

Universidade Federal de Mato Grosso do Sul
Faculdade de Engenharias, Arquitetura e Urbanismo e Geografia
Dissertação de mestrado do curso de Engenharia Elétrica

Thyago Vasconcelos Estrabis

**Desenvolvimento e Implementação de um
Controlador Preditivo Generalizado (GPC)
para Rastreamento de Referência Rampa**

Campo Grande - MS
06 de Dezembro de 2020

Universidade Federal de Mato Grosso do Sul
Faculdade de Engenharias, Arquitetura e Urbanismo e Geografia
Dissertação de mestrado do curso de Engenharia Elétrica

Thyago Vasconcelos Estrabis

**Desenvolvimento e Implementação de um
Controlador Preditivo Generalizado (GPC) para
Rastreamento de Referência Rampa**

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFMS para obtenção do Grau de Mestre em Engenharia Elétrica na área de Computação Flexível.

Orientador: Professor Dr. Raymundo Cordero García

Campo Grande - MS
06 de Dezembro de 2020



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



ATA DE DEFESA DE DISSERTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO

Aos dezesseis dias do mês de dezembro do ano de dois mil e vinte, às catorze horas e trinta minutos, por Videoconferência, reuniu-se a Banca Examinadora composta pelos membros: Raymundo Cordero Garcia (UFMS), Luís Guilherme Barbosa Rolim (UFRJ), Marcio Luiz Magri Kimpapa (UFMS) e Walter Issamu Suemitsu (UFRJ), sob a presidência do primeiro, para julgar o trabalho do aluno: THYAGO VASCONCELOS ESTRABIS, CPF 03812490188, do Programa de PósGraduação em Engenharia Elétrica, Curso de Mestrado, da Fundação Universidade Federal de Mato Grosso do Sul, apresentado sob o título "Desenvolvimento e Implementação de um Controlador Preditivo Generalizado (GPC) para Rastreamento de Referência Rampa" e orientação de Raymundo Cordero Garcia. O presidente da Banca Examinadora declarou abertos os trabalhos e agradeceu a presença de todos os Membros. A seguir, concedeu a palavra ao aluno que expôs sua Dissertação. Terminada a exposição, os senhores membros da Banca Examinadora iniciaram as arguições. Terminadas as arguições, o presidente da Banca Examinadora fez suas considerações. A seguir, a Banca Examinadora reuniu-se para avaliação, e após, emitiu parecer expresso conforme segue:

EXAMINADOR

Dr. Raymundo Cordero Garcia (Interno)

Dr. Cristiano Quevedo Andrea (Interno) (Suplente)

Dr. Luís Guilherme Barbosa Rolim (Externo) (Suplente)

Dr. Marcio Luiz Magri Kimpapa (Interno)

Dr. Walter Issamu Suemitsu (Externo)

RESULTADO FINAL:

(x) Aprovação () Aprovação com revisão () Reprovação

OBSERVAÇÕES:

Nada mais havendo a ser tratado, o Presidente declarou a sessão encerrada e agradeceu a todos pela presença.

Agradecimentos

Agradeço a minha família em primeiro lugar, sempre me dando suporte necessário para dar sequência nos meus estudos. Em especial a minha mãe Maria Aparecida Dourados Vasconcelos, pelos carinhos, suporte e zelo, meu pai José Estrabis pelo suporte, compreensão, a minha irmã Nayara Vasconcelos Estrabis, que é a minha inspiração de perseverança na área acadêmica, pelas dicas e todo carinho e "puxões de orelha".

Aos meus amigos que me proporcionaram momentos de descontração e de conversas que me permitiram crescer como pessoa, me ajudaram imensamente. Em especial ao Felipe Alexandre Monteiro, meu amigo de longa data que sempre esteve presente na minha vida, ao Gabriel Gentil, por ser essa pessoa incrível, pela parceria e suporte, a Renata Reis Kimpara, pela compreensão e pelas críticas construtivas, ao João Marcos Soares Anjos, pela parceria e suporte. Agradeço ao pessoal do capítulo estudantil PELS UFMS, em especial ao César Augusto Ferreira Santos, Matheus de Carvalho Pelzl, Henrique Matheus Gonçalves dos Santos e Mariana Espinola Ribeiro pelos momentos de descontração.

Agradeço aos professores que me deram apoio durante meu tempo no mestrado. Em especial meu orientador, Dr. Raymundo Cordero García por ser uma pessoa incrível e paciente, pelas dicas, conselhos e histórias contadas. Ao meu coorientador, Dr Edson Antonio Batista, pela ajuda e dicas, e por ter me apresentado o GPC, que acabou sendo parte do tema da dissertação. Ao professor Dr João Onfre Pereira Pinto, diretor do BATLAB, local onde desenvolvi a minha dissertação.

Agradeço aos meus companheiros e professores do BATLAB e do Programa de Pós Graduação em Engenharia Elétrica (PPGEE) da UFMS pelo apoio e auxílio durante o meu mestrado.

Resumo

Atualmente, o Controle por Modelo Preditivo (*Model Predictive Control* – MPC em Inglês) é uma das mais avançadas técnicas de controle aplicada em diferentes processos industriais. Entre todas as técnicas baseadas em MPC, o Controle Preditivo Generalizado (Generalized Predictive Control – GPC) é uma das mais estudadas e aplicadas. Em GPC, a saída do sistema é predita através de um modelo aumentado da planta, aplicando entradas presentes e futuras a ela, enquanto uma função custo é calculada sobre um horizonte de predição (um conjunto de resposta preditas). O objetivo do GPC é obter um conjunto de leis de controle futuras que reduz o erro entre as saídas preditas e a referência. Porém, o GPC convencional não pode ser aplicado para o rastreamento de referência rampa, a qual é usada em importantes aplicações práticas. Este trabalho tem por objetivo o desenvolvimento e a implementação experimental de um algoritmo GPC capaz de rastrear referências constantes e rampas. O operador de diferença de segunda ordem é utilizado para criar um novo modelo aumentado da planta onde a saída é o erro de rastreamento do sistema, no lugar da saída da planta. O modelo proposto possui dois integradores embutidos, o que permite o rastreamento de referências tipo rampa. A função custo e as técnicas de otimização usadas no algoritmo GPC convencional podem ser aplicadas no modelo aumentado proposto para obter a lei de controle. Resultados de simulação e experimentais comprovam que a técnica proposta pode rastrear adequadamente referências tipo constante e rampa.

Palavras-chaves: Controle por Modelo Preditivo, Controle Preditivo Generalizado, Operador de diferença de 2da ordem, rastreamento de referência rampa..

Abstract

Nowadays, Model Predictive Control (MPC) control is one of the most advanced control techniques applied in different industrial processes. Among all techniques based on MPC, Generalized Predictive Control (GPC) is one of the most studied and applied. In GPC, the system's output is predicted through an augmented model of the plant, applying present and future inputs to it, while a cost function is calculated over a prediction horizon (a set of predicted responses). The GPC aims to obtain a set of future control laws that reduces the error between the predicted outputs and the reference. However, conventional GPC cannot be applied for ramp reference tracking, which is used in critical practical applications. This Thesis proposes the design and experimental implementation of a GPC algorithm capable of tracking constant references and ramps. The second-order difference operator is used to create a new augmented plant model where the output is the system tracking error, instead of the plant output. The proposed model has two embedded integrators, allowing the tracking of ramp references. The cost function and optimization techniques used in the conventional GPC algorithm can be applied to the augmented model proposed to obtain the control law. Simulation and experimental results prove that the proposed technique can adequately track constant and ramp references .

Keywords: Model predictive control, generalized predictive control, second order difference operator, ramp reference tracking.

Lista de ilustrações

Figura 1.0.1 – Conceito do MPC.	13
Figura 1.0.2 – Estrutura básica do MPC	14
Figura 1.0.3 – Árvore Genealógica do MPC	16
Figura 2.1.1 – Diagrama dem bloco de um sistema GPC de malha fechada	26
Figura 2.2.1 – Controlador de modelo interno para referência tipo degrau.	28
Figura 2.2.2 – Controlador de modelo interno para referência tipo rampa.	30
Figura 3.1.1 – Sinal de Entrada da Referência do tipo Rampa	32
Figura 3.1.2 – Diagrama em bloco do SOD-GPC	37
Figura 3.1.3 – Sistema SOD-GPC	38
Figura 3.3.1 – Estrutura do sensor resolver	42
Figura 3.3.2 – Sistema ATO baseado no SOD-GPC	43
Figura 3.3.3 – Diagrama de bloco simplificado do sistema ATO para o SOD-GPC	45
Figura 4.1.1 – Referência de entrada do Tipo Rampa	47
Figura 4.1.2 – Diagrama em bloco do SOD-GPC	47
Figura 4.1.3 – Bancada Experimental	48
Figura 4.1.4 – Onda de referência da planta $G(s)$	48
Figura 4.1.5 – Resultados de Simualções. (a) GPC proposto usando $N_P = 3, N_C = 2, r_w = 0.0003$. (b) GPC proposto $N_P = 3, N_C = 2, r_w = 0.01$. (c) GPC proposto $N_P = 4, N_C = 3, r_w = 0.0003$. (d) GPC convencional usando $N_P = 3, N_C = 2, r_w = 0.0003$. (e) $C1(z)$. (f) $C2(z)$	49
Figura 4.1.6 – Resultados Experimentais. (a) GPC proposto usando $N_P = 3, N_C = 2, r_w = 0.0003$. (b) GPC proposto $N_P = 3, N_C = 2, r_w = 0.01$. (c) GPC proposto $N_P = 4, N_C = 3, r_w = 0.0003$. (d) GPC convencional usando $N_P = 3, N_C = 2, r_w = 0.0003$. (e) $C1(z)$. (f) $C2(z)$	50
Figura 4.2.1 – Curva de velocidade para a simulação do sensor resolver.	52
Figura 4.2.2 – Sistema RDC mencionado.	52
Figura 4.2.3 – Resultados de simulação for $N_p = 102, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.	53
Figura 4.2.4 – Resultados de simulação for $N_p = 120, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.	54
Figura 4.2.5 – Resultados de simulação for $N_p = 102, N_c = 10, r_w = 0.01$. a) Sem ruído, b) Com ruído.	55
Figura 4.2.6 – Resultados de simulação para o sistema RDC mencionado. a) Sem ruído, b) Com ruído.	55

Figura 4.2.7 – Bancada experimental para o teste do ATO proposto.	56
Figura 4.2.8 – Resultados experimentais para $N_p = 102, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.	57
Figura 4.2.9 – Resultados experimentais para $N_p = 120, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.	58
Figura 4.2.10 – Resultados experimentais para $N_p = 102, N_c = 10, r_w = 0.01$. a) Sem ruído, b) Com ruído.	58
Figura 4.3.2 – Resultados da simulação para $N_p = 3, N_c = 2, r_w = 0.01$	59
Figura 4.3.1 – Referência dente de serra.	59
Figura 4.3.3 – Resultados da simulação para $N_p = 3, N_c = 2, r_w = 0.1$	60
Figura 4.3.4 – Resultados da simulação para $N_p = 4, N_c = 2, r_w = 0.01$	60
Figura 4.3.5 – Resultados da simulação para $N_p = 4, N_c = 2, r_w = 0.1$	61
Figura 4.3.6 – Resultados da simulação para $N_p = 8, N_c = 2, r_w = 0.01$	61
Figura 4.3.7 – Resultados da simulação para $N_p = 8, N_c = 2, r_w = 0.1$	62

Lista de tabelas

Tabela 1	–	Resultados experimentais para a planta $G(s)$	51
Tabela 2	–	Parâmetros de simulação para o sensor resolver.	51
Tabela 3	–	Valores do rmse e do tempo de assentamento nos testes.	53
Tabela 4	–	Valores do rmse e do tempo de assentamento nos testes experimentais.	56

Sumário

1	Introdução	12
1.1	Motivação	18
1.2	Objetivos	18
1.2.1	Objetivo Geral	18
1.2.2	Objetivos Específicos	19
1.3	Organização do Trabalho	19
2	Fundamentação Teórica	20
2.1	GPC Convencional	20
2.1.1	Modelo Aumentado	20
2.1.2	Predição	22
2.1.3	Otimização	23
2.1.4	Sistema em Malha Fechada	25
2.2	Controlador de Modelo Interno	26
2.2.1	Rastreamento de Referência Tipo Degrau	27
2.2.2	Rastreamento de Referência Tipo Rampa	28
3	GPC para Rastreamento de Referência Tipo Rampa	31
3.1	GPC para Rastreamento de Referência Tipo Rampa	31
3.1.1	Operador de Diferença de Segunda Ordem	31
3.1.2	Preliminares	31
3.1.3	Modelo Aumentado	33
3.1.4	Cálculo da Lei de Controle	36
3.1.5	Função de Transferência do Modelo Aumentado Proposto	38
3.2	Discretização de Plantas Definidas em Tempo Contínuo	39
3.3	Sistema ATO baseado em GPC	42
3.3.1	Resolver	42
3.3.2	Sistema ATO proposto	42
4	Resultados	46
4.1	Controle de Planta Contínua $G(s)$	46
4.2	Desenvolvimento do ATO baseado em GPC	51
4.2.1	Resultados das Simulações	51
4.2.2	Resultados Experimentais	53
4.3	Ensaio Adicionais	57

Conclusão	63
Referências	64
Apêndices	68
A Códigos	69
A.1 MATLAB	69
A.2 VHDL	74

1 Introdução

O desenvolvimento de processos industriais e componentes eletrônicos cada vez mais complexos exigem técnicas de controle com melhor robustez e velocidade de resposta. Diferentes técnicas de controle foram propostas na literatura: controle adaptativo, controle robusto, técnicas de inteligência artificial, controle não linear, entre outros.

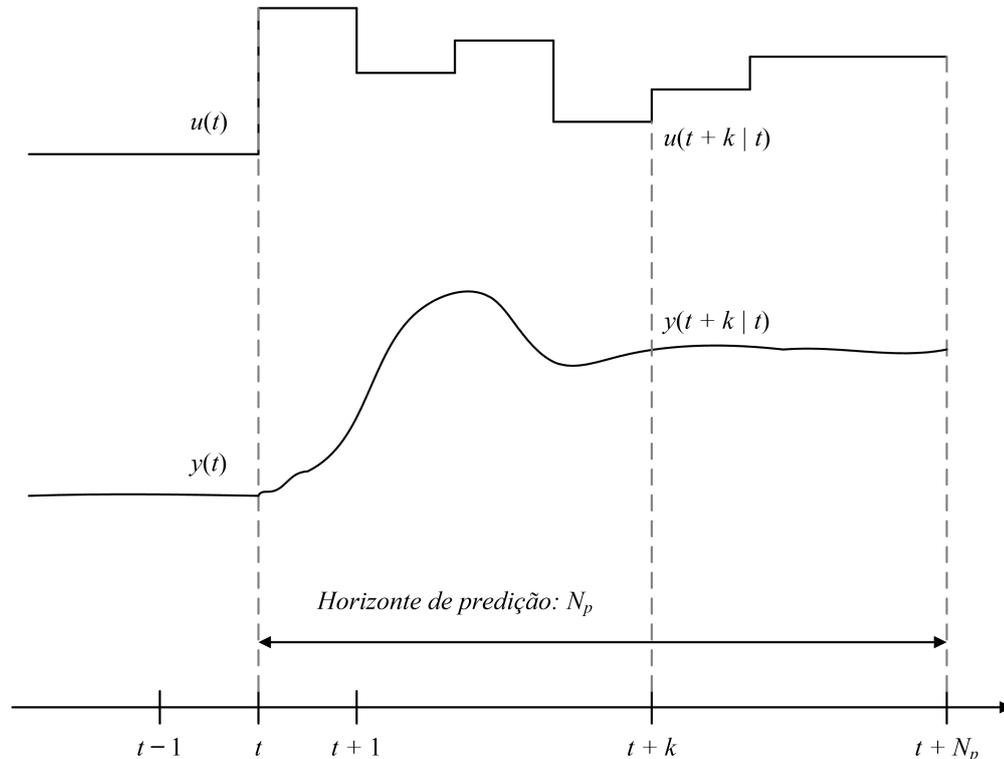
O Controle por Modelo Preditivo (*Model Predictive Control*, MPC, em Inglês) desenvolveu-se consideravelmente nas últimas décadas, tanto na área de pesquisa quanto na indústria. Esse sucesso pode ser atribuído ao fato de o MPC lidar com o problema de controle de processo no domínio do tempo. Em MPC, as respostas da planta são preditas com o objetivo de obter uma sequência de leis de controle futuras que minimizem o erro de controle. O controle preditivo foi integrado a técnicas de controle ótimo, controle estocástico, controle de processos com tempo morto, controle multivariável, entre outros [1–8]. Adicionalmente, é possível implementar restrições na formulação do MPC, enquanto também é possível aplicar esta técnica para o controle de sistemas não lineares.

O MPC possui muitas vantagens sobre outras técnicas de control [1]:

- Seus modelos são compostos por conceitos intuitivos;
- Pode ser usado para controlar uma grande variedade de processos com dinâmica simples, e para outras mais complexas, incluindo sistemas com longos tempos de atraso ou fases não mínimas ou instáveis;
- Pode ser facilmente aplicado para sistemas multivariável;
- Intrinsecamente possui compensação por tempos mortos;
- Introduz o controle de alimentação em avanço (*feedforward*) de maneira natural para compensar distúrbios mensuráveis;
- O controlador resultante é uma lei de controle de fácil implementação;
- Sua extensão ao tratamento de restrições é conceitualmente simples, e estes podem ser incluídos sistematicamente durante o processo de projeto do controlador;
- É uma metodologia totalmente aberta, baseada em certos princípios básicos que permite futuras extensões;
- Utiliza-se de um modelo explícito para prever as dinâmicas da planta no futuro;

Conforme pode ser visto na [Figura 1.0.1](#), os modelos existente no MPC possuem um comportamento em relação ao sinal a ser referenciado, sendo elas [1]:.

Figura 1.0.1 – Conceito do MPC.

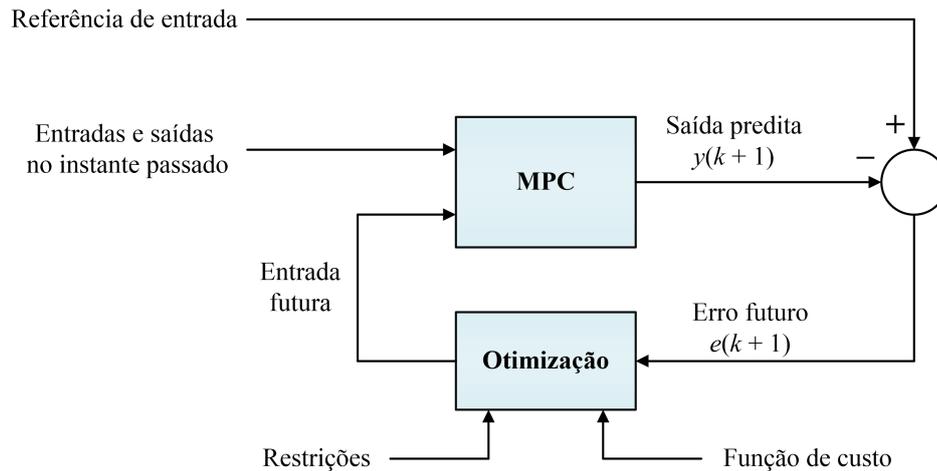


Fonte: Retirado de [1]

1. As saídas futuras para um horizonte definido N_P , chamado horizonte de predição, são previstas em cada instante de tempo t usando o modelo do processo. Essas saídas previstas $y(t+k|t)$ para $k = 0 \dots N_p - 1$, são calculadas os vetores de estado e os sinais de controle presentes e futuros;
2. O conjunto de sinais de controle futuros são calculados otimizando um determinado critério de forma a manter o processo mais próximo possível da trajetória de referência $w(t+k)$ (que pode ser o próprio *set - point* ou uma aproximação dele). Esse critério geralmente assume a forma de uma função quadrática dos erros entre o sinal de saída previsto e a trajetória de referência prevista. O esforço de controle é incluído na função objetivo na maioria dos casos. Uma solução explícita pode ser obtida se o critério é quadrático, o modelo é linear e não existirem restrições. Caso contrário, um método iterativo de otimização deve ser utilizado para obter as leis de controle futuras;
3. O sinal de controle $u(t|t)$ é enviado para o processo enquanto os sinais de controle seguintes são descartados, porque, no próximo instante de amostragem, $y(t+1)$

será conhecido, o passo 1 se repetirá com esse novo valor e todas as sequências serão atualizadas. Portanto, o $u(t + 1|t + 1)$ é calculado (que em princípio será diferente de $u(t + 1|t)$) porque novas informações foram apresentadas usando o conceito de horizonte deslizante.

Figura 1.0.2 – Estrutura básica do MPC



Fonte: Autor

Na [Figura 1.0.2](#), observa-se a estrutura básica do MPC. Essa é uma estrutura essencial e pode ser utilizada para um modelo de processo ou um modelo de perturbação. Em relação aos modelos de processos, eles podem ser descritos por:

- Resposta ao impulso – muito utilizado nas indústrias que utilizam referências do tipo Degrau;
- Função de transferência – usado tanto nas indústrias quanto na comunidade acadêmica;
- Modelo em espaço de estado – muito utilizado nas comunidades acadêmicas e científicas.

Quanto aos modelos de perturbação, normalmente usam-se da Regressão Automática Controlada e no médio movimento integrado (CARIMA) [9, 10].

