



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

João Pedro Brunoni

Controle preditivo de cômputo rápido aplicado a inversores trifásicos de dois níveis

Florianópolis
2024

João Pedro Brunoni

Controle preditivo de cômputo rápido aplicado a inversores trifásicos de dois níveis

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia de Automação e Sistemas.

Orientador: Prof. Rodolfo César Costa Flesch, Dr.
Coorientadores: Prof. Julio Elias Normey Rico, Dr.,
Prof. Vinícius Berndsen Peccin, Dr.

Florianópolis
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Brunoni, João Pedro

Controle preditivo de cômputo rápido aplicado a inversores trifásicos de dois níveis / João Pedro Brunoni ; orientador, Rodolfo César Costa Flesch, coorientador, Julio Elias Normey-Rico, coorientador, Vinícius Berndsen Peccin, 2024.

85 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2024.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Controle preditivo. 3. Inversores trifásicos. 4. Eletrônica de potência. 5. Sphere Decoding Algorithm. I. Flesch, Rodolfo César Costa. II. Normey-Rico, Julio Elias. III. Peccin, Vinícius Berndsen IV. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. V. Título.

João Pedro Brunoni

Controle preditivo de cômputo rápido aplicado a inversores trifásicos de dois níveis

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Daniel Martins Lima, Dr.
Universidade Federal de Santa Catarina

Prof. Telles Brunelli Lazzarin, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia de Automação e Sistemas.

Coordenação do Programa de Pós-Graduação

Prof. Rodolfo César Costa Flesch, Dr.
Orientador

Florianópolis, 2024.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que contribuíram de maneira significativa para a realização desta dissertação. Primeiramente, agradeço aos meus pais, Vilmar e Luciana, assim como à Kelli, por todo amor, apoio e incentivo que sempre me ofereceram. Vocês foram fundamentais em cada etapa da minha jornada acadêmica, e sem o suporte de vocês, este trabalho não teria sido possível.

Agradeço também às minhas irmãs, Julia e Sophia, por estarem sempre ao meu lado, compartilhando alegrias e desafios. A confiança e o incentivo de vocês foram essenciais para que eu pudesse seguir em frente. À minha namorada, Sandi, sou imensamente grato pelo amor, paciência e compreensão ao longo deste percurso. Sua presença constante e seu apoio incondicional foram cruciais para a conclusão deste trabalho.

Expresso minha gratidão ao meu orientador, Rodolfo, e também aos meus coorientadores, Julio e Vinícius, pelo suporte técnico e intelectual, bem como pelas valiosas contribuições indispensáveis para o desenvolvimento desta dissertação.

Aos meus amigos de Chapecó, Carlos, Guilherme, Ricardo, Leonardo e Marcos, e ao Bruno, amigo de longa data que conheço desde a infância, sou grato pela amizade e pelos momentos descontraídos. Vocês tornaram esta jornada mais leve e divertida. Aos meus amigos que fiz dentro da UFSC, João, Gabriel e Roberto, agradeço pela parceria e pelo apoio durante este percurso. Aos colegas de laboratório do LCA-Pesquisa, Nelson, Pedro, Bruno e Felipe, sou grato pela colaboração e pela camaradagem no dia a dia de trabalho. Compartilhar este caminho com todos vocês foi uma experiência enriquecedora e motivadora. A todos, deixo meu mais sincero agradecimento.

Este trabalho obteve o apoio financeiro do Programa de Recursos Humanos da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis — PRH-ANP, suportado com recursos provenientes do investimento de empresas petrolíferas qualificadas na Cláusula de P, D&I da Resolução ANP nº 50/2015.

*"Se fosse fácil achar o caminho das pedras
Tantas pedras no caminho não seria ruim."
Engenheiros do Hawaii*

RESUMO

Nesta dissertação de mestrado, são comparados algoritmos de cômputo rápido presentes na literatura para controle preditivo baseado em modelo com conjunto de controle finito (FCS-MPC, do inglês, *Finite Control Set Model Predictive Control*), quando aplicados a um inversor tipo fonte de tensão trifásico de dois níveis. Uma combinação entre duas das abordagens estudadas é proposta, resultando em um algoritmo inédito na literatura, denominado de vetores adjacentes pré-selecionados para algoritmo de decodificação via esferas (PAV-SDA, do inglês, *Preselected Adjacent Vectors for Sphere Decoding Algorithm*). Para comparar o desempenho dessas técnicas estudadas, os resultados obtidos por simulações em diferentes cenários são utilizados em análises de rastreamento de referência e esforço de cômputo dos algoritmos. Para horizontes de uma amostra de comprimento, a técnica FCS-MPC com conjunto simplificado de controle, em conjunto com o modelo de predição autorregressivo integrado com média móvel e entrada controlada, apresentou os melhores índices de seguimento de referência, com 21% do tempo de cômputo médio necessário pela segunda técnica mais rápida. Para horizontes de duas amostras ou mais, as técnicas que utilizam o modelo de predição autorregressivo com média móvel e entrada controlada obtêm melhores resultados com menos tempo de cômputo gasto. Por fim, a técnica proposta PAV-SDA demonstra ser a melhor opção para horizontes de três amostras ou mais, pois fornece os melhores índices de seguimento de referência com até 29% menos tempo de cômputo gasto.

Palavras-chave: Controle preditivo, FCS-MPC, controle de inversores elétricos, algoritmo de cômputo rápido.

ABSTRACT

In this master's dissertation, fast computation algorithms present in the literature for Finite Control Set Model Predictive Control (FCS-MPC) are compared when applied to a three-phase two-level voltage source inverter. A combination of two of the studied approaches is proposed, resulting in a novel algorithm, named Preselected Adjacent Vectors for Sphere Decoding Algorithm (PAV-SDA). To compare the performance of these studied techniques, the simulation results of different scenarios are analyzed regarding the reference tracking performance and the computational effort spent by the algorithms. For one-sample length horizons, the FCS-MPC with Simplified Control Set combined with the Controlled Auto-Regressive Integrated Moving Average model showed the best reference tracking indexes, with 21% of the average computational time required by the second fastest technique. For horizons of two or more samples, techniques using the Controlled Auto-Regressive Moving Average model achieve better results with less computational time spent. Finally, the proposed PAV-SDA technique proves to be the best option for horizons of three or more samples, providing the best reference tracking indices with up to 29% less computational time spent.

Keywords: Predictive control, FCS-MPC, electric inverter control, fast computation algorithm.

LISTA DE FIGURAS

Figura 1 – Esquema simplificado de um VSI trifásico	20
Figura 2 – Descrição das variáveis de controle	21
Figura 3 – Vetores de tensão de entrada representados no plano complexo. Adaptado de Rodríguez e Estay (2012).	22
Figura 4 – Implementação do FCS-MPC: (a) estrutura do algoritmo para compensação de atraso, com N e N_u de duas amostras; (b) exemplo de vetores de tensão e seus instantes de tempo correspondentes (apenas três opções de controle são consideradas em cada ramo por uma questão de representação).	34
Figura 5 – Diagrama esquemático de exemplo da seleção do vetor de tensão ótimo: (a) posicionamento de \mathbf{V}_{ref} e vetores utilizados na primeira comparação; (b) área delimitada pela primeira comparação e vetores utilizados na segunda comparação; (c) área resultante pela segunda comparação e vetores utilizados na terceira comparação. Adaptado de Guo <i>et al.</i> (2021).	40
Figura 6 – Fluxograma do algoritmo FCS-MPC-SCS para: (a) seleção do vetor de tensão não nulo mais próximo da referência; (b) seleção do vetor ótimo. Adaptado de Guo <i>et al.</i> (2021).	42
Figura 7 – Exemplo do FCS-MPC-SCS adaptado para o VSI trifásico de dois níveis: (a) posicionamento de \mathbf{V}_{ref} e vetores utilizados na primeira comparação; (b) área delimitada pela primeira comparação e vetores utilizados na segunda comparação; (c) área resultante pela segunda comparação e vetor nulo utilizado na última comparação.	43
Figura 8 – Fluxograma do algoritmo FCS-MPC-SCS com adaptações para: (a) seleção do vetor de tensão não nulo mais próximo da referência; (b) seleção do vetor ótimo.	44
Figura 9 – Exemplo de distâncias entre vetores do ponto de vista geométrico, para $\mathbf{V}_{opt}(k-1) = \mathbf{V}_1$	47
Figura 10 – Exemplo de funcionamento do algoritmo de vetores adjacentes.	48
Figura 11 – Visualização do SDA no plano ab para o horizonte $N = 1$. (a) Problema de otimização no sistema de coordenadas ortogonal. (b) Problema de otimização no sistema de coordenadas transformado criado por \mathbf{H}_{sda} . Adaptado de Geyer e Quevedo (2014).	50
Figura 12 – Diagrama resumido da simulação com a localização dos parâmetros descritos na Tabela 3.	58
Figura 13 – Comparação entre o tempo de cômputo necessário e o MSE para cada técnica com o modelo CARMA.	68
Figura 14 – Comparação entre o tempo de cômputo necessário e o DHT para cada técnica com o modelo CARMA.	69

Figura 15 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARMA - Seguimento de referência da tensão de saída e degrau de carga.	71
Figura 16 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARMA - Corrente na saída e degrau de carga.	72
Figura 17 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARIMA - Seguimento de referência da tensão de saída e degrau de carga.	72
Figura 18 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARIMA - Corrente na saída e degrau de carga.	73
Figura 19 – PAV-FCS-MPC e PAV-SDA com modelo de predição CARMA - Seguimento de referência da tensão de saída e degrau de carga.	73
Figura 20 – PAV-FCS-MPC e PAV-SDA com modelo de predição CARMA - Corrente na saída e degrau de carga.	74
Figura 21 – Comparação do seguimento de referência entre os grupos de técnicas.	75
Figura 22 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARMA - mudança de referência entre diferentes comprimentos de horizonte.	75
Figura 23 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARIMA - mudança de referência entre diferentes comprimentos de horizonte.	76
Figura 24 – PAV-FCS-MPC e PAV-SDA com modelo de predição CARMA - mudança de referência entre diferentes comprimentos de horizonte.	76

LISTA DE TABELAS

Tabela 1 – Tabela de pesquisa para os vetores de tensão de entrada no filtro LC	22
Tabela 2 – Tabela de pesquisa para encontrar a quantidade de troca de estados das chaves entre vetores.	46
Tabela 3 – Parâmetros do sistema controlado utilizados em todas as simulações. . . .	57
Tabela 4 – FCS-MPC padrão com predição feita por equação a diferenças.	59
Tabela 5 – FCS-MPC padrão com predição feita por modelo CARIMA.	59
Tabela 6 – FCS-MPC padrão com predição feita por modelo CARMA.	59
Tabela 7 – FCS-MPC-SCS para $N = 1$	61
Tabela 8 – PAV-FCS-MPC com predição feita por modelo CARMA - Parte 1.	61
Tabela 9 – PAV-FCS-MPC com predição feita por modelo CARMA - Parte 2.	62
Tabela 10 – SDA com predição feita por modelo CARMA.	64
Tabela 11 – SDA com predição feita por modelo CARIMA.	65
Tabela 12 – PAV-SDA com predição feita por modelo CARMA	65
Tabela 13 – Comparação do MSE, DHT e T_c entre as técnicas para $N = 1$ para o modelo CARMA.	66
Tabela 14 – Comparação do MSE, DHT e T_c entre as técnicas para $N = 1$ para o modelo CARIMA.	66
Tabela 15 – Comparação do MSE, DHT e T_c entre as técnicas para $N > 1$	67
Tabela 16 – Erro quadrático acumulado (e_{Σ}^2) para $0,03s \leq t \leq 0,036s$	77
Tabela 17 – Tempo de assentamento (t_a) da componente d da saída das técnicas com $\pm 7V$ de tolerância para a referência.	78

LISTA DE ABREVIATURAS E SIGLAS

CA	Corrente Alternada.
CARIMA	Autorregressivo integrado com média móvel e entrada controlada, do inglês <i>Controlled Auto-Regressive Integrated Moving Average</i> .
CARMA	Autorregressivo com média móvel e entrada controlada, do inglês <i>Controlled Auto-Regressive Moving Average</i> .
CC	Corrente Contínua.
CCS-MPC	MPC com conjunto contínuo de controle, do inglês <i>Continuous Control Set MPC</i> .
CEP	Conversor Eletrônico de Potência.
DHT	Distorção Harmônica Total.
FCS-MPC	MPC com conjunto finito de controle, do inglês <i>Finite Control Set MPC</i> .
FCS-MPC-SCS	FCS-MPC com conjunto simplificado de controle, do inglês <i>FCS-MPC-Simplified Control Set</i> .
FT	Função de Transferência.
GPC	Controle preditivo generalizado, do inglês <i>Generalized Predictive Control</i> .
Iter.	Quantidade média de iterações executadas pelo controlador em cada período de amostragem.
MFD	Descrição matricial fracionária, do inglês <i>Matrix Fraction Description</i> .
MIMO	Múltiplas-entradas múltiplas-saídas, do inglês <i>Multiple Input Multiple Output</i> .
Mod. pred.	Modelo de previsão utilizado pelo controlador.
MPC	Controle preditivo baseado em modelo, do inglês <i>Model Predictive Control</i> .
MSE	Erro médio quadrático, do inglês <i>Mean Squared Error</i> .
NPC	Ponto neutro grampeado, do inglês <i>Neutral Point Clamped</i> .
PAV-FCS-MPC	Vetores adjacentes pré-selecionados para FCS-MPC, do inglês <i>Preselected Adjacent Vectors-FCS-MPC</i> .
PAV-SDA	Vetores adjacentes pré-selecionados para algoritmo de decodificação via esferas, do inglês <i>Preselected Adjacent Vectors for Sphere Decoding Algorithm</i> .
PWM	Modulação por largura de pulso, do inglês <i>Pulse Width Modulation</i> .
RP	Regime Permanente.
RT	Regime Transitório.
SDA	Algoritmo de decodificação via esferas, do inglês <i>Sphere Decoding Algorithm</i> .
SISO	Entrada única e saída única, do inglês <i>Single Input Single Output</i> .
UPS	Fonte de energia ininterrupta, do inglês <i>Uninterruptible Power Supplies</i> .
VM	Variável Manipulada.

VP	Variável de Processo.
VSI	Inversor tipo fonte de tensão, do inglês <i>Voltage Source Inverter</i> .
ZOH	Sustentador de ordem zero, do inglês <i>Zero-Order Hold</i> .

LISTA DE SÍMBOLOS

N	Quantidade de amostras no horizonte de predição.
N_u	Quantidade de amostras no horizonte de controle.
\mathbf{V}	Vetor de tensão de entrada no filtro LC representado com valores complexos.
k	Tempo discreto amostrado.
\mathbf{H}_{sda}	Matriz que gera o espaço utilizado pelo algoritmo de decodificação via esferas.
L	Valor da indutância do filtro.
C	Valor da capacitância do filtro.
T_c	Tempo de cômputo médio necessário para o algoritmo implementado em cada amostra de controle.
e	Erro entre a saída e a referência do sistema.
t_a	Período de assentamento da resposta do sistema.
R	Valor da resistência da carga.
V	Tensão elétrica.
S	Estado da chave.
e	Número de Euler.
i	Corrente elétrica.
G	Função de transferência.
ω	Frequência.
ζ	Coefficiente de amortecimento.
T_s	Período de amostragem da planta.
y	Saída ou variável de processo do sistema.
u	Sinal de controle ou variável manipulada do sistema.
\mathbb{Z}	Conjunto dos números inteiros.
\hat{y}	Predição da saída do sistema.
\hat{u}	Sinal de controle estimado que ainda não foi definido e implementado.
N_2	Amostra final do horizonte de predição.
N_1	Amostra inicial do horizonte de predição.
N_{u1}	Amostra inicial do horizonte de controle.
N_{u2}	Amostra final do horizonte de controle.
d	Atraso de transporte considerado no sistema.
Δu	Incremento do sinal de controle.
$\hat{\mathbf{y}}$	Vetor com predições de saídas futuras do sistema.
$\Delta \mathbf{u}$	Matriz/vetor com os incrementos dos sinais de controle.
\mathbf{f}	Vetor coluna que armazena a resposta livre do sistema.
\mathbb{R}	Conjunto dos números reais.
\mathbf{G}	Matriz de resposta ao degrau do sistema.
\mathbf{y}	Vetor que armazena as saídas do sistema.

n_ε	Quantidade de entradas em um sistema do tipo MIMO.
n_φ	Quantidade de saídas em um sistema do tipo MIMO.
\mathbf{M}_t	Matriz de transferência.
φ	Índice da saída em um sistema do tipo MIMO.
ε	Índice da entrada em um sistema do tipo MIMO.
\mathbf{u}	Matriz/vetor com sinais de controle.
g	Função custo ou custo que deve ser minimizado pelo controlador utilizado.
λ	Fator de ponderação do erro do sistema.
\hat{e}	Predição do erro entre a saída predita e a referência futura do sistema.
w	Referência para a saída do sistema.
δ	Fator de ponderação do sinal de controle do sistema.
w	Vetor com as referências da amostra atual e das amostras futuras para a saída do sistema.
\mathbf{Q}_λ	Matriz bloco diagonal positiva definida que armazena as ponderações do erro do sistema.
\mathbf{Q}_δ	Matriz bloco diagonal positiva definida que armazena as ponderações do sinal de controle do sistema.
σ	Distância calculada entre vetores.
Q	Quantidade de chaves cujos estados são alterados para mudar de uma opção para outra.
ψ	Fluxo concatenado.
\mathbb{N}	Conjunto dos números naturais.
P	Transformada de Clarke.
e	Vetor que armazena o erro entre a saída e a referência do sistema.
i	Vetor com valores de corrente elétrica.
$\bar{\mathbf{u}}$	Vetor com sinais de controle transformados para o espaço gerado com o algoritmo de decodificação via esferas.
\mathbb{U}	Conjunto finito onde estão todas as sequências de soluções factíveis.
ρ	Raio da esfera no algoritmo de decodificação via esferas.
$\overline{\Delta \mathbf{u}}$	Vetor com os incrementos dos sinais de controle transformados para o espaço gerado com o algoritmo de decodificação via esferas.
F_w	Frequência da referência.
F_s	Frequência de amostragem da planta.
T_s	Período de amostragem da planta.
T_{sim}	Tempo utilizado para simulações.
T_{ana}	Tempo utilizado para análises.

SUMÁRIO

1	INTRODUÇÃO	17
1.1	PROBLEMA DE PESQUISA E SOLUÇÃO PROPOSTA	18
1.2	OBJETIVOS	18
1.2.1	Objetivo Geral	19
1.2.2	Objetivos específicos	19
1.3	ESTRUTURA DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	INVERSORES TRIFÁSICOS	20
2.2	MODELAGEM DINÂMICA	22
2.2.1	Modelo de tempo contínuo	23
2.2.2	Modelo de tempo discreto	24
2.3	CONTROLE PREDITIVO BASEADO EM MODELO	26
2.3.1	Predição da saída do sistema	26
2.3.2	MPC com conjunto finito de controle	32
2.3.3	Controle preditivo generalizado	34
2.4	CONSIDERAÇÕES	37
3	ALGORITMOS PARA REDUÇÃO DO TEMPO DE CÔMPUTO DO FCS-MPC	38
3.1	DIMINUIÇÃO DO TEMPO DE CÔMPUTO	38
3.1.1	Redução de vetores disponíveis por método iterativo: formulação original	39
3.1.2	Redução de vetores disponíveis por método iterativo: adaptações propostas	41
3.1.3	Redução de vetores disponíveis por vetores adjacentes: formulação original	45
3.1.4	Redução de vetores disponíveis por vetores adjacentes: adaptações propostas	45
3.1.5	Algoritmo de decodificação via esferas: formulação original	48
3.1.6	Algoritmo de decodificação via esferas: adaptações propostas	53
3.1.7	Vetores adjacentes pré-selecionados para algoritmo de decodificação via esferas	54
3.2	CONSIDERAÇÕES	55
4	RESULTADOS E ANÁLISES	57
4.1	ASPECTOS GERAIS	57
4.2	ANÁLISE DO REGIME PERMANENTE	58
4.3	COMPARAÇÃO ENTRE OS ÍNDICES DE REGIME PERMANENTE	66
4.4	ANÁLISE DO REGIME TRANSITÓRIO	71

4.5	CONSIDERAÇÕES	78
5	CONSIDERAÇÕES FINAIS	79
5.1	CONCLUSÕES	79
5.2	SUGESTÕES PARA TRABALHOS FUTUROS	80
	REFERÊNCIAS	81

1 INTRODUÇÃO

Conversores Eletrônicos de Potência (CEPs) são amplamente utilizados em processos nos quais tensão, corrente ou potência precisam ser controlados para que o processo opere de acordo com as especificações do usuário (BLAABJERG, 2021). No contexto de sistemas distribuídos para geração de energia, a principal topologia de CEP utilizada é o inversor tipo fonte de tensão (VSI, do inglês *Voltage Source Inverter*) trifásico (JUDEWICZ *et al.*, 2016). Essa topologia também é empregada em fontes de energia ininterrupta (UPSs, do inglês *Uninterruptible Power Supplies*), frequentemente utilizadas na indústria para prover uma fonte constante de potência em situações críticas (BORDONS; MONTERO, 2015). Nestes casos, os principais problemas encontrados estão relacionados com a qualidade da energia fornecida, como altos índices de Distorção Harmônica Total (DHT), que reduzem a eficiência do sistema e podem danificar os equipamentos que estão conectados à rede fornecida (PATHIK *et al.*, 2023). A topologia VSI também é utilizada em drives de frequência variável, dispositivos muito empregados para acionamento de máquinas elétricas com velocidade angular variável, como é o caso de bombas elétricas submersíveis na indústria de petróleo e gás (LIANG; FLEMING, 2013). Estes sistemas também são impactados com o aumento da DHT, visto que as harmônicas indesejáveis ocasionam perdas de potências nas bombas utilizadas e, conseqüentemente, reduzem a eficiência dos sistemas de elevação (LYAKHOMSKII *et al.*, 2022).

Existem diversas estratégias de controle que podem ser utilizadas em inversores, desde as mais simples, como controladores lineares (ALATHAMNEH; YANG; NELMS, 2021) ou por histerese (CHEREAU; AUGER; LORON, 2007), até alternativas mais complexas, como o controle por modos deslizantes (MEMIJE *et al.*, 2020) ou o controle preditivo baseado em modelo (MPC, do inglês *Model Predictive Control*) (JIN *et al.*, 2019). O uso de abordagens do tipo MPC em CEPs tem crescido significativamente nos últimos anos, pois oferta uma solução geral que pode lidar com diferentes cenários. O controle preditivo é uma estratégia de controle ótimo, elaborada nos anos 1970, que calcula a ação de controle por meio da minimização de uma função custo. Os termos dessa função podem ser escolhidos especificamente para cada problema de controle, o que torna o MPC um controlador flexível. Essas variáveis costumam depender de predições das saídas futuras e dos sinais de controle que seriam aplicados, dentro de horizontes denominados por horizonte de predição e de controle, respectivamente. A minimização dessa função custo também pode considerar restrições nas variáveis manipuladas, de processo e nos estados internos do sistema, caso a implementação seja feita em espaço de estados. O MPC ainda pode ser aplicado para processos não lineares e sistemas do tipo múltiplas entradas e múltiplas saídas (MIMO, do inglês *Multiple Input Multiple Output*). Como os CEPs podem ter restrições de operação e apresentam um número finito de opções de chaveamento, a utilização do controle preditivo sempre pareceu ser interessante, mas foi nos últimos anos, com o incremento do poder computacional das plataformas de controle, que o emprego do MPC em conversores passou a ser factível e amplamente utilizado (RODRÍGUEZ; ESTAY, 2012).

Ao tratar as implementações em conversores, especificamente, é possível classificar as estratégias de MPC em dois diferentes grupos: MPC com conjunto finito de controle (FCS-MPC, do inglês *Finite Control Set* MPC) e MPC com conjunto contínuo de controle (CCS-MPC, do inglês *Continuous Control Set* MPC). As técnicas que se enquadram dentro do CCS-MPC possuem infinitas possibilidades de sinal de controle, isto é, a solução ótima para a função custo pode ser qualquer valor dentro de um intervalo definido pelas restrições. Porém, devido às combinações possíveis entre as chaves semicondutoras do conversor, as possibilidades reais de aplicação são limitadas a um conjunto finito de opções. Deste modo, para atingir o sinal de controle calculado, utiliza-se um modulador para acionar o conversor através de técnicas de modulação, sendo a modulação por largura de pulso (PWM, do inglês *Pulse Width Modulation*) a opção mais comum (BORDONS; MONTERO, 2015). Além disso, essa formulação possibilita implementações com horizontes de múltiplas amostras, pois, além do impacto causado pelo aumento dos horizontes ser atenuado nessa estrutura de algoritmos (BORDONS; MONTERO, 2015), a utilização de um modulador faz com que os ciclos de controle possam acontecer em uma frequência mais lenta (BLAABJERG, 2021). A utilização do FCS-MPC em CEPs tem crescido consideravelmente, devido ao sucesso em lidar com as comutações discretas desses processos (MCNABB; WANG, 2023). A técnica explora o número finito de opções para sinal de controle, correspondente às opções de chaveamento disponíveis no conversor utilizado. Por conta disso, o controle é aplicado diretamente no conversor, sem precisar de qualquer tipo de modulação (BLAABJERG, 2021). Deste modo, a frequência do controlador impacta diretamente a qualidade do sinal da saída e, por isso, os controladores devem possuir baixos tempos de cômputo para atender a rápida dinâmica dos conversores. O resultado é um balanceamento entre desempenho e custo computacional no FCS-MPC que costuma definir um controlador com horizontes de uma única amostra (BLAABJERG, 2021).

1.1 PROBLEMA DE PESQUISA E SOLUÇÃO PROPOSTA

Como previamente comentado, a rápida dinâmica dos conversores eletrônicos faz com que o tempo de cômputo do controlador precise ser muito baixo e, por conta disso, o tamanho dos horizontes utilizados tende a ser reduzido, limitando o desempenho das técnicas.

Portanto, este trabalho visa implementar métodos para reduzir o esforço computacional de algoritmos de FCS-MPC e submetê-los a análises comparativas. Além disso, duas dessas técnicas são combinadas de forma inédita, com o intuito de explorar as vantagens de cada uma e reduzir ainda mais o tempo de cômputo obtido. Deste modo, a implementação de horizontes maiores em equipamentos que necessitam de curtos períodos de amostragem para serem controlados passa a ser uma possibilidade factível.

1.2 OBJETIVOS

Esta seção formaliza os objetivos do trabalho, conforme descrito a seguir.

1.2.1 Objetivo Geral

O objetivo geral desta pesquisa de mestrado é propor um algoritmo de cômputo rápido de FCS-MPC para implementação em inversores trifásicos de dois níveis e compará-lo com algoritmos semelhantes encontrados na literatura.

1.2.2 Objetivos específicos

Com base no objetivo geral descrito, foram elencados alguns objetivos específicos:

- implementar técnicas que diminuam o tempo de cômputo do FCS-MPC e comparar seus resultados;
- avaliar o desempenho computacional exigido para o aumento dos horizontes nessas formulações da literatura;
- propor um algoritmo de cômputo rápido para FCS-MPC;
- comparar os resultados obtidos com o algoritmo proposto e com outras técnicas encontradas na literatura para redução de tempo de cômputo do FCS-MPC.

1.3 ESTRUTURA DO TRABALHO

Esta dissertação está estruturada em 5 capítulos. O Capítulo 2 explica a fundamentação teórica sobre os assuntos abordados neste trabalho, como o princípio de funcionamento e a modelagem dinâmica de inversores trifásicos, assim como conceitos introdutórios de controle preditivo baseado em modelo e sua formulação FCS-MPC. No Capítulo 3 são descritos os algoritmos de cômputo rápido de FCS-MPC encontrados na literatura, em conjunto com as adaptações propostas utilizadas neste trabalho. Um algoritmo proposto que combina duas das técnicas abordadas é encontrado ao fim do capítulo. Os resultados obtidos com as técnicas abordadas são apresentados e discutidos no Capítulo 4. Por fim, o Capítulo 5 retrata as considerações finais sobre o trabalho desenvolvido e os resultados obtidos, assim como novas linhas de trabalhos futuros sugeridas para dar continuidade ao tema.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos introdutórios sobre os assuntos abordados neste trabalho. A Seção 2.1 apresenta um inversor trifásico de dois níveis e seu princípio de funcionamento. Na Seção 2.2, é apresentada a dedução dos modelos dinâmicos de tempo contínuo e tempo discreto do sistema. Por fim, a Seção 2.3 destaca características do MPC para CEPs, demonstrando como se calcula a previsão da saída do sistema para diferentes tipos de modelo e quais são os princípios de funcionamento das formulações FCS-MPC, controle preditivo generalizado (GPC, do inglês *Generalized Predictive Control*) e a sua variante CARMA-GPC.

2.1 INVERSORES TRIFÁSICOS

Os conversores eletrônicos podem ser classificados de acordo com a sua operação. Quando esses dispositivos são utilizados para transformar uma fonte de energia de Corrente Contínua (CC) em uma saída de Corrente Alternada (CA), eles são classificados como inversores. Esse tipo de saída em forma de onda é necessário em equipamentos como drives de velocidade ajustável e UPSs, que são aplicações amplamente utilizadas na indústria (RASHID, 2011).

Para que esses equipamentos sejam alimentados corretamente, o sinal CA fornecido costuma ter magnitude, fase e frequência controladas. Além disso, quando a alimentação CC do inversor é proveniente de uma fonte de tensão, a topologia é denominada como VSI. Em aplicações de média para alta potência, são utilizados VSIs trifásicos, nos quais a saída é formada por três sinais senoidais defasados em 120 graus (RASHID, 2011). A Figura 1 retrata uma visão simplificada do funcionamento de um VSI trifásico, topologia escolhida para este trabalho.

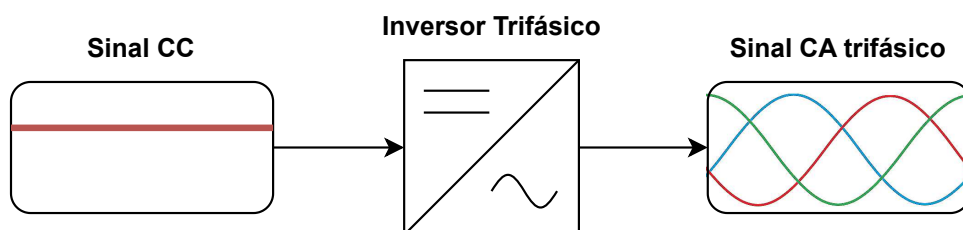


Figura 1 – Esquema simplificado de um VSI trifásico

O sistema controlado, conforme ilustrado na Figura 2, conta com um VSI trifásico de dois níveis, utilizado para fornecer sinal trifásico CA nas carga resistivas conectadas em cada fase. Entre o conversor e a carga, foi posicionado um filtro LC , configuração comum na indústria para reduzir a DHT do sinal da saída (MOHAMED *et al.*, 2013). Esse sistema é semelhante ao que foi apresentado em Bordons e Montero (2015) e conta com os mesmos valores para os componentes do filtro e da carga, ou seja, $R_x = 60\Omega$, $L_x = 2\text{mH}$, $C_x = 50\mu\text{F}$

e a mesma tensão no barramento CC ($V_{CC} = 400V$), o que facilitou os estágios iniciais de validação dos algoritmos.

