



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

YURI GIGLIO DE AZEVEDO

**APLICAÇÃO DE CONTROLE FCS-MPC COM FUNÇÃO DE CUSTO RESTRITA EM
CONVERSORES BOOST FOTOVOLTAICOS**

Recife
2025

YURI GIGLIO DE AZEVEDO

**APLICAÇÃO DE CONTROLE FCS-MPC COM FUNÇÃO DE CUSTO RESTRITA
EM CONVERSORES BOOST FOTOVOLTAICOS**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador(a): Prof. Dr. Rafael Cavalcanti Neto

Recife
2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Azevedo, Yuri Giglio de.

Aplicação de controle FCS-MPC com função de custo restrita em conversores boost fotovoltaicos / Yuri Giglio de Azevedo. - Recife, 2025.

102 p. : il., tab.

Orientador(a): Rafael Cavalcanti Neto

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2025.

Inclui referências, apêndices.

1. Eletrônica de Potência. 2. Controle Preditivo. 3. FCS-MPC. 4. Conversores Boost. 5. Sistemas Fotovoltaicos. I. Cavalcanti Neto, Rafael. (Orientação). II. Título.

620 CDD (22.ed.)

YURI GIGLIO DE AZEVEDO

**APLICAÇÃO DE CONTROLE FCS-MPC COM FUNÇÃO DE CUSTO RESTRITA
EM CONVERSORES BOOST FOTOVOLTAICOS**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 15/12/2025

BANCA EXAMINADORA

Prof. Dr. Rafael Cavalcanti Neto (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Fabrício Bradaschia (Examinador Interno)
Universidade Federal de Pernambuco

Eng. M. Sc. Gustavo Azevedo Teixeira (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Inicialmente, gostaria de agradecer à minha família, em especial aos meus pais que sempre fizeram de tudo para que eu e os meus irmãos alcançássemos nossos objetivos, sempre nos incentivaram a estudar, sendo verdadeiros símbolos de disciplina e comprometimento com o trabalho. Espero sempre orgulhá-los e conseguir retribuir nem que seja 10% de todo o carinho que me foi dado. Agradeço também aos meus irmãos que me incentivaram e serviram de exemplo para o caçula aqui.

Agradeço também aos amigos que fiz na Universidade, especialmente a Wellington, que me levou a questionar mais a fundo toda sorte de assuntos ligados à engenharia, sempre com um projeto novo a ser desenvolvido e a Rebeka que esteve presente por boa parte da minha graduação, sempre me incentivando e deixando meu dia mais leve.

Aos membros do Capibarib-E Racing, quero destinar os meus sinceros agradecimentos, por me fazer ver como a engenharia pode ser colocada em prática e me desenvolver em trabalhos em equipe. Estendo meus agradecimentos a todos da Interest Engenharia, que me apoiaram e foram pacientes comigo durante a preparação deste Trabalho.

Gostaria de agradecer a todos os professores que fizeram parte da minha trajetória na UFPE. Principalmente ao meu orientador, o professor Rafael Cavalcanti Neto. Quero expressar a minha gratidão a todos do GEPAE, em particular a Lucas Gabriel e Clarissa que me ajudaram a compreender o FCS-MPC e sempre foram solícitos quando precisei de ajuda.

Agradeço, por fim, aos integrantes da banca avaliadora deste trabalho, o professor Fabrício Bradaschia e o doutorando Gustavo Teixeira, pela leitura atenta, pelas contribuições criteriosas e pelas valiosas sugestões, que foram fundamentais para o aprimoramento desta pesquisa.

RESUMO

Em face da crescente demanda por eletricidade e do esforço global na adoção de fontes de energia renováveis, a necessidade de conversores de potência confiáveis e estratégias de controle eficientes atreladas a eles torna-se cada vez mais urgente. Sistemas fotovoltaicos (FV) dependem fortemente de conversores de alto desempenho para otimizar a produção de energia e realizar a interface entre os painéis e a rede elétrica interligada. Contudo, esses conversores introduzem desafios consideráveis às estratégias clássicas de controle, em função da sua natureza intrinsecamente não linear e variante no tempo. Nesse contexto, este trabalho propõe a aplicação do Controle Preditivo Baseado em Modelo com Conjunto de Controle Finito (FCS-MPC) em conversores CC-CC do tipo *boost* voltados para aplicações fotovoltaicas. Para isso, foi feita a modelagem do conversor tanto em espaço de estados quanto por meio da modelagem média de pequenos sinais, permitindo uma análise completa do comportamento dinâmico do sistema. A validação computacional das estratégias propostas foi realizada no ambiente MATLAB/Simulink, utilizando S-functions para o encapsulamento dos algoritmos de controle desenvolvidos em linguagem C. Os resultados obtidos foram comparados com aqueles provenientes de um controlador linear digital. O desempenho dinâmico dos controladores propostos foi analisado à luz de critérios quantitativos como tempo de assentamento, máximo sobressinal e métricas que avaliam a integral do sinal de erro: IAE, ISE, ITAE e ITSE. As simulações demonstraram que o método FCS-MPC proporciona uma resposta dinâmica significativamente mais rápida e eficaz, além de apresentar maior robustez, mostrando-se menos suscetível a perturbações nas variáveis de entrada. Essas características destacam o potencial da técnica como uma alternativa promissora para o controle de conversores em sistemas fotovoltaicos modernos.

Palavras-chave: Eletrônica de Potência; Controle Preditivo; FCS-MPC; Conversores Boost; Sistemas Fotovoltaicos.

ABSTRACT

In light of the rising demand for electricity and the global effort in adopting renewable energy sources, the need for reliable power converters and efficient control strategies in association with them becomes increasingly pressing. Photovoltaic systems heavily rely on high-performance converters to optimize energy production and serve as interface from panels to the interconnected power grid. Nevertheless, these converters introduce considerable challenges for classical control strategies due to their intrinsically non-linear and time-variant nature. Hence, this undergraduate thesis proposes the use of Finite Control Set Model Predictive Control (FCS-MPC) applied to DC-DC boost converters aimed at photovoltaic applications. In order to accomplish that, the converter was modeled both in state-space and average small-signal modeling, allowing for a full analysis of the system's dynamic behavior. The computational validation of the proposed strategies was carried out in the MATLAB/Simulink environment, employing S-functions to encapsulate control algorithms developed in C. The simulation results were compared to those from a linear digital controller. The dynamic performance of the proposed controllers was analyzed in view of quantitative criteria such as settling time, maximum overshoot, and metrics that evaluate the integral of the error signal: IAE, ISE, ITAE, and ITSE. The simulations demonstrated that the FCS-MPC method provides a significantly faster and more effective dynamic response, in addition to exhibiting a greater robustness, proving to be less susceptible to input disturbances. These characteristics highlight the technique's potential as a promising alternative for the control of converters in modern PV systems.

Keywords: Power Electronics; Predictive Control; FCS-MPC; Boost Converters; Photovoltaic Systems.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Evolução das chaves semicondutoras e tecnologias da eletrônica de potência..... | 16 |
| Figura 2 – Kit didático do <i>boost</i> PFC..... | 18 |
| Figura 3 – Gráficos de Publicações por ano com MPC e eletrônica de potência como temas. | 19 |
| Figura 4 – Circuito dos conversores (a) buck (b) <i>boost</i> (c) <i>buck-boost</i> | 24 |
| Figura 5 – Circuito fotovoltaico conectado à rede. | 26 |
| Figura 6 – Malha de controle linear com MPPT como referência..... | 26 |
| Figura 7 – Circuito de controle completo do <i>boost</i> fotovoltaico utilizando o MPPT como referência..... | 27 |
| Figura 8 – Malha de controle Genérica do controle por histerese..... | 28 |
| Figura 9 –Malha de controle linear com modulação PWM. | 30 |
| Figura 10 – Modelo Genérico dos controladores MPC..... | 31 |
| Figura 11 – Comportamento do FCS-MPC no tempo discreto. | 33 |
| Figura 12 – Gráfico de resposta transitória - Métricas de desempenho. | 37 |
| Figura 13 – Circuito do conversor <i>boost</i> convencional..... | 40 |
| Figura 14 – Circuito do conversor <i>boost</i> fotovoltaico..... | 42 |
| Figura 15 – Circuito equivalente do <i>boost</i> fotovoltaico com a chave ativa. | 44 |
| Figura 16 – Circuito equivalente do <i>boost</i> fotovoltaico com a chave desativada..... | 45 |
| Figura 17 – Quadro da tomada de decisão do FCS-MPC irrestrito. | 55 |
| Figura 18 – Quadro da tomada de decisão do FCS-MPC com restrição condicional. | 57 |
| Figura 19 – Quadro da tomada de decisão do FCS-MPC com extensão do horizonte de predição..... | 59 |
| Figura 20 – Circuito do conversor <i>boost</i> fotovoltaico em espaço de simulação. | 61 |
| Figura 21 – Simulação do Modelo linearizado por pequenos sinais..... | 61 |
| Figura 22 – Resultado Gráfico das variações em torno do ponto de operação separadamente. | 62 |
| Figura 23 – Resultado Gráfico das variações em torno do ponto de operação simultaneamente. | 63 |
| Figura 24 – Implementação do algoritmo em S-Function..... | 64 |

| | |
|---|----|
| Figura 25 – Resultado da aplicação do controle com a função custo do erro quadrático..... | 65 |
| Figura 26 – Gráfico de IAE x N x t' | 67 |
| Figura 27 – Gráfico de ISE x N x t' | 67 |
| Figura 28 – Gráfico de ITAE x N x t' | 68 |
| Figura 29 – Gráfico de ITSE x N x t' | 68 |
| Figura 30 – Gráfico de Máximo Sobressinal x N x t' | 69 |
| Figura 31 – Resultado da aplicação do controle com a função custo com restrição condicional. | 70 |
| Figura 32 – Gráfico comparativo entre o FCS-MPC Quadrático e o Condicional..... | 71 |
| Figura 33 – Gráfico de IAE x N x λ | 73 |
| Figura 34 – Gráfico de ISE x N x λ | 73 |
| Figura 35 – Gráfico de ITAE x N x λ | 74 |
| Figura 36 – Gráfico de ITSE x N x λ | 74 |
| Figura 37 – Resultado do controle com a função custo com restrição de tensão e horizonte estendido..... | 76 |
| Figura 38 – Comparação dos três algoritmos FCS-MPC propostos..... | 76 |
| Figura 39 – Gráfico de comparação FCS-MPC e controle linear..... | 78 |
| Figura 40 – Frequência de chaveamento do FCS-MPC ao longo da simulação..... | 79 |
| Figura 41 – Comparativo reação dos controles a perturbações..... | 81 |
| Figura 42 – ISE x N x λ - $f_{MPC} = 300kHz$ | 82 |
| Figura 43 – ITSE x N x λ - $f_{MPC} = 300kHz$ | 83 |
| Figura 44 – Gráfico de comparação FCS-MPC com $f_{MPC} = 300kHz$ e controle linear. | 84 |
| Figura 45 – Frequência de chaveamento do FCS-MPC ao longo da simulação. - $f_{MPC} = 300kHz$ | 85 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Consulta realizada na Plataforma <i>Web of Science</i> | 18 |
| Tabela 2 – Efeitos de aumentar os ganhos K_P , K_I e K_D na dinâmica da resposta ao degrau. | 29 |
| Tabela 3 – Comparação do FCS-MPC e CCS-MPC. | 32 |
| Tabela 4 – Parâmetros escolhidos para o conversor boost. | 43 |
| Tabela 5 – Índices de desempenho para o FCS-MPC - Erro Quadrático..... | 65 |
| Tabela 6 – Comparação das soluções - Restrição Condicional | 70 |
| Tabela 7 – Índices de desempenho para o FCS-MPC – Restrição Condicional | 71 |
| Tabela 8 – Comparação das soluções - Restrição de tensão – horizonte estendido. | 75 |
| Tabela 9 – Índices de desempenho para o FCS-MPC – Restrição de Tensão | 75 |
| Tabela 10 – Síntese da resposta dinâmica do sistema. | 77 |
| Tabela 11 – Índices de desempenho para o controlador linear..... | 79 |
| Tabela 12 – Comparação Parâmetros de desempenho – Integral do Erro | 80 |
| Tabela 13 – Comparação das soluções - $f_{MPC} = 300kHz$ | 83 |
| Tabela 14 – Índices de desempenho para o FCS-MPC (300kHz) e o controlador linear. | 84 |
| Tabela 15 – Comparação Parâmetros de desempenho – Integral do Erro - $f_{MPC} = 300kHz$ | 85 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|---------|---|
| CA | Corrente Alternada |
| CC | Corrente Contínua |
| CC-CC | Conversão de Corrente Contínua para Corrente Contínua |
| CCM | Modo de Condução Contínua (<i>Continuous Conduction Mode</i>) |
| CCS | Conjunto de Controle Contínuo (<i>Continuous Control Set</i>) |
| DCM | Modo de Condução Descontínua (<i>Discontinuous Conduction Mode</i>) |
| DSMC | Controle por Modo Deslizante Direto (<i>Direct Sliding Mode Controller</i>). |
| DSP | Processador de Sinais Digitais (<i>Digital Signal Processor</i>) |
| FCS-MPC | Controle Preditivo Baseado em Modelo com Conjunto de Controle Finito (<i>Finite Control Set – Model Predictive Control</i>) |
| FPGA | Arranjo de Portas Programáveis em Campo (<i>Field Programmable Gate Array</i>) |
| FV | Fotovoltaico |
| IAE | Integral do valor Absoluto do Erro (<i>Integral Absolute Error</i>) |
| ISE | Integral do valor Quadrático do Erro (<i>Integral Squared Error</i>) |
| ITAE | Integral do Erro Absoluto multiplicado pelo Tempo (<i>Integral of Time-weighted Absolute Error</i>) |
| ITSE | Integral do valor Quadrático do Erro multiplicado pelo Tempo (<i>Integral of Time-weighted Squared Error</i>) |
| MPC | Controle Baseado em Modelo (<i>Model Predictive Control</i>) |
| MPPT | Seguimento do Ponto de Máxima Potência (<i>Maximum Power Point Tracking</i>) |
| MMGD | Micro e Minigeração Distribuída |
| NPC | Inversor com ponto neutro fixo (<i>Neutral Point-Clamped</i>) |
| PFC | Correção de fator de potência (<i>Power Factor Correction</i>) |
| PI | Proporcional-Integral |
| PID | Proporcional-Integral-Derivativo |
| PR | Proporcional-Ressonante |
| PWM | Modulação por Largura de Pulso (<i>Pulse Width Modulation</i>) |
| RNA | Redes Neurais Artificiais |
| VSI | Inversor de Fonte de Tensão (<i>Voltage-Source Inverter</i>) |

LISTA DE SÍMBOLOS

| | |
|-----------------|--|
| A_i | Matrizes de estado |
| B_i | Matrizes de entrada |
| C_i | Matrizes de saída |
| E_i | Matrizes de transmissão direta |
| V_0 | Tensão de saída do conversor |
| V_{pv} | Tensão sobre o painel / entrada do conversor |
| I_{pv} | Corrente do painel fotovoltaico |
| C | Capacitor |
| L | Indutor |
| R_C | Resistência série do capacitor |
| R_L | Resistência série do indutor |
| f_s | Frequência de chaveamento |
| $e(t)$ | Sinal do erro no tempo |
| D | Ciclo de trabalho |
| T_S | Período de amostragem |
| $x(k)$ | Vetor de entrada discreto |
| $y(k)$ | Vetor de saída discreto |
| $g(k)$ | Ação do controlador sobre a chave |
| T_d | Tempo de atraso do sistema |
| f_c, ω_c | Frequência de passagem (Hz e rad/s) |
| f_{mpc} | Frequência do algoritmo MPC |
| N | Horizonte de predição |
| t' | Tempo de duração da restrição |
| J | Função custo |
| λ | Coeficiente do termo restritivo de tensão |
| V_{pv}^* | Valor de referência da tensão de entrada do conversor |
| Δv_{pv} | Ondulação da tensão de entrada do conversor em regime permanente |
| t_s | Tempo de assentamento |
| t_p | Tempo de pico |
| M_p | Máximo sobressinal |

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 15 |
| 1.1 | MOTIVAÇÃO | 17 |
| 1.2 | OBJETIVOS | 19 |
| 1.2.1 | Geral..... | 19 |
| 1.2.2 | Específicos | 20 |
| 1.3 | ORGANIZAÇÃO DO TRABALHO..... | 20 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 22 |
| 2.1 | CONVERSORES DE POTÊNCIA..... | 22 |
| 2.1.1 | Conversores CC-CA | 23 |
| 2.1.2 | Conversores CC-CC..... | 23 |
| 2.2 | SISTEMAS FOTOVOLTAICOS E MPPT | 25 |
| 2.3 | CONTROLE CLÁSSICO DE CONVERSORES ESTÁTICOS..... | 27 |
| 2.4 | CONTROLE PREDITIVO BASEADO EM MODELO | 30 |
| 2.4.1 | Conjunto de Controle Contínuo (CCS) e Finito (FCS) | 31 |
| 2.4.2 | Níveis de Abstração da Simulação | 35 |
| 2.5 | PARÂMETROS PARA ANÁLISE DE DESEMPENHO | 35 |
| 3 | MODELAGEM E CONTROLE PREDITIVO BASEADO EM MODELO | 39 |
| 3.1 | MODELAGEM DO CONVERSOR BOOST FOTOVOLTAICO | 39 |
| 3.1.1 | Conversor boost convencional | 39 |
| 3.1.2 | Adaptação para o boost fotovoltaico..... | 42 |
| 3.1.3 | Modelo em espaço de estados..... | 43 |
| 3.1.4 | Discretização do Modelo | 47 |
| 3.1.5 | Linearização e Modelo Médio por Pequenos Sinais | 49 |
| 3.1.6 | Método de Validação do Modelo | 51 |
| 3.1.7 | Proposta de Controle Linear..... | 52 |
| 3.2 | PROPOSTAS DE CONTROLE FCS-MPC | 53 |
| 3.2.1 | Controle por Erro Quadrático Sem Restrições | 54 |
| 3.2.2 | Controle com Restrição Condicional..... | 55 |
| 3.2.3 | Controle com Restrição na Tensão e Horizonte de Predição Estendido | 57 |
| 4 | RESULTADOS..... | 60 |
| 4.1 | AMBIENTE DE SIMULAÇÃO | 60 |
| 4.1.1 | Validação do sistema..... | 61 |
| 4.1.2 | Preparação para aplicar o FCS-MPC | 63 |
| 4.2 | SIMULAÇÃO DO FCS-MPC COM ERRO QUADRÁTICO | 64 |
| 4.3 | SIMULAÇÃO DO FCS-MPC COM RESTRIÇÃO CONDICIONAL..... | 66 |
| 4.4 | SIMULAÇÃO DO FCS-MPC COM RESTRIÇÃO DE TENSÃO COM HORIZONTE DE PREDIÇÃO ESTENDIDO..... | 72 |
| 4.5 | COMPARAÇÃO COM O CONTROLE LINEAR..... | 77 |

| | | |
|----------|---|-----------|
| 4.6 | ENSAIO COM AUMENTO DA FREQUÊNCIA DE EXECUÇÃO | 81 |
| 5 | CONCLUSÕES E PROPOSTAS DE CONTINUIDADE | 86 |
| 5.1 | TRABALHOS FUTUROS..... | 87 |
| | REFERÊNCIAS..... | 88 |
| | APÊNDICES..... | 92 |

1 INTRODUÇÃO

A demanda global por energia tem apresentado, ao longo da última década, um crescimento contínuo, sendo impulsionado pelo consumo da energia elétrica, que cresceu a uma taxa duas vezes maior que a energia como um todo (IEA, 2025). No Brasil, em especial, o Plano Decenal de Expansão de Energia (PDE) 2034, realizado pela Empresa de Pesquisa Energética (EPE), projeta um avanço médio de 3,3% a.a. entre 2024 e 2034 da carga demandada no Sistema Interligado Nacional (SIN). Além disso, o plano ainda ressaltou a tendência de eletrificação do país, seja por meio do consumo na rede ou via Micro e Minigeração Distribuída (MMGD) (EPE, 2024).

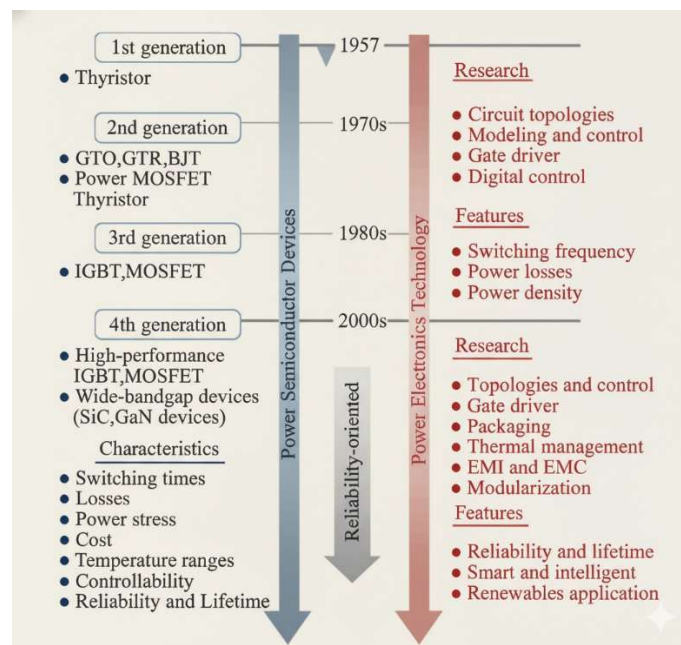
Em 2024, a matriz elétrica brasileira era composta quase 90% por energias de fontes renováveis, sendo a energia hidrelétrica dominante dentre elas. O Brasil encontra-se em um cenário favorável a redução de emissões de carbono, em consonância com as metas de transição energética global. No entanto, a forte dependência em uma só fonte, isto é, a hidráulica, torna-se um fator de risco, em função das mudanças climáticas, que tornam o regime pluviométrico menos previsível (ANA; MDR, 2024; EPE, 2024). Deste modo, a diversificação das fontes de energia, sobretudo das descentralizadas, se apresenta como uma opção para a segurança energética brasileira.

Nesse contexto, a energia solar fotovoltaica tem se apresentado como a alternativa viável para sistemas MMGD, representando 97% da geração de micro e minigeração distribuída no Brasil (EPE; MME, 2025). Contudo, este método de geração é acompanhado a dois desafios principais, o primeiro deles é a correta integração das gerações de distribuídas com a rede elétrica interligada, a fim de garantir a estabilidade da rede frente a injeção intrinsecamente variável dos sistemas fotovoltaicos (FV) (ONS, 2024; SAUAIA, 2018). Em adição a essa primeira questão, o processo de conversão de energia de forma eficiente e inteligente se impõe com uma barreira a ser ultrapassada (TANG, YANG e BLAABJBERG, 2022).

É neste cenário que surge a necessidade por conversores de potência e estratégias de controle efetivas atreladas a eles, atuando não só como uma medida de eficiência energética, mas também como a interface entre as fontes geradoras de energia e a rede elétrica interligada (OLABI e ABDELKAREEM, 2022). Em sistemas MMGD ou propriamente isolados (*off-grid*), essa necessidade se torna ainda mais evidente, seja

para se adequar o nível de tensão, armazenar a energia excedente ou tão somente para distribuí-la. Tang, Yang e Blaaberg (2022) enfatizam que há uma evolução rápida e consistente dos conversores de potência desde o surgimento das primeiras chaves semicondutoras, como ilustrado na Figura 1. No cenário atual, a busca por controle não se limita a estabilidade da fonte renovável, mas também da rede elétrica interligada, em grande parte pelo que se chama de controle orientado a confiabilidade (do inglês, *reliability-oriented*) (KARAFOTIS, EVANGELOPOULOS e GEORGILAKIS, 2022; JOOSHAKI, et al., 2020).

Figura 1 – Evolução das chaves semicondutoras e tecnologias da eletrônica de potência.



Fonte: adaptado de (TANG, YANG e BLAABJBERG, 2022).

Ao longo das últimas décadas uma série de técnicas de controle já foram apresentadas para lidar com esses conversores, sobretudo para lidar com um desafio inerente a eles (LIANG *et al.*, 2018). A natureza não-linear desses conversores acaba dificultando a atuação de controladores clássicos e lineares, como o PID, que demonstram limitações em cenários mais críticos (i.e., aplicações com requisitos dinâmicos rigorosos, cenários em que há variação de carga ou até presença de perturbações e distúrbios no sistema). Para além da não-linearidade, a depender da topologia do conversor CC-CC, um zero de fase não-mínima surge na modelagem, gerando desta maneira ainda mais um empecilho para os controles mais

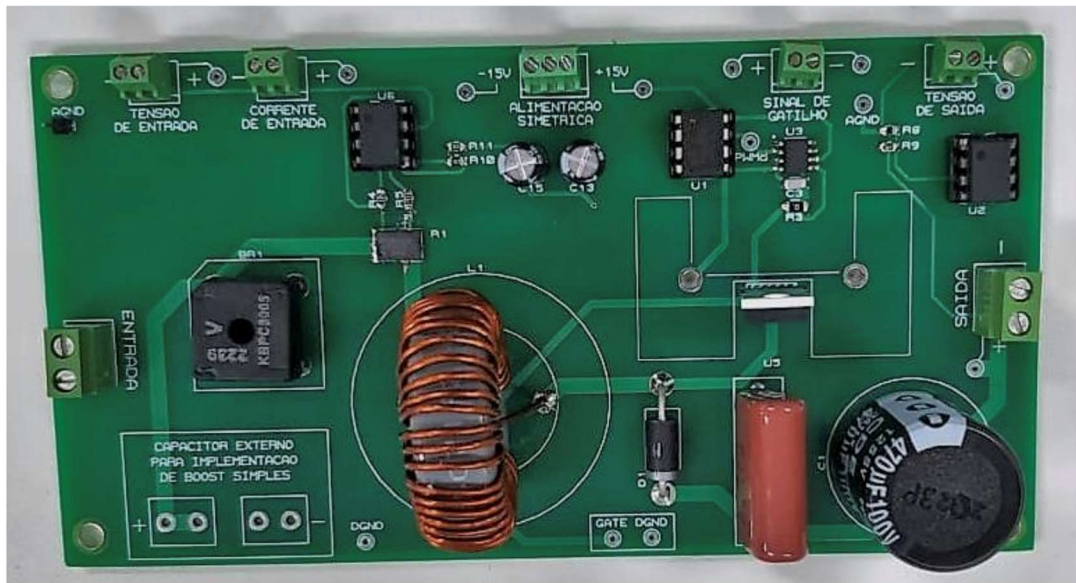
convencionais (DE KEYSER, BONILLA e IONESCU, 2006). Assim sendo, a maior parte das pesquisas, iniciadas ainda no século passado, focam seus esforços na procura por métodos não-lineares e inteligentes, que consigam lidar bem com as limitações dos conversores (ESCOBAR *et al.*, 1999).

Em anos mais recentes, que surge o controle preditivo baseado em modelo (*Model Predictive Control* – MPC), como alternativa para os conversores estáticos (LIANG *et al.*, 2018). Segundo Rodriguez *et al.* (2012), o MPC se destaca pela simplicidade de implementação e pelo seu desempenho positivo frente às adversidades encontradas pelos demais controladores.

1.1 Motivação

O presente trabalho tem como pedra fundamental a pesquisa realizada por Costa (2025) em sua monografia, servindo como base para a metodologia de estudo e análise de desempenho dos controladores encontrados. Com base em seu trabalho com o MPC aplicado aos conversores CC-CC abaixadores (*buck*), se tornou viável o estudo para outras topologias, no caso deste trabalho, os conversores do tipo elevadores (*boost*) voltados para uma aplicação fotovoltaica. O intuito desta pesquisa, além de explorar uma nova topologia, foi de dar um passo adiante no caminho da aplicação prática do controle preditivo baseado em modelo.

Para aproximar este estudo de uma futura validação prática, a parametrização do conversor *boost* utilizou como referência a planta didática desenvolvida por Melo (2023), ilustrada na Figura 2. No entanto, como o protótipo original foi projetado para atuar como pré-regulador de fator de potência (PFC) em modo de condução descontínua (DCM), o projeto passou por adequações para garantir a operação em modo contínuo de condução (CCM), essencial para a implementação de um modelo mais simples para o algoritmo MPC. Adaptações foram feitas para emular um conversor *boost* voltado para aplicações FV. Além disso, ressalta-se que os níveis de tensão adotados nas simulações foram estritamente condicionados pelos limites operacionais físicos da bancada, respeitando-se as restrições de tensão e corrente dos componentes do *hardware* preexistente.

Figura 2 – Kit didático do *boost* PFC.

Fonte: (MELO, 2023)

Em termos de aplicação dos métodos em ambiente de simulação, também houve um cuidado com o progresso para a avaliação experimental futura dos algoritmos MPC. Todos os códigos foram desenvolvidos através de S-functions, que permitem encapsular códigos nas linguagens C ou C++ às simulações em Matlab/Simulink. Desta forma, é possível testar um código de mais baixo nível de abstração que se aproxima do que será executado em *hardwares* dedicados, como os dispositivos de Processamento Digital de Sinais (DSP) por exemplo.

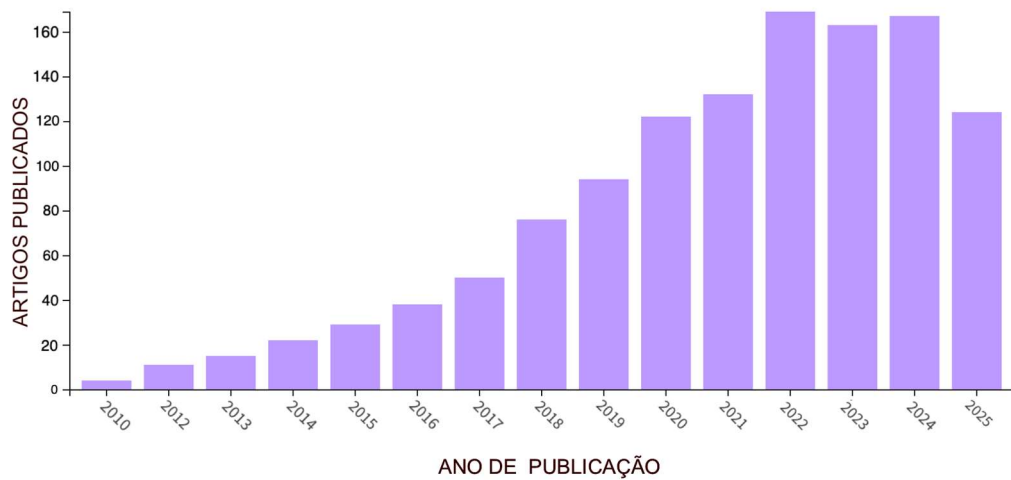
Para melhor se compreender o cenário acadêmico do controle MPC aplicado a conversores de potência, foi feita uma consulta através da plataforma Web of Science, como visto na Tabela 1, por artigos científicos que tratam tanto do tema “MPC” quanto de “eletrônica de potência”. O resultado desta busca pode ser visualizado na Figura 3, que mostra um crescimento consistente do tema nos últimos 15 anos.

Tabela 1 – Consulta realizada na Plataforma *Web of Science*.

| Consulta (campo “Query” no WoS) | Data | Nº de Resultados |
|---|------------|------------------|
| ("Finite Control Set" OR "MPC") and ("Boost" OR "Power Electronics") (All fields) | 21/07/2025 | 1245 |

Fonte: Elaborado pelo Autor (2025).

Figura 3 – Gráficos de Publicações por ano com MPC e eletrônica de potência como temas.



Fonte: Elaborado pelo Autor (2025).

Tendo em vista esse resultado, é importante de se analisar o que possibilitou este aumento dos trabalhos com esta temática. Estudos apontam que os avanços da capacidade computacional dos sistemas microcontrolados, DSPs e FPGAs, tem tornado controles digitais complexos como o MPC viáveis para aplicações de uma frequência relativamente alta (RODRIGUEZ *et al.*, 2012). Sendo Assim, abre-se o caminho para mais pesquisas, como é o caso deste trabalho.

1.2 Objetivos

1.2.1 Geral

Desenvolver e validar, em ambiente de simulação, um algoritmo de controle preditivo baseado em modelo com conjunto de controle finito (FCS-MPC), demonstrando sua viabilidade como alternativa aos controles clássicos e lineares para conversores *boost* em aplicações fotovoltaicas.

1.2.2 Específicos

Para que se alcance o objetivo geral foram estipulados alguns passos e objetivos em específico, eles são como se segue:

1. Realizar e validar a modelagem do conversor *boost* para aplicação fotovoltaica, tanto no espaço de estados quanto por um modelo médio;
2. Desenvolver o algoritmo MPC com diferentes funções de custo e adicionar restrições de diferentes naturezas;
3. Analisar os algoritmos MPC através de critérios de desempenho objetivos;
4. Comparar os resultados obtidos através do MPC com o desempenho obtidos com controladores lineares.

1.3 Organização do Trabalho

Este Trabalho de Conclusão de Curso foi organizado da seguinte maneira:

- **Capítulo 2 – Fundamentação Teórica:** É apresentada a base teórica sobre a qual o trabalho foi desenvolvido, descrevendo inicialmente os conversores de potência, a aplicação em sistemas FV e os algoritmos clássicos de controle a eles aplicados. Na sequência, são descritos métodos de controle inteligente propostos na literatura e entra-se em específico no funcionamento do MPC. Para finalizar, são mostrados os critérios de desempenho a serem utilizados durante este trabalho.
- **Capítulo 3 – Modelagem e Controle Preditivo Baseado em Modelo:** Apresenta o desenvolvimento do trabalho, abrangendo a modelagem matemática do conversor *boost* fotovoltaico e a implementação dos modelos em ambiente de simulação. Em seguida, é descrita a obtenção do modelo preditivo no domínio do tempo discreto e o processo de formulação do controle FCS-MPC.
- **Capítulo 4 – Resultados:** Reúne e analisa os resultados obtidos com a aplicação do controlador FCS-MPC ao conversor boost modelado. São apresentados os testes de validação do modelo no espaço de estados e os

resultados de simulação considerando diferentes configurações do algoritmo, com e sem restrição de função de custo. A análise é conduzida com base nas métricas de desempenho definidas no Capítulo 2, permitindo avaliar quantitativamente o comportamento dos controladores e compará-los com estratégias clássicas.