Na [Figura 1.0.2](#) nota-se a existência de um bloco de otimização, em que estão contidos a Função Custo do controlador e as Restrições que podem existir na planta. A Função Custo e as restrições são elementos contidos no processo de otimização. A Função Custo J atua penalizando os desvios das saídas preditas com a trajetória de referência. As restrições, apresentam uma dificuldade quando são aplicadas na entrada,

pois representam limitações no atuador do processo (que gera a lei de controle). Além desse motivo, as restrições são aplicadas na saída, devido a razões de segurança e controlar antecipadamente a saída, porque as variáveis de saída são afetadas pela dinâmica do processo. Existem vários tipos de restrições de entrada, saída ou estado:

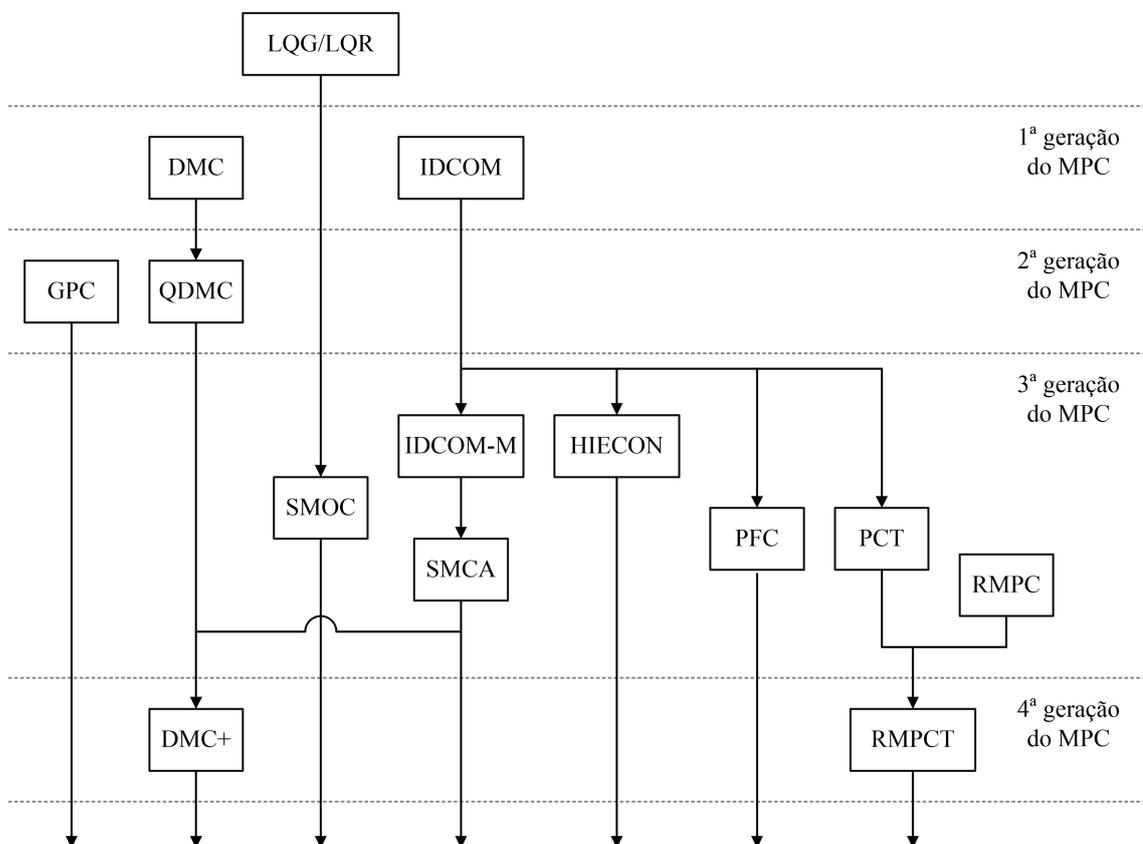
- Restrição de banda (*Band Constraints*);
- Restrições de ultrapassagem (*Overshoot Constraints*);
- Comportamento monotônico (*Monotonic behaviour*);
- Comportamento não mínimo da fase (*Non-minimal phase behaviour*);
- Não linearidades do atuador (*Actuator nonlinearities*);
- Restrições de igualdade do estado terminal (*Terminal state equality constraints*);
- Restrições do conjunto de terminais. (*Terminal set constraints*).

O surgimento do MPC foi no começo dos anos 60, pela apresentação dos trabalhos desenvolvido por Kalman e seus coautores [11], [12]. Os controladores que deram início ao MPC foram o LQR e LQG. Mesmo sendo criado na década de 60, o MPC começou a ser realmente utilizado no final do período de sua primeira geração. A [Figura 1.0.3](#) mostra a árvore genealógica do MPC e suas gerações:

- LQG/LQR: Regulador Linear Quadrático Gaussiano/Regulador Linear Quadrático (*Linear Quadratic Gaussian/Linear Quadratic Regulator*);
- DMC: Controle Matricial Dinâmico (*Dynamic Matrix Control*);
- IDCOM: Programa de Identificação e Comando (*Identification and Command*);
- GPC: Controle Preditivo Generalizado (*Generalized Predictive Control*);
- QMDC: Controle Matricial Dinâmico Quadrático (*Quadratic Dynamic Matrix Control*);
- SMOC: Controlador Multivariável Optimizado da *Shell* (*Shell Multivariable Optimizing Controller*);
- IDCOM-M: IDCOM Multivariável;
- HIECON: Controle de Restrição Hierárquica (*Hierarchical Constraint Control*);
- PFC: Controle Preditivo Funcional (*Predictive Functional Control*);
- PCT: Tecnologia de Controle Preditivo (*Predictive Control Technology*);

- SMCA: Arquitetura de Controle de Referência Multivariável (*Setpoint Multivariable Control Architecture*);
- RMPC: Controle por Modelo Preditivo Robusto (*Robust Model Predictive Control*);
- DMC+: Pacote de Controle Matricial Dinâmico da *Aspen Technology*;
- RMPCT: Tecnologia de Controle por Modelo Preditivo Robusto (*Robust Model Predictive Control Technology*).

Figura 1.0.3 – Árvore Genealógica do MPC



Fonte: Autor

O MPC é bem utilizado nas indústrias pelos seguintes fatores:

- Lida naturalmente com controle de multivariáveis;
- Leva em consideração as limitações do atuador;
- Permite que operações sejam mais próximas das restrições, aumentando o lucro;
- Possui flexibilidade em lidar com processos não mínimos e estáveis;
- Robusto a mudanças estruturais;

- É um método fácil de ajustar.

MPC Linear e MPC Não Linear são as duas principais classes do MPC. O MPC Linear pode possuir as seguintes características em sua formulação:

- Modelo Linear: $\dot{x} = Ax + Bu$;
- Função Custo Quadrático: $F = x^T Qx + u^T Ru$;
- Restrições Lineares: $Hx + Gu < 0$;
- Programação Quadrática.

No entanto, o MPC Não Linear pode possuir as seguintes características:

- Modelo Não Linear: $\dot{x} = f(x, u)$;
- Função Custo pode ser Não Quadrático: $F = (x, u)$;
- Restrições Não Lineares: $h(x, u) < 0$;
- Programação não Quadrática.

Uma das técnicas baseadas em modelo preditivo mais utilizadas é o Controle Preditivo Generalizado (*Generalized Predictive Control*, GPC em Inglês). Nesta técnica, as saídas são preditas aplicando sinais de entrada ao modelo da planta, enquanto uma função custo é calculada. A função custo é definida para um conjunto de resposta preditas, chamado de horizonte de predição. Esta função é uma medida do desempenho do sistema, e depende do erro entre as respostas preditas e o vetor de referência desejada para o horizonte de predição. O conjunto de leis de controle futuras é calculado para minimizar a função custo.

Convencionalmente, a referência do GPC é considerada como constante no horizonte de predição. Esta consideração é verdadeira para uma referência degrau [2]. Um modelo aumentado é utilizado para prever as respostas da planta. O modelo aumentado possui um integrador, o que permite reduzir o erro em regime permanente para uma referência tipo degrau. Porém, esta consideração não é verdadeira quando a referência da planta é uma rampa. Em [3] é descrita uma técnica de controle preditivo para rastreamento em rampa, mas esta técnica requer informação sobre o sinal de referência (o que não sempre é possível).

1.1 Motivação

O Controle baseado em Modelo Preditivo (MPC) antes utilizado em processos químicos e petroquímicos, acabou ganhando expressão em diversas áreas industriais e acadêmicas. Com passar do tempo, diferentes variações do algoritmos MPC foram propostas [1,2,4–8,13]. O Controle Preditivo Generalizado (GPC) foi a técnica escolhida neste trabalho, pela sua flexibilidade na implementação do algoritmo em processadores digitais, robustez para determinados sistemas e uma estrutura intuitiva. O GPC é um controlador que foi criado na década de 80 por Clarke em [9,10]. Com o passar dos anos, o GPC foi apresentando algumas otimizações para algumas aplicações em específico [1,2,6–10,13–16]. Mesmo tendo um acervo literário bem difundido no meio acadêmico, praticamente todos os trabalhos são para referência de entrada do tipo Degrau. Em [14] é mostrado uma aplicação complexa de um rastreamento de rampa usando GPC. A referência tipo rampa é utilizada em processos térmicos, robótica e servomecanismos [1]. O GPC convencional não pode ser aplicado diretamente no rastreamento de referência rampa principalmente por duas razões:

- No GPC convencional [2], o vetor de referência do horizonte de predição é constante. Esta consideração é falsa para um sinal de referência tipo rampa;
- O modelo aumentado do GPC convencional possui só um integrador. Porém, para o rastreamento de referência tipo rampa, dois integradores embutidos são necessários [17].

A motivação deste estudo é realizar uma modificação no GPC para que seja possível realizar o rastreamento em rampa em sistemas do tipo SISO e que tenha uma formulação simples de implementar nos sistemas que necessitem rastrear rampas.

Os modelos aumentados utilizados para a predição das respostas da planta estão baseados no cálculo da diferença de primeira ordem do vetor de estado. Neste trabalho, explora-se o uso do operador de diferença de maior ordem, para o desenvolvimento de modelos aumentados com um maior número de integradores embutidos.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento e teste experimental de um algoritmo GPC para o rastreamento de referência degrau e rampa. O sistema GPC proposto utiliza o operador de diferença de segunda ordem para obter um modelo aumentado com dois integradores embutidos e cuja saída é o erro de rastreamento. Desta maneira, as

técnicas de otimização utilizadas no GPC convencional podem ser utilizadas para obter a lei de controle.

1.2.2 Objetivos Específicos

1. Compreensão matemática do algoritmo GPC convencional.
2. Desenvolvimento do modelo aumentado para predição baseado no operador de diferença de segunda ordem.
3. Aplicação das técnicas de otimização para o cálculo da lei de controle.
4. Implementação de plantas experimentais para o teste do algoritmo proposto.

1.3 Organização do Trabalho

- **Capítulo 2:** é apresentada uma fundamentação teórica para o desenvolvimento do controlador.
- **Capítulo 3:** apresenta-se uma nova estrutura de GPC que possibilita realizar o rastreamento de sinais do tipo Rampa.
- **Capítulo 4:** são apresentados os resultados dos ensaios realizados.
- **Capítulo 5:** encontram-se as conclusões sobre os resultados obtidos.

2 Fundamentação Teórica

Neste capítulo serão descritas a técnica GPC convencional e o controlador de modelagem interna, os quais são a base do algoritmo proposto.

2.1 GPC Convencional

2.1.1 Modelo Aumentado

Uma planta linear, de ordem n , com uma entrada e uma saída (SISO) definida no tempo discreto pode ser descrita pelo modelo mostrado na equação 2.1.

$$\begin{aligned}x_m(k+1) &= A_m x_m(k) + B_m u(k) \\ y(k) &= C_m x_m(k)\end{aligned}\tag{2.1}$$

onde $u(k)$ é uma entrada manipulada ou manipulável, $y(k)$ é a saída do sistema, e $x_m(k)$ é o vetor da variável de estado com uma dimensão assumida em n . Considera-se que a planta não tem um integrador embutido. Sendo assim, é necessário o projeto de um sistema de controle que possua um integrador para que o erro em regime permanente seja zero para uma entrada degrau. Para o desenvolvimento do controlador no GPC, é necessário desenvolver um modelo aumentado com um integrador embutido. Para o desenvolvimento do modelo aumentado, é necessária uma consideração na formulação geral de um modelo espaço de estados, pois existe um termo direto do sinal de entrada $u(k)$ para a saída $y(k)$, como na equação 2.2:

$$y(k) = C_m x_m(k) + D_m u(k)\tag{2.2}$$

Porém, de acordo com o princípio do controle horizontal de recuo, para a predição de controle, a informação atual da planta é necessária. Adicionalmente, a entrada $u(k)$ não pode afetar a saída $y(k)$ no mesmo tempo [2]. Portanto, neste trabalho considera-se $D_m = 0$ no modelo da planta.

A partir da equação 2.1, o vetor da variável de estado no instante atual $x_m(k)$ pode ser descrita pela equação 2.3.

$$x_m(k) = A_m x_m(k-1) + B_m u(k-1)\tag{2.3}$$

Após realizar a operação de diferença entre as equações 2.1 e 2.3, obtém-se seguinte

equação:

$$\begin{aligned} x_m(k+1) - x_m(k) &= A_m x_m(k) + B_m u(k) - A_m x_m(k-1) - B_m u(k-1) \\ x_m(k+1) - x_m(k) &= A_m (x_m(k) - x_m(k-1)) + B_m (u(k) - u(k-1)) \end{aligned} \quad (2.4)$$

Definem-se as diferenças da variável de estado presente $\Delta x_m(k)$ e futuro $\Delta x_m(k+1)$ segundo as equações 2.5 e 2.6, respectivamente:

$$\Delta x_m(k) = x_m(k) - x_m(k-1) \quad (2.5)$$

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k) \quad (2.6)$$

Sendo Δ o operador de diferença de primeira ordem ($\Delta a(k) = a(k) - a(k-1)$, seja $a(k)$ um escalar ou vetor). A diferença da variável de controle é descrita por:

$$\Delta u(k) = u(k) - u(k-1) \quad (2.7)$$

A partir das equações 2.4 e 2.7, o sinal $\Delta x_m(k+1)$ pode ser descrito da seguinte maneira:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (2.8)$$

No modelo aumentado descrito em [2], os sinais $\Delta x_m(k)$ e $y(k)$ são utilizados. Assim, o vetor de estados do modelo aumentado é definido na equação 2.9:

$$x(k) = [\Delta x_m(k)^T y(k)^T]^T \quad (2.9)$$

A sigla T representa o operador utilizado para transpor uma matriz.

É necessário determinar os valores de $\Delta x_m(k+1)$ e $y(k+1)$ para definir o modelo em espaço de estados discreto. Considera-se $\Delta u(k)$ como a entrada do modelo aumentado. Assim, a equação 2.8 permite definir $\Delta x_m(k+1)$ a partir das variáveis de estado e da entrada do modelo aumentado. Na sequência, é necessário definir o valor de $y(k+1)$ a partir de $\Delta x_m(k)$ e $y(k)$ e $\Delta u(k)$. com base na equação 2.1:

$$y(k+1) = C_m x_m(k+1) \quad (2.10)$$

Encontrando a diferença entre as equações 2.10 e 2.1:

$$\begin{aligned} y(k+1) - y(k) &= C_m x_m(k+1) - C_m x_m(k) \\ y(k+1) &= C_m (x_m(k+1) - x_m(k)) + y(k) \end{aligned} \quad (2.11)$$

Substituindo $x_m(k+1)$ da equação 2.1 e $x_m(k)$ da equação 2.3 na equação 2.8, tem-se:

$$\begin{aligned} y(k+1) &= C_m ((A_m x_m(k) + B_m u(k)) - (A_m x_m(k-1) + B_m u(k-1))) + y(k) \\ y(k+1) &= C_m ((A_m x_m(k) - A_m x_m(k-1)) + (B_m u(k) - B_m u(k-1))) + y(k) \\ y(k+1) &= C_m A_m \Delta x_m + C_m B_m \Delta u(k) + y(k) \end{aligned} \quad (2.12)$$

A equação 2.13 descreve o modelo aumentado do GPC convencional em forma matricial [2]. As equações 2.8 e 2.12 formam esse modelo.

$$\begin{aligned} \begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} &= \overbrace{\begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix}}^A \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^B \Delta u(k) \\ y(k) &= \overbrace{\begin{bmatrix} o_m & 1 \end{bmatrix}}^C \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} \end{aligned} \quad (2.13)$$

As variáveis A , B e C , no modelo indicado na equação 2.13, são chamadas de variáveis do modelo aumentado. Estas variáveis são responsáveis por descrever o modelo aumentado. A variável o_m , que aparece na matriz A e na matriz C da equação 2.13, é uma matriz linha de zeros n_1 de comprimento $o_m = \overbrace{[0 \ 0 \ \dots \ 0]}^n$.

2.1.2 Predição

O próximo passo é definir a janela de otimização, com o cálculo da saída da planta prevista, junto do sinal futuro com as variáveis ajustáveis. Considera-se que, no instante k_i , o vetor de estados do modelo aumentado $x(k_i)$ é conhecido ou medido. A trajetória do controle futuro é descrita por:

$$\Delta u(k_i), \Delta u(k_i + 1) \cdots \Delta u(k_i + N_C - 1) \quad (2.14)$$

onde N_C é variável chamada de horizonte de controle, responsável por definir os parâmetros usados para capturar a trajetória do controle futuro. Com a informação $x(k_i)$ dada, o futuro das variáveis de estados são prevista para o número de amostras N_P , onde N_P é o comprimento da janela de otimização.

Definem-se as variáveis de estado preditas como:

$$x(k_i + 1|k_i), x(k_i + 2|k_i), \cdots x(k_i + m|k_i), x(k_i + N_P|k_i) \quad (2.15)$$

De acordo com as variáveis de estado do modelo aumentado e da trajetória do controle futuro (equações 2.13 e 2.14), as variáveis do estado e as saídas são preditas sequencialmente usando um conjunto de parâmetros de controle futuro [2]:

$$\begin{aligned} x(k_i + 1|k_i) &= Ax(k_i) + B\Delta u(k_i) \\ x(k_i + 2|k_i) &= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \\ &\vdots \\ x(k_i + N_P|k_i) &= A^{N_P}x(k_i) + A^{N_P-1}B\Delta u(k_i) + \\ &A^{N_P-2}B\Delta u(k_i + 1) + \cdots + A^{N_P-N_C}B\Delta u(k_i + N_C - 1) \end{aligned} \quad (2.16)$$

$$\begin{aligned}
y(k_i + 1|k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
y(k_i + 2|k_i) &= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\
&\vdots \\
y(k_i + N_P|k_i) &= CA^{N_P}x(k_i) + CA^{N_P-1}B\Delta u(k_i) + \\
&CA^{N_P-2}B\Delta u(k_i + 1) + \dots + CA^{N_P-N_C}B\Delta u(k_i + N_C - 1)
\end{aligned} \tag{2.17}$$

Pelas equações 2.16 e 2.17, pode-se observar que a variável predita depende do estado atual e o movimento do controle futuro. Define-se o vetor de saídas preditas (Y) e o vetor de controle futuro (ΔU) da seguinte forma:

$$Y = [y(k_i + 1|k_i) \quad y(k_i + 2|k_i) \quad \dots \quad y(k_i + N_P|k_i)]^T \tag{2.18}$$

$$\Delta U = [\Delta u(k_i) \quad \Delta u(k_i + 1) \quad \dots \quad \Delta u(k_i + N_C - 1)]^T \tag{2.19}$$

As equações 2.18 e 2.19 podem ser representados em uma matriz compacta, como é mostrado na equação a seguir:

$$Y = Fx(k_i) + \phi\Delta U \tag{2.20}$$

onde:

$$\begin{aligned}
F &= \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_P} \end{bmatrix} \\
\phi &= \begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CA & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ CA^{N_P-1} & CA^{N_P-1} & \dots & CA^{N_P-N_C} \end{bmatrix}
\end{aligned} \tag{2.21}$$

2.1.3 Otimização

O objetivo do sistema de controle preditivo é trazer a saída do sistema o mais próximo do sinal de referência (*set-point*). Considera-se $r(k_i)$ como a referência no instante de tempo k_i , dentro do sinal de predição. O *set-point*, para o modelo apresentado, será considerado constante na janela de otimização.

Na visão do projeto, o objetivo será utilizado para descobrir a melhor solução ΔU , de tal maneira que o erro entre o *set-point* e a saída prevista seja minimizada.

Define-se o vetor de referência da janela de otimização R_s da seguinte maneira:

$$R_s^T = \overbrace{[1 \quad 1 \quad \dots \quad 1]}^{N_P} \tag{2.22}$$

A função custo J é utilizada no GPC convencional para medir o desempenho da planta. De acordo com [2], a função custo possui a seguinte estrutura:

$$J = (R_S - Y)^T(R_S - Y) + \Delta U^T \bar{R} \Delta U \quad (2.23)$$

Como pode ser visto na equação 2.23, o primeiro termo é conectado no objetivo de minimização dos erros entre a saída e o sinal *set-point*, o segundo termo leva em consideração o tamanho da ação de controle ΔU quando a função objetivo J é criada para ser o menor possível. A variável \bar{R} que aparece no segundo termo da equação 2.23 é uma matriz diagonal e pode ser descrita na equação a seguir:

$$\bar{R} = r_w I_{N_C \times N_C} \quad (2.24)$$

Pode ser observado na equação 2.24 a presença de duas variáveis I e r_w . A variável I é uma matriz identidade $N_C \times N_C$, enquanto que r_w é um parâmetro de ajuste da função custo. Por esse motivo, a matriz \bar{R} se comporta como um parâmetro de ajuste em um sistema desejado de malha fechada. Quando $r_w = 0$, a função custo interpreta que a dimensão do ΔU pode ser desconsiderada, e os objetivos seriam destinados a obter o menor erro $(R_S - Y)^T(R_S - Y)$ possível. No caso de r_w ter um alto valor numérico positivo, a função de custo J , é interpretado como uma situação onde poderia ser considerado cuidadosamente a magnitude do sinal ΔU enquanto se reduz o valor do erro $(R_S - Y)^T(R_S - Y)$.

Para que a condição necessária do mínimo J seja obtida, é necessário aplicar o critério da primeira derivada. A primeira derivada da função custo é:

$$\frac{\partial J}{\partial \Delta u} = 2\phi^T(R_S - Fx(k_i)) + 2(\phi^T \phi + \bar{R})\Delta U \quad (2.25)$$

O critério da primeira derivada indica que o valor ótimo de ΔU faz a derivada da função igual a zero.

$$\frac{\partial J}{\partial \Delta u} = 0 \quad (2.26)$$

A partir das equações 2.23 e 2.25, o sinal de controle ΔU ótimo é definida pela a seguinte expressão [2]:

$$\Delta U = (\phi^T \phi + \bar{R})^{-1} \phi^T (R_S - Fx(k_i)) \quad (2.27)$$

Considera-se que $(\phi^T \phi + \bar{R})^{-1}$ exista. A matriz $(\phi^T \phi + \bar{R})^{-1}$, na literatura de otimização, é chamada de matriz Hessiana. A equação 2.27 pode ser descrita da seguinte forma:

$$\Delta U = (\phi^T \phi + \bar{R})^{-1} (\phi^T R_S - \phi^T Fx(k_i)) \quad (2.28)$$

2.1.4 Sistema em Malha Fechada

Em um sistema de malha fechada, a expressão $(\phi^T \phi + \bar{R})^{-1} \phi^T R_S$, encontrada na equação 2.28, corresponde à mudança do set-point, enquanto para $(\phi^T \phi + \bar{R})^{-1} \phi^T x(k_i)$ corresponde o estado do controle de realimentação dentro da estrutura do controle preditivo. Ambos dependem dos parâmetros do sistema, por isso, as suas matrizes são constantes para um sistema invariante no tempo. Pelo princípio de controle horizontal de recuo, somente o primeiro elemento de ΔU no tempo k_i , $\Delta u(k_i)$, é utilizado para calcular a lei de controle da planta. Portanto:

$$\Delta u(k_i) = \overbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}}^{N_c} (\phi^T \phi + \bar{R})^{-1} (\phi^T \bar{R}_S - \phi^T F x(k_i)) \quad (2.29)$$

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} x(k_i)$$

onde K_y é o primeiro elemento de $(\phi^T \phi + \bar{R})^{-1} (\phi^T \bar{R}_S)$ e K_{mpc} é a primeira linha de $(\phi^T \phi + \bar{R})^{-1} (\phi^T F x(k_i))$.

