



Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica



Trabalho de Conclusão de Curso

Estudo de Técnicas de Otimização para Sintonia de Controladores PID Aplicados ao Acionamento de Máquinas CC

Rafael Henrique Bastos Pessoa

João Monlevade, MG
2022

Rafael Henrique Bastos Pessoa

**Estudo de Técnicas de Otimização para
Sintonia de Controladores PID Aplicados ao
Acionamento de Máquinas CC**

Trabalho de Conclusão de curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.

Orientador: Prof. Dr. Márcio Feliciano Braga

Coorientador: Prof. Me. Thainan Santos Theodoro

**Universidade Federal de Ouro Preto
João Monlevade
2022**

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

P475e Pessoa, Rafael Henrique Bastos.

Estudo de técnicas de otimização para sintonia de controladores PID aplicados ao acionamento de máquinas CC. [manuscrito] / Rafael Henrique Bastos Pessoa. - 2022.

73 f.: il.: color., tab..

Orientador: Prof. Dr. Márcio Feliciano Braga.

Coorientador: Prof. Me. Thainan Santos Theodoro.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia Elétrica .

1. Motores elétricos de corrente contínua. 2. Controladores PID. 3. GRASP (Sistema operacional de computador). 4. Inteligência artificial. 5. Otimização combinatória. 6. Programação heurística. I. Braga, Márcio Feliciano. II. Theodoro, Thainan Santos. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 621.313

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526



FOLHA DE APROVAÇÃO

Rafael Henrique Bastos Pessoa

Estudo de Técnicas de Otimização para Sintonia de Controladores PID Aplicados ao Acionamento de Máquinas CC

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica

Aprovada em 21 de junho de 2022

Membros da banca

D.Sc. - Márcio Feliciano Braga - Orientador (Universidade Federal de Ouro Preto)
M.Sc. - Thainan Santos Theodoro - Coorientador (Universidade Federal de Ouro Preto)
D.Sc. - Renan Fernandes Bastos - (Universidade Federal de Ouro Preto)
D.Sc. - Víctor Costa da Silva Campos - (Universidade Federal de Minas Gerais)

Márcio Feliciano Braga, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 15/07/2022



Documento assinado eletronicamente por **Marcio Feliciano Braga, PROFESSOR DE MAGISTERIO SUPERIOR**, em 15/07/2022, às 09:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0362925** e o código CRC **4C6F2859**.

Agradecimentos

A Deus, primeiramente, pelo dom da vida, por me abençoar durante toda a minha caminhada e por me amar incondicionalmente, me dando forças e ajudando a superar todos os desafios.

À minha esposa, Larissa, por participar de cada momento e cada detalhe de minha vida, por me ajudar a me superar e por ser a melhor companheira que eu poderia ter. Sem ela, pouca coisa seria possível.

Aos meus pais, José Antônio e Ivana, por tudo que fizeram por mim, por todo amor e sacrifício dedicado, pela paciência, confiança e incentivo em todos os momentos. Também ao meu irmão, André, por todo apoio e vivência diante de todas as dificuldades.

A todos os professores que contribuíram em meu caminho até aqui, em especial aos professores Márcio Feliciano Braga e Thainan Santos Theodoro, por aceitarem me orientar, por estarem sempre disponíveis e pacientes, e por toda a ajuda desde então.

Por fim, a todos os meus amigos que caminharam junto comigo, pelo incentivo e força nos momentos difíceis e pelo companheirismo que construímos, fundamental em algumas de nossas etapas.

*“É possível viver a alma sacerdotal, aquela que não passa por ninguém sem ter deixado
um pouco de luz.”*
Italo Marsili

Resumo

Os motores de corrente contínua, mesmo com a evolução dos dispositivos de estado sólido, ainda têm grande aplicação nos dias atuais, e isso se deve à sua versatilidade e facilidade para se trabalhar, principalmente quando queremos controlá-los. Junto a isso, os controladores do tipo Proporcional, Integral e Derivativo são os mais utilizados na indústria, participando aproximadamente de 97% de todas as malhas de controle. Todavia, esse tipo de controlador pode causar muitos problemas se a forma como são sintonizados não for adequada. Em muitas situações, pode ser difícil encontrar a melhor sintonia utilizando somente as estratégias convencionais, o que faz com que os projetistas o sintonizem na tentativa e erro, isto é, manualmente. Dessa forma, neste trabalho, tentamos encontrar uma melhor solução de sintonia, que seja rápida e precisa, para controladores PID, na aplicação de um sistema de controle em cascata que aciona um motor de corrente contínua. Para tal, utilizamos dois algoritmos de meta-heurística, o *Simulated Annealing* e o GRASP. É feita, a partir desse estudo, a comparação entre a sintonia por meta-heurísticas e por outros métodos conhecidos na literatura baseando-se na qualidade de resposta de saída, uma vez que critérios de desempenho são preestabelecidos. Comparando os resultados, o GRASP foi o que se mostrou mais eficiente. Para os dois controladores, de velocidade e corrente, foi o que obteve melhor desempenho, além de ter tido o menor esforço de controle. O *Simulated Annealing* também obteve bons resultados para todos os casos, sendo melhor que a estratégia convencional na velocidade, mas sendo pior na corrente. No entanto, mostrou-se uma excelente solução de sintonia automática.

Palavras-chave: Máquina CC. Controladores PID. *Simulated Annealing*. GRASP. Inteligência artificial. Otimização. Meta-heurísticas.

Abstract

The direct current motors, even with the evolution of the solid-state devices, are still widely used nowadays, and this is due to their versatility and easy to work, especially when we want to control them. Along with this, the Proportional-Integral-Derivative controllers are the most used in the industry, occupying about 97% of all control loops. However, this kind of controller may cause a lot of problems if the way they are tuned is not appropriate. In many situations, it can be hard to find the best tune using just conventional strategies, which makes the designers to tune based on trial and error, in other words, manually. Thus, in this work, we try to find a better tuning solution, which is fast and precise, for PID controllers in the application of a cascade control system that drives a direct current motor. For such, we will use two metaheuristic algorithms, *Simulated Annealing* and GRASP. From this study, a comparison is made between the tuning by metaheuristics and by other methods known in the literature, based on the quality of the output response, since performance criteria are pre-established. Comparing the results, GRASP was the one that proved to be more efficient. Considering the two controllers, for speed and current, it was the one with the best performance, besides, it had the least control effort. The Simulated Annealing also had good results in all cases, being better than the conventional strategy at the speed, but worse at the current. However, proved to be an excellent automatic tuning solution.

Keywords: DC Machine. PID Controllers. Simulated Annealing. GRASP. Artificial Intelligence. Optimization. Metaheuristics.

Lista de ilustrações

Figura 1 – Esquema simples de uma máquina CC.	14
Figura 2 – Sistema de controle em malha fechada (realimentado).	15
Figura 3 – Sistema de controle em cascata.	16
Figura 4 – Sistema de controle antecipatório.	17
Figura 5 – Exemplo de minimização de uma função.	19
Figura 6 – Circuito motor CC de excitação independente.	22
Figura 7 – Sistema motor CC mais carga acoplados.	23
Figura 8 – Conversor CC-CC ponte H <i>chopper</i>	26
Figura 9 – Gráfico torque $\times$ velocidade quatro quadrantes.	27
Figura 10 – Funcionamento do PWM representado por formas de onda.	28
Figura 11 – Sistema de acionamento da máquina CC em malha aberta.	28
Figura 12 – Sistema de controle do projeto completo entrelaçado.	30
Figura 13 – Desentrelaçamento da malha de controle do projeto.	30
Figura 14 – Sistema de controle do projeto completo.	31
Figura 15 – Sistema de controle do projeto com controlador antecipatório.	31
Figura 16 – Sistema de controle do projeto com controlador antecipatório sem efeito perturbativo.	32
Figura 17 – Lugar geométrico das raízes resultantes para ζ e σ constantes.	35
Figura 18 – Aplicação da estratégia <i>anti-windup</i> no controlador PI.	37
Figura 19 – Fluxograma que descreve a meta-heurística <i>Simulated Annealing</i>	42
Figura 20 – Fluxograma que descreve a meta-heurística <i>GRASP</i>	43
Figura 21 – Resposta ao degrau do sistema de controle com controlador via <i>sisotool</i>	49
Figura 22 – Sistema de controle do projeto completo com valores numéricos.	49
Figura 23 – Bloco da máquina CC no simulink.	51
Figura 24 – Conversor CC-CC ponte H no simulink.	51
Figura 25 – Circuito PWM no simulink.	52
Figura 26 – Sistema de controle completo montado no <i>simulink</i>	52
Figura 27 – Velocidade da máquina CC com sistema funcionando sem <i>anti-windup</i>	53
Figura 28 – Corrente da máquina CC com sistema funcionando sem <i>anti-windup</i>	53
Figura 29 – Sinal de controle proveniente do controlador de corrente máquina CC com sistema funcionando sem <i>anti-windup</i>	54
Figura 30 – Velocidade da máquina CC com sistema funcionando com <i>anti-windup</i>	55
Figura 31 – Corrente da máquina CC com sistema funcionando com <i>anti-windup</i>	56
Figura 32 – Sinal de controle proveniente do controlador de corrente máquina CC com sistema funcionando com <i>anti-windup</i>	56

Figura 33 – Gráfico da função custo no processo de otimização da planta de velocidade utilizando o algoritmo <i>Simulated Annealing</i>	58
Figura 34 – Resposta ao degrau da malha de velocidade a partir do controlador $K_P = 260.8836$ e $K_I = 20520.6195$ encontrado utilizando o algoritmo <i>Simulated Annealing</i>	59
Figura 35 – Gráfico da função custo no processo de otimização do controlador de corrente utilizando o algoritmo <i>Simulated Annealing</i>	60
Figura 36 – Resposta ao degrau da malha de corrente a partir do controlador $K_P = 0.0132$ e $K_I = 0.6373$ encontrado utilizando o algoritmo <i>Simulated Annealing</i>	61
Figura 37 – Gráfico da função custo no processo de otimização da planta de velocidade utilizando o algoritmo GRASP.	63
Figura 38 – Resposta ao degrau da malha de velocidade a partir do controlador $K_P = 41.5323$ e $K_I = 483.4974$ encontrado utilizando o algoritmo <i>Simulated Annealing</i>	63
Figura 39 – Gráfico da função custo no processo de otimização do controlador de corrente utilizando o algoritmo GRASP.	64
Figura 40 – Resposta ao degrau da malha de corrente a partir do controlador $K_P = 0.2795$ e $K_I = 334.4815$ encontrado utilizando o algoritmo GRASP.	65
Figura 41 – Resposta da velocidade comparativa entre os controlares projetados por <i>sisotool</i> , SA e GRASP.	66
Figura 42 – Resposta da corrente comparativa entre os controlares projetados por <i>sisotool</i> , SA e GRASP.	67
Figura 43 – Resposta da corrente comparativa entre os controlares projetados por <i>sisotool</i> , SA e GRASP para um menor tempo.	68
Figura 44 – Esforço de controle comparativo entre os controlares projetados por <i>sisotool</i> , SA e GRASP.	69

Lista de tabelas

Tabela 1 – Critérios de desempenho para a malha de controle.	35
Tabela 2 – Critérios de desempenho para a malha de velocidade.	36
Tabela 3 – Penalidades da função objetivo das meta-heurísticas para o controlador de velocidade.	40
Tabela 4 – Penalidades da função objetivo das meta-heurísticas para o controlador de corrente.	41
Tabela 5 – Parâmetros da máquina CC, carga e engrenagem	46
Tabela 6 – Parâmetros da simulação utilizados durante a utilização do <i>Simulink</i> . .	50
Tabela 7 – Parâmetros utilizados para a implementação do algoritmo SA.	57
Tabela 8 – Resultado da otimização na sintonia do controlador de velocidade utilizando o algoritmo <i>Simulated Annealing</i>	58
Tabela 9 – Resultado da otimização na sintonia do controlador de corrente utilizando o algoritmo <i>Simulated Annealing</i>	60
Tabela 10 – Parâmetros utilizados para a implementação do algoritmo GRASP. . .	61
Tabela 11 – Resultado da otimização na sintonia do controlador de velocidade utilizando o algoritmo GRASP.	62
Tabela 12 – Resultado da otimização na sintonia do controlador de corrente utilizando o algoritmo GRASP.	64
Tabela 13 – Comparação numérica entre os controladores de velocidade projetados por <i>sisotool</i> , SA e GRASP.	66
Tabela 14 – Comparação numérica entre os controladores de corrente projetados por <i>sisotool</i> , SA e GRASP.	67
Tabela 15 – Comparação de tempo de assentamento sem atraso de transporte entre os controladores de corrente projetados por <i>sisotool</i> , SA e GRASP. . .	68

Lista de siglas

CA	Corrente Alternada
CC	Corrente Contínua
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
IGBT	<i>Insulated Gate Bipolar Transistor</i>
ISA	<i>International Society of Automation</i>
ITAE	<i>Integral of Time and Absolute Error</i>
LRC	Lista Restrita de Candidatos
LGR	Lugar Geométrico das Raízes
MATLAB	<i>Matrix Laboratory</i>
PI	<i>Proportional-Integral</i>
PID	<i>Proportional-Integral-Derivative</i>
PWM	<i>Pulse Width Modulation</i>
SA	<i>Simulated Annealing</i>

Sumário

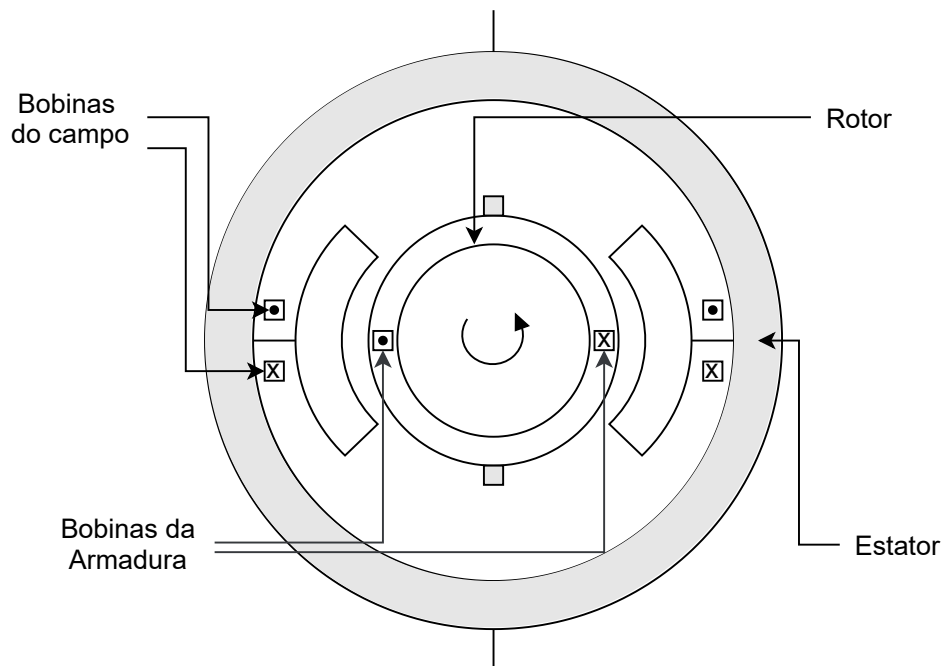
1	INTRODUÇÃO	14
1.1	Justificativa	17
1.2	Objetivos	21
1.2.1	Objetivos Específicos	21
1.3	Estrutura do Trabalho	21
2	FUNDAMENTOS TEÓRICOS	22
2.1	Motor CC de excitação independente mais carga	22
2.2	Controle em cascata do sistema	25
2.2.1	Conversores CC-CC e PWM	25
2.2.2	Sistema em cascata completo	28
2.2.3	Sistema de Controle Antecipatório	31
2.3	Projeto de controladores PID	32
2.3.1	Projeto do controlador e critérios de projeto	33
2.3.1.1	Critérios para a malha de corrente	35
2.3.1.2	Critérios para a malha de velocidade	36
2.3.2	Estratégia <i>anti-windup</i>	37
2.4	Algoritmos de meta-heurística	38
2.4.1	Variáveis de decisão e a busca na vizinhança	38
2.4.2	Função objetivo e restrições	39
2.4.3	Simulated Annealing	41
2.4.4	GRASP	42
2.5	Conclusões Parciais	44
3	APLICAÇÃO DAS TÉCNICAS APRESENTADAS NO CONTROLE DO MOTOR CC	46
3.1	Modelo numérico do motor CC e controlador projetado	47
3.1.1	Dinâmica das plantas	47
3.1.2	Controladores projetados	48
3.2	Resultados do sistema de controle da máquina CC	50
3.2.1	Resultados da máquina CC funcionando com controlador sem <i>anti-windup</i>	50
3.2.2	Resultados da máquina CC funcionando com controlador com <i>anti-windup</i>	54
3.3	Aplicação do algoritmo <i>Simulated Annealing</i>	57
3.3.1	Resultados encontrados para o controlador de velocidade	57
3.3.2	Resultados encontrados para o controlador de corrente	59
3.4	Aplicação do algoritmo GRASP	61

3.4.1	Resultados encontrados para o controlador de velocidade	62
3.4.2	Resultados encontrados para o controlador de corrente	62
3.5	Resultados dos controladores de velocidade na máquina	65
3.6	Resultados dos controladores de corrente na máquina	66
3.7	Conclusões parciais	69
4	CONCLUSÃO E TRABALHOS FUTUROS	71
4.1	Trabalhos Futuros	73
	REFERÊNCIAS	74

1 Introdução

As máquinas de Corrente Contínua (CC) são extremamente utilizadas em vários tipos de indústrias nos dias atuais, sendo em muitas delas, a maioria dentre os outros tipos de máquinas presentes. De acordo com Umans (2014), as máquinas CC podem produzir diferentes combinações quanto à excitação de seus enrolamentos de campo, podendo ser em derivação, série, composto, ímã permanente ou independente. Sendo assim, a máquina CC possui como principal característica a versatilidade, fazendo com que as análises de tensão *versus* corrente e velocidade *versus* conjugado tenham características variadas. A Figura 1 mostra um esquema simples de uma máquina CC.

Figura 1 – Esquema simples de uma máquina CC.



Fonte: Do autor.

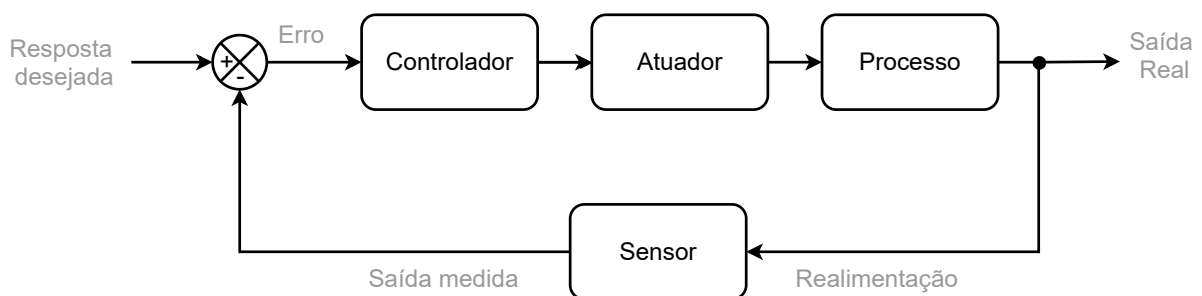
O rotor é a parte móvel da máquina, ou a parte que gira, e é onde se encontram as bobinas de armadura. O estator é a parte fixa da máquina e é onde se encontram as bobinas de campo. Como dito, há várias formas de se excitar o campo de uma máquina CC e uma delas é a excitação independente, que é a que vamos utilizar para este trabalho. Como o próprio nome diz, o campo neste tipo de máquina possui um circuito isolado, alimentado por uma fonte de tensão que conecta seus terminais diretamente nos terminais da máquina fazendo circular uma corrente nas bobinas. Uma vantagem dessa abordagem é que a corrente e a tensão no campo são sempre fixas, não sofrendo interferência do sistema em que a máquina está, seja ele para acionamento de uma carga ou para geração

de energia, diferente das outras abordagens em que o campo e a armadura pertencem ao mesmo circuito.

Assim sendo, neste trabalho faremos o uso da máquina CC como motor para acionamento de uma carga. Os motores CC são facilmente controlados e têm sido bastante aplicados em acionamentos com velocidade variável, isso dado a sua possibilidade de controle de velocidade em uma grande faixa e também pelo fato de, em sua maioria, terem um controle mais simples e barato que o controle de um acionamento de Corrente Alternada (CA) (RASHID, 2014).

Dito tudo isso, devemos projetar um sistema de controle para que o acionamento CC atinja um dado desempenho, não deixando a máquina ter uma sobrecarga, muito comum na partida, uma vez que o torque e a corrente são muito altos devido ao esforço inicial feito, e fazendo com que o erro seja o menor possível, dado que a máquina deve atuar a uma velocidade desejada. Posto isso, um sistema de controle convencional é composto por um controlador, um atuador, um processo e uma realimentação, que pode ser unitária ou não, comumente chamado de sistema em malha fechada ou sistema com realimentação, mostrado na Figura 2. Um sistema em malha fechada mede a saída do processo e a compara com a referência, ou resposta desejada, gerando um sinal de erro e aplicando-o no controlador. O controlador por sua vez deve corrigir o erro e, portanto, envia um sinal de controle para o atuador, o qual ajusta o processo para que o erro seja reduzido (DORF; BISHOP, 2018).

Figura 2 – Sistema de controle em malha fechada (realimentado).

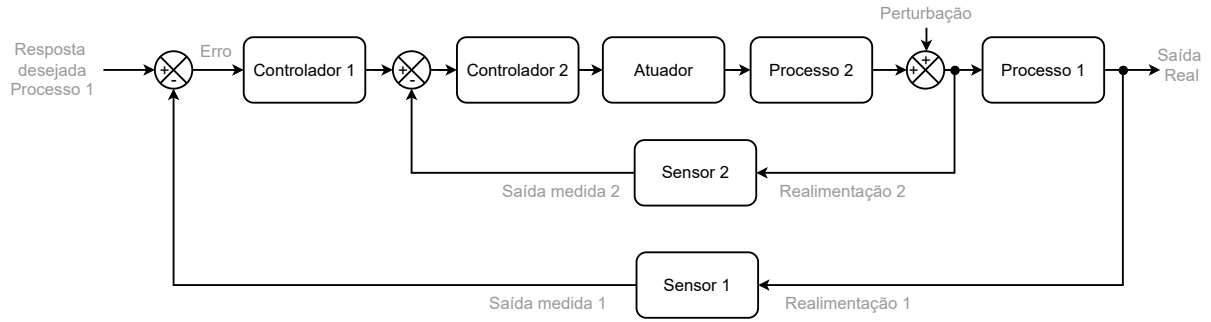


Fonte: Do autor.

Tendo isso em vista, o controle realizado por uma única malha de realimentação tem uma desvantagem, que é quanto à correção para possíveis perturbações existentes. O controlador, como dito, atua a partir de um erro existente, ou seja, somente quando houver uma variação na saída em relação a referência que ele é capaz de atuar. Todavia, caso haja perturbações, isso se torna uma desvantagem, pois é necessário que as perturbações sejam contornadas de forma rápida e com robustez, e mesmo que haja robustez na ação do controlador, ele é lento para resolver este problema, ou seja, ele não tem uma ação preventiva. Dessa forma, há várias estratégias de controle existentes que ajudam a reparar tal problema e uma delas é o controle em cascata, técnica que é largamente aplicada na

indústria, mostrada na Figura 3. Nesta estratégia são aplicadas duas malhas de controle por realimentação. Dessa forma, um controlador a mais é alocado e um segundo ponto de medição também, sendo esse último alocado de forma que ele reconheça a perturbação presente antes da variável controlada, ajudando a resolver o problema citado anteriormente. A malha interna, chamada de secundária ou escrava, e a malha externa, chamada de primária ou mestre, possuem períodos de resposta diferentes. Supondo o período de resposta da malha mestre τ_1 e o da malha escrava τ_2 , para o controle em cascata funcionar corretamente, é necessário que $\tau_1 \geq 4\tau_2$, ou seja, a malha interna deve ser muito mais rápida que a malha externa. Visa-se, assim, resolver o problema da perturbação na variável manipulada muito antes de seu controlador atuar e que as perturbações sejam controladas de forma mais rápida (FRANCHI, 2011).