- **Capítulo 5 – Conclusões e Propostas de Continuidade:** As conclusões oriundas desta pesquisa são evidenciadas, retornando aos objetivos da pesquisa e avaliando o sucesso do trabalho. Por fim são sugeridas de temas para futuros trabalhos com foco no avanço do controle de conversores estáticos e dos algoritmos MPC.

2 FUNDAMENTAÇÃO TEÓRICA

O propósito deste capítulo é fundamentar uma base teórica que revise os conceitos essenciais para o correto entendimento do presente trabalho. Optou-se por começar a explanação pelo objeto de estudo, os conversores estáticos, entrando mais especificamente no funcionamento geral daqueles denominados "CC-CC não isolados". Em seguida, passa-se a analisar as estratégias de controle encontradas na literatura, iniciando pelos controles clássicos, brevemente explorando os controles inteligentes e, para finalizar, comentando sobre o algoritmo chave deste trabalho, o MPC. Ao final do capítulo, um subtópico é reservado para os parâmetros de desempenho escolhidos para este estudo.

2.1 Conversores de Potência

A eletrônica de potência é o pilar da conversão e do gerenciamento de energia nos sistemas elétricos modernos, desde fontes de alimentação de baixa potência em dispositivos eletrônicos até a interface de alta potência de fontes de energia renovável com a rede elétrica. Os dispositivos que possibilitam tudo isso são os conversores de potência projetados para garantir a qualidade, eficiência, estabilidade e controle da energia elétrica (RODRIGUES, 2020).

De forma ampla, os conversores podem ser classificados como isolados e não isolados, a depender da existência ou não do isolamento galvânico entre a entrada e a saída do circuito. Em sua maior parte, os circuitos isolados utilizam os transformadores como artifício para obter este isolamento, isto é, não permitindo um caminho comum de corrente entre os dois estágios do conversor. Por outro lado, os conversores não isolados, o foco deste trabalho, são caracterizados por elementos e aspectos construtivos mais simples e menos onerosos financeiramente quando comparados aos isolados (CUNHA, 2021).

Os conversores podem ser agrupados ainda em quatro categorias CA-CA, CA-CC, CC-CA e CC-CC, como definido por Mohan, Undeland e Robbins (2003). Para as aplicações de geração fotovoltaica, destaca-se o uso dos conversores do tipo CC-CC e CC-CA como interface entre o arranjo fotovoltaico e a rede elétrica (KOURO *et al.*,

2015). Portanto, nas subseções a seguir será apresentada uma breve explanação dos conversores CC-CA e CC-CC, abordando seus princípios de funcionamento, principais topologias e aplicações no contexto de sistemas de geração fotovoltaica conectados à rede elétrica.

2.1.1 Conversores CC-CA

Comumente chamados de inversores, convertem energia contínua (CC) em energia alternada (CA). Por esse motivo, eles são indispensáveis em sistemas que transferem potência de fontes CC (como baterias ou painéis fotovoltaicos) para o domínio alternado, permitindo o funcionamento de equipamentos em CA ou até a distribuição de energia na rede elétrica (KOURO *et al.*, 2015).

As topologias fundamentais são os Inversores Fonte de Tensão (VSI - *Voltage Source Inverters*), como o inversor em ponte completa (*Full-bridge*), que são a base para aplicações de conexão monofásica de módulos fotovoltaicos à rede (KJAER, PEDERSEN e BLAABJERG, 2005).

Para aplicações de média e alta potência, os inversores multinível são amplamente preferidos, por obter uma melhor qualidade de onda e operar com uma menor frequência de chaveamento. As topologias clássicas de multinível incluem o NPC (*Neutral-Point Clamped*), o *Flying Capacitor* e o CHB (*Cascaded H-Bridge*) (RODRIGUEZ, LAI e PENG, 2002). Essas características tornam os inversores multinível especialmente adequados para a conexão de sistemas fotovoltaicos de grande escala à rede elétrica (KOURO *et al.*, 2015).

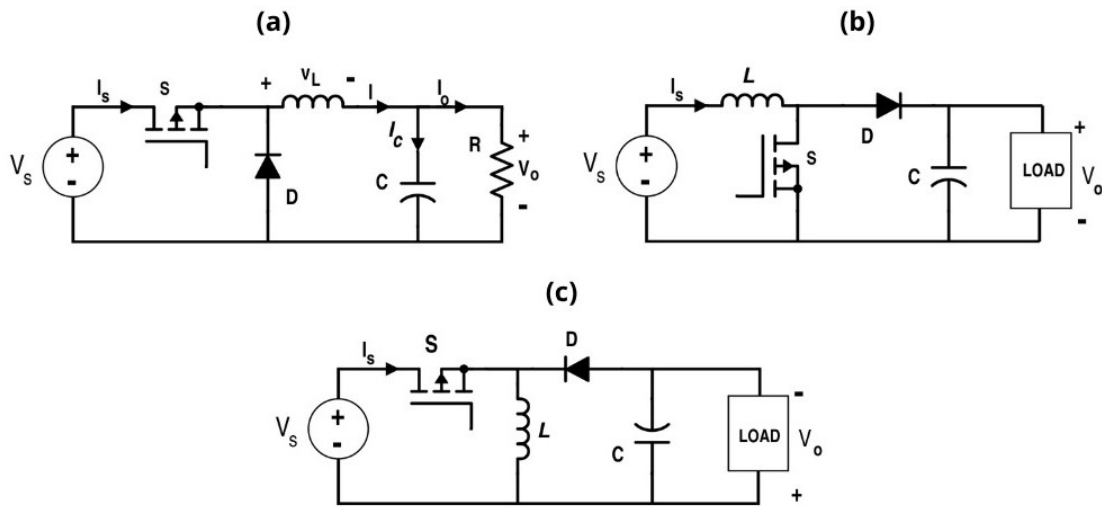
2.1.2 Conversores CC-CC

Conversores CC-CC (DC-DC) são utilizados para converter uma tensão contínua (CC) de uma fonte para outra tensão CC, seja para elevar (*Boost*), rebaixar (*Buck*) ou inverter a polaridade (*Buck-Boost*). Esses conversores são amplamente aplicados em sistemas embarcados, fontes chaveadas, eletrônica de potência e veículos elétricos, onde é essencial manter regulação e estabilidade da tensão de saída. Seu controle é geralmente realizado por técnicas de modulação por largura de pulso (PWM), mas

novas técnicas como o FCS-MPC (RODRIGUEZ *et al.*, 2012) e o controle por modo deslizante direto (DSMC) dispensam o uso do modulador PWM (AL-WESABI *et al.*, 2022).

Entrando mais em detalhes sobre topologias específicas dos conversores CC-CC, ilustradas na Figura 4, observa-se que o conversor *buck* é empregado na redução da tensão de saída em relação à tensão de entrada, sendo, portanto, classificado como um conversor abaixador de tensão (ABOLHASANI *et al.*, 2016; COSTA, 2025; RIBEIRO JR., 2023). O conversor *boost*, tem como finalidade elevar a potencial e possibilitar a entrega de energia eficiente mesmo para sistemas de baixa tensão (FOROUZESH *et al.*, 2017; ISLAM *et al.*, 2020). Já o conversor *buck-boost* combina as características dos dois anteriores, possibilitando tanto a elevação quanto a redução da tensão, além disso ele inverte a polaridade da tensão de saída (ISLAM *et al.*, 2020; HARZIG e GRAINGER, 2024).

Figura 4 – Circuito dos conversores (a) buck (b) boost (c) buck-boost.



Fonte: Adaptado de (RIBEIRO JR., 2023)

Estes conversores podem operar em dois regimes distintos, a depender do comportamento da corrente no indutor. Para as operações nas quais a corrente do indutor é zerada durante o chaveamento, diz-se que o conversor está operando em modo de condução descontínua (DCM - *Discontinuous Conduction Mode*) (MELO,

2023). Em contraste, caso a corrente do indutor se mantenha acima de zero durante todo o ciclo de operação, o conversor está em modo de condução contínua (CCM – *Continuous Conduction Mode*), o que torna possível realizar uma análise linear dos mesmos (MOHAN, UNDELAND e ROBBINS, 2003).

O presente trabalho almeja aplicar o controle preditivo baseado em modelo em um conversor CC-CC do tipo *boost* em um cenário de energia solar fotovoltaica. Portanto, torna-se imprescindível compreender os sistemas fotovoltaicos e o algoritmo que o acompanha, o MPPT.

2.2 Sistemas Fotovoltaicos e MPPT

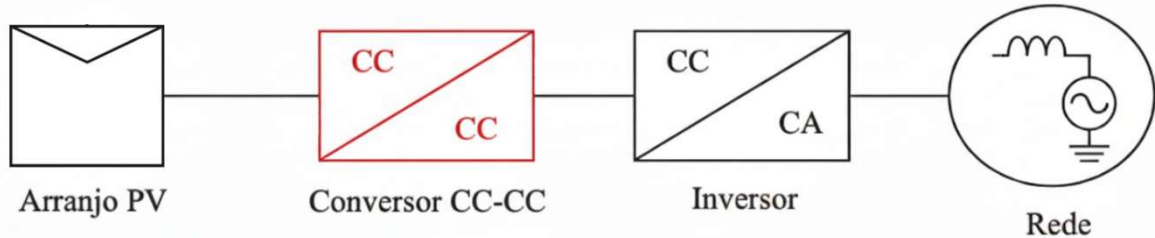
Ao longo das últimas décadas é notável o crescimento e a popularização da energia solar. Cenário este que também se observa no Brasil. De acordo com o Balanço Energético Nacional (BEM), realizado pela EPE em associação com o Ministério de Minas e Energia, a geração solar fotovoltaica cresceu apenas em 2024 39,6%, enquanto a sua capacidade instalada subiu em 28,1% em relação ao ano anterior (EPE; MME, 2025). Adicionalmente, projeta-se um crescimento de aproximadamente 7% a.a. nos próximos dez anos (EPE, 2024).

Neste sentido, é importante para a segurança energética brasileira que os sistemas FV sejam devidamente controlados e isto passa em grande parte pela a figura dos conversores estáticos. As topologias dos sistemas FV podem ser classificadas pelo número de estágios até a conexão com a rede elétrica. Alguns sistemas conectam os painéis fotovoltaicos diretamente aos inversores, caracterizando um estágio de conversão (RIBEIRO JR., 2023). Este trabalho se propõe a analisar um sistema com dois estágios, como ilustrado na Figura 5, onde é adicionado um conversor CC-CC, objeto desta pesquisa e por isso destacado em vermelho, entre o painel e o inversor. De acordo com Cunha (2021), o estágio CC-CC, no caso dos conversores isolados, leva ao isolamento galvânico do sistema, aumentando a segurança.

Kouro et al. (2015) explora o desenvolvimento das técnicas de conversão para essas topologias isoladas, como por exemplo de ponte-H e *Flyback*. De igual forma, o artigo destaca também as vantagens de se utilizar as topologias não isoladas, no

sentido de regular e elevar a tensão, além da possibilidade de realizar o controle para o Rastreamento do Ponto de Máxima Potência (MPPT).

Figura 5 – Circuito fotovoltaico conectado à rede.

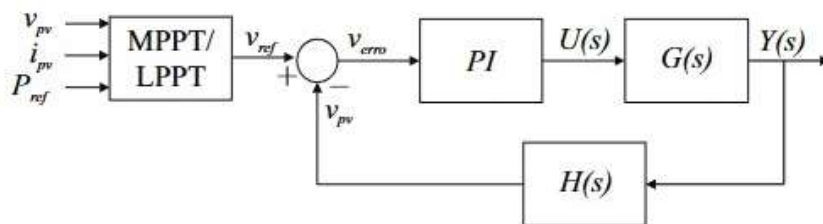


Fonte: Adaptado de Cunha (2021).

Apesar de não ser do escopo da presente pesquisa, o algoritmo MPPT (*Maximum Power Point Tracking*) é indispensável em sistemas FV, para que os painéis operem de forma eficiente em diferentes cenários de irradiância solar, temperatura e luminosidade, determinando a corrente e a tensão ideal para cada cenário. Assim, atinge-se a maior potência gerada (CUNHA, 2021; TANG, YANG e BLAABJBERG, 2022).

Em aplicações de controle em circuitos fotovoltaicos a tensão e corrente obtidas pelo MPPT, com o intuito de maximizar a potência gerada, servem de referência para uma malha de controle interna (CUNHA, 2021; RIBEIRO JR, 2023; TANG, YANG e BLAABJBERG, 2022; SANTOS *et al.*, 2020). A Figura 6 ilustra como se estrutura a malha de controle interna em um sistema FV, tendo como entrada a resposta do algoritmo MPPT.

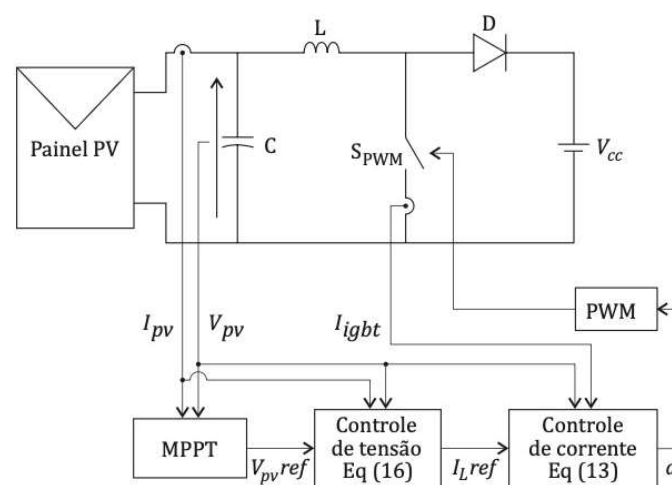
Figura 6 – Malha de controle linear com MPPT como referência.



Fonte: Adaptado de (SANTOS, COSTA, *et al.*, 2020).

O que este estudo se propõe a fazer é uma simplificação do esquema de controle mostrado na Figura 7. É feita uma abstração em relação ao algoritmo MPPT, assumindo um valor de tensão constante como referência, além da modelagem do painel fotovoltaico como uma fonte corrente, de modo que não se faz necessário controle de corrente. Resta, desta forma, unicamente o controle da tensão V_{pv} a ser realizado.

Figura 7 – Circuito de controle completo do *boost* fotovoltaico utilizando o MPPT como referência.



Fonte: Retirado de (CUNHA, 2021).

2.3 Controle Clássico de Conversores Estáticos

Como já mencionado e reforçado na literatura, o controle dos conversores é um desafio para os controladores. Isso é especialmente verdade para os controladores clássicos. De acordo com Rodrigues (2020), duas técnicas de controle clássico foram particularmente mais pesquisadas. A primeira delas é o controle por histerese e a segunda trata-se dos controladores lineares que utilizam um modulador para atuar nos conversores chaveados.

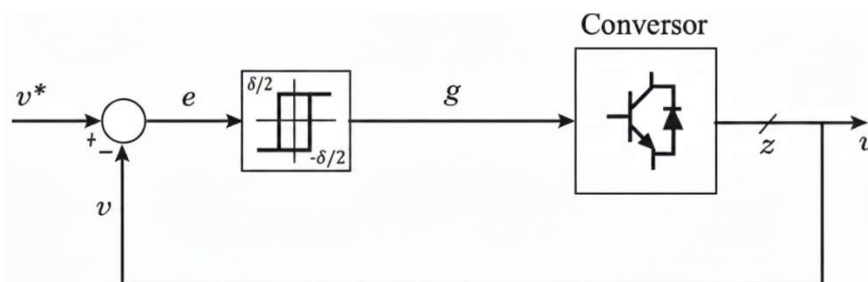
O termo ‘histerese’ tem origem no grego e significa algo como “atraso” ou “retardo” e isso categoriza bem o conceito de histerese nas mais diversas áreas de conhecimento, atendo-se à engenharia elétrica, a presença da histerese ocorre em temáticas desde o eletromagnetismo, por meio dos materiais ferromagnéticos, até

aplicações em eletrônica analógica, presente no comparador *Schmitt Trigger* (BOYLESTAD, 2012; UMANS, 2014; PERTENCE JR., 2003).

Em controle, o método por histerese é simples, utilizando-se de uma faixa de valores ao redor da referência chamada de banda de histerese (δ), o controle comuta o estado da chave no momento em que o sinal de saída ultrapassa os valores superior ou inferior da banda. Desta maneira, o controlador atua com um certo atraso, mas é capaz de manter a saída contida dentro da faixa de histerese. A Figura 8 ilustra bem o funcionamento do sistema de controle, o sinal de erro é avaliado pelo controlador e caso ele ultrapasse a banda de histerese, o controlador comuta o estado da chave (g).

Este método de controle acaba sendo acompanhado por um empecilho quando aplicado a conversores de potência. Apesar de apresentar uma resposta dinâmica satisfatória, para que seja possível realizar este controle digital, faz-se necessário uma frequência alta de amostragem (RODRIGUES, 2020).

Figura 8 – Malha de controle Genérica do controle por histerese.



Fonte: Adaptado de (RODRIGUES, 2020).

De certa forma, o *modus operandi* do controle por histerese é oposto ao que se espera do controle MPC. Enquanto um aguarda o sinal sair de uma certa faixa de valores para atuar, o outro se propõe a prever essa variação de modo que possa atuar antes mesmo que ela ocorra.

Outro método clássico amplamente aplicado em conversores na literatura são os controladores lineares em geral na forma de controladores PI, PID e compensadores de fase (i.e., avanço, atraso e avanço e atraso). Como definido por Ogata (2011) e Dorf e Bishop (2017), a estrutura dos controladores PI (Proporcional-Integrativo) e PID (Proporcional-Integrativo-Derivativo) é baseada no somatório dos ganhos:

proporcional (K_P), integral (K_I) e derivativo (K_D), com cada ação de controle tendo um impacto diferente sobre a dinâmica do sistema. De modo que o impacto do aumento de cada um desses ganhos na resposta ao degrau pode ser generalizado, conforme apresentado na Tabela 2.

Tabela 2 – Efeitos de aumentar os ganhos K_P , K_I e K_D na dinâmica da resposta ao degrau.

| | Máximo Sobressinal Percentual | Tempo de Acomodação | Erro em Regime Permanente |
|------------------|--------------------------------------|----------------------------|----------------------------------|
| Aumento de K_P | Aumenta | Impacto mínimo | Diminui |
| Aumento de K_I | Aumenta | Aumenta | Zera |
| Aumento de K_D | Diminui | Diminui | Não impacta |

Fonte: Traduzido de (DORF e BISHOP, 2017).

O controle por compensação de fase, por outro lado, está intimamente ligado com a análise no domínio da frequência e pode ser composto de duas formas, com um polo e um zero para os casos de avanço e de atraso de fase ou por dois polos e dois zeros para o caso do controlador por avanço e atraso, que seria o equivalente cascatear os dois controladores de primeira ordem em série (OGATA, 2011; DORF e BISHOP, 2017). O projeto destes controladores está intimamente ligado com a análise dos sistemas no domínio da frequência.

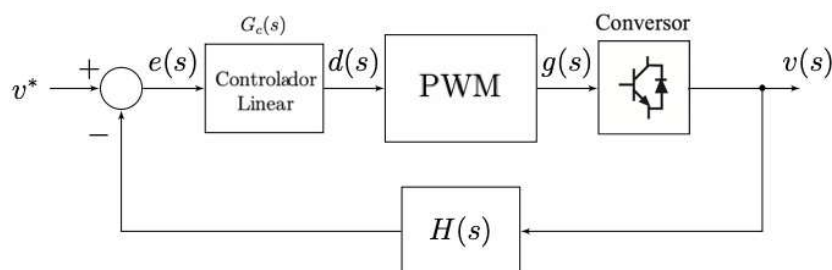
Em relação ao emprego do controle linear em conversores de potência há um empecilho claro que advém do fato que os conversores são naturalmente não lineares e variantes no tempo em função da comutação das chaves semicondutoras. Portanto, para que seja possível projetar um controle adequado é necessário fazer um modelo médio e linearizado do conversor (COSTA, 2025). Cunha (2021) e Zonetti et al. (2021) afirmam que a ação derivativa é tradicionalmente evitada nos controladores de potência em função da necessidade de se adicionar um filtro de saída para atenuar os ruídos de alta-frequência. Desta forma, os controladores mais comumente utilizados são PI como nos estudos de Haque, Wolfs e Alahakoon (2017) e Lahoti Shah e Kadam (2016). Além disso, os controles lineares dependem de um modulador por largura de pulso (PWM), como evidenciado na Figura 9.

O controle linear serve comumente como ponto de partida em termos de critério de desempenho dinâmico, de modo que usualmente compara-se o resultado do

controlador proposto pelo estudo ou artigo com um controlador clássico (COSTA, 2025; CUNHA, 2021 ; SGUAREZI FILHO *et al*, 2021; YOUNG e RODRIGUEZ, 2013).

Além disso, os controladores lineares também podem ser utilizados em junção com outros métodos, em um sistema de controle de malha dupla, como é o caso em Po *et al.* (2018), em que o controle PI é empregado como malha externa e serve de referência para um algoritmo FCS-MPC para a malha interna.

Figura 9 –Malha de controle linear com modulação PWM.



Fonte: Adaptado de (COSTA, 2025).

Para além destes modelos de controladores clássicos, o controle linear também figura em técnicas mais modernas de controle. É o caso, por exemplo, do controlador proporcional-ressonante (PR) presente no artigo de Ribeiro *et al.* (2018) aplicado a inversores meia-ponte.

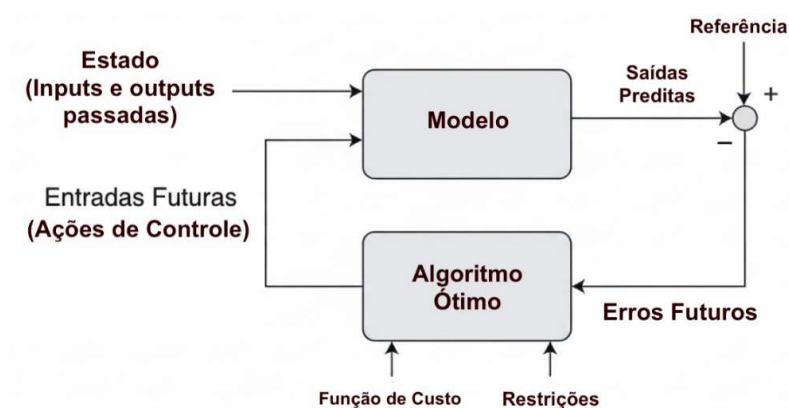
2.4 Controle Preditivo Baseado em Modelo

Em seu artigo introdutório sobre o controle preditivo baseado em modelo, Bordons e Montero (2015) descrevem de forma resumida a teoria que sustenta o MPC e, para além disso, mostram o funcionamento matemático deste algoritmo. Na realidade, os autores esclarecem de imediato que o MPC não se trata de um método de controle específico. Trata-se de um termo guarda-chuva para todo tipo de controle que se utiliza do modelo de um sistema para prever seus estados futuros e atua neste sistema com base na minimização de uma determinada função de custo. Para as mais diversas aplicações dos algoritmos MPC possíveis, (BORDONS e MONTERO, 2015) descrevem que somente três aspectos do controle mudam, o modelo da planta, a

escolha da função custo e como é feita a otimização. O diagrama da Figura 10 apresenta, de forma generalizada, como o algoritmo do MPC pode funcionar para o controle de um determinado sistema. Na prática, a técnica se impõe como um problema constante de otimização e minimização de uma função custo tendo que ainda operar em tempo real.

Este tipo de controlador teve sua origem na indústria de processos ainda no final da década de 1970 (RICHALET *et al.*, 1978), onde os sistemas conseguem operar em frequências mais baixas, especialmente quando comparados com aplicações de eletrônica de potência. A partir dos anos 1980, a técnica começou a ser aplicada a sistemas conversores de energia elétrica (HOLTZ e STADTFELD, 1983). No entanto, devido ao elevado esforço computacional, ele foi aplicado inicialmente a sistemas de frequência mais baixa (RODRIGUES, 2020).

Figura 10 – Modelo Genérico dos controladores MPC.



Fonte: Adaptado de (BORDONS e MONTERO, 2015)

2.4.1 Conjunto de Controle Contínuo (CCS) e Finito (FCS)

O Controle Preditivo Baseado em Modelo (MPC) aplicado a conversores de potência é categorizado em duas abordagens principais: o Conjunto de Controle Contínuo (CCS-MPC) e o Conjunto de Controle Finito (FCS-MPC). A diferença fundamental reside na etapa de atuação: enquanto o CCS gera um sinal contínuo (ciclo de trabalho) que necessita de um modulador PWM para ser sintetizado, o FCS aproveita a natureza discreta do conversor para atuar diretamente nos estados de

comutação das chaves a cada instante de amostragem (RODRIGUEZ *et al.*, 2012; KOURO *et al.*, 2009).

Historicamente, a implementação do FCS-MPC apresentou desafios técnicos não pela complexidade matemática, que é intuitiva, mas pela exigência de elevadas taxas de amostragem para garantir um desempenho comparável aos métodos modulados. Nesse contexto, o avanço dos processadores digitais (DSPs e FPGAs) nas últimas décadas foi determinante para viabilizar essa estratégia em aplicações de eletrônica de potência, permitindo cálculos rápidos dentro de períodos de amostragem reduzidos (KOURO *et al.*, 2009; ZUCUNI *et al.*, 2018).

Costa (2025) sintetiza as distinções operacionais entre os métodos, destacando o comportamento da frequência de chaveamento. O CCS-MPC, devido ao uso do modulador, opera com frequência de chaveamento fixa, resultando em um espectro harmônico previsível e semelhante aos controladores clássicos (como o PI). Em contrapartida, o FCS-MPC apresenta frequência de chaveamento variável, pois a mudança de estado das chaves depende estritamente da minimização da função custo a cada ciclo de operação. Embora Preindl e Bolognani (2013) citem que o CCS pode ser mais versátil devido ao espectro harmônico fixo, o FCS destaca-se pela resposta dinâmica superior e implementação direta sem moduladores. Ademais, o FCS-MPC tem uma maior facilidade de implementação, sobretudo na incorporação de restrições ao algoritmo. Na Tabela 3, encontra-se um resumo da comparação entre os dois algoritmos.

Tabela 3 – Comparação do FCS-MPC e CCS-MPC.

| Característica | FCS-MPC | CCS-MPC |
|---------------------------------------|---|--|
| Natureza de atuação | Sinal Discreto. Sem necessidade de modulador. | Sinal contínuo. Com necessidade de modulador. |
| Frequência de amostragem | Alta. Comumente opera em altas frequências. | Baixa. Comumente opera com valores menores que o FCS-MPC. |
| Frequência de chaveamento | Variável. | Fixa. |
| Esforço computacional (para $N = 1$) | Baixo. Independente da presença de restrições. | Baixo para casos irrestritos. Alto para casos restritos. |

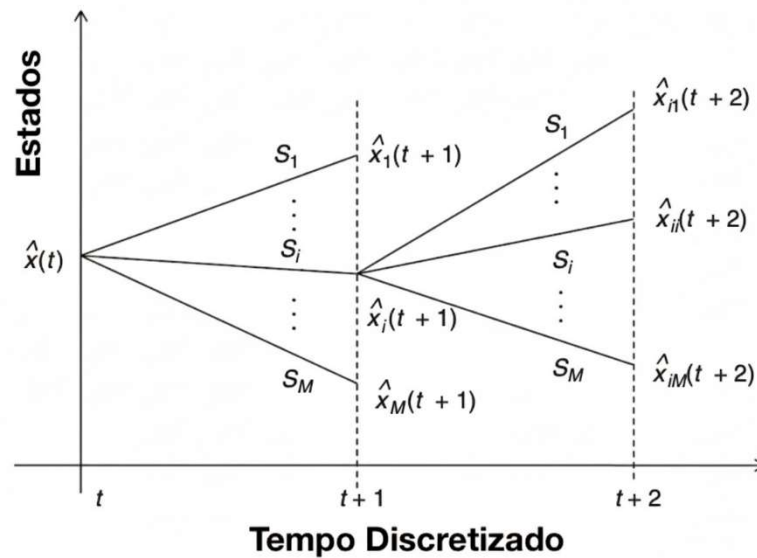
Fonte: (COSTA, 2025).

Rodriguez et al. (2012), assim como Vasquez et al. (2014), destacam que o FCS-MPC se tornou uma alternativa simples e eficiente para se realizar o controle de conversores de potência, que por sua natureza chaveada têm um número limitado de estados possíveis a cada instante de tempo discreto. Por exemplo, um conversor com apenas uma chave controlada será limitado a duas possibilidades para o próximo instante discreto ($k+1$). No entanto, ao passo que se avança por mais iterações, o número de possibilidades cresce exponencialmente. Portanto, temos a seguinte relação:

$$N_{\text{estados}} = M^{N_H} \quad (2.1)$$

Onde M é igual ao número de possibilidades a serem avaliadas a cada instante de tempo e N_H é o horizonte de predição. A Figura 11 ilustra bem esse funcionamento, que se assemelha ao de uma árvore de decisão.

Figura 11 – Comportamento do FCS-MPC no tempo discreto.



Fonte: Adaptado de (KOURO et al., 2009).

Para além da simplicidade, o FCS-MPC se sobressai perante os métodos de controle mais tradicionais pela possibilidade da adição de restrições, incorporação de múltiplas variáveis e metas distintas dentro de uma só função custo. Este algoritmo ainda pode operar em horizontes de predição mais longos, que por um lado, acarretam

em um custo computacional considerável, em razão do crescimento exponencial evidenciado anteriormente, mas por outro, como reforçado por Bordons e Montero (2015), resultam em uma melhora significativa da performance em regime permanente e diminuem a frequência de chaveamento e as distorções harmônicas da corrente. Ademais, a extensão do horizonte de predição permite, em casos de modelos com zero de fase não-mínima, que o controle enxergue os estados após a dinâmica inversa à mudança de referência, conseguindo assim aplicar um controle preciso e eficiente.

Os algoritmos MPC já foram estudados nos mais diversos cenários da eletrônica de potência e têm se provado como alternativa viável para a realização do controle de conversores de energia. Segundo Scoltock, Geyer e Madawala (2013), o MPC é especialmente útil em sistemas de média tensão, em função da sua capacidade de manter a frequência de chaveamento mais baixa, além de diminuir as perdas, enquanto mantém níveis aceitáveis de distorção harmônica.

A literatura acadêmica demonstra a versatilidade do FCS-MPC em diversas topologias de conversores CC-CC. Para o conversor *buck*, por exemplo, Azadi et al. (2023) explora o FCS-MPC em um conversor *buck* quadrático, com o intuito de aumentar a eficiência energética em data centers. Já para o *buck-boost*, Restrepo et al. (2011) propõe conversores de alta eficiência e com uma grande largura banda. Um enfoque especial é dado nos trabalhos acadêmicos para os conversores *boost*, como ressaltado na revisão de Li e He (2011), em grande parte pela presença de um zero de fase não-mínima (NMP) em sua modelagem.

As pesquisas sobre o controle preditivo baseado em modelo para o *boost* fotovoltaico, como em Sguarezi Filho et al. (2021), estão associadas intimamente ao algoritmo MPPT, sempre buscando o aprimoramento do método de controle. Em trabalhos como os de Cheng et al. (2018) e Cunha et al. (2019), o avanço científico é buscado ao abordar uma das principais desvantagens do FCS-MPC, a frequência de chaveamento variável. Portanto, ambos os estudos propuseram uma solução para MPC de conjunto de controle finito que mantivesse a frequência constante, melhorando a previsibilidade do sistema e facilitando o desenvolvimento de filtros para a aplicação.

2.4.2 Níveis de Abstração da Simulação

O processo de simulação constitui uma etapa fundamental na prototipagem e validação lógica de algoritmos de controle digital, especialmente em estratégias computacionalmente intensivas como o FCS-MPC. No entanto, a validação funcional em alto nível é insuficiente, pois a transição da simulação para a prática impõe dois desafios principais: a adaptação do algoritmo para uma linguagem de mais baixo nível, adequada ao *hardware* de destino (como um Processador Digital de Sinais – DSP), e o atendimento aos requisitos temporais da aplicação, de modo que o código seja executado integralmente dentro do período de amostragem.

Embora implementações em scripts MATLAB (.m) ou blocos padrão do Simulink sejam adequadas para testes iniciais, elas não representam fielmente a execução em *hardware*. Nesse contexto, a utilização de *S-functions* surge como uma etapa intermediária importante, pois permite uma codificação mais próxima da linguagem C ou C++ empregada em DSPs e proporciona uma execução mais eficiente no próprio ambiente, por ser previamente compilada para código de máquina e não interpretada pelo MATLAB (MATHWORKS, 2002). Apesar de não reproduzirem exatamente o comportamento em tempo real, as *S-functions* oferecem uma estimativa qualitativa do desempenho computacional e contribuem para uma transição mais confiável entre o ambiente de simulação e a implementação prática.

2.5 Parâmetros para Análise de Desempenho

Com o intuito de encontrar o método de controle mais eficiente, é preciso estabelecer algumas métricas objetivas para que se compreenda o que de fato se qualifica como um controle ótimo.