A equação de $\Delta u(k_i)$ é a forma padrão do controle de estado linear invariante no tempo. O vetor do ganho do estado de realimentação é descrito por K_{mpc} .

Sendo a estrutura do modelo aumentado no instante k_i :

$$x(k_i + 1) = Ax(k_i) + B\Delta u(k_i) \quad (2.30)$$

Para se obter o sistema de malha fechado é necessário realizar a substituição da equação 2.29 na equação 2.30. Logo o sistema de malha fechado pode ser descrito pela equação 2.31:

$$\begin{aligned} x(k_i) &= Ax(k_i) + B(K_y r(k_i) - K_{mpc} x(k_i)) \\ x(k_i + i) &= (A - BK_{mpc})x(k_i) + BK_y r(k_i) \end{aligned} \quad (2.31)$$

Os autovalores do sistema de malha fechada podem ser obtidos através da equação característica da malha fechada indicada na equação 2.32.

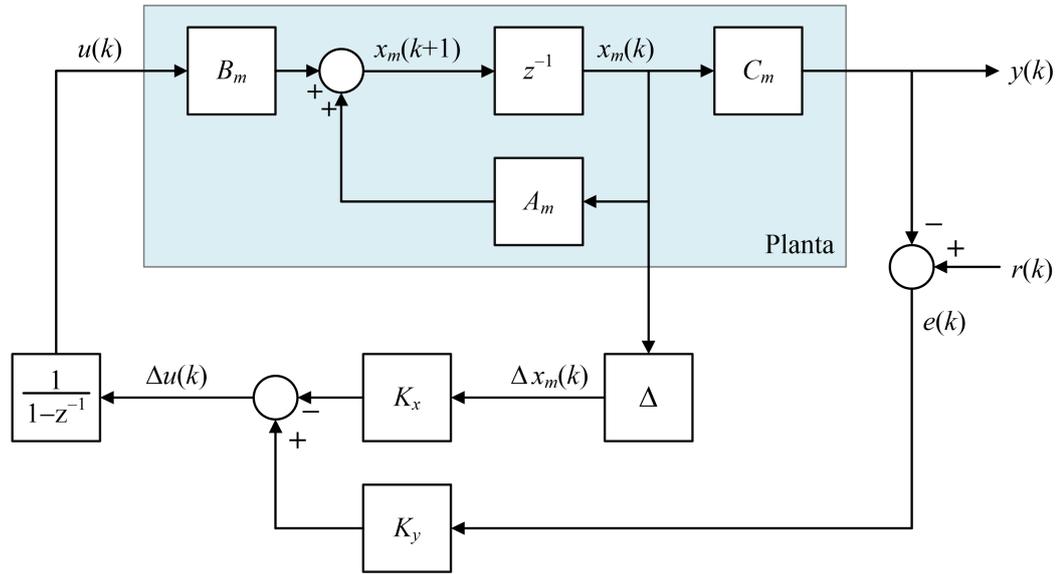
$$\det[\lambda I - (A - BK_{mpc})] = 0 \quad (2.32)$$

Por causa da estrutura especial das matrizes C e A , a última coluna de F é idêntica com \bar{R}_S , que é $\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T$. Consequentemente, o ganho K_y será o último elemento de K_{mpc} . Considerando as definições de $x(k_i)$, 2.9 e do ganho K_y , o vetor de ganho K_{mpc} pode ser reescrito da seguinte maneira:

$$K_{mpc} = [K_x \ K_y] \quad (2.33)$$

onde K_x corresponde ao vetor ganho de realimentação relacionado a $\Delta x_m(k)$. O ganho K_y será o ganho de realimentação relacionado a saída $\Delta y(k)$. Assim, o sistema GPC em malha fechada pode ser descrito segundo a [Figura 2.1.1](#).

Figura 2.1.1 – Diagrama dem bloco de um sistema GPC de malha fechada



Fonte: Autor

Na [Figura 2.1.1](#) o bloco z^{-1} indica o operador de deslocamento de atraso, $1 - z^{-1}$ é a função de transferência discreta equivalente ao operador Δ , enquanto que $\frac{1}{1-z^{-1}}$ corresponde a um integrador de tempo discreto.

2.2 Controlador de Modelo Interno

Considerando o modelo de estado variável de uma planta dada pela equação [2.34](#):

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{y} &= Cx \end{aligned} \tag{2.34}$$

onde x é o vetor de estado, u é a entrada e y é a saída. Considera-se também que as entradas de referências a ser gerado de um sistema linear da seguinte forma:

$$\begin{aligned} \dot{x} &= A_r x_r \\ r &= d_r x_r \end{aligned} \tag{2.35}$$

O modelo abaixo [2.36](#) é um modelo equivalente da referência de entrada $r(t)$, com

condições iniciais desconhecidas.

$$r^n = \alpha_{n-1}r^{(n-1)} + \alpha_{n-2}r^{(n-2)} + \dots + \alpha_1\dot{r} + \alpha_0r \quad (2.36)$$

Onde r^n é a n -ésima derivada de $r(t)$.

2.2.1 Rastreamento de Referência Tipo Degrau

Considera-se um projeto de um controlador para permitir o rastreamento de uma entrada degrau de referência com erro zero no estado estacionário, logo a entrada de referência pode ser gerada por:

$$\begin{aligned} \dot{x} &= 0 \\ r &= x_r \end{aligned} \quad (2.37)$$

Ou equivalente a:

$$\dot{r} = 0 \quad (2.38)$$

O erro por ser definido por:

$$e = y - r \quad (2.39)$$

A derivada do erro será:

$$\dot{e} = \dot{y} = C\dot{x} \quad (2.40)$$

A equação 2.41 pode ser obtida após definir $z = \dot{x}$ e $\omega = \dot{u}$ juntar o modelo da referência na entrada 2.38 com o modelo do processo indicado na equação 2.34 [17]:

$$\begin{bmatrix} \dot{e} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & C \\ 0 & A \end{bmatrix} \begin{bmatrix} e \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} \omega \quad (2.41)$$

A equação 2.42 é a forma da realimentação, e ela só pode ser definida se a equação 2.41 for controlável.

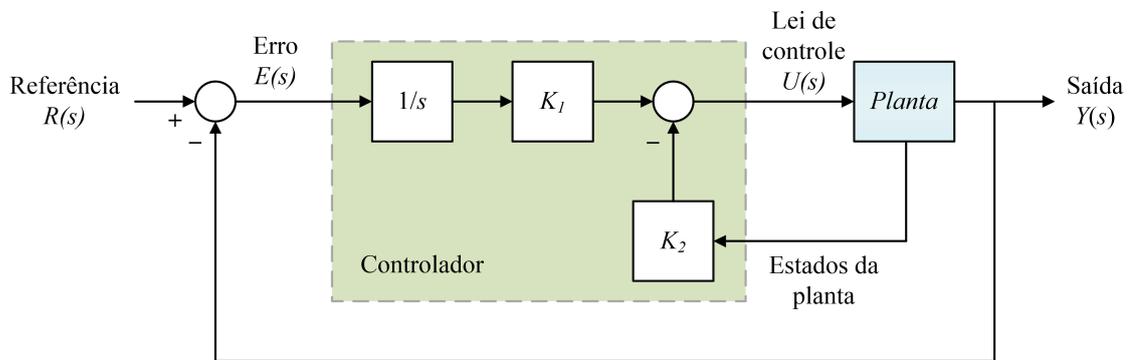
$$\omega = -K_1e - K_2z \quad (2.42)$$

O rastreamento do erro e é estável apenas se a equação 2.41 for estável. Com isso, é possível atingir o objetivo como sendo erro de rastreamento igual a zero no estado estacionário. A entrada de controle $u(t)$ 2.43 é obtida pela integral da equação 2.42.

$$u(t) = -K_1 \int_0^t e(\tau) d\tau - K_2 x(t) \quad (2.43)$$

A Figura 2.2.1 mostra em diagrama de bloco o modelo de design interno. Nota que o integrador é um compensador no modelo interno para uma entrada degrau [17].

Figura 2.2.1 – Controlador de modelo interno para referência tipo degrau.



Fonte: Autor

2.2.2 Rastreamento de Referência Tipo Rampa

Considera-se um projeto de um controlador para permitir o rastreamento de uma entrada de referência rampa com erro zero no estado estacionário 2.44 [17].

$$r(t) = Mt \quad (2.44)$$

Em que $t \geq 0$ e M é a magnitude da rampa, nesse caso o modelo da referência de entrada é descrito pela equação 2.45:

$$\begin{aligned} \dot{x}_r &= A_r x_r = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_r \\ r &= d_r x_r = \begin{bmatrix} 1 & 0 \end{bmatrix} x_r \end{aligned} \quad (2.45)$$

A referência tipo rampa pode ser modelada como uma função linear $r = at + b$. Portanto, pode-se demonstrar que a segunda derivada da referência é igual a zero.

$$\ddot{r} = 0 \quad (2.46)$$

Derivando duas vezes a equação do erro em 2.39, e considerando a equação 2.46, tem-se:

$$\ddot{e} = \ddot{y} = C\ddot{x} \quad (2.47)$$

A equação 2.48 pode ser obtida após definir $z = \ddot{x}$ e $z = \ddot{u}$ e juntar o modelo da referência na entrada 2.46 com o modelo do processo 2.34 [17]:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & C \\ 0 & 0 & A \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \omega \quad (2.48)$$

O sinal de controle é definido através de realimentação dos estados do modelo descrito na equação 2.48:

$$\omega = \ddot{u} = - \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ z \end{bmatrix} \quad (2.49)$$

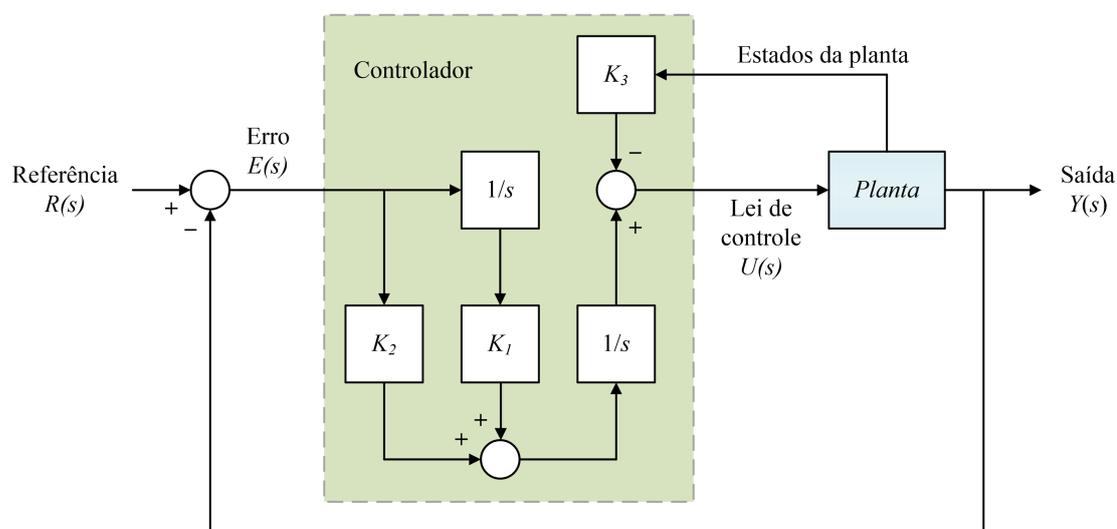
$$K_o = \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix} \quad (2.50)$$

onde K_o é a matriz de realimentação de estados. Substituindo a equação 2.49 em 2.48, tem-se o modelo em malha fechada do sistema de controle:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \\ \dot{z} \end{bmatrix} = \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & C \\ 0 & 0 & A \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} K_o \right\} \begin{bmatrix} e \\ \dot{e} \\ z \end{bmatrix} \quad (2.51)$$

Técnicas como a equação de Ackermann podem ser utilizadas para definir K_o para que os autovalores da equação 2.51 tenham os valores de projeto desejados (segundo os objetivos de projeto como o tempo de assentamento ou a ultrapassagem). Ademais, se o sistema da equação 2.51 é assintoticamente estável, conseqüentemente o erro rastreado tende a zero $e(t) \rightarrow 0$ com o $t \rightarrow \infty$. A lei de controle $u(t)$ que é enviada à planta é obtida integrando duas vezes o sinal ω definido na equação 2.51. A Figura 2.2.2 mostra o controlador de modelo interno para referência tipo rampa [17].

Figura 2.2.2 – Controlador de modelo interno para referência tipo rampa.



Fonte: Autor

No seguinte capítulo será explicado como a teoria de controlador de modelo interno é utilizado para desenvolver um algoritmo GPC para rastreamento de referências tipo rampa.

3 GPC para Rastreamento de Referência Tipo Rampa

3.1 GPC para Rastreamento de Referência Tipo Rampa

Neste capítulo será apresentado o desenvolvimento matemático do sistema GPC proposto, o qual está baseado principalmente no desenvolvimento de um novo modelo aumentado para a predição da resposta do sistema. O modelo aumentado proposto está baseado no controlador de modelo interno em tempo contínuo descrito em 2.2.2 e o operador de diferença de segunda ordem. A diferença para o GPC convencional, é que a saída do modelo aumentado proposto é o erro de rastreamento, no lugar da saída da planta.

3.1.1 Operador de Diferença de Segunda Ordem

Seja $a(k)$ uma variável (escalar ou vetor) definida no tempo discreto. Define-se o operador de diferença de primeira ordem como [18]:

$$\Delta a(k) = a(k) - a(k - 1) \quad (3.1)$$

O operador Δ foi utilizado em 2.1.1 para definir o modelo aumentado do GPC convencional. Este operador seria o equivalente discreto de uma derivada de primeira ordem. Da mesma maneira que existem derivadas de ordem superior, existem também operadores de diferenças de ordem maior. A equação 3.2 define o operador de diferença de segunda ordem (Δ^2) [18]:

$$\Delta^2 a(k) = \Delta[\Delta a(k)] = \Delta a(k) - \Delta a(k - 1) \quad (3.2)$$

Este operador é linear [18]: $\Delta^2[Ma(k) + Nb(k)] = M\Delta^2 a(k) + N\Delta^2 b(k)$, sendo M e N constantes ou matrizes de dimensão adequada tais que $Ma(k) + Nb(k)$ existe.

3.1.2 Preliminares

Nesta seção, diferentes propriedades sobre a referência tipo rampa serão demonstradas. Tais propriedades serão utilizadas na elaboração do modelo aumentado proposto.

Vai ser usada a equação 2.1, pois ela descreve o modelo discreto de ordem n de uma planta com uma entrada e uma saída (SISO), onde $u(k)$ é a entrada da planta, $y(k)$ é o processo de saída do sistema, e $x_m(k)$ é o vetor de estado da planta.

A Figura 3.1.1 mostra o comportamento de um sinal de entrada de referência do tipo rampa. Será considerado no sinal da referência de entrada que a angulação da rampa seja constante. Logo o sinal do tipo rampa pode ser descrito por:

$$r(k) = \alpha k + \beta \quad (3.3)$$

Portanto:

$$r(k+1) - r(k) = r(k) - r(k-1) = \alpha \quad (3.4)$$

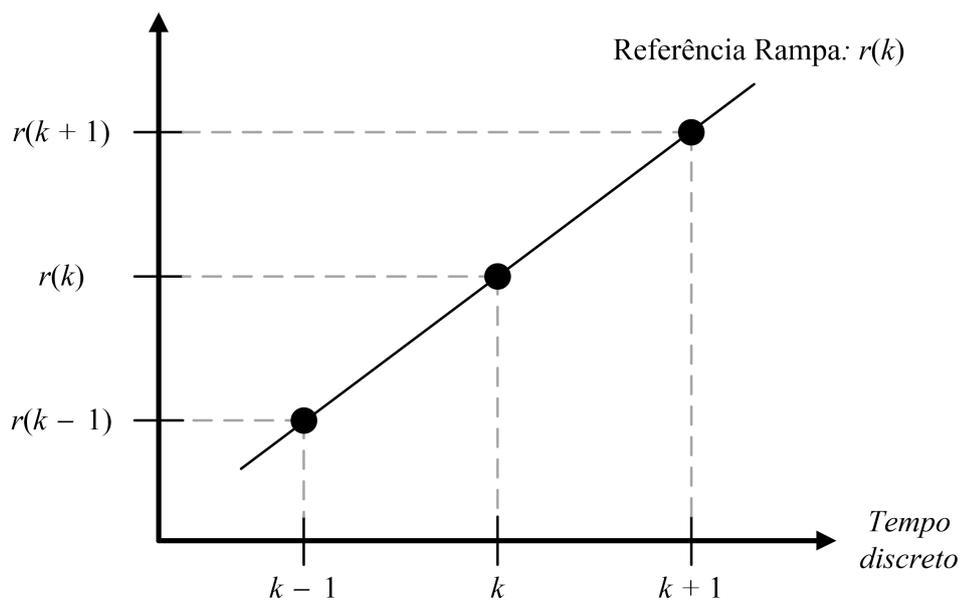
A diferença de outras técnicas de rastreamento de referência rampa (como em [3]), o conhecimento do valor do termo α na equação 3.4 não é necessário no algoritmo proposto. A partir da equação 3.4, tem-se:

$$\Delta r(k+1) = \Delta r(k) \quad (3.5)$$

A equação 3.6 pode ser obtida isolando o termo $r(k+1)$ da equação 3.4.

$$r(k+1) = 2r(k) - r(k-1) \quad (3.6)$$

Figura 3.1.1 – Sinal de Entrada da Referência do tipo Rampa



Fonte: Autor

O erro no instante k é definido por:

$$e(k) = r(k) - y(k) \quad (3.7)$$

E a primeira diferença do erro no instante k pode ser escrita na seguinte forma:

$$\Delta e(k) = e(k) - e(k-1) \quad (3.8)$$

Adicionalmente, aplicando o operador Δ a ambos os lados da equação 3.7, e considerando que este operador é linear, obtém-se:

$$\Delta e(k) = \Delta r(k) - \Delta y(k) \quad (3.9)$$

3.1.3 Modelo Aumentado

A estrutura do modelo aumentado e da lei de controle está baseado no controlador de modelo interno (CMI) para rastreamento de referência rampa descrito no capítulo anterior. Naquele controlador, os sinais que compõem o vetor de estado do modelo são o erro, a derivada do erro e a segunda derivada do vetor de estados da planta. Fazendo uma analogia com o controlador de modelagem interno previamente citado, o modelo aumentado será composto pelo erro $e(k)$, a diferença de primeira ordem do erro discreto $\Delta e(k)$, e a diferença de segunda ordem do vetor de estado da planta discreta $\Delta^2 x_m(k)$. Esta analogia, baseada no fato que o operador de diferença é um equivalente discreto da operação de derivação, é apresentado nas equações 3.10 e 3.11.

$$\dot{e} \rightarrow \Delta e(k) \quad (3.10)$$

$$z = \ddot{x} \rightarrow \Delta^2 x_m(k) \quad (3.11)$$

O operador $\Delta^2 x_m(k)$ pode ser descrito pela equação 3.12. Os operadores $\Delta e(k)$ e $e(k)$ já foram definidos pelas equações 3.8 e 3.7.

$$\Delta^2 x_m(k) = \Delta x_m(k) - \Delta x_m(k-1) \quad (3.12)$$

Baseados nas equações 3.12, 3.7 e 3.8, as variáveis que compõem o vetor de estados do modelo aumentado proposto são definidos da seguinte maneira:

$$\begin{bmatrix} \Delta^2 x_m(k) \\ \Delta e(k) \\ e(k) \end{bmatrix} = \begin{bmatrix} \Delta x_m(k) - \Delta x_m(k-1) \\ e(k) - e(k-1) \\ r(k) - y(k) \end{bmatrix} \quad (3.13)$$

sendo 3.14 a trajetória da diferença da variável de estado Δx_m no instante atual (k) até a sua primeira trajetória no futuro ($k + 1$).

$$\begin{aligned}\Delta x_m(k) &= x_m(k) - x_m(k-1) \\ \Delta x_m(k+1) &= x_m(k+1) - x_m(k)\end{aligned}\quad (3.14)$$

A partir da equação 2.1, as variáveis $x_m(k-1)$, $x_m(k)$, e $x_m(k+1)$ podem ser expressados da seguinte maneira:

$$\begin{aligned}x_m(k-1) &= A_m x_m(k-2) + B_m u(k-2) \\ x_m(k) &= A_m x_m(k-1) + B_m u(k-1) \\ x_m(k+1) &= A_m x_m(k) + B_m u(k)\end{aligned}\quad (3.15)$$

As equações 3.16 e 3.17 foram definidas a partir das equações 3.15 e 3.14:

$$\begin{aligned}\Delta x_m(k) &= [(A_m x_m(k-1) + B_m u(k-1)) + (A_m x_m(k-2) + B_m u(k-2))] \\ \Delta x_m(k) &= [(A_m x_m(k-1) - A_m x_m(k-2)) + (B_m u(k-1) - B_m u(k-2))] \\ \Delta x_m(k) &= [A_m \Delta x_m(k-1) + B_m \Delta u(k-1)]\end{aligned}\quad (3.16)$$

$$\begin{aligned}\Delta x_m(k+1) &= [(A_m x_m(k) + B_m u(k)) + (A_m x_m(k-1) + B_m u(k-1))] \\ \Delta x_m(k+1) &= [(A_m x_m(k) - A_m x_m(k-1)) + (B_m u(k) - B_m u(k-1))] \\ \Delta x_m(k+1) &= [A_m \Delta x_m(k) + B_m \Delta u(k)]\end{aligned}\quad (3.17)$$

Logo, pode ser descrito uma nova equação para o operador $\Delta^2 x_m(k+1)$ pela substituição das equações 3.16 e 3.17 na equação 3.13:

$$\begin{aligned}\Delta^2 x_m(k+1) &= [(A_m \Delta x_m(k) + B_m \Delta u(k)) + (A_m \Delta x_m(k-1) + B_m \Delta u(k-1))] \\ \Delta^2 x_m(k+1) &= [(A_m \Delta x_m(k) - A_m \Delta x_m(k-1)) + (B_m \Delta u(k) - B_m \Delta u(k-1))] \\ \Delta^2 x_m(k+1) &= [A_m \Delta^2 x_m(k) + B_m \Delta^2 u(k)]\end{aligned}\quad (3.18)$$

Sendo que o operador de diferença é linear, o sinal $\Delta e(k+1)$ pode ser encontrado após de aplicar o operador de diferença de primeira ordem (Δ) na equação 3.13:

$$\begin{aligned}\Delta e(k+1) &= \Delta[r(k+1) - y(k+1)] \\ \Delta e(k+1) &= \Delta r(k+1) - \Delta y(k+1)\end{aligned}\quad (3.19)$$

