



Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Departamento de Engenharia Elétrica
Engenharia de Controle e Automação

Pedro Vitor Soares Gomes de Lima

Desenvolvimento de Ferramenta no MATLAB para Suporte em Projetos de Controladores Repetitivos

Recife

2020

Pedro Vitor Soares Gomes de Lima

**Desenvolvimento de Ferramenta no MATLAB para Suporte em Projetos de
Controladores Repetitivos**

Trabalho de Conclusão de Curso submetido
ao Departamento de Engenharia Elétrica da
Universidade Federal de Pernambuco como
parte dos requisitos para obtenção do grau
de Bacharel em Engenharia de Controle e
Automação.

Orientador: Prof. Fabrício Bradaschia

Recife

2020

AGRADECIMENTOS

À minha família, por sempre acreditar em mim e me apoiar. Em especial, à minha mãe e ao meu pai, que estiveram ao meu lado em todas as etapas da minha vida.

Aos colegas e amigos que estão comigo há um longo tempo. Àqueles que são do tempo da escola e que permanecem até hoje, em especial, Gabriel. Àqueles que conheci desde o início do curso e se fizeram bastante importantes, em especial, Mariana, Leonardo, Felipe, Flávio e Valdemar.

Aos amigos que a Equipe Maracatronics me permitiu conhecer, em especial, Domingos, Ana Carolinna, Luís e Êmerson. Além desta ter feito parte de grande parte do meu período acadêmico e ser também responsável pelo profissional que me tornei.

Aos amigos que o intercâmbio me proporcionou e que contribuíram para que essa fosse em uma das experiências mais importantes da minha vida, em especial, Almir, Augusto, Maria Laura, João Guilherme e Julia, pela tamanha identificação e pelo convívio diário. E àqueles que primeiro conheci e que também se tornaram parceiros de viagens, Weider, Talissa e Nathanael.

Aos professores que contribuíram com a minha trajetória acadêmica, especialmente Fabrício Bradaschia, um exemplo de professor que, além de tê-lo como orientador deste trabalho, tive a oportunidade de ser seu monitor. Especialmente também ao professor Rafael Neto, por ser o idealizador deste projeto e por ser tão atencioso e paciente.

À universidade, pela sua responsabilidade social e por me proporcionar realizar um curso que sem ela não seria possível. Também por, apesar das diversas dificuldades, continuar com seu processo de formação do ser humano, do profissional e do cidadão.

A todos que, de alguma maneira, contribuíram para a realização deste trabalho.

RESUMO

Aplicações que demandam inversores trifásicos conectados à rede são cada vez mais presentes e, sendo o desempenho desses inversores relacionado à qualidade da energia gerada, é necessário que a estratégia de controle escolhida para eles seja de alta performance. Por isso, diversas abordagens são propostas a fim de garantir o controle eficiente de um determinado sistema; para este trabalho, o objeto de estudo é o controlador repetitivo complexo. Baseado em uma proposta de configuração para este controlador, o objetivo deste trabalho é criar um aplicativo no *software* MATLAB que funciona como suporte para projetos com a dita estrutura. Ele permite que o usuário visualize a influência dos parâmetros do controlador na estabilidade do sistema e encontre parâmetros ideais para cada caso. Para validar o funcionamento do aplicativo criado, primeiramente são propostos alguns testes através de simulações no próprio *software* MATLAB e, posteriormente, experimentos em uma planta real, envolvendo a análise do comportamento do sistema em relação ao esperado segundo a ferramenta desenvolvida. O sucesso obtido nessas avaliações possibilitou demonstrar tanto a correta implementação do controlador objeto de estudo, quanto a praticidade e importância da ferramenta criada, atingindo o objetivo inicial.

Palavras-chave: Projeto. Controlador Repetitivo. Aplicativo. MATLAB.

ABSTRACT

Three-phase grid-connected applications are increasingly present worldwide and choosing an appropriate control strategy for them is of high importance, since the performance of these inverters is related to the quality of power generation. Therefore, several approaches are proposed to ensure adequate control of a given system. In this work, the complex repetitive controller is the object of study. Based on a configuration proposed by other author for that controller, an application was created in MATLAB software to support projects with that structure. It allows the user to visualize the influence of the controller parameters on system stability and to find optimal parameters for each case. To validate the application, first, some tests are proposed through simulations in MATLAB software and then experiments in a real converter prototype, showing if the system behavior is as expected, according to the developed tool. The success obtained in these evaluations made it possible to demonstrate both the correct implementation of the studied controller, as well as the practicality and importance of the developed tool, reaching the initial objective.

Keywords: Design. Repetitive Controller. Application. MATLAB.

LISTA DE ILUSTRAÇÕES

Figura 1	Resposta em frequência de $\mathfrak{F}(\bar{s}_{\alpha\beta}^{(h_s)})$ para h_s positivo.....	13
Figura 2	Estrutura básica do controlador repetitivo.	14
Figura 3	Diagrama de blocos do controlador proposto por (NETO, 2018).	15
Figura 4	Resposta em frequência do controlador para diferentes valores de m e n . .	16
Figura 5	Diagrama de blocos para o estudo através do Teorema do Ganho Pequeno.	17
Figura 6	Sistema de controle genérico com o RC Complexo proposto em (NETO, 2020).	18
Figura 7	Sistema equivalente àquele com o RC Complexo Proposto.....	18
Figura 8	Exemplo de domínio de estabilidade (em verde) para um sistema com RC Complexo proposto em (NETO, 2020) com diversos valores de a	20
Figura 9	Resultado de simulação do sistema exemplo com o RC Complexo proposto em (NETO, 2020) para diferentes valores do parâmetro a	21
Figura 10	Sistema de controle genérico com o RC Complexo Modificado.	22
Figura 11	Comparação entre os domínios de estabilidade em sistemas com o RC Complexo proposto em (NETO, 2020) e com o RC Complexo Modificado.	23
Figura 12	Estabilidade do sistema com RC Complexo Modificado ($a = 0$) e com $Q(s)$ como um filtro passa-baixa.	24
Figura 13	Página inicial da ferramenta GUIDE.	27
Figura 14	Página inicial da ferramenta App Designer.....	28
Figura 15	Seção responsável pela definição da planta a ser controlada.	31
Figura 16	Seção responsável pelas informações a respeito da criação da ferramenta..	32
Figura 17	Seção para análise de estabilidade de acordo com os parâmetros do controlador.	34
Figura 18	Seções de configuração.	35
Figura 19	Seção para dimensionamento do filtro ideal para uma dada aplicação.	36
Figura 20	Fluxograma referente ao algoritmo de dimensionamento do filtro ideal. ...	43
Figura 21	Diagrama do sistema escolhido para validação da ferramenta.	44
Figura 22	Comportamento no tempo da corrente da fase a da carga não-linear i_{la}	45

Figura 23	Representação da corrente da fase a da carga não-linear i_{la} no domínio da frequência.....	45
Figura 24	Diagrama de blocos do sistema.....	47
Figura 25	Seção de análise de estabilidade da ferramenta criada com $Q(z) = 1$ e $a = 0.4$, após carregar os parâmetros do FAP.	48
Figura 26	Análise da estabilidade do sistema a partir da ferramenta criada para vários valores de a (com $Q(z) = 1$) e os resultados do sistema de controle simulado para cada caso, mostrando instabilidade em todas as situações..	49
Figura 27	Diagrama de Nyquist da planta e domínio de estabilidade do controlador em torno da origem, mostrando a impossibilidade de se garantir estabilidade com $Q(z) = 1$	50
Figura 28	Análise da estabilidade do sistema a partir da ferramenta criada com $a = 0,5$ e $Q(z) \neq 1$	51
Figura 29	Erro de corrente do sistema de controle para $a = 0,5$ e $Q(z) = 0,6$	51
Figura 30	Execução do algoritmo de dimensionamento do filtro $Q(z)$ para o FAP em análise.....	52
Figura 31	Projeto de um filtro FIR de ordem 6 a partir da curva limite de estabilidade de $Q(z)$ gerada pela ferramenta desenvolvida, que é aplicada no sistema de controle do FAP em análise.	53
Figura 32	Resultado experimental com filtro FIR dimensionado a partir da ferramenta, com $a = 1$	54
Figura 33	Resultado experimental com filtro FIR projetado e para diferentes valores do parâmetro a	55
Figura 34	Execução do algoritmo de dimensionamento do filtro ideal para planta da Equação (2.14).	56
Figura 35	Projeto dos filtros baseado no limite definido pela ferramenta para a planta da Equação (2.14).	57
Figura 36	Módulo do erro para diferentes filtros.	57

LISTA DE TABELAS

Tabela 1	Comparação entre ferramentas de criação de aplicativos do MATLAB. ...	30
Tabela 2	Parâmetros da aplicação.	44
Tabela 3	Componentes harmônicas da corrente da fase a da carga não-linear i_{la}	46
Tabela 4	Parâmetros e resultados da execução do algoritmo de dimensionamento do filtro ideal para o caso real.	52
Tabela 5	Parâmetros e resultados da execução do algoritmo de dimensionamento do filtro ideal para planta da Equação (2.14).	56

LISTA DE ABREVIATURAS E SIGLAS

CA	Corrente Alternada
CC	Corrente Contínua
FAP	Filtro Ativo de Potência
GUIDE	Ambiente de Desenvolvimento de Interface Gráfica do Usuário - <i>Graphical User Interface Development Environment</i>
MATLAB	MATrix LABoratory
PI	Controlador Proporcional-Integral
RC	Controlador Repetitivo - <i>Repetitive Controller</i>
SOGI	Integrador Generalizado de Segunda Ordem - <i>Second Order Generalized Integrator</i>
ZOH	Retentor de Ordem Zero - <i>Zero-Order Holder</i>