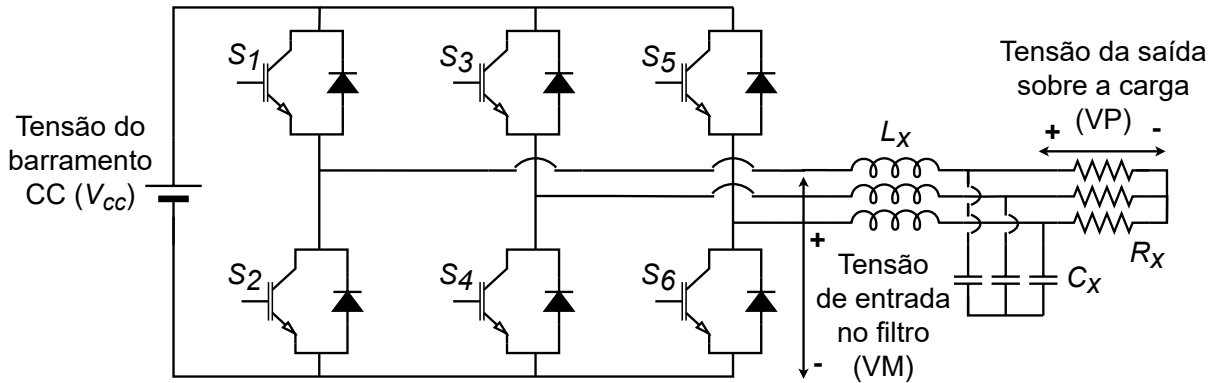


Figura 2 – Descrição das variáveis de controle

Para evitar que o barramento CC seja curto-circuitado, as duas chaves de cada fase (S_1 e S_2 , S_3 e S_4 , S_5 e S_6) devem ser operadas em diferentes estados de comutação (RODRÍGUEZ; ESTAY, 2012). Por exemplo, ao acionar S_1 , S_2 deve ser desativada. Deste modo é possível unir cada par em uma mesma variável de fase, representada por S_x para $x \in \{a; b; c\}$. Essa representação simplificada é definida da seguinte maneira:

$$S_a = \begin{cases} 1 & \text{se } S_1 = 1 \text{ e } S_2 = 0 \\ 0 & \text{se } S_1 = 0 \text{ e } S_2 = 1 \end{cases} \quad (1)$$

$$S_b = \begin{cases} 1 & \text{se } S_3 = 1 \text{ e } S_4 = 0 \\ 0 & \text{se } S_3 = 0 \text{ e } S_4 = 1 \end{cases} \quad (2)$$

$$S_c = \begin{cases} 1 & \text{se } S_5 = 1 \text{ e } S_6 = 0 \\ 0 & \text{se } S_5 = 0 \text{ e } S_6 = 1 \end{cases} \quad (3)$$

onde 1 significa que a chave está fechada, ou seja, está passando corrente, e 0 significa que a chave está aberta.

Além disso, como existe uma defasagem de 120° entre cada uma das fases, o vetor unitário $\mathbf{a} = e^{j2\pi/3} = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$ tipicamente é utilizado para definir o vetor de tensão de entrada no filtro LC da seguinte maneira:

$$\mathbf{V} = \frac{2}{3} (V_{aN} + \mathbf{a}V_{bN} + \mathbf{a}^2V_{cN}), \quad (4)$$

onde V_{aN} , V_{bN} , e V_{cN} são as tensões de fase para cada fase, que são iguais à tensão do barramento CC (V_{CC}) quando as chaves estão fechadas ou 0 caso estejam abertas. Portanto, ao considerar todas as possíveis combinações entre S_a , S_b e S_c , a Tabela 1 pode ser usada como Tabela de pesquisa para encontrar o valor de \mathbf{V} .

Os vetores de tensão de entrada representados na Tabela 1 são diretamente relacionados à representação no sistema de coordenadas $\alpha\beta$, sendo α a parte real e β o valor imaginário de \mathbf{V} . Portanto, os vetores podem ser visualizados no plano complexo, com as partes reais e imaginárias definindo as suas direções, conforme mostrando na Figura 3.

Tabela 1 – Tabela de pesquisa para os vetores de tensão de entrada no filtro *LC*

S_a	S_b	S_c	Vetor de tensão de entrada (\mathbf{V})
0	0	0	$\mathbf{V}_0 = 0$
1	0	0	$\mathbf{V}_1 = \frac{2}{3}V_{cc}$
1	1	0	$\mathbf{V}_2 = \frac{1}{3}V_{cc} + j\frac{\sqrt{3}}{3}V_{cc}$
0	1	0	$\mathbf{V}_3 = -\frac{1}{3}V_{cc} + j\frac{\sqrt{3}}{3}V_{cc}$
0	1	1	$\mathbf{V}_4 = -\frac{2}{3}V_{cc}$
0	0	1	$\mathbf{V}_5 = -\frac{1}{3}V_{cc} - j\frac{\sqrt{3}}{3}V_{cc}$
1	0	1	$\mathbf{V}_6 = \frac{1}{3}V_{cc} - j\frac{\sqrt{3}}{3}V_{cc}$
1	1	1	$\mathbf{V}_7 = 0$

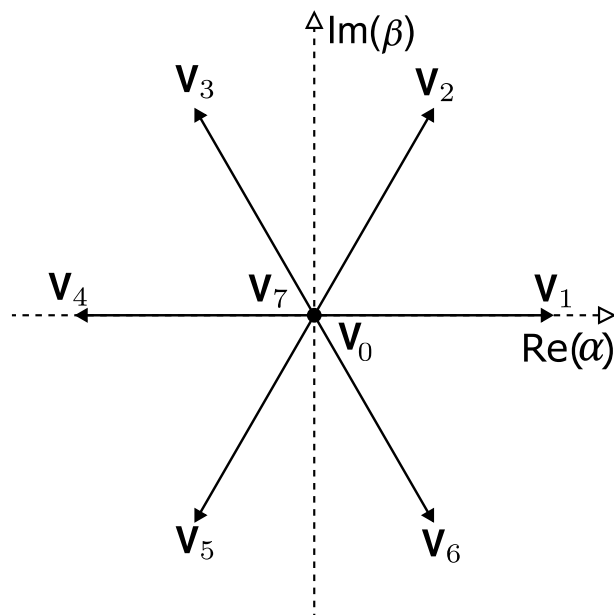


Figura 3 – Vetores de tensão de entrada representados no plano complexo. Adaptado de Rodríguez e Estay (2012).

2.2 MODELAGEM DINÂMICA

Obter um bom modelo dinâmico de um sistema é uma das etapas mais importantes de um projeto de controle. Para técnicas de controle preditivo, a qualidade do controlador costuma estar diretamente relacionada com a qualidade do modelo (RODRÍGUEZ; ESTAY, 2012). Nesta seção, são apresentados os passos necessários para encontrar os modelos de tempo contínuo e de tempo discreto utilizados para o desenvolvimento deste trabalho.

2.2.1 Modelo de tempo contínuo

O primeiro passo para obter o modelo dinâmico contínuo do sistema, ilustrado na Figura 2, é encontrar as suas equações dinâmicas. Essas equações podem ser deduzidas com uso da primeira e da segunda Lei de Kirchhoff, dos nós e das malhas, descritas por

$$i_{Lx} = i_{Cx} + i_{Rx}, \quad (5)$$

$$V_{xN} = V_{Lx} + V_{Cx}, \quad (6)$$

respectivamente. Como na Seção 2.1, x representa cada uma das fases do sistema (a , b ou c). Além disso, i_{Lx} , i_{Cx} e i_{Rx} são as correntes consumidas pelo indutor, capacitor e resistor, respectivamente. V_{Lx} e V_{Cx} representam as tensões sobre o indutor e o capacitor, enquanto V_{xN} é a tensão de fase na entrada do filtro.

No ponto de vista de problema de controle, o sinal senoidal trifásico ideal que se deseja rastrear na carga do sistema é definido como a referência. A tensão trifásica sobre as cargas resistivas (V_{Rx}), ou seja, a tensão de saída, é vista como Variável de Processo (VP). Vale ressaltar que, para esse sistema, por C_x ser paralelo a R_x , considera-se que $V_{Rx} = V_{Cx}$. Para atingir dada saída desejada, o vetor de estados das chaves do conversor (S_1 até S_6) é o que será efetivamente manuseado. Entretanto, uma simplificação de modelo pode ser feita ao considerar a tensão de entrada no filtro LC (V_{xN}) como a Variável Manipulada (VM). Desta forma, conforme descrito em Bordons e Montero (2015), a (5) pode ser manipulada, resultando em:

$$i_{Lx} = C_x \dot{V}_{Cx} + \frac{V_{Cx}}{R_x}. \quad (7)$$

De mesmo modo, (6) também é desenvolvida, encontrando a equação:

$$V_{xN} = L_x \dot{i}_{Lx} + V_{Cx}. \quad (8)$$

Ao inserir (7) em (8), encontra-se a equação dinâmica que relaciona a VM e a VP do processo, dada por:

$$V_{xN} = L_x C_x \ddot{V}_{Cx} + \frac{L_x}{R_x} \dot{V}_{Cx} + V_{Cx}. \quad (9)$$

Por fim, aplica-se a transformada de Laplace em (9), o que resulta na seguinte representação do sistema:

$$G_V(s) = \frac{Y(s)}{U(s)} = \frac{1}{L_x C_x s^2 + \frac{L_x}{R_x} s + 1}, \quad (10)$$

onde $U(s)$ é a VM e $Y(s)$ a VP, ambas no domínio da frequência. Esta função de transferência, denominada $G_V(s)$, descreve o comportamento dinâmico da saída V_{Cx} para uma entrada V_{xN} (BORDONS; MONTERO, 2015). Todavia, destaca-se que as variáveis estão sendo tratadas no sistema de coordenadas abc , visto que $x \in \{a, b, c\}$. Caso as variáveis estejam sendo representadas pelo sistema de coordenadas $\alpha\beta$, V_{xN} passa a ser o vetor \mathbf{V} , descrito na Tabela 1. Quanto

à VP, ao aplicar em V_{Cx} a transformada de Clarke, é possível encontrar $V_{C\alpha}$ e $V_{C\beta}$, e também representá-la na forma de vetor complexo, com $\mathbf{V}_{C\alpha\beta} = V_{C\alpha} + jV_{C\beta}$. A função de transferência mostrada em (10) também é válida para essas variáveis no sistema de coordenadas em $\alpha\beta$.

Na literatura, outra representação do sistema pode ser encontrada ao considerar o estado das chaves (S_x) como variável manipulada (GEYER; QUEVEDO, 2014), no lugar da tensão de entrada no filtro LC (V_{xN}). Essa substituição gera uma nova função de transferência, dada por

$$G_S(s) = \frac{V_{CC}}{L_x C_x s^2 + \frac{L_x}{R_x} s + 1}. \quad (11)$$

Nota-se que, neste caso, o comportamento dinâmico do sistema continua sendo o mesmo, porém, com uma mudança no ganho estático, oriunda da relação $V_{xN} = S_x V_{CC}$.

2.2.2 Modelo de tempo discreto

Para que seja possível realizar a predição da saída do sistema em uma plataforma de controle digital, é necessário obter o modelo de tempo discreto da planta. Existem diversos métodos para discretizar um modelo de tempo contínuo. Contudo, o método sustentador de ordem zero (ZOH, do inglês *Zero-Order Hold*) é muito comum em sistemas controlados digitalmente, uma vez que é um método simples e garante que, nos instantes amostrados, o valor do modelo discreto seja igual ao valor do modelo contínuo (ÅSTRÖM; WITTENMARK, 2011). Além disso, como em microcontroladores digitais o sinal da saída e o sinal de controle são mantidos constantes entre os instantes de amostragem, o ZOH é a técnica mais adequada para este tipo de implementação, pois garante que o modelo represente o que efetivamente está aplicado na planta. Portanto, foi o método definido para este trabalho.

A função de transferência apresentada em (10), por ter um par de polos complexos e não possuir zeros, pode ser reescrita na forma padrão de uma Função de Transferência (FT) de segunda ordem subamortecida:

$$G_V(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (12)$$

onde ω_0 é a frequência natural do sistema e ζ é o coeficiente de amortecimento que, por se tratar de uma planta subamortecida, assume $\zeta < 1$.

Para que essa relação ocorra, transforma-se a função descrita em (10) para a forma mônica, obtendo:

$$G_V(s) = \frac{1}{s^2 + \frac{1}{R_x C_x} s + \frac{1}{L_x C_x}}. \quad (13)$$

Com isso, ao igualar (13) com (12), obtêm-se

$$\omega_0 = \frac{1}{\sqrt{L_x C_x}}, \quad (14)$$

$$\zeta = \frac{\sqrt{L_x C_x}}{2R_x C_x}. \quad (15)$$

Com esses termos, é possível encontrar a representação de tempo discreto por ZOH desta planta, seguindo os passos descritos por Åström e Wittenmark (2011, p. 55). Primeiramente, calcula-se

$$\omega = \omega_0 \sqrt{1 - \zeta^2} = \frac{1}{2R_x C_x} \sqrt{\frac{(2R_x C_x)^2 - L_x C_x}{L_x C_x}}, \quad (16)$$

$$\alpha = e^{-\zeta \omega_0 T_s} = e^{-\frac{T_s}{2R_x C_x}}, \quad (17)$$

$$\beta = \cos(\omega T_s) = \cos\left(\frac{T_s}{2R_x C_x} \sqrt{\frac{(2R_x C_x)^2 - L_x C_x}{L_x C_x}}\right), \quad (18)$$

$$\gamma = \text{sen}(\omega T_s) = \text{sen}\left(\frac{T_s}{2R_x C_x} \sqrt{\frac{(2R_x C_x)^2 - L_x C_x}{L_x C_x}}\right), \quad (19)$$

onde T_s é o período de amostragem da planta.

Considerando uma função de transferência de tempo discreto de segunda ordem em sua forma padrão,

$$G(z) = \frac{y(k)}{u(k)} = \frac{B_{\text{tf}}(z^{-1})}{A_{\text{tf}}(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \quad (20)$$

então, conforme Åström e Wittenmark (2011), seus coeficientes são encontrados por

$$b_1 = 1 - \alpha \left(\beta + \frac{\zeta \omega_0}{\omega} \gamma \right),$$

$$b_1 = 1 - e^{-\frac{T_s}{2R_x C_x}} \left(\cos\left(\frac{\sqrt{1 - \frac{L_x}{4C_x R_x^2}} T_s}{\sqrt{L_x C_x}}\right) + \frac{\sqrt{L_x C_x} \sin\left(\frac{\sqrt{1 - \frac{L_x}{4C_x R_x^2}} T_s}{\sqrt{L_x C_x}}\right)}{2C_x R_x \sqrt{1 - \frac{L_x}{4C_x R_x^2}}}\right), \quad (21)$$

$$b_2 = \alpha^2 + \alpha \left(\frac{\zeta \omega_0}{\omega} \gamma - \beta \right),$$

$$b_2 = e^{-\frac{T_s}{R_x C_x}} + e^{-\frac{T_s}{2R_x C_x}} \left(-\cos\left(\frac{\sqrt{1 - \frac{L_x}{4C_x R_x^2}} T_s}{\sqrt{L_x C_x}}\right) + \frac{\sqrt{L_x C_x} \sin\left(\frac{\sqrt{1 - \frac{L_x}{4C_x R_x^2}} T_s}{\sqrt{L_x C_x}}\right)}{2C_x R_x \sqrt{1 - \frac{L_x}{4C_x R_x^2}}}\right), \quad (22)$$

$$a_1 = -2\alpha\beta,$$

$$a_1 = -2e^{-\frac{T_s}{2R_x C_x}} \cos\left(\frac{\sqrt{1 - \frac{L_x}{4C_x R_x^2}} T_s}{\sqrt{L_x C_x}}\right), \quad (23)$$

$$a_2 = \alpha^2, \quad (24)$$

$$a_2 = e^{-\frac{T_s}{R_x C_x}}.$$

Observa-se que, como $G_S(s) = V_{cc}G_V(s)$, $G_S(z)$ possui o mesmo denominador de $G_V(z)$ e os coeficientes do seu numerador podem ser encontrados multiplicando o numerador de $G_V(z)$ por V_{cc} .

2.3 CONTROLE PREDITIVO BASEADO EM MODELO

Desenvolvido no fim dos anos 1970, o MPC é uma estratégia de controle ótimo que calcula a ação de controle por meio da minimização de uma função custo (RODRÍGUEZ; ESTAY, 2012). Essa função custo costuma considerar o erro estimado para as amostras futuras, obtido com emprego do modelo dinâmico discreto do sistema. Nesta seção, são descritas as equações utilizadas para realizar a predição da saída utilizada neste cálculo do erro, assim como as formulações de MPC que foram implementadas nos algoritmos abordados.

2.3.1 Predição da saída do sistema

Com os modelos de tempo discreto obtidos na Subseção 2.2.2, a forma mais direta de realizar a predição da saída do sistema é transformar a função de transferência na forma de equação a diferenças. Partindo da FT discreta na forma padrão descrita em (20), multiplica-se cruzado os polinômios $A_G(z)$ e $B_G(z)$ com a saída $y(k)$ e com o sinal de controle $u(k)$, e isola-se o termo $y(k)$. Os termos descritos por z^{-n} representam um atraso de n amostras e, ao serem multiplicados por $y(k+i)$ e $u(k+i)$, deslocam estes sinais para $y(k+i-n)$ e $u(k+i-n)$, sendo $i \in \mathbb{Z}$ uma amostra qualquer no tempo. Deste modo, a equação a diferenças do sistema discretizado de segunda ordem passa a ser

$$y(k) = b_1u(k-1) + b_2u(k-2) - a_1y(k-1) - a_2y(k-2), \quad (25)$$

e pode ser deslocada no tempo para realizar o cálculo das predições das saídas, resultando em:

$$\begin{aligned} \hat{y}(k+1) &= b_1\hat{u}(k) + b_2u(k-1) - a_1y(k) - a_2y(k-1), \\ \hat{y}(k+2) &= b_1\hat{u}(k+1) + b_2\hat{u}(k) - a_1\hat{y}(k+1) - a_2y(k), \\ &\vdots \\ \hat{y}(k+j) &= b_1\hat{u}(k+j-1) + b_2\hat{u}(k+j-2) - a_1\hat{y}(k+j-1) - a_2\hat{y}(k+j-2), \end{aligned} \quad (26)$$

onde $j \in \mathbb{Z}$ é a amostra mais distante a ser estimada. Normalmente, os limites de predição são definidos pelo horizonte de predição (N), delimitado em função de variáveis auxiliares:

$$N = N_2 - N_1 + 1, \quad (27)$$

para N_1 e N_2 correspondentes às amostras inicial e final do horizonte de predição, respectivamente. Os sinais de controle também costumam ser ponderados até uma amostra específica e, a partir dela, o sinal pode passar a ser constante ou nulo, dependendo do modelo e da estratégia utilizados. O tamanho desse horizonte (N_u) também é definido por suas variáveis auxiliares, N_{u1} e N_{u2} .

Através de (26) é possível visualizar que, para uma planta de segunda ordem, a predição em $k+j$ depende das duas amostras anteriores, $k+j-1$ e $k+j-2$. Ou seja, para que a predição do instante $k+j$ possa ser feita, é necessário conhecer os valores associados a $y(k+j-1)$ e $y(k+j-2)$. Note que, para $j > 2$, as saídas nas amostras $k+j-1$ e $k+j-2$ serão apenas predições ($\hat{y}(k+j-1)$ e $\hat{y}(k+j-2)$), pois ainda não aconteceram. Para $j \leq 2$, alguns dos valores utilizados nos cálculos envolvem saídas que foram efetivamente medidas no sistema, como $y(k)$ até $y(k-2)$, pois estas aconteceram até o momento de início da amostra atual. Portanto, para que a predição de uma saída $\hat{y}(k+j)$ possa ser feita por essa abordagem de substituição recursiva, as predições anteriores entre k e $k+j$ já precisam ter sido calculadas. Os sinais de controle são considerados como predições a partir do instante k , pois a decisão do sinal de controle é feita só ao fim da amostra, o que significa que no momento da predição $\hat{u}(k)$ ainda não foi definido.

Vale ressaltar que por mais que o modelo possa ser considerado como um sistema de duas entradas e saídas distintas em α e em β , o seu cálculo de predição pode ser feito pela mesma equação a diferenças, ao considerar $u(k+j-n)$ e $y(k+j-n)$ variáveis complexas, com a componente real armazenando o seu valor no eixo α e a parte imaginária representando o valor sobre o eixo β .

Outro tipo de modelo muito utilizado em técnicas de MPC com conjunto contínuo de controle é o modelo autorregressivo integrado com média móvel e entrada controlada (CARIMA, do inglês *Controlled Auto-Regressive Integrated Moving Average*). A equação de um sistema de entrada única e saída única (SISO, do inglês *Single Input Single Output*) representado pelo modelo CARIMA pode ser retratada como:

$$A(z^{-1})y(k) = B(z^{-1})z^{-d}u(k-1) + C(z^{-1})\frac{\xi(k)}{\Delta}, \quad (28)$$

onde os polinômios $A(z^{-1})$ e $B(z^{-1})$ são obtidos através do denominador e do numerador da função de transferência discreta do modelo, no mesmo formato apresentado em (20), e

$$\Delta = 1 - z^{-1}. \quad (29)$$

É importante destacar que enquanto o sinal de controle do modelo CARIMA está uma amostra atrasado ($u(k-1)$), a função de transferência discreta, descrita em (20), utiliza um polinômio baseado na amostra atual ($u(k)$) e, portanto, o numerador da função de transferência deve ser multiplicado pelo operador z para chegar no valor do polinômio $B(z^{-1})$ do modelo CARIMA. O polinômio $C(z^{-1})$ tem como objetivo modelar o comportamento das perturbações não medidas do processo, mas uma prática comum é defini-lo como $C(z^{-1}) = 1$ por conta da dificuldade em obter esses dados no processo (NORMEY-RICO *et al.*, 2024). A saída do sistema é representada por $y(k)$ e um ruído branco de média nula é simbolizado por $\xi(k)$. Por fim, um atraso de transporte e um polinômio para a modelagem de perturbações não estacionárias são representados por d e Δ , respectivamente. Como Δ representa a derivada discreta, o integrador do modelo CARIMA é o resultado do termo $\frac{1}{\Delta}$, que integra o termo do ruído branco de média nula.

Partindo de (28), multiplica-se por Δ os dois lados da equação (NORMEY-RICO *et al.*, 2024), resultando em

$$A(z^{-1})\Delta y(k) = B(z^{-1})z^{-d}\Delta u(k-1) + C(z^{-1})\xi(k). \quad (30)$$

Então, considerando $C(z^{-1}) = 1$, $A(z^{-1}) = A_{\text{tf}}(z^{-1})$ e $B(z^{-1}) = zB_{\text{tf}}(z^{-1})$ para $A_{\text{tf}}(z^{-1})$ e $B_{\text{tf}}(z^{-1})$ obtidos por (20), e Δ dado em (29), (30) pode ser reescrita como

$$(1 + a_1z^{-1} + a_2z^{-2})(1 - z^{-1})y(k) = (b_1 + b_2z^{-1})z^{-d}\Delta u(k-1) + \xi(k), \quad (31)$$

o que leva a

$$y(k) = (1 - a_1)y(k-1) + (a_1 - a_2)y(k-2) + a_2y(k-3) + b_1z^{-d}\Delta u(k-1) + b_2z^{-d}\Delta u(k-2) + \xi(k), \quad (32)$$

onde Δu é o incremento de controle que passa a ser considerado como variável manipulada. A partir de (32), a predição da saída no instante $k+1$ pode ser obtida deslocando o modelo um instante de tempo para frente (NORMEY-RICO *et al.*, 2024), chegando em

$$y(k+1) = (1 - a_1)y(k) + (a_1 - a_2)y(k-1) + a_2y(k-2) + b_1z^{-d}\Delta u(k) + b_2z^{-d}\Delta u(k-1) + \xi(k+1). \quad (33)$$

Nesta equação, caso $d=0$, o valor futuro da saída ainda desconhecida ($y(k+1)$) depende dos seus próprios valores passados já conhecidos, do incremento de controle definido na amostra anterior ($\Delta u(k-1)$), do incremento de controle que será definido na amostra atual ($\Delta u(k)$) e do valor futuro do ruído branco de média nula $\xi(k+1)$, que também é desconhecido. Para definir esses valores futuros desconhecidos, a esperança matemática ($E[\cdot]$) é a melhor estimativa (NORMEY-RICO *et al.*, 2024) e, portanto, é aplicada sobre (33), conforme

$$E[y(k+1)] = E[(1 - a_1)y(k) + (a_1 - a_2)y(k-1) + a_2y(k-2) + b_1z^{-d}\Delta u(k) + b_2z^{-d}\Delta u(k-1)] + E[\xi(k+1)]. \quad (34)$$

Para os termos medidos ou conhecidos, a esperança matemática do seu valor é o seu próprio valor assumido. O incremento de controle na amostra atual $\Delta u(k)$, por mais que não seja conhecido no início da amostra, é calculado na lei de controle e não precisa ser estimado. Para a saída futura no instante $k+1$, define-se que $E[y(k+1)] = \hat{y}(k+1|k)$, ou seja, é o valor predito da saída para o instante $k+1$, calculado na amostra k . Quanto ao ruído, por possuir média nula, a sua esperança matemática é zero e o termo $\xi(k+1)$ pode ser retirado de (33). Com estas definições, encontra-se

$$\hat{y}(k+1|k) = (1 - a_1)y(k) + (a_1 - a_2)y(k-1) + a_2y(k-2) + b_1z^{-d}\Delta u(k) + b_2z^{-d}\Delta u(k-1). \quad (35)$$

Para calcular $y(k+2)$, desloca-se o modelo descrito em (32) duas amostras para frente, chegando em

$$y(k+2) = (1 - a_1)y(k+1) + (a_1 - a_2)y(k) + a_2y(k-1) + b_1z^{-d}\Delta u(k+1) + b_2z^{-d}\Delta u(k) + \xi(k+2). \quad (36)$$

Seguindo os conceitos de esperança matemática definidos anteriormente, encontra-se

$$\hat{y}(k+2|k) = (1-a_1)\hat{y}(k+1|k) + (a_1-a_2)y(k) + a_2y(k-1) + b_1z^{-d}\Delta u(k+1) + b_2z^{-d}\Delta u(k). \quad (37)$$

Substituindo $\hat{y}(k+1|k)$ pela expressão equivalente descrita em (35), obtém-se

$$\begin{aligned} \hat{y}(k+2|k) = & (1-a_1) \left[(1-a_1)y(k) + (a_1-a_2)y(k-1) + a_2y(k-2) + b_1z^{-d}\Delta u(k) + b_2z^{-d}\Delta u(k-1) \right] \\ & + (a_1-a_2)y(k) + a_2y(k-1) + b_1z^{-d}\Delta u(k+1) + b_2z^{-d}\Delta u(k), \end{aligned} \quad (38)$$

que pode ser organizada e reescrita por

$$\begin{aligned} \hat{y}(k+2|k) = & \left[(1-a_1)^2 + (a_1-a_2) \right] y(k) + \left[(1-a_1)(a_1-a_2) + a_2 \right] y(k-1) + (1-a_1)a_2y(k-2) \\ & + b_1z^{-d}\Delta u(k+1) + \left[(1-a_1)b_1 + b_2 \right] z^{-d}\Delta u(k) + (1-a_1)b_2z^{-d}\Delta u(k-1). \end{aligned} \quad (39)$$

Para qualquer outra amostra futura para a qual se deseja obter a saída predita, o mesmo processo pode ser executado. Essas equações de predição podem ser agrupadas para serem representadas de forma matricial, resultando em

$$\hat{\mathbf{y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{f}, \quad (40)$$

onde, $\hat{\mathbf{y}} \in \mathbb{R}^N$, é um vetor coluna, que armazena o valor das possíveis saídas preditas dentro de um horizonte de predição definido por N . O termo $\mathbf{G}\Delta\mathbf{u}$ está relacionado à resposta forçada do sistema, sendo que $\mathbf{G} \in \mathbb{R}^{N \times N_u}$ é a matriz que representa a resposta ao degrau do sistema e seu cômputo pode ser feito off-line, por meio de simulações ou pela separação das parcelas do modelo de predição. Esta parte da resposta depende do incremento de controle ($\Delta\mathbf{u} \in \mathbb{R}^{N_u}$) que ainda não foi implementado. Por outro lado, $\mathbf{f} \in \mathbb{R}^N$ é o vetor coluna que representa a resposta livre, e pode ser definido como

$$\mathbf{f} = \mathbf{F}\underline{\mathbf{y}} + \mathbf{H}\underline{\Delta\mathbf{u}}. \quad (41)$$

Nota-se que este vetor corresponde à parcela da resposta que depende dos sinais de controle definidos em amostras anteriores ($\underline{\Delta\mathbf{u}}$) e das saídas do sistema medidas até o instante de tempo atual ($\underline{\mathbf{y}}$). As matrizes \mathbf{H} e \mathbf{F} armazenam os coeficientes que multiplicam estes termos. Reorganizando (41), a equação do vetor \mathbf{f} pode ser reescrita por

$$\mathbf{f} = \begin{bmatrix} \mathbf{F} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{y}} & \underline{\Delta\mathbf{u}} \end{bmatrix}^T = \mathbf{f}_{\text{aux}} \begin{bmatrix} \underline{\mathbf{y}} & \underline{\Delta\mathbf{u}} \end{bmatrix}^T. \quad (42)$$

Deste modo, constata-se que o vetor \mathbf{f} tem parte calculada off-line ($\mathbf{f}_{\text{aux}} \in \mathbb{R}^N$), mas como depende de medições e sinais de controle antigos, tem que ser atualizado no início de cada período de amostragem.

As matrizes utilizadas nessa representação matricial podem ser definidas recursivamente, seguindo o que foi descrito anteriormente pelas equações (35) e (39), ou ainda por uma equação diofantina. A estrutura dessas matrizes e os passos para executar a equação diofantina podem ser encontrados em Normey-Rico *et al.* (2024).

Para um sistema MIMO com n_ε entradas e n_φ saídas, uma matriz de transferência $\mathbf{M}_t \in \mathbb{R}^{n_\varphi \times n_\varepsilon}$ é utilizada para descrever o comportamento dinâmico entre essas entradas e saídas:

$$\mathbf{M}_t(z) = \begin{bmatrix} \frac{N_{1,1}(z^{-1})}{D_{1,1}(z^{-1})} z^{-d_{1,1}} & \cdots & \frac{N_{1,n_\varepsilon}(z^{-1})}{D_{1,n_\varepsilon}(z^{-1})} z^{-d_{1,n_\varepsilon}} \\ \vdots & \ddots & \vdots \\ \frac{N_{n_\varphi,1}(z^{-1})}{D_{n_\varphi,1}(z^{-1})} z^{-d_{n_\varphi,1}} & \cdots & \frac{N_{n_\varphi,n_\varepsilon}(z^{-1})}{D_{n_\varphi,n_\varepsilon}(z^{-1})} z^{-d_{n_\varphi,n_\varepsilon}} \end{bmatrix}, \quad (43)$$

onde $N_{\varphi\varepsilon}(z^{-1})$, $D_{\varphi\varepsilon}(z^{-1})$ e $d_{\varphi\varepsilon}$ representam o numerador, denominador e atraso de transporte da função de transferência que relaciona a saída φ com a entrada ε , respectivamente. Decompondo \mathbf{M}_t em uma representação de descrição matricial fracionária (MFD, do inglês *Matrix Fraction Description*), obtêm-se as matrizes polinomiais utilizadas na generalização do modelo CARIMA para um sistema MIMO (PECCIN *et al.*, 2020), visualizadas em:

$$\mathbf{A}(z^{-1})y(k) = \mathbf{B}(z^{-1})\mathbf{D}(z^{-1})u(k-1) + \mathbf{C}(z^{-1})\frac{e(k)}{\Delta}. \quad (44)$$

Neste novo formato do modelo CARIMA, semelhante ao apresentado para o modelo SISO em (28), $\mathbf{A}(z^{-1}) \in \mathbb{R}^{n_\varphi \times n_\varphi}$ passa a ser uma matriz polinomial diagonal, e seus elementos da diagonal principal ($\varphi\varepsilon : \varphi = \varepsilon$) são os polinômios resultantes de um cálculo de mínimo múltiplo comum entre os denominadores da linha φ de \mathbf{M}_t . A matriz polinomial $\mathbf{B}(z^{-1}) \in \mathbb{R}^{n_\varphi \times n_\varepsilon}$ pode ser obtida por $\mathbf{B}(z^{-1}) = \mathbf{A}(z^{-1})\mathbf{M}_t$ e a matriz $\mathbf{D}(z^{-1}) \in \mathbb{R}^{n_\varphi \times n_\varphi}$ é diagonal com seus elementos representando o menor atraso de transporte entre as entradas com a saída da linha φ de \mathbf{M}_t .

