Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Trabalho de Conclusão de Curso

Comparação entre abordagens de otimização por meta-heurísticas e programação semidefinida para a sintonia de controladores

Dalton Alex da Silva

João Monlevade, MG
2018

Dalton Alex da Silva

Comparação entre abordagens de otimização por meta-heurísticas e programação semidefinida para a sintonia de controladores

Trabalho de Conclusão de curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia de Computação pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.
Orientador: Prof.^{o} Dr.^{o} Víctor Costa da Silva Campos
Coorientador: Prof.^{o} Dr.^{o} Márcio Feliciano Braga

Universidade Federal de Ouro Preto
João Monlevade
2018

ANEXO IV – Folha de Aprovação

Curso de Engenharia de Computação

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Comparação entre abordagens de otimização por meta-heurísticas e programação semidefinida para a sintonia de controladores

Dalton Alex da Silva

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação e aprovada pela Banca Examinadora abaixo assinada:

Victor Costa da Silva Campos

Prof. Dr. Víctor Costa da Silva Campos
DELT – UFMG
Professor Orientador

Márcio Feliciano Braga

Prof. Dr. Márcio Feliciano Braga
DEELT - UFOP
Professor Coorientador

TCosta

Prof. Drª. Tatiana Alves Costa
DECSI - UFOP
Professor Convidado

Marcelo Moreira Tiago

Prof. Dr. Marcelo Moreira Tiago
DEELT - UFOP
Professor Convidado

João Monlevade, 09 de julho de 2018.

Agradecimentos

Agradeço primeiramente a Deus por caminhar lado a lado comigo nessa árdua caminhada e principalmente por ter me dado forças para continuar a lutar principalmente nos momentos mais difíceis. Agradeço a minha mãe, Maria, por estar sempre me apoiando, aconselhando e impulsionando em tudo que faço. Agradeço também a minha tia Jaqueline e ao meu tio Nilton por terem começado essa importante etapa da minha vida ao meu lado e a toda minha família.

Agradeço de forma especial ao meu orientador Víctor e meu coorientador Márcio pelo aceite à minha orientação, pela paciência e pelo conhecimento que me passaram durante todo esse tempo.

Agradeço ainda à galera do laboratório Procsimos, todos os professores do ICEA pela contribuição à minha formação como engenheiro e ,por fim, agradeço à todos os meus amigos que tive o prazer de fazer durante a minha estada em João Monlevade, obrigado pelos ensinamentos, risadas e conversas.

"Lembre-se que as pessoas podem tirar tudo de você, menos o seu conhecimento."

– Albert Einstein

Resumo

Com o desenvolvimento tecnológico cada vez a passos mais largos, o controle se torna a cada dia mais e mais importante para a ciência e engenharia. Nessa área as indústrias vêm utilizando de forma crescente o controle automático para controlar variáveis como temperatura, umidade, pressão, entre outras. Esses sistemas têm motivado o estudo da teoria de controle aplicada.

Sabe-se que a sintonia de controladores é uma das áreas mais relevantes quando falamos na teoria de controle. Controladores vem sendo utilizados por décadas, para vários fins, em vários ramos industriais. Mesmo com todo esse tempo de utilização e se agregando a mesma quantidade de tempo em estudos e pesquisas, ainda existem controladores funcionando no mercado de forma insatisfatória. Esse contexto traz desafios de análises e novas tecnologias que podem solucionar essa questão.

Neste trabalho, são apresentadas as teorias por trás dos controladores (no domínio do tempo e na frequência), incertezas e robustez de controladores e métodos de otimização, abrangendo principalmente os controladores PID e controladores por realimentação de estados, ambos robustos. Bem como as meta-heurísticas e a programação semidefinidas.

O problema da sintonia de controladores é adaptado de forma que as duas abordagens de otimização possam ser capazes de resolvê-lo e, com isso, os parâmetros de sintonia juntamente com o erro final associado foram obtidos. Análises e comparações entre as abordagens puderam ser então realizadas.

Palavras-chave: Sintonia, Controladores, Robustez, Otimização

Abstract

Technological development ever more strides, this field is becoming more and more important for science and engineering. In the control field, industries are increasingly using automatic control to control variables such as temperature, humidity, pressure, among others. These systems have motivated the study of applied control theory.

It is known that controller tuning is one of the most relevant control fields when we talk about control theory. Controllers have been used for decades for various purposes in various industries. Even though, all this time in the industry and adding the same amount of time in studies and research, there are still controllers running in an unsatisfactory way. This context brings analysis challenges and new technologies that can solve this issue.

In this work, the theories behind the controllers (in the time domain and frequency domain), uncertainties and robustness of controllers and optimization methods are presented. It mainly covers PID controllers and state feedback controllers, both robust. As well as meta-heuristics and semidefinite programming.

The problem of controller tuning is adapted, so that, the two optimization approaches can be able to solve it and then, the tuning parameters with the associated final error were obtained. Analyzes and comparisons between the approaches could be performed.

Keywords: Tuning, Controllers, Robustness, Optimization

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

BIBO	Sistema com entrada e saída limitadas - do inglês *Bounded Input Bounded Output*
GA	Algoritmo Genético - do inglês *Genetic Algorithm*
GM	Margem de ganho - do inglês *Gain Margin*
ISE	Integral do erro ao quadrado - do inglês *Integral of squared error*
ITAE	Integral do erro absoluto ponderado no tempo - do inglês *Integral of time times absolute error*
LMI	Desigualdades lineares matriciais - do inglês *Linear Matrix Inequality*
PM	Margem de ganho - do inglês *Fase Margin*
PSO	Otimização por enxame de partículas - do inglês *Particle Swarm optimization*
SISO	Sistema com entrada e saída única - do inglês *Single Input Single Output*

Lista de símbolos

a	Número aleatório uniforme independente
A, B	Matrizes da representação em espaço de estados
C	Representação do controlador
D	Termo derivativo
e	Sinal de erro
G	Representação da planta
G_l	Melhor solução global
G_p	Representação de um conjunto de plantas incertas
H_∞	Norma H infinito
i	i-ésima posição
in	Contante de inércia
I	Termo Integral
$j\omega$	Representação de números complexos
K	Ganho do modelo
K_{cr}	Valor crítico do ganho
K_d	Ganho do termo derivativo
K_i	Ganho do termo integral
K_p	Ganho do termo proporcional
L	Representação do atraso
L_o	Melhor solução local
L_p	Caminho direto do sistema em malha aberta - do inglês *Loop transfer function*
MF	função transferência em malha fechada
M_p	Sobressinal máximo percentual - do inglês *overshoot*

M_S	O pico máximo da função de sensibilidade
M_T	O pico máximo da função de sensibilidade complementar
P	Termo proporcional
P_{cr}	Termo proporcional crítico
Q^T	Transposta da matriz Q
r	Sinal de referência
s	Variável no domínio complexo de Laplace
S	Função de sensibilidade
t	Tempo
T	Função de sensibilidade complementar
TF	Função de transferência
t_s	Tempo de assentamento - do inglês *settling time*
t_r	Tempo de subida - do inglês *rise time*
T_d	Tempo derivativo
T_i	Tempo integrativo
T_f	Representação do filtro derivativo
u	Ação de controle
u_0	Valor base da ação de controle
V	Velocidade
y	Resposta temporal
y_t	Resposta transitória
$y(\infty)$	Resposta em regime permanente
X	Posição
ω	Conponentes de frequência
ω_b	Largura de banda - do inglês *bandwidth*
ω_I	Peso multiplicativo

ω_p	Peso aplicado ao desempenho robusto
ϕ	Constante de aprendizagem

Sumário

1 Considerações iniciais

Inicialmente iremos fazer algumas considerações sobre os sistema de controle, alguns termos utilizados e peculiaridades. Além disso, falaremos do processo de otimização, a motivação para a realização dessa monografia, bem como os objetivos e sua estrutura.

1.1 Sistemas de Controle

Nos dias atuais, o termo controle é bastante utilizado nos mais variados contextos. Termos como controle de estoque, controle de qualidade industrial, controle de produção, controle de frota, entre outros, cobrem vários espectros de atividades. Todos esses tomam um sistema como base, cujo comportamento deseja-se influenciar, forçando assim, o sistema a agir de uma maneira desejada (FACCIN, 2004). Segundo Bolton (1995), sistema é uma série de componentes integrados em torno de uma condição limite, com um particular interesse na relação entre suas entradas e saídas.

O ser humano é citado por Kuo (1995) como o sistema de controle mais sofisticado e complexo, pois além de ser apto a desempenhar várias tarefas, ainda consegue tomar decisões quando são necessárias. Tarefas que podem ser rotineiras, como o simples movimento do corpo humano ao se locomover, ou terem a exigência de serem desempenhadas da melhor forma possível, tal como um atleta olímpico ao participar de alguma competição.

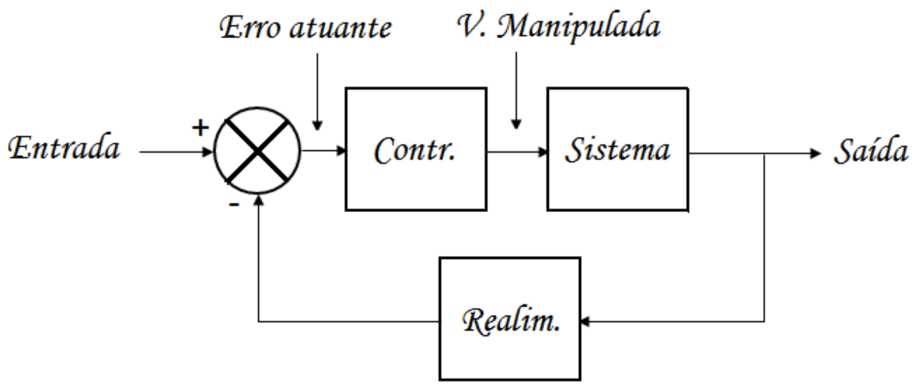
Os sistemas de controle se fazem essenciais independentemente do campo da engenharia ou da ciência. O controle automático é um componente essencial em sistemas de automação, robóticos industriais, manufatura, e quaisquer operações industriais que necessitem que variáveis como temperatura, pressão, umidade, viscosidade, vazão e muitas outras, sejam controladas (OGATA, 2010). Assim, sistemas de controle automáticos desempenham um papel cada vez mais importante no desenvolvimento da sociedade e suas tecnologias.

Antes de discutirmos os sistemas de controle, precisamos definir alguns termos utilizados nesse tipo de sistema. Inicialmente definimos o sistema a ser controlado, conhecido como planta (G). Este pode ser constituído por equipamentos, partes desses equipamentos ou um conjunto de componentes desses equipamentos que funcionem de maneira conjunta capazes de efetuar as ações necessárias visando desenvolver a operação ou resultado desejado (OGATA, 2010). Essa, na maioria dos casos, é a parte fixa do sistema de controle, isto é, a parte que normalmente não se pode modificar em virtude de ser fruto de um projeto, ou seja, todas as suas partes foram planejadas e seus componentes escolhidos em função do planejamento proposto. São exemplos de plantas: fornos, componentes robóticos, linhas de montagem, reatores, entre outros.

Em contrapartida, os outros dois importantes componentes presentes nesse tipo de sistema são: o controlador (C) e a realimentação. Estes atuam com a finalidade de melhorar o desempenho do sistema, ou seja, para que a planta possua estabilidade e opere com certa precisão, leveza e rapidez, seguindo as especificações uma vez estabelecidas. Logo, controle realimentado refere-se a comparação entre valores de referência e a saída do sistema (OGATA, 2010).

O resultado da comparação é o centro de toda a teoria estudada em sistemas de controle, e é denominado sinal de erro atuante (e). Após medido, esse sinal é levado ao C, que produz o que chamamos de sinal de controle (u), cuja função resume-se em reduzir o desvio entre a saída e o sinal desejado. A Figura 1 representa a estrutura de um sistema de controle (OGATA, 2010).

Figura 1 – Estrutura de um sistema de controle



Fonte: Elaborada pelo próprio autor.

Como dito anteriormente, plantas são modelos de um sistema real já implantado. Porque o ambiente real pode mudar com o tempo (componentes podem envelhecer ou seus parâmetros podem variar com temperatura ou outras condições ambientais), ou condições operacionais podem variar (mudanças de carga, perturbações), o sistema de controle deve ser capaz de resistir a variações. Mesmo se o ambiente não mudar, outro fato vital é a incerteza do modelo, pois qualquer representação matemática de um sistema envolve frequentemente hipóteses simplificadoras. A propriedade particular que um sistema de controle tem que possuir, para operar corretamente em situações realistas é chamada robustez. Matematicamente, significa que o controlador não deve funcionar satisfatoriamente só para uma planta, mas para uma família (ou conjunto) de plantas. Uma vez que a condição é atendida, o controlador é considerado robusto (SKOGESTAD; POSTLETHWAITE, 2005).

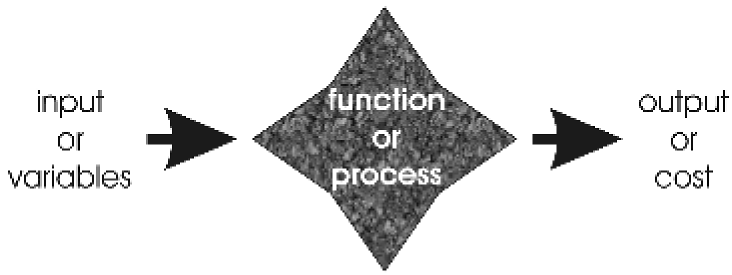
Uma vez trabalhando com sistemas robustos, necessitamos quantificar a incerteza dinâmica não modelada, para tal, a abordagem do domínio da frequência deve ser utilizada. Além disso, incertezas paramétricas também são frequentemente representadas por perturbações complexas, simplificando assim a análise.

Assim, se o sistema satisfaz as especificações de desempenho tais como: rastreamento de estado em regime, rejeição de perturbação, exigências de velocidade de resposta e além disso, o controlador satisfaz a todas as exigências para todos os valores na faixa de frequências especificada, podemos dizer que o sistema possui desempenho robusto (SKOGESTAD; POSTLETHWAITE, 2005).

1.2 Otimização

Otimização é o processo de extração do melhor rendimento possível, no que concerne a qualquer área de atividade. A ideia central consiste em tentar variações de um conceito inicial e usar essa informação extraída para melhorar a ideia a cada iteração. Em outras palavras, como apresentado na Figura 2, otimização consiste em ajustar alguns parâmetros de entrada ou processos, com a finalidade de encontrar o mínimo ou o máximo a partir desse sistema (HAUPT; HAUPT, 2004).

Figura 2 – Diagrama do processo de otimização.



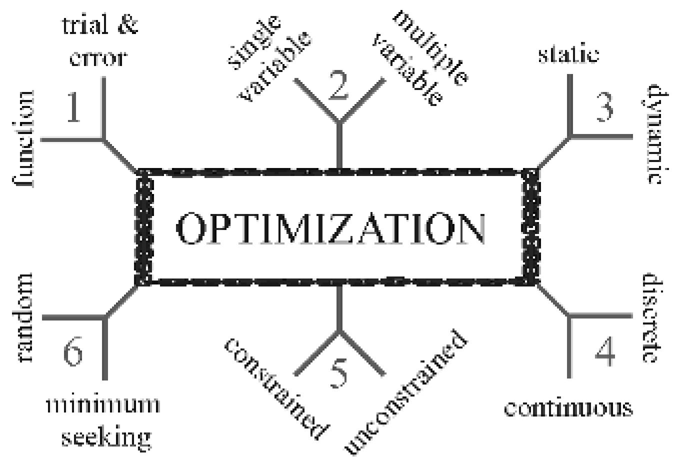
Retirada de (HAUPT; HAUPT, 2004).

Uma terminologia bastante utilizada na área de otimização é “a melhor solução”. Quando estamos procurando por uma determinada solução, na maioria das vezes, queremos a melhor solução. Um fato que precisamos ficar bastante atentos é que essa solução não é única e, além disso, as variações dessas soluções podem ter valores diferentes. A definição de qual é a melhor solução entre as encontradas dependerá de alguns fatores que precisamos levar em consideração, tais como: as tolerâncias permitidas, o problema na sua formulação e os métodos utilizados para a solução do problema (HAUPT; HAUPT, 2004). Como exemplificação para esse tipo de citação podemos tomar como base alguns problemas que em sua natureza não possuem raízes exatas, tais como: escolher a melhor música clássica de todos os tempos, definir o melhor jogador de futebol da atualidade, entre outros.

Se compararmos a otimização com os métodos de busca por raízes exatas, podemos fazer algumas considerações tais como: métodos de busca de raízes exatas buscam por zeros de uma determinada função, ou seja, buscam por uma solução exata, enquanto a otimização trabalha para encontrar zeros da função derivada, ou seja, buscam a melhor solução possível, levando em consideração algumas definições e restrições. Em adição a essa questão, algumas vezes as derivadas simplesmente não existem ou podem ser extremamente difíceis de se encontrar. Outra dificuldade quando falamos de otimização, é determinar se um mínimo encontrado é o mínimo global, ou simplesmente mais um mínimo local. Nos métodos de busca de raízes exatas, esse cenário não existe, pois qualquer raiz encontrada, desde que essa leve ao zero da função trabalhada, é válida (HAUPT; HAUPT, 2004).

As seis categorias dos algoritmos de otimização são divididos são apresentadas na Figura 3. Nesta monografia os algoritmos trabalhados são do tipo: funções, com variáveis únicas, estáticos, contínuos, com restrições e procura por mínimos globais (HAUPT; HAUPT, 2004).

Figura 3 – Seis categorias dos algoritmos de otimização.



Retirada de (HAUPT; HAUPT, 2004).

1.3 Motivação

O controle Proporcional Integral Derivativo (PID) continua sendo o mais utilizado no meio industrial. A grande maioria de todas as malhas existentes são do tipo PI / PID atingindo uma grande gama de aplicações, tais como: controle de processos, sistemas embarcados, indústria automobilística, controladores de voo, sistemas biológicos, entre outros (ARRUDA et al., 2008).

Por mais que os controladores PID vêm se mostrando dominantes no mercado por mais de 60 anos, os três parâmetros, ou seja, os termos proporcional (P), integral (I) e derivativo (D) que precisam ser ajustados no controlador, ainda geram desafios nos dias atuais. Muitos métodos de sitonia de controladores foram propostos diante do fato que os controladores podem ser ajustados para uma gama diferente de atividades (FACCIN, 2004).

Algumas variações dos controladores PID também são bastante utilizadas no mercado. O fato é que esse termo é bastante difícil de se sintonizar, então as empresas preferem de forma prática evitarem o controle derivativo (FACCIN, 2004).

Apesar de toda a sua importância em muitos ramos, os controladores PID ainda apresentam desempenho insuficiente em algumas aplicações. Podemos ver na literatura que, uma porcentagem significante dos controladores instalados operam em modo manual e que mais da metade dos PIDs que operam em modo automático apresentam grandes variações (ARRUDA et al., 2008).

Vários métodos para a sintonia de controladores foram desenvolvidos e melhorias vêm sendo apresentadas ao passar dos anos. Metodologias simples, algumas complexas, outras aplicadas na área de multivariáveis, controles robustos, baseados em modelos, entre outros.