SUMÁRIO

1	INTRODUÇÃO	10
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	VETORES ESPACIAIS	12
2.2	CONTROLADOR REPETITIVO COMPLEXO	14
2.3	ANÁLISE DE ESTABILIDADE	16
2.3.1	Norma \mathcal{H}_∞ e Teorema do Ganho Pequeno	16
2.3.2	RC Complexo proposto em (NETO, 2020)	18
2.3.3	RC Complexo Modificado	21
2.4	CONCLUSÃO	25
3	DESENVOLVIMENTO DA FERRAMENTA	26
3.1	FERRAMENTAS DO MATLAB PARA CRIAÇÃO DE APLICATIVOS ...	26
3.2	FUNCIONAMENTO	30
3.2.1	Definição da Planta	30
3.2.2	Parâmetros do Controlador	32
3.2.3	Dimensionamento do Filtro	35
3.3	DETALHES DE IMPLEMENTAÇÃO	36
3.3.1	Estrutura da Ferramenta	36
3.3.2	Compartilhamento de Dados entre Arquivos	37
3.3.3	Gráfico de Estabilidade	38
3.3.4	Algoritmo de Dimensionamento do Filtro Ideal	40
3.4	CONCLUSÃO	42
4	VALIDAÇÃO DA FERRAMENTA NUMA APLICAÇÃO REAL	44
4.1	RESULTADOS	48
4.1.1	Variação no Parâmetro a (com $Q(z) = 1$)	48
4.1.2	Variação nos Parâmetros a e $Q(z)$	48
4.1.3	Dimensionamento do Filtro	50
4.1.4	Impacto da Escolha da Equação do Filtro	53
4.2	CONCLUSÃO	58
5	CONCLUSÃO	59

1 INTRODUÇÃO

A energia produzida através de recursos energéticos distribuídos normalmente apresenta tensões incompatíveis ou fora de sincronia com a rede elétrica à qual devem ser conectados, sendo de suma importância que a conversão de energia seja feita de tal maneira à tornar essa ligação adequada à rede. Seja em aplicações onde a tensão de saída da fonte de energia possua uma frequência variável, como visto quando são utilizadas máquinas rotativas, a exemplo de turbinas eólicas e micro turbinas; ou seja em aplicações onde a tensão de saída da fonte de energia seja contínua e variável, como por exemplo em matrizes solares, o uso de inversores trifásicos é essencial para o sistema por ser o equipamento responsável pelo escoamento da energia gerada na rede elétrica (THIBAUT, 2006). Dessa forma, é de extrema importância que a estratégia de controle utilizada seja de alto desempenho, já que isso estará diretamente relacionado à qualidade da energia injetada (KAZMIERKOWSKI; MALESANI, 1998).

A escolha da estratégia de controle se resume a fazer com que as correntes de saída dos inversores trifásicos sejam controladas, mesmo elas apresentando alto conteúdo harmônico. Segundo o princípio do modelo interno (FRANCIS; WONHAM, 1975), um dado sistema em malha fechada segue uma determinada referência, sem erro de regime permanente, se o modelo matemático dessa referência estiver incluído no sistema realimentado. Como o tipo de aplicação que está sendo considerado se refere ao tratamento de sinais periódicos, soluções simples e comumente utilizadas, como é o caso do controlador Proporcional-Integral (PI), não são adequadas por não conseguirem eliminar o erro em regime permanente (NETO, 2018).

Nos casos onde as correntes são puramente senoidais, são alcançados resultados satisfatórios ao utilizar controladores PI aplicados em referencial síncrono dq ou controladores ressonantes, que são baseados nos Integradores Generalizados de Segunda Ordem (SOGI). Nos casos onde se faz necessário o controle de múltiplas componentes harmônicas, as propostas mais conhecidas são o uso de múltiplos SOGIs em paralelo e o uso de controladores repetitivos. De modo geral, todas essas soluções trabalham de forma a aplicar ganho infinito em um determinado conjunto de frequências (NETO et al., 2018).

Exceto pelo controlador repetitivo, as estruturas citadas anteriormente requerem que sejam utilizadas múltiplas células de controle em paralelo para cada harmônica que

se deseje compensar (NETO, 2018). Portanto, é claramente inviável a implementação dessas estruturas quando o número de componentes harmônicas torna-se excessivamente grande. O controlador repetitivo é uma alternativa para tal por possuir uma estrutura onde é possível compensar componentes harmônicas múltiplas da frequência fundamental do sinal de referência apenas a partir da escolha de determinados parâmetros.

Este trabalho explora a proposta de (NETO, 2020) de uma configuração para o controlador repetitivo, a qual será explicada com mais detalhes posteriormente. Por ser um controlador novo, não existe disponível uma ferramenta de projeto dos seus parâmetros em *softwares* comerciais, o que se faz interessante por ser um controlador que possui uma elevada complexidade na análise e no projeto. Então, o desenvolvimento de uma ferramenta para tal estrutura de controlador é uma maneira de facilitar futuros projetistas que se interessem por usá-la. O objetivo deste trabalho é o desenvolvimento de um aplicativo embarcado no *software* MATLAB que retrata precisamente o controlador proposto em (NETO, 2020) e onde é permitido ao usuário alterar livremente parâmetros do controlador repetitivo explorado e observar o impacto na estabilidade do sistema, além de também ser possível encontrar valores ideais para alguns parâmetros, dada uma determinada estrutura pré-definida pelo projetista. O usuário poderá, assim, projetar de maneira ágil e precisa os parâmetros para a sua aplicação. A validação do aplicativo criado se dá a partir de simulações de determinados sistemas que utilizam o controlador projetado pelo aplicativo na ferramenta Simulink, também presente no MATLAB, e de experimentos em um protótipo real, utilizando os parâmetros de projeto provenientes do aplicativo desenvolvido e observando se o comportamento do sistema é o esperado pelo projeto.

A organização do trabalho se dá da seguinte forma: no Capítulo 2, é apresentado de forma mais detalhada o tipo e a estrutura do controlador que serve como base para o aplicativo e como os parâmetros desse controlador interferem na estabilidade do sistema, dada uma determinada planta; o Capítulo 3 descreve o funcionamento, as interfaces e as funcionalidades do aplicativo desenvolvido, apresentando também detalhes de implementação; o Capítulo 4 é responsável pela validação da ferramenta criada, mostrando a metodologia utilizada e discorrendo sobre os resultados dos testes; por fim, o Capítulo 5 apresenta as principais contribuições deste trabalho, assim como os objetivos alcançados e sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é apresentar em detalhes o controlador repetitivo complexo que é objeto de estudo neste trabalho. Primeiramente, será feita uma breve introdução sobre vetores espaciais aplicados a sistemas de controle, que servirá de suporte para a explicação da utilidade de um controle repetitivo complexo. Por fim, será apresentada a configuração proposta por (NETO, 2018), que serve de base para o desenvolvimento do aplicativo.

2.1 VETORES ESPACIAIS

Seja um conjunto de sinais trifásicos periódicos distorcidos e desbalanceados, é possível representá-los genericamente em coordenadas abc da seguinte forma:

$$\begin{cases} s_a = \sum_{h=0}^{\infty} s_a^{(h)}; \\ s_b = \sum_{h=0}^{\infty} s_b^{(h)}; \\ s_c = \sum_{h=0}^{\infty} s_c^{(h)}, \end{cases} \quad (2.1)$$

onde h representa a h -ésima componente harmônica das fases a , b e c . Em que:

$$s_i^{(h)} = S_i^{(h)} \cos(h\omega_1 t + \phi_i^{(h)}) + S_o^{(h)} \cos(h\omega_1 t + \phi_o^{(h)}), \quad (2.2)$$

para $i = a, b, c$. Onde $S_i^{(h)}$ e $\phi_i^{(h)}$ representam a amplitude da componente harmônica h e a fase dos sinais, respectivamente, e ω_1 representa a frequência fundamental. Entende-se o índice “o” como referencia à componente homopolar.

Como demonstrado em (NETO et al., 2018), é possível representar, de forma genérica e no referencial $\alpha\beta$, um vetor espacial referente a uma componente harmônica qualquer, de sequência positiva ou negativa, da seguinte forma:

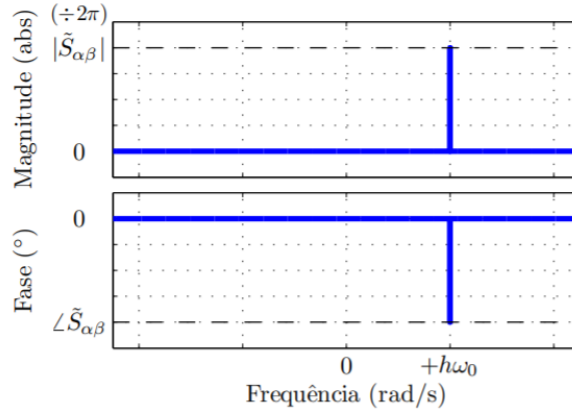
$$\vec{s}_{\alpha\beta}^{(h_s)} = \vec{S}_{\alpha\beta}^{(h_s)} e^{jh_s\omega_1 t}, \text{ onde } h_s = \begin{cases} h, & \text{para sequência positiva} \\ -h, & \text{para sequência negativa.} \end{cases} \quad (2.3)$$

Sabendo da ortogonalidade de funções exponenciais complexas (para $h_s \in \mathbb{Z}$), é possível representar um conjunto de sinais trifásicos periódicos qualquer a partir da soma de vetores de componentes harmônicas $\vec{s}_{\alpha\beta}^{(h_s)}$.