Note que na equação 3.19 existe a diferença de primeira ordem da saída futura $\Delta y(k+1)$, a qual deve ser calculada. Sendo que $y(k+1) = C_m x_m(k+1)$ e $y(k) = C_m x_m(k)$, a saída futura $y(k+1)$ pode-se expressar como indicado na equação 3.20:

$$y(k+1) = C_m (x_m(k+1) - x_m(k)) + y(k)\quad (3.20)$$

Aplicando o operador de diferença de primeira ordem a ambos os lados da equação 3.20, e considerando que este operador é linear, obtém-se:

$$\begin{aligned}\Delta y(k+1) &= \Delta[C_m \Delta x_m(k+1) + y(k)] \\ \Delta y(k+1) &= \Delta y(k) + C_m \Delta^2 x_m(k+1)\end{aligned}\quad (3.21)$$

Substituindo $\Delta y(k+1)$ da equação 3.21 na equação 3.19, tem-se:

$$\Delta e(k+1) = \Delta r(k+1) - \Delta y(k) + C_m \Delta^2 x_m(k+1) \quad (3.22)$$

Conforme foi definido pela equação 3.5, $\Delta r(k+1) = \Delta r(k)$. Assim, a equação 3.19 pode ser representada da seguinte forma:

$$\Delta e(k+1) = \Delta r(k) - \Delta y(k) + C_m \Delta^2 x_m(k+1) \quad (3.23)$$

Uma nova equação é encontrada 3.24 pela substituição das equações 3.9 e 3.18 na equação 3.23:

$$\begin{aligned}\Delta e(k+1) &= \Delta r(k) - [\Delta y(k) + C_m(A_m \Delta^2 x_m(k) + B_m \Delta^2 u(k))] \\ \Delta e(k+1) &= \Delta e(k) - C_m A_m \Delta^2 x_m(k) - C_m B_m \Delta^2 u(k)\end{aligned}\quad (3.24)$$

Devido à definição do operador de diferença na equação 3.25, o erro no instante futuro $e(k+1)$ pode ser encontrado segundo a equação 3.26:

$$\Delta e(k+1) = e(k+1) - e(k) \quad (3.25)$$

$$e(k+1) = \Delta e(k+1) + e(k) \quad (3.26)$$

Substituindo o termo $\Delta e(k+1)$ da equação 3.24 na equação 3.26, tem-se:

$$\begin{aligned}e(k+1) &= (\Delta e(k) - C_m A_m \Delta^2 x_m(k) - C_m B_m \Delta^2 u(k)) + e(k) \\ e(k+1) &= e(k) + \Delta e(k) - C_m A_m \Delta^2 x_m(k) - C_m B_m \Delta^2 u(k)\end{aligned}\quad (3.27)$$

Colocando as equações 3.18, 3.24 e 3.27 juntas, é possível definir o modelo aumentado proposto da seguinte maneira:

$$\begin{aligned}x(k+1) &= Ax(k) + B \Delta^2 u(k) \\ y(k) &= Cx(k) = e(k)\end{aligned}\quad (3.28)$$

onde

$$x(k) = \begin{bmatrix} \Delta^2 x_m(k) \\ \Delta e(k) \\ e(k) \end{bmatrix}, A = \begin{bmatrix} A_m & o_o^T & o_o^T \\ -C_m A_m & 1 & o_o^T \\ -C_m A_m & 1 & 1 \end{bmatrix}, B = \begin{bmatrix} B_m \\ -C_m B_m \\ -C_m B_m \end{bmatrix} \quad (3.29)$$

$$C = \begin{bmatrix} o_o & 0 & 1 \end{bmatrix}$$

A equação 3.28 define o modelo aumentado proposto que será utilizado na predição das respostas da planta. A variável o_o é um vetor composto por zeros, utilizado para completar as matrizes A e C . Pode-se observar que a entrada do modelo aumentado proposto é $\Delta^2 u(k)$, enquanto a saída é o erro de rastreamento $e(k)$, no lugar da saída da planta (como no GPC convencional). Pode-se definir uma analogia entre a lei de controle do controlador de modelo interno (ω) com a entrada do modelo aumentado proposto:

$$\omega \rightarrow \Delta^2 u(k) \quad (3.30)$$

3.1.4 Cálculo da Lei de Controle

Seja $e(k_i + l|k_i)$ a previsão do erro no instante $k_i + l$ ($l = 1, 2, \dots, N_p$), de acordo com o valor de $x(k_i)$. Seja $\Delta^2 u(k_i + m)$ a entrada do modelo proposto no instante $k_i + m$, $m = 0, 1, \dots, N_c - 1$. Tanto o modelo do GPC convencional (equação 2.13) quanto o modelo do GPC proposto (equação 3.29) são baseados na equação de estado $x(k_i + 1) = Ax(k_i) + Bu_{ma}(k_i)$, sendo $u_{ma}(k_i)$ a entrada do modelo aumentado. No GPC convencional, $u_{ma}(k_i) = \Delta u(k_i)$; enquanto que no modelo aumentado proposto $u_{ma}(k_i) = \Delta^2 u(k_i)$. Por outro lado, as previsões dos sinais do controle futuro e o vetor de saída desenvolvidos nas equações 2.16, 2.17, 2.18, 2.19 não dependem da estrutura interna dos sinais nem das matrizes que compõem o modelo aumentado. Portanto, é possível usar a metodologia aplicada na obtenção da equação 2.18 para prever as saídas do modelo proposto, ou seja, prever o erro de rastreamento:

$$Y_a = Fx(k_i) + \phi U_a \quad (3.31)$$

onde as variáveis F e ϕ são definidas em 2.21, enquanto que as variáveis Y_a e U_a tem a mesma estrutura que as variáveis Y e ΔU apresentada em 2.21, mas substituindo $\Delta u(k)$ e $y(k)$ por $\Delta^2 u(k)$ e $e(k)$ respectivamente:

$$\begin{aligned} u(k_i + m) &\rightarrow \Delta^2 u(k_i + m) \\ y(k_i + l|k) &\rightarrow (k_i + l|k) \end{aligned} \quad (3.32)$$

Portanto:

$$\begin{aligned} Y_a &= \begin{bmatrix} e(k_i + 1|k_i) & e(k_i + 2|k_i) & \cdots & e(k_i + N_p|k_i) \end{bmatrix} \\ U_a &= \begin{bmatrix} \Delta^2 u(k_i) & \Delta^2 u(k_i + 1) & \cdots & \Delta^2 u(k_i + N_c - 1) \end{bmatrix} \end{aligned} \quad (3.33)$$

O valor desejado do erro da planta sempre será zero. Portanto, o vetor de referência R_s para Y_2 é um conjunto de zeros ($R_s = 0$), ou seja, a referência da janela de otimização para o modelo aumentado proposto será uma constante. Definindo então $R = 0$, $Y = Y_a$ e $\Delta U = U_a$ nas equações 2.27 e 2.29, é possível aplicar a otimização e estratégias do horizonte de recuo usada no GPC convencional para obter o valor de $\Delta^2 u(k_i)$ (o primeiro elemento de U_a):

$$J = (0 - Y)^T(0 - Y) + U_a^T \bar{R}_a U_a \quad (3.34)$$

$$U_a = (\phi^T \phi + \bar{R}_a)^{-1} \phi^T (0 - Fx(k_i)) \quad (3.35)$$

$$\Delta^2 u(k_i) = - \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}}_{K_2} \underbrace{(\phi^T \phi + \bar{R}_a)^{-1} \phi^T Fx(k_i)}_{N_C} \quad (3.36)$$

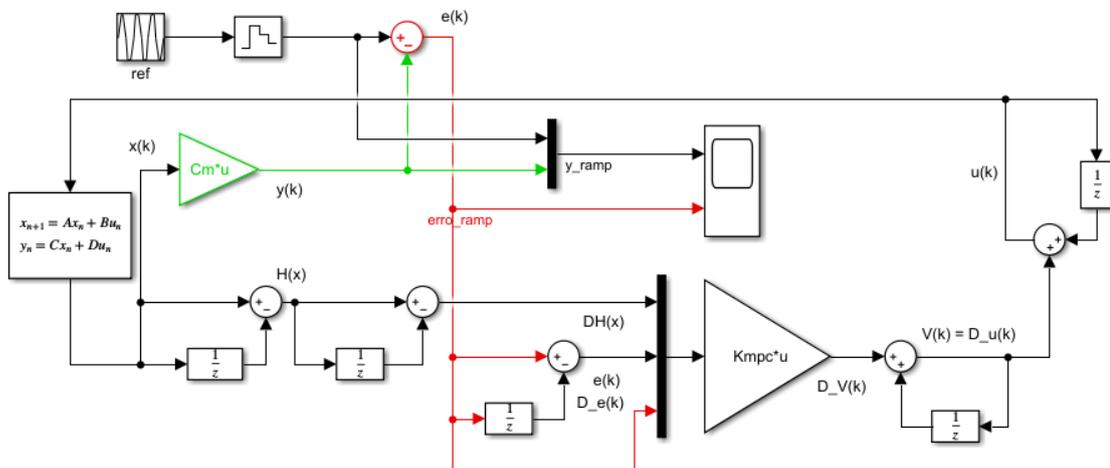
$$\bar{R}_a = r_w I_{N_C \times N_C} \quad (3.37)$$

O vetor K_2 é o vetor ganhos de realimentação. A lei de controle da planta é obtida pela dupla integração de $\Delta^2 u(k_i)$:

$$\begin{aligned} \Delta u(k_i) &= \Delta^2 u(k_i) + \Delta u(k_i - 1) \\ u(k_i) &= \Delta u(k_i) + u(k_i - 1) \end{aligned} \quad (3.38)$$

A Figura 3.1.2 mostra o diagrama de simulação do sistema de rastreamento rampa, enquanto que a Figura 3.1.3 mostra o esquemático do algoritmo GPC proposto. Sendo que

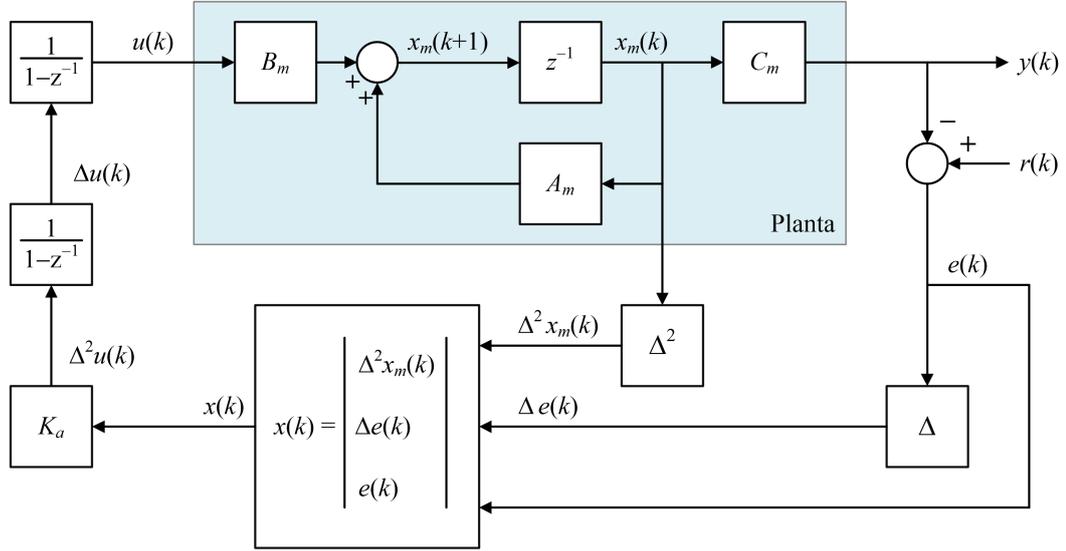
Figura 3.1.2 – Diagrama em bloco do SOD-GPC



Fonte: Autor

o operador de diferença de segunda ordem foi utilizado no modelo aumentado, o algoritmo proposto é chamada de *SOD-GPC* (*Second Order Difference GPC* em Inglês).

Figura 3.1.3 – Sistema SOD-GPC



Fonte: Autor

3.1.5 Função de Transferência do Modelo Aumentado Proposto

Para simplificar a notação, $M \in \mathfrak{R}^{m \times n}$ indica que M é uma matriz $m \times n$.

Considerar que o modelo da planta seja de ordem n . Seja $\sigma = n + 2$. Observe que $A \in \mathfrak{R}^{\sigma \times \sigma}$, $B \in \mathfrak{R}^{\sigma \times \sigma}$ e $C \in \mathfrak{R}^{\sigma \times \sigma}$. Seja $I \in \mathfrak{R}^{\sigma \times \sigma}$ e $I_a \in \mathfrak{R}^{\sigma \times \sigma}$ matrizes identidade. Segundo as equações 2.1 e 3.29, as funções de transferência discreta da planta e do modelo aumentado proposto são $G(z)$ e $G_o(z)$, respectivamente:

$$\begin{aligned} G(z) &= C_m(zI - A_m)^{-1}B_m \\ G_o(z) &= C(zI_a - A)^{-1}B \end{aligned} \quad (3.39)$$

Note que $I_a = \text{diag}(I, 1, 1)$. Definem-se as seguintes matrizes:

$$\begin{aligned} zI_a - A &= \begin{bmatrix} zI - A_m & 0 \\ -TC_m A_m & L \end{bmatrix}, \quad B = \begin{bmatrix} B_m \\ TC_m B_m \end{bmatrix}, \quad T = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \\ L &= \begin{bmatrix} z - 1 & 0 \\ -1 & z - 1 \end{bmatrix} \end{aligned} \quad (3.40)$$

Usando álgebra de sub-matrizes, e usando manipulações matemáticas, a equação 3.39 pode ser expressa como indicado nas equações 3.41:

$$G_o(z) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} zI - A_m & 0 \\ -TC_m A_m & L \end{bmatrix} \begin{bmatrix} B_m \\ TC_m B_m \end{bmatrix} \quad (3.41)$$

Observar que $A_m(zI - A_m)^{-1}B_m + B_m = [zI - (zI - A_m)](zI - A_m)^{-1}B_m + B_m = z(zI - A_m)^{-1}B_m - B_m + B_m$. Portanto:

$$A_m(zI - A_m)^{-1}B_m = z(zI - A_m)^{-1}B_m \quad (3.42)$$

Substituindo as equações 3.39, 3.40 e 3.42 na equação 3.41, é possível obter a relação entre $G_o(z)$ e $G(z)$:

$$\begin{aligned} G_o(z) &= \begin{bmatrix} 0 & 1 \end{bmatrix} L^{-1}TC_m[z(zI - A_m)^{-1}B_m] \\ G_o(z) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z-1 & 0 \\ z-1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ -1 \end{bmatrix} [C_m(zI - A_m)^{-1}B_m] \\ G_o &= - \left[\frac{z}{z-1} \right]^2 G(z) \end{aligned} \quad (3.43)$$

O termo $\frac{z}{(z-1)}$ que aparece na equação 3.43 é um integrador discreto. Portanto, a equação 3.43 indica que o modelo aumentado proposto tem 2 integradores embutidos. Segundo o princípio de modelagem interna [17], o modelo proposto pode efetuar rastreamento de referências tipo degrau e rampa com um erro que tende assintoticamente a zero. Adicionalmente, o sistema GPC proposto requer que a planta seja controlável e que $G(z)$ não tenha zeros em $z = 1$.

3.2 Discretização de Plantas Definidas em Tempo Contínuo

Muitas plantas a serem controladas são definidas por modelos em tempo contínuo (funções de transferência ou modelos em espaço de estados). Porém, o algoritmo GPC trabalha com modelos definidos em tempo discreto. Portanto, é necessário discretizar a planta para obter o modelo em tempo discreto utilizado no GPC.

Para a compreensão do processo de discretização, é necessário o entendimento sobre a Transformada Z. Esta transformada é o equivalente da transformada de Laplace no domínio discreto. No domínio laplaciano, as equações diferenciais são transformadas em equações algébricas. De forma análoga, a Transformada Z converte equações de diferenças em equações algébricas

Seja $x(k)$ um sinal real definida no tempo discreto, onde k representa a k -ésima amostra. A equação 3.44 define a transformada Z de $x(k)$, $X(z)$ [19]:

$$X(z) = \sum_{k=-\infty}^{+\infty} x(k)z^{-k} \quad (3.44)$$

onde $z = re^{j\Omega}$ é um numero complexo.

Define-se a transformada Z inversa da seguinte maneira:

$$x(k) = \frac{1}{2\pi j} \oint X(z)z^{k-1}dz \quad (3.45)$$

A existência de $X(z)$ depende de que a somatoria definida na equação 3.44 converga. Para garantir a convergência, deve-se cumprir a seguinte condição:

$$\sum_{k=-\infty}^{+\infty} |x(k)z^{-k}| < +\infty \quad (3.46)$$

Pode-se demonstrar que [19]:

$$|x(k)z^{-k}| = |x(k)r^{-k}| \quad (3.47)$$

sendo r a magnitude de z . Substituindo a equação 3.47 em 3.46, tem-se:

$$\sum_{k=-\infty}^{+\infty} |x(k)r^{-k}| < +\infty \quad (3.48)$$

Os valores de r para a qual a condição definida na equação 3.48 é satisfeita é denominada de região de convergência.

Considere que um sinal $x(k)$ está sendo aplicado a um sistema linear que possui uma resposta ao impulso discreto unitário $h(k)$. A saída do sistema, $y(k)$, é dada por

$$y(k) = x(k) * h(k) = \sum_{j=-\infty}^{+\infty} h(j)x(k-j) \quad (3.49)$$

onde $*$ denota convolução discreta. Aplicando a transformada Z na equação 3.49, obtém-se:

$$Y(z) = X(z) * H(z) \quad (3.50)$$

onde $Y(z)$, $X(z)$ e $H(z)$ são as transformadas Z de $y(k)$, $h(k)$ e $x(k)$, respectivamente. Define-se a função de transferência discreta $H(z)$ da seguinte maneira:

$$H(z) = \frac{Y(z)}{X(z)} \quad (3.51)$$

A função de transferência $H(z)$ pode ser descrita, com o auxílio da equação 3.44, como uma razão de polinômios:

$$H(z) = \frac{b_0 + b_1z^{-1} + \dots b_Mz^{-M}}{1 + a_1z^{-1} + \dots b_Nz^{-N}} \quad (3.52)$$

Substituindo a equação 3.51 em 3.52, obtém-se:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1z^{-1} + \dots b_Mz^{-M}}{1 + a_1z^{-1} + \dots a_Nz^{-N}} \quad (3.53)$$

Portanto:

$$Y(z)[1 + a_1z^{-1} + \dots a_Nz^{-N}] = X(z)[b_0 + b_1z^{-1} + \dots b_Mz^{-M}] \quad (3.54)$$

A transformada Z inversa de $z^{-m}X(z)$ é $x(k-m)$ [19]. Aplicando esta propriedade na equação 3.52, obtém-se as seguintes equações de diferenças:

$$y(k) + a_1y(k-1) + \dots + a_Ny(k-N) = b_0x(k) + b_1x(k-1) + \dots + b_Mx(k-M) \quad (3.55)$$

$$y(k) = b_0x(k) + b_1x(k-1) + \dots + b_Mx(k-M) - [a_1y(k-1) + \dots + a_Ny(k-N)] \quad (3.56)$$

Um sistema linear representado através da equação de diferenças definida na equação 3.56 pode ser implementado em um processador digital (DSP ou FPGA).

Para obter a função de transferência discreta ($G(z)$) a partir de uma planta $G(s)$ definida no tempo contínuo, podem-se usar técnicas de discretização: aproximar a variável laplaciana (s) por uma função definida em z . Seja t_s o tempo de amostragem dos sinais. As técnicas de discretização mais populares são [19]:

$$\text{Euler em atraso: } s \approx \frac{1 - z^{-1}}{t_s} \quad (3.57)$$

$$\text{Euler em avanço: } s \approx \frac{1 - z^{-1}}{t_s z^{-1}} \quad (3.58)$$

$$\text{Tustin: } s \approx \frac{2}{t_s} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (3.59)$$

Pela simplicidade, nesta dissertação foi utilizado a técnica Euler em atraso na discretização de sistemas contínuos.

A equação de diferenças que descreve o comportamento da planta é obtida a partir de $G(z)$. Diferentes modelos em espaço de estado podem ser obtidos a partir de uma equação de diferenças, dependendo da definição das variáveis de estado. Geralmente, as variáveis de estado são escolhidas como grandezas físicas que podem ser adquiridas. Por exemplo, utilizando a técnica da forma canônicas controlável, o modelo de $H(z)$ na equação 3.51 é dado por [20]:

$$x(k+1) = Ax(k) + Bu(k) \quad (3.60)$$

$$y(k) = Cx(k) + Du(k) \quad (3.61)$$

onde

$$A = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{L-1} & a_L \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad C = [\beta_1 \quad \beta_2 \quad \dots \quad \beta_L], \quad D = b_0$$

onde $L = \max(N, M)$ e $\beta_i = b_i - b_0 a_i$, $i = 1, \dots, L$. Softwares como MATLAB e OCTAVE possuem instruções que permitem obter o modelo em espaço de estados discretos a partir de funções de transferência contínuas.

3.3 Sistema ATO baseado em GPC

3.3.1 Resolver

O resolver é um sensor de posição angular muito utilizado em aplicações como veículos elétricos, antenas, CNC e robótica [21–23]. Este sensor pode suportar maiores temperaturas e vibrações que o sensor tipo encoder. A Figura 3.3.1 mostra a estrutura de um sensor resolver. Este sensor recebe um sinal senoidal de excitação de alta frequência (de 1 kHz a 10 kHz):

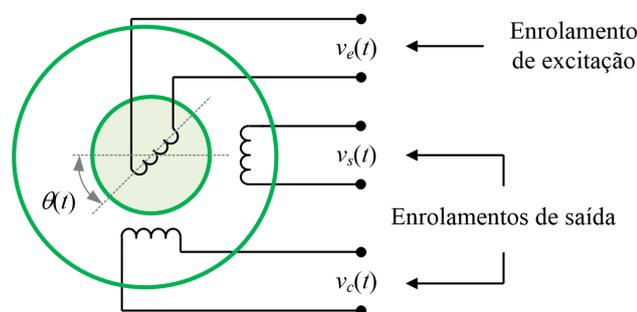
$$v_e(t) = a_r \cos(\omega_r t) \quad (3.62)$$

onde a_r é a amplitude de excitação, $\omega_r = 2\pi f_r$, enquanto que f_r é a frequência de excitação. A excitação induz duas tensões, $v_s(t)$ e $v_c(t)$, nos enrolamentos de saída do sensor [21–23]:

$$\begin{aligned} v_s(t) &= k_r v_e(t) \sin(\theta(t)) \\ v_c(t) &= k_r v_e(t) \cos(\theta(t)) \end{aligned} \quad (3.63)$$

onde k_r é um fator de transformação (típicamente $k = 0.5$). O observador chamado conversorresolver-digital (*Resolver-to-Digital Converter*, RDC em Inglês) é utilizado para calcular a posição angular a partir dos sinais do sensor resolver. Diferentes RDCs foram propostos [21–33]. Os RDCs podem ser circuitos de hardware externos. Porém, atualmente, os RDCs são implementados como algoritmos executados nos processadores digitais onde se encontram os algoritmos de controle de máquinas.