Mesmo com essa alteração no modelo para o caso MIMO, a predição do sistema é feita de forma semelhante ao que foi mostrado em (40), como pode ser visto por

$$\hat{\mathbf{y}}_m = \mathbf{G}_m \Delta \mathbf{u}_m + \mathbf{f}_m, \quad (45)$$

onde a matriz \mathbf{G}_m é formada por matrizes $\mathbf{G}_{\varphi\varepsilon} \in \mathbb{R}^{N_\varphi \times N_{u\varepsilon}}$, sendo N_φ e $N_{u\varepsilon}$ os horizontes de predição das saídas e de controle das entradas, respectivamente. Assim como no caso SISO, essas matrizes representam a resposta da saída φ ao degrau na entrada ε . Deste modo, a matriz \mathbf{G}_m é organizada da seguinte maneira

$$\mathbf{G}_m = \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \mathbf{G}_{1,n_\varepsilon} \\ \vdots & \ddots & \vdots \\ \mathbf{G}_{n_\varphi,1} & \cdots & \mathbf{G}_{n_\varphi,n_\varepsilon} \end{bmatrix}. \quad (46)$$

A resposta livre ($\mathbf{f}_m \in \mathbb{R}^{\sum_{\varphi=1}^{n_\varphi} N_\varphi}$), a saída predita do sistema ($\hat{\mathbf{y}}_m \in \mathbb{R}^{\sum_{\varphi=1}^{n_\varphi} N_\varphi}$) e o incremento de controle ($\Delta \mathbf{u}_m \in \mathbb{R}^{\sum_{\varepsilon=1}^{n_\varepsilon} N_{u\varepsilon}}$) persistem com o formato de vetor coluna, pois os vetores obtidos

para cada entrada ou saída do sistema são sobrepostos da seguinte forma:

$$\mathbf{f}_m = \begin{bmatrix} \mathbf{f}_1^T & \dots & \mathbf{f}_{n_\varphi}^T \end{bmatrix}^T, \quad (47)$$

$$\hat{\mathbf{y}}_m = \begin{bmatrix} \hat{\mathbf{y}}_1^T & \dots & \hat{\mathbf{y}}_{n_\varphi}^T \end{bmatrix}^T, \quad (48)$$

$$\Delta \mathbf{u}_m = \begin{bmatrix} \Delta u_1^T & \dots & \Delta u_{n_\varepsilon}^T \end{bmatrix}^T. \quad (49)$$

Por $\Delta \mathbf{u}_m$ estar organizado desta forma, ao multiplicar $\mathbf{G}_m \Delta \mathbf{u}_m$, cada matriz de coeficientes de resposta ao degrau ($\mathbf{G}_{\varphi\varepsilon}$) é correlacionada com o seu respectivo vetor de incrementos de controle ($\Delta \mathbf{u}_\varepsilon$) e, quando somadas, resultam na parcela de resposta forçada da saída predita correspondente ($\hat{\mathbf{y}}_\varphi$).

Pelo princípio do modelo interno, o integrador do modelo CARIMA o torna adequado para sinais de referência e perturbações do tipo degrau, isto é, sinais que são constantes sobre uma boa faixa de tempo, anulando o erro em regime permanente (NORMEY-RICO *et al.*, 2024). Portanto, ao trabalhar com o sistema de coordenadas em abc ou $\alpha\beta$, que possuem referências senoidais, as qualidades deste modelo podem não ser bem exploradas. Na literatura, os principais caminhos abordados para contornar este problema são: inserir um polo ressonante com a frequência do sistema (CORDERO *et al.*, 2022), trabalhar com um sistema de coordenadas que rotacione na mesma frequência e no mesmo sentido que a referência, para que os valores em regime permanente sejam vistos como sinais do tipo degrau (ZOU *et al.*, 2014) ou descartar o integrador do modelo e trabalhar com um modelo mais simples, que não tenha garantia de erro nulo em regime permanente (SONG; LIU, 2006). Neste trabalho, por conta do objetivo ser a redução do tempo de cômputo, a alternativa mais simples foi escolhida para ser avaliada e comparada com o modelo CARIMA. Esse modelo é denominado de autorregressivo com média móvel e entrada controlada (CARMA, do inglês *Controlled Auto-Regressive Moving Average*). Como o próprio nome especifica, a principal diferença é a ausência do integrador, o que significa que o termo $\frac{1}{\Delta}$ não está presente no modelo junto ao ruído, como pode ser visto na sua equação para o caso SISO:

$$A(z^{-1})y(k) = B(z^{-1})z^{-d}u(k-1) + C(z^{-1})e(k). \quad (50)$$

Para obter as predições da saída do sistema, os mesmos procedimentos aplicados ao modelo CARIMA devem ser executados neste caso. Todavia, diferente do que foi feito para o modelo CARIMA em (30), por não possuir o integrador junto ao erro, a multiplicação por Δ nos dois lados da equação não é realizada com o modelo CARMA. Deste modo, a partir da função de transferência discreta apresentada em (20) e fazendo as mesmas considerações descritas para o modelo CARIMA, encontra-se a equação da saída em k com o modelo CARMA

$$y(k) = -a_1y(k-1) - a_2y(k-2) + b_1z^{-d}u(k-1) + b_2z^{-d}u(k-2) + \xi(k), \quad (51)$$

que, ao ser deslocada no tempo, pode ser utilizada para encontrar as saídas preditas do sistema, como foi feito para o modelo CARIMA. Nota-se que neste modelo, a variável manipulada passa

a ser o sinal de controle em sua forma integral (u) e não apenas o seu incremento (Δu). Deste modo, existe uma clara semelhança entre o modelo CARMA e a equação a diferenças. A principal diferença entre esses modelos é a amostra considerada na variável manipulada e o ruído branco de média nula considerados por CARMA. Todavia, a diferença de amostras é compensada pelo grau do polinômio e o ruído é desconsiderado pelo modelo para o cálculo da predição, o que faz com que ambas as formas alcancem o mesmo resultado.

Para CARMA, a forma agrupada das equações de predição pode ser definida como

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}. \quad (52)$$

O vetor \mathbf{u} tem a mesma dimensão do vetor $\Delta\mathbf{u}$, definido para o CARIMA. A matriz \mathbf{G} e os vetores $\hat{\mathbf{y}}$ e \mathbf{f} também possuem as mesmas dimensões, mas valores diferentes por não incluírem o polinômio Δ em seu cálculo.

Para o caso MIMO, novamente todas as estruturas definidas para o modelo CARIMA podem ser reaproveitadas no modelo CARMA. Deste modo, a equação de predição da saída para o caso MIMO no modelo CARMA é definida por

$$\hat{\mathbf{y}}_m = \mathbf{G}_m\mathbf{u}_m + \mathbf{f}_m, \quad (53)$$

onde as dimensões dos vetores coluna $\hat{\mathbf{y}}_m$, \mathbf{G}_m e \mathbf{f}_m são iguais às que foram definidas em (45), assim como $\mathbf{u}_m \in \mathbb{R}^{\sum_{\varepsilon=1}^{n_\varepsilon} N_{u\varepsilon}}$, definido no seguinte formato:

$$\mathbf{u}_m = \begin{bmatrix} u_1^T & \dots & u_{n_\varepsilon}^T \end{bmatrix}^T. \quad (54)$$

2.3.2 MPC com conjunto finito de controle

Entre as abordagens de MPC para inversores trifásicos, as principais estão vinculadas à formulação FCS-MPC. Essa metodologia explora a natureza discreta do problema com o uso de variáveis binárias, relacionadas aos estados das chaves, que estão inclusas no algoritmo (MCNABB; WANG, 2023). Por conta disso, não precisa de modulador para aplicar o sinal de controle no conversor controlado, o que torna a técnica relativamente simples de ser implementada (BLAABJERG, 2021). Além disso, assim como outros algoritmos de MPC, pode adaptar a sua função custo conforme as necessidades do sistema controlado (RODRÍGUEZ; ESTAY, 2012).

Conforme mostrado na Seção 2.1, um VSI trifásico de dois níveis possui 8 (2^3) vetores disponíveis para o controlador em cada amostra. Para escolher a sequência ótima de vetores, um FCS-MPC clássico calcula o custo associado a cada sequência de entrada possível, utilizando a função custo definida para o projeto e, então, escolhe a opção com o menor custo associado. No caso convencional, a função custo pode ser representada por:

$$g_{\alpha\beta} = \lambda_{e,\alpha} \sum_{i=N_1}^{N_2} \hat{e}_\alpha^2(k+i) + \lambda_{e,\beta} \sum_{i=N_1}^{N_2} \hat{e}_\beta^2(k+i), \quad (55)$$

utilizando o sistema de coordenadas $\alpha\beta$ através da transformada de Clarke, ou ainda por:

$$g_{dq} = \lambda_{e,d} \sum_{i=N_1}^{N_2} \hat{e}_d^2(k+i) + \lambda_{e,q} \sum_{i=N_1}^{N_2} \hat{e}_q^2(k+i), \quad (56)$$

no sistema de eixo direto e em quadratura (dq), que utiliza a transformada de Park. Para ambos os casos, k é a amostra atual, $\lambda_{e,i}$ para $i \in \{\alpha; \beta; d; q\}$ são os pesos utilizados para sintonizar o controlador e \hat{e}_i para $i \in \{\alpha; \beta; d; q\}$ são os erros entre as previsões de VP no eixo i e as referências do eixo correspondente. Em casos de formulações mais genéricas, poderiam ser considerados pesos que variam no decorrer dos horizontes, mas isso aumentaria muito o número de parâmetros de sintonização, o que tornaria o projeto do controlador extremamente difícil (CASEIRO; MENDES; CRUZ, 2019). Além disso, para o caso convencional, é possível fixar um termo do tipo $\lambda_{e,i}$ em 1, reduzindo, assim, o número de pesos presente nas funções custo. Isso só pode ser feito porque a proporção entre os fatores é o que afeta a solução ótima da função custo. Com essa alteração, (55) e (56) podem ser simplificadas para:

$$g_{\alpha\beta} = \lambda_e \sum_{i=N_1}^{N_2} \hat{e}_\alpha^2(k+i) + \sum_{i=N_1}^{N_2} \hat{e}_\beta^2(k+i), \quad (57)$$

e

$$g_{dq} = \lambda_e \sum_{i=N_1}^{N_2} \hat{e}_d^2(k+i) + \sum_{i=N_1}^{N_2} \hat{e}_q^2(k+i), \quad (58)$$

respectivamente. Para obter os termos de erros futuros nas funções custo, os modelos de previsão mostrados na Subseção 2.3.1 são utilizados.

Como N_1 define a primeira amostra futura do cálculo de previsão, esse valor pode ser alterado para desconsiderar características indesejadas. Normalmente, em aplicações práticas, $N_1 = 2$, devido ao atraso adicionado pelo cálculo do processo de controle (CORTES *et al.*, 2012). Por isso, neste trabalho, a estrutura do algoritmo ilustrada na Figura 4(a) foi adotada para ser implementada em diferentes comprimentos de horizontes. O esquema mostrado na Figura 4(b) ilustra os vetores de tensão candidatos para serem avaliados pelo FCS-MPC, entretanto, em vez de representar os 8 vetores em $k+1$ e os 64 vetores em $k+2$, apenas 3 e 9 vetores de tensão são usados para facilitar a visualização. Em um caso real, todos os vetores possíveis são avaliados e apenas a sequência que minimiza a função custo (sequência ótima de controle) é utilizada na implementação. Vale ressaltar que, independente do comprimento de N e N_u , apenas a ação de controle calculada para N_{u1} será efetivamente aplicada no inversor, pois em $k+1$ todo o processo é repetido com as novas medições realizadas.

Como N_u define em quantas amostras serão considerados novos vetores e, para um VSI trifásico de dois níveis, são 8 vetores por amostra, o número de possíveis sequências de controle considerados por um controlador FCS-MPC convencional é 8^{N_u} . Deste modo, devido ao custo computacional, N e N_u frequentemente assumem um comprimento de uma única amostra (BLAABJERG, 2021). O Capítulo 3 descreve algumas propostas encontradas na literatura para a redução deste custo computacional.

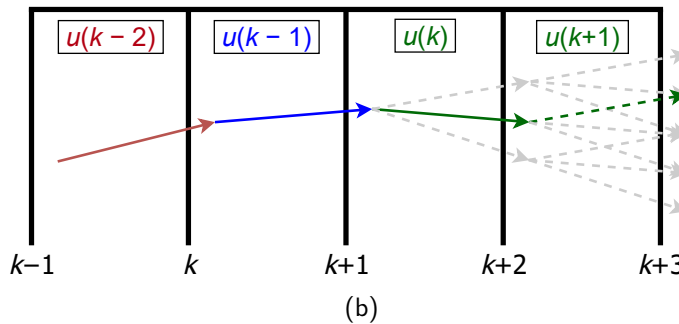
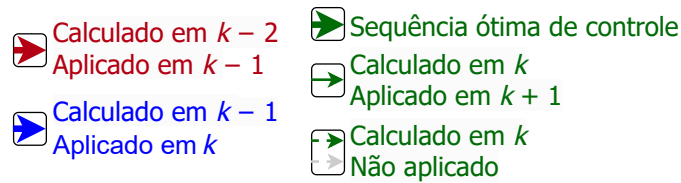
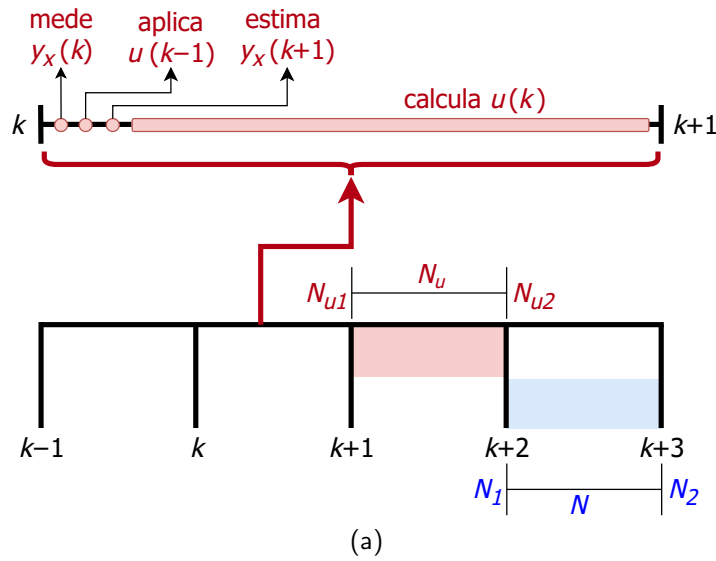


Figura 4 – Implementação do FCS-MPC: (a) estrutura do algoritmo para compensação de atraso, com N e N_u de duas amostras; (b) exemplo de vetores de tensão e seus instantes de tempo correspondentes (apenas três opções de controle são consideradas em cada ramo por uma questão de representação).

2.3.3 Controle preditivo generalizado

O controle preditivo generalizado foi proposto em 1987 e, desde então, se tornou um dos métodos mais utilizados na academia e na indústria (NORMEY-RICO *et al.*, 2024). Assim como foi demonstrado na Subseção 2.3.2, a formulação GPC também parte do princípio de encontrar o sinal de controle ótimo para o sistema através de um problema de minimização de uma determinada função custo, mas, por se tratar de uma formulação de conjunto de controle contínuo, suas infinitas possibilidades fazem com que a solução não seja restrita apenas às opções do inversor.

A função custo tradicional considerada pelo GPC para o caso MIMO é dada por:

$$g = \sum_{\varphi=1}^{n_{\varphi}} \sum_{j=N_{1\varphi}}^{N_{2\varphi}} \lambda_{\varphi} [\hat{y}_{\varphi}(k+j|k) - w_{\varphi}(k+j)]^2 + \sum_{\varepsilon=1}^{n_{\varepsilon}} \sum_{j=N_{u1\varepsilon}}^{N_{u2\varepsilon}} \delta_{\varepsilon} [\Delta u_{\varepsilon}(k+j-1)]^2, \quad (59)$$

onde o erro é representado pela subtração entre a saída φ predita no instante de tempo discreto futuro $k+j$, representada por $\hat{y}_{\varphi}(k+j|k)$, e a trajetória de referência no mesmo instante de tempo para a saída φ , representada por $w_{\varphi}(k+j)$. Além disso, λ_{φ} corresponde à ponderação do erro para a saída φ . O incremento de controle para a entrada ε é representado por Δu_{ε} e é acompanhado pela variável δ_{ε} , que pondera a variação do sinal de controle para essa mesma entrada. Para o modelo de predição CARIMA, modelo utilizado pelo GPC, Δu é a variável manipulada do sistema e nenhuma consideração deve ser feita para este termo. Entretanto, neste trabalho uma versão equivalente do GPC baseada no modelo de predição CARMA denominada como CARMA-GPC também é utilizada. Deste modo, para o controlador CARMA-GPC, como u é a variável manipulada, $\Delta u_{\varepsilon}(k)$ pode ser encontrado por

$$\Delta u_{\varepsilon}(k) = u_{\varepsilon}(k) - u_{\varepsilon}(k-1). \quad (60)$$

As variáveis $N_{1\varphi}$, $N_{2\varphi}$, $N_{u1\varepsilon}$ e $N_{u2\varepsilon}$ correspondem aos delimitadores que demarcam o início e o fim dos horizontes de predição e de controle utilizados no projeto, respectivamente. Esses valores podem ser diferentes para cada entrada ε e saída φ .

Seguindo os passos descritos em Normey-Rico *et al.* (2024), é possível encontrar a forma matricial dessa função para o modelo CARIMA, descrita por:

$$g_y = [\hat{\mathbf{y}} - \mathbf{w}]^T \mathbf{Q}_{\lambda} [\hat{\mathbf{y}} - \mathbf{w}] + \Delta \mathbf{u}^T \mathbf{Q}_{\delta} \Delta \mathbf{u}, \quad (61)$$

sendo que $\hat{\mathbf{y}} \in \mathbb{R}^{n_{\varphi}N}$ e $\Delta \mathbf{u} \in \mathbb{R}^{n_{\varepsilon}N_u}$ permanecem com as mesmas dimensões descritas na Subseção 2.3.1. A referência ($\mathbf{w} \in \mathbb{R}^{n_{\varphi}N}$) passa a ser um vetor coluna que é organizado da mesma forma que $\hat{\mathbf{y}}$ foi descrito em (48). Por fim, $\mathbf{Q}_{\lambda}^{(\varphi)} \in \mathbb{R}^{n_{\varphi} \times n_{\varphi}}$ e $\mathbf{Q}_{\delta}^{(\varepsilon)} \in \mathbb{R}^{n_{\varepsilon} \times n_{\varepsilon}}$ são matrizes bloco diagonais positivas definidas que armazenam as ponderações $\lambda^{(\varphi)}$ e $\delta^{(\varepsilon)}$.

A partir da relação estabelecida em (60), o vetor $\Delta \mathbf{u}$ para o modelo CARMA pode ser definido por

$$\Delta \mathbf{u} = \mathbf{S} \begin{bmatrix} u_1(k+N_{u1}) & \dots & u_{n_{\varepsilon}}(k+N_{u1}) \\ \vdots & \vdots & \vdots \\ u_1(k+N_{u2}) & \dots & u_{n_{\varepsilon}}(k+N_{u2}) \end{bmatrix} - \mathbf{E} \mathbf{u}(k-1), \quad (62)$$

onde

$$\mathbf{S} \in \mathbb{R}^{N_u \times N_u} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix}, \quad (63)$$

$$\mathbf{E} \in \mathbb{R}^{N_u} = [1 \ 0 \ \dots \ 0]^T, \quad (64)$$

$$\mathbf{u}(k-1) \in \mathbb{R}^{1 \times n_{\varepsilon}} = [u_1(k-1) \ \dots \ u_{n_{\varepsilon}}(k-1)]. \quad (65)$$

Portanto, a forma matricial da função custo para o modelo CARMA pode ser encontrada substituindo (62) em (61)

$$g_r = [\hat{\mathbf{y}} - \mathbf{w}]^T \mathbf{Q}_\lambda [\hat{\mathbf{y}} - \mathbf{w}] + [\mathbf{S}\mathbf{u} - \mathbf{E}\mathbf{u}(k-1)]^T \mathbf{Q}_\delta [\mathbf{S}\mathbf{u} - \mathbf{E}\mathbf{u}(k-1)]. \quad (66)$$

Vale ressaltar que para o modelo CARIMA, em (61), o vetor $\hat{\mathbf{y}}$ é encontrado com base no vetor do incremento de controle ($\Delta\mathbf{u}$) pela equação apresentada em (40). Para o modelo CARMA, o vetor $\hat{\mathbf{y}}$ em (66) é definido por (52) e é calculado utilizando o vetor com sinais de controle (\mathbf{u}).

Posteriormente, essas funções custos descritas em (61) e (66) podem ser reescritas nos formatos

$$g_y = \frac{1}{2} \Delta\mathbf{u}^T \mathbf{H}_y \Delta\mathbf{u} + \mathbf{b}_y^T \Delta\mathbf{u} + f_{0,I}, \quad (67)$$

$$g_r = \frac{1}{2} \mathbf{u}^T \mathbf{H}_r \mathbf{u} + \mathbf{b}_r^T \mathbf{u} + f_{0,M}, \quad (68)$$

onde

$$\mathbf{H}_y = 2 \left(\mathbf{G}_m^T \mathbf{Q}_\lambda \mathbf{G}_m + \mathbf{Q}_\delta \right), \quad (69)$$

$$\mathbf{H}_r = 2 \left(\mathbf{G}_m^T \mathbf{Q}_\lambda \mathbf{G}_m + \mathbf{S}^T \mathbf{Q}_\delta \mathbf{S} \right), \quad (70)$$

$$\mathbf{b}_y^T = -2 \mathbf{G}_m^T \mathbf{Q}_\lambda (\mathbf{w} - \mathbf{f}_m), \quad (71)$$

$$\mathbf{b}_r^T = -2 \left[\mathbf{G}_m^T \mathbf{Q}_\lambda (\mathbf{w} - \mathbf{f}_m) + \mathbf{S}^T \mathbf{Q}_\delta \mathbf{E}\mathbf{u}(k-1) \right], \quad (72)$$

$$f_{0,I} = (\mathbf{f}_m - \mathbf{w})^T (\mathbf{f}_m - \mathbf{w}), \quad (73)$$

$$f_{0,M} = (\mathbf{f}_m - \mathbf{w})^T (\mathbf{f}_m - \mathbf{w}) + (\mathbf{E}\mathbf{u}(k-1))^T \mathbf{Q}_\delta (\mathbf{E}\mathbf{u}(k-1)). \quad (74)$$

A partir de (67) e (68), as sequências de incrementos de controle ($\Delta\mathbf{u}_m$) e sinais de controle (\mathbf{u}_m) que minimizam os sistemas para o caso irrestrito, podem ser encontradas analiticamente ao igualar a derivada das funções g_y e g_r a zero, resultando em

$$\Delta\mathbf{u}_{\text{unc,opt}}(k) = -\mathbf{H}_y^{-1} \mathbf{b}_y^T, \quad (75)$$

$$\mathbf{u}_{\text{unc,opt}}(k) = -\mathbf{H}_r^{-1} \mathbf{b}_r^T, \quad (76)$$

que são equivalentes a

$$\Delta\mathbf{u}_{\text{unc,opt}}(k) = \mathbf{K}_y (\mathbf{w} - \mathbf{f}), \quad (77)$$

$$\mathbf{u}_{\text{unc,opt}}(k) = \mathbf{K}_{r,1} (\mathbf{w} - \mathbf{f}) + \mathbf{K}_{r,2} (\mathbf{u}(k-1)), \quad (78)$$

para

$$\mathbf{K}_y \in \mathbb{R}^{(n_\varepsilon N_u) \times (n_\varphi N)} = \left(\mathbf{G}_m^T \mathbf{Q}_\lambda \mathbf{G}_m + \mathbf{Q}_\delta \right)^{-1} \mathbf{G}_m^T \mathbf{Q}_\lambda, \quad (79)$$

$$\mathbf{K}_{r,1} \in \mathbb{R}^{(n_\varepsilon N_u) \times (n_\varphi N)} = \left(\mathbf{G}_m^T \mathbf{Q}_\lambda \mathbf{G}_m + \mathbf{S}^T \mathbf{Q}_\delta \mathbf{S} \right)^{-1} \mathbf{G}_m^T \mathbf{Q}_\lambda, \quad (80)$$

$$\mathbf{K}_{r,2} \in \mathbb{R}^{(n_\varepsilon N_u) \times (n_\varphi N)} = \left(\mathbf{G}_m^T \mathbf{Q}_\lambda \mathbf{G}_m + \mathbf{S}^T \mathbf{Q}_\delta \mathbf{S} \right)^{-1} \mathbf{S}^T \mathbf{Q}_\delta \mathbf{E}, \quad (81)$$

onde \mathbf{K}_y e \mathbf{K}_r são matrizes com os ganhos dos controladores GPC e CARMA-GPC de um sistema MIMO no caso irrestrito, respectivamente.

Nota-se que, caso a matriz de ponderação da variável manipulada (\mathbf{Q}_δ) seja nula em ambos os tipos de modelo, as matrizes \mathbf{H} , \mathbf{b} e \mathbf{f}_0 passam a utilizar as mesmas matrizes em seus cálculos, o que deixa o cômputo da predição da saída para esses diferentes modelos com esforço computacional mais semelhante. Caso contrário, são necessários muitos cálculos a mais para a predição da saída com o modelo CARMA.

Para o caso restrito, o problema de minimização passa a ser um problema de programação quadrática com restrições, descrito por

$$\Delta \mathbf{u}_{\text{opt}}(k) = \underset{\Delta \mathbf{u}_m(k)}{\text{argmin}} \|\mathbf{g}_y\|_2^2, \quad (82)$$

$$\mathbf{u}_{\text{opt}}(k) = \underset{\mathbf{u}_m(k)}{\text{argmin}} \|\mathbf{g}_r\|_2^2. \quad (83)$$

Para encontrar a sua solução ótima, podem ser utilizados algoritmos genéricos de programação quadrática, como o *quadprog* do software MATLAB[®]. Também existem algoritmos propostos na literatura que aceleram o processo de minimização, como o GPCADMM e o GPCGPAD (PECCIN *et al.*, 2020, 2021; BRUNONI *et al.*, 2022).

2.4 CONSIDERAÇÕES

Neste capítulo foram apresentados conceitos introdutórios necessários para o entendimento das técnicas abordadas neste trabalho. O princípio de funcionamento do inversor e a relação entre os vetores de tensão de entrada com os estados das chaves são necessários para compreender a estrutura dos algoritmos utilizados no Capítulo 3. Os conceitos do algoritmo de FCS-MPC padrão também servem como base para o entendimento dessas técnicas encontradas na literatura, visto que essas abordagens costumam ser modificações feitas sobre a estrutura convencional do algoritmo de controle preditivo. As soluções irrestritas do GPC e do CARMA-GPC são utilizadas nos algoritmos descritos na Subseção 3.1.6 e na Subseção 3.1.7. As diferentes formas de realizar a predição da saída do sistema trazem diferentes resultados e esforços ao algoritmo que podem ser visualizados no Capítulo 4.

3 ALGORITMOS PARA REDUÇÃO DO TEMPO DE CÔMPUTO DO FCS-MPC

Nos últimos anos, o controle preditivo baseado em modelo com conjunto de controle finito vem sendo aplicado eficientemente para resolver situações que exigem alto desempenho em drives e conversores eletrônicos (VAZQUEZ *et al.*, 2017). Entretanto, um fator que limita o desempenho desta formulação de controle preditivo é o tempo de cômputo necessário para executar o algoritmo de controle e o seu aumento exponencial conforme os horizontes são ampliados (LEZANA; AGUILERA; QUEVEDO, 2009).

Neste capítulo, são apresentados alguns dos métodos encontrados na literatura, bem como adaptações propostas pelo autor, para reduzir o fardo computacional do algoritmo de FCS-MPC. A Seção 3.1 demonstra três técnicas presentes na literatura para reduzir o esforço computacional de um algoritmo de FCS-MPC, junto com uma abordagem proposta que utiliza características de dois dos métodos descritos. Algumas considerações sobre as técnicas abordadas neste capítulo são encontradas na Seção 3.2.

3.1 DIMINUIÇÃO DO TEMPO DE CÔMPUTO

Por conta do alto custo de processamento requerido nos cálculos de algoritmos do tipo FCS-MPC, um objetivo em comum entre muitos pesquisadores da área é o desenvolvimento de algoritmos com baixo esforço computacional (ELMORSHEDY *et al.*, 2021).

Em Rashwan *et al.* (2017), a abordagem foi diminuir o tempo de execução do código sem diminuir a sua complexidade, ao dividir o esforço computacional entre um grupo de microcontroladores. Em Gutierrez e Kwak (2018), um método complexo de pré-seleção foi abordado, enquanto em Veysinejad, Rahmati e Shamlou (2019) e Veysinejad, Rahmati e Bina (2019), a estratégia foi seccionar a área do espaço vetorial.

Neste trabalho, este objetivo de redução do tempo de cômputo foi explorado através de três métodos diferentes. Um método iterativo que identifica qual o vetor mais próximo da referência sem precisar avaliar todas as opções, proposto por Guo *et al.* (2021), uma abordagem que reduz as opções de controle para aquelas adjacentes ao vetor atual, descrita em Cortés *et al.* (2010), e um algoritmo de decodificação via esferas (SDA, do inglês *Sphere Decoding Algorithm*), que propõe uma redução considerável do esforço computacional devido ao seu princípio de limites inteligentes (KARAMANAKOS *et al.*, 2016). Adicionalmente, este trabalho propõe de forma inédita a implementação de um algoritmo que combina as duas linhas de pesquisa, ao utilizar o algoritmo de decodificação via esferas em conjunto com a técnica de vetores adjacentes. Este algoritmo foi nomeado como vetores adjacentes pré-selecionados para algoritmo de decodificação via esferas (PAV-SDA, do inglês *Preselected Adjacent Vectors for Sphere Decoding Algorithm*). Os algoritmos correspondentes às técnicas mencionadas são explicados nesta seção.

3.1.1 Redução de vetores disponíveis por método iterativo: formulação original

Em um algoritmo padrão de FCS-MPC para inversores trifásicos de dois níveis, 8 opções de chaveamento podem ser consideradas pelo controlador, para cada amostra dentro do horizonte de controle. Portanto, para horizontes de controle maiores, a quantidade de opções cresce exponencialmente (8^{N_u}), visto que para cada opção em uma amostra, são criadas 8 combinações diferentes com a amostra seguinte. Então, para reduzir o esforço computacional desse algoritmo, alguns estudos são encontrados na literatura sobre a redução de vetores disponíveis ao controlador. Neste trabalho, uma das alternativas seguidas dentro desta linha consiste em uma escolha iterativa para encontrar o vetor mais próximo à referência, sem precisar considerar todos os vetores possíveis.

Este método, descrito em Guo *et al.* (2021) e intitulado por FCS-MPC com conjunto simplificado de controle (FCS-MPC-SCS, do inglês FCS-MPC *Simplified Control Set*), foi originalmente aplicado a um VSI de cinco fases e dois níveis. Neste tipo de conversor, são 32 vetores disponíveis para cada amostra do horizonte de controle. Convencionalmente, para excluir a componente de terceira harmônica da resposta do sistema com esta topologia de inversor, 10 vetores de tensão virtuais são gerados e utilizados como opção para o FCS-MPC abordado.

Essa técnica escolhe vetores de lados opostos do conjunto de vetores disponíveis e setoriza o local onde a resposta pode estar localizada dependendo do resultado das comparações feitas. Por exemplo, na Figura 5(a), o vetor denominado \mathbf{V}_{ref} é um vetor hipotético, não conhecido pelo controlador, que representa a melhor escolha para o sistema de controle naquele instante. Entretanto, a escolha do algoritmo FCS-MPC é restrita aos 10 vetores virtuais descritos na imagem. Deste modo, o primeiro passo para encontrar o vetor mais próximo de \mathbf{V}_{ref} consiste em comparar os custos associados a 2 vetores opostos, \mathbf{V}_1 e \mathbf{V}_6 . Considerando uma função custo que leve apenas o erro como um parâmetro de escolha, como neste caso \mathbf{V}_{ref} está mais próximo de \mathbf{V}_1 , o custo deste vetor ($g(\mathbf{V}_1)$) é menor que o custo de \mathbf{V}_6 ($g(\mathbf{V}_6)$). Portanto, define-se que o vetor ótimo está na seção destacada na Figura 5(b).

Com esta seção delimitada, os custos dos vetores adjacentes de \mathbf{V}_1 (\mathbf{V}_2 e \mathbf{V}_{10}) seriam computados e posteriormente comparados com o custo de \mathbf{V}_1 . Caso \mathbf{V}_6 fosse escolhido no passo anterior, os vetores computados seriam \mathbf{V}_5 e \mathbf{V}_7 . Neste exemplo, o vetor \mathbf{V}_2 seria definido como o mais próximo das 3 opções, ou seja, $g_{\min} = g(\mathbf{V}_2)$, pois $g(\mathbf{V}_2) < g(\mathbf{V}_1) < g(\mathbf{V}_{10})$. Isto resultaria em uma redução da região onde a referência se encontra, demonstrada na nova área destacada na Figura 5(c).

