados e do Filtro de Kalman, compondo assim o LQG . Todas estas etapas serão resumidas aqui de forma

a facilitar o entendimento do leitor quanto a metodologia utilizada.

Metodologia de projeto do controlador LQG :

- (i) Encontrar o modelo contínuo do conversor em espaço de estados.
- (ii) Utilizar a transformada ZOH e encontrar o equivalente discreto do modelo em espaço de estados.
- (iii) Obter a representação incremental do modelo discreto para o LQR_i , LQR_v e para o KF (cada controlador / observador requer uma representação única neste caso).
- (iv) Encontrar as matrizes de ponderação Q_{LQ} e R_{LQ} dos $LQRs$ e do KF através do algoritmo PSO.
- (v) Usar as matrizes Q_{LQ} e R_{LQ} encontradas para obter os ganhos dos controladores e do KF.

Percorridas estas etapas, foram obtidos o ganho K_i do controlador de corrente, o ganho K_v do controlador de tensão e o ganho L do Filtro de Kalman, os quais encontram-se dispostos em (3.49), (3.50) e (3.51), respectivamente. Lembrando que o ganho K_v do LQR_v multiplica os estados do modelo incremental de (3.27) resultando em $\Delta u_v(k) [A]$, o ganho K_i do LQR_i multiplica os estados do modelo incremental dado por (3.24) resultando em $\Delta u(k)[adimensional]$ (lei de controle $\Delta u(k) = -KX$) e o ganho L do KF multiplica o erro de estimação passado, dados através dos vetores $Y(k-1) = [i_L(k-1) \ v_o(k-1)]^T$ e $\hat{Y}(k-1) = [\hat{i}_L(k-1) \ \hat{v}_o(k-1)]^T$, produzindo o vetor de estados estimados dado por (3.33).

$$K_i = \begin{bmatrix} 0.0313 & -0.0378 & 0.0143 \end{bmatrix} \quad (3.49)$$

$$K_v = \begin{bmatrix} 2.4036 & 0.1645 \end{bmatrix} \quad (3.50)$$

$$L = \begin{bmatrix} 0,281 & -0,084 \\ -0,023 & 0,018 \\ 0,877 & -0,151 \\ -0,167 & 0,167 \end{bmatrix} \quad (3.51)$$

Tendo os ganhos em mãos, foram obtidos os resultados de simulação e experimentais do carregador de baterias controlado, que serão apresentados na seção 3.7.

3.7 RESULTADOS DO CONTROLE DO CARREGADOR DE BATERIAS

Os resultados do controlador LQG utilizado no carregador de baterias foram divididos em três etapas: i) Resposta em frequência dos componentes do LQG projetado, ii) resultados simulados do controlador e iii) resultados experimentais do controlador.

3.7.1 Resposta em frequência dos controladores

A análise de controladores no domínio da frequência é amplamente utilizada na eletrônica de potência, e por este motivo, foi realizada nesta dissertação.

Os resultados sobre os aspectos frequenciais do LQG como um todo foram obtidos através de simulações, considerando todos os componentes ideais, tanto os que compõe o conversor quanto os circuitos de condicionamento de sinais. Para todas as malhas projetadas foram traçados 3 diagramas de Bode. O primeiro, é a resposta em frequência da planta sem compensação $G(z)$, obtida através da representação genérica apresentada na equação (3.52). O segundo, é a resposta do sistema compensado em malha aberta $FTMA(z)$, obtido através da relação genérica dada por (3.53). O terceiro, é a resposta do sistema em malha fechada $FTMF(z)$, fornecida pela equação (3.54).

$$G(z) = C(sI - A)^{-1}B + D \quad (3.52)$$

$$FTMA(z) = K(sI - A)^{-1}B + D \quad (3.53)$$

$$FTMF(z) = C(sI - (A - BK))^{-1}(BK) + D \quad (3.54)$$

No projeto de controladores em conversores estáticos, algumas recomendações devem ser observadas no sistema compensado em malha aberta:

- alto ganho em baixas frequências: necessário para que haja rastreamento de referência. Em geral, para sistemas que não possuem integrador puro, o integrador na origem é utilizado para garantir erro nulo em regime permanente.
- atenuação em altas frequências: este requisito evita que os ruídos de alta frequência, em especial os de chaveamento, não sejam amplificados pelo controlador. Em geral, busca-se uma atenuação de -40 dB/dec em frequências da ordem de f_s ou maiores. Este é o motivo da utilização do KF neste projeto.
- frequência de cruzamento f_{cross} da curva de ganho da $FTMA(z)$ por 0 dB pelo menos 4 vezes menor que f_s : este item está ligado ao item anterior, assim, uma frequência de cruzamento menor garante melhor atenuação dos ruídos de chaveamento.
- cruzamento da curva de ganho por 0 dB com inclinação de -20 dB/dec .
- boa margem de fase: busca-se aqui um bom compromisso entre robustez de operação e rapidez na atuação do controlador. Assim, para cada projeto deve ser adotado um valor específico, que garanta um bom comportamento transitório sem comprometer a confiabilidade de operação, sendo frequentemente empregados valores de 30 a 60° .
- operação do controlador abaixo da frequência de amostragem: além de operar abaixo de f_s , deve-se projetar o controlador de forma que ele não atue próximo da frequência de amostragem f_a , onde começa-se a perder informações (amostrais) do sistema.

Apesar da frequência de Nyquist teórica ser $f_a/2$, recomenda-se na prática ao menos 4 amostras por período para que haja maior representatividade, resultando em uma frequência máxima de operação de $f_a/4$.

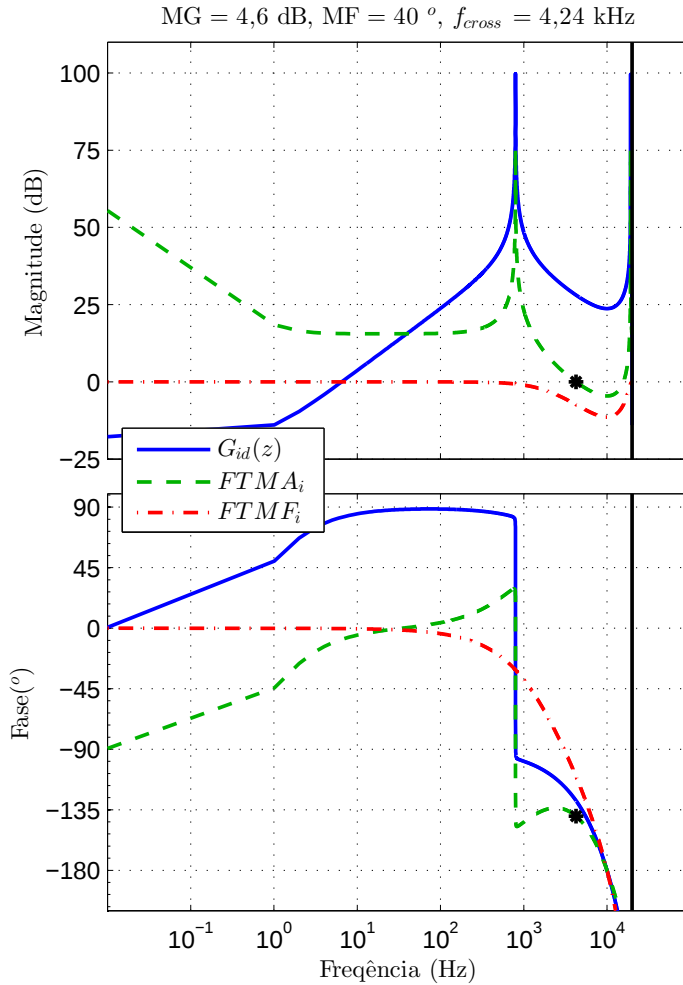
Uma vez conhecidos os aspectos frequenciais desejados no projeto dos controladores, passa-se então aos diagramas de Bode dos mesmos.

A Figura 17 mostra os diagramas de Bode referentes à malha interna de corrente, que utiliza o controlador LQR_i . Nota-se que a $FTMA_i$ do sistema possui um integrador na origem, atenuação em 0 dB e em altas frequências de -20 dB/dec , margem de fase $MF = 40^\circ$, assegurando bom compromisso entre rapidez e robustez, e frequência de cruzamento $f_{cross} = 4,24 \text{ kHz}$, valor este que é aproximadamente 5 vezes menor que f_a e 20 vezes menor que f_s , o que assegura boa confiabilidade ao sistema de controle, atendendo a todos os requisitos desejados.

Com relação ao controlador LQR_v , observa-se na Figura 18 que o mesmo também possui integrador na origem, atenuação em 0 dB e em altas frequências de -20 dB/dec , porém sua margem de fase é ainda maior, estando em $MF = 61^\circ$. Sua frequência de cruzamento é $f_{cross} = 460 \text{ Hz}$, estando mais que suficientemente longe de f_a e f_s , além de estar aproximadamente uma década abaixo da frequência de corte do LQR_i . Esta última característica é importante na configuração de controle em cascata, pois o controlador de corrente deve ser suficientemente mais rápido que a malha de tensão para que este seja enxergado apenas como um ganho estático pelo controlador de tensão. Estes dados atestam então que o controlador de tensão, em sua fase de projeto, também atende a todos os requisitos necessários.

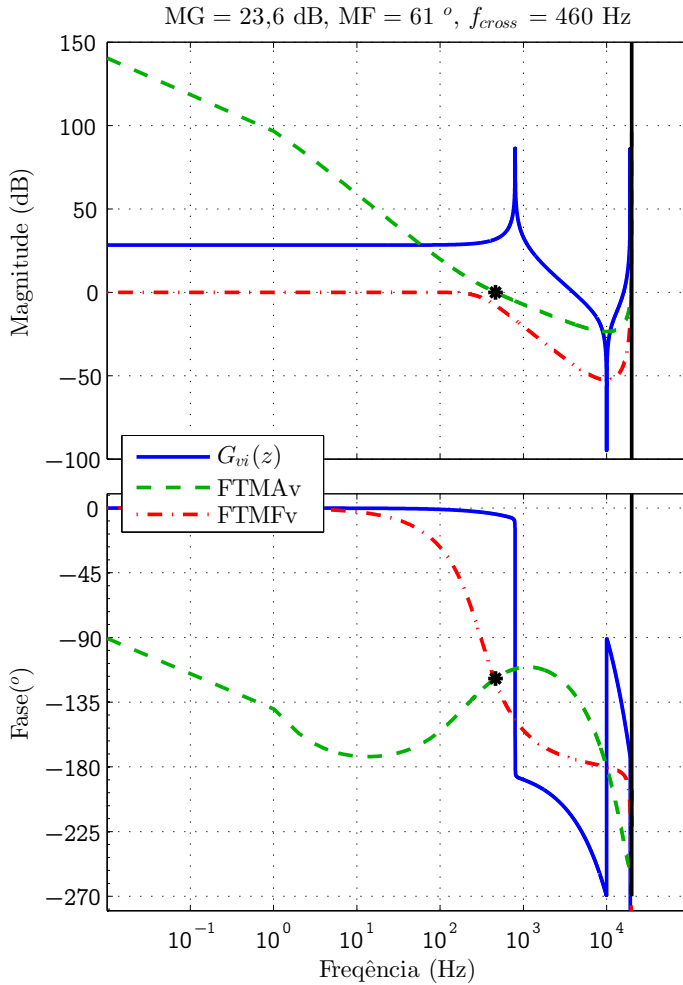
Por fim, a malha do Filtro de Kalman deve ser a responsável por filtrar ruídos, especialmente os do tipo gaussiano (ruído branco), que em geral são característicos nas frequências mais elevadas, o que acaba por auxiliar na atenuação de sinais próximos à frequência de comutação (mais uma atenuação de -20 dB/dec). É importante ainda destacar que, para que o KF seja capaz de “rastrear” (ou predizer) sinais com erro nulo em regime permanente, este também necessita de

Figura 17 – Diagrama de Bode da malha de interna de corrente: Planta sem compensador ($G_{id}(z)$ em azul), planta com compensador em malha aberta ($FTMA_i$ em verde) e planta compensada em malha fechada ($FTMF_i$ em vermelho).



Fonte: produção do próprio autor, 2016

Figura 18 – Diagrama de Bode da malha de externa de tensão: Planta sem compensador ($G_{vi}(z)$ em azul), planta com compensador em malha aberta ($FTMA_v$ em verde) e planta compensada em malha fechada ($FTMF_v$ em vermelho).



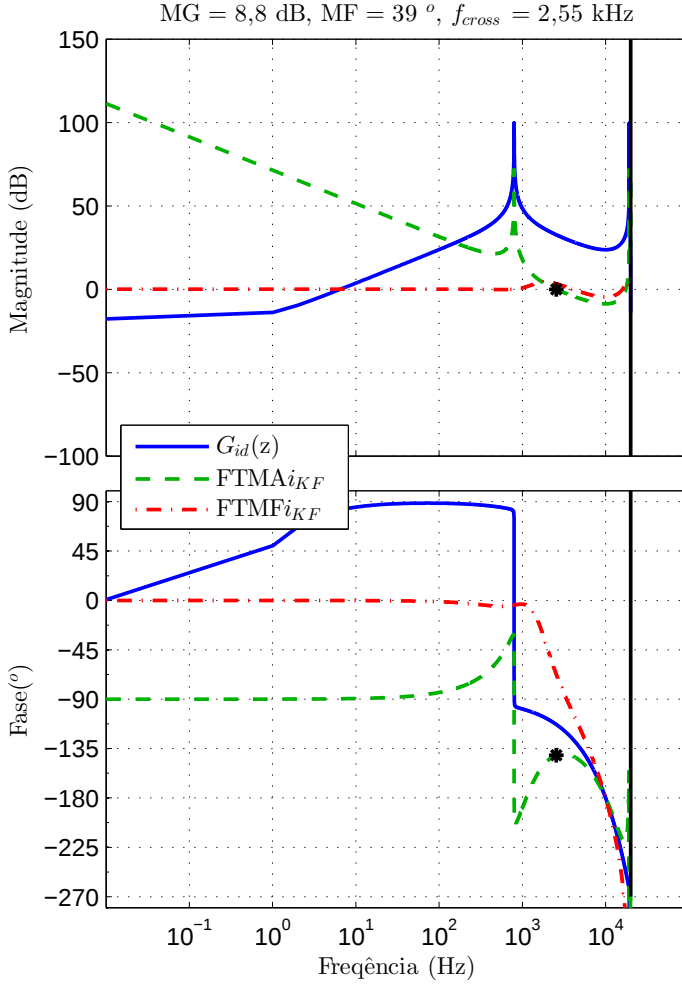
Fonte: produção do próprio autor, 2016

um integrador na origem, o que justifica a utilização da representação incremental durante o seu projeto.

O Filtro de Kalman foi projetado contendo duas saídas, i_L e v_o (conforme equação 3.31), havendo assim dois diagramas de Bode a serem analisados, um para cada saída. A Figura 19 mostra o diagrama de Bode da saída $i_L(k)$ em função da entrada $d(k)$, enquanto o diagrama de Bode da Figura 20 mostra o sistema contendo a saída $v_o(k)$ e a entrada $i_L(k)$. Nota-se a partir de tais figuras que o KF projetado atende aos requisitos necessários, atenuando sinais com frequências acima de $f_{cross} = 2,55 \text{ kHz}$ no caso do sistema com saída em corrente e $f_{cross} = 1,01 \text{ kHz}$ no caso do sistema com saída em tensão. As margens de fase ficaram em 39° e 52° , respectivamente.

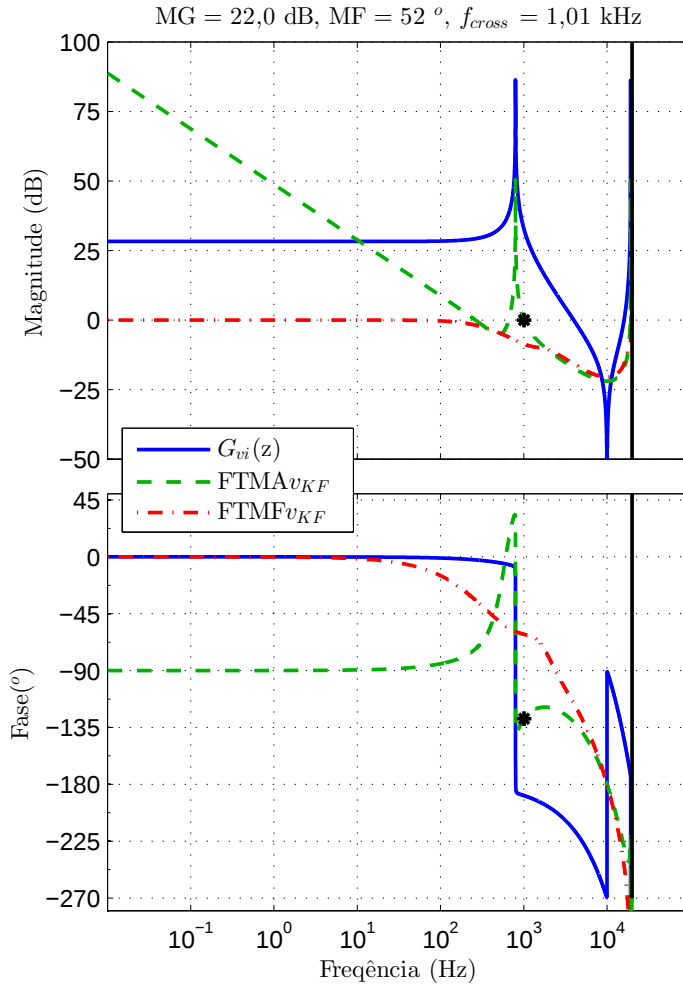
Mostrados os aspectos frequenciais dos componentes do LQG , passa-se agora aos resultados obtidos por simulação.

Figura 19 – Diagrama de Bode do KF (corrente): Planta em malha aberta ($G_{id}(z)$ em azul), planta com KF em malha aberta ($FTMA_{i_{KF}}$ em verde) e planta com KF em malha fechada ($FTMF_{i_{KF}}$ em vermelho).



Fonte: produção do próprio autor, 2016

Figura 20 – Diagrama de Bode do KF (tensão): Planta em malha aberta ($G_{vi}(z)$ em azul), planta com KF em malha aberta ($FTMA_{vKF}$ em verde) e planta com KF em malha fechada ($FTMF_{vKF}$ em vermelho).



Fonte: produção do próprio autor, 2016

3.7.2 Análise do controlador por simulação

Nesta seção, serão avaliados o comportamento transitório e de regime permanente do controlador e do conversor no domínio do tempo. Todos os resultados aqui obtidos, com exceção dos degraus de carga, utilizaram-se de uma carga constante de $0,8 \Omega$, para que pudessem ser confrontados com os resultados experimentais obtidos com a mesma carga constante, e para fossem obtidos os índices mais comuns de análise de resposta de controladores, como máximo sobressinal e tempo de assentamento.

Para os degraus na referência, serão utilizadas duas estratégias, visando validar o comportamento tanto do controle de tensão quanto do controle de corrente. Para validar o controle da tensão de saída do conversor, o valor limite da referência de corrente será mantido em um valor muito alto, de forma que este não influencie no controle de tensão. Será então dado um degrau na referência de tensão, e serão avaliados os sinais de tensão de saída do conversor, corrente no indutor, suas estimações obtidas através do KF e os sinais de controle u_v e u_i gerados pelo LQR_v e pelo LQR_i , respectivamente. Já para avaliar o controle de corrente, a única diferença em relação à configuração anterior é a utilização de um limite de referência de corrente mais baixo, de forma a limitar a corrente de saída do conversor, simulando a etapa de carga das baterias em corrente constante.

A Figura 21 mostra a resposta simulada do sistema frente a um degrau de $4,2 V$ dado na referência de tensão, partindo de condições nulas. Nota-se que os sinais de tensão de saída $\hat{v}_o$ e corrente no indutor $\hat{i}_L$ estimados (em azul), representam muito bem os sinais reais v_o e i_L (vermelho), com boa atenuação das componentes de alta frequência. Nota-se ainda que a tensão v_o atinge o valor médio de referência de $4,2 V$, enquanto a corrente no indutor acompanha o seu comportamento estabilizando no valor de $5,2 A$, o que é esperado dada uma carga resistiva de $0,8 \Omega$ e não havendo limitação no valor de corrente que o controlador pode impor. O tempo de estabilização da tensão de saída nesta simulação ficou próximo de $3 ms$, enquanto o sobressinal da tensão ficou em torno de 7%. Quanto aos sinais de controle, nota-se que a saída do controlador de tensão u_v tem um perfil próximo ao perfil

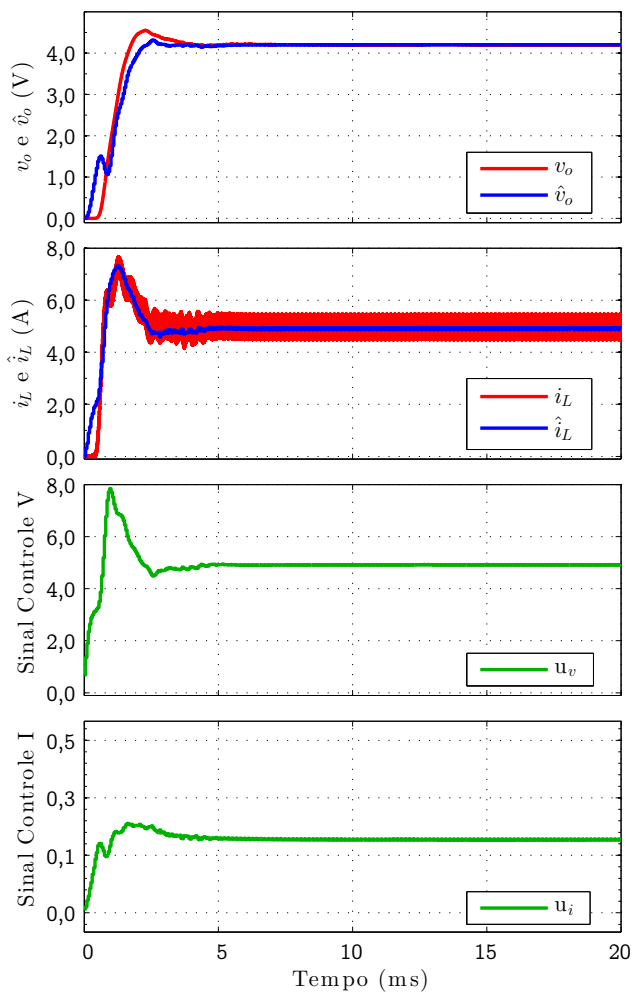
da corrente no indutor, fato que deve realmente acontecer, uma vez que este sinal é a referência que deve ser seguida pela malha de corrente. Já a ação de controle u_i , que é efetivamente a razão cíclica do conversor, mostra um comportamento suave, se estabilizando ao redor de 0,16.

Na Figura 22, tem-se a resposta simulada do sistema a um degrau de tensão também de 4,2 V, porém desta vez a referência de corrente está limitada em 3 A. Novamente, os sinais estimados apresentam comportamento próximo dos sinais reais, salvo a atenuação das componentes de alta frequência, demonstrando-se a eficácia do KF. Desta vez, a resposta do sistema se mostrou mais oscilatória. Uma vez que a limitação de referência pode ser encarada como uma saturação, pode-se justificar tal comportamento oscilatório devido a adição desta não linearidade ao sistema. O tempo de estabilização da corrente ficou em torno de 6 ms, tendo um valor alto de sobressinal, na ordem de 60%. Porém, mesmo sendo mais oscilatório, o sistema foi capaz de seguir o valor de referência de corrente o qual foi limitado, o que atesta sua operação durante a fase de corrente constante de carga das baterias. Com relação aos sinais de controle, enquanto u_v manteve-se saturado durante quase todo tempo, o sinal u_i também se mostrou mais oscilatório, corroborando com o comportamento de saída do sistema.

Por fim, o controle do conversor é validado por simulação frente a degraus de carga, uma vez que a própria característica das baterias é a de uma carga variável. Foram realizadas duas simulações, sendo que em uma delas a carga foi reduzida em 75% e em outra a carga foi aumentada em 300%. Em ambas as situações, é desejado que o controle de tensão mantenha a tensão do conversor em 4,2 V, simulando a entrada de baterias em paralelo.

Na Figura 23, foi simulada a retirada de 75% da carga no instante $t = 10$ ms, ou seja, após o sistema estabilizar no seu valor de tensão de referência por um tempo considerável. Notou-se que houve um ligeiro aumento da tensão de saída durante o transitório, mas a tensão retornou ao seu valor de referência após cerca de 7 ms. Já a corrente de carga sofreu redução, acompanhando a carga retirada do sistema. As variações dos sinais de controle também podem ser

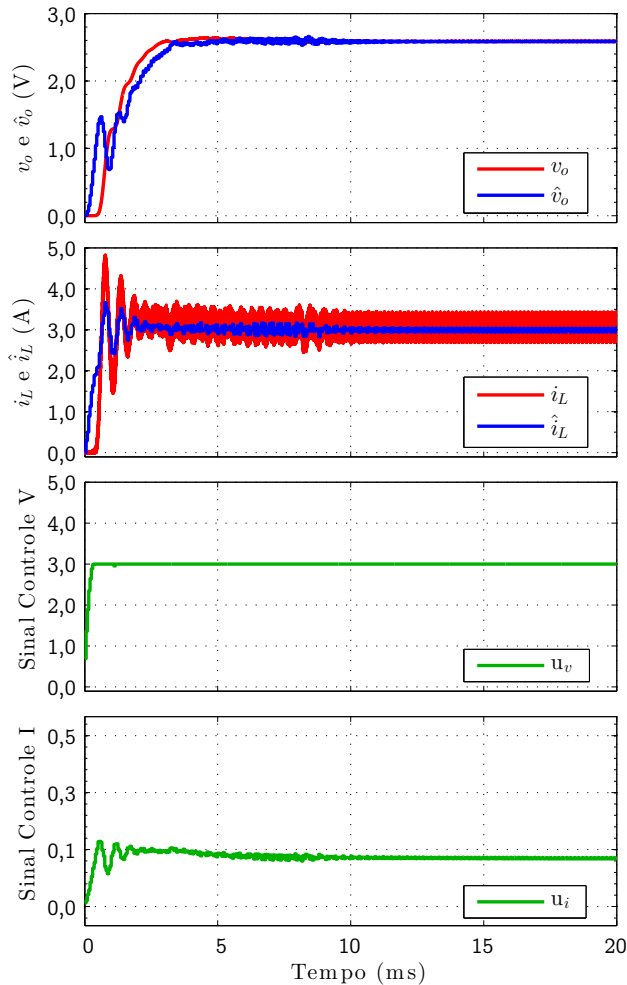
Figura 21 – Resposta simulada do sistema a um degrau de referência de tensão de 0 a 4,2 V sem limitação de corrente (controle de tensão com carga de 0,8 Ω).



Fonte: produção do próprio autor, 2016

observados nesta mesma figura.

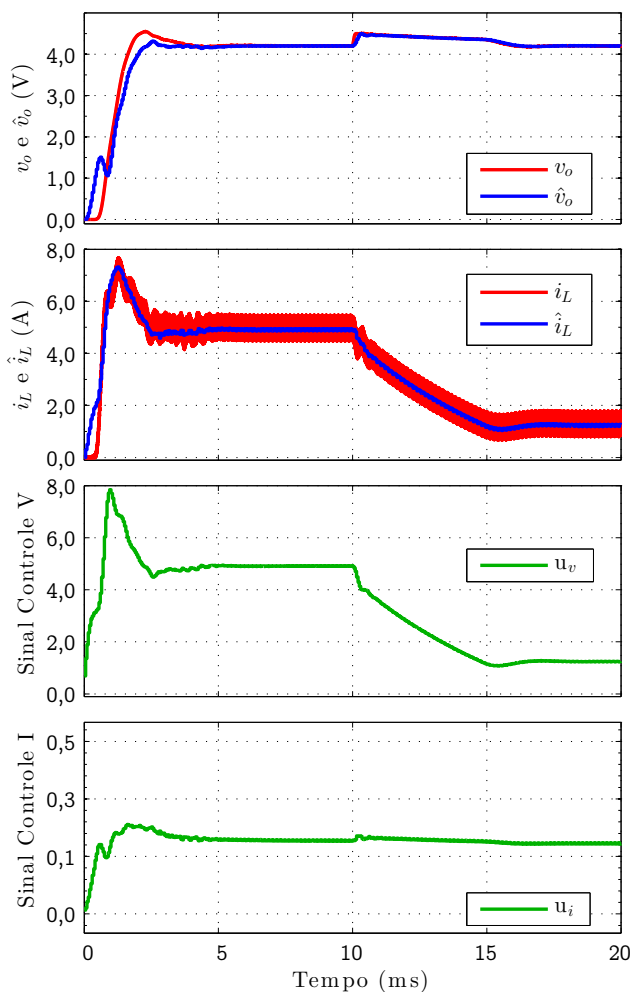
Figura 22 – Resposta simulada do sistema a um degrau de referência de tensão de 0 a 4,2 V com limitação de corrente em 3 A (controle de corrente com carga de 0,8 Ω).



Fonte: produção do próprio autor, 2016

Por fim, tem-se na Figura 24 os resultados simulados de um

Figura 23 – Resposta simulada do sistema a um degrau de carga -75% (de $0,8\ \Omega$ para $3,2\ \Omega$)



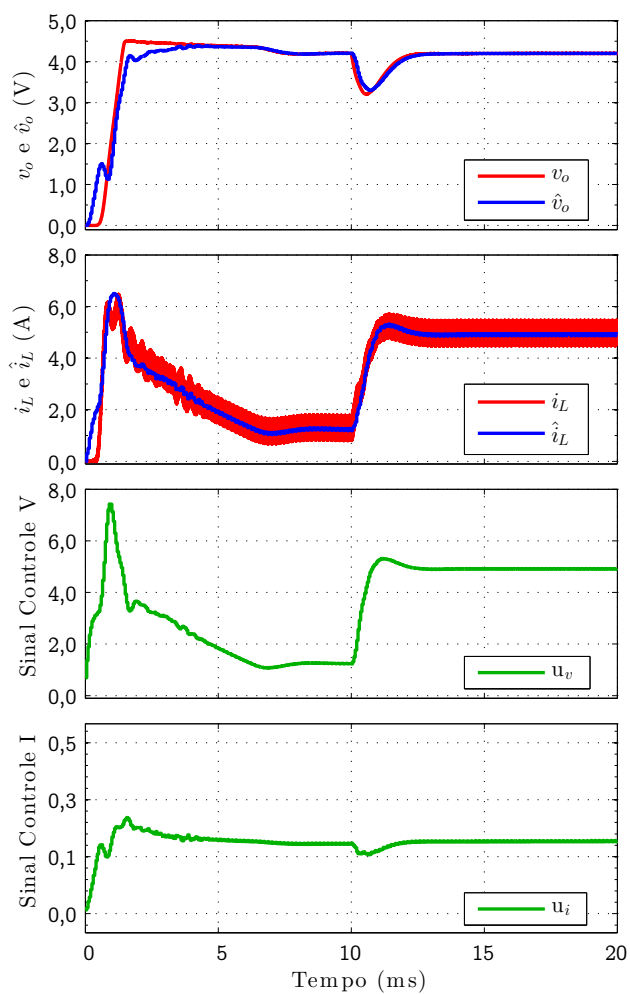
Fonte: produção do próprio autor, 2016

aumento de carga de 300%. De forma qualitativa, estes resultados são similares aos observados na Figura 23. Nota-se que na entrada de carga,

a tensão de saída sofre um afundamento ao invés de uma elevação, e que tal afundamento logo é corrigido pela estratégia de controle, levando ao aumento da corrente de carga. O degrau de carga também foi executado em 10 ms , e o tempo de acomodação do controle foi de cerca de 3 ms , validando a operação do sistema frente a perturbações de carga.

Estando a estratégia de controle validada por simulação, passa-se aos resultados experimentais.

Figura 24 – Resposta simulada do sistema a um degrau de carga +300% (de $3,2 \Omega$ para $0,8 \Omega$)



Fonte: produção do próprio autor, 2016

3.7.3 Resultados experimentais do controlador

Os resultados experimentais são compostos de formas de onda obtidas frente a degraus de referência (carga resistiva de $0,8 \Omega$) e formas de onda da operação de carga das baterias, o que já valida a sua operação frente a perturbações de carga. Todos os resultados aqui mostrados também encontram-se publicados em [39].

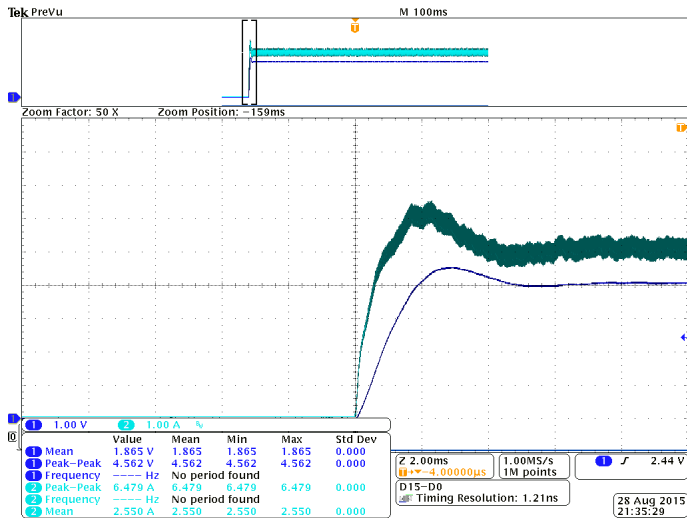
A Figura 25 mostra as formas de onda de tensão e corrente do conversor após dado um degrau na referência de tensão de $4,2 V$, mantendo-se o limite da referência de corrente livre. Nota-se que a tensão de saída segue o sinal de referência, apresentando um tempo de acomodação de aproximadamente $6 ms$ e um sobressinal de $8,6\%$. Apesar do comportamento das formas de onda serem semelhantes, foram observados valores de tempo de acomodação e sobressinal um pouco maiores que os vistos em simulação, que se devem provavelmente a não linearidades e outras características não modeladas.

Na Figura 26 tem-se as formas de onda de tensão e corrente do conversor durante um degrau na referência de tensão de $4,2 V$, porém com limitação na referência de corrente em $3 A$. A saída de corrente se estabiliza no seu valor de referência ($3 A$) em aproximadamente $3 ms$, apresentando um sobressinal de 26% , resultados melhores que os obtidos por simulação, fato que provavelmente está atrelado às não idealidades, que em geral, acrescentam maiores amortecimentos ao sistema, os quais não foram modelados.

Por fim, apresentam-se os resultados obtidos durante a carga de diferentes arranjos de baterias. Todos os dados do processo de carga foram obtidos através do analisador de energia Tektronix PA-4000, o qual possui uma frequência de amostragem de $1 Hz$. Em todos os casos, foram coletadas as formas de onda da tensão de saída do conversor, tensão terminal da bateria ou das baterias em série e corrente de saída do conversor, a mesma que alimenta todo o arranjo de baterias.

Na Figura 27, tem-se a carga de uma única célula, a qual estava completamente descarregada. O protocolo de carga CC-CV é empregado com sucesso pelo conversor. No início, enquanto a tensão da bateria está abaixo do seu valor máximo de $4,2 V$, o conversor é controlado em corrente através da saturação de sua referência no valor em

Figura 25 – Resposta experimental do sistema a um degrau de referência de tensão de 0 a 4,2 V sem limitação da corrente e carga resistiva de 0,8 Ω | CH1 (tensão, azul) = 1V/div | CH2 (corrente, verde) = 1A/div | tempo = 2ms/div

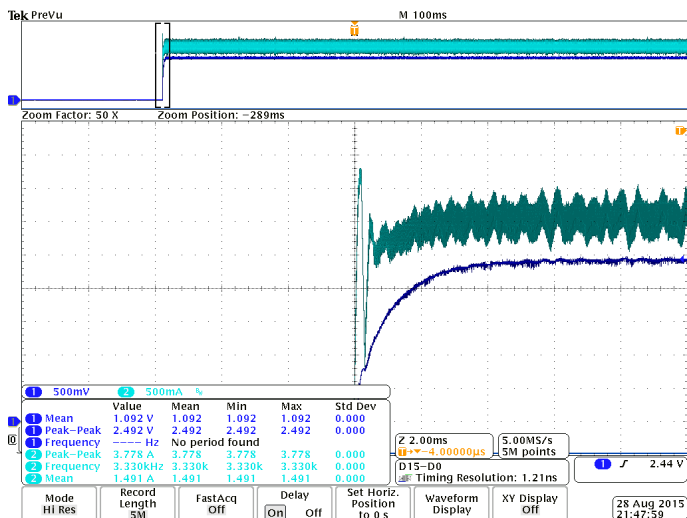


Fonte: produção do próprio autor, 2016

825 mA. A corrente de saída se mantém controlada em seu valor constante durante cerca de 3 h, enquanto a tensão do conversor v_o sobe lentamente acompanhando a tensão terminal da célula v_b . Quando a tensão da bateria atinge os 4,2 V, o controlador de tensão começa a atuar, reduzindo a corrente de saída e a tensão do conversor de forma a manter a tensão da bateria constante. Este processo continua até que a bateria atinja um valor muito baixo de corrente de carga, quando considera-se que a mesma está completamente carregada.

Na Figura 28 é mostrada a carga de um arranjo com duas baterias em série. A corrente de carga é mantida no mesmo patamar, enquanto a tensão de referência é atualizada para $2.v_b = 8,4$ V, o dobro da anterior. O comportamento da carga das duas baterias é muito semelhante, mudando-se apenas o fato de que ambas as tensões de saída do conversor e das baterias em série assumem valores maiores.

Figura 26 – Resposta do sistema a um degrau de referência de tensão de 0 a 4,2 V com limitação da corrente em 3 A e carga resistiva de $0,8 \Omega$ | CH1 (tensão, azul) = 500mV/div | CH2 (corrente, verde) = 500mA/div | tempo = 2ms/div



Fonte: produção do próprio autor, 2016

A Figura 29 apresenta agora uma associação de 6 baterias, sendo que 2 arranjos de 3 baterias em paralelo são conectados em série. Considerando que cada ramo de baterias em paralelo deva drenar a mesma corrente, foi definido um valor de referência de corrente limite aproximadamente três vezes maior que os anteriores, sendo este fixado em 3 A. Observou-se que neste caso, o perfil das tensões v_o e $2.v_b$ manteve-se muito próximo do observado na Figura 28, uma vez que a quantidade de células em série não mudou. Já a corrente fica próxima dos 3 A definidos, mas também com comportamento similar aos descritos anteriormente.

Em todos os resultados da carga de baterias, nota-se que há uma diferença entre a tensão do conversor (v_o) e a tensão do arranjo de baterias (v_b ou $2.v_b$), que é resultado das resistências ôhmicas de cabos e contatos e também da resistência do fusível de proteção, que encontra-

se em série com o arranjo. Além disso, nota-se também que há uma diferença entre o tempo de carga de diferentes arranjos, mesmo tendo-se uma corrente de $0,3C$ para cada bateria. Isto se deve a quantidade de carga inicial do arranjo de baterias, que difere em cada experimento. Isto não deve ser confundido com a quantidade de carga de cada bateria do arranjo.

É importante ainda frisar que em nenhum momento foram analisados desbalanceamentos de carga entre as baterias ou mesmo entre as correntes de carga, no caso do arranjo com baterias em paralelo. Na seção seguinte, são apresentadas as considerações sobre o projeto de controle e seus resultados.

Figura 27 – Carga de uma única bateria: tensão do conversor (v_o vermelho), tensão da bateria (v_b azul) e corrente de carga (i_b verde). Condição inicial de carga $SOC_o \approx 0$

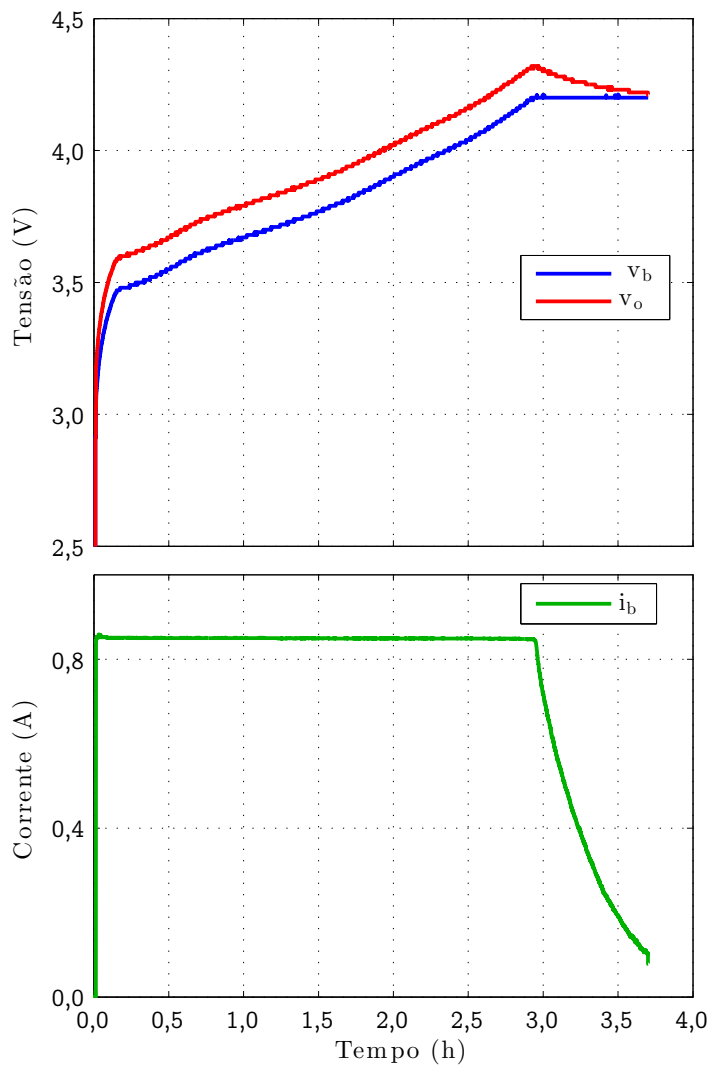
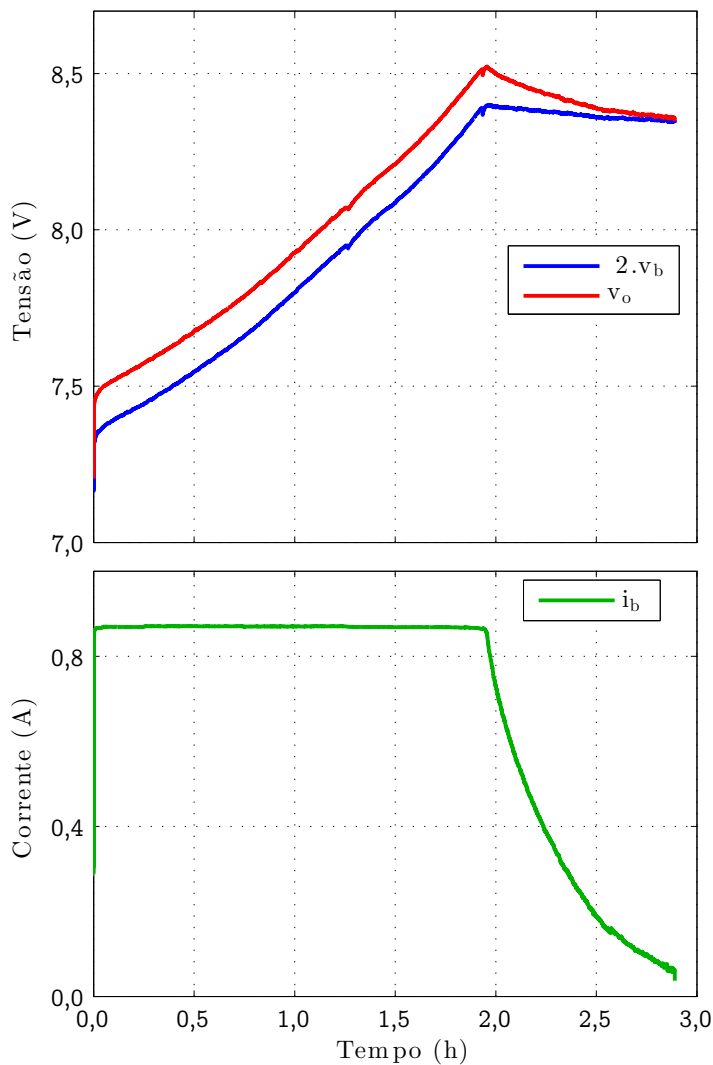
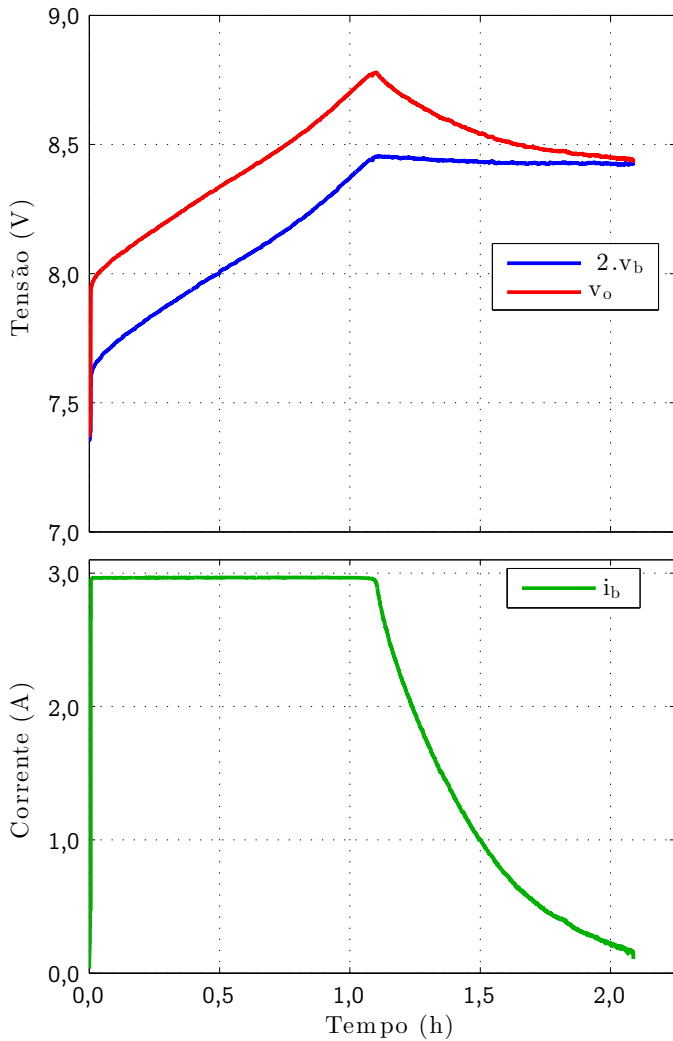


Figura 28 – Carga de duas baterias em série: tensão do conversor (v_o vermelho), tensão das baterias em série (v_b azul) e corrente de carga (i_b verde). Condição inicial de carga $SOC_o \approx 0,2$ para ambas as baterias



Fonte: produção do próprio autor, 2016

Figura 29 – Carga de um arranjo com 6 baterias (3 em paralelo em série com mais 3 em paralelo): tensão do conversor (v_o vermelho), tensão das baterias em série (v_b azul) e corrente de carga (i_b verde). Condição inicial de carga $SOC_o \approx 0,35$ para todas as baterias



Fonte: produção do próprio autor, 2016

3.8 CONSIDERAÇÕES SOBRE O CONTROLE DO CONVERSOR

Este capítulo, além de abordar sobre os circuitos de condicionamento de sinais e sobre as proteções utilizadas para as baterias, abordou sobre o projeto do controlador LQG e seus resultados. A metodologia de projeto do LQG com ajuste das matrizes de ponderação através do PSO foi realizada visando obter um controlador com bons resultados dinâmicos, porém não tendo por objetivo comprovar ou analisar o método em si.

A utilização do LQG com as matrizes de peso sintonizadas pelo algoritmo PSO se mostrou válida neste projeto, resultando em um sistema em malha fechada estável e capaz de realizar a carga de diferentes arranjos de baterias, além de se apresentar como uma metodologia sistêmica para a definição dos ganhos tanto dos controladores quanto do estimador, algo que muitas vezes não se é observado ao se utilizar de metodologias de projeto de controle do tipo LQ ou mesmo em metodologias de controle clássicas.

Os resultados obtidos no projeto deste controlador se mostraram satisfatórios, rastreando sinais de referência mesmo diante de variações, tanto nas referências quanto nas cargas. Tais resultados também se mostraram como uma boa alternativa na carga arranjos de baterias, as quais foram realizadas pelo controlador sem maiores dificuldades.

Uma vez comprovado o funcionamento do carregador de baterias experimentalmente e através de outros resultados simulados, encerra-se aqui o seu desenvolvimento. Os capítulos seguintes tratam da modelagem e simulação de baterias, além das técnicas de estimação do SOC e os seus resultados.

4 MODELAGEM E SIMULAÇÃO DE BATERIAS

Muitas das técnicas de estimação de indicadores de baterias necessitam de um modelo matemático que possibilite um melhor entendimento do problema em questão. Na literatura, existem quatro principais formas de se modelar uma bateria: i) modelagem eletroquímica, ii) modelagem estatística ou matemática, iii) modelagem caixa preta e iv) modelagem por circuitos elétricos [47].

Os modelos eletroquímicos são os mais representativos, porém a sua alta complexidade inviabiliza a sua utilização em larga escala. Estes modelos são compostos por equações diferenciais parciais e não lineares e dependem de inúmeras constantes químicas, como a taxa de difusividade, concentração molar do eletrólito, concentração dos eletrodos, raio das partículas, etc. [68], que garantem explicar as baterias com uma boa riqueza de detalhes necessária a quem estuda os seus aspectos físicos e fundamentais [8, 47, 69]. Alguns destes parâmetros devem ser obtidos através de ensaios experimentais, enquanto que outros necessitam de um vasto conhecimento nas áreas de química e eletroquímica para serem obtidos, o que acaba dificultando a utilização de tal modelo.

Os modelos estatísticos, também referenciados na literatura como modelos matemáticos, usam aproximações estocásticas e equações empíricas para descrever o comportamento das baterias. Apesar de ser um modelo pouco complexo, este pode acarretar em baixa precisão, podendo atingir índices de até 20% de erro [69], além de necessitar de uma boa gama de experimentos para que o modelo estatístico possa ter melhor representatividade. Vale ainda destacar que tal modelo só é válido para a bateria ou arranjo no qual foram extraídos os dados experimentais, não podendo este ser utilizado de forma generalista.

A modelagem por *Black Box* ou Caixa Preta modela por aprendizado o comportamento da bateria através de dados de entrada e saída obtidos empiricamente [69, 70], utilizando para tanto técnicas como ANN (*Artificial Neural Networks*), o que leva também a uma necessidade de grandes volumes de dados afim de obter um modelo com boa representatividade e, novamente, implica em um modelo pouco generalista para aplicação em outras baterias.

Por fim, os *Electric Circuit Models* (ECM - Modelos por Circuitos Elétricos) apresentam uma boa relação entre precisão e comple-

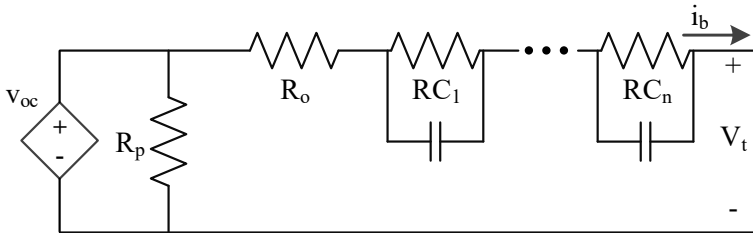
xidade [69–71].

Na modelagem descrita por [72], o modelo de Shepard é considerado. Neste caso, a bateria é modelada apenas por uma fonte de tensão variável em função da carga e uma resistência série. Este modelo então se utiliza de relações exponenciais para caracterizar a tensão de circuito aberto disponível em função da corrente drenada. Porém, o efeito Peukert, que é a perda de capacidade da bateria em função da corrente, não é considerado.

Outros modelos se utilizam de uma maior quantidade de componentes. Conforme descrito em [8, 47, 69, 71, 73, 74], o circuito elétrico da bateria também pode ser representado por uma fonte de tensão de circuito aberto v_{oc} , uma resistência série R_o que caracteriza as perdas pelas resistências dos contatos e dos eletrodos, uma resistência paralela R_p que representa as perdas de *self-discharge* que ocorrem lentamente ao longo do tempo, e uma associação de um ou mais pares RC que garantem ao circuito características dinâmicas.

Por possuir boa representatividade e capacidade de generalização, além de bom custo benefício entre complexidade e precisão e estar mais próximo da área de engenharia elétrica e controle, foi escolhido o ECM para modelagem das baterias. O circuito elétrico genérico aqui utilizado é mostrado na Figura 30.

Figura 30 – Modelo genérico de uma bateria representado por elementos de circuitos elétricos (ECM)



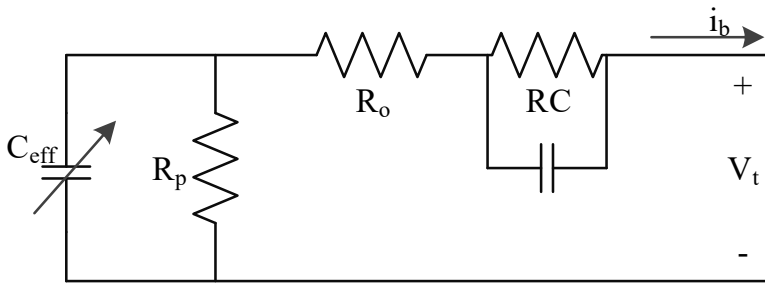
Fonte: produção do próprio autor, 2016

Tais elementos de circuito são geralmente obtidos através de ensaios. A resistência série R_o pode ser obtida analisando-se a queda

de tensão instantânea no momento que uma corrente de carga ou descarga é imposta à bateria. Já os pares RC podem ser descritos a partir da resposta da tensão terminal durante a aplicação de pulsos de corrente, na forma de constantes de tempo. O que dificulta a modelagem de baterias é que todos os seus parâmetros são dependentes da temperatura, do SOC e do SOH da bateria.

Outro fator importante a ser destacado é que a fonte de tensão de circuito aberto v_{oc} é afetada de forma predominante pelo SOC da bateria. Em [70], é proposto um ECM que substitui a fonte de tensão v_{oc} por uma capacitância efetiva C_{eff} , conforme Figura 31.

Figura 31 – Representação de um ECM com capacitância variável no lugar de uma fonte de tensão

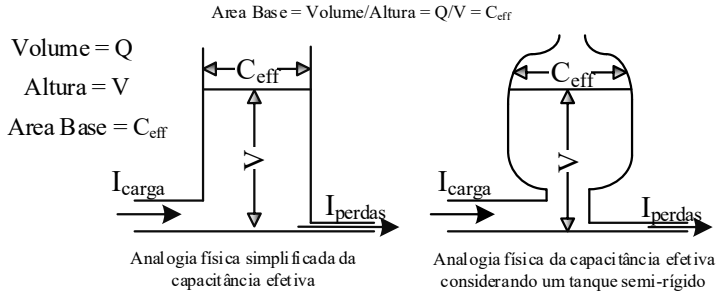


Fonte: Adaptado de C. Blanco et al. [70]

Tal capacitância é modelada como um tanque, que armazena o fluxo de carga inserido. A altura do tanque dita a tensão disponível na bateria. Sendo assim, o volume do tanque (carga armazenada Q) é proporcional a altura do tanque (tensão V) e a área da base do tanque (capacitância efetiva C_{eff}), chegando-se a uma relação entre a característica $v_{oc} \times SOC$. Em [70] também é demonstrado um modelo que inclui não linearidades da característica $v_{oc} \times SOC$ ao considerar o tanque semi-rígido. A Figura 32 mostra a analogia física do modelo de relação entre $v_{oc} \times SOC$ tanto para o caso simplificado quanto para o caso não ideal.

Outros autores como [71], [75], fazem o mapeamento entre a

Figura 32 – Analogia física entre um capacitor comum e um capacitor dependente da própria tensão



Fonte: Adaptado de C. Blanco et al. [70]

tensão de circuito aberto v_{oc} e o SOC através de tabelas de dados (*look-up tables*), obtidas através de experimentações ou dados de fabricantes. Uma outra opção, utilizada por [76], utiliza dados já conhecidos para levantar uma curva que descreve a relação $v_{oc} \times SOC$.

Todos os modelos de baterias mencionados previamente consideram uma única célula. Em [77] é demonstrado um modelo simplificado para várias células conectadas em série e paralelo, enquanto que [78] realiza uma modelagem mais profunda de um banco de baterias de grande escala utilizado em EVs. Neste trabalho, todo o desenvolvimento acerca da modelagem e da estimação de carga leva em consideração apenas uma única célula, por questões de simplicidade.

Na seção 4.1 é apresentada a modelagem da bateria que foi utilizada no projeto do controlador *LQG*. Já na seção 4.2 é detalhada a modelagem da bateria com parâmetros variáveis que é utilizada para a simulação da mesma.

4.1 MODELAGEM DA BATERIA PARA SIMULAÇÃO DO PROJETO DO CONTROLADOR

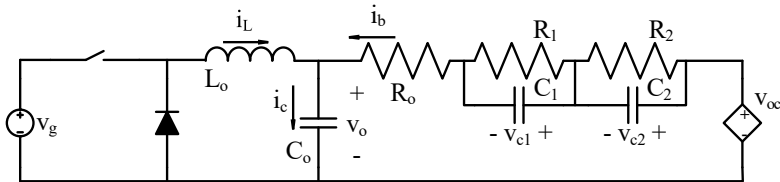
Na seção 3.6.1 foi mostrada a modelagem do conversor Duplo-Forward considerando-se uma carga pura resistiva. Foi a partir deste

modelo com carga resistiva que foram feitas as análises de controlabilidade, observabilidade e estabilidade do sistema, no qual foi assumido o pior caso sob a perspectiva de controle, onde a resistência de carga R_o é maior.

Já na seção 3.6.6 o algoritmo PSO foi utilizado para encontrar uma configuração de matrizes Q_{LQ} e R_{LQ} do LQG que fosse capaz de minimizar a função *fitness* dada por (3.42) durante a etapa de carga de uma bateria, onde ocorrem ambas as fases de corrente constante e tensão constante do perfil de carga CC-CV. Desta forma, nesta etapa de projeto foi simulado o conversor Duplo-Forward conectado a uma bateria e avaliado seu comportamento de carga durante um pequeno intervalo de tempo que pudesse ser representativo em termos de controle, que é durante a transição entre as fases de carga de corrente constante para a fase de tensão constante. Assim, foi necessário a modelagem do circuito do conversor conectado à bateria, que será tratada nesta seção.

A Figura 33 mostra o esquemático considerando do conversor juntamente com a bateria. Foram considerados dois pares RC neste modelo, visando separar as constantes de tempo em duas parcelas fixas, uma de curto prazo (segundos) e outra de longo prazo (minutos). A resistência R_p de *self-discharge* da bateria é desprezada, enquanto a R_o pode ser obtida através de dados de catálogo do fabricante para a corrente de $1C$.

Figura 33 – Representação do conversor Duplo-Forward conectado à uma bateria



Fonte: produção do próprio autor, 2016

A partir da Figura 33, podem ser obtidas as equações mostra-

das em (4.1).

$$\begin{cases} \dot{i}_L(t) = \frac{1}{L_o}(v_g(t) - v_o(t)) \\ \dot{v}_o(t) = \frac{1}{R_o C_o}(v_{oc}(t) - v_o(t)) + \frac{1}{C_o}i_L(t) - \frac{1}{R_o C_o}v_{c1}(t) - \frac{1}{R_o C_o}v_{c2}(t) \\ \dot{v}_{c1}(t) = \frac{1}{R_o C_1}(v_{oc}(t) - v_o(t)) + \left(\frac{-1}{R_o C_1} + \frac{-1}{R_1 C_1}\right)v_{c1}(t) - \frac{1}{R_o C_1}v_{c2}(t) \\ \dot{v}_{c2}(t) = \frac{1}{R_o C_2}(v_{oc}(t) - v_o(t)) - \frac{1}{R_o C_2}v_{c1}(t) + \left(\frac{-1}{R_o C_2} + \frac{-1}{R_2 C_2}\right)v_{c2}(t) \end{cases} \quad (4.1)$$

Manipulando-se as equações (4.1), tem-se a seguinte equação matricial, mostrada em (4.2).

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \\ \dot{v}_{c1}(t) \\ \dot{v}_{c2}(t) \end{bmatrix} = \begin{bmatrix} 0 & -1/L_o & 0 & 0 \\ 1/C_o & -1/R_o C_o & -1/R_o C_o & -1/R_o C_o \\ 0 & -1/R_o C_1 & \left(\frac{-1}{R_o C_1} + \frac{-1}{R_1 C_1}\right) & -1/R_o C_1 \\ 0 & -1/R_o C_2 & -1/R_o C_2 & \left(\frac{-1}{R_o C_2} + \frac{-1}{R_2 C_2}\right) \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_o(t) \\ v_{c1}(t) \\ v_{c2}(t) \end{bmatrix} + \begin{bmatrix} v_g(t)/L_o \\ v_{oc}/R_o C_o \\ v_{oc}/R_o C_1 \\ v_{oc}/R_o C_2 \end{bmatrix} \quad (4.2)$$

Agora, considerando a tensão $v_g(t)$ igual a V_{in}/N_t durante o tempo de condução do interruptor e zero durante o tempo de bloqueio do mesmo, tem-se as equações matriciais descritas por (4.3) e (4.4), respectivamente.

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \\ \dot{v}_{c1}(t) \\ \dot{v}_{c2}(t) \end{bmatrix} = \begin{bmatrix} 0 & -1/L_o & 0 & 0 \\ 1/C_o & -1/R_o C_o & -1/R_o C_o & -1/R_o C_o \\ 0 & -1/R_o C_1 & \left(\frac{-1}{R_o C_1} + \frac{-1}{R_1 C_1}\right) & -1/R_o C_1 \\ 0 & -1/R_o C_2 & -1/R_o C_2 & \left(\frac{-1}{R_o C_2} + \frac{-1}{R_2 C_2}\right) \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_o(t) \\ v_{c1}(t) \\ v_{c2}(t) \end{bmatrix} + \begin{bmatrix} V_{in}/N_t/L_o \\ v_{oc}/R_o C_o \\ v_{oc}/R_o C_1 \\ v_{oc}/R_o C_2 \end{bmatrix} \quad (4.3)$$

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \\ \dot{v}_{c1}(t) \\ \dot{v}_{c2}(t) \end{bmatrix} = \begin{bmatrix} 0 & -1/L_o & 0 & 0 \\ 1/C_o & -1/R_o C_o & -1/R_o C_o & -1/R_o C_o \\ 0 & -1/R_o C_1 & \left(\frac{-1}{R_o C_1} + \frac{-1}{R_1 C_1}\right) & -1/R_o C_1 \\ 0 & -1/R_o C_2 & -1/R_o C_2 & \left(\frac{-1}{R_o C_2} + \frac{-1}{R_2 C_2}\right) \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_o(t) \\ v_{c1}(t) \\ v_{c2}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ v_{oc}/R_o C_o \\ v_{oc}/R_o C_1 \\ v_{oc}/R_o C_2 \end{bmatrix} \quad (4.4)$$

Por fim, utilizando-se do conceito de média instantânea apresentado em 3.6.1 nas equações (4.3) e (4.4) e ainda considerando a tensão $v_{oc}(t)$ como uma entrada do sistema, tem-se o modelo de quarta ordem contínuo que representa conjuntamente as dinâmicas da bateria

e do conversor, conforme (4.5).

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_o(t) \\ \dot{v}_{c1}(t) \\ \dot{v}_{c2}(t) \end{bmatrix} = \begin{bmatrix} 0 & -1/L_o & 0 & 0 \\ 1/C_o & -1/R_o C_o & -1/R_o C_o & -1/R_o C_o \\ 0 & -1/R_o C_1 & \left(\frac{1}{R_o C_1} + \frac{1}{R_1 C_1}\right) & -1/R_o C_1 \\ 0 & -1/R_o C_2 & -1/R_o C_2 & \left(\frac{1}{R_o C_2} + \frac{1}{R_2 C_2}\right) \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_o(t) \\ v_{c1}(t) \\ v_{c2}(t) \end{bmatrix} + \begin{bmatrix} V_{in}/N_i L_o & 0 \\ 0 & 1/R_o C_o \\ 0 & 1/R_o C_1 \\ 0 & 1/R_o C_2 \end{bmatrix} \begin{bmatrix} d(t) \\ v_{oc}(t) \end{bmatrix} \quad (4.5)$$

Este modelo apresentado em (4.5) foi discretizado via ZOH e utilizado na sua forma discreta nas simulações do projeto do controlador LQG na seção 3.6.6.

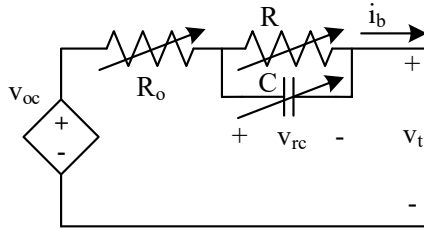
4.2 MODELAGEM POR CIRCUITOS ELÉTRICOS COM PARÂMETROS VARIÁVEIS

Conforme já comentado, o modelo descrito pela Figura 30 apresenta parâmetros que podem variar em função de diversos fatores, como corrente de carga, SOC da bateria e temperatura, porém todas estas variáveis podem ser vistas como funções do tempo. Desta forma, o circuito da Figura 34 pode também ser enxergado como um circuito com parâmetros variáveis no tempo. No caso da modelagem da bateria, foi ainda considerada variação da sua capacidade em função da sua corrente de carga (Efeito Peukert) [72]. Este modelo descrito será utilizado neste trabalho nas etapas de simulação das baterias, porém os efeitos da temperatura e do envelhecimento sobre a bateria serão desconsiderados. A configuração do circuito é mostrada na Figura 34.

Uma vez que o modelo é composto de parâmetros que se alteram no tempo, foi definida a utilização de apenas um par RC , que por ser variante no tempo, possui boa capacidade de representar a dinâmica da bateria.

Para obter a tensão de circuito aberto v_{oc} foi utilizada uma curva $v_{oc} \times SOC$ descrita por [79] para baterias de Lítio NCA, que são do mesmo tipo das baterias NCR 18650 Panasonic, possuindo assim características similares. A partir de tal curva, foi obtida uma aproximação polinomial de 7ª ordem, descrita pela equação 4.6, de forma a representar a tensão v_{oc} em função de SOC . Na Figura 35 é mostrada a curva polinomial obtida através dos pontos dados por [79], tendo in-

Figura 34 – ECM de uma bateria considerando parâmetros variáveis no tempo



Fonte: produção do próprio autor, 2016

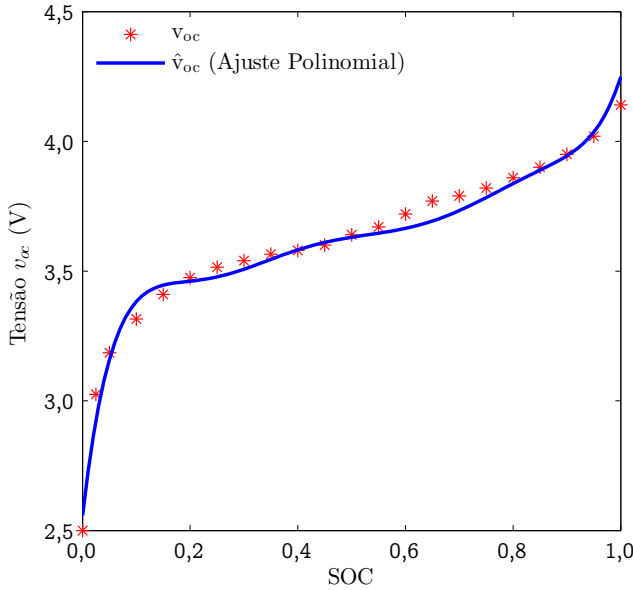
dice $R^2 = 0,9946$.

$$\left\{ \begin{array}{l} v_{oc}(SOC) = 2,558 + 17,89SOC - 144,5SOC^2 + \dots \\ \dots + 601,4SOC^3 - 1369SOC^4 + 1728SOC^5 + \dots \\ \dots - 1134SOC^6 + 301,9SOC^7 \end{array} \right. \quad (4.6)$$

Como o modelo é dependente do valor de SOC , este é obtido através de uma simples integração da corrente da bateria, que neste caso será descrita no tempo discreto k como uma soma recursiva, conforme equação (4.7). Neste caso de modelagem e simulação, considera-se que a corrente $i_b(k)$ utilizada para o modelo é a corrente real injetada ou drenada pela bateria, e não a corrente medida nos seus terminais (lembrando que, mesmo em um ambiente de simulação, pode-se realizar a inserção de ruídos pseudo-aleatórios em uma variável, diferenciando-se assim uma variável determinística de uma variável aleatória afetada por ruídos de medição). Logo, esta corrente não apresenta desvios ou ruídos inerentes de sistemas de medições, ou seja, o valor de $SOC(k)$ não apresenta a acumulação dos desvios em decorrência dos ruídos de medição da corrente $i_b(k)$.

$$SOC(k) = SOC(k-1) - \frac{i_b(k)T_a}{Q_{rate}(i_b(k))} \quad (4.7)$$

Figura 35 – Curva característica genérica de uma bateria Li-ion do tipo NCA e seu ajuste polinomial de 7ª ordem



Fonte: produção do próprio autor, 2016

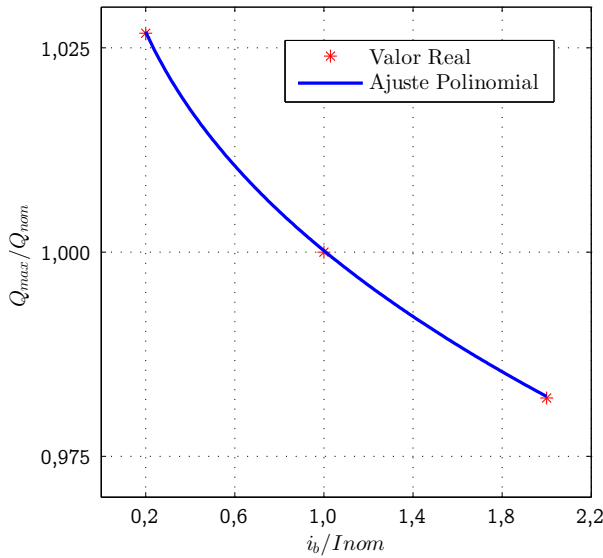
A capacidade da bateria Q_{rate} pode ser descrita pela Equação 4.8, que é uma aproximação polinomial de segunda ordem obtida através das curva de capacidade para diferentes níveis de corrente apresentada no catálogo do fabricante [56]

$$\frac{Q_{rate}(k)}{Q_{nom}} = -\left(\frac{160,5}{2800}\right)\left(\frac{i(k)}{I_{nom}}\right)^{0,3912} + \frac{2961}{2800} \quad (4.8)$$

Desta forma, o $SOC(k)$ depende do seu valor anterior $SOC(k-1)$, da corrente da bateria $i_b(k)$ e período de amostragem T_a . Nota-se ainda através da equação (4.7) que o valor do SOC é influenciado pela capacidade máxima instantânea da bateria Q_{rate} , que por sua vez varia com a magnitude da corrente $i_b(k)$ [80]. Como os efeitos do envelhecimento sobre as baterias foram desconsiderados,

a capacidade máxima da bateria é escrita somente em função da magnitude da corrente instantânea, não tendo dependência com o SOH da mesma, que é considerado sempre unitário. A Figura 36 mostra os dados de catálogo e a curva polinomial aproximada obtida, com um índice $R^2 = 1$, onde $i_b(k)/I_{nom}$ é a relação entre a corrente da bateria e sua corrente nominal de catálogo.

Figura 36 – Curva da capacidade máxima normalizada da bateria em função da sua corrente normalizada



Fonte: produção do próprio autor, 2016

É então realizada a modelagem do circuito, considerando-o inicialmente linear e invariante no tempo, já que que os parâmetros do sistema variam mais lentamente que suas variáveis de entrada e saída. Neste trabalho, serão abordados dois modelos da bateria, sendo que um deles considera a entrada do sistema como sendo a tensão terminal da bateria v_t e outro a sua corrente i_b , porém apenas este último será utilizado na simulação de baterias. A seguir, serão comentadas as obtenções de ambos os modelos da bateria.

4.2.1 Modelo de corrente da bateria

Inicialmente, será considerado o modelo que representa a bateria em sua fase de corrente constante, tendo-se então a corrente de carga como entrada do modelo e a tensão $v_{oc} - v_t$ como saída, o qual será chamado de modelo de corrente da bateria. Através da Figura 34, podem ser equacionadas as relações dinâmicas do circuito no tempo contínuo, descritas na equação (4.9).

$$\begin{cases} v_{oc}(t) - v_t(t) = R_o i_b(t) + v_{rc}(t) \\ v_{rc}(t) = R(i_b(t) - C\dot{v}_{rc}(t)) \end{cases} \quad (4.9)$$

Aplicando-se a Transformada de Laplace em (4.9), tem-se as seguintes equações mostradas em (4.10).

$$\begin{cases} v_{oc}(s) - v_t(s) = R_o i_b(s) + v_{rc}(s) \\ v_{rc}(s) = R(i_b(s) - sCv_{rc}(s)) \end{cases} \quad (4.10)$$

Isolando-se o termo $v_{rc}(s)$ e dividindo-o por $i_b(s)$ na segunda parcela da equação (4.10), chega-se à função de transferência descrita em (4.11).

$$\frac{v_{rc}(s)}{i_b(s)} = \frac{R}{1 + sRC} \quad (4.11)$$

Aplicando-se a transformada ZOH no sistema, tem-se então a função de transferência no domínio discreto (4.12), sendo $z_d = e^{-T_s/RC}$.

$$\frac{v_{rc}(z^{-1})}{i_b(z^{-1})} = \frac{R(1 - z_d)z^{-1}}{1 - z_d z^{-1}} \quad (4.12)$$

Uma vez obtida a função de transferência no domínio discreto em z , a mesma pode ser reescrita no domínio de tempo discreto k , conforme (4.13), representando assim a parcela dinâmica da bateria no tempo discreto.

$$v_{rc}(k) = z_d v_{rc}(k-1) + R(1 - z_d) i_b(k-1) \quad (4.13)$$

Por fim, através da discretização da primeira relação dada pela

equação (4.9), pode-se obter a seguinte relação para a saída, conforme (4.14).

$$v_{oc}(k) - v_t(k) = R_o i_b(k) + v_{rc}(k) \quad (4.14)$$

Em suma, o modelo de corrente da bateria pode ser descrito por (4.15) no espaço de estados, a partir das relações encontradas anteriormente, onde $v_{rc}(k)$ é o estado do sistema, $i_b(k)$ é a entrada e $v_{oc}(k) - v_t(k)$ é a saída, formando-se assim o modelo de corrente da bateria.

$$\begin{cases} v_{rc}(k) = z_d v_{rc}(k-1) + R(1 - z_d) i_b(k-1) \\ v_{oc}(k) - v_t(k) = R_o i_b(k) + v_{rc}(k) \end{cases} \quad (4.15)$$

Uma vez obtido o modelo de corrente, passa-se então a obtenção do modelo de tensão da bateria.