Figura 3.3.1 – Estrutura do sensor resolver



Fonte: Autor

3.3.2 Sistema ATO proposto

A Figura 3.3.2 mostra a estrutura de um ATO baseado no GPC proposto. Para simplificar a notação, o algoritmo proposto foi chamado de GPC de Diferença de Segunda

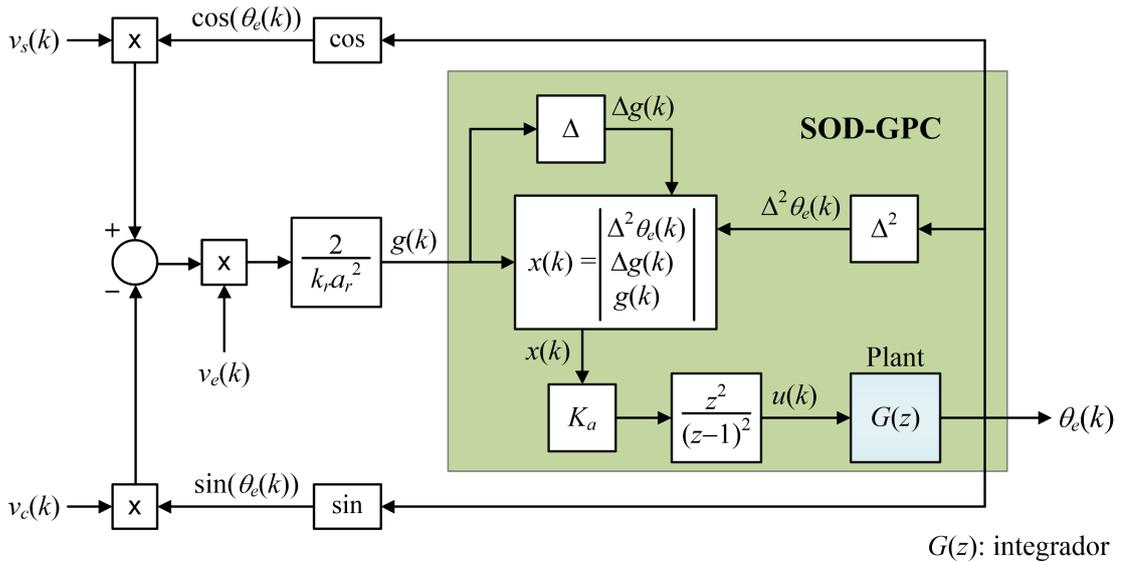
Ordem (*Second Order Difference Generalized Predictive Control - SOD-GPC*). Considera-se que o sinal de excitação é gerado pelo RDC. Então, o valor deste sinal é conhecido. Considera-se $t = k.t_s$, onde t é o tempo discretizado, t_s é o tempo de amostragem, enquanto k refere-se a quantidade de amostras. O ângulo atual e o estimado podem ser definidos por $\theta(k) = \theta(kt_s)$ e $\theta_e(k) = \theta_e(kt_s)$. Assim, os sinais $v_e(k)$, $v_s(k)$ e $v_c(k)$ do resolver podem ser descritos na seguinte forma:

$$\begin{aligned} v_e(k) &= a_r \cos(\omega_r t_s k) = a_r \cos(\omega_{rs} k) \\ v_s(k) &= k_r v_e(k) \sin(\theta(k)) \\ v_c(k) &= k_r v_e(k) \cos(\theta(k)) \end{aligned} \quad (3.64)$$

onde $\omega_{rs} = \omega_r t_s$. O termo $g(k)$ pode ser encontrado de acordo com o esquemático da Figura 3.3.2 e pela definição dos sinais $v_e(k)$, $v_s(k)$ e $v_c(k)$:

$$\begin{aligned} g(k) &= \frac{2}{k_r a_r^2} [v_s(k) \cos(\theta_e(k)) - v_c(k) \sin(\theta_e(k))] v_e(k) \\ g(k) &= \frac{2}{a_r^2} [\sin(\theta(k)) \cos(\theta_e(k)) - \cos(\theta(k)) \sin(\theta_e(k))] v_e^2(k) \\ g(k) &= 2 \sin(\theta(k) - \theta_e(k)) \cos^2(\omega_{rs} k) \\ g(k) &= \sin(\theta(k) - \theta_e(k)) [1 + \cos(2\omega_{rs} k)] \\ g(k) &= \sin(e_\theta(k)) + \sin(e_\theta(k)) \cos(2\omega_{rs} k), \end{aligned} \quad (3.65)$$

Figura 3.3.2 – Sistema ATO baseado no SOD-GPC



Fonte: Autor

Em que o $e_\theta(k) = \theta(k) - \theta_e(k)$ é o erro de estimação do ângulo. Assumindo que $e_\theta(k) \approx 0$, então $\sin(e_\theta(k)) \approx e_\theta(k)$. Logo a equação 3.66 pode ser reescrita da seguinte

forma:

$$g(k) \approx e_\theta(k) + e_\theta(k)\cos(2\omega_{rs}k). \quad (3.66)$$

Seja $h(k)$ a resposta ao impulso do ATO. Foi possível aplicar o princípio de superposição, pois o SOD-GPC é um sistema linear. Esta aplicação é necessária para analisar a resposta do ATO: A posição angular estimada (saída do ATO) é uma soma das respostas produzidas pelo $e_\theta(k)$ e $e_\theta(k)\cos(2\omega_{rs}k)$:

$$\begin{aligned} \theta_e(k) &= h(k) * g(k) \\ \theta_e(k) &= h(k) * [e_\theta(k) + e_\theta(k)\cos(2\omega_{rs}k)] \\ \theta_e(k) &= h(k) * e_\theta(k) + h(k) * [e_\theta(k)\cos(2\omega_{rs}k)], \end{aligned} \quad (3.67)$$

onde $*$ significa convolução. Através da propriedade de deslocamento da frequência, é possível provar que o termo $e_\theta(k)\cos(2\omega_{rs}k)$ é um sinal de alta frequência, enquanto o ATO atua como um filtro passa-baixa que rejeita sinais de alta frequência [33]. Como resultado, o efeito do sinal $e_\theta(k)\cos(2\omega_{rs}k)$ na estimação do ângulo poderá ser rejeitada pelo ATO:

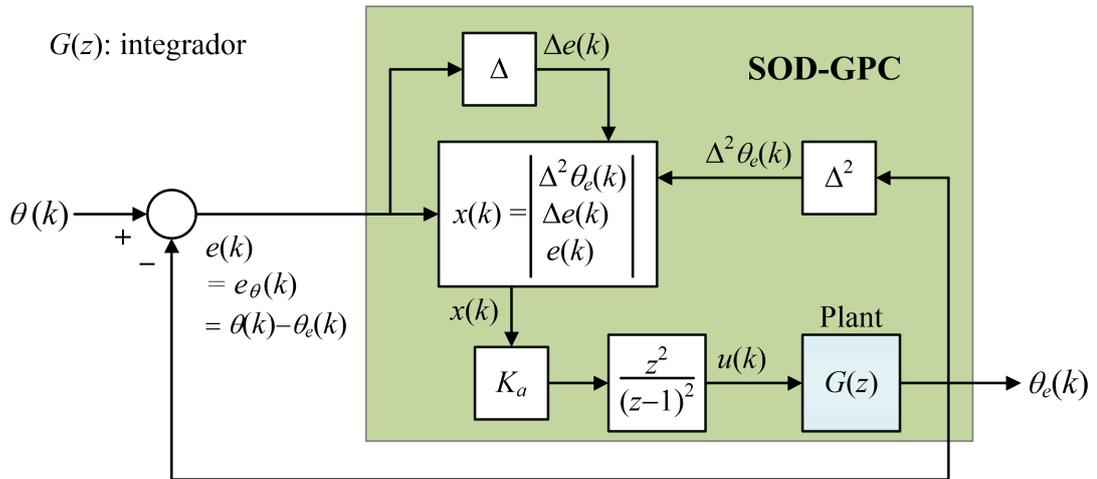
$$h(k) * [e_\theta(k)\cos(2\omega_{rs}k)] \approx 0. \quad (3.68)$$

Substituindo a equação 3.68 na equação 3.67:

$$\theta_e(k) \approx h(k) * e_\theta(k). \quad (3.69)$$

A equação 3.69 afirma que o ângulo estimado depende principalmente do termo $e_\theta(k)$. Em consequência, o ATO pode ser projetado apenas considerando a existência de $e_\theta(k)$. Assim, o ATO pode ser aproximado como o sistema de malha fechada da Figura 3.3.3, onde a referência é $\theta(k)$, a saída é $\theta_e(k)$ e o SOD-GPC é o controlador que reduz o erro $e_\theta(k)$.

Figura 3.3.3 – Diagrama de bloco simplificado do sistema ATO para o SOD-GPC



Fonte: Autor

4 Resultados

Para testar o desempenho do algoritmo GPC proposto, foram utilizados em duas aplicações diferentes:

- Caso 1: No controle da planta contínua $G(s) = \frac{53.48s + 5347.59}{s^2 + 253.5s + 10695.18}$, a qual será discretizada segundo as técnicas de discretização da seção 3.2.
- Caso 2: No desenvolvimento de um observador de rastreamento de ângulo (*angle tracking observer*, ATO em Inglês), segundo o algoritmo da subseção 3.3.2.

Testes de simulação e experimentais foram efetuados para cada caso

4.1 Controle de Planta Contínua $G(s)$

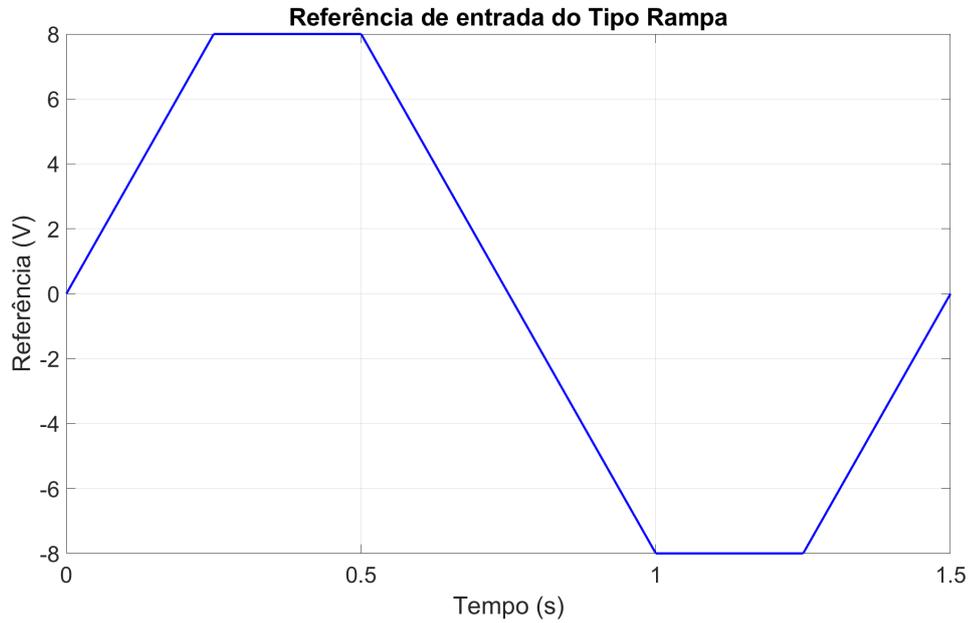
O algoritmo GPC proposto foi aplicado para controlar a saída da planta com função de transferência contínua $G(s) = \frac{53.48s + 5347.59}{s^2 + 253.5s + 10695.18}$. Para aplicar o algoritmo proposto, esta planta foi discretizada utilizando o MATLAB, segundo a fundamentação teórica descrita na seção 3.1. O tempo de amostragem utilizado na discretização é de $t_s = 0,001s$. Desta maneira, o modelo em espaço de estado discreto da planta é descrito pela equação 4.1:

$$A_m = \begin{bmatrix} 0.9479 & -0.0472 \\ 0 & 0.8187 \end{bmatrix}, B_m = \begin{bmatrix} 0.0496 \\ 0.0906 \end{bmatrix} \quad (4.1)$$

$$C_m = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

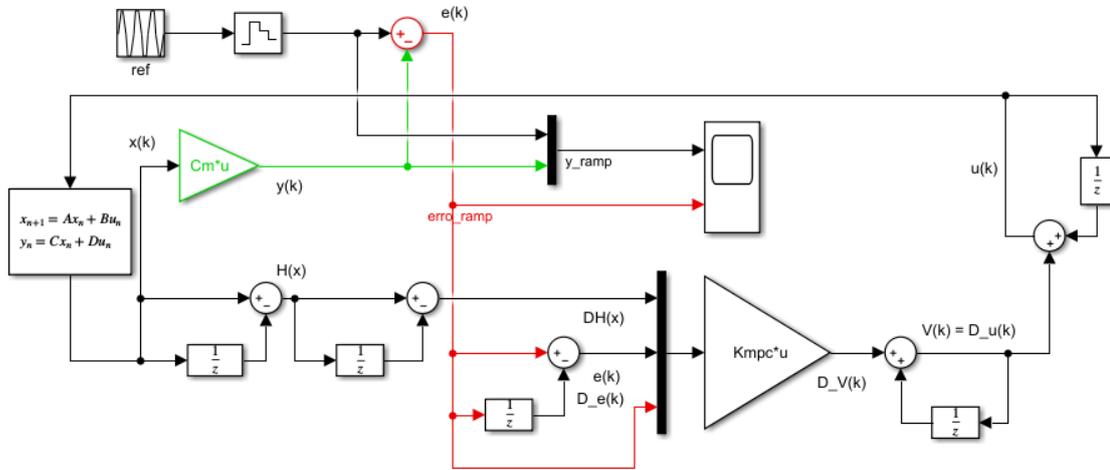
A Figura 4.1.1 mostra o sinal de referência utilizado nos ensaios de simulação e experimentais. Na Figura 4.1.2 mostra-se o diagrama de simulação do teste efetuado.

Figura 4.1.1 – Referência de entrada do Tipo Rampa



Fonte: Autor

Figura 4.1.2 – Diagrama em bloco do SOD-GPC



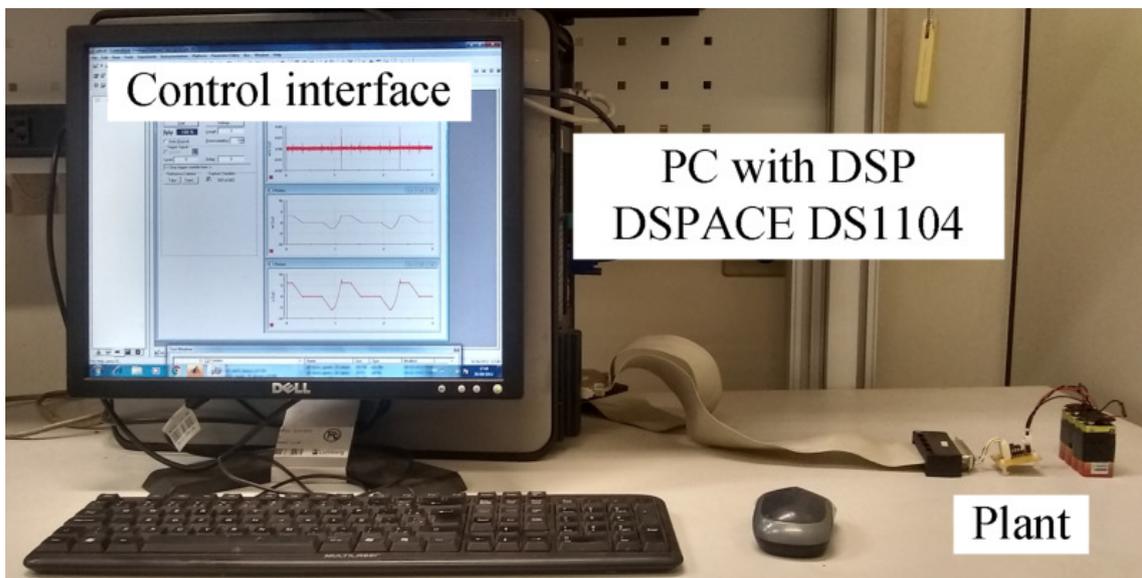
Fonte: Autor

O controlador GPC proposto (considerando diferentes valores de N_P , N_C , r_w) foi comparado com o algoritmo GPC convencional e com os controladores discreto $C1(z)$ e $C2(z)$, onde $C1(z) = \frac{6,359(z - 0,595)}{z - 1}$ e $C2(z) = \frac{35(z - 0,97)(z - 0,595)}{(z - 1)^2}$. A ferramenta SISO Design Tool do MATLAB foi utilizado para sintonizar $C1(z)$ e $C2(z)$.

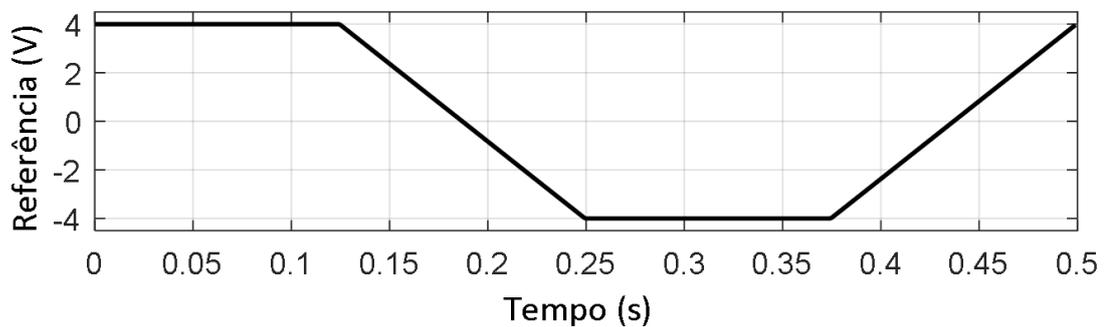
Adicionalmente, cada controlador mencionado foi implementado experimentalmente no DSP DSPACE DS1104, tal como mostrado na Figura 4.1.3. A planta $G(s)$ foi implementada com resistores, capacitores e o amplificador operacional TL 082. A referência e os parâmetros dos controladores foram os mesmos utilizados nas simulações.

As Figuras 4.1.5 e 4.1.6 mostram os resultados dos testes. A Tabela 1 lista a raiz do erro quadrático médio em estado estacionário (*root mean square error* em Inglês), o tempo de assentamento e o pico do erro para cada teste experimental. Os rmses para o GPC convencional e para $C_1(z)$ são altos devido ao erro de regime permanente quando a referência é uma rampa. O erro de pico para $C_2(z)$ tem o valor mais baixo. O rmse para o controlador proposto pode ser reduzido aumentando r_w , mas o pico de erro também é aumentado. No entanto, o tempo de acomodação para $C_2(z)$ é entre 8 a 27 vezes maior que o obtido para o controlador proposto (dependendo dos valores de N_P , N_C e r_w).

Figura 4.1.3 – Bancada Experimental

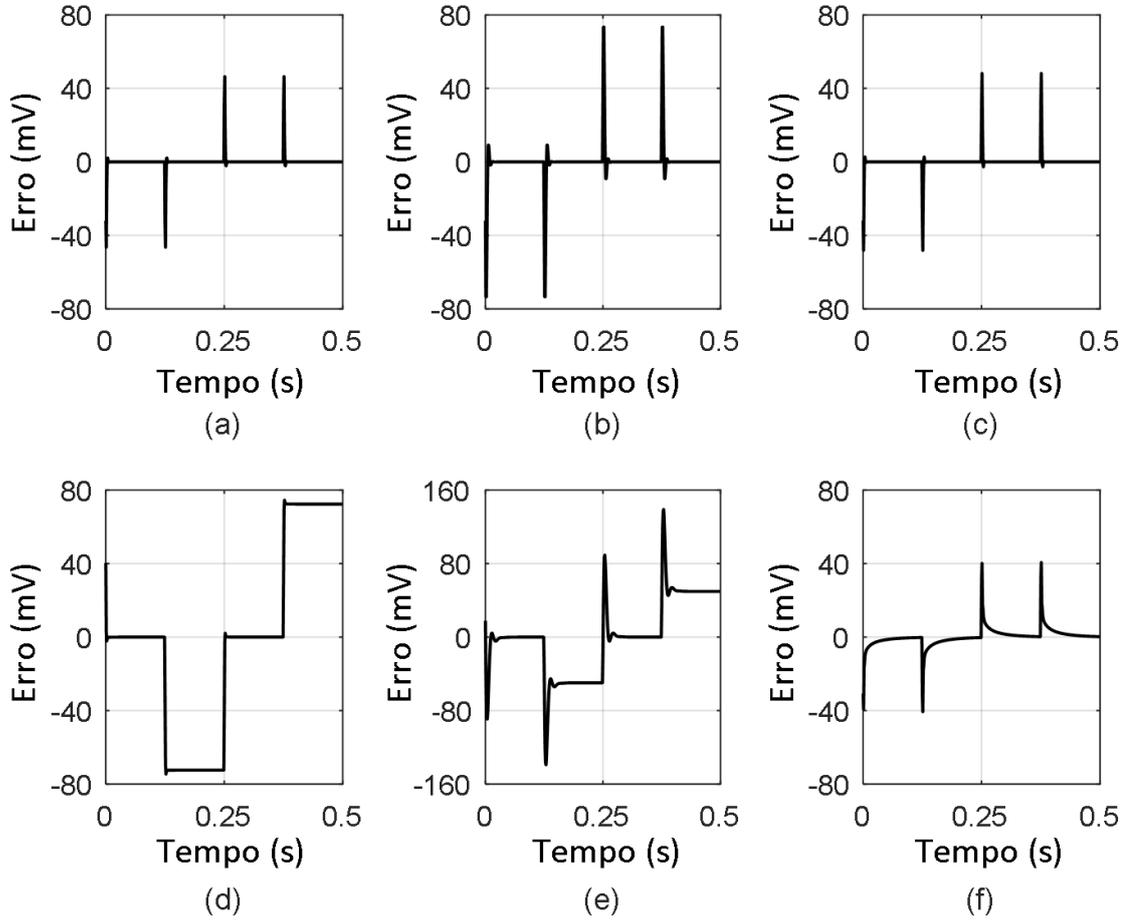


Fonte: Autor

Figura 4.1.4 – Onda de referência da planta $G(s)$ 

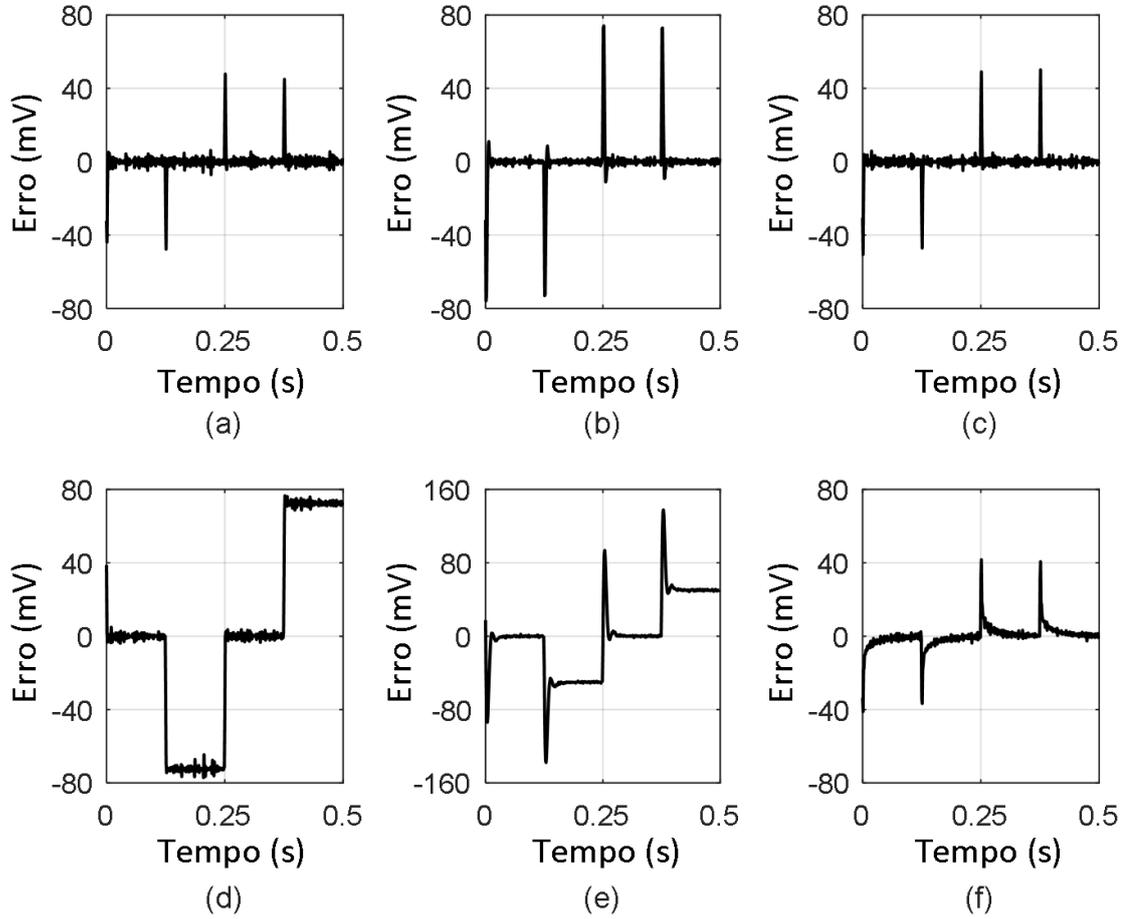
Fonte: Autor

Figura 4.1.5 – Resultados de Simulações. (a) GPC proposto usando $N_P = 3$, $N_C = 2$, $r_w = 0.0003$. (b) GPC proposto $N_P = 3$, $N_C = 2$, $r_w = 0.01$. (c) GPC proposto $N_P = 4$, $N_C = 3$, $r_w = 0.0003$. (d) GPC convencional usando $N_P = 3$, $N_C = 2$, $r_w = 0.0003$. (e) C1(z). (f) C2(z).