Figura 3 – Sistema de controle em cascata.



Fonte: Do autor.

Posto isso, os controladores para o sistema devem ser projetados de maneira adequada. De acordo com Castrucci, Bittar e Sales (2018), apesar do grande avanço tecnológico para o controle de aplicações modernas, uma arquitetura ainda muito utilizada, sendo de longe a mais utilizada na indústria, é a do controlador *Proportional-Integral-Derivative* (PID). Esse tipo de controlador demonstra, depois de vários anos, eficácia e praticidade nas aplicações industriais e isso se deve à sua simplicidade funcional, pois com o ajuste de seus parâmetros, é possível obter um bom desempenho. Sendo assim, o modelo matemático de um controlador PID, de acordo com o padrão da *International Society of Automation* (ISA), é dado por

$$m(t) = K_c \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right), \quad (1.1)$$

sendo $m(t)$ o sinal de saída do controlador, $e(t)$ o sinal de entrada e K_c , T_I e T_D os parâmetros de ajuste. O nome PID deriva do fato de sua expressão, assim como apresentada em (1.1), ser composta pela soma das ações proporcional, integral e derivativa. Contudo, tradicionalmente, os parâmetros de ajuste do controlador são apresentados como ganhos proporcional, integral e derivativo dentro de uma expressão no domínio da frequência, dada por

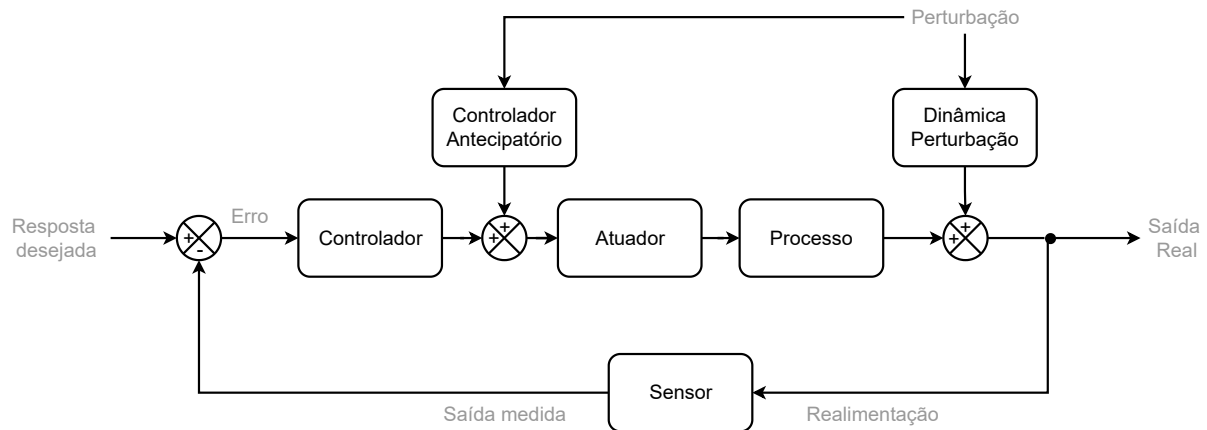
$$G_{pid}(s) = K_P + K_I \frac{1}{s} + K_D s, \quad (1.2)$$

em que

- $K_p = K_c$, ganho proporcional;
- $K_I = \frac{K_c}{T_I}$, ganho integral;
- $K_D = K_c T_D$, ganho derivativo;
- $G_{pid}(s) = \frac{M(s)}{E(s)}$, função de transferência do controlador.

Outro tipo de estratégia muito conhecida, que auxilia na rejeição às perturbações, além do sistema de controle em cascata, é o controle antecipatório, mostrado na Figura 4. O controle consiste na antecipação da perturbação na carga, controlando o processo antes que um impacto significativo aconteça na variável de projeto. Como pode ser visto, ele é utilizado juntamente com o controle por realimentação e conta com um controlador antecipatório, que, a partir da perturbação, faz os ajustes necessários no processo. É uma estratégia que têm larga utilização na indústria, principalmente no controle de alguns equipamentos como caldeiras, evaporadores, secadores, entre outros (FRANCHI, 2011).

Figura 4 – Sistema de controle antecipatório.



Fonte: Do autor.

1.1 Justificativa

Existem várias estratégias que nos permitem encontrar os parâmetros mais adequados de um controlador PID, desenvolvidas ao decorrer dos anos devido ao grande uso dessa estrutura de controlador, o que nos pode dar a entender que todos os problemas a cerca desse controlador estejam solucionados. De acordo com Franchi (2011), mais de

97% dos controladores são do tipo PID, sendo sua maioria configurada como *Proportional-Integral* (PI). Sendo assim, a partir de um estudo feito sobre o desempenho das malhas de controle nos EUA, tem-se que:

- Apenas 32% das malhas podiam ser classificadas com desempenho aceitável ou excelente;
- 32% dos controladores apresentavam um desempenho ruim, seja respondendo de forma lenta ou oscilatória;
- 36% dos controladores estavam em malha aberta, ou seja, eram manuais.

Essas conclusões foram baseadas a partir da análise de mais de 26.000 malhas de controles na indústria, o que implica que o desempenho das malhas não estava adequado a partir do que se esperava. Tendo esses resultados, vemos que nem todos os problemas foram solucionados e em muitas aplicações, os controladores têm seus parâmetros sintonizados manualmente, e mesmo quando são sintonizados por meio de alguma estratégia convencional, não se obtém uma sintonia ótima, constituindo um desafio de engenharia ainda presente. Dos benefícios que uma sintonia ótima pode trazer, podemos citar o aumento da vida útil de atuadores, melhor velocidade de resposta, entre outros. Assim sendo, a sintonia dos controladores PID por meio do uso de meta-heurísticas, busca corrigir as deficiências das estratégias convencionais e encontrar um padrão de sintonia para que possamos chegar a uma estratégia ótima, tema que vem sendo debatido durante os últimos anos (SOUZA, 2013).

Dito isso, precisamos entender primeiro o que é otimização. Segundo Souza (2005), a otimização de um sistema significa projetá-lo com maior eficiência e menor custo, fazendo com que o rendimento seja ótimo ou próximo do ótimo, dado um critério prévio de busca. A busca é feita sem a necessidade de testar todas as possibilidades, fazendo com que não sobrecarregue o processo, e pretende determinar a melhor correlação entre as variáveis de projeto. Neste sentido, a otimização tem ganhado seu espaço em vários campos da engenharia, tais como, sistemas de geração fotovoltaica, identificação de sistemas etc. Dessa maneira, um problema de otimização pode ser formulado da seguinte forma:

$$\begin{aligned} \min_{x_1, x_2, \dots, x_n} \quad & f(x_1, x_2, \dots, x_n) \\ \text{s.t.} \quad & g_1(x_1, x_2, \dots, x_n) \leq b_1, \\ & \vdots \\ & g_m(x_1, x_2, \dots, x_n) \leq b_m, \end{aligned}$$

em que

- $x_1, x_2, \dots, x_n$: variáveis de projeto;

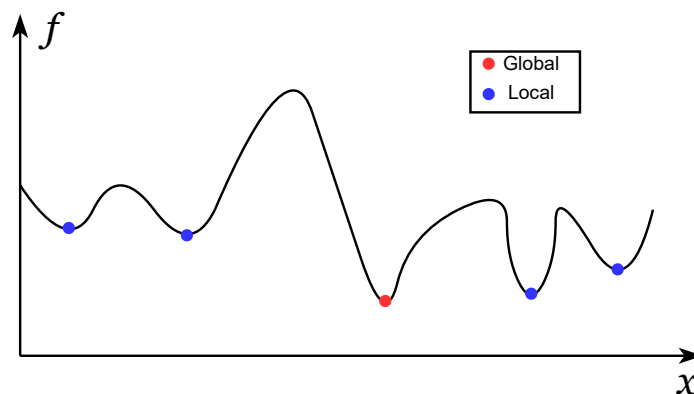
- $f(x_1, x_2, \dots, x_n)$: função objetivo;
- $g_1, g_2, \dots, g_m$: restrições.

Dessas considerações, podemos descrever que:

- Variáveis de projeto são as que se alteram durante o processo e podem ser reais, inteiras ou binárias;
- Função objetivo é a função que se deseja minimizar ou maximizar e tem uma ou mais variáveis de projeto que se deseja otimizar como parâmetros;
- Restrições são funções de igualdade ou desigualdade sobre as variáveis de projeto que descrevem quais situações de projeto não são desejáveis.

Tendo isso em vista, os métodos de otimização consistem em fazer aproximações cada vez mais perto da solução ótima, que podem ser locais ou globais. Uma das formas de busca da solução é feita por meio da implementação de estratégias heurísticas, que são algoritmos de pesquisa em que, a partir de uma solução inicial, buscam em sua vizinhança, de acordo com as características do problema, o melhor valor da função objetivo, isto é, o melhor valor dentre a busca. Supondo uma função objetivo f que queiramos minimizar e uma variável de projeto x , se utilizarmos um método de otimização heurístico adequado, provavelmente encontraríamos resultados parecidos como o mostrado na Figura 5. O que variaria esses resultados é a solução inicial que atribuiríamos ao projeto, sendo assim, ao otimizar um problema utilizando estratégias heurísticas, há chances de encontrarmos bons resultados, mas não necessariamente vamos encontrar os melhores, o que implica que poderíamos ficar presos nos ótimos locais (pontos em azul na Figura 5), e jamais chegar aos ótimos globais (ponto em vermelho na Figura 5).

Figura 5 – Exemplo de minimização de uma função.



Fonte: Do autor.

De todo modo, os métodos heurísticos, normalmente, possuem o problema de serem muito específicos para os problemas que foram criados. Assim sendo, os métodos

meta-heurísticos foram criados com o objetivo de fornecer métodos heurísticos que podem ser adaptados para diversos casos, e são procedimentos que guiam a otimização de forma mais inteligente, devolvendo soluções com uma maior qualidade e com um tempo de resposta aceitável (CARVALHO; ALMEIRA; ROCHA, 2020). Os algoritmos meta-heurísticos podem se dividir em duas categorias, de acordo com o princípio utilizado para exploração do espaço: busca local e busca populacional. Na busca local, a investigação é feita por meio de movimentos, aplicando uma solução corrente vizinha a cada passo. Na busca populacional, a investigação é feita dentro de um conjunto de boas soluções, que ao combiná-las, tenta-se produzir soluções ainda melhores.

Dito tudo isso, para este trabalho, utilizaremos dois algoritmos meta-heurísticos a fim de se obter a sintonia ótima de controladores PID aplicados em um sistema de controle de motor em cascata. O primeiro é o *Simulated Annealing* (SA), ou recozimento simulado, baseado em princípios da termodinâmica, mais especificamente do processo de recozimento empregado na fabricação de cerâmicas, cristais e vidros (KIRPATRICK; GELLAT; VECHI, 1983). O segundo é o *Greedy Randomized Adaptive Search Procedure* (GRASP), que é um procedimento de busca adaptativa, gulosa e aleatória, que consiste em duas fases: (i) gerar uma solução aleatória e; (ii) fazer a busca local a partir dessa solução (FEO; RESENDE, 1995). Ambos são métodos antigos e têm sido bastante citados em processos recentes, portanto, indicando a obtenção de bons resultados.

Cita-se, por exemplo, a dissertação de mestrado de Souza (2013), que utiliza a meta-heurística SA para encontrar a melhor sintonia de controladores PID no controle em cascata de um motor CC, em que o algoritmo retorna bons resultados, mas não melhora expressivamente o desempenho do sistema. O artigo de Zhang et al. (2018) utiliza o algoritmo SA para desenvolver uma nova metodologia eficiente para modelar e dimensionar um sistema híbrido de energia renovável, considerando dois tipos de armazenamento de energia. A meta-heurística é utilizada sozinha e em conjunto com outras variações, a fim de se encontrar o melhor armazenamento de energia de um sistema elétrico de fornecimento híbrido, eólico e solar.

O artigo de Corberán, Martí e Sanchis (2002) utiliza o algoritmo GRASP para resolver o problema do carteiro chinês, em que se deve encontrar um caminho de custo mínimo em um grafo de caminho fechado, onde todos os elos do gráfico devem ser percorridos pelo menos uma vez. De acordo com os testes feitos, utilizando comparações computacionais e com outras heurísticas, o GRASP foi superior às outras heurísticas e se aproximou do resultado ótimo. O artigo Skorin-Kapov e Kos (2006) tenta resolver o problema de roteamento multicast com restrição de atraso, com o máximo de atraso em uma sessão, limitado, por meio da meta-heurística GRASP. Ao final, percebeu-se que o algoritmo fornece um resultado quase ótimo em um tempo razoavelmente moderado e que, comparado a outras duas heurísticas, foi a que obteve o melhor resultado.

1.2 Objetivos

No presente trabalho, o objetivo geral é implementar técnicas de otimização heurísticas a fim de sintonizar controladores PID no acionamento de motores de corrente contínua. Para isso, serão comparados os desempenhos de dois algoritmos de meta-heurísticas, o *Simulated Annealing* e o *Greedy Randomized Adaptive Search Procedure*, na busca por uma melhor sintonia de controladores PID atuando em cascata e considerando uma malha anti-windup.

1.2.1 Objetivos Específicos

- Modelar e estruturar o sistema composto por uma máquina CC e um conversor CC-CC, e projetar controladores PID para o controle do sistema;
- Estudar e implementar algoritmos meta-heurísticos para que se encontre os melhores parâmetros dos controladores PID;
- Verificar a qualidade da resposta de saída do sistema quanto à utilização de meta-heurísticas na solução de parâmetros dos PID;
- Comparar os resultados obtidos por meio das meta-heurísticas com os resultados obtidos por meio de estratégias convencionais na sintonia de parâmetros PID, definir se é viável sua utilização a partir dessas comparações e qual solução retorna melhores resultados.

1.3 Estrutura do Trabalho

Este trabalho é estruturado em quatro capítulos organizados da seguinte maneira:

No Capítulo 1, é feita uma introdução, baseando-se em algumas referências, a respeito de alguns pontos básicos importantes para o desenvolvimento do trabalho, mostrando de onde partiremos e onde queremos chegar.

No Capítulo 2, são apresentadas todas as equações-base para a modelagem do motor CC, como é feito o acionamento do motor, as técnicas de controle que serão utilizadas, a estrutura final do sistema, as características do algoritmo de otimização e os fundamentos das meta-heurísticas que serão utilizadas.

No Capítulo 3, são mostrados os resultados obtidos no controle do motor CC, com e sem a estratégia *anti-windup*, os controladores obtidos por meio das meta-heurísticas SA e GRASP, utilizando controlador antecipatório para encontrar o controlador de corrente, e a aplicação dos controladores ao sistema.

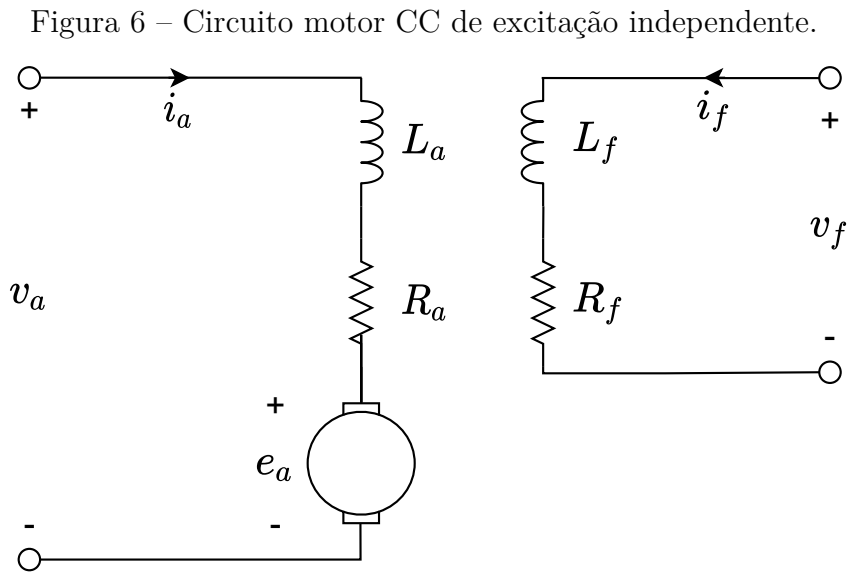
Por fim, o Capítulo 4 fornece uma conclusão final a partir de todos os resultados encontrados e apresenta possíveis etapas para a continuação deste trabalho.

2 Fundamentos Teóricos

Este capítulo discorre, de forma sintética, sobre o modelo matemático da máquina CC mais uma carga, a configuração em cascata para o sistema de controle, as estratégias de projeto dos controladores e os algoritmos de meta-heurística que serão implementados. Os fundamentos teóricos mostrados neste capítulo são a base para o desenvolvimento de todo o trabalho.

2.1 Motor CC de excitação independente mais carga

Como dito anteriormente, para este trabalho, utilizaremos o motor CC na configuração de excitação independente, ou seja, a corrente de campo i_f não depende da corrente de armadura i_a . As variações que ocorrem na corrente de armadura não surtem efeito sobre a corrente de campo, sendo a corrente de campo menor que a corrente de armadura. A Figura 6 mostra o circuito equivalente deste tipo de configuração.



Fonte: Do autor.

Sendo v_a a tensão de armadura, R_a a resistência de armadura e e_a a força eletromotriz, podemos escrever, a partir do circuito apresentado (Figura 6), que

$$v_a = e_a + R_a i_a + L_a \frac{di_a}{dt},$$

isolando $\frac{di_a}{dt}$, tem-se

$$L_a \frac{di_a}{dt} = v_a - e_a - R_a i_a,$$

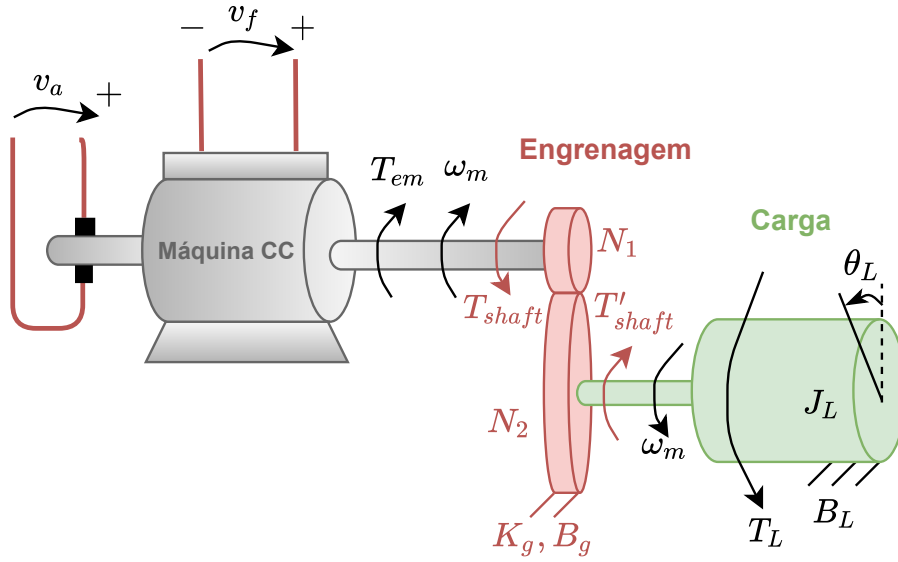
$$\frac{di_a}{dt} = \frac{v_a}{La} - \frac{e_a}{La} - \frac{R_a}{La}i_a.$$

Uma vez que a força eletromotriz pode ser representada pela multiplicação entre uma constante elétrica k_e e a velocidade angular da máquina ω_m , tem-se

$$\frac{di_a}{dt} = \frac{v_a}{La} - \frac{k_e}{La}\omega_m - \frac{R_a}{La}i_a. \quad (2.1)$$

Tendo o motor CC configurado, seu acionamento deve ser responsável pela movimentação de uma carga ao qual está acoplada, que em conjunto, compõe o sistema total. Desta forma, a Figura 7 mostra o esquema desse acoplamento.

Figura 7 – Sistema motor CC mais carga acoplados.



Fonte: Do autor.

A máquina CC é acionada da forma como mencionada anteriormente e, como podemos ver na Figura 7, o motor transmite movimento à carga por meio de engrenagens conectadas a cada um dos elementos. Isso posto, podemos então atribuir a esse sistema uma relação mecânica. Uma vez que T_{em} é o torque eletromagnético desenvolvido pelo motor, T_{shaft} é o torque no eixo, J_m é a inércia do motor e B_m é o coeficiente de atrito viscoso do motor, temos a relação dos torques dada por

$$T_{em} = J_m \frac{d\omega_m}{dt} + T_{shaft} + B_m \omega_m,$$

isolando $\frac{d\omega_m}{dt}$, tem-se

$$J_m \frac{d\omega_m}{dt} = T_{em} - T_{shaft} - B_m \omega_m,$$

$$\frac{d\omega_m}{dt} = \frac{T_{em}}{J_m} - \frac{T_{shaft}}{J_m} - \frac{B_m}{J_m} \omega_m.$$

O torque desenvolvido pode ser substituído pela multiplicação entre uma constante de torque k_t e a corrente de armadura, logo,

$$\frac{d\omega_m}{dt} = \frac{k_t}{J_m} i_a - \frac{T_{shaft}}{J_m} - \frac{B_m}{J_m} \omega_m. \quad (2.2)$$

Se observarmos como se dá a estrutura de (2.2), vemos que nela só há componentes referentes da máquina CC e, além disso, há um acoplamento da máquina com uma carga mecânica, que também exercerá influência sobre como o motor se comportará. Desse modo, para que as devidas alterações sejam feitas, devemos entender primeiro como essa carga transferirá esse movimento e como a máquina CC a sentirá. Há um conjunto de engrenagens entre esses componentes e, portanto, por meio delas, fluirá toda influência mecânica que a carga faz no sistema. A relação se estabelece devido à quantidade de dentes de engrenagens do lado do motor e do lado da carga, N_1 e N_2 , respectivamente. Logo, podemos escrever que

$$B_{eq} = B_m + B_L \left(\frac{N_1}{N_2} \right)^2 \quad (2.3)$$

e,

$$J_{eq} = J_m + J_L \left(\frac{N_1}{N_2} \right)^2, \quad (2.4)$$

dado que B_m e J_m são o coeficiente de atrito viscoso e a inércia da máquina, B_L e J_L são o coeficiente de atrito viscoso e a inércia da carga, e B_{eq} e J_{eq} são o coeficiente de atrito viscoso e a inércia equivalentes, referenciados do lado da máquina.

Desse modo, podemos substituir B_m e J_m em (2.2) por B_{eq} e J_{eq} , respectivamente, garantindo que a carga exerça influência sobre o resultado modelado. Portanto, temos que

$$\frac{d\omega_m}{dt} = \frac{k_t}{J_{eq}} i_a - \frac{T_{shaft}}{J_{eq}} - \frac{B_{eq}}{J_{eq}} \omega_m. \quad (2.5)$$

Da mesma forma como feito anteriormente, o torque T_{shaft} no eixo, relacionado ao lado motor, pode ser escrito como

$$T_{shaft} = T'_{shaft} \frac{N_2}{N_1},$$

sendo T'_{shaft} o torque no eixo, relacionado ao lado da carga. Neste sentido,

$$T'_{shaft} = K_g(\theta_1 - \theta'_2) + B_g(\omega_1 - \omega'_2),$$

sendo K_g e B_g o coeficiente de rigidez e o coeficiente de atrito viscoso do eixo do lado da máquina, θ_1 e ω_1 o ângulo e a velocidade do rotor da máquina e, θ'_2 e ω'_2 dados por

$$\theta'_2 = \theta_2 \frac{N_2}{N_1} \text{ e } \omega'_2 = \omega_2 \frac{N_2}{N_1},$$

uma vez que θ_2 e ω_2 são o ângulo e a velocidade do eixo da carga, isto é, o ângulo e a velocidade refletidas no lado da máquina.

Além das outras influências citadas e inseridas em (2.5), há também influências do torque por parte da carga. Entretanto, não substituiremos o valor de T_{shaft} na equação, pois será um valor proveniente de toda uma ação externa, propriamente da carga, dentro da modelagem. De toda forma, todas as relações apresentadas serão de grande importância no futuro para quando a carga for inserida e tiver interferência direta no torque de entrada do motor CC.