4.2.2 Modelo de tensão da bateria

Considerando a fase de carga da bateria sob tensão constante, tem-se então a tensão $v_{oc} - v_t$ como entrada e a corrente de carga i_b como saída, o qual será chamado de modelo de tensão da bateria.

Reescrevendo-se a equação (4.14) obtida na seção 4.2.1, tem-se a corrente i_b como saída do sistema em (4.16).

$$i_b(k) = \frac{1}{R_o} (v_{oc}(k) - v_t(k) - v_{rc}(k)) \quad (4.16)$$

Agora, substituindo a equação (4.16) na relação (4.13), obtém-se a nova relação dada por (4.17).

$$v_{rc}(k) = z_d v_{rc}(k-1) + (1 - z_d) \frac{R}{R_o} (v_{oc}(k-1) - v_t(k-1) - v_{rc}(k-1)) \quad (4.17)$$

Tendo-se a equação (4.16) e reescrevendo-se a equação (4.17), chega-se no modelo de tensão da bateria escrito em espaço de estados, descrito em (4.18), onde $v_{rc}(k)$ é o estado, $v_{oc}(k) - v_t(k)$ é a entrada

e $i_b(k)$ é a saída do sistema.

$$\begin{cases} v_{rc}(k) = \left(z_d + \frac{R}{R_o}(1 - z_d) \right) v_{rc}(k-1) + \frac{R}{R_o}(1 - z_d) (v_{oc}(k-1) - v_t(k-1)) \\ i_b(k) = \frac{1}{R_o} (v_{oc}(k) - v_t(k) - v_{rc}(k)) \end{cases} \quad (4.18)$$

Na sequência, serão mostradas as simulações de baterias feitas a partir dos modelos adquiridos.

4.3 SIMULAÇÃO DO MODELO DINÂMICO DA BATERIA

A simulação da bateria será construída a partir do modelo de corrente já obtido em espaço de estados. Inicialmente, a obtenção de tal modelo considerou o modelo linear e invariante no tempo. Agora, para a iteração k de simulação, os parâmetros que compõe o modelo em espaço de estados não serão mais considerados constantes, implicando assim em um sistema variante no tempo. Além disso, a característica não linear do sistema, fortemente influenciada pela relação $v_{oc} \times SOC$, será obtida através da aproximação polinomial conforme abordado na seção 4.2. Para encontrar os parâmetros do sistema a cada iteração k da simulação das baterias, foi utilizado um estimador baseado no algoritmo PSO. Este estimador é executado a cada iteração de simulação da bateria, buscando estimar os parâmetros $R_o(k)$, $R(k)$ e $C(k)$ da mesma, sendo que a execução do PSO numa dada iteração k de simulação da bateria abrange desde a etapa de inicialização do algoritmo até que este atinja a convergência ou o limite máximo de iterações. Na amostra posterior de simulação da bateria ($k + 1$), o algoritmo PSO é novamente executado, desde a sua inicialização até ser atingido o seu critério de parada, buscando estimar os parâmetros $R_o(k+1)$, $R(k+1)$ e $C(k+1)$, que variaram no tempo.

Com o objetivo de reduzir variações abruptas nas estimações dos parâmetros, a cada inicialização do PSO, uma das partículas recebe o valor dos parâmetros encontrados na iteração anterior.

É importante destacar que este simulador realizado não visa simular uma bateria em específico e nem validar sua simulação com dados experimentais. O objetivo do mesmo é servir de ferramenta para

estudos mais generalistas de baterias, que não necessitam de grande exatidão nos parâmetros, mas sim da representação do comportamento genérico de uma bateria, diminuindo-se assim a dependência de dados experimentais no estudo destas.

A operação do algoritmo PSO em si é muito semelhante à descrita na seção 3.6.6, porém, existem algumas particularidades com relação ao PSO na estimação dos parâmetros das baterias que são listadas a seguir.

- As partículas candidatas à solução da função *fitness* são compostas pelo vetor $x(i) = [\hat{R}_o \ \hat{R} \ \hat{C}]$.
- O espaço de busca é adequado ao novo vetor $x(i)$ e possui os valores descritos na Tabela 5. Nota-se que o par RC deve ficar entre um valor mínimo $RC_{min} = 0,1 \text{ s}$ e $RC_{max} = 360 \text{ s}$, conforme especificado pelo espaço de busca dos parâmetros R e C .
- A simulação da bateria será realizada unicamente a partir do modelo de corrente descrito em (4.15), ou seja, considerando $u(k) = i_b(k)$ e $v_{oc}(k) - v_t(k) = y(k)$.
- A função *fitness* foi modificada para que pudesse ser utilizada na estimação dos parâmetros da bateria e é uma função da saída $y(k)$ e também dos seus parâmetros. Esta função é descrita na equação (4.19).
- Parâmetros do algoritmo PSO como número de partículas, limite de iterações e etc. foram alterados, e estão dispostos na Tabela 5.

A cada iteração, o PSO irá calcular uma saída estimada $\hat{y}(k-1)$ em função dos parâmetros estimados $\hat{R}_o(k-1)$, $\hat{R}(k-1)$ e $\hat{C}(k-1)$, os quais compõem o vetor $\hat{\Theta}(k-1)$. Esta saída é calculada conforme as equações (4.15). Tendo-se o valor de $\hat{y}(k-1)$ calculado pelo PSO e o valor de $y(k-1)$ que já havia sido determinado anteriormente na

Tabela 5 – Parâmetros utilizados no algoritmo PSO do simulador de baterias

Parâmetro	Valor
x_{min}	[0, 05 0, 05 2]
x_{max}	[0, 2 0, 2 1800]
partículas	30
parâmetros	3
limite iterações	100
α_1	[0 0, 3]
α_2	[0 0, 3]
α_3	[0 0, 4]

Fonte: produção do próprio autor, 2016

simulação da bateria, pode-se então determinar o erro de estimação do algoritmo, que compõe uma das parcelas da função de custo do PSO. Ainda, para evitar que os valores encontrados variem bruscamente durante a simulação, foi adicionado um segundo termo, que corresponde à norma da variação dos parâmetros estimados, normalizada em função dos parâmetros estimados anteriormente, dividida pelo número de parâmetros. A minimização da norma da variação dos parâmetros atua como um LPF, evitando variações abruptas na estimação. A equação (4.19) mostra a função *fitness* definida.

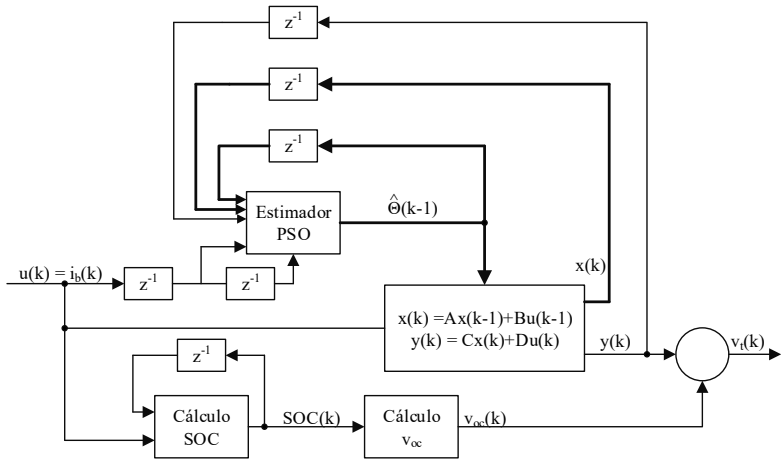
$$J_{PSO.batt} = |y(k-1) - \hat{y}(k-1)| + \frac{1}{3} \sqrt{\sum_{i=1}^3 \frac{\Delta \hat{\Theta}_i(k-1)}{\Theta_i(k-2)}} \quad (4.19)$$

É importante notar ainda que são utilizados os valores de saída atrasados em uma amostra, e conseqüentemente, a estimação dos parâmetros resultantes também é atrasada em uma amostra. Porém, como a variação dos parâmetros é mais lenta que a variação dos estados do sistema, pode-se considerar que os parâmetros na iteração k podem ser aproximados pelos parâmetros na iteração $k - 1$. Esta consideração é necessária para que se mantenha a causalidade do sistema, uma vez a obtenção dos estados na simulação da bateria assume que os parâmetros

já são conhecidos.

Estando definidos os parâmetros do sistema, é utilizada por fim a equação (4.8) para determinação da capacidade da bateria e a equação (4.7) para obtenção do SOC da mesma, seguidas da equação (4.15) para a determinação dos valores de entrada e saída da bateria. A Figura 37 mostra o diagrama de blocos do simulador da bateria utilizando-se o modelo de corrente.

Figura 37 – Diagrama de blocos do simulador de baterias utilizando o modelo de corrente



Fonte: produção do próprio autor, 2016

Mais detalhes sobre o simulador de baterias implementado podem ser obtidos através da análise dos códigos que compõe o mesmo, implementados em Matlab® e dispostos no Anexo C. Na seção 4.4, serão mostrados os resultados do simulador.

4.4 RESULTADOS DE SIMULAÇÃO DO MODELO DINÂMICO DA BATERIA

Os resultados do simulador de baterias e da modelagem utilizada no mesmo são compostos dos parâmetros obtidos ao longo do

tempo e das grandezas principais relacionadas às baterias, que são a tensão de circuito aberto v_{oc} e tensão terminal v_t , a corrente da bateria i_b e o SOC da mesma.

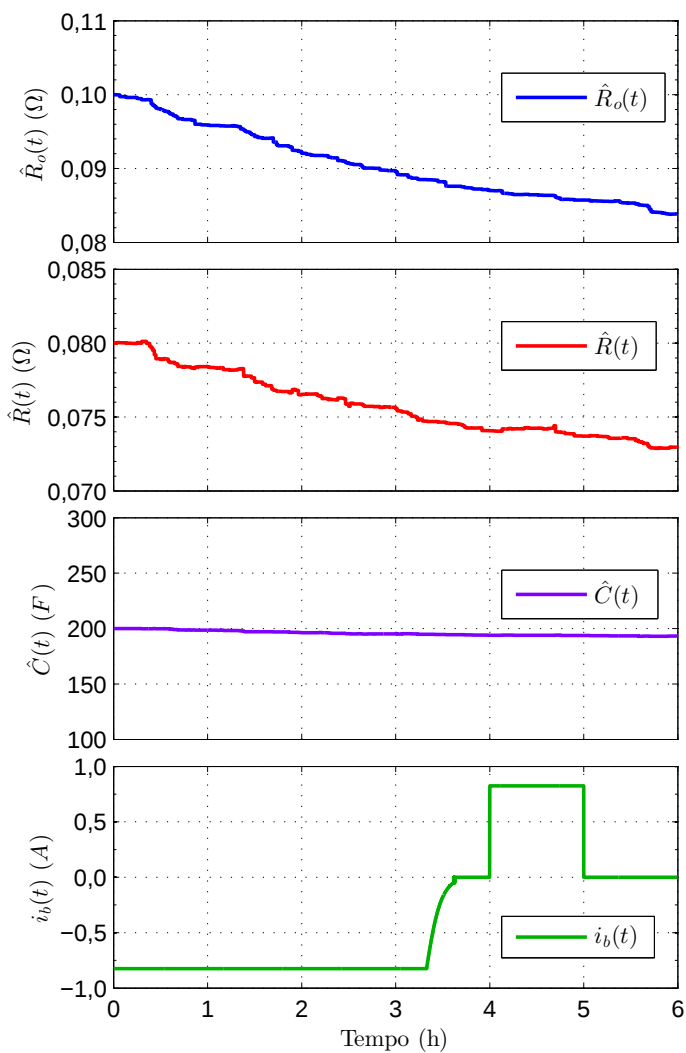
A Figura 38 mostra a variação dos parâmetros estimados $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ ao longo do tempo. Nota-se que tais parâmetros decrescem lentamente durante a fase de carga constante, dada pela corrente i_b negativa, caracterizando a variação dos mesmos frente ao aumento de carga na bateria. A partir do momento em que a corrente é nula, os parâmetros apresentam variações ligeiramente menores, mesmo durante a curta etapa de descarga da bateria ($i_b > 0$), que apresenta magnitude de corrente menor que a etapa de carga da mesma. Desta forma, não foi possível estabelecer uma correlação clara entre as variáveis de corrente e tensão da bateria e os parâmetros estimados.

Já a Figura 39 apresenta a característica da tensão v_{rc} e também o comportamento da constante de tempo $\hat{R}\hat{C}$ estimada. Nota-se que a constante de tempo $\hat{R}\hat{C}$ é maior também durante a etapa de carga sob corrente constante, porém esta continua a decrescer mesmo diante da etapa de descarga. Este comportamento também pode ser visualizado através da v_{rc} , que possui maior constante de tempo durante a etapa de carga e constante de tempo ligeiramente menor durante a etapa de descarga. A diferença entre o valor final de magnitude de v_{rc} obtido entre as etapas de carga e descarga se justifica quase que inteiramente pela diferença na magnitude das correntes destas etapas. Exceto por este último comentário, não foram observadas outras correlações entre as variáveis analisadas.

A Figura 40 traz as formas de onda das tensões v_{oc} e v_t da bateria, bem como a sua quantidade de carga dada pelo SOC, além da corrente i_b .

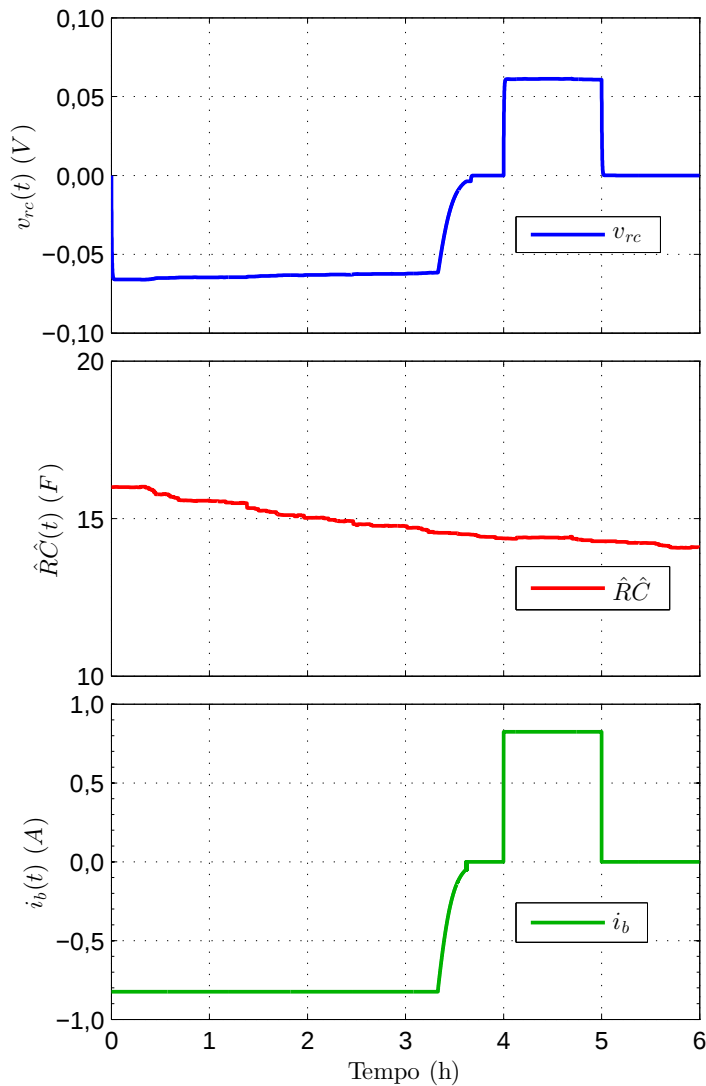
Nota-se que o comportamento da tensão terminal v_t é bem próximo do esperado, estando sempre maior que o valor de v_{oc} durante a carga e menor que v_{oc} durante a descarga. Nota-se ainda que o comportamento dinâmico de v_{rc} está presente na tensão v_t , conforme desejado, representando o efeito de relaxação da bateria, ou seja, sua dinâmica. A curva do SOC também apresenta comportamento esperado, crescendo linearmente durante a etapa de carga sob corrente constante e decrescendo linearmente durante a etapa de des-

Figura 38 – Resultados da estimação dos parâmetros $\hat{R}_o(t)$ (azul), $\hat{R}(t)$ (vermelho), $\hat{C}(t)$ (roxo), juntamente com a corrente $i_b(t)$ (verde) da bateria



Fonte: produção do próprio autor, 2016

Figura 39 – Resultados da constante de tempo $\hat{R}\hat{C}$ estimada (vermelho) e da tensão v_{rc} estimada (azul), juntamente com a corrente da bateria (verde)

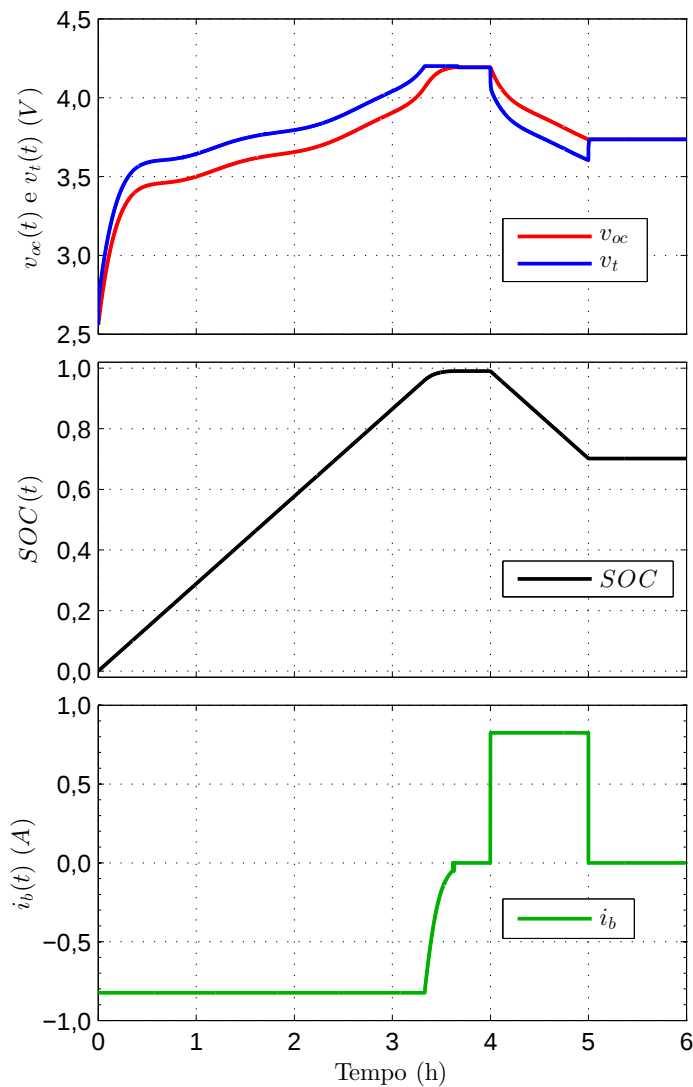


Fonte: produção do próprio autor, 2016

carga também com corrente constante. Durante a fase de carga sob tensão constante, que ocorre em um tempo em torno de 3,5 h , nota-se que a corrente de carga assume um perfil exponencial decrescente em magnitude, enquanto a tensão de carga é mantida, da mesma forma que foi observado nos resultados experimentais do carregador de baterias. Nesta fase, o crescimento do *SOC* também assume um perfil com a mesma característica da corrente de carga, não sendo mais linear.

A seguir, serão comentadas as considerações finais acerca da modelagem de baterias.

Figura 40 – Formas de onda das tensões v_t (azul) e v_{oc} (vermelho) da bateria, juntamente com seu SOC (preto) e sua corrente i_b (verde)



4.5 CONSIDERAÇÕES FINAIS SOBRE A MODELAGEM E SIMULAÇÃO DE BATERIAS

Neste capítulo, foram tratadas as questões referentes à modelagem de baterias e simulação das mesmas. Foram apresentados quatro tipos de modelagem frequentemente mencionadas na literatura, das quais a modelagem por circuitos elétricos ECM se mostrou a mais atrativa nesta dissertação, motivo pela qual esta foi utilizada.

Foi apresentada ainda neste capítulo a modelagem de uma bateria conectada ao conversor, sendo tal modelo utilizado na obtenção das matrizes de peso Q e R do controlador LQG , mostrado no capítulo 3.

Quando comparadas a dinâmica da bateria com a dinâmica do conversor, nota-se que o conversor possui constantes de tempo na ordem de ms enquanto a bateria apresenta constantes de tempo na ordem de segundos ou minutos, ou seja, a dinâmica do conversor é muito mais rápida que a dinâmica da bateria. Por este motivo, aproximações podem ser feitas tanto sob o ponto de vista da bateria quanto sob o ponto de vista do conversor. Com relação ao conversor, pode-se considerar para o projeto de controle que a bateria é uma carga R constante, uma vez que a sua tensão terminal e sua corrente variam muito mais lentamente que as grandezas do conversor. Do ponto de vista da bateria, como o conversor possui dinâmicas muito mais rápidas, este pode ser considerado como uma fonte ideal, tanto nas condições de carga de corrente constante (fonte de corrente) quanto nas condições de carga de tensão constante (fonte de tensão). Ambas as aproximações podem ser feitas tranquilamente sem prejuízos às análises de controle ou de estimação, uma vez que as dinâmicas destes sistemas se diferenciam em uma ordem de 1000 vezes. O modelo completo de quarta ordem, o qual considera a união do conversor Duplo-Forward com a bateria, foi demonstrado neste trabalho visando ilustrar a sua obtenção e também para que pudesse ser utilizado no projeto do controlador LQG um modelo mais detalhado do sistema em questão.

Em seguida, foram apresentadas as modelagens da bateria considerando-se como entrada tensões ou correntes constantes, resultando no modelo de corrente da bateria e no modelo de tensão da

bateria. Dentre estes modelos, o modelo de corrente foi escolhido para simular o comportamento das baterias, sendo que os parâmetros R_o , R e C foram assumidos como variantes no tempo e encontrados de forma iterativa através do algoritmo de otimização PSO. Este método de simulação mostrou um comportamento satisfatório, porém o mesmo não foi validado com dados experimentais de baterias, uma vez que o objetivo neste capítulo é apenas ilustrar os comportamentos mais genéricos de baterias e utilizar tais dados para algumas análises, principalmente do SOC de baterias de Lítio em geral, fornecendo uma maneira rápida de se estudar as baterias com menor dependência de dados experimentais.

Ainda, é importante salientar que não foram feitas análises sobre as variações dos dados simulados em função de diferentes execuções do simulador. Na verdade, a partir de algumas simulações, notou-se que o modelo de bateria obtido varia ligeiramente entre uma execução e outra, fator que provavelmente está atrelado às diferentes condições iniciais do vetor de parâmetros estimados $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ pelo PSO, à natureza estocástica do algoritmo e também podem estar relacionados à escolha de um espaço de busca pouco restritivo ou à má escolha de parâmetros do algoritmo PSO. Assim, o autor destaca que faltam estudos mais detalhados acerca da influência da inicialização aleatória dos parâmetros e como melhorar tal inicialização de parâmetros, bem como estudos que busquem definir um espaço de busca e outros parâmetros do algoritmo PSO mais adequados ao problema de modelagem de baterias. Apesar destas observações, os resultados aqui apresentados são suficientes para o propósito de simulação de baterias para a análise de estimação do SOC.

Feitas as considerações acerca deste capítulo, passa-se então ao capítulo 5 sobre a estimação do SOC de baterias.

5 ESTIMAÇÃO DE CARGA DE BATERIAS

No capítulo 1, foi realizada uma introdução acerca das tecnologias de baterias mais utilizadas atualmente e também sobre os métodos de estimação de carga utilizados, especialmente em baterias de Lítio. Esta introdução foi seguida de dois capítulos que trataram do desenvolvimento de um carregador de baterias, com o objetivo de auxiliar na manipulação do SOC das baterias e consequentemente no estudo de métodos de estimação de carga. No capítulo 4, foram comentados os diversos métodos de se modelar uma bateria, sua modelagem por circuitos elétricos e as simulações de tais modelos obtidos. Agora, utilizando-se os conhecimentos acerca da modelagem de baterias, serão abordados de forma mais aprofundada os conceitos sobre a estimação de carga das mesmas.

Este capítulo tem por objetivo apresentar ao leitor uma visão mais completa sobre a estimação do SOC de baterias e expor de forma sintética e concisa informações importantes espalhadas pela literatura acerca da estimação de carga. Ainda, será mostrado neste capítulo o desenvolvimento de técnicas de estimação do SOC de baterias visando a familiarização com as etapas de implementação de estratégias de estimação e mostrar os resultados obtidos através de tais desenvolvimentos.

Dentre os métodos de estimação implementados, foi utilizada a estimação através do *Extended Kalman Filter* (EKF), uma abordagem já conhecida na literatura. Outros dois métodos são aqui propostos, os quais utilizam um KF adaptativo para a estimação do SOC e que serão avaliados posteriormente neste capítulo.

Nas seções seguintes, algumas das técnicas de estimação do SOC mais recorrentes na literatura serão abordadas.

5.1 ESTIMAÇÃO DE CARGA POR COULOMB COUNTING E PELA TENSÃO TERMINAL

A estimação de carga de baterias é uma tarefa desafiadora para os pesquisadores. Primeiro, por ser um sistema não linear, em decorrência da relação entre a tensão de circuito aberto v_{oc} e o estado de carga SOC , em geral possuindo maiores derivadas nos extremos, quando há

pouca ou muita carga na bateria, e derivadas menores em faixas intermediárias de carga, conforme pode ser observado na Figura 35 da seção 4.2. Ainda, tal relação $v_{oc} \times SOC$ varia de bateria para bateria, o que pode ocorrer por inúmeros motivos, seja por variações paramétricas inerentes dos processos de fabricação ou pelos vários compostos químicos diferentes que podem ser utilizados.

Além das não linearidades presentes na curva $v_{oc} \times SOC$, há ainda o fato de que a tensão terminal v_t medida na bateria pode não ser necessariamente a tensão v_{oc} , em decorrência das resistências internas da bateria e da característica capacitiva da mesma, a qual agrega um efeito dinâmico entre o valor de tensão terminal v_t e o valor da tensão de circuito aberto v_{oc} . Desta forma, a única maneira de se medir a tensão v_{oc} de uma bateria é deixar a mesma em repouso ($i_b = 0 A$) por um longo período (em geral, por horas), até que a característica dinâmica da bateria leve a tensão v_t à convergir para o valor de v_{oc} , o que frequentemente é inviável para diversas aplicações.

Ainda, é importante destacar os efeitos do envelhecimento da bateria, os quais alteram o seu SOH e, por consequência, sua capacidade máxima de carga Q_{max} e também seus parâmetros resistivos e capacitivos, dificultando a estimação do SOC, já que esta é dependente de Q_{max} , e fazendo da bateria um sistema variante no tempo.

Por fim, conforme já comentado no capítulo 4, tanto os parâmetros da bateria, quanto a curva $v_{oc} \times SOC$ e o SOH da bateria são dependentes da temperatura. Destaca-se ainda que os parâmetros da bateria também variam de acordo com o SOC da mesma e também com a magnitude e sentido da corrente i_b da bateria. Em suma, é notório que as baterias e a estimação do SOC são temas de grande complexidade e que ainda carecem de mais estudos por parte da comunidade científica.

Conforme já foi comentado na seção 1.2.1, existem dois métodos de estimação do SOC que são extremamente utilizados devido à sua grande simplicidade, que são o *Coulomb Counting* e a estimação pela tensão terminal. Tais métodos são essencialmente estimadores em malha aberta.

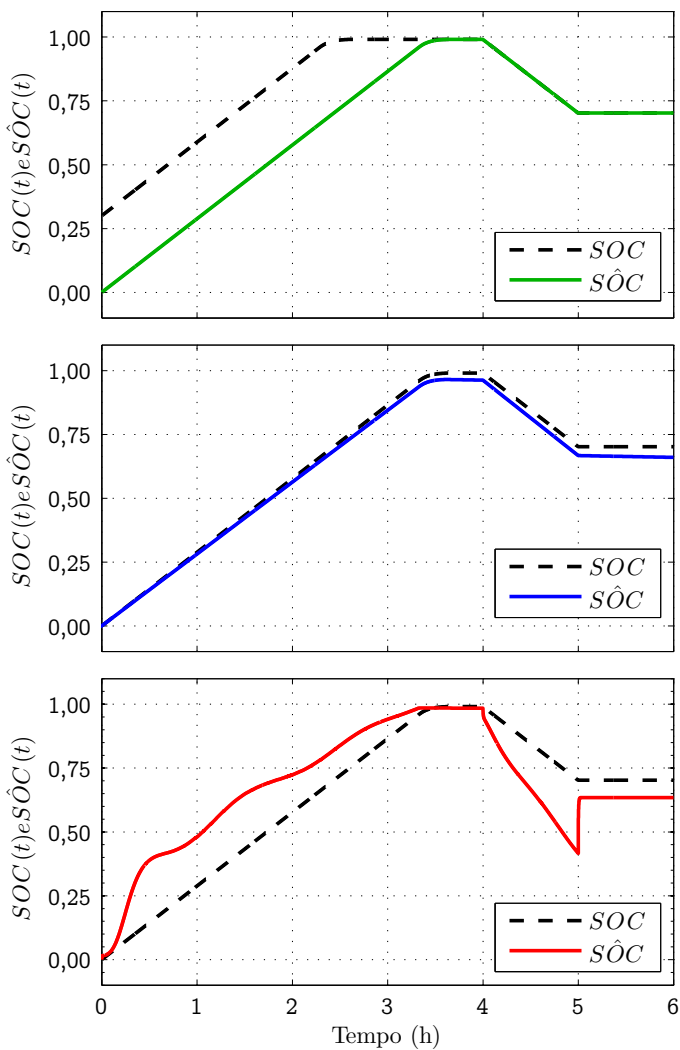
A dificuldade em estimar qual a condição inicial de carga SOC_o e a presença de ruídos de média não nula na medição na cor-

rente i_b da bateria são as principais dificuldades encontradas ao utilizar o *Coulomb Counting*. A Figura 41 demonstra três casos de estimação do SOC de baterias obtidos por simulação, onde em ambos a linha tracejada representa a forma de onda real do SOC. Na forma de onda em verde, tem-se o caso em que o *Coulomb Counting* foi inicializado com condição inicial errônea, porém desprezando-se o ruído de medição da corrente. Nota-se que erro de estimação inicial permanece constante durante todo o processo de estimação. Já na forma de onda em azul, tem-se o caso onde existe um erro de medição de corrente polarizado, porém, aqui foi assumido erro nulo na condição de carga inicial. Neste caso, o processo de integração da corrente acaba integrando o ruído conjuntamente. Como o ruído possui média não nula, tem-se como resultado uma estimação do SOC que diverge lentamente ao longo do tempo. Salienta-se ainda que em geral, tais problemas ocorrem conjuntamente durante o processo de estimação pelo *Coulomb Counting*. Vale destacar ainda que este método em geral é composto de saturações no valor do SOC estimado em 0 e 1, fato este que pode ser utilizado para calibração do algoritmo de tempos em tempos, quando é atingida a região de saturação.

No caso da estimação do SOC pela tensão terminal da bateria, também existem algumas dificuldades. A primeira delas, é que a curva $v_{oc} \times SOC$ deve ser conhecida para que seja realizado o processo de estimação, que em geral é realizado através de *look-up tables* decorrentes de dados experimentais. Ainda, destaca-se que esta curva deve ser atualizada de tempos em tempos, uma vez que sua característica se altera com o envelhecimento da bateria. Outro problema é que, conforme já mencionado, a tensão terminal medida muitas vezes difere da tensão v_{oc} da bateria, especialmente quando a corrente da bateria é não nula, o que dificulta uma boa estimação do SOC por este método quando a bateria não se encontra em repouso. A Figura 41 também ilustra o problema da utilização da tensão terminal v_t para a estimação do SOC (forma de onda em vermelho), considerando-se que a curva $v_{oc} \times SOC$ é bem conhecida.

Diante de tais dificuldades, os pesquisadores tem se debruçado na busca por outros métodos de estimação do SOC de baterias, sendo que muitos destes métodos se utilizam de estratégias de estimação em

Figura 41 – Erros de estimação utilizando-se os métodos de *Coulomb Counting* (verde e azul) e Estimação pela Tensão Terminal (vermelho)



Fonte: produção do próprio autor, 2016

malha fechada, almejando aumentar a confiabilidade da estimação. Na sequência, serão citados alguns dos artigos publicados na área de estimação do SOC destacando-se pontos de vista importantes dos mesmos, que são utilizados neste trabalho.

5.2 CONTEXTUALIZAÇÃO DE MÉTODOS DE ESTIMAÇÃO DO SOC PRESENTES NA LITERATURA

Apesar da determinação do valor de carga da bateria em si já ser bem consolidada na literatura, muitas vezes esta exige longos testes que dificultam sua aplicação prática, uma vez que a estimação de carga de baterias frequentemente deve ser feita *online*. Sejam BEVs, celulares ou notebooks, todos necessitam de uma estimação de carga contínua, inclusive enquanto estes encontram-se em operação, o que leva à situação de estimação de carga sob corrente não nula nas baterias. O desafio de realizar a estimação do SOC continuamente e tendo-se muitas das vezes apenas as medidas de corrente e tensão terminal da bateria ainda instiga pesquisadores, que buscam por métodos melhores e mais eficientes, não havendo ainda um método em malha fechada que seja amplamente adotado pela comunidade científica. Assim, são trazidos aqui alguns artigos e algumas das suas contribuições visando familiarizar o leitor diante de tantas metodologias distintas e condensar algumas das informações disponíveis sobre a estimação do SOC.

A proposta mostrada em [1] é um algoritmo adaptativo baseado na teoria de Lyapunov para realizar a estimação tanto do SOC quanto do SOH utilizando-se do mínimo de informações e dados disponíveis. Nesta proposta, os parâmetros do sistema são estimados por regressão via *Least-Squares* (LS - Mínimos-Quadrados) e a estimação supostamente necessita apenas dos dados de medição de corrente e tensão da bateria. Porém, o autor supõe que a curva $v_{oc} \times SOC$ já é conhecida, seja através do catálogo do fabricante ou de dados experimentais. A técnica desenvolvida estima de fato a tensão v_{oc} e então obtém o valor de SOC utilizando-se da curva $v_{oc} \times SOC$ já conhecida. Segundo o autor, a estimação é robusta frente a variações paramétricas da bateria.

Em [22], a estimação do SOC é feita através de um observador de Luenberger, sendo que o modelo de bateria utilizado para a esti-

mação considera apenas um único par RC , assim como o descrito na Figura 34 da seção 4.2, o qual é discretizado via ZOH. O observador é implementado na forma de um sistema embarcado e apresenta erros de estimação menores que 2% em regime permanente, porém são assumidos conhecidos os parâmetros elétricos do modelo e a influência da temperatura ou do SOH na estimação são desprezados.

É demonstrado em [81] utilização de um Proporcional Integral (PI) para estimação do SOC e a análise do erro do estimador em regime permanente. Neste caso, também foi utilizado um circuito de primeira ordem para representar a bateria e a curva $v_{oc} \times SOC$ foi aproximada por vários segmentos de reta, visando obter um sistema linear, sendo que todos os dados do modelo foram obtidos experimentalmente.

Em [81] foi feita ainda uma análise da influência da temperatura e da variação do SOH e do SOC sobre os parâmetros elétricos do modelo. Conforme mostrado em [81], a resistência total da bateria ($R_o + R$) diminui com a temperatura, sendo que a resistência série R_o sofre variações maiores se comparada à resistência do efeito de polarização R . Já a curva $v_{oc} \times SOC$ varia muito pouco em função da temperatura, podendo-se desprezar a influência da temperatura para aplicações práticas neste caso. Com relação ao envelhecimento da bateria, o qual leva a uma redução do SOH da mesma, nota-se um leve aumento da resistência interna ($R_o + R$), enquanto a curva $v_{oc} \times SOC$ sofre algumas variações pequenas, em especial nos extremos (SOC próximo de 0 ou 1). Por fim, [81] faz uma análise dos efeitos do SOC sobre os parâmetros da bateria, mostrando que as resistências da bateria são maiores para baixos valores de SOC (menores que 0,1), porém, permanecem praticamente constantes em uma faixa intermediária do SOC (entre 0,3 e 0,8), mesmo sob variações de temperatura.

Além das análises já comentadas, [81] avaliou o grau de influência que cada variável possui sobre o erro de estimação do SOC. Segundo [81], que utilizou uma aproximação da curva $v_{oc} \times SOC$ por segmentos de reta, as variáveis que mais influenciam a estimação do SOC são, nesta ordem: i) a inclinação da curva $v_{oc} \times SOC$, ii) o coeficiente linear da curva $v_{oc} \times SOC$, iii) a variação da resistência interna total ($R_o + R$) e por fim iv) a variação da carga máxima da bateria, ou seja, seu SOH. Nota-se então que a curva $v_{oc} \times SOC$ é o fator de

maior impacto na estimação do SOC de baterias.

Em [26], um EKF é implementado para estimação do SOC através de um ECM com dois pares RC ao invés de apenas um. A discretização do sistema neste caso foi feita através da aproximação de Euler (ou *backward*). Para a obtenção da curva $v_{oc} \times SOC$, foi realizada a carga e descarga da bateria com baixo valor de corrente ($0,02C$), sendo que a curva experimental $v_{oc} \times SOC$ utilizada é a média entre as curvas de carga e descarga, desprezando-se assim o efeito de histerese. Tal curva experimental foi aproximada por um polinômio de grau 12. Com relação aos parâmetros dos pares RC , foram obtidas as constantes de tempo de ambos através de dados experimentais. Um fato interessante deste artigo foi a utilização da resistência série R_o como um estado do sistema, o qual também é estimado conjuntamente com o SOC através do EKF. Os resultados mostraram erros de estimação do SOC abaixo de 2%.

Além de [26], outras publicações também se utilizam do EKF como ferramenta para a estimação do SOC. Em [27] é utilizado um EKF com um ECM composto de 7 pares RC , sendo que a obtenção destes parâmetros é feita por LS através de dados experimentais. A curva $v_{oc} \times SOC$ é aproximada pela combinação linear de um polinômio de grau 3 com um fator exponencial e um fator logarítmico ($\ln$). Já [29] também utiliza um EKF sob um sistema de segunda ordem, porém sua abordagem leva em consideração os efeitos da temperatura, do SOC e da corrente sobre os parâmetros do modelo, os quais são aproximados por funções dependentes destas três variáveis, e também os efeitos da temperatura sobre a capacidade máxima de carga da bateria. Em [25] é utilizado um estimador baseado no EKF com o objetivo de diminuir a influência de ruídos polarizados na estimação. Mais detalhes sobre o EKF serão descritos adiante na seção 5.3 deste trabalho.

Diversos autores também vêm se apoiando no uso do *H-Infinity Filter* (HIF) como ferramenta de estimação do SOC. Em [16] é utilizado o modelo no tempo contínuo para obtenção do SOC através de um HIF, o qual é obtido através de inequações matriciais lineares e desconsidera os efeitos do SOH na estimação do SOC. É importante salientar que, segundo [16], a curva $v_{oc} \times SOC$ é um fator crítico na estimação do SOC. Já [34] afirma que o HIF é um estimador mais robusto em relação

à variação dos parâmetros do modelo e em relação a ruídos polarizados. [34] também afirma que o HIF consegue lidar melhor com ruídos de medição na corrente do que na tensão, uma vez que ruídos de medição na tensão afetam mais diretamente a estimação da tensão v_{oc} , a qual está diretamente e não linearmente relacionada ao SOC, o que acarreta em um maior impacto na estimação.

Outra ferramenta bastante utilizada na estimação do SOC é o *Unscented Kalman Filter* (UKF), o qual é utilizado em [18] e apresenta boa tolerância à ruídos de medição na tensão e na corrente da bateria. Outras variantes do UKF também são utilizadas. O método proposto em [31] utiliza uma combinação entre o *Sigma Point KF* (SPKF - KF de Pontos Sigma), que é uma espécie de UKF, e *Support Vector Machine* (SVM - Máquinas de Vetores de Suporte), que se utiliza de técnicas de aprendizagem, para a estimação do SOC, sendo que os parâmetros do modelo são estimados por *Recursive LS* (RLS - LS Recursivo) e a capacidade da bateria também é estimada através de SVM. Em [32] a variante *Dual Spherical UKF* (DUKFST - UKF Dual Esférico) é utilizada para realizar a estimação do SOC levando em consideração os efeitos de temperatura e realizando conjuntamente a estimação dos parâmetros do modelo. Já em [33], a estimação do SOC combina uma outra variante do UKF com SVM e LS. Estes métodos baseados no UKF e suas variantes apresentam maior complexidade e abrem caminho para novas estratégias de estimação do SOC.

Uma proposta de estimação interessante para arranjos de baterias em série é a apresentada por [82], a qual propõe um *Collective Kalman Filter* (CKF - Filtro de Kalman Coletivo) para a estimação de carga das baterias. Fundamentalmente, este método propõe que, ao invés de se utilizar um KF separadamente para cada bateria, pode ser utilizado um único KF que leva em consideração as informações de todas as baterias de um arranjo série, ou ainda, a utilização de vários KF que compartilham informações da estimação do SOC. Por agregar mais informações para o estimador, este método tende a ser mais eficaz, mostrando melhores condições de observabilidade. Além do método proposto, [82] faz uma consideração importante acerca da observabilidade do sistema na estimação do SOC: o sistema tende a ser mais observável nos extremos da curva $v_{oc} \times SOC$, onde existem

derivadas maiores.

Dentre o contingente de trabalhos que abordam diferentes métodos de estimação do SOC, tem-se em [12] e [30] algumas comparações e considerações sobre tais técnicas. Em [12] são mostrados outros métodos de estimação baseados em ANN e Lógica Fuzzy, por exemplo, que também são abordados na literatura e uma breve comparação entre os diferentes métodos é comentada. Em [30], são comparadas três das técnicas aqui discutidas, o EKF, o UKF e o HIF, na estimação do SOC com foco em aplicações automotivas, considerando-se dois tipos de baterias de lítio distintas. Segundo [30], o UKF possui bons resultados frente aos demais quando a curva $v_{oc} \times SOC$ possui variações maiores, porém, problemas de convergência são observados caso o perfil $v_{oc} \times SOC$ possua derivadas menores (devido à menor condição de observabilidade do sistema). Com relação ao EKF, foi observado que este possui boa robustez na estimação, desde que não haja ruídos polarizados na medição de corrente da bateria. Já o HIF apresentou resultados um tanto imprevisíveis, possuindo variações ao longo do tempo no erro de estimação. Apesar deste fato, o erro de estimação se manteve limitado, e o estimador se mostrou relativamente bom, uma vez que praticamente não depende de parâmetros a serem ajustados.

Finalizando esta revisão sobre métodos de estimação do SOC, tem-se em [71] e [83] uma abordagem que considera um modelo adaptativo de primeira ordem, que é atualizado constantemente e utilizado na estimação do SOC. Enquanto [71] desenvolve esta técnica utilizando um observador de Luenberger, [83] mostra os resultados obtidos com a utilização de um KF. Nestes trabalhos, a curva $v_{oc} \times SOC$ também foi mapeada por diversos segmentos lineares através de dados já conhecidos, enquanto que os parâmetros do modelo são encontrados iterativamente através do algoritmo *Moving-Window LS* (MWLS - LS de Janela Móvel). Este conceito utilizado em [71] e [83] foi utilizado neste trabalho e será comentado na seção 5.5.

Na lista a seguir, são destacados os pontos principais na visão do autor deste trabalho sobre os trabalhos mencionados nesta seção, sendo que muitos deles servem de base para as análises que serão apresentadas ao longo deste capítulo.

- A curva característica $v_{oc} \times SOC$ é assumida conhecida em todos os trabalhos lidos e selecionados pelo autor, seja através de dados do fabricante ou de dados empíricos. A estimação desta curva ou mesmo de suas aproximações lineares por segmentos de reta não é realizada em nenhum destes trabalhos.
- Conforme citado por vários trabalhos, a curva $v_{oc} \times SOC$ é o fator de maior impacto na estimação do SOC.
- A observabilidade do sistema é maior nas regiões da curva $v_{oc} \times SOC$ com maiores derivadas.
- Os parâmetros do ECM (R_o , R e C) podem ser estimados com facilidade utilizando-se diversos métodos, à exemplo do algoritmo LS e de suas variantes. Muitos trabalhos também fazem a obtenção de tais parâmetros através de experimentos, podendo ou não levar em consideração fatores como temperatura, SOH, entre outros.
- Segundo [81], as resistências da bateria são maiores para baixos valores de SOC (menores que 0,1), porém, permanecem praticamente constantes em uma faixa intermediária do SOC (entre 0,3 e 0,8), mesmo sob variações de temperatura. Ainda segundo [81], as resistências da bateria têm um impacto menor sobre a estimação do SOC do que a curva $v_{oc} \times SOC$, porém possuem maior influência que o SOH.
- A utilização de estratégias baseadas em estimadores de estado, como o KF e suas variantes e o HIF tem se mostrado largamente utilizada por diversos pesquisadores no âmbito de estimação do SOC de baterias.

Feita esta revisão sobre alguns dos métodos de estimação encontrados na literatura, serão abordados na seção 5.3 em detalhes o KF e o EKF, os quais serão utilizados para estimação do SOC neste trabalho.

5.3 O FILTRO DE KALMAN E O FILTRO DE KALMAN ESTENDIDO

A história do Filtro de Kalman começou a ser desenhada inicialmente pelos pesquisadores Kolmogorov, Krein e Wiener, onde deu-se início à utilização de conceitos estatísticos nos problemas de estimação e controle, caracterizando sinais no tempo através de propriedades estocásticas como média e variância. A partir dos estudos desempenhados principalmente por Kolmogorov e Wiener, Kalman desenvolveu uma solução recursiva para o problema de filtragem e predição, a qual poderia ser implementada em sistemas digitais, os quais eram tecnologias emergentes na época. Esta solução passou então a ser conhecida como Filtro de Kalman [84].

A solução do KF busca encontrar a função densidade de probabilidade dos sinais de um sistema, considerando-se que os mesmos possuam característica gaussiana, o que resulta em encontrar a média e a variância de tais sinais, podendo-se assim reconstruir sua função densidade de probabilidade [84]. Em tal solução, o valor médio ou esperança encontrada pelo KF é nada mais que o vetor de estados do sistema $\hat{X}(k)$. Como os estados do sistema compõem um vetor $X(k)$, a variância de tais sinais não é dada apenas por um valor escalar σ^2 , mas sim por uma matriz, a chamada matriz de covariância do sistema P .

Na seção 3.6.4 um Filtro de Kalman foi desenvolvido com o objetivo de filtrar as componentes de alta frequência dos sinais utilizados no controlador, porém utilizando-se de dados passados para tal, o que configura um preditor. Nesta seção, o objetivo do Filtro de Kalman é realizar estimações sobre a bateria, em especial do SOC.

Para que esta noção entre predição e estimação fique mais clara, deve-se atentar à nomenclatura utilizada. Conforme dito por [84], um estimador é capaz de avaliar um conjunto de dados obtidos no presente, passado ou futuro, enquanto que um preditor tenta obter um determinado sinal a partir de dados já disponíveis no presente. Assim, o KF utilizado como preditor, conforme feito na seção 3.6.4, busca prever medições de corrente $\hat{i}_L(k)$ e tensão $\hat{v}_o(k)$ a partir dos valores lidos anteriormente $i_L(k-1)$ e $v_o(k-1)$, enquanto que o KF como estimador

que será mostrado neste capítulo almeja inferir, por exemplo, qual é o valor do SOC com base nos dados disponíveis de corrente $i_b(k)$ e tensão terminal $v_t(k)$ da bateria.

Com relação ao preditor implementado na seção 3.6.4, tem-se um ganho L fixo que é obtido através da solução da equação algébrica de Riccati, a qual fornece como resultado a matriz de covariância P do sistema em regime permanente. Assim, o Filtro de Kalman utilizado no capítulo 3 possui uma solução não recursiva, calculada *offline* e posteriormente utilizada na predição dos dados da planta. Como o ganho do KF é fixo nesta situação, os custos computacionais são reduzidos drasticamente graças à sua obtenção *offline*.

No caso da estimação do SOC de baterias, tem-se um sistema não-linear e variante no tempo. Sendo o KF um observador aplicável unicamente em sistemas lineares, este não se mostra adequado à estimação de carga da bateria sem que hajam adaptações, já que a obtenção do ganho L fixo considera que o sistema é linear e invariante no tempo. Para tornar o KF adequado à estimação de carga de baterias, é adotado o seu algoritmo recursivo, o qual pode ser utilizado em conjunto com técnicas de estimação adaptativas. Ainda, além do KF na sua forma recursiva, será demonstrado neste capítulo o EKF, que é uma variante do KF capaz de lidar com sistemas não lineares, mas que também necessita ser atualizado recursivamente. A seção 5.3.1 traz alguns fundamentos acerca do KF enquanto a seção 5.3.2 apresenta os princípios básicos do EKF.

5.3.1 O Filtro de Kalman

Para descrever resumidamente sobre o Filtro de Kalman, deve-se primeiramente entender a nomenclatura aqui utilizada para representação das estimativas *a priori* e *a posteriori*. Um sinal estimado *a priori* é denotado por $x(k|k-1)$ (lê-se x de k dado $k-1$), o qual representa o valor estimado de x no instante k (atual) levando-se em conta apenas as informações dadas no instante $k-1$. Já um sinal estimado *a posteriori* é denotado por $x(k|k)$ e representa um sinal x estimado no instante k dadas as informações contidas também no instante k . Entendido este conceito, passa-se rapidamente a uma descrição do KF.

Seja um sistema linear e invariante no tempo dado por (5.1).

$$\begin{cases} X(k) = AX(k-1) + Bu(k-1) + w(k) \\ y(k) = CX(k) + Du(k) + v(k) \end{cases} \quad (5.1)$$

Onde $w(k)$ é um vetor de ruídos presentes nos estados do processo, o qual contém média nula e covariância $Q_{kf}(k)$, ou seja, $w(k) \sim N(0, Q_{kf}(k))$, e $v(k)$ representa um vetor de ruídos presentes na medição da saída, o qual também possui média nula e covariância $R_{kf}(k)$, ou seja, $v(k) \sim N(0, R_{kf}(k))$. Agora, seja $\hat{x}(k|k)$ a estimativa dos estados do sistema dadas as medidas realizadas até o instante k e $P_{kf}(k|k)$ a matriz de covariância do vetor de erros de estimação dadas as observações também até o instante k , a qual representa o quanto os estados e consequentemente seu erro estão variando, de forma semelhante a variância σ^2 de uma variável escalar. Dados estes conceitos, apresenta-se as duas etapas, Predição e Atualização, que compõe o algoritmo do KF.

Predição:

$$\hat{X}(k|k-1) = A\hat{X}(k-1|k-1) + Bu(k-1)$$

$$P_{kf}(k|k-1) = AP_{kf}(k-1|k-1)A^T + Q_{kf}(k)$$

$$\hat{y}(k|k-1) = C\hat{X}(k|k-1)$$

Atualização:

$$e(k) = y(k) - \hat{y}(k|k-1)$$

$$S_{kf}(k) = CP_{kf}(k|k-1)C^T + R_{kf}(k)$$

$$L(k) = P_{kf}(k|k-1)C^TS^{-1}(k)$$

$$\hat{X}(k|k) = \hat{X}(k|k-1) + L(k)e(k)$$

$$P_{kf}(k|k) = (I - L(k)C)P_{kf}(k|k-1)$$

Conforme pode ser visto no algoritmo, a etapa de predição se utiliza apenas dos dados do modelo para estimar *a priori* o vetor de estados $\hat{X}(k|k-1)$, sua saída $\hat{y}(k|k-1)$ e a matriz de covariância $P_{kf}(k|k-1)$. A partir dos dados *a priori*, é calculado então o erro entre a saída estimada $\hat{y}(k|k-1)$ e a saída medida $y(k)$, e o resíduo de estimação da matriz de covariância $S_{kf}(k)$. Por fim, calcula-se o valor do ganho do KF $L(k)$, e através deste, obtém-se as estimações *a posteriori* de $\hat{X}(k|k)$ e $P_{kf}(k|k)$ do sistema.

Apesar da notação indicar que tais matrizes variam no tempo, muitas das vezes estas são assumidas constantes, em geral fixadas nos valores já esperados de covariância do sistema em regime permanente. Nota-se que a matriz $Q_{kf}(k)$ compõe uma parcela do cálculo da matriz $P_{kf}(k|k-1)$, enquanto a matriz $R_{kf}(k)$ compõe uma parcela da matriz de resíduo da covariância $S_{kf}(k)$. Uma vez que o KF atua buscando minimizar a matriz de covariância $P_{kf}(k|k)$, em uma sistema estável esta poderia vir a se tornar singular. Nos casos em que $Q_{kf}(k)$ ou $R_{kf}(k)$ são singulares, ter a matriz $P_{kf}(k|k)$ singular pode implicar em um valor de $S_{kf}(k)$ também singular, tornando tal matriz não invertível e impossibilitando o cálculo do ganho $L(k)$. Por este motivo opta-se sempre por matrizes $Q_{kf}(k)$ e $R_{kf}(k)$ definidas positivas, garantindo-se assim a convergência do algoritmo.

5.3.2 O Filtro de Kalman Estendido

O Filtro de Kalman Estendido é nada mais que uma generalização do Filtro de Kalman para sistemas não lineares. A ideia do EKF é linearizar um sistema não linear ao redor do seu ponto de operação, utilizando-se o aproximações por séries de Taylor. Considera-se um sistema qualquer dado pela equação (5.2).

$$\begin{cases} X(k) = f(X(k-1), u(k-1)) + w(k) \\ y(k) = h(X(k), u(k)) + v(k) \end{cases} \quad (5.2)$$

Onde $f(\bullet)$ e $h(\bullet)$ são funções que representam o sistema dinâmico, $w(k)$ e $v(k)$ são ruídos gaussianos de covariância $Q_{kf}(k)$ e $R_{kf}(k)$, respectivamente.

As matrizes Jacobianas deste sistema são dadas por (5.3) e (5.4).

$$A(k-1) = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}(k-1|k-1), u(k-1)} \quad (5.3)$$

$$C(k) = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}(k|k-1)} \quad (5.4)$$

Onde $A(k-1)$ e $C(k)$ são matrizes que representam a linearização do sistema dado o instante de tempo k . Assim como no KF, o EKF é dividido entre as etapas de Predição e Atualização, conforme mostrado a seguir.

Predição:

$$\hat{X}(k|k-1) = f(\hat{X}(k-1|k-1), u(k-1))$$

$$P_{kf}(k|k-1) = A(k-1)P_{kf}(k-1|k-1)A^T(k-1) + Q_{kf}(k)$$

$$\hat{y}(k|k-1) = h(\hat{X}(k|k-1), u(k))$$

Atualização:

$$e(k) = y(k) - \hat{y}(k|k-1)$$

$$S_{kf}(k) = C(k)P_{kf}(k|k-1)C^T(k) + R_{kf}(k)$$

$$L(k) = P_{kf}(k|k-1)C^T(k)S^{-1}(k)$$

$$\hat{X}(k|k) = \hat{X}(k|k-1) + L(k)e(k)$$

$$P_{kf}(k|k) = (I - L(k)C(k))P_{kf}(k|k-1)$$

Nota-se que o algoritmo do KF é muito similar ao algoritmo do EKF, diferindo-se apenas pelo cálculo do vetor de estados X e da saída y através das funções $f(\bullet)$ e $h(\bullet)$ e também pela utilizações das matrizes $A(k-1)$ e $C(k)$, que não são mais constantes. Ainda, caso $f(\bullet)$ e $h(\bullet)$ sejam funções lineares, recai-se novamente ao KF convencional, com matrizes A e C constantes.

Comentados brevemente os conceitos do KF e do EKF, passa-se então às estratégias de estimação do SOC utilizadas neste trabalho.

5.4 ESTIMAÇÃO DO SOC COM FILTRO DE KALMAN ESTENDIDO

Conforme abordado na seção 5.2, diversos trabalhos adotam o EKF como ferramenta para estimação do SOC, uma vez que este se mostra adequado para sistemas não lineares.

A partir dos equacionamentos obtidos na capítulo 4 sobre modelagem de baterias, destacam-se aqui três relações importantes, que tratam os sinais do SOC, da tensão dinâmica v_{rc} e da tensão terminal v_t da bateria, os quais são novamente mostrados na equação (5.5),

lembrando que $z_d = e^{-T_s/RC}$.

$$\begin{cases} SOC(k) = SOC(k-1) - \frac{T_a}{Q_{max}} i_b(k-1) \\ v_{rc}(k) = z_d v_{rc}(k-1) + R(1 - z_d) i_b(k-1) \\ v_t(k) = v_{oc}(k) - v_{rc}(k) - R_o i_b(k) \end{cases} \quad (5.5)$$

Nota-se a partir da equação (5.5) que $SOC(k)$ e $v_{rc}(k)$ são relações recursivas e independentes entre si, tendo em comum apenas o termo $i_b(k-1)$, as quais formam a equação de estados do sistema, sendo $X(k) = [SOC(k) \ v_{rc}(k)]^T$ e $u(k) = i_b(k)$. Ainda, é conveniente adotar a tensão $v_t(k)$ como saída do sistema, uma vez que ela é uma variável que pode ser mensurada. Escrevendo o sistema na forma padrão em espaço de estados, tem-se a equação (5.6).

$$\begin{bmatrix} SOC(k) \\ v_{rc}(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & z_d \end{bmatrix} \begin{bmatrix} SOC(k-1) \\ v_{rc}(k-1) \end{bmatrix} + \begin{bmatrix} \frac{-T_a}{Q_{max}} \\ R(1-z_d) \end{bmatrix} i_b(k-1) \quad (5.6)$$

$$v_t(k) = v_{oc}(k) - v_{rc}(k) - R_o i_b(k)$$

Conforme apresentado na seção 5.3.2, um sistema não linear pode ser descrito conforme a equação (5.2). Considerando que tais funções que descrevem o sistema são contínuas e deriváveis, que é o caso das baterias, pode-se obter então uma linearização do sistema a partir das relações dadas por (5.3) e (5.4). No caso do sistema apresentado pela equação (5.6), tem-se as matrizes do sistema linearizado dispostas nas equações (5.8) e (5.10).

$$A(k-1) = \begin{bmatrix} \frac{\partial SOC(k)}{\partial SOC(k-1)} & \frac{\partial SOC(k)}{\partial v_{rc}(k-1)} \\ \frac{\partial v_{rc}(k)}{\partial SOC(k-1)} & \frac{\partial v_{rc}(k)}{\partial v_{rc}(k-1)} \end{bmatrix} \bigg|_{X(k-1|k-1), u(k-1)} \quad (5.7)$$

$$A(k-1) = \begin{bmatrix} 1 & 0 \\ 0 & z_d \end{bmatrix} \quad (5.8)$$

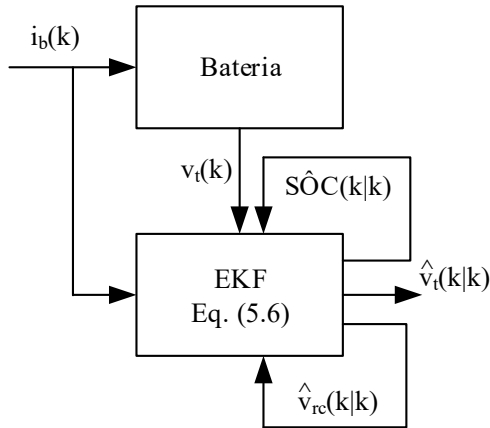
$$C(k) = \left[\frac{\partial v_t(k)}{\partial SOC(k)} \quad \frac{\partial v_t(k)}{\partial v_{rc}(k)} \right] \bigg|_{X(k|k-1), u(k)} \quad (5.9)$$

$$C(k) = [\beta_1(k) \quad -1] \quad (5.10)$$

Tais matrizes são variantes no tempo e são obtidas a cada período de amostragem. Nota-se que a matriz $C(k)$ é calculada tendo-se conhecimento da curva $v_{oc} \times SOC$, a qual também é um requisito para o cálculo da tensão de saída $v_t(k)$. Mais precisamente, a matriz $C(k)$ depende da inclinação da curva $v_{oc} \times SOC$, a qual é chamada de $\beta_1(k)$. Tendo-se tais informações, pode-se aplicar então o algoritmo do EKF descrito na seção 5.3.2 para estimação do SOC, desde que o sistema seja comprovadamente observável.

A Figura 42 ilustra o diagrama de blocos geral do algoritmo de estimação baseado no EKF.

Figura 42 – Diagrama de blocos do estimador com EKF



Fonte: produção do próprio autor, 2016

A prova de observabilidade do sistema é apresentada na sequência.

5.4.1 Análise de Observabilidade do Sistema Linearizado

A análise de observabilidade para o EKF é análoga à apresentada na seção 3.6.4.1, porém, deve-se ter em mente que neste caso, como o sistema é variante no tempo, suas matrizes A e C não são constantes, requisitando uma análise um pouco mais aprofundada. Através das matrizes $A(k-1)$ e $C(k)$, dadas pelas equações (5.8) e (5.10), pode-se obter a matriz de observabilidade do sistema, conforme equação (5.11).

$$M_o = \begin{bmatrix} \beta_1(k) & -1 \\ \beta_1(k) & -z_d \end{bmatrix} \quad (5.11)$$

Nota-se que tal matriz é linearmente independente desde que $z_d \neq 1$. Como $z_d = e^{-T_s/RC}$, esta condição de observabilidade é satisfeita desde que a constante de tempo RC não assuma valores muito grandes, tendendo ao infinito, o que pode ser garantido facilmente pela limitação do domínio dos parâmetros R e C . No caso do algoritmo do EKF, foram assumidos constantes os parâmetros R_o , R e C baseando-se nos conceitos ditos por [81] e no fato da curva $v_{oc} \times SOC$ ser o fator mais relevante na estimação do SOC, conforme abordado na seção 5.2, a qual é conhecida nesta implementação. Logo, a observabilidade do sistema é assegurada neste modelo.

Sabendo que o sistema é observável, o EKF foi aplicado para estimação do SOC e seus resultados encontram-se dispostos na seção 5.7.

5.5 ESTIMAÇÃO DO SOC COM FILTRO DE KALMAN A PARTIR DE UM MODELO ADAPTATIVO

Este método de estimação é de fato bem semelhante à ideia do EKF, utilizando-se também da linearização do sistema a cada iteração para então realizar a estimação do SOC. Sua principal diferença reside no fato de que a equação de saída também é linearizada, ao invés de se utilizar de sua forma não linear, como é feito no EKF. Para tal, descreve-se a curva $v_{oc} \times SOC$ a partir da relação linear dada por

(5.12).

$$v_{oc} = \beta_1(k).SOC(k) + \beta_o(k) \quad (5.12)$$

Na equação (5.12), $\beta_1(k)$ representa a inclinação da curva $v_{oc} \times SOC$ enquanto $\beta_o(k)$ representa o seu coeficiente linear. Tem-se evidente uma relação clara entre esta abordagem e a utilizada no EKF, uma vez que o fator $\beta_1(k)$ é o mesmo fator obtido na matriz $C(k)$. A partir das equações (5.6) e (5.12), tem-se que o sistema pode ser representado na sua forma linear conforme equação (5.13).

$$\begin{bmatrix} SOC(k) \\ v_{rc}(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & z_d \end{bmatrix} \begin{bmatrix} SOC(k-1) \\ v_{rc}(k-1) \end{bmatrix} + \begin{bmatrix} \frac{-T_a}{Q_{max}} \\ R(1-z_d) \end{bmatrix} i_b(k-1) \quad (5.13)$$

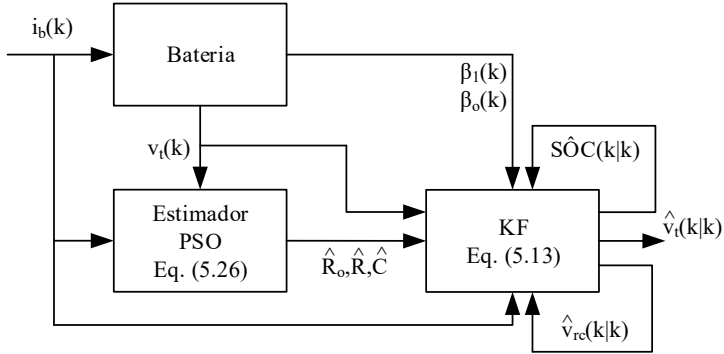
$$v_t(k) = \beta_1(k)SOC(k) + \beta_o(k) - v_{rc}(k) - R_o i_b(k)$$

Tal modelo foi proposto inicialmente por [71], no qual era realizada a estimação dos parâmetros R_o , R , C e β_1 através de uma variante do método dos mínimos quadrados. Porém, apesar de obter o parâmetro β_1 , [71] ainda se utiliza de valores pré-definidos da curva $v_{oc} \times SOC$ para encontrar o parâmetro β_o , não dispensando assim o conhecimento desta curva. Uma vez conhecidos os parâmetros do modelo linearizado da bateria, um KF era aplicado na estimação do SOC e de v_{rc} .

A proposta deste trabalho é uma adaptação do modelo apresentado por [71], porém, é realizada a estimação apenas dos parâmetros R_o , R e C através do algoritmo PSO, enquanto que os parâmetros β_1 e β_o são assumidos conhecidos. Ainda, como os parâmetros R_o , R e C variam lentamente em relação as demais variáveis do sistema, sua estimação é realizada a cada 60 s, com o objetivo de diminuir o custo computacional do algoritmo e consequentemente seu tempo de execução. A Figura 43 ilustra o diagrama de blocos do estimador.

Com relação a observabilidade, recai-se exatamente sobre o mesmo caso abordado na seção 5.4.1. Desta forma, o sistema é observável desde que a constante de tempo RC não assuma valores tendendo ao infinito, o que é facilmente garantido limitando-se o espaço de busca

Figura 43 – Diagrama de blocos do estimador com KF adaptativo



Fonte: produção do próprio autor, 2016

do algoritmo PSO.

Enquanto que na simulação de baterias o valor da tensão dinâmica v_{rc} é conhecido, na implementação tal valor é desconhecido, uma vez que se tem acesso apenas às medições de corrente i_b e tensão v_t . Desta forma, a estimação dos parâmetros R_o , R e C deve ser realizada apenas conhecendo-se estas duas medições, a qual será comentada na seção 5.5.1.

5.5.1 Identificação dos parâmetros resistivos e capacitivos da bateria

Enquanto que em [71] a identificação do sistema é feita a partir de um modelo contínuo posteriormente discretizado pela transformação de Tustin, aqui será utilizada a relação de Euler (*backward*) no modelo discreto. Uma vez que o sistema será avaliado dentro do algoritmo PSO, este será calculado diversas vezes a cada iteração, logo, a utilização de modelos mais simples diminuem o tempo de execução total do algoritmo de identificação de parâmetros.

Considera-se o modelo dado por (5.14), no qual todos os parâ-

metros são considerados constantes.

$$\begin{cases} SOC(k) = SOC(k-1) - \frac{T_a}{Q_{max}} i_b(k-1) \\ vrc(k) = z_d vrc(k-1) + R(1-z_d) i_b(k-1) \\ v_t(k) = \beta_1 SOC(k) + \beta_o - v_{rc}(k) - R_o i_b(k) \end{cases} \quad (5.14)$$

A partir das relações dadas em (5.14) e utilizando-se o conceito de diferenças finitas, pode-se chegar às equações dadas por (5.15), (5.16) e (5.17).

$$\Delta SOC(k) = \frac{-T_a}{Q_{max}} i_b(k-1) \quad (5.15)$$

$$\Delta v_{rc}(k) = (z_d - 1) v_{rc}(k) + R(1 - z_d) i_b(k-1) \quad (5.16)$$

$$\Delta v_t(k) = \beta_1 \Delta SOC(k) + \Delta \beta_o - \Delta v_{rc}(k) - R_o \Delta i_b(k) \quad (5.17)$$

Como β_o é assumido constante, $\Delta \beta_o \approx 0$. A partir do modelo (5.14), chega-se na equação (5.18).

$$v_t(k-1) = \beta_1 SOC(k-1) + \beta_o - v_{rc}(k-1) - R_o i_b(k-1) \quad (5.18)$$

Manipulando-se a equação (5.18), tem-se a equação (5.19).

$$v_{rc}(k-1) = \beta_1 SOC(k-1) + \beta_o - v_t(k-1) - R_o i_b(k-1) \quad (5.19)$$

Usando a equação (5.19) na relação (5.16), tem-se a equação (5.20).

$$\begin{cases} \Delta v_{rc}(k) = ... \\ (z_d - 1) [\beta_1 SOC(k-1) + \beta_o - v_t(k-1) - R_o i_b(k-1)] + ... \\ + R \cdot (1 - z_d) i_b(k-1) \end{cases} \quad (5.20)$$

Agora, usando (5.20) em (5.17), chega-se a relação (5.21).

$$\left\{ \begin{array}{l} \Delta v_t(k) = \beta_1 \Delta SOC(k) + ... \\ -(z_d - 1) [\beta_1 SOC(k-1) + \beta_o v_t(k-1) - R_o i_b(k-1)] + ... \\ - R(1 - z_d) i_b(k-1) - R_o \Delta i_b(k) \end{array} \right. \quad (5.21)$$

Utilizando a relação (5.15) em (5.21), obtém-se (5.22).

$$\left\{ \begin{array}{l} \Delta v_t(k) = \beta_1 \left(\frac{-T_a}{Q_{max}} i_b(k-1) \right) + ... \\ -(z_d - 1) [\beta_1 SOC(k-1) + \beta_o v_t(k-1) - R_o i_b(k-1)] + ... \\ - R(1 - z_d) i_b(k-1) - R_o \Delta i_b(k) \end{array} \right. \quad (5.22)$$

Agora, aplicando-se o operador Δ em (5.22), chega-se na seguinte expressão, dada por (5.23).

$$\left\{ \begin{array}{l} \Delta^2 v_t(k) = \beta_1 \left(\frac{-T_a}{Q_{max}} \Delta i_b(k-1) \right) + ... \\ -(z_d - 1) [\beta_1 \Delta SOC(k-1) + \Delta \beta_o v_t(k-1) - R_o \Delta i_b(k-1)] + ... \\ - R(1 - z_d) \Delta i_b(k-1) - R_o \Delta^2 i_b(k) \end{array} \right. \quad (5.23)$$

Ao utilizar novamente a relação (5.15) em (5.23), tem-se a equação (5.24).

$$\left\{ \begin{array}{l} \Delta^2 v_t(k) = \beta_1 \left(\frac{-T_a}{Q_{max}} \Delta i_b(k-1) \right) + ... \\ -(z_d - 1) \left[\beta_1 \frac{-T_a}{Q_{max}} i_b(k-2) + \Delta \beta_o v_t(k-1) - R_o \Delta i_b(k-1) \right] + ... \\ - R(1 - z_d) \Delta i_b(k-1) - R_o \Delta^2 i_b(k) \end{array} \right. \quad (5.24)$$

Rearranjando os termos em (5.24), chega-se em (5.25).

$$\left\{ \begin{aligned} &\Delta^2 v_t(k) + (1 - z_d)\Delta v_t(k-1) = -R_o\Delta^2 i_b(k) + \dots \\ &+ \Delta i_b(k-1) \left[\frac{-T_a\beta_1}{Q_{max}} - R_o(1 - z_d) - R(1 - z_d) \right] + \dots \\ &+ i_b(k-2) \left[z_d \frac{-T_a\beta_1}{Q_{max}} (1 - z_d) \right] \end{aligned} \right. \quad (5.25)$$

Expandindo os termos com o operador $\Delta = 1 - z^{-1}$ e reagrupando os termos similares, tem-se finalmente a equação (5.26) onde $v_t(k)$ é função $i_b(k)$ e de suas amostras passadas.

$$v_t(k) + a_1 v_t(k-1) + a_2 v_t(k-2) = b_o i_b(k) + b_1 i_b(k-1) + b_2 i_b(k-2) \quad (5.26)$$

Sendo as constantes dadas pelas relações de (5.27) a (5.31).

$$a_1 = -1 - z_d \quad (5.27)$$

$$a_2 = z_d \quad (5.28)$$

$$b_o = -R_o \quad (5.29)$$

$$b_1 = \frac{-T_a\beta_1}{Q_{max}} + R_o(1 - z_d) - R(1 - z_d) \quad (5.30)$$

$$b_2 = z_d \frac{T_a\beta_1}{Q_{max}} - z_d R_o + R(1 - z_d) \quad (5.31)$$

Este modelo, por ser função somente de medições v_t e i_b , as quais estão disponíveis, pode ser utilizado para estimação dos parâmetros R_o , R e C , sendo que o parâmetro β_1 é um parâmetro já conhecido, neste caso. Por ser baseado na relação de Euler e não de Tustin, tem-se uma menor complexidade menor custo computacional na implementação quando comparado ao utilizado por [71].