Fonte: Autor

Figura 4.1.6 – Resultados Experimentais. (a) GPC proposto usando $N_P = 3$, $N_C = 2$, $r_w = 0.0003$. (b) GPC proposto $N_P = 3$, $N_C = 2$, $r_w = 0.01$. (c) GPC proposto $N_P = 4$, $N_C = 3$, $r_w = 0.0003$. (d) GPC convencional usando $N_P = 3$, $N_C = 2$, $r_w = 0.0003$. (e) C1(z). (f) C2(z).



Fonte: Autor

Tabela 1 – Resultados experimentais para a planta $G(s)$.

Controladores	rmse (mV)	Tempo de Assentamento (s)	Pico do Erro (mV)
SOD - GPC com $N_p = 3, N_c = 2, r_w = 0.0003$	1.78	0.004	47.85
SOD - GPC com $N_p = 3, N_c = 2, r_w = 0.01$	1.04	0.013	74.10
SOD - GPC com $N_p = 4, N_c = 2, r_w = 0.0003$	1.58	0.007	50.16
GPC Convencional	51.12	0.007	74.86
C1 (z)	35.35	0.031	137.8
C2 (z)	1.03	0.108	41.75

4.2 Desenvolvimento do ATO baseado em GPC

4.2.1 Resultados das Simulações

Um conjunto de testes de simulação foi realizado no SIMULINK para demonstrar o desempenho do ATO baseado no GPC proposto. A taxa de amostragem foi fixada em 50 kHz. Os parâmetros do sensor do resolver estão listados na [Tabela 2](#). A posição angular corresponde à curva de velocidade mostrada na [Figura 4.2.1](#). O ATO proposto foi testado considerando diferentes valores de N_p , N_c e r_w :

- Configuração 1: $N_p = 102, N_c = 2, r_w = 0,01$;
- Configuração 2: $N_p = 120, N_c = 2, r_w = 0,01$;
- Configuração 3: $N_p = 102, N_c = 10, r_w = 0,01$.

Tabela 2 – Parâmetros de simulação para o sensor resolver.

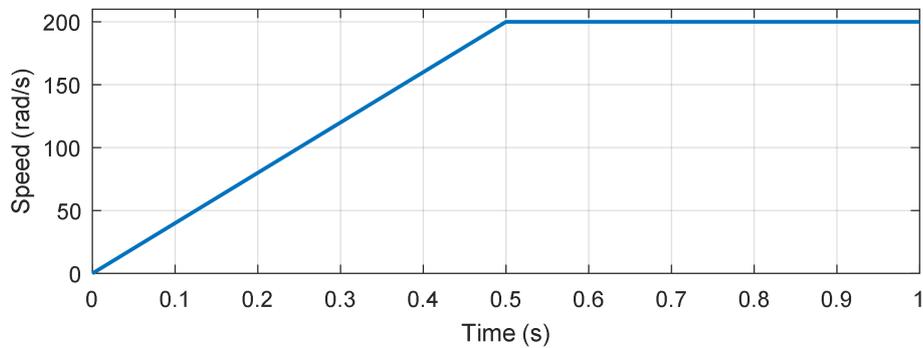
Parâmetros	Valores
Amplitude de excitação (a_r)	8,0 V
Frequência de excitação (f_r)	2,5 kHz
Relação de transformação (k_r)	0,5

A abordagem proposta foi comparada com o RDC descrito em [34]. O sistema de rastreamento tipo II mostrado na Figura 4.2.2 é usado em [34] para estimar a posição angular. A dinâmica desse ATO depende dos valores próprios da matriz $M = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 0, 5(k_r a_r)^2 \\ 0 \\ 0 \end{bmatrix} \underbrace{[k_0 - k_1 - k_2]}_K$. A fórmula de Ackermann pode ser usada para obter a matriz K a fim de definir os autovalores de M . Neste artigo, esses valores próprios foram definidos em $-100 + j100$, $-100 - j100$ e -500 .

Para cada configuração ATO, dois testes foram realizados: sem adicionar ruído às saídas do resolver e adicionar um ruído (média zero e variação de 0,0002) às saídas do resolver. As Figuras 4.2.3, 4.2.4, 4.2.5 e 4.2.6 mostram o erro de estimativa de ângulo para cada teste de simulação.

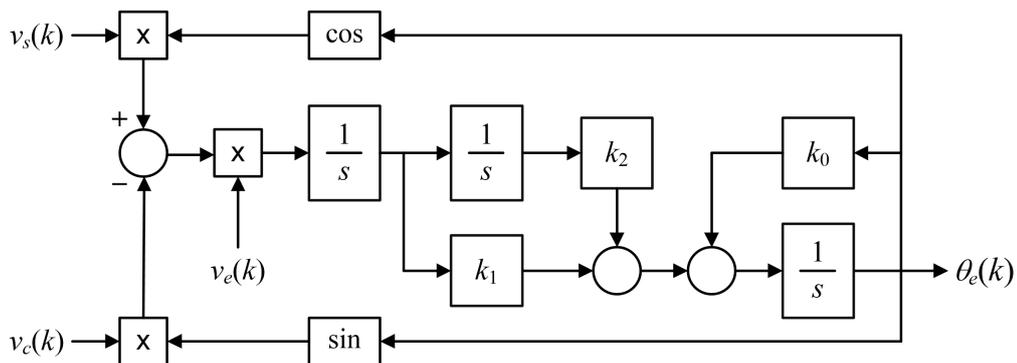
A Tabela 3 mostra a raiz quadrada média do erro (rmse) e o tempo de acomodação do erro (o tempo necessário para o sinal de erro atingir seu valor de estado estacionário) para cada teste de simulação. Os erros sem ruído são desprezíveis em todos os casos.

Figura 4.2.1 – Curva de velocidade para a simulação do sensor resolver.



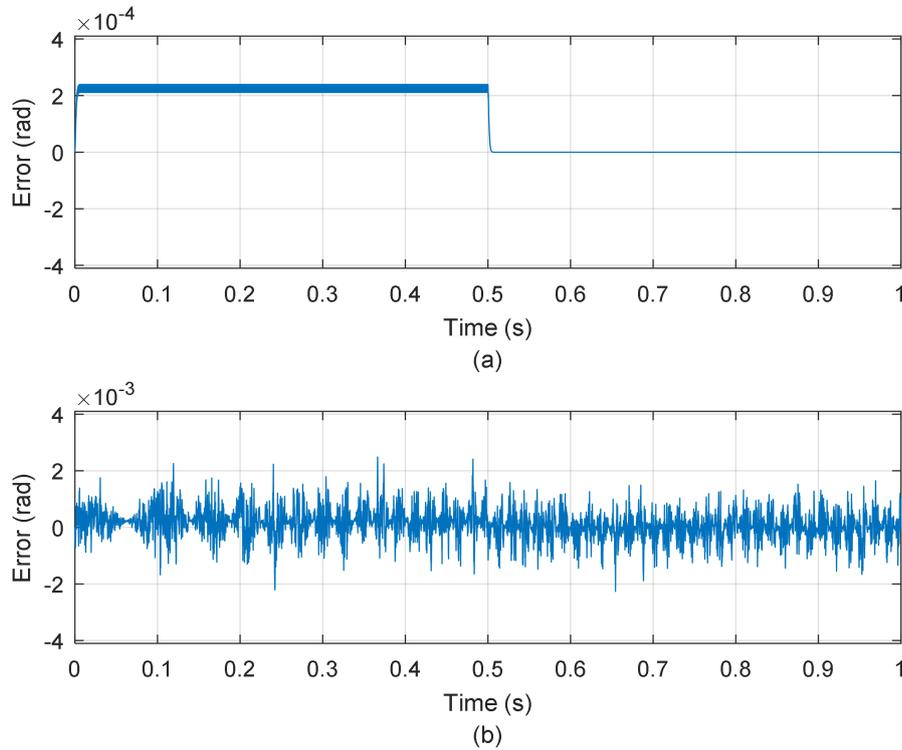
Fonte: Autor

Figura 4.2.2 – Sistema RDC mencionado.



Fonte: Autor

Figura 4.2.3 – Resultados de simulação for $N_p = 102, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.



Fonte: Autor

Porém, é possível ver que o ATO em [34] possui um rmse (quando ruído é adicionado nas saídas do resolver) e um tempo de acomodação maiores do que a abordagem proposta. Como resultado, o sistema RDC proposto baseado em SDO-GPC tem resposta mais rápida e mais robustez do que a abordagem em [34].

4.2.2 Resultados Experimentais

Para testar o ATO proposto, foi montada uma plataforma experimental baseada em hardware-in-the-loop (HIL). A configuração experimental mostrada na Figura 4.2.7

Tabela 3 – Valores do rmse e do tempo de assentamento nos testes.

Configuration	rmse ¹	rmse ²	Tass
$N_p = 102, N_c = 2, r_w = 0.01$	0.16×10^{-3}	0.48×10^{-3}	4.90×10^{-3}
$N_p = 120, N_c = 2, r_w = 0.01$	0.20×10^{-3}	0.47×10^{-3}	5.10×10^{-3}
$N_p = 102, N_c = 10, r_w = 0.01$	0.04×10^{-3}	0.68×10^{-3}	2.10×10^{-3}
ATO in [34]	0.61×10^{-3}	0.88×10^{-3}	55.0×10^{-3}

* $rmse^1$: sem ruído nas saídas do resolver. $rmse^2$: com ruído nas saídas do resolver.

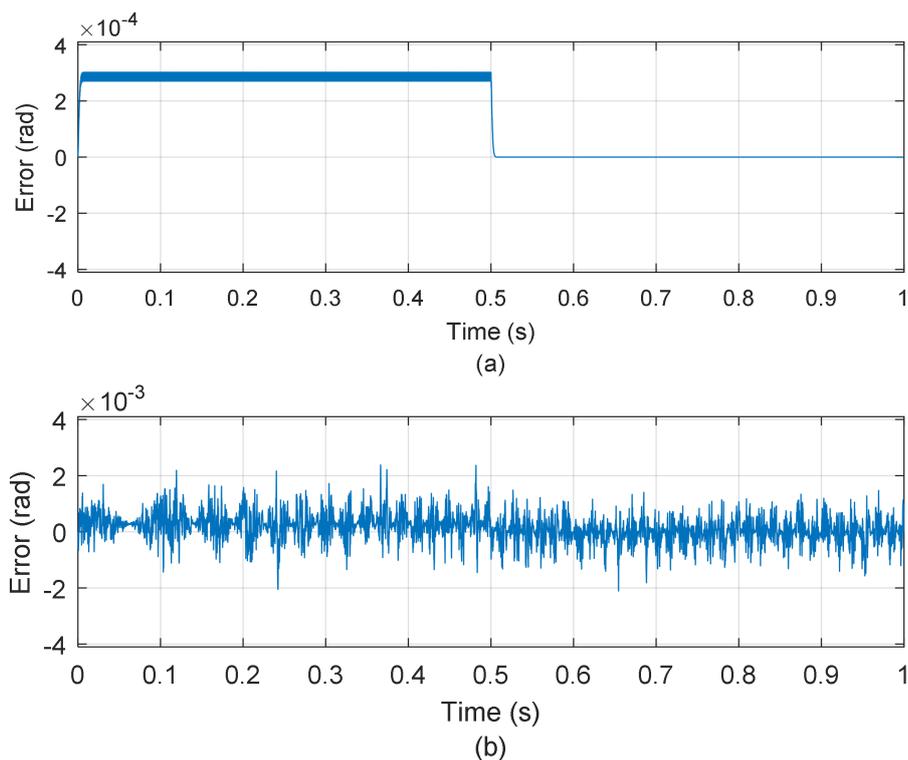
Tass: tempo de assentamento.

consiste em um PC e no FPGA Development and Education Board DE2-115 de ALTERA (baseado no FPGA EP4CE115F29C7). Um cabo Ethernet permite a comunicação entre a placa FPGA e o PC.

No método de teste HIL usado neste artigo, os sinais do resolver são gerados por meio de um diagrama de blocos SIMULINK no PC. Esses sinais são enviados para a placa FPGA por meio do cabo Ethernet. A placa FPGA recebe esses sinais, estima a posição angular através do algoritmo ATO proposto e envia essa estimativa para o diagrama de blocos no PC para calcular o erro de estimativa do ângulo. Uma vantagem de utilizar o método HIL é que a posição angular é conhecida a cada instante. Assim, o erro de estimativa do ângulo pode ser calculado com precisão. Por outro lado, quando um resolver prático é usado, um sensor de posição angular adicional com melhor precisão (que é difícil de obter) é necessário para obter a posição angular real.

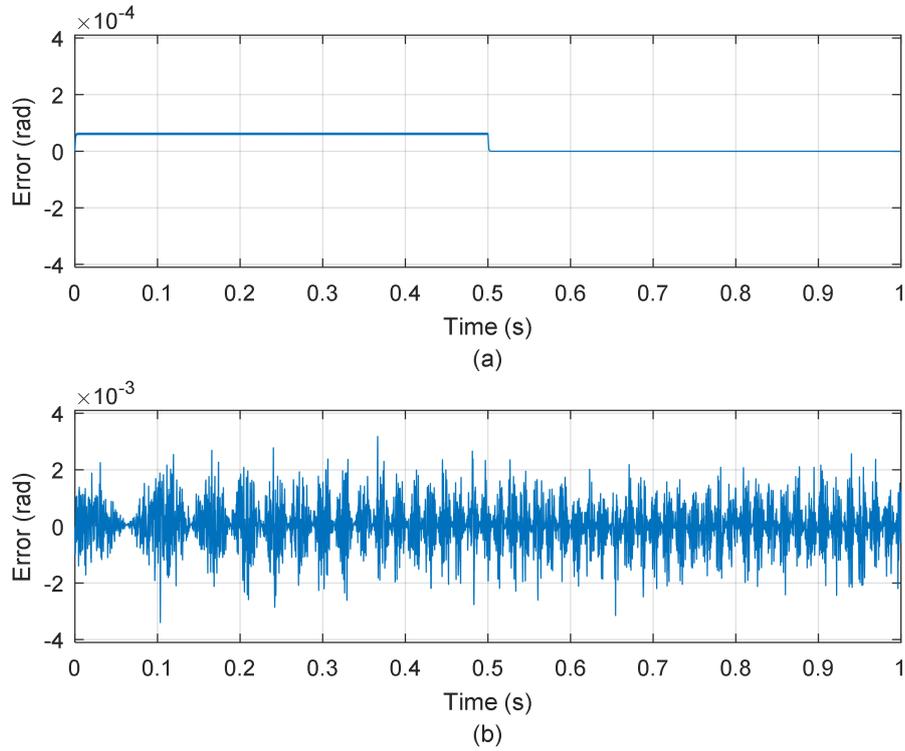
O ATO proposto foi implementado na placa FPGA através de um código VHDL, considerando a mesma frequência de amostragem do sinal (50 kHz) e os parâmetros SOD-GPC utilizados nos testes de simulação. Foi utilizado o formato de dados de ponto fixo de 32 bits para realizar as operações aritméticas. No entanto, a posição angular estimada foi representada em 40 bits (30 bits fracionários) para fazer uma melhor comparação entre o ângulo real e o estimado. Os sinais do resolver emulados são os mesmos usados nos testes de simulação.

Figura 4.2.4 – Resultados de simulação for $N_p = 120$, $N_c = 2$, $r_w = 0.01$. a) Sem ruído, b) Com ruído.



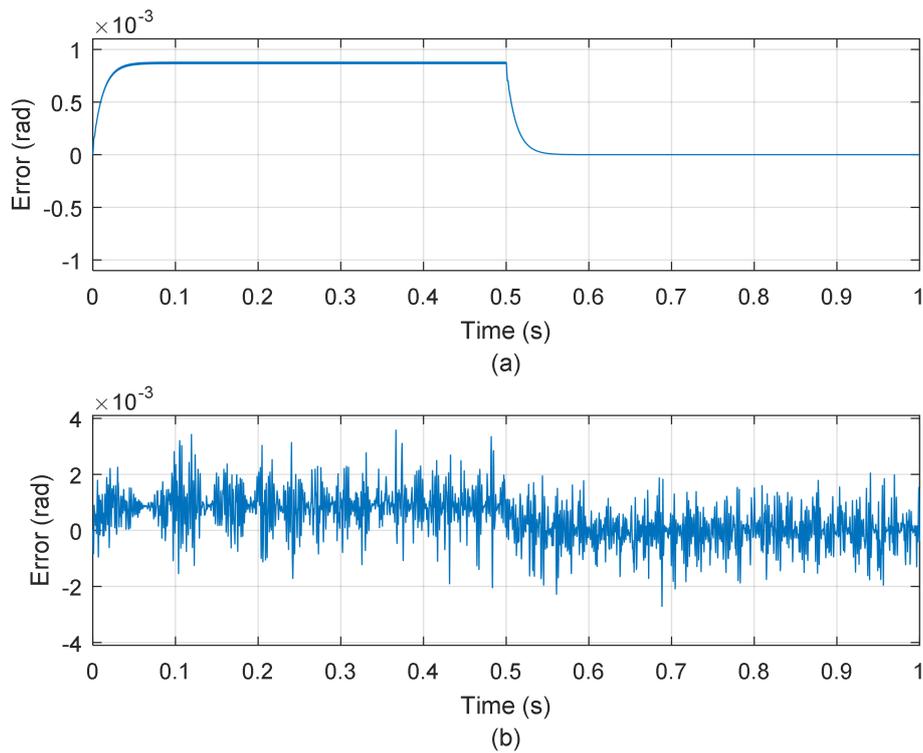
Fonte: Autor

Figura 4.2.5 – Resultados de simulação for $N_p = 102, N_c = 10, r_w = 0.01$. a) Sem ruído, b) Com ruído.



Fonte: Autor

Figura 4.2.6 – Resultados de simulação para o sistema RDC mencionado. a) Sem ruído, b) Com ruído.



Fonte: Autor

As figuras 4.2.8, 4.2.9 e 4.2.10 mostram os resultados dos testes HIL. Nos testes sem ruído, é possível observar uma ondulação no erro de posição angular durante a operação em velocidade constante (de 0 s a 1 s). Essa ondulação é produzida pelo número limitado de bits usados para as operações aritméticas. No entanto, essa ondulação é desprezível (essa ondulação tem uma amplitude menor que 5×10^{-5} rad). A Tabela 4 apresenta o rmse e o tempo de acomodação obtido em cada teste HIL. Esses valores são semelhantes aos obtidos em simulações. Quanto maior o valor de N_c , mais rápida será a resposta ATO, mas maior será o rmse.

Não foi possível implementar o ATO usado de referência no FPGA [34] utilizando uma aritmética em ponto fixo com 32 bits. O principal problema foi a discretização dos integradores. Considerando que o tempo de amostragem é de $t_s = 0,02$ ms, a quantidade de bits utilizados não eram suficientes para efetuar adequadamente os cálculos dos integradores. Aquela é uma vantagem do controlador preditivo: a estrutura do GPC é definida em um modelo de tempo discreto.

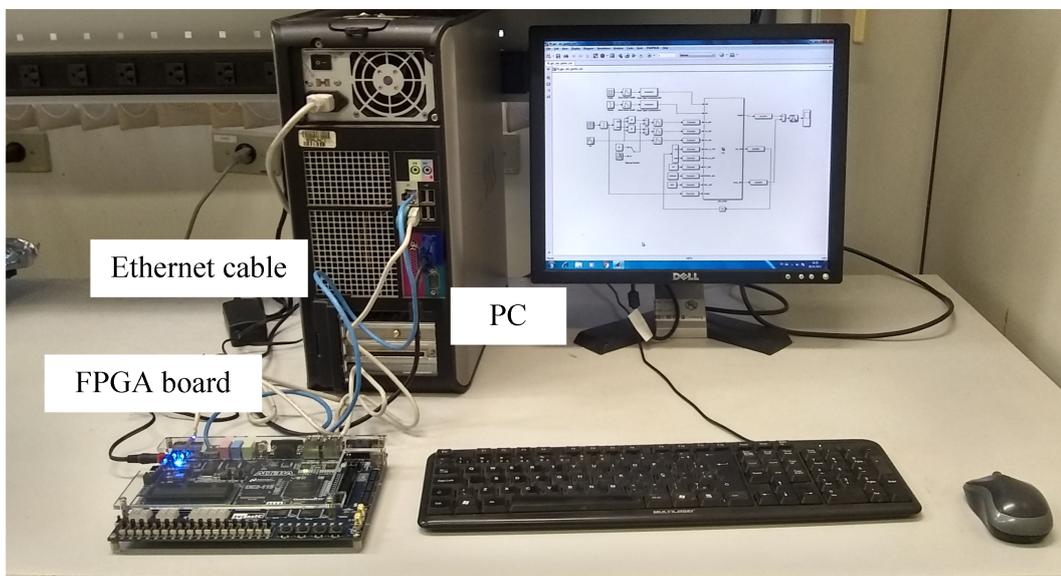
Tabela 4 – Valores do rmse e do tempo de assentamento nos testes experimentais.