Sequencialmente, o custo de \mathbf{V}_3 seria calculado e seria constatado que $g(\mathbf{V}_3) < g(\mathbf{V}_2)$. Portanto, \mathbf{V}_3 seria definido como o vetor não nulo (\mathbf{V}_{nn}) mais próximo da referência. O fluxograma apresentado na Figura 6(a) descreve o princípio de funcionamento do FCS-MPC-SCS para encontrar o \mathbf{V}_{nn} , com qualquer \mathbf{V}_{ref} , utilizando um VSI de cinco fases e dois níveis. Por fim, como é demonstrado na Figura 6(b), o custo do vetor nulo (\mathbf{V}_0) é computado e comparado com o custo associado ao vetor ótimo não nulo ($\mathbf{V}_{\text{nn}} = \mathbf{V}_3$). Deste modo, considerando que

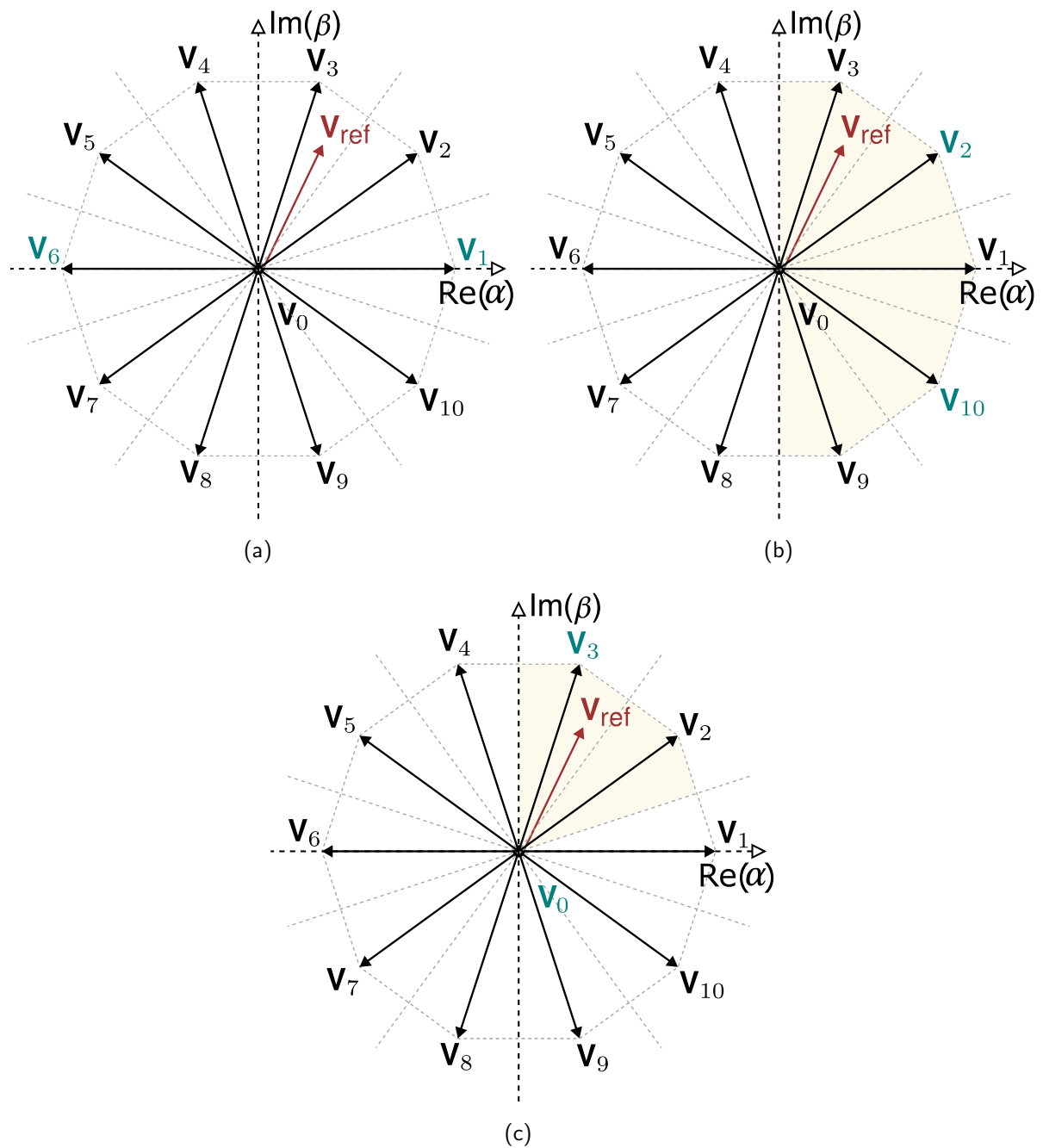


Figura 5 – Diagrama esquemático de exemplo da seleção do vetor de tensão ótimo: (a) posicionamento de \mathbf{V}_{ref} e vetores utilizados na primeira comparação; (b) área delimitada pela primeira comparação e vetores utilizados na segunda comparação; (c) área resultante pela segunda comparação e vetores utilizados na terceira comparação. Adaptado de Guo *et al.* (2021).

neste exemplo $g(\mathbf{V}_3) < g(\mathbf{V}_0)$, \mathbf{V}_3 seria definido como o vetor ótimo (\mathbf{V}_{opt}) e seria a escolha do controlador para esta amostra hipotética de controle.

3.1.2 Redução de vetores disponíveis por método iterativo: adaptações propostas

Para implementar o algoritmo FCS-MPC-SCS neste trabalho, poucas mudanças precisaram ser executadas, visto que em Guo *et al.* (2021) o atraso de cômputo já é considerado e a seleção de vetores já é feita em $\alpha\beta$. Entretanto, vale ressaltar que apenas a parcela relacionada à seleção iterativa do FCS-MPC-SCS foi implementada neste trabalho. Além disso, como o conversor utilizado neste trabalho difere do abordado no estudo de caso original, a sequência de vetores analisados foi adaptada para funcionar com o conjunto finito de opções descrito na Figura 3 (Capítulo 2, p. 22).

O primeiro passo do algoritmo adaptado continua sendo encontrar o lado mais próximo do vetor de referência. Todavia, considerando o conjunto finito de 8 opções disponíveis em um VSI trifásico de dois níveis, os vetores utilizados para a comparação passam a ser \mathbf{V}_1 e \mathbf{V}_4 . Em um novo exemplo hipotético, considerando novamente uma função que calcula o custo apenas pelo erro de predição, e um vetor de referência na posição indicada pela Figura 7 (a), é visto que $g(\mathbf{V}_1) < g(\mathbf{V}_4)$, fazendo com que a região direita do hexágono seja demarcada como a área onde está a referência, como mostra a Figura 7(b).

Posteriormente, seguindo a estrutura do algoritmo original, compara-se o custo de \mathbf{V}_1 com o custo dos seus vetores adjacentes, \mathbf{V}_2 e \mathbf{V}_6 . Entretanto, uma das adaptações deste trabalho envolve dividir esta etapa em duas partes. Primeiro avalia-se \mathbf{V}_2 . Caso o custo de \mathbf{V}_2 seja menor que o de \mathbf{V}_1 , não há a necessidade de computar o custo de \mathbf{V}_6 , o que reduz ainda mais o esforço do algoritmo. Se $g(\mathbf{V}_2) > g(\mathbf{V}_1)$, \mathbf{V}_6 é computado. O mesmo serve para o outro lado do hexágono, como é descrito na Figura 8(a). Como, neste exemplo, $g(\mathbf{V}_2) < g(\mathbf{V}_1)$, sabe-se que a área demarcada na Figura 7(c) é onde está o vetor de tensão ótimo e, portanto, o vetor ótimo não nulo (\mathbf{V}_{nn}) já pode ser definido como \mathbf{V}_2 .

Com \mathbf{V}_{nn} encontrado, a única comparação que resta ser feita é com um dos vetores nulos. A escolha do vetor nulo (\mathbf{V}_{null}) é feita com base na distância entre os estados das chaves dos dois vetores nulos redundantes com o último vetor escolhido pelo controlador ($\mathbf{V}_{\text{opt}}(k-1)$). O vetor nulo com menos trocas de estados nas chaves em relação a $\mathbf{V}_{\text{opt}}(k-1)$ tem seu custo calculado e comparado com \mathbf{V}_{nn} nessa fase, como ilustra o diagrama da Figura 8(b). Supondo que o vetor nulo mais próximo fosse \mathbf{V}_0 neste exemplo, o seu custo seria comparado com \mathbf{V}_2 . Por fim, a Figura 7(c) retrata que, como \mathbf{V}_{ref} está mais próximo de \mathbf{V}_2 , este seria definido como o vetor a ser implementado pelo algoritmo de controle.

Com este exemplo, é possível verificar a diminuição do esforço que esse algoritmo proporciona ao FCS-MPC, uma vez que em seu melhor caso para a versão adaptada (o vetor de cima é o mais próximo do vetor ótimo) o algoritmo calcula a predição do sistema para os vetores: \mathbf{V}_1 e \mathbf{V}_4 ; \mathbf{V}_2 ou \mathbf{V}_3 ; \mathbf{V}_0 ou \mathbf{V}_7 . Ou seja, apenas metade (4) dos vetores disponíveis seriam considerados. Para o caso no qual o vetor não está na região de cima são 5 vetores

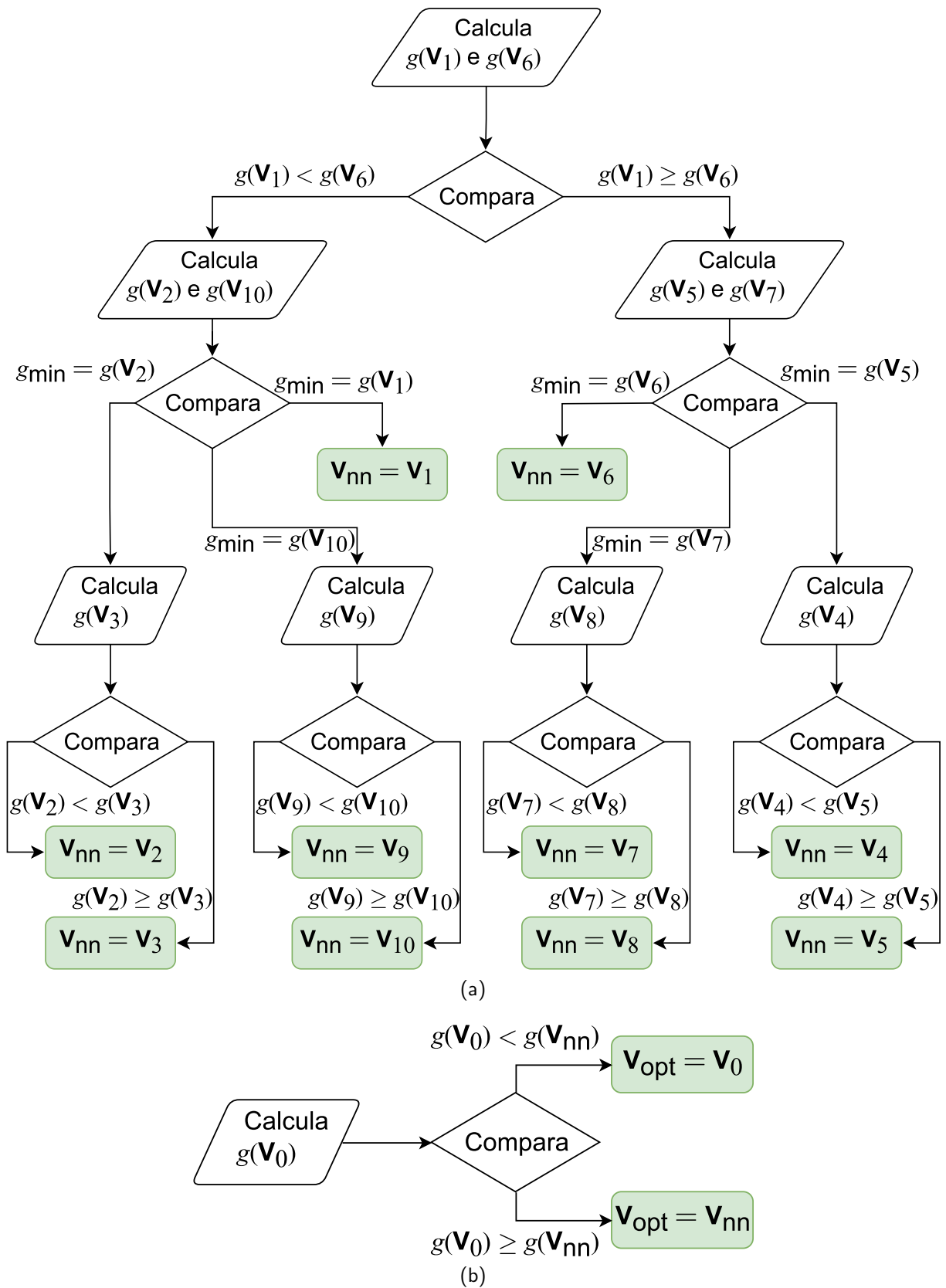


Figura 6 – Fluxograma do algoritmo FCS-MPC-SCS para: (a) seleção do vetor de tensão não nulo mais próximo da referência; (b) seleção do vetor ótimo. Adaptado de Guo *et al.* (2021).

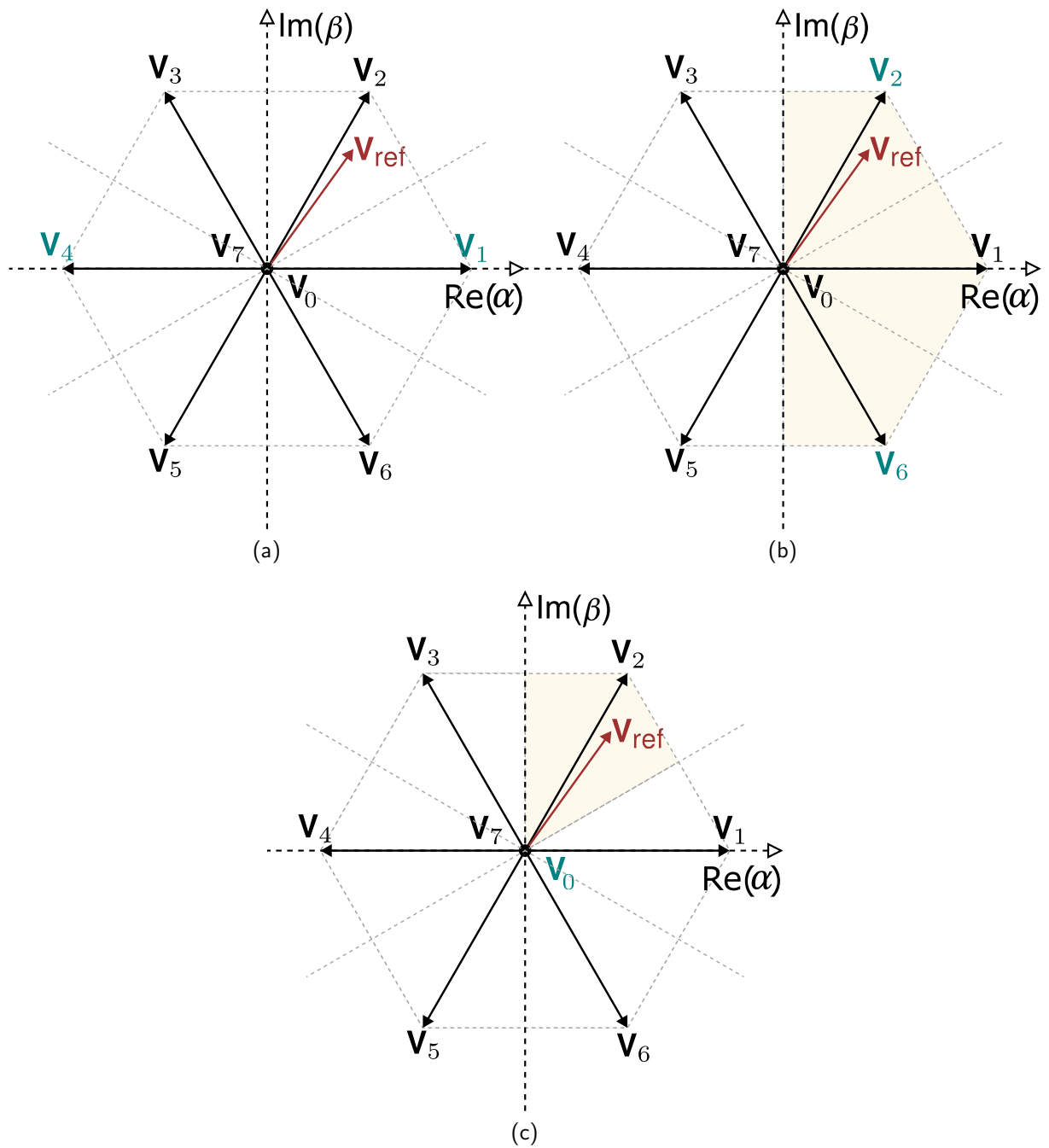
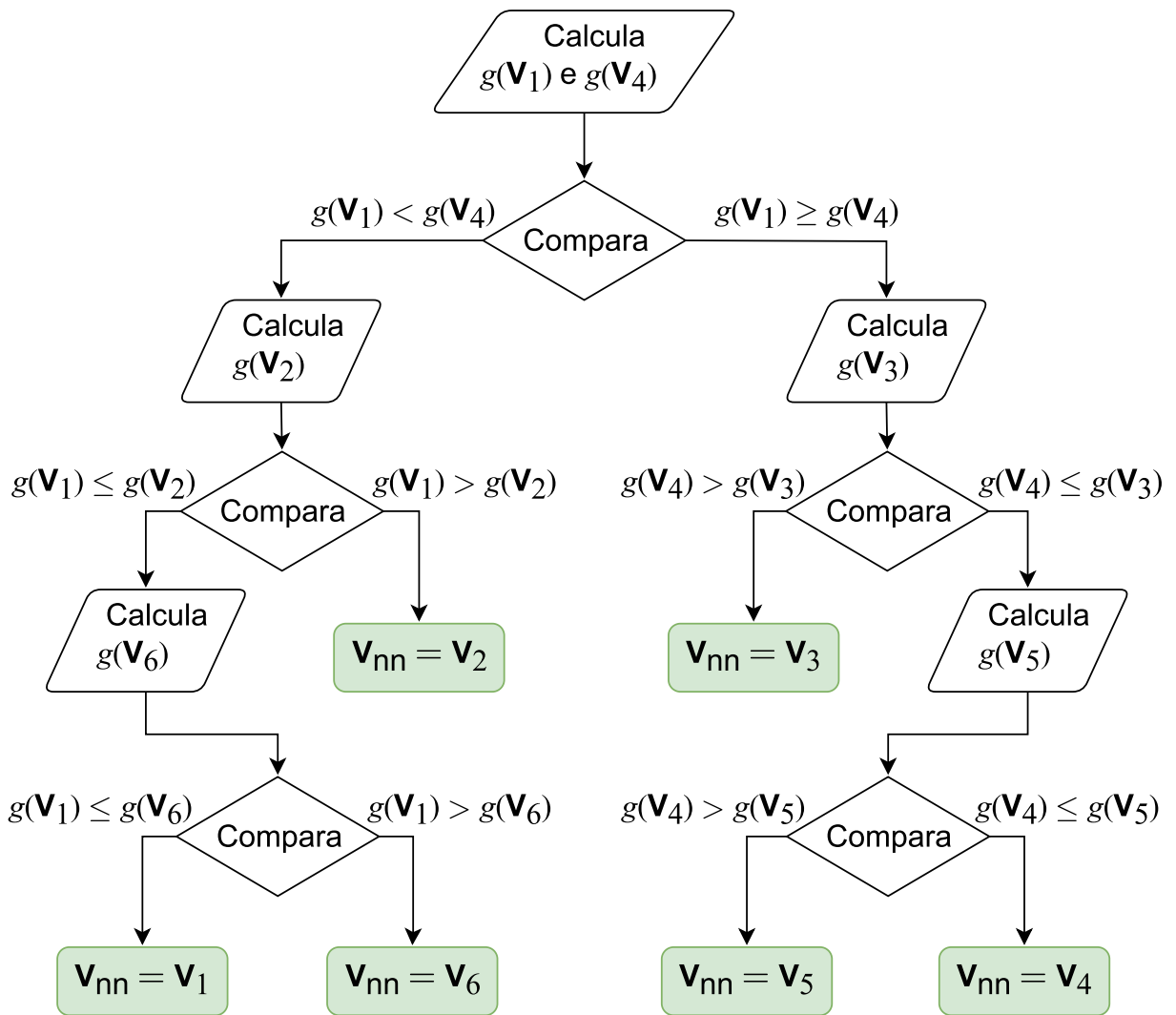
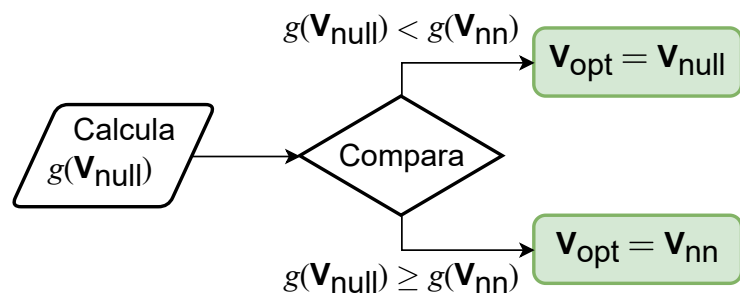


Figura 7 – Exemplo do FCS-MPC-SCS adaptado para o VSI trifásico de dois níveis: (a) posicionamento de \mathbf{V}_{ref} e vetores utilizados na primeira comparação; (b) área delimitada pela primeira comparação e vetores utilizados na segunda comparação; (c) área resultante pela segunda comparação e vetor nulo utilizado na última comparação.



(a)



(b)

Figura 8 – Fluxograma do algoritmo FCS-MPC-SCS com adaptações para: (a) seleção do vetor de tensão não nulo mais próximo da referência; (b) seleção do vetor ótimo.

considerados, frente aos 8 vetores que teriam seus custos calculados pela utilização do FCS-MPC convencional. Entretanto, é importante ressaltar que neste trabalho, assim como em Guo *et al.* (2021), essa técnica só foi utilizada em horizontes de uma única amostra, pois sua generalização para horizontes mais longos demonstrou ser complexa.

3.1.3 Redução de vetores disponíveis por vetores adjacentes: formulação original

Outro método implementado com o objetivo de diminuir a quantidade de vetores disponíveis foi apresentado em Cortés *et al.* (2010) e implementado para um inversor multinível ponte H em cascata. Neste conversor, a cada amostra de controle existem 125 opções de chaveamento disponíveis e, ao desconsiderar as redundâncias, 61 vetores de tensão diferentes podem ser encontrados.

A alternativa proposta em Cortés *et al.* (2010) consiste em elaborar um subconjunto com apenas os vetores adjacentes ao último vetor implementado ($\mathbf{V}(k-1)$). Para isso, utiliza-se a distância entre os vetores disponíveis no conjunto ($\mathbf{V}(k)$) e $\mathbf{V}(k-1)$, que pode ser calculada por

$$\sigma(\mathbf{V}(k-1), \mathbf{V}(k)) = \sqrt{(\mathbf{V}_\alpha(k-1) - \mathbf{V}_\alpha(k))^2 + (\mathbf{V}_\beta(k-1) - \mathbf{V}_\beta(k))^2}, \quad (84)$$

onde \mathbf{V}_α e \mathbf{V}_β são as coordenadas em α (parte real) e em β (parte imaginária) dos vetores $\mathbf{V}(k-1)$ e $\mathbf{V}(k)$. Esse cálculo é feito off-line para todas as opções disponíveis no conjunto de controle e, caso a distância entre $\mathbf{V}(k-1)$ e $\mathbf{V}(k)$ seja igual ou menor que $2V_{cc}/3$, ambos os vetores são considerados adjacentes. Para o conversor abordado em Cortés *et al.* (2010), ao fazer essa comparação de distância, é gerada uma lista para cada vetor com apenas 7 vetores adjacentes resultantes. Com este método, estes 7 vetores são os únicos cujos custos são calculados pelo algoritmo de FCS-MPC. Conforme um novo vetor é escolhido, novos 7 vetores adjacentes são considerados para o próximo ciclo de controle.

3.1.4 Redução de vetores disponíveis por vetores adjacentes: adaptações propostas

Neste trabalho, algumas modificações foram realizadas para a implementação da técnica de vetores adjacentes, proposta originalmente em Cortés *et al.* (2010). Quanto às redundâncias descartadas na formulação original, em um VSI trifásico de dois níveis não existem tantas redundâncias para serem desconsideradas, como foi em Cortés *et al.* (2010) com o inversor multinível ponte H em cascata. Apenas os dois vetores nulos são redundantes, mas ambos são mantidos no conjunto de opções do FCS-MPC para serem utilizados com seus vetores adjacentes, conforme descrito nesta subseção.

Na sua formulação original, esta técnica foi empregada para $N_1 = 1$. Portanto, a compensação do atraso relacionado ao cálculo de cômputo foi uma das adaptações realizadas neste trabalho. Deste modo, os vetores disponíveis em k para serem implementados em $k+1$ passam a ser os vetores adjacentes de $\mathbf{V}_{\text{opt}}(k-1)$, o vetor que foi escolhido na amostra anterior e implementado no início de k .

Além disso, em Cortés *et al.* (2010), apenas horizontes de uma amostra foram considerados. Com isso, a generalização para horizontes maiores foi proposta neste trabalho. Portanto, o vetor considerado para escolha em k ($\mathbf{V}_{\text{sel}}(k)$) passa a limitar a escolha em $k+1$ para apenas os seus vetores adjacentes. Uma vez que um vetor adjacente é selecionado em $k+1$ ($\mathbf{V}_{\text{sel}}(k+1)$), apenas os seus vetores adjacentes são considerados em $k+2$, e assim sucessivamente. Ao fim dos horizontes, um custo é associado à sequência avaliada e outros vetores adjacentes do vetor escolhido em $k-1$ podem ser considerados em k , resultando em diferentes vetores adjacentes avaliados no decorrer dos horizontes.

Nesta nova abordagem proposta, é definida pelo projetista qual a relação mais distante permitida pelo algoritmo (Q_{max}), ou seja, quantas chaves podem ter seus estados alterados para que o vetor possa ser considerado adjacente em relação a outro vetor. Com Q_{max} definido, os vetores que entrarão em cada subconjunto de adjacentes são selecionados de acordo com a distância descrita na Tabela 2. Considerando \mathbf{V}_1 como $\mathbf{V}_{\text{opt}}(k-1)$, um exemplo dessas distâncias pode ser visto na Figura 9. Se o Q correspondente a linha de um vetor com a coluna de outro vetor for menor ou igual a Q_{max} , os vetores são considerados adjacentes para o código.

Tabela 2 – Tabela de pesquisa para encontrar a quantidade de troca de estados das chaves entre vetores.

$S_{abc}(k-1) \mid S_{abc}(k)$	0	1	2	3	4	5	6	7
0	0	1	2	1	2	1	2	3
1	1	0	1	2	3	2	1	2
2	2	1	0	1	2	3	2	1
3	1	2	1	0	1	2	3	2
4	2	3	2	1	0	1	2	1
5	1	2	3	2	1	0	1	2
6	2	1	2	3	2	1	0	1
7	3	2	1	2	1	2	1	0

Por exemplo, se $\mathbf{V}_{\text{opt}}(k-1)$ for \mathbf{V}_2 (correspondente a $S_{abc} = [1, 1, 0]$), em k estarão disponíveis os seguintes vetores, ao considerar uma distância máxima (Q_{max}) de

- 0 chave: \mathbf{V}_2 (1 vetor disponível), pois na linha de \mathbf{V}_2 é a única coluna que possui valor menor ou igual a 0;
- 1 chave: \mathbf{V}_1 , \mathbf{V}_2 , \mathbf{V}_3 e \mathbf{V}_7 (4 vetores disponíveis), pois na linha de \mathbf{V}_2 são as colunas que possuem valores menores ou iguais a 1;
- 2 chaves: todos exceto \mathbf{V}_5 (7 vetores disponíveis), pois na linha de \mathbf{V}_2 é a única coluna que possui valor maior que 2;

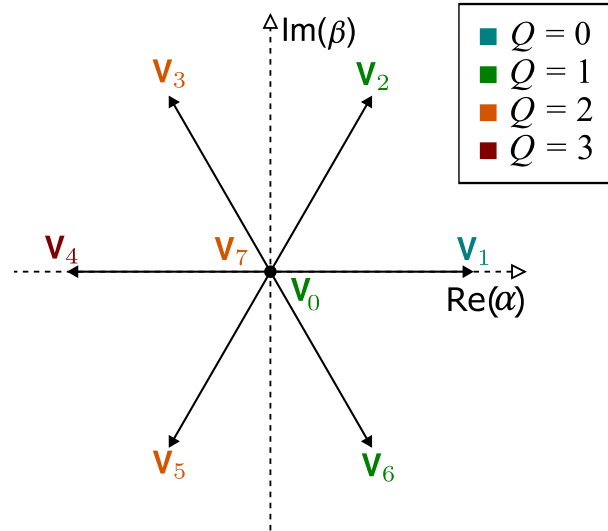


Figura 9 – Exemplo de distâncias entre vetores do ponto de vista geométrico, para $\mathbf{V}_{\text{opt}(k-1)} = \mathbf{V}_1$.

- 3 chaves: todos (8 vetores disponíveis), pois todas as colunas possuem valores menores ou iguais a 3.

Essa definição dos subconjuntos de vetores adjacentes pode ser feita antes da execução do algoritmo de controle para que o tempo de execução do código seja reduzido.

Em um inversor trifásico de dois níveis, utilizar apenas 1 chave ou menos de distância pode ser uma medida muito extrema, comprometendo muito a qualidade do seguimento de referência. Ao mesmo tempo, a escolha da distância máxima em 2 chaves representa uma diminuição de apenas 1 vetor por amostra. Portanto, uma alternativa proposta por esse trabalho, para que possam ser utilizadas menos opções que na distância de 2 chaves sem comprometer tanto a saída como acontece na distância de uma chave, consiste em definir quantos vetores nulos poderão estar presentes no conjunto adjacente (Q_{null}). Caso $Q_{\text{null}} = 2$, os dois vetores nulos poderão entrar no subconjunto de adjacentes se atenderem a distância imposta por Q_{max} . Todavia, para $Q_{\text{null}} = 1$, apenas o vetor nulo mais próximo pode ser selecionado para cada grupo de vetores adjacentes. Com isso, para os casos nos quais o vetor escolhido em $k-1$ não for um dos vetores nulos (\mathbf{V}_0 ou \mathbf{V}_7), apenas 6 vetores ficam disponíveis para serem escolhidos em k , ao definir $Q_{\text{max}} = 2$. Caso \mathbf{V}_0 ou \mathbf{V}_7 seja o vetor escolhido em $k-1$, a quantidade de vetores disponíveis permanece sendo 7.

Um exemplo dos vetores considerados pelo algoritmo proposto com vetores adjacentes, para $Q_{\text{max}} = 1$ e $\mathbf{V}_{\text{opt}(k-1)} = \mathbf{V}_1$, está presente na Figura 10.

Nota-se que com esta abordagem, não há garantia de que a solução ótima da função custo esteja no subconjunto de vetores adjacentes, portanto, a solução encontrada com esta técnica é categorizada como subótima. Além disso, em sua formulação original o algoritmo não foi nomeado. Porém, para facilitar a identificação da técnica, neste trabalho chamaremos esse algoritmo de vetores adjacentes pré-selecionados para FCS-MPC (PAV-FCS-MPC, do inglês *Preselected Adjacent Vectors-FCS-MPC*).

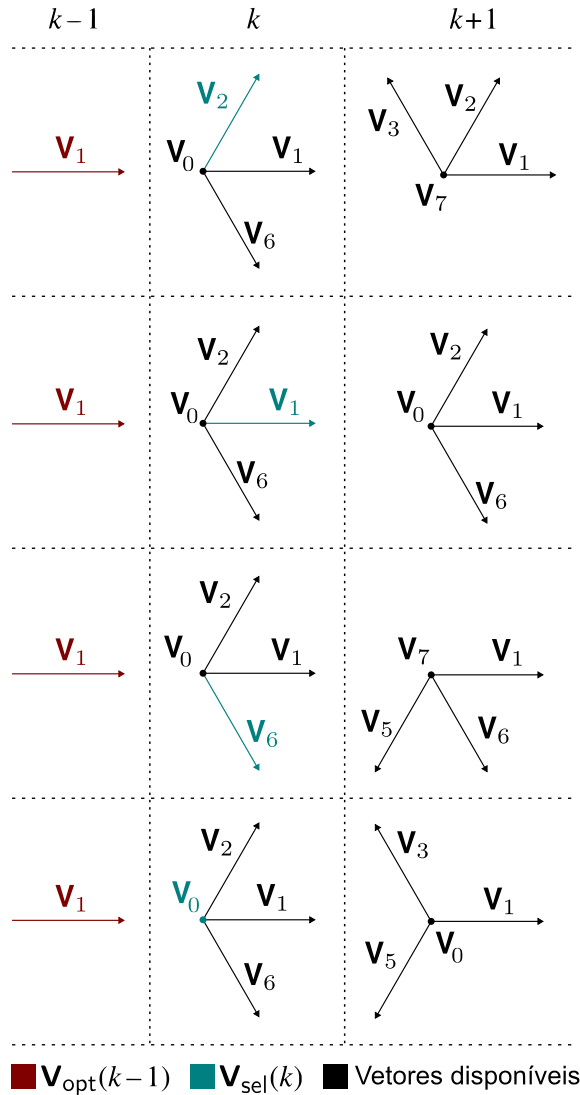


Figura 10 – Exemplo de funcionamento do algoritmo de vetores adjacentes.