5.6 ESTIMAÇÃO DO SOC COM FILTRO DE KALMAN A PARTIR DE UM MODELO ADAPTATIVO DE ORDEM 3

A estimação da relação não linear $v_{oc} \times SOC$ é muito pouco abordada na literatura, onde na maioria das vezes assume-se que tal relação já é conhecida. Porém, alguns fabricantes não disponibilizam tal relação em seu catálogos, e sua obtenção pode ser um tanto custosa, ainda mais se houver necessidade de se obter a curva para cada célula.

Para tentar sanar tal problema, é proposto neste trabalho um método que busca estimar todo o modelo completo da bateria juntamente com o SOC, na tentativa de obter um estimador capaz de obter tanto os parâmetros da bateria quanto o seu SOC, utilizando-se o fator $\beta_o(k)$ como um novo estado do sistema. A seção 5.6.1 demonstra as etapas realizadas para modelar tal parâmetro como um estado do sistema, bem como todas as suas considerações.

5.6.1 A obtenção da nova variável de estado da bateria

Na seção 5.5 foi apresentado um modelo linearizado da tensão de circuito aberto da bateria em função de dois parâmetros, β_1 e β_o , conforme equação (5.13). Na metodologia proposta por [71], β_1 é um parâmetro passível de ser estimado, enquanto que β_o não. Verificando-se a equação (5.26) dada na seção 5.5.1, nota-se que a dinâmica $v_t(z^{-1})/i_b(z^{-1})$ não depende do fator β_o , não podendo-se assim estimar tal parâmetro a partir deste equacionamento.

Considerando-se a equação (5.12) no tempo contínuo, tem-se a seguinte relação, dada por (5.32).

$$v_{oc}(SOC) = \beta_1(SOC)SOC + \beta_o(SOC) \quad (5.32)$$

Derivando a expressão (5.32) em função do SOC, tem-se (5.33).

$$\left\{ \begin{array}{l} \frac{\partial v_{oc}}{\partial SOC} = \frac{\partial \beta_1}{\partial SOC}SOC + \frac{\partial SOC}{\partial SOC}\beta_1 + \frac{\partial \beta_o}{\partial SOC} = \dots \\ \frac{\partial \beta_1}{\partial SOC}SOC + \beta_1 + \frac{\partial \beta_o}{\partial SOC} \end{array} \right. \quad (5.33)$$

Ao considerar os parâmetros β_1 e β_o constantes e derivar a expressão (5.32) em função do SOC, tem-se a relação aproximada, dada por (5.34).

$$\frac{\partial v_{oc}}{\partial SOC} \approx \beta_1 \quad (5.34)$$

Agora, substituindo a aproximação dada por (5.34) na relação (5.33), chega-se na expressão (5.35).

$$\frac{\partial \beta_1}{\partial SOC} SOC \approx -\frac{\partial \beta_o}{\partial SOC} \quad (5.35)$$

Esta aproximação traz um importante conceito acerca das variáveis β_1 e β_o . Suas derivadas parciais em função do SOC possuem sinais opostos. Apesar deste conceito estar contido em uma relação aproximada, ela de fato agrega uma informação adicional ao sistema. A partir desta relação encontrada no domínio contínuo, será buscada uma relação análoga no domínio discreto. Utilizando os conceitos de diferenças finitas sobre a equação (5.12), tem-se a relação (5.36).

$$\begin{cases} \Delta v_{oc}(k) = \Delta \beta_1(k) SOC(k) + \beta_1(k) \Delta SOC(k) + ... \\ - \Delta \beta_1(k) \Delta SOC(k) + \Delta \beta_o(k) \end{cases} \quad (5.36)$$

Novamente, assumindo-se que os parâmetros $\beta_1(k)$ e $\beta_o(k)$ são constantes e obtendo a primeira diferença de (5.12), tem-se a aproximação dada por (5.37).

$$\Delta v_{oc}(k) \approx \beta_1(k) \Delta SOC(k) \quad (5.37)$$

Ao substituir a aproximação dada em (5.37) na equação (5.36), simplificando-se os termos, chega-se à equação (5.38), a qual descreve a variação de $\beta_o(k)$ em função da variação de $\beta_1(k)$ e de $SOC(k-1)$.

$$\Delta \beta_o(k) \approx -\Delta \beta_1(k) SOC(K-1) \quad (5.38)$$

Tal relação descrita por (5.38) é linear em função dos estados, e dependente ainda do parâmetro $\Delta \beta_1(k)$, o qual pode ser estimado

a cada iteração. A partir da relação encontrada, pode-se manipular o sistema dado por (5.13) e descrevê-lo como um sistema de terceira ordem, conforme equação (5.39).

$$\begin{bmatrix} SOC(k) \\ v_{rc}(k) \\ \beta_o(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & z_d & 0 \\ -\Delta\beta_1(k) & 0 & 1 \end{bmatrix} \begin{bmatrix} SOC(k-1) \\ v_{rc}(k-1) \\ \beta_o(k-1) \end{bmatrix} + \begin{bmatrix} \frac{-T_a}{\bar{Q}_{max}} \\ R(1-z_d) \\ 0 \end{bmatrix} i_b(k-1) \quad (5.39)$$

$$v_t(k) = \beta_1(k)SOC(k) + \beta_o(k) - v_{rc}(k) - R_o i_b(k)$$

Ainda, será adicionado um fator λ , conhecido como fator de esquecimento, ao sistema, que posteriormente será utilizado no algoritmo, resultando no modelo dado por (5.40).

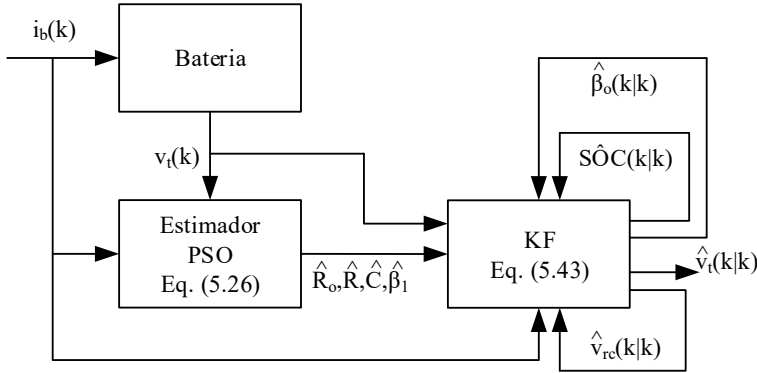
$$\begin{bmatrix} SOC(k) \\ v_{rc}(k) \\ \beta_o(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & z_d & 0 \\ -\Delta\beta_1(k) & 0 & \lambda \end{bmatrix} \begin{bmatrix} SOC(k-1) \\ v_{rc}(k-1) \\ \beta_o(k-1) \end{bmatrix} + \begin{bmatrix} \frac{-T_a}{\bar{Q}_{max}} \\ R(1-z_d) \\ 0 \end{bmatrix} i_b(k-1) \quad (5.40)$$

$$v_t(k) = \beta_1(k)SOC(k) + \beta_o(k) - v_{rc}(k) - R_o i_b(k)$$

Tendo-se o estado $\beta_o(k)$ no modelo, pode-se realizar a estimação do mesmo juntamente com a estimação do SOC através do KF, desde que o sistema seja observável. A Figura 44 mostra o diagrama de blocos de estimação utilizando-se esta configuração.

Na seção 5.6.2, será discutida a observabilidade deste modelo proposto.

Figura 44 – Diagrama de blocos do estimador com KF adaptativo e costimação da curva $v_{oc} \times SOC$



Fonte: produção do próprio autor, 2016

5.6.2 Análise de Observabilidade do Sistema Linearizado de ordem 3

A partir do sistema dado pela equação (5.40), tem-se as matrizes A e C do sistema, conforme relações (5.41) e (5.42).

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & z_d & 0 \\ -\Delta\beta_1(k) & 0 & \lambda \end{bmatrix} \quad (5.41)$$

$$C = [\beta_1(k) \quad -1 \quad 1] \quad (5.42)$$

A partir destas matrizes, monta-se a matriz de observabilidade do sistema, dada por (5.43).

$$M_o = \begin{bmatrix} \beta_1(k) & -1 & 1 \\ \beta_1(k) - \Delta\beta_1(k) & -z_d & \lambda \\ \beta_1(k) - \Delta\beta_1(k)(1 + \lambda) & -z_d^2 & \lambda^2 \end{bmatrix} \quad (5.43)$$

A partir desta matriz, obtém algumas informações importantes

acerca da observabilidade do sistema:

- No caso de $\lambda = 1$, o sistema é observável se, e somente se, $\Delta\beta_1(k) \neq 0$.
- No caso de $z_d = 1$, o sistema é observável se, e somente se, $\Delta\beta_1(k) \neq 0$.
- No caso de $\lambda \neq 1$, o sistema é observável desde que $\lambda \neq z_d$. O mesmo vale para o caso em que $z_d \neq 1$.

Dadas as considerações sobre a observabilidade do modelo, nota-se que este é observável dentro de uma boa margem de operação, já que $z_d \neq 1$ é sempre respeitada, $\lambda \neq 1$ é uma condição que pode ser facilmente imposta e $\lambda = z_d$ é uma condição rara. Ainda, caso ocorra, pode-se utilizar de restrições simples para tratar tal acontecimento. Dadas as considerações acerca deste modelo e também dos demais descritos anteriormente, passa-se à seção de resultados de cada um deles na estimativa do SOC de baterias.

5.7 RESULTADOS DA ESTIMAÇÃO DO SOC DE BATERIAS

Os resultados de cada um dos métodos de estimação de carga são apresentados nas seções 5.7.1, 5.7.3 e 5.7.4. Destes, apenas o método de estimação com EKF apresenta resultados experimentais e de simulação, enquanto os demais compõem-se apenas de resultados de simulação.

Todos os resultados obtidos por simulação na estimação do SOC levam em conta uma duração de 6 horas e três pontos de operação do valor de carga: 0, 0,35 e 0,7. O sistema então inicia com um valor de SOC dentre estes três mencionados, enquanto que a estimação sempre inicia em zero, visando demonstrar o funcionamento da mesma diante de erros de estimação do valor inicial.

Ainda, é utilizado o mesmo perfil de carga durante todas as simulações, iniciando-se com corrente constante de 825 mA até atingir-se o valor de 4,2 V, quando se mantém tensão constante até o final da carga. Ao final da carga, a bateria se mantém em repouso (corrente

nula) por um tempo até que se dê início à uma corrente de descarga constante de 825 mA por uma hora. Ao final da simulação, a bateria é mantida novamente em repouso.

Os resultados de estimação dos parâmetros R_o , R e C e também da tensão v_{rc} são obtidos apenas na situação em que o SOC parte com valor inicial nulo ($SOC_o = 0$), situação que abrange a estimação em toda a faixa do SOC (de 0 a 1). Esta medida foi adotada para condensar os resultados, uma vez que os parâmetros R_o , R e C têm pouca influência sobre a estimação de carga, conforme dito na seção 5.2. O foco dos resultados recai sobre a estimação do próprio SOC e dos parâmetros β_1 e β_o , quando for o caso, além da estimação da tensão v_t , que traz importantes informações. Todos os ruídos considerados possuem média nula (não polarizados).

Ao final deste capítulo, serão comentadas as considerações sobre os três métodos de estimação do SOC. A implementação dos algoritmos utilizados neste capítulo encontra-se no Anexo D.

5.7.1 Simulação de Estimação do SOC com EKF

O EKF se utiliza do modelo de segunda ordem da bateria, apresentado na equação (5.6), para realizar a estimação do SOC. Neste modelo, as equações não lineares do sistema são necessárias para o cálculo recursivo dos ganhos de estimação.

No caso do EKF e mesmo do KF, alguns parâmetros precisam ser ajustados, como por exemplo, as matrizes Q e R , conforme já abordado no capítulo 3. O ajuste dos parâmetros foi realizado a partir de tentativa e erro, chegando-se então ao resultado final de estimação. A Tabela 6 mostra os parâmetros utilizados no projeto do estimador com EKF.

A partir da Tabela 6 e da Figura 45, nota-se que os parâmetros estimados foram mantidos constantes para o modelo de estimação durante toda a simulação, o que é uma consideração válida, já que estes não possuem grande impacto sobre a estimação de carga.

Inicialmente considerando-se o sistema com condição inicial nula de carga, tem-se na Figura 45 a variação dos parâmetros obtidos através do simulador de bateria e os parâmetros do modelo de

Tabela 6 – Parâmetros de simulação do EKF

Descrição	Variável	Valor
Período de amostragem	T_a	1 s
Capacidade da bateria	Q_{max}	2750 mAh
Resistência série	R_o	0,09 Ω
Resistência do par RC	R	0,085 Ω
Capacitância do par RC	C	300 F
Matriz de covariância inicial	$P_{kf}(k=0)$	$diag(10^4 \ 1)$
Matriz de ponderação Q	Q_{kf}	$diag(10^{-4} \ 10^{-5})$
Matriz de ponderação R	R_{kf}	1
Variância do ruído de corrente	σ_i^2	0,0013 A^2
Variância do ruído de tensão	σ_v^2	0,005 V^2

Fonte: produção do próprio autor, 2016

estimação fixos ao longo de toda simulação.

Já na Figura 46 tem-se a constante de tempo $\hat{RC}$ frente a constante de tempo encontrada pelo simulador. A tensão v_{rc} e sua estimação a partir dos parâmetros RC também são mostradas. Nota-se que há um pequeno desvio entre a tensão v_{rc} simulada e a tensão $\hat{v}_{rc}$ pelo EKF, o que é natural, já que os parâmetros do modelo foram mantidos fixos e são diferentes dos parâmetros reais simulados.

A Figura 47 traz as tensões v_t e v_{oc} , tanto simuladas, quanto medidas. Aqui, percebe-se que o EKF é capaz de estimar a tensão v_t e a tensão v_{oc} muito bem, apresentando erros pequenos de estimação. Ainda, destaca-se que esta simulação foi obtida considerando-se ruídos de medição elevados na corrente i_b e na tensão v_t que alimentam o EKF com variâncias descritas na Tabela 6, o que não afetou a qualidade da estimação.

Por fim, tem-se a estimação do SOC considerando seu valor inicial nulo na Figura 48. Nota-se que a estimação do SOC segue perfeitamente a curva de carga real, e o erro de estimação fica limitado em valores menores que 0,03.

Considerando-se agora um $SOC_o = 0,35$ na bateria, tem-se as formas de onda de tensão v_t e v_{oc} e suas estimações $\hat{v}_t$ e $\hat{v}_{oc}$ na

Figura 49. Novamente, o sistema foi capaz de rastrear ambas as tensões, convergindo as suas estimações para os valores reais simulados.

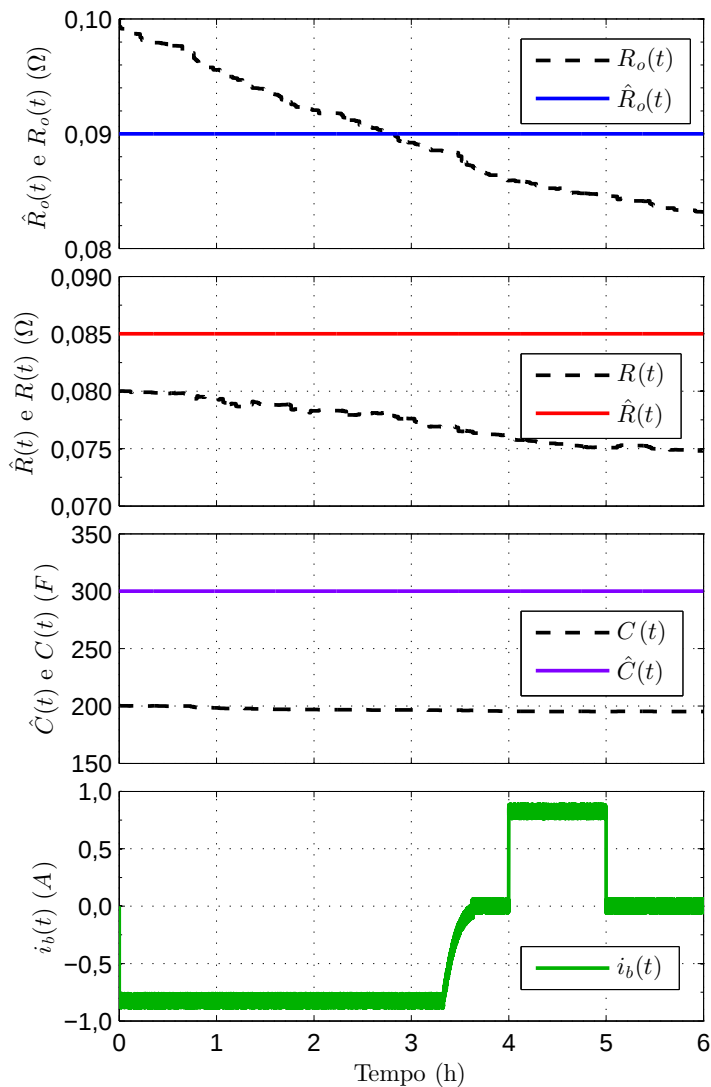
As formas de onda de estimação do SOC são mostradas na Figura 50 para o caso de $SOC_o = 0,35$. Nota-se que o $\hat{SOC}$ converge rapidamente para o valor real simulado, apresentando erros menores que 0,03 após cerca de 10 minutos.

Por fim, a Figura 51 mostra os valores de v_t e v_{oc} , juntamente com suas estimações, para um caso onde o $SOC_o = 0,7$. Nota-se que ainda assim, o sistema não apresenta maiores problemas em realizar as estimações.

A mesma conclusão pode ser observada na Figura 52 acerca da estimação do SOC, que apesar de apresentar um erro inicial de 0,7, converge rapidamente para o valor real simulado, apresentando erros também menores que 0,03.

Tendo sido apresentados os resultados de simulação do EKF como estimador do SOC, passa-se a seção 5.7.2, onde são apresentados os resultados experimentais de estimação obtidos.

Figura 45 – Parâmetros R_o (azul), R (vermelho) e C (roxo) estimados mantidos constantes para estimação do SOC com EKF ($SOC_o = 0$), juntamente com a corrente i_b (verde)



Fonte: produção do próprio autor, 2016

Figura 46 – Formas de onda da tensão $\hat{v}_{rc}$ (azul) e de $\hat{RC}$ (vermelho), juntamente da corrente i_b (verde), para o caso do EKF ($SOC_o = 0$)

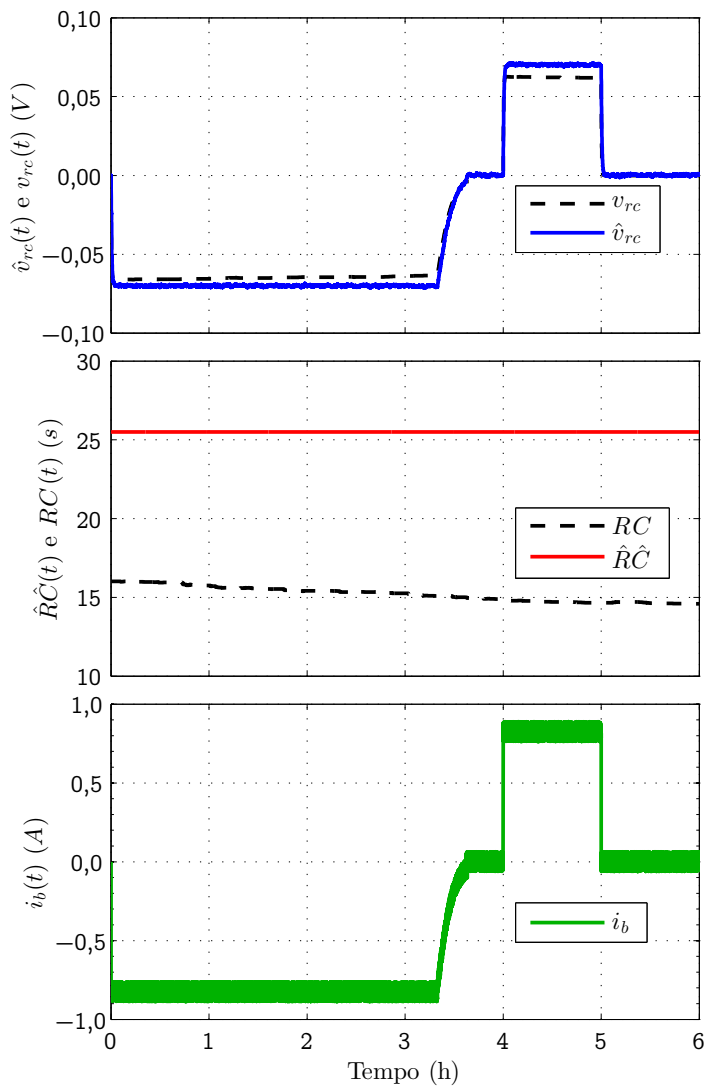
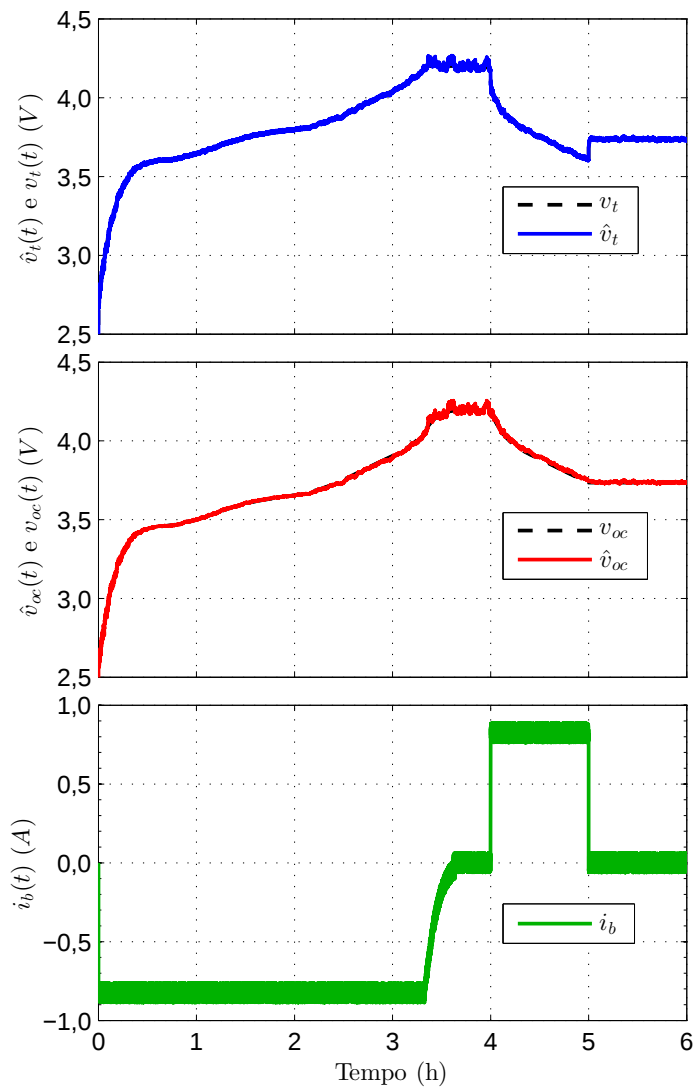
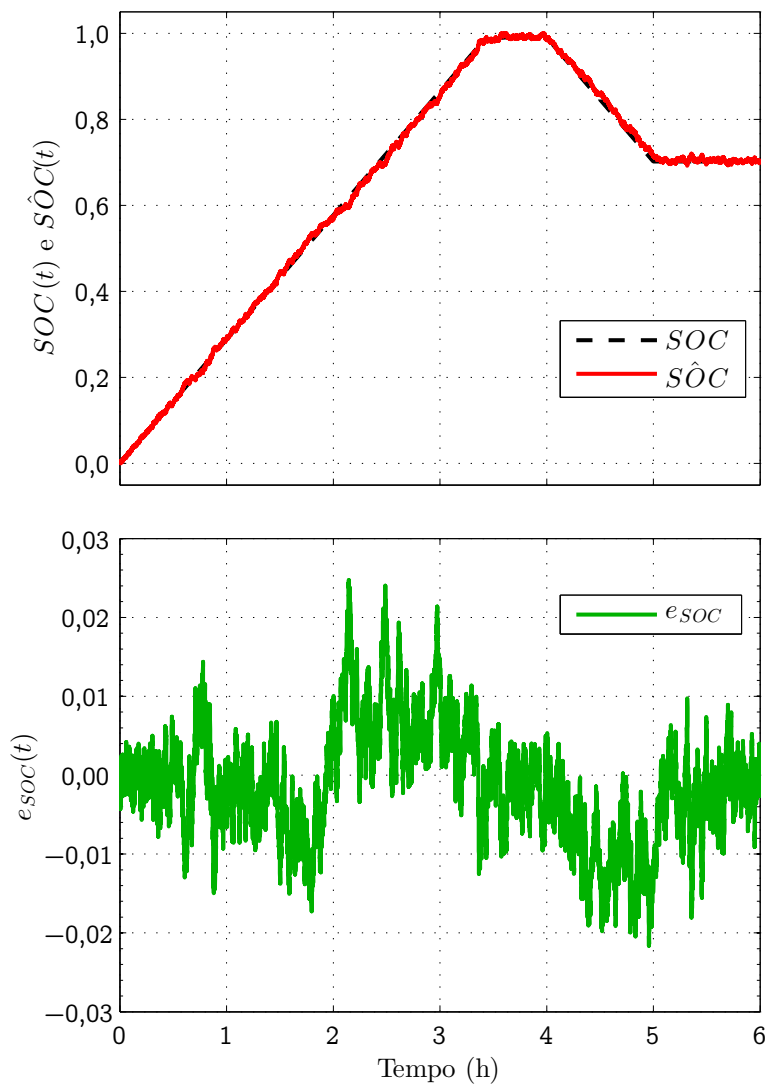


Figura 47 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo EKF, juntamente com a corrente i_b (verde) ($SOC_o = 0$)



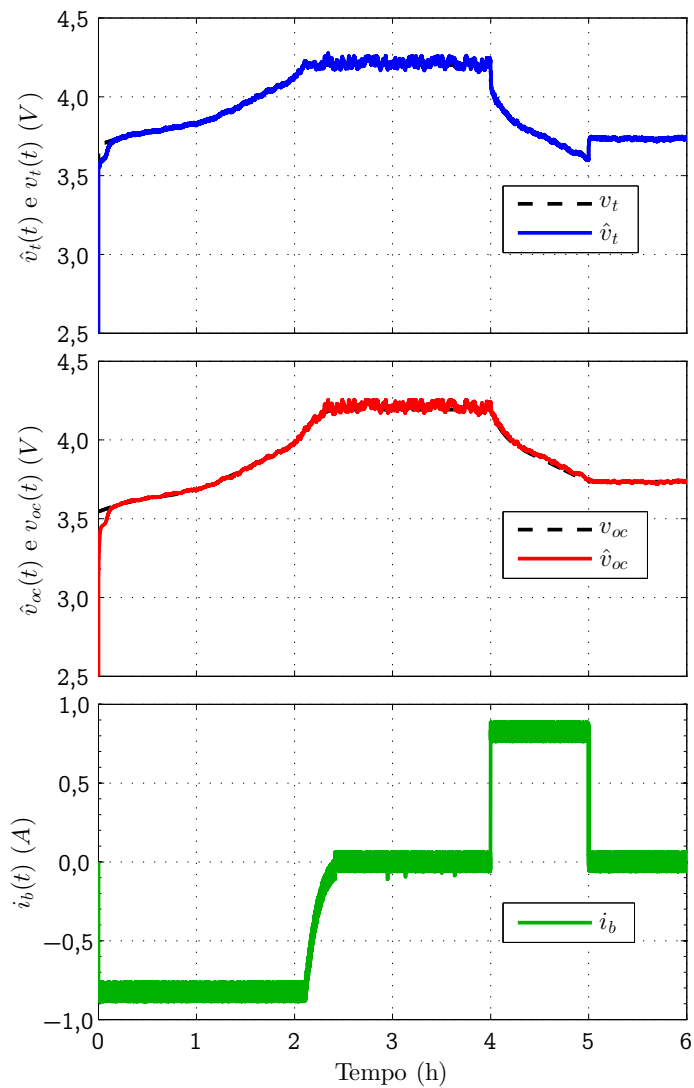
Fonte: produção do próprio autor, 2016

Figura 48 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) para o caso do EKF ($SOC_o = 0$)



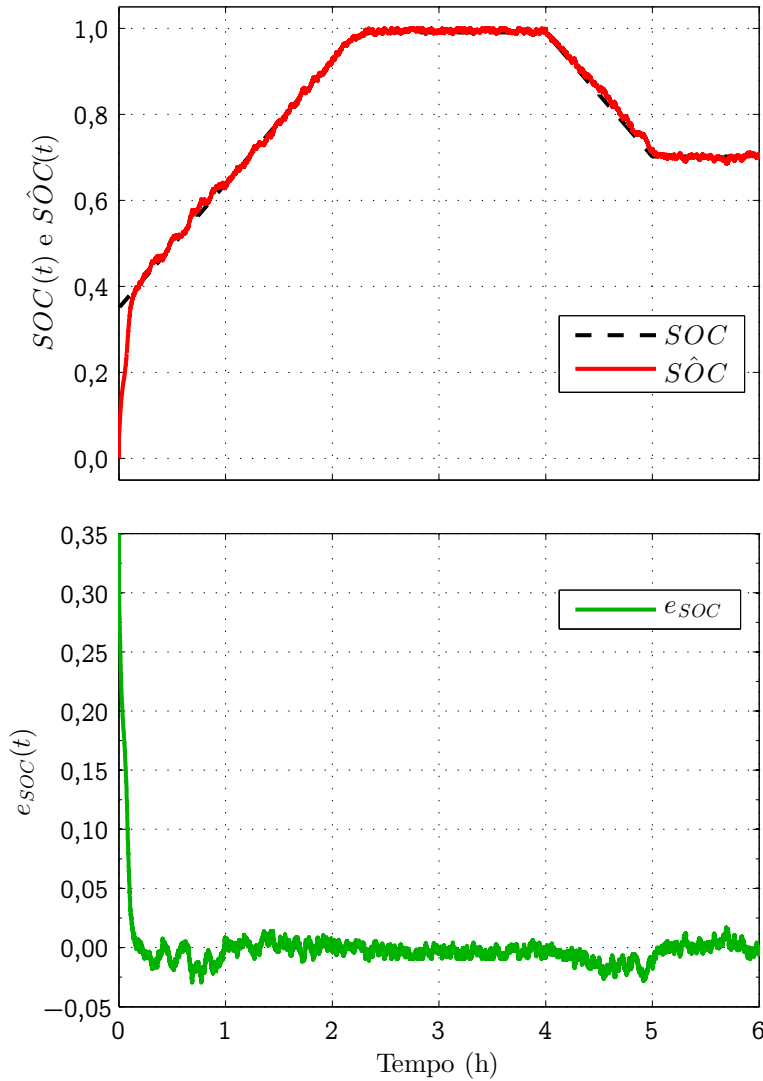
Fonte: produção do próprio autor, 2016

Figura 49 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo EKF, com a corrente i_b (verde) ($SOC_o = 0,35$)



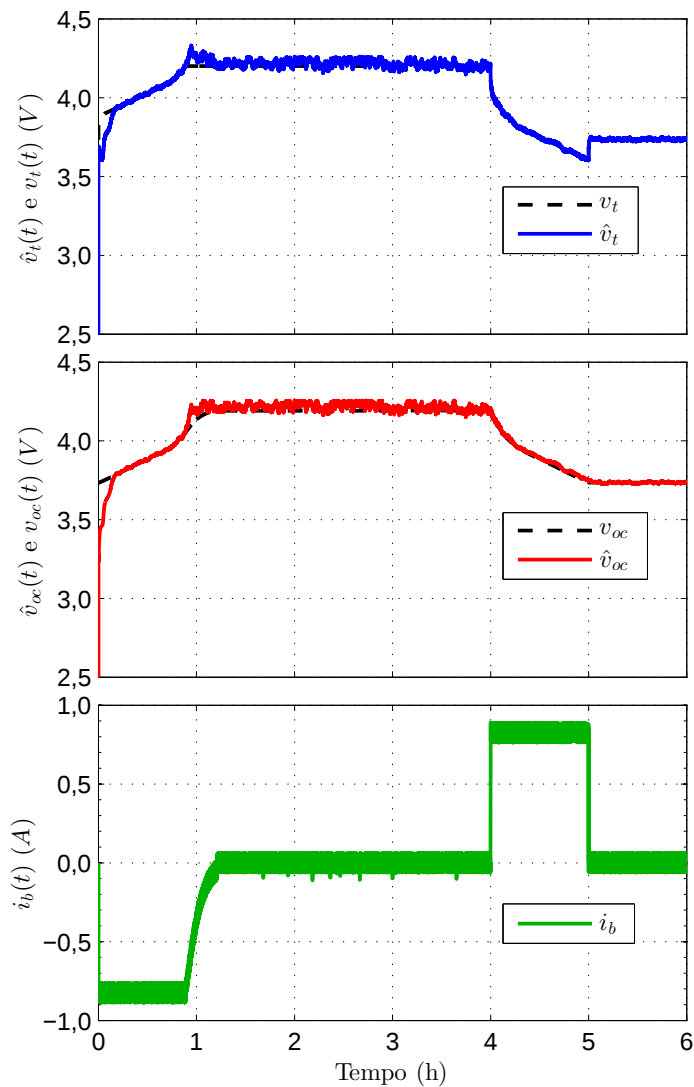
Fonte: produção do próprio autor, 2016

Figura 50 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) para o caso do EKF ($SOC_o = 0,35$)



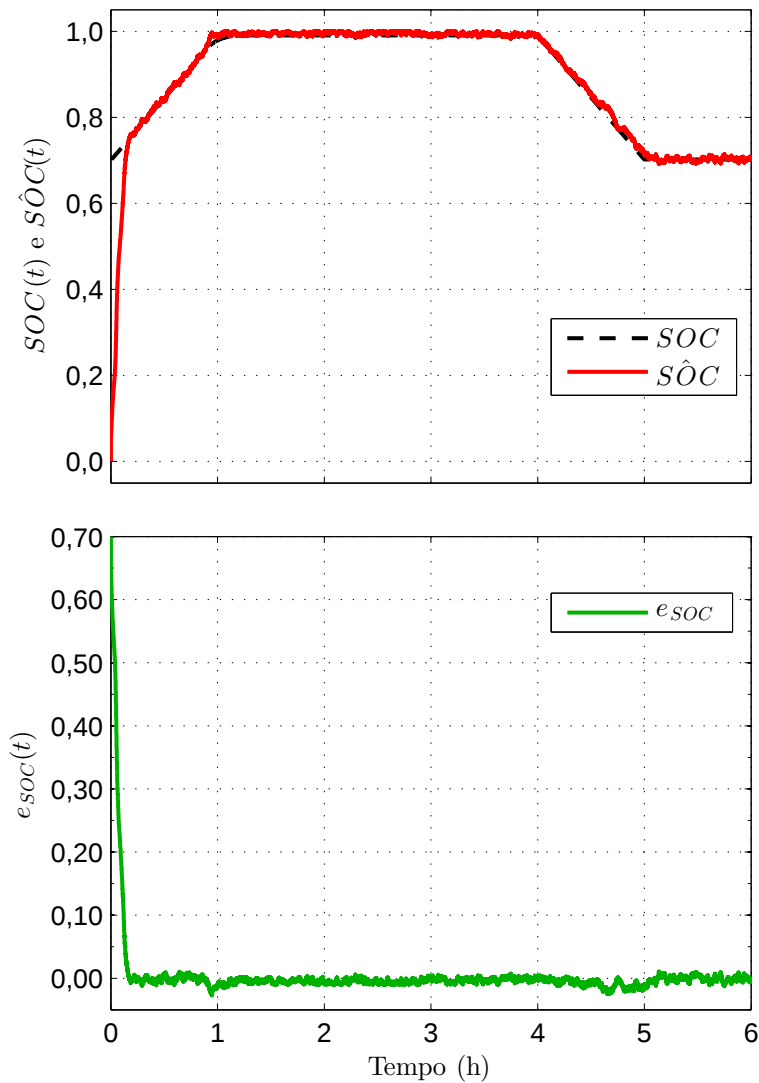
Fonte: produção do próprio autor, 2016

Figura 51 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo EKF, juntamente com a corrente i_b (verde) ($SOC_o = 0,7$)



Fonte: produção do próprio autor, 2016

Figura 52 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) para o caso do EKF ($SOC_o = 0,7$)



Fonte: produção do próprio autor, 2016

5.7.2 Resultados experimentais de estimação do SOC com EKF

Os resultados experimentais de estimação do SOC com EKF foram obtidos considerando-se a carga constante em três níveis diferentes, sendo estes novamente 0, 0,35 e 0,7. Ainda, para que pudesse ser feita uma comparação da estimação obtida experimentalmente com a estimação obtida através de simulação, são mostrados os resultados de simulação obtidos sob as mesmas condições do teste experimental.

A Figura 53 mostra os gráficos de estimação da tensão $\hat{v}_t$ e do $\hat{SOC}$ para o caso de uma bateria com carga próxima de zero. Nota-se que o valor de tensão $\hat{v}_t$ estimada converge para um valor de 2,864 V, enquanto o $\hat{SOC}$ estimado converge para um valor de 0,017. Ainda nesta mesma figura, podem ser observados diversos valores de medição e calculados pelo DSC. Nota-se que a tensão terminal da bateria (dada pela variável vb) medida é de 2,863 V, valor este bem próximo da tensão estimada. Ainda, nota-se que a tensão de circuito aberto estimada (dada pela variável Voc_{ekf}) tem um valor de 2,857 V, sendo este bem próximo da tensão terminal da bateria, o que é esperado, já que a bateria encontra-se em repouso.

Na Figura 54 são mostrados os resultados experimentais obtidos sob um SOC constante de 0,35. Nota-se que, novamente, a tensão estimada $\hat{v}_t$ (3,586 V) se aproxima da tensão medida v_t (3,587 V). Ainda, a estimação do $\hat{SOC}$ (0,343) também se mostrou convergindo para o valor real de 0,35. Novamente, vê-se que a tensão $\hat{v}_{oc}$ também converge para o valor lido de tensão terminal, o que condiz com a situação da bateria em repouso.

A título de comparação, foi realizada uma simulação do caso em que o SOC é mantido constante em 0,35 enquanto o algoritmo inicia a estimação a partir de condições iniciais nulas, a qual é mostrada na Figura 55. Nota-se que os resultados de simulação são de fato muito próximos dos observados na prática.

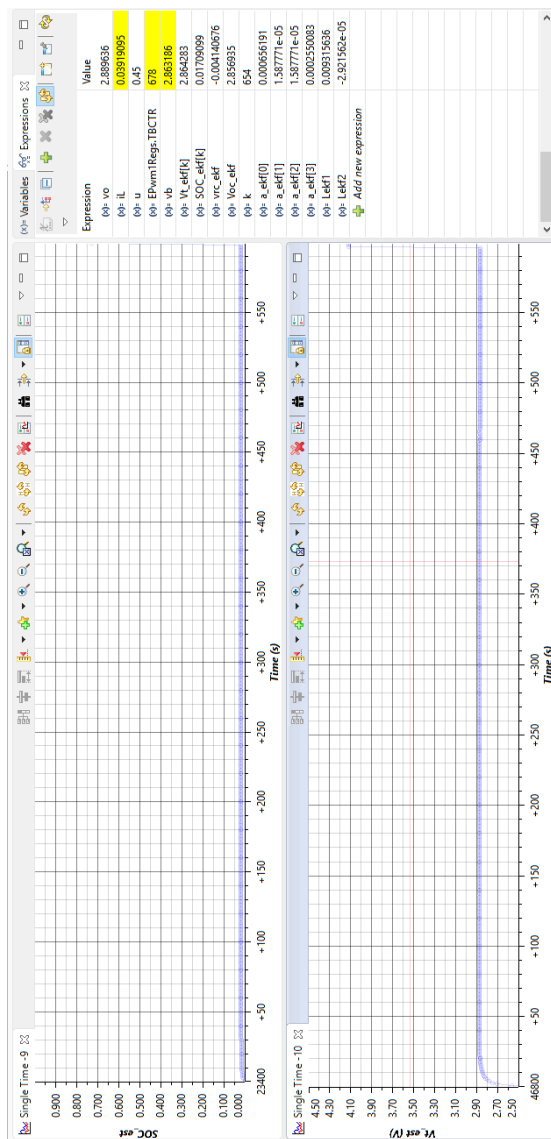
Finalizando os resultados de estimação experimentais com o EKF, tem-se a situação em que $SOC_o = 0,7$ na Figura 56, na qual a estimação do $\hat{SOC}$ convergiu para um valor de 0,729, a tensão $\hat{v}_t$ convergiu para 4,121 V, tendo-se um valor medido também de

4,121 V, enquanto que a tensão $\hat{v}_{oc}$ apresentou 4,113 V, novamente estando próxima do valor de tensão terminal.

Em todos os casos, a tensão estimada $\hat{v}_{rc}$ convergiu para um valor próximo de zero, o que satisfaz a condição de repouso da bateria. Além disso, as variáveis a_{ef} , que representam os elementos da matriz de covariância do sistema, convergiram para valores muito pequenos em todos os casos (próximo de zero) o que era esperado já que as medições apresentam baixos índices de ruído.

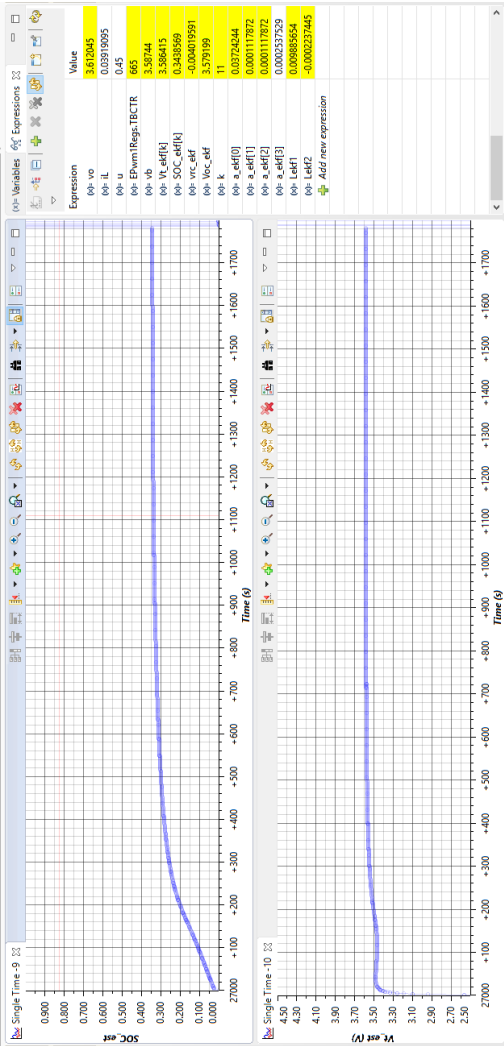
Apresentados os resultados experimentais do EKF implementado, segue-se então para os resultados de simulação da metodologia de estimação com KF adaptativo.

Figura 53 – Formas de onda experimentais do SOC (gráfico superior) e de v_t (gráfico inferior) estimados pelo EKF ($SOC \approx 0$)



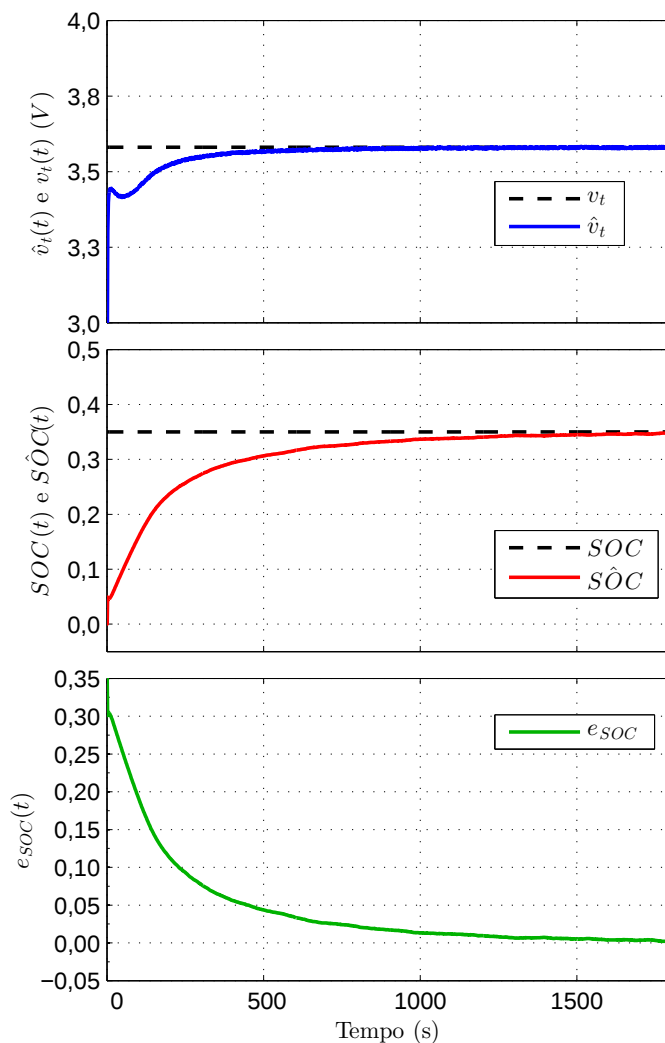
Fonte: produção do próprio autor, 2016

Figura 54 – Formas de onda experimentais do SOC (gráfico superior) e de v_t (gráfico inferior) estimados pelo EKF ($SOC = 0,35$)



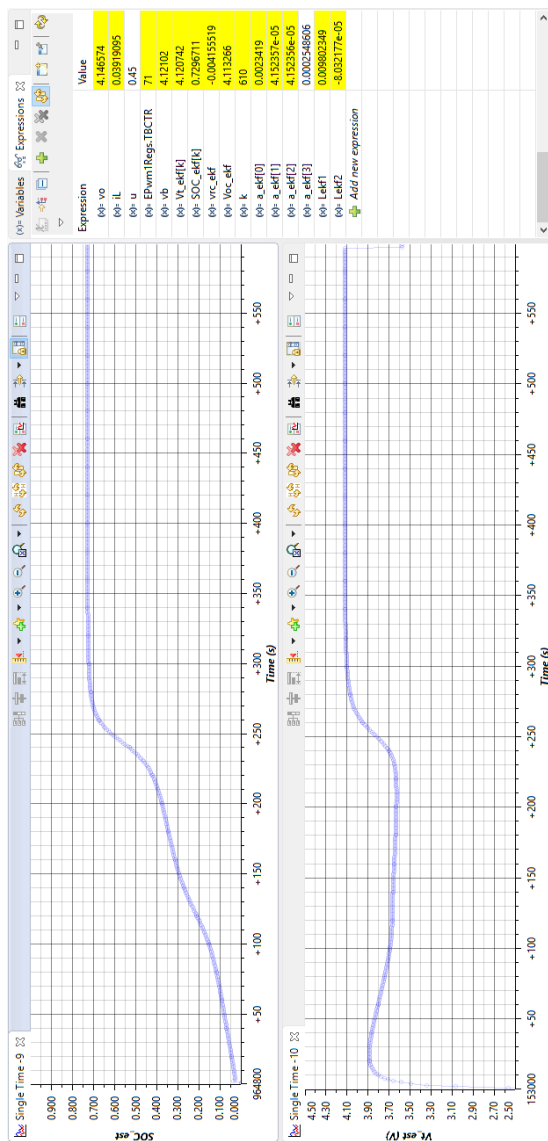
Fonte: produção do próprio autor, 2016

Figura 55 – Formas de onda simuladas do SOC (azul) e de v_t (vermelho) estimados por um EKF, reproduzindo-se as mesmas condições do teste experimental de estimação com EKF, juntamente com o erro de estimação do SOC (verde) ($SOC = 0,35$)



Fonte: produção do próprio autor, 2016

Figura 56 – Formas de onda experimentais do SOC (gráfico superior) e de v_t (gráfico inferior) estimados pelo EKF ($SOC = 0, 7$)



Fonte: produção do próprio autor, 2016

5.7.3 Simulação de Estimação do SOC com KF Adaptativo

A seguir, serão mostrados os resultados de simulação obtidos com a utilização de um KF na estimação do SOC, juntamente com a identificação dos parâmetros R_o , R e C através do algoritmo PSO.

A Tabela 7 traz os ajustes nos parâmetros que foram utilizados na obtenção destes resultados. Salienta-se que, como os parâmetros $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ não são mais constantes, foram indicados apenas os seus valores iniciais utilizados.

Tabela 7 – Parâmetros de simulação do KF adaptativo

Descrição	Variável	Valor
Período de amostragem	T_a	1 s
Capacidade da bateria	Q_{max}	2750 mAh
Resistência série inicial	$R_o(k=0)$	0,07 Ω
Resistência do par RC inicial	$R(k=0)$	0,08 Ω
Capacitância do par RC inicial	$C(k=0)$	250 F
Matriz de covariância inicial	$P_{kf}(k=0)$	$diag(10^4 \ 1)$
Matriz de ponderação Q	Q_{kf}	$diag(10^{-5} \ 10^{-5})$
Matriz de ponderação R	R_{kf}	1
Variância do ruído de corrente	σ_i^2	0,0013 A^2
Variância do ruído de tensão	σ_v^2	0,005 V^2

Fonte: produção do próprio autor, 2016

A Figura 57, traz os valores de estimação de $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ juntamente com os parâmetros reais obtidos pelo simulador, para a condição onde $SOC_o = 0$. A identificação destes parâmetros do modelo ocorre a cada 60 s e é filtrada por um LPF, para evitar variações abruptas que degradam o desempenho do KF. Os resultados mostram que os parâmetros identificados mantêm um comportamento quase constante, porém diferente daquele obtido através do simulador, fato que será melhor discutido adiante, mas que não traz problemas à estimação do SOC, conforme já discutido na seção 5.7.2 de resultados do EKF.

A partir dos parâmetros identificados, foi obtido o valor da constante de tempo estimada do sistema e também o valor da tensão

dinâmica estimada $\hat{v}_{rc}$, os quais encontram-se na Figura 58. Novamente, nota-se que os resultados estimados diferem ligeiramente dos resultados reais simulados.

Agora, na Figura 59, são mostradas as formas de onda da tensão v_t e da tensão v_{oc} , juntamente com as suas estimações. Ao analisar a tensão $\hat{v}_t$, percebe-se que esta consegue seguir a tensão v_t sem maiores problemas, mesmo diante de ruídos de medição na corrente e na tensão, os quais possuem a mesma variância daqueles utilizados na simulação do EKF. Porém, com relação a tensão $\hat{v}_{oc}$, nota-se que esta sofre pequenos desvios em relação ao seu valor real, mas que não chegam a comprometer os resultados de estimação. Isto se deve à estimação do SOC, que, conforme será mostrado adiante, mostrou resultados com erros ligeiramente maiores que os obtidos pelo EKF.

Na Figura 60, tem-se os resultados de estimação do SOC. Percebe-se que a estimação neste caso possui desvios um pouco maiores que os obtidos pelo EKF, chegando a valores de erro próximos de 0,09, porém, ainda seguindo a tendência do valor real simulado do SOC.

Na sequência, são mostrados os resultados obtidos a partir da simulação com um erro inicial na estimação do SOC, utilizando-se $SOC_o = 0,35$. A Figura 61 mostra que, novamente, o estimador foi capaz de oferecer bons resultados de estimação das tensões v_t e v_{oc} . O mesmo pode ser dito a respeito da estimação do SOC, a qual é mostrada na Figura 62. Quando comparado ao EKF, esta estrutura apresentou uma convergência mais rápida, diminuindo rapidamente o valor de erro inicial, porém mostrando um comportamento mais agressivo de estimação, que acarretou em valores de erro próximos de 0,05, maiores que os apresentados pelo EKF. Outro fator que comprova que este estimador possui uma dinâmica mais rápida é a observação de um pequeno sobressinal na estimação do SOC e de v_t .

Para caso em que $SOC_o = 0,7$, foram obtidos resultados muito similares aos analisados na simulação em que $SOC_o = 0,35$, conforme visto nas Figuras 63 e 64. Todos os valores de estimação apresentam resultados satisfatórios, e a estimação do SOC converge para valores de erro menores que 0,05 em regime permanente, apresentando também um pequeno sobressinal, comprovando mais uma vez uma ação de controle mais agressiva por parte do estimador.

Na seção seguinte, serão abordados os resultados do estimador do SOC realizado a partir de um KF adaptativo com coestimação da curva $v_{oc} \times SOC$.

Figura 57 – Parâmetros R_o (azul), R (vermelho) e C (roxo) mantidos constantes para estimação com KF adaptativo, juntamente com a corrente i_b (verde) ($SOC_o = 0$)

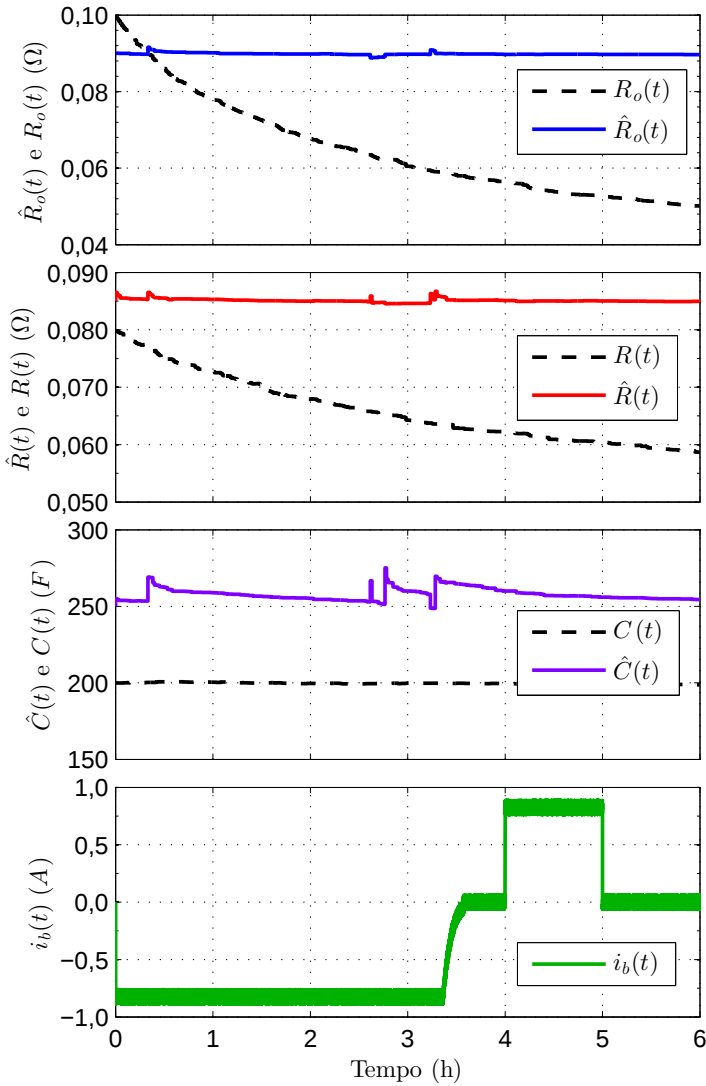
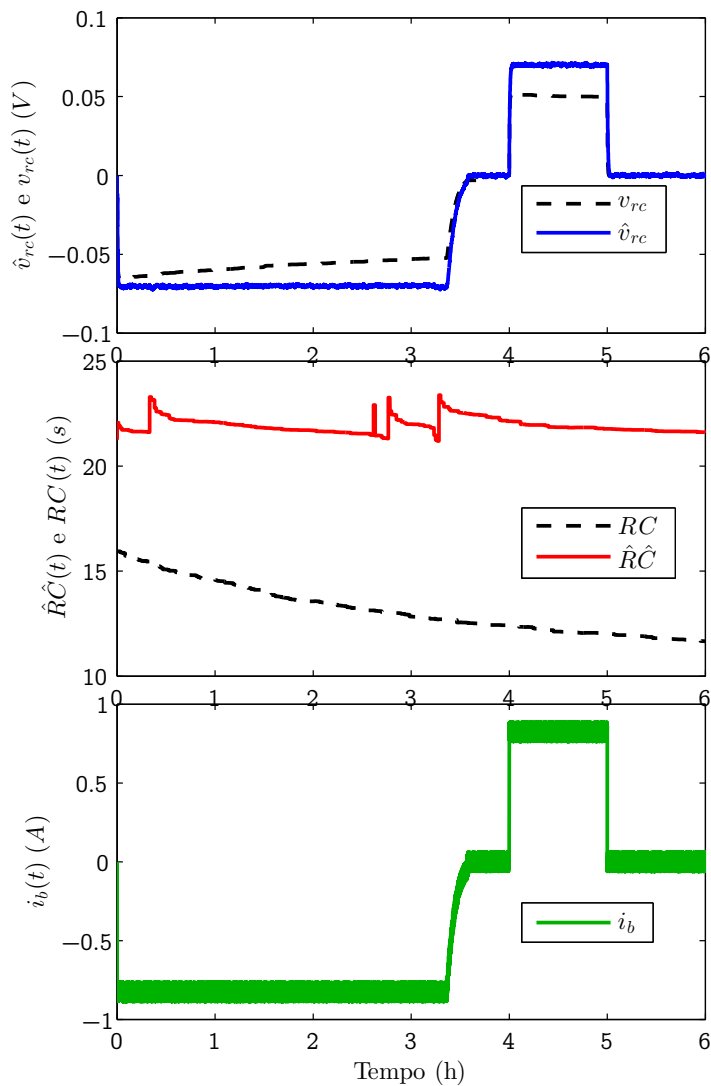


Figura 58 – Formas de onda da tensão $\hat{v}_{rc}$ (azul) e de $\hat{RC}$ (vermelho), juntamente com a corrente i_b (verde) para o caso do KF adaptativo ($SOC_o = 0$)



Fonte: produção do próprio autor, 2016

Figura 59 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) para o caso do KF adaptativo, juntamente com a corrente i_b (verde) ($SOC_o = 0$)

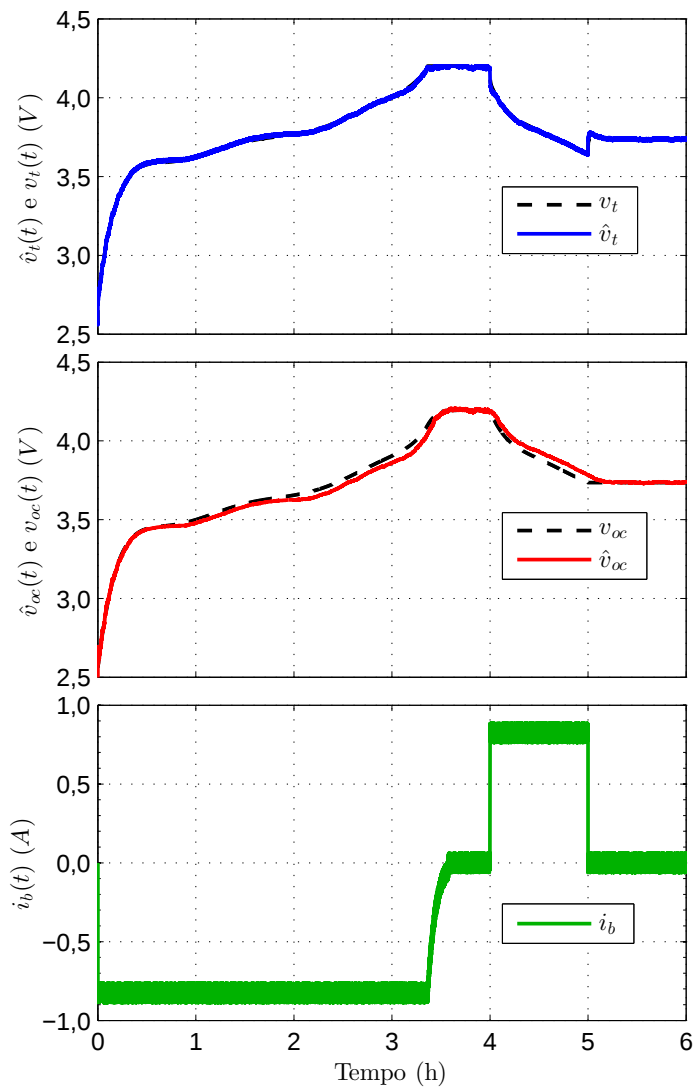
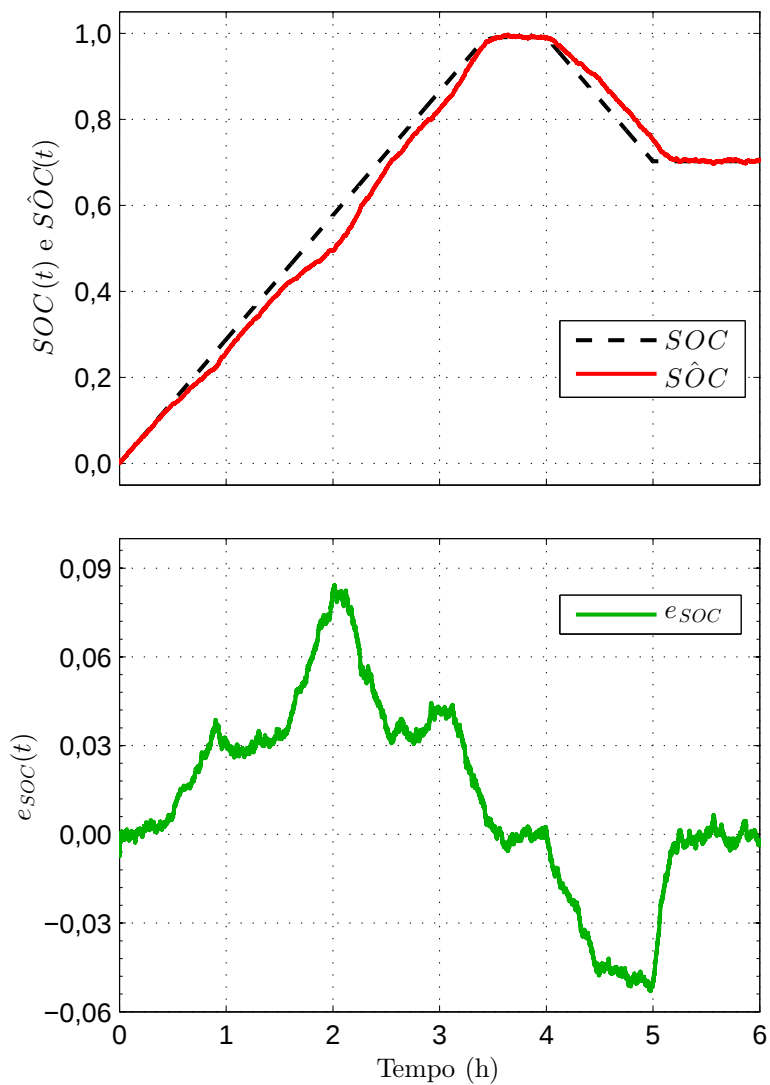


Figura 60 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) pelo KF adaptativo ($SOC_o = 0$)



Fonte: produção do próprio autor, 2016

Figura 61 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo KF adaptativo, juntamente com a corrente i_b (verde) ($SOC_o = 0,35$)

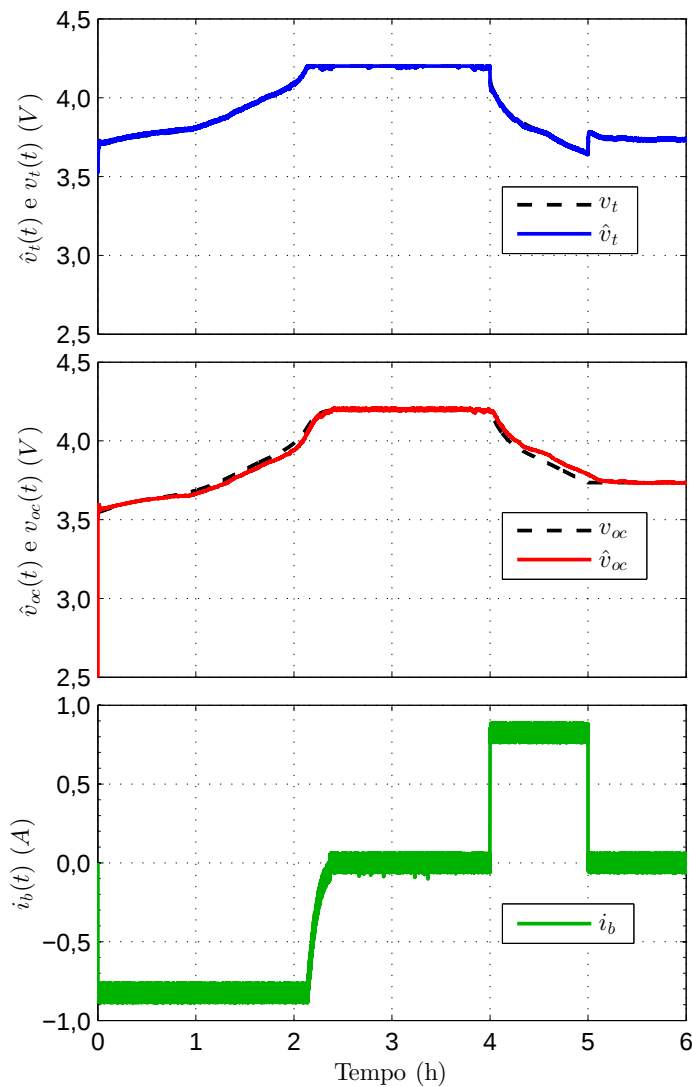
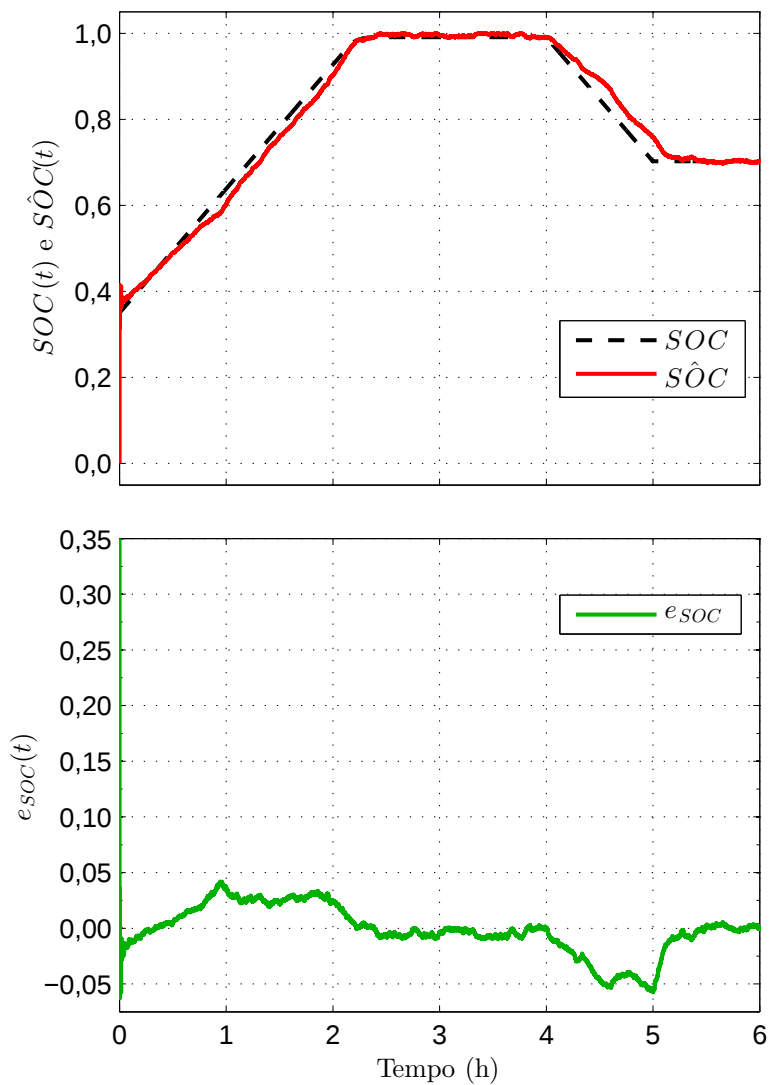


Figura 62 – Forma de onda do SOC estimado (vermelho) pelo KF adaptativo e do seu erro de estimação (verde) ($SOC_o = 0,35$)



Fonte: produção do próprio autor, 2016

Figura 63 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho), juntamente com a corrente i_b (verde), para o caso do KF adaptativo ($SOC_o = 0,7$)

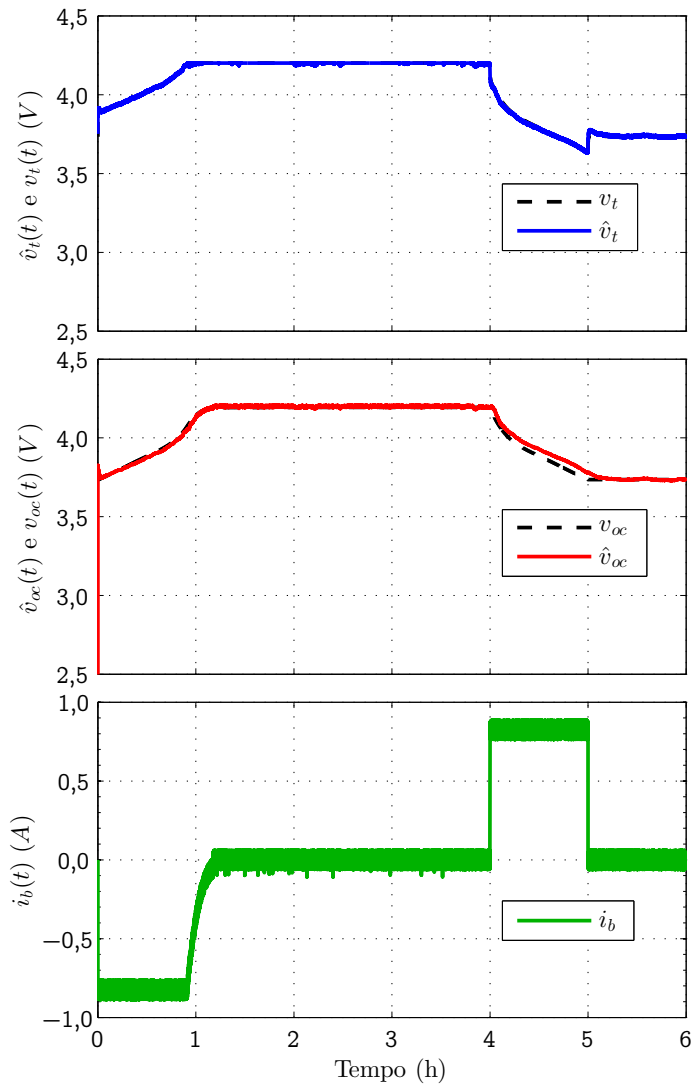
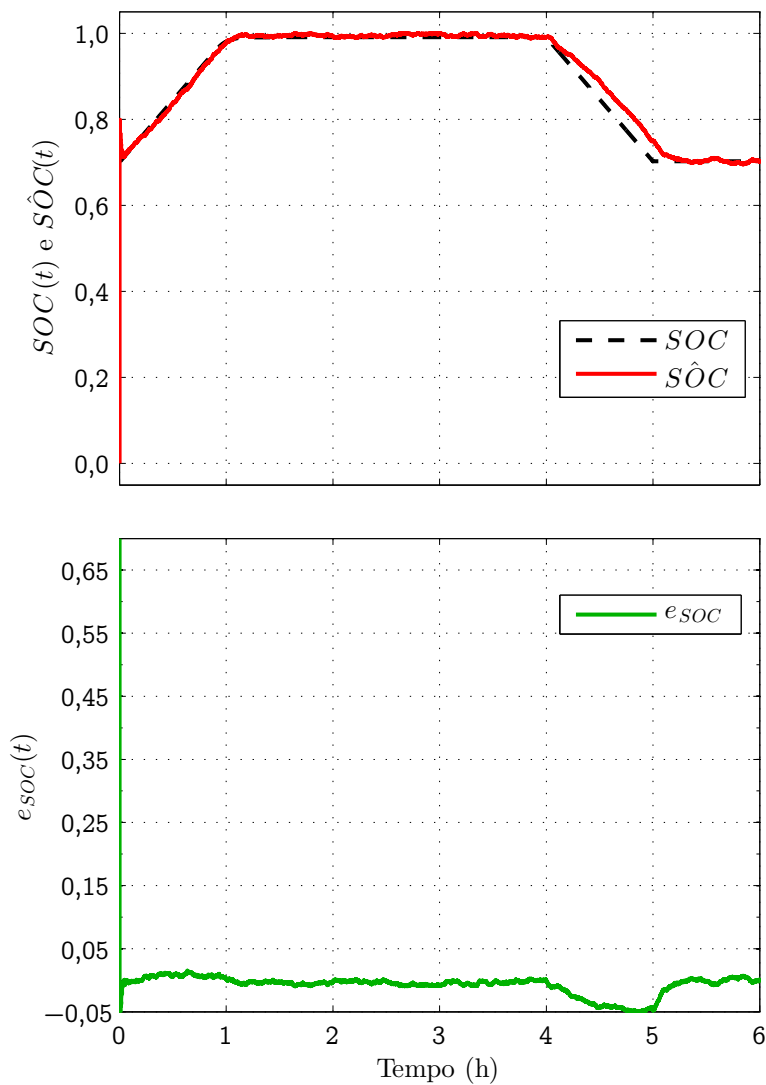


Figura 64 – Forma de onda do SOC estimado (vermelho) pelo KF adaptativo e do seu erro de estimação ($SOC_o = 0,7$)



Fonte: produção do próprio autor, 2016

5.7.4 Simulação de Estimação do SOC com KF adaptativo com coestimação da curva v_{oc} pela carga

Os resultados da estimação SOC e da curva $v_{oc} \times SOC$ serão abordados nesta seção. Os parâmetros utilizados nesta simulação são trazidos na Tabela 8.