Configuração	rmse ¹	rmse ²	T _{ass}
$N_p = 102, N_c = 2, r_w = 0.01$	0.18×10^{-3}	0.48×10^{-3}	5.50×10^{-3}
$N_p = 120, N_c = 2, r_w = 0.01$	0.23×10^{-3}	0.47×10^{-4}	5.90×10^{-3}
$N_p = 102, N_c = 10, r_w = 0.01$	0.06×10^{-3}	0.69×10^{-4}	3.90×10^{-3}

**rmse*¹: sem ruído nas saídas do resolver. *rmse*²: com ruído nas saídas do resolver.

T_{ass}: tempo de assentamento.

Figura 4.2.7 – Bancada experimental para o teste do ATO proposto.



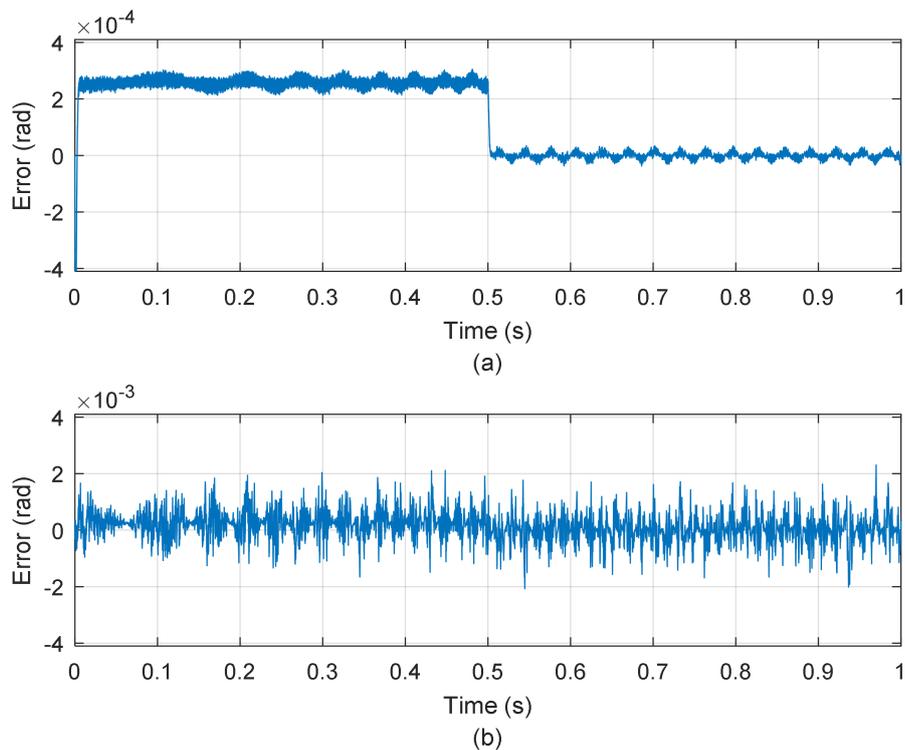
Fonte: Autor

4.3 Ensaio Adicionais

O algoritmo GPC proposto foi testado em simulações com uma referência tipo dente de serra mostrada na [Figura 4.3.1](#). para verificar o comportamento da planta ante uma variação abrupta de referência. A planta a controlar é a função de transferência $G(s) = \frac{53.48s+5347.59}{s^2+253.5s+10695.18}$ no primeiro grupo de ensaios.

As [Figura 4.3.2](#) até [Figura 4.3.7](#) mostram os resultados de simulação obtidos. Pode-se observar que, em todos os casos, o pico de error na transição é de 20 V. Porém, rapidamente (aproximadamente em 5 ms) o controlador corrige o erro. Melhoras no algoritmo proposto devem ser feitas para reduzir o pico de erro durante variações abruptas na referência.

Figura 4.2.8 – Resultados experimentais para $N_p = 102, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.



fonte: Autor

Figura 4.2.9 – Resultados experimentais para $N_p = 120, N_c = 2, r_w = 0.01$. a) Sem ruído, b) Com ruído.

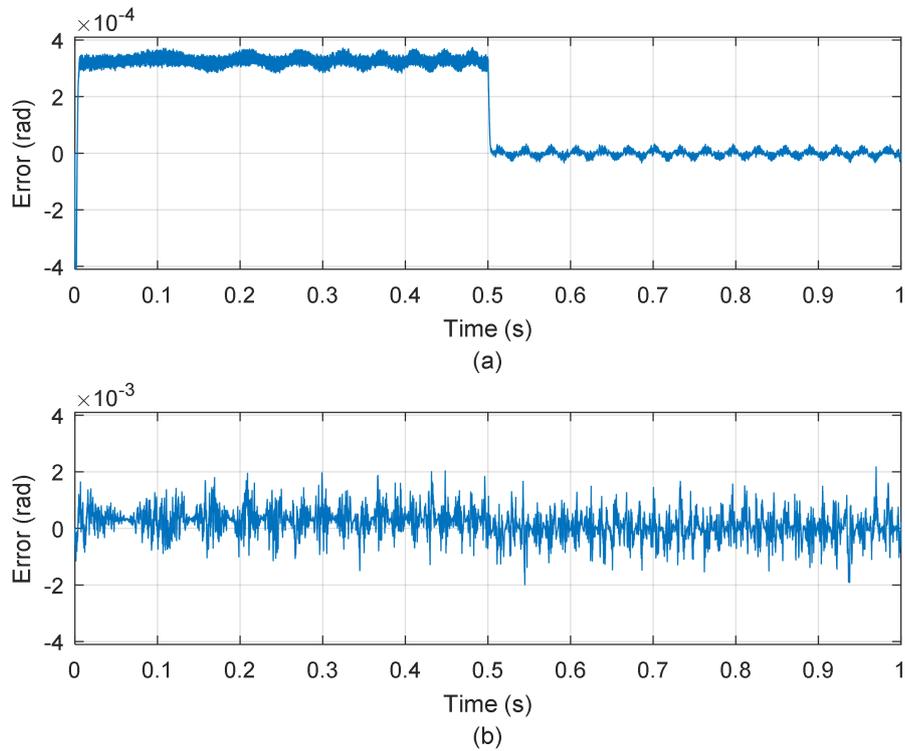
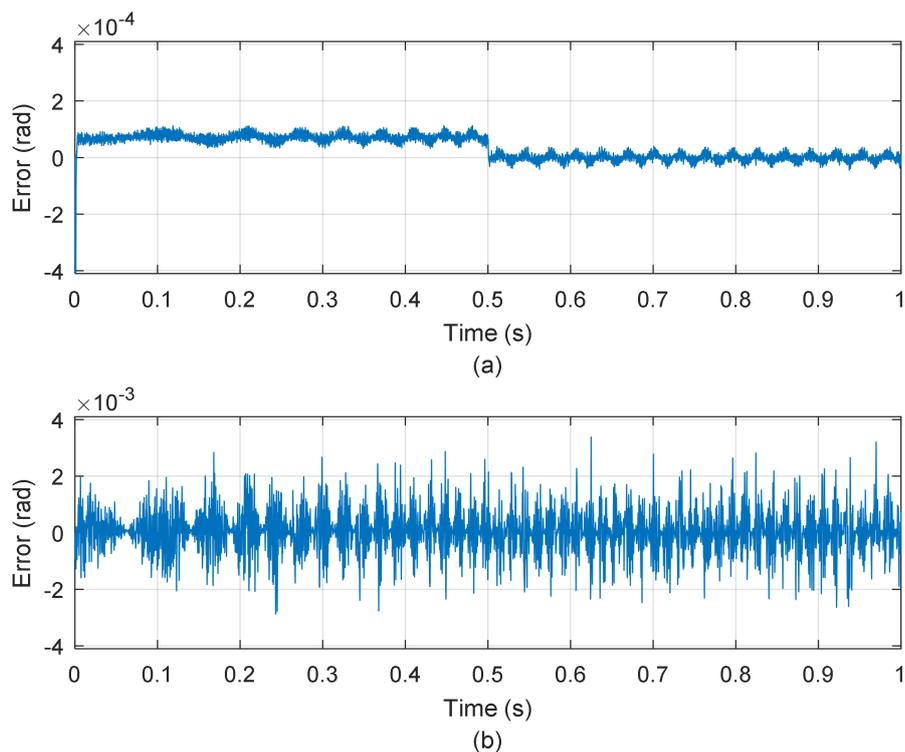
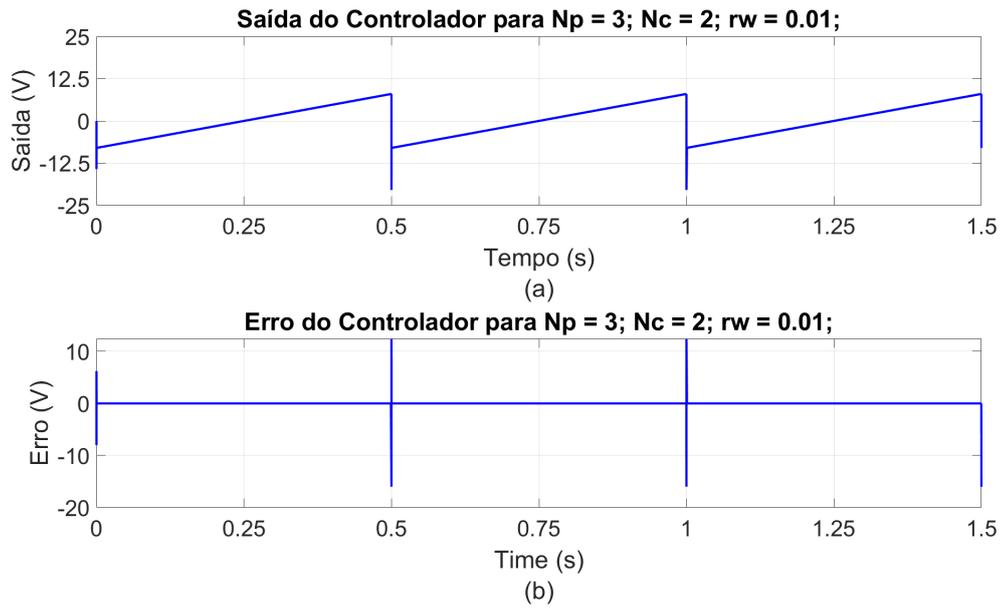


Figura 4.2.10 – Resultados experimentais para $N_p = 102, N_c = 10, r_w = 0.01$. a) Sem ruído, b) Com ruído.



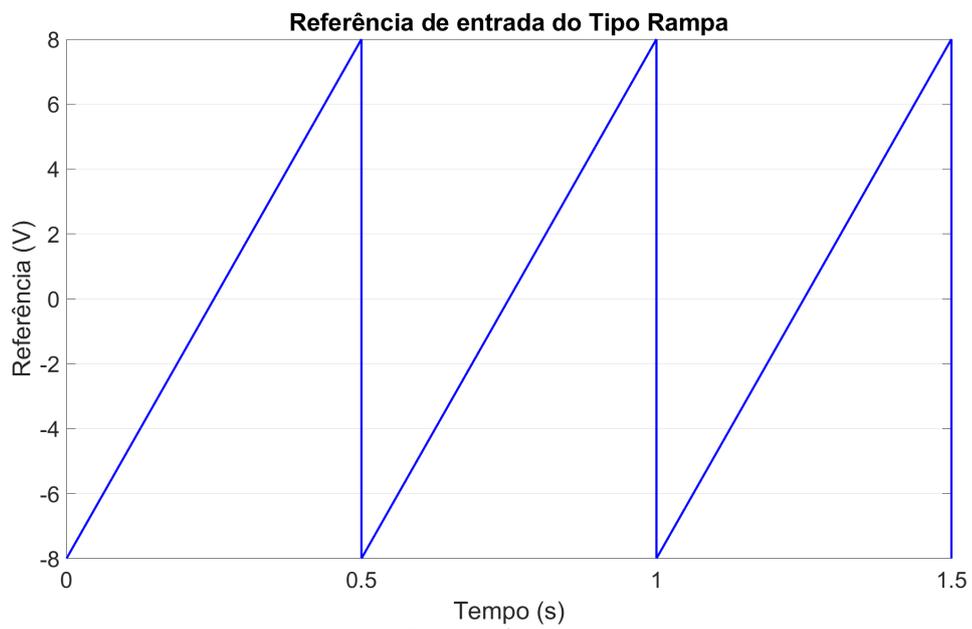
fonte: Autor

Figura 4.3.2 – Resultados da simulação para $N_n = 3$, $N_c = 2$, $r_w = 0.01$.



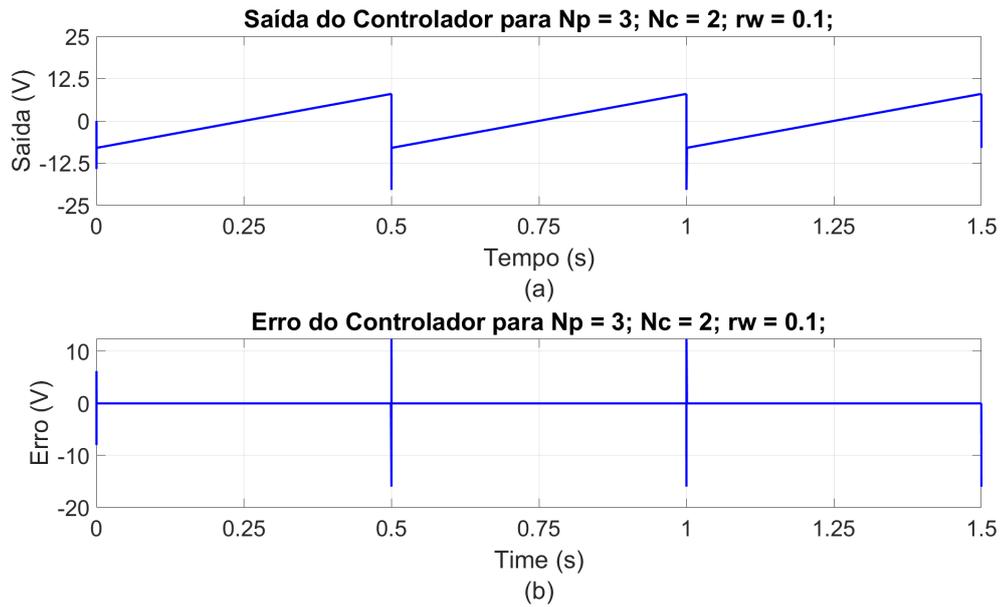
Fonte: Autor

Figura 4.3.1 – Referência dente de serra.



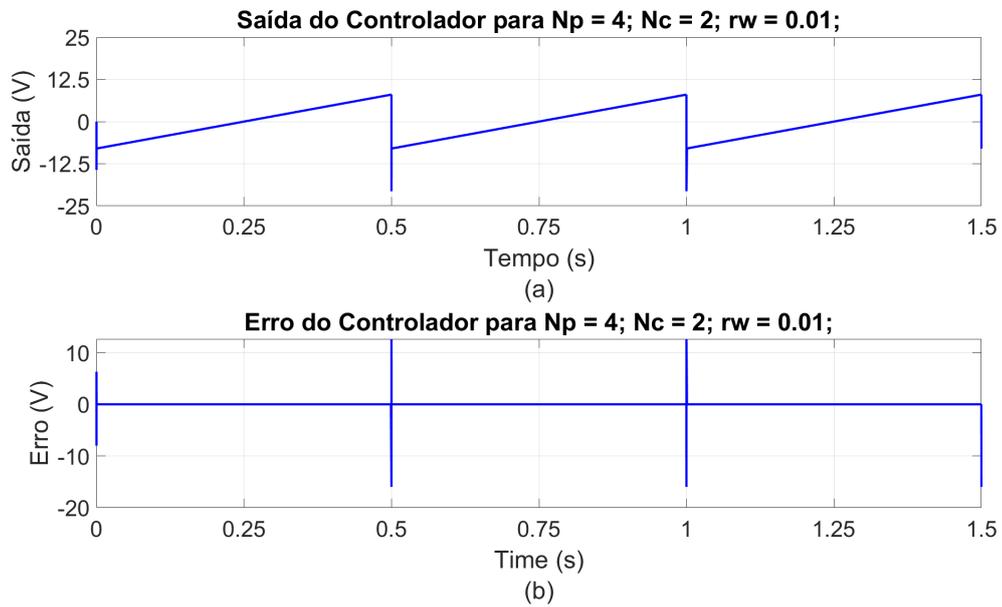
Fonte: Autor

Figura 4.3.3 – Resultados da simulação para $N_p = 3, N_c = 2, r_w = 0.1$.



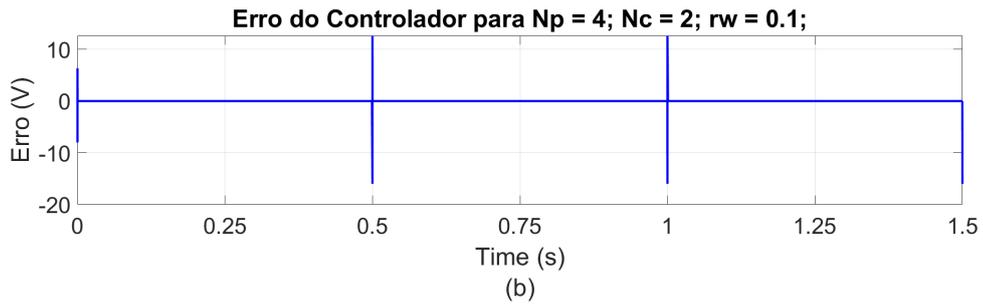
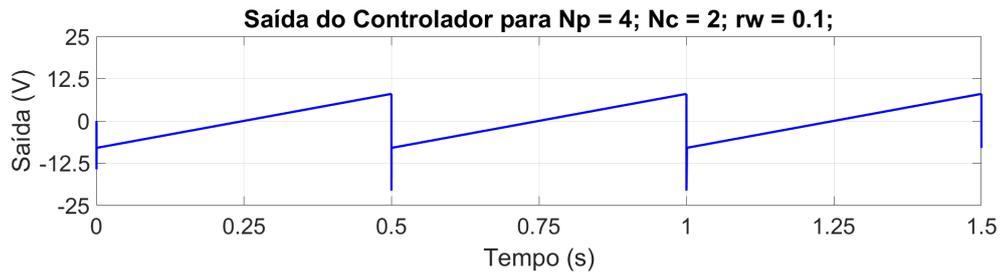
Fonte: Autor

Figura 4.3.4 – Resultados da simulação para $N_p = 4, N_c = 2, r_w = 0.01$.



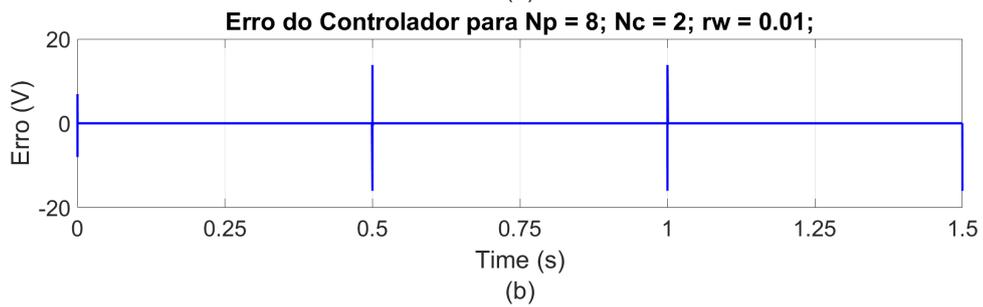
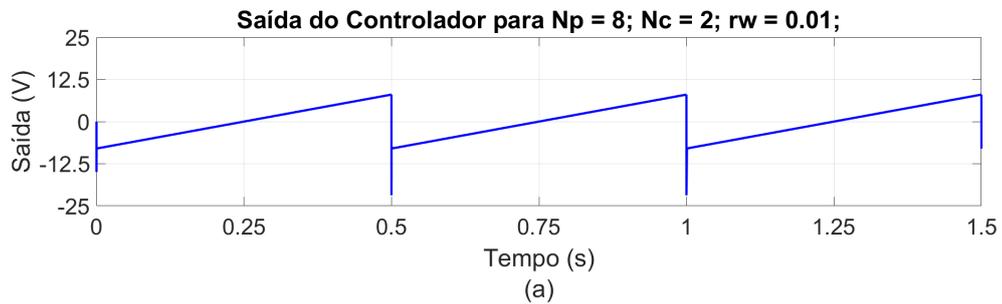
Fonte: Autor

Figura 4.3.5 – Resultados da simulação para $N_p = 4, N_c = 2, r_w = 0.1$.

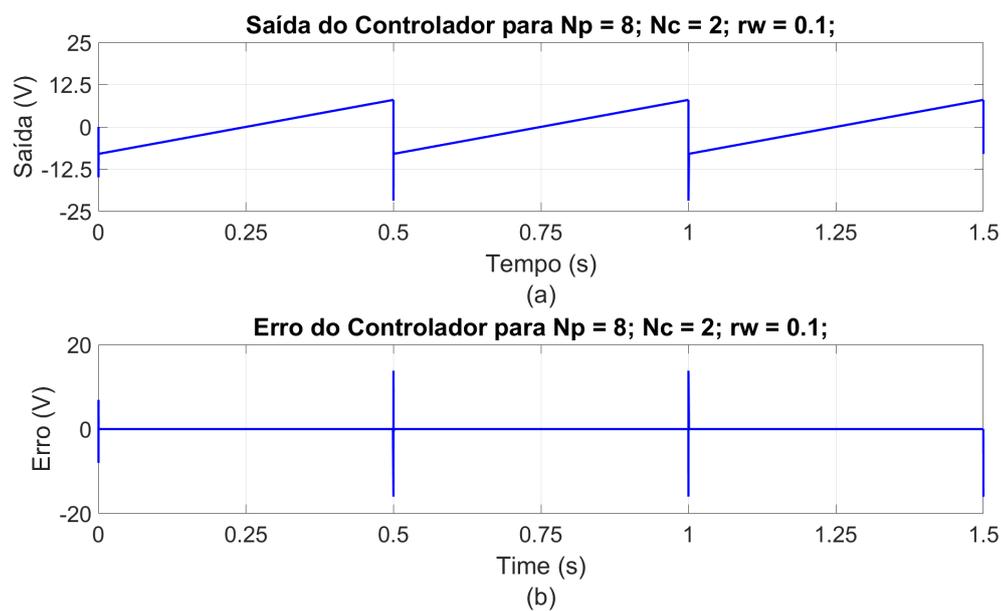


Fonte: Autor

Figura 4.3.6 – Resultados da simulação para $N_p = 8, N_c = 2, r_w = 0.01$.