Nos últimos tempos a indústria química vem utilizando o controle preditivo como estratégia de controle avançado. Esse tipo de processo requer malhas bem sintonizadas. Como os processos químicos são naturalmente multivariáveis, há uma grande necessidade de uma estratégia de sintonia e além disso, robusta (ARRUDA et al., 2008).

As citações anteriores representam as principais motivações para a realização deste trabalho. Por meio delas, podemos notar que o controlador PID é o mais utilizado industrialmente, bem como o mais estudado, mas controladores desse tipo não estão funcionando satisfatoriamente na maioria das aplicações industriais. Dentre todos os problemas que podem estar envolvidos, podemos citar alguns como: a ausência de conhecimentos dos operadores e engenheiros, a adoção de métodos de sintonia que não são compatíveis com o processo em análise, a grande variedade de controladores PIDs que podem levar a erros em aplicação de regras de sintonia, entre outros.

Assim, este problema continua desafiador e ainda há espaço pra o desenvolvimento de novas tecnologias voltadas para a sintonia de controladores.

1.4 Objetivos

Este estudo tem como objetivo principal a comparação entre a programação semidefinida e meta-heurísticas (por exemplo, Otimização por enxame de partículas - do inglês *Particle Swarm Optimization* (PSO), Algoritmo Genético - do inglês *Genetic Algorithm* (GA), Simulated Annealing) para a sintonia de controladores para sistemas lineares invariantes no tempo, em tempo contínuo.

Assim, este trabalho irá implementar estratégias de sintonia utilizando programação semidefinida (LMIs) em um *toolbox* livre disponível no Matlab para esse tipo de problema e estratégias de sintonia utilizando alguma abordagem meta-heurística. Tendo todas essas estratégias implementadas, uma comparação entre as abordagens será realizada.

1.5 Estrutura da monografia

No Capítulo 1, é feita uma breve explicação sobre os sistemas de controle e processos de otimização. Além disso, são apresentados a motivação e os objetivos desta monografia. O Capítulo 2 apresenta uma pequena parte da teoria sobre controladores PID, bem como, seus modos, tipos e projetos de sintonia. O capítulo apresenta também alguns conceitos aplicados a realimentação de estados e sua modelagem. No Capítulo 3, é apresentada uma análise de desempenho em malha fechada tanto no domínio do tempo quanto no domínio da frequência. Além disso, é feita uma abordagem sobre sistemas incertos e condições de robustez. O Capítulo 4 aborda uma teoria sobre os métodos de otimização e uma explicação do funcionamento dos algoritmos aplicados a este trabalho. No Capítulo 5, são apresentados os métodos e procedimentos adotados para resolver o problema de sintonia de controladores. No Capítulo 6, são apresentados resultados e realizadas análises acerca dos resultados obtidos. Por fim, no Capítulo 7, são mostradas as conclusões desta monografia.

2 Controladores

2.1 Modos de controle

Antes de falarmos dos métodos de sitonia precisamos conhecer quais as principais características das várias ações de controle constitutivas de um PID. Inicialmente precisamos definir qual a melhor escolha dos modos a se utilizar (proporcional, derivativo, integral, ou uma combinação deles). A partir dessa definição escolhemos quais variações podem ser utilizadas: P, PI, PD ou PID. (LOURENÇO, 1997).

2.1.1 Modo Proporcional

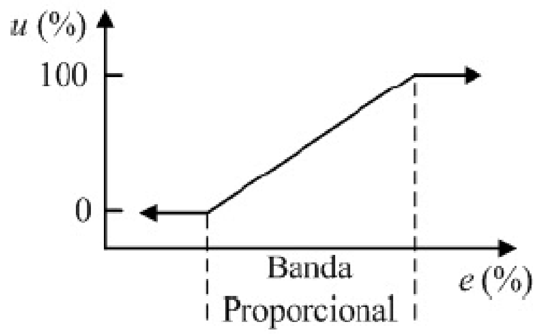
A ação de controle proporcional consiste essencialmente em um amplificador com um ganho constante, ou seja, ela é diretamente proporcional a sua entrada. Isso significa que o sinal de erro está em função do tempo. Isso pode ser demostrado por meio de (2.1) (FACCIN, 2004)

$$u(t) = K_p e(t) + u_0. \tag{2.1}$$

Sendo $u(t)$ a ação de controle (em função do tempo), K_p o ganho do termo proporcional e u_0 o valor base da ação de controle.

A ação de controle em função do erro gerada pela ação proporcional pode ser visualizada na Figura 4. Podemos notar que para cada valor de erro, há um único valor da ação de controle correspondente uma vez que a faixa de saturação foi excluída (FACCIN, 2004).

Figura 4 – Ação de controle proporcional em função do sinal de erro.



Retirada de (FACCIN, 2004).

Uma característica bastante pertinente da ação de controle proporcional é o rápido ajuste da variável manipulada. Porém, juntamente com essa característica vem sua principal desvantagem. Sempre que há alguma modificação nos parâmetros do sistema a se controlar, há também um erro residual permanente, ou seja, um erro em regime permanente. Esse erro estacionário é dependente de K_p, isso significa que o aumento do parâmetro pode diminuir o erro, porém, essa ação pode conduzir a um aumento do tempo de estabelecimento, tornando o controlador mais oscilatório (FACCIN, 2004; LOURENÇO, 1997).

2.1.2 Modo Integral

A ação de controle integral é diretamente dependente à integral do sinal de erro no domínio do tempo, como pode ser por verificado por (2.2) (FACCIN, 2004)

$$u(t) = K_i \int_0^t e(t)dt + u_0. \tag{2.2}$$

Sendo K_i o ganho do termo integral.

Note que (2.2) evidência que a ação de controle integral está intimamente ligada a dependência do sinal de erro ao longo do tempo desde seu início ($t = 0$) até o momento atual (FACCIN, 2004; SEBORG et al., 2004).

A principal função da ação de controle integral é a eliminação do erro em regime permanente (offset). Porém, a utilização dessa ação de controle pode reduzir a estabilidade da malha de controle (FACCIN, 2004; SEBORG et al., 2004).

2.1.3 Modo Derivativo

A ação de controle derivativo tem a função de antecipar a ação de controle considerando as taxas de mudanças a fim de que o processo reaja mais rapidamente, pois dependendo da dinâmica do processo, o sinal de controle estará em atraso para corrigir o erro. Isso significa que essa ação é proporcional à taxa de variação do sinal de erro, de acordo com (2.3) (PINTO, 2014; FACCIN, 2004; SEBORG et al., 2004)

$$u(t) = K_d \frac{de(t)}{dt} + u_0. \tag{2.3}$$

Sendo K_d o ganho do termo derivativo.

Analisando a equação (2.3) podemos perceber que, uma vez que o erro seja constante, ou seja, quando a derivada for nula, o valor de saída da ação de controle será simplesmente igual a u_0. Consequentemente, a ação de controle derivativa nunca poderá ser utilizada por si só e sim combinada com uma ação proporcional ou uma proporcional-integral, pois por mais que o erro seja grande, se não variar no tempo, o modo derivativo não atuará (SEBORG et al., 2004; FACCIN, 2004).

Assim, a ação de controle derivativa aspira a otimizar a resposta dinâmica da variável controlada pelo tempo de estabilização. Como essa ação trabalha de forma antecipatória, ela é capaz de aumentar a velocidade de correção do processo (SEBORG et al., 2004; FACCIN, 2004).

Por outro lado, se altas frequências estiverem presentes no sistema, tais como, ruídos ou flutuações aleatórias, então a derivada do sinal medido irá mudar descontroladamente. Consequentemente, amplificará altas frequências (SEBORG et al., 2004; FACCIN, 2004).

2.2 Controlador PID

O controlador proporcional integral e derivativo é gerado a partir da combinação dos três modos básicos de controle contínuo apresentados na Seção 2.1. O controlador PID gera a sua saída proporcionalmente ao sinal de erro, à integral e à derivada do mesmo. Resultando nas vantagens e desvantagens de um controlador PI e de um controlador PD. A forma geral do controlador PID é dada por (FACCIN, 2004; PINTO, 2014; SEBORG et al., 2004; LOURENÇO, 1997)

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}. \tag{2.4}$$

No controlador PID, o modo integral é utilizado para a eliminação do erro estacionário causado na maioria das vezes por variações de carga. O modo derivativo possibilita um aumento do ganho e faz com que as oscilações tendam a zero. Essas características resultam em uma velocidade de resposta superior se compararmos com um controlador PI (LOURENÇO, 1997).

Analisando o controlador PID podemos listar na Tabela 1 os efeitos na resposta do sistema em malha fechada. Note que a tabela deve ser utilizada uma vez que estamos definindo os parâmetros do controlador, pois, quando alteramos qualquer parâmetro podemos alterar o efeito de outra ação, uma vez que as correlações não são exatas. (LOURENÇO, 1997)

Tabela 1 – Efeitos na resposta do sistema em malha fechada para uma entrada degrau.

Resposta	Tempo de subida	Sobreelevação	Tempo de estabelecimento	Erro estacionário
Proporcional	Diminuição	Aumento	Sem alteração	Diminuição
Integral	Diminuição	Aumento	Aumento	Elimina
Derivativo	Sem alteração	Diminuição	Diminuição	Sem alteração

2.2.1 Tipos de algoritmos PID

Podemos encontrar várias formas de combinações entre as ações proporcional, integral e derivativa no mercado. Esse fato gera alterações no algoritmo dos controladores PID de diferentes fabricantes. Essas formas de controle são combinadas de acordo com a necessidade de desempenho de forma específica (FACCIN, 2004; PINTO, 2014).

Apesar da grande variação, duas delas se destacam no mercado e assim precisamos destaca-las. São elas a forma série ou interativa e a paralela ou não interativa (SEBORG et al., 2004; FACCIN, 2004; PINTO, 2014).

2.2.1.1 PID série ou interativa

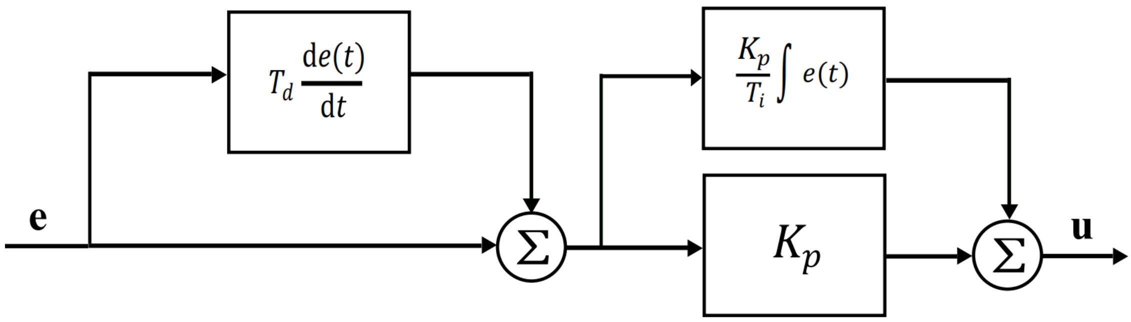
Na época dos controladores analógicos físicos (tanto os eletrônicos quanto os pneumáticos), elementos de um controlador PI e de um controlador PD operavam em série. Visando a continuidade desses tipos de controladores, alguns fabricantes permaneceram com essa mesma estrutura em seus dispositivos eletrônicos. O controlador do tipo série, após a aplicação da transformada de *Laplace* é (SEBORG et al., 2004; FACCIN, 2004; PINTO, 2014)

$$C(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{sT_i}\right)(1 + sT_d) . \tag{2.5}$$

Sendo s a variável no domínio complexo de Laplace, T_i o tempo integrativo e T_d o tempo derivativo.

Como o termo derivativo tem influência direta no termo integrativo, pois este é adicionado em série, esse tipo de algoritmo também é conhecido como PID interativo. Esse tipo de configuração não está presente nos controladores PID ideais, chamados de não interativos, pois um termo não influencia os outros. Vale ressaltar que quando formas alternativas que não contêm a parte derivativa são utilizadas, a forma ideal e a série são equivalentes. A representação esquemática de um PID série é representada na Figura 5 (PINTO, 2014; SEBORG et al., 2004; FACCIN, 2004).

Figura 5 – Representação esquemática da forma série do controlador PID.

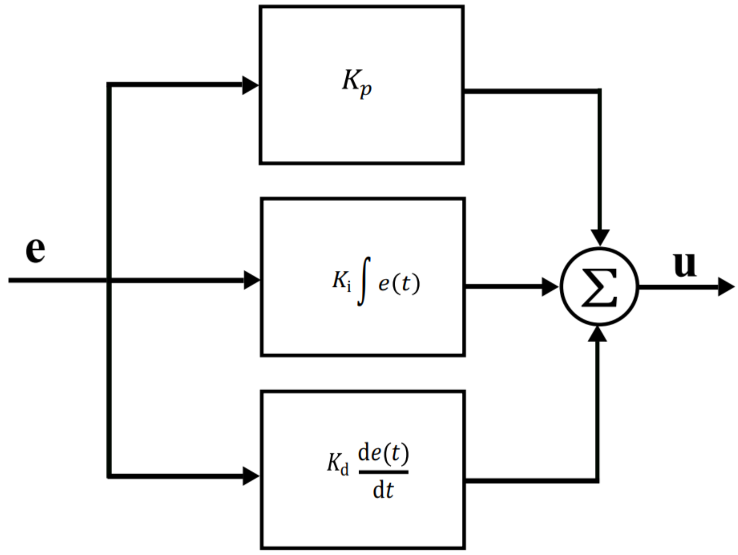


Retirada de (PINTO, 2014).

2.2.1.2 PID paralelo ou não interativa

Diferentemente da forma série, a forma paralela é assim chamada pois combina os modos de controle paralelamente. Logo esse tipo de controlador pode ser visto como três ações de controle trabalhando separadamente sobre o sinal de erro e sendo combinadas no final, ou seja, as ações são calculadas separadamente e somadas na saída para formar a ação do controlador. A Figura 6 ilustra as ações desse tipo de controlador (PINTO, 2014; SEBORG et al., 2004; FACCIN, 2004).

Figura 6 – Representação esquemática da forma paralela do controlador PID.



Retirada de (PINTO, 2014).

A forma do controlador PID paralelo, após a aplicação da transformada de Laplace é dada por

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + sK_d. \tag{2.6}$$

Infelizmente, o controlador tal como descrito em (2.6) é uma forma ideal e não é fisicamente realizável, pois não pode ser exatamente implementado utilizando-se tecnologia analógica ou digital. Assim, para controladores analógicos podemos aproximar (2.6) utilizando um filtro derivativo. Além de deixar o controlador realizável, reduz o efeito da derivada dos ruídos de medição (que normalmente é alto) na ação de controle, uma vez aplicado ao ganho K_d. A forma paralela do controlador aplicando o filtro derivativo, é dada por

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1}. \tag{2.7}$$

2.2.2 Projeto de um controlador PID

Para uma determinada aplicação precisamos definir qual tipo de controlador utilizar. Porém isso, na maioria das vezes, não é uma tarefa trivial. A seleção do tipo de controlador deve depender das condições de operação do sistema, além de certas especificações que precisam ser atendidas, tais como o tempo de estabelecimento permitido, erro estacionário máximo, sobre-elevação, entre outras (LOURENÇO, 1997).

Quando projetamos um controlador, é desejável que possamos ajustar alguns parâmetros e que esse, por sua vez, influenciem no desempenho. Mediante esse fato, a questão que temos para responder é: como selecionar os parâmetros dos controladores de modo a obtermos uma resposta satisfatória às especificações de um determinado sistema? (LOURENÇO, 1997; FACCIN, 2004).

Apesar de uma infinidade de métodos de ajustes que já foram propostos nos últimos sessenta anos, alguns controladores ainda estão sendo ajustados manualmente com os parâmetros escolhidos por tentativa e erro. Porém como estamos a cada dia desenvolvendo e trabalhando com sistemas mais complexos, deseja-se fazer a utilização de algum mecanismo sistemático, que possa ajustar esses parâmetros de uma forma à obter um melhor desempenho do sistema.

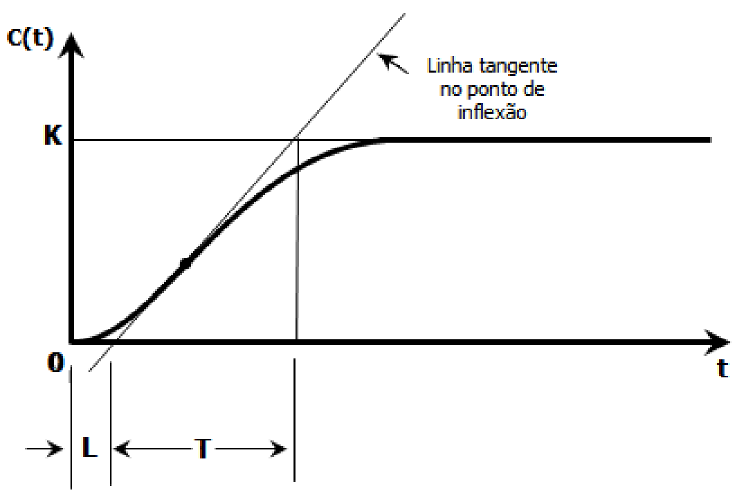
Dentre os vários métodos utilizados, os mais conhecidos são aqueles que se baseiam em um modelo aproximado do processo. Assim iremos discutir a seguir um dos mais famosos métodos que se baseiam nessa técnica.

2.2.2.1 Regras de sintonia de Ziegler-Nichols

J. G. Ziegler e N. B. Nichols sugeriram regras para a sintonia de controladores PID, ou seja, sugeriram ajustes para os valores de K_p, T_i e T_d. Os autores desenvolveram regras baseadas de forma experimental em uma entrada degrau ou no valor de K_p. Essas regras empíricas foram realizadas tendo como base o controlador comercial $Fulscope$ da $Taylor$. Vale ressaltar que esses valores são úteis uma vez que o modelo matemático da planta não é conhecido, porém, se for conhecido isso não impede a aplicação das regras. Assim podemos dizer que as regras de sintonia de Ziegler-Nichols são um ponto de partida para o ajuste de sintonia fina pra controladores PID. Existem dois métodos que denominam as regras de Ziegler-Nichols (OGATA, 2010; FACCIN, 2004).

O primeiro método utiliza uma entrada degrau unitário como experimento e então se obtém a resposta da planta. A Figura 7 mostra uma curva de resposta ao degrau unitário em aspecto de S, uma vez que a planta não possui integradores e nem polos complexos conjugados (OGATA, 2010).

Figura 7 – Curva de resposta em forma de S.



Retirada de (OGATA, 2010).

Analisando a imagem, podemos observar a existência de duas constantes, o atraso L e T que nesse cenário representa o tempo. Essas constantes podem ser determinadas através do traçado de uma linha tangente no ponto de inflexão da curva e assim determinando a interseção desta com o eixo dos tempos. Levando em consideração essas constantes, Ziegler e Nichols sugeriram valores de sintonia de acordo com a Tabela 2. Portanto, o controlador PID tem um polo na origem e zeros duplos em $s = \frac{-1}{L}$. (OGATA, 2010)

Tabela 2 – Efeitos na resposta do sistema em malha fechada.