Tabela 8 – Parâmetros de simulação do KF adaptativo com coestimação da curva $v_{oc} \times SOC$

Descrição	Variável	Valor
Período de amostragem	T_a	10 s
Capacidade da bateria	Q_{max}	2750 mAh
Resistência série inicial	$R_o(k=0)$	0,07 Ω
Resistência do par RC inicial	$R(k=0)$	0,08 Ω
Capacitância do par RC inicial	$C(k=0)$	300 F
Matriz de covariância inicial	$P_{kf}(k=0)$	$diag(10^3 \ 10 \ 10^5)$
Matriz de ponderação Q	Q_{kf}	$diag(1 \ 10^{-3} \ 10)$
Matriz de ponderação R	R_{kf}	10
Variância do ruído de corrente	σ_i^2	0 A ²
Variância do ruído de tensão	σ_v^2	0 V ²

Fonte: produção do próprio autor, 2016

Neste caso, a identificação dos parâmetros $\hat{R}_o$, $\hat{R}$, $\hat{C}$ e $\hat{\beta}_1$ é realizada a cada período de amostragem, e não a cada 60 s, como nos resultados obtidos nas seções anteriores. Isto porque não foram obtidos resultados satisfatórios de estimação do parâmetro β_1 com o aumento do período de amostragem de identificação dos parâmetros, por motivos desconhecidos. Por este motivo, para que a simulação não se tornasse extensa, foi adotado um $T_a = 10$ s, o qual ainda traz resultados satisfatórios na estimação de β_1 . Ainda, a utilização de um LPF sobre todos os parâmetros identificados acabava por degradar a estimação de β_1 . Assim, optou-se não utilizar um LPF após a estimação de β_1 , enquanto que o mesmo é mantido na estimação dos demais parâmetros. Destaca-se ainda que os ruídos na tensão e na corrente foram desconsiderados nesta análise ($\sigma_i^2 = 0$, $\sigma_v^2 = 0$).

A Figura 65 traz a estimação dos parâmetros $\hat{R}_o$, $\hat{R}$ e $\hat{C}$, que apresenta característica bem mais ruidosa, devido ao fato de serem realizadas identificações no sistema com menor período de amostragem, porém estas estimações se mantêm dentro de faixas bem delimitadas, e ainda, não devem influenciar fortemente na estimação do SOC.

A variação dos parâmetros $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ reflete em uma maior variância do sinal $\hat{v}_{rc}$ e da constante de tempo estimada, conforme visto na Figura 66.

Agora, na Figura 67, tem-se os valores estimados de β_1 e β_o , além da curva estimada de v_{oc} . Primeiramente, com relação à estimação de β_1 , nota-se que o algoritmo PSO consegue identificar bem a tendência da variável ao longo de quase toda a simulação, exceto quando a tensão da bateria é mantida constante, levando o algoritmo a divergir completamente da tendência da variável a ser estimada. Destaca-se ainda que, apesar de $\hat{\beta}_1$ seguir bem a tendência de β_1 durante os períodos em que v_t não é constante, há a presença de alguns *spikes* durante a estimação.

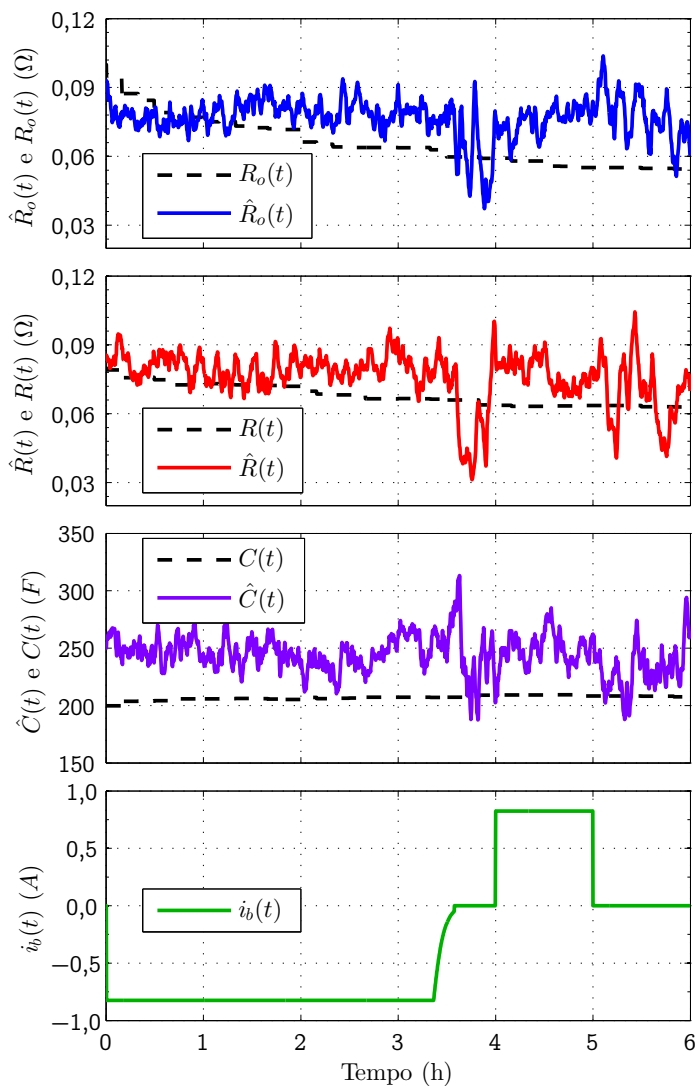
Com relação ao valor de β_o , tem-se uma divergência maior entre a curva real simulada e a curva estimada, estando o valor estimado $\hat{\beta}_o$ longe do real β_o na maioria do tempo de simulação. Este erro de estimação em β_o acarreta em um erro na estimação do SOC, que é muito sensível a este parâmetro, bem como ao parâmetro β_1 . Nota-se ainda nesta figura que a tensão estimada $\hat{v}_{oc}$ converge para a tensão v_{oc} em boa parte da simulação, mas apresenta desvios agressivos durante alguns instantes.

Com relação a estimação da tensão v_t , mostrada na Figura 68, percebe-se que seu valor também converge bem para o seu valor real em boa parte da simulação, assim como acontece com v_{oc} . Isto porque, como a tensão v_t medida é utilizada para realimentar o sistema, esta acaba sendo compensada de diversas formas para que o estimador convirja o valor estimado para o valor real medido. E neste caso, a forma encontrada pelo estimador para compensar o erro na estimação de β_o é variar o valor de $S\hat{O}C$ para obter a melhor representação de v_t possível, fato que é visto na Figura 68.

Nota-se na Figura 69 que o SOC estimado diverge completamente do valor real, apresentando erros na casa de até 0,8 na esti-

mação. A inicialização do algoritmo em condições iniciais de carga diferentes não muda este cenário, conforme observado nas Figuras 70 e 71. Nestes casos, nota-se que a estimação do SOC não converge para o seu valor real, mostrando valores de erros exacerbados. As Figuras 72 e 73 mostram a estimação dos parâmetros β_1 e β_o também para estes últimos dois casos, onde a condição inicial de carga da bateria é não nula. Assim como ocorreu para o caso onde $SOC_o = 0$, a estimação de β_1 consegue seguir bem a referência, porém, o valor de β_o estimado diverge do seu valor real, buscando compensar o erro de estimação no SOC, forçando assim a convergência do valor de tensão v_{oc} e também de v_t .

Figura 65 – Formas de onda estimadas dos parâmetros $\hat{R}_o$ (azul), $\hat{R}$ (vermelho) e $\hat{C}$ (roxo) no caso do KF Adaptativo com Co-estimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)



Fonte: produção do próprio autor, 2016

Figura 66 – Formas de onda da tensão $\hat{v}_{rc}$ (azul) e de $\hat{RC}$ (vermelho), juntamente com a corrente i_b (verde) no caso do KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)

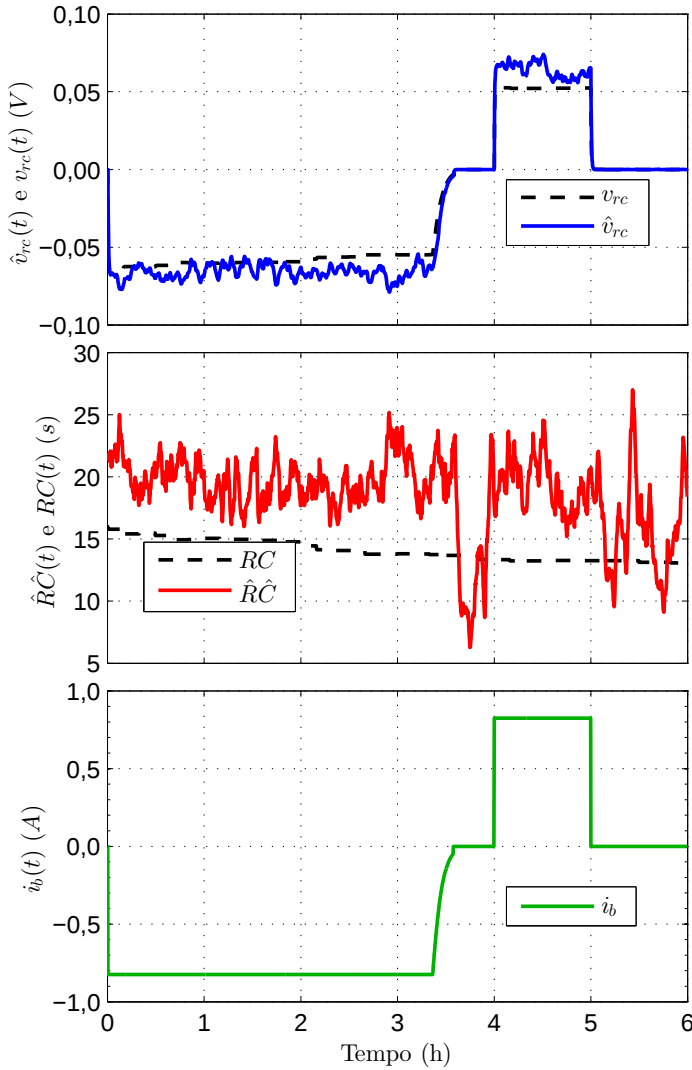
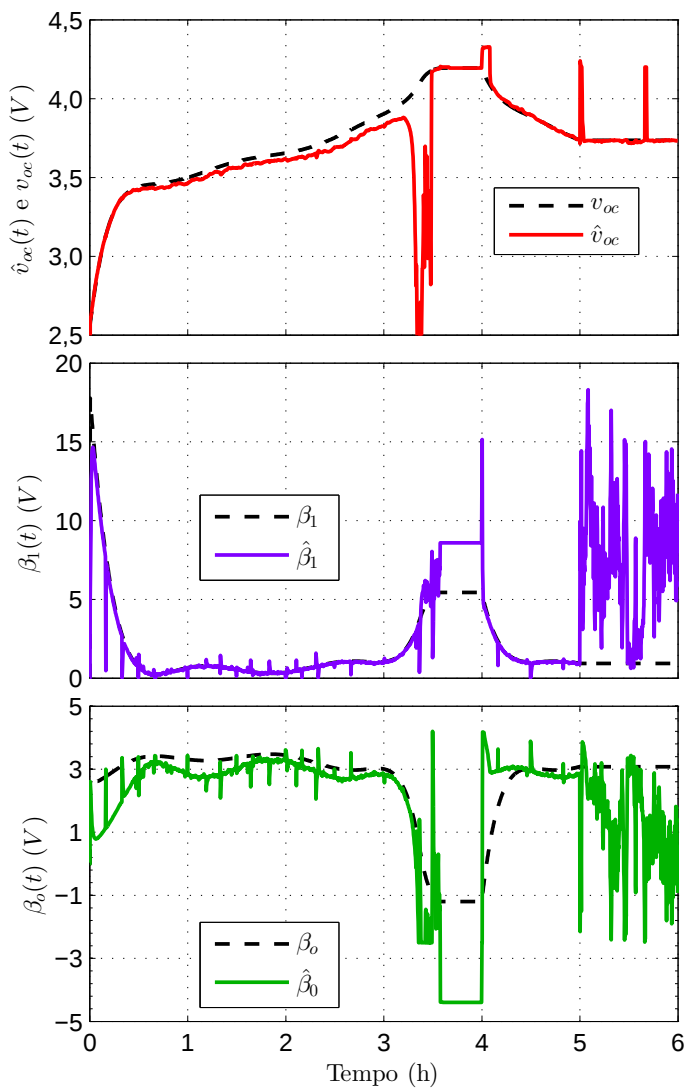


Figura 67 – Formas de onda de $\hat{\beta}_1$ (roxo), $\hat{\beta}_o$ (verde) e $\hat{v}_{oc}$ (vermelho) estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)



Fonte: produção do próprio autor, 2016

Figura 68 – Formas de onda de $\hat{v}_t$ (azul) e $\hat{v}_{oc}$ (vermelho) estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$, juntamente com a corrente i_b (verde) ($SOC_o = 0$)

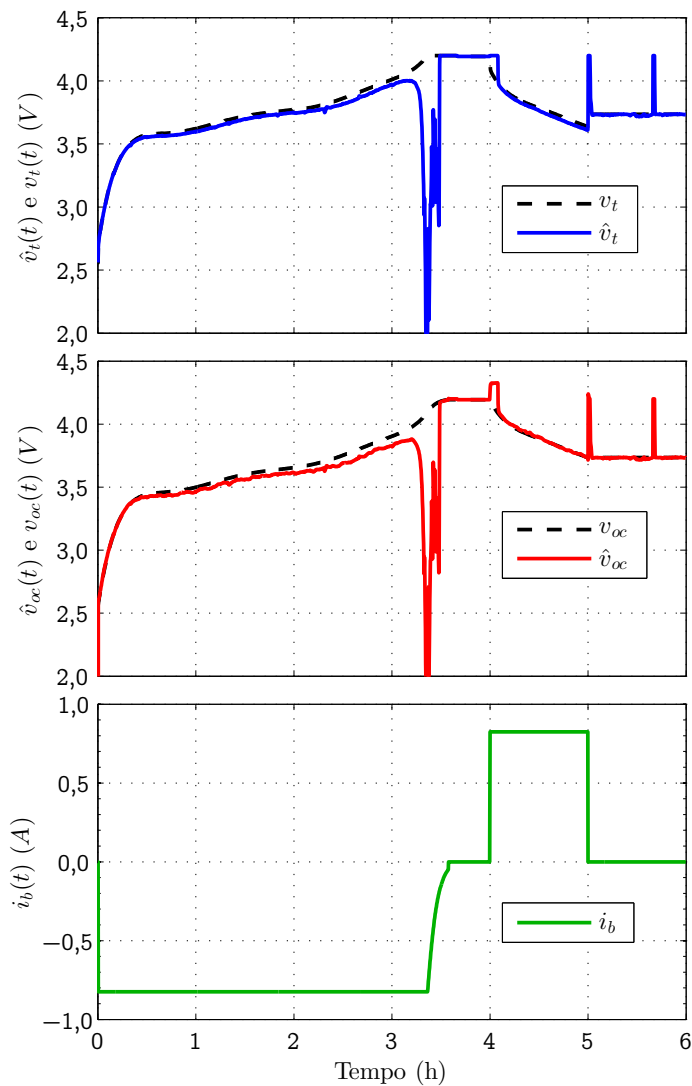
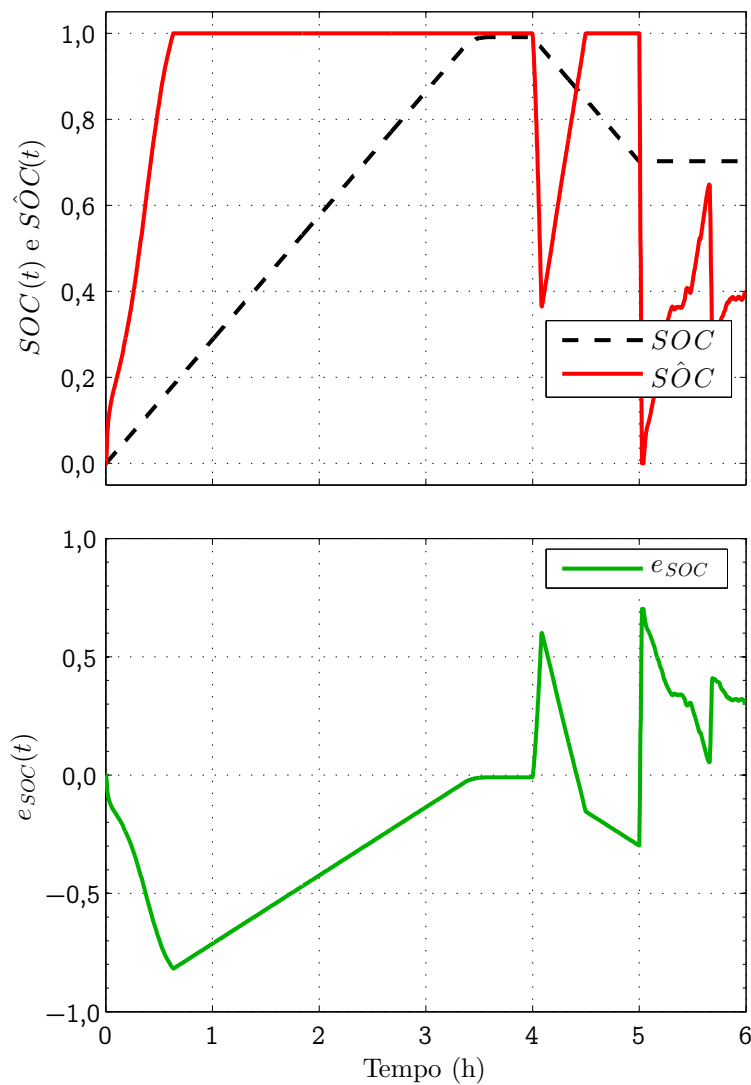


Figura 69 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) com o KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$)



Fonte: produção do próprio autor, 2016

Figura 70 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) com o KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0,35$)

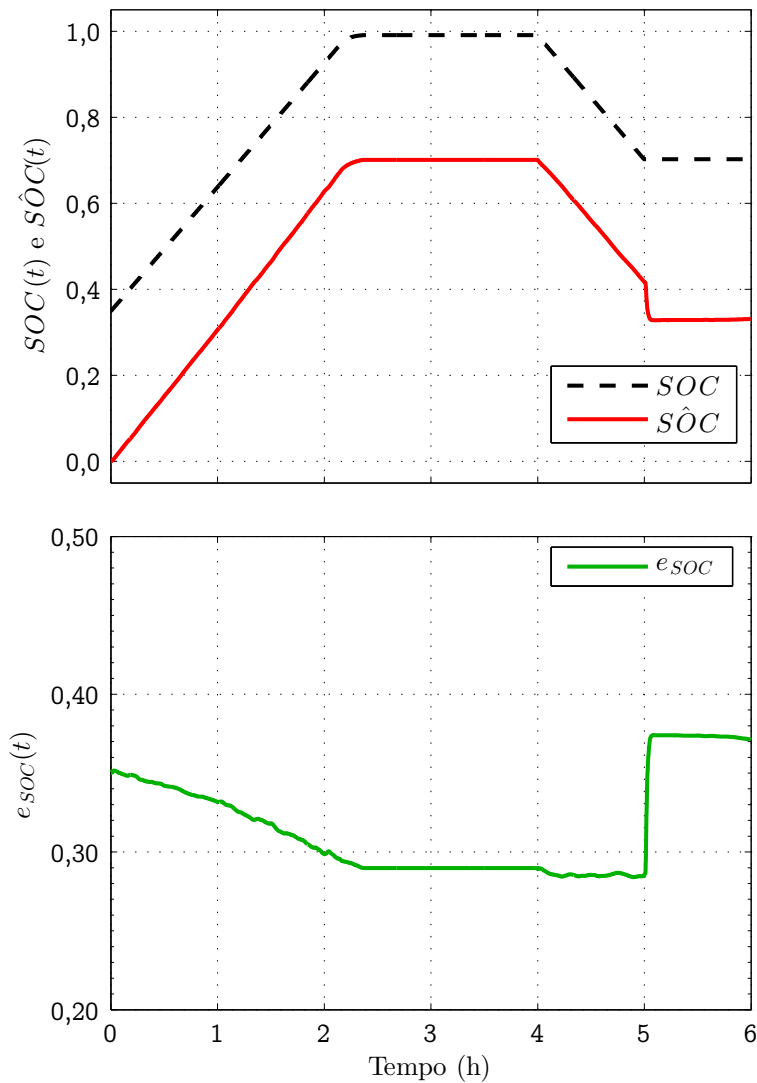
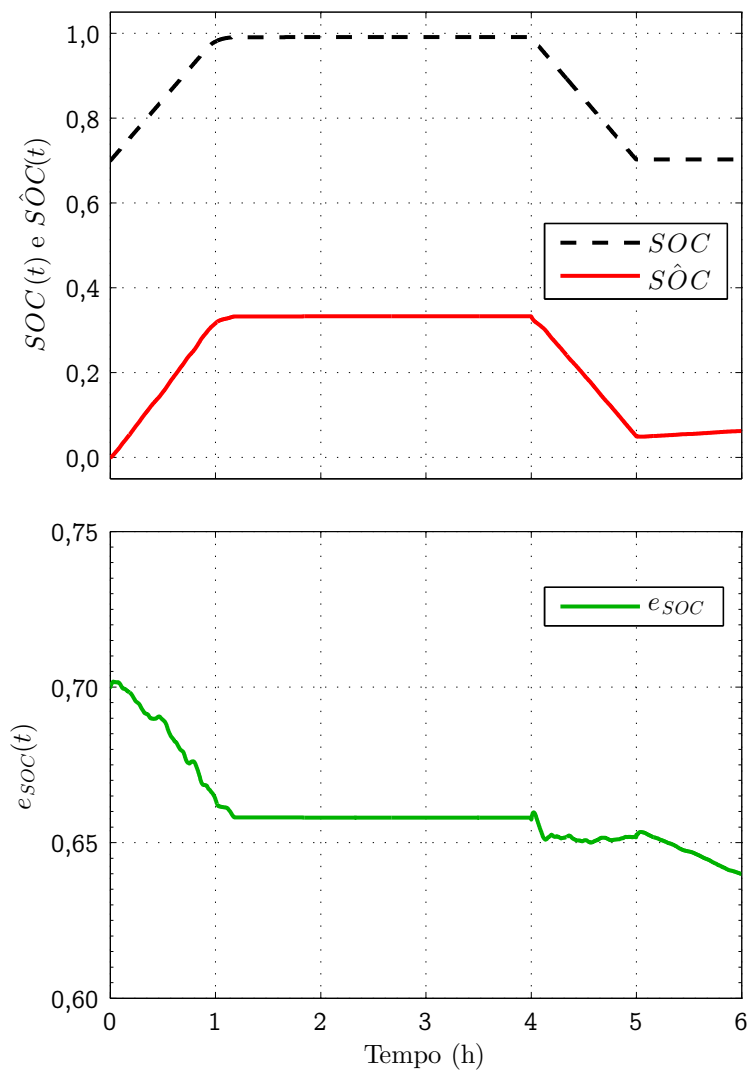


Figura 71 – Forma de onda do SOC estimado (vermelho) e do seu erro de estimação (verde) utilizando o KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0$) ($SOC_o = 0,7$)



Fonte: produção do próprio autor, 2016

Figura 72 – Formas de onda de $\hat{\beta}_1$ (roxo) e de $\hat{\beta}_o$ (verde), além da tensão $\hat{v}_{oc}$ (vermelho), estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0,35$)

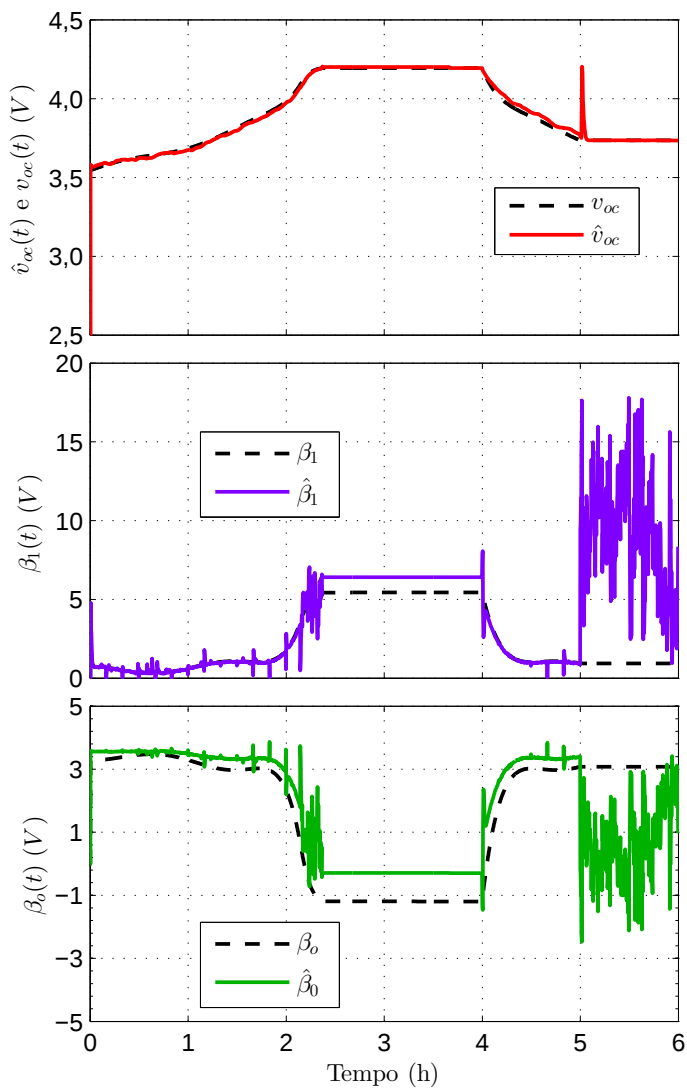
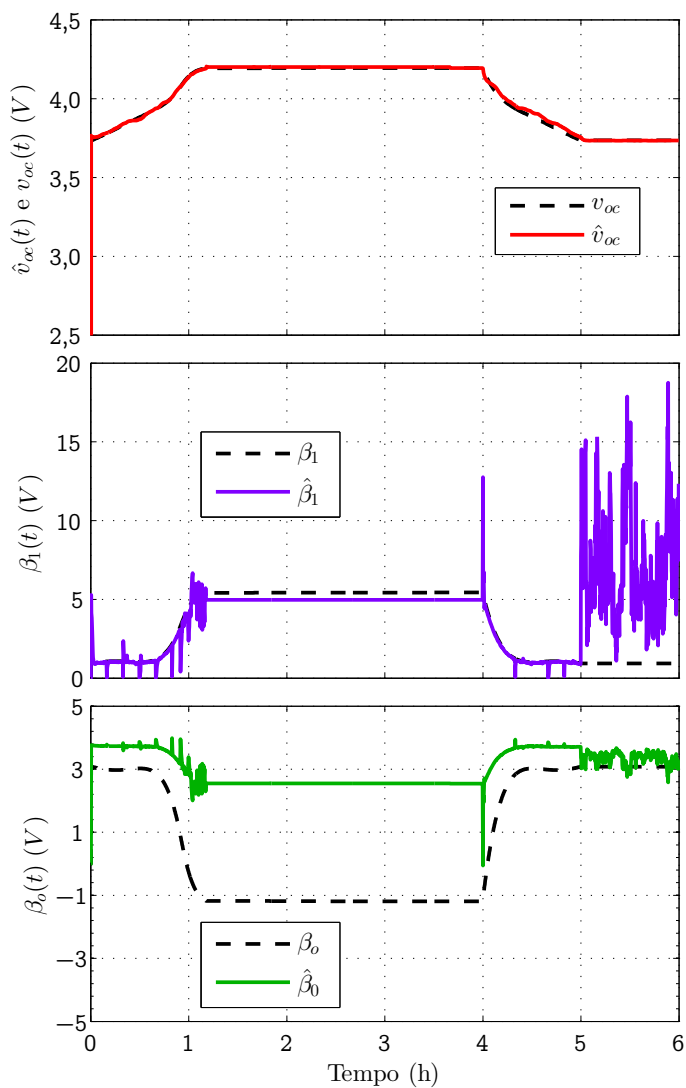


Figura 73 – Formas de onda de $\hat{\beta}_1$ (roxo) e de $\hat{\beta}_o$ (verde), além da tensão $\hat{v}_{oc}$ (vermelho), estimadas pelo KF Adaptativo com Coestimação da Curva $v_{oc} \times SOC$ ($SOC_o = 0,7$)



Fonte: produção do próprio autor, 2016

5.8 CONSIDERAÇÕES FINAIS SOBRE A ESTIMAÇÃO DE CARGA DE BATERIAS

A partir dos resultados aqui mostrados, serão feitas algumas considerações acerca dos métodos de estimação utilizados.

Com relação ao EKF, o qual já é um método bem consagrado e utilizado na literatura, não houve maiores problemas. Foi possível efetuar o ajuste do algoritmo de estimação com facilidade, e o mesmo se mostrou robusto mesmo diante dos parâmetros $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ errôneos, os quais foram mantidos constantes. Os erros nas estimações de tensão v_t , v_{oc} e do SOC se mostraram desprezíveis mesmo diante de ruídos na corrente e na tensão utilizadas pelo estimador.

Nos casos em que o erro de estimação inicial é diferente de zero, o algoritmo EKF também foi capaz de obter bons resultados de estimação em regime permanente, com erros desprezíveis, porém, levando um tempo maior para convergir que o algoritmo do KF adaptativo. Vale destacar que o tempo que o algoritmo leva para convergir pode ser ajustado através das matrizes Q e R .

A utilização da estratégia de estimação do KF com identificação dos parâmetros por PSO proposta também se mostrou adequada, conseguindo convergir os valores estimados de v_t , v_{oc} e do SOC mais rapidamente que o EKF. Uma das possíveis causas que levaram este algoritmo a uma estimação mais rápida é a característica da sua matriz $C(k)$, que utiliza o valor real de $\beta_1(k)$, assumido como conhecido, ao invés de calcular a derivada da função $v_{oc} \times SOC$ aplicando-se o valor de estimação $SOC(k|k-1)$ obtido na etapa de predição. Logo, ao retirar a dependência de $C(k)$ dos estados a serem estimados, consegue-se obter resultados melhores de convergência. Porém esta sua vantagem também é o seu problema, uma vez que em implementações práticas, não há como medir $\beta_1(k)$ diretamente, senão através da utilização do $\hat{SOC}$, o que faria com que este algoritmo se comportasse de forma muito semelhante ao EKF.

Com relação à identificação dos parâmetros $\hat{R}_o$, $\hat{R}$ e $\hat{C}$ no algoritmo do KF adaptativo, os mesmos novamente não se mostraram muito influentes nos resultados e mantiveram-se quase constantes nesta configuração. Desta forma, conclui-se que o custo computacional gasto na

estimação dos mesmos através do PSO talvez não seja viável, podendo-se manter tais parâmetros constantes, como foi o caso da estimação com EKF ou utilizar métodos menos custosos de identificação.

Por fim, com relação aos resultados da terceira proposta de estimação do SOC com KF adaptativo e coestimação da curva $v_{oc} \times SOC$, são notadas ainda diversas deficiências. Primeiramente, com relação a identificação dos parâmetros $\hat{R}_o$, $\hat{R}$ e $\hat{C}$, questiona-se novamente a necessidade do algoritmo PSO, já que os mesmos possuem pouca influência na estimação do SOC e poderiam ser substituídos por alternativas menos custosas do ponto de vista computacional. Com relação à estimação do parâmetro β_1 , a qual também é realizada dentro do próprio algoritmo PSO juntamente com $\hat{R}_o$, $\hat{R}$ e $\hat{C}$, foram obtidos resultados bons, mas que ainda carecem de aprimoramentos, uma vez que a identificação de $\hat{\beta}_1$ diverge do valor de β_1 quando o valor da tensão v_t é constante, além de apresentar variações abruptas em momentos específicos de execução do algoritmo (*spikes*). Já a estimação do parâmetro β_o não apresentou resultados satisfatórios, necessitando ainda de um estudo mais aprofundado visando melhorar tais resultados obtidos. Ainda, o erro de estimação desta variável acarreta em grandes erros na estimação do SOC.

Conclui-se através dos fatos expostos que o algoritmo proposto de estimação do SOC com o KF adaptativo e coestimação da curva $v_{oc} \times SOC$ ainda necessita de maiores estudos sobre a estimação do parâmetro β_o , visando melhorar os seus resultados e por consequência, obter melhores estimações do SOC. Destaca-se ainda através da análise dos resultados obtidos, que o sistema de fato não é observável, apesar da análise da seção 5.6.2 dizer o contrário. A razão desta afirmação reside no fato de o sistema dito como observável na seção 5.6.2 ser apenas uma aproximação do sistema real. Desta forma, apesar de comprovada a observabilidade do modelo para o modelo aproximado, tal condição não se mantém para o sistema real, o que compromete os resultados de estimação.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação apresentou um estudo generalista sobre baterias de Lítio e sua estimação de carga, onde foi apresentado o desenvolvimento de um conversor Duplo-Forward com um controlador LQG para ser utilizado como carregador de baterias, além da modelagem e simulação de baterias, as quais são utilizadas na análise dos métodos de estimação do SOC.

A implementação do conversor Duplo-Forward como carregador de baterias, apresentada no capítulo 2, se mostrou satisfatória, indicando que tal conversor pode ser utilizado para carregar pequenos arranjos de baterias. A utilização de um LQG no controle do conversor, apresentada no capítulo 3, também se mostrou adequada à aplicação, a qual funcionou adequadamente com a carga de baterias. Destaca-se ainda que o projeto do controlador LQG foi auxiliado pelo algoritmo PSO, o qual realizou obtenção de todas as matrizes de peso Q e R de forma sistemática, e que resultaram em um controlador com boas respostas dinâmicas. Esta metodologia de projeto do LQG com o ajuste das matrizes de ponderação através do PSO, apesar de ter se mostrado funcional, ainda necessita de estudos mais detalhados, uma vez que o PSO é um algoritmo com características estocásticas.

No capítulo 4, foi abordada a modelagem e simulação de baterias, as quais são necessárias nos estudos de estimação do SOC. Para a modelagem, foi adotado um ECM com parâmetros variáveis, o qual também é utilizado nas simulações de baterias. Os parâmetros do modelo são encontrados iterativamente através de um estimador PSO. Os resultados de simulação desta abordagem mostraram um comportamento muito próximo ao das baterias reais diante de diferentes simulações, porém, são percebidas mudanças sutis nos parâmetros encontrados a cada execução do simulador, que se devem a característica estocástica do PSO. Desta forma, este modelo se mostra adequado apenas para simulações mais generalistas de baterias, nas quais não são necessárias estimações precisas dos parâmetros do modelo.

As estratégias de estimação do SOC são aprofundadas no capítulo 5. Uma revisão da literatura de estimação mostra que a curva característica $v_{oc} \times SOC$ é o fator de maior impacto na estimação do SOC e que não foram encontradas estratégias de estimação autônomas,

capazes de estimar tal curva juntamente com o SOC, as quais teriam como benefício a menor dependência de dados de catálogo ou mesmo de dados experimentais.

Buscando realizar uma estimação mais autônoma, foi proposto nesta dissertação um KF adaptativo com coestimação da curva $v_{oc} \times SOC$, mostrado na seção 5.6. Tal proposta apresentou resultados que são característicos de sistemas não observáveis, compensando em $\hat{\beta}_o(k)$ a estimação errônea de $S\hat{O}C(k)$, ou vice-versa, o que leva a grandes erros de estimação. Apesar disso, notou-se que o parâmetro $\hat{\beta}_1(k)$ obtido através do PSO apresentou um bom comportamento de estimação no geral, e que $\hat{\beta}_o(k)$ possui uma característica de estimação relativamente próxima de $\beta_o(k)$, mas ainda não suficiente para levar a resultados satisfatórios de estimação do SOC. Destaca-se que, conforme mostrado na seção 5.5.1, no caso em que os valores de $\beta_1(k)$ e $\beta_o(k)$ são conhecidos, o KF adaptativo é capaz realizar boas estimativas do valor do SOC das baterias. Porém, vale salientar que $\beta_1(k)$ e $\beta_o(k)$ não podem ser mensurados diretamente da bateria, o que inviabiliza a utilização prática deste método, o qual foi abordado apenas para demonstrar que $\beta_1(k)$ e $\beta_o(k)$ são parâmetros fundamentais na estimação do SOC no caso de um KF adaptativo. O que se faz em geral, é saber da dependência de tais parâmetros em função do próprio SOC, o que implica no conhecimento da curva $v_{oc} \times SOC$, e utilizar o valor de estimação do SOC obtido a cada iteração para então realizar a obtenção da tensão $\hat{v}_{oc}$, recaindo-se assim sobre a estratégia mostrada na seção 5.3.2 com o EKF.

Por fim, conclui-se que o método proposto de estimação com um KF adaptativo com coestimação da curva $v_{oc} \times SOC$ ainda carece de maiores estudos, mas que talvez possa se mostrar uma estratégia de estimação válida se forem feitas algumas adaptações. Logo, este método pode servir como um bom ponto de partida para novas estratégias de estimação do SOC mais autônomas. Com relação à utilização do KF adaptativo, sem estimação da curva $v_{oc} \times SOC$, enfatiza-se a dificuldade da sua implementação prática apesar dos seus bons resultados. Já acerca da metodologia de estimação com EKF, foram confirmados bons resultados e a sua viabilidade prática, porém, destaca-se a necessidade do conhecimento da curva $v_{oc} \times SOC$.

A partir dos tópicos apresentados no trabalho, podem ser listadas as seguintes contribuições:

- Validação da aplicação do conversor Duplo-Forward como carregador de baterias de Lítio, para arranjos com poucas células.
- Validação da aplicação do controlador LQG, em uma configuração em cascata, para o controle do carregador de baterias.
- Apresentação de um método sistemático de obtenção das matrizes de ponderação do LQG do carregador de baterias através do PSO.
- Proposta de uma metodologia de simulação de baterias, utilizando o algoritmo PSO como estimador de parâmetros, podendo esta ser utilizada na simulação de genérica de baterias de Lítio, mas ainda carecendo de estudos mais aprofundados.
- Sintetização das informações disponíveis na literatura acerca da estimação do SOC e destaque para as informações mais relevantes.
- Observação de que o fator de maior impacto (curva $v_{oc} \times SOC$) é assumido conhecido em grande parte das técnicas de estimação do SOC.
- Validação do uso do EKF na estimação do SOC.
- Proposta de uma nova metodologia de estimação do SOC com coestimação da curva $v_{oc} \times SOC$, na seção 5.6. Tal metodologia não apresentou resultados satisfatórios, mas pode ser vista como um possível caminho para novas abordagens de estimação do SOC menos dependentes de parâmetros das baterias.

Diante das conclusões apresentadas sobre o presente trabalho observa-se que todos os objetivos propostos foram alcançados. O autor apresenta a possibilidade dos seguintes trabalhos futuros:

- Realização de um conversor CC-CC bidirecional, capaz de realizar tanto a carga quanto a descarga de baterias.
- Estudo estatístico acerca dos controladores LQG aplicados em conversores estáticos obtidos com o ajuste das matrizes de peso através do PSO.
- Estudo estatístico dos parâmetros e consequentemente dos modelos obtidos através do estimador PSO para a simulação de baterias, diante de diferentes fatores como: condições iniciais dos parâmetros estimados, influência dos parâmetros de ajuste do PSO (número de partículas, ponderações no cálculo do vetor de deslocamento das partículas, etc).
- Avaliação de diferentes formulações da função *fitness* utilizada no PSO para simulação de baterias.
- Utilização de outros algoritmos, baseados em Inteligência computacional ou não, na identificação de parâmetros do modelo da bateria para a estimação do SOC.
- Adaptação da estratégia de estimação proposta baseada no KF adaptativo com coestimação da curva $v_{oc} \times SOC$, visando uma metodologia mais autônoma.

REFERÊNCIAS

- [1] H. Chaoui, N. Golbon, I. Hmouz, R. Souissi, and S. Tahar, “Lyapunov-based adaptive state of charge and state of health estimation for lithium-ion batteries,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1610–1617, Mar. 2015.
- [2] G. H. Paul, “Estudo de métodos de armazenamento de energia e implementação de um sistema caes para aplicações com fontes renováveis,” TCC, Universidade do Estado de Santa Catarina (UDESC), Joinville, SC, Dec. 2015.
- [3] H. Ibrahima, A. Ilincaa, and J. Perron, “Energy storage systems—characteristics and comparisons,” *Renewable and Sustainable Energy Reviews*, pp. 1221–1250, 2008.
- [4] T. Appenzeller. (2004, Jun.) The end of cheap oil. National Geographic Magazine. [Online]. Available: <http://ngm.nationalgeographic.com/ngm/0406/feature5>
- [5] S. S. Williamson, A. K. Rathore, and F. Musavi, “Industrial electronics for electric transportation: Current state-of-the-art and future challenges,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 3021–3032, May 2015.
- [6] L. Dowbor, *Democracia Econômica: um passeio pelas teorias*. L. Dowbor, 2012, ch. Introdução, pp. 9–11. [Online]. Available: <http://dowbor.org/2012/07/democracia-economica-um-passeio-pelas-teorias-edicao-revista-em-nov-2010.html/>
- [7] S. Spoolman and G. T. Miller, *Ecologia e Sustentabilidade*, Cengage, Ed. Cengage, 2012. [Online]. Available: https://books.google.com.br/books?id=z_RdLwEACAAJ
- [8] K. Young *et al.*, “Electric vehicle battery technologies,” in *Electric Vehicle Integration into Modern Power Networks*, New York, USA, 2013, pp. 15–56.

- [9] H. Rahimi-Eichi *et al.*, “Battery management system - an overview of its application in the smart grid and electric vehicles,” *IEEE Industrial Electronics Magazine*, pp. 4–16, 2013.
- [10] A. Purwadi, A. Rizqiawan, A. Kevin, and N. Heryana, “State of charge estimation method for lithium battery using combination of coulomb counting and adaptive system with considering the effect of temperature,” in *Proceedings of the IEEE Conference on Power Engineering and Renewable Energy ICPERE*, 2014, pp. 91–95.
- [11] M. Yilmaz and P. T. Krein, “Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles,” *IEEE Transactions on Power Electronics*, vol. 28, no. 5, pp. 2151–2169, May 2013.
- [12] Y. Zhou and X. Li, “Overview of lithium-ion battery soc estimation,” in *Proceeding of the International Conference on Information and Automation (ICIA)*, Aug. 2015, pp. 2454–2459.
- [13] L. Gao, S. Liu, and R. A. Dougal, “Dynamic lithium-ion battery model for system simulation,” *IEEE Transactions on Components and Packaging Technologies*, vol. 25, no. 3, pp. 495–505, Sep. 2002.
- [14] O. Veneri, L. Ferraro, C. Capasso, and D. Iannuzzi, “Charging infrastructures for ev: Overview of technologies and issues,” in *Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS)*, 2012.
- [15] A. Ausswamaykin and B. Plangklang, “Design of real time management unit for power battery in pv-hybrid power supplies by application of coulomb counting method,” in *Proceedings of the International Electrical Engineering Congress*, 2014.
- [16] L. Xue, J. Jiuchun, Z. Caiping, Z. Weige, and S. Bingxiang, “Effects analysis of model parameters uncertainties on battery soc estimation using h-infinity observer,” in *Proceedings of*

the International Symposium on Industrial Electronics (ISIE), IEEE, Ed., Istanbul, Turkey, Jun. 2014, pp. 1647–1653.

- [17] L. Zhe, L. Languang, O. M. J. of Tsinghua University (Science, and Technology), “Comparison of methods for improving soc estimation accuracy through an ampere-hour integration approach,” *Journal of Tsinghua University (Science and Technology)*, vol. 50, no. 8, pp. 1293–1296, 2010.
- [18] Y. Tian, B. Xia, W. Sun, Z. Xu, and W. Zheng, “Soc estimation of lini1/3co1/3mn1/3o2 battery using unscented kalman filtering method,” in *Proceedings of the IEEE Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, Ed., Hangzhou, China, Jun. 2014, pp. 883–887.
- [19] S. Lee, J. Kim, J. Lee, and B. H. Cho, “State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge,” *J. Power Sources*, vol. 185, no. 2, pp. 1367–1373, Dec. 2008.
- [20] H. X. Lu, Z. F. T. Y. Lu, and S. J. Wang, “Soc dynamic power management using artificial neural network,” in *International Conference on Computational Intelligence and Natural Computing*, Oct. 2006, pp. 133–137.
- [21] G. Capizzi, F. Bonanno, and C. Napoli, “Hybrid neural networks architectures for soc and voltage prediction of new generation batteries storage,” in *International Conference Clean Electrical Power: Renewable Energy Resources*, Jun. 2011, pp. 341–344.
- [22] L. Zheng, J. Jiang, Z. Wang, T. Zhao, and T. He, “Embedded implementation of soc estimation based on the luenberger observer technique,” in *Proceedings of the IEEE Transportation Electrification Conference (ITEC)*, Aug. 2014, pp. 2–4.
- [23] P. Singh, C. Jr., and D. Reisner, “Fuzzy logic modelling of state of charge and available capacity of nickel metal hydride

- batteries,” *Journal of Power Sources*, vol. 136, pp. 322–333, Dec. 2004.
- [24] M. Charkhgard and M. Farrokhi, “State-of-charge estimation for lithiumion batteries using neural networks and ekf,” *IEEE Transactions on Industrial Electronics*, vol. 57, pp. 4178–4187, Dec. 2010.
 - [25] L. Du and X. Cheng, “The research on battery soc estimation within colored noise,” in *Proceedings of the IEEE Transportation Electrification Conference (ITEC)*, Aug. 2014, pp. 1–5.
 - [26] L. Lin, N. Kawarabayashi, M. Fukui, S. Tsukiyama, and I. Shirakawa, “A practical and accurate soc estimation system for lithium-ion batteries by ekf,” in *Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, Oct. 2014.
 - [27] L. Bo, Y. Xueqing, and Z. Lin, “Li-ion battery soc estimation based on ekf algorithm,” in *Proceedings of the IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER)*, Jun. 2015, pp. 1584–1588.
 - [28] L. Guo, C. Hu, and G. Li, “The soc estimation of battery based on the method of improved ampere-hour and kalman filter,” in *Proceedings of the IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Jun. 2015, pp. 1458–1460.
 - [29] E. Kamal, A. E. Hajjaji, and A. M. Mabwe, “State of charge estimation based on extened kalman filter algorithm forlithium-ion battery,” in *Proceedings of the Mediterranean Conference on Control and Automation (MED)*, Jun. 2015, pp. 734–739.
 - [30] B. Fridholm, M. Nilsson, and T. Wik, “Robustness comparison of battery state of charge observers for automotive applications,” in *Proceedings of the World Congress International Federation of Automatic Control (IFAC)*, Aug. 2014, pp. 2138–2146.
 - [31] P.-H. Michel and V. Heiries, “An adaptive sigma point kalman filter hybridized by support vector machine algorithm for bat-

tery soc and soh estimation,” in *Proceedings of the Vehicular Technology Conference (VTC)*, May 2015.

- [32] H. Aung and K. S. Low, “Temperature dependent state-of-charge estimation of lithium ion battery using dual spherical unscented kalman filter,” *Institution of Engineering and Technology Journal of Power Electronics*, vol. 8, no. 10, pp. 2026–2033, Apr. 2015.
- [33] J. Meng, G. Luo, and F. Gao, “Lithium polymer battery state-of-charge estimation based on adaptive unscented kalman filter and support vector machine,” *IEEE Transactions on Power Electronics*, vol. 31, no. 3, pp. 2226–2238, Mar. 2016.
- [34] Y. Chen, D. Huang, D. Feng, and K. Wei, “An h-infinity filter based approach for battery soc estimation with performance analysis,” in *Proceeding of the International Conference on Information and Automation (ICIA)*, Aug. 2015, pp. 1613–1618.
- [35] W. F. Schmitt, “Metodologia de avaliação de eficiência energética em veículos leves e seus impactos em termos de uso de combustíveis,” Tese, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, 2010.
- [36] O. of Energy Efficiency & Renewable Energy. (2016, Jan.) Fact 855 january 12, 2015 electric vehiclechargers by network and state. Department of Energy’s Alternative Fuels Data Center. [Online]. Available: <http://energy.gov/eere/vehicles/fact-855-january-12-2015-electric-vehicle-chargers-network-and-state>
- [37] R. Mayer, “Conversor cc-cc multifásico bidirecional em corrente não isolado aplicado a sistemas elétricos de tração de veículos elétricos e híbridos leves,” Master’s thesis, Universidade Regional de Blumenau (FURB), 2014.
- [38] R. Mayer, A. Péres, and S. V. G. Oliveira, “Conversor cc-cc multifásico bidirecional em corrente não isolado aplicado a sistemas elétricos de tração de veículos elétricos e híbridos,” *Re-*

- vista Brasileira de Eletrônica de Potência*, vol. 20, no. 3, p. 311, 2015.
- [39] C. L. Remes, M. B. Rosa, and S. V. G. Oliveira, “A two-switch forward converter application for battery charging,” in *Proceedings of the Brazilian Power Electronics Conference - Southern Power Electronics Conference (COBEP-SPEC)*, 2015.
 - [40] A. J. N. Borges, “Conversor duplo-forward pwm a duas chaves principais e duas chaves auxiliares com comutação suave,” *Dissertação*, Universidade Federal De Uberlândia (UFU), Uberlândia, May 2006.
 - [41] K. D. Coelho, “Conversores cc-cc com elevada tensão de entrada para fontes auxiliares,” *Tese*, Universidade Federal de Santa Catarina (UFSC), Florianópolis, Dec. 2004.
 - [42] C.-S. Leu, “A novel forward configuration for off-line applications: Two-switch built-in input filter forward converter (2sbiffc),” *IEEE Power Electronics Section*, pp. 1035–1040, 1996.
 - [43] R. P. T. Bascopé, “Conversores cc-cc zvs-pwm duplo forward com acoplamento magnético,” *Tese*, Universidade Federal de Santa Catarina (UFSC), Florianópolis, Dec. 2000.
 - [44] M. K. Kazimierczuk, *Pulse-width Modulated DC-DC Power Converters*, 1st ed. Wright State University, Dayton, Ohio, USA: John Wiley & Sons, Inc., 2008.
 - [45] M. Prudík and P. Vorel, “Advantages of using two-switch forward converter for high-voltage applications,” in *International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 2012, pp. 326–330.
 - [46] I. Barbi, *Projeto de Fontes Chaveadas*. Florianópolis, Brazil: Author’s Edition, 2001.

- [47] K. Thirugnanam *et al.*, “Mathematical modeling of li-ion battery using genetic algorithm approach for v2g applications,” *IEEE Transactions On Energy Conversion*, pp. 1–12, 2014.
- [48] (2014, May) Learn about batteries. Battery University. [Online]. Available: <http://batteryuniversity.com/>
- [49] celticrequiem. (2016, Jan.) Lithium ion battery explosion. Youtube. [Online]. Available: https://www.youtube.com/watch?v=SMY2_qNO2Y0
- [50] C. Channel. (2016, Jan.) Battery explosion compilation vol3 new battery test video. Youtube. [Online]. Available: <https://www.youtube.com/watch?v=dMSDXq2pDfo>
- [51] J.-J. Chen, F.-C. Yang, C.-C. Lai, Y.-S. Hwang, , and R.-G. Lee, “A high-efficiency multimode li-ion battery charger with variable current source and controlling previous-stage supply voltage,” *IEEE Transactions On Industrial Electronics*, vol. 56, no. 7, pp. 2469–2478, Jul. 2009.
- [52] B.-Y. Chen and Y.-S. Lai, “New digital-controlled technique for battery charger with constant current and voltage control without current feedback,” *IEEE Transactions On Industrial Electronics*, vol. 59, no. 3, pp. 1545–1553, Mar. 2012.
- [53] S. O. Yong and N. A. Rahim, “Development of on-off duty cycle control with zero computational algorithm for cc-cv li ion battery charger,” in *IEEE Conference on Clean Energy and Technology (CEAT)*, 2013.
- [54] J.-S. Moon, J.-H. Lee, I.-Y. Ha, T.-K. Lee, and C.-Y. Won, “An efficient battery charging algorithm based on state-of-charge estimation for electric vehicle,” in *International Conference on Electrical Machines and Systems (ICEMS)*, 2011.
- [55] K.-H. Park, Y.-J. Choi, S.-Y. Choi, and R.-Y. Kim, “Design consideration of cc-cv controller of llc resonant converter for

- li-ion battery charger,” in *International Future Energy Electronics Conference (IFEEC)*, 2015.
- [56] Panasonic. (2014, Dec.) Ncr18650 catalog nnp series. [Online]. Available: <http://industrial.panasonic.com/www-data/pdf2/ACA4000/ACA4000CE240.pdf>
- [57] R. W. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 2nd ed. New York, N.Y.: Kluwer Academic Publishers, 2000.
- [58] Z. Al-Hamouz, H. Al-Duwaish, and I. El-Amin, “A robust control for dc-dc switched mode power converters,” in *IEEE Industrial Applications Society Annual Meeting*, 1997.
- [59] L. Camacho-Solorio and A. Sariñana-Toledo, “I-lqg control of dc-dc boost converters,” in *Proceedings of the International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, IEEE, Ed., 2014.
- [60] P. C. Loh, M. J. Newman, D. N. Zmood, and D. G. Holmes, “A comparative analysis of multiloop voltage regulation strategies for single and three-phase ups systems,” *IEEE Transactions On Power Electronics*, vol. 18, no. 5, pp. 1176–1185, Sep. 2003.
- [61] F. O. Martinz, “Estudo de estratégias de rastreamento da corrente e da tensão de saída ca de um conversor do tipo fonte de tensão,” Dissertação, Universidade de São Paulo, São Paulo, 2007.
- [62] K. Ogata, *Modern Control Engineering*, P. E. International, Ed. Aeeizh, 2002.
- [63] C. L. Remes, R. Sebem, and M. S. M. Cavalcá, “Algoritmos bio-inspirados aplicados ao ajuste de um regulador linear quadrático,” *XX CBA - Congresso Brasileiro de Automática*, vol. 20, pp. 2909–2915, 2014.

- [64] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1995.
- [65] C. Wongsathan and C. Sirima, "Application of ga to design lqr controller for an inverted pendulum system," in *Proceedings of the 2008 IEEE*, 2009.
- [66] H. Selamat and S. D. A. Bilong, "Optimal controller design for a railway vehicle suspension system using particle swarm optimization," in *Proceedings of the IEEE*, 2013.
- [67] J. P. Hespanha, "Lecture notes on lqr/lqg controller design," *Lecture Notes*, 2005. [Online]. Available: <http://www.uz.zgora.pl/~wpaszke/materialy/kss/lqrnotes.pdf>
- [68] R. Ahmed, M. E. Sayed, I. Arasaratnam, J. Tjong, and S. Habibi, "Reduced-order electrochemical model parameters identification and soc estimation for healthy and aged li-ion batteries part i: Parameterization model development for healthy batteries," *IEEE Journal Of Emerging and Selected Topics in Power Electronics*, vol. 2, no. 3, pp. 659–677, 2014.
- [69] R. C. Kroeze and P. T. Krein, "Electrical battery model for use in dynamic electric vehicle simulations," in *Power Electronics Specialists Conference (PESC)*, Urbana, IL, USA, 2008, pp. 1336–1342.
- [70] C. Blanco *et al.*, "An equivalent circuit model with variable effective capacity for lifepo4 batteries," *IEEE Transactions on Vehicular Technology*, 2013.
- [71] H. Rahimi-Eichi, F. Baronti, and M.-Y. Chow, "Online adaptive parameter identification and state-of-charge coestimation for lithium-polymer battery cells," *IEEE Transactions On Industrial Electronics*, vol. 61, no. 4, pp. 2053–2061, 2014.
- [72] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A generic battery model for the dynamic simulation of hybrid electric

- vehicles,” in *Proceedings of the Vehicle Power and Propulsion Conference (VPPC)*, Canada, 2007, pp. 284–289.
- [73] E. reena Joy T.P., K. Thirugnanam, and P. Kumar, “Bidirectional contactless charging system using li-ion battery model,” in *Proceedings of the IEEE International Conference on Industrial and Information Systems (ICIIS)*, Guwahati, India, 2012, pp. 1–6.
 - [74] R. Das *et al.*, “Mathematical modeling for economic evaluation of electric vehicle to smart grid interaction,” *IEEE Transactions On Smart Grid*, vol. 5, no. 2, pp. 712–721, 2014.
 - [75] A. Baba and S. Adachi, “State of charge estimation of hev/ev battery with series kalman filter,” in *SICE Annual Conference 2012*, Akita, Japan, Aug. 2012, pp. 845–850.
 - [76] M. Zheng, B. Qi, and X. Du, “Dynamic model for characteristics of li-ion battery on electric vehicle,” in *Proceedings of the Industrial Electronics and Applications (ICIEA)*, Beijing, Chine, 2009, pp. 2867–2871.
 - [77] Z. Miao *et al.*, “An soc-based battery management system for microgrids,” *IEEE Transactions On Smart Grid*, vol. 5, no. 2, pp. 966–973, 2014.
 - [78] K. Li *et al.*, “Large-scale battery system modeling and analysis for emerging electric-drive vehicles,” in *Proceedings of the International Symposium on Low-Power Electronics and Design (ISLPED)*, Colorado, USA, 80309, 2010, pp. 277–282.
 - [79] K. A. Smith, “Electrochemical control of lithium-ion batteries,” *IEEE Control Systems Magazine*, pp. 18–25, 2010.
 - [80] L. W. Yao, J. A. Aziz, P. Y. Kong, and N. R. N. Idris, “Modeling of lithium-ion battery using matlab/simulink,” *IEEE Conference*, pp. 1729–1734, 2013.

- [81] T. Zhao, J. Jiang, C. Zhang, L. Zheng, and F. Wen, “Error analysis of soc estimation based on pi observer,” in *Proceedings of the IEEE Transportation Electrification Conference (ITEC)*, 2014, pp. 1–5.
- [82] J. Safi, M. Beeney, M. Kehs, J. Anstrom, S. Brennan, and H. Fathy, “Improving soc accuracy using collective estimation for lithium ion battery cells in series,” in *Proceedings of the American Control Conference (ACC)*, Jun. 2014, pp. 254–259.
- [83] H. Rahimi-Eichi, F. Baronti, and M.-Y. Chow, “Modeling and online parameter identification of li-polymer battery cells for soc estimation,” in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE)*, 2012.
- [84] R. F. Souto, “Estimadores robustos para sistema lineares e não-lineares,” Master’s thesis, Universidade de Brasília, 2009.
- [85] I. Barbi, *Eletrônica de Potência*, Autor, Ed. Autor, 2006.
- [86] A. Devices. (2016, Jan.) Reducing ground bounce in dc-to-dc converters—some grounding essentials. [online]. [Online]. Available: http://www.analog.com/library/analogdialogue/archives/41-06/ground_bounce.pdf

APÊNDICES

Apêndice – A DIMENSIONAMENTO DO CONVERSOR DUPLO-FORWARD

A.1 DIMENSIONAMENTO DO RETIFICADOR

Nesta seção de Apêndice são apresentados os passos de projeto do estágio de conversão CA-CC. A topologia é um retificador ponte completa com filtro capacitivo e todos os passos de projeto seguem a metodologia descrita por [46] e [85].

A.1.1 Dimensionamento do filtro capacitivo do retificador

Como requisito de projeto para o retificador, foi definida que a ondulação máxima de tensão no capacitor do filtro não pode ser maior que 3%. Através da Tabela 3, tem-se que a tensão máxima admita pelo conversor é $V_{in_{max}} = 326 \text{ V}$, sendo este valor também considerado como a máxima tensão de pico do retificador, enquanto que o valor mínimo é $V_{in_{min}} = 295 \text{ V}$. Para o desenvolvimento do retificador, foi assumido um rendimento $\eta = 0,7$ para o conversor, o que resulta em:

$$P_{in_{max}} = \frac{P_{o_{max}}}{\eta} = 171,43 \text{ W} \quad (\text{A.1})$$

Então, a impedância equivalente do conversor na condição de carga máxima (impedância mínima) é dada por (A.2):

$$R_{fc} = \frac{V_{in_{min}}^2}{P_{in_{max}}} = 507,65 \text{ } \Omega \quad (\text{A.2})$$

Conforme a metodologia de [85], uma ondulação de 3% resulta em um $\omega RC = 80 \text{ } \Omega.F.rad/s$. A frequência de operação da rede é de $\omega = 2\pi 60 \text{ rad/s}$. A partir destas informações, chega-se no valor de capacitância necessária ao filtro capacitor para atingir a ondulação de tensão desejada, segundo equação (A.3):

$$C_{fc_{calc}} = \frac{\omega RC}{\omega R_{fc}} = 418,02 \text{ } \mu F \quad (\text{A.3})$$

Tendo-se o valor de capacitância necessário para o filtro do retificador, deve-se então definir qual a corrente eficaz que o capacitor do filtro deve suportar. Sendo o valor de $\omega RC = 80 \Omega.F.rad/s$, seguindo os passos de [85], tem-se que o valor eficaz parametrizado da corrente no capacitor é de $R.I_{cef}/V_{pk} = 4$. Sendo assim, o valor eficaz da corrente a ser suportado pelo capacitor do filtro é:

$$I_{cef} = RI_{cef}/V_{pk} \frac{V_{pmax}}{R_{fc}} = 2,569 A \quad (A.4)$$

Por fim, foram selecionados dois capacitores Philips Snap In Cap 450 V 220 μF , com corrente eficaz máxima de 1,73 A cada, o que resulta em um filtro capacitivo com capacitância equivalente de $C_{fc} = 440 \mu F$ e capacidade máxima de corrente eficaz $I_{acfc} = 3,46 A$. A resistência série de cada capacitor do filtro é 0,4 Ω , o que resulta em uma resistência série equivalente $ESR_{fc} = 0,2 \Omega$.

A.1.2 Dimensionamento dos diodos do retificador

O dimensionamento dos diodos do estágio CA-CC foi realizado através de dados obtidos em simulações do software Cadence OrCAD®. Através das simulações, foram obtidos os valores de corrente eficaz $I_{efrect} = 1,6 A$ e média $I_{medrect} = 0,4 A$ para cada diodo. Sabe-se ainda que o diodo deve suportar uma tensão reversa de pico de ao menos 326 V. Assim, foi escolhido o diodo 1N5404, que suporta uma tensão reversa de pico $V_{RRM} = 400 V$ e uma corrente média de até 3,0 A, mais do que suficiente para a aplicação requisitada.

Para uma corrente média $I_{medrect} = 0,4 A$, cada diodo 1N5404 possui uma queda de tensão $V_{drect} = 0,2 V$. Desta forma, tem-se que a dissipação de potência por diodo 1N5404 é $P_{drect} = 0,08 W$. A resistência térmica junção-ambiente do diodo é $R_{thja} = 15 K/W$, resultando em uma elevação de temperatura de $\Delta T_{rect} = 1,2 ^\circ C$, dispensando assim a necessidade de dissipadores para estes semicondutores.

A seguir, serão descritas todas as etapas de projeto do conversor Duplo-Forward.

A.2 PROJETO DO CONVERSOR DUPLO-FORWARD

Nesta seção, serão descritos todos os passos para dimensionamento dos componentes e execução do projeto do estágio CC-CC do carregador de baterias. Todas as etapas a seguir são parte da metodologia de projeto do conversor Forward descrita em [46].

Visando uma menor complexidade nas etapas de projeto, as seguintes suposições serão utilizadas:

- Todos os componentes são ideais, sendo desconsideradas quaisquer resistências durante a condução e considerando que os interruptores são um circuito completamente aberto durante o bloqueio.
- O conversor opera em modo de condução contínua e em regime permanente.
- A fonte de tensão de entrada V_{in} é constante, sendo desconsiderada a ondulação do estágio retificador para o projeto do estágio CC-CC.

A.2.1 Relação de transformação

O conversor Duplo-Forward é um conversor que possui isolamento galvânica entre a entrada e a saída, sendo esta feita através de um transformador de alta frequência. Este mesmo transformador pode ser projetado de forma a adequar melhor os níveis de tensão de saída.

Uma vez que as especificações de entrada trabalham com tensões elevadas e as de saída com tensões muito menores (na ordem de 100 vezes), há uma dificuldade maior em se obter a mínima tensão de saída do conversor, agravada pelo fato da dificuldade prática em se trabalhar com razões cíclicas menores que 0,1. Assim, a relação de transformação será escolhida de forma a garantir que a menor tensão de saída seja alcançada com razão cíclica próxima de $D_{min} = 0,1$ mesmo com a maior tensão de entrada especificada. Ainda, o rendimento do conversor será considerado unitário ($\eta = 1$), o que neste caso, representa o pior caso. Definindo a relação de transformação $N_t = 12$,

tem-se através de (A.5) que a razão cíclica mínima necessária para atingir a tensão de saída mínima é:

$$D_{min} = \frac{V_{o_{min}}}{V_{in_{max}}} N_t = 0,092 \quad (\text{A.5})$$

Tal valor se mostra adequado para utilização, uma vez que é próximo de 0,1. Ainda, na prática, os resultados tendem a ser melhores, devido ao rendimento do conversor ser menor que a unidade. Logo, a relação de transformação N_t será fixada em 12 : 1, onde há uma espira no enrolamento secundário para cada 12 espiras do enrolamento primário. Ainda, a razão cíclica mínima será assumida como sendo $D_{min} = 0,1$.

A.2.2 Projeto do filtro de saída

O filtro de saída do 2SFC é um filtro de segunda ordem, composto por um indutor série e um capacitor em paralelo. Seguindo as especificações de projeto dadas na Tabela 3, pode-se calcular o valor de indutância necessária conforme a equação (A.6). O fator $\eta_v = 0,9$ representa qual o valor relativo de tensão no indutor em relação ao valor presente na entrada, como uma espécie de rendimento. Desta forma, tem-se:

$$L_{o_{calc}} = \frac{V_{in_{max}}}{4 \frac{N_t}{\eta_v} f_s \Delta I_{L_{max}}} \quad (\text{A.6})$$

O que resulta em $L_{o_{calc}} = 40,75 \mu H$. A partir deste valor, foi fixado um valor de indutância $L_o = 42 \mu H$ para o projeto do indutor. Para a determinação do valor da capacitância de saída, é utilizada a relação dada por (A.7), que resulta em $C_{o_{calc}} = 795,78 \mu F$.

$$C_{o_{calc}} = \frac{\Delta I_{L_{max}}}{2\pi f_s \Delta V_{o_{max}}} \quad (\text{A.7})$$

Por questões de disponibilidade, foram escolhidos dois capacitores Epcos B41821 – 50 V – 470 μF , que possuem tolerância de 20%. O valor de capacitância equivalente do filtro de saída é então

$$C_o = 940 \mu F.$$

A resistência série dos capacitores e a corrente eficaz suportada pelos mesmos devem ser obtidas através de alguns dados disponíveis no manual do fabricante. Dentre os dados disponíveis estão a $ESR_{C_{o_{cat}}} = 0,42 \Omega$ e $I_{ac_{cat}} = 0,760 A$, ambas a uma frequência de $120 Hz$ e temperatura de $20 ^\circ C$ (a operação deste componente se dá com uma frequência de $100 kHz$, que é a frequência de chaveamento). Também é dada a tangente de perdas $\tan\delta = 0,12$ e a corrente máxima normalizada suportada pelo capacitor de $1,4$ a uma frequência $f_s = 100 kHz$. A resistência série de cada capacitor é dada através da expressão (A.8). A corrente eficaz de cada capacitor é dada por (A.9).

$$ESR_{C_{o1}} = \tan\delta \frac{1}{2\pi f_s 0,5 C_o} = 0,4 m\Omega \quad (A.8)$$

$$I_{C_{o_{max}}} = 1,4 I_{ac_{cat}} = 1,064 A_{rms} \quad (A.9)$$

A resistência série máxima admitida pelos capacitores de saída é dada por (A.10):

$$ESR_{max} = \frac{\Delta V_{o_{max}}}{\Delta I_{L_{max}}} = 2 m\Omega \quad (A.10)$$

A corrente eficaz suportada pelos dois capacitores é de $2,128 A$, enquanto que a ondulação de corrente (parcela CA da corrente, que circula sobre ambos capacitores) é $\Delta I_{L_{max}} = 1,6 A$. Com relação a resistência série, a associação dos dois capacitores resulta em uma $ESR_{C_o} = 0,2 m\Omega$, enquanto que a máxima permitida é de $ESR_{max} = 2 m\Omega$. Logo, os capacitores escolhidos atendem tanto a capacidade de corrente necessária quanto a resistência série máxima admitida.

A.2.3 Projeto dos elementos magnéticos

Tendo sido feita a definição do filtro de saída e da relação de transformação do transformador de alta frequência, realiza-se então a etapa de projeto e execução dos elementos magnéticos do conversor, também seguindo a metodologia de [46].

A.2.3.1 Projeto e Execução do Indutor de Saída

Inicialmente será aqui demonstrado o projeto do indutor de saída. Tendo o valor da corrente de saída do conversor $I_{o_{med}}$, que representa o valor médio da corrente do indutor, e a ondulação de corrente, são obtidos os valores da corrente de pico $I_{L_{pk}}$ e corrente eficaz do indutor $I_{L_{ef}}$, mostrados na Tabela 9. Os valores de projeto definidos para a densidade de fluxo magnético B_{max} , fator de ocupação da janela k_w e máxima densidade de corrente J_{max} também estão descritos na mesma tabela.

Tabela 9 – Definições para o projeto do indutor

Descrição	Variável	Valor
Corrente média no indutor	$I_{o_{med}}$	12 A
Corrente eficaz no indutor	$I_{L_{ef}}$	12,43 A
Pico da corrente no indutor	$I_{L_{pk}}$	12,75 A
Densidade de fluxo magnético	B_{max}	0,29 T
Fator de ocupação da janela	k_w	0,5
Densidade de corrente máxima	J_{max}	650 A/cm ²

Fonte: produção do próprio autor, 2016

Através dos dados da Tabela 9, é calculado o fator $AeAw$ desejado, dado pela expressão (A.11):

$$AeAw_L = \frac{L_o I_{L_{o_{pk}}} I_{L_{o_{ef}}}}{B_{max} J_{max} k_w} = 0,706 \text{ cm}^4 \quad (\text{A.11})$$

Pela disponibilidade e proximidade do valor $AeAw_L$ calculado, foi escolhido o núcleo de ferrite Thornton NEE- 30/15/14, que possui um $AeAw_{cat} = 1,31 \text{ cm}^4$, calculado através de suas dimensões. Foi considerada que a área Aw da janela do carretel é 10% menor que a área da janela do núcleo de ferrite para este cálculo.

O número de enrolamentos do indutor pode ser obtido através

da expressão (A.12):

$$N_L = \frac{L_o I_{L_{opk}}}{B_{max} A_e} = 15,136 \quad (\text{A.12})$$

A partir de (A.12), foi definido que o número de voltas do enrolamento do indutor deve ser $N_L = 15$. O tamanho do entreferro (*gap*) do indutor é obtido através de (A.13):

$$l_{gap} = \frac{N_t^2 \mu_o A_e}{L_o} = 0,526 \text{ mm} \quad (\text{A.13})$$

Onde μ_o é a permeabilidade magnética do ar.

Ainda, há a necessidade de se calcular as seções dos condutores a serem utilizados. Uma vez que a densidade de corrente J_{max} e a corrente eficaz $I_{L_{ef}}$ estão definidas, tem-se então a seção total que o condutor como um todo deve ter em (A.14):

$$S_{total_{calc}} = \frac{I_{L_{ef}}}{J_{max}} = 1,913 \text{ mm}^2 \quad (\text{A.14})$$

Foi escolhido então um condutor de cobre com 18 AWG ($S_{fio} = 0,82 \text{ mm}^2$). Devido a componente CC (corrente média) possuir uma parcela muito mais representativa da corrente total do que a componente CA (ondulação de corrente), foi desconsiderada a ação do efeito pelicular para a escolha do condutor. Sabendo-se então da área total de seção necessária e a seção de cada condutor, calcula-se o número de condutores necessários:

$$n_{cond_L} = \frac{S_{total_{cond}}}{S_{fio}} = 2,33 \quad (\text{A.15})$$

Definindo o número de condutores em 2, tem-se então que a seção total dos condutores do indutor é $S_{total_{cond}} = 1,64 \text{ mm}^2$. Apesar da seção $S_{total_{cond}}$ ter um valor menor que a seção $S_{total_{calc}}$ calculada, foi mantida esta escolha a fim de manter um tamanho de indutor menor, pois, com maiores seções ou número de condutores, a execução do indutor não era possível com o núcleo NEE-30/15/14. O preço de tal

escolha são maiores perdas no indutor e maior elevação de temperatura.

Para o cálculo das perdas e da elevação de temperatura no indutor, deve-se primeiramente determinar a resistência do mesmo, obtida através de (A.16):

$$R_{cobre.L} = \frac{\rho_{fio} L_e N_L}{S_{total_{cond}}} = 0,012 \, \Omega \quad (A.16)$$

Tendo a corrente eficaz do indutor, as perdas nos condutores são dadas por (A.17):

$$P_{cobre.L} = R_{cobre.L} I_{Loef}^2 = 1,99 \, W \quad (A.17)$$

Já a potência dissipada no núcleo magnético é dada pela equação (A.18):

$$P_{nucleo.L} = (0,1 B_{max})^{2,4} (K_h f_s + K_f f_s^2) V_e = 0,013 \, W \quad (A.18)$$

Onde $K_h = 4 \cdot 10^{-5}$, $K_f = 4 \cdot 10^{-10}$ são constantes do ferrite e V_e é o volume do núcleo magnético.

Por fim, tendo-se as potências dissipadas no elemento magnético, deve-se calcular a resistência térmica do núcleo $R_{th_{nucleo.L}}$ a fim de conhecer a sua elevação de temperatura. Para um núcleo de ferrite, $R_{th_{nucleo.L}}$ é dada por (A.19):

$$R_{th_{nucleo.L}} = 23(AeAw_{cat})^{-0,37} = 20,813 \, K/W \quad (A.19)$$

E então, a elevação de temperatura do indutor é obtida em (A.20):

$$\Delta T_L = (P_{cobre.L} + P_{nucleo.L}) R_{th_{nucleo.L}} = 41,7 \, K \quad (A.20)$$

Assim, a uma temperatura ambiente $T_a = 40 \, ^\circ C$, este indutor deve trabalhar a cerca de $80 \, ^\circ C$, estando assim dentro dos limites do material de ferrite para uma boa operação.