Fonte: Autor

Figura 4.3.7 – Resultados da simulação para $N_p = 8$, $N_c = 2$, $r_w = 0.1$.

Fonte: Autor

Conclusão

O presente trabalho de dissertação de mestrado apresenta uma adaptação do algoritmo GPC convencional para o rastreamento de referência tipo rampa. Esta referência é utilizada em processos industriais. Ademais, os controladores tipo 2 (com 2 integradores embutidos) são utilizados em PLL, ATOs e estimadores. O algoritmo proposto é simples de implementar tanto em DSPs como em FPGAs. Uma vantagem do GPC é que o modelo de predição e a lei de controle são definidos em modelos em tempo discreto. Assim, o processo de discretização de controladores em tempo contínuo é eliminado. A resposta do GPC depende dos valores de N_p , N_c e r_w . A sintonização do GPC está além dos objetivos deste trabalho de Dissertação. É sugerido a utilização de técnicas heurísticas como algoritmos genéticos para obter os parâmetros adequados do controlador proposto. Outro trabalho futuro proposto é a adaptação do algoritmo proposto para ser aplicado em plantas MIMO (múltiplas entradas e múltiplas saída). Adicionalmente, melhoras no algoritmo GPC proposto deverão ser efetuadas para evitar grandes picos de erro durante variações abruptas da referência.

Referências

- [1] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer, London, 2 ed., 2007.
- [2] L. Wang, *Model Predictive Control System Design and Implementation Using Matlab*. Springer, London, 1 ed., 2009.
- [3] U. Maeder and M. Morari, “Offset-free reference tracking with model predictive control,” *Automatica*, vol. 46, no. 9, pp. 1469–1476, 2010.
- [4] P. Orukpe, “Model predictive control fundamentals,” *Nigerian Journal of Technology (NIJOTECH)*, vol. 31, pp. 139–148, July 2012.
- [5] S. J. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” *Transactions of ASME, Journal of Basic Engineering*, vol. 11, pp. 736–746, 2003.
- [6] L. W. S. Rakovic S.V., *Handbook of Model Predictive Control*. Springer, London, 1 ed., 2019.
- [7] D. M. M. Rawlings J. B., Mayne D. Q., *Model Predictive Control: Theory, Computation, and Design*. LLC, California, 2 ed., 2019.
- [8] H. S. Kwon W.H., *Receding Horizon Control: Model Predictive Control for State Models*. Springer, London, 1 ed., 2005.
- [9] M. C. Clarke D. W., “Generalized predictive control – part i. the basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.
- [10] M. C. Clarke D. W., “Generalized predictive control – part ii. extensions and interpretations,” *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.
- [11] R. E. Kalman, “Contributions to the theory of optimal control,” *Bulletin de la Sociedad Matematica Mexicana*, vol. 5, pp. 102–119, 1960.
- [12] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [13] R. N. R., “Model predictive control: History and development,” *International Journal of Engineering Trends and Technology (IJETT)*, vol. 4, 2013.
- [14] T. Sato, “Design of a gpc-based pid controller for controlling a weigh feeder,” *Control Engineering Practice* 18, pp. 105–113, 2010.

- [15] C. Pengzhan and B. Liu, “Combined pi and gpc for servo system,” *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering, CMCE 2010*, vol. 4, 08 2010.
- [16] T. V. Estrabis, R. C. García, E. A. Batista, C. Q. Andrea, and M. A. S. Grassi, “Application of model predictive control in a resolver-to-digital converter,” pp. 292–296, 2019.
- [17] R. C. Dorf and R. H. Bishop, *Modern Control Systems*. Addison-Wesley Longman, Menlo Park, CA, USA, 8 ed., 1998.
- [18] A. J. Jerry, *Difference Equations with Discrete Transform Methods*. Springer Science+Business Media, Dordrecht, Netherlands, 1996.
- [19] S. Haykin and B. V. Veen, *Signals and Systems*. USA: John Wiley and Sons, Inc., 2nd ed., 2002.
- [20] J. O. Smith, *Introduction to Digital Filters with Audio Applications*. <http://www.w3k.org/books/>: W3K Publishing, 2007.
- [21] M. Benammar, L. Ben-Brahim, and M. A. Alhamadi, “A high precision resolver-to-dc converter,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2289–2296, 2005.
- [22] M. Benammar, “A novel amplitude-to-phase converter for sine/cosine position transducers,” *International Journal of Electronics*, vol. 94, no. 4, pp. 353–365, 2007.
- [23] Y. Wang, Z. Zhu, and Z. Zuo, “A novel design method for resolver-to-digital conversion,” *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 3724–3731, 06 2015.
- [24] C. M. R. S, “Inverse tangent based resolver to digital converter-a software approach,” *IJAET*, vol. 4, pp. 228–235, 09 2012.
- [25] S. Sarma, V. K. Agrawal, and S. Udupa, “Software-based resolver-to-digital conversion using a dsp,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 371–379, 2008.
- [26] T. Shi, Y. Hao, G. Jiang, Z. Wang, and C. Xia, “A method of resolver-to-digital conversion based on square wave excitation,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7211–7219, 2018.
- [27] L. Ben-Brahim, M. Benammar, and M. Alhamadi, “A resolver angle estimator based on its excitation signal,” *Industrial Electronics, IEEE Transactions on*, vol. 56, pp. 574 – 580, 03 2009.

-
- [28] M. Benammar and A. S. P. Gonzales, “A novel pll resolver angle position indicator,” *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 1, pp. 123–131, 2016.
- [29] L. Pecly, R. Schindeler, D. Cleveland, and K. Hashtrudi-Zaad, “High-precision resolver-to-velocity converter,” *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 11, pp. 2917–2928, 2017.
- [30] L. Idkhajine, E. Monmasson, M. W. Naouar, A. Prata, and K. Bouallaga, “Fully integrated fpga-based controller for synchronous motor drive,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4006–resolver11, 2009.
- [31] N. Abou Qamar, C. J. Hatziaioniu, and H. Wang, “Speed error mitigation for a dsp-based resolver-to-digital converter using autotuning filters,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1134–1139, 2015.
- [32] M. Caruso, A. Di Tommaso, F. Genduso, R. Miceli, and G. Galluzzo, “A dsp-based resolver-to-digital converter for high performance electrical drive applications,” *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 1–1, 07 2016.
- [33] R. C. García, J. O. P. Pinto, W. I. Suemitsu, and J. O. Soares, “Improved demultiplexing algorithm for hardware simplification of sensed vector control through frequency-domain multiplexing,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6538–6548, 2017.
- [34] R. C. García, J. O. P. Pinto, I. E. S. Ono, V. d. S. Fahed, and M. Brito, “Simplification of the acquisition system for sensed vector control using resolver sensor based on fdm and current synchronous sampling,” pp. 1–7, 2018.
- [35] C. W. X.M. Sun, D. Wu and W. Wang, “A novel stability analysis for networked predictive control systems,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 14, pp. 453–457, 2014.
- [36] M. R. A. Oliveri and M. Storace, “Automatic domain partitioning of piecewise-affine simplicial functions implementing model predictive controllers,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 9, pp. 886–890, 2015.
- [37] K. M. Abughalieh and S. G. Alawneh, “A survey of parallel implementations for model predictive control,” *IEEE Access*, vol. 7, pp. 34348–34360, 2019.
- [38] Y. Yan, J. Yang, Z. Sun, S. Li, and H. Yu, “Non-linear-disturbance-observer-enhanced mpc for motion control systems with multiple disturbances,” *IET Control Theory Applications*, vol. 14, no. 1, pp. 63–72, 2020.
- [39] J. Rossiter, *Model-based Predictive Control-a Practical Approach*. 01 2003.

-
- [40] E. Fernandez-Car and E. Zuazua, “Control theory: History, mathematical achievements and perspectives,” 01 2003.
- [41] M. A. S. Grassi, E. A. Batista, C. Q. Andrea, R. C. Garcia, and E. S. D. Nascimento, “Controladores preditivos gpc para um sistema turbina-gerador: Testes em cosimulação e fpga-in-the-loop,” 2018.
- [42] B. Huang and R. Kadali, “Dynamic modeling, predictive control and performance monitoring: A data-driven subspace approach,” 2008.
- [43] G. Cheng, K. Peng, B. M. Chen, and T. H. Lee, “Improving transient performance in tracking general references using composite nonlinear feedback control and its application to high-speed xy -table positioning mechanism,” *IEEE Transactions on Industrial Electronics*, vol. 54, no. 2, pp. 1039–1051, 2007.

Apêndices

A Códigos

A.1 MATLAB

Para gerar os dados para a referência rampa e dente de serra, foi necessário definir os valores de KMPC, esses valores são obtidos a partir destes dois códigos.

```

1 function Kmpc=kkmpc(conf)
2 %% MPC-siso
3 clc; close all;
4 ts=10e-6;
5 Am=[-0.231 -0.600;0.600 -.448];
6 Bm= [2.612;3.078];
7 Cm=[0 1];
8 %% modelo aumentado
9 nm=length(Am);
10 A=[Am zeros(nm,1) zeros(nm,1); -Cm*Am 1 0; -Cm*Am 1 1];
11 B=[Bm; -Cm*Bm; -Cm*Bm];
12 C=[zeros(1,nm) 0 1];
13 Np = conf(1);
14 Nc = conf(2);
15 rw = conf(3);
16 %% matrizes do MPC
17 nau = length(A);
18 Rb = rw*eye(Nc,Nc);
19 F=zeros(Np,nau);
20
21 for i=1:Np
22     F(i,:)=C*(A^i);
23 end
24 %=====
25 Phi = zeros(Np, Nc);
26 %coluna 1:
27 for linha=1:Np
28     Phi(linha,1) = C*(A^(linha-1))*B;
29 end
30
31 %outras colunas

```

```

32 for col =2: Nc
33     Phi(col:Np, col) = Phi(1:Np-(col-1),1);
34 end
35 %=====
36 PTP = Phi'*Phi;
37 if (Nc == 1)
38     vetor10 = [1];
39 else
40     vetor10 = [ 1 zeros(1,Nc-1)];
41 end
42
43 Hessiana = PTP + Rb;
44 Kmpc = -1*vetor10*inv(Hessiana)*(Phi')*F;
45 Kmpc;
46 eig(A-B*Kmpc);

1 ts=1/50000;
2 Np=[3 4 8];
3 Nc=[2 4];
4 rw=[0 0.001 0.01 0.1 1];
5
6 conf=[3 2 0;3 2 0.001;3 2 0.01; 3 2 0.1; 3 2 1;
7       4 2 0;4 2 0.001;4 2 0.01; 4 2 0.1; 4 2 1;
8       8 2 0;8 2 0.001;8 2 0.01; 8 2 0.1; 8 2 1;
9       8 4 0;8 4 0.001;8 4 0.01; 8 4 0.1; 8 4 1];
10 for kk=1:20
11     Kmpc(kk,:) = kkmprc(conf(kk,:));
12 end
13 Kmpc=round(Kmpc,4)

```

Após a obtenção de dados, a análise dos picos máximos e mínimos, rmse e o tempo de assentamento, pode ser feita pelo seguinte código.

```

1 %Nc2
2     %Np3
3     %erro
4     erro_Np3Nc2_r=[Np3Nc2r0.signals(2).values'; Np3Nc2r001.
5                   signals(2).values';
6                   Np3Nc2r01.signals(2).values'; Np3Nc2r1.signals(2).
7                   values';
8                   Np3Nc2r1p.signals(2).values'];

```

```
7      erro_Np3Nc2_d=[Np3Nc2r0.signals(4).values'; Np3Nc2r001.
      signals(4).values';
8      Np3Nc2r01.signals(4).values'; Np3Nc2r1.signals(4).
      values';
9      Np3Nc2r1p.signals(4).values'];
10     y_Np3Nc2_r=[Np3Nc2r0.signals(1).values'; Np3Nc2r001.
      signals(1).values';
11     Np3Nc2r01.signals(1).values'; Np3Nc2r1.signals(1).
      values';
12     Np3Nc2r1p.signals(1).values'];
13     y_Np3Nc2_d=[Np3Nc2r0.signals(3).values'; Np3Nc2r001.
      signals(3).values';
14     Np3Nc2r01.signals(3).values'; Np3Nc2r1.signals(3).
      values';
15     Np3Nc2r1p.signals(3).values'];
16
17     %Np4
18     erro_Np4Nc2_r=[Np4Nc2r0.signals(2).values'; Np4Nc2r001.
      signals(2).values';
19     Np4Nc2r01.signals(2).values'; Np4Nc2r1.signals(2).
      values';
20     Np4Nc2r1p.signals(2).values'];
21     erro_Np4Nc2_d=[Np4Nc2r0.signals(4).values'; Np4Nc2r001.
      signals(4).values';
22     Np4Nc2r01.signals(4).values'; Np4Nc2r1.signals(4).
      values';
23     Np4Nc2r1p.signals(4).values'];
24     y_Np4Nc2_r=[Np4Nc2r0.signals(1).values'; Np4Nc2r001.
      signals(1).values';
25     Np4Nc2r01.signals(1).values'; Np4Nc2r1.signals(1).
      values';
26     Np4Nc2r1p.signals(1).values'];
27     y_Np4Nc2_d=[Np4Nc2r0.signals(3).values'; Np4Nc2r001.
      signals(3).values';
28     Np4Nc2r01.signals(3).values'; Np4Nc2r1.signals(3).
      values';
29     Np4Nc2r1p.signals(3).values'];
30
31     %Np8
```

```
32     erro_Np8Nc2_r=[Np8Nc2r0.signals(2).values'; Np8Nc2r001.
33         signals(2).values';
34         Np8Nc2r01.signals(2).values'; Np8Nc2r1.signals(2).
35         values';
36         Np8Nc2r1p.signals(2).values'];
37 erro_Np8Nc2_d=[Np8Nc2r0.signals(4).values'; Np8Nc2r001.
38     signals(4).values';
39     Np8Nc2r01.signals(4).values'; Np8Nc2r1.signals(4).
40     values';
41     Np8Nc2r1p.signals(4).values'];
42 y_Np8Nc2_r=[Np8Nc2r0.signals(1).values'; Np8Nc2r001.
43     signals(1).values';
44     Np8Nc2r01.signals(1).values'; Np8Nc2r1.signals(1).
45     values';
46     Np8Nc2r1p.signals(1).values'];
47 y_Np8Nc2_d=[Np8Nc2r0.signals(3).values'; Np8Nc2r001.
48     signals(3).values';
49     Np8Nc2r01.signals(3).values'; Np8Nc2r1.signals(3).
50     values';
51     Np8Nc2r1p.signals(3).values'];
52
53
54 %Np8Nc4
55 erro_Np8Nc4_r=[Np8Nc4r0.signals(2).values'; Np8Nc4r001.
56     signals(2).values';
57     Np8Nc4r01.signals(2).values'; Np8Nc4r1.signals(2).
58     values';
59     Np8Nc4r1p.signals(2).values'];
60 erro_Np8Nc4_d=[Np8Nc4r0.signals(4).values'; Np8Nc4r001.
61     signals(4).values';
62     Np8Nc4r01.signals(4).values'; Np8Nc4r1.signals(4).
63     values';
64     Np8Nc4r1p.signals(4).values'];
65 y_Np8Nc4_r=[Np8Nc4r0.signals(1).values'; Np8Nc4r001.
66     signals(1).values';
67     Np8Nc4r01.signals(1).values'; Np8Nc4r1.signals(1).
68     values';
69     Np8Nc4r1p.signals(1).values'];
70 y_Np8Nc4_d=[Np8Nc4r0.signals(3).values'; Np8Nc4r001.
```

```

        signals(3).values';
57     Np8Nc4r01.signals(3).values'; Np8Nc4r1.signals(3).
        values';
58     Np8Nc4r1p.signals(3).values'];
59
60 [x y] = size(erro_Np3Nc2_r);
61
62 for i=1:x
63     erro_med_r(i,:)=[rmse0(erro_Np3Nc2_r(i,:)) rmse0(
        erro_Np4Nc2_r(i,:))
64         rmse0(erro_Np8Nc2_r(i,:)) rmse0(erro_Np8Nc4_r(i,:))];
65     erro_med_d(i,:)=[rmse0(erro_Np3Nc2_d(i,:)) rmse0(
        erro_Np4Nc2_d(i,:))
66         rmse0(erro_Np8Nc2_d(i,:)) rmse0(erro_Np8Nc4_d(i,:))];
67     min_med_r(i,:)=[min(erro_Np3Nc2_r(i,:)) min(erro_Np4Nc2_r(i
        ,:))
68         min(erro_Np8Nc2_r(i,:)) min(erro_Np8Nc4_r(i,:))];
69     min_med_d(i,:)=[min(erro_Np3Nc2_d(i,:)) min(erro_Np4Nc2_d(i
        ,:))
70         min(erro_Np8Nc2_d(i,:)) min(erro_Np8Nc4_d(i,:))];
71     max_med_r(i,:)=[max(erro_Np3Nc2_r(i,:)) max(erro_Np4Nc2_r(i
        ,:))
72         max(erro_Np8Nc2_r(i,:)) max(erro_Np8Nc4_r(i,:))];
73     max_med_d(i,:)=[max(erro_Np3Nc2_d(i,:)) max(erro_Np4Nc2_d(i
        ,:))
74         max(erro_Np8Nc2_d(i,:)) max(erro_Np8Nc4_d(i,:))];
75     assent_med_r(i,:)=[assent(erro_Np3Nc2_r(i,:)) assent(
        erro_Np4Nc2_r(i,:))
76         assent(erro_Np8Nc2_r(i,:)) assent(erro_Np8Nc4_r(i,:))];
77     assent_med_d(i,:)=[assent(erro_Np3Nc2_d(i,:)) assent(
        erro_Np4Nc2_d(i,:))
78         assent(erro_Np8Nc2_d(i,:)) assent(erro_Np8Nc4_d(i,:))];
79 end
80 erro_med_r
81 erro_med_d
82 min_med_r
83 min_med_d
84 max_med_r
85 max_med_d

```

```
86 assent_med_r
87 assent_med_d
```

A.2 VHDL

Para realizar FPGA-in-the-loop, foi necessário implementar o código em VHDL.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
use ieee.fixed_float_types.all;
use ieee.fixed_pkg.all;

entity ato_main is port (
    clk          : in std_logic; -- Clock
    rst          : in std_logic; -- Reset
    vs_std       : in std_logic_vector(15 downto 0);
    vc_std       : in std_logic_vector(15 downto 0);
    ve_std       : in std_logic_vector(15 downto 0);
    sen_e_std: in std_logic_vector(17 downto 0);
    cos_e_std: in std_logic_vector(17 downto 0);
    --ganhos
    k1_std: in std_logic_vector(31 downto 0);
    k23n0_std: in std_logic_vector(31 downto 0);
    k2n_std: in std_logic_vector(31 downto 0);
    -- angulos
    angre        : in std_logic_vector(39 downto 0);
    angrs        : out std_logic_vector(39 downto 0);
    ae_std       : out std_logic_vector(39 downto 0);
    aep_std      : out std_logic_vector(39 downto 0)

    --aepp_std    : out std_logic_vector(39 downto 0)
);
end ato_main;

architecture arq1 of ato_main is
    signal cont          : std_logic_vector(2 downto 0);
    signal D2ae          : sfixed( 1 downto -39);
    signal u, up, upp, aux_u      : sfixed( 1 downto -39);
```

```

signal ae, aep, aepp, aeppp      : sfixed( 9 downto -30);
signal g, gp,aux_g              : sfixed( 1 downto -39);
signal k1, k23n0, k2n           : sfixed(1 downto -30);
signal n0                       : sfixed(1 downto -16);
signal areal                    : std_logic_vector(39 downto 0);
signal ven, vek23n0            : sfixed(1 downto -31);
signal vs,vc,ve                : sfixed(4 downto -11);
signal cos_e, sen_e            : sfixed( 1 downto -16);
signal ka                      : sfixed(1 downto -16);
signal c2n                     : sfixed(3 downto -1);

begin
process (clk,rst) begin
  if (rst = '1') then
    cont      <=(others=>'0'); --zerar contador
    ae       <=(others=>'0');
    aep      <=(others=>'0');
    aepp     <=(others=>'0');
    aeppp    <=(others=>'0');
    D2ae     <=(others=>'0');
    u        <=(others=>'0');
    up       <=(others=>'0');
    upp      <=(others=>'0');
    cos_e    <=to_sfixed(1.0,cos_e);
    sen_e    <=to_sfixed(0.0,cos_e);
    --ganhos -----
    k1       <= to_sfixed( -0.123652563620389, k1);
    k1       <= to_sfixed( k1_std, k1);
    k23n0    <= to_sfixed( k23n0_std, k23n0);
    k2n      <= to_sfixed( k2n_std, k2n);
    n0       <= to_sfixed( 0.0625000000000000, n0);
    c2n      <= to_sfixed( -2.0, c2n);
    -----
    ae_std   <=(others=>'0'); -- saidas iniciais
    aep_std  <=(others=>'0');
    --aepp_std <=(others=>'0');
    angrs    <=(others=>'0');
  elsif (rising_edge(clk)) then
    case cont is
      ----- cont = 0

```

```

when "000" =>
    cont          <= "001";
    vs            <= to_sfixed(vs_std, vs);
    vc            <= to_sfixed(vc_std, vc);
    ve            <= to_sfixed(ve_std, ve);
    cos_e <= to_sfixed(cos_e_std,cos_e);
    sen_e <= to_sfixed(sen_e_std,sen_e);
    D2ae          <= resize(aep + aeppp +
                           (c2n*aepp) , D2ae);
    areal <= angre;
----- cont = 1
when "001" =>
    cont          <= "010";
    aux_g          <= resize((vs*cos_e)-
                           (vc*sen_e), aux_g);
    ven            <= resize(ve*n0, ven);
    vek23n0 <= resize(ve*k23n0, vek23n0);
    aux_u          <= resize(up+up-upp, aux_u);
-----cont = 2
when "010" =>
    cont <= "011";
    u <= resize(aux_u+(k1*D2ae)+(k2n*gp)+
               (aux_g*vek23n0),u);
    g <= resize(aux_g*ven,g);
----- cont = 3
when "011" =>
    cont          <= "100";
    ae            <= resize(aep+u,ae);
----- cont = 4
when "100" =>
    cont          <="000"; --zerar contador
    -- enviando resultados
    ae_std        <= to_slv(ae);
    aep_std       <= to_slv(aep);
    --aepp_std <= to_slv(aepp);
    angrs         <= areal;
    -- atualização de estados passados
    upp           <= up;
    up            <= u;

```

```
aeppp <= aepp;
aepp   <= aep;
aep    <= ae;  --angulo atual;
gp     <= g;

-----

    when others =>
        cont      <="000"; --zerar contador
    end case;
end if;
end process;

end arq1;
```