Ogata (2011) caracteriza alguns índices de análise transitória como tempo de subida (t_r), tempo de pico (t_p), tempo de assentamento (t_s) e máximo sobressinal percentual ($M_{p\%}$), como destacado na Figura 12. Entre estes, se destacam os dois últimos como mais relevantes para fins de análise. O tempo de assentamento é determinado pelo tempo requerido para que o sinal chegue e se mantenha dentro de

uma faixa de porcentagem absoluta (em geral, 5% ou 2%) do valor final (OGATA, 2011). Já o máximo sobressinal percentual pode ser definido pela seguinte equação:

$$M_{P\%} = \frac{y(t_p) - y(\infty)}{y(\infty)}, \quad (2.2)$$

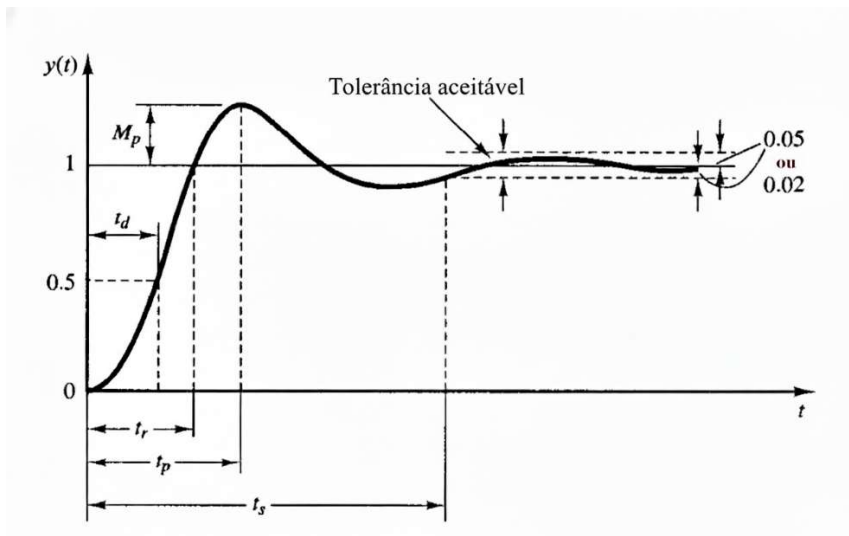
em que t_p representa o tempo de pico, por conseguinte $y(t_p)$ é igual ao valor de pico do sinal e $y(\infty)$ é o valor do sinal em regime permanente. Estas duas métricas, só são válidas para sistemas que convirjam para um valor finito.

Em sua monografia, Costa (2025) propôs uma maneira diferente de se avaliar o sobressinal, utilizando não do valor do sinal em regime permanente, mas sim da variação da referência, no intuito de levar em consideração a amplitude do degrau ao se comparar o *overshoot* do sinal. Nesse sentido, o máximo sobressinal relativo à amplitude do degrau pode ser definido como se segue:

$$M_{P\%rel.} = \frac{y(t_p) - y_{ref}}{\Delta y_{ref}}. \quad (2.3)$$

Para sistemas de controle de conversores chaveados a tensão não permanece constante em regime, há sempre uma oscilação em função do chaveamento. Assim sendo, a avaliação do desempenho dinâmico e de regime permanente dos controladores foi realizada de modo um pouco diferente daquilo usualmente apresentado em livros textos de controle como Ogata (2011) e o Dorf e Bishop (2017). Dois parâmetros foram analisados por método diferentes: a ondulação da tensão e o tempo de assentamento. Para a ondulação, foi definida uma janela de tempo em regime permanente para cada degrau, e o parâmetro foi quantificado como a diferença entre o valor máximo e o valor mínimo da tensão dentro desta janela (*range*). Para o tempo de assentamento, a metodologia consistiu em medir o intervalo de tempo desde a aplicação do degrau de referência até o instante em que a tensão de saída entra, e permanece, definitivamente dentro de uma faixa de tolerância. Esta faixa foi definida pelos próprios limites de ondulação (máximo e mínimo) do novo estado de regime permanente.

Figura 12 – Gráfico de resposta transitória - Métricas de desempenho.



Fonte: Adaptado de (OGATA, 2011).

Para além desses índices, o livro de Dorf e Bishop (2017) traz outras métricas quantitativas para se analisar a performance do sistema. Os quatro parâmetros IAE, ISE, ITAE e ITSE utilizam do erro do sistema para avaliar a eficiência do controlador.

O primeiro deles é caracterizado como a Integral do Erro Absoluto (IAE, do inglês *Integral of the Absolute Error*) e pode ser definida como:

$$\text{IAE} = \int_0^T |e(t)| dt, \quad (2.4)$$

em que $e(t)$ representa o sinal de erro, isto é, a diferença entre o sinal de saída e a referência e o T como limite superior da integral representa o tempo total da análise. O uso do módulo impede que erros positivos e negativos se cancelem, o que poderia mascarar o desempenho real do sistema. Deste modo, a integral passa a ser a soma infinitesimal da área entre a referência e o sinal de saída, resultando, de forma rudimentar, no erro acumulado ao longo do tempo.

Ademais, é possível estabelecer a métrica de ISE (Integral do Erro Quadrático), que de forma semelhante à IAE, mantém sempre valores positivos a serem integrados por elevar ao quadrado o valor do erro. A diferença entre os dois índices passa a ser como cada um penaliza as diferentes magnitudes do erro. O erro quadrático acaba penalizando mais duramente os erros maiores que um. Por outro lado, os erros menores são mais atenuados pela natureza quadrática da ISE, que pode ser calculado como:

$$\text{ISE} = \int_0^T e(t)^2 dt . \quad (2.5)$$

Ao se tentar otimizar o controle para cada um dos índices, essas divergências resultam em respostas dinâmicas distintas. Como os erros de maior magnitude são mais penalizados pela ISE, a tendência é que ele favoreça os controles que cheguem mais rápido ao valor de referência, levando a um tempo de subida menor, mesmo que isso signifique um maior sobressinal.

Os dois parâmetros restantes seguem o mesmo princípio dos dois anteriores, sendo um deles em função do módulo do erro e o outro em função do erro quadrático. O que os diferencia das métricas supracitadas é o fato de haver um termo de tempo multiplicando os termos do erro. Portanto temos a ITAE, Integral do erro absoluto multiplicado pelo tempo, que se calcula da seguinte maneira:

$$\text{ITAE} = \int_0^T t|e(t)| dt , \quad (2.6)$$

e a Integral do erro quadrático multiplicado pelo tempo, ITSE, dada por:

$$\text{ITSE} = \int_0^T te(t)^2 dt . \quad (2.7)$$

A adição do termo de tempo faz com que os erros do início da análise, isto é, próximo do momento em que se mudou de referência, sejam atenuados. Enquanto isso, os erros mais tardios, chegando ao tempo de regime permanente, são mais custosos para essas duas métricas. Isto posto, esses dois parâmetros tendem a levar a soluções menos oscilatórias em regime, mantendo, ainda assim, as particularidades dos dois índices anteriores devido à natureza quadrática e de valor absoluto de cada um.

Em resumo, esses foram os critérios de avaliação adotados para os controles deste estudo. Em conjuntura, eles serviram o propósito de métrica quantitativa de comparação para a definição dos controladores ótimos ao longo deste trabalho.

3 MODELAGEM E CONTROLE PREDITIVO BASEADO EM MODELO

Este capítulo apresenta o embasamento matemático do trabalho, desde o modelo do conversor até a escolha dos algoritmos de controle. Isto abrange, inicialmente, o detalhamento da modelagem do conversor *boost* fotovoltaico e, em seguida, os modelo em ambiente de simulação. Além disso, é exposta a elaboração do processo que compreende a obtenção do modelo preditivo no domínio do tempo discreto. Posteriormente, o foco é direcionado para a análise de desempenho dos sistemas de controle, a princípio explorando uma alternativa de controle linear para fins de comparação e, na sequência, apresentando o foco desta pesquisa o FCS-MPC.

3.1 Modelagem do Conversor *Boost* Fotovoltaico

Para realizar a implementação do algoritmo de MPC, como posto amplamente na literatura (KOURO et al., 2009; RODRIGUEZ et al., 2012; BORDONS e MONTERO, 2015), é necessário um modelo discretizado que corresponda, de forma precisa, ao conversor. Porém, devido à natureza chaveada do conversor, é encontrado um empecilho na implementação da representação dinâmica da planta, faz-se necessário considerar dois equacionamentos distintos: um para o conversor com a chave semicondutora ativa e um outro para o circuito enquanto a chave está desativada. Desta forma, deixa de ser possível sintetizar o modelo a uma única equação dinâmica e recorre-se a modelagem no espaço de estados (ERICKSON e MAKSIMOVIC, 2004).

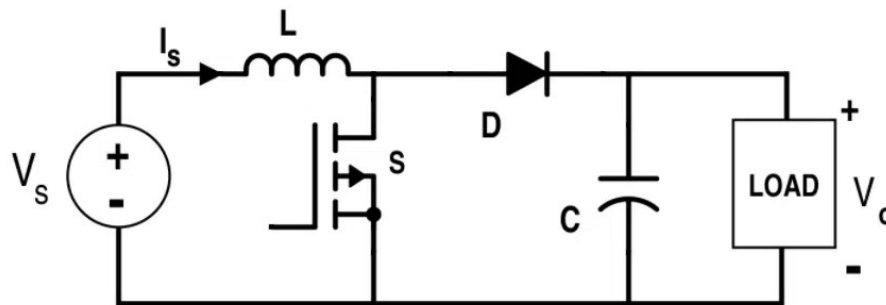
Isto posto, expõe-se a seguir o *boost* convencional, como mostrado no capítulo anterior, e em sequência o conversor aplicado a um sistema FV. Realiza-se a sua modelagem no espaço de estados, inicialmente em tempo contínuo, e a sua discretização para aplicações do controle FCS-MPC.

3.1.1 Conversor *boost* convencional

Como exposto anteriormente, o conversor *boost*, também conhecido como conversor *Step-Up*, é um conversor do tipo CC-CC que eleva a tensão do sistema. Como evidenciado na Figura 4(b) e aqui repetido para facilitar a visualização na Figura

13, o *boost* opera com dois dispositivos semicondutores, um deles é o diodo que trabalha passivamente, impedindo a condução de corrente em seu modo reverso. Já o outro é uma chave controlada, um MOSFET por exemplo, como é no caso da Figura 13 .

Figura 13 – Circuito do conversor *boost* convencional.



Fonte: Adaptado de (RIBEIRO JR., 2023)

O princípio de funcionamento deste circuito, como evidenciado no livro de Mohan, Undeland e Robbins (2003) é fundamentado na transferência de energia que é acumulada no indutor, aliada a comutação da chave. Quando a chave está ativa, permitindo a condução de corrente por ela, o indutor é capaz de acumular energia através do seu campo magnético induzido. Ao passo que a chave é desativada, a energia reservada no indutor pode ser destinada ao capacitor à medida que o seu campo magnético colapsa, a corrente I_s diminui e a polaridade da tensão do indutor é invertida. Dessa forma, o indutor atua como uma fonte para o capacitor e a carga, portanto, soma-se a sua tensão a tensão da fonte, efetivamente elevando a tensão de saída. Então pode-se determinar equações para a tensão do indutor, V_L , para a etapa condução da chave:

$$v_L(t) = v_s(t) , \quad (3.1)$$

e para a etapa de não condução:

$$v_L(t) = v_s(t) - v_o(t) . \quad (3.2)$$

A comutação da chave semicondutora, quando operada a uma frequência constante, segue uma lógica de razão cíclica (D), definido pela divisão do o tempo em que a chave está conduzindo (t_{on}) pelo período de chaveamento (T_s), como:

$$D = \frac{t_{on}}{T_s}. \quad (3.3)$$

Dado que o período de chaveamento é composto integralmente pela soma das parcelas de t_{on} e o tempo em que a chave não conduz (t_{off}), a razão D pode ser representada em termos de t_{off} pela expressão a seguir:

$$(1 - D) = \frac{t_{off}}{T_s}. \quad (3.4)$$

Seguindo o princípio da conservação de energia, sabe-se que a tensão média sobre o indutor é nula em regime permanente.

$$\frac{1}{T_s} \int_0^{T_s} v_L(t) dt = 0.$$

Então, é possível determinar uma expressão para o ganho estático do conversor *boost*, a partir de uma média ponderada pela razão cíclica da tensão no indutor, em modo de condução contínua (CCM):

$$\frac{1}{T_s} \int_0^{T_s} v_L(t) dt = V_s D - (V_s - V_o)(1 - D) = 0. \quad (3.5)$$

Simplificando, acha-se a seguinte razão e expressão para elevação da tensão de saída média:

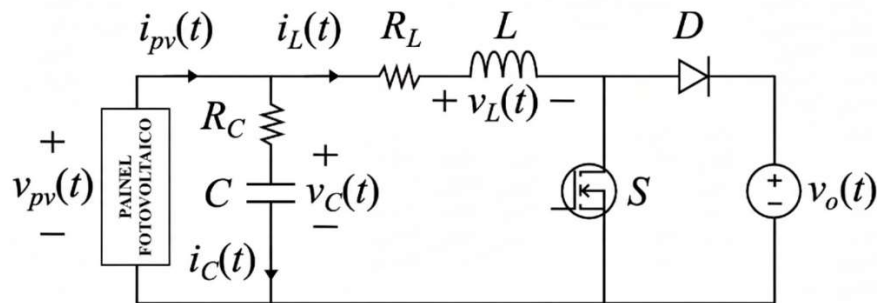
$$\frac{V_o}{V_s} = \frac{1}{1 - D} \quad ; \quad V_o = \frac{V_s}{1 - D}. \quad (3.6)$$

O modelo apresentado na Figura 13 não expõe algumas características geradoras de perdas ao conversor como aquelas causadas pela comutação da chave e as resistências parasitas associadas aos componentes, como o indutor e o capacitor.

3.1.2 Adaptação para o boost fotovoltaico

Como o intuito deste trabalho foi de simular de forma rudimentar um sistema FV, se fez necessário fazer algumas adaptações ao circuito, como mostrado na Figura 14. Seguiu-se a ideia de Cunha (2021), mostrada anteriormente na Figura 7, mas de modo simplificado. No lugar de uma outra malha de controle para determinar a corrente I_{pv} vinda do painel fotovoltaico, estipulou-se que esta corrente seria constante e aplicada por uma fonte de corrente. Assim como na Figura 7, a carga passou a ser representada por uma fonte CC de tensão V_o , e notadamente o capacitor muda de posição neste circuito, visto que nessa topologia a tensão de entrada passa a ser variável. Assim sendo, é a tensão de entrada do conversor (ou de saída do painel, V_{pv}) que se pretende filtrar e reduzir a oscilações (*ripple*). Para finalizar, foram adicionadas resistências em série com o capacitor e o indutor a fim de melhor representar as perdas ôhmicas de ambos os componentes.

Figura 14 – Circuito do conversor *boost* fotovoltaico.



Fonte: Adaptado de (SANTOS, COSTA, *et al.*, 2020)

Inicialmente, verificou-se a possibilidade de se utilizar o kit didático desenvolvido na monografia de Melo (2023) para determinar os valores dos parâmetros utilizados no circuito em ambiente de simulação. No entanto, encontrou-se um empecilho claro, o conversor *boost* desenvolvido trabalhava em modo descontínuo, o que dificultaria consideravelmente a determinação de seu modelo (KARAMANAKOS, GEYER e MANIAS, 2014). Portanto, optou-se por alterar os parâmetros de forma estudada para que fosse garantida a condução contínua durante toda a operação do sistema.

Chegou-se, portanto, aos valores presentes na Tabela 4. Apesar dessas mudanças, a relação do ganho estático do conversor continua seguindo a mesma equação:

$$\frac{V_O}{V_{pv}} = \frac{1}{1 - D} \quad ; \quad V_{pv} = (1 - D)V_O. \quad (3.7)$$

Tabela 4 – Parâmetros escolhidos para o conversor boost.

| Símbolo | Descrição | Valor |
|----------|--|---------------|
| V_O | Tensão de saída do conversor | 20 V |
| I_{pv} | Corrente do painel fotovoltaico | 8 A |
| C | Capacitância do conversor | 33 μ H |
| L | Indutância do conversor | 100 μ H |
| R_C | Resistência série capacitor | 0,05 Ω |
| R_L | Resistência série indutor | 0,1 Ω |
| f_s | Frequência de Chaveamento | 80kHz |
| D | Ponto de Operação do Ciclo de Trabalho | 0,5 |

Fonte: Elaborado pelo Autor (2025).

Reitera-se que os níveis de tensão e corrente escolhidos levam em consideração as limitações da planta física e não representam, de modo geral, as magnitudes utilizadas em sistemas fotovoltaicos.

3.1.3 Modelo em espaço de estados

Como comentado anteriormente para o *boost* convencional, a modelagem do sistema deve ser feita no espaço de estados, em função da sua não linearidade e variância no tempo. Portanto, analisa-se o circuito em suas fases ativa, como mostrado na Figura 15 e desativa, ilustrada na Figura 16. As expressões para o estado ativo trazem consigo o subscrito 1 como a seguir para a equação de estado, em que A representa a matriz de estado e B é a matriz de entrada:

$$\dot{x}(t) = A_1 x(t) + B_1 u(t) \quad (3.8)$$

e para a equação de saída, em que C é matriz de saída e E é a matriz de transmissão direta:

$$y(t) = C_1 x(t) + E_1 u(t) \quad (3.9)$$

De modo análogo, para a etapa desativa se adiciona um subíndice 2 aos termos:

$$\dot{x}(t) = A_2 x(t) + B_2 u(t) \quad (3.10)$$

$$y(t) = C_2 x(t) + E_2 u(t) \quad (3.11)$$

Antes mesmo de montar o sistema de equações, é preciso determinar quais são as variáveis de estado, bem como as entradas e saída do sistema. Como exposto por Ogata (2011), o número de variáveis de estado está intimamente ligado ao número de elementos acumuladores de energia do circuito. Neste caso, há dois deles, o capacitor e o indutor. Portanto, determina-se que a tensão sobre o capacitor e a corrente pelo indutor sejam as variáveis de estado:

$$x(t) = \begin{bmatrix} v_c(t) \\ i_L(t) \end{bmatrix} \quad (3.12)$$

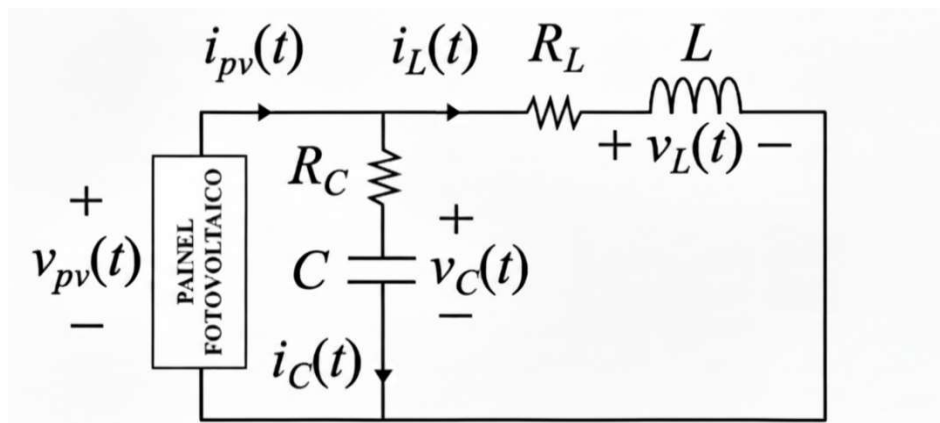
A tensão de saída do conversor e a corrente do painel fotovoltaico são as entradas.

$$u(t) = \begin{bmatrix} v_0(t) \\ i_{pv}(t) \end{bmatrix} \quad (3.13)$$

E a saída, $y(t)$, é a tensão de entrada do conversor, como se segue:

$$y(t) = [v_{pv}(t)] \quad (3.14)$$

Figura 15 – Circuito equivalente do *boost* fotovoltaico com a chave ativa.



Fonte: Elaborado pelo Autor (2025).

A análise do circuito da Figura 15 é realizada a partir das Leis de Kirchhoff de corrente e de tensão. Em termos de corrente, o circuito ativo é definido da seguinte maneira:

$$i_C(t) = -i_L(t) + i_{pv}(t) \quad (3.15)$$

Considerando a expressão da tensão sobre o capacitor, acha-se a primeira equação de estado:

$$i_C(t) = C \frac{dv_C}{dt} \therefore \dot{v}_C(t) = -\frac{1}{C}i_L(t) + \frac{1}{C}i_{pv}(t) \quad (3.16)$$

Fazendo o mesmo procedimento para as tensões, encontra-se a equação de estados da corrente no indutor como se segue:

$$v_L(t) + R_L i_L(t) = v_C(t) + (i_{pv}(t) - i_L(t))R_C \quad (3.17)$$

Portanto, a equação para a derivada da corrente pelo tempo é dada por:

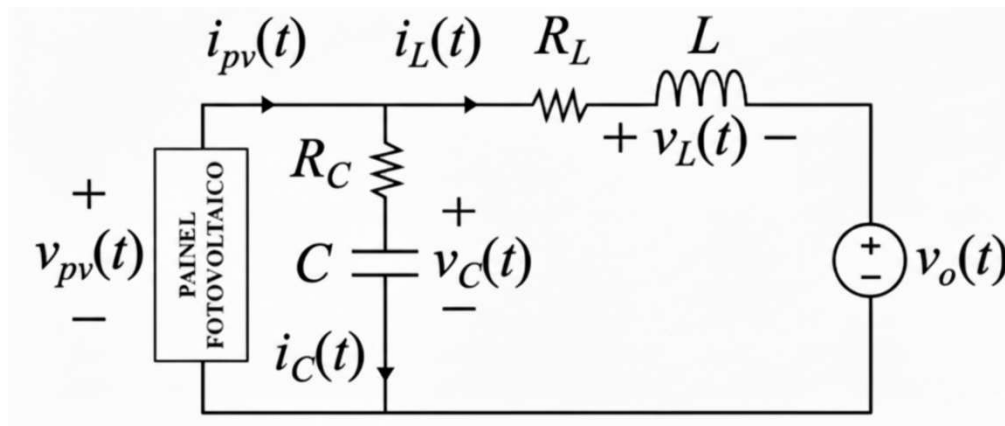
$$v_L(t) = L \frac{di_L}{dt} \therefore \dot{i}_L(t) = \frac{v_C(t)}{L} - \frac{R_L + R_C}{L}i_L(t) + \frac{R_C}{L}i_{pv}(t) \quad (3.18)$$

Resta agora determinar a expressão para a variável de saída,

$$v_{pv}(t) = v_C(t) - R_C \cdot i_L(t) + R_C \cdot i_{pv}(t) \quad (3.19)$$

De modo análogo, realiza-se esse procedimento para a etapa desativa como apresentada na Figura 16.

Figura 16 – Circuito equivalente do *boost* fotovoltaico com a chave desativada.



Fonte: Elaborado pelo Autor (2025)

Em termos de corrente, o circuito com a chave aberta é definido deste modo,

$$i_c(t) = -i_L(t) + i_{pv}(t) \quad (3.20)$$

Mais uma vez utilizando a expressão da tensão sobre o capacitor, acha-se a equação de estado, que é idêntica à equação para a etapa ativa,

$$\dot{v}_c(t) = -\frac{1}{C}i_L(t) + \frac{1}{C}i_{pv}(t) \quad (3.21)$$

Já para o caso das tensões, há uma diferença nas expressões, encontra-se, portanto, a equação de estados da corrente no indutor como se segue:

$$v_L(t) + R_L i_L(t) + v_o(t) = v_c(t) + (i_{pv}(t) - i_L(t))R_C \quad (3.22)$$

Portanto, a equação para a derivada da corrente no tempo é dada pela seguinte expressão, sendo adicionado apenas o termo da tensão de saída em relação ao circuito ativo.

$$\dot{i}_L(t) = v_c(t) - \frac{R_L + R_C}{L}i_L(t) - \frac{1}{L}v_o(t) + \frac{R_C}{L} \cdot i_{pv}(t) \quad (3.23)$$

Por fim, determina-se a expressão para a tensão de entrada do conversor, a saída do sistema

$$v_{pv}(t) = v_c(t) - R_C \cdot i_L(t) + R_C \cdot i_{pv}(t) \quad (3.24)$$

A partir das equações apresentadas acima se determinam as matrizes de estado: A_1 e A_2 ,

$$A_1 = A_2 = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{(R_L + R_C)}{L} \end{bmatrix} \quad (3.25)$$

Bem como as matrizes de entrada B_1 e B_2

$$B_1 = \begin{bmatrix} 0 & \frac{1}{C} \\ 0 & \frac{R_C}{L} \end{bmatrix}; B_2 = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & \frac{R_C}{L} \end{bmatrix} \quad (3.26)$$

As matrizes de saída: C_1 e C_2 ,

$$C_1 = C_2 = [1 \quad -R_C] \quad (3.27)$$

E por fim as matrizes de transmissão direta : E_1 e E_2 ,

$$E_1 = E_2 = [0 \quad R_C] \quad (3.28)$$

Em formato matricial, portanto as expressões ficam da seguinte forma para o subintervalo ativo,

$$\begin{cases} \begin{bmatrix} \dot{v}_c(t) \\ \dot{i}_L(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{(R_L + R_C)}{L} \end{bmatrix} \begin{bmatrix} v_c(t) \\ i_L(t) \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{C} \\ 0 & \frac{R_C}{L} \end{bmatrix} \begin{bmatrix} v_0(t) \\ i_{pv}(t) \end{bmatrix} \\ v_{pv}(t) = [1 \quad -R_C] \begin{bmatrix} v_c(t) \\ i_L(t) \end{bmatrix} + [0 \quad R_C] \begin{bmatrix} v_0(t) \\ i_{pv}(t) \end{bmatrix} \end{cases} \quad (3.29)$$

Enquanto para o intervalo t_{off} , muda-se somente o termo de $v_0(t)$ na equação de estado:

$$\begin{cases} \begin{bmatrix} \dot{v}_c(t) \\ \dot{i}_L(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{(R_L + R_C)}{L} \end{bmatrix} \begin{bmatrix} v_c(t) \\ i_L(t) \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & \frac{R_C}{L} \end{bmatrix} \begin{bmatrix} v_0(t) \\ i_{pv}(t) \end{bmatrix} \\ v_{pv}(t) = [1 \quad -R_C] \begin{bmatrix} v_c(t) \\ i_L(t) \end{bmatrix} + [0 \quad R_C] \begin{bmatrix} v_0(t) \\ i_{pv}(t) \end{bmatrix} \end{cases} \quad (3.30)$$

Dado este sistema no espaço de estados, é preciso discretizar o modelo, para que o algoritmo do FCS-MPC possa ser empregado. A seguir se expõe a discretização pelo método de *Euler Forward*.

3.1.4 Discretização do Modelo

Como mostrado por Ogata (1995), um modelo da planta no domínio discreto (k), em que k representa o índice da amostra na sequência discreta, é imprescindível para a aplicação dos controles digitais. Isto é ainda mais verdadeiro para o controle FCS-MPC, que deve antecipar os valores futuros das variáveis de estado, o que permite a tomada de medidas de controle de acordo com os resultados antecipados. Amplamente adotado, o método de discretização *Euler Forward*, conhecido também

como método das diferenças para frente, é ideal para a necessidade gerada pelo comportamento preditor do modelo. Na incapacidade de se ter um tempo de amostra infinitesimal, condição necessária para se determinar a derivada de um sinal pela definição de limite, esse método faz o mais próximo possível para uma taxa de amostragem T_s , aproximando linearmente a derivada do vetor de estado $x(t)$. Desta forma, utiliza-se o valor da amostra atual e da amostra seguinte para então se aproximar a derivada de $x(t)$ (OGATA, 1995). O emprego deste método é observado a seguir:

$$\dot{x}(t) \cong \frac{x(k+1) - x(k)}{T_s} \quad (3.31)$$

Portanto, substituindo na expressão em espaço de estados obtém-se,

$$\begin{cases} \frac{x(k+1) - x(k)}{T_s} = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Eu(k) \end{cases} \quad (3.32)$$

Isolando o termo $x(k+1)$, chega-se a seguinte relação:

$$\begin{cases} x(k+1) = (I + T_s A)x(k) + T_s Bu(k) \\ y(k) = Cx(k) + Eu(k) \end{cases}, \quad (3.33)$$

Em formato matricial isto se torna a seguinte expressão para o estado ativo

$$\begin{cases} \begin{bmatrix} v_c(k+1) \\ i_L(k+1) \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + T_s \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{(R_L + R_C)}{L} \end{bmatrix} \right) \begin{bmatrix} v_c(k) \\ i_L(k) \end{bmatrix} + T_s \begin{bmatrix} 0 & \frac{1}{C} \\ 0 & \frac{R_C}{L} \end{bmatrix} \begin{bmatrix} v_0(k) \\ i_{pv}(k) \end{bmatrix} \\ v_{pv}(k) = \begin{bmatrix} 1 & -R_C \end{bmatrix} \begin{bmatrix} v_c(k) \\ i_L(k) \end{bmatrix} + \begin{bmatrix} 0 & R_C \end{bmatrix} \begin{bmatrix} v_0(k) \\ i_{pv}(k) \end{bmatrix} \end{cases} \quad (3.34)$$

Enquanto para a etapa desativa,

$$\begin{cases} \begin{bmatrix} v_c(k+1) \\ i_L(k+1) \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + T_s \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{(R_L + R_C)}{L} \end{bmatrix} \right) \begin{bmatrix} v_c(k) \\ i_L(k) \end{bmatrix} + T_s \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & \frac{R_C}{L} \end{bmatrix} \begin{bmatrix} v_0(k) \\ i_{pv}(k) \end{bmatrix} \\ v_{pv}(k) = \begin{bmatrix} 1 & -R_C \end{bmatrix} \begin{bmatrix} v_c(k) \\ i_L(k) \end{bmatrix} + \begin{bmatrix} 0 & R_C \end{bmatrix} \begin{bmatrix} v_0(k) \\ i_{pv}(k) \end{bmatrix} \end{cases} \quad (3.35)$$

Simplificando as expressões chega-se, finalmente, ao resultado das variáveis de interesse no tempo discreto. A tensão no capacitor predita para uma amostra de tempo k a frente é dada pela seguinte equação.

$$v_C(k+1) = v_C(k) - \frac{T_s}{C} i_L(k) + \frac{T_s}{C} \cdot i_{pv}(k) \quad (3.36)$$

Nota-se que a expressão para um tempo de amostragem resulta na mesma tensão v_C tanto para chave ativa quanto desativa. Por outro lado, $i_L(k+1)$ muda conforme a chave muda de estado. Conforme representado na seguinte função:

$$i_L(k+1) = \frac{T_s}{L} v_C(k) + \left(1 - \frac{T_s}{L}(R_L + R_C)\right) i_L(k) - \frac{T_s}{L}(1 - g(k))v_o(k) + \frac{T_s}{L} R_C i_{pv}(k), \quad (3.37)$$

que expressa a ação da chave como $g(k)$ da seguinte forma:

$$\begin{cases} g(k) = 1, & \text{para a chave ativada,} \\ g(k) = 0, & \text{para a chave desativada} \end{cases} \quad (3.38)$$

Por fim, encontra-se a expressão para a tensão predita para a entrada do conversor

$$v_{pv}(k+1) = v_C(k+1) - R_C i_L(k+1) + R_C i_{pv}(k+1) \quad (3.39)$$

As expressões no tempo discreto bastariam para se realizar controle por meio do algoritmo do FCS-MPC. No entanto, para validar o modelo apresentado, e posteriormente, poder realizar um controle linear como base de comparação ao controle preditivo, optou-se por encontrar o modelo médio e linearizado.