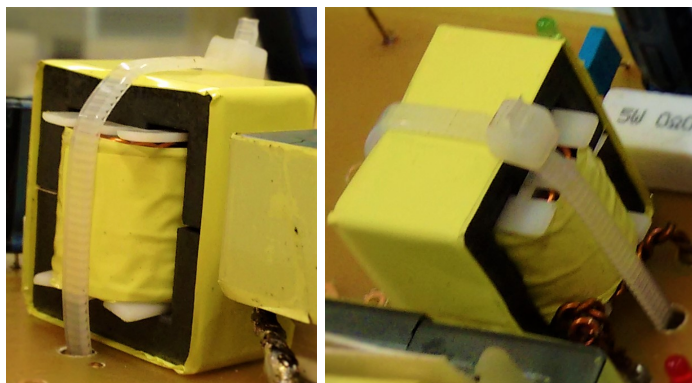
O fator de execução deste projeto de indutor, dado pela expressão (A.21), ficou em 0,462, o que mostra que sua execução é viável,

pois o fator possui valor menor que 1.

$$Exec_L = \frac{N_L S_{total_{cond}}}{k_w Aw} \quad (A.21)$$

Tendo tais parâmetros de projeto do indutor, foi realizada a execução do mesmo. A Figura 74 mostra o indutor pronto após sua fixação na placa do conversor.

Figura 74 – Imagens do indutor produzido



Fonte: produção do próprio autor, 2016

A Tabela 10 mostra os valores experimentais obtidos após a execução do indutor, estando estes bem próximos dos valores teóricos esperados.

Estando o indutor concluído, passa-se agora à etapa de projeto e execução do transformador.

A.2.3.2 Projeto e Execução do Transformador

O transformador é um elemento crítico de um conversor CC-CC isolado, isto porque uma má execução do transformador pode levar à indutâncias e capacitâncias parasitas indesejadas, responsáveis principalmente por picos de tensão que podem ser fatais aos semicondutores. Desta forma, uma boa execução deste componente é primordial para

Tabela 10 – Dados experimentais do indutor manufaturado.

Descrição	Variável	Valor
Resistência do indutor experimental	R_L	$12\text{ m}\Omega$
Potência dissipada no indutor (exp.)	$P_{d.L}$	$1,855\text{ W}$
Variação de Temperatura do Indutor	ΔT_L	$40\text{ }^\circ\text{C}$
Indutância experimental	L_o	$42,5\text{ }\mu\text{H}$

Fonte: produção do próprio autor, 2016

minimizar as sobretensões decorrentes dos bloqueios dos semicondutores, que pode levar ao sobredimensionamento dos mesmos ou necessidade de aumento da potência dissipada em circuitos de grampeamento e amortecimento, conhecidos como *snubbers*.

Os passos de projeto e execução do transformador são similares aos realizados para o indutor. A Tabela 11 contém as definições utilizadas para o projeto do transformador.

Tabela 11 – Definições para o projeto do transformador

Descrição	Variável	Valor
Densidade de corrente no primário	J_p	330 A/cm^2
Densidade de corrente no secundário	J_s	330 A/cm^2
Densidade de corrente máxima	J_{max_t}	330 A/cm^2
Densidade de fluxo magnético	B_{max_t}	$0,13\text{ T}$
Fator de ocupação da janela	k_{wt}	$0,4$
Fator de ocupação do primário	k_{wt}	$0,45$
Rendimento do transformador	η_t	$0,85$

Fonte: produção do próprio autor, 2016

Onde B_{max_t} é a máxima densidade de fluxo do transformador, que neste caso é menos da metade do valor escolhido para o indutor, uma vez que o transformador do Duplo-Forward opera em apenas um quadrante da curva $B - H$ e ainda deve-se garantir que o mesmo não

atingirá a saturação. k_{wt} é o fator de ocupação da janela do transformador, k_{pt} é o fator de ocupação do primário, J_p e J_s são as densidades de corrente dos enrolamentos primário e secundário, respectivamente, e η_t é o rendimento esperado do transformador. Por fim, J_{max_t} é o valor da densidade de corrente máxima do transformador, que neste caso, é igual à de ambos enrolamentos.

A equação (A.22) calcula o fator $AeAw_t$:

$$AeAw_t = \frac{2P_{o_{max}}}{k_{wt}k_{pt}J_{max_t}f_sB_{max_t}\eta_t} = 3,656 \text{ cm}^4 \quad (\text{A.22})$$

O núcleo magnético escolhido foi o núcleo de ferrite NEE-42/21/15 da Thornton, o qual possui um $AeAw_{cat_t} = 4,171 \text{ cm}^4$.

O número de voltas do enrolamento primário é definido pela equação (A.23):

$$N_{p_{calc}} = \frac{V_{in_{max}}}{2Aef_sB_{max_t}} = 69,273 \quad (\text{A.23})$$

Devido a um erro de cálculo durante a etapa de projeto, o número de espiras escolhidas no primário foi $N_p = 84$, e este valor então foi utilizado para as demais etapas de projeto e na execução do transformador. Assim, salienta-se que o transformador utilizado apresenta sobredimensionamento da sua indutância magnetizante.

Através da relação de transformação, obtém-se em (A.24) o número de espiras do enrolamento secundário:

$$N_s = \frac{N_p}{N_t} = 7 \quad (\text{A.24})$$

Para o cálculo dos condutores do transformador, deve-se levar em conta o efeito pelicular. A equação (A.25) traz uma relação entre o raio máximo que um condutor deve ter em função da frequência, de forma a minimizar o efeito pelicular.

$$r_{skin} = \frac{7,5}{\sqrt{f_s}} = 0,237 \text{ mm} \quad (\text{A.25})$$

Desta forma, tem-se que a seção máxima de cada condutor não deve ser maior que $S_{skin} = 0,177 \text{ mm}^2$. A área total da seção de cobre dos condutores dos enrolamentos primários e secundário deve ser maior que $Sp_{total} = 0,203 \text{ mm}^2$ e $Ss_{total} = 2,439 \text{ mm}^2$, respectivamente. Obedecendo a relação de seção máxima em função do efeito pelicular, foram escolhidos para o enrolamento primário 3 condutores de 27 AWG ($S_{fio_p} = 0,10 \text{ mm}^2$) e para o secundário 18 condutores de 25 AWG ($S_{fio_s} = 0,16 \text{ mm}^2$), obtendo desta forma uma seção total de $Sp_{total_{cond}} = 0,3 \text{ mm}^2$ e $Ss_{total_{cond}} = 2,88 \text{ mm}^2$ para o primário e para o secundário, nesta ordem.

Tendo os condutores, pode-se então determinar o fator de execução do transformador, conforme equação (A.26):

$$Exec_t = \frac{N_p Sp_{total_{cond}} + N_s Ss_{total_{cond}}}{k_{wt} Aw_t} = 0,492 \quad (\text{A.26})$$

Por fim, devem ser analisadas as perdas no transformador e sua elevação de temperatura. Os valores calculados de resistência dos enrolamentos e de resistência térmica do núcleo, além das potências dissipadas no núcleo e nos enrolamentos e da elevação de temperatura, são obtidos de maneira análoga à seção A.2.3.1 de projeto do indutor. Seus valores estão descritos na Tabela 12:

Tabela 12 – Cálculo das perdas do transformador

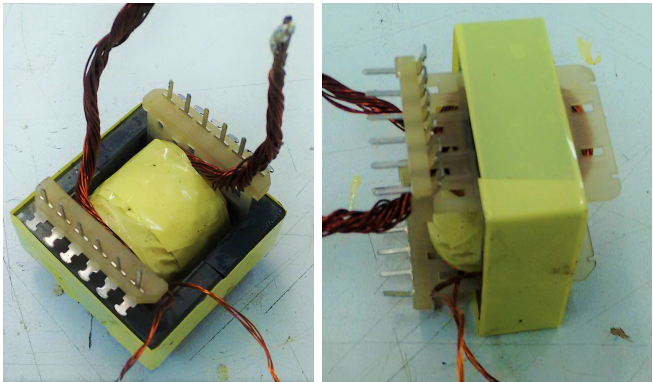
Descrição	Variável	Valor
Resistência calculada do primário	$R_{calc.1}$	$0,565 \Omega$
Resistência calculada do secundário	$R_{calc.2}$	$4,909 \text{ m}\Omega$
Perdas nos enrolamentos	$P_{calc.r.t}$	$0,573 \text{ W}$
Perdas no núcleo	$P_{calc.nucleo.t}$	$1,052 \text{ W}$
Elevação de Temperatura	$\Delta T_{calc.t}$	$22 ^\circ C$

Fonte: produção do próprio autor, 2016

Observa-se que os valores das perdas somadas nos enrolamentos e das perdas no núcleo são próximos, o que indica um bom dimensionamento do transformador. Além disso, a elevação de temperatura ficou

em $\Delta T_t = 22\text{ K}$, levando o magnético a operar sob temperaturas menores que $80\text{ }^{\circ}\text{C}$, considerando-se uma temperatura ambiente elevada $T_a = 40\text{ }^{\circ}\text{C}$. Tendo este projeto um fator de execução de 0,492 (menor que 1), este é então executado. A Figura 75 mostra fotos do transformador pronto. Na Tabela 13 são mostrados os valores calculados das resistências e indutâncias do transformador. Detalhes desta obtenção destes parâmetros podem ser vistos na seção de Apêndice B.1.

Figura 75 – Imagens do transformador produzido



Fonte: produção do próprio autor, 2016

Tabela 13 – Parâmetros do transformador executado e perdas totais

Descrição	Variável	Valor
Resistência do primário	R_1	$0,55\text{ }\Omega$
Resistência do secundário	R_2	$4,74\text{ m}\Omega$
Indutância de magnetização	L_m	$28,37\text{ mH}$
Indutância de dispersão do primário	L_{lk_1}	$26,77\text{ }\mu\text{H}$
Indutância de dispersão do secundário	L_{lk_2}	$0,04\text{ }\mu\text{H}$
Perdas totais	$P_{d.t}$	$1,625\text{ W}$

Fonte: produção do próprio autor, 2016

É importante ainda notar que a indutância magnetizante obtida é da ordem de 1000 vezes maior que a indutância de dispersão do primário, o que também indica uma boa execução do transformador, garantindo que hajam menores picos de sobretensão sobre os semicondutores.

A.2.4 Dimensionamento dos semicondutores

Aqui, serão mostrados os passos para dimensionamento dos semicondutores do 2SFC. O conversor necessita de seis semicondutores, sendo dois interruptores (Sw_1 e Sw_2) e dois diodos (D_{in1} e D_{in2}) no estágio de entrada, além de dois diodos no estágio de saída (D_{o1} e D_{o2}), os quais são mostrados na Figura 3.

Na sequência, será mostrado o dimensionamento dos interruptores Sw_1 e Sw_2 .

A.2.4.1 Dimensionamento dos interruptores

Para o dimensionamento dos interruptores, devem ser analisados os esforços de tensão e corrente sobre os mesmos. No caso do Duplo-Forward, a tensão que recai sobre cada interruptor no momento do bloqueio é idealmente a tensão de entrada V_{in} , enquanto a corrente que flui sobre eles é a corrente do secundário refletida para o primário, onde seus valores médio e eficaz são dados pelas equações (A.27) e (A.28), respectivamente.

$$I_{sw1_{med}} = I_{sw2_{med}} = \frac{I_{o_{max}}}{N_t} D_{max} = 0,45 \text{ A} \quad (\text{A.27})$$

$$I_{sw1_{ef}} = I_{sw2_{ef}} = \frac{I_{o_{max}}}{N_t} \sqrt{D_{max}} = 0,671 \text{ A} \quad (\text{A.28})$$

Desta forma, foram escolhidos interruptores MOSFET IRF740 Vishay, com tensão de bloqueio máxima de 400 V e capacidade de corrente de 10 A.

A.2.4.2 Dimensionamento dos diodos do estágio de entrada

Os diodos D_{in1} e D_{in2} tem por objetivo apenas a desmagnetização do transformador de alta frequência. Desta forma, a corrente que flui sobre os mesmos é apenas a corrente de desmagnetização do transformador, que em geral é pequena. A tensão reversa máxima vista nestes diodos é a própria tensão de entrada V_{in} , que é aplicada em ambos durante a etapa 1 de operação do conversor. Os valores médio e eficaz da corrente de desmagnetização do que flui por D_{in1} e D_{in2} são mostrados nas equações (A.29) e (A.30), respectivamente.

$$I_{din1_{med}} = I_{din2_{med}} = \frac{V_{in_{max}} D_{max}}{L_m f_s} = 0,052 \text{ A} \quad (\text{A.29})$$

$$I_{din1_{ef}} = I_{din2_{ef}} = \frac{V_{in_{max}} \sqrt{D_{max}}}{L_m f_s} = 0,052 \text{ A} \quad (\text{A.30})$$

Assim, diodos MUR460, com tensão de bloqueio de 600 V e corrente média de 4,0 A foram escolhidos por questão de disponibilidade, apesar do sobredimensionamento, além de serem diodos rápidos, necessários nesta função.

A.2.4.3 Dimensionamento dos diodos do estágio de saída

No estágio de saída, os diodos D_{o1} e D_{o2} apresentam maior valor de tensão reversa durante as etapas 2 e 1, nesta ordem, sendo este valor igual a tensão de entrada refletida V_{in}/N_t . A condução destes diodos se dá durante as etapas de condução dos interruptores no caso do diodo de condução direta D_{o1} e durante as etapas de bloqueio dos interruptores no caso do diodo de roda livre D_{o2} . Sendo assim, as equações (A.31), (A.32), (A.33) e (A.34) mostram os valores calculados de corrente média e eficaz para os diodos D_{o1} e D_{o2} , sendo considerado o pior caso para o dimensionamento de ambos.

$$I_{do1_{med}} = I_{o_{max}} D_{max} = 5,4 \text{ A} \quad (\text{A.31})$$

$$I_{do1_{ef}} = I_{o_{max}} \sqrt{D_{max}} = 8,05 \text{ A} \quad (\text{A.32})$$

$$I_{do2_{med}} = I_{o_{max}} (1 - D_{min}) = 10,675 \text{ A} \quad (\text{A.33})$$

$$I_{do2_{ef}} = I_{o_{max}} \sqrt{1 - D_{min}} = 11,318 \text{ A} \quad (\text{A.34})$$

Foram utilizados então diodos do tipo Schottky MBR- 1545CT, com tensão máxima de bloqueio de 45 V e corrente máxima de 15 A.

A seguir, o dimensionamento térmico dos semicondutores será apresentado.

A.2.4.4 Dimensionamento Térmico dos Semicondutores

Para o dimensionamento térmico dos interruptores, foram obtidos os dados de catálogo do MOSFET IRF740 através do manual técnico do fabricante, dispostos da Tabela 14.

Tabela 14 – Dados de catálogo do MOSFET IRF740

Descrição	Variável	Valor
Resistência do MOSFET (em 90 °C)	r_{mos}	1,6 Ω
Tempo para entrada em condução	ton_{mos}	41 ns
Tempo para entrada em bloqueio	$toff_{mos}$	74 ns
Carga de recuperação reversa (diodo)	Q_{rr}	3,8 nC
Resistência térmica junção-ambiente	$R_{th_{mos.ja}}$	62 K/W
Resistência térmica junção-cápsula	$R_{th_{mos.jc}}$	1 K/W
Resistência térmica cápsula-dissipador	$R_{th_{mos.cs}}$	0,5 K/W

Fonte: produção do próprio autor, 2016

As perdas de condução para cada um dos interruptores então podem ser obtidas através da expressão (A.35), enquanto as perdas de comutação estão presentes na equação (A.36).

$$P_{cond_{mos}} = r_{mos} I_{sw_{ef}}^2 = 0,72 \text{ W} \quad (\text{A.35})$$

$$P_{com_{mos}} = \left(\frac{V_{in_{max}}}{2} \frac{I_{o_{max}}}{N_t} (t_{on_{mos}} + t_{off_{mos}}) \right) f_s + (Q_{rr} V_{in_{max}}) f_s = 1,998 \text{ W} \quad (\text{A.36})$$

Somadas as perdas de comutação e condução, tem-se que cada interruptor dissipa uma potência $P_{mos} = 2,718 \text{ W}$. Sabendo o valor de resistência térmica junção-ambiente (Tabela 14), obtém-se então a elevação de temperatura de cada interruptor:

$$\Delta T_{mos} = R_{th.mos_{ja}} P_{mos} = 168,54 \text{ } ^\circ\text{C} \quad (\text{A.37})$$

Considerando uma temperatura ambiente de $40 \text{ } ^\circ\text{C}$, tem-se então que cada interruptor poderá atingir uma temperatura de junção $T_{j_{mos}} = 208,54 \text{ } ^\circ\text{C}$, estando muito acima da temperatura máxima indicada pelo fabricante de $150 \text{ } ^\circ\text{C}$. Logo, será necessário o uso de dissipadores.

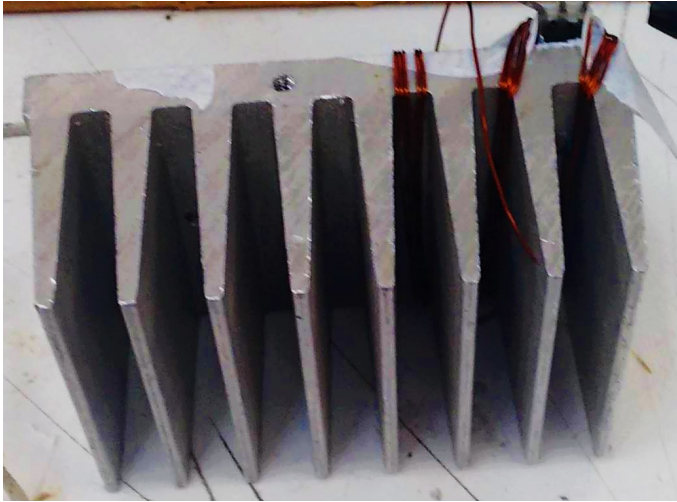
Fixando a temperatura máxima de junção dos interruptores em $T_{j_{max_{mos}}} = 135 \text{ } ^\circ\text{C}$ (10% menor que a indicada pelo fabricante), a temperatura ambiente em $T_a = 40 \text{ } ^\circ\text{C}$ e considerando que ambos interruptores serão acoplados ao mesmo dissipador, calcula-se em (A.38) o máximo valor de resistência térmica admitido para a escolha do dissipador.

$$R_{th.max.mos_{sa}} = \frac{T_{j.max_{mos}} - T_a}{2P_{mos}} - R_{th.mos_{jc}} - R_{th.mos_{cs}} = 15,977 \text{ K/W} \quad (\text{A.38})$$

Foi selecionado então um dissipador disponível de alumínio, liso, com dimensões $75 \times 35 \times 51 \text{ mm}$, o qual através de testes, estimou-se uma resistência térmica dissipador-ambiente menor que $R_{th.mos_{sa}} = 9,898 \text{ K/W}$, o que satisfaz os requisitos necessários para arrefecimento os interruptores. Os detalhes dos testes para determinação da resistência térmica do dissipador estão descritos na seção de Apêndice B.2. A Figura 76 mostra uma imagem do dissipador escolhido.

No caso dos diodos de entrada, foi obtida uma queda de tensão de condução direta $V_{fd_{in}} = 0,54 \text{ V}$ (em $T_j = 100 \text{ } ^\circ\text{C}$ e $I_f = 0,1 \text{ A}$). Logo, pode-se obter a potência dissipada por cada diodo,

Figura 76 – Imagem do dissipador dos interruptores



Fonte: produção do próprio autor, 2016

desconsiderando-se as perdas de comutação e as perdas na resistência de diodo, conforme equação (A.39):

$$P_{d_{in}} = V f_{d_{in}} I_{din1_{med}} = 0,028W \quad (A.39)$$

Tendo-se uma resistência térmica junção-ambiente, dada pelo fabricante, $R_{th.mos_{ja}} = 63 K/W$, a elevação de temperatura em cada diodo será menor que $2^{\circ}C$, dispensando o uso de dissipadores.

Por fim, será feito o dimensionamento térmico dos diodos de saída. A Tabela 15 mostra os dados de catálogo do Diodo MBR1545CT obtidos através do manual técnico do fabricante.

Aqui será adotada uma razão cíclica $D = 0,45$ para ambos os diodos ao invés de se adotar os valores de D_{max} para o diodo D_{o1} e D_{min} para o diodo D_{o2} , o que poderia levar a sobredimensionamentos, já que ambos os diodos sempre operam de forma complementar, podendo-se então garantir que quando um diodo está em condução com uma razão cíclica D , o outro estará em condução com uma razão cíclica

Tabela 15 – Dados de catálogo do diodo MBR1545CT

Descrição	Variável	Valor
Queda de tensão do diodo ¹	$V_{f_{do}}$	0,655 V
Pico na queda de tensão do diodo ¹	$V_{f_{pk_{do}}}$	0,72 V
Tensão reversa	$V_{r_{do}}$	27,17 V
Capacitância do diodo ²	C_{do}	340 pF
Corrente reversa ²	$I_{r_{do}}$	2.3 mA = 6 mA
Tempo de recuperação reversa	$t_{rr_{do}}$	30 ns
Tempo de recuperação direta	$t_{rf_{do}}$	320 ns
Corrente máxima reversa	$I_{RM_{do}}$	2.15 mA = 30 mA
Carga de recuperação reversa ³	$Q_{rr_{do}}$	0,45 nC

¹ $T_j = 125\text{ }^{\circ}\text{C}$ e $I_f = 12\text{ A}$

² $V_r = 27\text{ V}$

³ $Q_{rr_{do}} = 0.5 \cdot t_{r_{do}} \cdot I_{RM_{do}}$

Fonte: produção do próprio autor, 2016

$(1 - D)$.

As equações (A.40), (A.41) mostram os valores das perdas de condução direta e reversa para o diodo D_{o1} :

$$P_{do1_{cond.f}} = V_{f_{do}} I_{o_{max}} D = 3,537\text{ W} \quad (\text{A.40})$$

$$P_{do1_{cond.r}} = V_{r_{do}} I_{r_{do}} (1 - D) = 0,09\text{ W} \quad (\text{A.41})$$

As equações (A.42), (A.43) mostram os valores das perdas de comutação na entrada em condução e no bloqueio para o diodo D_{o1} :

$$P_{do1_{com.on}} = 0,5(V_{f_{pk_{do}}} - V_{f_{do}}) I_{o_{max}} t_{rf_{do}} f_s = 0,012\text{ W} \quad (\text{A.42})$$

$$P_{do1_{com.off}} = Q_{rr_{do}} V_{r_{do}} f_s = 1,223\text{ mW} \quad (\text{A.43})$$

A potência total dissipada pelo diodo D_{o1} é então dada por

(A.44):

$$P_{do1}=(P_{do1_{cond.f}}+P_{do1_{cond.r}})+(P_{do1_{com.on}}+P_{do1_{com.off}})=3,64 \text{ W} \quad (\text{A.44})$$

De maneira análoga, a potência dissipada pelo diodo D_{o2} pode ser obtida operando as equações de potência do D_{o1} com razão cíclica complementar, conforme mostrado nas equações (A.45), (A.46), (A.47), (A.48) e (A.49):

$$P_{do2_{cond.f}} = V_{f_{do}} I_{o_{max}} (1 - D) = 4,323 \text{ W} \quad (\text{A.45})$$

$$P_{do2_{cond.r}} = V_{r_{do}} I_{r_{do}} D = 0,073 \text{ W} \quad (\text{A.46})$$

$$P_{do2_{com.on}} = 0,5(V_{f_{pk_{do}}} - V_{f_{do}}) I_{o_{max}} t_{r_{f_{do}}} f_s = 0,012 \text{ W} \quad (\text{A.47})$$

$$P_{do2_{com.off}} = Q_{rr_{do}} V_{r_{do}} f_s = 1,223 \text{ mW} \quad (\text{A.48})$$

$$P_{do2}=(P_{do2_{cond.f}}+P_{do2_{cond.r}})+(P_{do2_{com.on}}+P_{do2_{com.off}})=4,41 \text{ W} \quad (\text{A.49})$$

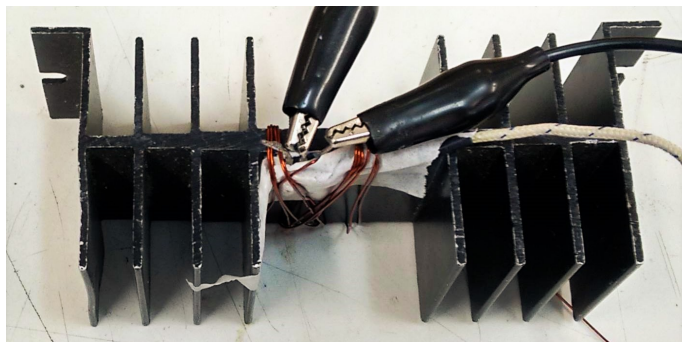
Colocando ambos os diodos D_{o1} e D_{o2} no mesmo dissipador, tem-se que a potência total dissipada pelos dois diodos é $P_{do} = 8,05 \text{ W}$. Sendo a resistência térmica junção-cápsula dos diodos $R_{th.do_{jc}} = 3 \text{ K/W}$ e a resistência térmica cápsula-dissipador $R_{th.do_{cs}} = 0,5 \text{ K/W}$, tem-se através de (A.50) a resistência térmica máxima dissipador-ambiente, considerando uma temperatura máxima de junção de $T_{dj_{max}} = 110 \text{ }^{\circ}\text{C}$ (aproximadamente 10% menor que a máxima suportada) e uma temperatura ambiente $T_a = 40 \text{ }^{\circ}\text{C}$:

$$R_{th.max.do_{sa}} = \frac{T_{dj_{max}} - T_a}{P_{do}} - R_{th.do_{jc}} - R_{th.do_{cs}} = 5,195 \text{ K/W} \quad (\text{A.50})$$

O dissipador selecionado possui dimensões $75 \times 35 \times 51 \text{ mm}$, o qual através de testes, estimou-se uma resistência térmica dissipador-

ambiente menor que $R_{th.mos_{sa}} = 5,149 \text{ K/W}$, o que satisfaz os requisitos necessários para arrefecimento os interruptores. Os detalhes dos testes para determinação da resistência térmica do dissipador estão descritos na seção B.2 dos Apêndices desta dissertação. A Figura 77 mostra uma imagem do dissipador escolhido.

Figura 77 – Imagem do dissipador dos diodos de saída



Fonte: produção do próprio autor, 2016

Finalizado o dimensionamento térmico dos componentes, passa-se então ao projeto dos circuitos de grampeamento utilizados.

A.2.5 Projeto dos circuitos de grampeamento

Circuitos de grampeamento e amortecimento, também conhecidos como *snnubbers*, são de grande auxílio em circuitos chaveados, pois reduzem os efeitos das indutâncias e capacitâncias parasitas sobre os semicondutores. O objetivo do uso de *snnubbers* aqui é controlar o valor máximo do pico de tensão sobre os diodos de saída D_{o1} e D_{o2} . É importante destacar que, no estágio de entrada do conversor, não é necessário a utilização de *snubbers* nos interruptores, uma vez que estes são grampeados naturalmente pelos diodos D_{in1} e D_{in2} , que permitem o escoamento da energia armazenada nas reatâncias do transformador.

Serão utilizados *snubbers* do tipo RCD visando grampear os picos de tensão em ambos os diodos D_{o1} e D_{o2} . A Tabela 16 mostra os valores utilizados para o projeto dos *snubbers*:

Tabela 16 – Definições e dados para projeto do *snubber*

Descrição	Variável	Valor
Corrente do snubber	$I_{s_{do}}$	12 A
Indutância de dispersão do secundário	L_{lk_2}	0,04 μH
Tensão de grampeamento	$V_{s_{clamp}}$	41 V
Variação da tensão de grampeamento	$\Delta V_{s_{do}}$	0,41 V
Tensão em regime (bloqueio) ¹	$V_{s_{ss}}$	25,808 V

¹Considerado 95% do valor ideal

Fonte: produção do próprio autor, 2016

Onde $I_{s_{do}}$ é a corrente que o diodo está conduzindo antes do bloqueio, L_{lk_s} é a indutância de dispersão do enrolamento secundário, $V_{s_{clamp}}$ é a tensão de grampeamento do secundário, $\Delta V_{s_{do}}$ é a variação da tensão de grampeamento permitida e $V_{s_{ss}}$ é a tensão de bloqueio do semiconductor após atingido o regime dentro do período. Nota-se que foi utilizado um fator de 0,95 nesta última, de forma a agregar as características não ideais de perda de tensão sobre os componentes.

A potência que deve ser dissipada no *snubber* é dada pela equação (A.51):

$$P_{s_{do1}} = \frac{1}{2} f_s L_{lk_s} I_{s_{do}}^2 \left(\frac{1}{1 - V_{s_{ss}}/V_{s_{clamp}}} \right) = 0,777 \text{ W} \quad (\text{A.51})$$

Desta forma, a resistência do *snubber* pode ser obtida em (A.52), enquanto a sua capacitância é dada por (A.53).

$$R_{s_{do1}} = \frac{V_{s_{clamp}}^2}{P_{s_{do1}}} = 2,163 \text{ k}\Omega \quad (\text{A.52})$$

$$C_{s_{do1}} = \frac{P_{s_{do1}}}{f_s V_{s_{clamp}} \Delta V_{s_{do}}} = 462,384 \text{ nF} \quad (\text{A.53})$$

Sendo assim, foram escolhidos um resistor de 2,2 k Ω – 3 W

(por disponibilidade), um capacitor de 470 nF e um diodo MUR460 (por disponibilidade) por ser do tipo ultra-rápido, além de suportar com folga os níveis de tensão e corrente exigidos.

Para o diodo de roda livre D_{o2} , são utilizados diferentes valores de $V_{s_{clamp}} = 34 \text{ V}$ e de $\Delta V_{s_{do}} = 0,34 \text{ V}$, sendo que os demais valores são os mesmos dispostos na Tabela 16. Tem-se então a potência dissipada pelo *snubber* do diodo D_{o2} em (A.54).

$$P_{s_{do2}} = \frac{1}{2} f_s L_{lk_s} I_{s_{do}}^2 \left(\frac{1}{1 - V_{s_{ss}}/V_{s_{clamp}}} \right) = 1,195 \text{ W} \quad (\text{A.54})$$

Desta forma, os valores de resistência e capacitância do *snubber* são obtidos através de (A.55) e (A.56) respectivamente.

$$R_{s_{do2}} = \frac{V_{s_{clamp}}^2}{P_{s_{do2}}} = 967,07 \, \Omega \quad (\text{A.55})$$

$$C_{s_{do2}} = \frac{P_{s_{do2}}}{f_s V_{s_{clamp}} \Delta V_{s_{do}}} = 1,034 \, \mu\text{F} \quad (\text{A.56})$$

Logo, foram escolhidos um resistor de $1 \text{ k}\Omega - 3 \text{ W}$, um capacitor de $1 \, \mu\text{F}$ e também um diodo MUR460 (por disponibilidade).

A próxima etapa de projeto consiste no dimensionamento do circuito de comando, que será descrita a seguir.

A.2.6 Dimensionamento do circuito de comando

Agora, será mostrado o caminho percorrido para o dimensionamento do circuito de comando. A seguir, será demonstrado o cálculo para obtenção da potência necessária para acionamento dos interruptores e o cálculo da resistência de *gate*.

A.2.6.1 Potência do circuito de comando e obtenção da resistência de *gate*

Inicialmente, serão calculados a potência requerida pelo *gate* do MOSFET para o correto acionamento dos interruptores. Considerando

que o valor da máxima tensão de gate seja $V_{gs_{max}} = 15 \text{ V}$ e sabendo que a carga necessária para carregar a capacitância do MOSFET é $Q_g = 63 \text{ nC}$, dada pelo fabricante, tem-se a potência requerida para o acionamento do interruptor em (A.57).

$$P_{gate} = V_{gs_{max}} Q_g f_s = 0,095 \text{ W} \quad (\text{A.57})$$

A corrente de pico necessária para acionamento do MOSFET é dada pela relação (A.58), onde foi considerado um $V_{gs_{mos}} = 15 \text{ V}$, e os valores $C_{iss_{mos}} = 1400 \text{ pF}$ e $t_{gr} = 25 \text{ ns}$ foram retirados do catálogo do fabricante.

$$I_{gpk_{mos}} = C_{iss_{mos}} \frac{V_{gs_{mos}}}{t_{gr}} = 0,84 \text{ A} \quad (\text{A.58})$$

Por fim, tendo o tempo $t_{on_{mos}} = 41 \text{ ns}$ através dos dados do fabricante, pode-se obter a resistência de *gate* através da equação (A.59):

$$R_g = \frac{t_{on_{mos}}}{2,2C_{iss_{mos}}} = 13,31 \text{ } \Omega \quad (\text{A.59})$$

Assim, foi escolhido um resistor $R_g = 15 \text{ } \Omega$.

A.2.6.2 Escolha dos optoacopladores

O conversor Duplo-Forward possui dois interruptores no estágio de entrada, sendo que o Sw_1 não possui o terminal de *source* conectado à referência da fonte V_{in} . Desta forma, para garantir uma tensão V_{gs} adequada, há a necessidade de que o comando deste interruptor seja isolado. Ainda, apesar do interruptor Sw_2 possuir o terminal de *source* conectado à referência da fonte V_{in} , por motivos práticos, esta não é a referência utilizada nas estratégias de controle. Desta forma, optou-se em também realizar o comando do interruptor Sw_2 de forma isolada, para que fossem evitados futuros problemas.

O isolamento do circuito de comando é feito através de optoacopladores. Foram aqui escolhidos os optoacopladores Fairchild FOD3180, que possuem capacidade de corrente de saída de 2 A , podem operar com frequências até 250 kHz , além de possuírem um atraso de

transporte de até 200 *ns*.

O optoacoplador recebe os sinais de PWM gerados pela estratégia de controle através de um fotodiodo. O fabricante recomenda que a corrente em cada fotodiodo fique entre 10 e 16 *mA*. Sabendo que o sinal de PWM possui uma tensão de 3,3 *V*, foram colocados dois resistores de 100 Ω em série (200 Ω no total) para cada fotodiodo, sendo um destes resistores também utilizados em um Filtro Passa Baixas LPF, que será comentado na seção A.2.6.3. Desta forma, tem-se uma corrente nos fotodiodos de 16,5 *mA*, um pouco acima da recomendada, mas que garante uma boa relação Sinal-Ruído.

A saída do optoacoplador é conectada diretamente em série com o resistor de gate para o acionamento dos interruptores. Apesar de consumir pouca potência em termos médios, o interruptor demanda altos picos de corrente. Além disso, a alimentação do estágio de saída do optoacoplador também deve ser isolada, para garantir o isolamento do circuito. Desta forma, a seção A.2.6.4 discute a escolha da fonte isolada para a aplicação e como fornecer os picos de corrente demandados pelo interruptor.

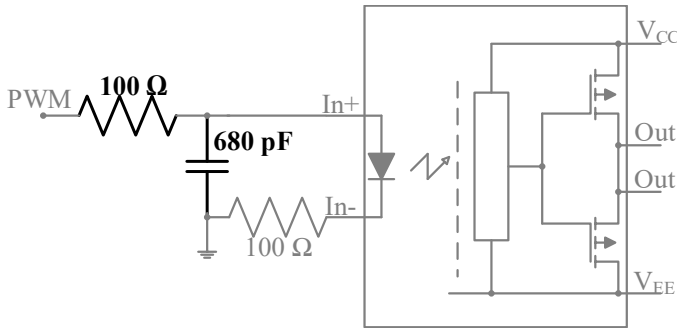
A.2.6.3 Filtro Passa Baixas para condicionamento dos optoacopladores

Conforme dito na seção A.2.6.2, foi adicionado um LPF, constituído de um circuito RC, para efetuar o desacoplamento dos sinais de PWM gerados pelo circuito de controle que são mandados ao optoacoplador.

Como a frequência de chaveamento é $f_s = 100 \text{ kHz}$, o filtro deve ser posicionado de forma a interferir o mínimo possível nas frequências próximas a f_s , de forma a não distorcer o sinal de PWM gerado. Assim, este será posicionado ao menos uma década acima de f_s . O próprio resistor de 100 Ω do circuito do optoacoplador será utilizado em série com o sinal de PWM, enquanto que foi escolhido um capacitor de 680 *pF* a ser colocado em paralelo com o sinal de PWM. Logo, tem-se uma frequência de corte $f_{c_{LPF}} = 2,341 \text{ MHz}$ no filtro, o que garante que o mesmo não irá exercer muita influência sobre o sinal de comando gerado. A Figura 78 ilustra o esquemático de utilização

do filtro.

Figura 78 – Esquemático do LPF na entrada do optoacoplador. Os componentes utilizados encontram-se destacados em preto.



Fonte: produção do próprio autor, 2016

O autor considerou importante a inclusão deste tópico, por este ter sido fundamental para a remoção de componentes de alta frequência que faziam com que o sinal de gatilho no optoacoplador se tornasse altamente instável, provocando chaveamentos indesejáveis no interruptor. Após a inclusão deste filtro, houveram melhoras significativas.

A.2.6.4 Fontes auxiliares isoladas para o circuito de comando

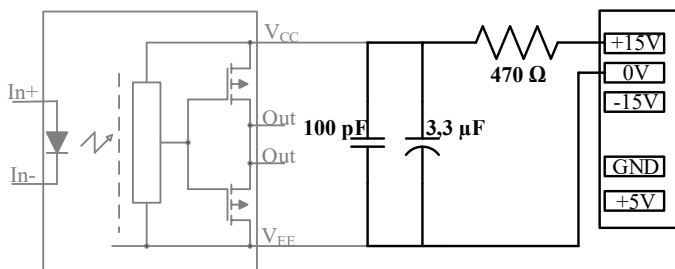
Conforme mostrado na equação (A.57), a potência requerida pelo circuito de acionamento ($0,095\text{ W}$) é pequena. Desta forma, optou-se por utilizar fontes CC-CC isoladas comerciais Murata NMA0515SC, uma para cada interruptor, adequando uma tensão de entrada de 5 V para 15 V com isolamento. Além disso, sua capacidade de corrente de saída é de 33 mA , podendo assim fornecer até $0,5\text{ W}$, sendo esta potência um valor maior que o necessário para acionamento dos interruptores.

Destaca-se que, apesar da potência média consumida pelos interruptores ser baixa, estes demandam altos picos de corrente ($0,84\text{ A}$ conforme equação (A.58)), que não podem ser fornecidos pela fonte

NMA0515SC. Assim, caso a fonte NMA0515SC seja conectada diretamente aos terminais do optoacopladores para acionamento dos interruptores, esta irá queimar por sobrecarga, fato que é destacado pelo autor, uma vez que estas fontes possuem um custo relativamente elevado e tiveram que ser substituídas neste projeto devido a este problema.

Buscando resolver o problema dos picos de corrente, foi adicionado de um resistor de $470\ \Omega$ em série entre a saída da fonte NMA0515SC e do optoacoplador, visando a limitar a corrente em $32\ mA$. Ainda, foram adicionados um capacitor eletrolítico de $3,3\ \mu F$ e um cerâmico de $100\ pF$ em paralelo com o optoacoplador, de forma que estes serão os responsáveis por suprir os picos de corrente demandados pelo interruptor, e não mais as fontes NMA0515SC. O esquemático aqui descrito é mostrado na Figura 79.

Figura 79 – Esquemático da alimentação isolada do optoacoplador, com os componentes utilizados destacados em preto.



Fonte: produção do próprio autor, 2016

Através da simples adição dos capacitores, foi possível então suprir os picos de corrente demandados para o acionamento dos interruptores.

Serão abordadas na sequência as proteções e os circuitos auxiliares utilizados neste conversor.

A.2.7 Dimensionamento das proteções e circuitos auxiliares

Afim de garantir a proteção do conversor, proteções simples por hardware foram implementadas, como a utilização de fusível e de

um termistor do tipo NTC.

O dimensionamento do fusível utiliza a equação (A.60):

$$I_{fuse.calc} = 1,25 I_{in_{max}} = 0,726 \text{ A} \quad (\text{A.60})$$

Assim, foi escolhido um fusível de 1 A, colocado na entrada do conversor.

Já o Termistor NTC é calculado definindo-se a máxima corrente de partida I_{rush} permitida no conversor. Porém, também deve ser observado o valor de resistência do NTC em regime permanente, que tende a ser maior conforme os requisitos de corrente de partida se tornam mais restritivos. Foi então definido um valor de $I_{rush} = 31 \text{ A}$, que leva ao valor de resistência aproximado do NTC à temperatura ambiente, conforme (A.61):

$$R_{NTC.calc} = \frac{V_{in_{max}}}{I_{rush}} = 10,516 \Omega \quad (\text{A.61})$$

Foi definida então a utilização do NTC-10, que possui uma resistência de 10Ω a uma temperatura de 25°C . Com pouco tempo de operação, a corrente que passa pelo NTC o aquece, fazendo com que sua resistência diminua. O NTC escolhido, por exemplo, possui uma resistência $R_{NTC.3A} = 0,395 \Omega$ após atingir o regime térmico com uma corrente de 3 A. Considerando que a relação entre sua resistência e a corrente é linear, (A.62) apresenta uma estimativa grosseira de qual deve ser a resistência do NTC dada a corrente de operação do conversor:

$$R_{NTC.ss} = R_{NTC.3A} \frac{3A}{I_{in_{max}}} = 2,039 \Omega \quad (\text{A.62})$$

Com o dimensionamento das proteções feito, encerram-se as etapas de projeto do estágio de potência. Detalhes de dimensionamento de outros circuitos auxiliares, como os LEDs indicativos, de ensaios de componentes, dentre outros, estão descritos no Apêndice B.

Apêndice – B INFORMAÇÕES COMPLEMENTARES DO ESTÁGIO DE POTÊNCIA DO CONVERSOR

B.1 OBTENÇÃO DOS PARÂMETROS DO TRANSFORMADOR DE ALTA FREQUÊNCIA

Para que fossem obtidos os parâmetros do transformador, ensaios de circuito aberto e de curto-circuito foram realizados no mesmo, além da medição da resistência dos enrolamentos.

Para realizar as medições, foi utilizada uma ponte RLC HIOKI 3511-50. O primeiro passo a ser feito na utilização da ponte RLC é sua correta calibração. Deve-se manter as duas ponteiros de medição separadas (circuito aberto), com a garra dos seus contatos fechada e devidamente posicionadas sobre uma bancada para que não se movam acidentalmente. Com esta montagem, aperta-se o botão OPEN, que realizará a calibração de circuito aberto da ponte RLC, o que leva apenas alguns segundos. Em seguida, faz-se a calibração de curto-circuito, que deve ser feito com o auxílio de um fio condutor ou estanho. Ambas as ponteiros deverão ser conectadas ao condutor, sendo que estas devem estar o mais próximo possível uma da outra, de forma a minimizar a resistência, formando um curto-circuito quase ideal. Feito este arranjo, aperta-se o botão SHORT, que fará a calibração de curto-circuito da ponte RLC.

Estando a ponte RLC devidamente calibrada, foram utilizadas as configurações da ponte conforme a Tabela 17.

Para a medição de resistência dos enrolamentos, foi selecionada a medição de resistência R (Ω) através do botão PARAMETER. Foram então conectadas as ponteiros da ponte RLC nos terminais do enrolamento primário do transformador, resultando na medição do valor de resistência deste enrolamento. O mesmo procedimento foi realizado para o enrolamento secundário. Os valores médios de resistência encontram-se na Tabela 13, presente na seção A.2.3.2.

Com relação aos parâmetros indutivos do transformador, estes não podem ser medidos diretamente. Sua obtenção é feita através de medições obtidas em ensaios, que então através de alguns cálculos, levam aos valores de indutância magnetizante e de indutâncias de dis-

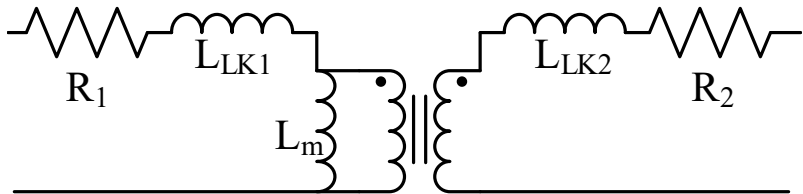
Tabela 17 – Dados de catálogo do diodo MBR1545CT

Descrição	Configuração
MODE	NORM
FREQ	1 kHz
LEVEL	1 V
SPEED	SLOW
CIRCUIT	AUTO
RANGE	AUTO
TRIG	default

Fonte: produção do próprio autor, 2016

persão. Desta forma, o modelo do transformador que será utilizado é mostrado na Figura 80.

Figura 80 – Representação do transformador não ideal



Fonte: produção do próprio autor, 2016

Para a medição das indutâncias do transformador, foi selecionada a medição de indutância L (H). Primeiramente, realizou-se o ensaio de circuito aberto no transformador. Neste ensaio, o enrolamento secundário é mantido em aberto, e as ponteiras da ponte RLC são conectadas nos terminais do enrolamento primário, fornecendo um valor de indutância de circuito aberto L_{oc} . Em seguida, são feitos os ensaios de circuito-circuito. Primeiramente, curto-circuitando os terminais do enrolamento secundário e conectando as ponteiras da ponte

RLC nos terminais do enrolamento primário, tem-se agora a medição da indutância de circuito-circuito L_{sc1} . Repetindo o processo para o outro enrolamento, tem-se a indutância de curto-circuito L_{sc2} . Os valores medidos de L_{oc} , L_{sc1} e L_{sc2} estão presentes da Tabela 18.

Tabela 18 – Medições de indutância do transformador

Descrição	Variável	Valor
Indutância de Circuito Aberto	L_{oc}	28,40 mH
Indutância de Curto-Circuito 1	L_{sc1}	28,20 μH
Indutância de Curto-Circuito 2	L_{sc2}	0,04 μH

Fonte: produção do próprio autor, 2016

Através do modelo mostrado na Figura 80, pode ser montado o seguinte sistema de equações não-lineares:

$$(B.1) \quad \begin{cases} L_{oc} = L_{lk1} + L_m \\ L_{sc1} = L_{lk1} + (L_m // L_{lk2})' = L_{lk1} + \frac{L_m \cdot (L_{lk2} N_t^2)}{L_m + (L_{lk2} N_t^2)} \\ L_{sc2} = L_{lk2} + (L_m // L_{lk1})' = L_{lk2} + \frac{1}{N_t^2} \frac{L_m L_{lk1}}{L_m + L_{lk1}} \end{cases}$$

Os resultados da resolução do sistema obtidos estão mostrados na Tabela 13, presente na seção A.2.3.2. O código em MATLAB® para resolução do sistema pode ser encontrado no Anexo A.1. Destaca-se ainda que o sistema em questão é mal condicionado, o que exige que sejam utilizadas técnicas para contornar tal problema. Neste trabalho, foi feita apenas a fixação da indutância L_{lk2} , que levava a erros de convergência.

B.2 OBTENÇÃO DA RESISTÊNCIA TÉRMICA DOS DISSIPADORES

A resistência térmica do dissipador pode ser obtida através de testes muito simples. Foi utilizado um diodo como fonte de calor e medida a diferença de temperatura entre a temperatura da cápsula do diodo e a temperatura ambiente. Os equipamentos utilizados foram:

- 1 Fonte CC – 15 V – 1 A.
- 2 Multímetros testados ICEL para medição de temperatura.
- 2 Multímetros testados Minipa para medições de grandezas elétricas.
- 2 Termopares (testados juntamente com os multímetros).
- 2 Resistores de 56 Ω – 200 W.
- 1 Diodo MUR630CT-D (U-620) como fonte de potência.
- 1 Dissipador a ser avaliado.

O procedimento de teste adotado é listado a seguir:

- (i) Ligar os dois resistores em paralelo.
- (ii) Conectar o diodo em série com os resistores, formando um circuito RD.
- (iii) Posicionar o termopar o mais próximo possível da capsula do diodo e fixá-lo.
- (iv) Posicionar e fixar o outro termopar para que fique em contato com o ar, para medição da temperatura ambiente.
- (v) Conectar os termopares aos multímetros ICEL.
- (vi) Conectar os 2 multímetros Minipa, um em série com o circuito para medição da corrente do diodo I_d , e outro em paralelo com o diodo para medição da queda de tensão V_f .

- (vii) Ligar a fonte de tensão.
- (viii) Aguardar 50 minutos, com a sala fechada e temperatura controlada, para atingir equilíbrio térmico.
- (ix) Fazer a leitura de tensão V_f e corrente I_d no diodo.
- (x) Fazer leitura das temperaturas do diodo T_c e do ambiente T_a .

Realizados os procedimento descritos acima, deve-se então calcular a potência dissipada pelo diodo P_d e o gradiente de temperatura cápsula-ambiente ΔT , conforme as equações:

$$P_d = V_f I_d \quad (\text{B.2})$$

$$\Delta T = T_c - T_a \quad (\text{B.3})$$

Agora, considerando que a resistência térmica capsula-dissipador é $R_{th_{cs}} = 0,5 K/W$, tem-se finalmente a resistência térmica do dissipador:

$$R_{th_{sa}} = \frac{\Delta T}{P_d} - R_{th_{cs}} \quad (\text{B.4})$$

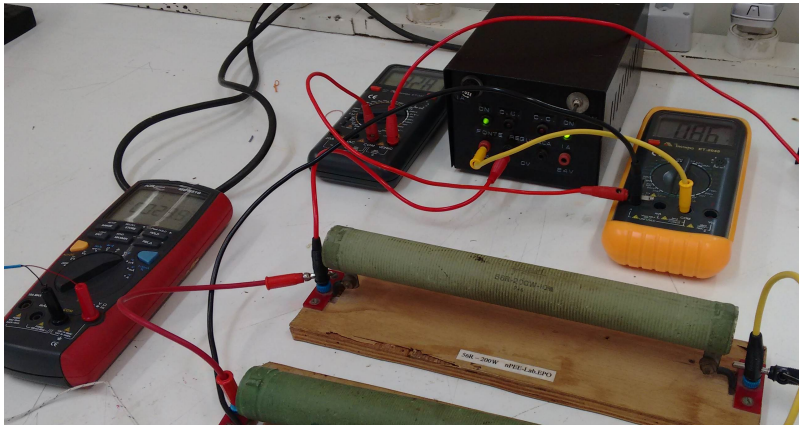
É importante destacar que o ideal é que sejam feitas várias medidas de tensão, corrente e temperatura, em várias configurações diferentes, para que se possa então avaliar a média e desvio padrão dos dados obtidos e a partir disto montar a função densidade de probabilidade da resistência térmica, porém este estudo é altamente custoso, além de estar fora do escopo deste trabalho. Desta forma, foram assumidos os valores de resistência térmica encontrados a partir da metodologia aqui mostrada.

B.2.1 Resultados do dissipador dos interruptores

O dissipador utilizado para os interruptores é feito de alumínio com dimensões $75 \times 35 \times 51 \text{ mm}$. Após a configuração do experimento, foram coletadas as medidas para o cálculo da resistência térmica do dissipador. A Figura 81 mostra a montagem dos testes com os valores

de tensão, corrente e temperatura ambiente medidos. Todos os dados coletados estão dispostos na Tabela 19:

Figura 81 – Imagem do ensaio do dissipador dos interruptores



Fonte: produção do próprio autor, 2016

Tabela 19 – Medições do ensaio do dissipador dos interruptores

Descrição	Configuração
V_f	0,828 V
I_d	0,860 A
T_c	29,9 °C
T_a	24,2 °C

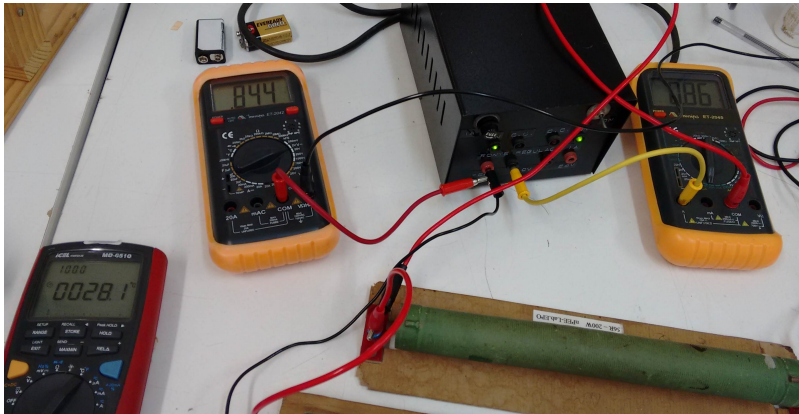
Fonte: produção do próprio autor, 2016

A partir destes dados, foi então estimada a resistência térmica do dissipador $R_{th_{sa}} = 7,505 \text{ K/W}$.

B.2.2 Resultados do dissipador dos diodos de saída

Para os diodos, foi utilizado um dissipador com dimensões $121 \times 49 \times 40 \text{ mm}$. Após a configuração do experimento, novamente foram coletadas as medidas para o cálculo da resistência térmica do dissipador. A Figura 82 mostra a montagem dos testes com os valores de tensão, corrente e temperatura na cápsula medidos. Todos os dados coletados estão dispostos na Tabela 20:

Figura 82 – Imagem do ensaio do dissipador dos diodos



Fonte: produção do próprio autor, 2016

Tabela 20 – Medições do ensaio do dissipador dos diodos

Descrição	Configuração
V_f	0,844 V
I_d	0,860 A
T_c	28,1 °C
T_a	24,0 °C

Fonte: produção do próprio autor, 2016

A partir destes dados, foi então estimada a resistência térmica do dissipador $R_{th_{sa}} = 5,149 \text{ K/W}$.

B.3 LEDS INDICADORES DE OPERAÇÃO

No conversor, foram inseridos 3 LEDs para facilitar o manuseio e operação, sendo um para indicar que há tensão no barramento CC, representado pela fonte V_{in} , outro para indicar que há alimentação nos circuitos de comando do conversor, e um último como indicativo de tensão na saída do conversor. Todos os LEDs operam com uma corrente $I_{LED} = 5 \text{ mA}$, e serão consideradas nulas as suas quedas de tensão. Assim, é necessário apenas o cálculo dos resistores para limitação da corrente dos LEDs.

Para o LED indicativo do barramento CC, tem-se:

$$R_{LED_{vin}} = \frac{V_{in_{max}}}{I_{LED}} = 65,2 \text{ k}\Omega \quad (\text{B.5})$$

$$P_{LED_{vin}} = R_{LED_{cc}} I_{LED}^2 = 1,63 \text{ W} \quad (\text{B.6})$$

Por falta de disponibilidade de resistores com esta faixa de resistência e potência maior que 1 W , foi selecionado um resistor de $100 \text{ k}\Omega$ e 1 W , que irá operar com uma corrente ligeiramente menor, e consequentemente com um brilho de menor intensidade.

Para o LED indicativo do circuito de comando, tem-se:

$$R_{LED_{com}} = \frac{5V}{I_{LED}} = 1 \text{ k}\Omega \quad (\text{B.7})$$

$$P_{LED_{com}} = R_{LED_{com}} I_{LED}^2 = 0,025 \text{ W} \quad (\text{B.8})$$

Logo, foi selecionado um resistor de $1 \text{ k}\Omega$ e $1/4 \text{ W}$. Finalmente, para o LED indicativo da saída do conversor, tem-se:

$$R_{LED_{out}} = \frac{V_{o_{max}}}{I_{LED}} = 2 \text{ k}\Omega \quad (\text{B.9})$$

$$P_{LED_{out}} = R_{LED_{out}} I_{LED}^2 = 0,05 \text{ W} \quad (\text{B.10})$$

Escolheu-se então um resistor de $2\text{ k}\Omega$ e $1/4\text{ W}$.

B.4 DEMAIS RESULTADOS DO CONVERSOR DUPLO-FORWARD

Nesta seção de apêndice, serão mostrados os demais resultados do conversor Duplo-Forward, apenas com o objetivo de complementar àqueles já mostrados, e servir de apoio para o desenvolvimento de outros trabalhos que possam se beneficiar deste projeto.

A seguir na seção B.4.1, serão mostrados os resultados juntamente de alguns comentários do projeto da PCI do conversor Duplo-Forward.

B.4.1 Resultados do projeto da placa de circuito impresso

A Placa de Circuito Impresso (PCI) do conversor 2SFC passou por diversas versões até chegar no seu estágio final de desenvolvimento. Isto porque o *layout* da PCI é decisivo para reduzir os ruídos que podem atrapalhar a operação do conversor. Logo, deve-se atentar muito a esta parte do desenvolvimento e gastar quanto tempo for necessário durante sua etapa de projeto.

Dentre as principais ações que devem ser tomadas no projeto de uma PCI, destacam-se:

- Prever a utilização de componentes durante a etapa de projeto: mesmo quando não se tem certeza da utilização de certos componentes, é fundamental haja um espaço reservado para os mesmos. A exemplo dos *snubbers* dos interruptores, que foram incluídos no projeto da PCI, mesmo estes sendo dispensáveis em teoria. Isto porque, caso não haja espaço reservado na PCI e a utilização de tais componentes se mostre necessária na prática, podem ser necessárias adaptações técnicas visando sanar o problema de espaço dos componentes, porém podendo gerar outros efeitos indesejados.
- Adição de capacitores de desacoplamento: este item tem relação com o item anterior. Deve-se deixar espaço para a inclusão de

capacitores de desacoplamento entre as referências do circuito e a carcaça, além da inclusão de um capacitor entre os terminais do primário e do secundário do transformador. Tais capacitores foram incluídos no projeto da PCI, porém não se mostraram úteis na prática neste caso, o que dispensou sua utilização posterior.

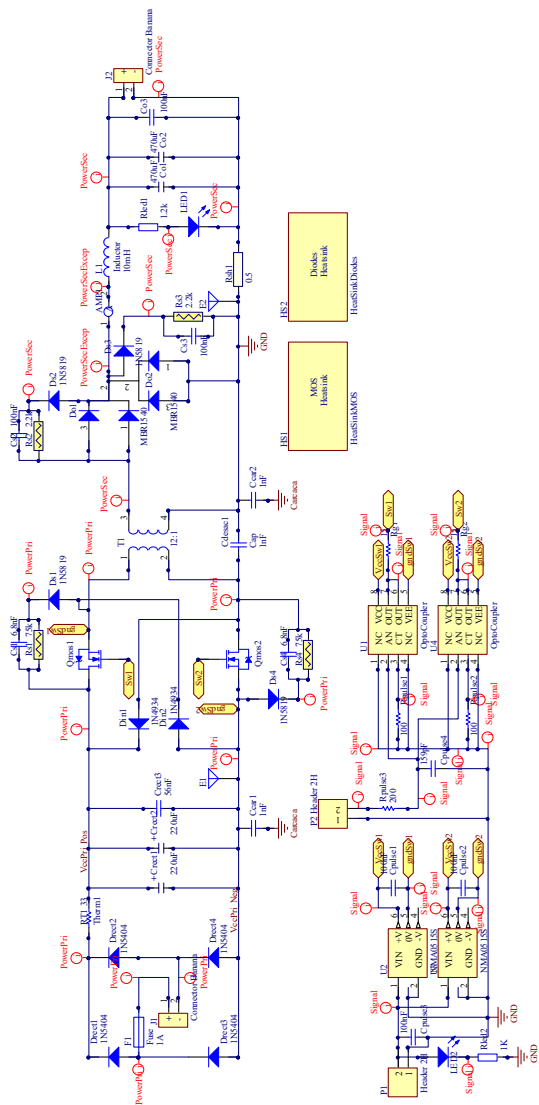
- Inclusão de pontos de medição no circuito: este item é fundamental quando se quer aprofundar os estudos no conversor em questão, já que a medição de corrente e tensão em semicondutores, por exemplo, é em geral custosa quando não é planejada. Desta forma, deve-se planejar já na etapa de projeto da PCI os pontos de medição de tensão e corrente de todos os componentes desejados. Porém, vale destacar que a adição de pontos de medição pode gerar reatâncias parasitas desnecessárias. Desta forma, sua utilização também deve ser usada com parcimônia. Assim, este trabalho optou pela adição de poucos pontos de medição, já que o estudo do conversor não está entre os objetivos do mesmo, visando reduzir os efeitos do ruído, em troca da obtenção de poucos resultados experimentais do conversor.
- Inserção de capacitores cerâmicos na PCI: A utilização de capacitores cerâmicos é sempre benéfica na prática, por estes capacitores possuírem menor resistência série e menor impedância em altas frequências. Quando a sua utilização for inviabilizada devido a necessidade de altas capacitâncias, deve-se então optar pela utilização de capacitores cerâmicos de menor capacidade em paralelo com àquele projetados. Neste trabalho, tem-se o exemplo dos capacitores do filtro do retificador. Além da utilização de dois capacitores eletrolíticos de $220\mu F$ cada, foi utilizado ainda um capacitor cerâmico na ordem de nF em paralelo.
- Posicionamento de *Snubbers* junto dos semicondutores: procure posicionar os circuitos de *snubber* o mais próximo possível dos semicondutores em questão, visando maximizar o seu efeito.
- Evitar o *Ground Bounce*: a variação da área enlaçada pela cor-

rente durante os chaveamentos pode provocar oscilações na tensão de referência de circuito. Estas oscilações ocorrem de forma periódica durante os chaveamentos, e por estarem presentes na referência do circuito, são extremamente difíceis de serem removidas ou filtradas. Este problema não foi observado durante a etapa de projeto e se faz presente no conversor aqui projetado. Seus efeitos acarretam em instabilidade na comunicação do DSC com o computador, o que dificultava a leitura de variáveis em tempo real. Apesar de haver uma pequena melhora nos efeitos do *Ground Bounce* após a adição de alguns capacitores para a referência e de adaptações nas trilhas, o problema não foi completamente sanado. Mais informações sobre o *Ground Bounce* podem ser vistas em [86].

- Proximidade no posicionamento dos componentes sem comprometimento: uma dica comum entre os projetistas de PCI é alocar os componentes o quão próximo possível uns dos outros. Em adição, é importante que este posicionamento também leve em consideração a facilidade de manuseio após a implementação da PCI. Desta forma, posicione os componentes o mais próximo possível uns dos outros, mas evitando dificultar o acesso dos mesmos às ponteiros de osciloscópio, por exemplo. É uma dica simples, mas que pode fazer muita falta caso seja esquecida.

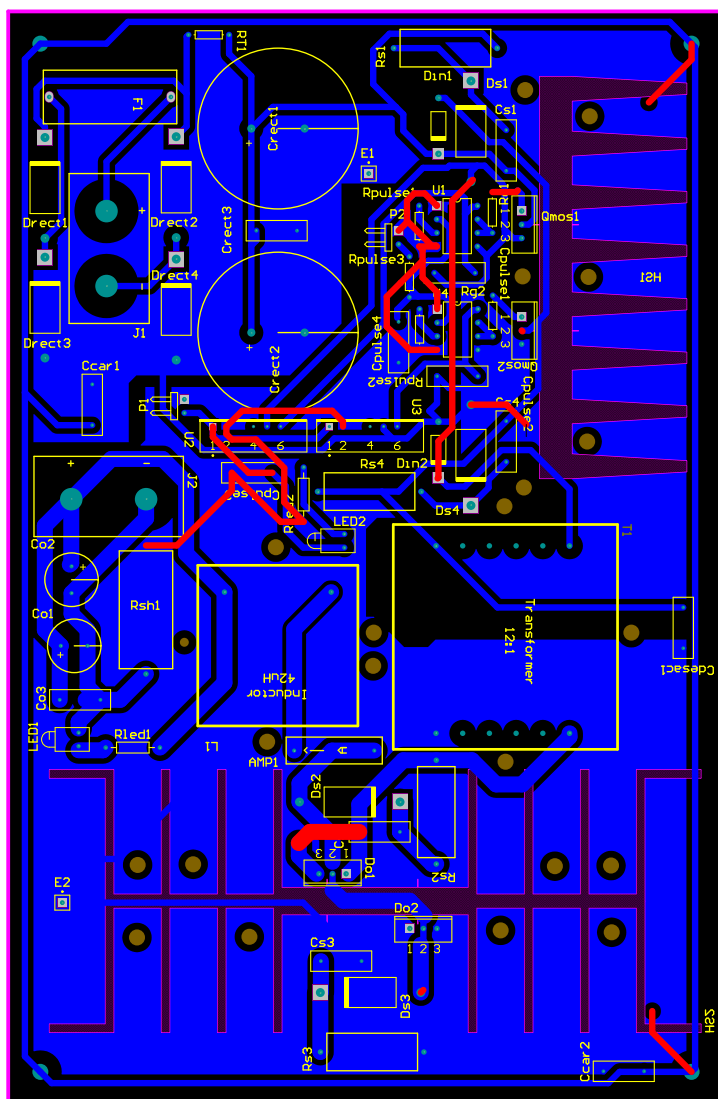
O esquemático e o projeto da PCI utilizada no conversor Duplo-Forward estão presentes nas Figuras 83 e 84.

Figura 83 – Esquemático da PCI do Retificador e do Conversor Duplo-Forward



Fonte: produção do próprio autor, 2016

Figura 84 – Projeto da PCI do Retificador e do Conversor Duplo-Forward



Fonte: produção do próprio autor, 2016

B.4.2 Resultados complementares do conversor

Nesta seção, são mostradas as formas de onda de tensão e corrente nos semicondutores e no transformador, além das formas de onda da potência média instantânea nos semicondutores, obtidas por simulação. Ainda, são mostradas as formas de onda obtidas experimentalmente das tensões nos interruptores e nos diodos D_{o1} e D_{o2} . Todos os resultados são obtidos com a operação em regime permanente.

Inicialmente, a Figura 85 mostra as formas de onda da corrente de *gate* dos interruptores e da tensão V_{gs} aplicada obtidas através de simulações. Na Figura 86 tem-se a forma de onda da tensão V_{gs} obtida experimentalmente. Ambas possuem características muito semelhantes, a não ser pelo fato da tensão experimental ter magnitude menor (11,8 V) que a tensão mostrada na simulação (13,8 V), o que se deve às não idealidades.

Já a Figura 87 traz os resultados simulados de corrente nos semicondutores Sw_1 , Sw_2 , D_{in1} , D_{in2} , D_{o1} e D_{o2} . Estas correntes não foram obtidas experimentalmente devido à ausência de pontos de medição de corrente no protótipo do conversor, conforme comentado na seção B.4.1.

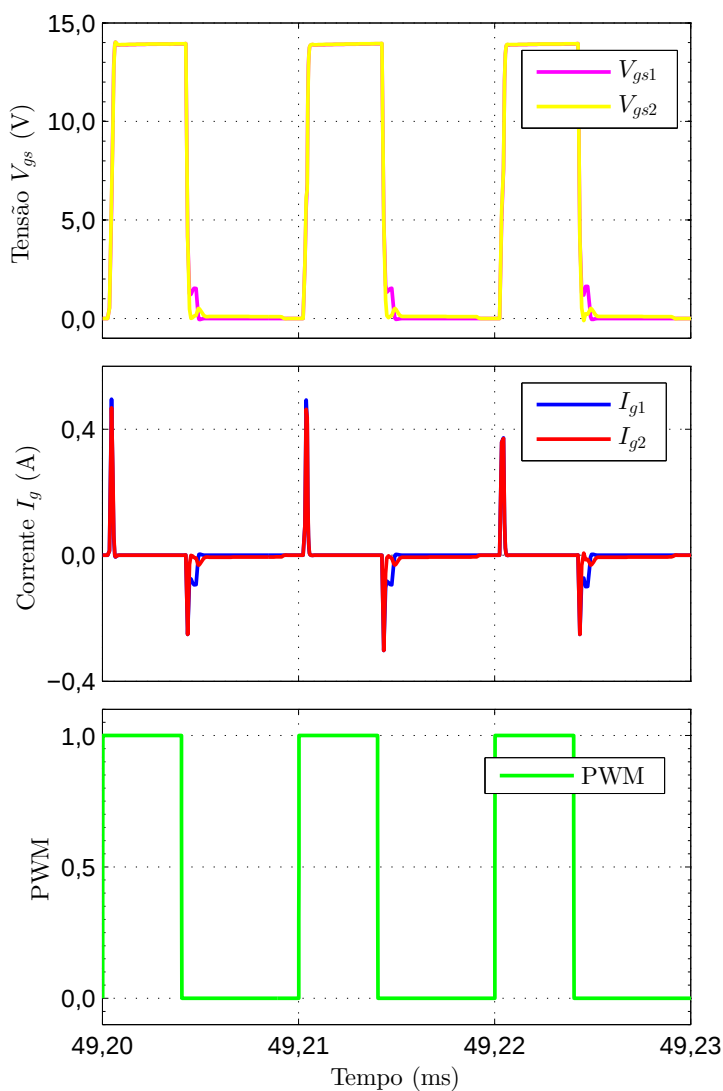
As tensões nos semicondutores também são muito importantes no projeto. A Figura 88 traz os resultados destas tensões obtidos por simulações. Estes resultados também foram obtidos experimentalmente para os interruptores e para os diodos de saída D_{o1} e D_{o2} , os quais são apresentados nas Figuras 89 e 90.

Na Figura 91 são vistas as formas de onda simuladas das tensões e correntes no transformador.

Na Figura 92, tem-se a forma de onda da medição de dois pontos de referência do circuito (medição entre dois pontos de “terra”), que teoricamente deveriam estar sobre o mesmo potencial. Este resultado mostra o efeito do *Ground Bounce* no circuito. Nota-se a presença de uma oscilação de alta frequência e com amplitude de cerca de 0,9 V, a qual é rapidamente amortecida.

Por fim, tem-se na Figura 93 as potências médias instantâneas, obtidas por simulação, dissipadas em cada semiconductor do circuito, as quais se aproximam dos valores calculados presentes na seção A.2.4.4.