A transformada de Fourier é útil nesta aplicação por fornecer informações importantes sobre o comportamento em frequência de um determinado vetor espacial. A Equação (2.4) mostra que a transformada de Fourier de um vetor espacial genérico é um impulso de amplitude $2\pi\vec{S}_{\alpha\beta}^{(h_s)}$ deslocado em $h_s\omega_1$ e a Figura 1 ilustra graficamente esse resultado.

$$\begin{aligned}\mathfrak{F}(\vec{s}_{\alpha\beta}^{(h_s)}) &= \int_{-\infty}^{\infty} \vec{S}_{\alpha\beta}^{(h_s)} e^{jh_s\omega_1 t} e^{-j\omega t} dt, \\ &= \vec{S}_{\alpha\beta}^{(h_s)} \int_{-\infty}^{\infty} e^{-j(\omega - h_s\omega_1)t} dt, \\ \mathfrak{F}(\vec{s}_{\alpha\beta}^{(h_s)}) &= 2\pi \vec{S}_{\alpha\beta}^{(h_s)} \delta(\omega - h_s\omega_1).\end{aligned}\tag{2.4}$$

Figura 1: Resposta em frequência de $\mathfrak{F}(\vec{s}_{\alpha\beta}^{(h_s)})$ para h_s positivo.



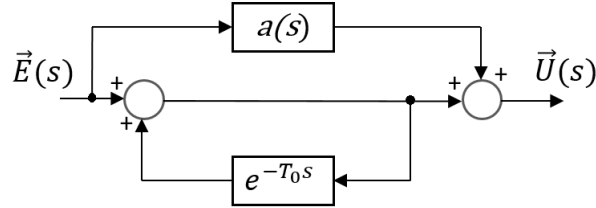
Fonte: (NETO et al., 2018).

Sabendo que as componentes harmônicas de sequência positiva são representadas por frequências positivas e que componentes harmônicas de sequência negativa são representadas por frequências negativas, é possível observar que a resposta em frequência apresentada acima não é simétrica em relação ao eixo das ordenadas. Em outras palavras, com o uso de vetores espaciais, é possível identificar as componentes harmônicas de sequência positiva e sequência negativa de um dado sinal.

2.2 CONTROLADOR REPETITIVO COMPLEXO

Inicialmente proposto por (HARA; OMATA; NAKANO, 1985), o controlador repetitivo visa garantir o seguimento de referências periódicas ou a rejeição de distúrbios periódicos. Fundamentado pelo princípio do modelo interno (FRANCIS; WONHAM, 1975), o controlador repetitivo apresenta o que pode ser chamado de núcleo gerador de sinais periódicos (NETO, 2018), que funciona a partir de um termo de atraso temporal de T_0 segundos na malha de controle representando o período fundamental dos sinais a serem seguidos ou rejeitados. Assim, existe um ganho infinito para uma frequência selecionada $\omega_0 = \frac{2\pi}{T_0} \text{ rad/s}$ e para todas suas harmônicas $\omega = k \omega_0 \text{ rad/s}$, onde $k \in \mathbb{N}$. Posto isso, foi proposto por (HARA; OMATA; NAKANO, 1985) o acréscimo de uma segunda malha direta com ganho $a(s)$ para tornar possível a alocação de zeros. Toda essa estrutura é apresentada na Figura 2, onde $E(s)$ e $U(s)$ são, respectivamente, os sinais de entrada e saída do controlador.

Figura 2: Estrutura básica do controlador repetitivo.



Fonte: (HARA; OMATA; NAKANO, 1985), Adaptado.

De acordo com o demonstrado em (NETO, 2018), é possível definir a configuração utilizada para implementação do RC ao selecionar um valor entre 0 e 1 para o parâmetro $a(s)$, porém é preciso atenção pois o domínio de estabilidade do sistema depende disso. Sabendo do comportamento de um controlador repetitivo, observa-se a possibilidade de alterar a família de componentes harmônicas em que o RC atua a partir da variação do termo T_0 do tempo de atraso, sem que seja alterada a frequência do sinal de referência.

Diversos trabalhos apresentam soluções baseadas em RC para obtenção de ganho infinito nas harmônicas de uma família genérica do tipo $H = \{nk \pm m \mid k \in \mathbb{Z} \text{ e } m, n \text{ constantes e definidos pelo usuário}\}$ (NETO, 2018), em vez de um ganho infinito em todas as frequências múltiplas da fundamental. Quando o sinal de referência do sistema de controle também possui harmônicos distribuídos segundo a família H , é clara a vantagem da utilização desse tipo de controle em relação ao RC convencional, já que o controlador

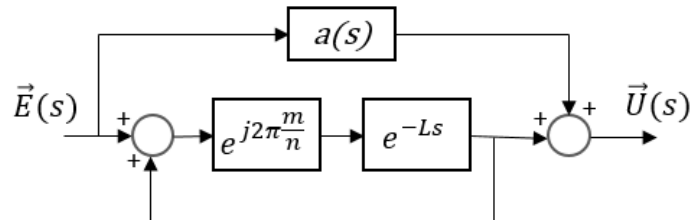
obtido possui ordem reduzida em relação a esse. Porém, em casos onde o sinal de referência possui componentes harmônicas de sequência positiva diferentes das de sequência negativa (não simétrico em relação ao zero), por exemplo o conjunto $h_s = \{-7, -4, -1, 2, 5, 8\}$, o uso de controladores $nk \pm m$ não se faz mais tão eficiente.

Por isso, controladores complexos são uma boa alternativa por conseguir reduzir ainda mais a ordem do controlador, visto que eles operam de forma que é obtida uma resposta em frequência assimétrica em relação ao eixo das ordenadas ($H = \{nk+m \mid k \in \mathbb{Z} \text{ e } m, n \text{ constantes e definidos pelo usuário}\}$).

Fundamentado em tudo que foi explicado anteriormente, uma estrutura de controlador de ordem e estrutura genérica foi proposta por (NETO, 2018). A Figura 3 mostra o seu diagrama de blocos, onde $E(s)$ e $U(s)$ são, respectivamente, os sinais de entrada e saída do controlador e:

- **Ganho “ $a(s)$ ”:** diretamente relacionado com a alocação dos zeros do controlador e com a estabilidade do sistema;
- **Atraso “ e^{-Ls} ”:** atraso genérico de $L = \frac{T_0}{n}$, onde T_0 é o período da componente fundamental e o parâmetro n está relacionado à periodicidade da família de componentes harmônicas a ser controlada ($H_s = \{nk \mid k \in \mathbb{Z}\}$);
- **Ganho complexo “ $e^{j2\pi \frac{m}{n}}$ ”:** em cascata com o atraso genérico, permite transladar a resposta em frequência do controlador, de forma que, a partir da escolha do parâmetro m , a família de componentes harmônicas a ser controlada torna-se $H_s = \{nk + m \mid k \in \mathbb{Z}\}$, onde $0 \leq m \leq (n - 1)$.

Figura 3: Diagrama de blocos do controlador proposto por (NETO, 2018).

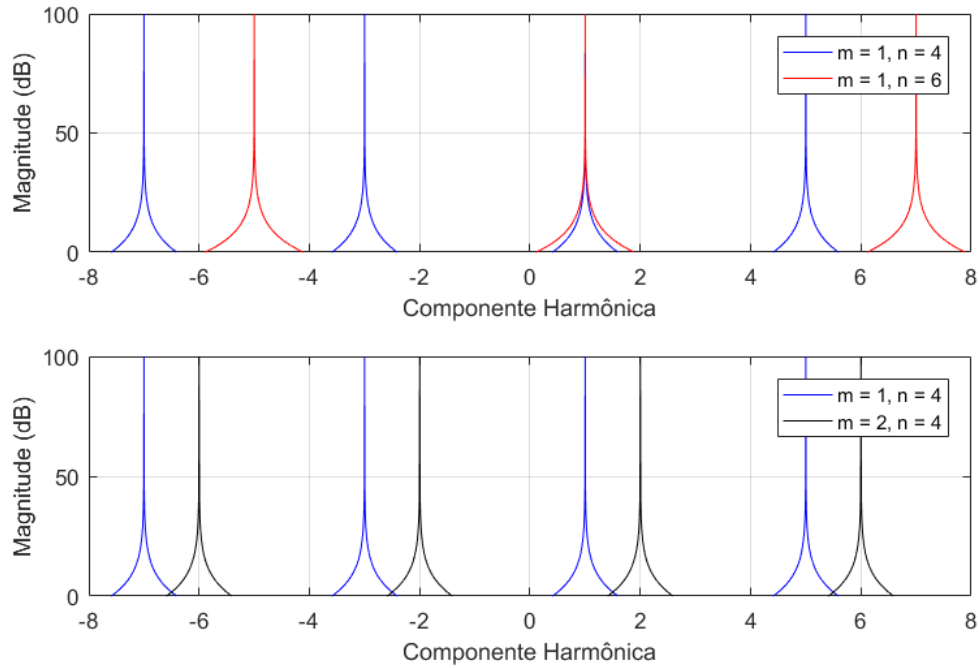


Fonte: (NETO, 2018), Adaptado.

Para deixar claro o funcionamento do ganho complexo em cascata com o atraso genérico, é mostrada na Figura 4 a resposta em frequência do controlador com diferentes

valores para m e n . É possível observar que o parâmetro m seleciona uma componente harmônica desejada (que define o eixo de simetria da resposta em frequência) e que o parâmetro n define a periodicidade da família de harmônicos onde é aplicado ganho elevado.

Figura 4: Resposta em frequência do controlador para diferentes valores de m e n .



Fonte: Próprio Autor.

2.3 ANÁLISE DE ESTABILIDADE

A estabilidade é um ponto de extrema importância quando se trata de análise e projeto de sistemas de controle e é possível defini-la de diferentes maneiras. Este trabalho se concentrará no conceito de BIBO-estabilidade (*bounded input, bounded output*), que diz respeito a um sistema que produz uma saída limitada em amplitude ao ser excitado por uma entrada limitada em amplitude.