2.2 Controle em cascata do sistema

Como falado anteriormente, o sistema de controle em cascata possui certas vantagens e, devido a isso, é bastante utilizado na indústria. Para o sistema em estudo, utilizaremos essa mesma abordagem, contudo, precisamos entender como é realizado o acionamento do motor CC e assim projetar a estrutura final do projeto.

2.2.1 Conversores CC-CC e PWM

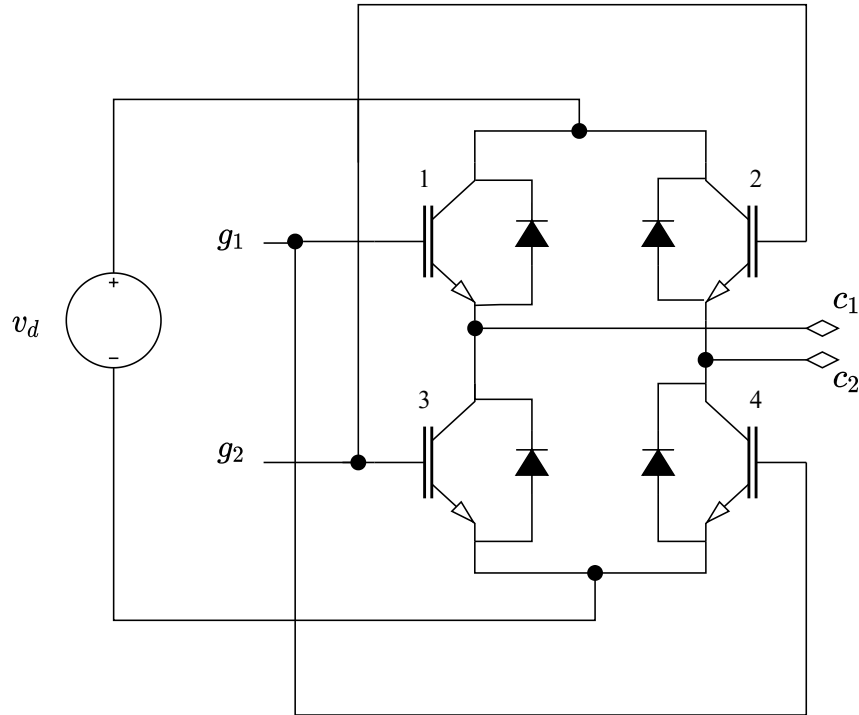
Utilizando dispositivos eletrônicos, somos capazes de converter potência elétrica de uma forma para a outra, utilizando os chamados circuitos eletrônicos de potência. Esses dispositivos são os semicondutores que, usados como chave, modificam ou controlam o valor de tensão ou corrente de um circuito. Dito isso, os circuitos eletrônicos de potência têm a finalidade de, a partir de uma fonte, corresponder às condições de tensão e corrente de uma carga e converter potência. Fazem isso modificando o tipo ou o nível de uma forma de onda de tensão ou corrente. Por esse motivo, esses circuitos são chamados de *conversores*, funcionando como uma interface entre fonte e carga (HART, 2011).

Dessa forma, os conversores classificam-se conforme a relação de entrada e saída. Por exemplo, um conversor que recebe na entrada um sinal CA e sua saída é um sinal CC é chamado de conversor CA-CC, assim, é possível formar quatro tipos de conversores, CA-CC, CC-CA, CC-CC e CA-CA. A escolha do conversor, mais uma vez, dependerá de qual é a configuração do projeto a ser executado e, portanto, depende especificamente da fonte e da carga a ser acionada.

Para este projeto, utilizaremos o conversor CC-CC, uma vez que a fonte ou o barramento de alimentação é CC e a carga, que no caso é o motor, é CC. Teoricamente, como ambos são de mesmo tipo, não seria necessário um conversor de acionamento, contudo, os conversores também podem converter níveis e fazem isso geralmente a partir da alteração dos disparos de seus dispositivos eletrônicos, podendo diminuir ou aumentar a tensão. Como o sistema que estamos trabalhando é dinâmico, caso queiramos variar a velocidade do motor, esperamos que o controlador faça essa alteração automaticamente. Para isso acontecer, a tensão de armadura deve ser alterada, e não convém que o operador mude manualmente a tensão aplicada, uma vez que o sistema deve ser automático. Sendo assim, a Figura 8 mostra o esquema de um conversor CC-CC em ponte H, chamado de

conversor *chopper*. O circuito é composto por quatro dispositivos semicondutores nomeados *Insulated Gate Bipolar Transistor* (IGBT), sendo um sinal g_1 responsável por acionar o par 1 e 4, e um sinal g_2 responsável por acionar o par 2 e 3, uma vez que esse circuito é alimentado pela tensão v_d e, c_1 e c_2 são as saídas do circuito. Há também um diodo ligado em anti-paralelo para cada IGBT, a fim de que, caso os IGBTs sofram uma sobretensão, seja evitado que ele se queime.

Figura 8 – Conversor CC-CC ponte H *chopper*.



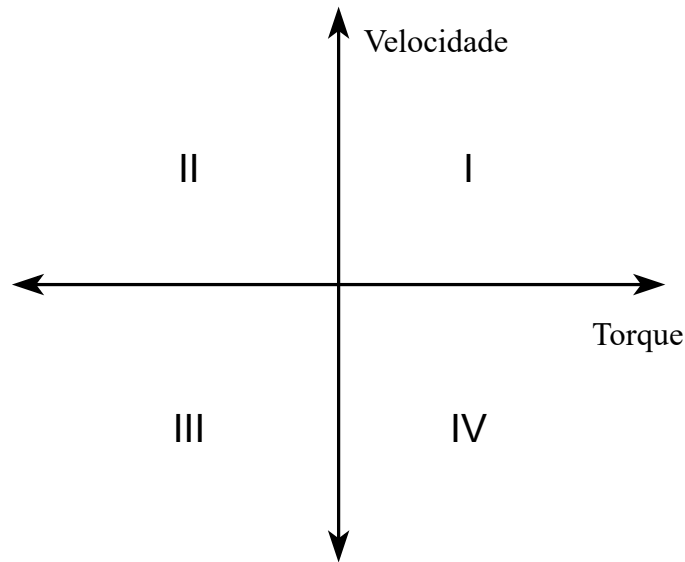
Fonte: Do autor.

Dito tudo isso, o conversor também permite que a máquina funcione nos quatro quadrantes do plano torque $\times$ velocidade, assim como ilustrado na Figura 9. Isso quer dizer que a máquina pode operar em quatro modos diferentes, um referido a cada quadrante do plano, que têm por características:

- **Quadrante I:** o torque e a velocidade são positivos, isso é, a máquina está no sentido de rotação avante e está acelerando, funcionamento motor.
- **Quadrante II:** o torque é negativo e a velocidade é positiva, ou seja, a máquina está no sentido de rotação avante, mas em uma operação de frenagem, funcionamento gerador.
- **Quadrante III:** o torque é negativo e a velocidade é negativa, portanto, a máquina está no sentido de rotação reverso ou à ré, e a máquina está acelerando, funcionamento motor.

- **Quadrante IV:** o torque é positivo e a velocidade é negativa, desse modo, a máquina está com a rotação no sentido à ré e na operação de frenagem, funcionamento gerador.

Figura 9 – Gráfico torque $\times$ velocidade quatro quadrantes.



Fonte: Do autor.

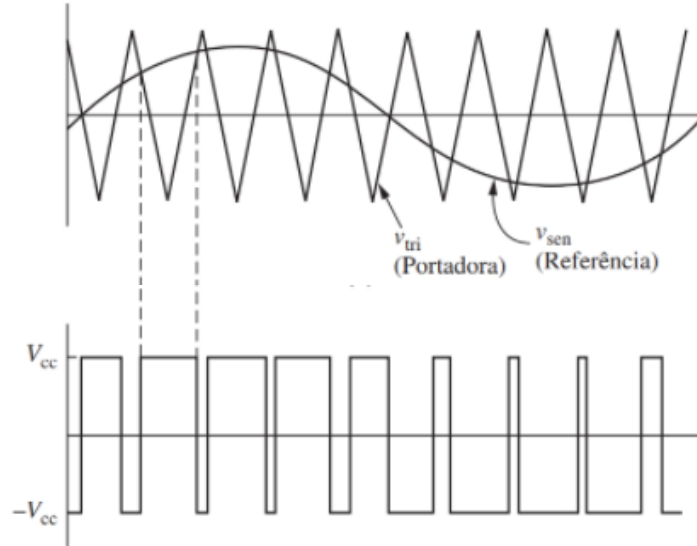
À vista disso, nosso sistema contará com esse tipo de conversor para o acionamento da máquina. O acionamento é feito a partir do disparo do semicondutor, que no caso, são os terminais chamados de portas, conectados a g_1 e g_2 . Quando o sinal emitido nas portas estão em nível lógico alto, há passagem de corrente pelo IGBT, quando está em nível lógico baixo, não há passagem de corrente, sendo g_1 o inverso de g_2 . A quantidade de tempo que a saída permanece acionada é o que determina o valor médio da tensão. Portanto, quanto maior o tempo, maior o valor da tensão e quanto menor o tempo, menor o valor de tensão. O controlador é quem determinará como esses dispositivos funcionarão a partir da condição que se encontra o sistema. Todavia, essa conexão entre o controle e o conversor não é feita diretamente, dado que o sinal de controle não pode ser responsável pelos acionamentos. Desse modo, é necessário que utilizemos a técnica nomeada *Pulse Width Modulation* (PWM), que receberá o sinal de controle e agirá de forma que o conversor funcione de acordo com o que o controle estabelece.

Para ilustrar o funcionamento básico de um PWM, podemos observar a Figura 10. O primeiro gráfico mostra duas ondas, a onda triangular v_{tri} é gerada pelo próprio dispositivo e a onda v_{sen} é a entrada do PWM. Sua saída pode ser vista no segundo gráfico, v_o , que é formada a partir dos seguintes critérios:

$$v_o = V_{cc} \quad \text{para} \quad v_{sen} > v_{tri}$$

$$v_o = -V_{cc} \quad \text{para} \quad v_{sen} < v_{tri}.$$

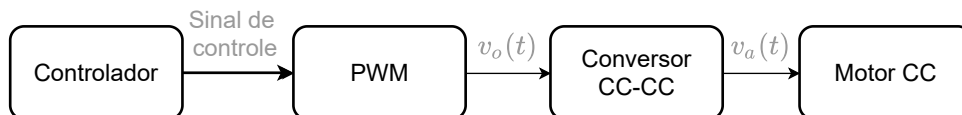
Figura 10 – Funcionamento do PWM representado por formas de onda.



Fonte: Adaptada de (HART, 2011)

Desse modo, temos que, quando o sinal de entrada for maior que o sinal triangular, a saída recebe nível alto, e quando o sinal de entrada é menor que o sinal triangular, a saída recebe nível baixo, assim sendo, temos a saída assim como descrita na Figura 10. Como dito anteriormente, o sinal resultante é aplicado ao conversor, que alimentará a tensão de armadura da máquina de acordo com a forma que o controlador indicar, pois o sinal de controle proveniente do controlador será o sinal de entrada do PWM. Finalmente, com estes dispositivos operando em conjunto, conseguimos fazer com que o controle atue sobre a máquina e o sistema responda como esperado. A visualização desse acionamento em malha aberta pode ser visto na Figura 11.

Figura 11 – Sistema de acionamento da máquina CC em malha aberta.



Fonte: Do autor.

2.2.2 Sistema em cascata completo

Tendo as informações obtidas até então, podemos começar a desenvolver teoricamente nosso sistema a fim de que ele se assemelhe ao mostrado na Figura 3. Tomemos o

resultado obtido em (2.5), reescrevendo-o no domínio de Laplace, temos que

$$s\omega_m(s) = \frac{k_t}{J_{eq}}I_a(s) - \frac{T_{shaft}(s)}{J_{eq}} - \frac{B_{eq}}{J_{eq}}\omega_m(s),$$

isolando, $\omega_m(s)$, tem-se

$$\omega_m(s) \left(s + \frac{B_{eq}}{J_{eq}} \right) = \frac{k_t}{J_{eq}}I_a(s) - \frac{T_{shaft}(s)}{J_{eq}},$$

logo,

$$\omega_m(s) = \frac{k_t I_a(s) - T_{shaft}(s)}{(J_{eq}s + B_{eq})}. \quad (2.6)$$

Do mesmo modo, tomando (2.1) e reescrevendo no domínio de Laplace, obtém-se

$$sI_a(s) = \frac{V_a(s)}{L_a} - \frac{k_e}{L_a}\omega_m(s) - \frac{R_a}{L_a}I_a(s),$$

isolando, $I_a(s)$, tem-se

$$I_a(s) \left(s + \frac{R_a}{L_a} \right) = \frac{V_a(s)}{L_a} - \frac{k_e}{L_a}\omega_m(s),$$

logo,

$$I_a(s) = \frac{V_a(s) - k_e\omega_m(s)}{(L_a s + R_a)}. \quad (2.7)$$

As duas equações apresentadas em (2.6) e (2.7) serão de grande importância para estruturação do projeto atual, pois, a partir delas, podemos montar um sistema de controle no domínio da frequência. Entretanto, antes disso, sabemos que o sistema em cascata deve possuir dois controladores para duas plantas distintas e, além disso, o sinal de controle proveniente do controlador é enviado para o PWM, que por sua vez controla o acionamento do motor a partir do conversor CC-CC. Assim sendo, a estrutura PWM mais conversor também deve ser representado junto ao modelo do sistema, uma vez que interferirá diretamente no processo. A título de simplificação, vamos assumir que não há efeito nenhum além de um ganho, que chamaremos de K_{PWM} , dado por

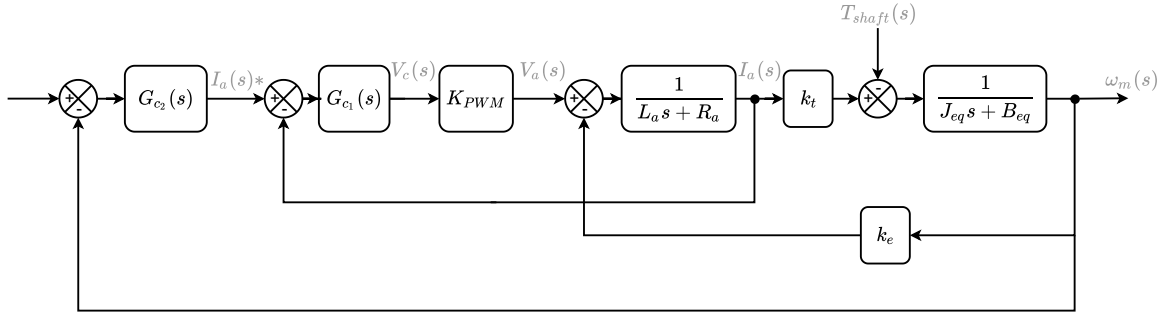
$$K_{PWM} \times v_c = v_a, \quad (2.8)$$

sendo v_c o sinal de controle e v_a a tensão aplicada à máquina.

Tendo isso em vista, a partir das informações obtidas em (2.6), (2.7) e em (2.8), juntamente com o conhecimento sobre estrutura em cascata, dispomos o sistema de controle do motor CC como ilustrado na Figura 12.

Por meio da Figura 12, vemos que o sistema não está totalmente de acordo com a proposta do sistema em cascata, visto que há um entrelaçamento de malhas em seu meio. Dessa forma, precisamos desentrelaçá-las com o intuito de, no fim, obtermos um sistema em cascata assim como definido. Para fazer isso, é necessário que a malha que sai de $\omega_m(s)$ e se soma com $V_a(s)$, passe seu ponto de partida para $I_a(s)$. Porém, ao fazê-lo,

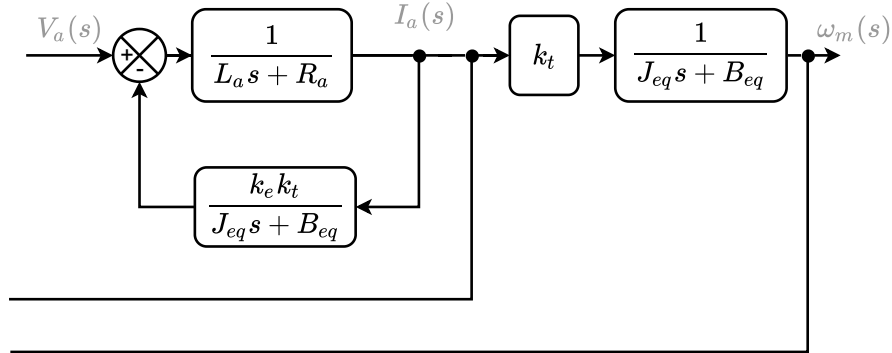
Figura 12 – Sistema de controle do projeto completo entrelaçado.



Fonte: Do autor.

devemos multiplicar todos os ganhos da malha original, de forma que a dinâmica real não seja perdida. Além disso, por simplicidade, retiraremos o valor de T_{shaft} do somatório. Sendo assim, a nova malha pode ser vista na Figura 13, na qual é mostrada, para melhor entendimento, apenas a parte que foi modificada.

Figura 13 – Desentrelaçamento da malha de controle do projeto.



Fonte: Do autor.

A partir da Figura 13, basta obter a função de transferência da malha fechada interna que se encontra entre os sinais $V_a(s)$ e $I_a(s)$. Fazendo isso, temos que

$$G_1(s) = \frac{1}{(L_a s + R_a) \left(1 + \left(\frac{1}{(L_a s + R_a)} \right) \left(\frac{k_t k_e}{J_{eq} s + B_{eq}} \right) \right)},$$

operando as manipulações,

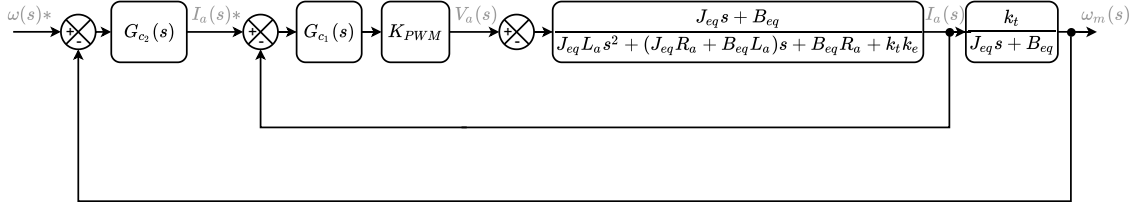
$$G_1(s) = \frac{J_{eq} s + B_{eq}}{J_{eq} L_a s^2 + (J_{eq} R_a + B_{eq} L_a) s + B_{eq} R_a + k_t k_e}. \quad (2.9)$$

Do mesmo modo, podemos simplificando a parte entre $I_a(s)$ e $\omega_m(s)$ em somente uma equação, que é descrita por

$$G_2(s) = \frac{k_t}{J_{eq} s + B_{eq}}. \quad (2.10)$$

Os resultados de $G_1(s)$ e $G_2(s)$ encontrados em (2.9) e (2.10), respectivamente, representam as duas plantas do sistema em cascata e serão utilizadas para o projeto do controlador futuramente. Dessa forma, a Figura 14 mostra o sistema final de controle em cascata do projeto.

Figura 14 – Sistema de controle do projeto completo.



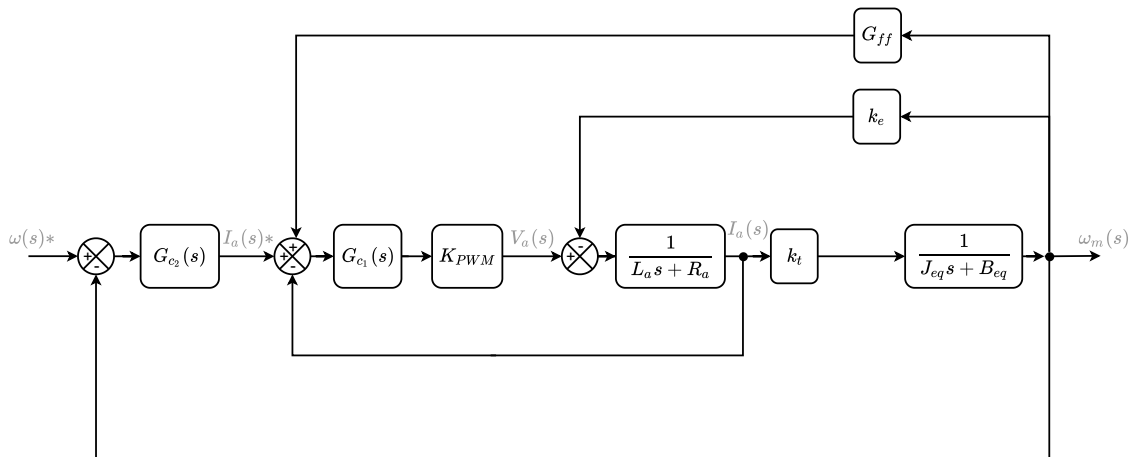
Fonte: Do autor.

Tendo feito tudo isso, temos parte dos resultados que precisamos para o controle final, pois, ainda é necessário definir como o controlador será projetado e quais serão os critérios de projeto. Posto isso, a seguir apresenta-se como se dá o controle antecipatório no projeto e, posteriormente, como o controlador PID pode atuar da melhor forma dentro de um sistema, para, ao final, obtermos respostas satisfatórias.

2.2.3 Sistema de Controle Antecipatório

Uma outra abordagem, que resolve o problema do entrelaçamento das malhas da Figura 12, é a utilização do controle antecipatório. Fazendo uma assimilação à Figura 4, podemos considerar que $\omega_m(s)$ seja a “perturbação” e k_e a dinâmica da perturbação. Portanto, o objetivo seria eliminar o efeito de k_e na malha, fazendo com que, ao final, a dinâmica da corrente seja bem mais simples que a encontrada na Figura 14. Com a aplicação do controlador, o sistema fica da forma como mostra a Figura 15.

Figura 15 – Sistema de controle do projeto com controlador antecipatório.



Fonte: Do autor.

Para projetar o controlador antecipatório $G_{ff}(s)$, é necessário considerar a perturbação como a entrada e zerar a dinâmica *Perturbação/Saída*, portanto,

$$\frac{-K_e G_{1ff} + G_{ff} G_{c1} K_{PWM} G_{1ff}}{1 + G_{c1} K_{PWM} G_{1ff}} = 0,$$

sendo,

$$G_{1ff} = \frac{1}{L_a s + R_a}. \quad (2.11)$$

Como só a parte do numerador importa para a equação, temos que:

$$-K_e G_1 + G_{ff} G_{c1} K_{PWM} G_1 = 0,$$

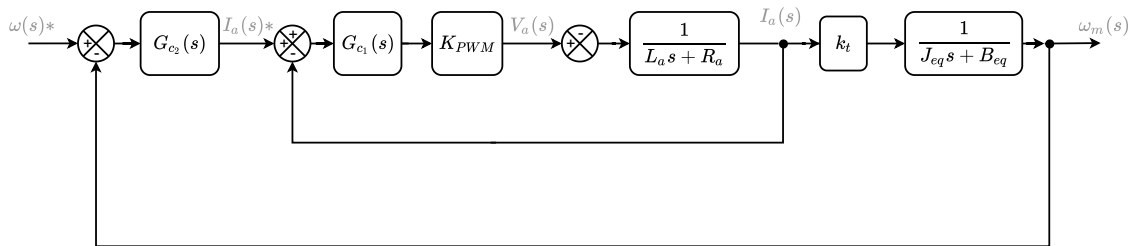
desse modo,

$$G_{ff}(s) = \frac{k_e}{G_{c1} K_{PWM}}. \quad (2.12)$$

Uma das limitações da aplicação de um controlador antecipatório é a alta dependência do modelo do processo, assim como vemos em (2.12). Se a dinâmica do processo não estiver sendo bem representada pelo modelo, o controlador antecipatório pode não funcionar tão bem quanto o esperado e comprometer todas as outras partes do sistema, levando-o até a instabilidade.

Tendo o controlador antecipatório calculado e, considerando que ele cancelará o efeito “perturbativo” do k_e , o sistema teria um comportamento assim como mostrado na Figura 16. A malha de corrente passa a ter uma dinâmica bem mais simples do que a apresentada (2.9) e pode ser uma interessante abordagem para o cômputo dos resultados.

Figura 16 – Sistema de controle do projeto com controlador antecipatório sem efeito perturbativo.