Figura 85 – Formas de onda da tensão V_{gs} (rosa e amarelo) e da corrente de *gate* (azul e vermelho) dos interruptores



Fonte: produção do próprio autor, 2016

Figura 87 – Formas de onda simuladas das correntes nos semicondutores: I_{sw1} (rosa), I_{sw2} (amarelo), I_{din1} (azul claro), I_{din2} (preto), I_{do1} (azul) e I_{do2} (vermelho)

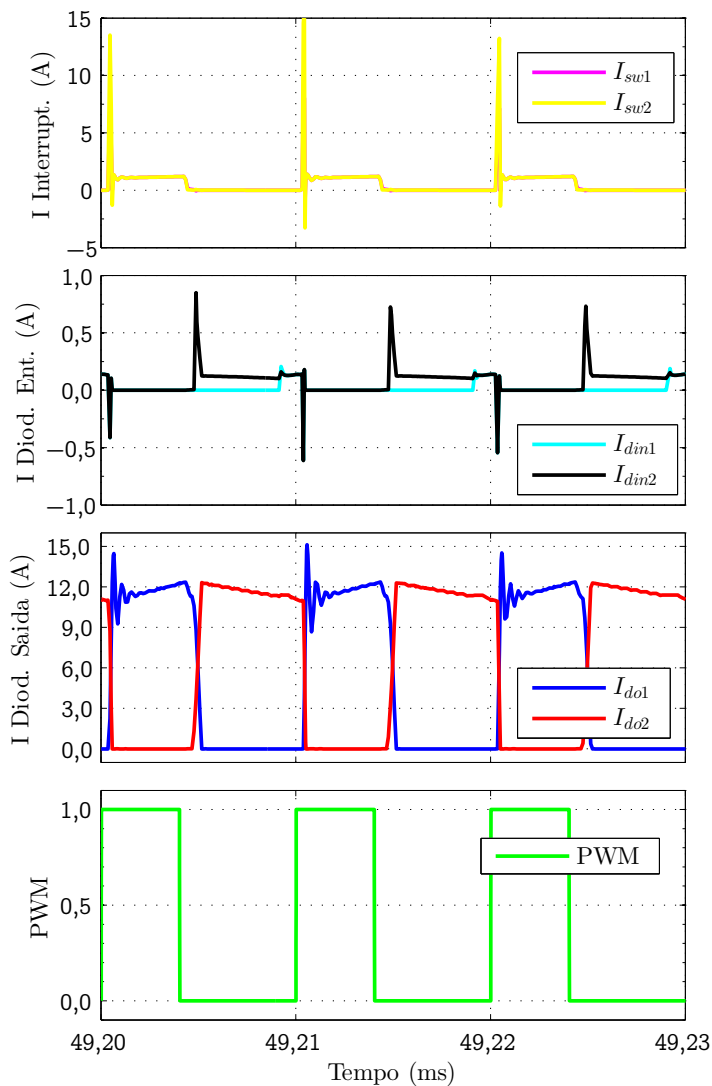


Figura 88 – Formas de onda simuladas das tensões nos semicondutores:
 V_{sw1} (rosa), V_{sw2} (amarelo), V_{din1} (azul claro), V_{din2} (preto), V_{do1} (azul) e V_{do2} (vermelho)

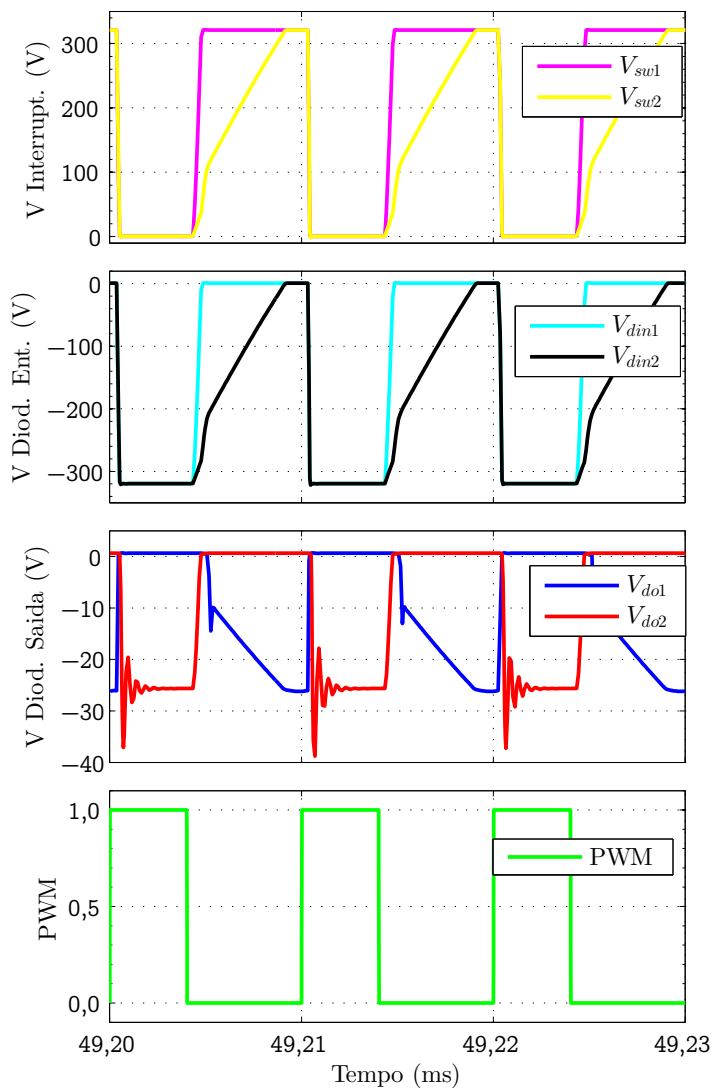
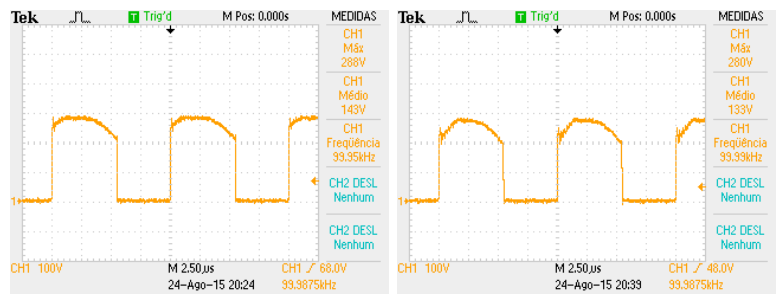
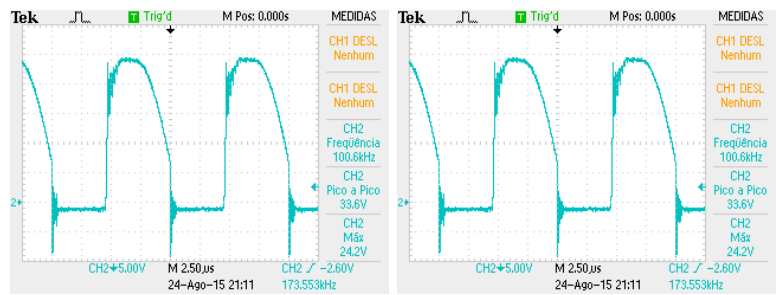


Figura 89 – Formas de onda experimentais da tensão sobre os interruptores S_{w1} (esquerda) e S_{w2} (direita)



Fonte: produção do próprio autor, 2016

Figura 90 – Formas de onda experimentais da tensão sobre os diodos D_{o1} (esquerda) e D_{o2} (direita)



Fonte: produção do próprio autor, 2016

Figura 91 – Formas de onda simuladas das tensões e correntes do transformador: V_p (rosa), V_s (amarelo), I_p (azul), I_s (vermelho)

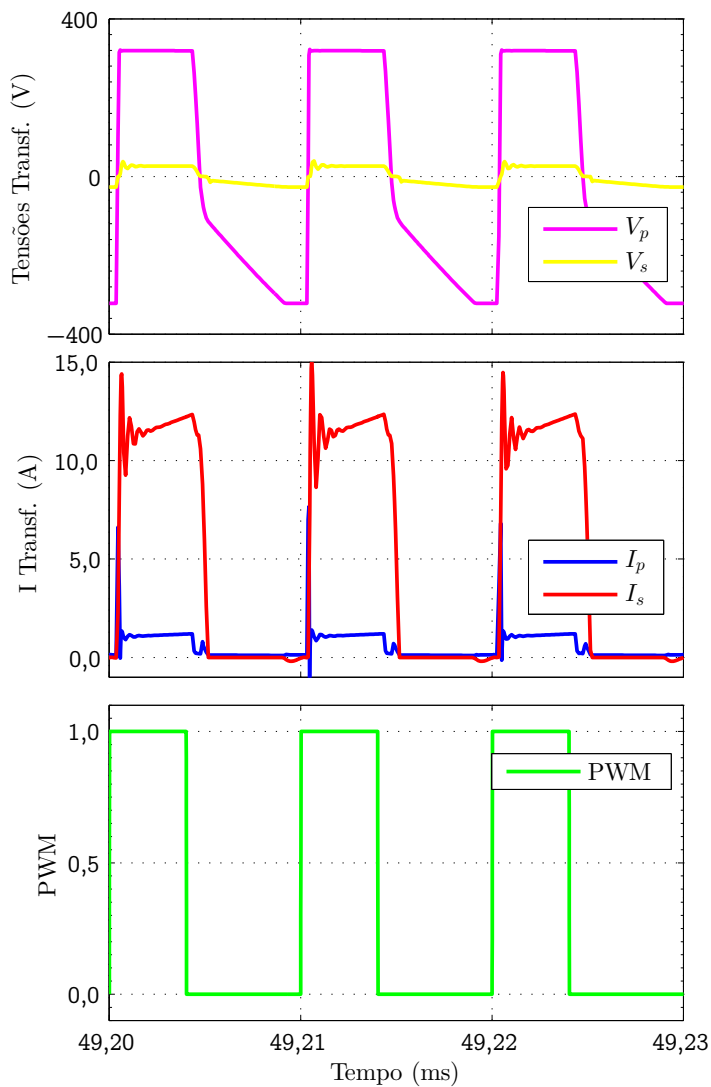
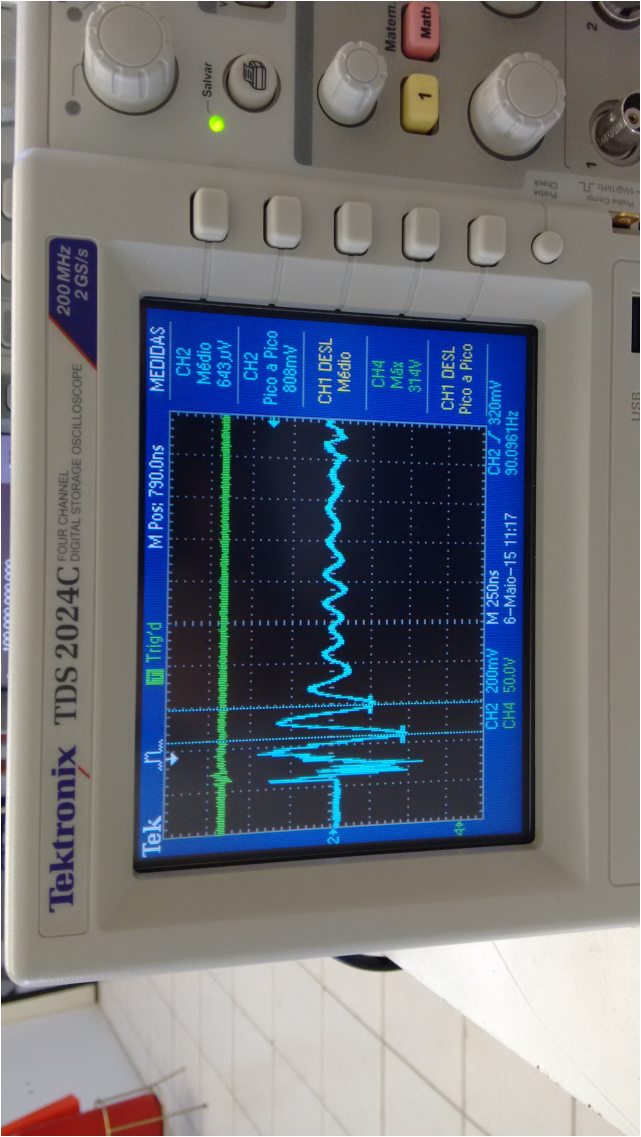
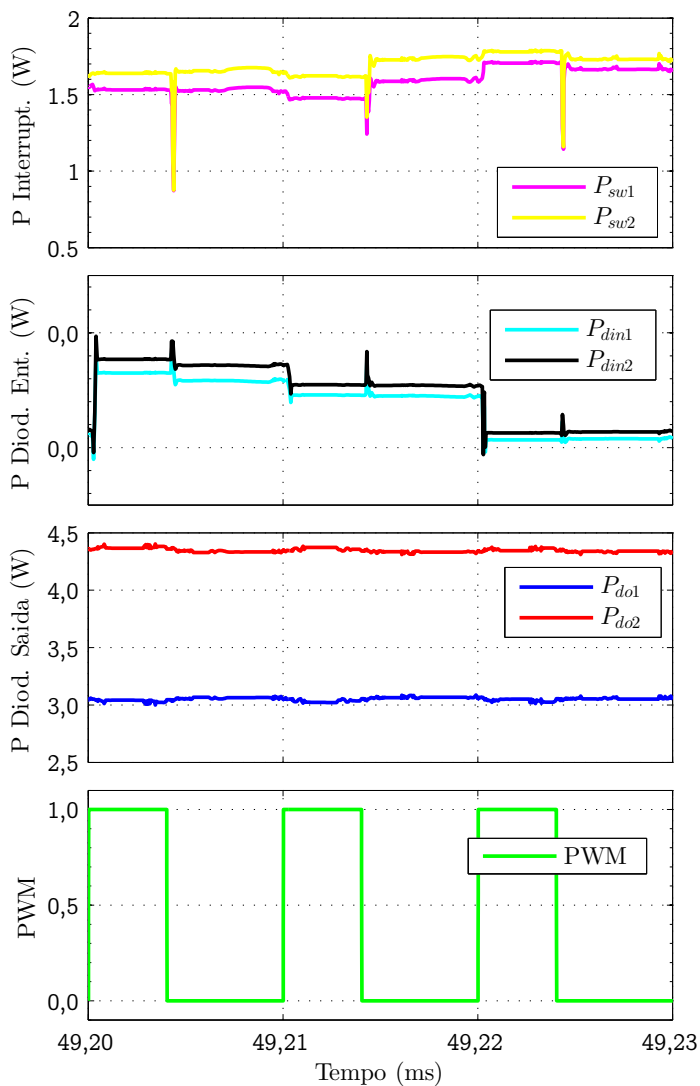


Figura 92 – Forma de onda experimental demonstrando o efeito do *Ground Bounce* (azul claro - CH2)



Fonte: produção do próprio autor, 2016

Figura 93 – Potências médias instantâneas simuladas nos semicondutores: P_{sw1} (rosa), P_{sw2} (amarelo), P_{din1} (azul claro), P_{din2} (preto), P_{do1} (azul) e P_{do2} (vermelho)



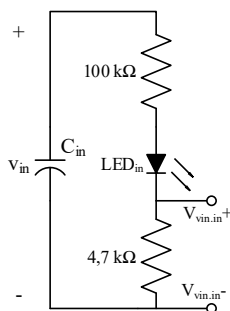
Apêndice – C INFORMAÇÕES COMPLEMENTARES SOBRE OS CIRCUITOS DE CONDICINAMENTO

C.1 CIRCUITOS DE CONDICIONAMENTO UTILIZADOS

C.1.1 Condicionamento do sinal da tensão de entrada do conversor

A tensão no barramento CC de entrada do conversor Duplo-Forward é medida para que possa ser feito um monitoramento contínuo da mesma através do DSC. Além disso, tal medida pode ser útil em projetos futuros que se utilizem de estratégias de controle do tipo *Feed-Forward* para compensação da ondulação da tensão v_{in} . A tensão de entrada é obtida pela medição diferencial em um divisor resistivo, no mesmo ramo em que se encontra o LED indicativo de tensão de entrada, conforme a Figura 94.

Figura 94 – Esquema de medição do sinal v_{in} para condicionamento



Fonte: produção do próprio autor, 2016

Com base na Figura 94 e sabendo que o valor máximo da tensão de entrada $V_{in_{max}}$, tem-se então o valor máximo de tensão diferencial medida em 14,63 V. Com o intuito de que o sinal de saída do circuito de condicionamento possa excursionar entre 0 e 3 V, foram escolhidos os resistores comerciais que melhor se encaixam nesta condição. Sendo o valor dos resistores conhecido, define-se na sequência

o capacitor que atende aos requisitos do LPF. A Tabela 21 mostra os resistores e capacitores escolhidos, além valor do ganho CC do circuito de condicionamento.

Tabela 21 – Dados do circuito de condicionamento de v_{in}

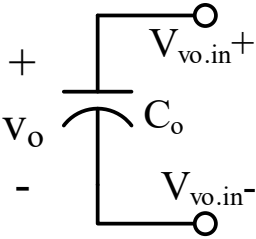
Descrição	Variável	Valor
Resistores de Ganho	$R_1 \ R_2$	$12 \ k\Omega$
Resistores de Ganho	$R_3 \ R_4$	$2,2 \ k\Omega$
Ganho do condicionamento de v_{in}	$G_{op.vin}$	$0,183 \ V/V$
Capacitor de filtro do cond.	C_3	$6,8 \ nF$

Fonte: produção do próprio autor, 2016

C.1.2 Condicionamento do sinal da tensão de saída do conversor

A tensão de saída do conversor deve ser medida caso seja necessário o seu controle e também para o seu monitoramento. Como a tensão de saída do conversor é baixa, sua medição diferencial é realizada diretamente através dos seus terminais, conforme Figura 95.

Figura 95 – Esquema de medição do sinal v_o para condicionamento.



Fonte: produção do próprio autor, 2016

A tensão máxima de saída $V_{o_{max}}$ de 10 V é adaptada para excursionar até 3 V utilizando-se os resistores do circuito de condicionamento descritos na Tabela 22.

Tabela 22 – Dados do circuito de condicionamento de v_o

Descrição	Variável	Valor
Resistores de Ganho	$R_1 R_2$	10 $k\Omega$
Resistores de Ganho	$R_3 R_4$	2,7 $k\Omega$
Ganho do condicionamento de v_{in}	$G_{op.vo}$	0,27 V/V
Capacitor de filtro do cond.	C_3	5,6 nF

Fonte: produção do próprio autor, 2016

C.1.3 Condicionamento do sinal da corrente no indutor

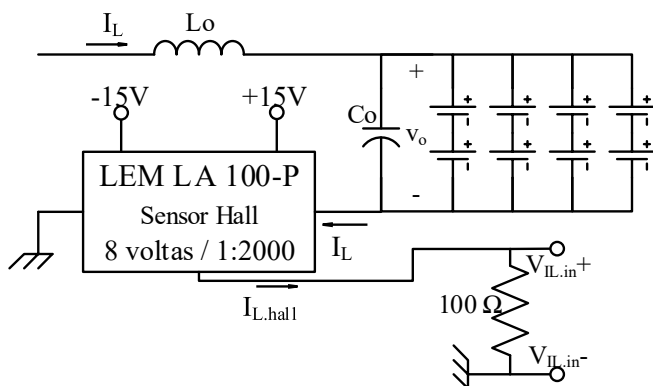
Uma vez que a corrente no indutor e consequentemente a corrente de saída do conversor deverá ser controlada, esta também deve ser condicionada. Como a corrente do conversor é a própria corrente de carga das baterias, que pode variar de 50 mA a 12 A, há a necessidade de que esta medida seja precisa em toda a excursão do sinal, para que o carregador seja capaz de controlar/impor o valor correto de corrente de carga.

Inicialmente este projeto utilizou um resistor *shunt* para a medição da corrente do indutor. Porém, uma vez que o carregador de baterias trabalha com uma impedância de saída muito baixa (baixa tensão e elevada corrente), a resistência do *shunt* deveria ser ainda menor (na ordem de 10 mΩ) para que não houvessem perdas significativas no conversor. Ao trabalhar com um valor de resistência muito baixo, tinha-se um valor de tensão no resistor *shunt* muito pequeno (em torno de 120 mV sob uma corrente de 12 A), resultando em uma relação sinal-ruído pobre, o que afetava as medidas de corrente principalmente sob operação com valores abaixo de 2 A.

Para solucionar o problema da medição do sinal de corrente foi necessária a substituição do resistor *shunt* por um sensor de efeito Hall

(LEM LA 100-P). A Figura 96 mostra o esquema de medição utilizado. O sensor foi alimentado com uma fonte externa Supplier isolada de $\pm 15\text{ V}$. Foram utilizadas ainda 8 voltas do condutor da corrente i_L de forma a aumentar a magnitude da corrente medida em 8 vezes. Como o sensor Hall possui um ganho de $1/2000$, tem-se então que a corrente produzida pelo sensor excursiona de 0 a 48 mA com uma corrente i_L variando de 0 a 12 A . Por fim, passando a corrente do sensor por um resistor de $100\ \Omega$, tem-se então uma medição de tensão diferencial entre 0 e $4,8\text{ V}$ para a corrente do indutor, a qual será então condicionada.

Figura 96 – Esquema de medição do sinal i_L para condicionamento.



Fonte: produção do próprio autor, 2016

A Tabela 23 mostra os resistores e capacitores utilizados no circuito de condicionamento da corrente no indutor.

Tabela 23 – Dados do circuito de condicionamento de i_L

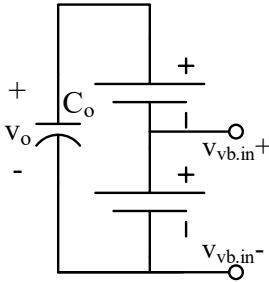
Descrição	Variável	Valor
Resistores de Ganho	$R_1 \ R_2$	$36 \ k\Omega$
Resistores de Ganho	$R_3 \ R_4$	$22 \ k\Omega$
Ganho do condicionamento de v_{in}	$G_{op.iL}$	$0,611 \ V/V$
Capacitor de filtro do cond.	C_3	$0,68 \ nF$

Fonte: produção do próprio autor, 2016

C.1.4 Condicionamento dos sinais de tensão das baterias

A tensão individual de cada célula do arranjo de baterias também é medida de forma que esta possa ser monitorada e controlada. Como cada bateria pode atingir uma tensão de até $4,2 \ V$, foi realizada a medição diferencial diretamente dos seus terminais, conforme Figura 97.

Figura 97 – Esquema de medição do sinal v_b para condicionamento.



Fonte: produção do próprio autor, 2016

Para o circuito de condicionamento, tem-se os valores dos componentes selecionados dispostos na Tabela 24.

Tabela 24 – Dados do circuito de condicionamento de v_b

Descrição	Variável	Valor
Resistores de Ganho	$R_1 R_2$	$10\text{ k}\Omega$
Resistores de Ganho	$R_3 R_4$	$4,7\text{ k}\Omega$
Ganho do condicionamento de v_{in}	$G_{op.vb}$	$0,47\text{ V/V}$
Capacitor de filtro do cond.	C_3	$3,3\text{ nF}$

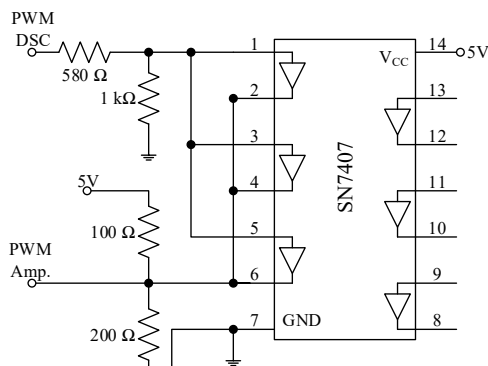
Fonte: produção do próprio autor, 2016

C.1.5 Condicionamento do sinal de PWM (saída)

Se por um lado há a necessidade de condicionamento dos sinais analógicos que serão enviados ao DSC, também há a necessidade de condicionamento do sinal de *Pulse-Width Modulation* (PWM - Modulação por Largura de Pulso) gerado pelo mesmo, uma vez que as portas lógicas do DSC TMS320F28335 possuem apenas 4 mA de capacidade máxima de corrente, enquanto a corrente requerida pelos circuitos dos fotodiodos é de $16,5\text{ mA}$, conforme dito na seção A.2.6.2. De forma a amplificar o sinal de corrente do DSC, fora utilizado o buffer SN7407. A Figura 98 mostra o esquema completo utilizado, o qual foi verificado por simulação e implementado na sequência.

Estando todos os sinais necessários já condicionados, tem-se então toda a estrutura necessária ao DSC para que este possa fazer todas as leituras dos sinais e enviar os sinais de comando para o conversor. Outros circuitos de condicionamento que foram projetados e implementados, porém não utilizados, estão descritos na seção C.2 deste Apêndice.

Figura 98 – Esquema de condicionamento do sinal de PWM.



Fonte: produção do próprio autor, 2016

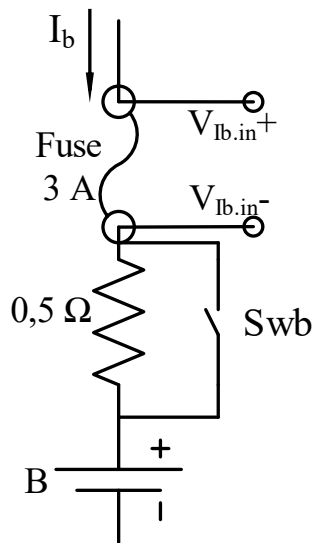
C.2 CIRCUITOS DE CONDICIONAMENTO NÃO UTILIZADOS

C.2.1 Condicionamento dos sinais de corrente das baterias

As correntes individuais de cada uma das baterias, apesar de não terem sido utilizadas neste projeto, também receberam um circuito dedicado de condicionamento pois estavam previstas para serem utilizadas inicialmente. Desta forma, uma vez que o circuito de condicionamento da corrente das baterias foi executado, este será documentado neste trabalho apenas para fins informativos.

Como a utilização de um sensor de efeito Hall para cada bateria não é uma opção atrativa do ponto de vista econômico e a utilização de resistores *shunt*, sempre conectados em série com as baterias, também não se mostrava a melhor das alternativas por agregar perdas elevadas ou não oferecer uma boa relação sinal-ruído, fora pensado na utilização do próprio fusível do ramo de cada bateria como transdutor, uma vez que este componente sempre estaria em série com sua bateria, ao contrário do resistor de balanceamento de $0,5\Omega$, que durante a operação de carga das baterias é contornado pelo interruptor de *by-pass*. A Figura 99 mostra o esquema de medição utilizado no ramo de uma das baterias.

Figura 99 – Esquema de medição do sinal i_b para condicionamento.



Fonte: produção do próprio autor, 2016

Os valores de resistores e capacitores projetados para o condicionamento encontram-se na Tabela 25 e foram obtidos considerando-se uma resistência de $50 \text{ m}\Omega$ para os fusíveis.

Tabela 25 – Dados do circuito de condicionamento de i_b

Descrição	Variável	Valor
Resistores de Ganho	$R_1 \ R_2$	$2,7 \text{ k}\Omega$
Resistores de Ganho	$R_3 \ R_4$	$47 \text{ k}\Omega$
Ganho do condicionamento de I_b	$G_{op.Ib}$	$17,41 \text{ V/V}$
Capacitor de filtro do cond.	C_3	$0,33 \text{ nF}$

Fonte: produção do próprio autor, 2016

C.2.2 Condicionamento dos sinais de temperatura das baterias

Nesta pesquisa, foi feito também o circuito de condicionamento da medição de temperatura. Este circuito também acabou não sendo utilizado na prática devido à não conclusão da câmara térmica, que iria se utilizar da medição de temperatura ambiente para a realização do controle da mesma. Com a falta da câmara térmica, o controle de temperatura foi feito manualmente através de condicionadores de ar instalados na sala, juntamente com uma caixa térmica de isopor, com o objetivo de aumentar a inércia térmica de sistema e manter a temperatura ambiente o mais constante possível.

A medição de temperatura foi idealizada contendo um transdutor LM35, o qual possui um coeficiente de $10\text{ mV}/^\circ\text{C}$, resultando em uma tensão de saída que varia linearmente com a temperatura medida. Esta tensão de saída é entregue através de um dos pinos do LM35, desta forma optou-se por não realizar a medição diferencial neste caso. O circuito de condicionamento é mostrado na Figura 100. Fazendo $R_1 = R_2$, o ganho CC desta estrutura é $G_{op}(0) = R_3/R_1 + 1$, enquanto a frequência de corte do LPF é dada por $1/2\pi R_1.C_T$.

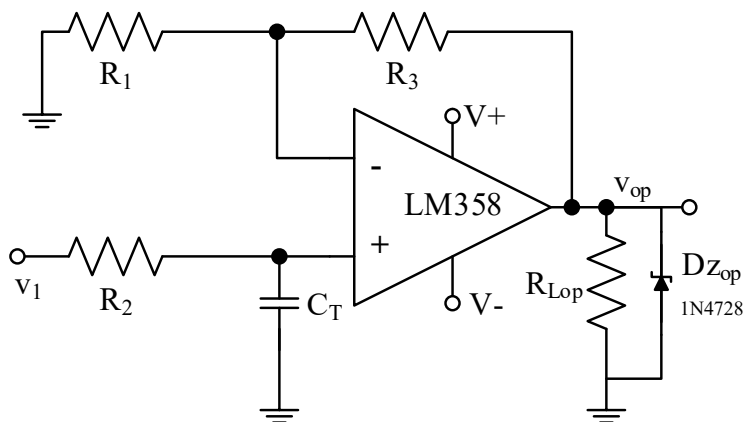
A frequência de corte neste caso foi selecionada em 1 kHz . Foram utilizados os seguintes resistores e capacitores para no circuito de condicionamento, dispostos na Tabela 26.

Tabela 26 – Dados do circuito de condicionamento de v_b

Descrição	Variável	Valor
Resistores de Ganho	$R_1\ R_2$	$4,7\text{ k}\Omega$
Resistores de Ganho	$R_3\ R_4$	$27\text{ k}\Omega$
Ganho do condicionamento de Temperatura	$G_{op.T}$	$6,75\text{ V/V}$
Capacitor de filtro do cond.	C_3	33 nF

Fonte: produção do próprio autor, 2016

Figura 100 – Esquema de condicionamento do sinal de temperatura do LM35.



Fonte: produção do próprio autor, 2016

ANEXOS

Anexo – A ANEXOS DO DESENVOLVIMENTO DO CON- VERSOR DUPLO-FORWARD

A.1 CÓDIGOS PARA OBTENÇÃO DAS INDUTÂNCIA DO TRANSFORMADOR

```

1 %main code for Inductances
2 clear all, clc
3
4 Lxo = [28.34e-3 0.0283e-3];%0.04e-6; %initial guess
5 %for [Lm Lp1 Lp2]
6 [Lx, fval] = fsolve(@myfun2,Lxo)%options) % Call solver
7 norm(fval)
8
9 Lm = Lx(1)
10 Lp1 = Lx(2)
11 %Lp2 = Lx(3) %fixed due convergence problem.
12 Lp2 = 0.04e-6 %fixed due convergence problem.
13
14 Lm/Lp1

```

```

1 %function used on Inductance calculation
2 function F = myfun2(x,y,N)
3
4 N = 12;
5 %Values from OC and SC tests
6 Lop = 28.4e-3;
7 Lsc1 = 0.0282e-3;
8 Lsc2 = 0.04e-6;
9
10 %α(1) = Lm
11 %α(2) = Lp1
12 %α(3) = Lp2
13 Lp2 = 0.01e-6; %fixed due convergence problem.
14 y = [Lop Lsc1];
15
16 %Non-linear functinos
17 F = [x(1) + x(2) - y(1);
18      x(2) + (x(1)*Lp2*N^2)/(x(1) + Lp2*N^2) - y(2)];
19 %α(2) + (x(1)*x(3)*N^2)/(x(1) + x(3)*N^2) - y(2);
20 %α(3) + (1/N^2)*(x(1)*x(2))/(x(1) + x(2));

```

Anexo – B ANEXOS DO DESENVOLVIMENTO DO CONTROLADOR

B.1 CÓDIGOS PARA CONFIGURAÇÃO DO DSC

B.1.1 Configuração do ADC

```

1  /*
2
3  INEP – UFSC
4
5
6  TÍTULO :Projeto do Conversor Back to Back
7  DATA: 05/07/2009
8  ATUALIZAÇÃO: 08/08/2015
9  VERSÃO:1.1
10
11  DESENVOLVIMENTO: MÁRCIO S. ORIMANN SAMIR AHMAD MUSSA
12  ATUALIZAÇÃO: CHRYSTIAN LENON REMES
13
14  OBSERVAÇÕES:
15  */
16
17 #include <adc.h>
18 #include "DSP2833x_Device.h"
19 #include "Define.h"
20
21 #define ADC_Cal (void) (*) (void) 0x380080 //Pointer to ADC_Cal() Routine
22     ↳ programmed by TI
23
24 void adc_config(void){
25
26     extern void DSP28x_usDelay(Uint32 Count);
27
28     // Enables ADC Clock and Calibrates ADC;
29     EALLOW;
30     SysCtrlRegs.PCLKCR0.bit.ADCENCLK = 1; //Enables ADC Clock
31     (*ADC_Cal)(); //ADC
32     ↳ Calibration: Calls ADC_Cal() Routine
33     SysCtrlRegs.PCLKCR0.bit.ADCENCLK = 0; //Disables ADC Clock
34     EDIS;
35
36     EALLOW;
37
38     // Determining the sequences for buffer of conversions.
39     AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0; // ADG-A0
40     AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 1; // ADG-A1
41     AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 2; // ADG-A2
42     AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 3; // ADG-A3
43
44     AdcRegs.ADCCHSELSEQ2.bit.CONV04 = 4; // ADG-A4
45     AdcRegs.ADCCHSELSEQ2.bit.CONV05 = 5; // ADG-A5
46     AdcRegs.ADCCHSELSEQ2.bit.CONV06 = 6; // ADG-A6
47     AdcRegs.ADCCHSELSEQ2.bit.CONV07 = 7; // ADG-A7
48
49     AdcRegs.ADCCHSELSEQ3.bit.CONV08 = 8; // ADG-B0
50     AdcRegs.ADCCHSELSEQ3.bit.CONV09 = 9; // ADG-B1
51     AdcRegs.ADCCHSELSEQ3.bit.CONV10 = 10; // ADG-B2
52     AdcRegs.ADCCHSELSEQ3.bit.CONV11 = 11; // ADG-B3
53
54     AdcRegs.ADCCHSELSEQ4.bit.CONV12 = 12; // ADG-B4
55     AdcRegs.ADCCHSELSEQ4.bit.CONV13 = 13; // ADG-B5
56     AdcRegs.ADCCHSELSEQ4.bit.CONV14 = 14; // ADG-B6
57     AdcRegs.ADCCHSELSEQ4.bit.CONV15 = 15; // ADG-B7
58
59     // ADC Control Register 1

```

```

58     AdcRegs.ADCCTRL1.bit.RESET = 0;           //No Action
59     AdcRegs.ADCCTRL1.bit.SUSMOD = 0;          //Suspension mode 2 (by
        ↳ debugger): complete conversion before stop
60     AdcRegs.ADCCTRL1.bit.ACQ_PS = 2;          //Aquisition window size (
        ↳ ACQ_PS + 1) ADCCLK periods
61     AdcRegs.ADCCTRL1.bit.CPS = 1;             //Divide HSPCLK frequency (0
        ↳ is /1 and 1 is /2) (prescaler)
62     AdcRegs.ADCCTRL1.bit.CONT_RUN = 0;        //0 is Start-Stop mode and 1
        ↳ is continuous mode
63     AdcRegs.ADCCTRL1.bit.SEQ_OVRD = 0;        //Sequence override (0 dis, 1
        ↳ en)
64     AdcRegs.ADCCTRL1.bit.SEQ_CASC = 1;        //0 is Dual-sequencer, 1 is
        ↳ Cascade mode

65 //     ADC Control Register 2
66     AdcRegs.ADCCTRL2.bit.EPWM_SOCA_SEQ = 0;   //0 No Action, 1 allows the cascade
        ↳ SEQ to be started by dPWM SOCB
67     AdcRegs.ADCCTRL2.bit.RST_SEQ1 = 0;        //0 No Action, 1 resets the SEQ1 or
        ↳ cascade SEQ
68     AdcRegs.ADCCTRL2.bit.SOC_SEQ1 = 0;        //0 Clear pending "start of
        ↳ conversion", 1 prepare for start conversion
69     AdcRegs.ADCCTRL2.bit.INT_ENA_SEQ1 = 1;     //Enable interrupts by SEQ1
70     AdcRegs.ADCCTRL2.bit.INT_MOD_SEQ1 = 0;    //INT_SEQ1 is setted in all sequences
71     AdcRegs.ADCCTRL2.bit.EPWM_SOCA_SEQ1 = 1;  //0 No Action, 1 allows the cascade
        ↳ SEQ to be started by dPWM SOCA
72     AdcRegs.ADCCTRL2.bit.EXT_SOC_SEQ1 = 0;    //1 enables ADC autoconversion to be
        ↳ started by GPIOA (need config as XINT2)
73     AdcRegs.ADCCTRL2.bit.RST_SEQ2 = 0;        //0 No Action, 1 resets the
        ↳ SEQ2
74     AdcRegs.ADCCTRL2.bit.SOC_SEQ2 = 0;        //0 Clear pending "start of
        ↳ conversion", 1 prepare for start conversion
75     AdcRegs.ADCCTRL2.bit.INT_ENA_SEQ2 = 0;    //Desabilita o pedido de INT pelo
        ↳ SEQ2
76     AdcRegs.ADCCTRL2.bit.INT_MOD_SEQ2 = 0;    //INT_SEQ1 is setted in all sequences
77     AdcRegs.ADCCTRL2.bit.EPWM_SOCA_SEQ2 = 0; //0 No Action, 1 allows the SEQ2 to
        ↳ be started by dPWM SOCB
78
79 //     ADC Control Register 3
80     AdcRegs.ADCCTRL3.bit.ADCBGRFDN = 3;       //Power up bandgap/reference
81     AdcRegs.ADCCTRL3.bit.ADCPWIN = 1;         //Power up ADC core circuitry
82     AdcRegs.ADCCTRL3.bit.ADCCLKPS = 2;        //Clock Divider (HSPCLK/
        ↳ ADCCLKPS), HSPCLK = 150MHz, ADCCLK is ADCCTRL1[7]
83     AdcRegs.ADCCTRL3.bit.SMODE_SEL = 0;       //Sequential sampling mode
84
85 //     ADC Maximum Conversions
86     AdcRegs.ADCMAXCONV.all = 7;                //2 (3 conversions)
87     ↳ 11 conversões
88
89 //     ADC Offset
90     AdcRegs.ADCOFFTRIM.all = 0x00;            //0 offset; -256 ~
        ↳ 255 in 2-complement format
91
92     EDIS;
93 }
94
95 void analog_trigger(void){
96
97     EALLOW;
98     AdcRegs.ADCCTRL2.bit.RST_SEQ1 = 1;        // Reseta Seq1
99     AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;       // limpa flag de int
100     AdcRegs.ADCCTRL2.bit.SOC_SEQ1 = 1;        // Disparo para início de
        ↳ conversão
101
102     EDIS;
103
104     while(AdcRegs.ADCST.bit.INT_SEQ1==0){    // loop a espera do fim da conversão
105         ;
106     }
107 }

```

B.1.2 Configuração do PWM

```

1  #include "DSP2833x_Device.h"
2  #include "Define.h"
3  #include <epwm_config.h>
4
5  void epwm_config(void)
6  {
7
8      EALLOW;
9      SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC=0; // for carriers sinc
10     EDIS;
11
12     // Time-Base (TB) Configuration
13     EALLOW;
14     EPwm1Regs.TBPRD = TBPRD_EPWM1; //Time-base period
15     // TBPRD ... Tpwm = (TBPRD+1)*TBCLK; TBCLK = 1/SYSCLKOUT = 1/150MHz;
16     // Tpwm = 10us Tpwm = 2*TBPRD*TBCLK (triangular)
17
18     EPwm1Regs.TBPHS.half.TBPHS = 0; //Value, where the counter TBCTR will
19     // continue
20
21     //EPwm1Regs.TBCTR = 0x00; //Initializes counter
22     // in 0
23
24     EPwm1Regs.TBCIL.bit.FREE_SOFT = 2; //Free run during debugging
25     EPwm1Regs.TBCIL.bit.PHSDIR = 1; //Count up after sync event
26     EPwm1Regs.TBCIL.bit.CLKDIV = 0; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV)
27     // ... 0 is /1 and 7 is /128...
28     EPwm1Regs.TBCIL.bit.HSPCLKDIV = 0; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV)
29     // ... 0 is /1 and 7 is /14...
30     EPwm1Regs.TBCIL.bit.SWFSYNC = 1; //Sync by SW... 0 has no effect. 1
31     // forces a one-time sync sw pulse
32     EPwm1Regs.TBCIL.bit.SYNCOSEL = 1; //Select the output Sync.. 0 is
33     // epwmSYNCO = epwmSYNC; 1 is SYNCO when CTR = 0;
34     EPwm1Regs.TBCIL.bit.PRDL = 0; //Enables the TBPRD shadow register
35     EPwm1Regs.TBCIL.bit.PHSEN = 0; //Don't load TBCTR from TBPHS
36     EPwm1Regs.TBCIL.bit.CTRMODE = 2; //Counter mode -> Up-down-count mode
37     // (triangular)
38     EDIS;
39
40     // Counter-Compare (CC) Configuration
41     EALLOW;
42     EPwm1Regs.CMPCTL.bit.SHOWMODE = 0; //CPMB Shadow mode enable
43     EPwm1Regs.CMPCTL.bit.SHOWMODE = 0; //CPMA Shadow mode enable
44     EPwm1Regs.CMPCTL.bit.LOADMODE = 2; //CPMB is loaded from shadow at CTR =
45     // 0 and CTR = PRD;
46     EPwm1Regs.CMPCTL.bit.LOADMODE = 2; //CPMA is loaded from shadow at CTR =
47     // 0 and CTR = PRD;
48     EDIS;
49
50     // Action-Qualifier (AQ) Configuration
51     EALLOW;
52     EPwm1Regs.AQCTLA.bit.CED = 0; //00 "Do nothing" Action in pwmA when
53     // CTR = CPMB and counting is down
54     EPwm1Regs.AQCTLA.bit.CBU = 0; //00 "Do nothing" Action in pwmA when
55     // CTR = CPMB and counting is up
56     EPwm1Regs.AQCTLA.bit.CAD = 2; //01 "Clear" Action in pwmA when CTR
57     // = CPMA and counting is down
58     EPwm1Regs.AQCTLA.bit.CAU = 1; //10 "Set" Action in pwmA when CTR =
59     // CPMA and counting is up
60     EPwm1Regs.AQCTLA.bit.PRD = 0; //00 "Do nothing" Action when CTR =
61     // PRD
62     EPwm1Regs.AQCTLA.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
63     // ZERO
64
65     EPwm1Regs.AQCTLB.bit.CED = 0; //00 "Do nothing" Action in pwmB when
66     // CTR = CPMB and counting is down
67     EPwm1Regs.AQCTLB.bit.CBU = 0; //00 "Do nothing" Action in pwmB when
68     // CTR = CPMB and counting is up

```

```

50     EPwm1Regs.AQCTLB.bit.CAD = 1;           //01 "Clear" Action in pwmA when CTR
        ↳ = CPMA and counting is down
51     EPwm1Regs.AQCTLB.bit.CAU = 2;           //10 "Set" Action in pwmA when CTR =
        ↳ CPMA and counting is up
52     EPwm1Regs.AQCTLB.bit.PR0 = 0;           //00 "Do nothing" Action when CTR =
        ↳ PR0
53     EPwm1Regs.AQCTLB.bit.Z0 = 0;           //00 "Do nothing" Action when CTR =
        ↳ ZERO
54
55     EPwm1Regs.AQSFRC.bit.RLDCSF = 0;         //AQ load from shadow registers. 00
        ↳ is in CTR = 0
56     EPwm1Regs.AQSFRC.bit.OTSFB = 0;         //OTSFB = 1 trigger a software event
        ↳ out B... 0 no action
57     EPwm1Regs.AQSFRC.bit.OTSFA = 0;         //Action when OTSFB = 1... 00 do
        ↳ nothing
58     EPwm1Regs.AQSFRC.bit.OTSFA = 0;         //OTSFA = 1 trigger a software event
        ↳ out A... 0 no action
59     EPwm1Regs.AQSFRC.bit.OTSFA = 0;         //Action when OTSFA = 1... 00 do
        ↳ nothing
60
61     EPwm1Regs.AQSFRC.bit.CSFB = 0;           //continuous force low/high on out B
        ↳ ... 00 forcing is disabled, 01 forces continuous low
62     EPwm1Regs.AQSFRC.bit.CSFA = 0;           //continuous force low/high on out A
        ↳ ... 00 forcing is disabled, 01 forces continuous low
63     EDIS;
64
65 //     Dead-band (DB) Configuration
66 EALLOW;
67     EPwm1Regs.DBCTL.bit.IN_MODE = 0;         //dPwMA is the source for rise-edge
        ↳ and fall-edge for delay (dead-band)
68     EPwm1Regs.DBCTL.bit.P0ISEL = 2;         //Type of delay. 10 is Active High
        ↳ Complementary (PwMB is inverted) See report 2.5 fig 32
69     EPwm1Regs.DBCTL.bit.OUT_MODE = 3;       //dead-band is enabled for both rise
        ↳ and fall edges.
70
71     EPwm1Regs.DBERED = 1;                   //DBERED = 1, dead-band = 0.01
        ↳ us (see report 2.5 tab 15)
72     EPwm1Regs.DBFED = 1;                   //DBFED = 1, dead-band = 0.01
        ↳ us (see report 2.5 tab 15)
73     EDIS;
74
75 //     PWMChopper (PC) Configuration
76 EALLOW;
77     EPwm1Regs.PCCIL.bit.CHPDUTY = 0;         //Chopping Clock Duty Cycle
78     EPwm1Regs.PCCIL.bit.CHPFREQ = 0;         //Chopping Clock Frequency Divider
79     EPwm1Regs.PCCIL.bit.OSHIWIH = 0;         //One-shot Pulse Width (80ns) (see
        ↳ manual)
80     EPwm1Regs.PCCIL.bit.CHPEN = 0;           //Chopping disable. Bypass PWM
        ↳ chopping function
81     EDIS;
82
83 //     Trip-Zone (TZ) Configuration
84 EALLOW;
85     EPwm1Regs.TZSEL.all = 0x0100;           //Enable Trip OneShot to dPwM1
86     EPwm1Regs.TZCTL.bit.TZB = 2;           //Forces Low State on Trip Event to
        ↳ dPwMB
87     EPwm1Regs.TZCTL.bit.TZA = 2;           //Forces Low State on Trip Event to
        ↳ dPwMA
88     EPwm1Regs.TZEINT.bit.OST = 1;           //1 enable OneShot Trip Interrupt in
        ↳ PIE
89     EPwm1Regs.TZEINT.bit.CBC = 0;           //0 disable cycle-by-cycle interrupt
        ↳ (not used)
90     EPwm1Regs.TZCLR.all = 0x0000;           //Clear all Trips
91     EDIS;
92
93 //     Event-trigger (ET) Configuration
94 EALLOW;
95     EPwm1Regs.ETSEL.bit.SOCBEN = 1;         //1 enable ADC start of conversion
        ↳ after a dPwMB pulse
96     EPwm1Regs.ETSEL.bit.SOCSEL = 1;         //Generate a ADC pulse when CTR = 0

```

```

97     EPwm1Regs.ETSEL.bit.SOCAEN = 1;           //1 enable ADC start of conversion
98         ↳ after a dPWMB pulse
99     EPwm1Regs.ETSEL.bit.SOCASEL = 1;           //Generate a ADC pulse when CTR = 0
100    EPwm1Regs.ETSEL.bit.INTEN = 0;             //1 Enable PWM interrupt generation (
101         ↳ not used)
102    EPwm1Regs.ETSEL.bit.INTSEL = 0;             //(0 = Reserved) Select when
103         ↳ interrupt occurs
104
105    EPwm1Regs.ETPS.bit.SOCBPRD = 0;             //Disable SOCB event counter
106    EPwm1Regs.ETPS.bit.SOCAPRD = 0;             //Disable SOCA event counter
107    EPwm1Regs.ETPS.bit.INTPRD = 0;             //Disable interrupt event counter.
108    EDIS;
109
110    // Initialize CPMA
111    EPwm1Regs.CMPA.half.CMPA = 0x0001;         //CPMA = 1
112    EPwm1Regs.CMPB = 0x0001;                   //CPMB = 1
113
114    // dPWM2
115    EALLOW;
116    EPwm2Regs.TBPRD = TBPRD_EPWM2;             //Time-base period
117         ↳ TBPRD ... TBCLK = 1/SYSCLKOUT = 1/150MHz; TpwM = (TBPRD+1)*TBCLK (UP
118         ↳ or Down); TpwM = 2*TBPRD*TBCLK (UpDown); TpwM =
119    EPwm2Regs.TBPHS.half.TBPHS = 0;             //Value, where the counter
120         ↳ TBCIR will continue
121    //EPwm2Regs.TBCIR = 0x00;                   //Initializes counter
122         ↳ in 0
123    EPwm2Regs.TBCIL.bit.FREE_SOFT = 2;          //Free run during debugging
124    EPwm2Regs.TBCIL.bit.PHSDIR = 1;             //Count up after sync event
125    EPwm2Regs.TBCIL.bit.CLKDIV = 0;             //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV) ... 0
126         ↳ is /1 and 7 is /128...
127    EPwm2Regs.TBCIL.bit.HSPCLKDIV = 0;          //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV) ... 0
128         ↳ is /1 and 7 is /14...
129    EPwm2Regs.TBCIL.bit.SWFSYNC = 1;            //Sync by SW... 0 has no effect. 1
130         ↳ forces a one-time sync sw pulse
131    EPwm2Regs.TBCIL.bit.SYNCOSEL = 1;           //Select the output Sync.. 0 is
132         ↳ epwmSYNCO = epwmSYNC; 1 is SYNCO when CTR = 0;
133    EPwm2Regs.TBCIL.bit.PRDLID = 0;             //Enables the TBPRD shadow register
134    EPwm2Regs.TBCIL.bit.PHSEN = 0;             //Don't load TBCIR from TBPHS
135    EPwm2Regs.TBCIL.bit.CTRMODE = 2;           //Counter mode -> Up-down-count mode
136    EDIS;
137    EALLOW;
138    EPwm2Regs.CMPCTL.bit.SHOWMODE = 0;         //CPMB Shadow mode enable
139    EPwm2Regs.CMPCTL.bit.SHOWMODE = 0;         //CPMA Shadow mode enable
140    EPwm2Regs.CMPCTL.bit.LOADMODE = 2;         //CPMB is loaded from shadow at CTR =
141         ↳ 0 and CTR = PRD;
142    EPwm2Regs.CMPCTL.bit.LOADMODE = 2;         //CPMA is loaded from shadow at CTR =
143         ↳ 0 and CTR = PRD;
144    EDIS;
145    EALLOW;
146    EPwm2Regs.AQCILA.bit.CBD = 0;              //00 "Do nothing" Action in pwmA when
147         ↳ CTR = CPMB and counting is down
148    EPwm2Regs.AQCILA.bit.CBU = 0;              //00 "Do nothing" Action in pwmA when
149         ↳ CTR = CPMB and counting is up
150    EPwm2Regs.AQCILA.bit.CAD = 1;              //01 "Clear" Action in pwmA when CTR
151         ↳ = CPMA and counting is down
152    EPwm2Regs.AQCILA.bit.CAU = 2;              //10 "Set" Action in pwmA when CTR =
153         ↳ CPMA and counting is up
154    EPwm2Regs.AQCILA.bit.PRD = 0;              //00 "Do nothing" Action when CTR =
155         ↳ PRD
156    EPwm2Regs.AQCILA.bit.ZRO = 0;              //00 "Do nothing" Action when CTR =
157         ↳ ZERO
158    EPwm2Regs.AQCILB.bit.CBD = 0;              //00 "Do nothing" Action in pwmB when
159         ↳ CTR = CPMB and counting is down
160    EPwm2Regs.AQCILB.bit.CBU = 0;              //00 "Do nothing" Action in pwmB when
161         ↳ CTR = CPMB and counting is up
162    EPwm2Regs.AQCILB.bit.CAD = 2;              //01 "Clear" Action in pwmA when CTR
163         ↳ = CPMA and counting is down
164    EPwm2Regs.AQCILB.bit.CAU = 1;              //10 "Set" Action in pwmA when CTR =
165         ↳ CPMA and counting is up
166    EPwm2Regs.AQCILB.bit.PRD = 0;              //00 "Do nothing" Action when CTR =
167         ↳ PRD

```

```

144 EPwm2Regs.AQCTLB.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↳ ZERO
145 EPwm2Regs.AQSFRC.bit.RLDCSF = 0; //AQ load from shadow registers. 00
    ↳ is in CTR = 0
146 EPwm2Regs.AQSFRC.bit.OTSFB = 0; //OTSFB = 1 trigger a software event
    ↳ out B... 0 no action
147 EPwm2Regs.AQSFRC.bit.ACTSFB = 0; //Action when OTSFB = 1... 00 do
    ↳ nothing
148 EPwm2Regs.AQSFRC.bit.OTSFA = 0; //OTSFA = 1 trigger a software event
    ↳ out A... 0 no action
149 EPwm2Regs.AQSFRC.bit.ACTSFA = 0; //Action when OTSFA = 1... 00 do
    ↳ nothing
150 EPwm2Regs.AQSFRC.bit.CSFB = 0; //continuous force low/high on out B
    ↳ ... 00 forcing is disabled, 01 forces continuous low
151 EPwm2Regs.AQSFRC.bit.CSFA = 0; //continuous force low/high on out A
    ↳ ... 00 forcing is disabled, 01 forces continuous low

152 EDIS;
153 EALLOW;
154 EPwm2Regs.DBCCTL.bit.IN_MODE = 0; //dPWMxA is the source for rise-edge
    ↳ and fall-edge for delay (dead-band)
155 EPwm2Regs.DBCCTL.bit.POLSEL = 2; //Type of delay. 10 is Active High
    ↳ Complementary (PwMB is inverted) See report 2.5 fig 32
156 EPwm2Regs.DBCCTL.bit.OUT_MODE = 3; //dead-band is enabled for both rise
    ↳ and fall edges.
157 EPwm2Regs.DBFED = 1; //DBFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)
158 EPwm2Regs.DBFED = 1; //DBFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)

159 EDIS;
160 EALLOW;
161 EPwm2Regs.PCCCTL.bit.CHPDUTY = 0; //Chopping Clock Duty Cycle
162 EPwm2Regs.PCCCTL.bit.CHPFREQ = 0; //Chopping Clock Frequency Divider
163 EPwm2Regs.PCCCTL.bit.OH1WIH = 0; //One-shot Pulse Width (80ns) (see
    ↳ manual)
164 EPwm2Regs.PCCCTL.bit.CHPEN = 0; //Chopping disable. Bypass PWM
    ↳ chopping function

165 EDIS;
166 EALLOW;
167 EPwm2Regs.TZSEL.all = 0x0000; //Disable all Trip-Zone sources (not
    ↳ used)
168 EPwm2Regs.TZCTL.bit.TZB = 3; //11 Do nothing when a trip occurs (
    ↳ not used)
169 EPwm2Regs.TZCTL.bit.TZA = 3; //11 Do nothing when a trip occurs (
    ↳ not used)
170 EPwm2Regs.TZEINT.bit.OST = 0; //0 disable One shot interrupt (not
    ↳ used)
171 EPwm2Regs.TZEINT.bit.CBC = 0; //0 disable cycle-by-cycle interrupt
    ↳ (not used)

172 EDIS;
173 EALLOW;
174 EPwm2Regs.ETSEL.bit.SOCBEN = 0; //1 enable ADC start of conversion
    ↳ after a dPWMB pulse
175 EPwm2Regs.ETSEL.bit.SOCBSEL = 0; //Generate a ADC pulse when CTR = 0
176 EPwm2Regs.ETSEL.bit.SOCAEN = 1; //1 enable ADC start of conversion
    ↳ after a dPWMA pulse
177 EPwm2Regs.ETSEL.bit.SOCASEL = 1; //Generate a ADC pulse when CTR = 0
178 EPwm2Regs.ETSEL.bit.INTEN = 0; //1 Enable PWM interrupt generation (
    ↳ not used)
179 EPwm2Regs.ETSEL.bit.INTSEL = 0; //Generate interruption when CTR = 0
180 EPwm2Regs.EITPS.bit.SOCBPRD = 0; //Disable SOCB event counter
181 EPwm2Regs.EITPS.bit.SOCAPRD = 0; //Disable SOCA event counter
182 EPwm2Regs.EITPS.bit.INTPRD = 0; //Disable interrupt event counter.
183 EDIS;
184 EPwm2Regs.CMPA.half.CMPA = 300; //CPMA = 300 (aprox. 10% of Tprwm)
185 EPwm2Regs.CMPB = 300; //CPMB = 300 (aprox. 10% of
    ↳ Tprwm)

186 //
187 dPWM-3
188 EALLOW;
189 EPwm3Regs.TBPRD = 1499; //Time-base period TBPRD ...
    ↳ Tprwm = (TBPRD+1)*TBCLK; TBCLK = 1/SYSCLKOUT = 1/150MHz; Tprwm = 10us

```

```

190 EPwm3Regs.TBPHS.half.TBPHS = 0x00; //Value, where the
    ↪ counter TBCTR will continue
191 EPwm3Regs.TBCTR = 0x00; //Initializes counter in 0
192 EPwm3Regs.TBCIL.bit.FREE_SOFT = 2; //Free run during debugging
193 EPwm3Regs.TBCIL.bit.PHSDIR = 1; //Count up after sync event
194 EPwm3Regs.TBCIL.bit.CLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV
    ↪ )... 0 is /1 and 7 is /128...
195 EPwm3Regs.TBCIL.bit.HSPCLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV) ...
    ↪ 0 is /1 and 7 is /14...
196 EPwm3Regs.TBCIL.bit.SWFSYNC = 1; //Sync by SW... 0 has no effect. 1
    ↪ forces a one-time sync sw pulse
197 EPwm3Regs.TBCIL.bit.SYNCOSEL = 1; //Select the output Sync.. 0 is
    ↪ epwmSYNCO = epwmSYNC; 1 is SYNCO when CTR = 0;
198 EPwm3Regs.TBCIL.bit.PRDL = 0; //Enables the TBPRD shadow register
199 EPwm3Regs.TBCIL.bit.PHSEN = 0; //Don't load TBCTR from TBPHS
200 EPwm3Regs.TBCIL.bit.CTRMODE = 2; //Counter mode -> Up-down-count mode
201 EDIS;
202 FALLOW;
203 EPwm3Regs.CMPCIL.bit.SHOWMODE = 0; //CPMB Shadow mode enable
204 EPwm3Regs.CMPCIL.bit.SHOWMODE = 0; //CPMA Shadow mode enable
205 EPwm3Regs.CMPCIL.bit.LOADMODE = 2; //CPMB is loaded from shadow at CTR =
    ↪ 0 and CTR = PRD;
206 EPwm3Regs.CMPCIL.bit.LOADMODE = 2; //CPMA is loaded from shadow at CTR =
    ↪ 0 and CTR = PRD;
207 EDIS;
208 FALLOW;
209 EPwm3Regs.AQCILA.bit.CBD = 0; //00 "Do nothing" Action in pwmA when
    ↪ CTR = CPMB and counting is down
210 EPwm3Regs.AQCILA.bit.CBU = 0; //00 "Do nothing" Action in pwmA when
    ↪ CTR = CPMB and counting is up
211 EPwm3Regs.AQCILA.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↪ = CPMA and counting is down
212 EPwm3Regs.AQCILA.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↪ CPMA and counting is up
213 EPwm3Regs.AQCILA.bit.PRD = 0; //00 "Do nothing" Action when CTR =
    ↪ PRD
214 EPwm3Regs.AQCILA.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↪ ZERO
215 EPwm3Regs.AQCILB.bit.CBD = 0; //00 "Do nothing" Action in pwmB when
    ↪ CTR = CPMB and counting is down
216 EPwm3Regs.AQCILB.bit.CBU = 0; //00 "Do nothing" Action in pwmB when
    ↪ CTR = CPMB and counting is up
217 EPwm3Regs.AQCILB.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↪ = CPMA and counting is down
218 EPwm3Regs.AQCILB.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↪ CPMA and counting is up
219 EPwm3Regs.AQCILB.bit.PRD = 0; //00 "Do nothing" Action when CTR =
    ↪ PRD
220 EPwm3Regs.AQCILB.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↪ ZERO
221 EPwm3Regs.AQSFRC.bit.RLDCSF = 0; //AQ load from shadow registers. 00
    ↪ is in CTR = 0
222 EPwm3Regs.AQSFRC.bit.OTSFB = 0; //OTSFB = 1 trigger a software event
    ↪ out B... 0 no action
223 EPwm3Regs.AQSFRC.bit.ACTSFB = 0; //Action when OTSFB = 1... 00 do
    ↪ nothing
224 EPwm3Regs.AQSFRC.bit.OTSFA = 0; //OTSFA = 1 trigger a software event
    ↪ out A... 0 no action
225 EPwm3Regs.AQSFRC.bit.ACTSFA = 0; //Action when OTSFA = 1... 00 do
    ↪ nothing
226 EPwm3Regs.AQSFRC.bit.CSFB = 0; //continuous force low/high on out B
    ↪ ... 00 forcing is disabled, 01 forces continuous low
227 EPwm3Regs.AQSFRC.bit.CSFA = 0; //continuous force low/high on out A
    ↪ ... 00 forcing is disabled, 01 forces continuous low
228 EDIS;
229 FALLOW;
230 EPwm3Regs.DBCIL.bit.IN_MODE = 0; //pwmA is the source for rise-edge
    ↪ and fall-edge for delay (dead-band)
231 EPwm3Regs.DBCIL.bit.POLSEL = 2; //Type of delay. 10 is Active High
    ↪ Complementary (FWMB is inverted) See report 2.5 fig 32

```

```

232 EPwm3Regs.DECTL.bit.OUT_MODE = 3; //dead-band is enabled for both rise
    ↳ and fall edges.
233 EPwm3Regs.DEFED = 1; //DEFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)
234 EPwm3Regs.DEFED = 1; //DEFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)
235 EDIS;
236 EALLOW;
237 EPwm3Regs.PCCCTL.bit.CHPDUTY = 0; //Chopping Clock Duty Cycle
238 EPwm3Regs.PCCCTL.bit.CHPFREQ = 0; //Chopping Clock Frequency Divider
239 EPwm3Regs.PCCCTL.bit.OWIWIH = 0; //One-shot Pulse Width (80ns) (see
    ↳ manual)
240 EPwm3Regs.PCCCTL.bit.CHPEN = 0; //Chopping disable. Bypass PWM
    ↳ chopping function
241 EDIS;
242 EALLOW;
243 EPwm3Regs.TZSEL.all = 0x0000; //Disable all Trip-Zone sources (not
    ↳ used)
244 EPwm3Regs.TZCTL.bit.TZB = 3; //11 Do nothing when a trip occurs (
    ↳ not used)
245 EPwm3Regs.TZCTL.bit.TZA = 3; //11 Do nothing when a trip occurs (
    ↳ not used)
246 EPwm3Regs.TZEINT.bit.OST = 0; //0 disable One shot interrupt (not
    ↳ used)
247 EPwm3Regs.TZEINT.bit.CBC = 0; //0 disable cycle-by-cycle interrupt
    ↳ (not used)
248 EDIS;
249 EALLOW;
250 EPwm3Regs.ETSEL.bit.SOCBEN = 1; //1 enable ADC start of conversion
    ↳ after a dPWMB pulse
251 EPwm3Regs.ETSEL.bit.SOCBSEL = 1; //Generate a ADC pulse when CTR = 0
252 EPwm3Regs.ETSEL.bit.SOCAEN = 1; //1 enable ADC start of conversion
    ↳ after a dPWMB pulse
253 EPwm3Regs.ETSEL.bit.SOCASEL = 1; //Generate a ADC pulse when CTR = 0
254 EPwm3Regs.ETSEL.bit.INTEN = 0; //1 Enable PWM interrupt generation (
    ↳ not used)
255 EPwm3Regs.ETSEL.bit.INTSEL = 0; //Generate interruption when CTR = 0
256 EPwm3Regs.ETPS.bit.SOCBPRD = 0; //Disable SOCB event counter
257 EPwm3Regs.ETPS.bit.SOCAPRD = 0; //Disable SOCA event counter
258 EPwm3Regs.ETPS.bit.INTPRD = 0; //Disable interrupt event counter.
259 EDIS;
260
261 // dPWM4
262 EALLOW;
263 EPwm4Regs.TBPRD = 1499; //Time-base period TBPRD ...
    ↳ Tpwm = (TBPRD+1)*TBCLK; TBCLK = 1/SYSCLKOUT = 1/150MHz; Tpwm = 10us
264 EPwm4Regs.TBPHS.half.TBPHS = 0x00; //Value, where the
    ↳ counter TBCTR will continue
265 EPwm4Regs.TBCTR = 0x00; //Initializes counter in 0
266 EPwm4Regs.TBCTL.bit.FREE_SOFT = 2; //Free run during debugging
267 EPwm4Regs.TBCTL.bit.PHSDIR = 1; //Count up after sync event
268 EPwm4Regs.TBCTL.bit.CLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV)
    ↳ ... 0 is /1 and 7 is /128...
269 EPwm4Regs.TBCTL.bit.HSPCLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV) ...
    ↳ 0 is /1 and 7 is /14...
270 EPwm4Regs.TBCTL.bit.SWFSYNC = 1; //Sync by SW... 0 has no effect. 1
    ↳ forces a one-time sync sw pulse
271 EPwm4Regs.TBCTL.bit.SYNCOSEL = 1; //Select the output Sync.. 0 is
    ↳ epwmSYNCO = epwmSYNC; 1 is SYNCO when CTR = 0;
272 EPwm4Regs.TBCTL.bit.PRDL = 0; //Enables the TBPRD shadow register
273 EPwm4Regs.TBCTL.bit.PHSEN = 0; //Don't load TBCTR from TBPHS
274 EPwm4Regs.TBCTL.bit.CTRMODE = 2; //Counter mode -> Up-down-count mode
275 EDIS;
276 EALLOW;
277 EPwm4Regs.CMPCCTL.bit.SHOWMODE = 0; //CPMB Shadow mode enable
278 EPwm4Regs.CMPCCTL.bit.SHVAMODE = 0; //CPMA Shadow mode enable
279 EPwm4Regs.CMPCCTL.bit.LOADMODE = 2; //CPMB is loaded from shadow at CTR =
    ↳ 0 and CTR = PRD;
280 EPwm4Regs.CMPCCTL.bit.LOADMODE = 2; //CPMA is loaded from shadow at CTR =
    ↳ 0 and CTR = PRD;
281 EDIS;

```

```

282 EALLOW;
283 EPwm4Regs.AQCILA.bit.CBD = 0; //00 "Do nothing" Action in pwmA when
    ↳ CTR = CPMB and counting is down
284 EPwm4Regs.AQCILA.bit.CBU = 0; //00 "Do nothing" Action in pwmA when
    ↳ CTR = CPMB and counting is up
285 EPwm4Regs.AQCILA.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↳ = CPMA and counting is down
286 EPwm4Regs.AQCILA.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↳ CPMA and counting is up
287 EPwm4Regs.AQCILA.bit.PRD = 0; //00 "Do nothing" Action when CTR =
    ↳ PRD
288 EPwm4Regs.AQCILA.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↳ ZERO
289 EPwm4Regs.AQCILB.bit.CBD = 0; //00 "Do nothing" Action in pwmB when
    ↳ CTR = CPMB and counting is down
290 EPwm4Regs.AQCILB.bit.CBU = 0; //00 "Do nothing" Action in pwmB when
    ↳ CTR = CPMB and counting is up
291 EPwm4Regs.AQCILB.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↳ = CPMA and counting is down
292 EPwm4Regs.AQCILB.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↳ CPMA and counting is up
293 EPwm4Regs.AQCILB.bit.PRD = 0; //00 "Do nothing" Action when CTR =
    ↳ PRD
294 EPwm4Regs.AQCILB.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↳ ZERO
295 EPwm4Regs.AQSFRC.bit.RLDCSF = 0; //AQ load from shadow registers. 00
    ↳ is in CTR = 0
296 EPwm4Regs.AQSFRC.bit.OTISFB = 0; //OTISFB = 1 trigger a software event
    ↳ out B... 0 no action
297 EPwm4Regs.AQSFRC.bit.ACISFB = 0; //Action when OTISFB = 1... 00 do
    ↳ nothing
298 EPwm4Regs.AQSFRC.bit.OTISFA = 0; //OTISFA = 1 trigger a software event
    ↳ out A... 0 no action
299 EPwm4Regs.AQSFRC.bit.ACISFA = 0; //Action when OTISFA = 1... 00 do
    ↳ nothing
300 EPwm4Regs.AQCSFRC.bit.CSFB = 0; //continuous force low/high on out B
    ↳ ... 00 forcing is disabled, 01 forces continuous low
301 EPwm4Regs.AQCSFRC.bit.CSFA = 0; //continuous force low/high on out A
    ↳ ... 00 forcing is disabled, 01 forces continuous low
302 EDIS;
303 EALLOW;
304 EPwm4Regs.DBCIL.bit.IN_MODE = 0; //PWMxA is the source for rise-edge
    ↳ and fall-edge for delay (dead-band)
305 EPwm4Regs.DBCIL.bit.POLSEL = 2; //Type of delay. 10 is Active High
    ↳ Complementary (FWMB is inverted) See report 2.5 fig 32
306 EPwm4Regs.DBCIL.bit.OUT_MODE = 3; //dead-band is enabled for both rise
    ↳ and fall edges.
307 EPwm4Regs.DBERD = 1; //DERED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)
308 EPwm4Regs.DBFED = 1; //DEFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)
309 EDIS;
310 EALLOW;
311 EPwm4Regs.PCCIL.bit.CHPDUTY = 0; //Chopping Clock Duty Cycle
312 EPwm4Regs.PCCIL.bit.CHPFREQ = 0; //Chopping Clock Frequency Divider
313 EPwm4Regs.PCCIL.bit.OSHIWIH = 0; //One-shot Pulse Width (80ns) (see
    ↳ manual)
314 EPwm4Regs.PCCIL.bit.CHPEN = 0; //Chopping disable. Bypass PWM
    ↳ chopping function
315 EDIS;
316 EALLOW;
317 EPwm4Regs.TZSEL.all = 0x0000; //Disable all Trip-Zone sources (not
    ↳ used)
318 EPwm4Regs.TZCIL.bit.TZB = 3; //11 Do nothing when a trip occurs (
    ↳ not used)
319 EPwm4Regs.TZCIL.bit.TZA = 3; //11 Do nothing when a trip occurs (
    ↳ not used)
320 EPwm4Regs.TZEINT.bit.OST = 0; //0 disable One shot interrupt (not
    ↳ used)
321 EPwm4Regs.TZEINT.bit.CBC = 0; //0 disable cycle-by-cycle interrupt
    ↳ (not used)

```

```

322 EDIS;
323 EALLOW;
324 EPwm4Regs.ETSEL.bit.SOCBEN = 1; //1 enable ADC start of conversion
    ↳ after a dPWMB pulse
325 EPwm4Regs.ETSEL.bit.SOCSEL = 1; //Generate a ADC pulse when CTR = 0
326 EPwm4Regs.ETSEL.bit.SOCAEN = 1; //1 enable ADC start of conversion
    ↳ after a dPWMB pulse
327 EPwm4Regs.ETSEL.bit.SOCASEL = 1; //Generate a ADC pulse when CTR = 0
328 EPwm4Regs.ETSEL.bit.INTEN = 0; //1 Enable PWM interrupt generation (
    ↳ not used)
329 EPwm4Regs.ETSEL.bit.INTSEL = 0; //Generate interruption when CTR = 0
330 EPwm4Regs.ETPS.bit.SOCBPRD = 0; //Disable SOCB event counter
331 EPwm4Regs.ETPS.bit.SOCAPRD = 0; //Disable SOCA event counter
332 EPwm4Regs.ETPS.bit.INTPRD = 0; //Disable interrupt event counter.
333 EDIS;
334
335 // dPWM4-5
336 EALLOW;
337 EPwm5Regs.TBPRD = 1499; //Time-base period TBPRD ...
    ↳ Tpwm = (TBPRD+1)*TBCLK; TBCLK = 1/SYSCLKOUT = 1/150MHz; Tpwm = 10us
338 EPwm5Regs.TBPHS.half.TBPHS = 0x00; //Value, where the
    ↳ counter TBCTR will continue
339 EPwm5Regs.TBCTR = 0x00; //Initializes counter in 0
340 EPwm5Regs.TBCTL.bit.FREE_SOFT = 2; //Free run during debugging
341 EPwm5Regs.TBCTL.bit.PHSDIR = 1; //Count up after sync event
342 EPwm5Regs.TBCTL.bit.CLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV
    ↳ ) ... 0 is /1 and 7 is /128...
343 EPwm5Regs.TBCTL.bit.HSPCLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV) ...
    ↳ 0 is /1 and 7 is /14...
344 EPwm5Regs.TBCTL.bit.SWFSYNC = 1; //Sync by SW... 0 has no effect. 1
    ↳ forces a one-time sync sw pulse
345 EPwm5Regs.TBCTL.bit.SYNCOSEL = 1; //Select the output Sync.. 0 is
    ↳ epwm5SYNCO = epwm5SYNC; 1 is SYNCO when CTR = 0;
346 EPwm5Regs.TBCTL.bit.PRDL = 0; //Enables the TBPRD shadow register
347 EPwm5Regs.TBCTL.bit.PHSEN = 0; //Don't load TBCTR from TBPHS
348 EPwm5Regs.TBCTL.bit.CTRMODE = 2; //Counter mode -> Up-down-count mode
349 EDIS;
350 EALLOW;
351 EPwm5Regs.CMPCCTL.bit.SHADOWMODE = 0; //CPMB Shadow mode enable
352 EPwm5Regs.CMPCCTL.bit.SHADWAMODE = 0; //CPMA Shadow mode enable
353 EPwm5Regs.CMPCCTL.bit.LOADEMODE = 2; //CPMB is loaded from shadow at CTR =
    ↳ 0 and CTR = PRD;
354 EPwm5Regs.CMPCCTL.bit.LOADAMODE = 2; //CPMA is loaded from shadow at CTR =
    ↳ 0 and CTR = PRD;
355 EDIS;
356 EALLOW;
357 EPwm5Regs.AQCTLA.bit.CBD = 0; //00 "Do nothing" Action in pwmA when
    ↳ CTR = CPMB and counting is down
358 EPwm5Regs.AQCTLA.bit.CBU = 0; //00 "Do nothing" Action in pwmA when
    ↳ CTR = CPMB and counting is up
359 EPwm5Regs.AQCTLA.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↳ = CPMA and counting is down
360 EPwm5Regs.AQCTLA.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↳ CPMA and counting is up
361 EPwm5Regs.AQCTLA.bit.PRD = 0; //00 "Do nothing" Action when CTR =
    ↳ PRD
362 EPwm5Regs.AQCTLA.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↳ ZERO
363 EPwm5Regs.AQCTLB.bit.CBD = 0; //00 "Do nothing" Action in pwmB when
    ↳ CTR = CPMB and counting is down
364 EPwm5Regs.AQCTLB.bit.CBU = 0; //00 "Do nothing" Action in pwmB when
    ↳ CTR = CPMB and counting is up
365 EPwm5Regs.AQCTLB.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↳ = CPMA and counting is down
366 EPwm5Regs.AQCTLB.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↳ CPMA and counting is up
367 EPwm5Regs.AQCTLB.bit.PRD = 0; //00 "Do nothing" Action when CTR =
    ↳ PRD
368 EPwm5Regs.AQCTLB.bit.ZRO = 0; //00 "Do nothing" Action when CTR =
    ↳ ZERO