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0{,}9\frac{T}{L}$	$\frac{L}{0{,}3}$	0
PID	$1{,}2\frac{T}{L}$	2L	0,5L

O segundo método define inicialmente os valores de T_i=∞ e T_d=0. Em seguida usa-se o valor de K_p iniciando de zero ao seu valor crítico (K_{cr}) , no qual a saída deve exibir inicialmente uma oscilação sustentada. Caso a saída não exiba essa oscilação pela primeira vez para qualquer valor de K_p, esse método não pode ser aplicado. Ziegler e Nichols sugeriram valores de sintonia de acordo com a Tabela 3. Portanto, o controlador PID tem um polo na origem e zeros duplos em $s = \frac{-4}{P_{cr}}$. Sendo P_{cr} o período crítico (OGATA, 2010)

Tabela 3 – Efeitos na resposta do sistema em malha fechada.

Tipo de controlador	K_p	T_i	T_d
P	$0{,}5K_{cr}$	∞	0
PI	$0{,}45K_{cr}$	$\frac{1}{1{,}2}P_{cr}$	0
PID	$0{,}6K_{cr}$	$0{,}5P_{cr}$	$0{,}125P_{cr}$

2.3 Realimentação de estados

A teoria de controle moderno foca no conceito de estados. A modelagem de cada estado inclui o *status* interno do sistema além do comportamento único quanto a entrada-saída em um determinado instante de tempo. A determinação de um certo estado está atrelada ao conhecimento do estado anterior e do entrada aplicado ao sistema. Porém uma vez que o novo estado é definido, ele não dependerá da estado anterior (BROGAN, 1991).

Podemos claramente perceber as vantagens inseridas neste contexto, uma vez que todos os estados estão disponíveis para uso na implementação da lei de controle. Porém, esse segmento é somente para interesses acadêmicos, pois por definição o único sinal acessível de todo o sistema é a saída (BROGAN, 1991).

Mesmo com a situação citada no parágrafo anterior, referente aos estados existentes disponíveis para a ação de controle, há algumas razões pelas quais os $feedbacks$ dos estados são considerados na teoria de controle: (BROGAN, 1991):

1. Em cada estado estão contidas informações sobre o sistema como um todo e, assim, podemos estudar o que poderia ser realizado naquele espaço de tempo.

2. Há instâncias para as quais as saídas de cada estado são mensuráveis.

3. Várias leis de controle ótimo fazem uso de $feedback$ de cada estado.

4. Existem algumas formas eficazes para se estimar ou reconstruir as variáveis de estado utilizando entradas ou saídas disponíveis.

Para que possamos aplicar a técnica de realimentação de estados para o projeto do sistema de controle, precisamos garantir que seja controlável. Controlabilidade é a propriedade que envolve a entrada do sistema e o estado em questão, assim envolve somente as matrizes A e B do sistema em espaços de estados.

Um sistema

$$\dot{x} = Ax + Bu \tag{2.8}$$

é dito linear controlável em um determinado instante de tempo se existe alguma função de entrada que transferirá o estado para o estado final em um tempo finito (BROGAN, 1991)

Matematicamente de acordo com Brogan (1991), um sistema em espaço de estados, (2.8), é controlável se e somente se a matriz de controlabilidade,

$$M_{controlabilidade} = [B|AB|...|A^{n-1}B] \tag{2.9}$$

tem posto igual a |n|, uma vez que a dimensão dessa matriz é |n x np|, em que p pode ser qualquer valor inteiro.

Ainda de acordo com Brogan (1991), se e somente se um sistema em malha aberta é controlável, então qualquer conjunto de autovalores desejáveis pode ser alcançado utilizando a matriz de ganho de uma sistema em espaço de estados. Uma forma simples de se obter os valores dessa matriz é resolvendo sua equação característica dada pelo determinante de $det[sI - (A - BK)]$ e considerando que a posição desejada dos polos é conhecida, logo o determinante calculado anteriormente pode ser igualado aos polos. Assim, os elementos da matriz de ganho são obtidos por simples casamento de coeficientes. Uma outra forma de obter esses valores é com a utilização da forma canônica controlável, a partir da qual os elementos da matriz podem ser obtidos diretamente.

3 Análise de desempenho em malha fechada

3.1 Desempenho no domínio do tempo

Como dito anteriormente, a obtenção do modelo matemático que descreve a planta a ser controlada é o primeiro passo a ser dado. Uma vez o modelo obtido, é possível realizar várias análises do sistema quanto ao seu desempenho (OGATA, 2010).

Tanto na análise quanto no projeto de sistemas de controle, devemos estabelecer uma base de comparação detalhando-se os sinais de entrada de teste específicos, e assim, comparando-se as respostas do sistema com esses sinais. Muitos dos critérios de projeto e análise desse tipo de sistema leva como base as respostas a esses sinais juntamente às mudanças do sistema quanto as condições iniciais (OGATA, 2010).

Utilizam-se geralmente funções degrau, rampa, parábola de aceleração, impulso, senoidais e ruídos branco como sinais de entrada que formam uma base de teste. A partir desses, tanto a análise experimental como a matemática podem ser realizadas uma vez que esses sinais estão no domínio do tempo. Assim, uma vez que o sistema de controle seja projetado com base nos sinais de teste, o uso dos mesmos possibilita a comparação do desempenho em relação à mesma base. (OGATA, 2010).

A resposta temporal de um sistema é dividida em duas partes: a resposta transitória ($y_t(t)$) e a resposta estacionária (em regime permanente) ($y(\infty)$). A primeira entende-se como a que vai do estado inicial ao estado final, ou seja, é definida como a parte da resposta que tende a zero quando o tempo tende ao infinito. Pela estacionária, entende-se como o comportamento da saída do sistema à medida que t tende ao infinito. Logo a resposta $y(t)$ é dada pela equação (3.1) (OGATA, 2010)

$$y(t) = y_t(t) + y(\infty). \tag{3.1}$$

Uma das características mais importantes do comportamento dinâmico de um sistema de controle é a estabilidade. Considera-se um sistema de controle linear e invariante no tempo como estável, se a saída retornar ao estado de equilíbrio quando o sistema é submetido a uma condição inicial. Outro comportamento importante do sistema que necessita de uma atenção especial é o erro estacionário. A resposta transitória de um sistema de controle prático na maioria das vezes apresenta oscilações amortecidas antes de atingir o estado permanente. Uma vez que o sinal de saída não coincida exatamente com a entrada, dizemos que o sistema apresenta um erro estacionário.

Neste trabalho foram aplicadas 5 penalidades ao sistema conforme especificadas no Capítulo 5. Dentre as penalidades, temos uma que leva em consideração a estabilidade do sistema, ou seja, a posição dos polos no plano S, outra que se preocupa com o esforço máximo de controle, restando 3 penalidades que são especificações da resposta transitória. Quando tratamos de sistemas reais, a resposta transitória de um sistema de controle apresenta, quase sempre, oscilações amortecidas. A resposta transitória especifica na maioria das vezes as características de desempenho de um sistema de controle. Como utilizamos 3 dessas características para avaliar o sistema apresentado neste trabalho, elas são definidas como (OGATA, 2010):

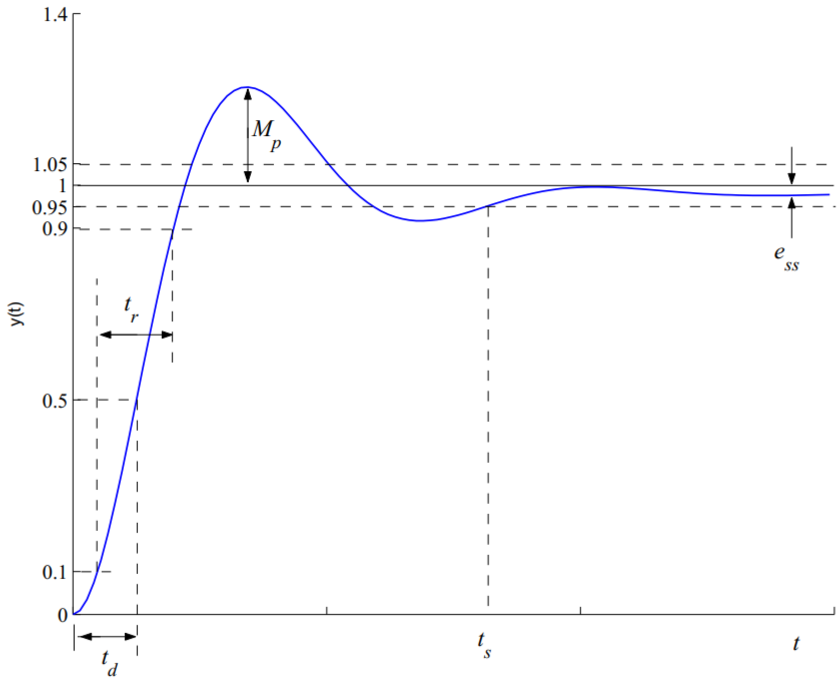
1. Máximo sobressinal percentual (M_p), do inglês *overshoot*: é a diferença entre o valor máximo de pico atingido e o valor final em percentual do valor final.

2. Tempo de assentamento (t_s), do inglês *settling time*: é o tempo gasto para o sinal acomodar na faixa de ±2% a ±5% porcento do valor final.

3. Tempo de subida (t_r, do inglês *rise time*): é o tempo necessário para a sinal de saída variar de 10% a 90% (sistemas sobre-amortecidos) ou de 0% a 100% (sistemas subamortecidos) do valor final.

Graficamente podemos observar todas as características de desempenho por meio de uma curva de resposta em degrau unitário como mostra a Figura 8 (OGATA, 2010; SEBORG et al., 2004).

Figura 8 – Curva de resposta em degrau unitário que mostra as características de desempenho no domínio do tempo.

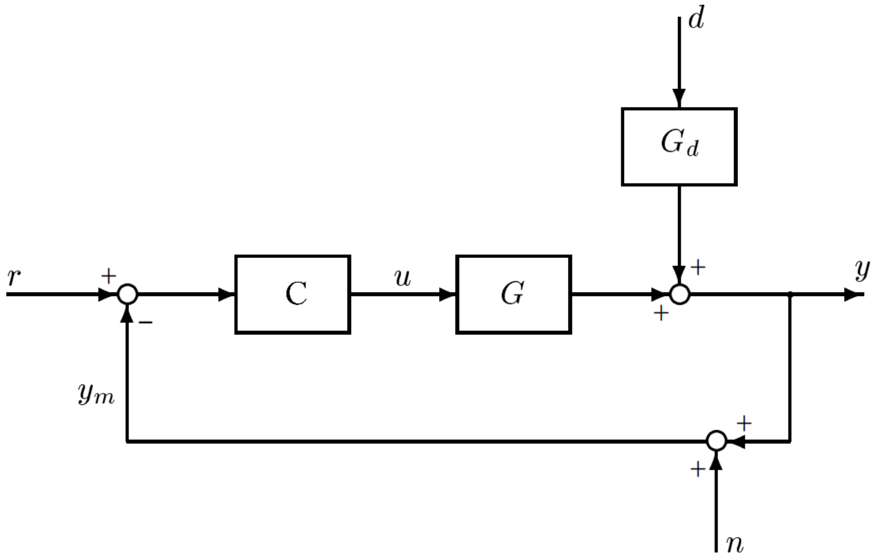


Retirada das notas de aula do Professor Reinaldo M. Palhares - UFMG.

3.2 Desempenho no domínio da frequência

Além da análise no domínio do tempo, pode-se utilizar técnicas de resposta em frequência para executar a análise no domínio da frequência. As técnicas vem sendo utilizadas por engenheiros de controle há décadas e se provaram indispensáveis quando se deseja analisar os benefícios, limitações e problemas em sistemas de controle. Uma vez que substituímos s por $j\omega$ em G obtemos então a chamada descrição da resposta em frequência, ou seja, $G(j\omega)$ é a resposta em frequência de $G(s)$ e depende diretamente da frequência (SKOGESTAD; POSTLETHWAITE, 2005).

Figura 9 – Diagrama de blocos de um sistema de controle de um grau de liberdade. Onde ym = y+n é o sinal de saída somado ao ruído.



Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

Considerando o modelo em diagrama de blocos apresentado pela Figura 9, podemos escrever o modelo da planta como

$$y = G(s)u + G_d(s)d \tag{3.2}$$

e assim, u pode ser descrito conforme

$$u = C(s)(r - y - n). \tag{3.3}$$

Substituindo (3.3) em (3.2) obtemos a resposta em malha fechada descrita por

$$y = (1 + GC)^{-1}GCr + (1 + GC)^{-1}G_d d - (1 + GC)^{-1}GCn. \tag{3.4}$$

A partir de (3.4) podemos extrair dois itens essenciais para a análise no domínio da frequência. Esses dois itens são a função de sensibilidade (S), dada por

$$S = (1 + GC)^{-1}, \tag{3.5}$$

que é a MF a partir do sinal de referência para a saída total do sistema e a função de sensibilidade complementar (T), dada por

$$T = (1 + GC)^{-1}GC. \tag{3.6}$$

que é a função transferência em malha fechada (MF) a partir da saída da pertubação para a saída total do sistema. Ambas podem ser definidas como (SKOGESTAD; POSTLETHWAITE, 2005):

A partir da análise dos termos S e T, podemos determinar a identidade definida como em (SKOGESTAD; POSTLETHWAITE, 2005)

$$S + T = 1. \tag{3.7}$$

Uma vez que a resposta em frequência e suas respectivas funções para análise são caracterizadas, podemos utilizar esse conceito para caracterizar o desempenho de uma ou várias funções de transferência em malha fechada (SKOGESTAD; POSTLETHWAITE, 2005).

A seguir apresentaremos alguns métodos utilizados para descrever desempenho no domínio da frequência. São eles: margem de ganho e fase, o pico máximo de S e T, e largura de banda.

1. Margem de ganho e fase: Considere o caminho direto do sistema em malha aberta (L_p) sobre um $feedback$ negativo, a margem de ganho (GM) é definida por

   $$GM = \frac{1}{|L_p(j\omega_{180})|}. \tag{3.8}$$

   Em que a frequência de cruzamento de fase (180°) representa o local em que em um gráfico de Nyquist (técnica gráfica utilizada para determinar a estabilidade de um sistema dinâmico), a curva cruza o eixo real entre os valores -1 e 0. Assim, o GM nos diz o quanto podemos acrescentar ao ganho do sistema operando em malha fechada antes que se torne instável (SKOGESTAD; POSTLETHWAITE, 2005).

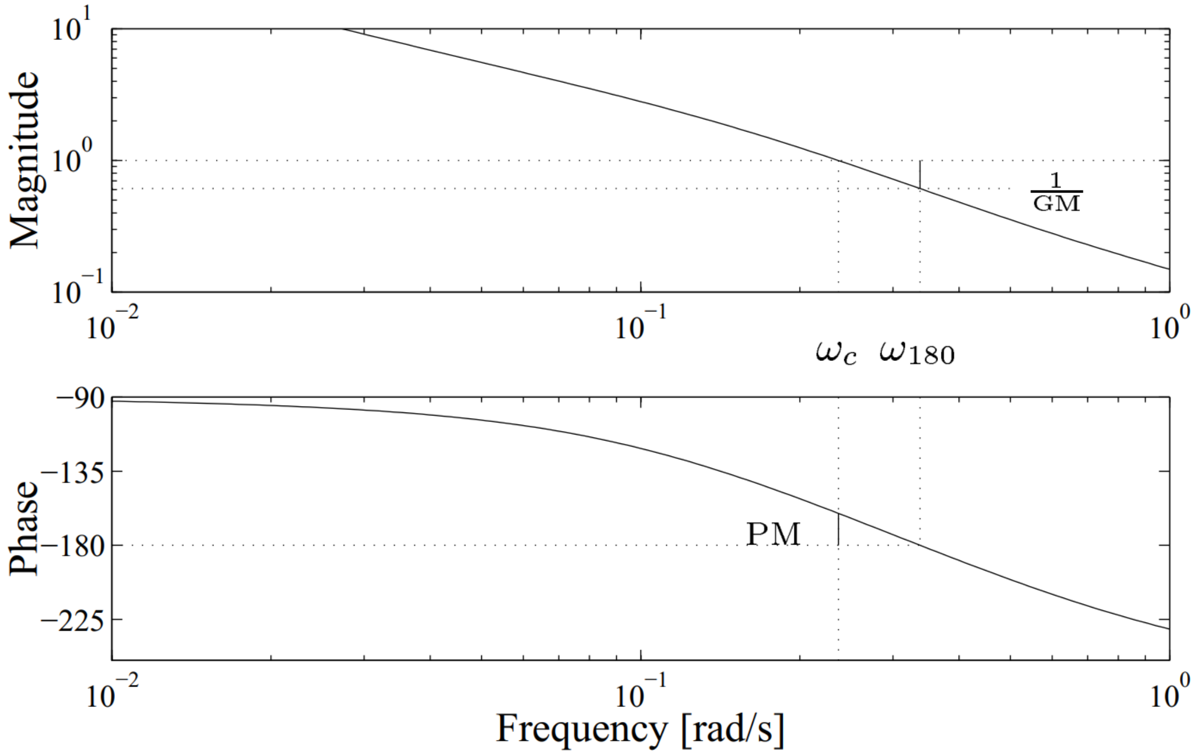
   A margem de fase (PM) é definida como

   $$PM = \angle L_p(j\omega_c) + 180°. \tag{3.9}$$

   Onde a frequência de cruzamento de ganho representa o local onde em um gráfico de Nyquist, $\angle M$F cruza pela primeira vez o valor 1. Assim, PM nos diz o quanto de fase pode ser adicionado ao sistema operando em malha fechada na mesma frequência antes de se tornar instável, ou seja, antes da fase se tornar -180°.

   A Figura (10) ilustra ambas as margens de ganho e fase em um típico diagrama de Bode (SKOGESTAD; POSTLETHWAITE, 2005).

Figura 10 – Diagrama de Bode ilustrando a margem de ganho e fase.



Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

2. O pico máximo de S e T: Os picos máximos das funções de sensibilidade (M_S) e sensibilidade complementar (M_T) são definidas como se segue (SKOGESTAD; POSTLETHWAITE, 2005):

$$M_S = max|S(j\omega)| \tag{3.10}$$

$$M_T = max|T(j\omega)|. \tag{3.11}$$

Os valores de M_S e M_T são indicadores tanto de desempenho quanto de robustez uma vez que valores altos de ambos os picos indicam um fraco desempenho e não robustez em qualquer frequência. Como temos a identidade representada em (3.7) aplicada a S e T, e como as equações que representam os picos máximos são em função dessas variáveis, a mesma identidade se aplica aos picos máximos. Logo para questões de robustez, estabilidade e desempenho precisamos que o valor de M_S seja o mais próximo possível do valor 1. (SKOGESTAD; POSTLETHWAITE, 2005).

Existe uma relação bastante interessante entre aspectos no domínio do tempo com outros no domínio da frequência. A principal relação está entre os picos máximos no domínio da frequência e o M_p ou a variação total, ambos no domínio do tempo.

Podemos representar essa relação por

$$M_T \leq variação total \leq (2n+1)M_T \tag{3.12}$$

e a Figura 11 apresenta as relações entre os valores nos dois domínios (SKOGESTAD; POSTLETHWAITE, 2005).