3.1.5 Algoritmo de decodificação via esferas: formulação original

O algoritmo de decodificação via esferas tem se destacado por sua efetividade em sistemas lineares com entradas de variáveis do tipo inteiro, como acontece no contexto de eletrônica de potência (KARAMANAKOS *et al.*, 2016). Em Geyer e Quevedo (2014), o algoritmo foi aplicado para o controle de um VSI trifásico de três níveis com ponto neutro grampeado (NPC, do inglês *Neutral Point Clamped*), que possui 27 vetores disponíveis para cada amostra de controle, empregado para o acionamento de um motor por indução.

A representação do modelo de tempo contínuo do motor de indução é feita no espaço de estados, por

$$\frac{dx(t)}{dt} = \Phi x(t) + Y u_{\alpha\beta}(t), \tag{85}$$

$$y(t) = C x(t), \tag{86}$$

onde,

$$\mathbf{x}(t) = \begin{bmatrix} i_{s\alpha}(t) & i_{s\beta}(t) & \psi_{r\alpha}(t) & \psi_{r\beta}(t) \end{bmatrix}^T, \quad (87)$$

$$\mathbf{u}_{\alpha\beta}(t) = \begin{bmatrix} u_{\alpha}(t) & u_{\beta}(t) \end{bmatrix}^T, \quad (88)$$

Φ é a matriz de estados, \mathbf{Y} é a matriz de entradas e \mathbf{C} é a matriz de saídas. Estas matrizes podem ser encontradas em Geyer e Quevedo (2014). As variáveis de estado i_{sx} e ψ_{rx} , para $x \in \{\alpha, \beta\}$, são as correntes no estator e o fluxo concatenado no rotor nas coordenadas α e β , respectivamente.

A partir deste modelo em tempo contínuo, encontra-se a representação em espaço de estados para o tempo discreto, dada por

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}_{\alpha\beta}(k), \quad (89)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \quad (90)$$

com $k \in \mathbb{N}$, onde $\mathbf{A} = e^{\Phi T_s}$ e $\mathbf{B} = -\Phi^{-1}(\mathbf{I} - \mathbf{A})\mathbf{Y}$. O vetor de entradas $\mathbf{u}_{\alpha\beta}$ é o resultado da transformada de Clarke (\mathbf{P}) aplicada ao vetor \mathbf{u}_{abc}

$$\mathbf{u}_{\alpha\beta}(k) = \mathbf{P} \begin{bmatrix} u_a(k) & u_b(k) & u_c(k) \end{bmatrix}^T, \quad (91)$$

onde $u_x \in \{-1, 0, 1\}$, para $x \in \{a, b, c\}$, representa o vetor com os estados das chaves do conversor. Como os estados das chaves em $\mathbf{u}_{\alpha\beta}$, ao contrário de \mathbf{u}_{abc} , não são valores inteiros, a equação (89) é modificada para que as entradas do sistema estejam contidas em $\{-1, 0, 1\}$, resultando em

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{P}\mathbf{u}_{abc}(k). \quad (92)$$

A equação (92) pode ser utilizada para realizar a predição do modelo de tempo discreto do sistema. Assim como foi exemplificado na Subseção 2.3.3, a solução irrestrita de uma função custo quadrática é encontrada a partir do seu ponto de mínimo, ou seja, o ponto definido ao igualar a derivada da função a zero, e os seus elementos encontrados não são necessariamente valores inteiros. Portanto, através da função custo utilizada em Geyer e Quevedo (2014), descrita por

$$g = \sum_{i=1}^N \left(\|\mathbf{e}_{i,\alpha\beta}(k+i)\|_2^2 + \lambda_u \|\Delta\mathbf{u}_{abc}(k+i-1)\|_2^2 \right), \quad (93)$$

para

$$\Delta\mathbf{u}_{abc}(k) = \mathbf{u}_{abc}(k) - \mathbf{u}_{abc}(k-1), \quad (94)$$

$$\mathbf{e}_{i,\alpha\beta}(k+1) = w_{i,\alpha\beta}(k+1) - \mathbf{C}\mathbf{x}(k+1), \quad (95)$$

sendo $w_{i,\alpha\beta}$ a referência da corrente no sistema de coordenadas $\alpha\beta$, o vetor $\mathbf{u}_{\text{unc,opt}} \in \mathbb{R}^{3N}$ é encontrado e sua definição é dada por

$$\mathbf{u}_{\text{unc,opt}}(k) = -\mathbf{Q}^{-1}\Theta(k), \quad (96)$$

onde $\mathbf{Q} \in \mathbb{R}^{3N \times 3N}$ e $\mathbf{\Theta} \in \mathbb{R}^{3N}$. Ambos os termos dependem das matrizes da representação de tempo discreto em espaço de estados e são definidos em Geyer e Quevedo (2014). Como \mathbf{Q} é uma matriz positiva definida e simétrica por definição, a seguinte decomposição de Cholesky pode ser feita

$$\mathbf{H}_{\text{sda}}^{-1} \mathbf{H}_{\text{sda}}^{-T} = \mathbf{Q}^{-1}, \quad (97)$$

onde $\mathbf{H}_{\text{sda}} \in \mathbb{R}^{3N \times 3N}$ é uma matriz inversível triangular inferior, denominada como matriz geradora de nós e é responsável por gerar o espaço discreto onde está localizada a solução do problema. Com esta matriz, é possível encontrar a solução irrestrita nesse novo espaço gerado ($\bar{\mathbf{u}}_{\text{unc,opt}} \in \mathbb{R}^{3N}$), através de

$$\bar{\mathbf{u}}_{\text{unc,opt}}(k) = \mathbf{H}_{\text{sda}} \mathbf{u}_{\text{unc,opt}}(k). \quad (98)$$

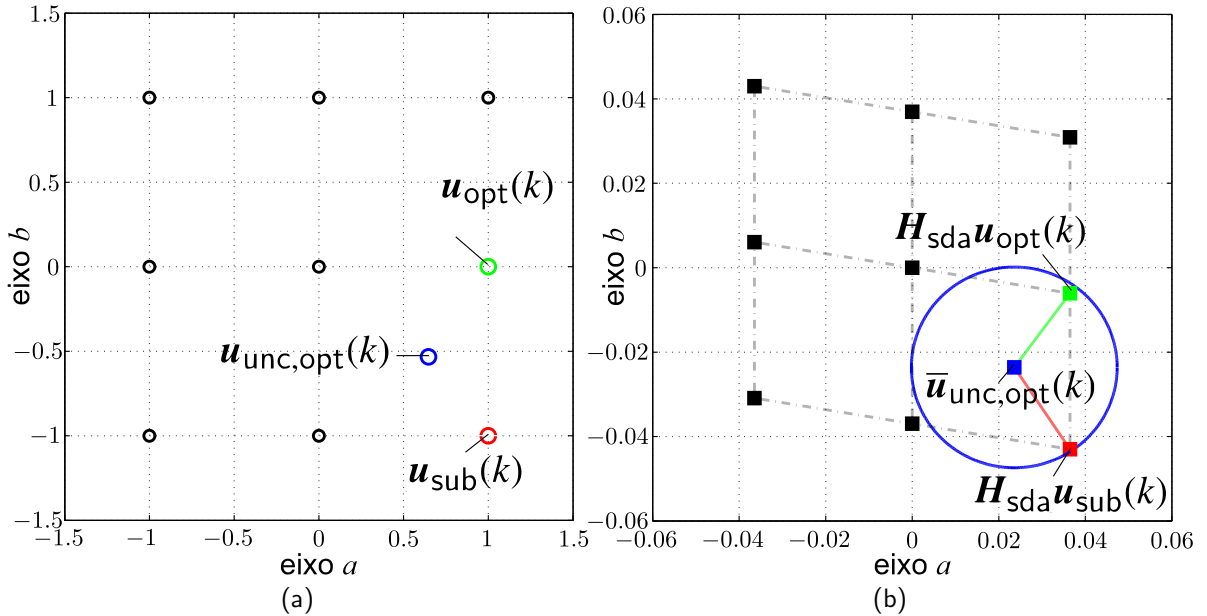


Figura 11 – Visualização do SDA no plano ab para o horizonte $N = 1$. (a) Problema de otimização no sistema de coordenadas ortogonal. (b) Problema de otimização no sistema de coordenadas transformado criado por \mathbf{H}_{sda} . Adaptado de Geyer e Quevedo (2014).

Então, seguindo os procedimentos descritos em Geyer e Quevedo (2014), a função custo apresentada em (93) é reformulada para o formato de mínimos quadrados inteiros, dado por

$$\mathbf{u}_{\text{opt}}(k) = \arg \min_{\mathbf{u}(k) \in \mathbb{U}} \|\mathbf{H}_{\text{sda}} \mathbf{u}(k) - \bar{\mathbf{u}}_{\text{unc,opt}}(k)\|_2^2, \quad (99)$$

onde $\mathbf{u}_{\text{opt}}(k) \in \mathbb{U}^{3N}$ é um dos vetores $\mathbf{u}(k) \in \mathbb{U}^{3N} = [\mathbf{u}_{\text{abc}}^T(k) \ \dots \ \mathbf{u}_{\text{abc}}^T(k+N-1)]^T$ que descreve uma dada sequência de vetores definida como solução do problema de minimização. \mathbb{U} é o conjunto finito onde estão todas as sequências de soluções factíveis, ou seja, todas as sequências geradas pela combinação dos 27 vetores disponíveis para cada amostra do horizonte.

Portanto, os vetores com os estados das chaves que são disponíveis ao controlador, originalmente representados no sistema de coordenadas ortogonal abc , conforme ilustrado na Figura 11(a), são transformados para um novo espaço gerado pela matriz \mathbf{H}_{SDA} , descrito na Figura 11(b). Neste novo espaço, a distância euclidiana entre a solução irrestrita transformada ($\bar{\mathbf{u}}_{\text{unc,opt}}(k)$) e os vetores disponíveis transformados ($\bar{\mathbf{u}}(k)$) pode ser utilizada como métrica para definir qual a solução ótima do problema. Vale ressaltar que o vetor mais próximo de $\mathbf{u}_{\text{unc,opt}}(k)$ antes da transformação espacial pode ser apenas um vetor que representa a solução subótima do problema ($\mathbf{u}_{\text{sub}}(k)$) dentro do novo espaço transformado, como pode ser visto na Figura 11.

Pelo ponto de vista computacional, o SDA minimiza a função custo (99) através da estrutura descrita pelo Algoritmo 1. A inicialização deste algoritmo é dada por

$$\mathbf{u}_{\text{opt}}(k) = \text{MSphDec} \left([], 0, 1, \rho_{\text{ini}}^2, \bar{\mathbf{u}}_{\text{unc,opt}} \right), \quad (100)$$

onde $[]$ é um vetor vazio. Os parâmetros considerados para a chamada da função do SDA (SphDec) são: o vetor com a sequência avaliada (\mathbf{u}), inicializado por um vetor vazio; a distância quadrática entre o vetor avaliado e a solução irrestrita no espaço transformado (σ^2), que começa em 0; o índice da camada que será avaliada (i), que inicia em 1; o raio da esfera ao quadrado ($\rho(k)^2$), inicializado pelo raio inicial ao quadrado (ρ_{ini}^2); e a sequência de controle ótima para o caso irrestrito, transformada para o novo espaço gerado ($\bar{\mathbf{u}}_{\text{unc,opt}}$), que é mantida constante dentro de cada amostra de controle.

Algoritmo 1: Pseudocódigo para SDA. Adaptado de Geyer e Quevedo (2014).

```

Function SphDec( $\mathbf{u}$ ,  $\sigma^2$ ,  $i$ ,  $\rho^2$ ,  $\bar{\mathbf{u}}_{\text{unc,opt}}$ ):
  para cada  $u_x \in \{-1, 0, 1\}$  faça
     $\mathbf{u}^{(i)} = u_x$ 
    Calcula  $\sigma'^2$  recursivamente para os termos relacionados a  $i$ , conforme (105)
    se  $\sigma'^2 \leq \rho^2$  então
      se  $i < 3N$  então
        | SphDec( $\mathbf{u}$ ,  $\sigma'^2$ ,  $i+1$ ,  $\rho^2$ ,  $\bar{\mathbf{u}}_{\text{unc,opt}}$ )
      senão
        | se  $\mathbf{u} \in \mathbb{U}$  então
          |  $\mathbf{u}_{\text{opt}}(k) = \mathbf{u}$ 
          |  $\rho^2 = \sigma'^2$ 

```

O raio inicial da esfera (ρ_{ini}) é calculado através de uma solução estimada (\mathbf{u}_{est}) pelo algoritmo. Deseja-se que este raio seja iniciado com um valor mínimo para que a solução seja encontrada da forma mais rápida possível, mas que não seja tão pequeno a ponto de não existir soluções factíveis dentro da esfera (GEYER; QUEVEDO, 2014). Existem duas abordagens que são mais utilizadas para estimar esta solução inicial. A primeira maneira, denominada de “estimador Babai”, consiste em utilizar a solução irrestrita não transformada da função custo ($\mathbf{u}_{\text{unc,opt}}$) e arredondá-la para um vetor de números inteiros, que esteja contido

dentro do conjunto finito de sequências disponíveis para o controlador (\mathbb{U}), obtendo assim $\mathbf{u}_{\text{est}} = \mathbf{u}_{\text{bab}} = \lfloor \mathbf{u}_{\text{unc,opt}} \rfloor$ (DORFLING *et al.*, 2020; KARAMANAKOS *et al.*, 2016). A segunda técnica, implementada em Geyer e Quevedo (2014) e denominada de “palpite fundamentado”, consiste em utilizar a solução restrita não transformada encontrada pelo algoritmo na amostra anterior como palpite inicial na amostra atual, resultando então em $\mathbf{u}_{\text{est}} = \mathbf{u}_{\text{pf}} = \mathbf{u}_{\text{opt}}(k-1)$. Nota-se que esta última abordagem pode funcionar muito bem quando o sistema está em regime permanente, mas talvez deixe a desejar com o sistema no período transitório, quando o vetor ótimo de uma amostra pode ser distante da solução definida na amostra anterior. Por conta disto, em Dorfling *et al.* (2020) e Karamanakos *et al.* (2016), as duas técnicas são implementadas e são obtidos ρ_{bab} e ρ_{pf} , onde

$$\rho_{\text{bab}} = \|\mathbf{H}_{\text{sda}}\mathbf{u}_{\text{bab}} - \bar{\mathbf{u}}_{\text{unc,opt}}(k)\|_2^2, \quad (101)$$

$$\rho_{\text{pf}} = \|\mathbf{H}_{\text{sda}}\mathbf{u}_{\text{pf}} - \bar{\mathbf{u}}_{\text{unc,opt}}(k)\|_2^2. \quad (102)$$

Posteriormente, ρ_{ini} é definido conforme

$$\rho_{\text{ini}} = \min\{\rho_{\text{bab}}, \rho_{\text{pf}}\}. \quad (103)$$

O princípio de funcionamento deste algoritmo é considerar iterativamente as sequências candidatas que pertençam ao conjunto finito de sequências factíveis (\mathbb{U}) e que, após a transformação para o novo espaço formado, estejam dentro do raio da esfera centrada em $\bar{\mathbf{u}}_{\text{unc,opt}}(k)$, ou seja, satisfaçam

$$\|\mathbf{H}_{\text{sda}}\mathbf{u}(k) - \bar{\mathbf{u}}_{\text{unc,opt}}(k)\|_2^2 = \sigma \leq \rho(k). \quad (104)$$

Como \mathbf{H}_{sda} é uma matriz triangular inferior, (104) é equivalente à seguinte inequação

$$\begin{aligned} \rho(k)^2 \geq & \left(\mathbf{H}_{\text{sda}}^{(1,1)}\mathbf{u}^{(1)}(k) - \bar{\mathbf{u}}_{\text{unc,opt}}^{(1)}(k) \right)^2 + \left(\mathbf{H}_{\text{sda}}^{(2,1)}\mathbf{u}^{(1)}(k) + \mathbf{H}_{\text{sda}}^{(2,2)}\mathbf{u}^{(2)}(k) - \bar{\mathbf{u}}_{\text{unc,opt}}^{(2)}(k) \right)^2 \\ & + \dots + \left(\mathbf{H}_{\text{sda}}^{(3N,1)}\mathbf{u}^{(1)}(k) + \dots + \mathbf{H}_{\text{sda}}^{(3N,3N)}\mathbf{u}^{(3N)}(k) - \bar{\mathbf{u}}_{\text{unc,opt}}^{(3N)}(k) \right)^2 = \sigma^2, \end{aligned} \quad (105)$$

onde $\mathbf{H}_{\text{sda}}^{(i,j)}$ representa o elemento da i -ésima linha e da j -ésima coluna da matriz \mathbf{H}_{sda} , enquanto $\bar{\mathbf{u}}_{\text{unc,opt}}^{(i)}$ e $\mathbf{u}^{(i)}$ representam os elementos da i -ésima linha dos vetores $\bar{\mathbf{u}}_{\text{unc,opt}}$ e \mathbf{u} , para $\{i, j\} \in [1, 3N]$. A maior parte do esforço computacional requisitado pelo SDA é relacionado ao cômputo da distância calculada, descrito em (104). Todavia, com a utilização de (105), este cálculo pode ser feito sequencialmente, pois apenas as somas dos termos ao quadrado relacionadas ao i -ésimo componente de \mathbf{u} devem ser incrementadas, o que torna o cômputo mais leve (GEYER; QUEVEDO, 2014). Quando \mathbf{u} estiver completo e (105) for satisfeita, $\rho(k)^2$ passa a assumir o valor de σ^2 para as avaliações seguintes.

Nota-se que o algoritmo utiliza a solução irrestrita da função custo ($\mathbf{u}_{\text{unc,opt}}$) como heurística para resolver o problema restrito, através de uma busca que se baseia nos métodos de ramificar e limitar (em inglês, *Branch and Bound*). Os ramos são feitos a partir dos estados

disponíveis para cada fase. A limitação acontece ao considerar apenas as soluções que se encontram dentro da esfera centrada em $\bar{\mathbf{u}}_{\text{unc,opt}}$. Este tipo de técnica é muito mais eficiente em termos de tempo de cômputo do que o método de busca exaustiva utilizado pelo algoritmo de FCS-MPC padrão (GEYER; QUEVEDO, 2014), descrito na Subseção 2.3.2. Ao utilizá-la em cenários com horizontes longos, por exemplo, algumas opções não serão avaliadas pelo algoritmo, reduzindo o fardo computacional da técnica.

3.1.6 Algoritmo de decodificação via esferas: adaptações propostas

Neste trabalho, diferentemente da implementação em Geyer e Quevedo (2014), todas as variáveis do modelo são mantidas em $\alpha\beta$ como é padrão ao FCS-MPC. Isto significa que os vetores $\bar{\mathbf{u}}_{\text{unc,opt}}$, \mathbf{u}_{opt} e \mathbf{u} , utilizados no problema de mínimos quadrados inteiros, descrito em (99), têm suas dimensões alteradas de $3N$ para $2N$. Nota-se que considerar N para o tamanho desses vetores traz a ideia de que N e N_u possuem o mesmo comprimento. Neste trabalho, por mais que $N = N_u$ em todos os casos, considera-se que $\{\mathbf{u}, \mathbf{u}_{\text{opt}}\} \in \mathbb{U}^{2N_u}$ e $\bar{\mathbf{u}}_{\text{unc,opt}} \in \mathbb{R}^{2N_u}$ para deixar de forma mais genérica.

Por mais que \mathbf{u} esteja em $\alpha\beta$ para o cálculo de σ , para manter o bom funcionamento do SDA com números inteiros, dentro do algoritmo, as novas opções são formadas com \mathbf{u} em abc , como é feito em sua forma clássica. Portanto, antes que ocorra a transformação com a matriz \mathbf{H}_{sda} , é necessário realizar a transformada de Clarke no \mathbf{u}_{abc} gerado. Para que isso seja possível, a cada três camadas i passadas, quando as componentes das fases a , b e c já foram definidas para aquela amostra de predição avaliada dentro da sequência, \mathbf{u}_{abc} é transformado para $\mathbf{u}_{\alpha\beta}$. Então, a transformação para o espaço discreto onde está a solução ótima e o cálculo da distância para o vetor ótimo transformado podem ser executados, em $\alpha\beta$.

Por conta disso, a matriz \mathbf{H}_{sda} tem sua dimensão alterada para $\mathbf{H}_{\text{sda}} \in \mathbb{R}^{2N_u \times 2N_u}$. Além disso, a matriz não é mais completamente triangular inferior, pois passa a ser uma matriz do tipo bloco diagonal, com cada bloco não nulo sendo triangular inferior, ou seja,

$$\mathbf{H}_{\text{sda}} = \begin{bmatrix} \mathbf{T}_{\text{inf}} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{\text{inf}} \end{bmatrix}, \quad (106)$$

onde $\mathbf{T}_{\text{inf}} \in \mathbb{R}^{N_u \times N_u}$ é uma matriz triangular inferior e $\mathbf{0} \in \mathbb{N}^{N_u \times N_u}$ é uma matriz de zeros.

Como as componentes de α e β dependem da transformada de Clarke, a cada três camadas ambas são encontradas simultaneamente, formando um vetor $\mathbf{u}_{\alpha\beta}(k+l-1)$, para $l \in [1, N_u]$, que faz parte de uma sequência com N_u vetores. Isto significa que o cálculo sequencial da distância quadrática (σ^2) também é alterado. Na formulação clássica, apenas os termos presentes na equação (105) que multiplicam o sinal de controle da i -ésima camada ($\mathbf{u}^{(i)}$) são inseridos no cálculo. Nesta adaptação, são os termos que multiplicam os sinais de controle em $k+l-1$ que são inseridos, ou seja, $\mathbf{u}^{(l)}$ e $\mathbf{u}^{(l+N_u)}$.

Outra observação importante é que em sua formulação original, a matriz \mathbf{H}_{sda} e o vetor $\mathbf{u}_{\text{unc,opt}}$ são encontrados através do modelo em espaço de estados do sistema. Neste

trabalho, o vetor $\mathbf{u}_{\text{unc,opt}}$ pode ser encontrado através da solução irrestrita dos controladores GPC ou CARMA-GPC, utilizando o modelo CARIMA ou o modelo CARMA, respectivamente. O processo para alcançar esta solução é explicado na Subseção 2.3.3. Ao comparar (96) com (75) e (76), nota-se que \mathbf{H}_y e \mathbf{H}_r estão para \mathbf{Q} , assim como \mathbf{b}_y^T e \mathbf{b}_r^T estão para Θ . Portanto, a matriz \mathbf{H}_{sda} , que era encontrada por (97), passou a ser calculada por

$$\mathbf{H}_{\text{sda}}^{-1} \mathbf{H}_{\text{sda}}^{-T} = \mathbf{H}^{-1}, \quad (107)$$

onde $\mathbf{H} \in \{\mathbf{H}_y, \mathbf{H}_r\}$ depende do modelo utilizado. Vale ressaltar que, neste trabalho, as matrizes \mathbf{H} e $\mathbf{b}^T \in \{\mathbf{b}_y^T, \mathbf{b}_r^T\}$ foram formadas com a matriz bloco diagonal \mathbf{Q}_δ nula, fazendo com que a minimização da função custo não pondere o incremento de controle. Esta sintonia foi definida para que os resultados obtidos com esta técnica sejam equivalentes aos resultados encontrados pelo algoritmo FCS-MPC padrão e outros semelhantes.

Ao utilizar o modelo CARIMA, o algoritmo SDA também utiliza \mathbf{u}_{abc} para se beneficiar da estrutura com números inteiros e binários. Todavia, após a transformada de Clarke aplicada a cada três amostras, um passo extra é adicionado. O vetor $\mathbf{u}_{\alpha\beta}$ resultante é utilizado para encontrar $\Delta\mathbf{u}_{\alpha\beta}$, através de

$$\Delta\mathbf{u}_{\alpha\beta}(k+l-1) = \mathbf{u}_{\alpha\beta}(k+l-1) - \mathbf{u}_{\alpha\beta}(k+l-2). \quad (108)$$

Então, $\Delta\mathbf{u}_{\alpha\beta}(k+l-1)$ é inserido dentro do vetor que armazena as sequências avaliadas, $\Delta\mathbf{u}$, que posteriormente é transformado por \mathbf{H}_{sda} para $\overline{\Delta\mathbf{u}}$. Assim como é feito com o modelo CARMA, o procedimento para encontrar $\Delta\mathbf{u}_{\text{opt}}$ passa a ser encontrar o vetor $\Delta\mathbf{u}$ que minimize o problema

$$\Delta\mathbf{u}_{\text{opt}}(k) = \underset{\Delta\mathbf{u}(k) \forall \mathbf{u}(k) \in \mathbb{U}}{\text{argmin}} \quad \|\mathbf{H}_{\text{sda}} \Delta\mathbf{u}(k) - \overline{\Delta\mathbf{u}}_{\text{unc,opt}}(k)\|_2^2, \quad (109)$$

Claramente, a estrutura do SDA é bem distinta do algoritmo padrão para FCS-MPC. Deste modo, no momento de avaliar o esforço de cômputo, uma simples contagem de ciclos não é a medida mais justa, pois o esforço carregado por cada iteração pode ser bem diferente para cada algoritmo.

3.1.7 Vetores adjacentes pré-selecionados para algoritmo de decodificação via esferas

Uma contribuição inédita na literatura proposta por este trabalho consiste na combinação da técnica adaptada de vetores adjacentes com o SDA modificado, resultando no algoritmo denominado PAV-SDA. Como o algoritmo PAV-FCS-MPC resulta em soluções subótimas e o algoritmo SDA retorna soluções ótimas, a técnica PAV-SDA também encontra soluções subótimas.

Conforme descrito na Subseção 3.1.4, uma quantidade máxima de chaves que podem trocar de estado (Q_{max}) é definida. Com isso, para cada vetor avaliado, é visto quantas chaves devem alterar o seu estado (Q) para alcançar o vetor definido na amostra anterior. Esse valor é

fornecido através da Tabela 2 (p. 46). Caso $Q \leq Q_{\max}$, define-se que o vetor avaliado pertence ao subconjunto de vetores adjacentes e sua avaliação dentro do SDA é permitida pelo algoritmo proposto. Caso contrário, o vetor é desconsiderado. Vale ressaltar que, caso o vetor avaliado seja um dos vetores nulos, se $Q_{\text{null}} = 1$, o vetor é considerado dentro do subconjunto dos vetores adjacentes quando for o vetor nulo mais próximo, ou seja, apenas se $Q \leq 1$. Caso $Q_{\text{null}} = 2$, basta que $Q \leq Q_{\max}$.

Nota-se que a inicialização de ρ_{ini} não pode ser feita como estava sendo feita no SDA, pois não existe garantia que o método ρ_{bab} utilizará um \mathbf{u}_{bab} que esteja dentro do subconjunto de vetores adjacentes definidos pela parte do algoritmo relacionada ao PAV-FCS-MPC. Quanto ao ρ_{pf} , como \mathbf{u}_{pf} é sempre a sequência definida na amostra anterior, sabe-se que, garantidamente, cada opção está localizada no subconjuntos de vetores adjacentes do vetor considerado para a amostra de predição anterior.

3.2 CONSIDERAÇÕES

A diminuição do tempo de cômputo é um tópico em constante desenvolvimento na literatura e pode ser implementada de diversas maneiras. Duas linhas de pesquisa que sugerem uma pré-seleção das opções de controle que são consideradas pelo FCS-MPC foram avaliadas neste trabalho, com o intuito de que menos iterações sejam necessárias para encontrar a sequência ótima de controle. Uma das linhas aborda um método iterativo, aplicado para horizontes de uma amostra, que não precisa considerar todas as opções para obter o resultado ótimo. A outra abordagem considera apenas vetores adjacentes como opções do algoritmo de busca do FCS-MPC. Além disso, o algoritmo de decodificação via esferas foi adaptado para os controladores GPC e CARMA-GPC irrestritos, com o intuito de avaliar o impacto que as podas inteligentes causam no esforço computacional e no rastreamento de referências. Resumidamente, as adaptações propostas para cada uma destas técnicas da literatura são descritas abaixo:

- FCS-MPC-SCS:
 - adaptação para o VSI trifásico de dois níveis;
 - criação de mais uma etapa para deixar o algoritmo mais leve.

- PAV-FCS-MPC:
 - adaptação para o VSI trifásico de dois níveis;
 - compensação do atraso de cômputo;
 - extrapolação para horizontes maiores;
 - tabela de pesquisa para obter Q ;
 - inserção dos conceitos Q_{\max} e Q_{null} .

- SDA:
 - adaptação para o VSI trifásico de dois níveis;
 - compensação do atraso de cômputo;
 - solução irrestrita oriunda do GPC/CARMA-GPC;
 - transformada de Clarke e modelo de predição em $\alpha\beta$.

Adicionalmente, foi proposto um algoritmo que combina a abordagem de vetores adjacentes com o algoritmo de decodificação via esferas. Os resultados obtidos com todas estas técnicas são discutidos no Capítulo 4.

4 RESULTADOS E ANÁLISES

Neste capítulo, são apresentados os resultados obtidos via simulação entre os diferentes algoritmos de FCS-MPC abordados no Capítulo 3. Todas as simulações foram executadas na ferramenta Simulink do software MATLAB[®]. O sistema controlado é um VSI trifásico de dois níveis, apresentado na Figura 2 (Seção 2.1, p. 21). Os parâmetros utilizados em todas as simulações são apresentados na Seção 4.1. Os resultados obtidos por cada técnica em Regime Permanente (RP) são apresentados na Seção 4.2. As análises comparativas desses resultados podem ser vistas na Seção 4.3. Por fim, a Seção 4.4 apresenta a resposta temporal das técnicas para possibilitar a avaliação dos Regimes Transitórios (RT).

4.1 ASPECTOS GERAIS

Os valores dos parâmetros utilizados neste sistema para todas as simulações são descritos na Tabela 3. O inversor opera com uma tensão no barramento CC (V_{cc}) de 400 V. Do outro lado, deve rastrear uma referência em forma de onda senoidal trifásica com tensão eficaz ($V_{w,rms}$) de 120 V e frequência (F_w) de 50 Hz. A frequência de amostragem (F_s) foi configurada para 40 kHz, resultando em um período de amostragem (T_s) disponível para o controle de 25 μ s. Essas configurações são semelhantes ao que foi apresentado em Bordons e Montero (2015).

Tabela 3 – Parâmetros do sistema controlado utilizados em todas as simulações.

Parâmetro	Descrição	Valor
V_{cc}	Tensão no barramento CC	400 V
R_{sim}	Valor da resistência da carga na simulação	54 Ω
L_{sim}	Valor da indutância do filtro na simulação	1,8 mH
C_{sim}	Valor da capacitância do filtro na simulação	45 μ F
R_{mod}	Valor da resistência da carga no modelo de predição	60 Ω
L_{mod}	Valor da indutância do filtro no modelo de predição	2 mH
C_{mod}	Valor da capacitância do filtro no modelo de predição	50 μ F
V_n	Variância de tensão na medição ruidosa	2 V
F_s	Frequência de amostragem do controlador	40 kHz
$V_{w,rms}$	Tensão eficaz da referência	120 V
F_w	Frequência da referência	50 Hz
T_{sim}	Período total da simulação	0,06 s
T_{ana}	Período da simulação submetido a análises	0,02 s
N_1	Quantas amostras à frente está o início do horizonte de predição	2
N_{u1}	Quantas amostras à frente está o início do horizonte de controle	1

Existe uma diferença de 10% entre os valores dos componentes utilizados na simulação (R_{sim} , L_{sim} e C_{sim}) e aqueles utilizados no modelo de predição (R_{mod} , L_{mod} e C_{mod}), de modo que seja possível avaliar o desempenho dos algoritmos quando existem erros de modelagem. Uma medição ruidosa de média nula e variância V_n foi considerada na saída do sistema. Além

disso, como comentado na Subseção 2.3.2, $N_1 = 2$ para compensar o atraso adicionado pelo custo computacional e, portanto, $N_{u1} = 1$.