```

```

369 EPwm5Regs.AQSFRC.bit.RLDCSF = 0;           //AQ load from shadow registers. 00
      ↳ is in CIR = 0
370 EPwm5Regs.AQSFRC.bit.OTSFb = 0;             //OTSFB = 1 trigger a software event
      ↳ out B... 0 no action
371 EPwm5Regs.AQSFRC.bit.ACTSFb = 0;            //Action when OTSFb = 1... 00 do
      ↳ nothing
372 EPwm5Regs.AQSFRC.bit.OTSFA = 0;            //OTSFA = 1 trigger a software event
      ↳ out A... 0 no action
373 EPwm5Regs.AQSFRC.bit.ACTSFA = 0;           //Action when OTSFA = 1... 00 do
      ↳ nothing
374 EPwm5Regs.AQSFRC.bit.CSFB = 0;             //continuous force low/high on out B
      ↳ ... 00 forcing is disabled, 01 forces continuous low
375 EPwm5Regs.AQSFRC.bit.CSFA = 0;            //continuous force low/high on out A
      ↳ ... 00 forcing is disabled, 01 forces continuous low
376 EDIS;
377 EALLOW;
378 EPwm5Regs.DBCIL.bit.IN_MODE = 0;           //dPWMxA is the source for rise-edge
      ↳ and fall-edge for delay (dead-band)
379 EPwm5Regs.DBCIL.bit.POLSEL = 2;            //Type of delay. 10 is Active High
      ↳ Complementary (FWMb is inverted) See report 2.5 fig 32
380 EPwm5Regs.DBCIL.bit.OUT_MODE = 3;          //dead-band is enabled for both rise
      ↳ and fall edges.
381 EPwm5Regs.DBERED = 1;                      //DERED = 1, dead-band = 0.01
      ↳ us (see report 2.5 tab 15)
382 EPwm5Regs.DBERED = 1;                      //DERED = 1, dead-band = 0.01
      ↳ us (see report 2.5 tab 15)
383 EDIS;
384 EALLOW;
385 EPwm5Regs.PCCIL.bit.CHPDUTY = 0;           //Chopping Clock Duty Cycle
386 EPwm5Regs.PCCIL.bit.CHPFREQ = 0;          //Chopping Clock Frequency Divider
387 EPwm5Regs.PCCIL.bit.CSHITWIH = 0;         //One-shot Pulse Width (80ns) (see
      ↳ manual)
388 EPwm5Regs.PCCIL.bit.CHPEN = 0;            //Chopping disable. Bypass PWM
      ↳ chopping function
389 EDIS;
390 EALLOW;
391 EPwm5Regs.TZSEL.all = 0x0000;              //Disable all Trip-Zone sources (not
      ↳ used)
392 EPwm5Regs.TZCIL.bit.TZB = 3;               //11 Do nothing when a trip occurs (
      ↳ not used)
393 EPwm5Regs.TZCIL.bit.TZA = 3;               //11 Do nothing when a trip occurs (
      ↳ not used)
394 EPwm5Regs.TZEINT.bit.OST = 0;              //0 disable One shot interrupt (not
      ↳ used)
395 EPwm5Regs.TZEINT.bit.CBC = 0;              //0 disable cycle-by-cycle interrupt
      ↳ (not used)
396 EDIS;
397 EALLOW;
398 EPwm5Regs.ETSEL.bit.SOCBEN = 1;            //1 enable ADC start of conversion
      ↳ after a dPWMb pulse
399 EPwm5Regs.ETSEL.bit.SOCBSEL = 1;           //Generate a ADC pulse when CIR = 0
400 EPwm5Regs.ETSEL.bit.SOCAEN = 1;           //1 enable ADC start of conversion
      ↳ after a dPWMb pulse
401 EPwm5Regs.ETSEL.bit.SOCASEL = 1;           //Generate a ADC pulse when CIR = 0
402 EPwm5Regs.ETSEL.bit.INTEN = 0;             //1 Enable PWM interrupt generation (
      ↳ not used)
403 EPwm5Regs.ETSEL.bit.INTSEL = 0;            //Generate interruption when CIR = 0
404 EPwm5Regs.ETPS.bit.SOCBPRD = 0;           //Disable SOCB event counter
405 EPwm5Regs.ETPS.bit.SOCAPRD = 0;           //Disable SOCA event counter
406 EPwm5Regs.ETPS.bit.INTPRD = 0;            //Disable interrupt event counter.
407 EDIS;
408
409 // dPWM4
410 EALLOW;
411 EPwm6Regs.TBPRD = 1499;                    //Time-base period TBPRD ...
      ↳ Tpwm = (TBPRD+1)*TBCLK; TBCLK = 1/SYSCLKOUT = 1/150MHz; Tpwm = 10us
412 EPwm6Regs.TBPHS.half.TBPHS = 0x00;        //Value, where the
      ↳ counter TBCTR will continue
413 EPwm6Regs.TBCIR = 0x00;                    //Initializes counter in 0
414 EPwm6Regs.TBCIL.bit.FREE_SOFT = 2;         //Free run during debugging
415 EPwm6Regs.TBCIL.bit.PHSDIR = 1;           //Count up after sync event

```

```

416 EPwm6Regs.TBCTL.bit.CLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV
    ↳ )... 0 is /1 and 7 is /128...
417 EPwm6Regs.TBCTL.bit.HSPCLKDIV = 0x00; //TBCLK = SYSCLKOUT/(HSPCLKDIV*CLKDIV)...
    ↳ 0 is /1 and 7 is /14...
418 EPwm6Regs.TBCTL.bit.SWFSYNC = 1; //Sync by SW... 0 has no effect. 1
    ↳ forces a one-time sync sw pulse
419 EPwm6Regs.TBCTL.bit.SYNCOSEL = 1; //Select the output Sync.. 0 is
    ↳ epwmSYNCO = epwmSYNCO; 1 is SYNCO when CIR = 0;
420 EPwm6Regs.TBCTL.bit.PRDL = 0; //Enables the TBPRD shadow register
421 EPwm6Regs.TBCTL.bit.PHSEN = 0; //Don't load TBCTR from TBPHS
422 EPwm6Regs.TBCTL.bit.CIRMODE = 2; //Counter mode -> Up-down-count mode
423 EDIS;
424 EALLOW;
425 EPwm6Regs.CMPCCTL.bit.SHADOWMODE = 0; //CPMB Shadow mode enable
426 EPwm6Regs.CMPCCTL.bit.SHADOWMODE = 0; //CPMA Shadow mode enable
427 EPwm6Regs.CMPCCTL.bit.LOADMODE = 2; //CPMB is loaded from shadow at CIR =
    ↳ 0 and CTR = PRD;
428 EPwm6Regs.CMPCCTL.bit.LOADMODE = 2; //CPMA is loaded from shadow at CIR =
    ↳ 0 and CTR = PRD;
429 EDIS;
430 EALLOW;
431 EPwm6Regs.AQCTLA.bit.CBD = 0; //00 "Do nothing" Action in pwmA when
    ↳ CTR = CPMB and counting is down
432 EPwm6Regs.AQCTLA.bit.CBU = 0; //00 "Do nothing" Action in pwmA when
    ↳ CTR = CPMB and counting is up
433 EPwm6Regs.AQCTLA.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↳ = CPMA and counting is down
434 EPwm6Regs.AQCTLA.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↳ CPMA and counting is up
435 EPwm6Regs.AQCTLA.bit.PRD = 0; //00 "Do nothing" Action when CIR =
    ↳ PRD
436 EPwm6Regs.AQCTLA.bit.ZRO = 0; //00 "Do nothing" Action when CIR =
    ↳ ZERO
437 EPwm6Regs.AQCTLB.bit.CBD = 0; //00 "Do nothing" Action in pwmB when
    ↳ CTR = CPMB and counting is down
438 EPwm6Regs.AQCTLB.bit.CBU = 0; //00 "Do nothing" Action in pwmB when
    ↳ CTR = CPMB and counting is up
439 EPwm6Regs.AQCTLB.bit.CAD = 1; //01 "Clear" Action in pwmA when CTR
    ↳ = CPMA and counting is down
440 EPwm6Regs.AQCTLB.bit.CAU = 2; //10 "Set" Action in pwmA when CTR =
    ↳ CPMA and counting is up
441 EPwm6Regs.AQCTLB.bit.PRD = 0; //00 "Do nothing" Action when CIR =
    ↳ PRD
442 EPwm6Regs.AQCTLB.bit.ZRO = 0; //00 "Do nothing" Action when CIR =
    ↳ ZERO
443 EPwm6Regs.AQSFRC.bit.RLDCSF = 0; //AQ load from shadow registers. 00
    ↳ is in CTR = 0
444 EPwm6Regs.AQSFRC.bit.OTSFB = 0; //OTSFB = 1 trigger a software event
    ↳ out B... 0 no action
445 EPwm6Regs.AQSFRC.bit.ACTSFB = 0; //Action when OTSFB = 1... 00 do
    ↳ nothing
446 EPwm6Regs.AQSFRC.bit.OTSFA = 0; //OTSFA = 1 trigger a software event
    ↳ out A... 0 no action
447 EPwm6Regs.AQSFRC.bit.ACTSFA = 0; //Action when OTSFA = 1... 00 do
    ↳ nothing
448 EPwm6Regs.AQSFRC.bit.CSFB = 0; //continuous force low/high on out B
    ↳ ... 00 forcing is disabled, 01 forces continuous low
449 EPwm6Regs.AQSFRC.bit.CSFA = 0; //continuous force low/high on out A
    ↳ ... 00 forcing is disabled, 01 forces continuous low
450 EDIS;
451 EALLOW;
452 EPwm6Regs.DBCCTL.bit.IN_MODE = 0; //EPWMA is the source for rise-edge
    ↳ and fall-edge for delay (dead-band)
453 EPwm6Regs.DBCCTL.bit.POLSEL = 2; //Type of delay. 10 is Active High
    ↳ Complementary (PWMB is inverted) See report 2.5 fig 32
454 EPwm6Regs.DBCCTL.bit.OUT_MODE = 3; //dead-band is enabled for both rise
    ↳ and fall edges.
455 EPwm6Regs.DBFED = 1; //DBFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)
456 EPwm6Regs.DBFED = 1; //DBFED = 1, dead-band = 0.01
    ↳ us (see report 2.5 tab 15)

```

```

457     EDIS;
458     EALLOW;
459     EPwm6Regs.PCCIL.bit.CHPDUTY = 0;           //Chopping Clock Duty Cycle
460     EPwm6Regs.PCCIL.bit.CHPFREQ = 0;          //Chopping Clock Frequency Divider
461     EPwm6Regs.PCCIL.bit.OSHTWIH = 0;          //One-shot Pulse Width (80ns) (see
        ↪ manual)
462     EPwm6Regs.PCCIL.bit.CHPEN = 0;            //Chopping disable. Bypass PWM
        ↪ chopping function
463     EDIS;
464     EALLOW;
465     EPwm6Regs.TZSEL.all = 0x0000;            //Disable all Trip-Zone sources (not
        ↪ used)
466     EPwm6Regs.TZCTL.bit.TZB = 3;             //11 Do nothing when a trip occurs (
        ↪ not used)
467     EPwm6Regs.TZCTL.bit.TZA = 3;            //11 Do nothing when a trip occurs (
        ↪ not used)
468     EPwm6Regs.TZEINT.bit.OST = 0;           //0 disable One shot interrupt (not
        ↪ used)
469     EPwm6Regs.TZEINT.bit.CBC = 0;           //0 disable cycle-by-cycle interrupt
        ↪ (not used)
470     EDIS;
471     EALLOW;
472     EPwm6Regs.ETSEL.bit.SOCBEN = 1;         //1 enable ADC start of conversion
        ↪ after a dFWMB pulse
473     EPwm6Regs.ETSEL.bit.SOCBSEL = 1;        //Generate a ADC pulse when CTR = 0
474     EPwm6Regs.ETSEL.bit.SOCAEN = 1;         //1 enable ADC start of conversion
        ↪ after a dFWMB pulse
475     EPwm6Regs.ETSEL.bit.SOCASEL = 1;        //Generate a ADC pulse when CTR = 0
476     EPwm6Regs.ETSEL.bit.INTEN = 0;          //1 Enable PWM interrupt generation (
        ↪ not used)
477     EPwm6Regs.ETSEL.bit.INTSEL = 0;         //Generate interruption when CTR = 0
478     EPwm6Regs.ETPS.bit.SOCBPRD = 0;        //Disable SOCB event counter
479     EPwm6Regs.ETPS.bit.SOCAPRD = 0;        //Disable SOCA event counter
480     EPwm6Regs.ETPS.bit.INTPRD = 0;         //Disable interrupt event counter.
481     EDIS;
482
483     EALLOW;
484     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC=1;    // for carriers sync
485     EDIS;
486 }

```

B.1.3 Configuração das GPIOs

```

1  /*
2  *  gpios.c
3  *
4  *  Created on: 06/07/2015
5  *  Author: Chrystian
6  */
7
8  #include <gpios.h>
9
10 void init_gpios(void){
11
12     EALLOW;
13
14     GpioCtrlRegs.GPACR0.all = 0x0000;       //All GPIO (0 to 15) sampling
        ↪ period = Tsysclkout
15
16     GpioCtrlRegs.GPADIR.all = 0x0000;       // GPIO0-GPIO31 are inputs
17     GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;     // only GPIO0 is output
18     GpioCtrlRegs.GPADIR.bit.GPIO2 = 1;     // only GPIO2 is output
19     GpioCtrlRegs.GPADIR.bit.GPIO11 = 1;    // only GPIO11 is output
20
21     GpioCtrlRegs.GPAQSEL1.all = 0x0000;    // ALL GPIO (0 to 15) Synch
        ↪ to SYSCLKOUT

```

```

22     GpioCtrlRegs.GPAPUD.all = 0x0000;           // Pullup's enabled GPIO0-
        ↳ GPIO31
23
24     GpioDataRegs.GPADAT.all = 0xFFFF;           //all outputs in 1, which is
        ↳ off, because of pull-up;
25     GpioDataRegs.GPADAT.bit.GPIO11 = 0;         //GPIO11 in 0, which is on;
26
27     //EPWM1A is GPIO 0
28     GpioCtrlRegs.GPAMUX1.all = 0x0000;         // GPIO functionality GPIO0-
        ↳ GPIO15
29     GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 1;         //GPAMUX1[1:0] = 01 uses
        ↳ GPIO0 as EPWM;
30     GpioCtrlRegs.GPAMUX1.bit.GPIO2 = 1;         //GPAMUX1[1:0] = 01 uses
        ↳ GPIO2 as EPWM;
31     EDIS;
32
33 }

```

B.1.4 Configuração das Interrupções

```

1  #include "DSP2833x_Device.h"
2  #include "DSP28x_Project.h"
3  #include <interrupts.h>
4
5  extern interrupt void cpu_timer0_isr(void);
6  extern interrupt void epwm1_timer_isr(void);
7  extern interrupt void epwm2_timer_isr(void);
8
9  void InitTimerInterrupt(void){
10
11     DINT;
12
13     EALLOW;
14     PieVectTable.TINT0 = &cpu_timer0_isr; //Timer Interrupt 0 (TINT0) receives
        ↳ &cpu_timer0_isr
15     EDIS;
16
17     InitCpuTimers(); // For this
        ↳ example, only initialize the Cpu Timers
18
19     #if (CPU_FREQ_150MHZ) // Configure CPU-
        ↳ Timer 0 to interrupt every 1000 milliseconds: 150MHz CPU Freq, 50
        ↳ millisecond Period (in uSeconds)
20     ConfigCpuTimer(&CpuTimer0, 150, 1000000);
21 #endif
22     #if (CPU_FREQ_100MHZ) // Configure CPU-
        ↳ Timer 0 to interrupt every 1000 milliseconds: 100MHz CPU Freq, 50
        ↳ millisecond Period (in uSeconds)
23     ConfigCpuTimer(&CpuTimer0, 100, 1000000);
24 #endif
25
26     CpuTimer0Regs.TCR.all = 0x4001; // Use write-only instruction
        ↳ to set TSS bit = 0
27
28     IER |= M_INT1; // Enable CPU
        ↳ INT1 which is connected to CPU-Timer 0:
29
30     PieCtrlRegs.PIEIER1.bit.INTx7 = 1; // Enable TINT0 in the PIE:
        ↳ Group 1 interrupt 7
31
32     // Enable global Interrupts and higher priority real-time debug events:
33     EINT; // Enable Global interrupt INIM
34     ERIM; // Enable Global realtime interrupt DBGM
35 }
36
37 void InitPWMInterrupt(void){

```

```

38     DINT;
39
40     EALLOW; // This is needed to write to EALLOW protected registers
41     PieVectTable.EPWM1_INT = &epwm1_timer_isr;
42     PieVectTable.EPWM2_INT = &epwm2_timer_isr;
43     EDIS;
44
45     EALLOW;
46     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0; //Stop all the TB clocks
47     EDIS;
48
49     EPwm1Regs.ETSEL.bit.INTSEL = 1; //Select INT on Zero
50     ↪ event
51     EPwm1Regs.ETSEL.bit.INTEN = 1; //Enable INT
52     EPwm1Regs.ETPS.bit.INIPRD = 1; //Generate INT on 1st
53     ↪ event
54     EPwm2Regs.ETSEL.bit.INTSEL = 1; //Select INT on Zero
55     ↪ event
56     EPwm2Regs.ETSEL.bit.INTEN = 1; //Enable INT
57     EPwm2Regs.ETPS.bit.INIPRD = 1; //Generate INT on 1st
58     ↪ event
59     EALLOW;
60     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1; //Start all the timers synced
61     IER |= 4;
62     ↪ //Enable CPU INT3 which is connected to EPWM1-6 INT:
63     PieCtrlRegs.PIEIER3.bit.INTx1 = 1; //Enable Interrupts
64     ↪ Group IER3 INT1 (dPWM1)
65     PieCtrlRegs.PIEIER3.bit.INTx2 = 1; //Enable Interrupts
66     ↪ Group IER3 INT1 (dPWM2)
67     EDIS;
68     EINT; ↪ // Enable Global interrupt INTM
69     ERIM; ↪ // Enable Global realtime interrupt DBGM
70 }

```

B.1.5 Configuração de sistema

```

1  #include "DSP2833x_Device.h"
2  #include "DSP2833x_Examples.h"
3  #include "Define.h"
4  #include <sys_config.h>
5  #include <watchdog.h>
6
7  #define ADC_Cal (void (*)(void)) 0x380080 // pointer for ADC calibration in OIP
8
9  #pragma CODE_SECTION(flash_config, "ramfuncs");
10
11  extern void DSP28x_usDelay(unsigned long Count); // Rotina de
12  ↪ delay em ms
13
14  void sys_config(void){
15     DINT; //Disable all Interrupts
16
17     EALLOW;
18     SysCtrlRegs.WDCR = 0x0068; //Disable Watchdog
19     EDIS;
20
21     InitPll(DSP28_PLLCR,DSP28_DIVSEL);
22

```

```

23     EALLOW;
24
25     //Set Clock for Normal and Low Power Operations
26     SysCtrlRegs.HISPCP.bit.HSPCLK = 0;           //High Clock
27     ↪ /1
28     SysCtrlRegs.LOSPCP.bit.LSPCLK = 5;           //Low Clock
29     ↪ /10
30
31     // XCLKOUT to SYSCLKOUT ratio. By default XCLKOUT = 1/4 SYSCLKOUT
32     /*XintfRegs.XINTCNF2.bit.XTIMCLK = 1;        //XTIMCLK =
33     ↪ SYSCLKOUT/2
34     XintfRegs.XINTCNF2.bit.CLKMODE = 1;          //XCLKOUT =
35     ↪ XTIMCLK/2
36     XintfRegs.XINTCNF2.bit.CLKOFF = 0;           // Enable
37     ↪ XCLKOUT*/
38
39     //Peripheral Clocks
40     SysCtrlRegs.PCLKCR0.bit.ADCENCLK = 1;        // ADC
41     (*ADC_Cal)();                                //ADC
42     ↪ Calibration: Calls ADC_Cal() Routine
43
44     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 0;        //Disable TBCLK within the
45     ↪ dPWM
46     SysCtrlRegs.PCLKCR1.bit.EPWMENCLK = 1;        //1 dPWM1 is clocked by
47     ↪ SYSCLKOUT, 0 dPWM1 is not clocked.
48     SysCtrlRegs.PCLKCR1.bit.EPWMENCLK = 1;        // dPWM2
49     SysCtrlRegs.PCLKCR1.bit.EPWMENCLK = 1;        // dPWM3
50     SysCtrlRegs.PCLKCR1.bit.EPWMENCLK = 1;        // dPWM4
51     SysCtrlRegs.PCLKCR1.bit.EPWMENCLK = 1;        // dPWM5
52     SysCtrlRegs.PCLKCR1.bit.EPWMENCLK = 1;        // dPWM6
53     SysCtrlRegs.PCLKCR0.bit.TBCLKSYNC = 1;        //Enable TBCLK within the
54     ↪ dPWM
55
56     SysCtrlRegs.PCLKCR3.bit.CPUTIMERENCLK = 1;    //CPU Timer 0
57     SysCtrlRegs.PCLKCR3.bit.CPUTIMERENCLK = 1;    //CPU Timer 1
58     SysCtrlRegs.PCLKCR3.bit.CPUTIMERENCLK = 1;    //CPU Timer 2
59
60     SysCtrlRegs.PCLKCR3.bit.DMAENCLK = 1;         //DMA Clock
61     SysCtrlRegs.PCLKCR3.bit.XINTFENCLK = 1;        //XTIMCLK
62     SysCtrlRegs.PCLKCR3.bit.GPIOPENCLK = 1;        //GPIO input clock
63
64     EDIS;
65 }
66
67 void flash_config(void){
68
69     EALLOW;
70
71     FlashRegs.FOFT.bit.ENPIPE = 1;                //Enable Flash Pipeline mode
72     ↪ to improve performance of code executed from Flash.
73
74     //
75     CAUTION
76     //Minimum waitstates required for the flash operating
77     //at a given CPU rate must be characterized by TI.
78     //Refer to the datasheet for the latest information.
79
80     // For 150 MHZ Operation
81     FlashRegs.FBANKWAIT.bit.PAGEWAIT = 5;          //Set the Paged Waitstate for the
82     ↪ Flash
83     FlashRegs.FBANKWAIT.bit.RANDWAIT = 5;          //Set the Random Waitstate for the
84     ↪ Flash
85     FlashRegs.FOIPWAIT.bit.OIPWAIT = 8;            //Set the Waitstate for the OIP
86
87     //
88     CAUTION
89     //ONLY THE DEFAULT VALUE FOR THESE 2 REGISTERS SHOULD BE USED
90     FlashRegs.FSIDBYWAIT.bit.SIDBYWAIT = 0x01FF;
91     FlashRegs.FACTIVEWAIT.bit.ACTIVEWAIT = 0x01FF;
92     EDIS;
93
94     //Force a pipeline flush to ensure that the write to

```

```

82 //the last register configured occurs before returning.
83 asm(" RPT #7 || NOP");
84 }
85
86 void InitPll(Uint16 val, Uint16 divsel){
87
88     if (SysCtrlRegs.PLLSTS.bit.MCLKSTS != 0){           //Make sure the PLL is not
89         ↪ running in limp mode
90         asm("          ESTOP0");                         //Missing
91         ↪ external clock has been detected. Replace this line with a
92         ↪ call to an appropriate SystemShutdown(); function.
93     }
94
95     if (SysCtrlRegs.PLLSTS.bit.DIVSEL != 0){             //DIVSEL MUST be 0 before
96         ↪ PLLCR can be changed from 0x0000. It is set to 0 by an external reset
97         ↪ XRSn. This puts us in 1/4
98         EALLOW;
99         SysCtrlRegs.PLLSTS.bit.DIVSEL = 0;
100         EDIS;
101     }
102
103     if (SysCtrlRegs.PLLCR.bit.DIV != val){               // Change the PLLCR
104         EALLOW;
105         SysCtrlRegs.PLLSTS.bit.MCLKOFF = 1;             // Before setting
106         ↪ PLLCR turn off missing clock detect logic
107         SysCtrlRegs.PLLCR.bit.DIV = val;
108         EDIS;
109
110         // Optional: Wait for PLL to lock.During this time the CPU will
111         ↪ switch to OSCCLK/2 until the PLL is stable. Once the PLL is
112         ↪ stable the CPU will switch to the new PLL value.
113
114         // This time-to-lock is monitored by a PLL lock counter.
115         // Code is not required to sit and wait for the PLL to lock. However,
116         ↪ if the code does anything that is timing critical, and
117         ↪ requires the correct clock be locked, then it is best to wait
118         ↪ until this switching has completed.
119
120         // Wait for the PLL lock bit to be set.
121         DisableDog();                                   //
122         ↪ Watchdog should be disabled
123
124         while(SysCtrlRegs.PLLSTS.bit.PLLLOCKS != 1){
125             // ServiceDog();
126         }
127
128         EALLOW;
129         SysCtrlRegs.PLLSTS.bit.MCLKOFF = 0;
130         EDIS;
131     }
132
133     if((divsel == 1)|| (divsel == 2)){                   // If switching to 1/2
134         EALLOW;
135         SysCtrlRegs.PLLSTS.bit.DIVSEL = divsel;
136         EDIS;
137     }
138
139     // NOTE: ONLY USE THIS SETTING IF PLL IS BYPASSED (I.E. PLLCR = 0) OR OFF If
140     ↪ switching to 1/1
141     // * First go to 1/2 and let the power settle
142     // The time required will depend on the system, this is only an example
143     // * Then switch to 1/1
144     if(divsel == 3){
145         EALLOW;
146         SysCtrlRegs.PLLSTS.bit.DIVSEL = 2;
147         DELAY_US(50L);
148         SysCtrlRegs.PLLSTS.bit.DIVSEL = 3;
149         EDIS;
150     }
151 }
152

```

B.1.6 Configuração do Watchdog

```

1  /*
2  * Watchdog.h
3  *
4  * Created on: 06/07/2015
5  * Author: Chrystian
6  */
7
8  ///#include "DSP2833x_Device.h"
9  #include <Watchdog.h>
10
11 void DisableDog(void)
12 {
13     EALLOW;
14     //SysCtrlRegs.WDCR= 0x0068;
15     SysCtrlRegs.WDCR= 0x00EF; //disables watchdog 0b-1,1,101,111 = flag (1 to reset
16     ↪ WD), enable (1 to disable), check (always 101), PS (111 for clk/64)
17     EDIS;
18 }
19
20 void ServiceDog(void)
21 {
22     EALLOW;
23     SysCtrlRegs.WKEY= 0x0055;
24     SysCtrlRegs.WKEY= 0x00AA;
25     EDIS;
26 }
27
28 void EnableDog(void)
29 {
30     EALLOW;
31     SysCtrlRegs.WDCR= 0x00AF; //enables watchdog 0b-1,0,101,111 = flag (1 to
32     ↪ reset WD), enable (0 to enable), check (always 101), PS (111 for clk
33     ↪ /64)
34     SysCtrlRegs.WKEY= 0x0055; //feeds watchdog
35     SysCtrlRegs.WKEY= 0x00AA; //feeds watchdog
36     EDIS;
37 }

```

B.2 CÓDIGOS PARA OBTENÇÃO DAS MATRIZES Q E R E DO LQG

B.2.1 Obtenção das matrizes de peso

```

1  clc, clear all, close all
2  %%
3  %field x vs y
4  Qi_min = 1e-3*[1 1 1];
5  ri_min = 1e-3;
6  Qi_max = 1e5*[1 1 1];
7  ri_max = 1e7;
8
9  Qv_min = 1e-3*[1 1];
10 rv_min = 1e-3;
11 Qv_max = 1e5*[1 1];
12 rv_max = 1e7;
13
14 Qkf_min = 1e-4*[1 1 1 1];
15 Rkf_min = 1e-4*[1 1];
16 Qkf_max = 1e5*[1 1 1 1];
17 Rkf_max = 1e7*[1 1];

```

```

18
19 bottom = [Qi_min Qv_min Qkf_min ri_min rv_min Rkf_min];
20 top = [Qi_max Qv_max Qkf_max ri_max rv_max Rkf_max];
21 goal = 0;
22
23 %%
24 %DForward Parameters
25 Ta = 1/20e3;
26 Vinnom = 311; N = 12; Lo = 42.5e-6; Co = 940e-6;
27 r1 = 12e-3; rc = 0.2e-3;
28 Vonom = 4.2; Vomax = 10; Iomin = 50e-3; Iomax = 3;
29 Vin = Vinnom/N;
30 Ro = Vomax/Iomin;
31 dnom = Vonom/Vin;
32
33 %Battery Parameters
34 R1 = 0.05; R2 = 0.12; Rs = 0.054; C1 = 400; C2 = 1500;
35
36 %DForward + Battery State-Space Model
37 Ac = [0 -1/Lo 0 0;
38       1/Co -1/(Rs*Co) -1/(Rs*Co) -1/(Rs*Co);
39       0 -1/(Rs*C1) -(1/(Rs*C1)+1/(R1*C1)) -1/(Rs*C1);
40       0 -1/(Rs*C2) -1/(Rs*C2) -(1/(Rs*C2)+1/(R1*C2))];
41 Bc = [Vin/Lo 0; 0 1/(Rs*Co); 0 1/(Rs*C1); 0 1/(Rs*C2)];
42 C = [1 0 0 0];
43 D = [0 0];
44
45 %Discretization DForward + Battery
46 [A,B] = c2dm(Ac,Bc,C,D,Ta);
47
48 %%
49 %Dforward vo_iL / d Model
50 Aci_df = [0 -1/Lo; 1/Co -1/(Ro*Co)];
51 Bci_df = [Vin/Lo; 0];
52 Cdf_i = [1 0];
53 Cdf_v = [0 1];
54 Ddf = 0;
55
56 %DForward vo / iL Model
57 Acv_df = -1/(Ro*Co);
58 Bcv_df = 1/Co;
59
60 %Discretization DForward
61 [Adf, Bdf] = c2dm(Aci_df,Bci_df,Cdf_i,Ddf,Ta);
62 [Avdf,Bvdf] = c2dm(Acv_df,Bcv_df,1,Ddf,Ta);
63
64 %%
65 %Extended model
66 Aia = [Adf zeros(2,1); Cdf_i*Adf 1];
67 Bia = [Bdf; Cdf_i*Bdf];
68 Cia = [0 0 1];
69 Dia = 0;
70
71 Ava = [Avdf 0; 1*Avdf 1];
72 Bva = [Bvdf; 1*Bvdf];
73 Cva = Cdf_v;
74 Dva = 0;
75
76 %%
77 %Extended Kalman Filter Model
78 Akf = [Adf zeros(2,2); [Cdf_i;Cdf_v]*Adf eye(2)];
79 Bkf = [Bdf; [Cdf_i;Cdf_v]*Bdf];
80 Ckf = [0 0 1 0; 0 0 0 1];
81 Dkf = zeros(2);
82
83 %%
84 %swarm
85 particle_number = 15;
86 particle_parameters = 13;
87 for j = 1:particle_parameters

```

```

88     particles(j,:) = bottom(j)*ones(1,particle_number)+(top(j)-bottom(j))*rand(1,
      ↳ particle_number);
89 end
90 particles_dpos = zeros(particle_parameters,particle_number);
91 best_particle = randi(particle_number,1,1); %initialize best particle
92 particles_best = 1*rand(particle_parameters, particle_number) + 1e-6*ones(
      ↳ particle_parameters, particle_number); %initialize the best past of each
      ↳ particle
93 %particles(:,best_particle) = [10 1 0.001 1 10 0.01 0.001 1 0.001 0.1 5000000 10000
      ↳ 5000000 1000]';
94 particles(:,best_particle) = [1 100 1 1000 1 100 10 0.001 0.001 500 150 700 50]';
95
96 m=0;
97 error = 1;
98
99 while m < 50
100     m = m+1
101     error_acum = 0;
102     for i=1:l:particle_number %para cada uma das 20 candidatas
103         i
104         if particles(:,i) ~= particles(:,best_particle) %se a candidata i for
            ↳ diferente da melhor candidata
105             for j=1:l:particle_parameters
106                 particles_dpos(j,i) = 0.3*rand*(particles(j,best_particle) -
                    ↳ particles(j,i)) + 0.45*rand*(particles_best(j,i) -
                    ↳ particles(j,i)) + 0.25*particles_dpos(j,i); %calcula a
                    ↳ derivada da posicao para todas as particulas
107                 if (particles_dpos(j,i)+particles(j,i)) > top(j) || (particles_dpos(j
                    ↳ ,i)+particles(j,i)) < bottom(j)
108                     particles_dpos(j,i) = -particles_dpos(j,i); %inverte o sentido
                    ↳ caso a partícula extrapole o campo de busca
109                 end
110                 particles(j,i) = particles_dpos(j,i) + particles(j,i); %atualiza a
                    ↳ posicao das particulas
111                 if particles(j,i) < 0
112                     particles(j,i) = -particles(j,i)
113                 end
114             end
115         end
116         z_best = PSO_desired_function(particles_best(:,i),A,B,Aia,Bia,Ava,Bva,Akf,Bkf
            ↳ ,Ckf,Ta) %o melhor valor dentre os já encontrados pela própria
            ↳ partícula
117         best_z = PSO_desired_function(particles(:,best_particle),A,B,Aia,Bia,Ava,Bva,
            ↳ Akf,Bkf,Ckf,Ta) %o melhor valor de todas as particulas
118         z = PSO_desired_function(particles(:,i),A,B,Aia,Bia,Ava,Bva,Akf,Bkf,Ckf,Ta) %
            ↳ o valor atual encontrado da partícula i
119         % if abs(z_best) < abs(best_z)
120         %     best_particle = i;
121         % end
122         if abs(z_best) < abs(best_z)
123             best_particle = i;
124             particles(:,i) = particles_best(:,i);
125         end
126         if abs(z) < abs(z_best)
127             particles_best(:,i) = particles(:,i);
128         end
129         if abs(z) < abs(best_z)
130             best_particle = i;
131         end
132         error_acum = abs(z_best) - goal + error_acum;
133     end
134     error = error_acum/i;
135 end
136
137 %movie(movie1,1,2)
138 best_particle
139 QR_IGG = particles(:,best_particle)';
140 best_z
141
142 particles(:,best_particle)
143

```

```

144 save best_z.dat QR_LQG -ascii;
145 %% = x^2 + y^2

```

B.2.2 Obtenção dos ganhos do LQG

```

1  %Charger Project
2  %Forward Converter
3  clear all, close all
4  clc
5
6  QR_LQG = load('best_z.dat')
7  %%
8  %DForward Parameters
9  Ta = 1/20e3; fs = 100e3;
10 Vinnom = 311; N = 12; Lo = 42.5e-6; Co = 940e-6;
11 r1 = 12e-3; rc = 0.2e-3;
12 Vonom = 4.2; Vomax = 10; Iomin = 50e-3; Iomax = 3;
13 Vin = Vinnom/N;
14 Ro = Vomax/Iomin;
15 dnom = Vonom/Vin;
16
17 %Battery Parameters
18 R1 = 0.05; R2 = 0.12; Rs = 0.054; C1 = 400; C2 = 1500;
19
20 %%
21 %Sensor Gains
22 %Vo Converter Voltage
23 Gain_Vo = 0.27;
24 Vo_sense_max = Vomax*Gain_Vo
25 fc_Vo_sense = 10.526e3;
26
27 %IL Converter Current
28 Rshunt = 0.01; %Ohm
29 Vsh_max = Rshunt*12; %V
30 Gain_Op_IL = 23.5; %W/V
31 Gain_IL = Rshunt*Gain_Op_IL
32 IL_sense_max = Gain_IL*12
33 fc_IL_sense = 10.261e3;
34
35 %Vin Converter Line Voltage
36 R_sensor_in = 4.7e3;
37 R_led_in = 100e3;
38 Gain_R_Vin = R_sensor_in/(R_sensor_in+R_led_in)
39 Gain_Vin_Op = 0.2;
40 Gain_Vin = Gain_R_Vin*Gain_Vin_Op
41 Vin_sense_max = Gain_Vin*(Vinnom*1.048)
42 fc_Vin_sense = 11.7e3;
43
44 %Vbat Battery Voltage
45 Gain_Vbat = 0.47;
46 Vbat_sense_max = 4.2*Gain_Vbat
47 fc_Vbat_sense = 10.261e3;
48
49 %Ibat Battery Current
50 R_fuse = 0.045;
51 Gain_Ibat_Op = 17.407;
52 Gain_Ibat = R_fuse*Gain_Ibat_Op
53 Ibat_sense_max = Gain_Ibat*3
54 fc_Ibat_sense = 10.261e3;
55
56 %%
57 %DForward + Battery State-Space Model
58 Ac = [0 -1/Lo 0 0;
59       1/Co -1/(Rs*Co) -1/(Rs*Co) -1/(Rs*Co);
60       0 -1/(Rs*C1) -(1/(Rs*C1)+1/(R1*C1)) -1/(Rs*C1);

```

```

62     0 -1/(Rs*C2) -1/(Rs*C2) -(1/(Rs*C2)+1/(R1*C2))];
63 Bc = [Vin/L0 0; 0 1/(Rs*Co); 0 1/(Rs*C1); 0 1/(Rs*C2)];
64 C = [1 0 0 0];
65 D = [0 0];
66
67 %Discretization DForward + Battery
68 [A,B] = c2dm(Ac,Bc,C,D,Ta);
69
70 %%
71 %Dforward vo_iL / d Model
72 Aci_df = [0 -1/L0; 1/Co -1/(Ro*Co)];
73 Bci_df = [Vin/L0; 0];
74 Cdf_i = [1 0];
75 Cdf_v = [0 1];
76 Ddf = 0;
77
78 %DForward vo / iL Model
79 Acv_df = -1/(Ro*Co);
80 Bcv_df = 1/Co;
81
82 %Discretization DForward
83 [Adf, Bdf] = c2dm(Aci_df,Bci_df,Cdf_i,Ddf,Ta);
84 [Avdf,Bvdf] = c2dm(Acv_df,Bcv_df,1,Ddf,Ta);
85
86 %%
87 %Extended model
88 Aia = [Adf zeros(2,1); Cdf_i*Adf 1];
89 Bia = [Bdf; Cdf_i*Bdf];
90 Cia = [0 0 1];
91 Dia = 0;
92
93 Ava = [Avdf 0; 1*Avdf 1];
94 Bva = [Bvdf; 1*Bvdf];
95 Cva = Cdf_v;
96 Dva = 0;
97
98 %%
99 %Controller Project
100 Mci = ctrb(Aia,Bia);
101 rank(Mci)
102 Mcv = ctrb(Ava,Bva);
103 rank(Mcv)
104
105 % Qia = diag([1 100 1]);
106 % Ria = 500;
107 Qia = diag(QR_LQG(1:3));
108 Ria = QR_LQG(10);
109
110 Kia = dlqr(Aia,Bia,Qia,Ria)
111
112 % Qva = diag([1000 1]);
113 % Rva = 150;
114 Qva = diag(QR_LQG(4:5));
115 Rva = QR_LQG(11);
116
117 Kva = dlqr(Ava,Bva,Qva,Rva)
118
119 %%
120 %Bode Plots LQR Controller
121 bodeopts = bodeoptions;
122 bodeopts.FreqUnits = 'Hz';
123 bodeopts.Grid = 'on';
124
125 [Nz_Gi,Dz_Gi] = ss2tf(Adf,Bdf,Cdf_i,Ddf);
126 [Nz_Gv,Dz_Gv] = ss2tf(Adf,Bdf,Cdf_v,Ddf);
127 Gidz = tf(Nz_Gi,Dz_Gi,Ta);
128 Gvdz = tf(Nz_Gv,Dz_Gv,Ta);
129
130 [Nz_iOL,Dz_iOL] = ss2tf(Aia,Bia,Kia,Ddf);
131 [Nz_vOL,Dz_vOL] = ss2tf(Ava,Bva,Kva,Ddf);
132 CGidz_OL = tf(Nz_iOL,Dz_iOL,Ta);

```

```

133 CGvdz_OL = tf(Nz_vOL,Dz_vOL,Ta);
134
135 [Nz_iCL,Dz_iCL] = ss2tf(Aia-Bia*Kia,Bia*Kia(3),Cia,Ddf);
136 [Nz_vCL,Dz_vCL] = ss2tf(Ava-Bva*Kva,Bva*Kva(2),Cva,Ddf);
137 CGidz_CL = tf(Nz_iCL,Dz_iCL,Ta);
138 CGvdz_CL = tf(Nz_vCL,Dz_vCL,Ta);
139
140 figure
141 bode(Gidz,CGidz_OL,CGidz_CL,bodeopts)
142 legend('Gi(z)','Open Loop Sys','Closed Loop Sys')
143 set(findall(gcf,'type','line'),'linewidth',2)
144 figure
145 bode(Gvdz,CGvdz_OL,CGvdz_CL,bodeopts)
146 legend('Gv(z)','Open Loop Sys','Closed Loop Sys')
147 set(findall(gcf,'type','line'),'linewidth',2)
148
149 %%
150 %Kalman Filter Project
151 Akf = [Adf zeros(2,2); [Cdf_i;Cdf_v]*Adf eye(2)]
152 Bkf = [Bdf; [Cdf_i;Cdf_v]*Bdf]
153 Ckf = [0 0 1 0;0 0 0 1]
154 Dkf = zeros(2)
155
156 Mo = obsv(Akf,Ckf);
157 rank(Mo);
158
159 % Qkf = diag([100 10 0.001 0.001]);
160 % Rkf = diag([700 50]);
161 Qkf = diag(QR_LQG(6:9));
162 Rkf = diag(QR_LQG(12:13));
163
164 L = dlqr(Akf',Ckf',Qkf,Rkf)';
165
166 %Bode Plots Kalman Filter
167 [Nz_kfi,Dz_kfi] = ss2tf(Akf,L,Ckf(1,:),Dkf(1,:),1);
168 [Nz_kfv,Dz_kfv] = ss2tf(Akf,L,Ckf(2,:),Dkf(2,:),2);
169 KFi_OL = tf(Nz_kfi,Dz_kfi,Ta);
170 KFv_OL = tf(Nz_kfv,Dz_kfv,Ta);
171
172 [Nz_kfi_CL,Dz_kfi_CL] = ss2tf(Akf-L*Ckf,L,Ckf(1,:),Dkf(1,:),1);
173 [Nz_kfv_CL,Dz_kfv_CL] = ss2tf(Akf-L*Ckf,L,Ckf(2,:),Dkf(2,:),2);
174 KFi_CL = tf(Nz_kfi_CL,Dz_kfi_CL,Ta);
175 KFv_CL = tf(Nz_kfv_CL,Dz_kfv_CL,Ta);
176
177 figure
178 bode(Gidz,KFi_OL,KFi_CL,bodeopts)
179 legend('Gi(z)','Open Loop KF','Closed Loop KF')
180 set(findall(gcf,'type','line'),'linewidth',2)
181 figure
182 bode(Gvdz,KFv_OL,KFv_CL,bodeopts)
183 legend('Gv(z)','Open Loop KF','Closed Loop KF')
184 set(findall(gcf,'type','line'),'linewidth',2)
185
186 %%
187 %Initialization
188 t = 0:Ta:0.05;
189 n = length(t)
190
191 Xa_ant = zeros(4,1);
192 du_ant = zeros(2,1);
193
194 il = zeros(1,n);
195 vo = zeros(1,n); vo(1:3) = Bat_calculate_Voc(0.9)*ones(1,3);
196
197 Xa_est_ant = zeros(4,1);
198 il_est = zeros(1,n);
199 vo_est = zeros(1,n);
200
201 ib = zeros(1,n);
202 SOC = zeros(1,n); SOC(1:3) = 0.9;

```

```

203 Voc = zeros(1,n); Voc(1) = Bat_calculate_Voc(SOC(1)); Voc(2) = Bat_calculate_Voc(SOC
    ↪ (2)); Voc(3) = Bat_calculate_Voc(SOC(3));
204 dVoc = 0;
205
206 ui = zeros(1,n);
207 dui = zeros(1,n);
208 uv = zeros(1,n);
209 duv = zeros(1,n);
210
211 ri = 3;
212 rv = 4.2;
213
214 %%
215 %Iterative section
216 for k=3:n
217     if mod(k,10000)==0
218         k
219     end
220
221 %DForward + Battery Dynamics
222 Xa = A*Xa_ant + B*du_ant;
223 il(k) = il(k-1) + Xa(1);% + i_rand(k) + 0.75*sin(w*k*Ta);
224 vo(k) = vo(k-1) + Xa(2);% + v_rand(k) + 15e-3*sin(w*k*Ta);
225 %Diode restriction
226 if il(k) < 0
227     il(k) = 0;
228     Xa(1) = il(k) - il(k-1);
229     %Xa(1) = dil(k);
230 end
231
232 %Battery VOC and SOC update
233 ib(k) = -il(k) + Co/Ta*Xa(2);
234 SOC(k) = Bat_calculate_SOC(ib(k),SOC(k-1),Ta);
235 Voc(k) = Bat_calculate_Voc(SOC(k));
236 dVoc = Voc(k) - Voc(k-1);
237
238 %Kalman Filter dynamics
239 Xa_est = Akf*Xa_est_ant + Bkf*dui(k-1) + L*([il(k-1);vo(k-1)] - [il_est(k-1);
    ↪ vo_est(k-1)]);
240 il_est(k) = Xa_est(3);
241 vo_est(k) = Xa_est(4);
242 %Diode restriction for estimated il
243 if il_est(k) < 0
244     il_est(k) = 0;
245     Xa_est(1) = il_est(k) - il_est(k-1);
246 end
247
248 %calculate reference through vo/il controler
249 duv(k) = -Kva(1)*Xa_est(2) - Kva(2)*(vo_est(k) - rv);
250 uv(k) = uv(k-1) + duv(k);
251
252 %decide which reference to use
253 if uv(k) > ri
254     uv(k) = ri;
255 else if uv(k) < 0
256     uv(k) = 0;
257 end
258 end
259
260 %Control Action Calculus
261 dui(k) = -Kia(1:2)*Xa_est(1:2) - Kia(3)*(il_est(k) - uv(k));
262
263 %saturation
264 ui(k) = ui(k-1) + dui(k);
265 if ui(k) > 0.45
266     ui(k) = 0.45;
267     dui(k) = ui(k) - ui(k-1);
268 else if ui(k) < 0
269     ui(k) = 0;
270     dui(k) = ui(k) - ui(k-1);
271 end

```

```

272     end
273
274     %Update
275     Xa_ant = Xa;
276     du_ant = [dui(k-1); dVoc]; %du(k-1) because the DSP 1/z delay
277     Xa_est_ant = Xa_est;
278 end
279
280 %%
281 %Plots Section
282 m = n-1;
283 n_p = 1000;
284
285 figure
286 subplot(311)
287 plot(t(1:m/n_p:n),vo(1:m/n_p:n),'b','linewidth',1.5);
288 hold on
289 plot(t(1:m/n_p:n),il(1:m/n_p:n),'g','linewidth',1.5);
290 hold on
291 plot(t(1:m/n_p:n),vo_est(1:m/n_p:n),'b--','linewidth',1.5);
292 hold on
293 plot(t(1:m/n_p:n),il_est(1:m/n_p:n),'g--','linewidth',1.5);
294 legend('vo','il')
295
296 subplot(312)
297 plot(t(1:m/n_p:n),dui(1:m/n_p:n),'r-','linewidth',1.5)
298 hold on
299 plot(t(1:m/n_p:n),ui(1:m/n_p:n),'r','linewidth',1.5)
300 legend('u')
301
302 subplot(313)
303 plot(t(1:m/n_p:n),duv(1:m/n_p:n),'r-','linewidth',1.5)
304 hold on
305 plot(t(1:m/n_p:n),uv(1:m/n_p:n),'r','linewidth',1.5)
306 legend('u')
307
308 %
309 % figure
310 % subplot(211)
311 % plot(t(1:m/n_p:n),vo(1:m/n_p:n),'b','linewidth',1.5)
312 % hold on
313 % plot(t(1:m/n_p:n),Voc(1:m/n_p:n),'r','linewidth',1.5)
314 % legend('vo','Voc')
315 %
316 % subplot(212)
317 % plot(t(1:m/n_p:n),-ib(1:m/n_p:n),'g--','linewidth',1.5)

```

B.3 FUNÇÕES DO CONTROLADOR LQG IMPLEMENTADOS EM C

```

1  #include <lqg.h>
2
3  float lqr_controller_v(float dXv_est, float vo_est, float uv_ant, float ri, float rv)
4  ↪ {
5
6      float duv, uv;
7
8      //calculate reference through vo/il controler
9      duv = -Kva1*dXv_est -Kva2*(vo_est - rv);
10     uv = uv_ant + duv;
11
12     //decide which reference to use
13     if (uv > ri){
14         uv = ri;
15     }
16 }

```

```

14     }
15     else if (uv < 0){
16         uv = 0;
17     }
18
19     return uv;
20 }
21
22 float lqr_controller_i(float dXi_est, float dXv_est, float iL_est, float uv, float
    ↪ ui_ant){
23
24     float dui, ui;
25
26     dui = -Kia1*dXi_est -Kia2*dXv_est -Kia3*(iL_est - uv);
27     ui = ui_ant + dui;
28
29     return ui;
30 }
31
32
33 float saturation(float u, float umin, float umax){
34     if (u > umax){
35         u = umax;
36         return u;
37     } else if (u < umin){
38         u = umin;
39         return u;
40     } else{
41         return u;
42     }
43 }

```

```

1  #include <lqg.h>
2
3  float* kalman_filter(float dXi_est_ant, float dXv_est_ant, float iL_est_ant, float
    ↪ vo_est_ant, float iL_ant, float vo_ant, float dui_ant, float *X_est){
4
5     float dXi_est, dXv_est, iL_est, vo_est;
6
7     dXi_est = 0.968878126705188*dXi_est_ant - 1.163991008622261*dXv_est_ant +
    ↪ 30.173219575190053*dui_ant + L11*(iL_ant - iL_est_ant) + L12*(vo_ant
    ↪ - vo_est_ant);
8     dXv_est = 0.052627253049411*dXi_est_ant - 0.968457108680792*dXv_est_ant +
    ↪ 0.806575216223891*dui_ant + L21*(iL_ant - iL_est_ant) + L22*(vo_ant -
    ↪ vo_est_ant);
9     iL_est = 0.968878126705188*dXi_est_ant - 1.163991008622261*dXv_est_ant +
    ↪ iL_est_ant + 30.173219575190053*dui_ant + L31*(iL_ant - iL_est_ant) +
    ↪ L32*(vo_ant - vo_est_ant);
10    vo_est = 0.052627253049411*dXi_est_ant - 0.968457108680792*dXv_est_ant +
    ↪ vo_est_ant + 0.806575216223891*dui_ant + L41*(iL_ant - iL_est_ant) +
    ↪ L42*(vo_ant - vo_est_ant);
11
12    //diode restriction
13    if (iL_est < 0){
14        iL_est = 0;
15        dXi_est = iL_est - iL_est_ant;
16    }
17
18    *X_est = dXi_est;
19    *(X_est+1) = dXv_est;
20    *(X_est+2) = iL_est;
21    *(X_est+3) = vo_est;
22    return X_est;
23 }

```

B.4 CÓDIGO MAIN.C DO CONTROLADOR E SUAS PROTEÇÕES SEM ESTIMAÇÃO DO SOC

```

1  #include <adc.h>
2  #include "DSP2833x_Device.h"
3  #include "DSP28x_Project.h"
4  #include "C28x_FPU_FastRTS.h"
5  #include "stdio.h"
6  #include "Define.h"
7  #include "Structs.h"          ///<
8  #include "C28x_FPU_FastRTS.h"
9  #include <math.h>
10 #include "Const_disc8.h"
11 #include <epwm_config.h>
12 #include <gpios.h>
13 #include <sys_config.h>
14 #include <Watchdog.h>
15 #include <interrupts.h>
16 #include <lqg.h>
17
18 #define ref_i 1
19 #define ref_v 4.2
20 #define n_parallel 3
21 #define counts_prot 20000
22 #define vb_min 2.4
23 #define vb_max 4.25
24 #define iL_min 0.05
25 #define iL_max 12
26 #define vo_max 10
27
28 void InitTimerInterrupt(void);
29 void protection_counters(void);
30 void protection_verification(void);
31 void protection_on(void);
32 void protection_off(void);
33 void update_variables(void);
34 void initialize_references(void);
35 void verify_end_of_charge(void);
36 extern void DSP28x_usDelay(unsigned long Count);          ///< Rotina de
   ↳ delay em ms
37 interrupt void cpu_timer0_isr(void);
38 interrupt void epwm1_timer_isr(void);
39 interrupt void epwm2_timer_isr(void);
40
41 Uint16 i=0;
42 Uint16 start_control=0;
43
44 float uv = 0, u = 0, du = 0;
45 float vo=0, iL=0, vin=0, vo_ant = 0, iL_ant = 0;
46 float vb1, vb2, vb3, vb4, vb5;
47 float vb_l=0, vb=0, vb_ant=0;
48
49 float dXi_est_ant = 0, dXv_est_ant = 0, iL_est_ant = 0, vo_est_ant = 0;
50 float vb1_est=0, vb2_est=0, vb_est=0;
51 float dXi_est = 0, dXv_est = 0, iL_est = 0, vo_est = 0;
52 float vb1_est_ant=0, vb2_est_ant=0, vb_est_ant=0;
53
54 float uv_ant = 0, u_ant = 0, du_ant = 0;;
55 float ri = 0, rv = 0;
56
57 float a[4] = {0,0,0,0};
58 float *X_est;
59
60 Uint16 count_overvoltage=0, count_undervoltage=0, count_overload=0, count_shortC=0,
   ↳ count_turn_off_protection=0;
61 Uint16 count_overvoltage_vb=0, count_undervoltage_vb=0, count_lowcurrent=0;
62 Uint16 flag_overvoltage=0, flag_undervoltage=0, flag_overcurrent=0, flag_shortC=0;
63

```

```

64 float u_graph[501];
65 Uint16 k=0;
66
67
68 void main(void){
69
70     X_est = &a[0];
71
72     DisableDog(); //Disables Watchdog
73     sys_config();
74
75     DINT;
76
77     InitPieCtrl();
78
79     InitPieVectTable();
80
81     InitTimerInterrupt();
82
83     DSP28x_usDelay(10000000);
84
85     init_gpios(); //Initialize GPIOS, GPIO11 as out
86     adc_config();
87     epwm_config();
88
89     DSP28x_usDelay(10000000);
90
91     InitPWMInterrupt();
92
93     for(;;){
94
95     }
96 }
97
98
99 void protection_verification(void){
100     if (count_oversvoltage > counts_prot || count_oversvoltage_vb > counts_prot ||
        ↪ count_oversvoltage > counts_prot || count_shortC > counts_prot ||
        ↪ count_undersvoltage_vb>counts_prot){
101         count_turn_off_protection=0;
102         protection_on();
103         if (count_oversvoltage > counts_prot || count_oversvoltage_vb >
            ↪ counts_prot){
104             flag_oversvoltage = 1; count_oversvoltage = 0;
            ↪ count_oversvoltage_vb = 0;
105         }
106         else{
107             if (count_undersvoltage_vb > counts_prot){
108                 flag_undersvoltage = 1; count_undersvoltage_vb = 0;
109             }
110             else{
111                 if (count_oversvoltage > counts_prot){
112                     flag_oversvoltage = 1; count_oversvoltage = 0;
113                 }
114                 else{
115                     flag_shortC = 1; count_shortC = 0;
116                 }
117             }
118         }
119     } else{
120         count_turn_off_protection++;
121         if (count_turn_off_protection > 10){
122             count_turn_off_protection=0;
123             protection_off();
124         }
125     }
126 }
127
128
129 void protection_counters(void){

```

```

130         if (vo > vo_max) count_oversvoltage++;           //Oversvoltage
131         ↪ protection counter
132         if (vb > vb_max) count_oversvoltage_vb++;         //Oversvoltage
133         ↪ protection counter
134         if (vb_l < vb_min) count_undersvoltage_vb++;      //Oversvoltage
135         ↪ protection counter
136         if (iL > iL_max) count_overload++;                 //Overcurrent
137         ↪ protection counter
138         if (iL < iL_min) count_lowcurrent++;
139         if (vo < 0.1 && iL > 1) count_shortC++;           //Short Circuit protection
140         ↪ counter
141     }
142
143     void protection_on(void){
144         du = -u;
145         ri = 0; rv = 0; u = 0; uv = 0; u_ant = 0; uv_ant = 0;
146         count_oversvoltage = 0;
147         EALLOW;
148         EPwm1Regs.TZFRM.bit.OST = 1;                     //Force Interrupt
149         EDIS;
150     }
151
152     void protection_off(void){
153         initialize_references();
154         count_oversvoltage = 0; count_undersvoltage = 0; count_overload = 0;
155         ↪ count_shortC = 0;
156         flag_oversvoltage = 0; flag_undersvoltage = 0; flag_overcurrent = 0;
157         ↪ flag_shortC = 0;
158         EALLOW;
159         EPwm1Regs.TZCLR.bit.OST = 1;                     //Clear Interrupt
160         EDIS;
161     }
162
163     void update_variables(void){
164         dXi_est_ant = dXi_est;
165         dXv_est_ant = dXv_est;
166         iL_est_ant = iL_est;
167         vo_est_ant = vo_est;
168         vb1_est_ant = vb1_est;
169         vb2_est_ant = vb2_est;
170         vb_est_ant = vb_est;
171         iL_ant = iL;
172         vo_ant = vo;
173         vb_ant = vb;
174         du_ant = du;
175         uv_ant = uv;
176         u_ant = u;
177     }
178
179     void initialize_references(void){
180         rv = ref_v;
181         ri = n_parallel*ref_i;
182     }
183
184     void verify_end_of_charge(void){
185         if (count_lowcurrent >= counts_prot){
186             if (vb >= 4.2){
187                 count_turn_off_protection=0;
188                 protection_on();
189             }
190         }
191     }
192
193     interrupt void cpu_timer0_isr(void){
194         GpioDataRegs.GPATOGGLE.bit.GPIO11 = 1;

```

```

194
195     protection_verification();
196
197     if ((i >= 5) && start_control == 0){
198         initialize_references();
199         start_control = 1;
200     }
201     i++;
202     if (i > 5) i = 0;
203
204     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;           // Acknowledge this interrupt
205     ↪ to receive more interrupts from group 1
206 }
207
208
209 interrupt void epwm1_timer_isr(void){
210     EPwm1Regs.CMPA.half.CMPA = (Uint16)(u*TBPRD_EPWM1);
211
212     EPwm1Regs.ETCLR.bit.INT = 1;                       // Clear INT flag for
213     ↪ this timer
214     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;           // Acknowledge this
215     ↪ interrupt to receive more interrupts from group 3
216 }
217
218 interrupt void epwm2_timer_isr(void){
219     analog_trigger();                                  //starts ADC conversion
220
221     vo = 0.0026461865127994600*((float)(AdcRegs.ADCRESULT0>>4));
222     iL = 0.0030146882754948800*((float)(AdcRegs.ADCRESULT1>>4));
223     vin = 0.11806753959139265*((float)(AdcRegs.ADCRESULT2>>4));
224     vb1 = 0.0015432670830300000*((float)(AdcRegs.ADCRESULT3>>4));
225     vb4 = 0.0015432670830300000*((float)(AdcRegs.ADCRESULT4>>4));
226     vb2 = 0.0015432670830300000*((float)(AdcRegs.ADCRESULT5>>4));
227     vb3 = 0.0015432670830300000*((float)(AdcRegs.ADCRESULT6>>4));
228     vb5 = 0.0015432670830300000*((float)(AdcRegs.ADCRESULT7>>4));
229
230     if (vb1 > vb4){
231         vb = vb1; vb_l = vb4;
232     } else {
233         vb = vb4; vb_l = vb1;
234     }
235
236     //Calculate the estimate states from Kalman Filter
237     X_est = kalman_filter(dXi_est_ant, dXv_est_ant, iL_est_ant, vb_est_ant,
238     ↪ iL_ant, vb_ant, du_ant, X_est);
239     dXi_est = *(X_est);    dXv_est = *(X_est+1);    iL_est = *(X_est+2);
240     ↪ vb_est = *(X_est+3);
241
242     //Calculate control action uv from voltage LQR controller
243     uv = lqr_controller_v(dXv_est, vb_est, uv_ant, ri, rv);
244     uv = saturation(uv, 0, iL_max);
245
246     //Calculate control action u from current LQR controller
247     u = lqr_controller_i(dXi_est, dXv_est, iL_est, uv, u_ant);
248     u = saturation(u, 0, 0.45);    //Saturation of u
249
250     du = u - u_ant;
251
252     if (k>500) k = 0;
253     u_graph[k] = u;
254     k++;
255
256     if (start_control == 0){
257         protection_on();
258     }
259
260     //Protections
261     protection_counters();

```

```

260 |
261 | //Update delayed variables
262 | update_variables();
263 |
264 | //verify end of charge condition
265 | verify_end_of_charge();
266 |
267 | EPwm2Regs.ETCLR.bit.INT = 1; // Clear INT flag for
    | ↪ this timer
268 | PieCtrlRegs.PIEACK.all = PIEACK_GROUP3; // Acknowledge this
    | ↪ interrupt to receive more interrupts from group 3
269 | }

```

Anexo – C CÓDIGOS PARA SIMULAÇÃO DA BATERIA

C.1 CÓDIGO PRINCIPAL DO SIMULADOR DE BATERIAS

```

1  %Battery Simulation
2  clear all, close all
3  clc
4  format long
5
6  %Definitions
7  Inom = 2750; %mAh
8  Qmaxnom = 2750;
9  Ts = 1;
10 Rso = 0.054545; %from datasheet dV/dI
11 Cc = [1 1];
12 Dc = 0;
13 time = 6; %hours
14 hour = 3600/Ts;
15 n = ceil(time*3600/Ts)
16
17 %initialize field x vs y for PSO estimator
18 [bottom,top] = init_field_bat();
19
20 %capacity rate – power func parameters
21 [ap, cp, cpv] = init_capacity_rate();
22
23 %initialization current profile
24 i = init_current_profile(Inom,n,Ts);
25
26 i_med(1:2) = [0 0];
27 Cr(1:2) = i(1:2)*1000/Inom;
28 SOC(1:2) = 0.0*[1 1]
29
30 Voc = zeros(1,n);
31 Voc(1) = calculate_Voc(cp,SOC(1));
32 Voc(2) = calculate_Voc(cp,SOC(2));
33 Vt = zeros(1,n);
34 Vt(1:2) = Voc(1:2);
35 Vt_med = zeros(1,n);
36 Vt_med(1) = Vt(1)*(1+0.01*rand(1));
37 Vt_med(2) = Vt(2)*(1+0.01*rand(1));
38 vc(1:2) = [0 0];
39 Qmax = zeros(1,n);
40 Q_max = zeros(1,n);
41 SOC_medl = zeros(1,n);
42 SOC_medv = zeros(1,n);
43
44 %noise initialization
45 noise_i = init_noise(n,0,0.0013); %(n,u,sigma)
46 noise_v = init_noise(n,0,0.005); %(n,u,sigma) ... var = 0.01 -> |noise| = 0,4
47 %plot(noise)
48
49 %Battery Parameters Initialization
50 Parameters = zeros(n,3);
51 Parameters(1,:) = [0.1 0.08 200];
52 Parameters(2,:) = Parameters(1,:);
53 zd = exp(-Ts/(Parameters(2,2)*Parameters(2,3)));
54
55 azd = exp(-Ts/12);
56
57 for k = 3:n
58  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59  %Simulating Battery Behavior
60  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61
62  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
63  %Estimate Battery Parameters Via PSO
64  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65  if (k==3)||(mod(k,60) == 0) %! estimation per 1 minute

```

```

66         if (mod(k,1000)==0)
67             k
68         end
69         batterySys = [ i(k-2) i(k-1) vc(k-2) Voc(k-1) Vt(k-1)];
70         Parameters(k,:) = PSOfunc_current(bottom,top,batterySys,Parameters(k-1,:),Ts)
           ↪ ;
71     else
72         Parameters(k,:) = Parameters(k-1,:);
73     end
74     %Filtering PSO Parameters
75     Parameters(k,:) = azd*Parameters(k-1,:) + (1-azd)*Parameters(k,:);
76
77     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78     %Battery Dynamics
79     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
80     zd = exp(-Ts/(Parameters(k,2)*Parameters(k,3)));
81
82     if (i(k) < 0) %charging
83         if Vt(k-1) < 4.2
84             A = zd;
85             B = Parameters(k,2)*(1-zd);
86             C = 1;
87             D = Parameters(k,1);
88             %Calculate SOC(k)
89             [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
90             %Calculate Voc through SOC
91             Voc(k) = calculate_Voc(cp,SOC(k));
92             %Saturation in Voc
93             if Voc(k) > 4.2
94                 Voc(k) = 4.2;
95             end
96             vc(k) = A*vc(k-1) + B*i(k-1);
97             Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
98         else
99             A = zd - (Parameters(k,2)/Parameters(k,1))*(1-zd);
100             B = (Parameters(k,2)/Parameters(k,1))*(1-zd);
101             C = -1/Parameters(k,1);
102             D = 1/Parameters(k,1);
103             Vt(k) = 4.2;
104             vc(k) = A*vc(k-1) + B*(Voc(k-1)-Vt(k-1));
105
106             iy = C*vc(k) + D*(Voc(k-1)-Vt(k-1));
107             if abs(iy) <= abs(i(k))
108                 i(k) = iy;
109             end
110             if i(k) > -0.05
111                 i(k) = 0;
112             end
113             [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
114             Voc(k) = calculate_Voc(cp,SOC(k));
115             if Voc(k) > 4.2
116                 Voc(k) = 4.2;
117             end
118         end
119     else %discharging or i = 0
120         A = zd;
121         B = Parameters(k,2)*(1-zd);
122         C = 1;
123         D = Parameters(k,1);
124         %Calculate SOC(k)
125         [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
126         %Calculate Voc through SOC
127         Voc(k) = calculate_Voc(cp,SOC(k));
128         %Saturation in Voc
129         if Voc(k) > 4.2
130             Voc(k) = 4.2;
131         end
132         vc(k) = A*vc(k-1) + B*i(k-1);
133         Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
134     end
135     if Vt(k) > 4.2

```

```

136         Vt(k) = 4.2;
137     end
138     i_med(k) = i(k) + noise_i(k);
139     Vt_med(k) = Vt(k) + noise_v(k);
140     SOC_med(k) = calculate_SOC(Inom,i_med(k),SOC_medi(k-1),Ts,ap);
141     SOC_medv(k) = calculate_SOCv(cpv,Vt_med(k));
142     Q_max(k) = Q_max(k-1) - (5/6)*(1/Q_maxnom)*abs(i_med(k)*1000)*(Ts/3600);
143
144     %Least Square Algorithm
145
146     %KF Online Estimation
147
148 end
149
150 t = 0:Ts:Ts*n-Ts;
151
152 filename = 'BatterySimulator.xlsx';
153 xlsxwrite(filename,t,'Results','A')
154 xlsxwrite(filename,Parameters,'Results','B')
155 %xlsxwrite(filename,Parameters(:,2),'Results','C')
156 %xlsxwrite(filename,Parameters(:,3),'Results','D')
157 xlsxwrite(filename,Parameters(:,2).*Parameters(:,3),'Results','E')
158 xlsxwrite(filename,vc(1,:),'Results','F')
159 xlsxwrite(filename,Vt,'Results','G')
160 xlsxwrite(filename,Voc,'Results','H')
161 xlsxwrite(filename,i,'Results','I')
162 xlsxwrite(filename,SOC,'Results','J')
163
164 figure(1)
165 subplot(311)
166 hold on
167 plot(t,Parameters(:,1))
168 title('Rs(t)')
169 subplot(312)
170 hold on
171 plot(t,Parameters(:,2),'r')
172 title('R1(t)')
173 subplot(313)
174 hold on
175 plot(t,Parameters(:,3),'g')
176 title('C1(t)')
177 xlabel('tempo(s)')
178
179 figure(2)
180 plot(t,Parameters(:,2).*Parameters(:,3),'b')
181 title('RC(t)')
182 xlabel('tempo(s)')
183
184 figure(3)
185 plot(t,vc(1,:),'b')
186 title('Vrc(t)')
187 xlabel('tempo(s)')
188
189 figure(5)
190 subplot(211)
191 hold on
192 plot(t,Vt,'b','linewidth',2)
193 plot(t,Voc,'r','linewidth',2)
194 title('Vt(t)')
195 subplot(212)
196 hold on
197 plot(t,i,'g','linewidth',2)
198 title('i(t)')
199 xlabel('tempo(s)')
200
201 figure(6)
202 subplot(211)
203 hold on
204 plot(t,Vt,'b','linewidth',2)
205 plot(t,Voc,'r','linewidth',2)
206 title('Voc(t)')

```

```

207 subplot(212)
208 hold on
209 plot(t,SOC,'k','linewidth',2)
210 title('SOC(t)')
211 xlabel('tempo(s)')

```

C.2 FUNÇÕES DE INICIALIZAÇÃO DO ALGORITMO DE SIMULAÇÃO DE BATERIAS

C.2.1 Inicialização do perfil de corrente da bateria

```

1 function [i] = init_current_profile(Inom,n,Ts)
2
3 n_hour = 3600/Ts;
4 n_min = 60/Ts;
5 n_seg = Ts;
6 no = (5*n_hour+40*n_min)
7 n_cycles = floor(n/no)
8
9 %generate cycles
10 for j = 1:n_cycles
11     i(no*(j-1)+1:no*(j-1)+3*n_hour+40*n_min) = -0.3*Inom/1000;
12     i(no*(j-1)+3*n_hour+40*n_min+1:no*(j-1)+4*n_hour) = 0;
13     i(no*(j-1)+4*n_hour+1:no*(j-1)+5*n_hour) = 0.3*Inom/1000;
14     i(no*(j-1)+5*n_hour+1:no*(j-1)+5*n_hour+20*n_min) = 0;
15 end
16 i(no*n_cycles:n) = 0;
17 i(1:2) = -0.3*Inom/1000;

```

C.2.2 Inicialização da curva de capacidade da bateria em função da corrente

```

1 function [ap, cp, cpv] = init_capacity_rate()
2
3 ap = [-160.5 0.3912 2961];
4 cp = [301.9 -1134 1728 -1369 601.4 -144.5 17.89 2.558]; %polynomial coefficients
5 cpv = [4.885 6.906 3.432 3.942 -0.26 -8.5];

```

C.2.3 Inicialização do campo de busca do algoritmo PSO

```

1 function [bottom,top] = init_field_bat()
2
3 Rs_LL = 0.05;
4 Rs_UL = 0.2;
5
6 Rl_LL = 0.05;
7 Rl_UL = 0.2;
8
9 RC1_limits = [0.1 360]; %constant time
10 Cl_LL = RC1_limits(1)/Rl_LL; %Lower Limit for C1
11 Cl_UL = RC1_limits(2)/Rl_UL; %Upper Limit for C1
12 bottom = [Rs_LL Rl_LL Cl_LL];
13 top = [Rs_UL Rl_UL Cl_UL];

```

C.2.4 Inicialização de ruídos para análise de medições de grandezas da bateria

```

1 function noise = init_noise(n,u,sigma)
2
3 noise_ini=rand(1,n);
4 noise_ini=2*noise_ini -1;
5 noise = sqrt(sigma*3)*noise_ini + u;

```

C.3 FUNÇÕES PARA CÁLCULO DO SOC E DA TENSÃO DE CIRCUITO ABERTO

C.3.1 Cálculo do SOC através da corrente

```

1 function [SOC,Qmax] = calculate_SOC(Inom,i,SOCant,Ts,ap)
2
3 Cr = abs(i)*1000/Inom; %Obtain Crate
4 Qmax = ap(1)*(Cr*ap(2))+ap(3); %Obtain Qmax in function of current
5 SOC = SOCant - i*1000*(Ts/3600)/Qmax; %Calculate SOC

```

C.3.2 Cálculo do SOC através da tensão

```

1 function SOC = calculate_SOCv(cpv,V)
2
3 SOC = 1/(1+cpv(1)*exp(-cpv(2)*(V-cpv(3))))+1/(1+cpv(4)*exp(-cpv(5)*(V-cpv(6))));

```

C.3.3 Cálculo da tensão de circuito aberto

```

1 function Voc = calculate_Voc(cp,SOC)
2
3 Voc = cp(1)*SOC^7 + cp(2)*SOC^6 + cp(3)*SOC^5 + cp(4)*SOC^4 + ...
4       cp(5)*SOC^3 + cp(6)*SOC^2 + cp(7)*SOC + cp(8);

```

C.4 FUNÇÕES DE EXECUÇÃO DO ALGORITMO PSO PARA BATERIAS

C.4.1 Algoritmo PSO para encontrar parâmetros no modelo de corrente

```

1 function Parameters = PSOfunc_current(bottom,top,batterySys,Parameters_ant,T)
2
3 %field x vs y
4 goal = 0;
5
6 %swarm
7 particle_number = 30;
8 particle_parameters = 3;
9 particles = zeros(particle_parameters,particle_number);
10 for j = 1:particle_parameters
11     particles(j,:) = bottom(j)*ones(1,particle_number)+(top(j)-bottom(j))*rand(1,
12         ↪ particle_number);
13 end
14 particles_dpos = zeros(particle_parameters,particle_number);
15 best_particle = 1; %initialize best particle
16 particles_best = 0.02*ones(particle_parameters, particle_number); %initialize the
17     ↪ best past of each particle
18 particles_best(:,20) = Parameters_ant;
19
20 m=0;
21 error = 1;
22
23 while (error > 0.00005) && (m < 100)
24     m = m+1;
25     error_acum = 0;
26     for i=1:1:particle_number %para cada uma das 20 candidatas
27         if particles(:,i) ~= particles(:,best_particle) %se a candidata i for
28             ↪ diferente da melhor candidata
29             for j=1:1:particle_parameters
30                 particles_dpos(j,i) = 0.3*rand*(particles(j,best_particle) -
31                     ↪ particles(j,i)) + 0.3*rand*(particles_best(j,i) -
32                     ↪ particles(j,i)) + 0.4*particles_dpos(j,i); %calcula a
33                     ↪ derivada da posicao para todas as particulas
34                 if particles_dpos(j,i)+particles(j,i) > top(j) || particles_dpos(j,i)
35                     ↪ +particles(j,i) < bottom(j)
36                     particles_dpos(j,i) = -particles_dpos(j,i); %inverte o sentido
37                     ↪ caso a particula extrapole o campo de busca
38                 end
39                 particles(j,i) = particles_dpos(j,i) + particles(j,i); %atualiza a
40                     ↪ posicao das particulas
41             end
42         end
43
44         z_best = desired_function_current(particles_best(:,i),Parameters_ant,
45             ↪ batterySys,T);
46         best_z = desired_function_current(particles(:,best_particle),Parameters_ant,
47             ↪ batterySys,T);
48         z = desired_function_current(particles(:,i),Parameters_ant,batterySys,T);
49         if abs(z) < abs(z_best)
50             particles_best(:,i) = particles(:,i);
51             if abs(z) < abs(best_z)
52                 best_particle = i;
53             end
54         end
55         error_acum = abs(z_best) - goal + error_acum;
56     end
57     error = error_acum/i;
58 end
59 %plot(error)
60
61 %movie(movie1,1,2)
62 Parameters = particles(1:particle_parameters,best_particle);
63
64 %z = x^2 + y^2

```

C.5 FUNÇÕES PARA CÁLCULO DO FITNESS DOS ALGORITMOS PSO

C.5.1 Cálculo da função fitness do PSO para o modelo de corrente

```

1  function [ z ] = desired_function_current(particles_parameters,Parameters_ant,
    ↪ batterySys,T)
2
3  %%#codegen
4
5  % Ro_k1 = particles_parameters(1);
6  % R_k1 = particles_parameters(2);
7  % C_k1 = particles_parameters(3);
8
9  % ib_k2 = batterySys(1);
10 % ib_k1 = batterySys(2);
11 % vrc_k2 = batterySys(3)';
12 % voc_k1 = batterySys(4);
13 % vt_k1 = batterySys(5);
14
15 %X(k) = [ vrc(k) ];
16 %u(k) = ib(k);
17
18 zd = exp(-T/(particles_parameters(2)*particles_parameters(3))); %zd_k1
19
20 vrc_k1 = zd*batterySys(3) + particles_parameters(2)*(1-zd)*batterySys(1);
21
22 vt_k1 = batterySys(4) - vrc_k1 - particles_parameters(1)*batterySys(2);
23
24 D_Theta = particles_parameters - Parameters_ant';
25 D_Theta = D_Theta./Parameters_ant';
26
27 z = sqrt(((batterySys(5) - vt_k1))^2) + sqrt(sum(D_Theta.*D_Theta))/3;
28
29 end