Figura 11 – Relação entre os valores de pico, variação total e *overshoot* para um sistema de segunda ordem

	Time domain		Frequency domain	
ζ	Overshoot	Total variation	M_T	M_S
2.0	1	1	1	1.05
1.5	1	1	1	1.08
1.0	1	1	1	1.15
0.8	1.02	1.03	1	1.22
0.6	1.09	1.21	1.04	1.35
0.4	1.25	1.68	1.36	1.66
0.2	1.53	3.22	2.55	2.73
0.1	1.73	6.39	5.03	5.12
0.01	1.97	63.7	50.0	50.0

Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

3. Largura de banda: Em um sistema de controle precisamos considerar além do desempenho, a rapidez da resposta. Esse último nos leva a considerar a largura de banda do sistema. Um relação entre uma banda larga e uma banda estreita pode ser especificada com relação a questões que podem influenciar o comportamento do sistema (SKOGESTAD; POSTLETHWAITE, 2005).

   Uma banda larga corresponde a um rápido tempo de assentamento e também indica um sistema que é sensível a ruídos e variações de parâmetros. Por outro lado, uma banda estreita de faixa corresponde geralmente a um tempo de resposta maior e a um sistema menos sensível a ruídos (SKOGESTAD; POSTLETHWAITE, 2005).

Os valores máximos de pico descritos anteriormente podem ser descritos em termos da norma H_∞, em que $M_s = ||S||_\infty$ e $M_T = ||T||_\infty$, assim precisamos definir e caracterizar essa norma. A norma H_∞ de uma TF pode ser definida como o valor de pico dessa função no domínio da frequência, que é representada por (SKOGESTAD; POSTLETHWAITE, 2005)

$$||TF(s)||_\infty = max|TF(j\omega)|. \tag{3.13}$$

3.2.1 Sensibilidade ponderada

Para sistemas de uma única entrada e única saída (SISO), a função S pode ser utilizada como indicador de desempenho em malha fechada. Como estamos preocupados somente com sua magnitude, ou seja, $|S|$, pois idealmente a queremos pequena, não precisamos levar em consideração a sua fase.

Uma forma comum de se utilizar os valores de norma H_∞ para estudar o desempenho do sistema é por meio de uma função de transferência de peso, levando à chamada sensibilidade ponderada. A ideia nesse caso é que o valor $1/wp$ especifica um comportamento limite para o sistema, de forma que se

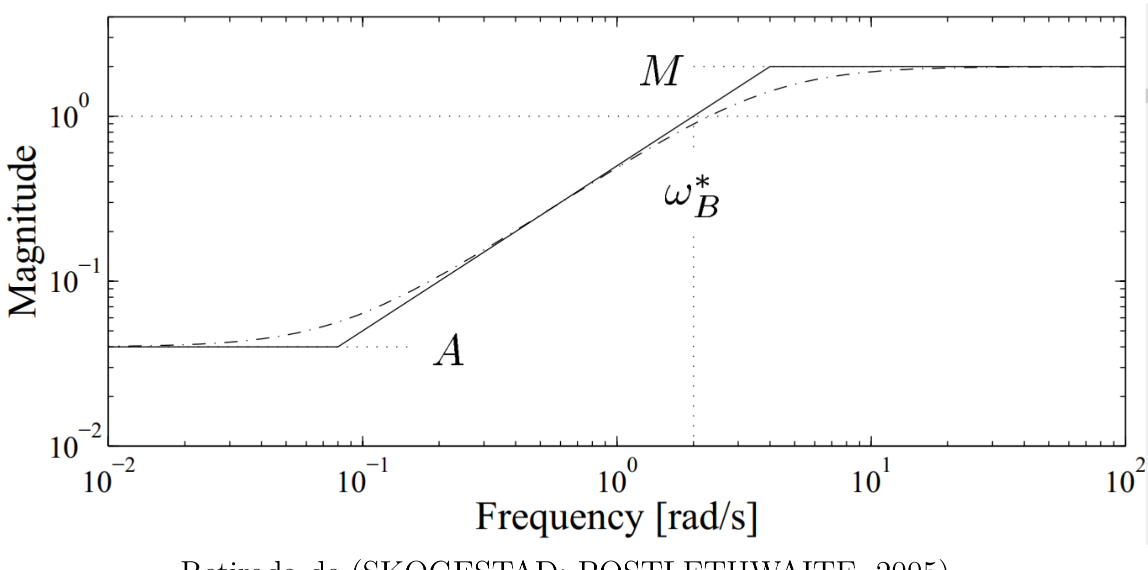
$$||\omega_p S||_\infty < 1 \tag{3.14}$$

for atendida para todo ω, podemos garantir que o desempenho do sistema é melhor ou igual ao especificado

Para selecionarmos esse peso utilizamos a assintota mostrada na Figura 12. Assim o peso pode ser representado por:

$$\omega_p(s) = \frac{s/M_s + \omega_b}{s + \omega_b A} \tag{3.15}$$

Figura 12 – Inverso do peso do desempenho.



Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

3.3 Estabilidade e desempenho robustos

Controladores devem ser capazes de operar independentemente das incertezas das plantas. Controladores desse tipo são denominados robustos. A seguir iremos abordar dois aspectos que são essenciais nesse contexto, a estabilidade e o desempenho robusto.

3.3.1 Sistemas incertos e suas representações

Uma vez trabalhando com sistemas robustos necessitamos de quantificar a incerteza dinâmica não modelada. Para tal, a abordagem do domínio da frequência é a solução utilizada neste trabalho (SKOGESTAD; POSTLETHWAITE, 2005).

A incerteza no modelo matemático de G pode se dar por vários aspectos, onde esses podem ser agrupados em duas classes como se segue (SKOGESTAD; POSTLETHWAITE, 2005):

1. Incerteza paramétrica: Nessa classe, a estrutura do modelo, incluindo sua ordem, são conhecidas, porém, alguns parâmetros são incertos. Esse tipo de incerteza pode ser quantificada se assumirmos que cada parâmetro incerto estará dentro de uma região limite $[\alpha_{max}, \alpha_{min}]$. Assim, podemos descrever a incerteza em função da região limite como:

$$\alpha = \overline{\alpha}(1 + b_\alpha \triangle). \tag{3.16}$$

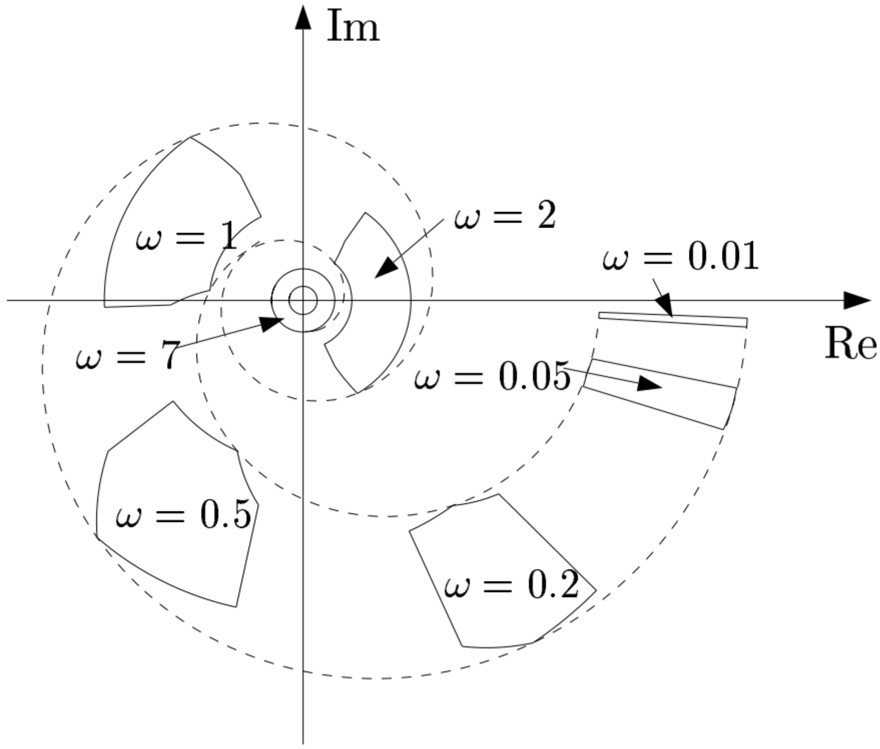
   Em que, $\overline{\alpha}$ é o valor médio, $b_\alpha = (\alpha_{max} - \alpha_{min}/(\alpha_{max} - \alpha_{min})$ representa a relativa incerteza no parâmetro e $\triangle$ nesse contexto é qualquer constante real desde que $|\triangle| \leq 1$.

2. Incerteza dinâmica não modelada: Nessa classe o modelo está em estado de erro pela falta de dinâmica, seja por falta de entendimento em representar seu processo físico ou por falta de dinâmica geralmente em altas frequências.

3.3.2 Representando regiões incertas através de perturbações complexas

Para demonstrar como parâmetros incertos são ilustrados no domínio da frequência, considere a Figura 13. Essa figura mostra uma região de números complexos gerada para cada variação dos parâmetros incertos da planta. Como podemos perceber, essas regiões incertas têm formas bastante complexas. Logo para fins de estudos e simplificação, aproximamos cada região por um círculo (SKOGESTAD; POSTLETHWAITE, 2005).

Figura 13 – Gráfico de Nyquist representando as regiões geradas a partir de uma planta incerta com parâmetros incertos.



Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

Essa aproximação resulta em uma incerteza aditiva, ou seja, adicionando perturbações complexas limitadas por normas sobre G. Esse processo pode ser descrito por(SKOGESTAD; POSTLETHWAITE, 2005)

$$G_p(s) = G(s) + \omega_A(s)\triangle_A(s). \tag{3.17}$$

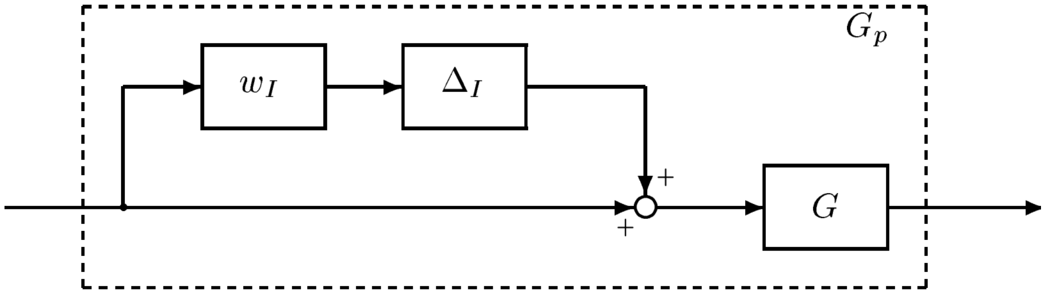
Em que, $\triangle_A(s)$ é qualquer TF desde que, seja estável e não maior ou igual a 1 em magnitude, ou seja, $|\triangle_A(j\omega)| \leq 1 \forall \omega$ e ω_A que representa pesos aditivos à incerteza, é na maioria da vezes uma TF racional, que geralmente é estável e com fase mínima.

Uma forma alternativa de se representar pesos incertos é por meio de sua forma multiplicativa. Essa forma geralmente é preferida porque seu valor numérico é mais informativo. Essa forma alternativa pode ser demostrada pela Figura 14 representada por

$$G_p(s) = G(s)(1 + \omega_I(s)\triangle_I(s)) \tag{3.18}$$

(SKOGESTAD; POSTLETHWAITE, 2005). Utilizaremos esse tipo de peso para as incertezas nesta monografia.

Figura 14 – Planta com incerteza multiplicativa.



Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

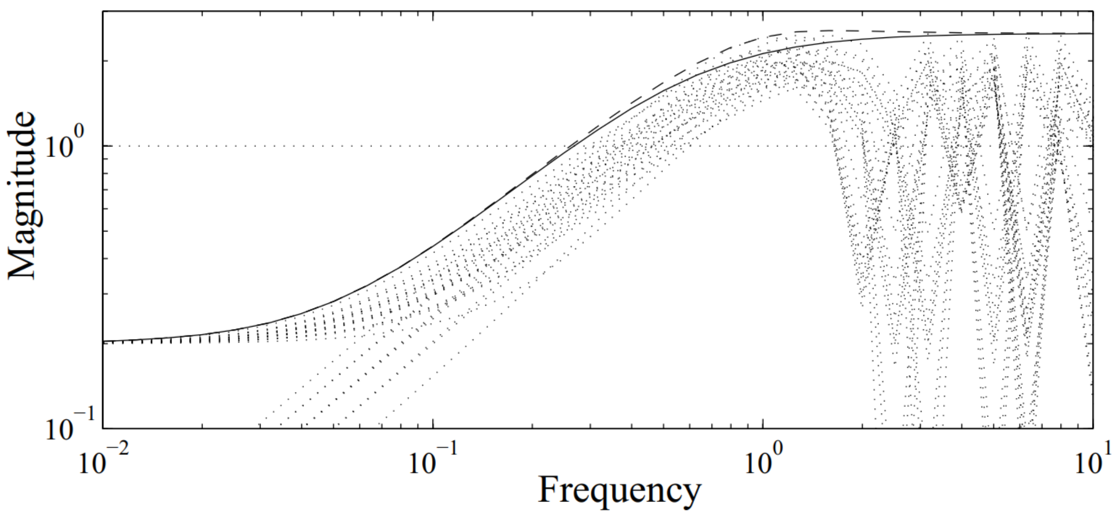
Em que, $\triangle_I(s)$ é qualquer TF desde que, a mesma seja estável e não maior ou igual a 1 em magnitude, ou seja, $|\triangle_I(j\omega)| \leq 1 \forall\omega$ e ω_I representa pesos multiplicativos à incerteza, é na maioria da vezes uma TF racional, que geralmente é estável e com fase mínima.

Considerando um conjunto de plantas, queremos representar esse peso como uma perturbação complexa. Para a incerteza multiplicativa temos sua forma conforme descrita por

$$|\omega_I(j\omega)| \geq max. \left(|\frac{G_p(j\omega) - G(j\omega)}{G(j\omega)}| \right). \tag{3.19}$$

Assim, precisamos que o valor do peso fique acima de todos os modelos candidatos do sistema em toda faixa de frequência. Esse processo pode ser exemplificado pela Figura 15, na qual temos o peso multiplicativo (ω_I) representado pela linha tracejada.

Figura 15 – Representação de um peso multiplicativo sobre uma família de plantas.



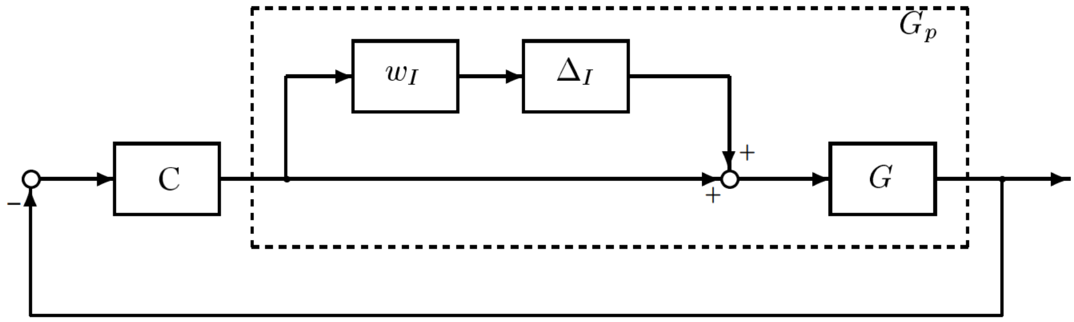
Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

3.3.3 Condições de estabilidade e desempenho robustos

Uma vez que o sistema satisfaz as especificações de desempenho tais como, o rastreamento de estado em regime, rejeição de perturbação, e exigências de velocidade de resposta, e além disso, o controlador satisfaz a todas as exigências para todos os valores na faixa de frequências especificada, podemos dizer que o sistema possui desempenho robusto (SKOGESTAD; POSTLETHWAITE, 2005).

Precisamos derivar algumas condições para assegurarmos que o sistema permanece estável mediante a todas as perturbações dentro do conjunto de incertezas e ao mesmo tempo garantir que tenha um desempenho dentro das especificações. Considere o sistema em malha fechada especificado pela Figura 16 (SKOGESTAD; POSTLETHWAITE, 2005).

Figura 16 – Sistema em MF com incerteza multiplicativa.



Retirada de (SKOGESTAD; POSTLETHWAITE, 2005).

Estamos preocupados nesta subseção com a estabilidade e desempenho robustos, logo para questões de simplificações iremos considerar que o sistema é estável em malha aberta. Podemos então utilizar o critério de estabilidade de Nyquist para testar a estabilidade robusta do sistema (SKOGESTAD; POSTLETHWAITE, 2005).

Uma vez o gráfico de Niquist de uma L_p traçado, iremos encontrar a distância entre o ponto -1 ao centro do disco representado pela L_p. Podemos extrair desse gráfico que essa distância será igual a $|1 + L_p|$ e o raio desse disco será $\omega_I L_p$. Assim podemos encontrar o critério de estabilidade robusta para esse tipo de sistema baseado nessas informações e as derivando como segue (SKOGESTAD; POSTLETHWAITE, 2005):

$$|\omega_I L_p| < |1 + L_p|, \qquad \forall\omega \tag{3.20}$$

$$|\frac{\omega_I L_p}{1 + L_p}| < 1, \qquad \forall\omega \tag{3.21}$$

$$|\omega_I T| < 1, \qquad \forall\omega \tag{3.22}$$

$$||\omega_I T||\infty < 1. \tag{3.23}$$

A partir de (3.23) podemos perceber que para uma L_p ser estável quanto a robustez, a norma H_∞ precisa ser menor do que 1.

De forma semelhante, pode-se testar o desempenho robusto de um sistema pela desigualdade descrita por

$$max(|\omega_p S| + |\omega_I T|) < 1. \tag{3.24}$$

Maiores detalhes sobre como derivar tal condição podem ser vistos em (BOYD et al., 2010).

4 Métodos de otimização

As meta-heurísticas são métodos heurísticos que direcionam outras heurísticas. Assim o espaço de soluções para uma busca vai muito além do ótimo local explorando novas regiões promissoras. São utilizadas combinações de escolhas aleatórias e o conhecimento histórico dos resultados para se guiarem e realizarem suas buscas pelo espaço de busca.

Dentre as várias categorias de métodos de otimização existentes, iremos focar neste trabalho a que abrange os métodos baseados na otimização natural. Essa categoria se assemelha com as demais quanto a abordagem básica da posição partindo de um ponto arbitrário, porém, diferem na direção e distância dos movimentos (HAUPT; HAUPT, 2004).

Além das meta-heurísticas não podemos deixar de falar da programação semidefinida. Este método de otimização é bastante aplicado a teoria de controle, uma vez que, em algumas situações podemos reduzir alguns problemas complexos à problemas de otimização padrão convexos, que envolvem desigualdades matriciais (BOYD et al., 2010).

O principal ponto da programação semidefinida está em sua solução numérica aplica para a resolução de problemas que podem ser resolvidos em tempo polinomial e são tratáveis (BOYD et al., 2010).

4.1 Métodos de otimização natural

Os métodos de otimização natural aplicam operadores em seus pontos atuais, gerando assim novos pontos em seu espaço de busca. A partir desses novos pontos o método utiliza estatística para se movimentar nesse espaço. Estão embutidos nessa categoria, métodos estatísticos e um método de busca inteligente (HAUPT; HAUPT, 2004).