2.3.1 Norma \mathcal{H}_∞ e Teorema do Ganho Pequeno

O Teorema do Ganho Pequeno será responsável pela avaliação da estabilidade entrada-saída do sistema. Antes, porém, é necessário que o conceito de norma \mathcal{H}_∞ seja definido. Seja um sistema $H(s)$, sua norma \mathcal{H}_∞ pode ser determinada, no espaço frequencial (FRANCIS, 1987)

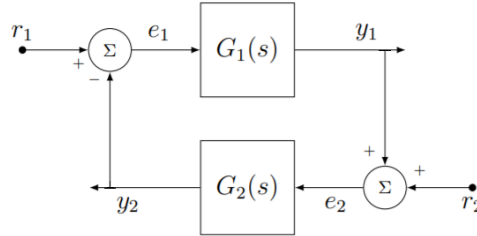
$$\|H(s)\|_\infty \triangleq \sup_{\omega \in \mathbb{R}} |H(j\omega)|. \quad (2.5)$$

O que significa que a norma \mathcal{H}_∞ é indicada através do máximo valor absoluto da função de transferência do sistema. Em sistemas de controle, ela equivale ao máximo ganho de um sistema, considerando suas entradas limitadas em energia (NETO, 2020).

O Teorema do Ganho Pequeno fornece uma condição suficiente para caracterizar a estabilidade de um sistema em malha fechada. Seja o diagrama de blocos mostrado na Figura 5, com os sinais de entrada r_1 e r_2 , os sinais de saída y_1 e y_2 , os sinais de erro e_1 e e_2 e os sistemas $G_1(s)$ e $G_2(s)$ estáveis, o critério de Nyquist garante a estabilidade do sistema se, e somente se, o diagrama de Nyquist de $G_1(s)G_2(s)$ não contém o ponto $+1$. Logo, a condição abaixo precisa ser respeitada:

$$\begin{aligned} 1 - G_1(j\omega)G_2(j\omega) &\neq 0, \\ \sup_{\omega \in \mathbb{R}} |G_1(j\omega)G_2(j\omega)| &< 1. \end{aligned} \quad (2.6)$$

Figura 5: Diagrama de blocos para o estudo através do Teorema do Ganho Pequeno.



Fonte: (NETO, 2020).

Em termos da norma \mathcal{H}_∞ , a condição da Equação (2.6) pode ser reescrita pela condição suficiente:

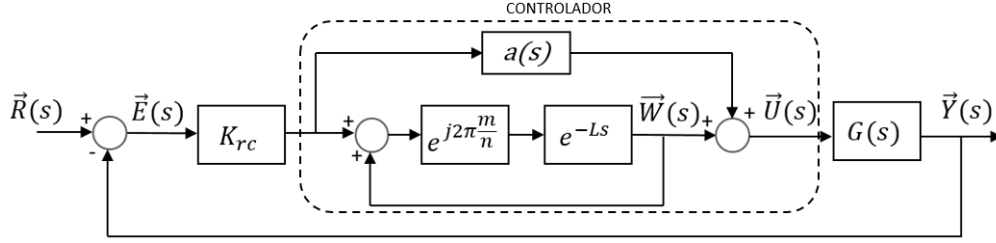
$$\begin{aligned} \|G_1(j\omega) G_2(j\omega)\|_\infty &< 1, \\ \|G_1(j\omega) G_2(j\omega)\|_\infty &< \|G_1(j\omega)\|_\infty \|G_2(j\omega)\|_\infty < 1, \\ \|G_1(j\omega)\|_\infty \|G_2(j\omega)\|_\infty &< 1. \end{aligned} \quad (2.7)$$

Logo, para definir a estabilidade de um dado sistema, é necessário avaliar se as funções $G_1(s)$ e $G_2(s)$ são limitadas (em amplitude e em energia, por exemplo) e se a condição da Equação (2.7) é respeitada.

2.3.2 RC Complexo proposto em (NETO, 2020)

Agora, o objetivo é aplicar o Teorema do Ganho Pequeno e determinar a estabilidade de um sistema que utiliza o controlador repetitivo apresentado na Figura 3. A Figura 6 apresenta uma aplicação genérica com esse controlador, onde $R(s)$, $E(s)$ e $Y(s)$ são, respectivamente, os sinais de referência, de erro e de saída. A planta do sistema é representada por $G(s)$, o ganho K_{rc} representa um ganho que pode aumentar ou diminuir a influência do erro no sistema e, no controlador, o parâmetro $a(s)$ é considerado como constante. Os sinais $U(s)$ e $W(s)$ se referem, respectivamente, à saída e ao ramo de realimentação do controlador.

Figura 6: Sistema de controle genérico com o RC Complexo proposto em (NETO, 2020).

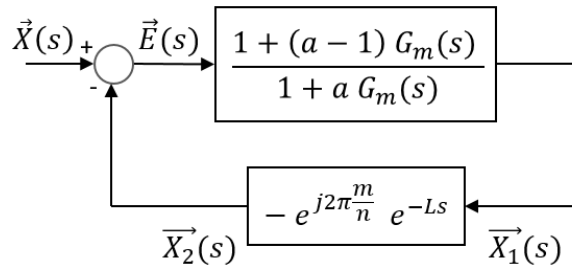


Fonte: (NETO, 2020), Adaptado.

Para que o Teorema do Ganho Pequeno seja aplicado, é necessário rearranjar o diagrama de blocos para uma forma equivalente, mas que esteja mais próxima àquela que o teorema foi apresentado. Como demonstrado em (NETO, 2020), é possível chegar à estrutura apresentada na Figura 7, onde têm-se:

- $\vec{X}(s) = \frac{\vec{D}_e(s)}{1 + a G_m(s)}$,
- $\vec{D}_e(s) = (1 - e^{j2\pi \frac{m}{n}} e^{-Ls}) [\vec{R}(s) - \vec{Y}(s)] - \vec{W}(s) G(s)$,
- $G_m(s) = K_{rc} G(s)$.

Figura 7: Sistema equivalente àquele com o RC Complexo Proposto.



Fonte: (NETO, 2020), Adaptado.

O trabalho de (NETO, 2020) também apresenta um desenvolvimento para provar a estabilidade do sistema a partir deste diagrama de blocos e das condições necessárias do Teorema do Ganho Pequeno. Ao final de toda análise apresentada pelo autor, duas condições precisam ser satisfeitas:

$$(i) \frac{G_m(s)}{1 + a G_m(s)} \text{ precisa possuir norma } \mathcal{H}_\infty, \quad (2.8)$$

$$(ii) \left\| \frac{1 + (a - 1) G_m(s)}{1 + a G_m(s)} \right\|_\infty < 1. \quad (2.9)$$

É importante deixar claro que essas condições, válidas para sistemas em tempo contínuo, também são válidas para o sistema em tempo discreto.

De agora em diante, será considerado que a planta escolhida é tal que a condição da Equação (2.8) é sempre válida. Assim, somente é necessário testar se os valores de a , K_{rc} e $G(s)$ tornam a Inequação (2.9) verdadeira. Ao definir um valor para o parâmetro a , o controlador garantirá a estabilidade em uma determinada região do plano complexo e é preciso que o diagrama de Nyquist de $G_m(s)$ esteja contido nessa região.

A fim de simplificar a Inequação (2.9), os termos serão escritos de forma genérica e, com isso, a inequação será desmembrada. Portanto, o parâmetro a torna-se $Re\{a\} + j Im\{a\}$ e o parâmetro $G_m(j\omega)$ torna-se $Re\{G_m(j\omega)\} + j Im\{G_m(j\omega)\}$. É provado por (NETO, 2020) que é possível encontrar o domínio de estabilidade do sistema a partir da seguinte inequação:

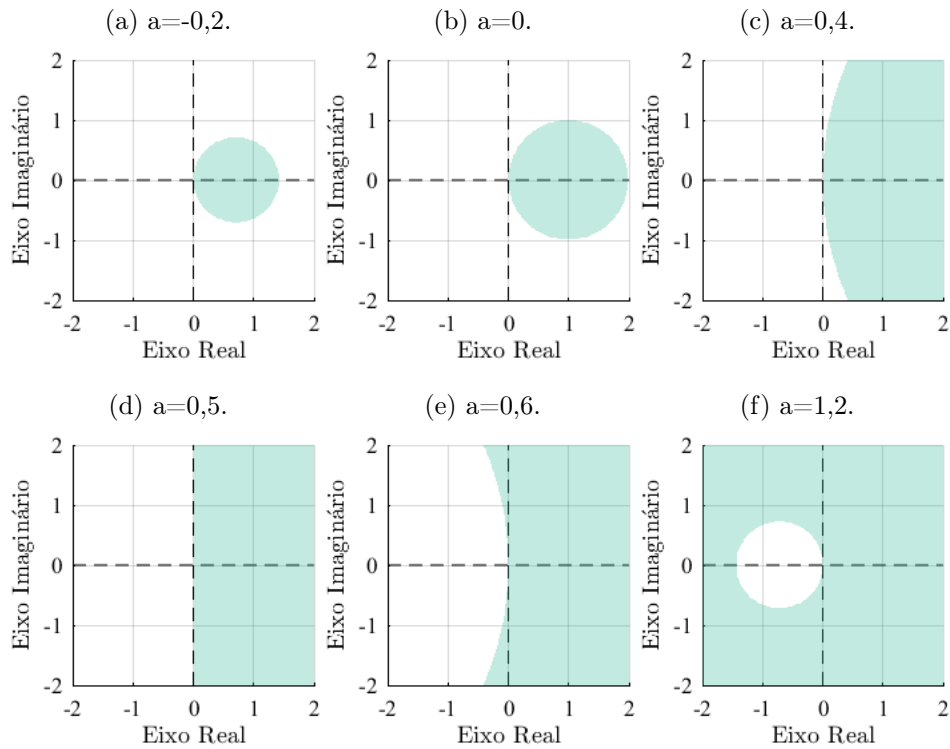
$$(1 - 2Re\{a\}) Re\{G_m(j\omega)\}^2 - 2Re\{G_m(j\omega)\} + (1 - 2Re\{a\}) Im\{G_m(j\omega)\}^2 < 0 \quad (2.10)$$

É possível observar que, independente da escolha, apenas a parte real do parâmetro a influenciará na estabilidade do sistema. Portanto, é desnecessário que um valor complexo seja escolhido para ele. Ainda, é possível ratificar a afirmação feita na Seção 2.2, a qual relaciona diretamente a estabilidade do sistema ao parâmetro a do controlador. A Figura 8 mostra como esse parâmetro influencia no domínio de estabilidade (em verde).