Com o objetivo de medir a qualidade da resposta em malha fechada obtida, a DHT e o erro médio quadrático (MSE, do inglês *Mean Squared Error*) da saída em RP são utilizados. O custo computacional para adquirir esse desempenho é medido através do tempo de cômputo médio (T_c) necessário para o algoritmo do FCS-MPC em cada amostra de controle. Com uso do comando *tic-toc*, calcula-se o tempo de cômputo do algoritmo. Durante o tempo utilizado para análises (T_{ana}), é calculado o tempo médio de cômputo daquela simulação. Esse processo é repetido para 10 simulações e, então, T_c assume a média dos valores médios calculados. Estas simulações são rodadas em um computador com um processador Intel i7-13700 com 2,10 GHz e 32 GB de RAM. É importante destacar que, mesmo se esse valor for maior que o período de amostragem disponível para o controlador, isso não significa que a técnica não pode ser implementada, uma vez que as implementações foram realizadas em computador pessoal, com sistema operacional Windows e outras tarefas síncronas que requerem um esforço adicional. Deste modo, é recomendado que a análise de T_c seja feita através do seu aumento ou decréscimo relativo, que é diretamente relacionado ao fardo computacional do algoritmo. Todavia, nos casos em que o tempo de cômputo do algoritmo ultrapassa o período de amostragem disponível ao controlador, os valores de T_c são destacados na cor vermelha.

O tempo total de simulação (T_{sim}) utilizado corresponde a 3 ciclos de referência. Então, como um período de referência é igual a 0,02 s, T_{sim} assume 0,06 s. Todavia, para avaliar a resposta em RP do sistema, os dois primeiros ciclos de referência são descartados. Por isso, T_{ana} é definido em 0,02 s. Na Figura 12, é possível visualizar onde todos estes parâmetros apresentados na Tabela 3 foram empregados dentro do ambiente de simulação.

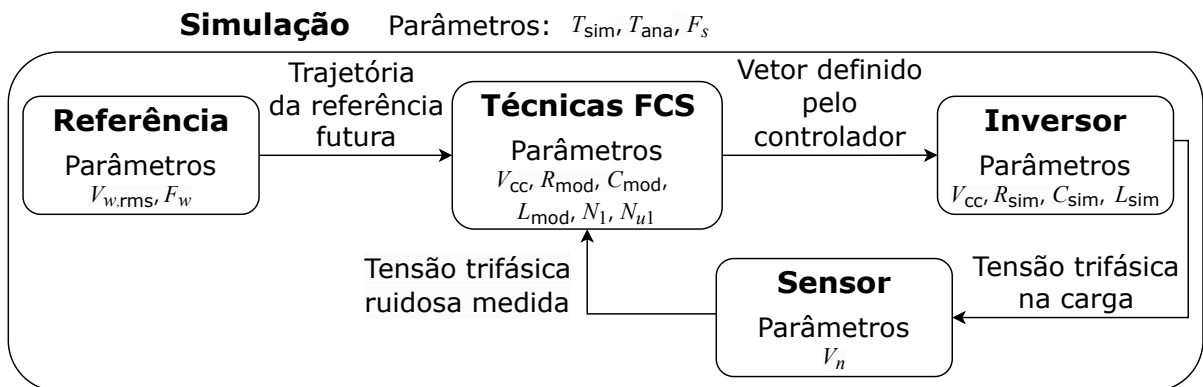


Figura 12 – Diagrama resumido da simulação com a localização dos parâmetros descritos na Tabela 3.

4.2 ANÁLISE DO REGIME PERMANENTE

A primeira técnica avaliada foi o FCS-MPC padrão, através de três variantes, que se distinguem pelos modelos de predição utilizados. A primeira variante realiza suas predições por

uma equação a diferenças, a segunda pelo modelo CARIMA e a terceira pelo modelo CARMA. Todos estes modelos de predição são apresentados na Subseção 2.3.1. Os algoritmos foram executados com diferentes tamanhos de horizonte de predição e seus resultados podem ser vistos nas Tabelas 4, 5 e 6. Vale ressaltar que, neste trabalho, os horizontes de controle e de predição são definidos com o mesmo tamanho.

Tabela 4 – FCS-MPC padrão com predição feita por equação a diferenças.

N	DHT (%)	Dif. DHT (%)	MSE (V^2)	Dif. MSE (%)	Iter. (-)	T_c (ms)	Dif. T_c (x)
1	2,53	-	18,23	-	8	0,003	-
2	1,42	-44	2,91	-84	128	0,014	4,7
3	1,36	-4	2,62	-10	1536	0,123	8,8
4	1,33	-2	2,64	1	16 384	1,220	9,9
5	1,32	-1	2,58	-2	163 840	13,843	11,3

Tabela 5 – FCS-MPC padrão com predição feita por modelo CARIMA.

N	DHT (%)	Dif. DHT (%)	MSE (V^2)	Dif. MSE (%)	Iter. (-)	T_c (ms)	Dif. T_c (x)
1	2,35	-	11,11	-	8	0,003	-
2	1,62	-31	5,00	-55	64	0,015	5,0
3	1,52	-6	4,03	-19	512	0,073	4,9
4	1,51	-1	3,92	-3	4096	0,776	10,6
5	1,51	0	3,96	1	32 768	11,424	14,7

Tabela 6 – FCS-MPC padrão com predição feita por modelo CARMA.

N	DHT (%)	Dif. DHT (%)	MSE (V^2)	Dif. MSE (%)	Iter. (-)	T_c (ms)	Dif. T_c (x)
1	2,53	-	18,23	-	8	0,003	-
2	1,42	-44	2,91	-82	64	0,010	3,3
3	1,36	-4	2,62	-10	512	0,072	7,2
4	1,33	-2	2,64	1	4096	0,754	10,5
5	1,32	-1	2,58	-2	32 768	10,960	14,5

Nota-se que, para $N = 1$, o algoritmo com a predição feita com o modelo CARIMA apresenta os melhores índices de desempenho para o seguimento de referência, 2,35% de DHT e $11,11 V^2$ de MSE, que representam aproximadamente uma redução de 7% na DHT e 39% no MSE obtidos com os outros modelos, enquanto o tempo de cômputo médio necessário foi praticamente o mesmo. Entretanto, para $N > 1$, o desempenho do modelo CARIMA passa a ser pior que com os outros dois modelos. Isso acontece pois, como discutido na Subseção 2.3.1, o modelo é projetado com um integrador discreto em sua formulação, o que o torna adequado para sinais de referência e perturbações do tipo degrau e, portanto, ao trabalhar com o sistema de coordenadas em $\alpha\beta$, as características do modelo podem atrapalhar o seguimento das

referências senoidais. Deste modo, por mais que a inclusão do integrador contribua em $N = 1$ para lidar com os erros de modelagem, ao aumentar o tamanho do horizonte os outros modelos conseguem compensar a ausência do integrador e melhoram o desempenho em relação ao CARIMA. Outro ponto negativo do modelo evidenciado na Tabela 5 é que o T_c necessário para a técnica é maior do que o tempo gasto pelo modelo CARMA. Neste caso, isto se deve ao fato de que a variável manipulada utilizada para os cálculos de predição com este modelo é o incremento de controle, então, partindo do conjunto finito de controle do inversor, um passo adicional deve ser executado para encontrar o incremento que aquele vetor representa em comparação ao vetor considerado no instante de tempo anterior ao avaliado, o que gera um custo computacional extra para o modelo.

Também é possível visualizar que o código com predição feita por equação a diferenças traz os mesmos índices de rastreamento de referência do algoritmo com o modelo CARMA, o que está de acordo com o que foi explicado na Subseção 2.3.1. Todavia, como o algoritmo com equação a diferenças calcula a predição da saída iterativamente, sem o agrupamento feito pela representação matricial utilizado nas outras duas variantes, mais iterações são necessárias nos casos nos quais $N > 1$. Essas iterações são mais leves que as iterações dos algoritmos com CARMA e CARIMA, mas ainda fazem com que, nesses casos com horizontes mais compridos, essa variante seja mais lenta do que as outras. Deste modo, para $N > 1$, o algoritmo com predição feita por CARMA demonstra ser a opção com melhor rastreamento de referência e menor tempo de cômputo gasto.

Com ambos os modelos, o aumento do horizonte de predição melhora o seguimento de referência em detrimento do aumento do tempo de cômputo. Para facilitar a visualização do impacto do aumento dos horizontes, foram incluídas junto aos índices DHT, MSE e T_c as suas respectivas diferenças de proporção com relação aos valores encontrados por $N-1$. Por exemplo, na Tabela 6 é descrito que, para o modelo CARMA, aumentar N de 1 para 2 amostras ocasiona uma redução de 44% de DHT (2,53% para 1,42%), 82% de MSE ($18,23 V^2$ para $2,91 V^2$), ao custo de deixar o tempo de cômputo do código aproximadamente 3,3 vezes maior (0,003 ms para 0,010 ms). Portanto, ao analisar as Tabelas 4, 5 e 6 através destas métricas, percebe-se que $N=2$ traz o melhor custo benefício, visto que tem o maior ganho de desempenho sobre o menor aumento do tempo de cômputo. Além disso, os valores em vermelho nas três tabelas destacam que, com todos os modelos avaliados, o emprego de horizontes com três amostras de comprimento ou mais seria impossível para um hardware com poder de cômputo semelhante ao computador que rodou as simulações.

A segunda técnica implementada foi o FCS-MPC-SCS. Nesta técnica, como a predição por equação a diferenças atinge os mesmos resultados de CARMA de forma mais lenta, essa variante não foi considerada neste algoritmo. Além disso, conforme descrito na Subseção 3.1.2, essa técnica não foi generalizada para horizontes maiores. Portanto, a Tabela 7 descreve os resultados obtidos com esta técnica utilizando os modelos de predição CARMA e CARIMA, para $N = 1$.

Tabela 7 – FCS-MPC-SCS para $N = 1$.

Mod. pred.	DHT (%)	MSE (V^2)	Iter. (-)	T_c (ms)
CARMA	2,53	18,23	4,67	0,000 474
CARIMA	2,35	11,11	4,67	0,000 557

Percebe-se que esta técnica é capaz de encontrar a solução ótima do problema de minimização, uma vez que resulta nos mesmos índices de rastreamento de referência encontrados com a busca exaustiva utilizada no algoritmo padrão de FCS-MPC, ou seja, 2,53% de DHT e $18,23V^2$ de MSE com o modelo CARMA e 2,35% de DHT e $11,11V^2$ de MSE com o CARIMA. Além disso, por precisar de menos iterações para encontrar a solução ótima e não aumentar de forma expressiva a complexidade do código, o algoritmo reduz cerca de 80% do tempo de cômputo gasto pelo algoritmo padrão de FCS-MPC, pois esse passa de 0,003 ms para aproximadamente 0,0006 ms.

O terceiro algoritmo implementado foi o PAV-FCS-MPC. Este algoritmo substitui o conjunto finito de controle do algoritmo convencional de FCS-MPC por um subconjunto com apenas vetores próximos ao último vetor aplicado. Por ser a opção com melhor custo-benefício ao ampliar os horizontes, o modelo CARMA foi utilizado para fazer a predição da saída com esta técnica. Os resultados obtidos com este algoritmo são encontrados na Tabela 8 e na Tabela 9.

Tabela 8 – PAV-FCS-MPC com predição feita por modelo CARMA - Parte 1.

N	Q_{\max}	Q_{null}	DHT (%)	Dif. DHT (%)	MSE (V^2)	Dif. MSE (%)	T_c (ms)
1	1	1	10,81	-	188,48	-	0,002
1	2	1	3,54	-	41,30	-	0,002
1	2	2	3,54	-	41,30	-	0,002
2	1	1	4,08	-62	30,87	-84	0,005
2	2	1	1,54	-56	3,38	-92	0,009
2	2	2	1,54	-56	3,38	-92	0,011
3	1	1	1,81	-56	6,38	-79	0,011
3	2	1	1,34	-13	2,66	-21	0,035
3	2	2	1,34	-13	2,66	-21	0,047
4	1	1	1,62	-10	3,53	-45	0,049
4	2	1	1,33	-1	2,60	-2	0,256
4	2	2	1,33	-1	2,60	-2	0,420
5	1	1	1,56	-4	3,53	0	0,267
5	2	1	1,32	-1	2,55	-2	2,194
5	2	2	1,32	-1	2,55	-2	4,118

Vale ressaltar que os valores encontrados através da combinação $Q_{\max} = 1$ e $Q_{\text{null}} = 1$ são idênticos aos resultados alcançados com $Q_{\max} = 1$ e $Q_{\text{null}} = 2$, visto que não existem 2 vetores nulos no subconjunto de vetores adjacentes ao definir $Q_{\max} = 1$. Por isso, esta combinação foi desconsiderada nesta seção.

Tabela 9 – PAV-FCS-MPC com previsão feita por modelo CARMA - Parte 2.

N	Q_{\max}	Q_{null}	DHT (%)	MSE (V^2)	Iter. (-)	T_c (ms)	Dif. T_c (\times)
1	1	1	10,81	188,48	4	0,002	-
1	2	1	3,54	41,30	6	0,002	-
1	2	2	3,54	41,30	7	0,002	-
2	1	1	4,08	30,87	16	0,005	2,5
2	2	1	1,54	3,38	38	0,009	4,5
2	2	2	1,54	3,38	49	0,011	5,5
3	1	1	1,81	6,38	64	0,011	2,2
3	2	1	1,34	2,66	233	0,035	3,9
3	2	2	1,34	2,66	343	0,047	4,3
4	1	1	1,62	3,53	256	0,049	4,5
4	2	1	1,33	2,60	1437	0,256	7,3
4	2	2	1,33	2,60	2401	0,420	8,9
5	1	1	1,56	3,53	1024	0,267	5,4
5	2	1	1,32	2,55	8877	2,194	8,6
5	2	2	1,32	2,55	16 807	4,118	9,8

Diferente do FCS-MPC-SCS, o resultado obtido com este algoritmo não vai ao encontro daquele obtido pelo FCS-MPC padrão. Isto acontece porque não é em todos os casos que a solução ótima está dentro do subconjunto de vetores adjacentes disponíveis. Portanto, o algoritmo PAV-FCS-MPC fornece uma solução subótima para o problema. Para $N = 1$, independente da configuração utilizada, o rastreamento de referência é bem comprometido. Por exemplo, para o melhor dos casos, ou seja, $Q_{\max} = 2$ e $Q_{\text{null}} = \{1,2\}$, um aumento de aproximadamente 127% no MSE, de $18,23 V^2$ para $41,30 V^2$, e 40% na DHT, de $18,23 V^2$ para $41,30 V^2$, são obtidos em relação ao algoritmo FCS-MPC padrão. Entretanto, para horizontes maiores, existem cenários em que utilizar esta técnica resulta em melhores índices do que utilizar o FCS-MPC padrão, pois nem sempre a solução ótima para o problema teórico resulta na solução ótima do problema prático, que possui erros de modelagem e ruídos medidos na saída do sistema. Deste modo, a solução subótima não é necessariamente um problema para o sistema controlado.

A configuração $Q_{\max} = 1$ e $Q_{\text{null}} = 1$ para $N > 1$ resulta em tempos de cômputo menores, ao custo de piorar os índices de desempenho MSE e DHT. Para $N = 3$, apenas esta sintonia se encaixa dentro do período de amostragem disponível, pois requer aproximadamente $11 \mu s$ de T_c . Contudo, a utilização de $Q_{\max} = 2$ em $N = 2$ resulta em 15% menos DHT, de 1,81% para 1,54%, e 47% menos MSE, de $6,38 V^2$ para $3,38 V^2$, o que faz com que a combinação de $Q_{\max} = 1$, $Q_{\text{null}} = 1$ e $N = 3$ não seja uma boa alternativa. Ampliando o comprimento do horizonte para $N = 5$, a sintonia com $Q_{\max} = 1$ passou a ser mais competitiva com $Q_{\max} = 2$ para o mesmo comprimento de horizonte, pois por mais que exista um aumento de 38% do MSE, de $2,55 V^2$ para $3,53 V^2$, e 18% da DHT, de 1,32% para 1,56%, o tempo de cômputo necessário é reduzido em no mínimo 88%, de 2,194 ms para 0,267 ms. Todavia, este seu melhor caso ainda é pior

do que a sintonia $Q_{\max} = 2$ para $N = 2$, pois além de possuir aproximadamente 5% mais MSE e 1% mais DHT, o seu T_c é no mínimo 24 vezes maior, de 0,011 ms para 0,267 ms. Portanto, conclui-se que a configuração $Q_{\max} = 1$ e $Q_{\text{null}} = 1$ não possui um bom custo-benefício.

Nota-se que a configuração $Q_{\max} = 2$ e $Q_{\text{null}} = 1$ resulta nos mesmos índices de desempenhos obtidos por $Q_{\max} = 2$ e $Q_{\text{null}} = 2$, entretanto, custando menos T_c para a plataforma de controle, em qualquer tamanho de horizonte. Isso acontece pois os vetores nulos são redundantes, ou seja, independente do vetor nulo implementado a saída vai para o mesmo ponto. Todavia, é importante destacar que por mais que os vetores sejam redundantes, deixar apenas um deles disponível sem avaliar a distância para o vetor anterior pode resultar em perdas de potência desnecessárias. Nesta implementação, como o vetor nulo disponível para $Q_{\text{null}} = 1$ é sempre o mais próximo do vetor anterior, estas comutações desnecessárias não acontecem. Deste modo, $Q_{\max} = 2$ e $Q_{\text{null}} = 1$ demonstra ser a melhor configuração para ser implementada com esta técnica.

Nas Tabelas 8 e 9, as métricas de diferença de proporção entre os índices DHT, MSE e T_c também foram apresentadas para a análise do impacto do aumento dos horizontes, de forma semelhante ao que foi feito nas Tabelas 4 a 6. Entretanto, é importante destacar que nesta técnica as diferenças de proporção foram calculadas entre as configurações com mesmos valores de Q_{\max} e Q_{null} , mas mantendo a comparação apenas entre N e $N-1$. Por exemplo, ao ampliar o horizonte de $N = 1$ para $N = 2$, com a sintonia $Q_{\max} = 2$ e $Q_{\text{null}} = 1$, a DHT reduz 56%, de 3,54% para 1,54%, o MSE reduz 92%, de $41,30 V^2$ para $3,38 V^2$, e o tempo de cômputo é aproximadamente 4,5 vezes maior. Então, com o auxílio destas métricas, é possível constatar que, novamente, $N = 2$ traz o melhor custo benefício entre melhoria de desempenho e aumento do tempo de cômputo. Todavia, como T_c aumenta de forma mais lenta do que com o FCS-MPC padrão, $N = 3$ pode ser uma boa alternativa para os casos nos quais a minimização dos índices de desempenho do rastreamento de referência tem mais importância que a redução do tempo de cômputo, pois, comparado com $N = 2$, reduz 13% da DHT, de 1,54% para 1,34%, e 21% do MSE, de $3,38 V^2$ para $2,66 V^2$, ao custo de aumentar em 3,9 vezes o T_c , de 0,009 ms para 0,035 ms. Para $N > 3$, a baixa melhoria de desempenho não justifica o alto incremento do tempo de cômputo.

A última técnica adaptada da literatura foi o SDA. Três formas distintas foram implementadas para cada horizonte de predição, com o raio inicial ρ_{ini} sendo ρ_{bab} , ρ_{pf} , ou calculando os dois e escolhendo o menor entre eles. Os resultados obtidos com esta técnica para os modelos CARMA e CARIMA são descritos nas Tabelas 10 e 11.

Assim como na técnica FCS-MPC-SCS, a solução encontrada é a mesma apresentada pelo FCS-MPC padrão, ou seja, a solução é ótima. Como o cômputo dos raios iniciais é feito on-line e existe um custo computacional associado a estes cálculos, para $N \leq 2$, com ambos os modelos, $\rho_{\text{ini}} = \min\{\rho_{\text{bab}}, \rho_{\text{pf}}\}$ é a alternativa mais lenta entre as três opções. Como a escolha do raio inicial não interfere na solução ótima encontrada, esta alternativa é a que apresenta o pior custo benefício para horizontes de até duas amostras, mesmo diminuindo a quantidade de

Tabela 10 – SDA com predição feita por modelo CARMA.

N	ρ_{ini}	DHT (%)	MSE (V^2)	Iter. (-)	T_c (ms)	Dif. T_c (\times)
1	ρ_{bab}	2,53	18,23	8	0,003	-
1	ρ_{pf}	2,53	18,23	8	0,003	-
1	$\min\{\rho_{bab}, \rho_{pf}\}$	2,53	18,23	8	0,004	-
2	ρ_{bab}	1,42	2,91	41	0,010	3,3
2	ρ_{pf}	1,42	2,91	40	0,010	3,3
2	$\min\{\rho_{bab}, \rho_{pf}\}$	1,42	2,91	37	0,011	2,8
3	ρ_{bab}	1,36	2,62	144	0,033	3,3
3	ρ_{pf}	1,36	2,62	130	0,032	3,2
3	$\min\{\rho_{bab}, \rho_{pf}\}$	1,36	2,62	119	0,029	2,6
4	ρ_{bab}	1,33	2,64	455	0,096	2,9
4	ρ_{pf}	1,33	2,64	369	0,079	2,5
4	$\min\{\rho_{bab}, \rho_{pf}\}$	1,33	2,64	333	0,070	2,4
5	ρ_{bab}	1,32	2,58	1303	0,274	2,9
5	ρ_{pf}	1,32	2,58	942	0,199	2,5
5	$\min\{\rho_{bab}, \rho_{pf}\}$	1,32	2,58	862	0,184	2,6

iterações. Conforme o horizonte aumenta, a melhoria provocada pela escolha desta alternativa fica mais evidente pois a diferença de iterações que devem ser computadas aumenta. Deste modo, para $N \geq 3$, a melhor opção passa a ser definir o raio inicial como o menor raio entre ρ_{bab} e ρ_{pf} . Até $N \leq 2$, a diferença de tempo entre ρ_{bab} e ρ_{pf} é mínima e a escolha da técnica pode depender do sistema controlado. Para um sistema com variações bruscas e frequentes na referência, ρ_{bab} pode ser a melhor opção. Neste caso, como a referência opera em 50 Hz e o período de amostragem é de 25 μ s, a melhor alternativa passa a ser ρ_{pf} , pois a solução anterior normalmente será uma boa estimativa para a solução atual.

O aumento do tamanho dos horizontes para o SDA resulta nas mesmas melhorias dos índices de desempenho obtidas com o algoritmo de FCS-MPC padrão. Portanto, nas Tabelas 10 e 11, as métricas de diferença percentual entre os diferentes tamanhos de horizontes para os índices DHT e MSE foram retiradas, visto que são os mesmos valores apresentados nas Tabelas 5 e 6. Todavia, para avaliar o impacto causado no tempo de cômputo ao aumentar o tamanho do horizonte com o método de busca utilizado pelo SDA, a métrica de diferença proporcional do T_c , apresentada para os métodos FCS-MPC padrão e PAV-FCS-MPC, também é avaliada neste cenário. Por exemplo, para $\rho_{ini} = \min\{\rho_{bab}, \rho_{pf}\}$ e com o modelo de predição CARMA, ao aumentar o horizonte de $N = 1$ para $N = 2$, o tempo de cômputo aumenta 2,8 vezes, de 0,004 ms para 0,011 ms. Mantendo a mesma sintonia e definindo $N = 3$, o T_c aumenta 2,6 vezes, de 0,011 ms para 0,029 ms. É possível observar que o princípio de ramificações e podas do SDA faz com que o impacto causado no tempo de cômputo do código seja mais suave, fazendo com que a alternativa $N = 3$ passe a oferecer um balanço competitivo. Entretanto, a métrica demonstra que utilizar $N = 2$ ainda é a melhor opção custo-benefício para o SDA. Adicionalmente, observa-se que com esta técnica, independente do modelo, horizontes maiores

Tabela 11 – SDA com predição feita por modelo CARIMA.

N	ρ_{ini}	DHT (%)	MSE (V^2)	Iter.(-)	T_c (ms)	Dif. T_c (\times)
1	ρ_{bab}	2,35	11,11	8	0,004	-
1	ρ_{pf}	2,35	11,11	8	0,004	-
1	$\min\{\rho_{bab}, \rho_{pf}\}$	2,35	11,11	8	0,004	-
2	ρ_{bab}	1,62	5,00	49	0,013	3,3
2	ρ_{pf}	1,62	5,00	48	0,013	3,3
2	$\min\{\rho_{bab}, \rho_{pf}\}$	1,62	5,00	46	0,013	3,3
3	ρ_{bab}	1,52	4,03	195	0,046	3,5
3	ρ_{pf}	1,52	4,03	162	0,039	3,0
3	$\min\{\rho_{bab}, \rho_{pf}\}$	1,52	4,03	154	0,038	2,9
4	ρ_{bab}	1,51	3,92	680	0,157	3,4
4	ρ_{pf}	1,51	3,92	509	0,118	3,0
4	$\min\{\rho_{bab}, \rho_{pf}\}$	1,51	3,92	482	0,110	2,9
5	ρ_{bab}	1,51	3,96	2276	0,586	3,7
5	ρ_{pf}	1,51	3,96	1417	0,340	2,9
5	$\min\{\rho_{bab}, \rho_{pf}\}$	1,51	3,96	1327	0,316	2,9

que duas amostras também ultrapassam o período de amostragem disponível ao controlador.

Por fim, o último algoritmo avaliado é o PAV-SDA, proposto de forma inédita neste trabalho ao combinar as técnicas PAV-FCS-MPC e SDA. Os resultados obtidos com esta técnica utilizando o modelo de predição CARMA, para diferentes valores de Q_{max} , Q_{null} e N , são descritos na Tabela 12.

Tabela 12 – PAV-SDA com predição feita por modelo CARMA

N	Q_{max}	Q_{null}	DHT (%)	MSE (V^2)	Iter. (-)	T_c (ms)	Dif. T_c (\times)
1	1	1	10,81	188,48	8	0,003	-
1	2	1	3,54	41,30	8	0,003	-
1	2	2	3,54	41,30	8	0,004	-
2	1	1	4,08	30,87	35	0,008	2,7
2	2	1	1,54	3,38	36	0,009	3,0
2	2	2	1,54	3,38	39	0,010	2,5
3	1	1	1,81	6,38	86	0,016	2,0
3	2	1	1,34	2,66	105	0,025	2,8
3	2	2	1,34	2,66	113	0,026	2,6
4	1	1	1,62	3,53	176	0,031	1,9
4	2	1	1,33	2,60	265	0,056	2,2
4	2	2	1,33	2,60	310	0,068	2,6
5	1	1	1,56	3,53	366	0,063	2,0
5	2	1	1,32	2,55	614	0,131	2,3
5	2	2	1,32	2,55	746	0,165	2,4

Vale ressaltar que, diferentemente do que ocorre no algoritmo SDA, não é recomendado utilizar $\rho_{ini} = \rho_{bab}$ nesta técnica, pois não é garantido que u_{bab} esteja dentro do subconjunto

de vetores adjacentes, o que poderia resultar em um problema sem solução. Deste modo, na Tabela 12, apenas $\rho_{ini} = \rho_{pf}$ é utilizado. Além disso, nota-se que a técnica PAV-SDA encontra as mesmas soluções obtidas por PAV-FCS-MPC, com um tempo de cômputo reduzido. Portanto, assim como para o SDA, apenas a diferença proporcional do tempo de cômputo foi apresentada.

Novamente, através da métrica o impacto do aumento dos horizontes pode ser avaliado entre as sintonias. Para $Q_{max} = 2$ e $Q_{null} = 1$, por exemplo, o aumento de $N = 1$ para $N = 2$, faz com que o tempo de cômputo fique 3,0 vezes maior, passando de 0,003 ms para 0,009 ms. Para $N = 3$, o tempo de cômputo aumenta 2,8 vezes, pois sobe para 0,025 ms. Como este tempo não ultrapassa o período de amostragem, por mais que $N = 2$ ainda seja a melhor opção, $N = 3$ passa a ser bem factível e pode ser uma boa escolha caso o seguimento de referência seja um requisito crítico ao projeto e haja um hardware com poder de cômputo semelhante à plataforma de simulação.

4.3 COMPARAÇÃO ENTRE OS ÍNDICES DE REGIME PERMANENTE

Nesta seção, algumas análises comparativas são feitas entre os algoritmos. Primeiramente, os melhores resultados obtidos por cada uma das técnicas com o modelo CARMA e CARIMA, para $N = 1$, são encontrados nas Tabelas 13 e 14, respectivamente. Para os algoritmos PAV-FCS-MPC e PAV-SDA, a configuração $Q_{max} = 2$ e $Q_{max} = 1$ foi selecionada, enquanto o SDA utiliza $\rho_{ini} = \rho_{pf}$.

Tabela 13 – Comparação do MSE, DHT e T_c entre as técnicas para $N = 1$ para o modelo CARMA.

Algoritmo	DHT (%)	MSE (V^2)	T_c (μs)
FCS-MPC	2,53	18,23	2,810
FCS-MPC-SCS	2,53	18,23	0,474
PAV-FCS-MPC	3,54	41,30	2,134
SDA	2,53	18,23	3,347
PAV-SDA	3,54	41,30	3,417

Tabela 14 – Comparação do MSE, DHT e T_c entre as técnicas para $N = 1$ para o modelo CARIMA.

Algoritmo	DHT (%)	MSE (V^2)	T_c (μs)
FCS-MPC	2,35	11,11	2,595
FCS-MPC-SCS	2,35	11,11	0,557
SDA	2,35	11,11	4,130

Nota-se que, para este cenário com apenas horizontes de uma amostra, as combinações do modelo de previsão CARIMA com o algoritmo FCS-MPC padrão e as técnicas SDA e FCS-MPC-SCS alcançaram os menores valores de DHT e MSE entre todas as técnicas avaliadas. Porém, o melhor custo-benefício é obtido com o algoritmo FCS-MPC-SCS, pois opera de forma

mais rápida, com 21% do T_c necessário pelo FCS-MPC convencional, de aproximadamente $2,6 \mu\text{s}$ para $0,6 \mu\text{s}$ e 13% do T_c do SDA, cujo valor aproximado é de $4,1 \mu\text{s}$. Em $N = 1$ é o único cenário que qualquer técnica obtém resultados melhores com o modelo de predição CARIMA em relação com o modelo de predição CARMA.

Ao manter $2 \leq N \leq 5$, os melhores resultados obtidos são descritos na Tabela 15. Neste caso, para $N = 2$, a técnica SDA utiliza $\rho_{\text{ini}} = \rho_{\text{pf}}$, mas para $N \geq 3$, passa a ser $\rho_{\text{ini}} = \min\{\rho_{\text{bab}}, \rho_{\text{pf}}\}$. As técnicas PAV-FCS-MPC e PAV-SDA continuam com $Q_{\text{max}} = 2$ e $Q_{\text{max}} = 1$.

Tabela 15 – Comparação do MSE, DHT e T_c entre as técnicas para $N > 1$.

Algoritmo	N	DHT (%)	MSE (V^2)	T_c (ms)
FCS-MPC	2	1,42	2,91	0,010
PAV-FCS-MPC	2	1,54	3,38	0,009
SDA	2	1,42	2,91	0,010
PAV-SDA	2	1,54	3,38	0,009
FCS-MPC	3	1,36	2,62	0,072
PAV-FCS-MPC	3	1,34	2,66	0,035
SDA	3	1,36	2,62	0,029
PAV-SDA	3	1,34	2,66	0,025
FCS-MPC	4	1,33	2,64	0,754
PAV-FCS-MPC	4	1,33	2,60	0,256
SDA	4	1,33	2,64	0,070
PAV-SDA	4	1,33	2,60	0,056
FCS-MPC	5	1,32	2,58	10,960
PAV-FCS-MPC	5	1,32	2,55	2,194
SDA	5	1,32	2,58	0,184
PAV-SDA	5	1,32	2,55	0,131

Os resultados apresentados nas Tabelas 13 e 15 também podem ser visualizados através dos gráficos presentes nas Figuras 13 e 14, que apresentam comparações entre o aumento do tempo de cômputo médio, gasto por cada técnica com o modelo CARMA ao aumentar o tamanho do horizonte, e a redução do MSE e da DHT, respectivamente. Vale ressaltar que ambas as figuras retratam o tempo de cômputo médio em escala logarítmica, enquanto os índices de desempenho MSE e DHT são representados na escala linear.

Nota-se que para $N = 2$, os resultados obtidos são muito próximos e a melhor opção não é tão clara. Enquanto o FCS convencional e o SDA obtêm os melhores índices (8% menos DHT, de 1,54% para 1,42%, e 14% menos MSE, de $3,38 V^2$ para $2,91 V^2$), os algoritmos PAV-FCS-MPC e PAV-SDA gastam quase 10% menos tempo de cômputo, de $10 \mu\text{s}$ para $9 \mu\text{s}$.