4.1.1 Padrão comum

Analisando de uma forma geral, os métodos de otimização da categoria de otimização natural seguem um padrão quando modelam sua dinâmica de operação.

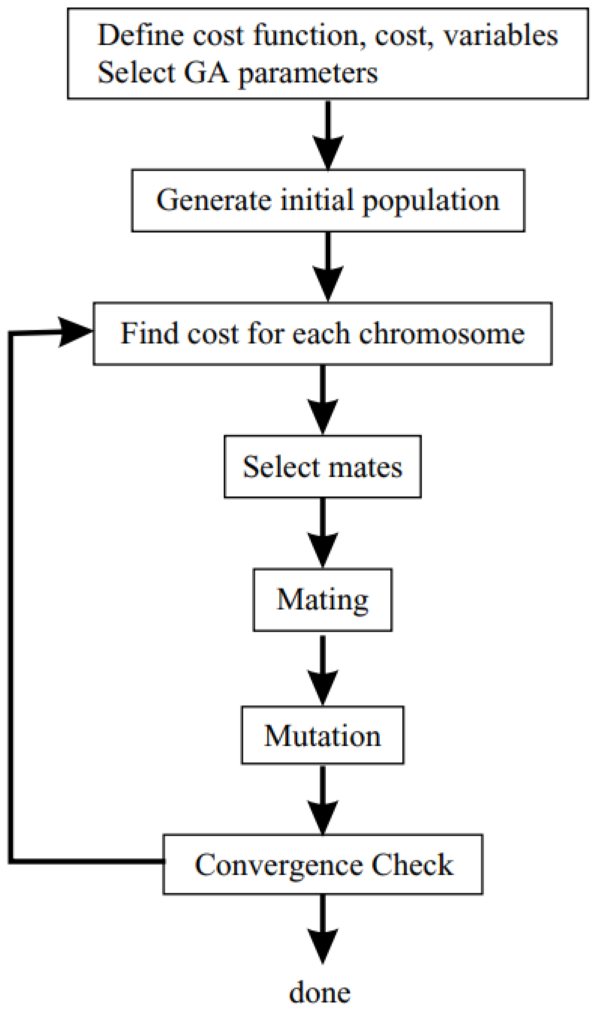
Algoritmos baseados na otimização natural produzem uma população inicial, composta por diversos indivíduos que será evoluída para um estado que minimizará o custo de uma determinada função, sobre regras de seleção específicas. Essa população será de um tamanho constante durante topo o processo (HAUPT; HAUPT, 2004; JONG, 2006).

A partir da população inicial, novos indivíduos são gerados e então passamos a ter um problema, pois a população deve ser constante. Logo passamos para a segunda parte da seleção natural, a evolução. Utilizando uma heurística elitista, o algoritmo mantém sempre o melhor indivíduo atual vivo, considerando gerações diferentes. Alguns métodos pertencentes a essa categoria ainda utilizam a mutação, que é uma forma de reprodução assexuada para gerar indivíduos após as seleções aplicadas (HAUPT; HAUPT, 2004; JONG, 2006).

4.1.2 Algoritmo Genético (GA)

Iremos iniciar essa explicação apresentando um fluxograma que exibe cada fase do *GA*. A Figura 17 representa esse fluxograma e cada parte desse será explicada posteriormente (HAUPT; HAUPT, 2004).

Figura 17 – Fluxo do processo do *GA*.



Retirada de (HAUPT; HAUPT, 2004).

Inicialmente precisamos definir a função de custo e alguns parâmetros de seleção. Essas informações serão úteis para calcular o custo de cada cromossomo (indivíduo) em cada iteração, bem como para definir parâmetros de seleção (HAUPT; HAUPT, 2004).

Com todos os parâmetros selecionados, passamos para a fase de geração aleatória da população inicial. Uma vez a população inicial gerada, passamos a mesma pela função de custo, que irá calcular o custo de cada cromossomo particularmente (HAUPT; HAUPT, 2004).

Com o custo de cada indivíduo nas mãos, passamos para a próxima fase do algoritmo, a seleção natural. Como o próprio nome diz, a seleção natural irá selecionar quais cromossomos passam de uma geração para outra. Inicialmente ela é aplicada à população inicial para definir quais cromossomos estão aptos a gerarem "filhos". A seleção natural utiliza o custo de cada cromossomo como um ranking e assim sobrevive aqueles com o menor custo (HAUPT; HAUPT, 2004; JONG, 2006).

Passando para a fase seguinte, temos a geração de "filhos". Dois indivíduos são selecionados e um terceiro indivíduo é gerado a partir de alguma combinação entre eles. Esse processo é chamado de *crossover* e existem vários métodos de crossover que podem ser aplicados no *GA*. Adewauya (1996) e Michalewicz(1994) propuseram alguns interessantes.

Uma vez os filhos gerados e a seleção natural aplicada, teremos o mesmo número de indivíduos tão quanto a população inicial. Então podemos passar para uma fase bastante importante do algoritmo, a mutação. Essa fase é importante pois força o algoritmo a vasculhar toda a superfície de custo. Uma vez executando essa procura, o algoritmo corre menos risco de ficar preso em um mínimo local. Geralmente algoritmos desse tipo possuem uma taxa de mutação que gira em torno de 20% (HAUPT; HAUPT, 2004).

A última fase se baseia na convergência do algoritmo, ou seja, se o algoritmo encontrou o mínimo global durante as gerações. Isso poderá ser avaliado utilizando-se a função de custo e os parâmetros de aceitação da função definidos no início da aplicação do método(HAUPT; HAUPT, 2004; JONG, 2006).

4.1.3 Otimização por enxame de partículas (PSO)

Formulado por Edward e Kennedy em 1995 e inspirado no comportamento dos animais, assim como todos os outros algoritmos da categoria de otimização natural, o *PSO* é bastante similar ao *GA* no início do seu processo ao gerar sua população inicial aleatoriamente. Eles se diferem na parte de seleção e movimentação, uma vez que o *PSO* não utiliza operadores evolucionários (HAUPT; HAUPT, 2004).

Similar aos cromossomos do *GA*, no *PSO* os indivíduos são conhecidos como partículas. Cada uma dessas partículas contém dentro da superfície de custo, uma posição (X) dada por

$$X_{t+1} = X(t) + V(t) \tag{4.1}$$

e uma velocidade (V), dada por

$$V_{t+1} = inV(i) + \phi_1 a_1(L_o(i) - X(i)) + \phi_2 a_2(G_l(i) - X(i)). \tag{4.2}$$

Em que, in a constante de inércia, ϕ_1 e ϕ_2 constantes de aprendizagem, a um número aleatório uniforme independente e i a i-ésima posição. Cada uma utiliza como base para atualizar tanto sua posição atual quanto sua velocidade, a melhor solução local (L_o) e a melhor solução global (G_l) (HAUPT; HAUPT, 2004).

Analisando a equação de atualização da velocidade, equação (4.2), podemos perceber que esta depende tanto da melhor solução global até o presente momento, quando da melhor solução local encontrada pela atual geração. Uma vez que a solução local ultrapasse a global com respeito ao menor custo, a global é substituída pela local. Já a equação de atualização do posicionamento depende somente da melhor solução local de forma indireta (HAUPT; HAUPT, 2004).

O PSO gera uma população inicial e então a cada iteração após a atualização da posição e velocidade de cada partícula, encontramos a partícula mais próxima do mínimo global. Uma vez encontrada, direcionamos o restante das partículas naquela posição (HAUPT; HAUPT, 2004; JONG, 2006).

4.2 Programação semidefinida

Alguns conceitos utilizados nessa abordagem precisam ser definidos de antemão para o melhor entendimento do assunto.

Uma matriz pode ser definida pela sua forma quadrática associada. Assim, uma matriz Q é definida positiva se a sua forma quadrática é definida positiva. Matematicamente podemos representar essa relação por (BROGAN, 1991)

$$x^T Q x > 0, \qquad \forall x \neq 0. \tag{4.3}$$

Uma matriz Q é semidefinida positiva se a sua forma quadrática é semidefinida positiva. Matematicamente podemos representar essa relação por (BROGAN, 1991)

$$x^T Q x \geq 0, \qquad \forall x \neq 0. \tag{4.4}$$

As representação de uma matriz definida negativa e semidefinida negativa seguem os mesmos conceitos abordados acima, porém voltados para a parte negativa.

Sempre que citamos o termo sinal de uma matriz, estamos considerando apenas matrizes simétricas pois, uma matriz pode ser decomposta como uma parte simétrica e uma parte assimétrica. Porém, matrizes assimétricas têm a propriedade de suas formas quadráticas serem sempre zero, logo essa parte não contribui para a parte quadrática. Assim, uma vez que estamos definindo o sinal de uma matriz, estamos nos referindo somente à sua parte simétrica.

A programação semidefinida tem como objetivo encontrar, dentre todos os pontos nos quais a função de restrição for semidefinida negativa, aquele que minimize a função objetivo (LING, 2001; MASSERA, 2010).

4.2.1 Desigualdades matriciais lineares (LMI)

Uma das ferramentas amplamente utilizadas na teoria de controle são as desigualdades matriciais lineares - do inglês *Linear Matrix Inequalities* ou LMIs. Essas desigualdades surgiram com os trabalhos de Lyapunov há mais de cem anos, porém com poucas soluções numéricas. Recentemente com a criação de alguns algoritmos que exploram a convexidade dos problemas LMIs, podemos obter resultados numéricos mais confiáveis (SKOGESTAD; POSTLETHWAITE, 2005; LING, 2001).

Na maioria das aplicações de controle, LMIs surgem como funções de variáveis matriciais. Essas funções podem ser definidas da forma

$$R + M_1^T V_1 N_1 + N_1^T V_1 M_1 + ... + M_n^T V_n N_n + N_n^T P_n M_n < 0. \tag{4.5}$$

Em que M_i, N_i e R, i=1,2,...,n, são matrizes fixas, e V_i são variáveis matriciais. O sinal de desigualdade significa definida negativa (vide Seção 4.2) (SCHERER; SIEPWEILAND, 2015).

As LMIs podem ser aplicadas em diversas situações para resolver problemas que envolvem muitas variáveis matriciais e diversas estruturas podem ser impostas a essas variáveis. Além dessas vantagens, esse método é totalmente flexível, podendo assim, uma grande gama de problemas serem diretamente transformados em problemas LMIs (SKOGESTAD; POSTLETHWAITE, 2005).

Precisamos estar atentos à aplicação de LMIs quando necessitamos que o sistema de controle tenha desempenho robusto, pois em projetos tradicionais utilizando LMIs, esse quesito não é abordado de forma satisfatória. Uma forma de obter um melhor desempenho do sistema é através da localização dos seus polos. Porém, uma vez que estamos lidando com sistemas incertos, seus polos podem não permanecer no local pré especificado. Regiões abordadas em problemas LMIs, geralmente são discos, cones ou até mesmo a parte esquerda do plano S.

5 Metodologia

A sintonia de controladores pode ser alcançada identificando e otimizando seus parâmetros ajustáveis. Assim tanto a programação semidefinida quanto as meta-heurísticas devem sem capazes de identificar esses parâmetros tanto para modelos certos quanto para uma família de modelos incertos. A seguir são apresentados os métodos e procedimentos adotados para identificar tais parâmetros.

Objetivando uma abordagem simplificada para a obtenção das constantes utilizadas pelas meta-heurísticas, foi utilizado o modelo matemático de uma planta exata de um elevador fornecido por alunos que desenvolveram um trabalho para a disciplina de Controle II. O modelo matemático da planta, já aplicada a transformada de Laplace é dada por

$$G = \frac{48}{0.84s + 1}. \tag{5.1}$$

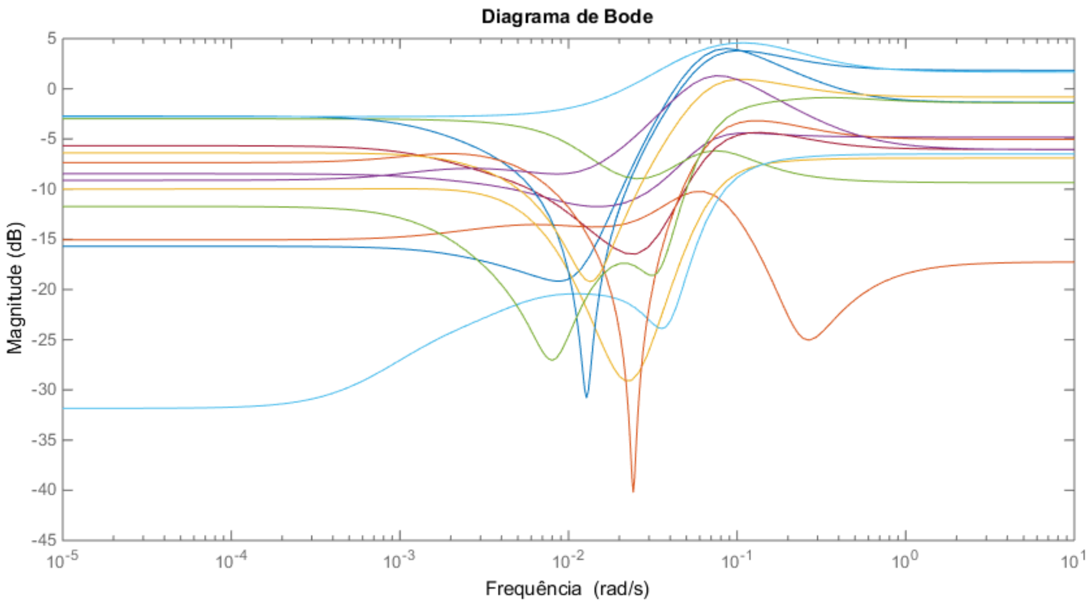
Como ferramenta para análise foi utilizado um sistema que consiste de uma câmara termoeletricamente controlada por meio de módulos Peltier TEC 12715 cuja temperatura é medida por meio de um sensor digital DS18B20. Considera-se inicialmente que o comportamento do sistema é descrito por 12 funções de transferência diferentes obtidas via métodos de identificação de sistemas (PEREIRA et al., 2018). A Figura 18 mostra o sistema acima citado.

Figura 18 – Câmara termoeletricamente controlada.

Elaborada pelo próprio autor.

Para uso na metodologia, considerou-se um modelo médio gerado a partir da média de todos os modelos e uma incerteza multiplicativa. A TF média utilizada é de uma ordem muito elevada, logo não será apresentada. De forma a se obter o peso que representa a incerteza multiplicativa, utilizou-se a abordagem apresentada em (SKOGESTAD; POSTLETHWAITE, 2005) explicada na subseção 3.3.2 e aproximou-se de forma gráfica seu diagrama de bode de magnitude conforme Figura 19.

Figura 19 – Ajuste da incerteza multiplicativa. A curva de magnitude que está sobre todas as outras representa a TF. que foi ajustada



Elaborada pelo próprio autor.

Como ferramenta para o cálculo do custo, foram implementados dois métodos, a Integral do erro ao quadrado - do inglês *Integral of squared error* (ISE) e a Integral do erro absoluto ponderado no tempo - do inglês *Integral of time times absolute error* (ITAE), a partir da resposta do sistema em malha fechada a um degrau unitário. Porém, por questões de melhor compatibilidade com o problema em questão e já pensando em algumas possibilidades futuras, somente o ISE foi utilizado. Esse método pode ser descrito por

$$ISE = \int_0^t e^2(\tau)d\tau. \tag{5.2}$$

Por fim, precisamos citar as penalidades e restrições impostas aos sistemas, tanto o simplificado quanto o robusto.

Foram impostas ao sistema (5.1) cinco penalidades utilizadas para o caso sem incertezas, quando os algoritmos estavam processando com a finalidade de determinar as constantes utilizadas pelas meta-heurísticas. Todas estas estão no domínio do tempo e tomam as seguintes formas:

A primeira penalidade é aplicada quando o sistema se encontra instável em algum momento. Uma vez que o sistema se encontre nesse estado, um custo bastante elevado é adicionado ao custo previamente calculado utilizando o ISE

$$custo = custo + \begin{cases} 0, & \text{se} \quad max(polos) < 0 \\ 100000, & \text{se} \quad max(polos) \geq 0. \end{cases}$$

A segunda penalidade é aplicada sobre o máximo esforço de controle. Uma vez que essa variável ultrapasse o valor 10, essa penalidade é aplicada, ou seja, um valor de 100 vezes o máximo esforço de controle é adicionado ao custo previamente calculado utilizando o ISE

$$custo = custo + \begin{cases} 0, & \text{se} \quad max(|u|) \leq 10 \\ 100.max(|u|), & \text{se} \quad max(|u|) > 10. \end{cases}$$

A terceira penalidade é aplicada sobre o sobressinal ($overshoot$). Uma vez que essa variável ultrapasse o valor 10, essa penalidade é aplicada, ou seja, um valor de 100 vezes o valor do $overshoot$ é adicionado ao custo previamente calculado utilizando o ISE

$$custo = custo + \begin{cases} 0, & \text{se} \quad overshoot \leq 10 \\ 100.overshoot, & \text{se} \quad overshoot > 10. \end{cases}$$

A quarta penalidade é aplicada sobre o tempo de assentamento ($settlingTime$). Uma vez que essa variável ultrapasse o valor 5, essa penalidade é aplicada, ou seja, um valor de 100 vezes o valor do $settlingTime$ é adicionado ao custo previamente calculado utilizando o ISE

$$custo = custo + \begin{cases} 0, & \text{se} \quad settlingTime \leq 10 \\ 100.settlingTime, & \text{se} \quad settlingTime > 10. \end{cases}$$

A quinta e última penalidade é aplicada sobre o tempo de subida (riseTime). Uma vez que essa variável ultrapasse o valor 1, essa penalidade é aplicada, ou seja, um valor de 100 vezes o valor do riseTime é adicionado ao custo previamente calculado utilizando o ISE

$$custo = custo + \begin{cases} 0, & \text{se} \quad riseTime \leq 10 \\ 100.riseTime, & \text{se} \quad riseTime > 10. \end{cases}$$

Uma vez que as constantes utilizadas pelas meta-heurísticas foram definidas, as cinco penalidades acima citadas foram substituídas por novas restrições e penalidades, aplicadas agora ao sistema robusto utilizado como ferramenta de análise (câmara termoeletricamente controlada). Foram impostas duas penalidades, a primeira com relação a estabilidade e desempenho robustos do sistema, e a segunda para evitar a utilização de uma ação de controle muito elevada. Ambas foram implementadas no domínio da frequência e tomam as seguintes formas:

A primeira penalidade é aplicada para garantir tanto a estabilidade quanto o desempenho robusto do sistema. Para tal, necessitamos que o valor máximo da norma H_∞ seja menor que o valor 1. Para isso precisamos calcular a soma entre as multiplicações de $\omega_p S$ e $\omega_I T$ e logo após calcular o valor máximo de sua H_∞ como descrito por

$$H_\infty = max(\|\omega_p S + \omega_I T||_\infty) < 1. \tag{5.3}$$

Uma vez que essa variável seja igual ou ultrapasse o valor 1, essa penalidade é aplicada, ou seja, um valor de 100000 vezes o valor de H_∞ é adicionado ao custo previamente calculado utilizando o ISE.

$$custo = custo + \begin{cases} 0, & se \quad H_\infty < 1 \\ 100000(H_\infty - 1), & se \quad H_\infty \geq 1. \end{cases}$$

A segunda penalidade é imposta de forma a evitar que se utilize um controlador com uma ação de controle muito elevada. Essa penalidade é calculada a partir do esforço máximo de controle obtido na resposta do sistema em malha fechada a um degrau unitário. Essa penalidade pode ser calculada como:

$$custo = custo + \begin{cases} 0, & se \quad |u| \leq 30 \\ 100|u|, & se \quad |u| > 30. \end{cases}$$

5.1 Sintonia PID por meta-heurísticas

Como mencionado anteriormente, existem vários métodos para a sintonia de controladores PID. Métodos empíricos como o desenvolvido por Ziegler-Nichols em 1942, métodos modernos baseados em técnicas de inteligência artificial, tais como redes neurais, logica fuzzy e neuro-fuzzy. Além destes, alguns métodos que levam a computação evolucionária e meta-heurísticas como base também podem ser aplicados na resolução desse tipo de problema. Assim, a seguir serão apresentados duas meta-heurísticas (PSO e AG) aplicadas a sintonia de controladores PID (STIMAC; BRAUT; ZIGULIC, 2014).