Fonte: Do autor.

2.3 Projeto de controladores PID

A partir de agora, é necessário descrever como projetaremos os controladores PID do sistema. Dessa maneira, foi mencionado anteriormente que a estrutura PI do PID é a mais utilizada dentro da indústria. Isso se deve ao fato de sua simplicidade de implementação, além de evitar os problemas decorrentes da utilização da parte derivativa que

amplifica sinais de alta frequência, exigindo que precauções adicionais sejam tomadas. Portanto, a estrutura dos controladores PI para esse sistema, a partir de (1.2), é dada por

$$G_c(s) = K_P + K_I \frac{1}{s}.$$

Tendo isso em vista, a sintonia da malha de controle é o processo no qual se pretende encontrar os melhores parâmetros de um controlador. Como mencionado, em muitos casos, os parâmetros são encontrados via tentativa e erro, o que não é necessariamente uma maneira incorreta de se fazer, porém, é um processo que pode ser demorado e trabalhoso. No entanto, atualmente, existem algumas estratégias predefinidas que auxiliam no processo de sintonia, uma vez que a partir de valores retirados do modelo, o método retorna a melhor sintonia de controlador para o sistema (FRANCHI, 2011). Essas estratégias são formuladas, normalmente, a partir de alguns critérios preestabelecidos que devem reger toda a malha projetada, tais como:

- Velocidade de resposta;
- Erro em regime permanente;
- Robustez;
- Desempenho.

O desempenho depende do que se está buscando na resposta do controle, e a robustez é a garantia de estabilidade para uma larga faixa de operação, garantindo também boa resposta às perturbações. De todo modo, essas estratégias não resolvem todos os casos, ou ainda, retornam um controlador que não possui características adequadas ao que buscamos, e isso faz com que muitos sistemas permaneçam sendo sintonizados manualmente. Sendo assim, para este trabalho, tomaremos uma abordagem computacional de projeto, utilizando uma ferramenta gráfica para a sintonia.

2.3.1 Projeto do controlador e critérios de projeto

Todo o trabalho será desenvolvido em ambiente computacional, dessa maneira, precisaremos de uma ferramenta confiável e precisa, de modo que os resultados devolvidos sejam de qualidade e não comprometam o projeto. Dito isso, utilizaremos o *software* de computação numérica *Matrix Laboratory* (MATLAB), que dentro de um dos seus pacotes fornecidos, há uma funcionalidade que permite o projeto de sintonia de controladores.

Ao abrir o programa, na aba “*Command Window*”, devemos digitar o comando “*sisotool(G)*”, sendo G a planta a ser controlada. Fazendo isso, uma nova janela será aberta, mostrando três gráficos distintos e relacionados à planta inserida, são eles:

- O diagrama de Bode do caminho direto;

- O lugar geométrico das raízes;
- A resposta ao degrau em malha fechada.

O Lugar Geométrico das Raízes (LGR) é um gráfico no plano s , no qual a partir dos polos e zeros da malha aberta do sistema, conseguimos traçar o comportamento em malha fechada, dado o ajuste do ganho do projeto. Este gráfico será de grande utilidade, em razão da possibilidade de, com base nas especificações de desempenho, ajustar o controlador de modo que as raízes do sistema fiquem dentro da região requerida. Para definir essa região, suponhamos que as duas especificações de desempenho sejam o tempo de assentamento, t_s , e o máximo sobressinal, M_p , dados por

$$t_s = \frac{4}{\zeta\omega_n} \quad (2.13)$$

e

$$M_p = e^{-\frac{\zeta}{\sqrt{1-\zeta^2}}\pi}. \quad (2.14)$$

Supondo, por exemplo, que $t_s < 4$ s, aplicando em (2.13), temos que

$$\zeta\omega_n > 1.$$

Como $\sigma = \zeta\omega_n$, logo,

$$\sigma > 1.$$

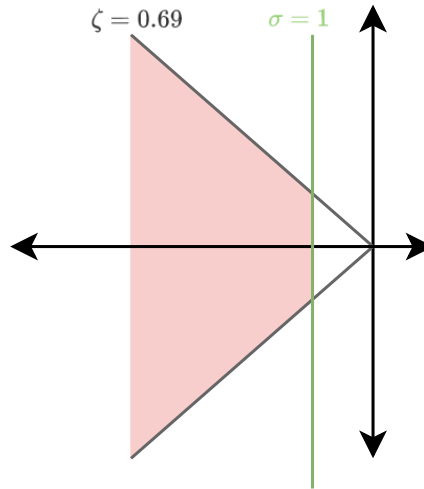
Supondo, agora que $M_p < 5\%$, aplicando em (2.14),

$$\zeta > 0.69.$$

Tendo em vista esses resultados, podemos traçar, no LGR, as retas que limitarão onde as raízes do sistema devem ficar, ilustradas da Figura 17. As retas traçadas são para σ e ζ constantes, nos valores indicados, sendo a área colorida de rosa, onde os polos da malha fechada devem estar, ou seja, é a área em que $\sigma > 1$ e $\zeta > 0.69$. Desse modo, tendo a dinâmica da planta, projetamos o controlador, de modo que as especificações sejam atendidas e é como projetaremos os controladores via *sisotool*.

A resposta ao degrau em malha fechada refere-se à saída do sistema no domínio do tempo, dado um sinal degrau unitário aplicado à sua entrada, com o sistema atuando em malha fechada. Este gráfico também será de grande importância, visto que ao ajustar os polos no gráfico do LGR, a saída do sistema mudará de acordo com o ajuste feito e, portanto, seremos capazes de posicionar os polos de forma a obter a melhor resposta de saída possível.

Para nosso caso, precisaremos fazer esse processo duas vezes, uma vez que o sistema apresentado na Figura 14 conta com dois controladores, um para cada planta do sistema.

Figura 17 – Lugar geométrico das raízes resultantes para ζ e σ constantes.

Fonte: Do autor.

2.3.1.1 Critérios para a malha de corrente

A malha de corrente do sistema em cascata é a malha interna, composta pelo controlador $G_{c1}(s)$, o qual queremos projetar, o ganho do PWM-conversor K_{PWM} e a planta a ser controlada $G_1(s)$, descrita em (2.9). Desse modo, o projeto do controlador deve considerar a multiplicação entre K_{PWM} e $G_1(s)$ como uma só planta, formando a configuração

$$G_{1_{PWM}}(s) = \frac{K_{PWM}(J_{eq}s + B_{eq})}{J_{eq}L_a s^2 + (J_{eq}R_a + B_{eq}L_a)s + B_{eq}R_a + k_t k_e}. \quad (2.15)$$

A partir do modelo descrito em (2.15), as configurações de projeto estipuladas são mostradas na Tabela 1.

Tabela 1 – Critérios de desempenho para a malha de controle.

Tipo de critério	Critério estipulado
Tempo de assentamento	≤ 0.003 s
Máximo sobressinal	$\leq 5\%$
Erro em regime permanente	Nulo

Fonte: Do autor.

A escolha do tempo de assentamento se deve ao fato da malha interna ter que ser muito mais rápida que a malha externa. Além disso, essa malha necessita que a corrente responda da forma mais rápida possível, fazendo assim com que a escolha do tempo seja adequada. O máximo sobressinal escolhido é um baixo valor, para que a máquina não se danifique na partida, por sobrecarga. O erro em regime permanente nulo é ideal para a maioria dos projetos, uma vez que a saída seguiria firmemente a entrada durante

sua atuação, e a vantagem da utilização da estrutura PI é que, o elemento integrador, principalmente quando a planta não tem um integrador, tem a função de anular o erro em regime permanente para uma entrada degrau, proporcionando robustez. Destaca-se que, em alguns casos isso não acontece, devido a fatores como não linearidades (saturação, banda morta etc.), no entanto, vamos assumir, por enquanto, que o erro nulo é alcançável somente com a utilização do PI. Destaca-se ainda que a introdução do controlador PI é capaz de rejeitar perturbações no processo.

Para o caso em que o controlador antecipatório é aplicado, partindo agora de (2.11), $G_{1ff,PWM}$ fica da forma como mostrado na equação (2.16). Da mesma maneira, para esse caso, os mesmos critérios de desempenhos da Tabela 1 são desejados.

$$G_{1PWM}(s) = \frac{K_{PWM}}{L_a s + R_a}. \quad (2.16)$$

2.3.1.2 Critérios para a malha de velocidade

A malha de velocidade do sistema em cascata é a malha externa, composta pelo controlador $G_{c2}(s)$, o qual queremos projetar, e a planta a ser controlada $G_2(s)$, descrita em (2.10).

Para essa malha, as configurações de projeto estipuladas estão listadas na Tabela 2.

Tabela 2 – Critérios de desempenho para a malha de velocidade.

Tipo de critério	Critério estipulado
Tempo de assentamento	$\geq 0.5 \text{ s}$ e $\leq 5 \text{ s}$
Máximo sobressinal	$\leq 5\%$
Erro em regime permanente	Nulo

Fonte: Do autor.

A escolha do tempo de assentamento foi feita a partir da ideia de que, além da malha externa ter que ser mais lenta, devido às características do sistema em cascata, um motor, usualmente, demora um pequeno tempo para acelerar até chegar seu estado em regime permanente, impossibilitando valores muito baixos. O máximo sobressinal foi o mesmo que o anterior e, mais uma vez, um baixo valor, a fim de não interferir negativamente na partida da máquina. O erro em regime permanente é nulo, assim como proposto para o caso anterior, devido à espera de resposta de saída com a utilização do PI e por ser uma condição ideal de operação.

A partir desses critérios e das considerações feitas sobre como o projeto será feito, temos uma forma de se chegar a bons controladores para este sistema. Entretanto, há um efeito na operação do sistema, que acontece na prática devido à limitação dos atuadores, em que a saída de controle é saturada, gerando mal funcionamento do controle. Desse

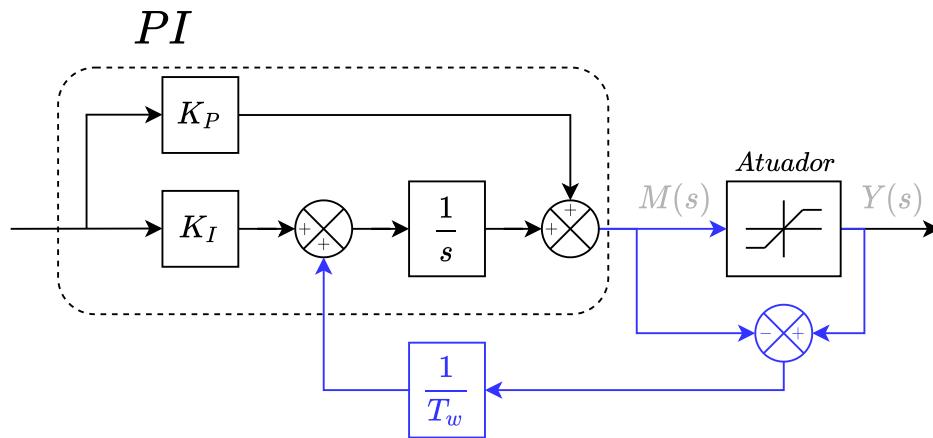
modo, para um melhor entendimento, apresentaremos na próxima seção uma estratégia que será utilizada para contornar o problema..

2.3.2 Estratégia *anti-windup*

Como mencionado, a utilização do controlador PI na prática pode trazer uma inconveniência, que se não considerada no projeto, pode gerar resultados ruins durante a operação. O nome dado a este problema é *efeito windup* e é causado devido à ação integral do controlador. Suponhamos um controlador PI atuando em uma malha de controle e que essa malha possua um desvio na saída, isso por sua vez, faz com que o modo integral do controlador continue enviando sinal de controle para ajuste do erro. Contudo, se o atuador do processo atinge um valor máximo ou mínimo, ele saturará, o erro continuará a existir e o controlador continuará a integrar, aumentando cada vez mais a saturação. Como não é uma ação esperada, isso prejudicaria o processo, quebrando a malha de realimentação (GARCIA, 2017).

Com o intuito de contornar tal problema, devemos fazer com que o controlador volte a trabalhar na região linear, ou seja, que ele saia da região de saturação, fazendo o termo integral se “descarregar”, sendo essa estratégia intitulada como *anti-windup*. Ao fazer isso, espera-se que o sinal de erro troque de sinal e, por um certo período de tempo, seja aplicado um sinal oposto de erro na entrada da ação integral do controlador. A desvantagem quanto à utilização dessa estratégia é que a resposta transitória pode ficar lenta e oscilatória, características que como mencionamos, diminuem o desempenho do sistema. Na Figura 18, mostra-se um exemplo de controlador PI projetado com a estratégia *anti-windup*.

Figura 18 – Aplicação da estratégia *anti-windup* no controlador PI.



Fonte: Do autor.

A parte em azul da Figura 18 é a que representa o *anti-windup*. Assim como mencionado, é necessário injetar um sinal de erro no controlador na parte integral para

que quando a saída sature, o sinal de controle seja recalculado de forma a permanecer no valor limite do atuador. Para tal, é calculada a diferença entre a entrada e a saída do atuador, gerando um sinal, o qual multiplicado por um ganho $\frac{1}{T_w}$ é ligado ao controlador, dando uma dinâmica de tempo atuação, e não corrigindo instantaneamente. O valor de T_w , segundo uma regra empírica sugerida, pode ser calculado como

$$T_w = \sqrt{T_D T_I} = \sqrt{\frac{K_D}{K_I}}. \quad (2.17)$$

Para o controlador do tipo PI, assume-se $K_D = 1$ ou $T_D = 1$.

Desse modo, ao encontrar o valor de T_w a partir de (2.17) e aplicar a estratégia assim como mostrado na Figura 18, o sistema será capaz de contornar o efeito *windup* e operar da forma desejada (BAZANELLA; DA SILVA JR, 2000).

Até aqui, foram apresentadas todas as ferramentas necessárias para se modelar todo o sistema de controle que queremos trabalhar, desde a modelagem da carga até o projeto de controladores, passando por todas as etapas detalhadamente. Por conseguinte, foi mencionado no início desse texto sobre a possibilidade de sintonia de controladores diferentemente das estratégias que são usuais. Desse modo, a seguir, introduziremos o funcionamento dos algoritmos meta-heurísticos mencionados e como podemos utilizá-los a fim de encontrar os melhores parâmetros do controlador PI. O objetivo, ao final, será comparar as abordagens dentro de todos os aspectos a partir de resultados obtidos em simulação.

2.4 Algoritmos de meta-heurística

As meta-heurísticas são procedimentos que, a partir de um problema de otimização, devem encontrar uma solução adequada e, em alguns casos, a melhor solução global. Atualmente, existem várias estratégias de meta-heurísticas, sendo elas diferenciadas somente pela forma como saem dos ótimos locais, o que pode se tornar também um critério de comparação entre estratégias ou confirmação de resultados (SOUZA, 2011).

Dito isso, utilizaremos as estratégias SA e GRASP para este trabalho, mas, antes de falarmos sobre o funcionamento de cada uma, devemos definir os elementos em comum entre elas que compõe a otimização, tais como variáveis de decisão, função objetivo, restrições e a busca na vizinhança.

2.4.1 Variáveis de decisão e a busca na vizinhança

O problema de otimização que desejamos resolver com este trabalho é a sintonia ótima de controladores PI. Desse modo, as variáveis de projeto serão os valores dos ganhos K_P e K_I , representados por um vetor de duas posições da forma

$$S = [K_P \ K_I]. \quad (2.18)$$

Assim sendo, é necessário possuir uma condição inicial, ou seja, devemos atribuir ao vetor S , no início do algoritmo, um valor aleatório, de forma que na primeira iteração seja possível calcular o valor da função objetivo. Posto isso, qualquer valor pode ser escolhido e deve mudar a cada iteração, dado o fato de que a heurística deve proceder com a busca local, a fim de se encontrar o melhor resultado. Portanto, além da inicialização do vetor, deve existir uma função que altere esse valor, recorrentemente. Esse processo, conhecido como geração de vizinhos, não pode percorrer por somente uma direção, ou seja, não pode somente decrescer ou crescer de valor. Consequentemente, temos que fazer com que os valores dentro do vetor S , descrito em (2.18), variem de forma aleatória e não muito dispersa. Para tal, a função receberá o vetor inicial, ou o da iteração anterior, e retornará o novo vizinho da forma

$$S' = S \times (0.95 + (1.05 - 0.95) * (Rand)),$$

sendo S' o novo valor e $Rand$ um vetor de duas posições, gerado aleatoriamente, cujos valores variam entre 0 e 1. Essa fórmula altera o valor dos ganhos em no máximo $\pm 5\%$, ou seja, não percorre somente uma direção e não dispersa os valores para outros locais muito distintos.

2.4.2 Função objetivo e restrições

Além de alterar os valores dos ganhos, devemos testar a qualidade desses valores, e dentro do problema de otimização, esse papel é realizado pela função objetivo. A função objetivo é projetada de modo particular a cada problema, bem como sua otimização, que pode ter como objetivo minimizar ou maximizar. Sendo assim, para nosso caso, a função objetivo recebe como parâmetros de entrada o vetor de sintonias descrito em (2.18) e a planta do sistema que se deseja otimizar. A partir desses parâmetros, calcula a resposta em degrau da malha fechada e retira, a partir desse cálculo, elementos importantes para o computo final, sendo eles:

- $e(t)$ = Sinal de erro;
- $s_c(t)$ = Sinal de controle;
- t_s = Tempo de assentamento;
- m_s = Máximo sobressinal;

Nosso objetivo será minimizar a função objetivo, portanto, devemos construí-la de modo que haja penalidades somadas quando não é retornado um valor desejado. Desse modo, como sabemos, é indesejável valores altos para o tempo de assentamento e máximo sobressinal, além disso, o sinal de controle não pode saturar, assim sendo, colocaremos estes valores para serem as penalidades. Essas penalidades devem ser somadas a uma

medida de desempenho que, quanto menor o valor, maior a qualidade da resposta. Dito isso, um índice de desempenho utilizado em sistemas de controle é o *Integral of Time and Absolute Error* (ITAE), que calcula a integral da multiplicação entre o tempo e o valor absoluto do sinal de erro. A partir dessa informações, podemos escrever a função objetivo como

$$E = a_0 \int_0^T t|e(t)|dt + a_1 \int_0^T s_c(t)^2 dt + a_2((t_s - t_{s,max})^2 + (t_s - t_{s,min})^2) + a_3(m_s), \quad (2.19)$$

sendo $\int_0^T t|e(t)|dt$ o parâmetro de medição de qualidade e, o restante, as penalidades, dado que a_1 , a_2 e a_3 são as constantes que determinam o quanto a função objetivo será penalizada, e a_0 determinando a qualidade da resposta. Vale ressaltar que, o valor de sinal de controle ao quadrado, $s(t)^2$, é devido à condição do sinal de controle não poder saturar positivamente ou negativamente. Além do mais, esse sinal deve ser integrado para que seja retornado apenas um valor. O parâmetro $t_{s,min}$ é o valor mínimo de tempo de assentamento que se deseja no projeto e caso o usuário não tenha um valor mínimo requerido, basta assumi-lo como 0, bem como $t_{s,max}$, que é o valor máximo, quando houver.

Seguindo este raciocínio, as penalidades de cada termo serão valores específicos para cada tipo de controlador que deseja se encontrar. Tendo os índices de desempenho predefinidos, os valores das penalidades foram escolhidos a partir de testes sucessivos, feitos sobre cada algoritmo e cada controlador, considerando o que seria a resposta mais ideal para nós, a partir dos índices de desempenho, tempo de simulação, quantidade de iterações necessárias para conseguir um bom resultado, etc. A variação foi feita, aumentando e diminuindo os valores das penalidades e combinando essas mudanças entre si, a fim de atingir a melhor função custo possível. Sendo assim, as Tabelas 3 e 4 mostram esses valores para os controladores de velocidade e de corrente, nesta ordem.

Tabela 3 – Penalidades da função objetivo das meta-heurísticas para o controlador de velocidade.

Penalidade	Valor
a_0	3
a_1	0
a_2	0
a_3	1

Fonte: Do autor.

Não somente devemos calcular a função objetivo, a fim de se encontrar os melhores valores, devemos impor restrições de projeto que, se não cumpridas, não servem para o nosso objetivo final. Desse modo, as restrições de projeto serão os índices de desempenho mostrados anteriormente, nas Tabelas 1 e 2, para cada tipo de controlador.

Tendo isso em vista, somos capazes de aplicar todas as funções e condições apresentadas a um código de otimização. Sendo assim, a seguir apresentaremos os dois algoritmos

Tabela 4 – Penalidades da função objetivo das meta-heurísticas para o controlador de corrente.

Penalidade	Valor
a_0	3
a_1	0.1
a_2	0.1
a_3	0

Fonte: Do autor.

de meta-heurística e seu funcionamento, para que, futuramente, possamos utilizá-los junto ao que foi proposto e tentar encontrar os melhores valores na sintonia do controlador PI.

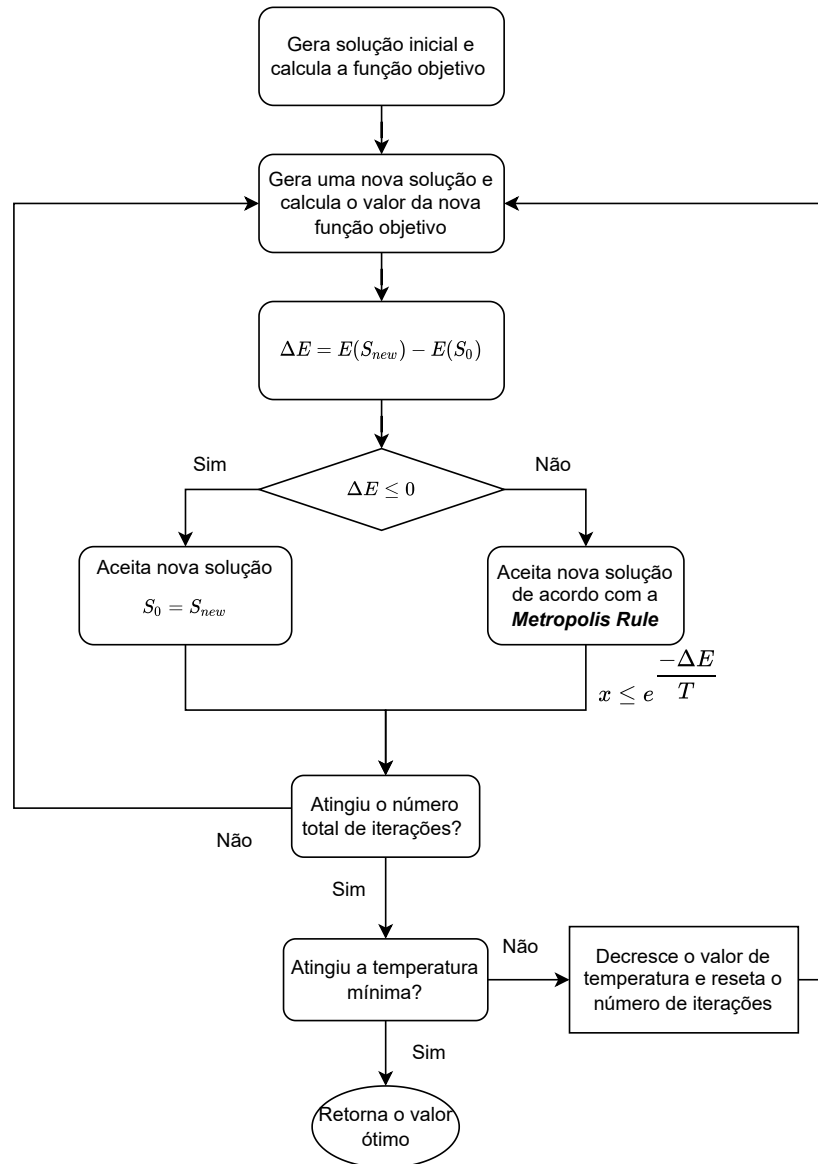
2.4.3 Simulated Annealing

A meta-heurística *Simulated Annealing* (SA) foi proposta originalmente por Kirkpatrick, Gelatt e Vecchi (1983). O método se baseia no processo físico de recozimento de um material como o metal, por exemplo. Trata-se de um processo de resfriamento em que se eleva a temperatura do metal a uma temperatura muito elevada a fim de atingir o estado líquido e depois resfria-o lentamente até que o metal se solidifique. Dessa forma, surge o nome recozimento simulado dado ao algoritmo, sendo que a cada decréscimo de temperatura são geradas soluções do problema de otimização, assemelhando-se aos átomos buscando aleatoriamente a condição de menor energia. Dito tudo isso, a Figura 19 mostra o fluxograma que descreve o funcionamento da meta-heurística.