3.1.5 Linearização e Modelo Médio por Pequenos Sinais

A abordagem usada para representar a dinâmica média dos conversores chaveados é a de ponderação linear média dos modelos, considerando as equações de cada estado de condução pelo tempo que estes ocupam no ciclo de chaveamento (ERICKSON e MAKSIMOVIC, 2004). Essa técnica requer a linearização do sistema em torno de um ponto de operação fixo, assumindo pequenas perturbações dos sinais. Da seguinte forma:

$$x(t) = X + \hat{x}(t), \quad \text{com } |\hat{x}(t)| \ll |X|, \quad (3.40)$$

$$y(t) = Y + \hat{y}(t), \quad \text{com } |\hat{y}(t)| \ll |Y|, \quad (3.41)$$

$$u(t) = U + \hat{u}(t), \quad \text{com } |\hat{u}(t)| \ll |U|, \quad (3.42)$$

$$d(t) = D + \hat{d}(t), \quad \text{com } |\hat{d}(t)| \ll |D|, \quad (3.43)$$

Em paralelo, efetua-se a ponderação média dos sinais no tempo em função do ciclo de trabalho $d(t)$ e o seu complementar $d'(t)$. Deste modo, as expressões para o circuito ativo e desativo, encontradas para o sistema no espaço de estados, são combinadas em um único modelo médio.

$$\dot{x}(t) = \frac{d(t)}{d(t) + d'(t)} \cdot (A_1 x(t) + B_1 u(t)) + \frac{d'(t)}{d(t) + d'(t)} \cdot (A_2 x(t) + B_2 u(t)), \quad (3.44)$$

$$y(t) = \frac{d(t)}{d(t) + d'(t)} \cdot (C_1 x(t) + E_1 u(t)) + \frac{d'(t)}{d(t) + d'(t)} \cdot (C_2 x(t) + E_2 u(t)) \quad (3.45)$$

Chega-se a uma expressão linearizada após a simplificação rigorosa demonstrada em Erikson e Maksimovic (2004), na qual as multiplicações dos pequenos sinais se igualou a zero por serem ordens de grandeza menores que o resto da expressão.

$$\begin{cases} \hat{x}(t) = A_m \hat{x}(t) + [B_m (A_1 - A_2)X + (B_1 - B_2)U] \begin{bmatrix} \hat{u}(t) \\ \hat{d}(t) \end{bmatrix} \\ \hat{y}(t) = C_m \hat{x}(t) + [E_m (C_1 - C_2)X + (E_1 - E_2)U] \begin{bmatrix} \hat{u}(t) \\ \hat{d}(t) \end{bmatrix} \end{cases} \quad (3.46)$$

Em que a matrizes do espaço do estado são representadas como uma média ponderada em função do ciclo de trabalho D , do seguinte modo:

$$A_m = DA_1 + (1 - D)A_2, \quad (3.47)$$

$$B_m = DB_1 + (1 - D)B_2, \quad (3.48)$$

$$C_m = DC_1 + (1 - D)C_2, \quad (3.49)$$

$$E_m = DE_1 + (1 - D)E_2, \quad (3.50)$$

E as equações para o regime permanente são estas:

$$\begin{cases} 0 = A_m X + B_m U, \\ Y = C_m X + E_m U. \end{cases} \quad (3.51)$$

Tendo como base o sistema apresentado na equação (3.45) é possível determinar a matriz de transferência do sistema linearizado utilizando a expressão a seguir:

$$H(s) = C(sI - A)^{-1}B + E, \quad (3.52)$$

na qual, as matrizes A, B, C e E são representadas por:

$$A = A_m \quad (3.53)$$

$$B = B_m (A_1 - A_2)X + (B_1 - B_2)U, \quad (3.54)$$

$$C = C_m, \quad (3.55)$$

$$E = E_m (C_1 - C_2)X + (E_1 - E_2)U. \quad (3.56)$$

Efetuada as contas e substituindo os valores dos parâmetros do conversor como mostrados na Tabela 4, chega-se a seguinte matriz de transferência para o sistema MISO (*Multiple Inputs Single output*), um sistema com múltiplas entradas, mas apenas uma saída:

$$\hat{v}_{pv}(s) = \begin{bmatrix} \frac{250(s+6,06 \times 10^5)}{s^2+1500s+3,03 \times 10^8} & \frac{0,05(s+6,06 \times 10^5)(s+1000)}{s^2+1500s+3,03 \times 10^8} & \frac{-1000(s+6,06 \times 10^5)}{s^2+1500s+3,03 \times 10^8} \end{bmatrix} \begin{bmatrix} \hat{v}_o(s) \\ \hat{i}_{pv}(s) \\ \hat{d}(s) \end{bmatrix}. \quad (3.57)$$

3.1.6 Método de Validação do Modelo

Foi implementada uma simulação em ambiente MATLAB/Simulink para validar a acurácia do modelo linearizado no domínio do tempo. A sua resposta dinâmica é comparada com a de um modelo de simulação do conversor. Esta análise comparativa foca nos transitórios da tensão de entrada do conversor v_{pv} . Para verificar a validade do modelo, então, aplicou-se uma perturbação às três entradas distintas do sistema MISO originado pelo modelo linearizado separadamente e, por fim, aplicou-se as mesmas perturbações, mas em todas as variáveis ao mesmo tempo. Os resultados estão apresentados no Capítulo 4.

3.1.7 Proposta de Controle Linear

Após a validação do modelo, é chegada a hora de se realizar o controle do conversor *boost* estudado. Inicialmente, a proposta foi de aplicar um controle linear ao sistema para se obter uma base de comparação para os controladores FCS-MPC que que viriam a ser implementados durante o trabalho.

Portanto, foi seguida a metodologia exposta no capítulo 9 de Erikson e Maksimovic (2004). A análise do conversor parte da modelagem por pequenos sinais, como evidenciada na matriz de transferências da equação (3.57), mais especificamente, volta-se o interesse à função de transferência da variação da tensão de entrada (\hat{v}_{pv}) pela variação no ciclo de trabalho (\hat{d}), como abaixo,

$$\frac{\hat{v}_{pv}(s)}{\hat{d}(s)} = \frac{-1000(s + 6,06 \times 10^5)}{s^2 + 1500s + 3,03 \times 10^8} . \quad (3.58)$$

Nota-se de antemão que a relação é negativa, em função da natureza inversamente proporcional da razão cíclica e a tensão de entrada, isto é, quanto maior a razão de trabalho, menor será a tensão na saída do painel fotovoltaico nesse sistema. Tendo isso em mente, a primeira medida a ser tomada foi considerar um controlador com ganho negativo.

Isto posto, parte-se para a análise no domínio da frequência, a partir daqui segue-se o procedimento estabelecido por Erikson e Maksimovic (2004) e replicado em grande parte pelo estudo de Giraldo (2021), levou-se então em consideração o atraso da implementação do controle digital de um período de chaveamento (T_s) e meio período pela natureza do modulador PWM. Logo, o atraso total do sistema é dado por:

$$T_d = T_s + \frac{T_s}{2} . \quad (3.59)$$

Então, optou-se por estabelecer a frequência de passagem almejada (ω_c) como um vigésimo da frequência de chaveamento, f_s , seguindo o exemplo de Erikson e Maksimovic (2004), neste caso, $f_c = 4kHz$ e, portanto,

$$\omega_c = 2\pi \cdot 4kHz , \quad em \ rad/s . \quad (3.60)$$

Definiu-se ainda que o controlador seria um compensador com dois polos e dois zeros, assim como aqueles escolhidos como base de comparação em Costa (2025) e Cunha (2021), seguindo o seguinte formato:

$$G_C(s) = K \frac{(s + z_1)(s + z_2)}{s(s + p)}. \quad (3.61)$$

Determinou-se, na sequência, a frequência do polo e dos zeros. Optou-se por igualar z_1 e z_2 , criando zeros duplos a uma frequência quatro vezes mais baixa que f_C , para garantir uma margem de fase considerável em f_C .

$$z_{1,2} = \frac{\omega_C}{4}, \quad \text{em rad/s}. \quad (3.62)$$

Enquanto isso, o polo foi definido uma frequência uma década abaixo de f_S , de modo que não houvesse tanta influência na desta forma:

$$p = \frac{2\pi f_S}{10}, \quad \text{em rad/s}. \quad (3.63)$$

Em seguida, achou-se o ganho na frequência de passagem com K unitário, para que fosse possível calcular o ajuste de K necessário para levar a frequência de 4kHz à magnitude de 0 dB, garantindo assim que a frequência ω_C escolhida fosse de fato a frequência de passagem. Assim, encontra-se o controlador:

$$G_C(s) = -\frac{0,1148 s^2 + 1442 s + 4,53 \cdot 10^6}{s^2 + 50270 s}. \quad (3.64)$$

Discretizando o controlador pela transformada bilinear (método de Tustin), chega-se ao controlador digital final:

$$G_C(z) = -\frac{0,09432 z^2 - 0,1744 z + 0,0806}{z^2 - 1,522 z + 0,5219}. \quad (3.65)$$

3.2 Propostas de controle FCS-MPC

As propostas deste trabalho para o controle preditivo baseado em modelo com conjunto de controle finito (FCS-MPC) dão sequência ao estudo desenvolvido em Costa (2025) para o conversor *buck*. Três controladores FCS-MPC são postos a

comparação: com erro quadrático, com restrição condicional temporal e, por fim, com restrição de tensão com um horizonte de predição estendido.

Como exposto por Rodriguez e Cortes (2012) e reforçado por Costa (2025), a função de custo com erro quadrático pode ser bem utilizada com o intuito de manter o valor da variável de interesse próximo do valor desejado de operação. Como apresentado em Rodrigues, (2020) a implementação do controle digital causa um atraso inerentemente no sistema de predição. Portanto, exige-se que se faça a predição de no mínimo duas amostras à frente. Em vista disso, são apresentadas a seguir as propostas de controle em sequência.

3.2.1 Controle por Erro Quadrático Sem Restrições

Inicialmente, a proposta foi de se desenvolver um algoritmo MPC que servisse como base para a aplicação de restrições daí por diante. Optou-se por utilizar a expressão do erro quadrático como função custo como escrita a seguir, em que J é o valor da função custo e V_{pv}^* é a referência.

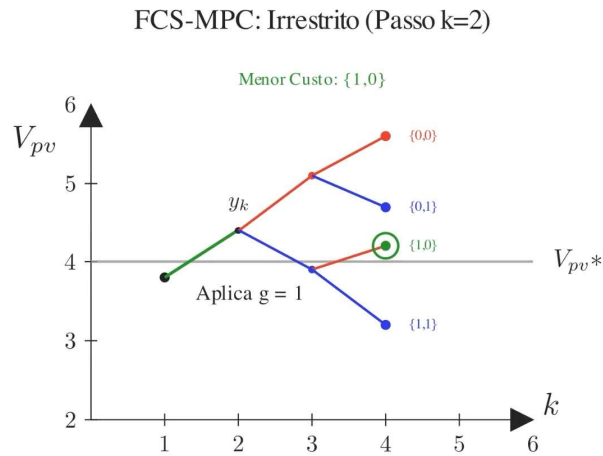
$$J = (V_{pv}^* - v_{pv}(k + 2))^2. \quad (3.66)$$

O algoritmo, como apresentado no fluxograma do Apêndice A, computa a cada iteração duas amostras futuras, o que resulta em 4 possibilidades e por conseguinte quatro custos. Ao determinar qual custo é o mínimo, o controlador aplica a ação de controle g à chave semicondutora com base no primeiro estado da chave do menor custo. Por exemplo, se J for minimizado pela sequência de chaveamento ativo seguida por desativo ($g(k) = 1$ e $g(k + 1) = 0$), a ação de controle $g(k)$ será para ativar a chave ($g(k) = 1$)

Dada a natureza dinâmica do Controle Preditivo baseado em Modelo (MPC), a visualização estática das equações pode não ser suficiente para demonstrar intuitivamente a tomada de decisão do algoritmo em tempo real. Para mitigar essa lacuna e ilustrar o funcionamento do algoritmo proposto, foi desenvolvida uma

simulação visual utilizando a linguagem Python.¹ A Figura 17 apresenta um quadro representativo desta simulação, destacando a escolha pelo menor custo predito.

Figura 17 – Quadro da tomada de decisão do FCS-MPC irrestrito.



3.2.2 Controle com Restrição Condicional

Com o objetivo de atenuar o sobressinal excessivo que pode ser gerado pela função custo sem restrição, foi proposta uma adaptação ao algoritmo MPC. Esta abordagem introduz uma restrição condicional à função custo, baseada em um horizonte de predição estendido por N instantes discretos de tempo ($k + N$) e projetada para atuar seletivamente apenas nos regimes transitórios do conversor, ou seja, nos instantes subsequentes a uma alteração da tensão de referência (v_{pv}^*).

A estratégia de controle opera monitorando as variações da referência (v_{pv}^*). Ao detectar uma mudança nesse *setpoint*, um temporizador Δt^* é iniciado e a restrição permanece ativa enquanto Δt^* for menor ou igual a um tempo limite predefinido t' . Durante esse intervalo, além da minimização da função custo irrestrita (calculada para o horizonte $k + N$), o algoritmo executa uma verificação de trajetória futura para N passos adiante.

Diferentemente de um MPC com horizonte longo, que avaliaria 2^N possibilidades de sequência de chaveamento, esta rotina simplificada projeta apenas duas trajetórias

¹ Para visualizar a dinâmica temporal deste algoritmo, uma animação completa em vídeo foi elaborada e encontra-se disponível através do link: https://youtu.be/8B_VpyNdWYQ.

extremas: a manutenção contínua da chave no estado ativado ($g(k) = 1$) por N amostras de tempo ou a permanência da chave no estado desativado ($g(k) = 0$) também por N instantes preditos. O objetivo é antecipar-se à possibilidade futura de um cruzamento excessivo do novo valor de referência, evitando um sobressinal expressivo.

A lógica da restrição a função custo é aplicada de modo diferente a depender do sentido da variação. Isto é, há dois modos de funcionamento: um para uma variação positiva da referência e outro para variação negativa.

1. Variação positiva da referência: O algoritmo verifica a trajetória futura assumindo que a chave permaneça desativada (ação tipicamente eleva a tensão). Se a predição indicar que a tensão superará o valor da referência dentro de N amostras de tempo (caracterizando um *overshoot*), as ações de controle que iniciam com a chave desativada ($g(k) = 0$) recebem uma penalidade infinita na função custo, forçando o controlador a escolher a chave ativada como próxima ação de controle (portanto, $g(k) = 1$). A formulação matemática para a função custo para este caso é dada por:

$$J = \begin{cases} \infty, & \text{se } v_{pv}(k+N)|_{g=0} > v_{pv}^*, \quad \text{para } \Delta t^* \leq t' \text{ e } g(k) = 0, \\ (v_{pv}^* - v_{pv}(k+2))^2, & \text{para os demais casos.} \end{cases} \quad (3.67)$$

2. Variação negativa da referência: De modo análogo, verifica-se a trajetória com a chave mantida sempre ativada, se a predição indicar que a tensão cairá abaixo da tensão de referência nova (*undershoot*), as sequências de chaveamento que iniciam com a chave ativada ($g(k) = 1$) são bloqueadas e o algoritmo atua desativando a chave ($g(k) = 0$).

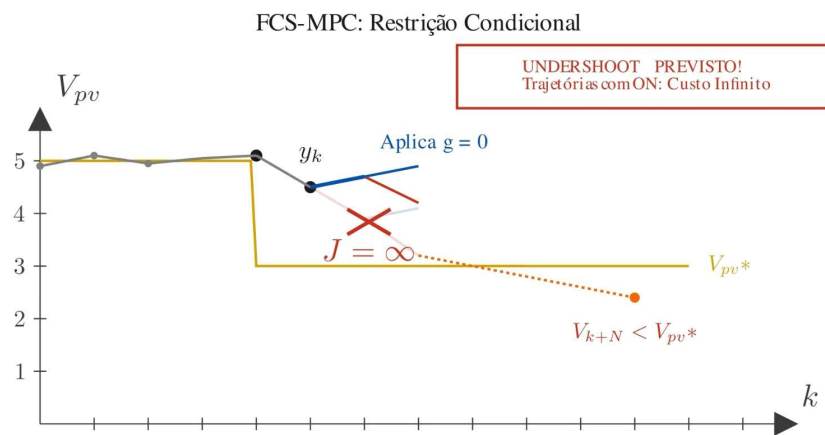
$$J = \begin{cases} \infty, & \text{se } v_{pv}(k+N)|_{g=1} > v_{pv}^*, \quad \text{para } \Delta t^* \leq t' \text{ e } g(k) = 1, \\ (v_{pv}^* - v_{pv}(k+2))^2, & \text{para os demais casos.} \end{cases} \quad (3.68)$$

Ressalta-se que fora do intervalo transitório ($\Delta t^* \leq t'$) e para os casos em que a predição não antevê um *overshoot* ou *undershoot*, o algoritmo e o cálculo das funções custo são realizados da mesma forma que o FCS-MPC sem restrições.

Esta lógica do algoritmo é apresentada no fluxograma do Apêndice B. Além disso, de modo análogo ao feito para o caso do FCS-MPC Irrestrito, foi desenvolvida uma animação utilizando a linguagem Python para ilustrar melhor a dinâmica do algoritmo.²

Esta animação permite observar o comportamento do horizonte de predição estendido ($k + N$) nos instantes após a mudança de referência, bem como a penalização da função custo caso a restrição seja violada como ilustrado no quadro da Figura 18.

Figura 18 – Quadro da tomada de decisão do FCS-MPC com restrição condicional.



Fonte: Elaborado pelo Autor (2025).

3.2.3 Controle com Restrição na Tensão e Horizonte de Predição Estendido

Por fim, como alternativa à restrição condicional rígida, foi implementada uma função de custo composta que integra a regulação de curto prazo com uma avaliação de tendência de longo prazo. Esta estratégia estende o horizonte de predição da tensão de entrada para N_1 amostras, onde $N_1 > 2$, introduzindo um termo de penalidade ponderado por um coeficiente λ .

Esta solução une premissas das duas propostas anteriores. A base do algoritmo continua sendo o erro quadrático e o termo do erro da tensão predita após dois instantes de tempo continua sendo utilizado. Ademais, de modo análogo ao algoritmo com restrição condicional, realiza-se a extensão do horizonte de predição,

² Para visualizar a dinâmica temporal deste algoritmo, uma animação completa em vídeo foi elaborada e encontra-se disponível através do link: <https://youtu.be/Fmn52tKTYoE>.

considerando apenas os casos extremos. Isto é, não são consideradas todas as 2^{N_1} possibilidades de chaveamento a cada iteração, apenas os casos de manutenção da chave ativada ($g(k) = 1$) ou desativada ($g(k) = 0$) por N_1 instantes de tempo consecutivos. No entanto, essa ampliação não se limita aos períodos transitórios com anteriormente, neste caso, os horizontes longos são calculados sempre que o código é executado.

Portanto, a função de custo proposta é definida pela soma ponderada do erro quadrático no horizonte de controle padrão ($k + 2$) e do erro quadrático no horizonte estendido ($k + N_1$), expressa matematicamente por:

$$J = (V_{pv}^* - v_{pv}(k + 2))^2 + \lambda(V_{pv}^* - v_{pv}(k + N_1))^2 \quad (3.69)$$

A composição das funções custos finais guarda ainda um detalhe técnico. Como são feitas duas previsões a longo prazo, é preciso entender qual delas será utilizada em quais funções custo. Assim como para os métodos anteriores, ao final da execução do código quatro funções custo são calculadas. O algoritmo associa a previsão a longo prazo à primeira ação de controle tomada na sequência de chaveamento predita. Isto significa, por exemplo, que se a sequência começa com $g(k) = 1$, utiliza-se a penalidade calculada pela trajetória continuamente ativa ($v_{pv\ N,on}$). Caso inicie com a chave sem conduzir, $g(k) = 0$, utiliza-se a trajetória desativada ($v_{pv\ N,off}$).

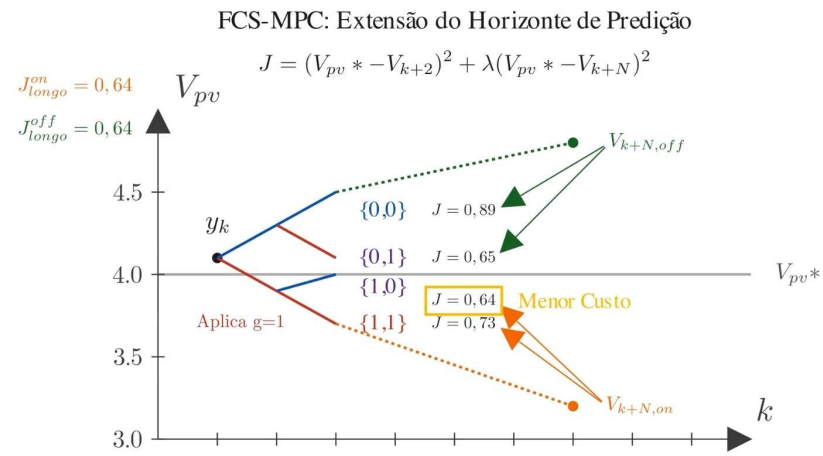
Desta forma as funções custo podem ser representadas da seguinte forma:

$$\begin{cases} J_{on,on} = (V_{pv}^* - v_{pv}(k + 2))^2 + \lambda(V_{pv}^* - v_{pv\ N,on})^2 \\ J_{on,off} = (V_{pv}^* - v_{pv}(k + 2))^2 + \lambda(V_{pv}^* - v_{pv\ N,on})^2 \\ J_{off,on} = (V_{pv}^* - v_{pv}(k + 2))^2 + \lambda(V_{pv}^* - v_{pv\ N,off})^2 \\ J_{off,off} = (V_{pv}^* - v_{pv}(k + 2))^2 + \lambda(V_{pv}^* - v_{pv\ N,off})^2 \end{cases} \quad (3.70)$$

O fluxo detalhado de execução deste código encontra-se no fluxograma do Apêndice C. Como elaborado para os demais métodos, também foi feita uma animação para este algoritmo.³ A Figura 19 ilustra a tomada de decisão a partir da restrição de tensão, de acordo com o resultado da soma dos custos das previsões em dois instantes de tempo ($k + 2$) e dos horizontes mais longos ($k + N_1$).

³ Para visualizar a dinâmica temporal deste algoritmo, uma animação completa em vídeo foi elaborada e encontra-se disponível através do link: <https://youtu.be/MXgeFKAHUos>.

Figura 19 – Quadro da tomada de decisão do FCS-MPC com extensão do horizonte de previsão.



Fonte: Elaborado pelo Autor (2025).

4 RESULTADOS

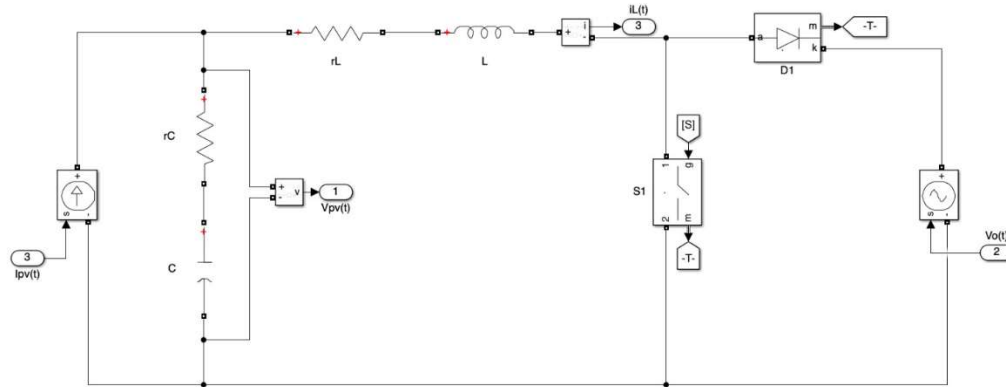
Neste capítulo, são detalhados os resultados da aplicação decorrente da utilização do conversor *boost* com os algoritmos FCS-MPC que foram propostos no capítulo anterior. Começa-se pelos resultados da validação do modelo no espaço de estados e linearizado. Em seguida, aprofundando a análise, o objetivo é direcionado para a observação da tensão de entrada do conversor, sob a ação de cada um dos algoritmos propostos.

As métricas postas na seção 2.5 são utilizadas a fim de quantificar o desempenho dessas estratégias de controle. É realizada a comparação entre a operação do FCS-MPC com e sem restrição de função de custo, o que permite averiguar as melhorias advindas das restrições. Para isso, os parâmetros das restrições são adequados, com a intenção de obter os menores valores para os parâmetros IAE, ITAE, ISE e ITSE. Posteriormente, os controles são avaliados em termos do tempo de assentamento, máximo sobressinal percentual e ondulação da tensão de saída. Finalmente, o algoritmo que apresenta o melhor desempenho é comparado com as aplicações de controle clássico.

4.1 Ambiente de Simulação

A simulação da configuração do conversor *boost* fotovoltaico como apresentado na Figura 14 foi implementada com auxílio do *software* do MATLAB/Simulink, como é possível observar na Figura 20, e é a base para obtenção dos resultados deste trabalho. Nota-se que o painel fotovoltaico é emulado sendo uma fonte ideal de corrente, que serve como uma das entradas do sistema, em conjunto com a fonte de tensão na saída do conversor. Os parâmetros da simulação seguem sendo os mesmos da Tabela 4.

Figura 20 – Circuito do conversor *boost* fotovoltaico em espaço de simulação.

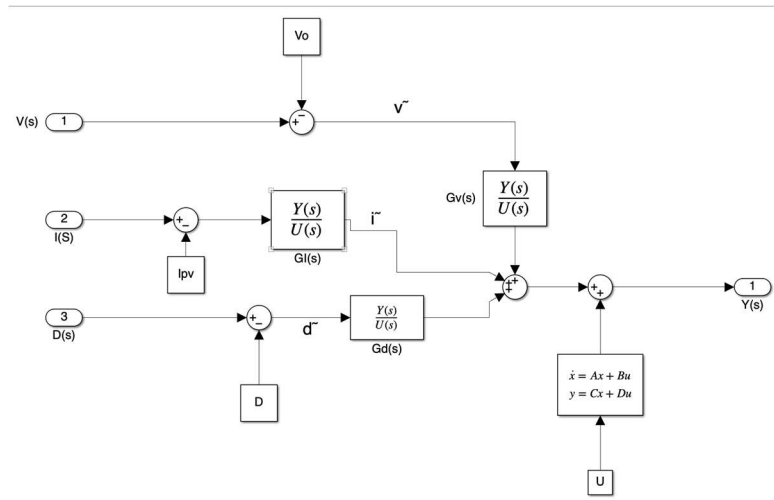


Fonte: Elaborado pelo Autor (2025).

4.1.1 Validação do sistema

Com o circuito montado em ambiente de simulação, parte-se para a validação do modelo como evidenciado na subseção 3.1.6. A Figura 21 mostra em mais detalhes como a validação foi obtida. Para se obter a saída do sistema $Y(s)$, neste caso a tensão de entrada do conversor $v_{pv}(s)$, foram utilizadas as funções de transferência do modelo de pequenos sinais como apresentadas na equação (3.57) e o modelo em médio em espaço de estados ponderado pela razão cíclica D , como nas equações (3.47) até (3.50). Desta forma, se somam as parcelas das variações das entradas com a saída do modelo médio se alcança a resposta do modelo como um todo.

Figura 21 – Simulação do Modelo linearizado por pequenos sinais.



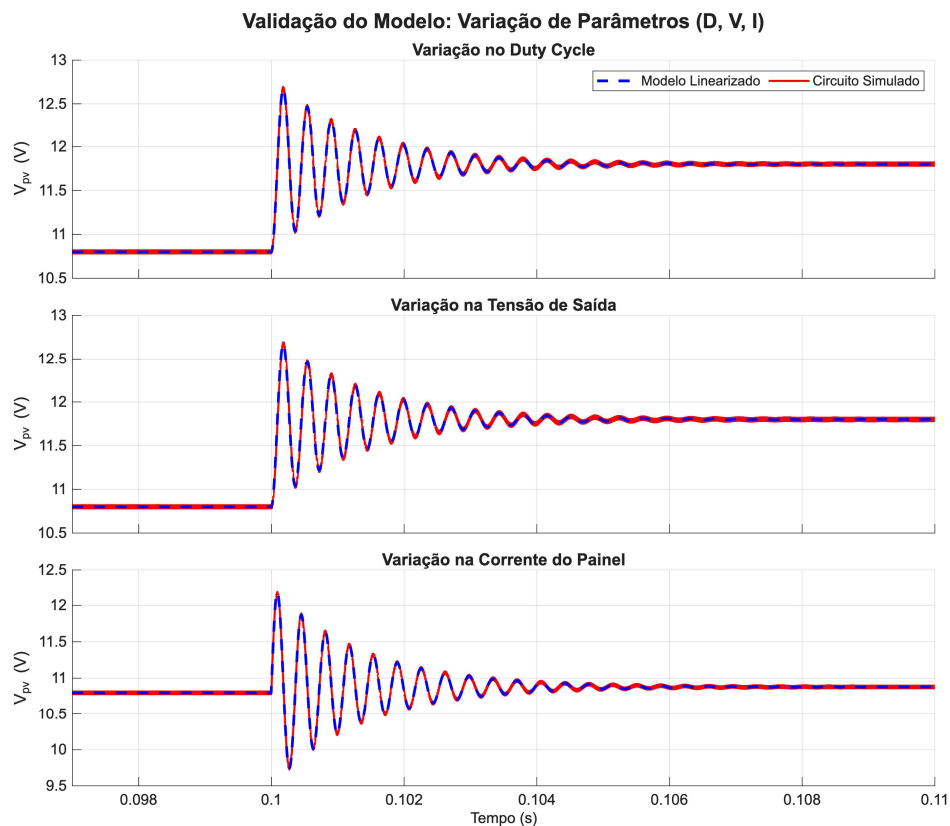
Fonte: Elaborado pelo Autor (2025)

Os testes para validação seguiram a seguinte sequência e estão presentes nas Figura 22 e 23 ,

1. Uma variação de -10% ao ciclo de trabalho D , enquanto as demais entradas se mantiveram no ponto de operação.
2. Um degrau de 10% à tensão de saída v_o , mantendo as demais entradas no ponto de operação;
3. Uma mudança de 10% à corrente do painel I_{pv} , com as demais entradas no ponto de operação;
4. Por fim aplicou-se a variação de 10% a todas as entradas simultaneamente.

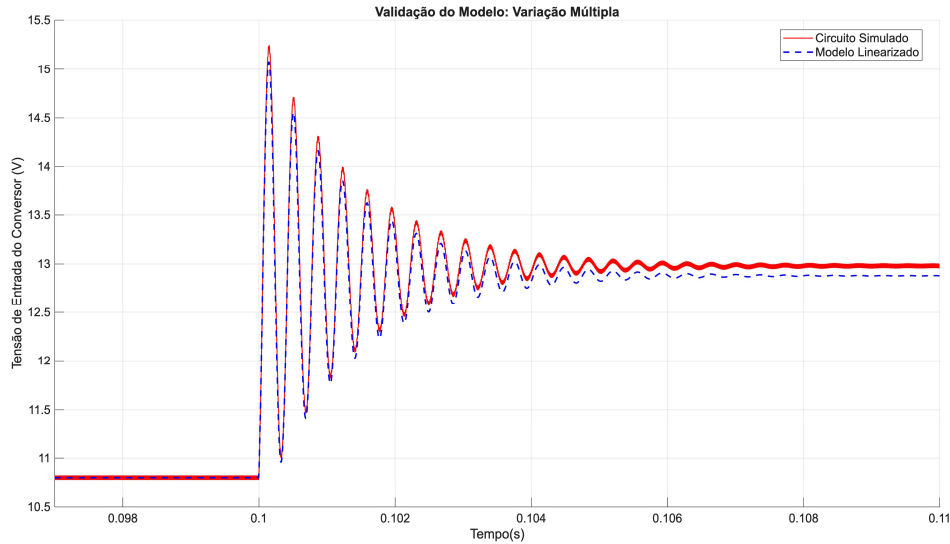
Analisando os gráficos da Figura 22 é possível afirmar que o modelo é bem sucedido em emular a resposta dinâmica do conversor proposto quando há uma variação em um dos sinais de entrada.

Figura 22 – Resultado Gráfico das variações em torno do ponto de operação separadamente.



Nota-se, entretanto, que quando há uma variação em todas as entradas simultaneamente, o modelo linearizado passa a ter um pequeno desvio (*offset*) em relação ao circuito simulado, como ilustrado na Figura 23.

Figura 23 – Resultado Gráfico das variações em torno do ponto de operação simultaneamente.



Fonte: Elaborado pelo Autor (2025).

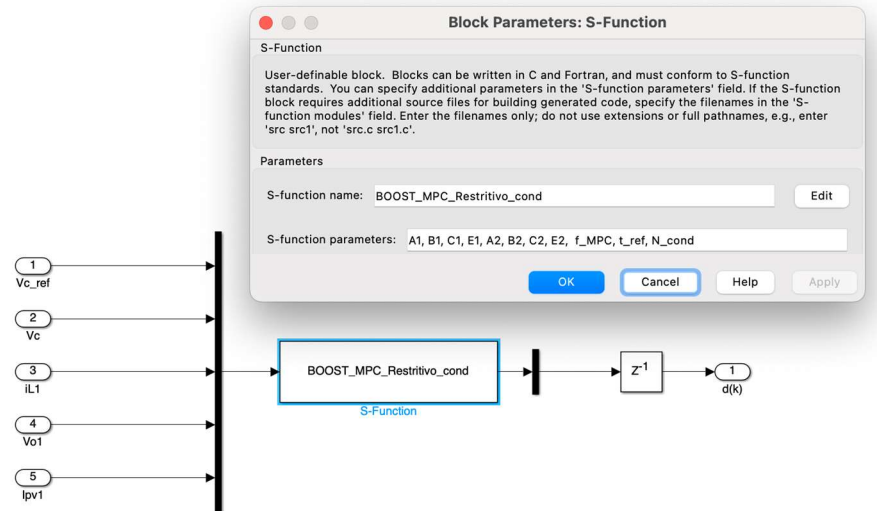
4.1.2 Preparação para aplicar o FCS-MPC

Tendo validado o modelo, passa-se ao algoritmo MPC. Por se tratar de um algoritmo digital é preciso determinar uma frequência de amostragem fixa para o FCS-MPC, portanto para averiguar as condições de operação do algoritmo, foram feitos testes com diferentes frequências com a função custo do erro quadrático sem restrições (COSTA, 2025). Chegou-se a um resultado considerado satisfatório, com uma ondulação em regime permanente menor que 10% da tensão de referência ($\Delta V_{pv} < 0,1 \cdot V_{pv}^*$), a uma frequência f_{MPC} de 200kHz. A partir disso, as simulações foram realizadas com base na taxa de amostragem e ação do MPC estipuladas.

Com a frequência de operação determinada, é preciso que o controlador atue periodicamente de forma consistente, diferentemente das funções em MATLAB que necessita de um sinal de disparo externo, o código na linguagem C encapsulado dentro de um bloco S-Function, como ilustrado na Figura 24, pode definir internamente a frequência de operação do controlador. Além disso, o bloco de S-Function permite

determinar parâmetros estáticos durante a simulação, o que simplifica a implementação das simulações.

Figura 24 – Implementação do algoritmo em S-Function.