Para comprovar a afirmação de que o controlador garante a estabilidade de um sistema desde que o diagrama de Nyquist da planta a ser controlada esteja contido no seu domínio de estabilidade, será tomado o seguinte exemplo onde:

$$\left\{ \begin{array}{l} K_{rc} = 3, \\ G_m(s) = \frac{2s+1}{2s+5}, \\ m = 1, \\ n = 3. \end{array} \right.$$

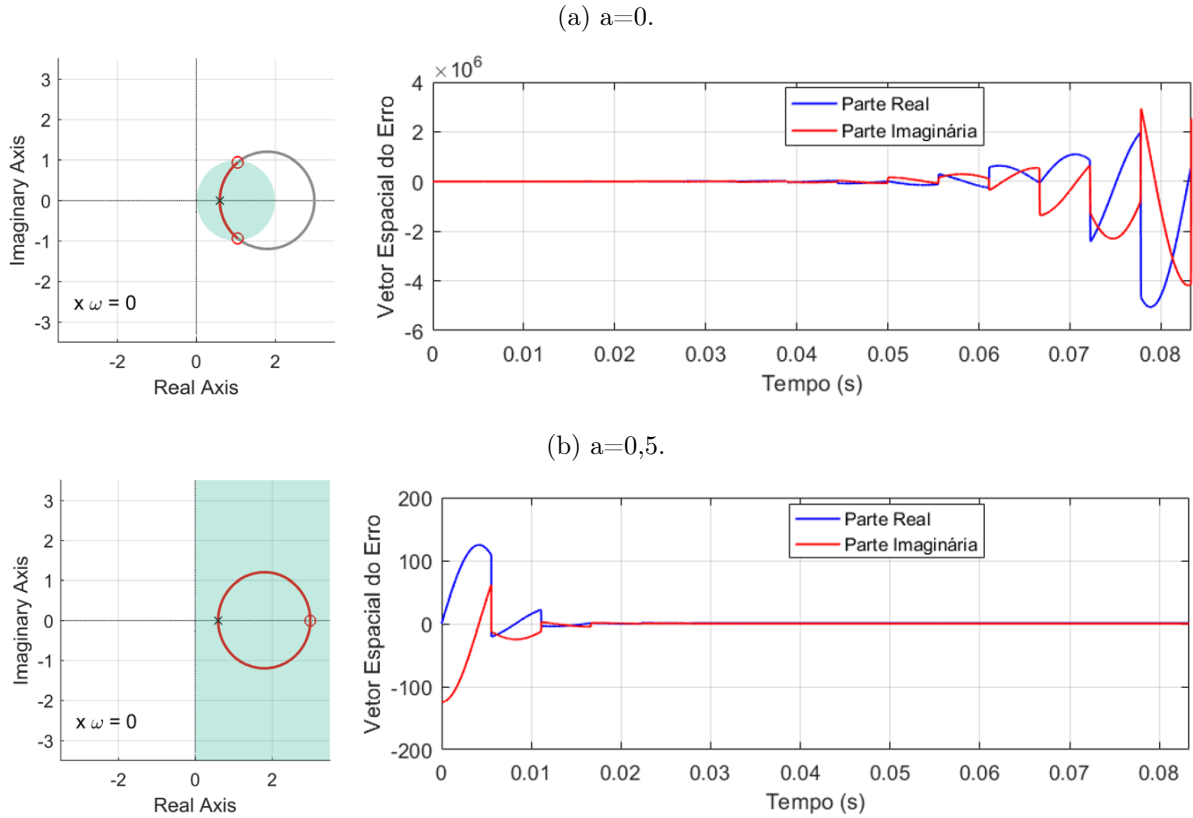
Figura 8: Exemplo de domínio de estabilidade (em verde) para um sistema com RC Complexo proposto em (NETO, 2020) com diversos valores de a .



Fonte: (NETO, 2020), Adaptado.

A ilustração se dará com a utilização de dois valores diferentes para o parâmetro a do controlador: $a = 0$ e $a = 0,5$. Os resultados são apresentados na Figura 9, onde do lado esquerdo é apresentado o domínio de estabilidade para um dado valor de a , juntamente com o diagrama de Nyquist de $G_m(s)$ e do lado direito é apresentado o vetor espacial do erro do respectivo sistema. No caso (a), parte do diagrama de Nyquist de $G_m(s)$ se localiza fora do domínio de estabilidade do controlador, gerando uma resposta instável; no caso (b), o diagrama de Nyquist de $G_m(s)$ se localiza completamente dentro do domínio de estabilidade do controlador, gerando uma resposta estável.

Figura 9: Resultado de simulação do sistema exemplo com o RC Complexo proposto em (NETO, 2020) para diferentes valores do parâmetro a .



Fonte: Próprio Autor.

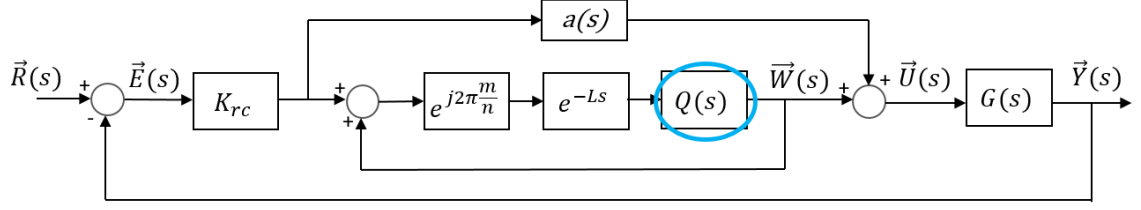
2.3.3 RC Complexo Modificado

Sabendo que em aplicações de controle em eletrônica de potência é muito comum encontrar funções de transferência estritamente próprias, ou seja, aquelas onde o grau do denominador é maior do que o do numerador, é possível observar uma limitação na estrutura do controlador utilizada até então. Seja $F(s)$ uma função estritamente própria, sabe-se que $F(j\infty) = 0$. Logo, nota-se que este tipo de planta nunca poderia ser controlada por um RC Complexo, já que o ponto $(0,0)$ não pertence ao domínio de estabilidade do controlador, de acordo com a Equação (2.10).

As condições de estabilidade de um RC Complexo podem se tornar menos rígidas a partir da redução do ganho do laço em que seu gerador de sinal periódico encontra-se. Para tal, duas abordagens são possíveis: adicionar uma atenuação constante ou um filtro passa-baixa (NETO, 2020). Seja $Q(s)$ a função de transferência responsável por esta tarefa, uma nova estrutura é proposta e uma aplicação é mostrada na Figura 10.

Agora, é necessário reaplicar o Teorema do Ganho Pequeno para encontrar as novas

Figura 10: Sistema de controle genérico com o RC Complexo Modificado.



Fonte: (NETO, 2020), Adaptado.

inequações referentes ao controlador e, a partir disso, refazer a análise de estabilidade. Seguindo o mesmo raciocínio do caso anterior, o trabalho de (NETO, 2020) mostra que as novas condições que precisam ser satisfeitas são:

$$(i) \frac{G_m(s)}{1 + a G_m(s)} \text{ precisa possuir norma } \mathcal{H}_\infty, \quad (2.11)$$

$$(ii) \left\| Q(s) \cdot \frac{1 + (a - 1) G_m(s)}{1 + a G_m(s)} \right\|_\infty < 1. \quad (2.12)$$

Da mesma forma que foi feito anteriormente, a seguinte inequação agora é responsável por definir o domínio de estabilidade do sistema:

$$[X^2 + Y^2] \cdot f_1(a, |Q(j\omega)|) + X \cdot f_2(a, |Q(j\omega)|) < 1 - |Q(j\omega)|^2 \quad (2.13)$$

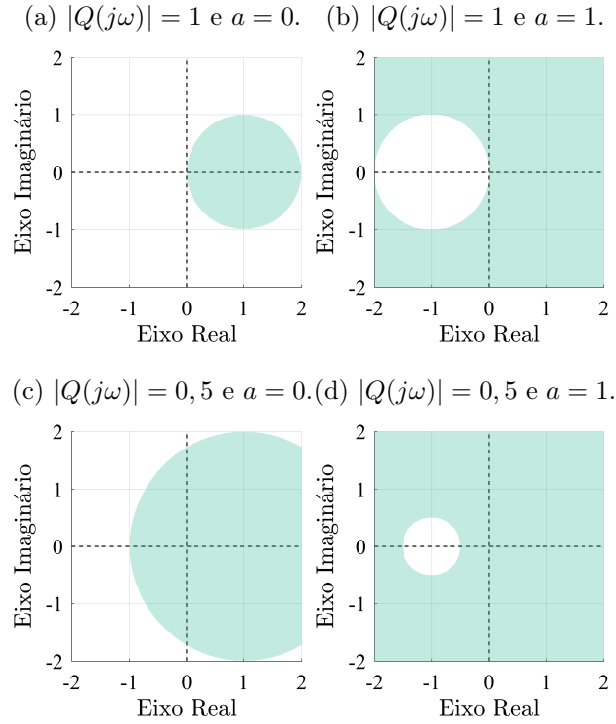
com:

$$\begin{cases} X = \text{Re}\{G_m(j\omega)\}, \\ Y = \text{Im}\{G_m(j\omega)\}, \\ f_1(a, |Q(j\omega)|) = a^2 |Q(j\omega)|^2 - 2a |Q(j\omega)|^2 - a^2 + |Q(j\omega)|^2, \\ f_2(a, |Q(j\omega)|) = 2a |Q(j\omega)|^2 - 2|Q(j\omega)|^2 - 2a. \end{cases}$$

Onde é possível observar que ela se torna igual à Inequação (2.10) quando $Q(s) = 1$.