Inicialmente, é gerada uma solução inicial S_0 e calculada a função objetivo E_0 , que serão, conseqüentemente, os melhores resultados da busca, S_{better} e E_{better} . Após isso, o algoritmo entra no *loop* de iterações, gera uma nova solução vizinha S_{new} e calcula sua função objetivo E_{new} . Para aceitar a nova solução como a melhor solução, faz-se o cálculo da diferença entre os valores de E_{new} e E_0 , e se o resultado, ΔE , for menor ou igual a 0, o novo valor se torna o novo parâmetro de comparação, ou seja, $S_0 = S_{new}$ e $E_0 = E_{new}$. Caso contrário, ele aceita a solução de acordo com a *Metropolis rule*. *Metropolis rule* é o mecanismo da meta-heurística SA que aceita soluções piores a fim de sair dos ótimos locais e, para fazer isso, é necessário gerar um valor aleatório x entre 0 e 1 e fazer a comparação

$$x \leq e^{\left(\frac{-\Delta E}{T}\right)},$$

sendo T a temperatura atual. Se essa condição for satisfeita, o algoritmo aceita a solução atual, S_{new} e E_{new} , como os novos parâmetros de comparação, S_0 e E_0 . Se em alguns desses dois casos aceitos, E_{new} for menor que E_{better} , S_{new} se tornará a melhor solução da busca. Após isso, o procedimento é seguido, sendo que, se não atingir o número de

Figura 19 – Fluxograma que descreve a meta-heurística *Simulated Annealing*.

Fonte: Do autor.

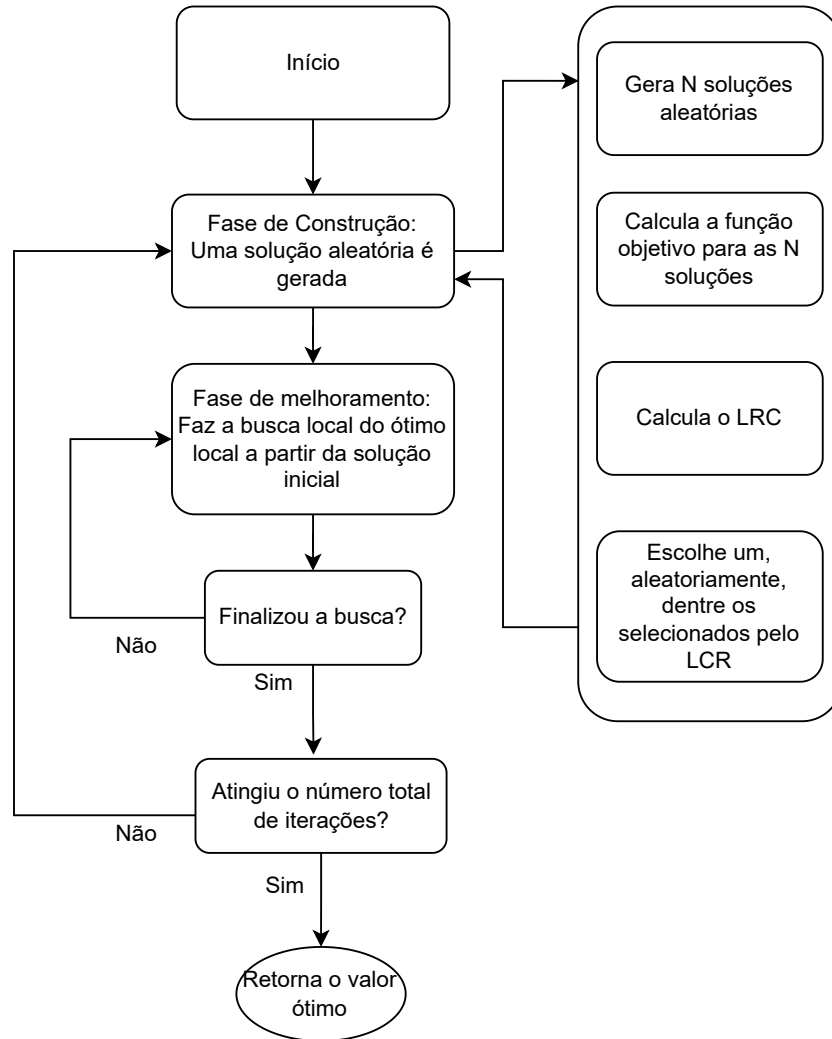
iterações total, ele repete o processo. Se atingir o número máximo de iterações, confere se a temperatura mínima foi alcançada, caso sim, o algoritmo se encerra, caso contrário, a temperatura é decrescida por um fator α e as iterações resetadas. Outro ponto importante é que, independentemente se a solução for melhor ou pior, o algoritmo só pode aceitar a nova solução se as condições impostas forem atendidas.

2.4.4 GRASP

A meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) foi proposta pela primeira vez por Feo e Resende (1995). Como o próprio nome diz, é uma busca adaptativa, gulosa e randômica. Isso se deve ao fato de, a cada iteração, o algoritmo

passar por uma fase de construção, em que várias soluções aleatórias são geradas e uma é retirada como solução inicial. A partir dessa solução, há uma busca do ótimo local, para que, ao final, o ótimo global possa ser encontrado. Desse modo, a Figura 20 mostra o fluxograma que descreve o funcionamento dessa meta-heurística.

Figura 20 – Fluxograma que descreve a meta-heurística *GRASP*.



Fonte: Do autor.

Dado um número de iterações n , a cada iteração, o algoritmo passa pela primeira fase, que é a fase de construção. Dentro da fase de construção, são geradas N soluções aleatórias, chamados de elementos candidatos, e calcula-se o valor da função objetivo para todas elas. Depois disso, calcula-se a Lista Restrita de Candidatos (LRC), que são os melhores elementos dentre os candidatos, dado por

$$LRC = f_{s,min} + \alpha(f_{s,max} - f_{s,min}). \quad (2.20)$$

Os valores $f_{s,min}$ e $f_{s,max}$ são o menor e o maior valor dentre as funções objetivo calculadas, respectivamente. O α é um valor entre 0 e 1, de modo que, quando $\alpha = 0$, a construção é

puramente gulosa, e quando $\alpha = 1$, a construção é puramente aleatória. Uma construção gulosa tem menos variação, mas possui maior qualidade, enquanto a construção aleatória é totalmente ao contrário. Assim sendo, o parâmetro α controla a qualidade dos elementos da LRC e pode ser escolhida de acordo com o critério de quem está configurando a heurística. Encontrado o valor de LRC em (2.20), a Lista Restrita de Candidatos será formada por todos os elementos cujos valores dos benefícios são menores ou iguais a LRC . Dentre esses, um é selecionado, aleatoriamente, e se tornará a solução inicial.

Passado a fase de construção, cada solução entrará em um *loop* de m iterações para a fase de melhoramento ou busca do ótimo local. Terminado as iterações, o algoritmo voltará para a fase de construção até que as iterações não se encerrem e, ao final, o melhor resultado dentre todas as buscas é retornado. O elemento de seleção aleatória, gulosa e adaptativa, permite que a heurística não fique presa em ótimos locais.

2.5 Conclusões Parciais

Nesse capítulo, vimos toda parte técnica descrita, a partir da teoria apresentada no capítulo introdutório:

- Inicialmente, apresentamos o circuito de um motor de corrente contínua de excitação independente e como o motor é acoplado à carga. Discutimos sobre como é o funcionamento desse sistema, as equações diferenciais que o compõe e seus modelos dinâmicos.
- Depois, mostramos um conversor CC-CC, ponte H *chopper*, o seu funcionamento, a partir de um sinal do PWM, e como ele vai ser representado no sistema de controle.
- Dado as dinâmicas dos componentes previamente apresentados, falamos sobre como o sistema de controle em cascata completo é composto. O sistema contava com um entrelaçamento de malhas e duas formas de resolvê-la foram apresentadas. A primeira, utilizando o desentrelaçamento das malhas, por cálculo matemático. O segundo, a partir da utilização de um controlador antecipatório, cujo a forma de calculá-lo também foi explicitada. Das duas maneiras, foram apresentados as plantas de velocidade e corrente, sendo somente a de corrente diferente entre os dois métodos.
- Para o projeto de controladores, discutimos quais seriam os critérios de desempenho que o sistema precisava atingir para que o controlador fosse considerado bom ou aceitável. Dentre as técnicas que serão aplicadas, temos a utilização do *sisotool*, que é uma ferramenta do *software* MATLAB, e a obtenção por meio das meta-heurísticas.

- Discutimos e apresentamos, qual seria a função objetivo, como ela foi escolhida, quais são as penalidades, o porquê da escolha das penalidades e como funcionam as meta-heurísticas SA e GRASP, por meio de diagramas de fluxo.
- Por fim, apresentamos a utilização de uma malha *anti-windup*, quais são suas vantagens e porque é importante a sua utilização.

3 Aplicação das Técnicas Apresentadas no Controle do Motor CC

Foi apresentada toda fundamentação teórica por trás deste trabalho e compreendemos melhor todas as ferramentas que seriam necessárias para o desenvolvimento e aplicação real. De todo modo, como mencionado, este trabalho será aplicado totalmente de forma computacional utilizando o *software* MATLAB. Assim sendo, apresentaremos neste capítulo os resultados obtidos nas simulações e discorreremos um pouco sobre cada resultado.

Para iniciar, é necessário primeiramente escolher qual o modelo de motor CC que será trabalhado e quais são os seus parâmetros, bem como os parâmetros da carga e a relação de engrenagens. No nosso caso, será utilizado uma máquina de médio-grande porte, cujas características podem ser vistas juntamente com as características da carga e das engrenagens na Tabela 5.

Tabela 5 – Parâmetros da máquina CC, carga e engrenagem

Tipo	Descrição	Parâmetro	Valor	Unidade
Máquina CC	Potência nominal	P_{nom}	200	HP
	Tensão de armadura nominal	$V_{a,nom}$	500	V
	Corrente de armadura nominal	$I_{a,nom}$	298,4	A
	Torque nominal	$T_{em,nom}$	814,14	$N \cdot m$
	Velocidade Nominal	n_{nom}	1750	rpm
	Tensão de campo nominal	$V_{f,nom}$	310	V
	Corrente de campo nominal	$I_{f,nom}$	1,0	A
	Resistência de armadura	R_a	0,076	Ω
	Indutância de Armadura	L_a	0,00157	H
	Resistência de campo	R_f	310	Ω
	Indutância de campo	L_f	232,25	H
	Indutância Mútua	L_{af}	3,320	H
	Inércia da Máquina	J_m	4	$K_g \cdot m^2$
	Coefficiente de atrito viscoso da máquina	B_m	0,16	$N \cdot m \cdot s$
	Torque de fricção de Coulomb	T_f	0	$N \cdot m$
Engrenagens	Raio da engrenagem motor	r_m	100	mm
	Raio da engrenagem carga	r_L	500	mm
	Coefficiente de rigidez (lado máquina CC)	K_g	115	$N \cdot m/rad$
	Coefficiente de atrito (lado máquina CC)	B_g	0.001	$N \cdot m \cdot s$
Carga	Inércia do motor	J_L	150	$k_g \cdot m^2$
	Coefficiente de atrito viscoso do motor	B_L	4,0	$N \cdot m \cdot s$

Fonte: Do autor.

A partir desses valores, podemos começar a desenvolver o projeto final, inicialmente encontrando o modelo da planta e, posteriormente, projetando os controladores.

3.1 Modelo numérico do motor CC e controlador projetado

3.1.1 Dinâmica das plantas

Tendo em vista os parâmetros apresentados na Tabela 5, podemos encontrar o modelo numérico da carga do sistema. Desse modo, como foi visto, o valor do ganho K_{PWM} depende do valor da tensão no barramento. Nesse caso utilizaremos, como tensão de entrada, o valor da tensão nominal de armadura, pois o controlador ficará encarregado de mensurar a quantidade de tensão necessária para o processo através do sinal enviado ao PWM, portanto, não implica que a tensão será mantida sempre no nível nominal. Sendo assim, temos inicialmente que

$$v_d = 500V,$$

logo, a partir de (2.8), sendo v_c o sinal de controle que determinará o valor final de tensão aplicado à máquina, v_a , podemos dizer que

$$K_{PWM} = 500.$$

Do mesmo modo, como foi apresentado, precisamos encontrar os valores de J_{eq} , em (2.4) e B_{eq} , em (2.3), que dependem da relação entre as engrenagens do sistema. Entretanto, foi falado sobre a relação utilizando a quantidade de dentes das engrenagens e a Tabela 5 nos fornece o raio de cada engrenagem. Ao final, isso não terá relevância, pois a mesma relação continua se estabelecendo da forma

$$\frac{N_1}{N_2} = \frac{r_1}{r_2},$$

sendo r_1 e r_2 o raio das engrenagens do lado do motor e da carga, respectivamente, apresentadas na Tabela 5 como r_m e r_L . Dessa maneira, tem-se que

$$\frac{N_1}{N_2} = \frac{r_m}{r_L} = \frac{100}{500} = 0,2,$$

logo,

$$B_{eq} = 0,32 \text{ e } J_{eq} = 10.$$

Além desses resultados, é necessário fazer o cálculo das constantes k_e e k_t . O valor de k_e pode ser escrito como a indutância mútua da máquina multiplicada pela corrente de campo. Como o motor terá seu campo alimentado com tensão nominal, a corrente dele também será nominal, logo,

$$k_e = L_{af} \times I_f = 3,32 \times 1,0 = 3,32.$$

O valor de k_t , por sua vez, pode ser definido como sendo o mesmo valor de k_e , portanto,

$$k_t = k_e = 3,32.$$

Em seguida, partindo de (2.15), tendo todos esses resultados e substituindo os valores dados na Tabela 5, temos que

$$G_{1_{PWM}}(s) = \frac{500(10s + 0.32)}{0.0157s^2 + 0.7605s + 11.05}, \quad (3.1)$$

sendo esse resultado o modelo numérico da dinâmica da corrente do motor.

No caso em que o controlador antecipatório é aplicado, vimos que o modelo da planta pode ser simplificado, logo, partindo de (2.16),

$$G_{1_{ff,PWM}}(s) = \frac{500}{0.00157s + 0.076}.$$

Da mesma forma, utilizando os resultados encontrados e partindo de (2.10), podemos escrever que

$$G_2(s) = \frac{3.32}{10s + 0.32}, \quad (3.2)$$

o qual representa o modelo numérico da dinâmica da velocidade do motor.

3.1.2 Controladores projetados

Uma vez tendo as plantas definidas, dadas em (3.1) e (3.2), é possível encontrar os controladores para o sistema. Como mencionado, os controladores serão encontrados computacionalmente através do *software* MATLAB, utilizando a função *sisotool*. Dentro desta função, podemos ver o comportamento do sistema perante a introdução do controlador, visto que os parâmetros de projeto apresentados devem ser considerados. Então, começando pelo modelo dinâmico da corrente, (3.1), encontramos o controlador

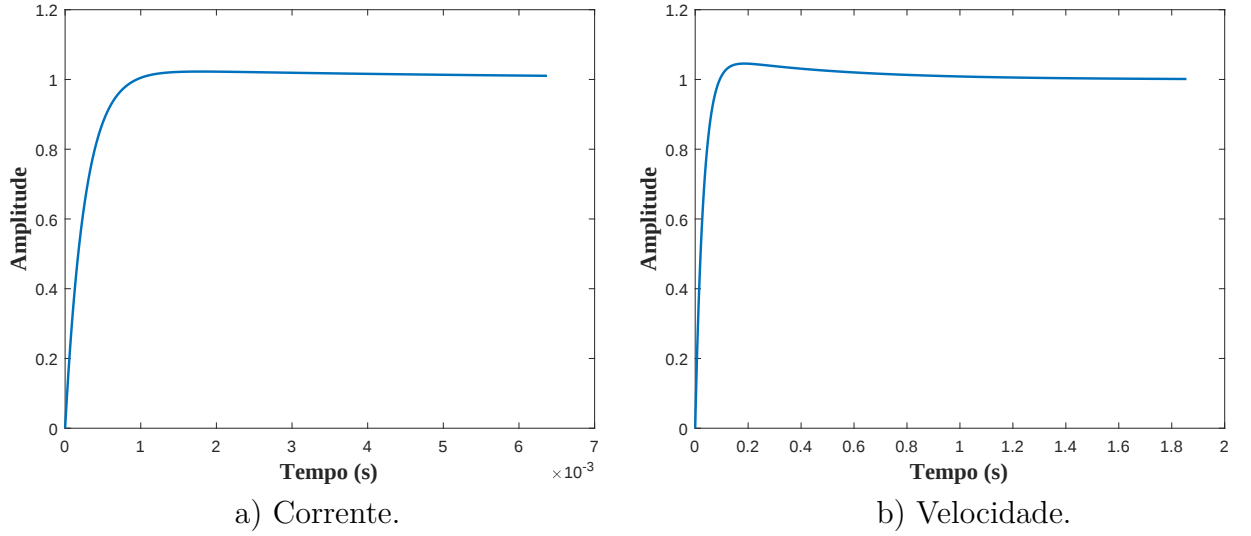
$$G_{c_1}(s) = 0.01238 + \frac{2}{s}. \quad (3.3)$$

A resposta ao degrau do sistema, composto pelo controlador mais planta, em malha fechada, pode ser observado na Figura 21-a. O tempo de assentamento nesse caso é 0.0028s e o máximo sobressinal é 2.27%, ou seja, está dentro dos parâmetros preestabelecidos na Tabela 1 e, portanto, é um resultado adequado às especificações predefinidas.

Projetando agora o controlador a partir do modelo dinâmico da velocidade, (3.2), temos o controlador

$$G_{c_1}(s) = 100 + \frac{200}{s}. \quad (3.4)$$

A resposta ao degrau deste sistema, composto pelo controlador mais planta, em malha fechada, pode ser observado na Figura 21-b. O tempo de assentamento nesse caso

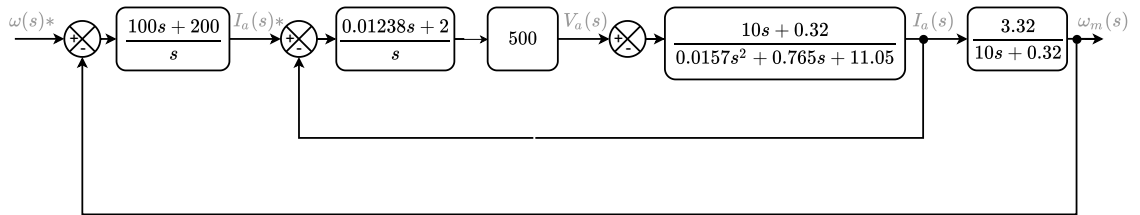
Figura 21 – Resposta ao degrau do sistema de controle com controlador via *sisotool*.

Fonte: Do autor.

é $0.604s$ e o máximo sobressinal é 4.55% , ou seja, está dentro dos parâmetros preestabelecidos na Tabela 2 e, portanto, é um resultado adequado às especificações predefinidas.

Encontrado os controladores (3.3) e (3.4) por meio do MATLAB e as plantas por meio dos cálculos, se observarmos a Figura 14, temos que o sistema de controle, inicialmente, está completo e pode ser representado assim como ilustrado na Figura 22. Esta configuração permanecerá durante a apresentação dos resultados. As únicas mudanças serão a adição do *anti-windup*, para aplicação real, e a troca dos controladores, para quando os calcularmos por meio das meta-heurísticas. Além disso, para alguns casos, utilizaremos a malha de corrente com o controlador antecipatório.

Figura 22 – Sistema de controle do projeto completo com valores numéricos.



Fonte: Do autor.

Feito isso, partiremos para a aplicação completa da forma como é mostrado na Figura 22, e para tal, simularemos o sistema utilizando mais uma ferramenta simulacional, fornecida por meio do MATLAB. Os detalhes e resultados serão apresentados a seguir.

3.2 Resultados do sistema de controle da máquina CC

Uma vez que todo trabalho é realizado computacionalmente, precisaremos de um simulador de circuitos, mais uma vez, confiável e que seja possível fazer análises minuciosas. Para isso, utilizaremos a biblioteca *SimPowerSystems*, contida no *Simulink*, que é um ambiente desenvolvido para que seja possível simular sistemas de controle, eletrônica de potência, sistemas de potência etc, dentro do MATLAB. A partir de agora, então, nossas análises serão retiradas dos resultados obtidos dessa ferramenta, baseando-nos no sistema de controle da Figura 22.

Algumas informações importantes a cerca do ambiente em que o sistema foi simulado, são os parâmetros da simulação, listados na Tabela 6

Tabela 6 – Parâmetros da simulação utilizados durante a utilização do *Simulink*.

Parâmetro	Valor
Método de Integração	Ode15s
Passo de simulação máximo	$1\mu s$
Biblioteca	SimPowerSystems
Modo de simulação (powergui)	$50 \times 10\mu s$

Fonte: Do autor.

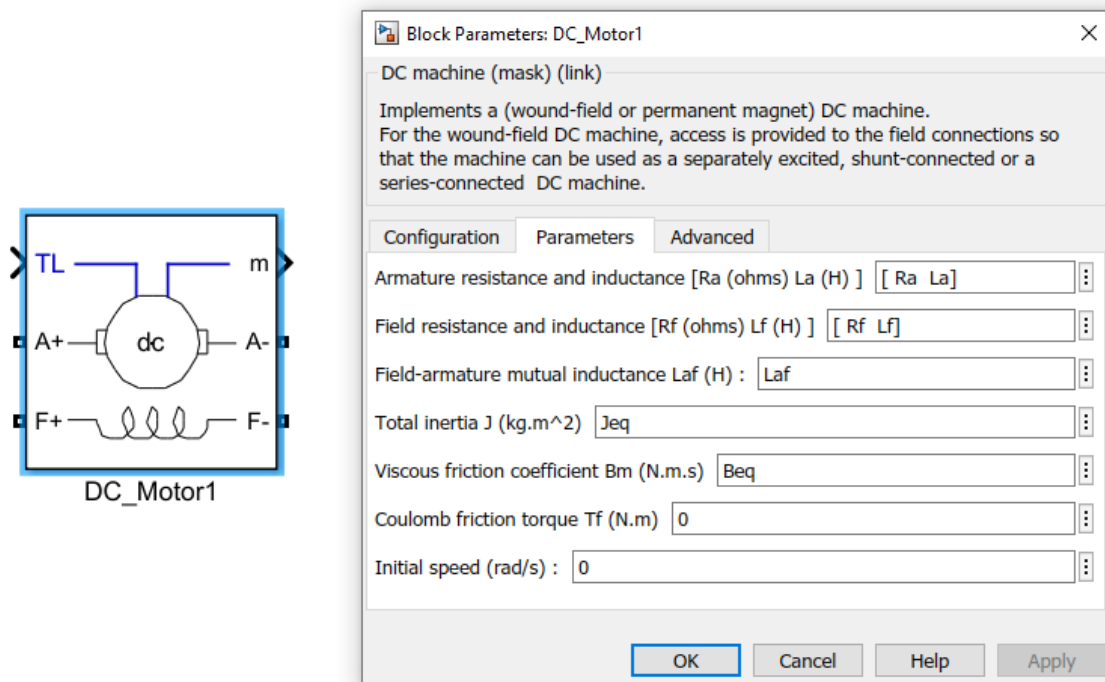
3.2.1 Resultados da máquina CC funcionando com controlador sem *anti-windup*

Para montar o sistema de controle no *Simulink*, utilizaremos os diagramas de bloco para inserir os controladores e utilizaremos uma máquina *built-in* do próprio *software*, isto é, um modelo de máquina já construído. Para usá-la, temos somente que inserir alguns valores de entrada, que podemos retirar da Tabela 5 e dos cálculos feitos, conectar um torque de entrada em seu terminal e conectar seus terminais de armadura e campo conforme precisamos. O campo será de excitação independente, portanto será ligado à uma fonte com tensão nominal $V_f = 310V$. A armadura faz parte do modelo e do circuito que desejamos controlar, dessa forma, será ligado ao sistema de controle através dos conversores. A Figura 23 mostra o bloco do motor e seus parâmetros inseridos.