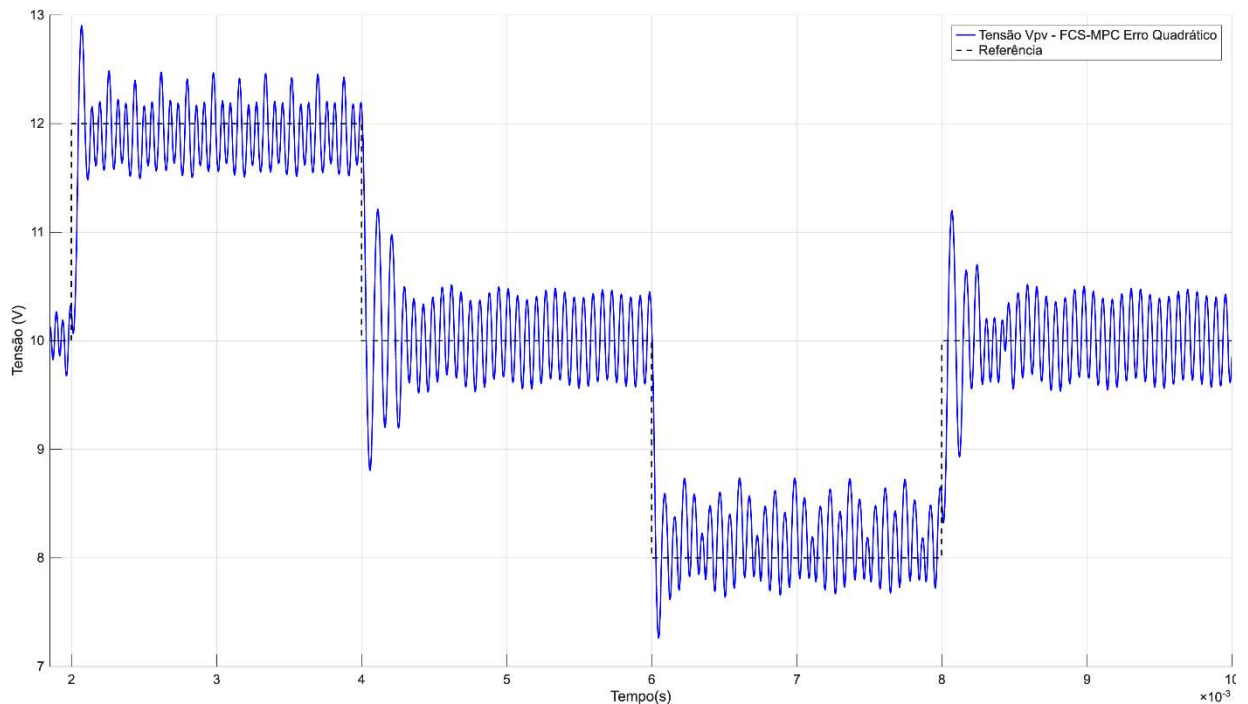


Fonte: Elaborado pelo Autor (2025)

4.2 Simulação do FCS-MPC com Erro Quadrático

Após a construção do cenário de simulação em MATLAB/Simulink, o próximo passo foi a aplicação da primeira proposta de controle preditivo baseado em modelo deste estudo, com o código base mostrado no Apêndice D, para a função de custo do erro quadrático sem restrições. Varia-se a referência por quatro vezes, com degraus de +2, -2, -2 e +2V, nessa ordem, e o resultado gráfico da simulação do controle da tensão de entrada do conversor *boost* fotovoltaico se encontra na Figura 25. A análise do seu desempenho é dada em termos dos parâmetros expostos na seção 2.5. São eles: a ondulação, ΔV_{pv} , o tempo de assentamento, t_s e os três termos de máximo sobressinal: o absoluto, o percentual e o relativo ao degrau aplicado. Os resultados numéricos desta análise encontram-se na Tabela 5.

Figura 25 – Resultado da aplicação do controle com a função custo do erro quadrático



Fonte: Elaborado pelo Autor (2025).

Ao se analisar a resposta no tempo e os índices de desempenho do algoritmo, é possível tirar algumas conclusões preliminares. A primeira delas é que o controle foi bem-sucedido e rápido dinamicamente. Antes de 1ms ele chegou à condição de regime permanente em todos os degraus. No entanto, é importante considerar que em sua operação, o nível de *ripple* (ΔV_{pv}) ainda está relativamente alto. Em termos relativos a tensão de referência a ondulação gira em torno de 10%. Assim como o máximo sobressinal percentual que chega a ser de 12% em duas das respostas ao degrau avaliadas.

Tabela 5 – Índices de desempenho para o FCS-MPC - Erro Quadrático.

| Referência | $M_{p_{abs}}$ | $M_p(\%)$ | $M_{p_{rel}}(\%)$ | $t_s(\mu s)$ | ΔV_{pv} |
|------------|---------------|-----------|-------------------|--------------|-----------------|
| 10 → 12V | 0,90 | 7,5 | 45,15 | 980,05 | 0,95 |
| 12 → 10V | 1,20 | 12,0 | 59,8 | 948,8 | 0,93 |
| 10 → 8V | 0,74 | 9,3 | 37,03 | 988,20 | 1,06 |
| 8 → 10V | 1,20 | 12,0 | 59,99 | 984,35 | 0,93 |

Fonte: Elaborado pelo Autor (2025).

4.3 Simulação do FCS-MPC com Restrição Condicional

Foi com o intuito de se reduzir o *overshoot* da resposta do erro quadrático que se propôs o segundo algoritmo FCS-MPC deste trabalho. A restrição condicional como exposta no capítulo anterior, opera justamente nos momentos propícios para a ocorrência do sobressinal, durante o transitório entre duas referências.

No entanto, nota-se que o algoritmo apenas menciona os termos N e t' , mas não define valores para eles. Isto foi feito visando testar diversas combinações possíveis dos dois termos, para se avaliar o desempenho do controle para cada possibilidade, através da métricas de integral do erro e pelo máximo sobressinal apresentadas na seção 2.5. a faixa de valores analisados para t' foi de:

$$0,00 \leq t' \leq 1ms, \quad \text{com } \Delta t' = 5\mu s; \quad (4.1)$$

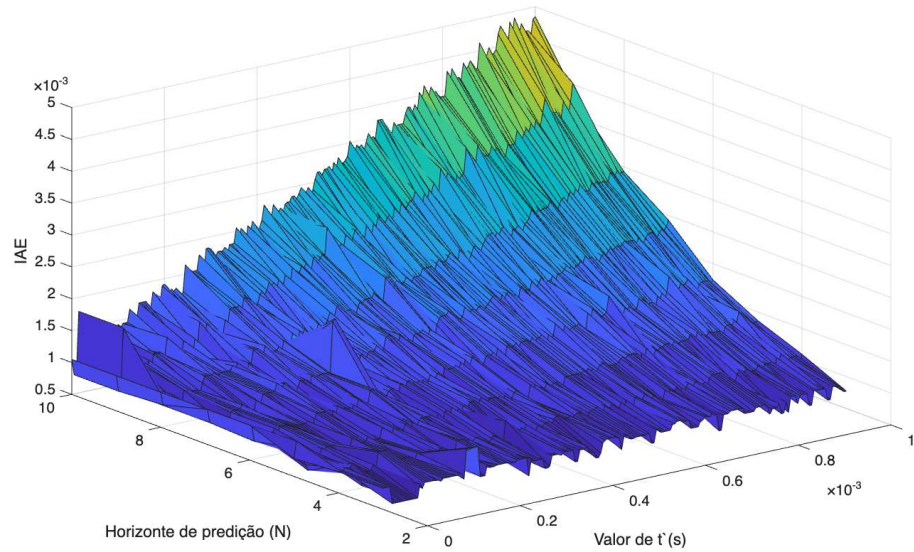
O passo foi definido como o inverso da frequência f_{MPC} de modo que o algoritmo sempre coincida com o tempo limite da restrição, t' . Além disso o valor final da faixa analisada é de 1ms, tempo considerado suficiente para o sinal chegar em regime permanente, como observado pelo desempenho do algoritmo irrestrito apresentado há pouco.

Já o termo de extensão do horizonte de predição (N) foi operado na seguinte faixa:

$$3 \leq N \leq 10, \quad \text{com } \Delta N = 1. \quad (4.2)$$

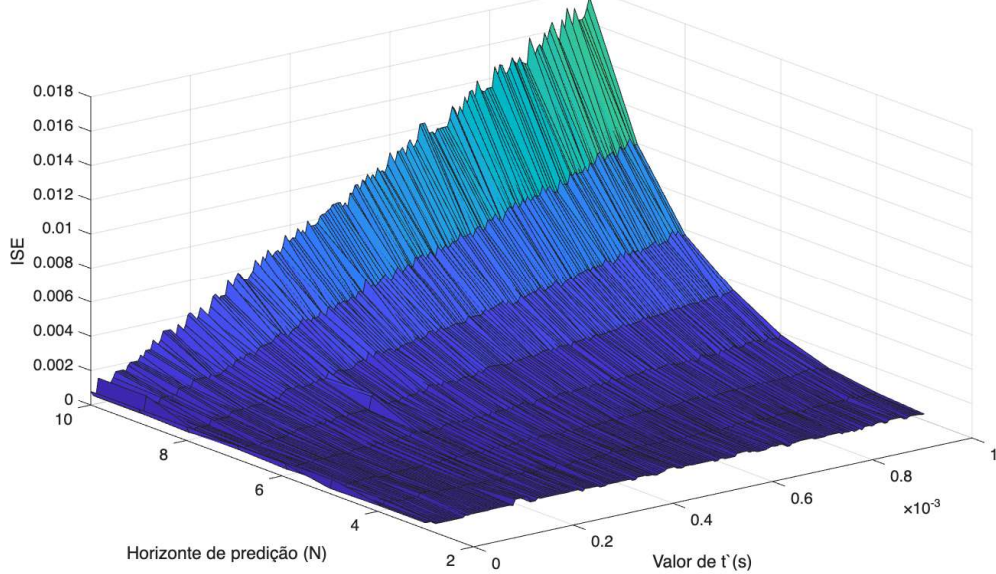
Os resultados gráficos da varredura para os parâmetros de desempenho: IAE, ISE ITAE, ITSE e sobressinal estão representados nas Figuras 26, 27, 28, 29 e 30 respectivamente. Investigando os gráficos, fica evidente que o melhor desempenho é dado por horizontes de predição relativamente baixos, entre 3 e 5 ou para os casos de t' baixo. A curva que mais divergiu desse padrão foi a de sobressinal, mas esse resultado é natural, dado que o princípio matemático do sobressinal é totalmente distinto dos demais.

Figura 26 – Gráfico de IAE x N x t'.



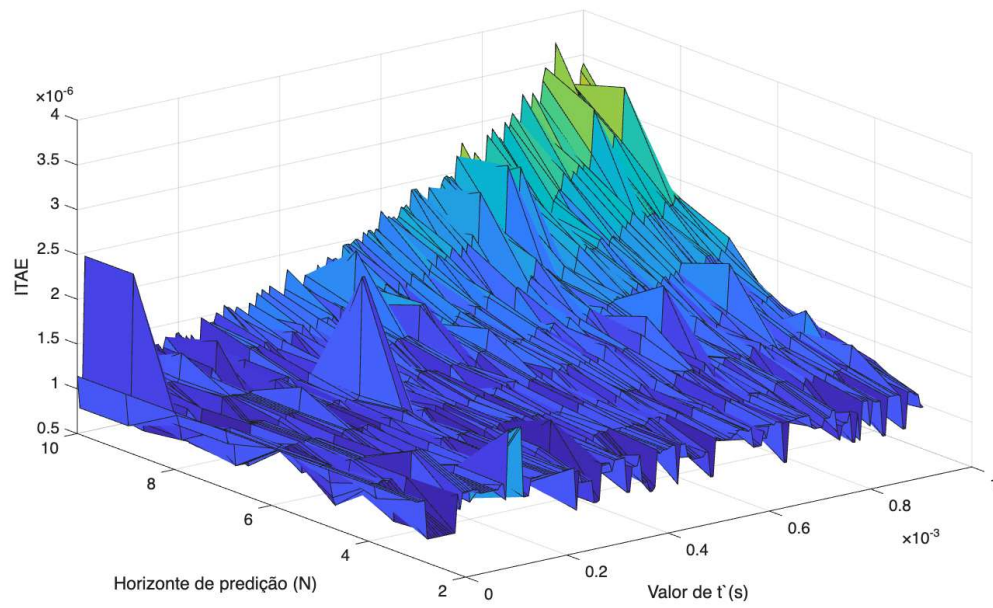
Fonte: Elaborado pelo Autor (2025)

Figura 27 – Gráfico de ISE x N x t'.



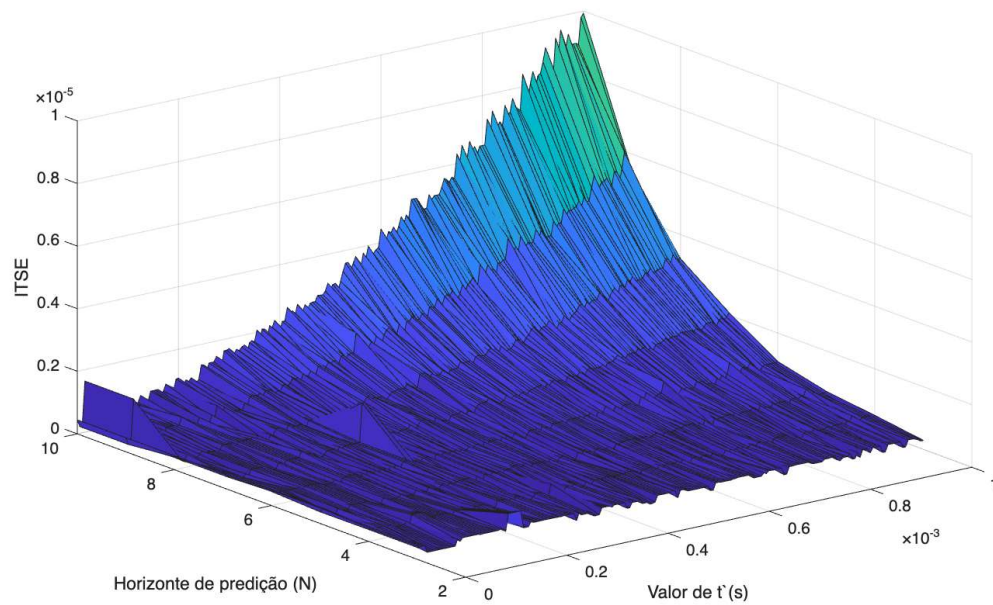
Fonte: Elaborado pelo Autor (2025)

Figura 28 – Gráfico de ITAE x N x t' .



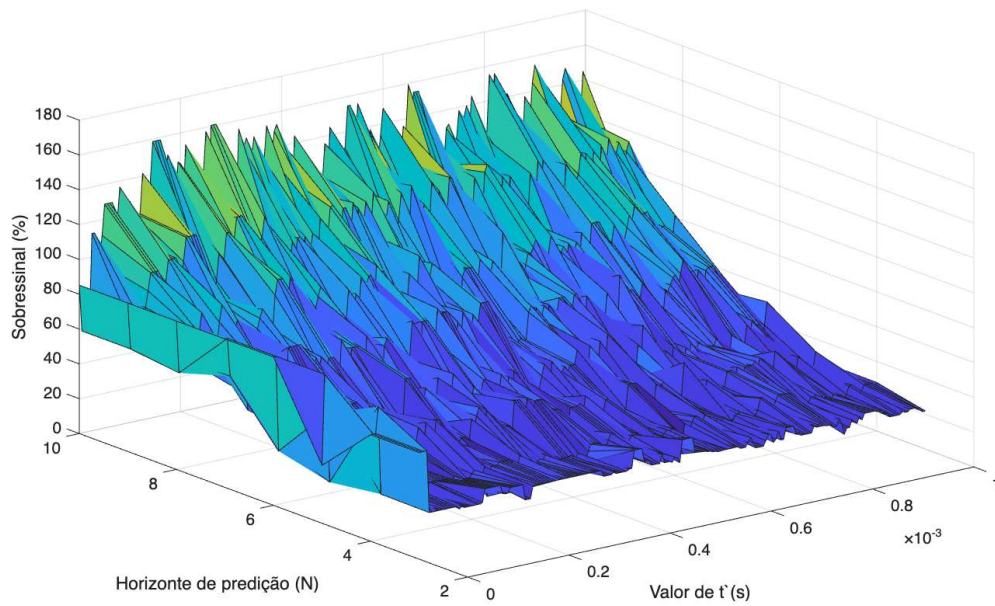
Fonte: Elaborado pelo Autor (2025)

Figura 29 – Gráfico de ITSE x N x t' .



Fonte: Elaborado pelo Autor (2025)

Figura 30 – Gráfico de Máximo Sobressinal x N x t' .



Fonte: Elaborado pelo Autor (2025).

Foram encontradas as combinações de t' e N que minimizam cada um dos parâmetros de desempenho. A fim de escolher qual é a melhor solução globalmente, calculou-se o desempenho relativo de cada métrica em relação ao seu valor mínimo global para cada dupla de t' e N . Como exemplo, o valor relativo de ISE dada uma combinação qualquer de t' e N é calculado utilizando a seguinte equação:

$$ISE_{rel}(t', N) = \frac{ISE_{rel}(t', N) - ISE_{min}}{ISE_{min}} \times 100. \quad (4.3)$$

A Tabela 6 apresenta as combinações que conseguiram minimizar ao menos um dos parâmetros e valor relativos das demais métricas.

A solução com as variáveis $t' = 50$ e $N = 4$, apresentou um resultado ótimo para três das cinco métricas. Ainda que a combinação 2 tenha um desempenho ótimo em relação aos indicadores de integral do erro, ela é mais problemática na questão do máximo sobressinal, que vai de encontro à proposta original deste algoritmo de amortecer o sistema e atenuar o *overshoot*. Escolheu-se, portanto, a primeira combinação.

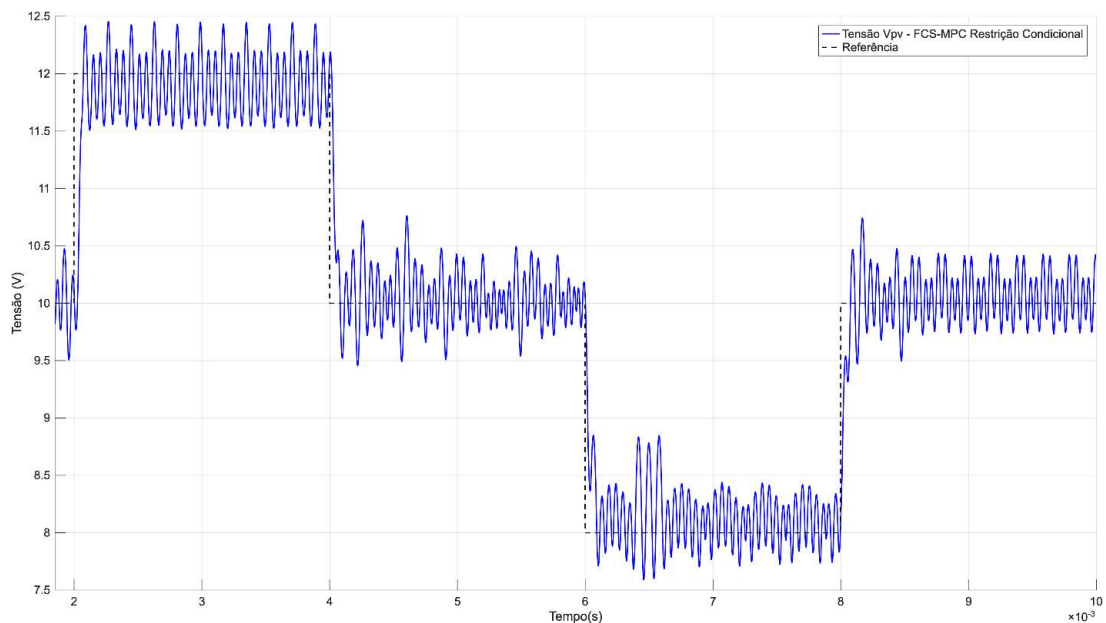
Tabela 6 – Comparação das soluções - Restrição Condicional

| Combinação | t' (μ s) | N | $ITAE_{rel}$ | $ITSE_{rel}$ | IAE_{rel} | ISE_{rel} | $Mp(\%)$ |
|------------|-----------------|---|--------------|--------------|-------------|-------------|----------|
| 1 | 50 | 4 | 0% | 0% | 1,4% | 33,5% | 133,4% |
| 2 | 315 | 3 | 0,2% | 0,2% | 0% | 5,2% | 169,1% |
| 3 | 20 - 75 | 3 | 6,1% | 15,7% | 5,9% | 0% | 144,1% |
| 4 | 275 | 4 | 13,0% | 45,8% | 22,5% | 83% | 0% |

Fonte: Elaborado pelo Autor (2025).

Realiza-se a simulação a partir do código do Apêndice E com a combinação 1 e a resposta dinâmica do controle MPC condicionalmente restrito, é apresentada na Figura 31. A princípio, não se nota visualmente uma diferença tão considerável em relação à dinâmica obtida pelo algoritmo anterior, especialmente em termos de tempo de assentamento e ondulação.

Figura 31 – Resultado da aplicação do controle com a função custo com restrição condicional.



Fonte: Elaborado pelo Autor (2025).

No entanto, os índices de desempenho, expostos na Tabela 7, mostram uma redução pela metade no máximo sobressinal e uma redução da ondulação para os

dois degraus finais. Pode-se afirmar que a restrição condicional foi bem-sucedida em amortecer o sinal durante os transitórios.

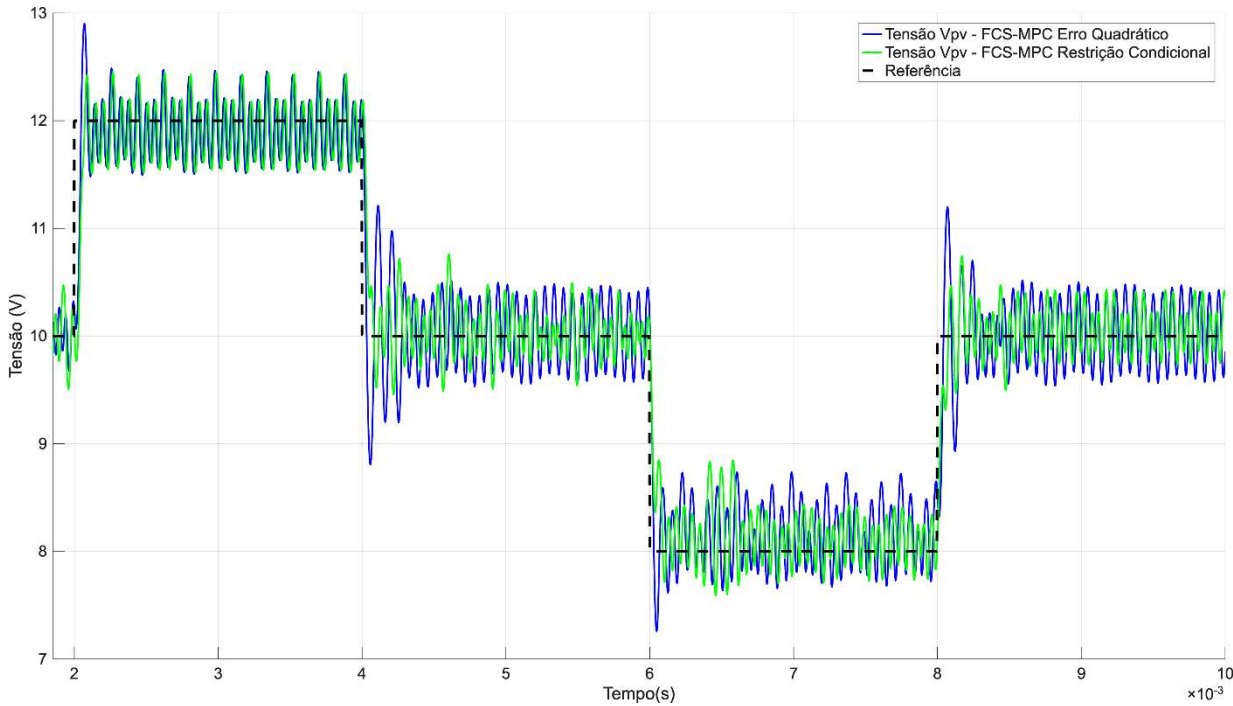
Tabela 7 – Índices de desempenho para o FCS-MPC – Restrição Condicional

| Referência | Mp_{abs} | $Mp_{(%)}$ | $Mp_{rel} \text{ (}\%)$ | $t_s \text{ (}\mu\text{s)}$ | ΔV_{pv} |
|------------|------------|------------|-------------------------|-----------------------------|-----------------|
| 10 → 12V | 0,46 | 3,8 | 22,83 | 989,32 | 0,93 |
| 12 → 10V | 0,54 | 5,4 | 27,23 | 913,6 | 0,95 |
| 10 → 8V | 0,41 | 5,1 | 20,55 | 895,3 | 0,73 |
| 8 → 10V | 0,74 | 7,4 | 37,15 | 764,82 | 0,70 |

Fonte: Elaborado pelo Autor (2025).

Por fim, compara-se graficamente os sinais dos controles apresentados até aqui. A Figura 32 torna mais claro aquilo evidenciado pela tabela dos índices de desempenho. É perceptível que o sistema passou a ter menos sobressinal e que a faixa de oscilação é levemente menor para as duas últimas variações de referência.

Figura 32 – Gráfico comparativo entre o FCS-MPC Quadrático e o Condicional.



Fonte: Elaborado pelo Autor (2025)

4.4 Simulação do FCS-MPC com Restrição de Tensão com Horizonte de Predição Estendido

A última função custo proposta é a operação do conversor com o FCS-MPC com a função custo restrita em termos de tensão com horizonte estendido proporciona a resposta representada determinação do termo de ponderação e do horizonte de predição estendido, λ_1 e N_1 respectivamente, é indispensável para desenvolver a operação do FCS-MPC com restrição em termos de tensão, com horizonte de predição estendido, definido na subseção 3.2.3. De modo análogo ao ocorrido para a função custo condicional, foi realizada uma varredura automática, que consiste na variação dos valores dentro de uma faixa definida previamente, como

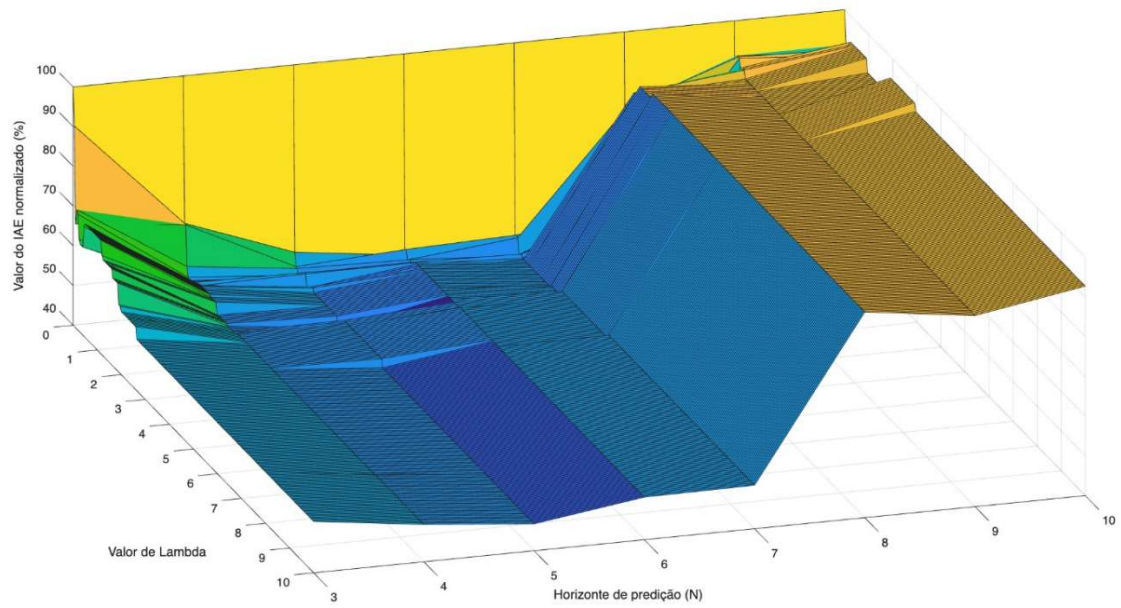
$$0,00 \leq \lambda_1 \leq 10,00, \quad \text{com } \Delta\lambda = 0,05; \quad (4.4)$$

$$3 \leq N_1 \leq 10, \quad \text{com } \Delta N_1 = 1. \quad (4.5)$$

Os resultados obtidos para a varredura para os parâmetros IAE, ISE, ITAE e ITSE normalizados são mostrados nas Figuras 33, 34, 35, 36 respectivamente. As curvas mostram consistentemente um vale na faixa de 5 horizontes de predição se estendendo praticamente por todos os valores de λ_1 . Destaca-se de antemão que para o valor nulo de lambda o controle mostra o pior desempenho para todos os parâmetros.

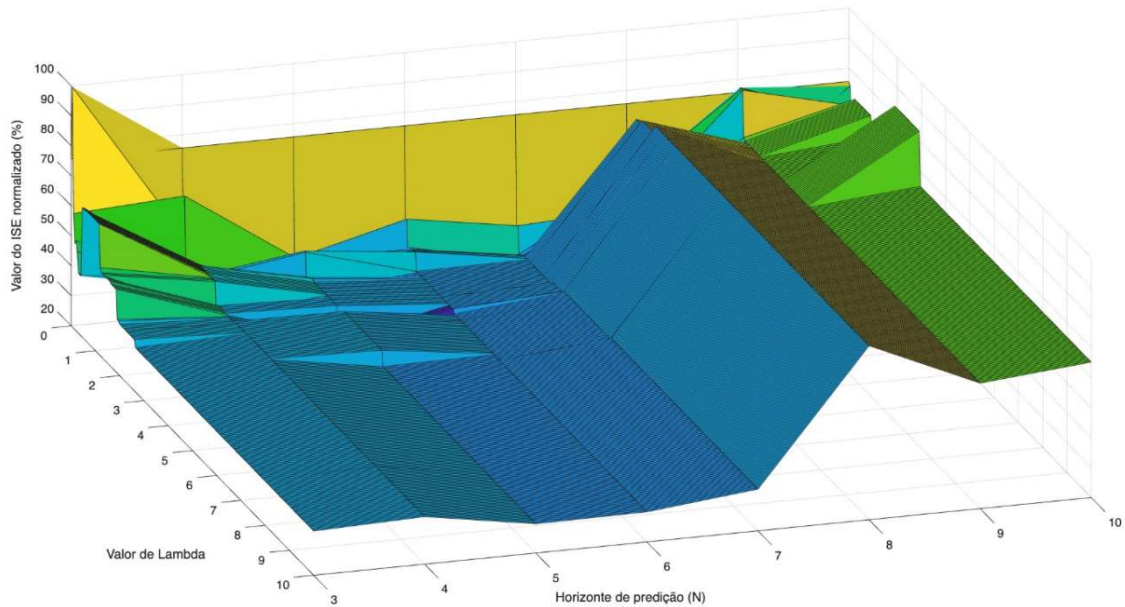
Nota-se que fazer λ_1 igual a zero é o mesmo que manter a função custo com o erro quadrático sem restrição. Portanto, antes mesmo de se analisar a resposta desta solução de controle no tempo, é possível concluir que o desempenho dinâmico do controle com restrição de tensão e horizonte estendido será melhor que o puramente irrestrito, sendo verdade para qualquer dupla de valores de λ_1 e N_1 .

Figura 33 – Gráfico de IAE x N x λ .



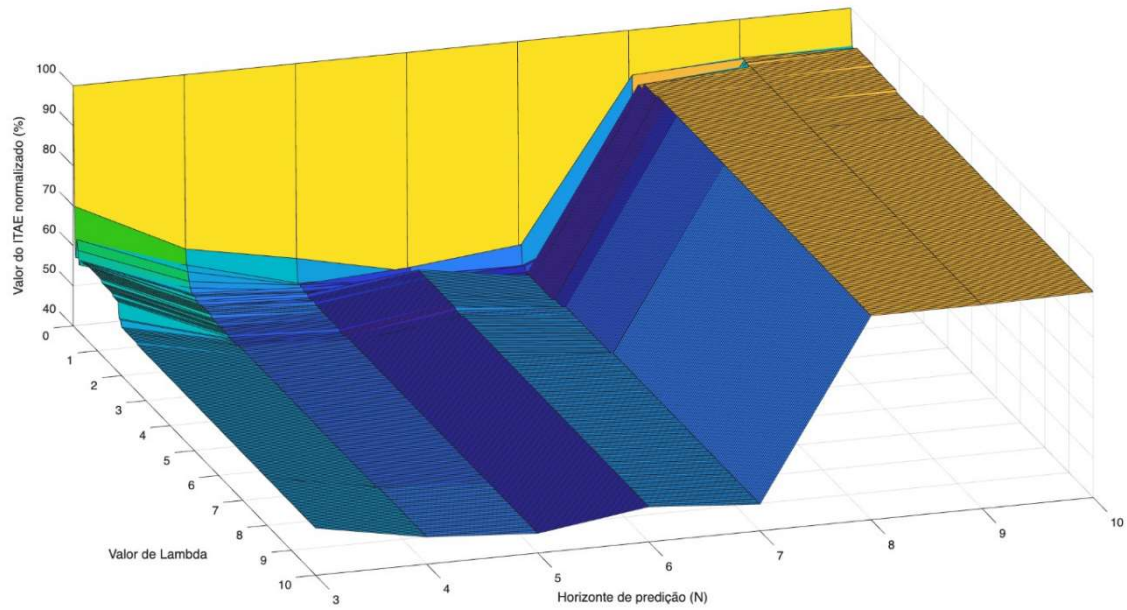
Fonte: Elaborado pelo Autor (2025)

Figura 34 – Gráfico de ISE x N x λ .