5.1.1 PSO aplicado a sintonia PID

Como estamos utilizando neste trabalho o algoritmo PSO na sua versão linear, então precisamos inicialmente definir os parâmetros constantes in, ϕ_1 que é conhecido nesse âmbito como parâmetro de aprendizagem cognitivo e ϕ_2 que é conhecido nesse ambiente como parâmetro de aprendizagem social. Além disso precisamos definir, o número de partículas que serão utilizadas, o número de iterações e o tipo de custo.

Não há um método específico para ajuste desses parâmetros, nem mesmo tabelas de consultas ou algum algoritmo que possa indicar quais parâmetros melhor se ajustariam. Para a solução desse problema inicial (ajuste das constantes) foi adotado o método de

Monte Carlo. Esse método pode ser descrito como um método de simulação estatística que utiliza sequências de números aleatórios para desenvolver simulações. Em outras palavras, é visto como um método numérico universal para resolver problemas por meio de amostragem aleatória. Baseado nesse método e utilizando a planta descrita pela equação (5.1), o número de partículas foi ajustado para 100, o número de iterações para 100, o tipo de custo para o ISE e essa configuração foi executada para todas as combinações possíveis de in, ϕ_1 e ϕ_2, dentro de um loop de 100 iterações. A cada combinação diferente dos parâmetros, o algoritmo foi programado para salvar uma matriz de custo associada a essa combinação.

Uma vez que todas as contantes foram definidas, o PSO foi reprogramado para resolver o problema da sintonia PID em questão. Para essa tarefa o número de partículas foi elevado de 100 para 1000, o número de iterações de 100 para 500 e o loop foi retirado. Além disso, a planta exata antes utilizada foi substituída pelo sistema incerto.

5.1.2 GA aplicado a sintonia PID

Similarmente ao PSO quanto a definição de suas constantes, no GA precisamos definir inicialmente o método de seleção que iremos utilizar para selecionar os "pais", ou seja, os indivíduos que estarão aptos a gerarem um terceiro cromossomo. Os métodos de seleção para os cromossomos pais que foram implementados são: seleção uniforme, roleta, torneio com dois elementos e torneiro com três elementos. Além disso, optamos por manter uma heurística elitista sobre os filhos de uma geração para a outra.

Outro parâmetro que precisamos definir antes da execução definitiva do algoritmo é o método que iremos utilizar para a seleção de sobrevivência, ou seja, precisamos manter um número constante de indivíduos tal qual o número da população inicial. Como serão gerados "filhos", precisamos selecionar os melhores indivíduos que irão compor a população. Os métodos de seleção de sobrevivência implementados foram os mesmos utilizados para a seleção dos cromossomos "pais".

Assim como no PSO, não existe um método específico para ajuste desses métodos, nem mesmo tabelas de consultas ou algum algoritmo que possa indicar quais métodos melhor se ajustariam. Como temos basicamente o mesmo cenário apresentado anteriormente, logo utilizamos novamente o método de Monte Carlo (este método é explicado na Seção 5.1.1).

Baseado no método de Monte Carlo, o número inicial da população foi ajustado para 100, o número de gerações para 100, o tipo de custo para o ISE e essa configuração foi executada para todas as possíveis combinações dos métodos de seleção para os pais e sobrevivência dentro de um loop de 100 iterações. A cada combinação diferente dos métodos, o algoritmo foi programado para salvar uma matriz de custo associada a essa combinação.

Uma vez que os métodos de seleção dos pais e os métodos de sobrevivência foram definidos, o GA pôde ser executado para resolver o problema da sintonia PID em questão. Para essa tarefa o número inicial da população foi elevado de 100 para 1000, o número de gerações foi elevado de 100 para 500 e o *loop* foi retirado. Além disso, a planta exata antes utilizada foi substituída pelo sistema incerto.

5.2 Sintonia por LMIs

Bem como as meta-heurísticas, a otimização por LMIs é uma ferramenta bastante poderosa e pode ser aplicada no auxilio ao problema de sintonia de controladores. A seguir será apresentada a metodologia utilizada para a aplicação dessa técnica.

Antes de realmente detalharmos essa parte da metodologia, precisamos de antemão deixar claro algumas considerações que serão utilizadas nessa seção.

Considerando um sistema linear em espaço de estados, uma forma de se analisar a estabilidade desse sistema é utilizando o Teorema de Estabilidade de *Lyapunov*. Assim como no caso não linear, basta que encontremos uma função que seja definida positiva e cuja derivada temporal seja definida negativa para provarmos a estabilidade assintótica do ponto de equilíbrio na origem. Para sistemas lineares, essa estabilidade assintótica implica em estabilidade BIBO.

Teorema de estabilidade de *Lyapunov*: Um sistema linear autônomo com a dinâmica $\dot{x} = Ax$ é assintoticamente estável se, e somente se, existe uma função quadrática definida positiva

$$V(x) = x^T Q x > 0, \qquad \forall x \neq 0, \tag{5.4}$$

com derivada temporal definida negativa

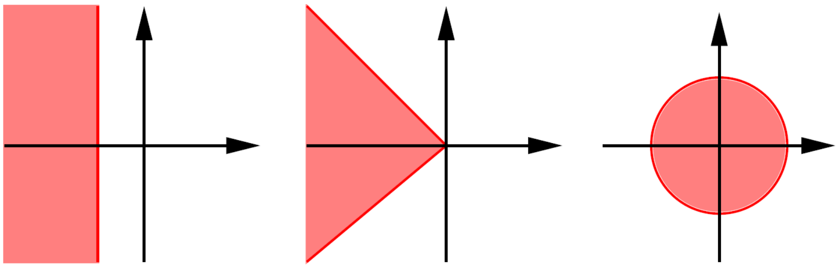
$$\dot{V}(x) = \dot{x}^T Q x + x^T Q x = x^T (A^T Q + QA) x < 0, \qquad \forall x \neq 0. \tag{5.5}$$

Vamos supor que todos os estados do sistema estão disponíveis para a lei de controle. Em um sistema linear invariante no tempo, a realimentação de estados possibilita alocar os polos do sistema em qualquer posição, desde que o sistema seja controlável (vide Seção 2.3).

No projeto tradicional por LMIs, o controlador ideal é projetado de tal forma que o sistema em malha fechada é assintoticamente estável, resultando em um sistema estável robusto que também é menos sensível ao distúrbio de carga externa. Outro problema no projeto tradicional por LMIs é a falta de percepção sobre a escolha do ganho superior. Para ser praticamente útil, sua ordem de grandeza deve ser suficientemente pequena(LING, 2001).

Se voltarmos a sessão 4.2.1 podemos verificar os requisitos para que o sistema tenha desempenho. A Figura 20 mostra as regiões mais abordadas em problemas LMIs. Para esse trabalho iremos definir duas regiões. A primeira será um cone e a segunda será um semiplano, e então essas duas regiões são combinadas para formar a região trabalhada.

Figura 20 – Regiões geralmente abordadas em problemas LMIs.



Adaptada de (HENRION, 2010).

Para o cone precisamos definir um ângulo θ que será utilizado para limitar a região abrangente. Esse ângulo é encontrado por meio do ζ escolhido para ser o ζ constante das retas que formam o cone. Foi escolhido um ζ=0,707 e assim o ângulo θ pode ser encontrado ($\zeta = \cos\,\theta$) e vale 0,785 radiandos. Uma vez definido o valor de θ, precisamos estabelecer duas variáveis que serão utilizadas para montar as LMIs. A primeira variável é chamada de D_0=0 e a segunda variável chamada de D_1 pode ser definida pela matriz abaixo (SCHERER; SIEPWEILAND, 2015; HENRION, 2010)

$$\mathbf{D_1} = \begin{pmatrix} sin(\theta) & cos(\theta) \\ -cos(\theta) & sin(\theta) \end{pmatrix}.$$

Para o semiplano precisamos definir um valor chamado α, utilizado para deslocar o semi plano a partir do zero em direção ao lado esquerdo do plano S, que será utilizado para localizar os polos no semiplano esquerdo do plano S. Esse valor α foi estabelecido com o valor de 0,0025 através de um diagnóstico que permite verificar se as LMIs tiveram soluções.

Como estamos lidando com um problema de rastreamento (quando a saída do sistema segue um sinal de referência), uma solução para esse tipo de problema que pode ser incorporado as LMIs, é a adição de um integrador ao controlador (sistema aumentado), logo teremos uma lei de controle por realimentação de estados e integral do erro. Essa relação pode ser representada por

$$u = k(i)x(i) + K_i \int (\tau) d\tau. \tag{5.6}$$

Em que, k representa o ganho de cada estado x do sistema.

Uma vez estabelecida a região de estabilidade e desempenho, bem como acrescentado o integrador, podemos definir o conjunto de LMIs seguindo o teorema de *Lyapunov* que representa esse problema. Para ambas regiões (cone e semiplano) precisamos de uma condição que é definida por

$$Q > 0. \tag{5.7}$$

Para o semiplano, as LMIs foram codificadas para que o conjunto seja menor do que $-2\alpha F$ (vide equação (5.5)). Para o cone precisamos utilizar o produto de Kronecker cujo resultado é (SCHERER; SIEPWEILAND, 2015)

$$D_0 \otimes Q + D_1 \otimes AQ + D_1^T \otimes QA^T < 0. \tag{5.8}$$

Para introduzir o conceito do produto de *Kronecker*, considere duas matrizes A(|m|x |n|) e B (|k| x |l|), o produto de *Kronecker* dessas duas matrizes é dado pela matriz (SCHERER; SIEPWEILAND, 2015)

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} A_{11}B & ... & A_{1n}B \\ \vdots & & \vdots \\ A_{m1}B & ... & A_{mn}B \end{pmatrix}.$$

Com todas as etapas acima citadas codificadas, o algoritmo foi programado e uma vez executado, as LMIs foram resolvidas e podemos obter os valores dos ganhos de cada estado associado ao sistema, bem como o valor de K_i.

6 Apresentação e análise dos resultados

Nesse capítulo são apresentados os resultados obtidos na resolução do problema em questão, tanto para a programação semidefinida quanto para o uso de meta-heurísticas. Logo após uma comparação entre as duas abordagem é realizada.

6.1 Sintonia PID por meta-heurísticas

Com toda a metodologia de meta-heurísticas, bem como todos os métodos adotados (PSO e GA) implementados, podemos agora apresentar os resultados obtidos para cada meta-heurística anteriormente mostrada.

6.1.1 PSO aplicado a sintonia PID

A primeira etapa de processamento diz respeito a definição das contantes como já dito anteriormente. O algoritmo foi programado de acordo com a Seção 5.1.1 para encontrar as constantes. Após mais de 90 horas de processamento, obtivemos todas as matrizes de custo associadas a cada diferente combinação das constantes. Com esses dados em mãos, uma inspeção foi executada. Essa inspeção se baseou na média e no desvio padrão da matriz de custo de cada configuração. Após uma análise sobre todas as matrizes, podemos chegar a melhor configuração para as constantes.

A próxima etapa, já com as constantes definidas foi recodificar o algoritmo para a resolução do problema de sintonia. Os parâmetros para a resolução do problema em si foram ajustados de acordo com o último parágrafo da Seção 5.1.1 e novamente voltamos para a etapa de processamento. Após a execução do algoritmo, os parâmetros de sintonia encontrados podem ser observados na Tabela 4 e a TF do controlador, codificada através desses parâmetros, já aplicada a transformada de Laplace, está descrita na equação (6.1).

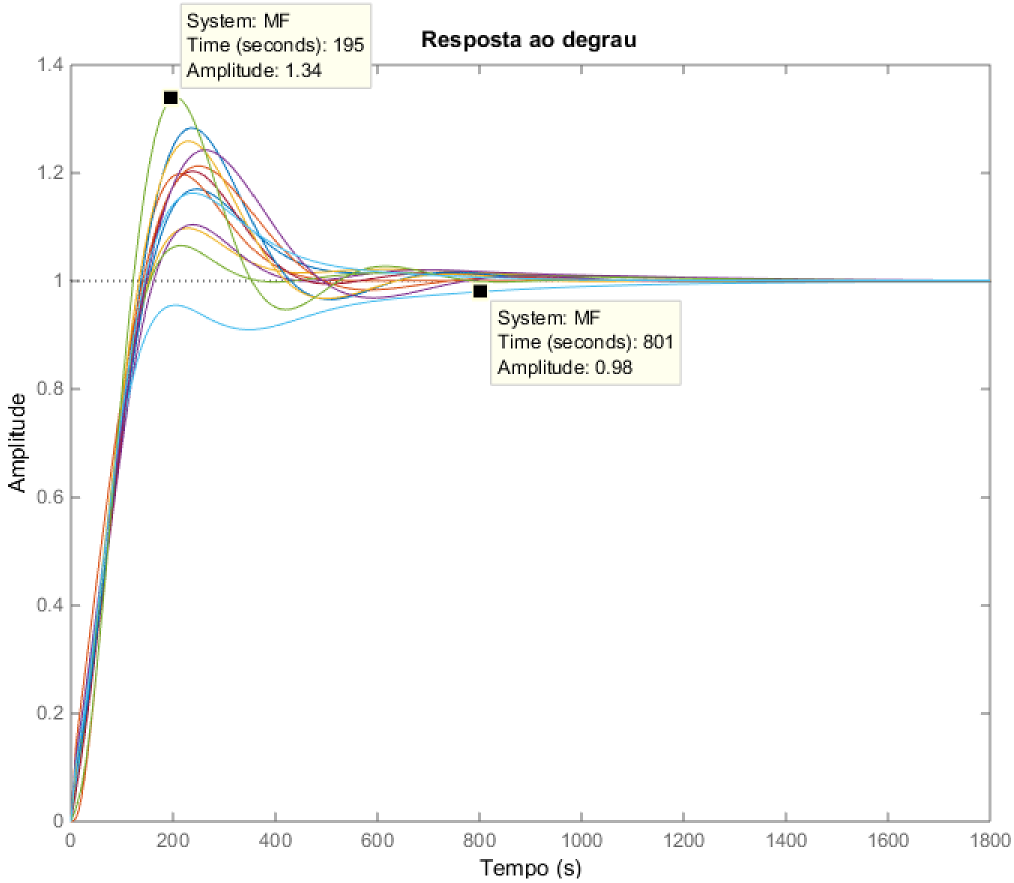
Tabela 4 – Parâmetros de sintonia PID e custo ISE - PSO

K_p	K_i	K_d	T_d	ISE
5.0209	0.0200	13.1412	1.0321	48.6255

$$C = \frac{-18.32s^2 - 5.041s - 0.01998}{1.032s^2 + s}. \tag{6.1}$$

Uma vez disponível os parâmetros de sintonia e a TF do controlador um novo método foi codificado para avaliar G_p uma vez que o C foi aplicado. Um entrada degrau unitário foi aplicada ao sistema e a Figura 21 mostra o comportamento de G_ps mais a planta média em malha fechada.

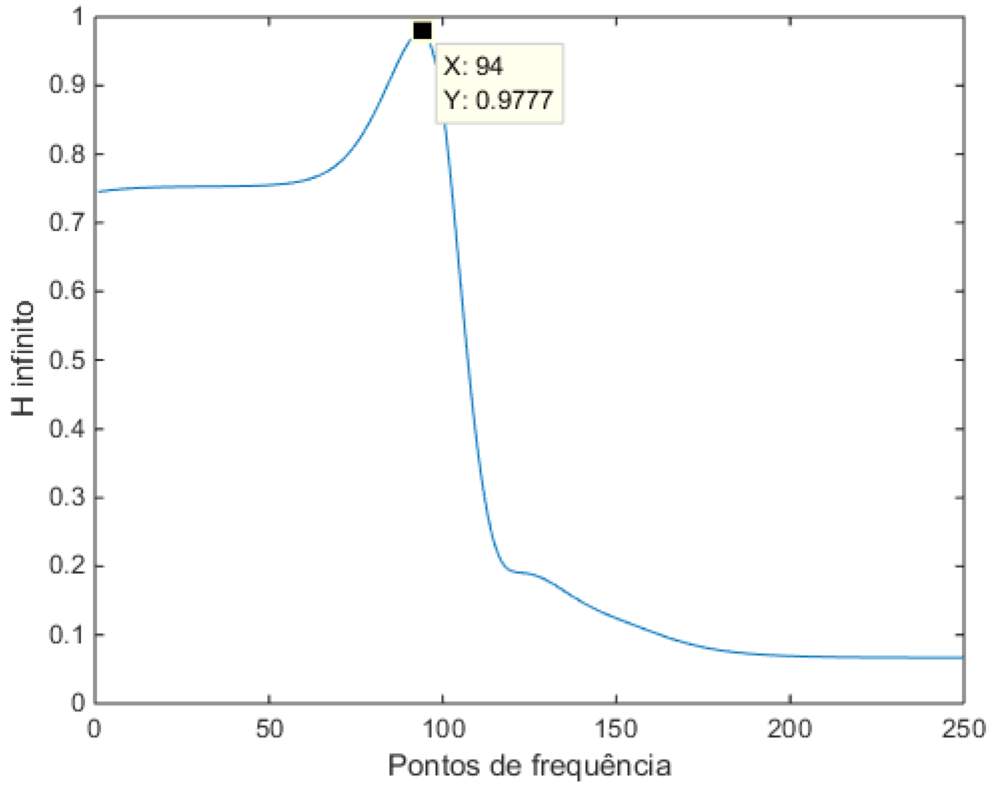
Figura 21 – Resposta ao degrau aplicado ao conjunto de plantas mais a planta média em malha fechada - *PSO*.



Elaborada pelo próprio autor.

Além da análise da aplicação da entrada degrau unitário é necessário verificar se G_p está dentro das especificações de estabilidade e desempenho robusto. Para isso precisamos verificar o valor máximo de H_∞ dentro do espectro de frequência determinado. Foram gerados 250 pontos entre os valores 1e-5 e 1e+3 espaçados logaritmicamente. Na Figura 22 é possível observar que o sistema obedeceu a restrição imposta ao mesmo (Vide Capítulo 5) e que o máximo valor de H_∞ presente no sistema foi de 0.9777.