```

Anexo – D CÓDIGOS PARA ESTIMAÇÃO DO SOC

D.1 CÓDIGO PRINCIPAL DE SIMULAÇÃO DE ESTIMAÇÃO DO SOC COM EKF

```

1 %Battery Simulation
2 clear all, clc
3 format long
4
5 %Definitions
6 Inom = 2750; %mA
7 Qmaxnom = 2750; %mAh
8 QmaxC = Qmaxnom*3600/1000; %C = Coulombs
9 Ts = 1;
10 Rso = 0.054545; %from datasheet dV/dI
11 time = 6; %hours
12 hour = 3600/Ts;
13 n = ceil(time*3600/Ts)
14
15 %initialize field x vs y for PSO estimator
16 [bottom,top] = init_field_bat();
17
18 %capacity rate – power func parameters
19 [ap, cp, cpv] = init_capacity_rate();
20
21 %initialization current profile
22 i = init_current_profile(Inom,n,Ts);
23
24 i_med(1:2) = [0 0];
25 Cr(1:2) = i(1:2)*1000/Inom;
26 SOC(1:2) = 0.7*[1 1]
27
28 Voc = zeros(1,n);
29 Voc(1) = calculate_Voc(cp,SOC(1));
30 Voc(2) = calculate_Voc(cp,SOC(2));
31 Vt = zeros(1,n);
32 Vt(1:2) = Voc(1:2);
33 Vt_med = zeros(1,n);
34 Vt_med(1) = Vt(1)*(1+0.01*rand(1));
35 Vt_med(2) = Vt(2)*(1+0.01*rand(1));
36 vc(1:2) = [0 0];
37 Qmax = zeros(1,n);
38
39 %noise initialization
40 noise_i = init_noise(n,0,0.0013); %(n,u,sigma)
41 noise_v = init_noise(n,0,0.005); %(n,u,sigma) ... var = 0.01 -> |noise| = 0,4
42 %plot(noise)
43
44 %Battery Parameters Initialization
45 Parameters = zeros(n,3);
46 Parameters(1,:) = [0.1 0.08 200];
47 Parameters(2,:) = Parameters(1,:);
48 zd = exp(-Ts/(Parameters(2,2)*Parameters(2,3)));
49
50 %SOC estimation Model Parameters Initialization
51 Ro_est = 0.09*ones(1,n);
52 R_est = 0.085*ones(1,n);
53 C_est = 300*ones(1,n);
54 zd_est = exp(-Ts/(R_est(1)*C_est(1)))
55
56 %SOC estimation Initialization
57 %X(k|k)
58 SOC_est = zeros(1,n);
59 vrc_est = zeros(1,n);
60 Vt_est = 2.5*ones(1,n);
61 Voc_est = zeros(1,n);
62
63 %X(k|k-1)

```

```

64 SOC_est_ant = zeros(1,n);
65 vrc_est_ant = zeros(1,n);
66 Vt_est_ant = zeros(1,n);
67 Voc_est_ant = zeros(1,n);
68
69 %Extended Kalman Filter Initialization
70 Pkf = zeros(2,2,n);
71 Pkf_ant = zeros(2,2,n);
72 Skf = zeros(1,n);
73 Lkf = zeros(2,n);
74 %Constants and initial values
75 Pkf(:, :, 1) = diag([Ts*10000 1*Ts]);
76 Pkf(:, :, 2) = Pkf(:, :, 1);
77 Pkf(:, :, 3) = Pkf(:, :, 1);
78
79 Qkf = diag([Ts*0.0001 Ts*0.00001]);
80 Rkf = 1/Ts;
81
82 azd = exp(-Ts/12);
83
84 tic
85
86 for k = 3:n
87     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88     %Simulating Battery Behavior
89     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90
91     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
92     %Estimate Battery Parameters Via PSO
93     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94     if (k==3)|(mod(k,60) == 0) %l estimation per 1 minute
95         if (mod(k,1000)==0)
96             k
97         end
98         batterySys = [i(k-2) i(k-1) vc(k-2) Voc(k-1) Vt(k-1)];
99         Parameters(k,:) = PSOfunc_current(bottom,top,batterySys,Parameters(k-1,:),Ts)
100         ↪ ;
101     else
102         Parameters(k,:) = Parameters(k-1,:);
103     end
104     %Filtering PSO Parameters
105     Parameters(k,:) = azd*Parameters(k-1,:) + (1-azd)*Parameters(k,:);
106
107     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
108     %Battery Dynamics
109     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
110     zd = exp(-Ts/(Parameters(k,2)*Parameters(k,3)));
111     if (i(k) < 0) %charging
112         if Vt(k-1) < 4.2
113             A = zd;
114             B = Parameters(k,2)*(1-zd);
115             C = 1;
116             D = Parameters(k,1);
117             %Calculate SOC(k)
118             [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
119             %Calculate Voc through SOC
120             Voc(k) = calculate_Voc(cp,SOC(k));
121             %Saturation in Voc
122             if Voc(k) > 4.2
123                 Voc(k) = 4.2;
124             end
125             vc(k) = A*vc(k-1) + B*i(k-1);
126             Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
127         else
128             A = zd - (Parameters(k,2)/Parameters(k,1))*(1-zd);
129             B = (Parameters(k,2)/Parameters(k,1))*(1-zd);
130             C = -1/Parameters(k,1);
131             D = 1/Parameters(k,1);
132             Vt(k) = 4.2;
133             vc(k) = A*vc(k-1) + B*(Voc(k-1)-Vt(k-1));

```

```

134     iy = C*vc(k) + D*(Voc(k-1)-Vt(k-1));
135     if abs(iy) <= abs(i(k))
136         i(k) = iy;
137     end
138     if i(k) > -0.05
139         i(k) = 0;
140     end
141     [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
142     Voc(k) = calculate_Voc(cp,SOC(k));
143     if Voc(k) > 4.2
144         Voc(k) = 4.2;
145     end
146 end
147 else %discharging or i = 0
148     A = zd;
149     B = Parameters(k,2)*(1-zd);
150     C = 1;
151     D = Parameters(k,1);
152     %Calculate SOC(k)
153     [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
154     %Calculate Voc through SOC
155     Voc(k) = calculate_Voc(cp,SOC(k));
156     %Saturation in Voc
157     if Voc(k) > 4.2
158         Voc(k) = 4.2;
159     end
160     vc(k) = A*vc(k-1) + B*i(k-1);
161     Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
162 end
163 if Vt(k) > 4.2
164     Vt(k) = 4.2;
165 end
166 i_med(k) = i(k) + noise_i(k);
167 Vt_med(k) = Vt(k) + noise_v(k);
168
169 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
170 %SOC Estimation Algorithm
171 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
172
173 %EKF Prediction phase... X_est(k|k-1) = f(X(k-1|k-1),u(k-1))
174 zd_est = exp(-Ts/(R_est(k)*C_est(k)));
175
176 SOC_est_ant(k) = SOC_est(k-1) - i_med(k-1)*(Ts/QmaxC);
177 vrc_est_ant(k) = zd_est*vrc_est(k-1) + R_est(k)*(1-zd_est)*i_med(k-1);
178
179 %Saturation SOC(k|k-1)
180 if SOC_est_ant(k) > 1
181     SOC_est_ant(k) = 1;
182 else if SOC_est_ant(k) < 0
183     SOC_est_ant(k) = 0;
184 end
185 end
186
187 Akf = [1 0; 0 zd_est];
188 Pkf_ant(:, :, k) = Akf*Pkf(:, :, k-1)*Akf' + Qkf; %F(k-1) = Ak
189 Voc_est_ant(k) = calculate_Voc(cp,SOC_est_ant(k));
190 dVocSoc(k) = calculate_dVoc(cp,SOC_est_ant(k));
191 Vt_est_ant(k) = -vrc_est_ant(k) - Ro_est(k)*i_med(k) + Voc_est_ant(k);
192 Ckf(k,:) = [dVocSoc(k) -1];
193
194 %Saturation Vt(k|k-1)
195 if Vt_est_ant(k) > 4.2
196     Vt_est_ant(k) = 4.2;
197 else if Vt_est_ant(k) < 2.5
198     Vt_est_ant(k) = 2.5;
199 end
200 end
201
202 %EKF Propagation phase.. X(k|k) = X(k|k-1) + L(k)*e(k)
203 e_kf(k) = Vt_med(k) - Vt_est_ant(k);
204 Skf(k) = Ckf(k,:) *Pkf_ant(:, :, k)*Ckf(k,:)' + Rkf;

```

```

205     Lkf(:,k) = Pkf_ant(:,k)*Ckf(k,:)/(Skf(k));
206     Pkf(:,k) = (eye(2) - Lkf(:,k)*Ckf(k,:))*Pkf_ant(:,k);
207
208     SOC_est(k) = SOC_est_ant(k) + Lkf(1,k)*e_kf(k);
209     vrc_est(k) = vrc_est_ant(k) + Lkf(2,k)*e_kf(k);
210     Voc_est(k) = calculate_Voc(cp,SOC_est(k));
211     Vt_est(k) = -vrc_est(k) - Ro_est(k)*I_med(k) + Voc_est(k);
212
213     %Saturation SOC(k|k)
214     if SOC_est(k) > 1
215         SOC_est(k) = 1;
216     else if SOC_est(k) < 0
217         SOC_est(k) = 0;
218     end
219 end
220
221 toc
222
223 t = 0:Ts:Ts*n-Ts;
224
225 filename = 'EKF_Results_07_20160209.xlsx';
226
227 xlswrite(filename,t','Results','A')
228 xlswrite(filename,Vt','Results','B')
229 xlswrite(filename,Vt_est','Results','C')
230 xlswrite(filename,Voc','Results','D')
231 xlswrite(filename,Voc_est','Results','E')
232 xlswrite(filename,i_med','Results','F')
233 xlswrite(filename,Parameters(:,1),'Results','G') %Ro
234 xlswrite(filename,Ro_est','Results','H') %Ro_est
235 xlswrite(filename,Parameters(:,2),'Results','I') %R
236 xlswrite(filename,R_est','Results','J') %R_est
237 xlswrite(filename,Parameters(:,3),'Results','K') %C
238 xlswrite(filename,C_est','Results','L') %C_est
239 xlswrite(filename,Parameters(:,2).*Parameters(:,3),'Results','M') %RC
240 xlswrite(filename,(R_est.*C_est)', 'Results','N') %RC_est
241 xlswrite(filename,vc','Results','O') %vrc
242 xlswrite(filename,vrc_est','Results','P') %vrc_est
243 xlswrite(filename,SOC','Results','Q') %SOC
244 xlswrite(filename,SOC_est','Results','R') %SOC_est
245 xlswrite(filename,(SOC-SOC_est)', 'Results','S') %eSOC_est
246
247 figure(1)
248 subplot(211)
249 hold on
250 plot(t,Vt,'b','linewidth',2)
251 plot(t,Voc,'r','linewidth',2)
252 ylim([2 5])
253 title('Vt_{est}(t)')
254 subplot(212)
255 hold on
256 plot(t,i,'g','linewidth',2)
257 title('i(t)')
258 xlabel('tempo(s)')
259
260 figure(2)
261 hold on
262 plot(t,Parameters(:,2).*Parameters(:,3),'k-','linewidth',1.5)
263 plot(t,R_est.*C_est,'b','linewidth',1.5)
264 title('RC(k)')
265 xlabel('tempo(s)')
266
267 figure(3)
268 subplot(311)
269 hold on
270 plot(t,Ro_est,'b','linewidth',2)
271 plot(t,Parameters(:,1),'k--','linewidth',1.5)
272 title('R_o(k)')
273 subplot(312)
274 hold on
275 plot(t,R_est,'r','linewidth',2)

```

```

276 plot(t,Parameters(:,2),'k--','linewidth',1.5)
277 title('R(k)')
278 subplot(313)
279 hold on
280 plot(t,C_est,'g','linewidth',2)
281 plot(t,Parameters(:,3),'k--','linewidth',1.5)
282 title('C(k)')
283
284 figure(4)
285 subplot(211)
286 hold on
287 plot(t,Vt,'b','linewidth',2)
288 plot(t,Vt_med,'k','linewidth',2)
289 plot(t,Vt_est,'r','linewidth',2)
290 title('Vt_{est}(t)')
291 ylim([2 5])
292 subplot(212)
293 hold on
294 plot(t,i,'g','linewidth',2)
295 plot(t,i_med,'k','linewidth',2)
296 title('i(t)')
297 xlabel('tempo(s)')
298
299 figure(5)
300 subplot(211)
301 hold on
302 plot(t,Vt,'b','linewidth',2)
303 plot(t,Vt_med,'k','linewidth',2)
304 plot(t,Vt_est,'r','linewidth',2)
305 title('Vt_{est}(t)')
306 ylim([2 5])
307 subplot(212)
308 hold on
309 plot(t,SOC,'k','linewidth',2)
310 plot(t,SOC_est,'r','linewidth',2)
311 title('SOC_{est}(t)')
312 xlabel('tempo(s)')
313
314 figure(6)
315 hold on
316 plot(t,vc,'b','linewidth',1.5)
317 plot(t,vrc_est,'k--','linewidth',1.5)
318 title('V_{rc}(k)')
319 xlabel('tempo(s)')
320
321 figure(7)
322 hold on
323 plot(t,Voc,'k--','linewidth',2)
324 plot(t,Voc_est,'r','linewidth',2)
325 title('V_{oc}_{est} and V_{oc}')
326 ylim([2 5])

```

D.2 CÓDIGO PRINCIPAL DE SIMULAÇÃO DE ESTIMAÇÃO DO SOC COM KF ADAPTATIVO

```

1 %Battery Simulation
2 clear all,
3 clc
4 format long
5
6 %Definitions
7 Inom = 2750; %mA
8 Qmaxnom = 2750; %mAh
9 QmaxC = Qmaxnom*3600/1000; %C = Coulombs

```

```

10 Ts = 1;
11 time = 6; %hours
12 hour = 3600/Ts;
13 n = ceil(time*3600/Ts)
14
15 %initialize field x vs y
16 [bottom,top] = init_field_bat();
17 [bottom_Theta,top_Theta] = init_field_Theta();
18
19 %capacity rate - power func parameters
20 [ap, cp, cpv] = init_capacity_rate();
21
22 %initialization current profile
23 i = init_current_profile(Inom,n,Ts);
24
25 i_med(1:2) = [0 0];
26 Cr(1:2) = i(1:2)*1000/Inom;
27 SOC(1:2) = 0.7*[1 1]
28
29 Voc = zeros(1,n);
30 Voc(1) = calculate_Voc(cp,SOC(1));
31 Voc(2) = calculate_Voc(cp,SOC(2));
32 Vt = zeros(1,n);
33 Vt(1:2) = Voc(1:2);
34 Vt_med = zeros(1,n);
35 Vt_med(1) = Vt(1)*(1+0.01*rand(1));
36 Vt_med(2) = Vt(2)*(1+0.01*rand(1));
37 vc(1:2) = [0 0];
38 Beta1 = zeros(1,n);
39 Beta0 = zeros(1,n);
40 Qmax = zeros(1,n);
41
42 SOC_est = zeros(1,n); SOC_est(1:3) = 0*[1 1 1];
43 vrc_est = zeros(1,n);
44 Vt_est = 2.0*ones(1,n);
45 Voc_est = zeros(1,n);
46 Theta = zeros(n,3);
47 SOC_est_ant = zeros(1,n);
48 vrc_est_ant = zeros(1,n);
49 Vt_est_ant = zeros(1,n);
50
51
52 %noise initialization
53 noise_i = init_noise(n,0,0.0013); %(n,u,sigma)
54 noise_v = init_noise(n,0,0.005); %(n,u,sigma) ... var = 0.01 -> |noise| = 0,4
55 %plot(noise)
56
57 %Battery Parameters Initialization
58 Parameters = zeros(n,3);
59 Parameters(1,:) = [0.1 0.08 200];
60 Parameters(2,:) = Parameters(1,:);
61 zd = exp(-Ts/(Parameters(2,2)*Parameters(2,3)));
62
63 %SOC estimation Parameters (Theta) Intialization
64 Vt_est(1:2) = Vt_med(1:2);
65 Theta(1,:) = [0.07 0.08 300];
66 Theta(2,:) = Theta(1,:);
67 Ro_est = 0.09*ones(1,n);
68 R_est = 0.085*ones(1,n);
69 C_est = 250*ones(1,n);
70 zd_est = exp(-Ts/(R_est(1)*C_est(1)))
71
72 %SOC estimation Initialization
73
74 %Kalman Filter Initialization
75
76 Pkf = zeros(2,2,n);
77 Pkf_ant = zeros(2,2,n);
78 Skf = zeros(1,n);
79 Lkf = zeros(2,n);
80 %Constants and initial values

```

```

81 Pkf(:, :, 1) = diag([10000*Ts 1*Ts]);
82 Pkf(:, :, 2) = Pkf(:, :, 1);
83 Pkf(:, :, 3) = Pkf(:, :, 1);
84
85 Qkf = diag([Ts*0.0001 Ts*0.00001]);
86 Rkf = 1/Ts;
87
88 azd = exp(-Ts/60);
89
90 tic
91
92 for k = 3:n
93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94 %Simulating Battery Behavior
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96
97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 %Estimate Battery Parameters Via PSO
99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 if (k==3)|| (mod(k,60) == 0) %l estimation per 1 minute
101     if (mod(k,1000)==0)
102         k
103     end
104     batterySys = [i(k-2) i(k-1) vc(k-2) Voc(k-1) Vt(k-1)];
105     Parameters(k, :) = PSOfunc_current(bottom, top, batterySys, Parameters(k-1, :), Ts)
106         ↪ ;
107     Parameters(k, :) = Parameters(k-1, :);
108 end
109 %Filtering PSO Parameters
110 Parameters(k, :) = azd*Parameters(k-1, :) + (1-azd)*Parameters(k, :);
111
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 %Battery Dynamics
114 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
115 zd = exp(-Ts/(Parameters(k,2)*Parameters(k,3)));
116 if (i(k) < 0) %charging
117     if Vt(k-1) < 4.2
118         A = zd;
119         B = Parameters(k,2)*(1-zd);
120         C = 1;
121         D = Parameters(k,1);
122         %Calculate SOC(k)
123         [SOC(k), Qmax(k)] = calculate_SOC(Inom, i(k), SOC(k-1), Ts, ap);
124         %Calculate Voc through SOC
125         Voc(k) = calculate_Voc(cp, SOC(k));
126         %Saturation in Voc
127         if Voc(k) > 4.2
128             Voc(k) = 4.2;
129         end
130         vc(k) = A*vc(k-1) + B*i(k-1);
131         Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
132     else
133         A = zd - (Parameters(k,2)/Parameters(k,1))*(1-zd);
134         B = (Parameters(k,2)/Parameters(k,1))*(1-zd);
135         C = -1/Parameters(k,1);
136         D = 1/Parameters(k,1);
137         Vt(k) = 4.2;
138         vc(k) = A*vc(k-1) + B*(Voc(k-1)-Vt(k-1));
139
140         iy = C*vc(k) + D*(Voc(k-1)-Vt(k-1));
141         if abs(iy) <= abs(i(k))
142             i(k) = iy;
143         end
144         if i(k) > -0.05
145             i(k) = 0;
146         end
147         [SOC(k), Qmax(k)] = calculate_SOC(Inom, i(k), SOC(k-1), Ts, ap);
148         Voc(k) = calculate_Voc(cp, SOC(k));
149         if Voc(k) > 4.2
150             Voc(k) = 4.2;

```

```

151         end
152     end
153 else %discharging or i = 0
154     A = zd;
155     B = Parameters(k,2)*(1-zd);
156     C = 1;
157     D = Parameters(k,1);
158     %Calculate SOC(k)
159     [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
160     %Calculate Voc through SOC
161     Voc(k) = calculate_Voc(cp,SOC(k));
162     %Saturation in Voc
163     if Voc(k) > 4.2
164         Voc(k) = 4.2;
165     end
166     vc(k) = A*vc(k-1) + B*i(k-1);
167     Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
168 end
169 if Vt(k) > 4.2
170     Vt(k) = 4.2;
171 end
172 i_med(k) = i(k) + noise_i(k);
173 Vt_med(k) = Vt(k) + noise_v(k);
174
175 if (SOC(k)-SOC(k-1))==0
176     Beta1(k) = Beta1(k-1);
177 else
178     Beta1(k) = (Voc(k)-Voc(k-1))/(SOC(k)-SOC(k-1));
179 end
180 Beta0(k) = Voc(k) - Beta1(k)*SOC(k);
181
182 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
183 %SOC Estimation Algorithm
184 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
185
186 %PSO SOC Estimation Algorithm
187 if (k==3) || (mod(k,60) == 0) %1 estimation per 1 minute
188     phi = [Vt_med(k) Vt_med(k-1) Vt_med(k-2) i_med(k) i_med(k-1) i_med(k-2) Beta1
189           ↪ (k)];
189     Theta(k,:) = PSO_Theta(bottom_Theta,top_Theta,Theta(k-1,:),phi,QmaxC,Ts,k);
190 else
191     Theta(k,:) = Theta(k-1,:);
192 end
193
194 %Filtering PSO Estimations
195 alpf = exp(-Ts/60);
196 Ro_est(k) = alpf*Ro_est(k) + (1-alpf)*Theta(k,1);
197 R_est(k) = alpf*R_est(k) + (1-alpf)*Theta(k,2);
198 C_est(k) = alpf*C_est(k) + (1-alpf)*Theta(k,3);
199
200 zd_est = exp(-Ts/(R_est(k)*C_est(k)));
201
202 %%%Prediction Phase
203 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
204 SOC_est_ant(k) = SOC_est(k-1) - (Ts/QmaxC)*i_med(k-1);
205 vrc_est_ant(k) = zd_est*vrc_est(k-1) + R_est(k)*(1-zd_est)*i_med(k-1);
206
207 %Saturation SOC(k|k-1)
208 if SOC_est_ant(k) > 1
209     SOC_est_ant(k) = 1;
210 else if SOC_est_ant(k) < 0
211     SOC_est_ant(k) = 0;
212 end
213 end
214
215 %Matrices A, C and P
216 Akf = [1 0; 0 zd_est];
217 Pkf_ant(:, :, k) = Akf*Pkf(:, :, k-1)*Akf' + Qkf;
218 Ckf = [Beta1(k) -1];
219
220 %%%Update Phase

```

```

221 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
222 Vt_est_ant(k) = Beta1(k)*SOC_est_ant(k) + Beta0(k) -vrc_est_ant(k) - Ro_est(k)*
    ↪ i_med(k);
223
224 %Saturation Vt(k|k-1)
225 if Vt_est_ant(k) > 4.2
226     Vt_est_ant(k) = 4.2;
227 else if Vt_est_ant(k) < 2.5
228     Vt_est_ant(k) = 2.5;
229 end
230 end
231
232 Skf(k) = Ckf*Pkf_ant(:, :, k)*Ckf' + Rkf;
233 Lkf(:, k) = (Pkf_ant(:, :, k)*Ckf')/Skf(k);
234 Pkf(:, :, k) = (eye(2) - Lkf(:, k)*Ckf)*Pkf_ant(:, :, k);
235
236 %Final State Estimation X(k|k)
237 SOC_est(k) = SOC_est_ant(k) + Lkf(1, k)*(Vt_med(k) - Vt_est_ant(k));
238 vrc_est(k) = vrc_est_ant(k) + Lkf(2, k)*(Vt_med(k) - Vt_est_ant(k));
239
240 %Saturation SOC(k|k)
241 if SOC_est(k) > 1
242     SOC_est(k) = 1;
243 else if SOC_est(k) < 0
244     SOC_est(k) = 0;
245 end
246 end
247
248 %Output results
249 Voc_est(k) = Beta1(k)*SOC_est(k) + Beta0(k);
250 Vt_est(k) = Voc_est(k) - vrc_est(k) - Ro_est(k)*i_med(k);
251
252 %Saturation Vt(k|k)
253 if Vt_est(k) > 4.2
254     Vt_est(k) = 4.2;
255     Voc_est(k) = Vt_est(k) + vrc_est(k) + Ro_est(k)*i_med(k);
256 else if Vt_est(k) < 2;
257     Vt_est(k) = 2;
258     Voc_est(k) = Vt_est(k) + vrc_est(k) + Ro_est(k)*i_med(k);
259 end
260 end
261 end
262
263 Beta1_est = Beta1;
264 Beta0_est = Beta0;
265
266 toc
267
268 t = 0:Ts:Ts*n-Ts;
269
270 % filename = 'PSO_KF_2nd_Results_07_20160209.xlsx';
271 % xlswrite(filename, t, 'Results', 'A')
272 % xlswrite(filename, Vt, 'Results', 'B')
273 % xlswrite(filename, Vt_est, 'Results', 'C')
274 % xlswrite(filename, Voc, 'Results', 'D')
275 % xlswrite(filename, Voc_est, 'Results', 'E')
276 % xlswrite(filename, i_med, 'Results', 'F')
277 % xlswrite(filename, Parameters(:, 1), 'Results', 'G') %Ro
278 % xlswrite(filename, Ro_est, 'Results', 'H') %Ro_est
279 % xlswrite(filename, Parameters(:, 2), 'Results', 'I') %R
280 % xlswrite(filename, R_est, 'Results', 'J') %R_est
281 % xlswrite(filename, Parameters(:, 3), 'Results', 'K') %C
282 % xlswrite(filename, C_est, 'Results', 'L') %C_est
283 % xlswrite(filename, Parameters(:, 2).*Parameters(:, 3), 'Results', 'M') %RC
284 % xlswrite(filename, (R_est.*C_est), 'Results', 'N') %RC_est
285 % xlswrite(filename, vc, 'Results', 'O') %vrc
286 % xlswrite(filename, vrc_est, 'Results', 'P') %vrc_est
287 % xlswrite(filename, SOC, 'Results', 'Q') %SOC
288 % xlswrite(filename, SOC_est, 'Results', 'R') %SOC_est
289 % xlswrite(filename, (SOC-SOC_est), 'Results', 'S') %SOC_est
290 % xlswrite(filename, Beta1, 'Results', 'T') %C_est

```

```

291 % xlswrite(filename,Beta0','Results','U') %C_est
292
293 close all
294
295 figure(1)
296 subplot(211)
297 hold on
298 plot(t,Vt,'b','linewidth',2)
299 plot(t,Voc,'r','linewidth',2)
300 ylim([2 5])
301 title('Vt_{est}(t)')
302 subplot(212)
303 hold on
304 plot(t,i,'g','linewidth',2)
305 title('i(t)')
306 xlabel('tempo(s)')
307
308 figure(2)
309 subplot(211)
310 hold on
311 plot(t,Vt,'b','linewidth',2)
312 plot(t,Vt_med,'k','linewidth',2)
313 plot(t,Vt_est,'r','linewidth',2)
314 title('Vt_{est}(t)')
315 ylim([2 5])
316 subplot(212)
317 hold on
318 plot(t,i,'g','linewidth',2)
319 plot(t,i_med,'k','linewidth',2)
320 title('i(t)')
321 xlabel('tempo(s)')
322
323 figure(3)
324 subplot(211)
325 hold on
326 plot(t,Vt,'b','linewidth',2)
327 plot(t,Vt_med,'k','linewidth',2)
328 plot(t,Vt_est,'r','linewidth',2)
329 title('Vt_{est}(t)')
330 ylim([2 5])
331 subplot(212)
332 hold on
333 plot(t,SOC,'k','linewidth',2)
334 plot(t,SOC_est,'r','linewidth',2)
335 title('SOC_{est}(t)')
336 xlabel('tempo(s)')
337
338 figure(4)
339 hold on
340 plot(t,vc,'b','linewidth',1.5)
341 plot(t,vrc_est,'k-','linewidth',1.5)
342 title('V_{rc}(k)')
343 xlabel('tempo(s)')
344
345 figure(5)
346 hold on
347 plot(t,Parameters(:,2).*Parameters(:,3),'k-','linewidth',1.5)
348 plot(t,R_est.*C_est,'b','linewidth',1.5)
349 title('RC(k)')
350 xlabel('tempo(s)')
351
352 figure(6)
353 subplot(311)
354 hold on
355 plot(t,Ro_est,'b','linewidth',2)
356 plot(t,Parameters(:,1),'k--','linewidth',1.5)
357 title('R_o(k)')
358 subplot(312)
359 hold on
360 plot(t,R_est,'r','linewidth',2)
361 plot(t,Parameters(:,2),'k--','linewidth',1.5)

```

```

362 title('R(k)')
363 subplot(313)
364 hold on
365 plot(t,C_est,'g','linewidth',2)
366 plot(t,Parameters(:,3),'k--','linewidth',1.5)
367 title('C(k)')
368
369 figure(7)
370 subplot(211)
371 hold on
372 plot(t,Beta0_est,'c','linewidth',2)
373 plot(t,Beta0,'k--','linewidth',2)
374 title('Beta_0(k)')
375 subplot(212)
376 hold on
377 plot(t,Beta1_est,'m','linewidth',2)
378 plot(t,Beta1,'k--','linewidth',2)
379 title('Beta_1(k)')
380
381 figure(8)
382 hold on
383 plot(t,Voc,'k--','linewidth',2)
384 plot(t,Voc_est,'r','linewidth',2)
385 title('V_{oc}_{est} and V_{oc}')
386 ylim([2 5])
387

```

D.2.1 Função de inicialização do campo de busca dos parâmetros do estimador do SOC

```

1 function [bottom_Theta,top_Theta] = init_field_Theta()
2
3 %Ro, R, C, Beta0, Beta1
4 Romin = 0.01;
5 Romax = 0.2;
6
7 Rmin = 0.01;
8 Rmax = 0.2;
9
10 RCmin = .1;
11 RCmax = 360;
12
13 Cmin = RCmin/Rmin;
14 Cmax = RCmax/Rmax;
15
16 bottom_Theta = [Romin Rmin Cmin];
17 top_Theta = [Romax Rmax Cmax];

```

D.2.2 Função do estimador PSO de parâmetros do modelo

```

1 function Parameters = PSO_Theta(bottom,top,Theta_ant,phi,Qmax,Ts,k)
2
3 %field x vs y
4 goal = 0;
5
6 %swarm
7 particle_number = 30;
8 particle_parameters = 3;
9 particles = zeros(particle_parameters,particle_number);
10

```

```

11 for j = 1:particle_parameters
12     particles(j,:) = bottom(j)*ones(1,particle_number)+(top(j)-bottom(j))*rand(1,
        ↳ particle_number);
13 end
14 particles_dpos = zeros(particle_parameters,particle_number);
15 best_particle = 1; %initialize best particle
16 particles_best = [0.07 0.08 300]*ones(1, particle_number); %initialize the best past
        ↳ of each particle
17 particles_best(:,20) = Theta_ant;
18
19 m=0;
20 error = 1;
21
22 while (error > 0.0000001) && (m < 100)
23     m = m+1;
24     error_acum = 0;
25     for i = 1:particle_number %para cada uma das 20 candidatas
26
27         if particles(:,i) ~= particles(:,best_particle) %se a candidata i for
            ↳ diferente da melhor candidata
28             for j = 1:particle_parameters
29                 particles_dpos(j,i) = 0.3*rand*(particles(j,best_particle) -
                    ↳ particles(j,i)) + 0.3*rand*(particles_best(j,i) -
                    ↳ particles(j,i)) + 0.4*particles_dpos(j,i); %calcula a
                    ↳ derivada da posicao para todas as particulas
30                 if particles_dpos(j,i)+particles(j,i) > top(j) || particles_dpos(j,i)
                    ↳ +particles(j,i) < bottom(j)
31                     particles_dpos(j,i) = -particles_dpos(j,i); %inverte o sentido
                    ↳ caso a partícula extrapole o campo de busca
32                 end
33                 particles(j,i) = particles_dpos(j,i) + particles(j,i); %atualiza a
                    ↳ posicao das particulas
34             end
35         end
36
37         z_best = desired_function_Theta(particles_best(:,i),Theta_ant,phi,Qmax,Ts,k);
38         best_z = desired_function_Theta(particles(:,best_particle),Theta_ant,phi,Qmax
            ↳ ,Ts,k);
39         z = desired_function_Theta(particles(:,i),Theta_ant,phi,Qmax,Ts,k);
40         if abs(z) < abs(z_best)
41             particles_best(:,i) = particles(:,i);
42             if abs(z) < abs(best_z)
43                 best_particle = i;
44             end
45         end
46         error_acum = abs(z_best) - goal + error_acum;
47
48     end
49     error = error_acum/i;
50 end
51
52 %movie(movie1,1,2)
53 Parameters = particles(1:particle_parameters,best_particle);
54
55 %a = x^2 + y^2

```

D.2.3 Função de cálculo da função de custo da estimação de parâmetros com PSO

```

1 function [ z ] = desired_function_Theta(particles_parameters,Theta_ant,phi,Qmax,Ts,k)
2
3 %passing regressors to variables
4 Vt_med = phi(1);
5 Vt_med1 = phi(2);
6 Vt_med2 = phi(3);

```

```

7 | i_med = phi(4);
8 | i_med1 = phi(5);
9 | i_med2 = phi(6);
10 | Beta1 = phi(7);
11 |
12 | %passing parameters (Theta) to variavles
13 | Ro = particles_parameters(1);
14 | R = particles_parameters(2);
15 | C = particles_parameters(3);
16 |
17 | zd = exp(-Ts/(R*C));
18 |
19 | a1 = -1 -zd;
20 | a2 = zd;
21 | bo = -Ro;
22 | b1 = -Beta1*Ts/Qmax + Ro*(1+zd) - R*(1-zd);
23 | b2 = zd*Beta1*Ts/Qmax - Ro*zd + R*(1-zd);
24 |
25 | Vt_est = -a1*Vt_med1 - a2*Vt_med2 + bo*i_med + b1*i_med1 + b2*i_med2;
26 |
27 | DTheta = (particles_parameters' - Theta_ant)./Theta_ant;
28 |
29 | Jv = sqrt((Vt_est - Vt_med)^2);
30 |
31 | JTheta = sqrt(DTheta*DTheta')/3;
32 |
33 | z = Jv + 1*JTheta;
34 |
35 | end

```

D.3 CÓDIGO PRINCIPAL DE SIMULAÇÃO DE ESTIMAÇÃO DO SOC COM KF ADAPTATIVO E COSTIMAÇÃO DA CURVA VOC

```

1 | %Battery Simulation
2 | clear all,
3 | clc
4 | format long
5 |
6 | %Definitions
7 | Inom = 2750; %mA
8 | Qmaxnom = 2750; %mAh
9 | QmaxC = Qmaxnom*3600/1000; %C = Coulombs
10 | Ts = 10;
11 | time = 6; %hours
12 | hour = 3600/Ts;
13 | n = ceil(time*3600/Ts)
14 |
15 | %initialize field x vs y
16 | [bottom,top] = init_field_bat();
17 | [bottom_Theta,top_Theta] = init_field_Theta();
18 |
19 | %capacity rate - power func parameters
20 | [ap, cp, cpv] = init_capacity_rate();
21 |
22 | %initialization current profile
23 | i = init_current_profile(Inom,n,Ts);
24 |
25 | Cr(1:2) = i(1:2)*1000/Inom;
26 | SOC = zeros(1,n);
27 | SOC(1:2) = 0.0*[1 1];
28 |
29 | Voc = zeros(1,n);

```

[illegible]

```

101     if (k==3) || (mod(k,60) == 0) %! estimation per 1 minute
102         if (mod(k,1000)==0)
103             k
104         end
105         batterySys = [ i(k-2) i(k-1) vc(k-2) Voc(k-1) Vt(k-1)];
106         Parameters(k,:) = PSOfunc_current(bottom,top,batterySys,Parameters(k-1,:),Ts)
107     else
108         Parameters(k,:) = Parameters(k-1,:);
109     end
110     %Filtering PSO Parameters
111     Parameters(k,:) = azd*Parameters(k-1,:) + (1-azd)*Parameters(k,:);
112
113     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
114     %Battery Dynamics
115     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
116     zd = exp(-Ts/(Parameters(k,2)*Parameters(k,3)));
117     if (i(k) < 0) %charging
118         if Vt(k-1) < 4.2
119             A = zd;
120             B = Parameters(k,2)*(1-zd);
121             C = 1;
122             D = Parameters(k,1);
123             %Calculate SOC(k)
124             [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
125             %Calculate Voc through SOC
126             Voc(k) = calculate_Voc(cp,SOC(k));
127             %Saturation in Voc
128             if Voc(k) > 4.2
129                 Voc(k) = 4.2;
130             end
131             vc(k) = A*vc(k-1) + B*i(k-1);
132             Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
133         else
134             A = zd - (Parameters(k,2)/Parameters(k,1))*(1-zd);
135             B = (Parameters(k,2)/Parameters(k,1))*(1-zd);
136             C = -1/Parameters(k,1);
137             D = 1/Parameters(k,1);
138             Vt(k) = 4.2;
139             vc(k) = A*vc(k-1) + B*(Voc(k-1)-Vt(k-1));
140
141             iy = C*vc(k) + D*(Voc(k-1)-Vt(k-1));
142             if abs(iy) <= abs(i(k))
143                 i(k) = iy;
144             end
145             if i(k) > -0.05
146                 i(k) = 0;
147             end
148             [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
149             Voc(k) = calculate_Voc(cp,SOC(k));
150             if Voc(k) > 4.2
151                 Voc(k) = 4.2;
152             end
153         end
154     else %discharging or i = 0
155         A = zd;
156         B = Parameters(k,2)*(1-zd);
157         C = 1;
158         D = Parameters(k,1);
159         %Calculate SOC(k)
160         [SOC(k),Qmax(k)] = calculate_SOC(Inom,i(k),SOC(k-1),Ts,ap);
161         %Calculate Voc through SOC
162         Voc(k) = calculate_Voc(cp,SOC(k));
163         %Saturation in Voc
164         if Voc(k) > 4.2
165             Voc(k) = 4.2;
166         end
167         vc(k) = A*vc(k-1) + B*i(k-1);
168         Vt(k) = Voc(k) - (D*i(k) + C*vc(k));
169     end
170     if Vt(k) > 4.2

```

```

171         Vt(k) = 4.2;
172     end
173     i_med(k) = i(k) + noise_i(k);
174     Vt_med(k) = Vt(k) + noise_v(k);
175
176     if (SOC(k)-SOC(k-1)==0)
177         Beta1(k) = Beta1(k-1);
178     else
179         Beta1(k) = (Voc(k)-Voc(k-1))/(SOC(k)-SOC(k-1));
180     end
181     Beta0(k) = Voc(k) - Beta1(k)*SOC(k);
182
183     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
184     %SOC Estimation Algorithm
185     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
186
187     %PSO SOC Estimation Algorithm
188     %alp = exp(-Ts/150);
189     %Vt_lpf(k) = alp*f*Vt_lpf(k-1) + (1-alp)*Vt_med(k);
190     if (mod(k,1) == 0)
191         phi = [Vt_med(k) Vt_med(k-1) Vt_med(k-2) i_med(k) i_med(k-1) i_med(k-2)];
192         Theta(k,:) = PSO_Theta(bottom_Theta,top_Theta,Theta(k-1,:),phi,QmaxC,Ts,k);
193     else
194         Theta(k,:) = Theta(k-1,:);
195     end
196
197     alp = exp(-Ts/150);
198     Ro_est(k) = alp*f*Ro_est(k-1) + (1-alp)*Theta(k,1);
199     R_est(k) = alp*f*R_est(k-1) + (1-alp)*Theta(k,2);
200     C_est(k) = alp*f*C_est(k-1) + (1-alp)*Theta(k,3);
201     Beta1_est(k) = Theta(k,4);
202     zd_est = exp(-Ts/(R_est(k)*C_est(k)));
203
204     lambdaSOC = 1;
205     lambdaB0 = exp(-1/100);
206     %lambdaB0 = exp(-1/10000);
207
208     dB1 = Beta1_est(k) - Beta1_est(k-1);
209     if dB1 == 0
210         dB1 = 1e-3;
211     end
212
213     SOC_est_ant(k) = lambdaSOC*SOC_est(k-1) - (Ts/QmaxC)*i_med(k-1);
214     vrc_est_ant(k) = zd_est*vrc_est(k-1) + R_est(k)*(1-zd_est)*i_med(k-1);
215     Beta0_est_ant(k) = lambdaB0*Beta0_est(k-1) - dB1*SOC_est(k-1); % + (dB1*Ts/QmaxC)*
216     ↪ i_med(k-1);
217
218     if SOC_est_ant(k) > 1
219         SOC_est_ant(k) = 1;
220     else if SOC_est_ant(k) < 0
221         SOC_est_ant(k) = 0;
222     end
223
224     if Beta0_est_ant(k) > 4.2
225         Beta0_est_ant(k) = 4.2;
226     else if Beta0_est_ant(k) < -2.5
227         Beta0_est_ant(k) = -2.5;
228     end
229
230     if k==3
231         Beta0_est_ant(k) = 2.5;
232     end
233
234     Akf = [lambdaSOC 0 0; 0 zd_est 0; -dB1 0 lambdaB0];
235     Pkf_ant(:, :, k) = Akf*Pkf(:, :, k-1)*Akf' + Qkf;
236     Ckf = [Beta1_est(k) -1 1];
237
238     if (rank(observ(Akf,Ckf))<3)
239         fprintf('non-observable at k=%d, i(k-1)=%.4f, Beta(k-1)=%.7f, SOC(k-1)=%.7f\n
240         ↪ ',k,i_med(k-1),Beta1_est(k-1),SOC_est(k-1));

```

```

240     Akf;
241     Ckf;
242     Mo = obsv(Akf,Ckf);
243 end
244
245 %Update Phase
246 Vt_est_ant(k) = Beta1_est(k)*SOC_est_ant(k) + Beta0_est_ant(k) -vrc_est_ant(k) -
    ↪ Ro_est(k)*i_med(k);
247
248 if Vt_est_ant(k) > 4.2
249     Vt_est_ant(k) = 4.2;
250 else if Vt_est_ant(k) < 2.5
251     Vt_est_ant(k) = 2.5;
252 end
253 end
254
255 Skf(k) = Ckf*Pkf_ant(:, :, k)*Ckf' + Rkf;
256 if (Vt_med(k) >= 4.19) && (i_med(k)==0 || i_med(k-1)==0 || i_med(k-2)==0)
257     Lkf(:, k) = zeros(3,1);
258     Pkf(:, :, k) = zeros(3,3);
259 else
260     Lkf(:, k) = (Pkf_ant(:, :, k)*Ckf')/Skf(k);
261     Pkf(:, :, k) = (eye(3) - Lkf(:, k)*Ckf)*Pkf_ant(:, :, k);
262 end
263
264 SOC_est(k) = SOC_est_ant(k) + Lkf(1,k)*(Vt_med(k) - Vt_est_ant(k));
265 vrc_est(k) = vrc_est_ant(k) + Lkf(2,k)*(Vt_med(k) - Vt_est_ant(k));
266 Beta0_est(k) = Beta0_est_ant(k) + Lkf(3,k)*(Vt_med(k) - Vt_est_ant(k));
267
268 if SOC_est(k) > 1
269     SOC_est(k) = 1;
270 else if SOC_est(k) < 0
271     SOC_est(k) = 0;
272 end
273 end
274 if Beta0_est(k) > 4.2
275     Beta0_est(k) = 4.2;
276 else if Beta0_est(k) < -2.5
277     Beta0_est(k) = -2.5;
278 end
279 end
280
281 if (Vt_med(k) >= 4.19) && (i_med(k)==0 || i_med(k-1)==0 || i_med(k-2)==0)
282     Voc_est(k) = Vt_med(k) - vrc_est(k) - Ro_est(k)*i_med(k);
283     Beta0_est(k) = Voc_est(k) - Beta1_est(k)*SOC_est(k);
284     Vt_est(k) = Vt_med(k);
285 else
286     Voc_est(k) = Beta1_est(k)*SOC_est(k) + Beta0_est(k);
287     Vt_est(k) = Voc_est(k) - vrc_est(k) - Ro_est(k)*i_med(k);
288 end
289 if Vt_est(k) > 4.2
290     Vt_est(k) = 4.2;
291     Voc_est(k) = Vt_est(k) + vrc_est(k) + Ro_est(k)*i_med(k);
292 else if Vt_est(k) < 2;
293     Vt_est(k) = 2;
294     Voc_est(k) = Vt_est(k) + vrc_est(k) + Ro_est(k)*i_med(k);
295 end
296 end
297 end
298
299 toc
300
301 t = 0:Ts:Ts*n-Ts;
302
303 close all
304
305 filename = 'PS0_KF_3rd_Results_00_20160210.xlsx';
306 xlswrite(filename,t', 'Results', 'A')
307 xlswrite(filename,Vt', 'Results', 'B')
308 xlswrite(filename,Vt_est', 'Results', 'C')
309 xlswrite(filename,Voc', 'Results', 'D')

```

```

310 xlswrite(filename,Voc_est','Results','E')
311 xlswrite(filename,i_med','Results','F')
312 xlswrite(filename,Parameters(:,1),'Results','G') %Ro
313 xlswrite(filename,Ro_est','Results','H') %Ro_est
314 xlswrite(filename,Parameters(:,2),'Results','I') %R
315 xlswrite(filename,R_est','Results','J') %R_est
316 xlswrite(filename,Parameters(:,3),'Results','K') %C
317 xlswrite(filename,C_est','Results','L') %C_est
318 xlswrite(filename,Parameters(:,2).*Parameters(:,3),'Results','M') %RC
319 xlswrite(filename,(R_est.*C_est),'Results','N') %RC_est
320 xlswrite(filename,vc','Results','O') %vrc
321 xlswrite(filename,vrc_est','Results','P') %vrc_est
322 xlswrite(filename,SOC','Results','Q') %SOC
323 xlswrite(filename,SOC_est','Results','R') %SOC_est
324 xlswrite(filename,(SOC-SOC_est),'Results','S') %SOC_est
325 xlswrite(filename,Betal','Results','T') %B1
326 xlswrite(filename,Betal_est','Results','U') %B1_est
327 xlswrite(filename,Beta0','Results','V') %B0
328 xlswrite(filename,Beta0_est','Results','W') %B0_est
329
330 figure(1)
331 subplot(211)
332 hold on
333 plot(t,Vt,'b','linewidth',2)
334 plot(t,Voc,'r','linewidth',2)
335 ylim([2 5])
336 title('Vt_{est}(t)')
337 subplot(212)
338 hold on
339 plot(t,i,'g','linewidth',2)
340 title('i(t)')
341 xlabel('tempo(s)')
342
343 figure(2)
344 subplot(211)
345 hold on
346 plot(t,Vt_lpf,'b','linewidth',2)
347 plot(t,Vt_med,'k','linewidth',2)
348 plot(t,Vt_est,'r','linewidth',2)
349 title('Vt_{est}(t)')
350 ylim([2 5])
351 subplot(212)
352 hold on
353 plot(t,i,'g','linewidth',2)
354 plot(t,i_med,'k','linewidth',2)
355 title('i(t)')
356 xlabel('tempo(s)')
357
358 figure(3)
359 subplot(211)
360 hold on
361 plot(t,Vt,'b','linewidth',2)
362 plot(t,Vt_med,'k','linewidth',2)
363 plot(t,Vt_est,'r','linewidth',2)
364 title('Vt_{est}(t)')
365 ylim([2 5])
366 subplot(212)
367 hold on
368 plot(t,SOC,'k','linewidth',2)
369 plot(t,SOC_est,'r','linewidth',2)
370 title('SOC_{est}(t)')
371 xlabel('tempo(s)')
372
373 figure(4)
374 hold on
375 plot(t,vc,'b','linewidth',1.5)
376 plot(t,vrc_est,'k','linewidth',1.5)
377 title('V_{rc}(k)')
378 xlabel('tempo(s)')
379
380 figure(5)

```

```

381 hold on
382 plot(t, Parameters(:,2).*Parameters(:,3), 'k-', 'linewidth', 1.5)
383 plot(t, R_est.*C_est, 'b', 'linewidth', 1.5)
384 title('RC(k)')
385 xlabel('tempo(s)')
386
387 figure(6)
388 subplot(311)
389 hold on
390 plot(t, Ro_est, 'b', 'linewidth', 2)
391 plot(t, Parameters(:,1), 'k--', 'linewidth', 1.5)
392 title('R_o(k)')
393 subplot(312)
394 hold on
395 plot(t, R_est, 'r', 'linewidth', 2)
396 plot(t, Parameters(:,2), 'k--', 'linewidth', 1.5)
397 title('R(k)')
398 subplot(313)
399 hold on
400 plot(t, C_est, 'g', 'linewidth', 2)
401 plot(t, Parameters(:,3), 'k--', 'linewidth', 1.5)
402 title('C(k)')
403
404 figure(7)
405 subplot(211)
406 hold on
407 plot(t, Beta0_est, 'c', 'linewidth', 2)
408 plot(t, Beta0, 'k--', 'linewidth', 2)
409 title('Beta_0(k)')
410 subplot(212)
411 hold on
412 plot(t, Beta1_est, 'm', 'linewidth', 2)
413 plot(t, Beta1, 'k--', 'linewidth', 2)
414 title('Beta_1(k)')
415
416 figure(8)
417 hold on
418 plot(t, Voc, 'k--', 'linewidth', 2)
419 plot(t, Voc_est, 'r', 'linewidth', 2)
420 title('V_{oc}_{est} and V_{oc}')
421
422 ylim([2 5])

```

D.3.1 Função de inicialização do campo de busca dos parâmetros do estimador do SOC com coestimação da curva voc

```

1 function [bottom_Theta, top_Theta] = init_field_Theta()
2
3 %Ro, R, C, Beta0, Beta1
4 Romin = 0.01;
5 Romax = 0.15;
6
7 Rmin = 0.01;
8 Rmax = 0.15;
9
10 RCmin = 1;
11 RCmax = 60;
12
13 Cmin = RCmin/Rmin;
14 Cmax = RCmax/Rmax;
15
16 B1min = 1e-5;
17 B1max = 20;

```

```

18 bottom_Theta = [Romin Rmin Cmin Blmin];
19 top_Theta = [Romax Rmax Cmax Blmax];
20

```

D.3.2 Função do estimador PSO de parâmetros do modelo com coestimação da curva voc

```

1 function Parameters = PSO_Theta(bottom,top,Theta_ant,phi,Qmax,Ts,k)
2
3 %field x vs y
4 goal = 0;
5
6 %swarm
7 particle_number = 30;
8 particle_parameters = 4;
9 particles = zeros(particle_parameters,particle_number);
10
11 Vt_med = phi(1);
12 Vt_med1 = phi(2);
13 Vt_med2 = phi(3);
14 i_med = phi(4);
15 i_med1 = phi(5);
16 i_med2 = phi(6);
17
18 if (Vt_med >= 4.19) && (i_med==0 || i_med1==0 || i_med2==0)
19     bottom(4) = Theta_ant(4);
20     top(4) = Theta_ant(4);
21 end
22
23 for j = 1:particle_parameters
24     particles(j,:) = bottom(j)*ones(1,particle_number)+(top(j)-bottom(j))*rand(1,
        ↳ particle_number);
25 end
26 particles_dpos = zeros(particle_parameters,particle_number);
27 best_particle = 1; %initialize best particle
28 for j = 1:particle_parameters
29     particles_best(j,:) = bottom(j)*ones(1,particle_number)+(top(j)-bottom(j))*rand
        ↳ (1,particle_number); %initialize the best past of each particle
30 end
31 particles_best(:,particle_number) = Theta_ant;
32
33 m=0;
34 error = 1;
35
36 while (error > 0.0000001) && (m < 50)
37     m = m+1;
38     error_acum = 0;
39     for i = 1:particle_number %para cada uma das 20 candidatas
40
41         if particles(:,i) ~= particles(:,best_particle) %se a candidata i for
            ↳ diferente da melhor candidata
42             for j = 1:particle_parameters
43                 particles_dpos(j,i) = 0.2*rand*(particles(j,best_particle) -
                    ↳ particles(j,i)) + 0.4*rand*(particles_best(j,i) -
                    ↳ particles(j,i)) + 0.4*particles_dpos(j,i); %calcula a
                    ↳ derivada da posicao para todas as particulas
44                 if particles_dpos(j,i)+particles(j,i) > top(j) || particles_dpos(j,i)
                    ↳ +particles(j,i) < bottom(j)
45                     particles_dpos(j,i) = -particles_dpos(j,i); %inverte o sentido
                    ↳ caso a partícula extrapole o campo de busca
46                 end
47                 particles(j,i) = particles_dpos(j,i) + particles(j,i); %atualiza a
                    ↳ posicao das particulas
48             end
49         end

```

```

50     z_best = desired_function_Theta(particles_best(:,i),Theta_ant,phi,Qmax,Ts,k);
51     best_z = desired_function_Theta(particles(:,best_particle),Theta_ant,phi,Qmax
52     ↪ ,Ts,k);
53     z = desired_function_Theta(particles(:,i),Theta_ant,phi,Qmax,Ts,k);
54     if abs(z) < abs(z_best)
55         particles_best(:,i) = particles(:,i);
56     end
57     if abs(z) < abs(best_z)
58         best_particle = i;
59     end
60     error_acum = abs(z_best) - goal + error_acum;
61
62     end
63     error = error_acum/i;
64 end
65 %plot(error)
66
67 %anovie(movie1,1,2)
68 Parameters = particles(1:particle_parameters,best_particle);
69
70 %z = x^2 + y^2

```

D.3.3 Função de cálculo da função de custo da estimação de parâmetros com PSO com coestimação da curva voc

```

1  function [ z ] = desired_function_Theta(particles_parameters,Theta_ant,phi,Qmax,Ts,k)
2
3  %passing regressors to variables
4  Vt_med = phi(1);
5  Vt_med1 = phi(2);
6  Vt_med2 = phi(3);
7  i_med = phi(4);
8  i_med1 = phi(5);
9  i_med2 = phi(6);
10
11 %passing parameters (Theta) to variavles
12 Ro = particles_parameters(1);
13 R = particles_parameters(2);
14 C = particles_parameters(3);
15 Beta1 = particles_parameters(4);
16
17 zd = exp(-Ts/(R*C));
18
19 a1 = -1 -zd;
20 a2 = zd;
21 bo = -Ro;
22 b1 = -Beta1*Ts/Qmax + Ro*(1+zd) - R*(1-zd);
23 b2 = zd*Theta_ant(4)*Ts/Qmax - Ro*zd + R*(1-zd);
24
25 Vt_est = -a1*Vt_med1 - a2*Vt_med2 + bo*i_med + b1*i_med1 + b2*i_med2;
26
27 DTheta = (particles_parameters' - Theta_ant)./Theta_ant;
28
29 Jv = ((Vt_est - Vt_med)/Vt_med)^2;
30 JTheta = DTheta*DTheta';
31
32 z = Jv + 1e-10*JTheta;
33
34 end

```

D.4 CÓDIGO PRINCIPAL ESTIMAÇÃO DO SOC COM EKF UTILIZADO NO DSC

```

1  #include "DSP2833x_Device.h"
2  #include "Define.h"
3
4  #include "stdio.h"
5
6  #include <epwm_config.h>
7  #include <adc.h>
8  #include <gpios.h>
9  #include <sys_config.h>
10 #include <Watchdog.h>
11 #include <interrupts.h>
12
13 #include <lqg.h>
14 #include <estimator.h>
15
16 #define ref_i 0.275
17 #define ref_v 4.2
18 #define n_parallel 1
19 #define counts_prot 20000
20 #define vb_min 2.4
21 #define vb_max 4.25
22 #define iL_min 0.05
23 #define iL_max 12
24 #define vo_max 10
25
26 void InitTimerInterrupt(void);
27 void protection_counters(void);
28 void protection_verification(void);
29 void protection_on(void);
30 void protection_off(void);
31 void update_variables(void);
32 void initialize_references(void);
33 void verify_end_of_charge(void);
34 extern void DSP28x_usDelay(unsigned long Count); // Rotina de
    ↪ delay em ms
35 interrupt void cpu_timer0_isr(void);
36 interrupt void epwm1_timer_isr(void);
37 interrupt void epwm2_timer_isr(void);
38
39 Uint16 i=0, j=0;
40 Uint16 start_control=0;
41
42 float uv = 0, u = 0, du = 0;
43 float vo=0, iL=0, vin=0, vo_ant = 0, iL_ant = 0;
44 float vb1, vb2, vb3, vb4, vb5;
45 float vb_l=0, vb=0, vb_ant=0;
46
47 float dXi_est_ant = 0, dXv_est_ant = 0, iL_est_ant = 0, vo_est_ant = 0;
48 float vb1_est=0, vb2_est=0, vb_est=0;
49 float dXi_est = 0, dXv_est = 0, iL_est = 0, vo_est = 0;
50 float vb1_est_ant=0, vb2_est_ant=0, vb_est_ant=0;
51
52 float uv_ant = 0, u_ant = 0, du_ant = 0;;
53 float ri = 0, rv = 0;
54
55 float a[4] = {0,0,0,0};
56 float *X_est;
57
58 float a_ekf[10] = {10000,0,0,1,0,0,0,0,0,0};
59 float *EKF_results;
60 float SOC_ekf_k, vrc_ekf, Voc_ekf, Vt_ekf_k_k1, Voc_ekf_k_k1;
61 float SOC_cc=0.2;
62 float Lekf1, Lekf2, dVocSoc;
63 float i_med_k1=0;//, i_med_k=0;
64 Uint16 Start_SOC_Estimation=1, SOC_estimation_complete=0;

```

```

65
66 float Vt_ekf[1800];
67 float SOC_ekf[1800];
68
69 Uint16 count_overnoltage=0, count_undervoltage=0, count_overload=0, count_shortC=0,
    ↪ count_turn_off_protection=0;
70 Uint16 count_overnoltage_vb=0, count_undervoltage_vb=0, count_lowcurrent=0;
71 Uint16 flag_overnoltage=0, flag_undervoltage=0, flag_overcurrent=0, flag_shortC=0;
72
73 Uint16 k=0;
74
75
76 void main(void){
77
78     X_est = &a[0];
79     EKF_results = &a_ekf[0];
80     Vt_ekf[0] = 0;
81     SOC_ekf[0] = 0;
82
83     DisableDog(); //Disables Watchdog
84     sys_config();
85
86     DSP28x_usDelay(10000000);
87
88     DINT;
89
90     InitPieCtrl();
91
92     InitPieVectTable();
93
94     InitTimerInterrupt();
95
96     DSP28x_usDelay(10000000);
97
98     init_gpios(); //Initialize GPIOs, GPIO11 as out
99     adc_config();
100    epwm_config();
101
102    DSP28x_usDelay(10000000);
103
104    InitPWMInterrupt();
105
106    for(;;){
107        if (Start_SOC_Estimation == 1){
108            k++;
109            if (k>=1799) {
110                k=1; Vt_ekf[0] = Vt_ekf[1799]; SOC_ekf[0] = SOC_ekf
                    ↪ [1799];
111            }
112
113            Start_SOC_Estimation = 0;
114            EKF_results = SOC_estimation_EKF(i_med_k1, X_est, EKF_results
                    ↪ );
115
116            SOC_ekf[k] = *(EKF_results+4);          vrc_ekf = *(
                    ↪ EKF_results+5);          Voc_ekf = *(EKF_results
                    ↪ +6);          Vt_ekf[k] = *(EKF_results+7);
117            Lekf1 = *(EKF_results+8);          Lekf2 = *(EKF_results
                    ↪ +9);
118
119            SOC_cc = SOC_estimation_CC(i_med_k1, SOC_cc);
120
121            i_med_k1 = *(X_est+2);
122        }
123    }
124 }
125
126
127 void protection_verification(void){
128     if (count_overnoltage > counts_prot || count_overnoltage_vb > counts_prot ||
        ↪ count_overload > counts_prot || count_shortC > counts_prot ||

```

```

129         ↪ count_undervoltage_vb>counts_prot){
130         count_turn_off_protection=0;
131         protection_on();
132         if (count_undervoltage > counts_prot || count_undervoltage_vb >
            ↪ counts_prot){
133             flag_undervoltage = 1; count_undervoltage = 0;
            ↪ count_undervoltage_vb = 0;
134         }
135         else{
136             if (count_undervoltage_vb > counts_prot){
137                 flag_undervoltage = 1; count_undervoltage_vb = 0;
138             }
139             else{
140                 if (count_overload > counts_prot){
141                     flag_overcurrent = 1; count_overload = 0;
142                 }
143                 else{
144                     flag_shortC = 1; count_shortC = 0;
145                 }
146             }
147         }
148         count_turn_off_protection++;
149         if (count_turn_off_protection > 10){
150             count_turn_off_protection=0;
151             protection_off();
152         }
153     }
154 }
155
156 void protection_counters(void){
157     if (vo > vo_max) count_undervoltage++; //Overvoltage
158     ↪ protection counter
159     if (vb > vb_max) count_undervoltage_vb++; //Overvoltage
160     ↪ protection counter
161     if (vb_l < vb_min) count_undervoltage_vb++; //Overvoltage
162     ↪ protection counter
163     if (iL > iL_max) count_overload++; //Overcurrent
164     ↪ protection counter
165     if (iL < iL_min) count_lowcurrent++;
166     if (vo < 0.1 && iL > 1) count_shortC++; //Short Circuit protection
167     ↪ counter
168 }
169
170 void protection_on(void){
171     du = -u;
172     ri = 0; rv = 0; u = 0; uv = 0; u_ant = 0; uv_ant = 0;
173     count_undervoltage = 0;
174     EALLOW;
175     EPwm1Regs.TZFRM.bit.OST = 1; //Force Interrupt
176     EDIS;
177 }
178
179 void protection_off(void){
180     initialize_references();
181     count_undervoltage = 0; count_undervoltage_vb = 0; count_overload = 0;
182     ↪ count_shortC = 0;
183     flag_undervoltage = 0; flag_undervoltage_vb = 0; flag_overcurrent = 0;
184     ↪ flag_shortC = 0;
185     EALLOW;
186     EPwm1Regs.TZCLR.bit.OST = 1; //Clear Interrupt
187     EDIS;
188 }
189
190 void update_variables(void){
191     dXi_est_ant = dXi_est;
192     dXv_est_ant = dXv_est;

```

```

190         iL_est_ant = iL_est;
191         vo_est_ant = vo_est;
192         vbl_est_ant = vbl_est;
193         vb2_est_ant = vb2_est;
194         vb_est_ant = vb_est;
195         iL_ant = iL;
196         vo_ant = vo;
197         vb_ant = vb;
198         du_ant = du;
199         uv_ant = uv;
200         u_ant = u;
201     }
202
203     void initialize_references(void){
204         rv = ref_v;
205         ri = n_parallel*ref_i;
206     }
207
208     void verify_end_of_charge(void){
209         if (count_lowcurrent >= counts_prot){
210             if (vb >= 4.2){
211                 count_turn_off_protection=0;
212                 protection_on();
213             }
214         }
215     }
216 }
217
218 interrupt void cpu_timer0_isr(void){
219     GpioDataRegs.GPATOGGLE.bit.GPIO11 = 1;
220
221     protection_verification();
222
223     if ((i >= 5) && start_control == 0){
224         initialize_references();
225         start_control = 1;
226     }
227     if (i>5) i=0;
228     i++;
229
230     if (Start_SOC_Estimation == 0){
231         Start_SOC_Estimation = 1;
232         j++;
233     }
234
235     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1; // Acknowledge this interrupt
236     ↪ to receive more interrupts from group 1
237 }
238
239
240
241 interrupt void epwm1_timer_isr(void){
242     EPwm1Regs.CMPA.half.CMPA = (Uint16)(u*TBPRD_EPWM1);
243
244     EPwm1Regs.ETCLR.bit.INT = 1; // Clear INT flag for
245     ↪ this timer
246     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3; // Acknowledge this
247     ↪ interrupt to receive more interrupts from group 3
248 }
249
250
251 interrupt void epwm2_timer_isr(void){
252     analog_trigger(); //starts ADC conversion
253
254     vo = 0.0026461865127994600*((float)(AdcRegs.ADCRESULT0>>4));
255     iL = 0.0030146882754948800*((float)(AdcRegs.ADCRESULT1>>4));
256     vin = 0.11806753959139265*((float)(AdcRegs.ADCRESULT2>>4));
257     vbl = 0.0015376941741190583*((float)(AdcRegs.ADCRESULT3>>4));

```

```

258 vb4 = 0.0015376941741190583*((float)(AdcRegs.ADCRESULT4>>4));
259 vb2 = 0.0015376941741190583*((float)(AdcRegs.ADCRESULT5>>4));
260 vb3 = 0.0015376941741190583*((float)(AdcRegs.ADCRESULT6>>4));
261 vb5 = 0.0015376941741190583*((float)(AdcRegs.ADCRESULT7>>4));
262
263 if (vb1 > vb4){
264     vb = vb1; vb_l = vb4;
265 } else {
266     vb = vb4; vb_l = vb1;
267 }
268
269 //Calculate the estimate states from Kalman Filter
270 X_est = kalman_filter(dXi_est_ant, dXv_est_ant, iL_est_ant, vb_est_ant,
    ↪ iL_ant, vb_ant, du_ant, X_est);
271 dXi_est = *(X_est);    dXv_est = *(X_est+1);    iL_est = *(X_est+2);
    ↪ vb_est = *(X_est+3);
272
273 //Calculate control action uv from voltage LQR controller
274 uv = lqr_controller_v(dXv_est, vb_est, uv_ant, ri, rv);
275 uv = saturation(uv, 0, iL_max);
276
277 //Calculate control action u from current LQR controller
278 u = lqr_controller_i(dXi_est, dXv_est, iL_est, uv, u_ant);
279 u = saturation(u, 0, 0.45);    //Saturation of u
280
281 du = u - u_ant;
282
283 if (start_control == 0){
284     protection_on();
285 }
286
287 //Protections
288 protection_counters();
289
290 //Update delayed variables
291 update_variables();
292
293 //verify end of charge condition
294 verify_end_of_charge();
295
296 EPwm2Regs.ETCLR.bit.INT = 1;    // Clear INT flag for
    ↪ this timer
297 PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;    // Acknowledge this
    ↪ interrupt to receive more interrupts from group 3
298 }

```

D.4.1 Função para estimação de carga utilizando Coulomb Counting

```

1  #include <estimator.h>
2
3  float SOC_estimation_CC(float i_med_k1, float SOC_k1){
4      float SOC_k;
5
6      SOC_k = SOC_k1 + i_med_k1*(Ts/QmaxC);
7
8      if (SOC_k > 1){
9          return 1;
10     } else if (SOC_k < 0){
11         return 0;
12     } else{
13         return SOC_k;
14     }
15 }
16

```

D.4.2 Função para estimação de carga utilizando EKF

```

1  #include <estimator.h>
2  #include <lqg.h>
3
4  float* SOC_estimation_EKF(float i_med_kl, float *X_est, float *estimation){
5
6      float SOC_est_k_kl, vrc_est_k_kl, Voc_est_k_kl, Vt_est_k_kl, dVocSoc;
7      float Skf, Lkf1, Lkf2;
8      float p11_k_kl, p12_k_kl, p21_k_kl, p22_k_kl;
9      //dXi_est = *(X_est);   dXv_est = *(X_est+1);   iL_est = *(X_est+2);
10     ↪ vb_est = *(X_est+3);
11     //e = p11;               *e+1 = p12;               *e+2 = p21;       *e+3 = p22;
12     //e+4 = SOC              *e+5 = vrc                *e+6 = Voc         *e+7 = Vt
13     //e+8 = Vt_k_kl
14     //SOC_est = *(SOC_estimation+4);   vrc_est = *(SOC_estimation+5);
15
16     SOC_est_k_kl = *(estimation+4) + i_med_kl*(Ts/QmaxC);
17     SOC_est_k_kl = saturation(SOC_est_k_kl, 0, 1);
18     vrc_est_k_kl = zd_est*(*(estimation+5)) - R_est*(1-zd_est)*i_med_kl;
19
20     p11_k_kl = *estimation + qsoc;                               //p11_k_kl =
21     ↪ p11 + qsoc;
22     p12_k_kl = (*(estimation+1))*zd_est;                         //p12_k_kl = p12*
23     ↪ zd_est;
24     p21_k_kl = (*(estimation+2))*zd_est;                         //p21_k_kl = p21*
25     ↪ zd_est;
26     p22_k_kl = (*(estimation+3))*zd_est*zd_est + qvrc;          //p22_k_kl =
27     ↪ p22*zd_est*zd_est + qvrc;
28
29     Voc_est_k_kl = calculate_Voc(SOC_est_k_kl);
30     dVocSoc = calculate_dVoc(SOC_est_k_kl);
31
32     Vt_est_k_kl = -vrc_est_k_kl + Ro_est*(*(X_est+2)) + Voc_est_k_kl;
33     //Vt_est_k_kl = saturation(Vt_est_k_kl, 2, 4.2);
34
35     //EKF Propagation phase.. X(k|k) = X(k|k-1) + L(k)*e(k)
36     Skf = p11_k_kl*dVocSoc*dVocSoc - dVocSoc*(p12_k_kl + p21_k_kl) + p22_k_kl +
37     ↪ Rkf;
38     Lkf1 = (dVocSoc*p11_k_kl - p12_k_kl)/Skf;
39     Lkf2 = (dVocSoc*p21_k_kl - p22_k_kl)/Skf;
40
41     /*p11*/ *estimation = (1-Lkf1*dVocSoc)*p11_k_kl + Lkf1*p21_k_kl;
42     ↪ //p11 = (1-Lkf1*dVocSoc)*p11_k_kl + Lkf1*p21_k_kl;
43     /*p12*/ *(estimation+1) = (1-Lkf1*dVocSoc)*p12_k_kl + Lkf1*p22_k_kl;
44     ↪ //p12 = (1-Lkf1*dVocSoc)*p12_k_kl + Lkf1*p22_k_kl;
45     /*p21*/ *(estimation+2) = -Lkf2*dVocSoc*p11_k_kl + (1+Lkf2)*p21_k_kl;
46     ↪ //p21 = -Lkf2*dVocSoc*p11_k_kl + (1+Lkf2)*p21_k_kl;
47     /*p22*/ *(estimation+3) = -Lkf2*dVocSoc*p12_k_kl + (1+Lkf2)*p22_k_kl;
48     ↪ //p22 = -Lkf2*dVocSoc*p12_k_kl + (1+Lkf2)*p22_k_kl;
49
50     /*SOC_est_k*/ *(estimation+4) = SOC_est_k_kl + Lkf1*(*(X_est+3)) -
51     ↪ Vt_est_k_kl;                               //(Vt_med_k - Vt_est_k_kl)
52     *(estimation+4) = saturation(*(estimation+4), 0, 1)
53     ↪ ;
54     /*vrc_est_k*/ *(estimation+5) = vrc_est_k_kl + Lkf2*(*(X_est+3)) -
55     ↪ Vt_est_k_kl;                               //(Vt_med_k - Vt_est_k_kl)
56     /*Voc_est_k*/ *(estimation+6) = calculate_Voc(*(estimation+4));
57     /*Vt_est_k*/ *(estimation+7) = -(*(estimation+5)) + Ro_est*(*(X_est+2)) + *(
58     ↪ estimation+6);
59
60     *(estimation+8) = Lkf1;
61     *(estimation+9) = Lkf2;
62
63     return estimation;
64 }
65
66 float calculate_Voc(float SOC){
67     return (2.559 + 21.030000*SOC - 234.70000*SOC*SOC + 1511.0000*SOC*SOC*SOC -
68     ↪ 5802.0000*SOC*SOC*SOC*SOC + 13690.0000*SOC*SOC*SOC*SOC*SOC -

```

```

54         ↪ 19980.0000*SOC*SOC*SOC*SOC*SOC*SOC + 17560.0000*SOC*SOC*SOC*SOC*SOC*
55         ↪ SOC*SOC - 8505.0000*SOC*SOC*SOC*SOC*SOC*SOC*SOC + 1745.0000*SOC*
56         ↪ SOC*SOC*SOC*SOC*SOC*SOC*SOC);
57     }
58     float calculate_dVoc(float SOC){
59         return (21.030000 - 2*234.70000*SOC + 3*1511.0000*SOC*SOC - 4*5802.0000*SOC*
60         ↪ SOC*SOC + 5*13690.0000*SOC*SOC*SOC*SOC - 6*19980.0000*SOC*SOC*SOC*SOC
61         ↪ *SOC + 7*17560.0000*SOC*SOC*SOC*SOC*SOC*SOC - 8*8505.0000*SOC*SOC*SOC
62         ↪ *SOC*SOC*SOC + 9*1745.0000*SOC*SOC*SOC*SOC*SOC*SOC*SOC);
63     }

```

As baterias, em especial as de Lítio, vêm se tornando cada vez mais integradas ao cotidiano das pessoas através de inúmeras aplicações, sejam estas celulares, notebooks ou mesmo os promissores Veículos Elétricos. Diante deste cenário, técnicas de estimação do Estado de Carga (SOC) das baterias têm chamado atenção dos pesquisadores, que buscam o aprimoramento dos métodos de estimação já existentes. Este trabalho traz um estudo geral sobre as técnicas de estimação do SOC já disponíveis na literatura sintetizando os principais pontos de vista apontados por diversos autores, além da implementação da estimação do SOC através do Filtro de Kalman Estendido, método já conhecido na literatura, e de duas metodologias aqui propostas baseadas no Filtro de Kalman Adaptativo, juntamente com os resultados e análises de cada uma das metodologias de estimação. Além da estimação do SOC, esta dissertação também aborda a modelagem e simulação de baterias, necessárias ao estudo de estimação do SOC, e a construção de um conversor Duplo-Forward com um controlador LQG, utilizado como carregador de baterias.

Orientador: Sérgio Vidal Garcia Oliveira

Joinville, 2016