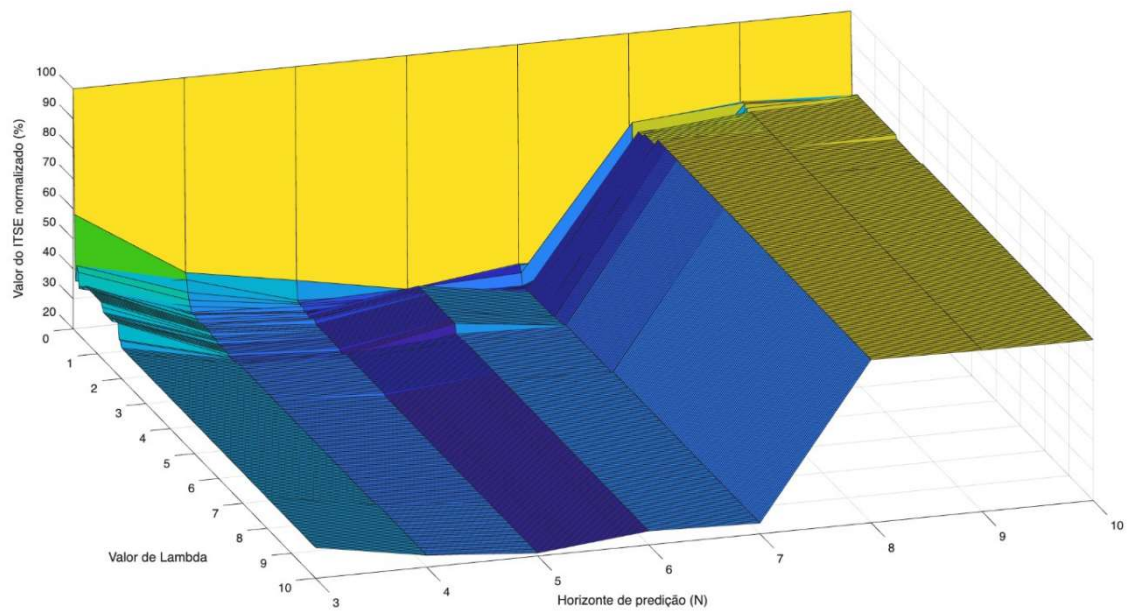
Fonte: Elaborado pelo Autor (2025)

Figura 35 – Gráfico de ITAE x N x λ .



Fonte: Elaborado pelo Autor (2025)

Figura 36 – Gráfico de ITSE x N x λ .



Fonte: Elaborado pelo Autor (2025).

Assim como para o algoritmo anterior, foram encontradas as combinações das variáveis da varredura que otimizam cada um dos parâmetros. Os resultados estão reunidos na Tabela 8. Para os índices IAE, ISE, ITSE, quatro possibilidades de combinação são ótimas, enquanto apenas uma solução minimiza ITAE. Dado que os resultados mostram uma convergência para $\lambda \approx 2$ e $N = 5$. Estes foram os valores escolhidos para formar função custo.

Tabela 8 – Comparação das soluções - Restrição de tensão – horizonte estendido.

| Combinação | λ | N | IAE_{rel} | $ITAE_{rel}$ | ISE_{rel} | $ITSE_{rel}$ |
|------------|-----------|---|-------------|--------------|-------------|--------------|
| 1 | 1,8 – 2,0 | 5 | 0% | 0,2% | 0% | 0% |
| 2 | 2,05 | 5 | 0,9% | 0% | 2,9% | 0,5% |

Fonte: Elaborado pelo Autor (2025).

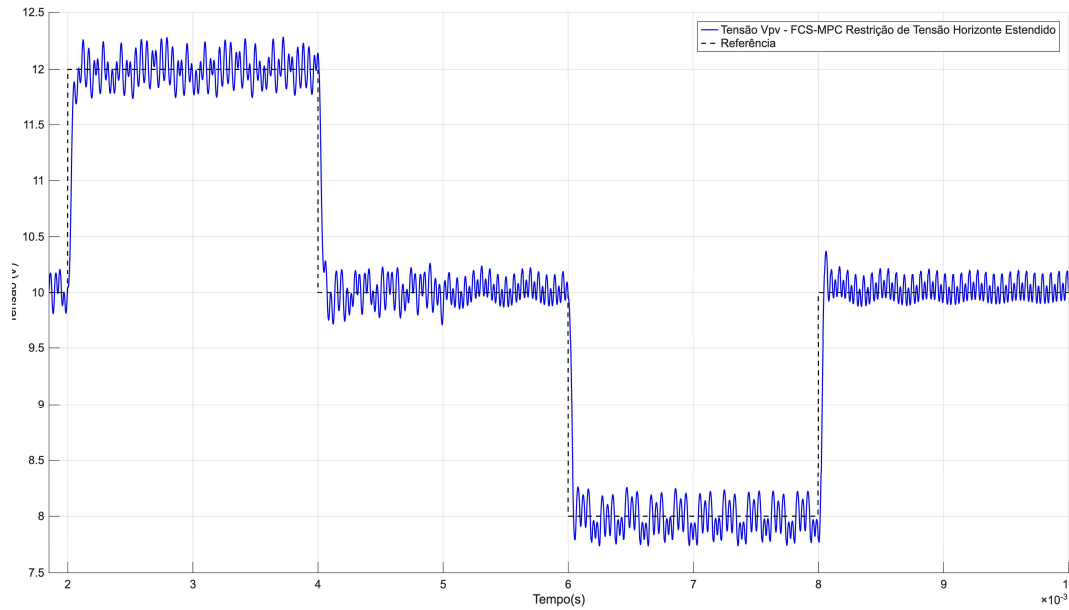
Em seguida, o código, como evidenciado no Apêndice F, foi executado e a resposta dinâmica do controle MPC com restrição de tensão com a combinação escolhida é apresentada na Figura 37. É possível notar que o desempenho deste controle foi melhor que os dois anteriormente mostrados, especialmente em termos de sobressinal e ondulação. Fato este que é corroborado pelos índices expostos na Tabela 9. Destaca-se a diminuição do sobressinal, assim como a redução significativa da ondulação em regime permanente quando comparado com as demais soluções.

Tabela 9 – Índices de desempenho para o FCS-MPC – Restrição de Tensão

| Referência | Mp_{abs} | $Mp_{(%)}$ | $Mp_{rel} (%)$ | $t_s (\mu s)$ | ΔV |
|------------|------------|------------|----------------|---------------|------------|
| 10 → 12V | 0,28 | 2,3 | 13,93 | 522,27 | 0,54 |
| 12 → 10V | 0,29 | 2,9 | 14,32 | 1000,05 | 0,45 |
| 10 → 8V | 0,27 | 3,4 | 13,4 | 862,45 | 0,50 |
| 8 → 10V | 0,37 | 3,7 | 18,61 | 885,8 | 0,32 |

Fonte: Elaborado pelo Autor (2025).

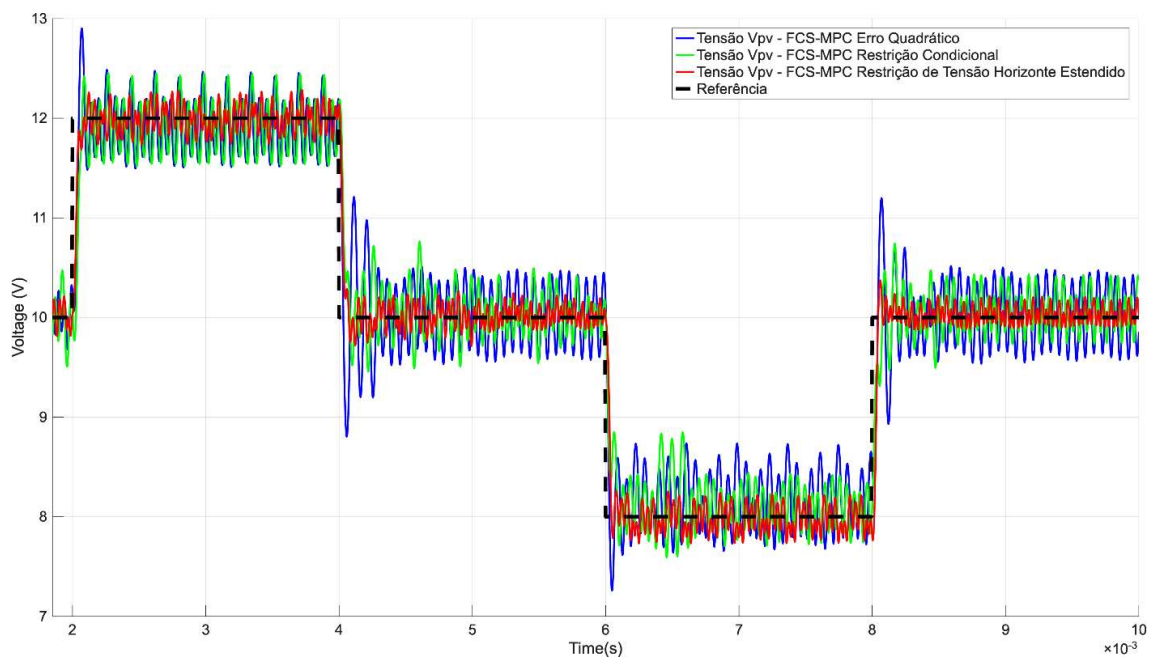
Figura 37 – Resultado do controle com a função custo com restrição de tensão e horizonte estendido



Fonte: Elaborado pelo Autor (2025)

Por fim, é feita a seguir uma comparação gráfica da resposta dos três controles, como visualizado na Figura 38. Além disso é realizada uma síntese dos resultados numéricos para cada solução proposta na Tabela 10. Pela qual é possível concluir ainda com mais clareza que o último algoritmo apresentou a melhor das dinâmicas.

Figura 38 – Comparação dos três algoritmos FCS-MPC propostos.



Fonte: Elaborado pelo Autor (2025).

Tabela 10 – Síntese da resposta dinâmica do sistema.

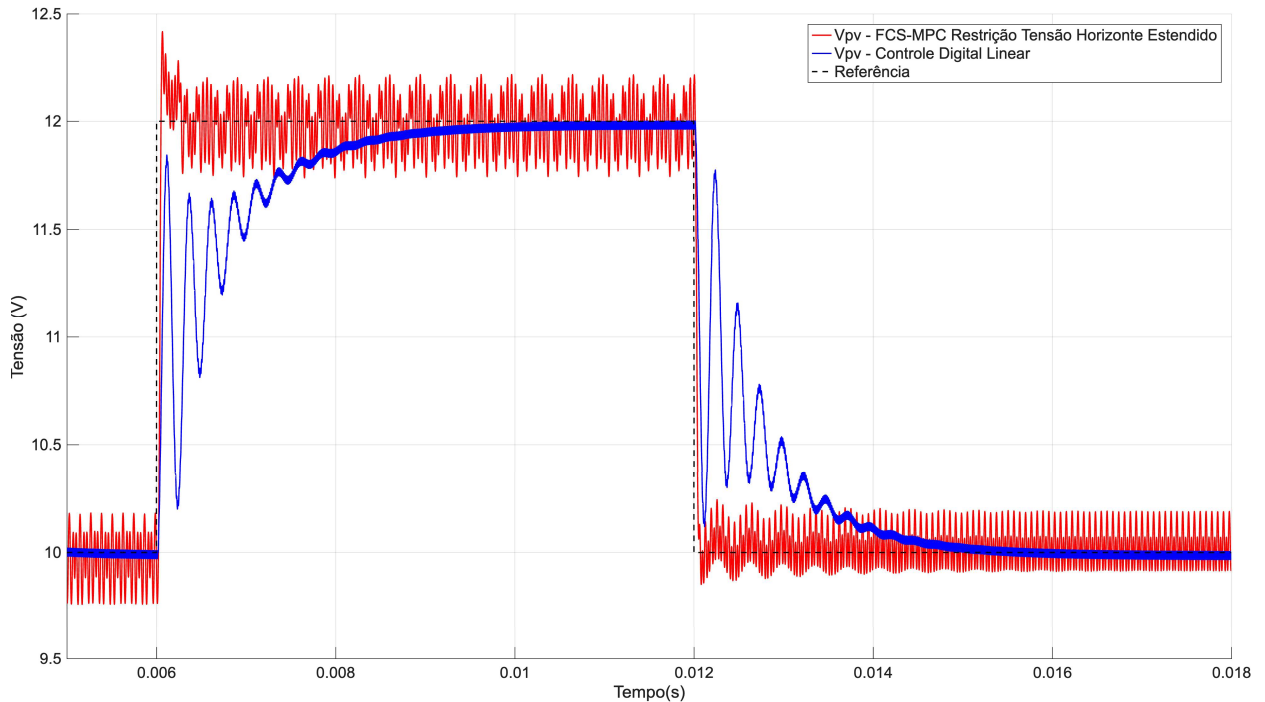
| Função Custo | Referência | Mp(%) | t_s (μ s) | ΔV |
|-----------------------|----------------------|-------|------------------|------------|
| Erro Quadrático | 10 \rightarrow 12V | 7,5 | 980,05 | 0,95 |
| Erro Quadrático | 12 \rightarrow 10V | 12,0 | 948,8 | 0,93 |
| Erro Quadrático | 10 \rightarrow 8V | 9,3 | 988,20 | 1,06 |
| Erro Quadrático | 8 \rightarrow 10V | 12,0 | 984,35 | 0,93 |
| Restrição Condicional | 10 \rightarrow 12V | 3,8 | 989,32 | 0,93 |
| Restrição Condicional | 12 \rightarrow 10V | 5,4 | 913,6 | 0,95 |
| Restrição Condicional | 10 \rightarrow 8V | 5,1 | 895,3 | 0,73 |
| Restrição Condicional | 8 \rightarrow 10V | 7,4 | 764,82 | 0,70 |
| Restrição de Tensão | 10 \rightarrow 12V | 2,3 | 522,27 | 0,54 |
| Restrição de Tensão | 12 \rightarrow 10V | 2,9 | 1000,05 | 0,45 |
| Restrição de Tensão | 10 \rightarrow 8V | 3,4 | 862,45 | 0,50 |
| Restrição de Tensão | 8 \rightarrow 10V | 3,7 | 885,8 | 0,32 |

Fonte: Elaborado pelo Autor (2025).

4.5 Comparação com o Controle Linear

Finalmente, tendo sido estabelecida a função custo que apresenta o melhor desempenho para o sistema, é importante realizar a comparação da solução proposta com o controlador linear digital projetado na subseção 3.1.7. A Figura 39 apresenta um comparativo da resposta dinâmica em controle de malha fechada para a tensão de saída do painel fotovoltaico (v_{pv}). Analisando o gráfico é notável que o controle linear é consideravelmente mais lento e apresenta um transitório mais oscilatório, resultado de uma necessidade de conciliar a rapidez do controle com a sua estabilidade.

Figura 39 – Gráfico de comparação FCS-MPC e controle linear



Fonte: Elaborado pelo Autor (2025).

A Tabela 11 apresenta alguns índices de desempenho para o controlador linear. No entanto, a comparação de tempo de assentamento t_s , com o critério utilizado para as soluções FCS-MPC, passa a não fazer justiça à disparidade entre o controle com restrição de tensão e horizonte estendido e o compensador de dois polos e dois zeros. Utilizando o critério de 2% para o controlador linear, chega-se a um t_s aproximado de 1,5ms, o que é rápido, mas frente ao MPC passa a ser considerado bem mais demorado. Enquanto isso, o controle preditivo por pouco não mantém o sinal dentro da faixa 2% em regime permanente, em contrapartida a frequência de chaveamento do MPC acaba sendo mais baixa apesar do algoritmo operar em 200kHz.

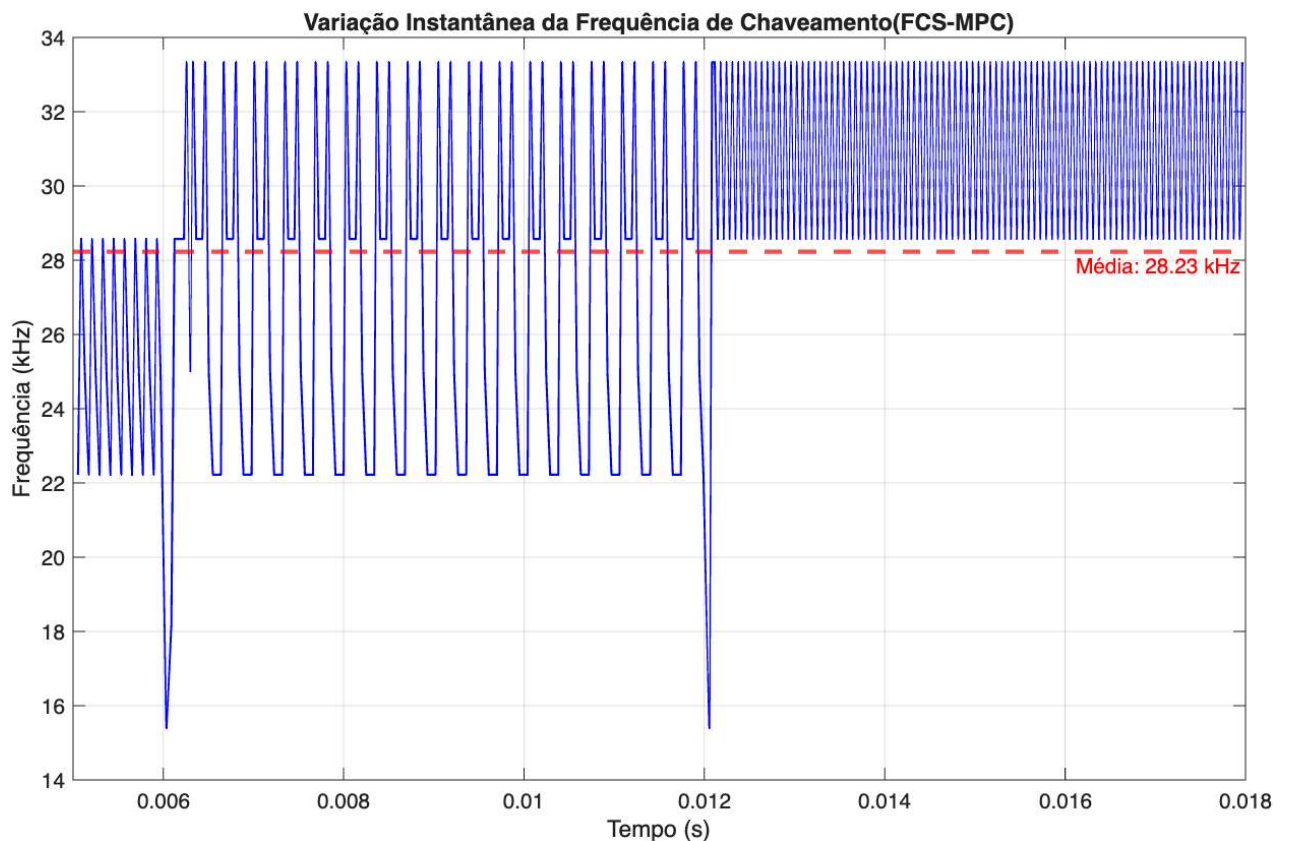
Para o ensaio realizado na Figura 39, o chaveamento ocorre a uma frequência média em torno de 28kHz como apresentado na Figura 40. Isto representa quase um terço da frequência de chaveamento fixa do controle linear ($f_s = 80\text{kHz}$), desta forma, as perdas por comutação podem ser reduzidas em cenários práticos. Nota-se ainda que em o controlador linear em média não garante erro nulo para a referência de 12V, com a tensão variando entre 12V e 11,96V e a média ficando um pouco a baixo da referência em torno de 11,98V.

Tabela 11 – Índices de desempenho para o controlador linear.

| Referência | $Mp_{(\%)}$ | $t_{s(2\%)} (\mu s)$ | ΔV_{pv} |
|------------|-------------|----------------------|-----------------|
| 10 → 12V | 0 | 1553 | 0,0369 |
| 12 → 10V | 0 | 1553 | 0,03759 |

Fonte: Elaborado pelo Autor (2025).

Figura 40 – Frequência de chaveamento do FCS-MPC ao longo da simulação.



Fonte: Elaborado pelo Autor (2025)

Um fator, no entanto, no qual o controlador linear se destaca é a ondulação em regime, que, como evidenciado na Tabela 11, gira em torno de 4mV. Neste sentido, o controle linear supera por uma margem considerável o controle FCS-MPC proposto, mesmo após a adição de restrições.

Para evidenciar a problemática gerada por uma maior ondulação em regime, foi feito um outro ensaio com esses dois controladores, desta vez aplicando apenas um degrau variando a tensão de referência de 10 a 12V e mantendo a simulação por 10ms em regime permanente. Na Tabela 12, estão expostos, para os dois

controladores em análise, os quatro critérios de desempenho de integral do erro previamente utilizados.

Tabela 12 – Comparação Parâmetros de desempenho – Integral do Erro

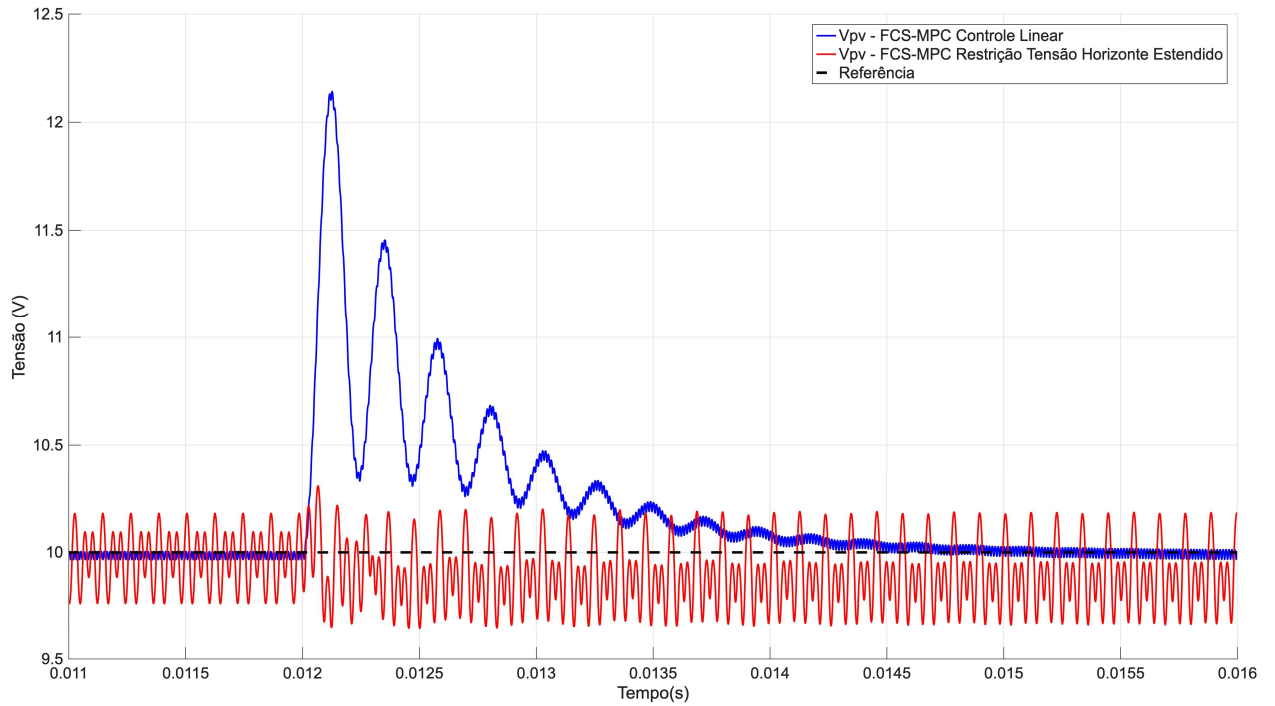
| Controlador | IAE | ISE | ITAE | ITSE |
|-----------------------------|----------------------|----------------------|----------------------|----------------------|
| Controlador Linear Digital | $5,24 \cdot 10^{-4}$ | $8,34 \cdot 10^{-5}$ | $1,36 \cdot 10^{-6}$ | $6,74 \cdot 10^{-8}$ |
| FCS-MPC Restrição de Tensão | $1,24 \cdot 10^{-3}$ | $1,91 \cdot 10^{-4}$ | $6,81 \cdot 10^{-6}$ | $1,05 \cdot 10^{-6}$ |

Fonte: Elaborado pelo Autor (2025).

A partir destes resultados, torna-se evidente que a longo prazo a ondulação leva o controle FCS-MPC a ter um desempenho pior que o controle linear segundo as métricas de integral do erro, ainda que a sua dinâmica transitória seja consideravelmente mais rápida.

Para finalizar, foi feita uma análise da resposta a perturbações ao sistema, especificamente aumentando em 25% a tensão de saída do conversor V_o e o resultado gráfico é ilustrado na Figura 41. Fica evidente que este é um dos pontos fortes do controle preditivo baseado em modelo, que se adequa à situação nova de forma rápida e prevê a possibilidade de sobressinal e penaliza estes cenários através da função custo. Por outro lado, esse é um dos pontos fracos do compensador de dois polos e dois zeros, visto que ele foi projetado a partir de um modelo linearizado em torno de um ponto de operação específico, à medida que se distância um pouco dessa referência, o controle passa por adversidades.

Figura 41 – Comparativo reação dos controles a perturbações.



Fonte: Elaborado pelo Autor (2025)

4.6 Ensaio com Aumento da Frequência de Execução

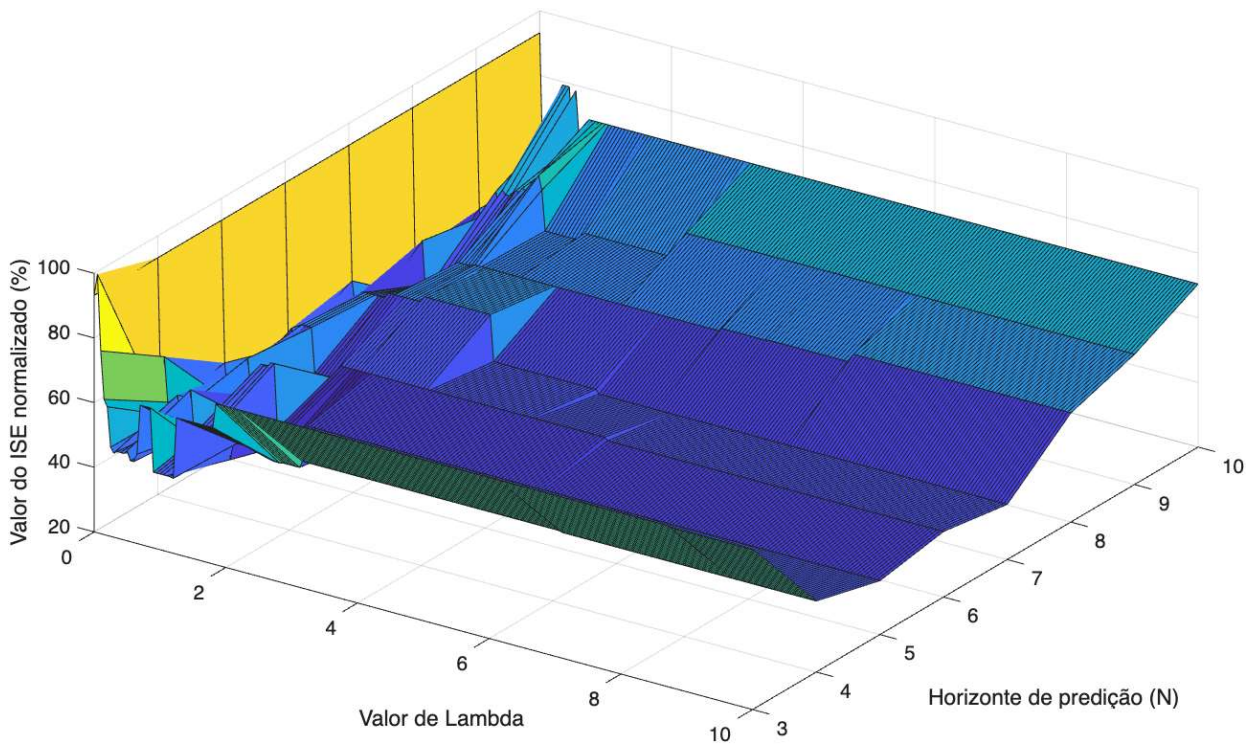
Como o FCS-MPC apresentou problemas em relação à ondulação em regime permanente. É relevante que se busque alternativas para a melhorar o desempenho do controle neste sentido. Uma das possibilidades seria aumentar os níveis de tensão de operação do circuito, o que muito provavelmente faria com que a ondulação relativa à tensão de referência diminuísse percentualmente. No entanto, como explanado anteriormente, os níveis de tensão das simulações presentes neste estudo são limitados em função da planta didática desenvolvida por Melo (2023), com um intuito de que se possa utilizar esta pesquisa como base para uma aplicação experimental futura. Deste modo, não se torna interessante alterar os níveis de tensão e corrente do conversor.

Portanto, resta uma outra possível solução que seria aumentar a frequência de execução do algoritmo MPC, f_{MPC} , com o intuito de aumentar também a frequência de chaveamento (f_s) e por consequência diminuir a ondulação. Para evidenciar que o aumento da frequência traria este benefício, escolheu-se um f_{MPC} de 300kHz.

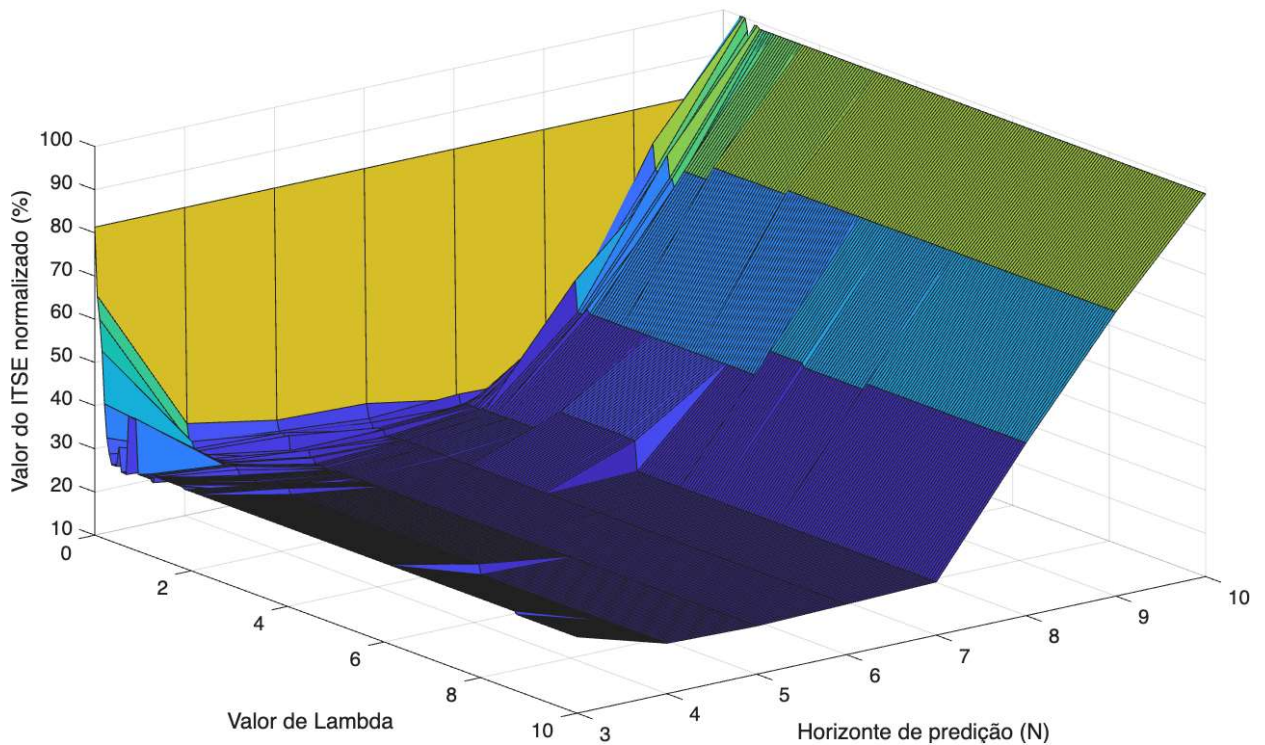
Sendo o último ensaio deste estudo, optou-se por realizá-lo apenas com a proposta do FCS-MPC que apresentou o melhor desempenho, o MPC com restrição de tensão e horizonte de predição estendido. Portanto, foi realizada a mesma análise de varredura presente nas equações (4.4) e (4.5).

Como os resultados de IAE, ITAE e ITSE se mostraram muito semelhantes visualmente, apenas os gráficos de ISE e ITSE serão evidenciados, nas Figuras 42 e 43 respectivamente, como forma de sintetizar a análise. Assim como nas outras análises as combinações de parâmetros que minimizaram as métricas de integral do erro se encontram reunidas na Tabela 13, com a variação relativa para a métricas que não são minimizadas por determinada combinação.

Figura 42 – ISE x N x λ - $f_{MPC} = 300kHz$.



Fonte: Elaborado pelo Autor (2025).

Figura 43 – ITSE x N x λ - $f_{MPC} = 300kHz$.

Fonte: Elaborado pelo Autor (2025).

Tabela 13 — Comparação das soluções - $f_{MPC} = 300kHz$.

| Combinação | λ | N | IAE_{rel} | $ITAE_{rel}$ | ISE_{rel} | $ITSE_{rel}$ |
|------------|-------------|---|-------------|--------------|-------------|--------------|
| 1 | 1,05 – 1,25 | 5 | 0% | 6,2% | 0% | 7,68% |
| 2 | 0,35 | 5 | 7,32% | 0% | 29,7% | 0% |

Fonte: Elaborado pelo Autor (2025).