A ponte H que será ligada ao motor foi montada manualmente no programa, utilizando os IGBTs fornecidos pelo próprio *software*. O circuito montado pode ser observado na Figura 24, sendo que, à direita é mostrado o conversor em forma de bloco para facilidade de implementação, e, à esquerda, o circuito presente no bloco. O conversor recebe em sua entrada g o sinal proveniente do PWM, a alimentação CC da fonte, e gera uma saída CC pulsada.

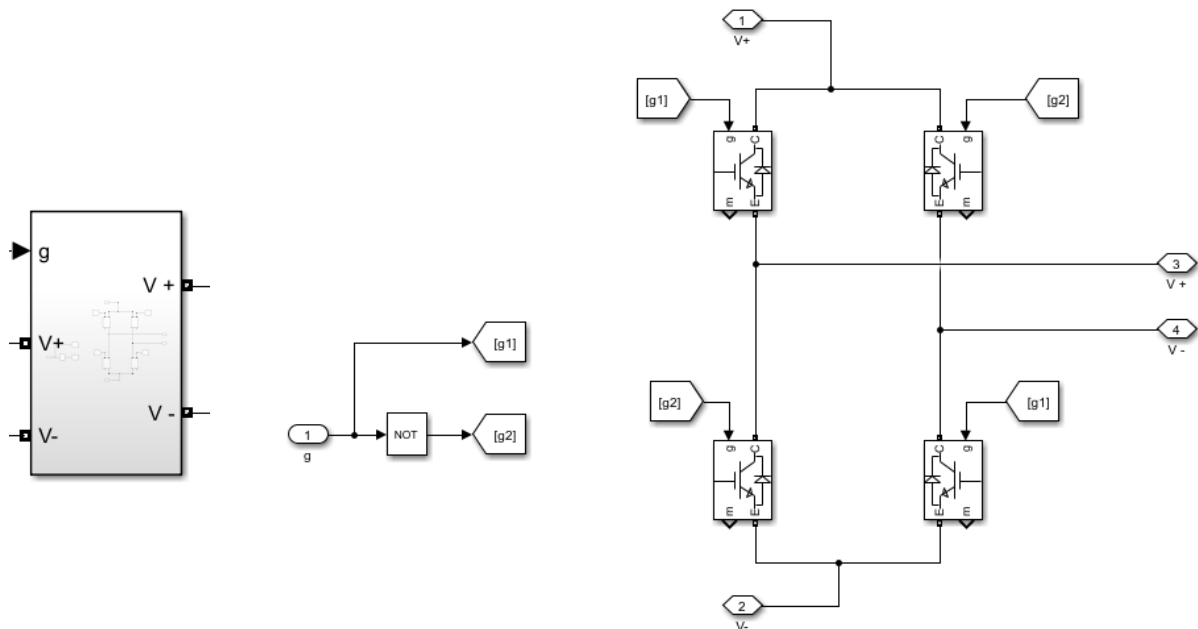
Para controlar o acionamento do conversor, um PWM também foi inserido, sendo este mostrado como implementado na Figura 25. Mais uma vez, à esquerda, está o bloco do

Figura 23 – Bloco da máquina CC no simulink.



Fonte: Do autor.

Figura 24 – Conversor CC-CC ponte H no simulink.

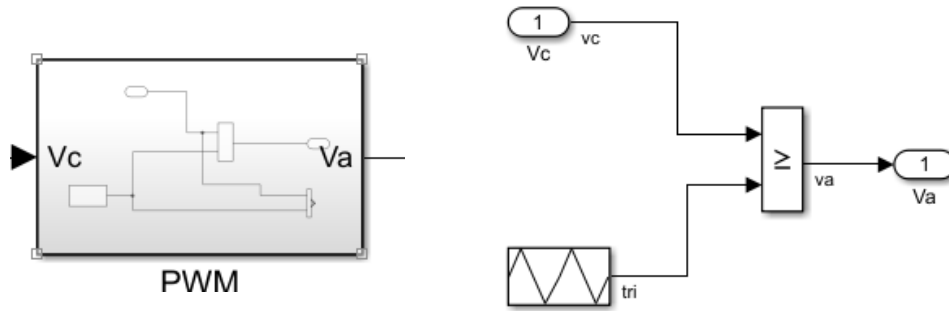


Fonte: Do autor.

dispositivo e, à direita, seu circuito interno, de modo que o sinal de controle é comparado com um sinal triangular, cujo a frequência de chaveamento é de 5 KHz , e o sinal resultante enviado para a saída.

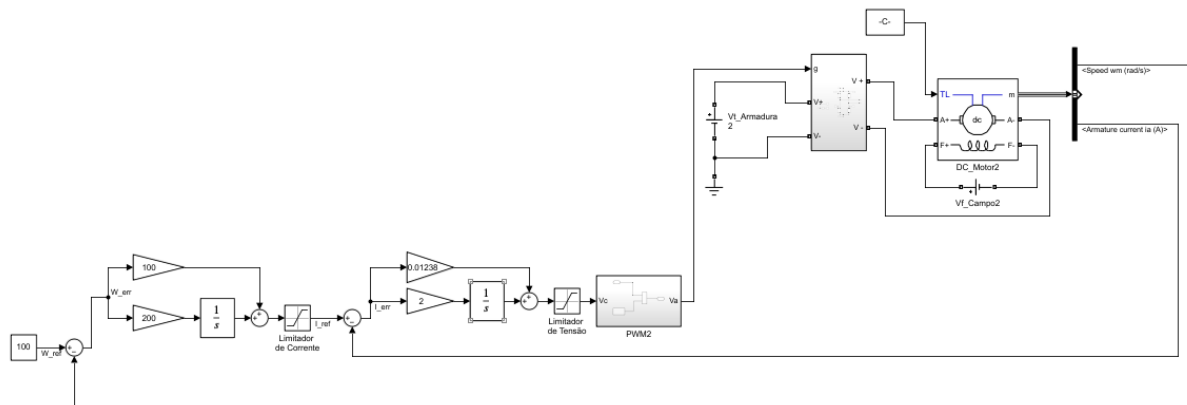
A parte de controle foi feita através de blocos de ganho e função de transferência,

Figura 25 – Circuito PWM no simulink.



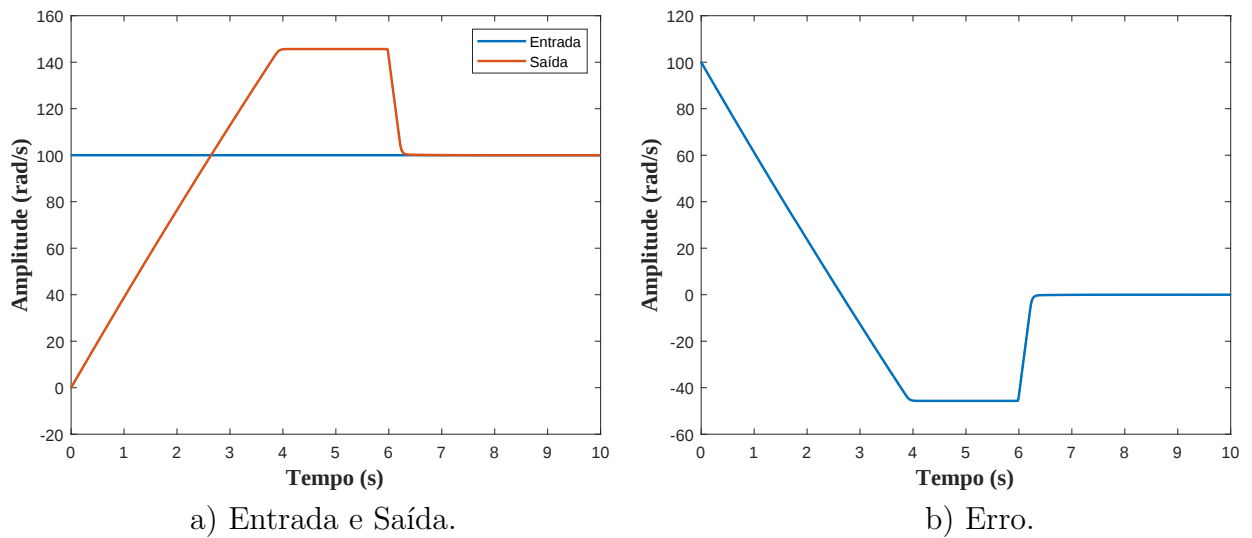
Fonte: Do autor.

inserindo assim os dois controladores PIs encontrados em (3.3) e (3.4). Conectando todas essas partes e fechando a malha em cascata, retirando os sinais de saída de corrente e velocidade do motor através de um bloco auxiliar, temos o circuito completo montado no *simulink* exposto na Figura 26. Vale ressaltar também que limitadores de corrente e tensão foram inseridos, pelo fato de, fisicamente, ser impossível chegar a alguns valores que o *software* computa, que são os limites dos atuadores. Dito isso, foram configurados com limites máximo e mínimo de $\pm 3.35\%$ e $\pm 15\%$, para a corrente a tensão, nesta ordem. Em outras palavras, os sinais de controle serão de -310 a +310 para o de corrente e -1.15 a +1.15 para o de tensão, que entra no PWM.

Figura 26 – Sistema de controle completo montado no *simulink*.

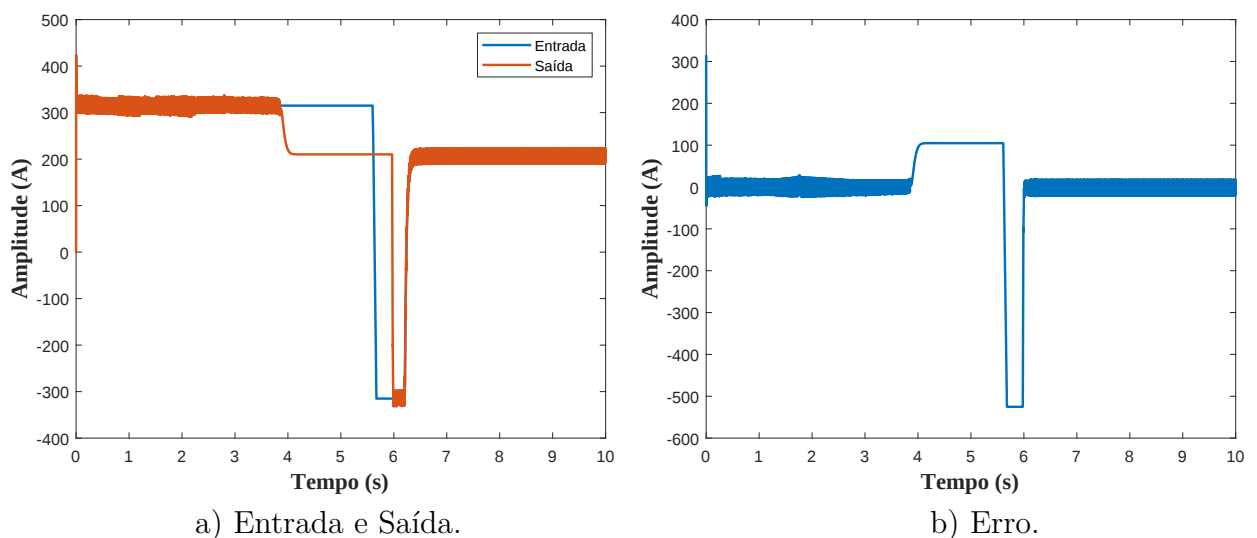
Fonte: Do autor.

Obtido o circuito, podemos simulá-lo e ver o resultado. Uma observação é que o *anti-windup* não foi inserido, logo, os primeiros resultados servirão de comparação quanto a sua importância. Dito tudo isso, para a primeira simulação, vamos atribuir à referência do sistema uma velocidade de 100 rad/s e um torque de entrada da máquina T_{shaft} igual a 80% do torque nominal, ou seja, $T_{shaft} = 651.312 \text{ N.m}$. Fazendo isso e simulando o sistema, obtivemos, inicialmente, o comportamento da velocidade, mostrado na Figura 27.

Figura 27 – Velocidade da máquina CC com sistema funcionando sem *anti-windup*.

Fonte: Do autor.

Observando a Figura 27-a, onde é mostrado o comportamento da saída em relação à entrada, podemos observar que a velocidade aumenta rapidamente até seu valor máximo, se mantém e depois cai, em regime permanente, para o valor final, que é o valor da referência, 100rad/s . Do mesmo modo, a Figura 28 ilustra o comportamento da corrente para esse caso.

Figura 28 – Corrente da máquina CC com sistema funcionando sem *anti-windup*.

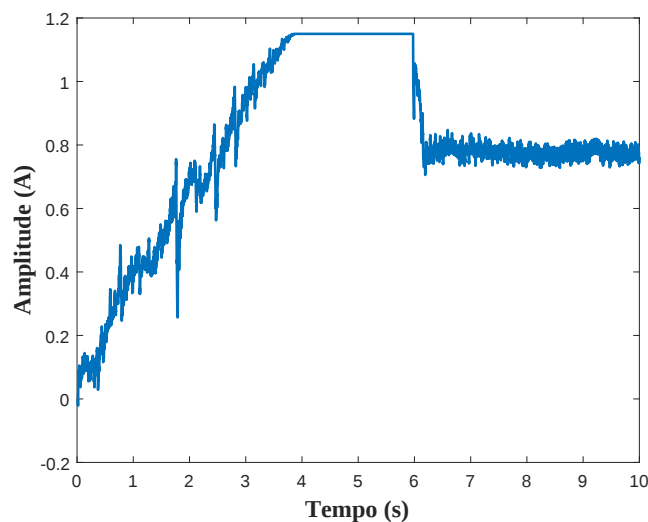
Fonte: Do autor.

A partir da Figura 28-a, que representa a referência e a saída de corrente, vemos que a corrente tem uma resposta muito rápida, em que há um sobressinal relevante, mas logo se mantém no valor de referência. Depois de um certo tempo, quando a velocidade atinge seu

valor máximo, a corrente de saída decai para o valor final, tem um pico negativo máximo, proveniente da resposta do controlador de velocidade, que é a referência de corrente, para fazer com que a velocidade diminua até a desejada, e logo volta para seu valor em regime permanente.

Esse tipo de padrão em que os valores variam bruscamente se devem ao fato de não ter um sistema *anti-windup* instalado e, conseqüentemente, faz com que o controlador carregue demais. Logo quando percebe que precisa diminuir a ação de controle, também envia de forma brusca, fazendo com que a máquina tenha o tipo de comportamento apresentado acima. Para comprovar isso, a Figura 29 ilustra o sinal de controle proveniente do controlador de corrente. Como esse sinal é muito oscilante entre seus valores máximos de 1.15 e -1.15, a imagem foi retirada a partir da média desse sinal. Nela, vemos um formato muito parecido com a velocidade, apresentada na Figura 27-a.

Figura 29 – Sinal de controle proveniente do controlador de corrente máquina CC com sistema funcionando sem *anti-windup*.



Fonte: Do autor.

Apresentado tudo isso, vimos o quão importante é atuar para impedir que ocorra a saturação da parte integradora do controlador. Por isso, a partir da próxima seção, apresentaremos os resultados do sistema com a aplicação da estratégia *anti-windup*.

3.2.2 Resultados da máquina CC funcionando com controlador com *anti-windup*

Assim como foi comentado, de (2.17), podemos encontrar o valor da constante de tempo que aplicada ao sistema *anti-windup*, fará com que o controlador descarregue o seu integrador e evite a saturação do sinal de controle. Uma vez que temos dois controladores,

podemos implementar um *anti-windup* em cada. Sendo assim,

$$T_{w_1} = \frac{1}{\sqrt{2}} = 0.7071,$$

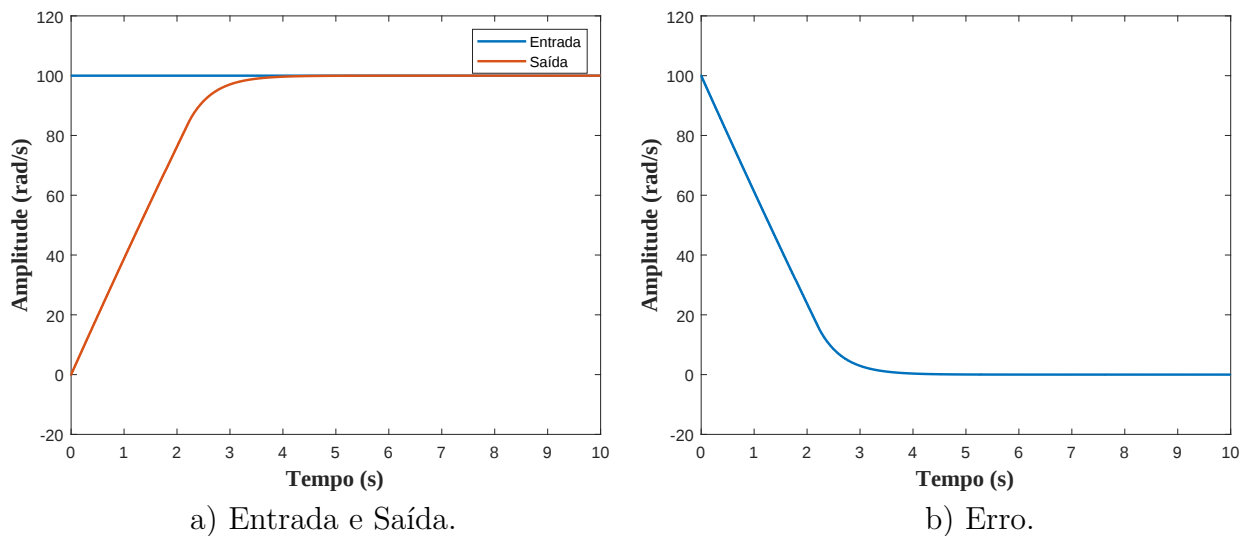
e

$$T_{w_2} = \frac{1}{\sqrt{200}} = 0.07071,$$

para T_{w_1} o valor para $G_{c_1}(s)$ e T_{w_2} para $G_{c_2}(s)$.

Configurando estes valores no sistema da simulação, podemos obter os mesmos resultados anteriores para comparação. O primeiro, novamente, é a velocidade, mostrado na Figura 30. A saída em relação à entrada, Figura 30-a, mostra um comportamento totalmente diferente agora, uma vez que a velocidade aumenta suavemente até seu valor em regime permanente.

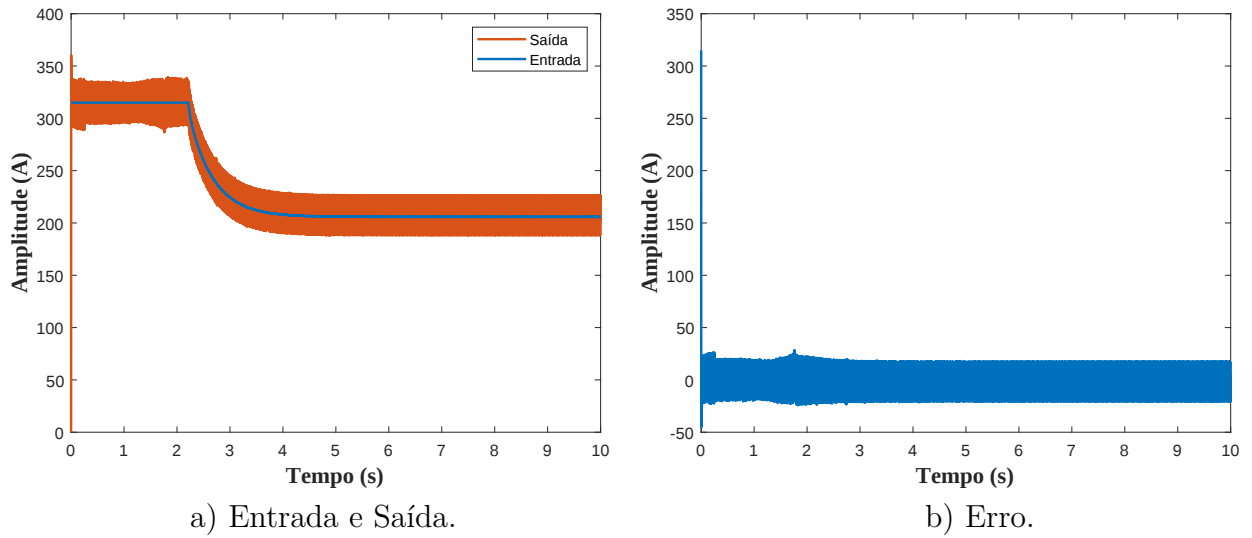
Figura 30 – Velocidade da máquina CC com sistema funcionando com *anti-windup*.



Fonte: Do autor.

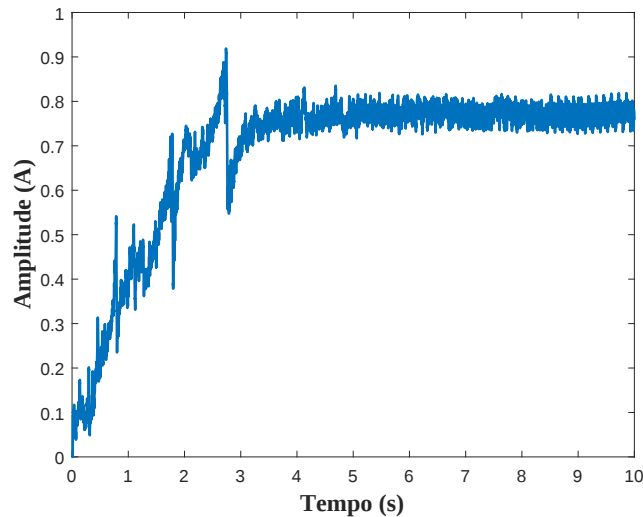
Do mesmo modo, a Figura 31 demonstra a corrente do sistema. Na comparação entrada e saída de corrente, Figura 31-a, vemos que a saída, por mais que tenha uma grande oscilação, segue muito bem a referência. Ela satura durante um certo tempo e, quando a velocidade começa a assentar, diminui seu valor, suavemente, até o regime permanente.

A partir dessas análises, podemos ver o quão importante se faz a utilização do *anti-windup* no sistema, que não permite que a parte integradora do controlador não se carregue totalmente e envia o sinal de controle de forma que o sistema consiga responder suavemente. Como a entrada da corrente é o esforço de controle do controlador de velocidade, a Figura 32 mostra o esforço de controle do controlador de corrente. Da mesma forma, como há muita oscilação, a ilustração mostra a média do sinal, para uma melhor

Figura 31 – Corrente da máquina CC com sistema funcionando com *anti-windup*.

Fonte: Do autor.

análise de comportamento. A forma da curva é parecida como a da velocidade de saída, Figura 30-a e se comporta de forma mais branda que o sinal de controle sem *anti-windup*.

Figura 32 – Sinal de controle proveniente do controlador de corrente máquina CC com sistema funcionando com *anti-windup*.

Fonte: Do autor.

Tendo em vista os resultados obtidos e apresentados, vimos que o controlador projetado atendeu às especificações, e que a estratégia *anti-windup* é fundamental em sistemas reais utilizando controladores PI ou PID. Desse modo, os resultados finais da simulação do sistema foram satisfatórios e atenderam ao esperado. Todavia, a nossa tentativa é conseguir melhorar ainda mais o controlador projetado. Em vista disso, apresentaremos

adiante a implementação das meta-heurísticas, a fim de se alcançar esse objetivo. Os resultados obtidos serão comparados com os anteriores e discutidos sobre a qualidade de cada um.

3.3 Aplicação do algoritmo *Simulated Annealing*

Toda teoria envolvendo o procedimento do SA foi percorrida no capítulo anterior, não só isso como os outros tópicos de importância para a otimização do nosso problema, como a função objetivo, restrições etc. Portanto, entraremos diretamente na implementação, aplicando os parâmetros escolhidos, os quais são mostrados na Tabela 7, e obtendo os resultados.

Tabela 7 – Parâmetros utilizados para a implementação do algoritmo SA.

Parâmetro	Descrição	Valor
N	Número de iterações por temperatura	150
T_o	Temperatura inicial	25
T_{end}	Temperatura de parada	10^{-8}
α	Fator de resfriamento	0.95

Fonte: Do autor.