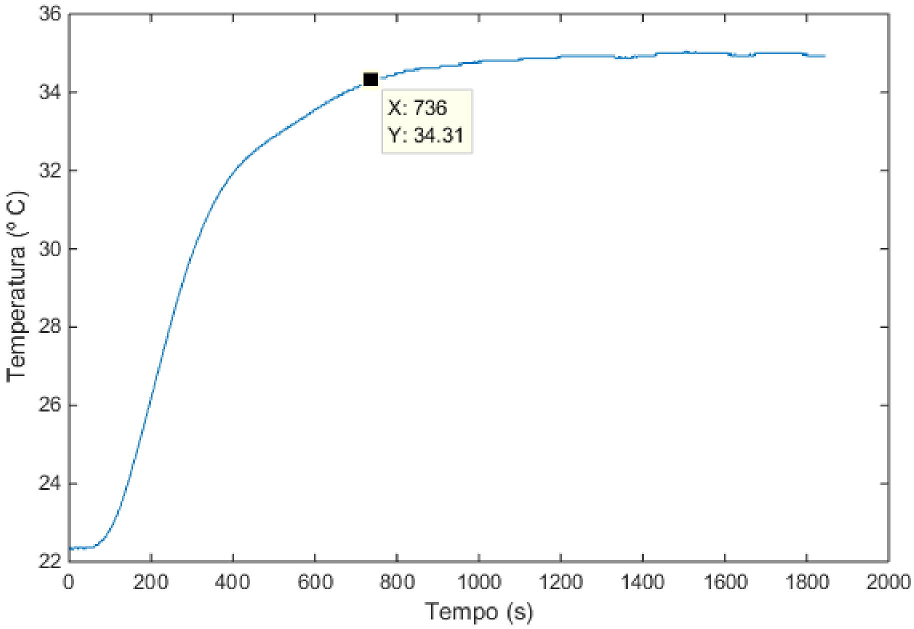
Figura 22 – Verificação gráfica quanto a estabilidade e desempenho robusto - PSO.



Elaborada pelo próprio autor.

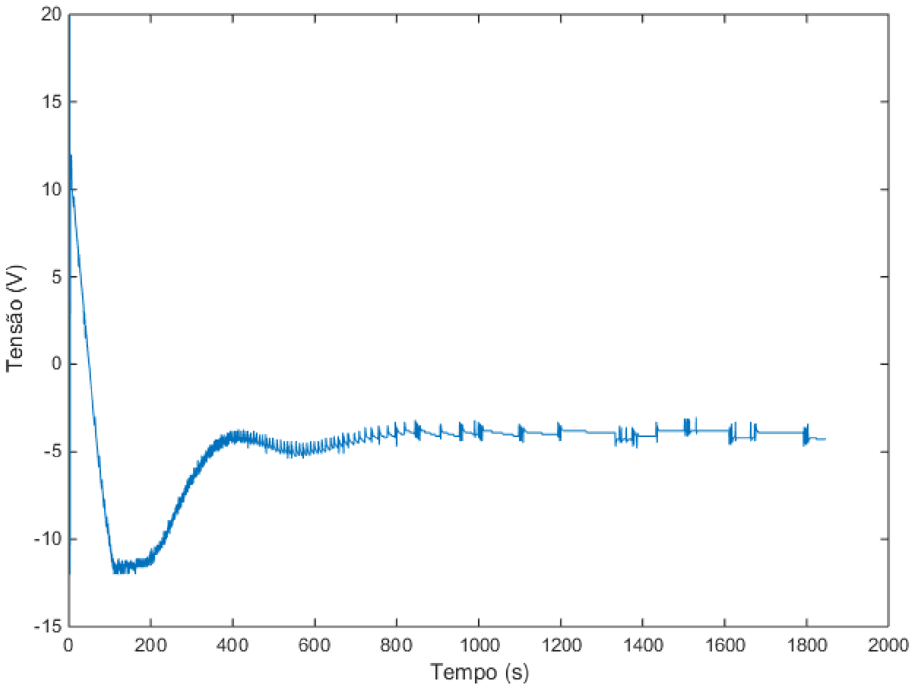
Após todo o processo de simulações e análises passamos para a aplicação no sistema real (Figura 18). A partir da TF contínua apresentada em (6.1) e considerando um tempo de amostragem igual a 0.8 segundos, foi obtida a função de transferência discreta. O controlador discreto foi implementado em um Raspberry Pi utilizado para controlar o sistema. Para uma referência de 35 graus Celsius, a resposta do sistema em malha fechada é apresentada na Figura 23 e tensão aplicada pelo controlador ao sistema ao longo do processo (esforço de controle) é apresentada na Figura 24. Nota-se que o sistema apresentou uma resposta sem sobressinal e com um tempo de assentamento de aproximadamente 736 segundos.

Figura 23 – Resposta do sistema em malha fechada para uma referência de 35 graus Celsius.



Elaborada pelo próprio autor.

Figura 24 – Tensão aplicada pelo controlador ao sistema para uma referência de 35 graus Celsius.



Elaborada pelo próprio autor.

6.1.2 GA aplicado a sintonia PID

Inicialmente nos deparamos com o problema de escolha dos métodos de seleção. Para resolver tal problema, o algoritmo foi modificado e programado conforme mencionado na Seção 5.1.2 para encontrar tais parâmetros. Com aproximadamente 17 horas de processamento todas as matrizes de custo associadas a cada diferente configuração de métodos foram salvas. Uma vez essas matrizes salvas uma inspeção foi executada sobre as mesmas. Essa inspeção assim como na análise para encontrar as constantes do PSO, se baseou na média e no desvio padrão da matriz de custos de cada configuração. Após uma análise sobre todas as matrizes, podemos chegar na melhor configuração para os métodos.

Com os métodos de seleção dos pais e sobrevivência definidos, o algoritmo do GA foi reprogramado para a resolução do problema de sintonia PID. Os parâmetros para a resolução do problema em si foram ajustados de acordo com o ultimo parágrafo da Seção 5.1.2 e novamente voltamos para a etapa de processamento. Após a execução do algoritmo os parâmetros de sintonia encontrados podem ser observados na Tabela 5 e a TF do controlador, codificada através desse parâmetros, já aplicada a transformada de Laplace, está descrita na equação (6.2).

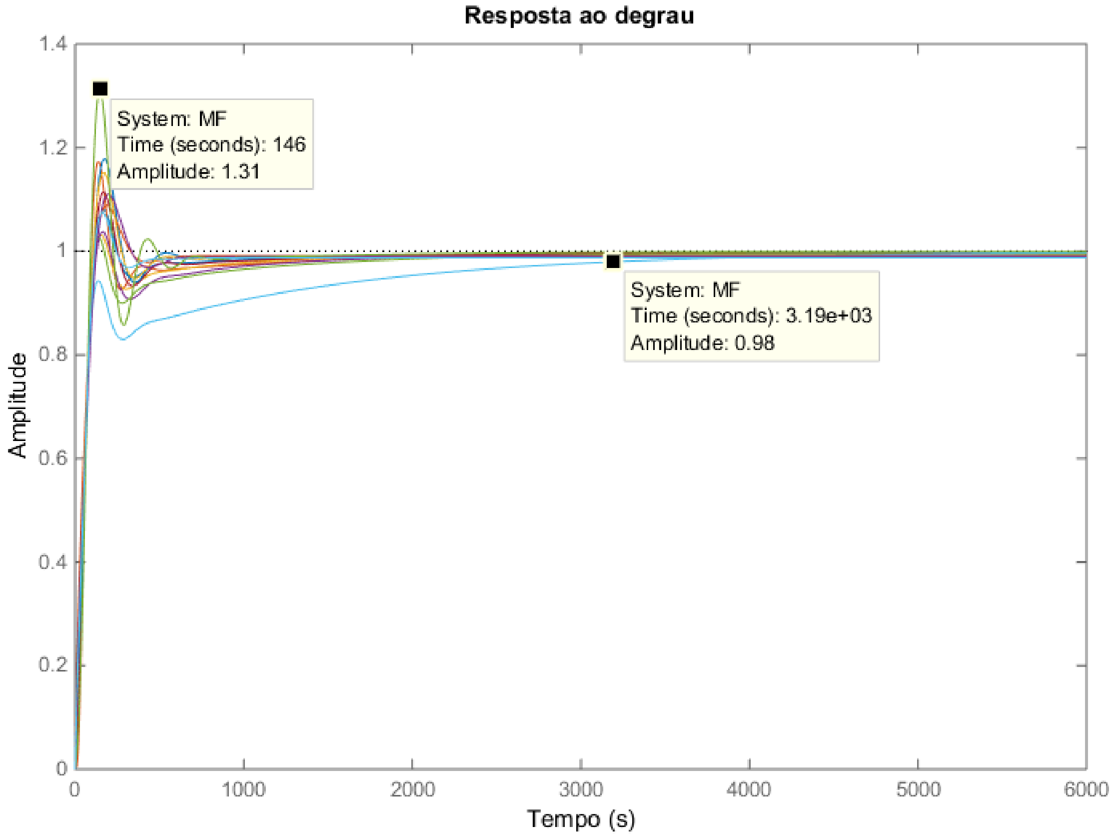
Tabela 5 – Parâmetros de sintonia PID e custo ISE - GA

K_p	K_i	K_d	T_d	ISE
8.9825	0.0077	0.7856	2.1140	33.4855

$$C = \frac{-19.77s^2 - 8.999s - 0.007665}{2.114s^2 + s}. \tag{6.2}$$

Uma vez disponível os parâmetros de sintonia e a TF do controlador o mesmo método utilizado para avaliar G_p no PSO foi utilizado para avaliar o GA uma vez que o C foi aplicado. Um entrada degrau unitário foi aplicada ao sistema e a Figura 25 mostra o comportamento de G_ps mais a planta média em malha fechada.

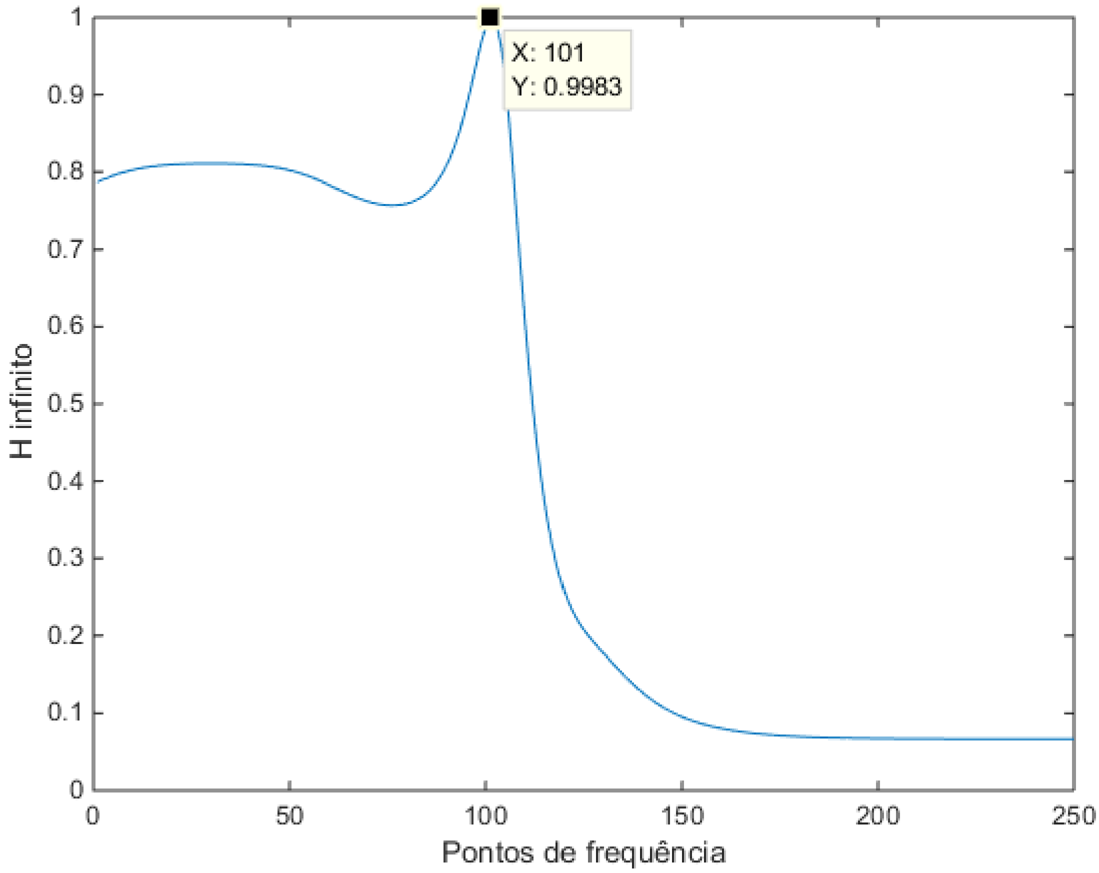
Figura 25 – Resposta ao degrau aplicado ao conjunto de plantas mais a planta média em malha fechada - GA.



Elaborada pelo próprio autor.

Além da análise da aplicação da entrada degrau unitário é necessário verificar se G_p está dentro das especificações de estabilidade e desempenho robusto. Para isso precisamos verificar o valor máximo de H_∞ dentro do espectro de frequência determinado. Foram gerados 250 pontos entre os valores 1e-5 e 1e+3 espessados logaritmicamente. Na Figura 26 é possível observar que o sistema obedeceu a restrição imposta (Vide Capítulo 5) e que o máximo valor de H_∞ presente no sistema foi de 0.9983.

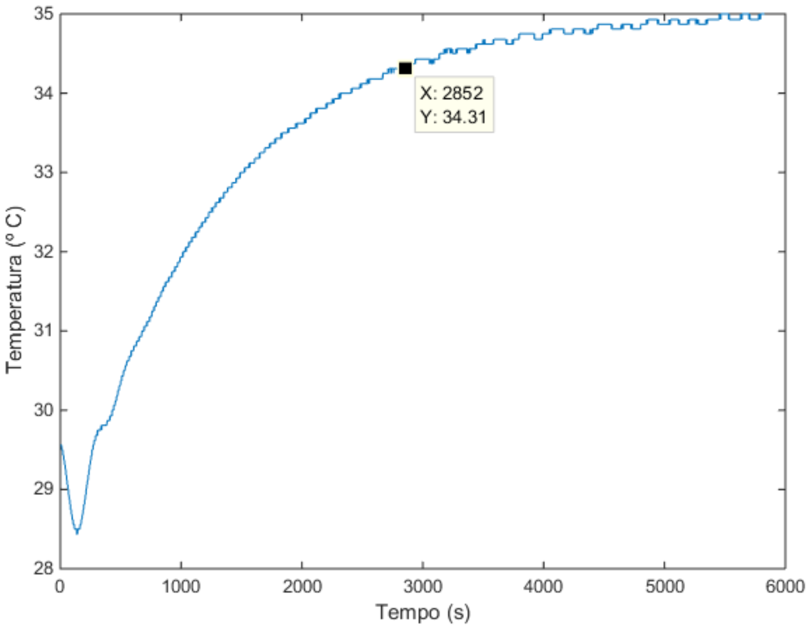
Figura 26 – Verificação gráfica quanto a estabilidade e desempenho robusto - GA.



Elaborada pelo próprio autor.

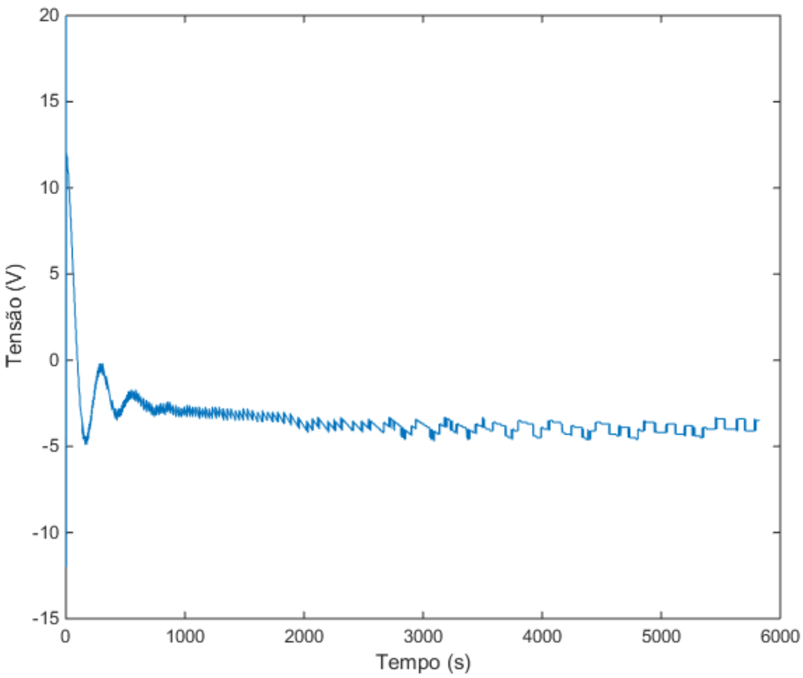
Após todo o processo de simulações e análises passamos para a aplicação no sistema real (Figura 18). A partir da TF contínua apresentada em (6.2) e considerando um tempo de amostragem igual a 0.8 segundos, foi obtida a função de transferência discreta. O controlador discreto foi implementado em um Raspberry Pi utilizado para controlar o sistema. Para uma referência de 35 graus Celsius, a resposta do sistema em malha fechada é apresentada na Figura 27 e tensão aplicada pelo controlador ao sistema ao longo do processo é apresentada na Figura 28. Nota-se que o sistema apresentou uma resposta sem sobressinal e com um tempo de assentamento de aproximadamente 2852 segundos.

Figura 27 – Resposta do sistema em malha fechada para uma referência de 35 graus Celsius.



Elaborada pelo próprio autor.

Figura 28 – Tensão aplicada pelo controlador ao sistema para uma referência de 35 graus Celsius.



Elaborada pelo próprio autor.

6.2 Sintonia por LMIs

Com toda a metodologia de LMIs, bem como todos os algoritmos implementados, podemos agora apresentar os resultados obtidos.

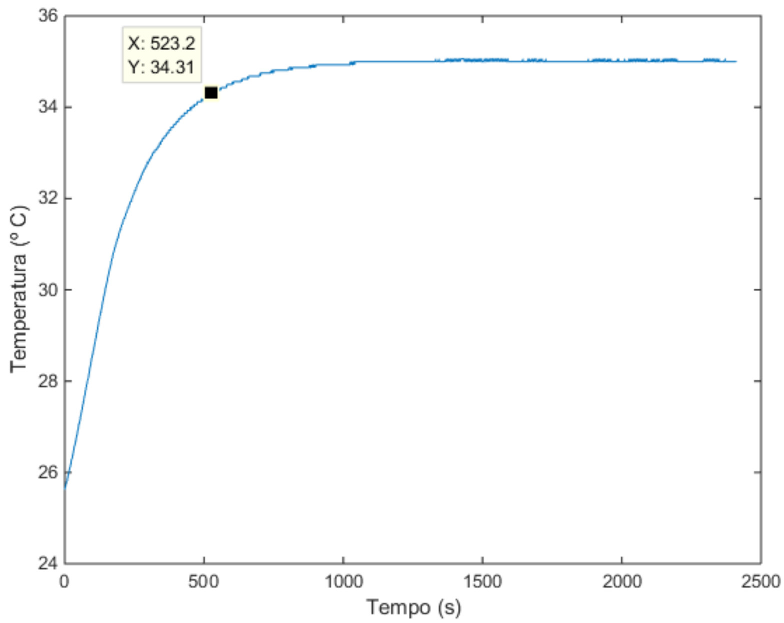
Seguindo a Seção 6.2 os parâmetros para a resolução do problema foram ajustados e após a execução do algoritmo, os parâmetros de sintonia encontrados podem ser observados na Tabela 6.