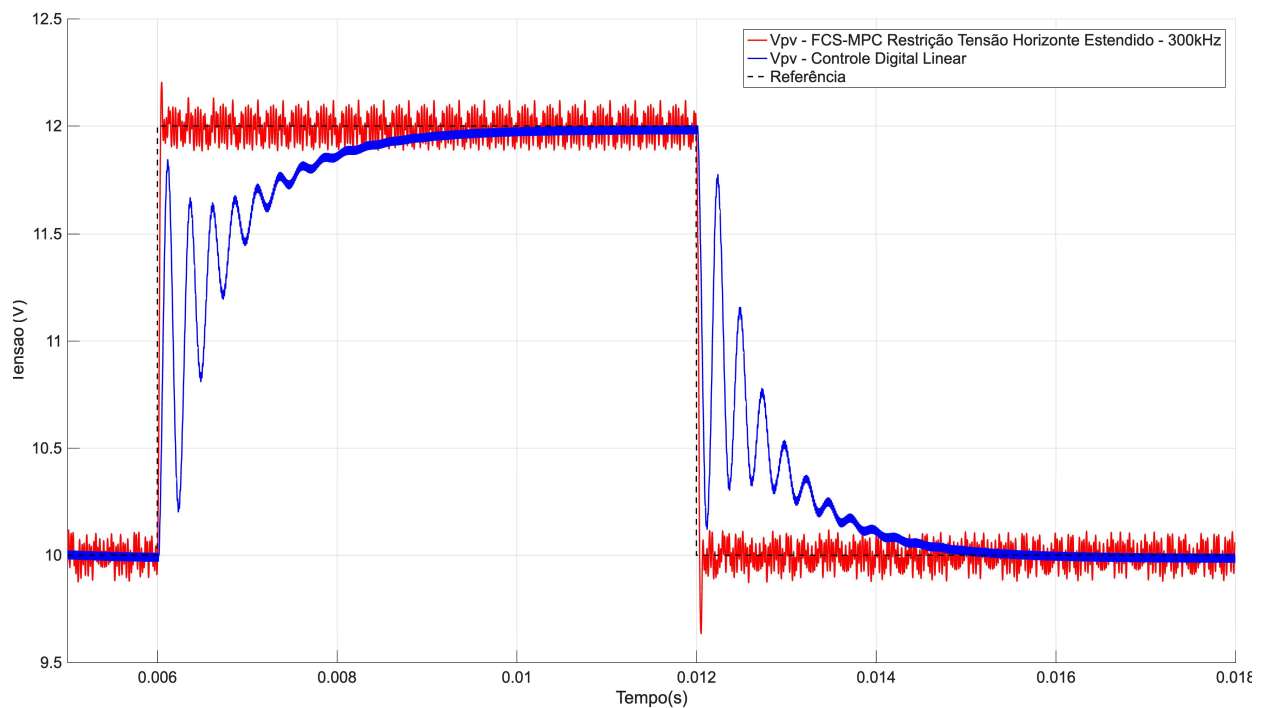
A partir dos resultados da Tabela 13, optou-se pela combinação 1, que na média mostrou ser mais estável para todas as métricas. Ainda fez-se um ensaio com ambas as combinações e a primeira apresentou um desempenho ligeiramente melhor nos critérios de ondulação em regime, sobressinal e tempo de assentamento.

Tendo sido escolhida a combinação 1, fez-se necessário determinar o valor de λ . Baseado na ideia que o termo de previsão longa ($k + N_1$) da função custo leva o sinal a ficar mais próximo da referência, optou-se pelo maior valor da faixa de possibilidades da combinação 1. Portanto, com $\lambda = 1,25$ e $N_1 = 5$, a uma frequência de 300kHz, o algoritmo foi executado e o seu resultado dinâmico se encontra na Figura 44, em comparação com controle linear digital. Nota-se de antemão que houve uma redução

da ondulação, de modo que este algoritmo pôde ser avaliado pelo critério mais comum de tempo de assentamento ($t_{s(2\%)}$). Algumas das métricas de avaliação do controle estão expostas na Tabela 14.

Destaca-se a redução da ondulação pela metade em relação ao melhor algoritmo executado a 200kHz, além do tempo de assentamento consideravelmente mais veloz que o controle linear.

Figura 44 – Gráfico de comparação FCS-MPC com $f_{MPC} = 300kHz$ e controle linear.



Fonte: Elaborado pelo Autor (2025).

Tabela 14 – Índices de desempenho para o FCS-MPC (300kHz) e o controlador linear.

| Controlador | Referência | Mp (%) | $t_{s(2\%)} (\mu s)$ | ΔV_{pv} |
|----------------------------|------------|--------|----------------------|-----------------|
| Controlador Linear Digital | 10 → 12V | 0 | 1553 | 0,0369 |
| Controlador Linear Digital | 12 → 10V | 0 | 1553 | 0,03759 |
| FCS-MPC – 300kHz | 10 → 12V | 2,0% | 54,33 | 0,26 |
| FCS-MPC – 300kHz | 12 → 10V | 3,2% | 67,00 | 0,21 |

Fonte: Elaborado pelo Autor (2025).

Seguindo o mesmo procedimento realizado para obter os resultados da Tabela 12, encontram-se as métricas de integral do erro para o algoritmo FCS-MPC com a frequência aumentada, presentes na Tabela 15, a partir da qual fica evidente a melhora do desempenho do algoritmo após o aumento da frequência de amostragem.

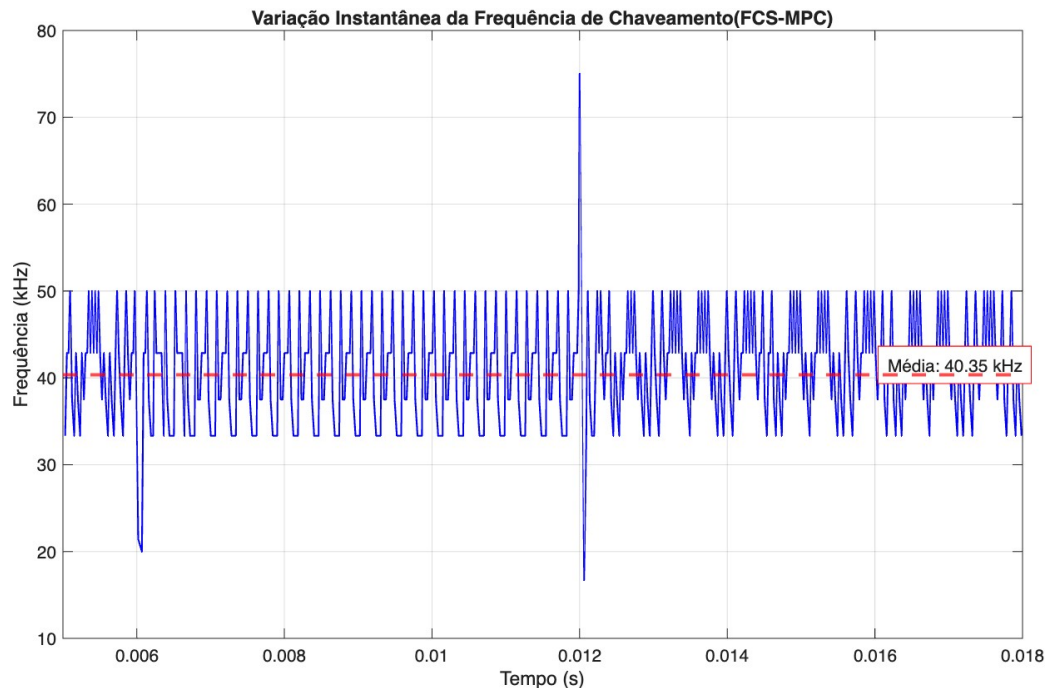
Tabela 15 – Comparação Parâmetros de desempenho – Integral do Erro - $f_{MPC} = 300kHz$.

| Controlador | IAE | ISE | ITAE | ITSE |
|----------------------------|----------------------|----------------------|----------------------|----------------------|
| Controlador Linear Digital | $5,24 \cdot 10^{-4}$ | $8,34 \cdot 10^{-5}$ | $1,36 \cdot 10^{-6}$ | $6,74 \cdot 10^{-8}$ |
| FCS-MPC – 200kHz | $1,24 \cdot 10^{-3}$ | $1,91 \cdot 10^{-4}$ | $6,81 \cdot 10^{-6}$ | $1,05 \cdot 10^{-6}$ |
| FCS-MPC – 300kHz | $5,71 \cdot 10^{-4}$ | $4,12 \cdot 10^{-5}$ | $3,14 \cdot 10^{-6}$ | $2,26 \cdot 10^{-7}$ |

Fonte: Elaborado pelo Autor (2025).

Por fim, ressalta-se, a partir do gráfico Figura 45, que o aumento da frequência de amostragem realmente se traduziu em um aumento na frequência de chaveamento média durante a execução do código, passando de 28kHz para 40kHz, ainda assim consideravelmente mais baixa que a frequência de chaveamento do controle linear.

Figura 45 – Frequência de chaveamento do FCS-MPC ao longo da simulação. - $f_{MPC} = 300kHz$.



Fonte: Elaborado pelo Autor (2025).

5 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Este Trabalho de Conclusão de Curso buscou investigar a aplicação do controle preditivo baseado em modelo, especificamente na forma do FCS-MPC, aplicado a um conversor *boost* fotovoltaico. Para isso, estruturou-se uma abordagem progressiva que abrangeu desde a construção de uma base teórica sólida até o desenvolvimento de uma modelagem matemática adequada e a avaliação quantitativa dos controladores propostos.

Inicialmente, foram apresentados os conceitos mais gerais da pesquisa para facilitar a compreensão da proposta deste estudo. Passou-se por uma introdução breve aos conversores de potência, dando enfoque àqueles do tipo CC-CC não isolados. Em seguida, comentou-se sobre a possibilidade de se aplicar o controle do *boost* em sistemas fotovoltaicos. Por fim, foram apresentados os mais diversos métodos de controle, finalizando com o FCS-MPC.

A partir do estudo do conversor, o trabalho foi bem sucedido em encontrar a modelagem do *boost* proposto. O sistema foi modelado tanto no espaço de estados, quanto via um modelo médio por pequenos sinais, sendo um passo importante para a aplicação dos controles FCS-MPC e para o controlador linear, utilizado como referência de avaliação dos controladores apresentados.

Por meio de simulações em *software*, os resultados deste trabalho levam à conclusão que o controle FCS-MPC pode ser uma alternativa viável pra sistemas de eletrônica de potência, mas com certas ressalvas, apesar do desempenho excelente em transitórios, o resultado em regime permanente deixou a desejar, sobretudo para o caso de níveis de tensão abaixo do usual para aplicações de potência ou aplicações fotovoltaicas. Faz-se necessário destacar também a necessidade de uma taxa de amostragem alta para a execução exitosa do algoritmo. Como evidenciado pelo último ensaio, o aumento da frequência de execução traz uma melhora significativa em relação à ondulação em regime.

Além disso, os resultados reforçam a relevância da adição de termos restritivos e a extensão do horizonte de predição para que se obtenha um controle mais eficiente dinamicamente e com menos ondulação em regime permanente.

Ademais, deu-se um passo à frente no caminho da aplicação dos algoritmos MPC com *hardware* dedicado e controladores reais, através da utilização das *S-functions*, permitindo encapsular códigos em C no ambiente Simulink. Assim, este trabalho contribui modestamente para o avanço do estudo e aplicação do controle preditivo em conversores estáticos, oferecendo uma base para futuras pesquisas e desenvolvimentos na área, como comentado a seguir.

5.1 Trabalhos Futuros

O presente trabalho demonstrou, em simulação, a viabilidade da estratégia de controle FCS-MPC aplicada ao conversor *boost* para sistemas fotovoltaicos. A partir dos resultados obtidos, identificam-se diversas oportunidades para a continuidade e aprofundamento desta pesquisa.

- **Validação Experimental:** Um passo fundamental para consolidar os achados deste TCC é a validação experimental da estratégia de controle. Propõe-se a implementação do algoritmo FCS-MPC apresentado em uma plataforma de *hardware* em tempo real, como um DSP. Para averiguar o desempenho controle na prática e, para além disso, realizar uma análise crítica do custo computacional do processamento do algoritmo.
- **Aplicação do FCS-MPC em topologias de Fase Não-Mínima:** Como é o caso do *boost* convencional em que se deseja controlar a tensão de saída do conversor, a Fase Não-Mínima impõe um desafio tanto para os controladores lineares quanto para controles preditivos. Uma oportunidade de pesquisa surge em torno dessa adversidade, com formas de contorná-la seja pela extensão do horizonte preditivo ou uma restrição condicional.
- **Extensão do estudo para a aplicação em conjuntos com MPPT:** Por fim, ressalta-se o ensejo por aplicar o controle estudado neste trabalho aliado ao algoritmo MPPT e possivelmente ligado a rede elétrica através de um inversor. O controle, portanto, atuaria no conjunto completo de um sistema FV, sendo um desafio de grande valor para aplicações reais.

REFERÊNCIAS

- ABOLHASANI et al. A comparison between buck and boost topologies as module integrated converters to mitigate partial shading effects on PV arrays. **7th Power Electronics and Drive Systems Technologies Conference (PEDSTC)**., Tehran, fevereiro 2016. 367-372.
- AL-WESABI, I. et al. Direct Sliding Mode Control for Dynamic Instabilities in DC-Link Voltage of Standalone Photovoltaic Systems with a Small Capacitor. **Electronics**, 1 Janeiro 2022. 1-23.
- ANA; MDR. **Relatório Conjuntura dos Recursos Hídricos**. AGÊNCIA NACIONAL DE ÁGUAS E SANEAMENTO BÁSICO; MINISTÉRIO DA INTEGRAÇÃO E DO DESENVOLVIMENTO REGIONAL. Brasília. 2024.
- AZADI et al. FCS-Model Predictive Control of a Quadratic Buck Converter for more Efficient Data Centers. **17th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)**, Tallinn, 2023. 1-5.
- BORDONS, CARLOS; MONTERO, CARLOS. Basic Principles of MPC for Power Converters. **IEEE industrial electronics magazine** , 21 Setembro 2015. 31-43.
- BOYLESTAD, Robert L. Circuitos Magnéticos. In: BOYLESTAD, Robert L. **Introdução à Análise de Circuitos**. 12^a. ed. [S.l.]: Pearson Education Inc., 2012. Cap. 12, p. 429-451.
- CHENG, Long et al. Model Predictive Control for DC–DC Boost Converters With Reduced-Prediction Horizon and Constant Switching Frequency. **IEEE Transactions on Power Electronics**, 2018. 9064-9075.
- COSTA, Lucas. Aplicação de Controle Preditivo Baseado em Modelo com Restrição de Função de Custo em Conversores Chaveados. **Trabalho de Conclusão de Curso - Universidade Federal de Pernambuco (UFPE)**, Recife, 2025.
- CUNHA, R. B. A. et al. Constant switching frequency finite control set model predictive control. **Solar Energy**, 2019. 57-66.
- CUNHA, Rafael B. A. Controle Preditivo do tipo finite control set aplicado ao conversor elevador de um sistema fotovoltaico com operação em frequência constante. **Tese de Doutorado - Escola Politécnica da Universidade de São Paulo**, São Paulo, 2021. 100.
- DE KEYSER, R.; BONILLA, J.; IONESCU, C. A Comparative Study of Several Control. **IEEE 10th Int Conf on Optimisation of Electrical and Electronic Equipment OPTIM**, Brasov, 2006. 71-78.
- DORF, Richard C.; BISHOP, Robert H. **Modern Control Systems**. 13^a. ed.
- EPE. **Plano Decenal de Expansão de Energia 2034**. Empresa de Pesquisa Energética. Brasília, p. 579. 2024.
- EPE; MME. **Balanço Energético Nacional (BEN) - Relatório Síntese 2025 - Ano base 2024**. Empresa de Pesquisa Energética; Ministério de Minas e Energia. [S.l.], p. 74. 2025.
- ERICKSON, Robert; MAKSIMOVIC, Dragan. **Fundamentals of Power Electronics**. 2^a. ed.
- ESCOBAR, G. et al. An experimental comparison of several nonlinear controllers for power converters. **IEEE Control Systems Magazine**, 1 Fevereiro 1999. 66-82.
- FOROUZESH, Mojtaba et al. Step-Up DC–DC Converters: A Comprehensive Review of Voltage-Boosting Techniques, Topologies, and Applications. **IEEE Transactions on Power Electronics**, Dezembro 2017. 9143-9178.
- GIRALDO, Néstor I. M. **Modelagem e Controle de Conversor CC-CC Cuk no Modo de Condução Contínua (MCC)**. Universidade Federal de Pernambuco. Recife. 2021.

HAQUE et al. Small signal modeling and control of isolated three port DC-DC Converter for PV-battery system. **IEEE Region 10 Humanitarian Technology Conference**, Dhaka, 2017. 263-266.

HARZIG, Thibaut; GRAINGER, Brandon. Time-Optimal finite control se model predictive control of non-isolated DC-DC Converters. **IET Electric Power Applications**, 20 Junho 2024. 1626-1637.

HOLTZ, J.; STADTFELD, S. Field-Oriented Control by Forced Motor Currents in a Voltage Fed Inverter Drive. **IFAC Control in Power Electronics and Electrical Drives**, Lausanne, 1983. 103-110.

IEA. **Global Energy Review 2025**. International Energy Agency. [S.l.], p. 43. 2025.

ISLAM, Omar K. et al. Comprehensive Comparison Between Boost and Buck-Boost Converters in Solar MPPT With ANN. **Emerging Technology in Computing Communication and Electronics (ETCCE)**, 2020. 1-6.

JOOSHAKI, MOHAMMAD et al. Reliability-Oriented Electricity Distribution System Switch and Tie Line Optimization. **IEEE Access - IEEE POWER & ENERGY SOCIETY SECTION**, 28 julho 2020. 130967-130978.

KARAFOTIS, Panagiotis A.; EVANGELOPOULOS, Vasileios A.; GEORGILAKIS, Pavlos S. Reliability-Oriented Reconfiguration of Power Distribution Systems Considering Load and RES Production Scenarios. **IEEE Transactions on Power Delivery**, 23 fevereiro 2022. 4668-4678.

KARAMANAKOS, Petros; GEYER, Tobias; MANIAS, Stefanos. Direct Voltage Control of DC-DC Boost Converters Using Enumeration-Based Model Predictive Control. **IEEE Transactions on Power Electronics**, fevereiro 2014.

KJAER, S. B.; PEDERSEN, J.K.; BLAABJERG, F. A review of single-phase grid-connected inverters for photovoltaic modules. **EEE Transactions on Industry Applications**, Outubro 2005. 1292-1306.

KOURO, Samir et al. Model Predictive Control — A Simple and Powerful Method to Control Power Converters. **IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS**, Junho 2009. 1826-1838.

KOURO, Samir et al. Grid-Connected Photovoltaic Systems: An Overview of Recent Research and Emerging PV Converter Technology. **IEEE Industrial Electronics Magazine**, Março 2015. 47-61.

LAHOTI, G.; SHAH, U.; KADAM, P. Robust control of boost converter for flexible operation in PV based systems. **2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)**, Nagercoil, 2016. 345-349.

LIANG, Yan et al. Model Predictive Control for Interleaved DC-DC Boost Converter Based on Kalman Compensation. **2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)**, Shenzhen, 2018. 1-5.

MATHWORKS. **Writing S-Functions**. 5ª. ed.

MELO, Iago. Desenvolvimento de kit didático para controle de corrente de um conversor boost PFC. **Trabalho de Conclusão de Curso - Universidade Federal de Pernambuco**, Recife, setembro 2023.

MOHAN, Ned; UNDELAND, Tore; ROBBINS, William. **Power Electronics: Converters, Applications and Design**. 3ª. ed.

OGATA, Katsuhiko. **Discrete-Time Control Systems**. 2ª. ed.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5ª. ed.

OLABI, A.; ABDELKAREEM, M. A. S. Renewable energy and climate change. **Renewable and Sustainable Energy Reviews**, Abril 2022.

ONS. **PAR/PEL 2024 - Plano da Operação Elétrica de Médio Prazo do SIN - Ciclo 2025 - 2029**. Operador Nacional do Sistema. [S.l.], p. 74. 2024.

PERTENCE JR., Antonio. Aplicações não-lineares com AOPs. In: PERTENCE JR., Antonio **Eletrônica Analógica - Amplificadores Operacionais e Filtros Ativos**. 6ª. ed. Porto Alegre: Bookman Companhia Editora, 2003. Cap. 5, p. 99-144.

PO, Li et al. Composite adaptive model predictive control for DC–DC boost converters. **IET Power Electronics**, 2018. 1706-1717.

PREINDL, Matthias; BOLOGNANI, Silverio. Comparison of direct and PWM model predictive control for power electronic and drive systems. **2013 Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)**, Long Beach, 27 Maio 2013. 2526-2533.

RESTREPO, C, et al. A Noninverting Buck–Boost DC–DC Switching Converter With High Efficiency and Wide Bandwidth. **IEEE Transactions on Power Electronics**, Setembro 2011. 2490-2503.

RIBEIRO JR., Marcelo A. Controle Preditivo para Conversor Buck Aplicado a um Sistema Fotovoltaico. **Trabalho de Conclusão de Curso - Universidade Federal do Espírito Santo**, Vitória, Julho 2023.

RIBEIRO, Adeilson S. B. et al. ANALYSIS AND DESIGN OF PROPORTIONAL-RESONANT CONTROLLERS BASED ON POLE PLACEMENT APPROACH. **2018 Simposio Brasileiro de Sistemas Eletricos (SBSE)**, Niterói, maio 2018. 1-6.

RICHALET, J. et al. Model predictive heuristic control: Application to industrial processes. **AUTOMATICA**, 1978. 413-428.

RODRIGUES, Gleice. **Controle Preditivo Aplicado a Conversores Estáticos de Potência**. Dissertação de Mestrado - Universidade Federal da Paraíba. João Pessoa, p. 111. 2020.

RODRIGUEZ, J et al. State of the Art of Finite Control Set Model Predictive Control in Power Electronics. **IEEE Transactions on Industrial Informatics**, 2012. 1003-1016.

RODRIGUEZ, J.; LAI, Jih-Sheng; PENG, Fang Z. Multilevel inverters: a survey of topologies, controls, and applications. **IEEE Transactions on Industrial Electronics**, Agosto 2002. 724-738.

RODRIGUEZ, José; CORTES, Patrício. **Predictive control of power converters and electrical drives**. 1ª. ed.

SANTOS, Cássia et al. Estratégia de Controle para Conversor Boost Fotovoltaico operando nos modos MPPT e LPPT. **Eletrônica de Potência**, Fortaleza, setembro 2020. 326-336.

SAUAIA, Rodrigo. **Energia Solar Fotovoltaica: Panorama, Oportunidades e Desafios**. ABSOLAR - Associação Brasileira de Energia Fotovoltaica. Rio de Janeiro, p. 50. 2018.

SCOLTOCK, Jason; GEYER, Tobias; MADAWALA, Udaya. Model Predictive Direct Current Control for a Grid-Connected Converter: LCL-Filter versus L-Filter. **IEEE International Conference on Industrial Technology (ICIT)**, Cape Town, 2013. 576-581.

SQUAREZI FILHO, Alfeu et al. Predictive Control Applied to a Boost Converter of a Photovoltaic System. **Journal of Control, Automation and Electrical Systems**, 04 Setembro 2021. 393-405.

TANG, Zhogting; YANG, Yongheng; BLAABJBERG, Frede. Power Electronics: The Enabling Technology for Renewable Energy. **CSEE Journal of Power and Energy Systems**, Janeiro 2022. 39-52.

UMANS, D. S. Máquinas Elétricas de Fitzgerald e Kingsley. In: UMANS; D., Stephen **Máquinas Elétricas de Fitzgerald e Kingsley**. 7ª. ed. [S.l.]: AMGH Editora LTDA., 2014. Cap. 1, p. 1-62.

VASQUEZ, Sergio et al. Model Predictive Control for Single-Phase NPC Converters Based on Optimal Switching Sequences. **IEEE Transactions on Industrial Electronics**, Dezembro 2016. 7533-7541.

VAZQUEZ, Sergio et al. Model Predictive Control: A Review of Its Applications in Power Electronics. **IEEE Industrial Electronics Magazine**, 1 Março 2014. 16-31.

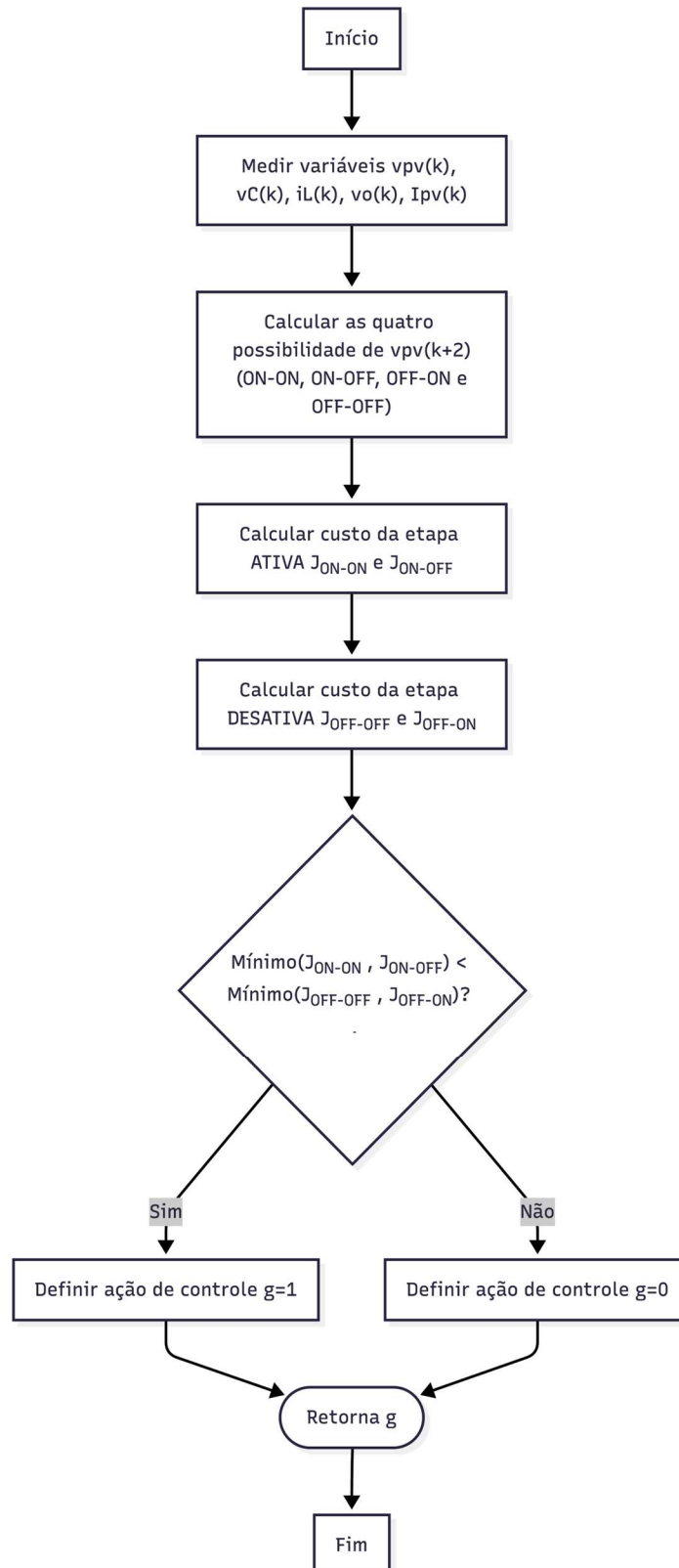
YOUNG, H.; RODRIGUEZ, J. Comparison of finite-control-set model predictive control versus a SVM-based linear controller. **15th European Conference on Power Electronics and Applications (EPE)**, Lille, 2013. 1-8.

ZONETTI, Daniele et al. PID passivity-based droop control of power converters: Large-signal stability, robustness and performance. **International Journal of Robust and Nonlinear Control**, 26 novembro 2021. 1769-1795.

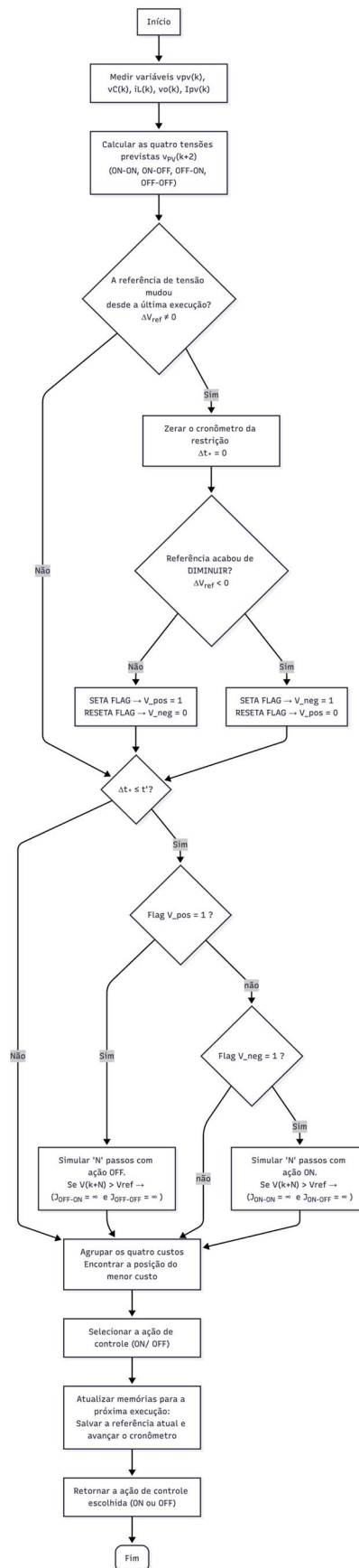
ZUCUNI, J. et al. Comparison of model predictive current controllers for grid-tied inverters. **Proceedings of the 11th Seminar on Power Electronics and Control (SEPOC 2018)**, Santa Maria, 2018. 1-5.