Para $N \geq 3$, o algoritmo PAV-SDA obtém os menores valores de DHT, MSE e T_c , tornando-se a melhor alternativa. Por exemplo, para $N = 3$, como pode ser visto na Tabela 15, é a única técnica que possui T_c menor que o período de amostragem para F_s de até 40 kHz. Com as outras técnicas, utilizar esta frequência de amostragem só é possível para horizontes com até duas amostras. Em comparação ao SDA, a segunda técnica mais rápida para $N \geq 3$, o PAV-SDA reduz 14% do T_c para $N = 3$ ($0,029 \text{ ms}$ para $0,025 \text{ ms}$), 20% para $N = 4$ ($0,070 \text{ ms}$

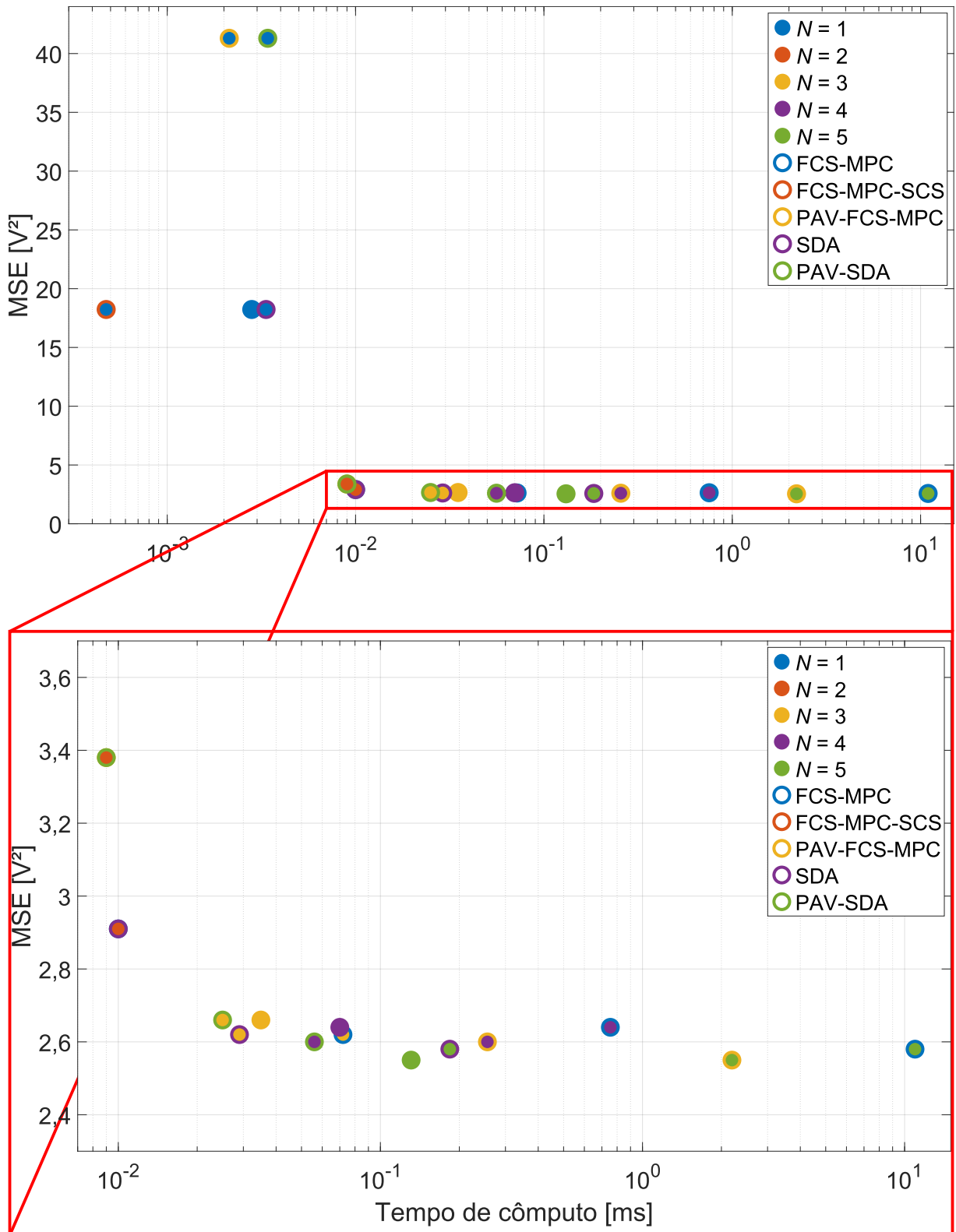


Figura 13 – Comparação entre o tempo de cômputo necessário e o MSE para cada técnica com o modelo CARMA.

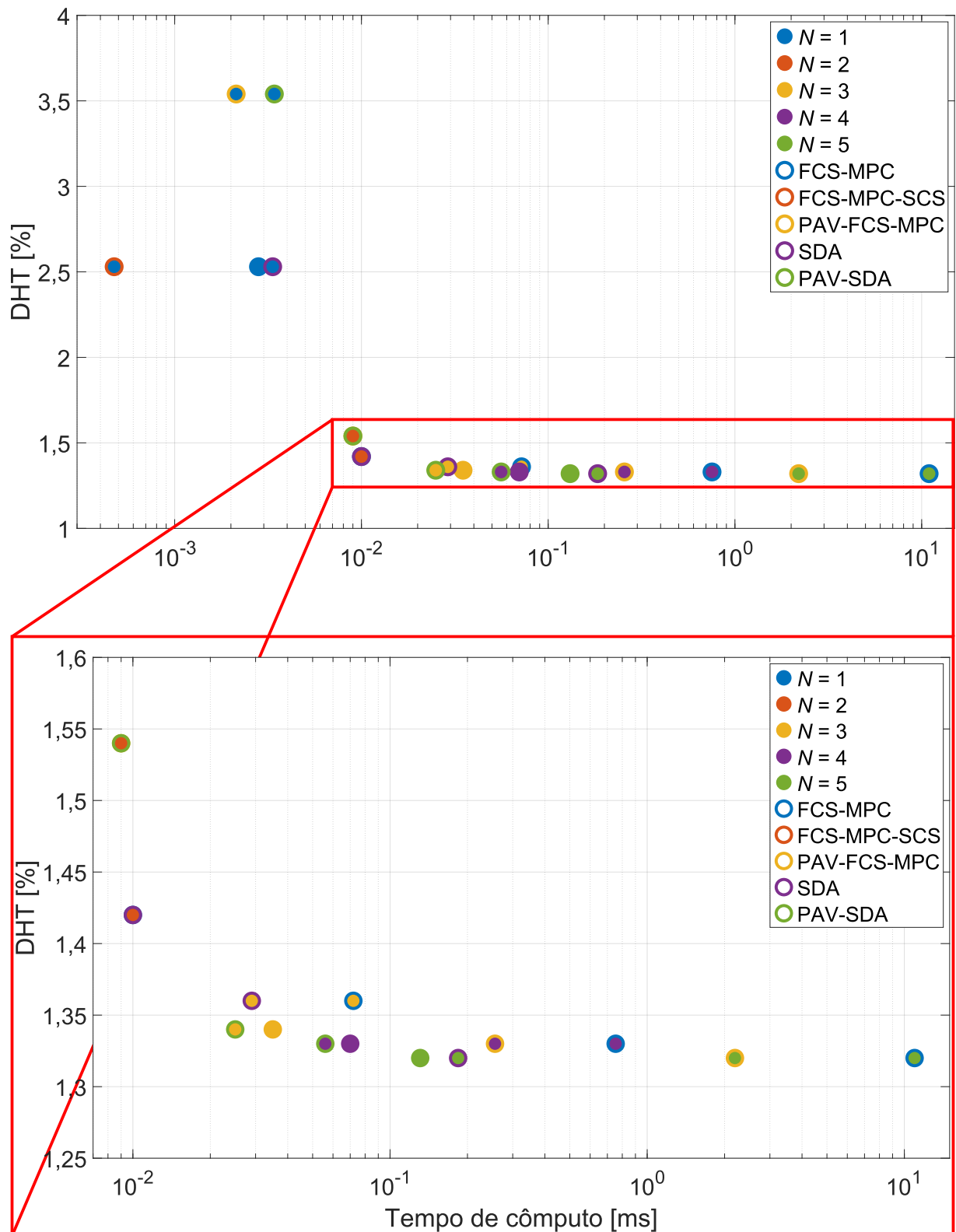


Figura 14 – Comparação entre o tempo de cômputo necessário e o DHT para cada técnica com o modelo CARMA.

para 0,056 ms) e 29% para $N = 5$ (0,184 ms para 0,131 ms).

4.4 ANÁLISE DO REGIME TRANSITÓRIO

Com o objetivo de avaliar o regime transitório dos controladores, dois tipos de ensaios foram simulados com as técnicas avaliadas. Primeiramente, quando a resposta das técnicas já havia alcançado o RP, uma resistência de $12\ \Omega$ foi acionada em paralelo com a carga resistiva. Com isso, em $0,03\ \text{s}$, a resistência para cada uma das fases da carga nas simulações passou de $R_{\text{sim}} = 54\ \Omega$ para $9,82\ \Omega$. Para este cenário, apenas o comprimento de horizonte $N = 1$ foi utilizado.

Como visto na Seção 4.3, existem três grupos que levam a respostas diferentes entre as combinações de técnicas e modelos avaliados neste trabalho. Como as técnicas FCS-MPC padrão, FCS-MPC-SCS e SDA apresentam os mesmos resultados, podem ser agrupadas em dois grupos, para os modelos CARMA e CARIMA. O outro grupo é oriundo das técnicas subótimas com vetores adjacentes, PAV-FCS-MPC e PAV-SDA, que só foram simuladas com o modelo CARMA.

Como pode ser visto na Figura 15, mesmo com o degrau de carga diminuindo em quase 80% a resistência nas fases, em um panorama geral, a resposta da técnica sofre uma leve alteração que rapidamente é consertada pelo controlador. Na Figura 16 é possível comprovar o degrau na resistência da carga, visto que o aumento da corrente foi inversamente proporcional ao degrau da carga. Este bom comportamento se mantém para todas as técnicas, como pode ser visto entre as Figuras 17 e 20.

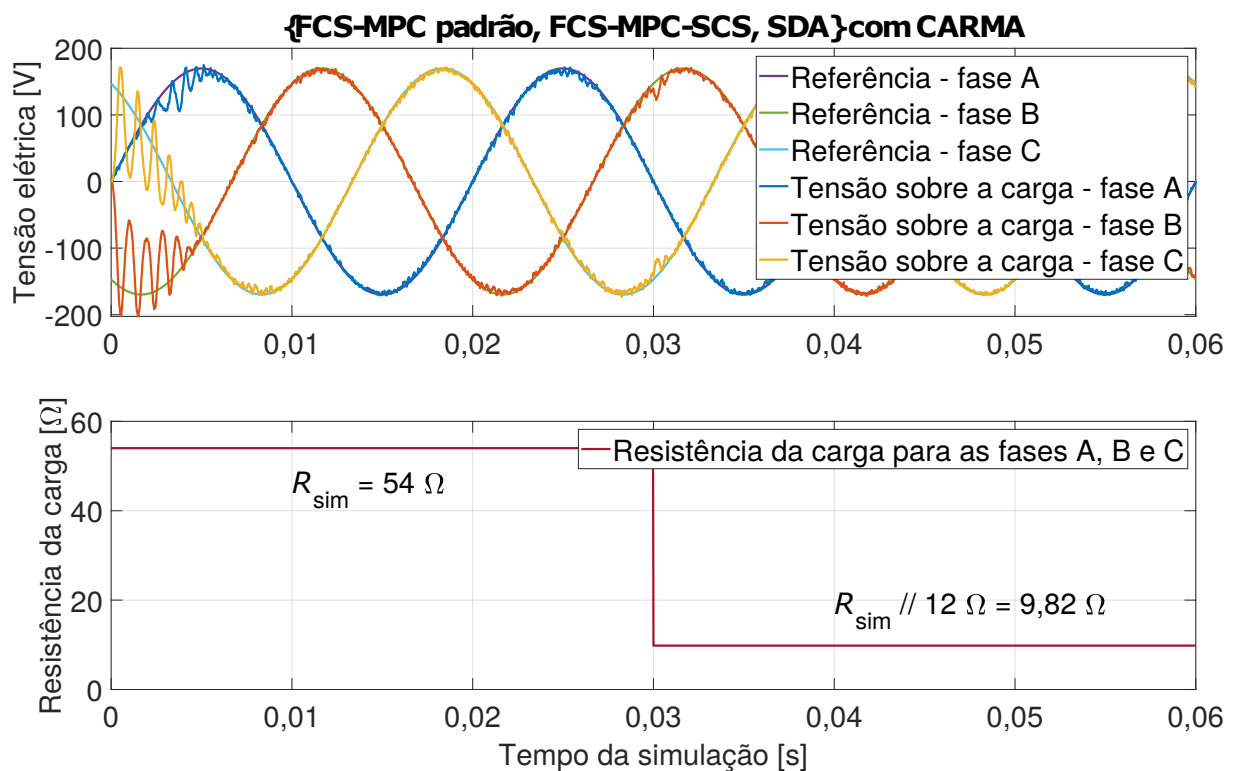


Figura 15 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de previsão CARMA - Seguimento de referência da tensão de saída e degrau de carga.

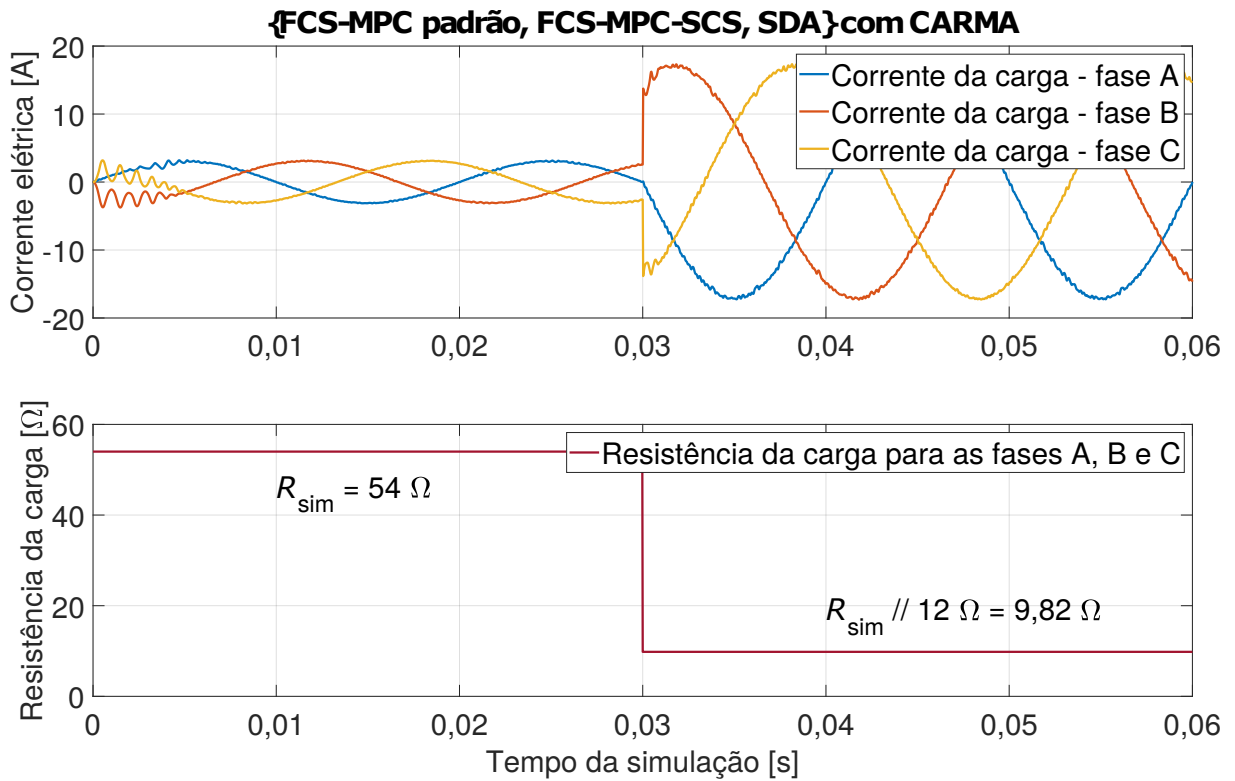


Figura 16 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de previsão CARMA - Corrente na saída e degrau de carga.

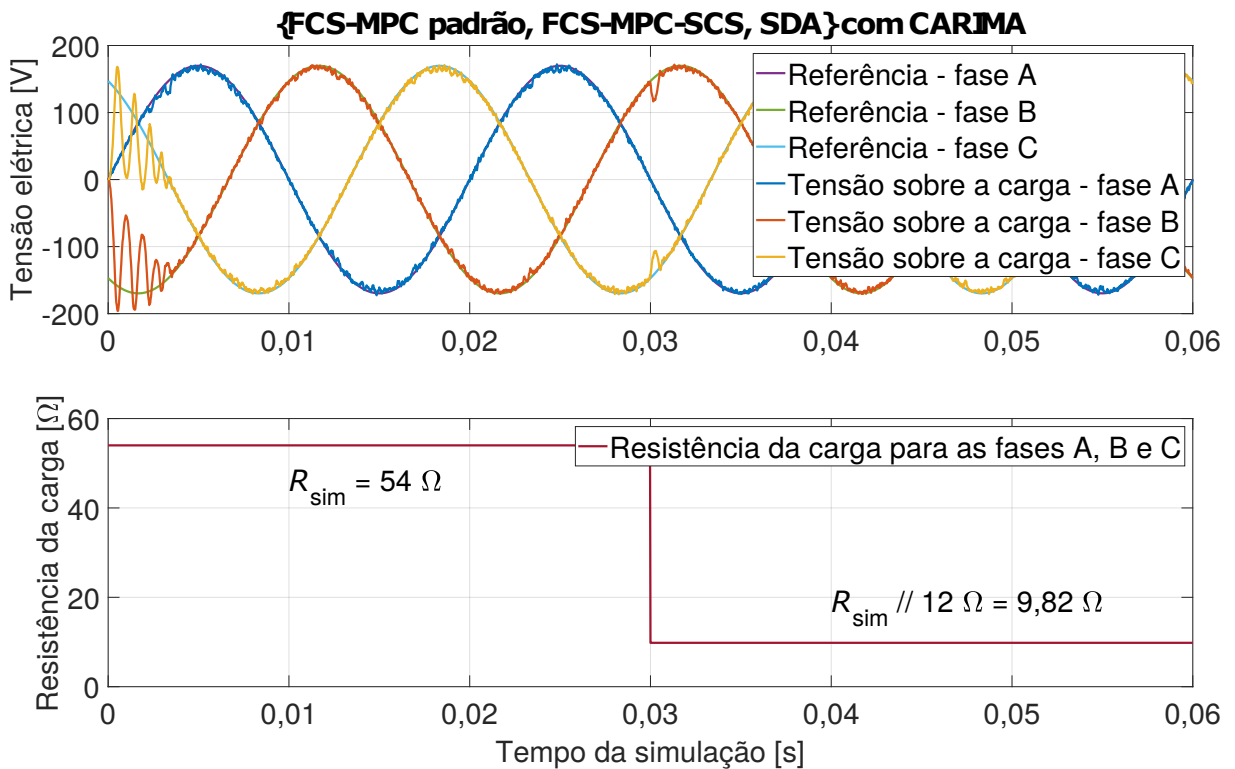


Figura 17 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de previsão CARIMA - Seguimento de referência da tensão de saída e degrau de carga.

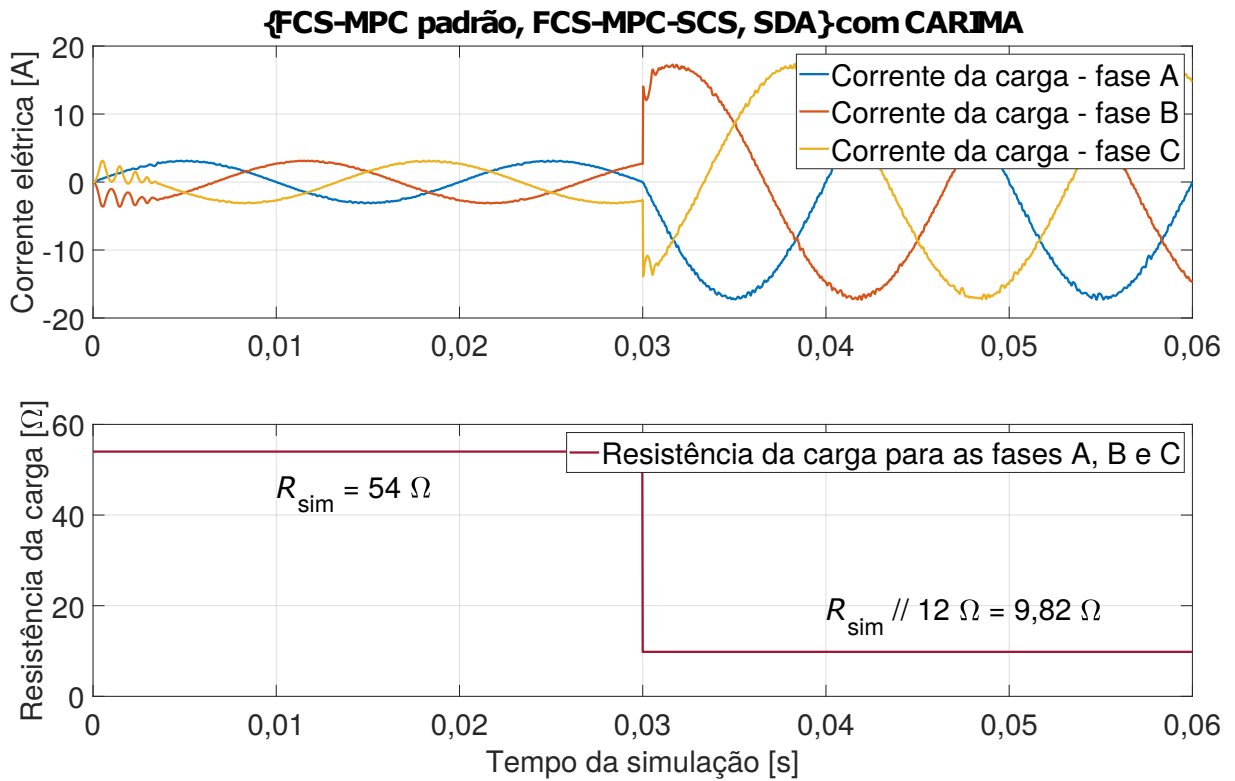


Figura 18 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARIMA - Corrente na saída e degrau de carga.

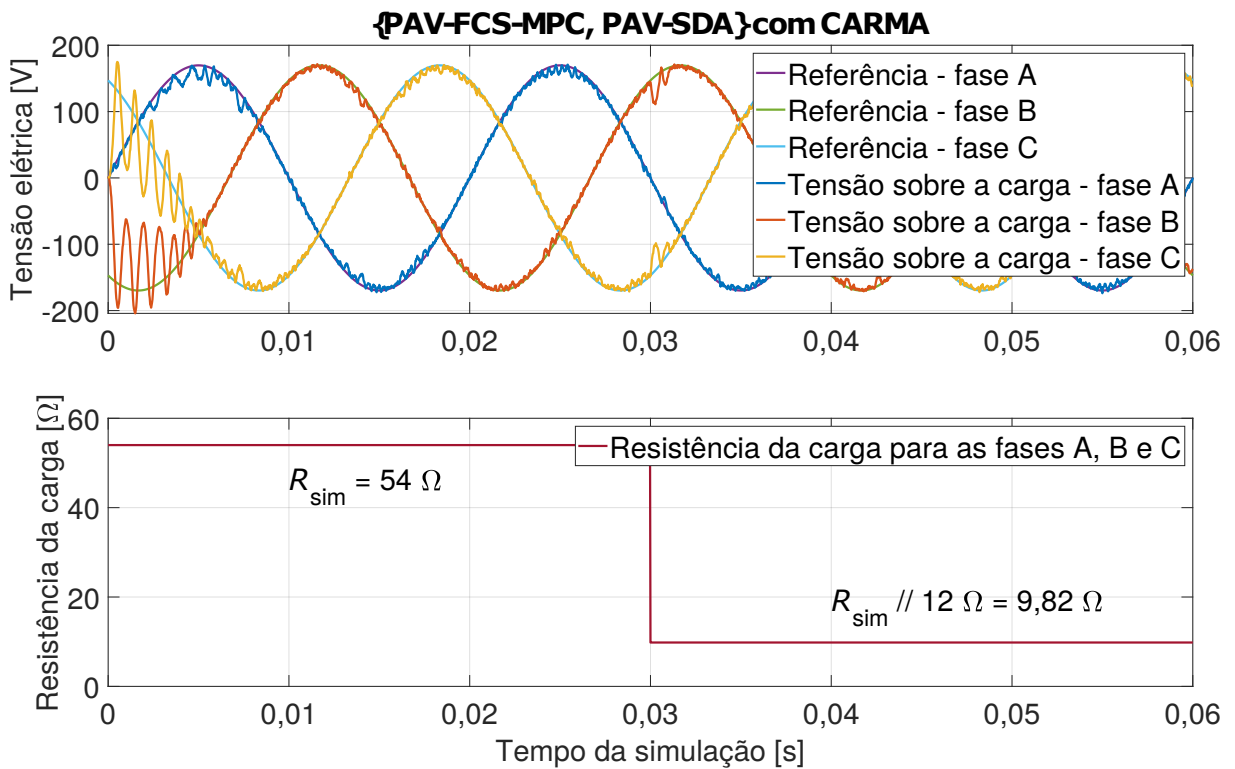


Figura 19 – PAV-FCS-MPC e PAV-SDA com modelo de predição CARMA - Seguimento de referência da tensão de saída e degrau de carga.

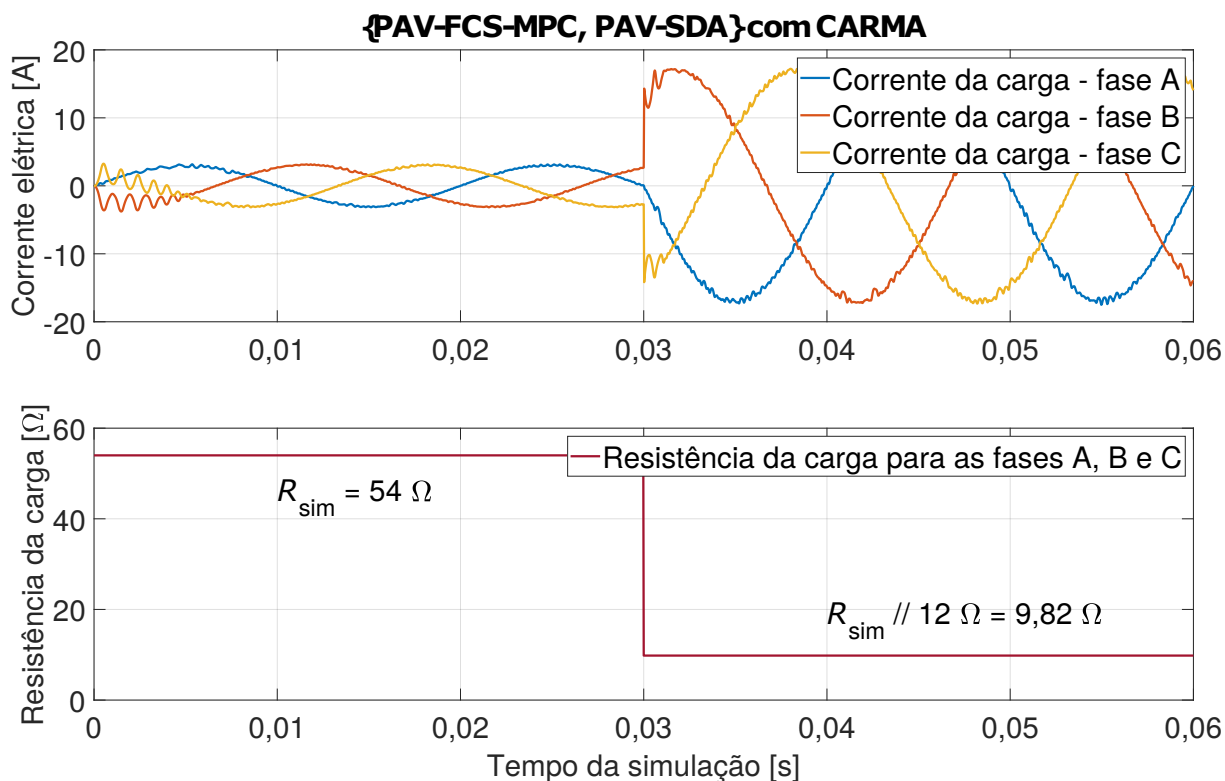


Figura 20 – PAV-FCS-MPC e PAV-SDA com modelo de previsão CARMA - Corrente na saída e degrau de carga.

Para possibilitar a comparação de desempenho entre as técnicas, a Figura 21 pode ser observada. É visto que, por mais que todas possuam comportamentos muito semelhantes, as respostas com as técnicas que encontram a solução ótima ao problema teórico são ligeiramente melhores que a resposta com PAV-FCS-MPC e PAV-SDA, sendo o modelo CARMA o que apresenta o melhor resultado, pois apresenta a menor variação na amplitude do sinal e se assenta próximo à referência de forma mais rápida.

O segundo ensaio utilizado para avaliar o regime transitório consistiu na variação da amplitude da referência e na análise de métricas quantitativas do desempenho de cada técnica para horizontes entre uma e três amostras de comprimento. A variação da amplitude da referência também aconteceu aos 0,03s, quando o sistema já estava em RP, e foi de $120 V_{rms}$ para $60 V_{rms}$, sem informar previamente o controlador através das referências futuras, de modo que o seguimento do degrau de referência se iniciou apenas no instante em que efetivamente havia sido implementado o degrau. As técnicas foram agrupadas da mesma forma que no ensaio anterior, ou seja, entre os modelos de previsão CARMA e CARIMA, e com o modelo CARMA, entre técnicas com respostas ótimas ou subótimas. O tempo de simulação foi de novamente 0,06s, mas os resultados demonstrados entre as Figuras 22 e 24 estão ampliados entre 0,02s e 0,04s para facilitar a visualização da diferença de desempenho. Além das figuras, as Tabelas 16 e 17 podem ser utilizadas para verificar as métricas quantitativas utilizadas na análise do regime transitório oriunda deste ensaio.

Nota-se que, diferentemente do que ocorre nas avaliações realizadas em RP e no ensaio

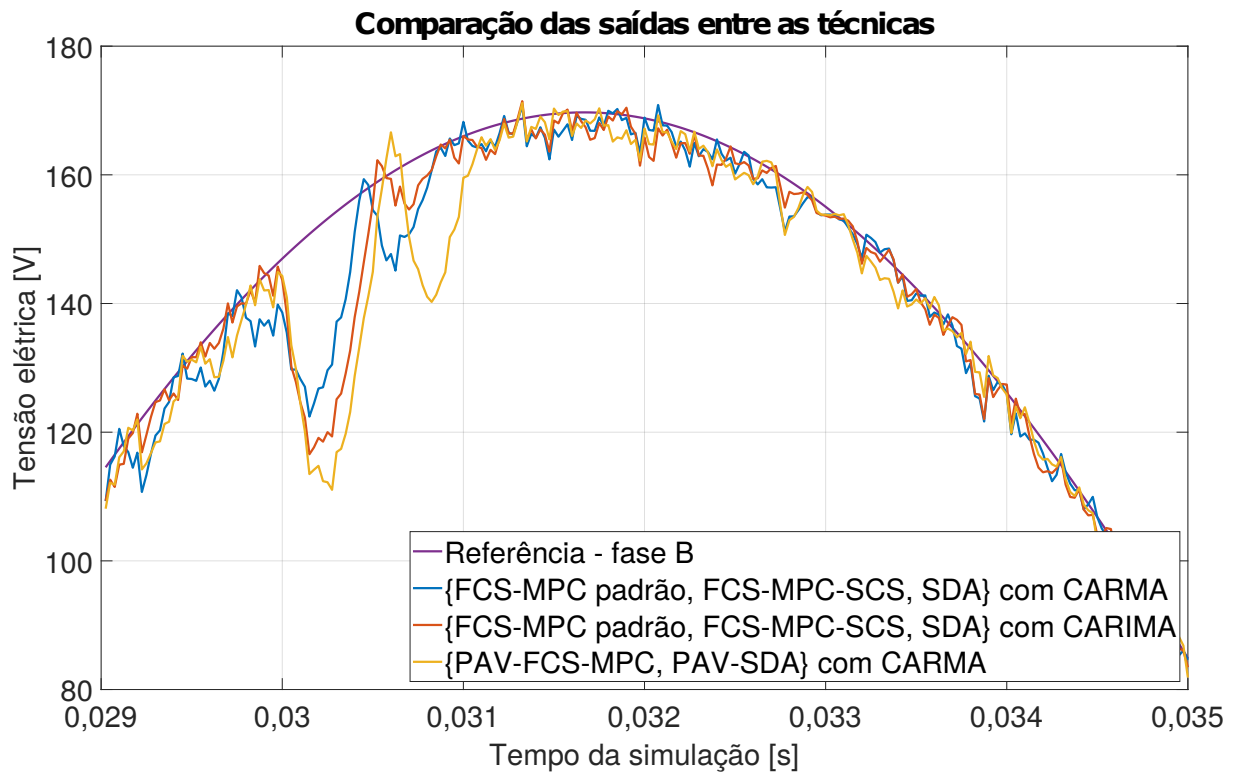


Figura 21 – Comparação do seguimento de referência entre os grupos de técnicas.