Uma das possíveis escolhas para a função de transferência $Q(s)$ é que ela tenha um valor constante e menor do que 1. Quanto menor esse valor, maior se torna o domínio de estabilidade do sistema. Esse comportamento pode ser observado com ajuda da Figura 11, onde são mostrados os domínios de estabilidade para sistemas com o RC Complexo proposto em (NETO, 2020) ($|Q(j\omega)| = 1$; $a = 0$ e $a = 1$) e com o RC Complexo Modificado ($|Q(j\omega)| = 0,5$; $a = 0$ e $a = 1$). Nota-se que a origem do plano complexo pertence ao domínio para ambas as configurações do RC Complexo Modificado.

Figura 11: Comparação entre os domínios de estabilidade em sistemas com o RC Complexo proposto em (NETO, 2020) e com o RC Complexo Modificado.



Fonte: Próprio Autor.

Porém, é provado por (NETO, 2020) que, no caso onde $|Q(j\omega)|$ está com um valor fixo, mesmo que as condições (2.11) e (2.12) sejam respeitadas para um determinado valor de $|Q(j\omega)|$ e o sistema permaneça estável, o vetor espacial do erro não diverge, mas também não converge para zero. Uma forma de melhorar esse comportamento é fazer de tal maneira que o valor de $|Q(j\omega)|$ diminua continuamente com o aumento da frequência, o que pode ser traduzido por implementar um filtro passa-baixa para o parâmetro $Q(j\omega)$.

A segunda opção de estrutura para a função de transferência $Q(s)$, e a que será considerada de agora em diante, é que ela tenha o comportamento de um filtro passa-baixa. A ideia se resume em aumentar o domínio de estabilidade enquanto o vetor espacial de erro é mantido limitado em energia. Independente das características do filtro, a condição de estabilidade (2.12) precisa ser válida para todo ω . Isso pode ser tratado com a hipótese de um filtro com ganho unitário em sua banda de passagem, atenuação constante em sua banda de rejeição e frequência de corte $\omega_c = 2\pi f_c$ (NETO, 2020).

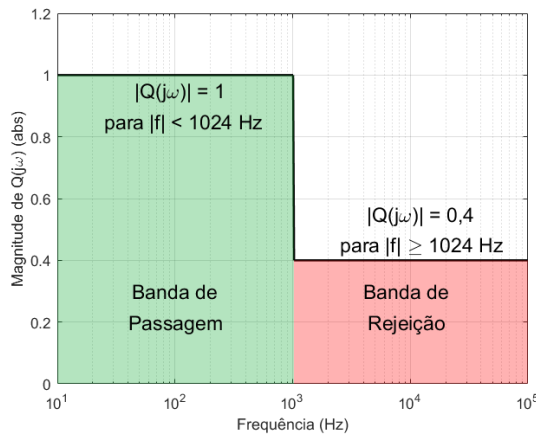
Um exemplo é apresentado em (NETO, 2020) e será usado para ilustrar o que está sendo dito. Seja um sistema onde é utilizado o RC Complexo Modificado com $a = 0$ e

$$G_m(s) = \frac{550s^2 + 3,459 \cdot 10^7 s + 2,171 \cdot 10^9}{s^3 + 2628s^2 + 5,911 \cdot 10^7 + 3,635 \cdot 10^{10}}. \quad (2.14)$$

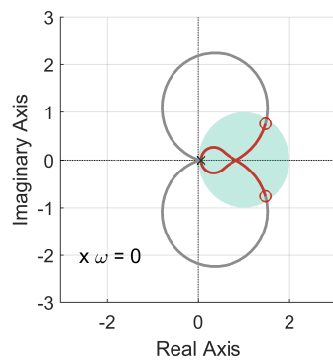
Quando $|Q(j\omega)| = 1$, o diagrama de Nyquist de $G_m(j\omega)$ pertence ao domínio de estabilidade do controlador apenas até $f \approx 1024 \text{ Hz}$, o que é, portanto, a frequência de corte máxima aceitável. Agora, é necessário que o valor de $|Q(j\omega)|$ seja reduzido de tal maneira que o domínio de estabilidade do controlador contenha também o restante do diagrama Nyquist de $G_m(j\omega)$, o que acontece com $|Q(j\omega)| = 0,4$. A Figura 12 mostra o filtro passa-baixa ideal referente ao exemplo (Figura 12a) e também os domínios de estabilidade do controlador juntamente com o diagrama de Nyquist de $G_m(s)$ para a banda de passagem (Figura 12b) e para a banda de rejeição (Figura 12c).

Figura 12: Estabilidade do sistema com RC Complexo Modificado ($a = 0$) e com $Q(s)$ como um filtro passa-baixa.

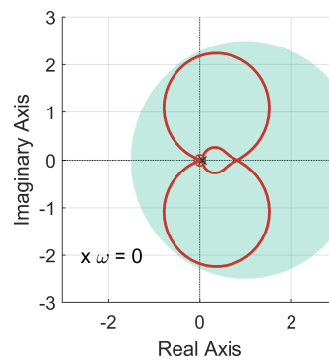
(a) Resposta em frequência do filtro com $f_c = 1024 \text{ Hz}$ e ganho de 0,4 na banda de rejeição.



(b) Domínio de estabilidade e diagrama de Nyquist de $G_m(s)$ para $|f| < 1024$.



(c) Domínio de estabilidade e diagrama de Nyquist de $G_m(s)$ para $|f| \geq 1024$.



Fonte: (NETO, 2020), Adaptado.

2.4 CONCLUSÃO

Tendo em vista que um sinal trifásico pode ser representado a partir da utilização de vetores espaciais e que com isso é possível identificar componentes harmônicas de sequência positiva e sequência negativa desse sinal, o uso de controladores complexos corresponde a uma opção conveniente para sistemas de controle com referências trifásicas.

Apesar de garantir a estabilidade para uma variedade de plantas, o RC Complexo proposto em (NETO, 2020) não atende a um critério importante que faz parte de um campo onde a sua utilização é bastante conveniente. Dessa forma, é apresentado ainda em (NETO, 2020) uma modificação ao mesmo, onde um novo termo é adicionado com o objetivo de tornar o uso do controlador mais abrangente.

Assim, mais um termo do controlador pode ser alterado a fim de encontrar a melhor configuração para uma determinada planta. Logo, na continuação deste trabalho será considerada a estrutura do RC Complexo Modificado, onde o dimensionamento de um filtro ideal, dada uma aplicação, fará parte da ferramenta criada.

3 DESENVOLVIMENTO DA FERRAMENTA

Após toda fundamentação teórica, chega-se ao real objeto deste trabalho. Neste capítulo, é apresentada a ferramenta criada para dar suporte a projetos que utilizam o RC Complexo Modificado. Agora, é necessário implementar algumas das equações mostradas anteriormente, de forma que o usuário tenha a liberdade de observar a influência dos parâmetros do controlador na estabilidade do sistema e de dimensionar valores ideais para cada aplicação. Primeiramente, será apresentado o motivo e como esta ferramenta foi criada. Então, serão mostradas suas interfaces e funcionalidades e, por fim, como foram implementados alguns de seus recursos.

Após a proposta de uma nova configuração de controlador, a criação de uma ferramenta de suporte de projetos de parâmetros desse controlador é uma forma de viabilizar mais projetos deste tipo e favorecer o seu uso e popularização. Apesar de inúmeras maneiras de criar aplicativos, seja diretamente com uma linguagem de programação ou seja a partir de *softwares* comerciais, o MATLAB foi escolhido para tal tarefa. Esta escolha se deu principalmente por dois motivos:

- **Utilização mundial:** o MATLAB é amplamente usado em todo o mundo, sobretudo em projetos de sistemas de controle, por causa de suas ferramentas de modelagem, simulação e análise de sistema, como é o caso do *Simulink*, o *Control System Designer*, *PID Tuning*, etc. Segundo (MATHWORKS, s.d.[b]), o MATLAB é utilizado em mais de 6500 universidades pelo mundo e em mais de 5000 aceleradoras, incubadoras e startups nas indústrias;
- **Compatibilidade:** Caso o usuário já possua modelagem de sistemas e projetos de controladores no MATLAB, é possível vinculá-los facilmente à ferramenta criada, poupando um retrabalho caso a implementação fosse feita em uma outra plataforma.