A cada iteração, a temperatura decrescerá 5% até atingir a temperatura de parada e o código é interrompido. A variável de projeto também deve ser inicializada, e para esse caso, assumiremos ambos os valores de K_I e K_P iguais a 100, ou seja,

$$S = [100 \ 100],$$

valor escolhido com base em testes feitos sobre qual seria a melhor condição inicial.

Esse valor será variado a cada iteração e, ao final, espera-se encontrar o melhor resultado. Tendo em vista os valores apresentados na Tabela 7, unido à função objetivo (2.19), somos capazes de implementar o algoritmo computacionalmente, utilizando o *software* MATLAB.

3.3.1 Resultados encontrados para o controlador de velocidade

Para o cômputo do controlador de velocidade, utilizaremos as Tabelas 2 e 3. A partir desses valores, nossa função objetivo é pode ser dada por

$$J = 3 \int_0^T t|e(t)|dt + (m_s).$$

Neste sentido, utilizaremos agora (3.2) para o cálculo da função objetivo. Com o intuito de encontrar o melhor controlador possível, fizemos a discretização do sistema de controle composto por controlador de velocidade e planta, para que o *anti-windup*

pudesse ser inserido nas operações matemáticas. Ao fazer isso, encontramos os resultados mostrados na Tabela 8.

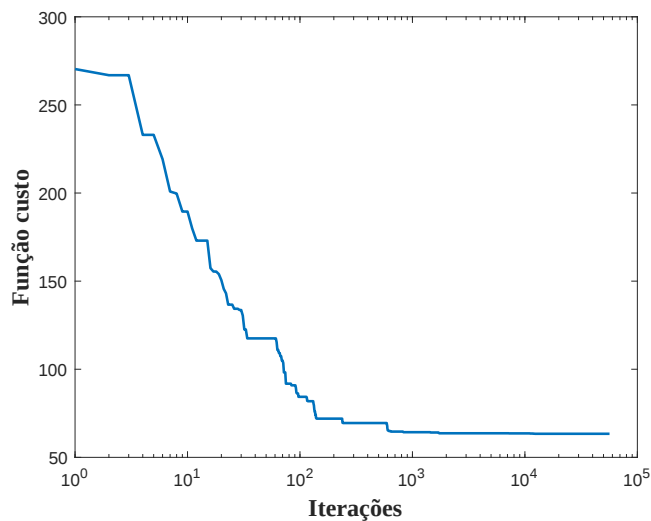
Tabela 8 – Resultado da otimização na sintonia do controlador de velocidade utilizando o algoritmo *Simulated Annealing*.

Representação	Descrição	Resultado
-	Tempo de processamento	107 s
-	Número de Iterações	56700
J_{min}	Melhor Custo	63.3757
$G_{c1}(s)$	Melhor Controlador	$260.8836 + \frac{20520.6195}{s}$

Fonte: Do autor.

O gráfico de otimização para esse caso está ilustrado na Figura 33. A partir dele, vemos que o algoritmo realmente otimizou e o gráfico diminuiu até encontrar seu ótimo global. A escala está em logaritmo para que tivéssemos uma melhor visualização.

Figura 33 – Gráfico da função custo no processo de otimização da planta de velocidade utilizando o algoritmo *Simulated Annealing*.



Fonte: Do autor.

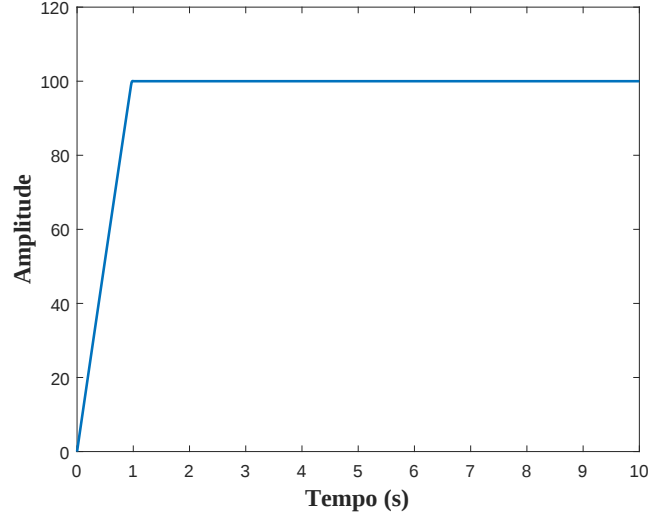
A seguir, aplicamos uma entrada degrau de 100 *rad/s* no sistema $G_{c2}(s)$ - $G_2(s)$, com anti-windup, cujo a constante de tempo é

$$T_{w_{wm,sa}} = \frac{1}{\sqrt{20520.6195}} = 143.2502.$$

A saída, neste caso, dentro de um tempo de 10 s, é mostrada na Figura 34.

A resposta da Figura 34 tem um tempo de assentamento de 0.9513 s, um máximo sobressinal de 0.0439% e erro em regime permanente igual a 0. Todas essas características atendem ao nosso critério de desempenho, nesse caso.

Figura 34 – Resposta ao degrau da malha de velocidade a partir do controlador $K_P = 260.8836$ e $K_I = 20520.6195$ encontrado utilizando o algoritmo *Simulated Annealing*.



Fonte: Do autor.

3.3.2 Resultados encontrados para o controlador de corrente

Agora, trataremos a resolução do controlador de corrente, em que as Tabelas 2 e 3 serão a base para o cômputo do controlador. Nesse, sentido, a função objetivo é dada por

$$J = 3 \int_0^T t|e(t)|dt + 0.1 \int_0^T s_c(t)^2 dt + 0.1((t_s - 0.003)^2 + (t_s - 0.001)^2).$$

Como os valores da função objetivo no cálculo do controlador de corrente eram muito baixos, havia um problema de aproximação no *software*, que fazia com que a comparação dos melhores resultados falhasse. Dessa forma, multiplicamos o resultado da função objetivo por 1000, com o intuito de não deixar o problema acontecer.

Para computar controlador de corrente, fizemos testes utilizando a dinâmica (2.15), entretanto, os resultados encontrados, depois de várias diferentes tentativas, não foram satisfatórios, além da complexidade maior dessa função na sua discretização. Desse modo, para executar os algoritmos de forma mais simples e tomar uma abordagem diferente, utilizaremos a dinâmica da planta, descrita em (2.16), ou seja, vamos considerar o sistema funcionando com o controlador antecipatório. Para calcular o controlador, tomando (2.12), temos que

$$G_{ff}(s) = \frac{k_e}{G_{c1} K_{PWM}},$$

logo,

$$G_{ff}(s) = \frac{3.32}{500G_{c1}},$$

sendo G_{c1} o controlador que queremos encontrar. Assim sendo, discretizamos o sistema de controle composto pelo controlador de corrente e planta, para que a malha *anti-windup* seja inserida nas operações. Sendo assim, a Tabela 9 mostra os resultados obtidos.

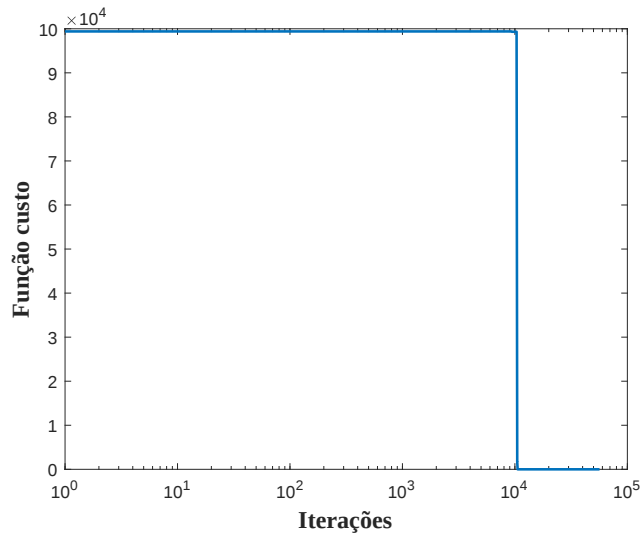
Tabela 9 – Resultado da otimização na sintonia do controlador de corrente utilizando o algoritmo *Simulated Annealing*.

Representação	Descrição	Resultado
-	Tempo de processamento	111 s
-	Número de Iterações	56700
J_{min}	Melhor Custo	0.0736
$G_{c1}(s)$	Melhor Controlador	$0.0132 + \frac{0.6373}{s}$

Fonte: Do autor.

O gráfico de otimização está ilustrado na Figura 35. A partir dele, vemos que o algoritmo realmente otimizou até o ótimo global. A escala está em logaritmo para que tivéssemos uma melhor visualização.

Figura 35 – Gráfico da função custo no processo de otimização do controlador de corrente utilizando o algoritmo *Simulated Annealing*.



Fonte: Do autor.

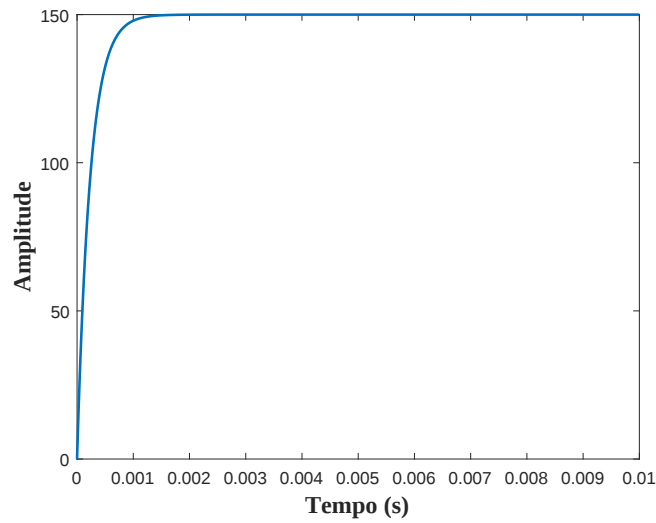
A partir desses resultados, aplicamos um degrau de 150 A no sistema $G_{c1}(s)$ - $G_{1PWM}(s)$, com o anti-windup, cujo valor da constante de tempo é

$$T_{wi,sa} = \frac{1}{\sqrt{0.6373}} = 0.7983.$$

A saída, neste caso, dentro de um tempo de 0.01 s, é mostrada na Figura 36.

A resposta da Figura 36 tem um tempo de assentamento de 0.0009 s, um máximo sobressinal de 0 e erro em regime permanente igual a 0.0001. O erro não foi nulo, mas como

Figura 36 – Resposta ao degrau da malha de corrente a partir do controlador $K_P = 0.0132$ e $K_I = 0.6373$ encontrado utilizando o algoritmo *Simulated Annealing*.



Fonte: Do autor.

foi um número muito pequeno, é de se esperar uma diferença, seja por causa dos cálculos matemáticos do *software*, ou porque o tempo não foi suficiente para que ele chegasse à 0.

3.4 Aplicação do algoritmo GRASP

Toda teoria envolvendo o procedimento do GRASP foi percorrida no capítulo anterior, não só isso como os outros tópicos de importância para a otimização do nosso problema, como a função objetivo, restrições etc. Portanto, entraremos diretamente na implementação, aplicando os parâmetros escolhidos, os quais são mostrados na Tabela 10, e obtendo os resultados.

Tabela 10 – Parâmetros utilizados para a implementação do algoritmo GRASP.

Parâmetro	Descrição	Valor
N_c	Número de iterações para construção	250
N_b	Número de iterações para busca	400
N	Soluções aleatórias na construção	10
a	Valor mínimo do parâmetro na construção	20
b	Valor máximo do parâmetro na construção	500
α	Fator qualidade	0.9

Fonte: Do autor.

O valor do fator de qualidade, α , mostra que optamos por algo menos seletivo e de mais qualidade. Os valor do vetor que representa a variável de projeto estará entre 20 e 500, e o programa iterará 100000 vezes. Tendo em vista os valores apresentados na

Tabela 10, unido à função objetivo (2.19), podemos, novamente, implementar o algoritmo computacionalmente.

3.4.1 Resultados encontrados para o controlador de velocidade

Como a função objetivo se mantém para todos os métodos heurísticos, para o cômputo do controlador de velocidade, utilizaremos as Tabelas 2 e 3. A partir desses valores, nossa função objetivo é pode ser dada por

$$J = 3 \int_0^T t|e(t)|dt + (m_s).$$

Do mesmo modo, utilizaremos agora (3.2) para o cálculo da função objetivo, discretizando planta e controlador, para que o anti-windup possa ser incluído nos cálculos e o controlador encontrado seja o melhor possível. Ao fazer isso, encontramos os resultados mostrados na Tabela 11.

Tabela 11 – Resultado da otimização na sintonia do controlador de velocidade utilizando o algoritmo GRASP.

Representação	Descrição	Resultado
-	Tempo de processamento	178 s
-	Número de Iterações	100000
J_{min}	Melhor Custo	63.7369
$G_{c1}(s)$	Melhor Controlador	$41.5323 + \frac{483.4974}{s}$

Fonte: Do autor.

O gráfico de otimização para esse caso está ilustrado na Figura 37. A partir dele, vemos que o algoritmo realmente otimizou e o gráfico diminuiu até encontrar seu ótimo global. A escala está em logaritmo para que tivéssemos uma melhor visualização.

A seguir, aplicamos uma entrada degrau de 100 *rad/s* no sistema $G_{c2}(s)$ - $G_2(s)$, com anti-windup, cujo a constante de tempo é

$$T_{wm,grasp} = \frac{1}{\sqrt{483.4974}} = 21.9886.$$

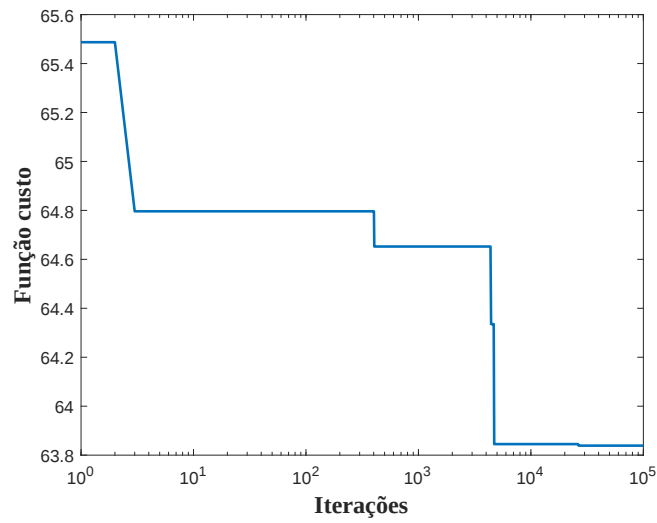
A saída, neste caso, dentro de um tempo de 10 s, é mostrada na Figura 38.

A resposta da Figura 34 tem um tempo de assentamento de 1.1005 s, um máximo sobressinal de 2.1116% e erro em regime permanente igual a 0. Todas essas características atendem ao nosso critério de desempenho, nesse caso.

3.4.2 Resultados encontrados para o controlador de corrente

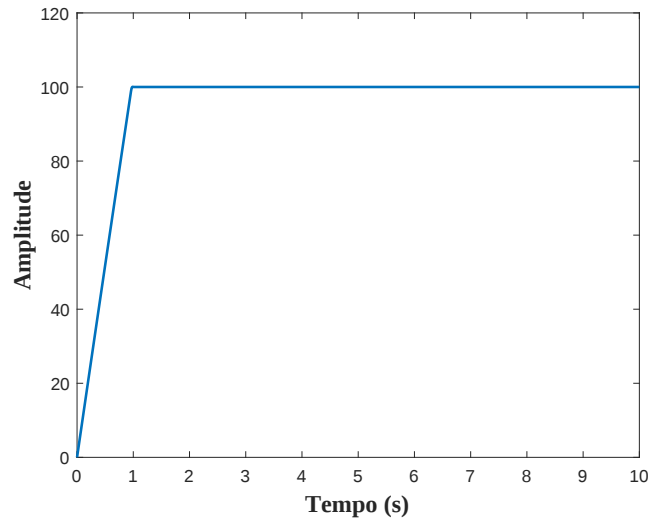
Novamente, trataremos a resolução do controlador de corrente, em que as Tabelas 2 e 3 serão a base para o cômputo do controlador. Logo, a função objetivo é dada por

Figura 37 – Gráfico da função custo no processo de otimização da planta de velocidade utilizando o algoritmo GRASP.



Fonte: Do autor.

Figura 38 – Resposta ao degrau da malha de velocidade a partir do controlador $K_P = 41.5323$ e $K_I = 483.4974$ encontrado utilizando o algoritmo *Simulated Annealing*.



Fonte: Do autor.

$$J = 3 \int_0^T t|e(t)|dt + 0.1 \int_0^T s_c(t)^2 dt + 0.1((t_s - 0.003)^2 + (t_s - 0.001)^2).$$

Da mesma maneira, o valor da função é multiplicado por 1000, a fim de evitar os erros de cálculo por aproximação. O cômputo do controlador será da mesma forma como no SA, com a discretização de controlador de corrente mais a planta, descrita em (2.16), para que o *anti-windup* possa ser inserido. Os resultados encontrados são mostrados na

Tabela 12.

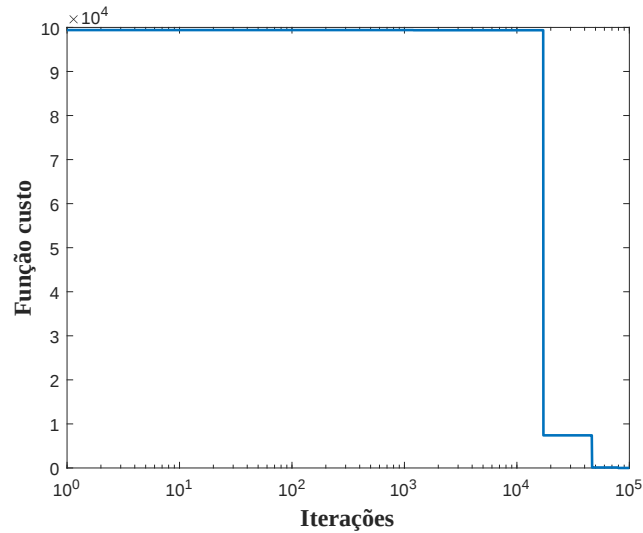
Tabela 12 – Resultado da otimização na sintonia do controlador de corrente utilizando o algoritmo GRASP.

Representação	Descrição	Resultado
-	Tempo de processamento	173 s
-	Número de Iterações	100000
J_{min}	Melhor Custo	125.1237
$G_{c1}(s)$	Melhor Controlador	$0.2795 + \frac{334.4815}{s}$

Fonte: Do autor.

O gráfico de otimização está ilustrado na Figura 39. A partir dele, vemos que o algoritmo realmente otimizou até o ótimo global. A escala está em logaritmo para que tivéssemos uma melhor visualização.

Figura 39 – Gráfico da função custo no processo de otimização do controlador de corrente utilizando o algoritmo GRASP.



Fonte: Do autor.

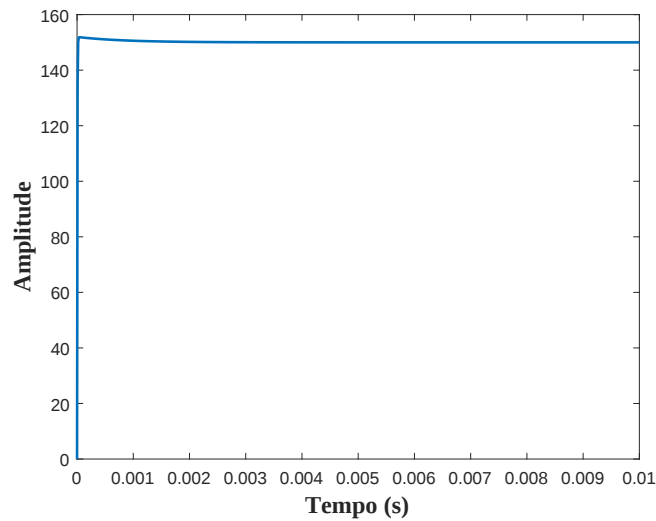
A partir desses resultados, aplicamos um degrau de 150 A no sistema $G_{c1}(s)$ - $G_{1PWM}(s)$, com o anti-windup, cujo valor da constante de tempo é

$$T_{w_{i,grasp}} = \frac{1}{\sqrt{334.4815}} = 18.2888.$$

A saída, neste caso, dentro de um tempo de 0.01 s, é mostrada na Figura 40.

A resposta da Figura 40 tem um tempo de assentamento de 0.000018 s, um máximo sobressinal de 1.2330% e erro em regime permanente igual a 0. Todas essas características atendem ao nosso critério de desempenho, nesse caso.

Figura 40 – Resposta ao degrau da malha de corrente a partir do controlador $K_P = 0.2795$ e $K_I = 334.4815$ encontrado utilizando o algoritmo GRASP.



Fonte: Do autor.

3.5 Resultados dos controladores de velocidade na máquina

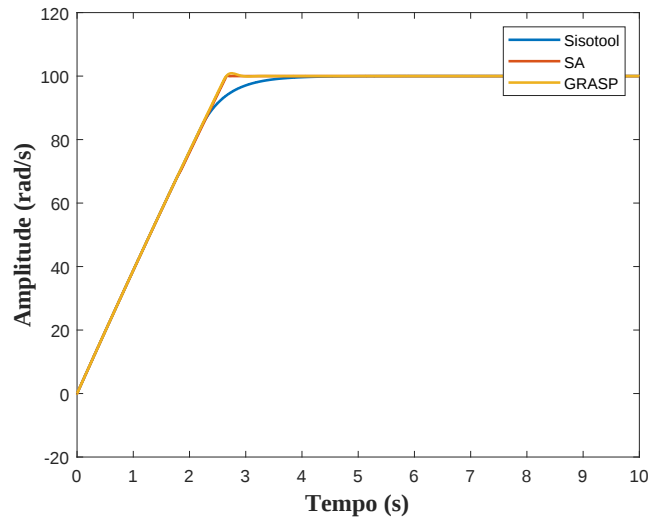
Uma vez que temos os controladores de velocidade projetados pelos três métodos descritos, pelo *sisotool* e pelas meta-heurísticas SA e GRASP, precisamos fazer uma análise comparativa dos resultados obtidos quando esses controladores são aplicados ao sistema de controle da máquina, mostrado na Figura 26. Uma vez que vimos a importância de se usar uma malha *anti-windup*, todos os resultados apresentados pressupõem a sua utilização. O resultado do controlador encontrado pelo *sisotool* foi apresentado na Figura 30, porém, retornaremos com ele, nessa seção, a fins de comparação. Quanto aos resultados dos controladores pelas meta-heurísticas, mostraremos, agora, os resultados obtidos da aplicação na máquina, fazendo o comparativo com que já tínhamos. Dessa forma, a Figura 41 mostra os resultados de saída do três controladores aplicados na máquina.

Com relação ao sobressinal, os controladores por *sisotool* e SA tiveram um sobressinal melhor. Quanto ao tempo de assentamento, os obtidos por SA e GRASP tiveram uma resposta 18% mais rápida que a do *sisotool*. Em termos numéricos, a Tabela 13 mostra os resultados de cada resposta, sendo o índice de desempenho ITAE apresentado com o intuito de definir qual foi a melhor resposta. Assim sendo, a lista é apresentada em ordem crescente ao seu valor.

De acordo com índice de desempenho, o GRASP parece ter tido a melhor resposta. Foi tão rápido quanto o controlador do SA e não teve um sobressinal tão discrepante dos outros.

Os controladores de velocidade foram simulados com seus respectivos controladores de corrente encontrados em cada técnica. Isso implica que nos caso dos controladores por

Figura 41 – Resposta da velocidade comparativa entre os controlares projetados por *sisotool*, SA e GRASP.



Fonte: Do autor.

Tabela 13 – Comparação numérica entre os controladores de velocidade projetados por *sisotool*, SA e GRASP.

Controlador	ITAE	Tempo de assentamento	Máximo sobressinal
GRASP	0.84×10^7	2.59 s	0.890%
SA	1.01×10^7	2.59 s	0.150%
<i>Sisotool</i>	1.12×10^7	3.18 s	0.002%

Fonte: Do autor.