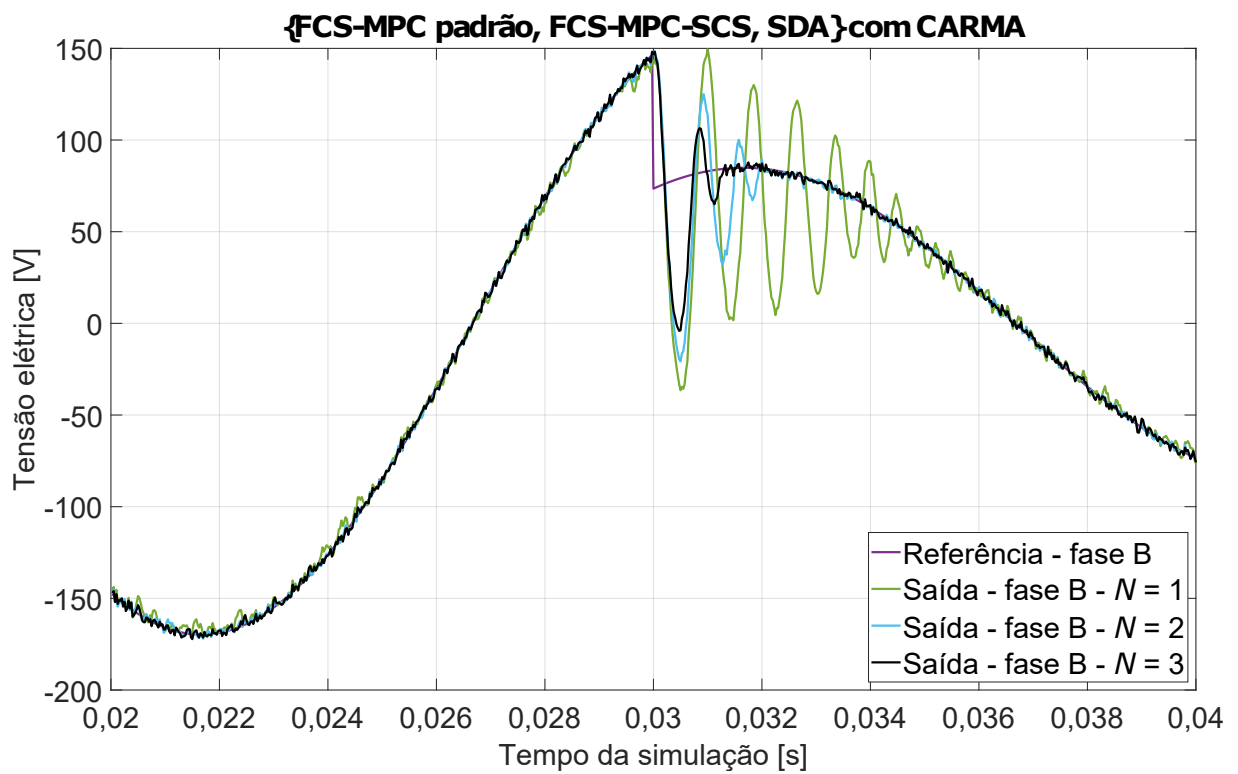


Figura 22 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de predição CARMA - mudança de referência entre diferentes comprimentos de horizonte.

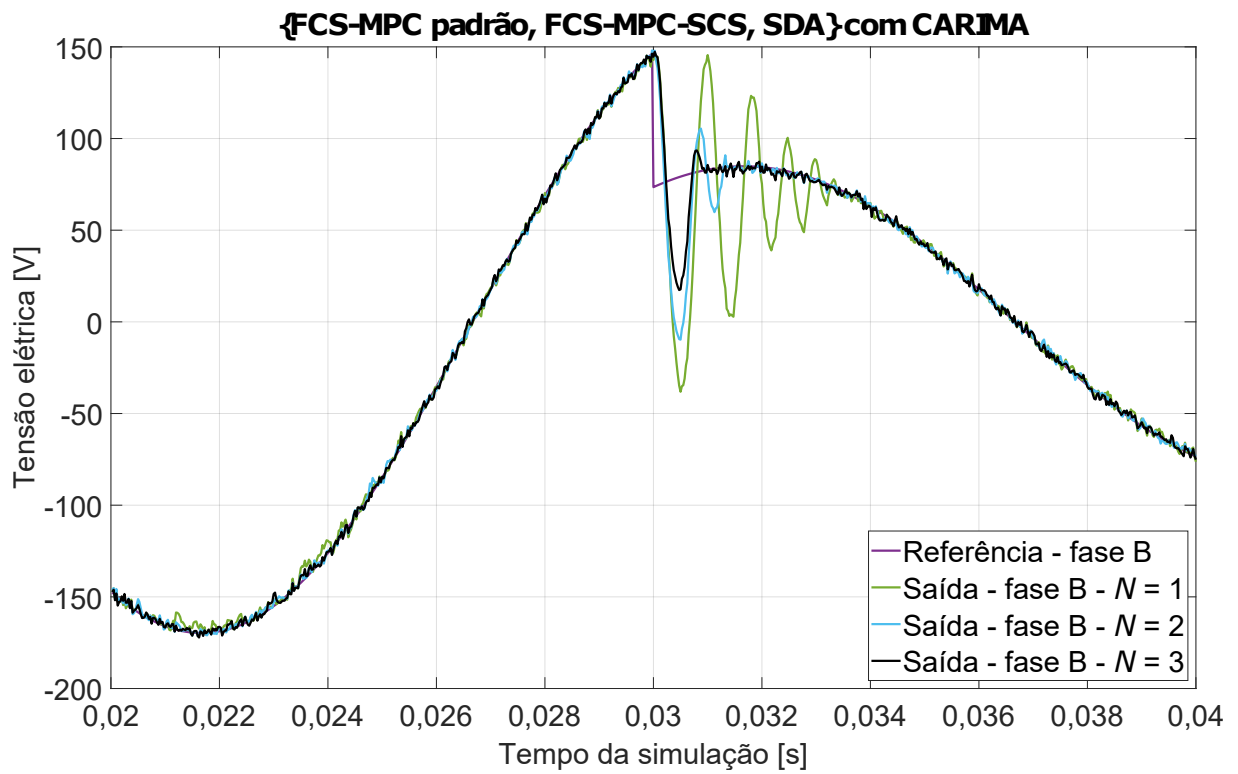


Figura 23 – FCS-MPC padrão, FCS-MPC-SCS e SDA com modelo de previsão CARIMA - mudança de referência entre diferentes comprimentos de horizonte.

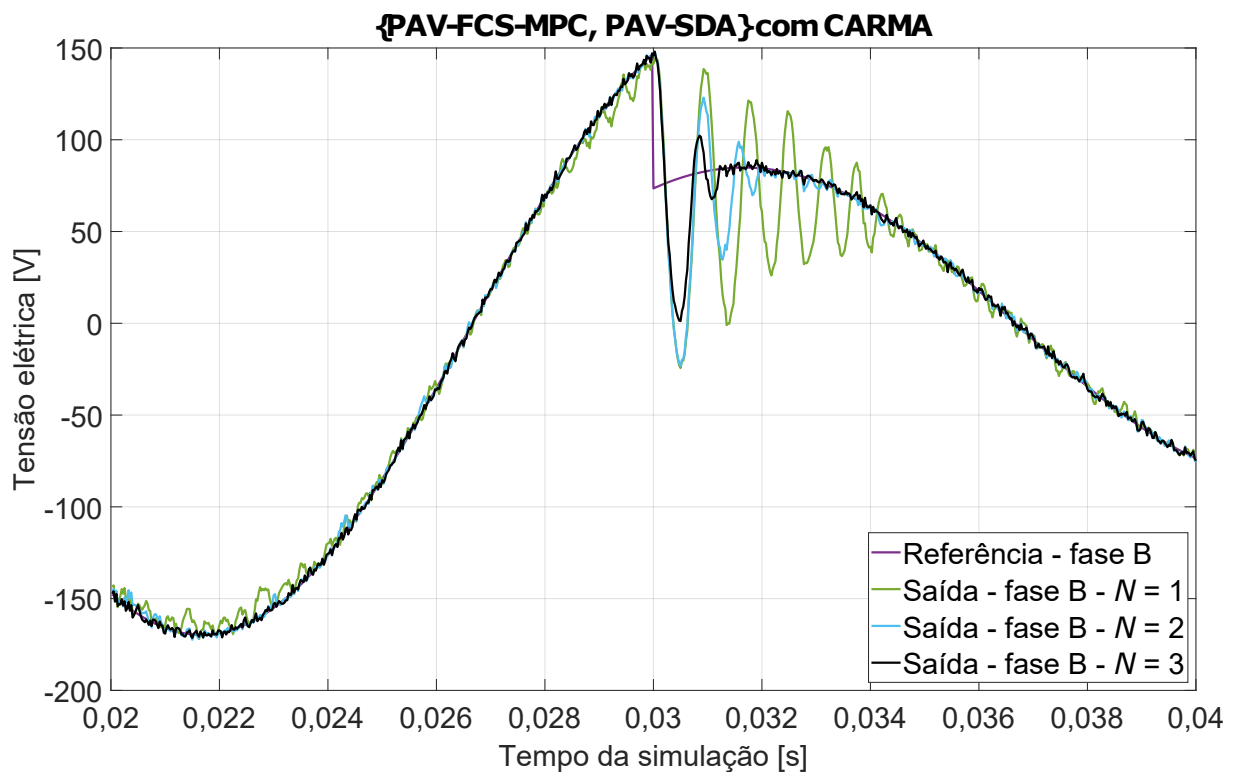


Figura 24 – PAV-FCS-MPC e PAV-SDA com modelo de previsão CARMA - mudança de referência entre diferentes comprimentos de horizonte.

anterior para análise do RT, a utilização do modelo CARIMA resultou nos melhores regimes transitórios na saída do sistema. Comparando a Figura 23 com as Figuras 22 e 24, é possível visualizar que com o modelo de predição CARIMA, de forma independente do tamanho do horizonte, o regime transitório obteve menos sobressinal e foi mais curto.

Na Tabela 16, a métrica apresentada para auxiliar na análise do regime transitório é o erro quadrático acumulado (e_{Σ}^2). Diferente do MSE apresentado nas análises de regime permanente, a média não é calculada nesta etapa. O procedimento para obter esta métrica é descrito na equação (110).

$$e_{\Sigma}^2 = \sum_{x=a}^c \sum_{t=0,03}^{0,036} \left[(w_x(t) - y_x(t))^2 \right] \quad (110)$$

Vale ressaltar que, por mais que o período analisado seja iniciado em 0,03 s e finalizado em 0,036 s, o passo da integral discreta respeita o T_s utilizado na simulação. Com essa métrica, é possível dimensionar a área gerada pelo erro entre a saída e a referência durante o período analisado, para cada técnica e comprimento de horizonte. Observa-se que esta métrica corrobora a análise visual das figuras, pois demonstra que o modelo CARIMA apresenta os melhores índices em RT, independente do comprimento do horizonte. Para os grupos com o modelo de predição CARMA, a melhor técnica de acordo com esta métrica vai depender do comprimento do horizonte, mas, de modo geral, as técnicas do grupo CARMA subótimo apresentam mais consistência e, em $N = 1$, um resultado bem próximo àqueles obtidos com as técnicas que utilizam o modelo CARIMA, devido à componente em RP que interfere na análise.

Tabela 16 – Erro quadrático acumulado (e_{Σ}^2) para $0,03 \text{ s} \leq t \leq 0,036 \text{ s}$.

Grupo	e_{Σ}^2 [V ²]		
	$N = 1$	$N = 2$	$N = 3$
CARMA ótimo	21,6862	8,8281	5,3436
CARIMA ótimo	16,0860	5,9132	3,4455
CARMA subótimo	16,1186	9,0452	4,8056

Para uma análise sem interferências do RP e que possa ser avaliada quantitativamente, os períodos de assentamento (t_a) para cada combinação entre grupo e comprimento de horizonte foram medidos. Para facilitar esse procedimento e trabalhar com valores de referência constantes, os valores da referência e da saída trifásica do sistema foram convertidos através da transformada de Park e utilizados no sistema de coordenadas rotativo dq . Além disso, para desconsiderar a oscilação de tensão presente na resposta em regime permanente, foi estabelecida uma tolerância de $\pm 7 \text{ V}$ em torno da componente d da referência. O tempo que a componente d da saída do sistema levou para se assentar dentro dessa faixa de tolerância foi medido e é apresentado na Tabela 17.

Como já havia sido comentado na análise visual, é possível visualizar na Tabela 17 que as técnicas com o modelo CARIMA levam menos tempo para se estabelecer na faixa delimitada

Tabela 17 – Tempo de assentamento (t_a) da componente d da saída das técnicas com $\pm 7V$ de tolerância para a referência.

Grupo	t_a [s]		
	$N=1$	$N=2$	$N=3$
CARMA ótimo	0,0343	0,0319	0,0312
CARIMA ótimo	0,0330	0,0313	0,0309
CARMA subótimo	0,0341	0,0319	0,0312

neste ensaio. Além disso, é possível visualizar que, por mais que o degrau de referência não foi previamente informado ao controlador, conforme o comprimento do horizonte de predição aumenta, menor é o t_a das técnicas.

4.5 CONSIDERAÇÕES

Neste capítulo, as técnicas avaliadas neste trabalho foram submetidas a análises em RP e RT. Para quase todos os cenários, além da avaliação do controlador, diferentes comprimentos de horizontes foram implementados para avaliar o impacto do seu incremento.

Para $N=1$, a combinação da técnica FCS-MPC-SCS com o modelo de predição CARIMA obteve os melhores resultados em RP e faz parte do grupo com a melhor resposta transitória no ensaio com degrau de referência. Para $N \geq 2$, independente do algoritmo selecionado, o modelo de predição CARIMA trouxe os piores índices para RP. Nas análises de RT, este modelo foi o intermediário no degrau de carga e o melhor em degrau de referência, para todos os horizontes avaliados. Como todas as técnicas apresentaram um resultado satisfatório para o RT, recomenda-se que a escolha da técnica a ser utilizada seja norteada pelos resultados obtidos em RP, pois foi neste tipo de análise que alguns controladores se destoaram entre as técnicas e os comprimentos de horizonte avaliados. Deste modo, para $N=2$ as melhores opções seriam as combinações entre o modelo de predição CARMA e os algoritmos FCS-MPC e SDA, que apresentaram os menores valores de DHT e MSE entre as opções comparadas, assim como as técnicas PAV-FCS-MPC e PAV-SDA, que consumiram o menor tempo de cômputo para serem executadas. A escolha entre essas técnicas pode variar de acordo com as prioridades do projeto. Para $N \geq 3$, o algoritmo PAV-SDA apresentou os melhores índices de DHT e MSE em quase todos os horizontes e o menor tempo de cômputo. Por mais que no RT apresentou valores menos competitivos que o modelo CARIMA, a sua melhoria em RP assume prioridade e foi definido como a melhor opção para estes casos.

5 CONSIDERAÇÕES FINAIS

Este capítulo discorre sobre as conclusões e considerações finais do presente trabalho, os objetivos atingidos nesse contexto e, também, sobre possíveis sugestões de trabalhos futuros no tema.

5.1 CONCLUSÕES

Neste trabalho, são apresentadas comparações de resultados entre diferentes algoritmos para redução do tempo de cômputo do FCS-MPC. Em conjunto com técnicas abordadas na literatura, um algoritmo denominado PAV-SDA foi proposto neste trabalho, ao combinar duas das técnicas estudadas. Os algoritmos são aplicados via simulação para realizar o controle de tensão de um inversor tipo fonte de tensão trifásico de dois níveis. Análises relacionadas à distorção harmônica total e ao erro médio quadrático nas tensões de saída são utilizadas para avaliar o desempenho do rastreamento de referência após dois ciclos de referência, enquanto o tempo médio de cômputo de cada algoritmo em cada amostra de controle representa o esforço computacional relacionado a cada técnica. Ainda, duas análises complementares sobre o regime transitório das técnicas foram realizadas em cenários com variação de carga, com uma redução de resistência na ordem de 80%, e degrau de amplitude na referência, levando a tensão eficaz para 50% do valor inicial. Concluiu-se que, por mais que houvesse uma diferença nos resultados das técnicas, todas atingiram baixo sobressinal e tempo de assentamento curto o suficiente para serem consideradas satisfatórias. Com isso, é recomendado que a escolha da técnica se baseie nos índices de RP, discutidos a seguir.

É visto que ao aumentar o tamanho dos horizontes, o desempenho do rastreamento de referência melhora. Todavia, em todos os cenários considerados, para $N > 3$ o incremento no tempo de cômputo do algoritmo é muito grande, enquanto os índices de desempenho DHT e MSE se mantêm muito semelhantes aos valores encontrados com $N = 3$ e, portanto, opções com horizontes muito grandes não são consideradas boas alternativas. Para horizontes com apenas uma amostra, as combinações do modelo de predição CARIMA com o algoritmo FCS-MPC padrão e as técnicas SDA e FCS-MPC-SCS alcançaram os menores valores de DHT e MSE entre todas as técnicas avaliadas. Entretanto, a técnica FCS-MPC-SCS foi a que obteve o melhor custo-benefício, pois utilizou 21% do tempo de cômputo necessário pela segunda técnica mais rápida naquele cenário, o FCS-MPC convencional. Para horizontes maiores, o modelo de predição com melhores índices de desempenho foi o CARMA. Para $N = 2$, a decisão da melhor técnica depende das restrições do sistema controlado. Se o sistema precisa operar com o melhor rastreamento de referência possível, a escolha fica empatada entre o algoritmo convencional de FCS-MPC e o SDA, pois em relação as outras técnicas alcançam 8% menos DHT e 14% menos MSE. Para os casos nos quais o tempo de cômputo precisa ser reduzido ao máximo, as técnicas PAV-FCS-MPC e PAV-SDA utilizam 10% menos tempo de cômputo. Para horizontes com três amostras ou mais, o algoritmo PAV-SDA obtém os menores valores

de DHT, MSE e tempo de cômputo médio, tornando-se a melhor alternativa. Para $N = 5$, em comparação ao SDA, a segunda técnica mais rápida neste comprimento de horizonte, o algoritmo proposto necessita de 29% menos tempo de cômputo com praticamente os mesmos índices de MSE e DHT.

Por fim, conclui-se que os objetivos desta pesquisa foram cumpridos, visto que foram implementadas três diferentes técnicas de cômputo rápido para FCS-MPC encontradas na literatura, duas delas foram avaliadas quanto ao aumento dos horizontes, um algoritmo de cômputo rápido para FCS-MPC foi proposto e os resultados obtidos com as simulações de todas as técnicas abordadas foram comparados quanto ao seu tempo de cômputo e índices de rastreamento de referência.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

No decorrer do desenvolvimento deste trabalho, algumas linhas de pesquisa surgiram como opções para possíveis trabalhos futuros. Primeiramente, sugere-se ter um bom modelo de predição no sistema de coordenadas dq e implementar os algoritmos com $N_u < N$. Com isso, seria possível diminuir ainda mais o tempo de cômputo e alcançar horizontes maiores com os mesmos algoritmos, melhorando os índices de desempenho da resposta do sistema. Para o modelo de predição utilizado no sistema de coordenadas $\alpha\beta$, este tipo de estratégia não é adequada visto que a variação da variável manipulada entre N_u e N seria nula. Uma outra alternativa ao modelo em dq seria modelar o problema com uma estrutura similar àquela do modelo CARIMA, porém, trocando o integrador discreto por um termo com modos na frequência da referência utilizada. Isto poderia resultar em um modelo que ofereça melhores resultados do que os obtidos com os modelos CARMA e CARIMA.

Outra estratégia sugerida consiste em implementar algumas destas técnicas de cômputo rápido em conjunto com funções custo modificadas, encontradas na literatura, que reduzem o erro em regime permanente e ponderam variáveis como a quantidade de comutações realizadas dentro de uma amostra de controle. Essa integração também contribuiria com a melhora do rastreamento de referência e da eficiência do conversor, ao custo de um leve aumento de tempo de cômputo.

Por fim, como a técnica FCS-MPC-SCS apresentou os melhores resultados para horizontes de apenas uma amostra e sua generalização para horizontes maiores ainda não foi discutida na literatura, outro caminho sugerido como trabalho futuro seria testar esta generalização do FCS-MPC-SCS ao combiná-lo com uma estrutura de algoritmo semelhante ao SDA. Caso funcione, esse desenvolvimento tem o potencial de resultar nos menores tempos de cômputo entre as técnicas e ainda obter solução ótima, ou seja, apresentar a mesma solução fornecida pelo FCS-MPC padrão. Esta técnica ainda poderia ser combinada com as funções custo modificadas, discutidas previamente.

REFERÊNCIAS

ALATHAMNEH, M.; YANG, X.; NELMS, R. M. Power Control of a Three-phase Grid-connected Inverter using a Proportional-Resonant Control Method under Unbalanced Conditions. en. *In: IECON 2021 – 47th Annu. Conf. of the IEEE Ind. Electron. Soc. Toronto, ON, Canada: IEEE, out. 2021. P. 1–6. DOI: 10.1109/IECON48115.2021.9589120. Acesso em: 16 out. 2023.*

ÅSTRÖM, K. J.; WITTENMARK, B. **Computer-controlled systems: theory and design.** Third edition. [S.l.]: Dover publications, 2011. ISBN 978-0-486-28404-0.

BLAABJERG, F. **Control of power electronic converters and systems. 3rd Volume.** London: Academic, 2021. ISBN 978-0-12-819432-4.

BORDONS, C.; MONTERO, C. Basic Principles of MPC for Power Converters: Bridging the Gap Between Theory and Practice. **IEEE Ind. Electron. Mag.**, v. 9, n. 3, p. 31–43, set. 2015. ISSN 1941-0115. DOI: 10.1109/MIE.2014.2356600.

BRUNONI, J. P. *et al.* IMPLEMENTAÇÃO DE GPC DE CÔMPUTO RÁPIDO COM RESTRIÇÕES BASEADA EM MICROCONTROLADOR. pt. *In.* DOI: 10.20906/CBA2022/3378.

CASEIRO, L. M. A.; MENDES, A. M. S.; CRUZ, S. M. A. Dynamically Weighted Optimal Switching Vector Model Predictive Control of Power Converters. **IEEE Trans. Ind. Electron.**, v. 66, n. 2, p. 1235–1245, fev. 2019. ISSN 0278-0046, 1557-9948. DOI: 10.1109/TIE.2018.2829689.

CHEREAU, V.; AUGER, F.; LORON, L. A new three-phase varying-band hysteresis current controller for voltage-source inverters. en. *In: 2007 7th Int. Conf. on Power Electron. and Drive Syst. Bangkok, Thailand: IEEE, nov. 2007. P. 1179–1186. DOI: 10.1109/PEDS.2007.4487855. Acesso em: 16 out. 2023.*

CORDERO, R. *et al.* Development of a Resonant Generalized Predictive Controller for Sinusoidal Reference Tracking. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 69, n. 3, p. 1218–1222, 2022. DOI: 10.1109/TCSII.2021.3102535.

CORTES, P. *et al.* Delay Compensation in Model Predictive Current Control of a Three-Phase Inverter. **IEEE Trans. Ind. Electron.**, v. 59, n. 2, p. 1323–1325, fev. 2012. ISSN 0278-0046, 1557-9948. DOI: 10.1109/TIE.2011.2157284. Acesso em: 30 jun. 2023.

CORTÉS, P. *et al.* Model Predictive Control of Multilevel Cascaded H-Bridge Inverters. **IEEE Trans. Ind. Electron.**, v. 57, n. 8, p. 2691–2699, ago. 2010. ISSN 1557-9948. DOI: 10.1109/TIE.2010.2041733.

DORFLING, T. *et al.* Long-Horizon Finite-Control-Set Model Predictive Control With Nonrecursive Sphere Decoding on an FPGA. **IEEE Trans. on Power Electron.**, v. 35, n. 7, p. 7520–7531, jul. 2020. ISSN 0885-8993, 1941-0107. DOI: 10.1109/TPEL.2019.2956213.

ELMORSHEDY, M. F. *et al.* Recent Achievements in Model Predictive Control Techniques for Industrial Motor: A Comprehensive State-of-the-Art. **IEEE Access**, v. 9, p. 58170–58191, 2021. ISSN 2169-3536. DOI: 10.1109/ACCESS.2021.3073020.

GEYER, T.; QUEVEDO, D. E. Multistep Finite Control Set Model Predictive Control for Power Electronics. **IEEE Transactions on Power Electronics**, v. 29, n. 12, p. 6836–6846, 2014. DOI: 10.1109/TPEL.2014.2306939.

GUO, Y. *et al.* A FCS-MPC Method Based on Simplified Control Set for Five-Phase PMSMs. *In: 6TH IEEE Int. Conf. on Predictive Contr. of Elect. Drives and Power Electron. (PRECEDE)*. Jinan, China: IEEE, nov. 2021. P. 317–321. DOI: 10.1109/PRECEDE51386.2021.9680956.

GUTIERREZ, B.; KWAK, S.-S. Modular Multilevel Converters (MMCs) Controlled by Model Predictive Control With Reduced Calculation Burden. **IEEE Trans. on Power Electron.**, v. 33, n. 11, p. 9176–9187, nov. 2018. ISSN 1941-0107. DOI: 10.1109/TPEL.2018.2789455.

JIN, T. *et al.* Model Predictive Voltage Control Based on Finite Control Set With Computation Time Delay Compensation for PV Systems. **IEEE Trans. Energy Convers.**, v. 34, n. 1, p. 330–338, mar. 2019. ISSN 1558-0059. DOI: 10.1109/TEC.2018.2876619.

JUDEWICZ, M. G. *et al.* Generalized Predictive Current Control (GPCC) for Grid-Tie Three-Phase Inverters. **IEEE Trans. on Indu. Electron.**, v. 63, n. 7, p. 4475–4484, jul. 2016. ISSN 1557-9948. DOI: 10.1109/TIE.2015.2508934.

KARAMANAKOS, P. *et al.* Computationally efficient sphere decoding for long-horizon direct model predictive control. *In: 2016 IEEE Energy Conversion Congress and Exposition (ECCE)*. [S.l.: s.n.], 2016. P. 1–8. DOI: 10.1109/ECCE.2016.7854955.

LEZANA, P.; AGUILERA, R.; QUEVEDO, D. Steady-state issues with finite control set model predictive control. *In: 2009 35th Annu. Conf. of IEEE Ind. Electron.* Porto, Portugal: IEEE, nov. 2009. P. 1776–1781. DOI: 10.1109/IECON.2009.5414812.

LIANG, X.; FLEMING, E. Electrical Submersible Pump Systems: Evaluating Their Power Consumption. **IEEE Ind. Appl. Mag.**, v. 19, n. 6, p. 46–55, nov. 2013. ISSN 1558-0598. DOI: 10.1109/MIAS.2012.2215660.

LYAKHOMSKII, A. *et al.* Assessment of the Harmonics Influence on the Power Consumption of an Electric Submersible Pump Installation. *en. Energies*, v. 15, n. 7, p. 2409, mar. 2022. ISSN 1996-1073. DOI: 10.3390/en15072409.

MCNABB, L.; WANG, L. An Improved Horizon-1 Finite-Control-Set Model-Predictive Current Control Design for an Induction Machine With Integrated Anti-Windup. **IEEE Trans. Control. Syst. Technol.**, v. 31, n. 2, p. 985–988, mar. 2023. ISSN 1558-0865. DOI: 10.1109/TCST.2022.3193500.

MEMIJE, D. *et al.* Reduction of Grid Background Harmonics in Three-Phase Inverters by Applying Sliding Mode Control. en. *In: 2020 IEEE 29th Int. Symp. on Ind. Electron. (ISIE)*. Delft, Netherlands: IEEE, jun. 2020. P. 191–196. DOI: 10.1109/ISIE45063.2020.9152589. Acesso em: 16 out. 2023.

MOHAMED, I. S. *et al.* Three-phase inverter with output LC filter using predictive control for UPS applications. *In: PROC. Int. Conf. Control, Decis. Inf. Technol.(CoDIT)*. Hammamet, Tunisia: IEEE, mai. 2013. P. 489–494. DOI: 10.1109/CoDIT.2013.6689593.

NORMEY-RICO, J. E. *et al.* **Controle preditivo baseado em modelo: caso monovariável linear**. São Paulo, SP: Editora Edgard Blucher, fev. 2024. ISBN 978-85-212-2128-9.

PATHIK, B. B. *et al.* Total Harmonic Distortion Reduction in a Power Grid: A UPQC Based Approach. *In: 2023 3rd Int. Conf. on Robotics, Electrical and Signal Processing Techniques (ICREST)*. [S.l.: s.n.], 2023. P. 264–268. DOI: 10.1109/ICREST57604.2023.10070091.

PECCIN, V. B. *et al.* Fast algorithms for constrained generalised predictive control with on-line optimisation. en. **IET Cont. Theory & Appl**, v. 15, n. 4, p. 545–558, 2021. ISSN 1751-8652. DOI: 10.1049/cth2.12060.

PECCIN, V. B. *et al.* Fast Constrained Generalized Predictive Control with ADMM Embedded in an FPGA. **IEEE Latin America Trans.**, v. 18, p. 422–429, fev. 2020. ISSN 1548-0992. DOI: 10.1109/TLA.2020.9085299.

RASHID, M. H. (Ed.). **Power electronics handbook: devices, circuits, and applications**. 3rd ed. Burlington, MA: Butterworth-Heinemann, 2011. ISBN 978-0-12-382036-5.

RASHWAN, A. *et al.* Predictive Controller Based on Switching State Grouping for a Modular Multilevel Converter With Reduced Computational Time. **IEEE Trans. Power Deliv.**, v. 32, n. 5, p. 2189–2198, out. 2017. ISSN 1937-4208. DOI: 10.1109/TPWRD.2016.2639529.

RODRÍGUEZ, J.; ESTAY, P. C. **Predictive control of power converters and electrical drives**. Chichester, West Sussex, UK: Wiley, 2012. ISBN 978-1-119-96398-1.

SONG, Q.; LIU, F. The Direct Approach to Unified GPC Based on ARMAX/CARIMA/CARMA Model and Application for Pneumatic Actuator Control. *In: FIRST Int. Conf. on Innovative Computing, Information and Control - Volume I (ICICIC'06)*. [S.l.: s.n.], 2006. P. 336–339. DOI: 10.1109/ICICIC.2006.186.

VAZQUEZ, S. *et al.* Model Predictive Control for Power Converters and Drives: Advances and Trends. **IEEE Trans. Ind. Electron.**, v. 64, n. 2, p. 935–947, fev. 2017. ISSN 1557-9948. DOI: 10.1109/TIE.2016.2625238.

VEYSINEJAD, A.; RAHMATI, S.; BINA, M. T. Predictive Direct Torque Control with Reduced Number of Predictions for a Three-level Inverter Driven Permanent Magnet Synchronous Motor. *In: 27TH Iranian Conf. on Elect. Eng. (ICEE)*. Yazd, Iran: IEEE, abr. 2019. P. 813–818. DOI: 10.1109/IranianCEE.2019.8786449.

VEYSINEJAD, A.; RAHMATI, S.; SHAMLOU, S. Predictive Direct Torque Control with Reduced Number of Considered States in Synchronous Reluctance Motor Drive. *In: 10TH Int. Power Electron., Drive Syst. and Technol. Conf. (PEDSTC)*. Shiraz, Iran: IEEE, fev. 2019. P. 90–95. DOI: 10.1109/PEDSTC.2019.8697735.

ZOU, C. *et al.* Stationary Frame Equivalent Model of Proportional-Integral Controller in dq Synchronous Frame. **IEEE Transactions on Power Electronics**, v. 29, n. 9, p. 4461–4465, 2014. DOI: 10.1109/TPEL.2013.2296789.