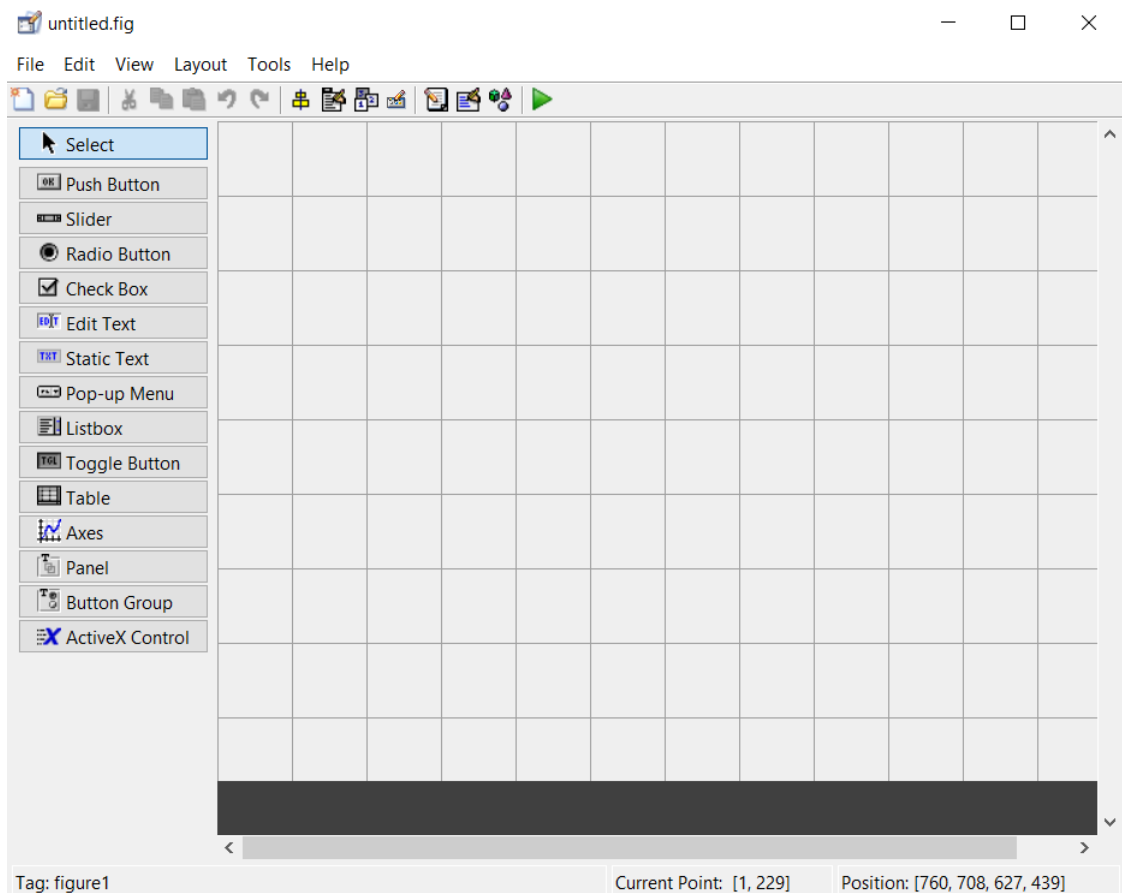
3.1 FERRAMENTAS DO MATLAB PARA CRIAÇÃO DE APLICATIVOS

No geral, o MATLAB disponibiliza duas ferramentas de criação de aplicativos: o GUIDE e o App Designer. Em algumas versões, apenas uma dessas ferramentas é disponibilizada, o GUIDE para versões mais antigas e o App Designer para versões mais

novas. Neste trabalho, foi utilizada a versão R2016a do MATLAB, que contém ambas as ferramentas. Logo, alguns critérios foram atribuídos para de fato optar por uma delas.

O GUIDE (*Graphical User Interface Development Environment*) está presente no MATLAB desde 1996 (MATLAB 5.0) e é um ambiente que permite a criação manual de interfaces gráficas de forma rápida e fácil. A Figura 13 mostra a página principal desta ferramenta, onde é possível escolher quais componentes (canto esquerdo) o usuário deseja adicionar à aplicação. Informações sobre as propriedades da janela, seus componentes e layout são salvas em um arquivo *.fig*. Associada à esta figura criada, o MATLAB cria automaticamente um arquivo de texto (arquivo *.m*), onde o usuário programa o comportamento da interface através de funções específicas de acordo com os componentes escolhidos. Por exemplo: se um usuário deseja que sua aplicação tenha um botão como agente de uma tarefa, primeiramente é preciso adicionar este componente no arquivo *.fig* e, só depois, no arquivo *.m*, define-se o que acontecerá quando este botão for pressionado.

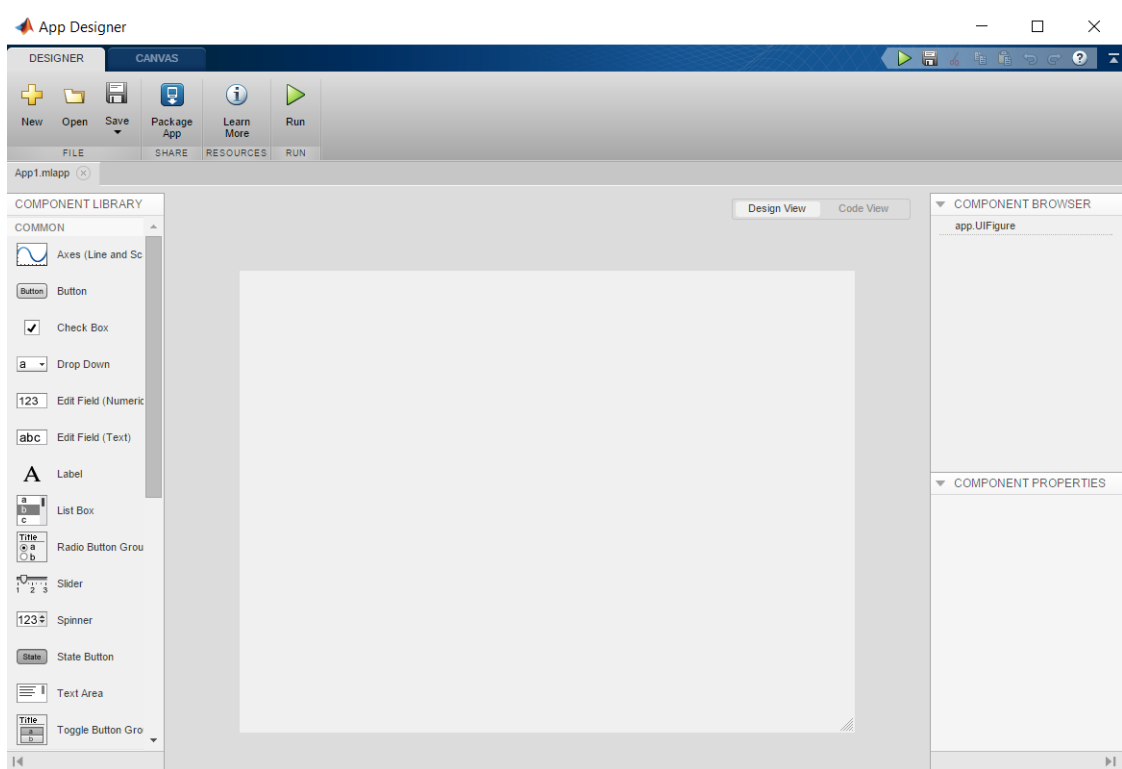
Figura 13: Página inicial da ferramenta GUIDE.



Fonte: Extraído do MATLAB R2016a.

O App Designer pode ser encontrado a partir da versão R2016a do MATLAB e, de maneira geral, funciona da mesma forma que o GUIDE, ou seja, primeiramente é preciso definir os componentes que vão compor o aplicativo e, com isso, definir o comportamento de cada um. A Figura 14 mostra a página principal desta ferramenta, a qual disponibiliza uma maior quantidade de componentes, uma experiência mais agradável ao usuário e outras particularidades que serão discutidas em breve.

Figura 14: Página inicial da ferramenta App Designer.



Fonte: Extraído do MATLAB R2016a.

Atualmente, a própria MathWorks (empresa responsável pelo MATLAB) desencoraja o uso do GUIDE e incentiva que as pessoas usem o App Designer para criar aplicativos (MATHWORKS, s.d.[a]). Apesar disso, o App Designer é uma ferramenta nova e que, mesmo com muito potencial, ainda apresenta algumas limitações. Por isso, no momento de criar o aplicativo objeto deste trabalho, para escolher qual ferramenta seria utilizada, foram definidos alguns critérios para observar qual das ferramentas oferecia melhores recursos para esta aplicação. Os critérios foram:

1. Compatibilidade entre diferentes versões do MATLAB;
2. Usabilidade;

3. Quantidade de arquivos para Interface e Código;
4. Componentes disponibilizados;
5. Disponibilidade de material didático.

A Tabela 1 mostra a comparação realizada entre as duas ferramentas, segundo os critérios supracitados, sendo que as explicações para cada critério são detalhadas em seguida:

1. O GUIDE é melhor neste quesito, pois, independente do aplicativo criado, são gerados apenas arquivos *.fig* e *.m*, os quais podem ser executados em qualquer versão do MATLAB. No App Designer, diferentemente, o tipo de arquivo criado obriga que seja especificada a versão que o mesmo será utilizado. E para versões anteriores à R2016a, não é possível abri-lo;
2. O App Designer tem uma aparência mais moderna e fornece uma melhor experiência ao usuário, ele gera de forma automática o código referente ao objeto escolhido e também é possível fazer de maneira que o aplicativo responda a diferentes tamanhos de tela, por exemplo, sem que uma lógica complexa de redimensionamento tenha que ser escrita. Ou seja, ele se torna mais fácil de usar do que o GUIDE;
3. Apesar de ambas as ferramentas funcionarem de maneira parecida, o App Designer não cria dois arquivos, como acontece no GUIDE. O App Designer possui um editor de código integrado, que dispensa que o trabalho seja feito em dois locais diferentes;
4. Por ser uma ferramenta mais atual, o App Designer oferece uma maior variedade de componentes, seja com uma nova categoria (exemplo: componentes de instrumentação industrial) ou seja com aqueles que eliminam um trabalho de tratamento de informação (exemplo: nele é oferecido campo de edição para texto e campo de edição para número, enquanto que no GUIDE apenas o primeiro é disponibilizado);
5. Como o GUIDE está presente no MATLAB há mais tempo que o App Designer e era, até então, a única ferramenta de criação de aplicativo disponível, é possível encontrar mais material para ele em fóruns na internet e em documentações do próprio MATLAB.