Tabela 6 – Parâmetros de sintonia LMI e K_i.

K_1	K_2	K_3	K_4	K_5	K_i
-1.1623	6.2967	5.5664	0.3165	8.3728	-0.0683

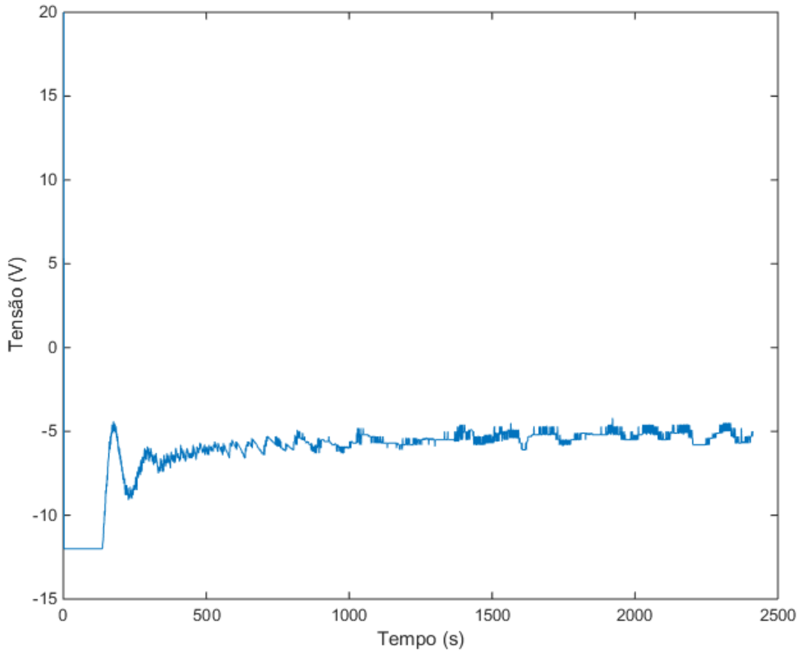
Após todo o processo de análises passamos para a aplicação no sistema real (Figura 18). A partir dos parâmetros da Tabela 6 e considerando um tempo de amostragem igual a 0.8 segundos, o integrador foi discretizado e juntamente com os ganhos, os parâmetros foram implementados no Raspberry Pi utilizado para controlar o sistema. Para uma referência de 35 graus Celsius, a resposta do sistema em malha fechada é apresentada na Figura 29 e tensão aplicada pelo controlador ao sistema ao longo do processo (esforço de controle) é apresentada na Figura 30. Nota-se que o sistema apresentou uma resposta sem sobressinal e com um tempo de assentamento de aproximadamente 523 segundos. Além disso a Figura 31 mostra a formação das regiões para a D-estabilidade com todos os autovalores dos vértices das matrizes do sistema em malha fechada.

Figura 29 – Resposta do sistema em malha fechada para uma referência de 35 graus Celsius.



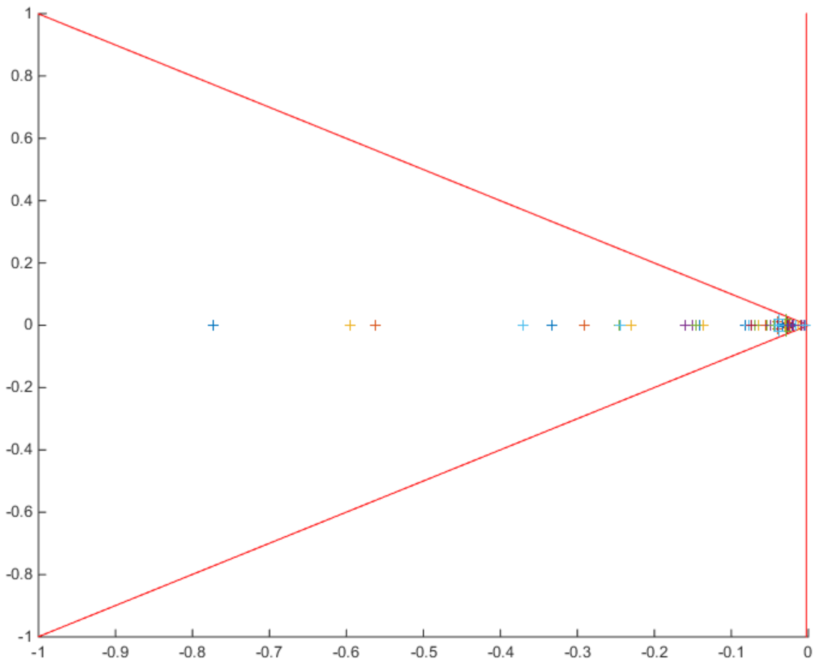
Elaborada pelo próprio autor.

Figura 30 – Tensão aplicada pelo controlador ao sistema para uma referência de 35 graus Celsius.



Elaborada pelo próprio autor.

Figura 31 – Região cônica localizada no semiplano esquerdo do plano S mostrando todos os autovalores dos vértices das matrizes do sistema em malha fechada



Elaborada pelo próprio autor.

6.3 Comparação entre abordagens por meta-heurísticas e LMIs

Considerando as informações apresentadas no capítulo 5 e os resultados obtidos no capítulo 6 é possível realizar uma comparação entre as duas abordagens aplicadas neste trabalho. Para tal, serão considerados o tempo de assentamento do sistema e o esforço computacional dos algoritmos, afim de determinar qual abordagem melhor se adapta a esse cenário (Figura 18).

6.3.1 Tempo de assentamento

O tempo de assentamento (t_s) é o tempo gasto para o sinal acomodar na faixa de $\pm 2\%$ a $\pm 5\%$ porcento do valor final.

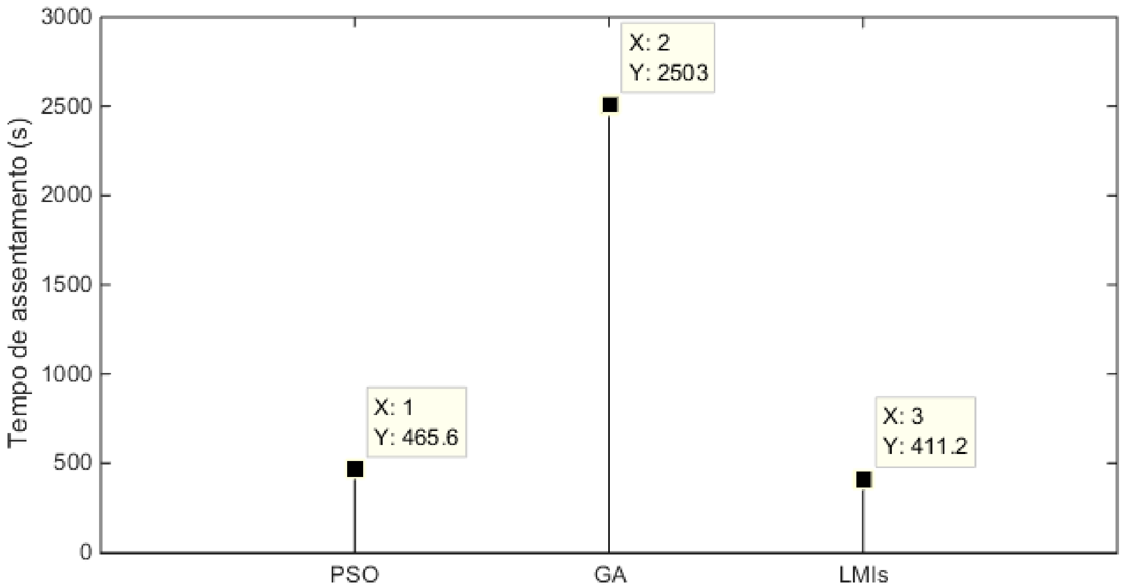
Na Seção 6 foram apresentados os resultados com relação a essa métrica para cada metodologia aplicada ao sistema, bem como uma representação gráfica apresentando a resposta do sistema em malha fechada para uma referência de 35 graus Celsius. Cada representação gráfica dessa seção apresenta o ensaio como um todo, ou seja, cada algoritmo é inicializado em uma temperatura inicial aleatória, uma vez que não há como prever a temperatura do sistema antes do início de cada algoritmo.

Para uma melhor comparação entre as abordagens vamos inicialmente considerar parte desse ensaio onde a temperatura inicial será considerada comum a todos as abordagens e inicializada em 29 graus Celsius. A Figura 32 representa essa comparação. Analisando esta, podemos perceber que as LMIs obtiveram o melhor tempo de assentamento, ficando em 411,2 segundos, o PSO obteve um tempo 13,21% maior, ficando em 465,6 segundos e por fim, o GA obteve um tempo 508,71% pior, ficando em 2503 segundos.

Considerando agora o ensaio como um todo, vamos analisar o tempo de assentamento para cada algoritmo. Voltando a Seção 6 podemos obter os dados com relação ao tempo de assentamento para cada abordagem. As LMIs assim como no ensaio fixo de temperatura inicial, obtiveram o melhor tempo de assentamento, ficando em 523,2 segundos, para uma temperatura inicial de 25,62 graus Celsius. O PSO obteve um tempo 40,67% maior, ficando em 736 segundos, para uma temperatura inicial de 22,31 graus Celsius. Por fim, o GA obteve um tempo 445,11% maior, ficando em 2852 segundos, para uma temperatura inicial de 29,56 graus Celsius.

Uma observação bastante pertinente parte da análise da comparação entre as duas meta-heurísticas no ensaio total. Podemos perceber que o PSO mesmo partindo de uma temperatura inicialmente 32,50% menor do que o GA, conseguiu um tempo de assentamento 74,19% menor.

Figura 32 – Comparação do tempo de assentamento para os três métodos de otimização aplicados ao sistema.



Elaborada pelo próprio autor.

6.3.2 Esforço computacional

Consideraremos como esforço computacional nessa monografia o tempo de processamento para cada algoritmo. Se voltarmos nas seções 5.1.1 e 5.1.2 podemos ver que ambas meta-heurísticas necessitam de um pré-processamento para definição de contantes utilizadas no *PSO* e métodos de seleção para os pais e sobrevivência utilizados no *GA*. Para determinar esse parâmetros para ambos algoritmos o métodos de Monte Carlo foi utilizado. Além disso alguns outros parâmetros foram configurados de forma a igualar o tempo entre os algoritmos. O número de partículas no *PSO* foi igualado ao número inicial da população no *GA*, o número de iterações no *PSO* foi igualado ao número de gerações no *GA* e ambas meta-heurísticas submetidas a um *loop* de 100 iterações.

Para o *PSO*, o valor de 3 constantes precisam ser atribuídos e para determiná-las todas as combinações possíveis precisaram ser geradas e então analisadas. Um total de 90 horas de processamento foi gasto para gerar todas as 90 matrizes de combinações, logo temos 1 hora para o processamento de cada configuração. Uma vez que todas as constantes estavam definidas foram gastos mais 13223 segundos (aproximadamente 3,67 horas) para o processamento do algoritmo aplicado a resolução do problema de sintonia.

Para o *GA*, dois métodos de seleção precisam ser definidos e para determiná-los todas as combinações possíveis precisaram ser geradas e então analisadas. Um total de 17 horas de processamento foi gasto para gerar todas as 16 matrizes de combinações, logo temos 56 minutos para o processamento de cada configuração. Uma vez que todas os métodos foram definidos foram gastos mais 7256 segundos (aproximadamente 2,02 horas) para o processamento do algoritmo aplicado a resolução do problema de sintonia.

Para as LMIs não temos esse tipo de pré processamento. Uma vez programado, o algoritmo gastou 7,54 segundos para encontrar a solução.

Novamente vemos que as LMIs foram superiores quanto ao tempo de processamento. Porém esse método não necessita de um pré-processamento. Assim comparando somente os tempos de execução entre os algoritmo para resolver o problema de sintonia, uma vez que a diferença entre o tempo de processamento do pré-problema do PSO e do GA não foi relevante, temos: O GA é aproximadamente 962 vezes mais lento quando comparado as LMIs, enquanto que o PSO é 1754 vezes mais lento quando comparado as LMIs.

7 Considerações finais

Este trabalho apresentou um solução para o problema de sintonia robusta de controladores utilizando duas abordagens de otimização, por meta-heurísticas e por programação semidefinida. Esta sintonia foi aplicada a um sistema de controle térmico real e, ambas as abordagens obtiveram resultados interessantes.

Para a abordagem envolvendo meta-heurísticas, o controlador escolhido para a sintonia foi do tipo PID. Após tratar cada particularidade de cada algoritmo implementado (PSO e GA), impor penalidade e restrições ao sistema, a sintonia foi obtida e como mostrado na Seção 6.1, ambos algoritmos respeitaram as restrições e obtiveram resultados que satisfizeram tanto o desempenho robusto quando o tempo de assentamento uma vez em teste no sistema real.

Considerando a abordagem de síntese robusta no domínio do tempo por meio de modelos em espaço de estados e desigualdades matriciais lineares (LMIs), um controlador robusto por realimentação de estados foi escolhido para a sintonia. A Seção 6.2 apresenta a análise e os resultados dessa abordagem, e podemos perceber que as LMIS implementadas obtiveram soluções que obedeceram o critério da D-estabilidade. Uma vez aplicando os parâmetros de sintonia no sistema real, os resultados foram bastante satisfatórios, assim como na primeira abordagem.

Comparando essas duas abordagens através de 2 métricas - Tempo de assentamento e Esforço computacional - para um ponto de referência comum, foram realizados 2 tipos de teste aplicados à primeira métrica e um único tipo aplicado a segunda.

Com relação a primeira métrica - Tempo de assentamento - o primeiro teste, obedece um ponto comum de inicialização do sistema real para os 3 métodos de otimização. Já no segundo teste, cada algoritmo tem o seu próprio ponto inicial de temperatura, uma vez que é impossível prevê-la antes de inicializá-lo. Para ambos os teste, as LMIS se mostraram superiores quando comparadas aos outros algoritmos. Se classificarmos como em um *ranking* teríamos: LMIs, seguidas pelo PSO e então pelo GA, para os dois testes.

Com relação a segunda métrica - Esforço computacional - o único teste realizado foi a comparação entre os tempos de execução de cada algoritmo para resolver o problema de sintonia. Novamente as LMIs se mostraram superioras e podemos classificar os algoritmos através do seguinte *ranking*: LMIs, GA e então PSO. Assim, as LMIs foram a melhor solução abordada nessa monografia que solucionou o problema de sintonia de controlados com o melhor desempenho.

Referências

ARRUDA, L. V. R. de et al. Um método evolucionário para sintonia de controladores pi/pid em procesos multivariáveis. *Revista Controle & Automação*, v. 19, n. 19, p. 2, 1 2008. Meses: 1, 2 e 3. 4, 5

BOYD, S. et al. *Linear Matrix Inequalities in System and Control Theory*. 15. ed. Philadelphia: Library of Congress Cataloging-in-Publication Data, 2010. 26, 27

BROGAN, W. L. *Modern Control Theory*. 3. ed. New Jersey 07458: Prentice Hall, 1991. 13, 14, 30

FACCIN, F. *Abordagem Inovadora no Projeto de Controladores PID*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul - Programa de Pós-Graduação em Engenharia Química, Porto Alegre, 2004. 4, 7, 8, 9, 10, 12

HAUPT, R.; HAUPT, S. E. *Practical genetic algorithms*. 2. ed. Hoboken - New Jersey: John Wiley & Sons, 2004. 3, 4, 27, 28, 29, 30

HENRION, D. *STATE-SPACE ANALYSIS*. 2010. Belgian Graduate School on Systems, Control, Optimization and Networks. Lecture notes. 38

JONG, K. de. *Evolutionary Computation - A Unified Approach*. 2. ed. Cambridge, MA: A Bradford Book - The MIT Press, 2006. 27, 28, 29, 30

KUO, B. C. *Sistemas De Controle Automático*. 4. ed. São Paulo: Phb, 1995. 1

LING, B. Stat feedback regional pole placement via lmi optimization. *American Control Conference*, p. 3475–3480, 6 2001. 30, 31, 37

LOURENÇO, J. *Sintonia de Controladores P.I.D.* Manaus, 1997. Realizado em Janeiro de 96 e revisto em Janeiro de 97. 7, 8, 9, 11, 12

MASSERA, J. M. A. *Técnicas de Programação Semidefinida e Aplicações em Otimização de Material*. Tese (Doutorado) — Pós-Graduação e Pesquisa de Engenharia (COPPE) da Universidade Federal do Rio de Janeiro - RJ, Rio de Janeiro, 7 2010. 30

OGATA, K. *Engenharia de controle moderno*. 5. ed. São Paulo: Pearson Prentice Hall, 2010. 1, 2, 12, 13, 15, 16

PEREIRA, A. M. F. et al. Tensor product model tranformation simplification of takagi-sugeno control and estimation laws - an aplication to a thermoelectric controlled chamber. *Acta Polytechnica Hungarica*, 2018. Aceito para publicação. 32

PINTO, J. *Aplicação Prática do método de Sintonia de Controladores PID Utilizando o Método do Relé com Histerese*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte - Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Natal, 2014. 8, 9, 10, 11

SCHERER, C.; SIEPWEILAND. *Linear Matrix Inequalities in Control*. 2015. University of Stuttgart and Eindhoven University of Technology. Lecture notes. 31, 38, 39

SEBORG, D. E. et al. *Process Dynamics and Control*. 3. ed. Hoboken - New Jersey: John Wiley & Sons, 2004. 8, 9, 10, 16

SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable Feedback Control - Analysis and Design*. 2. ed. New York: John Wiley & Sons, 2005. 2, 3, 17, 18, 19, 20, 21, 22, 23, 24, 25, 31, 33

STIMAC, G.; BRAUT, S.; ZIGULIC, R. Comparative analysis of pso algorithms for pid controller tuning. *CHINESE JOURNAL OF MECHANICAL ENGINEERING*, v. 27, n. 5, p. 1, 5 2014. 35

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE ENGENHARIA DE COMPUTAÇÃO

ANEXO III – Termo de Responsabilidade

TERMO DE RESPONSABILIDADE

Eu, Valtan Alex da Silva, declaro que o texto do trabalho de conclusão de curso intitulado "Comparação entre abordagens de otimização por meta-heurísticas e programação semidefinida para a síntese de controladores" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 09 de julho de 2018

Valtan Alex da Silva
Assinatura do aluno

Certifico que o aluno **Dalton Alex da Silva,** autor do trabalho de conclusão de curso intitulado "**Comparação entre abordagens de otimização por meta-heurísticas e programação semidefinida para a sintonia de controladores**"**,** efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

Víctor Costa da Silva Campos

Víctor Costa da Silva Campos
Orientador

Belo Horizonte, 11 de julho de 2018.