APÊNDICES

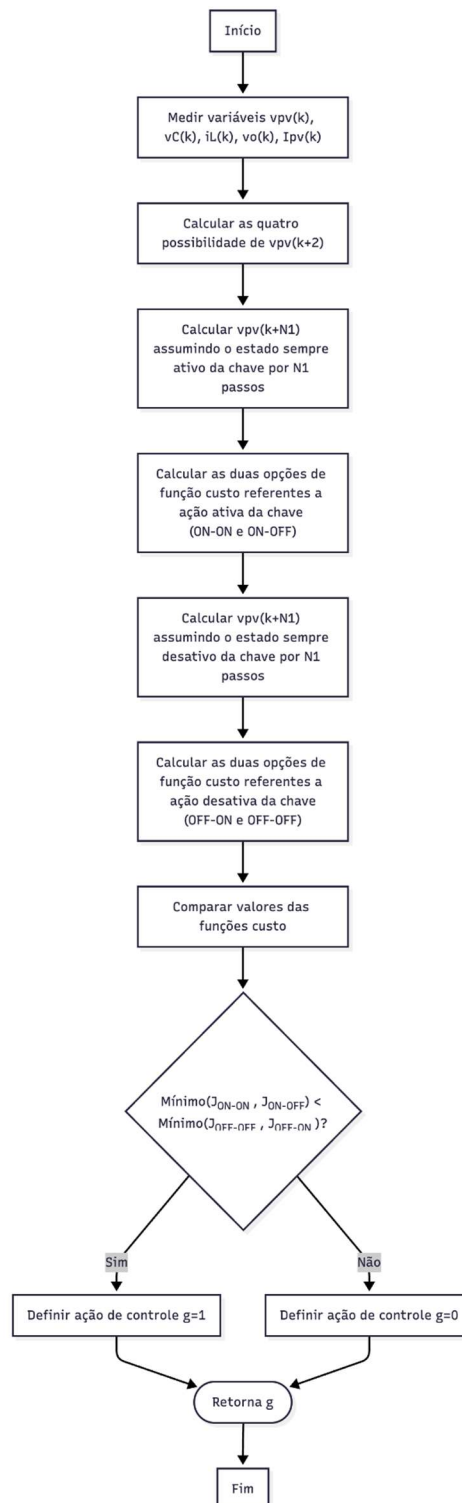
APÊNDICE A – FLUXOGRAMA FCS-MPC ERRO-QUADRÁTICO



APÊNDICE B – FLUXOGRAMA FCS-MPC COM RESTRIÇÃO CONDICIONAL



APÊNDICE C – FLUXOGRAMA FCS-MPC COM RESTRIÇÃO DE TENSÃO E HORIZONTE DE PREDIÇÃO ESTENDIDO



APÊNDICE D – CÓDIGO S-FUNCTION FCS-MPC ERRO QUADRÁTICO

```

/*-----*/
/* Filename: BOOST_FCS_MPC_ERRO_QUAD.c */
/* Description: Controle FCS-MPC para Conversor Boost */
/* Função Custo: Erro Quadrático */
/* Autor: Yuri Giglio de Azevedo */
/* Adaptado de Código em Matlab para o Conversor Buck de */
/* Autoria de Lucas Gabriel Pedrosa de Lima Costa */
/*-----*/
#define S_FUNCTION_NAME BOOST_FCS_MPC_ERRO_QUAD
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include "math.h"

/* --- Definições do Bloco --- */

/* 5 Entradas: [Vpv_ref, vC, iL, Vo, Ipv] */
#define NINPUTS 5

/* 1 Saída: [g] (sinal de controle 0 ou 1) */
#define NOUTPUTS 1

/* 5 Parâmetros: [f_MPC, RL, Rc, L, C] */
#define NPARAMS 5

/* Índices dos Parâmetros (para leitura) */
#define F_MPC_INDEX 0
#define RL_INDEX 1
#define RC_INDEX 2
#define L_INDEX 3
#define C_INDEX 4

/*=====
 * S-function methods *
=====*/

/* Function: mdlInitializeSizes ===== */
static void mdlInitializeSizes(SimStruct *S) {
    ssSetNumSFcnParams(S, NPARAMS);

    // Checagem de erro dos parâmetros
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        ssSetErrorStatus(S, "Numero incorreto de parametros passados para a S-Function FCS_MPC_BOOST. Esperado: 5.");
        return;
    }

    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, 0);

    /* Configuração da Porta de Entrada */
    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, NINPUTS);
    ssSetInputPortDirectFeedThrough(S, 0, 1);

    /* Configuração da Porta de Saída */
    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, NOUTPUTS);

    ssSetNumSampleTimes(S, 1);

    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

/* Function: mdlInitializeSampleTimes =====*/
static void mdlInitializeSampleTimes(SimStruct *S) {
    // Lê o parâmetro f_MPC para definir o tempo de amostragem do bloco
    real_T f_MPC = mxGetScalar(ssGetSFcnParam(S, F_MPC_INDEX));
    real_T Ts = 1.0 / f_MPC;

    ssSetSampleTime(S, 0, Ts);
    ssSetOffsetTime(S, 0, 0.0);
}

/* Function: mdlOutputs =====*/
static void mdlOutputs(SimStruct *S, int_T tid) {

    // --- 1. Ponteiros de Entrada e Saída ---
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S, 0);
    real_T *y = ssGetOutputPortRealSignal(S, 0); // Ponteiro da Saída y[0]

    // --- 2. Leitura das 5 Entradas (medições atuais k) ---
    real_T Vpv_ref = *uPtrs[0];
    real_T vC = *uPtrs[1];
    real_T iL = *uPtrs[2];
    real_T Vo = *uPtrs[3];
    real_T Ipv = *uPtrs[4];

    // --- 3. Leitura dos 5 Parâmetros Estáticos ---
    real_T f_MPC = mxGetScalar(ssGetSFcnParam(S, F_MPC_INDEX));
    real_T RL = mxGetScalar(ssGetSFcnParam(S, RL_INDEX));
    real_T Rc = mxGetScalar(ssGetSFcnParam(S, RC_INDEX));
    real_T L = mxGetScalar(ssGetSFcnParam(S, L_INDEX));
    real_T C = mxGetScalar(ssGetSFcnParam(S, C_INDEX));

```

```

// --- 4. Lógica de Função Custo do MPC ---

real_T Ts = 1.0 / f_MPC;

// Variáveis intermediárias para a predição
real_T vC_A, iL_A, vC_D, iL_D;
real_T vC_AA, vC_AD, vC_DA, vC_DD;
real_T iL_AA, iL_AD, iL_DA, iL_DD;
real_T Vpv_AA, Vpv_AD, Vpv_DA, Vpv_DD;
real_T Fc_11, Fc_12, Fc_21, Fc_22;
real_T g_out;

// Termos comuns para otimizar cálculos
real_T term_vC_common = vC - (Ts*iL/C) + (Ts*Ipv/C);
real_T term_iL_dyn_common = (1.0 - Ts/L*(RL + Rc));
real_T term_iL_in_common = (Ts*vC/L) + (Ts*Rc*Ipv/L);

/* --- Predição k+1 (baseado no estado atual k) --- */
vC_A = term_vC_common; // (g=1, Ativa)
iL_A = term_iL_in_common + term_iL_dyn_common * iL;

vC_D = term_vC_common; // (g=0, Desativa) - Equação de vC é idêntica
iL_D = term_iL_in_common + term_iL_dyn_common * iL - (Ts*Vo/L);

/* --- Predição k+2 (árvore de decisão) --- */
// Caminho 1: A->A (Ativa -> Ativa)
vC_AA = vC_A - (Ts*iL_A/C) + (Ts*Ipv/C);
iL_AA = (Ts*vC_A/L) + (Ts*Rc*Ipv/L) + term_iL_dyn_common * iL_A;

// Caminho 2: A->D (Ativa -> Desativa)
vC_AD = vC_A - (Ts*iL_A/C) + (Ts*Ipv/C); // vC_AD é igual a vC_AA
iL_AD = (Ts*vC_A/L) + (Ts*Rc*Ipv/L) + term_iL_dyn_common * iL_A - (Ts*Vo/L);

// Caminho 3: D->A (Desativa -> Ativa)
vC_DA = vC_D - (Ts*iL_D/C) + (Ts*Ipv/C);
iL_DA = (Ts*vC_D/L) + (Ts*Rc*Ipv/L) + term_iL_dyn_common * iL_D;

// Caminho 4: D->D (Desativa -> Desativa)
vC_DD = vC_D - (Ts*iL_D/C) + (Ts*Ipv/C); // vC_DD é igual a vC_DA
iL_DD = (Ts*vC_D/L) + (Ts*Rc*Ipv/L) + term_iL_dyn_common * iL_D - (Ts*Vo/L);

/* --- Saídas Preditas em k+2 --- */
Vpv_AA = vC_AA - Rc*iL_AA + Rc*Ipv;
Vpv_AD = vC_AD - Rc*iL_AD + Rc*Ipv;
Vpv_DA = vC_DA - Rc*iL_DA + Rc*Ipv;
Vpv_DD = vC_DD - Rc*iL_DD + Rc*Ipv;

/* --- Funções Custo --- */
// Calcula os erros quadrados k+2
real_T err_k2_AA = Vpv_ref - Vpv_AA;
real_T err_k2_AD = Vpv_ref - Vpv_AD;
real_T err_k2_DA = Vpv_ref - Vpv_DA;
real_T err_k2_DD = Vpv_ref - Vpv_DD;

// Custo = (erro_k+2)^2
Fc_11 = (err_k2_AA * err_k2_AA);
Fc_12 = (err_k2_AD * err_k2_AD);
Fc_21 = (err_k2_DA * err_k2_DA);
Fc_22 = (err_k2_DD * err_k2_DD);

/* --- Escolha da ação de controle da chave (g(k)) --- */
// 0 vetor de ações g(k) correspondente é [1, 1, 0, 0]

real_T Fc_min = Fc_11; // Assume que o primeiro custo é o mínimo
g_out = 1.0; // Ação correspondente ao Fc_11 é g(k)=1

if (Fc_12 < Fc_min) {
    Fc_min = Fc_12;
    g_out = 1.0; // Ação correspondente ao Fc_12 é g(k)=1
}

if (Fc_21 < Fc_min) {
    Fc_min = Fc_21;
    g_out = 0.0; // Ação correspondente ao Fc_21 é g(k)=0
}

if (Fc_22 < Fc_min) {
    // Fc_min = Fc_22; // Não precisa atualizar Fc_min, apenas g_out
    g_out = 0.0; // Ação correspondente ao Fc_22 é g(k)=0
}

/* --- 5. Atribuição à Saída --- */
y[0] = g_out;
}

/* Function: mdlTerminate ===== */
static void mdlTerminate(SimStruct *S) {

}

/* Boilerplate para compilação MEX/Simulink */
#ifdef MATLAB_MEX_FILE
#include "simulink.c"
#else
#include "cg_sfun.h"
#endif

```

APÊNDICE E – CÓDIGO S-FUNCTION FCS-MPC - RESTRIÇÃO CONDICIONAL

```

/*-----*/
/* Filename: BOOST_MPC_Restritivo_cond.c */
/* Description: S-Function para o algoritmo MPC-FCS (k+2) */
/* COM ROTINA RESTRIÇÃO CONDICIONAL. */
/* Level: S-Function Nível 2 */
/* Autor: Yuri Giglio de Azevedo */
/* Adaptado de código Matlab de Lucas Gabriel Pedrosa de Lima Costa */
/*-----*/
#define S_FUNCTION_NAME BOOST_MPC_Restritivo_cond
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include <math.h>
#include <string.h>
#ifdef MATLAB_MEX_FILE
#include "mex.h"
#endif

/* --- Parâmetros --- */
enum { P_A1 = 0, P_B1, P_C1, P_E1, P_A2, P_B2, P_C2, P_E2, P_F MPC,
       P_T_REF, P_N, NUM_PARAMS }; // 11

/* Variáveis persistentes */
enum { DWORK_VPV_REF_ANT = 0, DWORK_DELTA_T, DWORK_REF_POS, DWORK_REF_NEG,
       NUM_DWORKS }; //

#define NINPUTS 5
#define NOUTPUTS 1
#define DSTATES 0

/* ===== Funções Auxiliares ===== */

static void euler_forward(const real_T* A, const real_T* B,
                        const real_T* x, const real_T* u,
                        int nx, int nu, real_T Ts, real_T* x_next)
{
    int i, j;
    for (i = 0; i < nx; ++i) {
        real_T accAx = 0.0;
        for (j = 0; j < nx; ++j) {
            accAx += A[i + j*nx] * x[j];
        }

        real_T accBu = 0.0;
        for (j = 0; j < nu; ++j) {
            accBu += B[i + j*nx] * u[j];
        }

        x_next[i] = x[i] + Ts * accAx + Ts * accBu;
    }
}

static void calculate_output(const real_T* C, const real_T* E,
                           const real_T* x, const real_T* u,
                           int nx, int nu, int ny, real_T* y_out)
{
    int i, j;
    for (i = 0; i < ny; ++i) {
        y_out[i] = 0.0;
        // Parte C*x
        for (j = 0; j < nx; ++j) {
            y_out[i] += C[i + j*ny] * x[j];
        }
        // Parte E*u
        for (j = 0; j < nu; ++j) {
            y_out[i] += E[i + j*ny] * u[j];
        }
    }
}

/* * predict_N_steps:
 * Replica o loop 'for i = 1:N' para predição de horizonte longo.
 */
static void predict_N_steps(const real_T* A, const real_T* B,
                           const real_T* x_initial, const real_T* u,
                           int nx, int nu, real_T Ts, int N,
                           real_T* x_final)
{
    int i, j;
    // Buffers temporários para o loop
    real_T* x_temp = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T* x_next = (real_T*) mxMalloc(nx, sizeof(real_T));

    // Inicializa x_temp com o estado inicial
    for(i=0; i < nx; ++i) {
        x_temp[i] = x_initial[i];
    }

    // Executa a predição N vezes
    for (i = 0; i < N; ++i) {
        // x_next = euler(x_temp)
        euler_forward(A, B, x_temp, u, nx, nu, Ts, x_next);

        // Copia x_next de volta para x_temp para a próxima iteração
        // x_temp = x_next
        for(j=0; j < nx; ++j) {
            x_temp[j] = x_next[j];
        }
    }

    // Copia o resultado final para o ponteiro de saída
    for(i=0; i < nx; ++i) {
        x_final[i] = x_temp[i];
    }

    // Libera memória dos buffers
    mxFree(x_temp);
    mxFree(x_next);
}

```

```

/* ===== Métodos da S-Function ===== */

static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, NUM_PARAMS); // Atualizado para 11
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        ssSetErrorStatus(S, "Número incorreto de parâmetros. Esperado: 11.");
        return;
    }

#ifdef MATLAB_MEX_FILE
    {
        /* Validação das matrizes */
        const mxArray *A1 = ssGetSFcnParam(S, P_A1), *B1 = ssGetSFcnParam(S, P_B1);
        const mxArray *C1 = ssGetSFcnParam(S, P_C1), *E1 = ssGetSFcnParam(S, P_E1);
        const mxArray *A2 = ssGetSFcnParam(S, P_A2), *B2 = ssGetSFcnParam(S, P_B2);
        const mxArray *C2 = ssGetSFcnParam(S, P_C2), *E2 = ssGetSFcnParam(S, P_E2);

        mwSize nx = mxGetM(A1);
        if (nx != mxGetN(A1) || nx != mxGetM(A2) || nx != mxGetN(A2)) {
            ssSetErrorStatus(S, "As matrizes A1 e A2 devem ser quadradas e de mesma dimensão.");
        }
        return;
        mwSize nu = mxGetN(B1);
        if (nx != mxGetM(B1) || nx != mxGetM(B2) || nu != mxGetN(B2)) {
            ssSetErrorStatus(S, "Dimensões de B1/B2 incompatíveis com A1/A2."); return;
        }
        mwSize ny = mxGetM(C1);
        if (ny != mxGetM(C2) || nx != mxGetN(C1) || nx != mxGetN(C2)) {
            ssSetErrorStatus(S, "Dimensões de C1/C2 incompatíveis com A1/A2."); return;
        }
        if (ny != mxGetM(E1) || ny != mxGetM(E2) || nu != mxGetN(E1) || nu != mxGetN(E2)) {
            ssSetErrorStatus(S, "Dimensões de E1/E2 incompatíveis com B1/B2."); return;
        }

        if (mxGetNumberOfElements(ssGetSFcnParam(S, P_T_REF)) != 1) {
            ssSetErrorStatus(S, "Parâmetro 't_ref' (P10) deve ser um escalar."); return;
        }
        if (mxGetNumberOfElements(ssGetSFcnParam(S, P_N)) != 1) {
            ssSetErrorStatus(S, "Parâmetro 'N' (P11) deve ser um escalar."); return;
        }
    }
#endif

    /* Portas de Entrada e Saída */
    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, NINPUTS);
    ssSetInputPortDirectFeedThrough(S, 0, 1);

    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, NOUTPUTS);

    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, 0);

    /* Configuração dos DWork Vectors */
    ssSetNumDWork(S, NUM_DWORKS);
    int i;
    for (i = 0; i < NUM_DWORKS; ++i) {
        ssSetDWorkWidth(S, i, 1);
        ssSetDWorkDataType(S, i, SS_DOUBLE);
        ssSetDWorkComplexSignal(S, i, COMPLEX_NO);
        ssSetDWorkUsedAsDState(S, i, 1); // Usado para salvar estado entre passos
    }

    ssSetNumSampleTimes(S, 1);
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

static void mdlInitializeSampleTimes(SimStruct *S)
{
    real_T f_MPC = mxGetScalar(ssGetSFcnParam(S, P_F MPC));

    real_T Ts = 1.0 / f_MPC;
    ssSetSampleTime(S, 0, Ts);
    ssSetOffsetTime(S, 0, 0.0);
}

#define MDL_START
static void mdlStart(SimStruct *S)
{
    // Obtém ponteiros para os DWork vectors
    real_T *Vpv_ref_anterior = (real_T*) ssGetDWork(S, DWORK_VPV_REF_ANT);
    real_T *delta_t_ref = (real_T*) ssGetDWork(S, DWORK_DELTA_T);
    real_T *Ref_pos = (real_T*) ssGetDWork(S, DWORK_REF_POS);
    real_T *Ref_neg = (real_T*) ssGetDWork(S, DWORK_REF_NEG);

    *Vpv_ref_anterior = 0.0;
    *delta_t_ref = 0.0;
    *Ref_pos = 0.0;
    *Ref_neg = 0.0;
}

```



```

static void mdlOutputs(SimStruct *S, int_T tid)
{
    /* --- Ponteiros de Entrada --- */
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);
    real_T Vpv_ref = *uPtrs[0];
    real_T vC      = *uPtrs[1];
    real_T iL      = *uPtrs[2];
    real_T Vo      = *uPtrs[3];
    real_T Ipv     = *uPtrs[4];

    /* --- Ponteiro de Saída --- */
    real_T *g = ssGetOutputPortRealSignal(S,0);

    /* --- Parâmetros --- */
    real_T f_MPC = mxGetScalar(ssGetSFcnParam(S, P_F MPC));
    real_T Ts    = 1.0 / f_MPC;
    real_T t_ref = mxGetScalar(ssGetSFcnParam(S, P_T_REF));
    int    N     = (int) mxGetScalar(ssGetSFcnParam(S, P_N));

    /* Ponteiros para Matrizes */
    const mxArray *A1_mx = ssGetSFcnParam(S, P_A1), *B1_mx = ssGetSFcnParam(S, P_B1);
    const mxArray *C1_mx = ssGetSFcnParam(S, P_C1), *E1_mx = ssGetSFcnParam(S, P_E1);
    const mxArray *A2_mx = ssGetSFcnParam(S, P_A2), *B2_mx = ssGetSFcnParam(S, P_B2);
    const mxArray *C2_mx = ssGetSFcnParam(S, P_C2), *E2_mx = ssGetSFcnParam(S, P_E2);

    const real_T *A1 = mxGetPr(A1_mx), *B1 = mxGetPr(B1_mx);
    const real_T *C1 = mxGetPr(C1_mx), *E1 = mxGetPr(E1_mx);
    const real_T *A2 = mxGetPr(A2_mx), *B2 = mxGetPr(B2_mx);
    const real_T *C2 = mxGetPr(C2_mx), *E2 = mxGetPr(E2_mx);

    int nx = (int) mxGetM(A1_mx);
    int nu = (int) mxGetN(B1_mx);
    int ny = (int) mxGetM(C1_mx);

    /* Ponteiros para DWork (Variáveis Persistentes) */
    real_T *Vpv_ref_anterior_ptr = (real_T*) ssGetDWork(S, DWORK_VPV_REF_ANT);
    real_T *delta_t_ref_ptr      = (real_T*) ssGetDWork(S, DWORK_DELTA_T);
    real_T *Ref_pos_ptr          = (real_T*) ssGetDWork(S, DWORK_REF_POS);
    real_T *Ref_neg_ptr          = (real_T*) ssGetDWork(S, DWORK_REF_NEG);

    // Copia valores atuais do DWork para variáveis locais
    real_T Vpv_ref_anterior = *Vpv_ref_anterior_ptr;
    real_T delta_t_ref      = *delta_t_ref_ptr;
    real_T Ref_pos          = *Ref_pos_ptr;
    real_T Ref_neg          = *Ref_neg_ptr;

    /* --- Alocação de Memória --- */
    // Buffers para predição k+2
    real_T *xo = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *u  = (real_T*) mxMalloc(nu, sizeof(real_T));
    real_T *x_1 = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *x_2 = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *x_11 = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *x_12 = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *x_21 = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *x_22 = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *y_11 = (real_T*) mxMalloc(ny, sizeof(real_T));
    real_T *y_12 = (real_T*) mxMalloc(ny, sizeof(real_T));
    real_T *y_21 = (real_T*) mxMalloc(ny, sizeof(real_T));
    real_T *y_22 = (real_T*) mxMalloc(ny, sizeof(real_T));

    // Buffers para predição restritiva k+N
    real_T *x_N_on = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *y_N_on = (real_T*) mxMalloc(ny, sizeof(real_T));
    real_T *x_N_off = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T *y_N_off = (real_T*) mxMalloc(ny, sizeof(real_T));

    /* --- Inicialização dos vetores --- */
    xo[0] = vC; xo[1] = iL;
    u[0] = Vo; u[1] = Ipv;

    /* Definindo a direção da variação de referência */
    if (Vpv_ref_anterior != Vpv_ref) {
        delta_t_ref = 0.0; // Contagem reiniciada
        if (Vpv_ref_anterior < Vpv_ref) { // Analisando se a nova referência é positiva
            Ref_neg = 0.0; // false
            Ref_pos = 1.0; // true
        } else { // Se a nova referência é negativa
            Ref_pos = 0.0; // false
            Ref_neg = 1.0; // true
        }
    }

    /* --- Predição da árvore de decisão k+2 --- */
    euler_forward(A1, B1, xo, u, nx, nu, Ts, x_1); /* g(k) = ON */
    euler_forward(A2, B2, xo, u, nx, nu, Ts, x_2); /* g(k) = OFF */

    euler_forward(A1, B1, x_1, u, nx, nu, Ts, x_11); /* ON -> ON */
    calculate_output(C1, E1, x_11, u, nx, nu, ny, y_11);
    euler_forward(A2, B2, x_1, u, nx, nu, Ts, x_12); /* ON -> OFF */
    calculate_output(C2, E2, x_12, u, nx, nu, ny, y_12);
    euler_forward(A1, B1, x_2, u, nx, nu, Ts, x_21); /* OFF -> ON */
    calculate_output(C1, E1, x_21, u, nx, nu, ny, y_21);
    euler_forward(A2, B2, x_2, u, nx, nu, Ts, x_22); /* OFF -> OFF */
    calculate_output(C2, E2, x_22, u, nx, nu, ny, y_22);

    /* --- Extração das predições de Vpv k+2 --- */
    real_T Vpv_11 = y_11[0], Vpv_12 = y_12[0], Vpv_21 = y_21[0], Vpv_22 = y_22[0];

    /* --- Cálculo das Funções Custo k+2 --- */
    real_T Fc[4];
    Fc[0] = (Vpv_ref - Vpv_11) * (Vpv_ref - Vpv_11); // Fc_11
    Fc[1] = (Vpv_ref - Vpv_12) * (Vpv_ref - Vpv_12); // Fc_12
    Fc[2] = (Vpv_ref - Vpv_21) * (Vpv_ref - Vpv_21); // Fc_21
    Fc[3] = (Vpv_ref - Vpv_22) * (Vpv_ref - Vpv_22); // Fc_22

```

```

/* --- Rotina Restritiva k+N --- */

// Predições k+N para a etapa ATIVA (g = 1)
if ((delta_t_ref <= t_ref) && (Ref_neg == 1.0)) {
    // Simula N passos com g=1 (A1, B1, C1, E1)
    predict_N_steps(A1, B1, xo, u, nx, nu, Ts, N, x_N_on);
    calculate_output(C1, E1, x_N_on, u, nx, nu, ny, y_N_on);

    real_T VpvN_on = y_N_on[0];
    if (VpvN_on < Vpv_ref) { // Se a predição indica sobressinal
        Fc[0] = INFINITY; // Penaliza Fc_11
        Fc[1] = INFINITY; // Penaliza Fc_12
    }
}

// Predições k+N para a etapa DESATIVA (g = 0)
if ((delta_t_ref <= t_ref) && (Ref_pos == 1.0)) {
    // Simula N passos com g=0 (A2, B2, C2, E2)
    predict_N_steps(A2, B2, xo, u, nx, nu, Ts, N, x_N_off);
    calculate_output(C2, E2, x_N_off, u, nx, nu, ny, y_N_off);

    real_T VpvN_off = y_N_off[0];
    if (VpvN_off > Vpv_ref) { // Se a predição indica sobressinal
        Fc[2] = INFINITY; // Penaliza Fc_21
        Fc[3] = INFINITY; // Penaliza Fc_22
    }
}

/* --- Escolha da Ação de Controle --- */
real_T minFc = Fc[0];
int arg_min = 0;
int i;
for (i = 1; i < 4; ++i) {
    if (Fc[i] < minFc) {
        minFc = Fc[i];
        arg_min = i;
    }
}
// acoes_g = [1 1 0 0] -> se arg_min for 0 ou 1, g = 1 (ON)
g[0] = (arg_min < 2) ? 1.0 : 0.0;

/* --- Atualização das Variáveis Persistentes (DWork) --- */
*Vpv_ref_anterior_ptr = Vpv_ref;
*delta_t_ref_ptr = delta_t_ref + Ts;
*Ref_pos_ptr = Ref_pos;
*Ref_neg_ptr = Ref_neg;

/* --- Liberação da Memória --- */
mxFree(xo); mxFree(u); mxFree(x_1); mxFree(x_2);
mxFree(x_11); mxFree(x_12); mxFree(x_21); mxFree(x_22);
mxFree(y_11); mxFree(y_12); mxFree(y_21); mxFree(y_22);

// Libera buffers da predição k+N
mxFree(x_N_on); mxFree(y_N_on);
mxFree(x_N_off); mxFree(y_N_off);
}

static void mdlTerminate(SimStruct *S)
{
}

#ifdef MATLAB_MEX_FILE
#include "simulink.c"
#else
#include "cg_sfuns.h"
#endif

```


APÊNDICE F – CÓDIGO S-FUNCTION FCS-MPC - RESTRIÇÃO DE TENSÃO E HORIZONTE DE PREDIÇÃO ESTENDIDO

```

/*-----*/
/* Filename:   BOOST_FCS_MPC_Ten_Restrito.c */
/* Description: S-Function para o algoritmo MPC-FCS (k+2) */
/* COM RESTRIÇÃO NA TENSÃO E HORIZONTE DE PREDIÇÃO ESTENDIDO. */
/* Level:      S-Function Nível 2 */
/* Autor:      Yuri Giglio de Azevedo */
/* Adaptado de código Matlab de Lucas Gabriel Pedroso de Lima Costa */
/*-----*/
#define S_FUNCTION_NAME BOOST_FCS_MPC_Ten_Restrito
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include <math.h>
#include <string.h>
#ifdef MATLAB_MEX_FILE
#include "mex.h"
#endif

enum {
    P_A1 = 0, P_B1, P_C1, P_E1, P_A2, P_B2, P_C2, P_E2, // MATRIZES
    P_LAMBDAA1,
    P_N,
    P_FMP,
    NUM_PARAMS // 11 parâmetros
};

#define NOUTPUTS 1
#define DSTATES 0

/* Funções auxiliares euler_forward, calculate_output, n_steps */

/* Função para achar o valor predito no próximo estado k+1 */
static void euler_forward(const real_T* A, const real_T* B, const real_T* x, const real_T* u,
                        int nx, int nu, real_T Ts, real_T* xnext) {
    int i, j;
    for (i = 0; i < nx; ++i) {
        real_T accAx = 0.0;
        for (j = 0; j < nx; ++j) accAx += A[i + j*nx] * x[j];
        xnext[i] = x[i] + Ts * accAx;
        if (u) {
            real_T accBu = 0.0;
            for (j = 0; j < nu; ++j) accBu += B[i + j*nx] * u[j];
            xnext[i] += Ts * accBu;
        }
    }
}

/* Calcula o valor predito para a saída */
static void calculate_output(const real_T* C, const real_T* E, const real_T* x, const real_T* u,
                        int ny, int nx, int nu, real_T* y_out) {
    int i, j;
    for (i = 0; i < ny; ++i) {
        real_T accCx = 0.0;
        for (j = 0; j < nx; ++j) accCx += C[i + j*ny] * x[j];
        y_out[i] = accCx;
        if (u) {
            real_T accEu = 0.0;
            for (j = 0; j < nu; ++j) accEu += E[i + j*ny] * u[j];
            y_out[i] += accEu;
        }
    }
}

/* Algoritmo que chama simulação do sistema por N passos de tempo */
static void n_steps(const real_T* A, const real_T* B, const real_T* x0, const real_T* u,
                    int nx, int nu, real_T Ts, int K, real_T* x_out) {
    int k;
    real_T* xcur = (real_T*) mxMalloc(nx, sizeof(real_T));
    real_T* xnext = (real_T*) mxMalloc(nx, sizeof(real_T));
    memcpy(xcur, x0, nx*sizeof(real_T));
    for (k = 0; k < K; ++k) {
        euler_forward(A, B, xcur, u, nx, nu, Ts, xnext);
        memcpy(xcur, xnext, nx*sizeof(real_T));
    }
    memcpy(x_out, xcur, nx*sizeof(real_T));
    mxFree(xcur); mxFree(xnext);
}

/* ===== Métodos da S-Function ===== */
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, NUM_PARAMS);
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        ssSetErrorStatus(S, "Número incorreto de parâmetros. Esperado 11.");
        return;
    }

    ssSetNumInputPorts(S, 1);
    ssSetInputPortWidth(S, 0, DYNAMICALLY_SIZED);
    ssSetInputPortDirectFeedThrough(S, 0, 1);

    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, NOUTPUTS);

    ssSetNumContStates(S, 0);
    ssSetNumDiscStates(S, DSTATES);
    ssSetNumSampleTimes(S, 1);
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}

```

```

static void mdlInitializeSampleTimes(SimStruct *S)
{
    real_T f_MPC = mxGetScalar(ssGetSFcnParam(S, P_F MPC)); // P_F MPC é o índice 10
    real_T Ts = 1.0 / f_MPC;
    ssSetSampleTime(S, 0, Ts);
    ssSetOffsetTime(S, 0, 0.0);
}

static void mdlOutputs(SimStruct *S, int_T tid)
{
    InputRealPtrsType u_all = ssGetInputPortRealSignalPtrs(S, 0);
    real_T *y = ssGetOutputPortRealSignal(S, 0);

    /* Dimensões baseadas nas matrizes */
    int nx = (int) mxGetM(ssGetSFcnParam(S, P_A1));
    int nu = (int) mxGetM(ssGetSFcnParam(S, P_B1));
    int ny = (int) mxGetM(ssGetSFcnParam(S, P_C1));

    real_T f_MPC = mxGetScalar(ssGetSFcnParam(S, P_F MPC)); // P_F MPC (índice 10)
    real_T Ts = 1.0 / f_MPC;

    const real_T *A1 = mxGetPr(ssGetSFcnParam(S, P_A1)), *B1 = mxGetPr(ssGetSFcnParam(S, P_B1)), *C1 =
mxGetPr(ssGetSFcnParam(S, P_C1)), *E1 = mxGetPr(ssGetSFcnParam(S, P_E1));
    const real_T *A2 = mxGetPr(ssGetSFcnParam(S, P_A2)), *B2 = mxGetPr(ssGetSFcnParam(S, P_B2)), *C2 =
mxGetPr(ssGetSFcnParam(S, P_C2)), *E2 = mxGetPr(ssGetSFcnParam(S, P_E2));

    real_T Vref = *u_all[0];
    real_T *x0 = (real_T*) mxCalloc(nx, sizeof(real_T));
    real_T *uvec = (real_T*) mxCalloc(nu, sizeof(real_T));
    for (int i = 0; i < nx; ++i) x0[i] = *u_all[1 + i];
    for (int i = 0; i < nu; ++i) uvec[i] = *u_all[1 + nx + i];

    {
        // Parâmetros lidos com os novos índices da enum
        real_T lambda = mxGetScalar(ssGetSFcnParam(S, P_LAMBDA1)); // P_LAMBDA1 (índice 8)
        int N = (int) mxGetScalar(ssGetSFcnParam(S, P_N)); // P_N (índice 9)
        if (N < 1) N = 1;

        real_T *x_1 = (real_T*) mxCalloc(nx, sizeof(real_T)), *x_2 = (real_T*) mxCalloc(nx, sizeof(real_T));
        real_T *x_11 = (real_T*) mxCalloc(nx, sizeof(real_T)), *x_12 = (real_T*) mxCalloc(nx, sizeof(real_T));
        real_T *x_21 = (real_T*) mxCalloc(nx, sizeof(real_T)), *x_22 = (real_T*) mxCalloc(nx, sizeof(real_T));
        real_T *x_N_on = (real_T*) mxCalloc(nx, sizeof(real_T)), *x_N_off = (real_T*) mxCalloc(nx, sizeof(real_T));
        real_T *y_11 = (real_T*) mxCalloc(ny, sizeof(real_T)), *y_12 = (real_T*) mxCalloc(ny, sizeof(real_T));
        real_T *y_21 = (real_T*) mxCalloc(ny, sizeof(real_T)), *y_22 = (real_T*) mxCalloc(ny, sizeof(real_T));
        real_T *y_on = (real_T*) mxCalloc(ny, sizeof(real_T)), *y_off = (real_T*) mxCalloc(ny, sizeof(real_T));

        euler_forward(A1, B1, x0, uvec, nx, nu, Ts, x_1);
        euler_forward(A2, B2, x0, uvec, nx, nu, Ts, x_2);

        euler_forward(A1, B1, x_1, uvec, nx, nu, Ts, x_11); calculate_output(C1, E1, x_11, uvec, ny, nx, nu, y_11); //
Caso chave on-on
        euler_forward(A2, B2, x_1, uvec, nx, nu, Ts, x_12); calculate_output(C2, E2, x_12, uvec, ny, nx, nu, y_12); //
Caso chave on-off
        euler_forward(A1, B1, x_2, uvec, nx, nu, Ts, x_21); calculate_output(C1, E1, x_21, uvec, ny, nx, nu, y_21); //
Caso chave off-on
        euler_forward(A2, B2, x_2, uvec, nx, nu, Ts, x_22); calculate_output(C2, E2, x_22, uvec, ny, nx, nu, y_22); //
Caso chave off-off

        real_T Vpv_11 = y_11[0], Vpv_12 = y_12[0], Vpv_21 = y_21[0], Vpv_22 = y_22[0];

        n_steps(A1, B1, x0, uvec, nx, nu, Ts, N, x_N_on); calculate_output(C1, E1, x_N_on, uvec, ny, nx, nu, y_on);
        // Caso ON até k+N
        n_steps(A2, B2, x0, uvec, nx, nu, Ts, N, x_N_off); calculate_output(C2, E2, x_N_off, uvec, ny, nx, nu, y_off);
        // Caso OFF até k+N

        real_T VpvN_on = y_on[0], VpvN_off = y_off[0];

        real_T Fc[4];
        Fc[0] = pow(Vref - Vpv_11, 2.0) + lambda * pow(Vref - VpvN_on, 2.0);
        Fc[1] = pow(Vref - Vpv_12, 2.0) + lambda * pow(Vref - VpvN_on, 2.0);
        Fc[2] = pow(Vref - Vpv_21, 2.0) + lambda * pow(Vref - VpvN_off, 2.0);
        Fc[3] = pow(Vref - Vpv_22, 2.0) + lambda * pow(Vref - VpvN_off, 2.0);

        int min_idx = 0;
        for (int i = 1; i < 4; ++i) if (Fc[i] < Fc[min_idx]) min_idx = i; // Determina o menor valor
        y[0] = (min_idx < 2) ? 1.0 : 0.0; // Se min_idx for 0 ou 1 (caminhos 11 ou 12), a ação g(k) é 1 (ON)

        mxFree(x_1); mxFree(x_2); mxFree(x_11); mxFree(x_12); mxFree(x_21); mxFree(x_22);
        mxFree(x_N_on); mxFree(x_N_off); mxFree(y_11); mxFree(y_12); mxFree(y_21);
        mxFree(y_22); mxFree(y_on); mxFree(y_off);
    }

    mxFree(x0); mxFree(uvec);
}

static void mdlTerminate(SimStruct *S) {}

#ifdef MATLAB_MEX_FILE
#include "simulink.c"
#else
#include "cg_sfun.h"
#endif

```