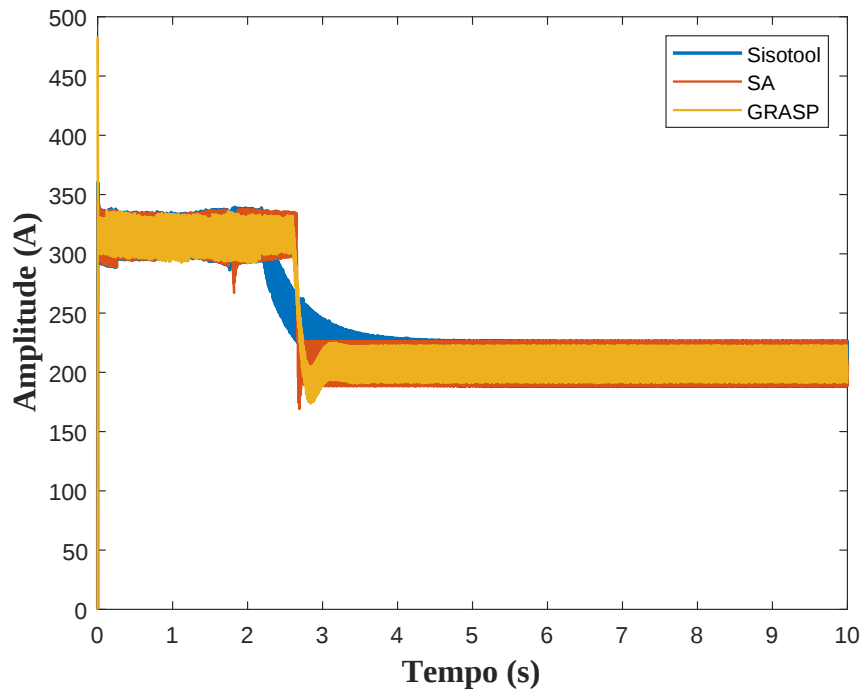
meta-heurísticas, os controladores antecipatórios foram implementados. Os resultados pros controladores de corrente serão discutidos, a seguir.

3.6 Resultados dos controladores de corrente na máquina

Para a comparação do controlador de corrente, o procedimento será o mesmo que foi feito anteriormente para o controlador de velocidade. Simularemos ambos os controladores encontrados em cada técnica: *sisotool*, SA e GRASP. No caso das meta-heurísticas, o controlador antecipatório será inserido. Utilizaremos a malha *anti-windup*, simulando o sistema da Figura 26. Dito isso, a Figura 42 mostra a saída de corrente para os três controladores.

A partir da análise dos resultados obtidos, vemos que a corrente atinge seu valor máximo, e se estabiliza depois de algum tempo. Isso é devido ao fato de sua ação ser bem mais rápida que a ação da velocidade, desse modo, enquanto a velocidade não se estabiliza, a corrente continua a ficar em nível máximo. Na seção anterior, observamos que,

Figura 42 – Resposta da corrente comparativa entre os controlares projetados por *sisotool*, SA e GRASP.



Fonte: Do autor.

a velocidade, utilizando as meta-heurísticas, tinham um comportamento de estabilização mais rápido, e, da mesma forma, a corrente também possui esse comportamento, dado que a corrente por *sisotool* se estabiliza de forma mais suave. Tendo isso em vista, o tempo de assentamento das meta-heurísticas também foi melhor. Com relação ao sobressinal, GRASP e *sisotool* tiveram maiores valores, mas sem muita discrepância do SA. Todos esses resultados, incluindo o ITAE, são listados na Tabela 14.

Tabela 14 – Comparação numérica entre os controladores de corrente projetados por *sisotool*, SA e GRASP.

Controlador	ITAE	Tempo de assentamento	Máximo sobressinal
GRASP	3.47×10^5	0.0184 s	3.24%
<i>Sisotool</i>	8.47×10^5	0.0167 s	3.00%
SA	10.70×10^5	0.0203 s	1.72%

Fonte: Do autor.

Os resultados listados na Tabela 14 foram obtidos a partir da análise da média da corrente de saída e, de acordo com os índices de desempenho, o GRASP, mais uma vez, foi o melhor. O tempo de assentamento descrito parece não condizer com o que foi preestabelecido, entretanto, a corrente teve um atraso de transporte inicial nos três casos, de 0.01666 s. Portanto, o tempo foi menor que o computado, sendo coerente com o que

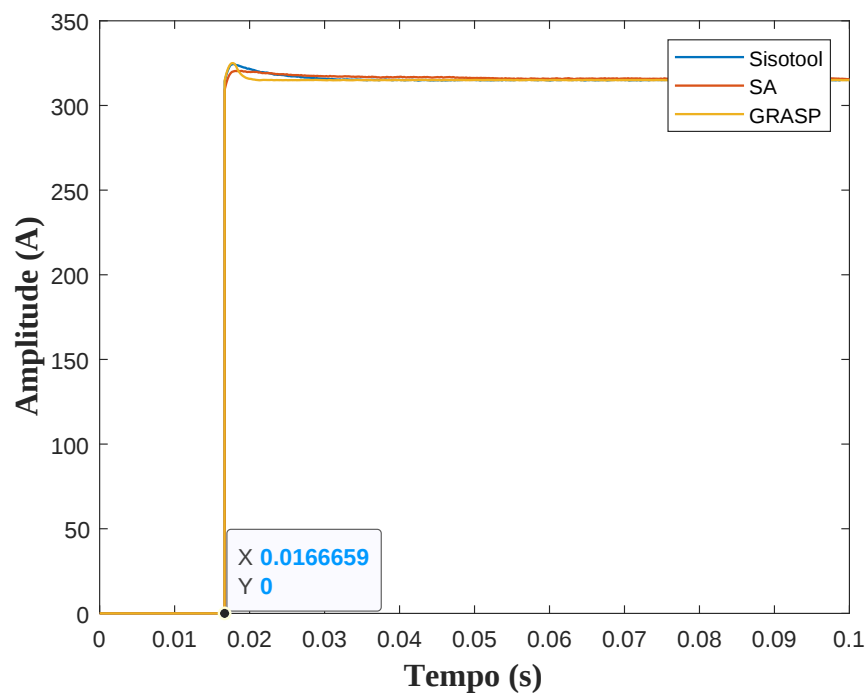
tínhamos previsto, mostrado na Tabela 15. Para comprovar o atraso, a Figura 43 mostra a saída de corrente média, nos três casos, em um menor tempo.

Tabela 15 – Comparação de tempo de assentamento sem atraso de transporte entre os controladores de corrente projetados por *sisotool*, SA e GRASP.

Controlador	Tempo de assentamento
GRASP	0.00174 s
<i>Sisotool</i>	0.00004 s
SA	0.00364 s

Fonte: Do autor.

Figura 43 – Resposta da corrente comparativa entre os controlares projetados por *sisotool*, SA e GRASP para um menor tempo.

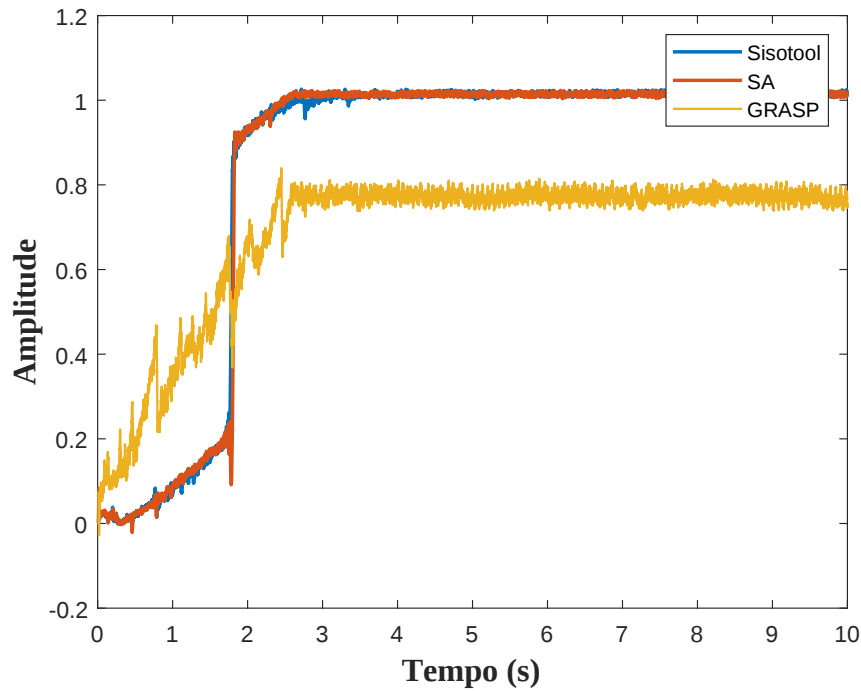


Fonte: Do autor.

Outro parâmetro que podemos observar é o esforço de controle do controlador de corrente. Vimos que o seu comportamento remete à como a velocidade se comporta, que também interfere na referência de corrente, que é o sinal de controle proveniente do controlador de velocidade. Dessa forma, vamos fazer uma última análise comparativa, que mostra como se comporta o sinal de controle para os três controladores e pode ser visto na Figura 44. Os controladores por *sisotool* e SA tiveram um comportamento muito parecido, começaram suavemente, tiveram um grande esforço, que forma uma reta quase que vertical, e depois foram se estabilizando até um valor final próximo a 1. Por outro lado, o esforço de controle do controlador por GRASP, foi um pouco mais oscilatório,

começou menos suave, mas manteve seu crescimento constante até que se atingisse um valor final de, aproximadamente, 0.8. Tendo isso em vista, podemos dizer que, por mais que tenha sido oscilatório, o esforço de controle do controlador por GRASP foi menor e mais suave em toda simulação.

Figura 44 – Esforço de controle comparativo entre os controlares projetados por *sisotool*, SA e GRASP.



Fonte: Do autor.

3.7 Conclusões parciais

Nessa seção, desenvolvemos toda teoria e analisamos os resultados obtidos.

- Utilizando os controladores obtidos por *sisotool*, geramos os resultados aplicados no sistema de controle com a máquina, modelo *built-in*, do *Simulink*. Os resultados remetem-se a utilização da malha *anti-windup* ou não. A partir disso, provou-se que é imprescindível a sua utilização nesse tipo de sistema, com relação ao desempenho que esperávamos.
- Encontramos ambos os controladores, de velocidade e corrente, a partir da heurística SA. Simulamos como os controladores reagiam, junto com o *anti-windup*, dado uma referência de velocidade ou corrente, mostrando a saída, sinal de erro e esforço de controle. Os resultados antederam as especificações de desempenho.

- Assim como no SA, encontramos os controladores de velocidade e corrente por meio do GRASP. Da mesma forma, foram testados com *anti-windup*, para uma entrada em degrau. Também foram atendidos as especificações de desempenho.
- Contendo os controladores obtidos pelos três métodos distintos, fizemos um comparativo entre a resposta de saída dos três. Para todos os métodos, os controladores de velocidade e corrente foram bem sucedidos. Para o de velocidade, os controladores obtidos por meta-heurística se mostraram melhores que o obtido por *sisotool*, o que afirma nosso objetivo principal. De acordo com o índice de desempenho ITAE, o GRASP foi o que melhor se comportou. Para o de corrente, não houve muita diferenciação entre os resultados, sendo que os controladores obtidos por meta-heurística, utilizaram-se de uma dinâmica mais simples para cálculo, dado que consideramos um controlador antecipatório no processo, a fins de simplificação. De todo modo, pelo índice de desempenho ITAE, o GRASP se mostrou, novamente, o melhor. No entanto, o SA não superou o controlador encontrado por *sisotool*.

4 Conclusão e Trabalhos Futuros

Com o crescente desenvolvimento e a necessidade de processos automáticos, fica inviável, com o passar do tempo, a utilização de malhas de controle não automáticas ou de baixo desempenho. A competitividade do mercado e o investimento em tecnologia também aumenta, e problemas envolvendo o processo, principalmente quando causados pelos controladores, não podem mais existir de forma tão persistente. Desse modo, este trabalho abordou a utilização de novos meios de sintonia de controladores além das convencionais, uma através de um *software* computacional e outra através de um algoritmo de meta-heurística, calculando dois controladores a fim de se utilizar no sistema de controle em cascata para acionamento de um motor CC

Inicialmente, foi obtido, a partir de equações que regem a máquina CC, o modelo da carga do sistema, dado os parâmetros de um motor CC de médio/grande porte, acoplado a uma carga por meio de engrenagens. Desse modelo, foram definidos os critérios de desempenho que se esperavam ao final e calculado os controladores utilizando a ferramenta *sisotool* do MATLAB. A ferramenta proporcionou que a escolha fosse feita utilizando o Lugar das Raízes, em que, preestabelecidos os critérios de desempenho, os polos estariam dentro de uma região desejada. Após a validação dos resultados, foi montado o sistema, implementado em ambiente computacional, biblioteca *SimPowerSystems* do *Simulink*, no MATLAB, no qual simulamos o sistema de controle em cascata completo. Os primeiros resultados foram obtidos, utilizando os controladores encontrados por meio do *sisotool*. Esses resultados consideraram o mesmo sistema, porém, para duas situações distintas, uma em que a malha *anti-windup* está presente e outra em que não está. Quando não há *anti-windup*, o controlador é saturado pela limitação do atuador, a velocidade sobre até alcançar seu valor máximo e só depois de, aproximadamente, 6 segundos, se estabiliza, assim como está na Figura 27. A partir disso, a corrente, nos primeiros segundos, satura, decai um pouco de valor, e quando a velocidade chega em regime permanente, tem um salto máximo negativo, e volta rapidamente, representando um mal comportamento, mostrado na Figura 28. Em contrapartida, ao simular o sistema com o *anti-windup*, a velocidade, por sua vez, não satura, mas vai aumentando de valor, suavemente, até chegar em seu valor final, por volta de 4 segundos, ilustrado na Figura 30. A corrente, satura inicialmente, mas quando a velocidade se estabiliza, ela cai para seu valor final, de forma suave, Figura 31. Tendo isso em vista, podemos dizer que é imprescindível a utilização do *anti-windup*, dado que queremos que a velocidade se comporte de forma tênue, e que a corrente não alterne de valor tão bruscamente.

Posteriormente, na intenção de encontrar melhores resultados, implementamos as meta-heurísticas *Simulated Annealing* e o *Greedy Randomized Adaptive Search Procedure*. O objetivo era desenvolver uma ferramenta que possa ser utilizada em trabalhos futuros

como uma nova forma de sintonia de controladores. A Tabela 7 mostra os dados de entrada da meta-heurística SA, que foi simulada e encontrou os controladores de velocidade e corrente, exibidos nas Tabelas 8 e 9, respectivamente. As tabelas também mostram os dados gerais das buscas. Utilizando as malhas *anti-windup* com os controladores encontrados, plotamos os comportamentos de velocidade e corrente do sistema, expostos nas Figuras 34 e 36, nessa ordem, dado uma referência degrau em ambas. Os resultados mostraram que os controladores atingiram os critérios de desempenho e podem ser testados no sistema com a máquina. Da mesma forma, a Tabela 10 apresenta os dados de entrada para a busca dos controlador e por meio da heurística GRASP. Os controladores encontrados e o comportamento dos procedimentos são expressos nas Tabelas 11 E 12, para velocidade e corrente, respectivamente. Utilizando também a malha *anti-windup*, junto aos controladores encontrados, tendo em vista uma entrada em degrau para ambos os casos, exibimos nas Figuras 38 e 40, os comportamentos de velocidade e corrente, por essa ordem. Igualmente, as respostas satisfizeram os critérios de desempenho. Para os algoritmos de busca dos controladores por meta-heurísticas, vale lembrar que, no caso dos controladores de corrente, consideramos um controlador antecipatório na malha, previamente calculado, o que fez com que a dinâmica da corrente fosse mais simples, passando de uma função de segunda para primeira ordem, facilitando os cálculos da discretização. Além disso, os resultados obtidos pela dinâmica da função de segunda ordem, não forneciam resultados satisfatórios.

Com todos os controladores calculados, fizemos um comparativo entre eles, aplicando-os no sistema completo, no Simulink, com a máquina, em cascata. Para o caso dos controladores encontrados pelas heurísticas, utilizamos um controlador antecipatório, devidamente implementado. No caso do controlador de velocidade, os resultados de saída são mostrados na Figura 41, sendo os valores de comparação, apresentados na Tabela 13. A partir desses dados, vemos que os controladores obtidos pelas meta-heurísticas, neste caso, foram mais satisfatórios. Não somente pelo fato do índice de desempenho ITAE indicar, mas pelo fato da resposta ter sido um pouco mais rápida e sem muita diferenciação de sobressinal. O GRASP, apesar de ter tido um sobressinal maior, teve um tempo de assentamento semelhante ao SA, e, pelo ITAE, foi o melhor controlador. No comportamento da corrente, a Figura 42 e a Tabela 14 exibem os resultados dos controladores aplicados. Novamente, pelo ITAE, o controlador por GRASP foi o melhor controlador, no entanto, o SA ficou pior que o obtido por *sisotool*. Visualmente falando, com o SA, o sistema teve uma resposta bem parecida com o GRASP, talvez até melhor, com o sobressinal mais baixo de todos, porém, com tempo de assentamento, mostrado na Tabela 15, maior que os outros. Além disso, os resultados para a corrente tiveram oscilação com um *ripple* considerável, o que pode fazer com que, durante o cômputo desses índices no *software*, os resultados não tenham sido bem definidos como esperávamos. Todavia, todos se mostraram eficazes em ambos os casos, sendo os controladores encontrados por GRASP os melhores. Ademais,

por fim, comparamos o esforço de controle dos controladores de corrente, mostrados na Figura 44. As curvas dos controladores por SA e *Sisotool* foram bem parecidas, contudo, a do controlador obtido por GRASP foi mais suave em todo processo, além de, ao final, ter um menor esforço para gerar o melhor resultado.

Posto tudo isso, por mais que os controladores obtidos por *sisotool* tenham sido satisfatórios, a utilização das meta-heurísticas para tal objetivo foi mais rápida e mais prática. Primeiramente pelo fato de o GRASP ter conseguido o melhor resultado em todos os quesitos, nos dois controladores e no esforço de controle. O SA, apesar de parecer ter ficado levemente pior no controle de corrente, foi melhor no controle de velocidade e igual no esforço de controle. A utilização matemática ou sintonias de projeto previamente criados, não permitem que obtenhamos a melhor configuração possível, muitas vezes. Mesmo no caso como fizemos, utilizando o Lugar das Raízes, em que o controlador pode ser ajustado para que fique dentro de uma grande região, o tempo gasto com ajustes e análise podem ser bastante demorados, dependendo do caso. Com a utilização dos métodos heurísticos, o que basta fazer é iniciar a busca e aguardar até que eles retornem os valores.

4.1 Trabalhos Futuros

Para trabalhos futuros, sugerimos como possíveis objetivos:

- Otimizar a busca do controlador de corrente, pois, apesar dos bons resultados, acreditamos que os resultados ainda possam melhorar. Isso pode ser feito, alterando os pesos da função objetivo, ou a própria função objetivo, em algum aspecto, aplicando outras meta-heurísticas, o que pode ser de grande valia para o de velocidade também, pelo menos a título de comparação.
- Aplicar os controladores na máquina, pois a resposta da corrente teve uma grande oscilação, o que dificultou, em alguns casos, analisar criticamente os resultados obtidos.
- Como um sistema geral, também fazer o acoplamento da máquina à carga e verificar o comportamento do sistema, simulando, assim, uma situação real.
- Discutir a ligação entre a simulação e a implementação real, o que envolve discretizar o controlador e inserir atrasos de transporte no modelo. Colocar também ruídos randômicos nas medidas e analisar a estabilidade do sistema, reproduzindo um caso real.
- Simulação do sistema funcionando em um *hardware-in-the-loop*.

Referências

- BAZANELLA, A. S.; DA SILVA JR, J. M. G. *Ajuste de Controladores PID*. 2000. Curso de extensão do Departamento de Engenharia Elétrica, UFRGS (Universidade Federal do Rio Grande do Sul), Rio Grande do Sul, Brasil. Disponível em: <<http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/apostila.html>>. 38
- CARVALHO, R. L. d.; ALMEIRA, T. d. S.; ROCHA, M. L. *Introdução às Metaheurísticas*. Palmas: EDUFT, 2020. Disponível em: <<http://hdl.handle.net/11612/2486>>. 20
- CASTRUCCI, P. d. L.; BITTAR, A.; SALES, R. M. *Controle automático*. Rio de Janeiro: Editora LTC, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/books/9788521635628>>. 16
- CORBERÁN, A.; MARTÍ, R.; SANCHIS, J. A grasp heuristic for the mixed chinese postman problem. *European Journal of Operational Research*, Science Direct, v. 142, p. 70–80, 2002. DOI: <[https://doi.org/10.1016/S0377-2217\(01\)00296-X](https://doi.org/10.1016/S0377-2217(01)00296-X)>. 20
- DORF, R. C.; BISHOP, R. H. *Sistemas de Controle Modernos, 13ª Edição*. Rio de Janeiro: Editora LTC, 2018. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788521635147/>>. 15
- FEO, T.; RESENDE, M. Greedy randomized adaptive search procedures. *Global Optimization*, Kluwer Academic Publisher, v. 6, p. 109–133, 1995. DOI: <<https://doi.org/10.1007/BF01096763>>. 20, 42
- FRANCHI, C. M. *Controle de processos industriais: princípios e aplicações*. São Paulo: Editora Érica, 2011. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788536518282/>>. 16, 17, 33
- GARCIA, C. *Controle de processos industriais: estratégias convencionais*. São Paulo: Edgard Blücher Ltda, 2017. Disponível em: <<https://integrada.minhabiblioteca.com.br/books/9788521211860>>. 37
- HART, D. W. *Eletrônica de Potência*. Porto Alegre: AMGH Editora, 2011. Disponível em: <<https://integrada.minhabiblioteca.com.br/books/9788580550474>>. 25, 28
- KIRPATRICK, S.; GELLAT, D.; VECHI, M. Optimization by simulated annealing. *Science*, American Association for the Advancement of Science, v. 220, p. 671–680, 1983. DOI: <<https://doi.org/10.1126/science.220.4598.671>>. 20, 41
- RASHID, M. H. *Eletrônica de Potência: Dispositivos, circuitos e aplicações*. São Paulo: Pearson Education do Brasil Ltda., 2014. Disponível em: <<https://plataforma.bvirtual.com.br/Leitor/Loader/10210/pdf>>. 15
- SKORIN-KAPOV, N.; KOS, M. A grasp heuristic for the delay-constrained multicast routing problem. *Telecommun Syst*, Springer Science, v. 32, p. 55–69, 2006. DOI: <<https://doi.org/10.1007/s11235-006-8202-2>>. 20

- SOUZA, D. A. d. *Otimização pelo método dos algoritmos genéticos e dimensionamento de estruturas tubulares metálicas espaciais com barras cruzadas para coberturas*. Tese (Doutorado) — Universidade Federal de Uberlândia, 2005. Disponível em: <<https://repositorio.ufu.br/handle/123456789/14791>>. 18
- SOUZA, J. O. d. O. d. *Metaheurísticas Aplicadas na Sintonia de Controladores PID: Estudo de Casos*. Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos, 2013. Disponível em: <<http://www.repositorio.jesuita.org.br/handle/UNISINOS/4457>>. 18, 20
- SOUZA, M. J. F. *Inteligência computacional para Otimização*. 2011. Notas de Aula do Departamento de Computação, UFOP (Universidade Federal de Ouro Preto), João Monlevade, Brasil. Disponível em: <<http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf>>. 38
- UMANS, S. D. *Máquinas Elétricas de Fitzgerald e Kingsley*. Porto Alegre: AMGH Editora Ltda., 2014. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788580553741>>. 14
- ZHANG, W. et al. Optimization with a simulated annealing algorithm of a hybrid system for renewable energy including battery and hydrogen storage. *Energy*, Science Direct, v. 163, p. 191–207, 2018. DOI: <<http://doi.org/10.1109/ACCESS.2019.2930408>>. 20



ANEXO X - TERMO DE RESPONSABILIDADE

O texto do trabalho de conclusão de curso intitulado “Estudo de técnicas de Otimização para Sintonia de Controladores PID aplicados ao Acionamento de Máquinas CC” é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico ou qualquer outro material pertencente a terceiros sem a devida citação ou consentimento dos referidos autores.

João Monlevade, 15 de Julho de 2022

Rafael Henrique Bastos Pessoa

Nome completo do(a) aluno(a)