Tabela 1: Comparação entre ferramentas de criação de aplicativos do MATLAB.

Ferramenta	Critérios				
	1	2	3	4	5
GUIDE	x				x
App Designer		x	x	x	

Fonte: Próprio Autor.

Mesmo que o App Designer tenha se sobressaído na maioria dos critérios escolhidos, a ferramenta utilizada foi o GUIDE. Apesar do incentivo da MathWorks para a utilização do App Designer e também de ser mais atual e moderno, o que mais pesou na escolha de utilizar o GUIDE foi a quantidade de material de suporte disponível e também a possibilidade de executar o aplicativo criado em qualquer versão do MATLAB, visto que muitos usuários do MATLAB nem sempre acompanham as atualizações de versão e continuam utilizando versões mais antigas.

3.2 FUNCIONAMENTO

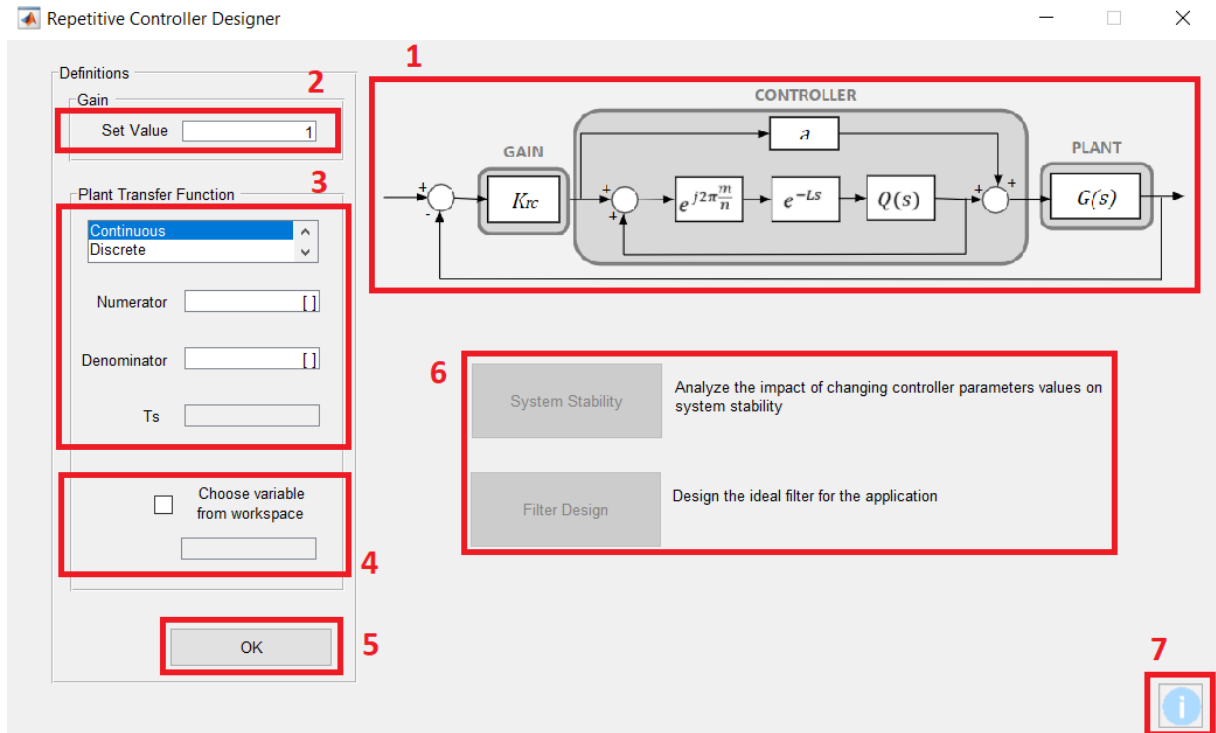
O aplicativo criado neste trabalho é dividido em três principais seções. A primeira seção é responsável pela determinação da planta que se deseja controlar, apenas depois dela é possível prosseguir para as outras duas seções. A segunda seção é responsável pela visualização da influência dos parâmetros do controlador na estabilidade do sistema e, na terceira seção, é possível dimensionar um filtro ideal para o parâmetro $Q(s)$ do controlador. A seguir serão dados mais detalhes para cada seção.

3.2.1 Definição da Planta

A primeira janela do aplicativo é mostrada na Figura 15. Em (1), é ilustrado o diagrama de blocos de um sistema genérico que utiliza o RC Complexo Modificado. O primeiro passo para o usuário é definir o ganho K_{rc} em (2) e qual a planta $G(s)$ a ser controlada. Para a planta, duas abordagens são possíveis: digitar manualmente os parâmetros da planta (3) ou escolher uma variável já definida no *workspace* do MATLAB (4). Apenas após o botão *OK* (5) ser pressionado, os dois botões presentes em (6) são habilitados, permitindo ao usuário a análise de estabilidade do sistema (botão *System Stability*) ou o dimensionamento de um filtro ideal (botão *Filter Design*). Apertar o botão em (7) abre uma página onde é possível encontrar informações a respeito da criação da

ferramenta e um pequeno resumo de seu funcionamento (Figura 16).

Figura 15: Seção responsável pela definição da planta a ser controlada.



Fonte: Próprio Autor.

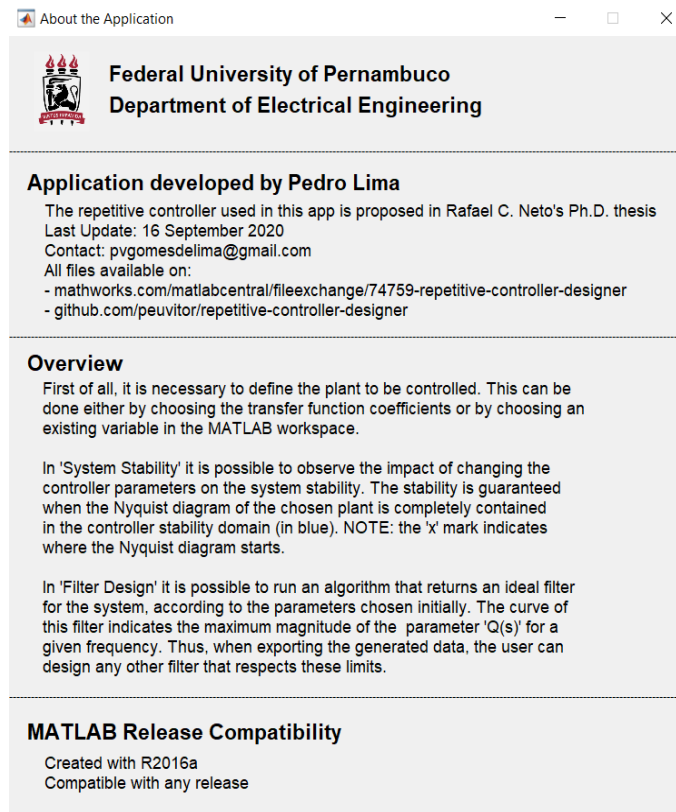
É importante comentar que foi necessária também a implementação de tratamento para possíveis erros de digitação e preenchimento. Para o caso onde o usuário digita os parâmetros da planta, testa-se:

- Se o ganho é um número real de dimensão 1x1;
- Se o numerador é um vetor linha;
- Se o denominador é um vetor linha;
- Numa função discreta, se o período de amostragem é um número real de dimensão 1x1.

Para o caso onde o usuário escolhe uma variável do *workspace*, testa-se:

- Se o ganho é um número real de dimensão 1x1;
- Se a variável escolhida existe e se ela é de alguns dos seguintes tipos: *tf* (função de transferência), *zpk* (zero-polo-ganho) ou *ss* (espaço de estados).

Figura 16: Seção responsável pelas informações a respeito da criação da ferramenta.



Fonte: Próprio Autor.

Só é permitido ao usuário continuar com a aplicação se todas as condições forem satisfeitas. Do contrário, aparece uma mensagem de erro que impossibilita a continuação até que o mesmo seja corrigido.

3.2.2 Parâmetros do Controlador

Após a definição da planta que se deseja controlar, uma das opções que o usuário possui é a de analisar a estabilidade do sistema em função dos parâmetros do controlador. Como já é de conhecimento, os parâmetros do controlador são: m , n , L , $a(s)$ e $Q(s)$ (Figura 10). Os três primeiros são constantes definidas pelo usuário as quais se referem às componentes harmônicas a serem controladas, o que não influenciará no desempenho do controlador em estabilizar o sistema. Logo, restam os parâmetros $a(s)$ e $Q(s)$ que interferem diretamente na estabilidade do sistema (Equação 2.13).

Como pode ser visto na Figura 17, são dadas duas possibilidades ao usuário para observar a influência desses dois parâmetros (1). É possível ou variar o valor dos parâmetros com a ajuda de *sliders* ou definir livremente valores reais. O impacto dessas variações

é imediatamente observado no gráfico ao lado (3), onde, além do domínio de estabilidade em verde, também é mostrado o diagrama de Nyquist da planta em cinza e, em vermelho, a frequência limite até onde o diagrama de Nyquist pertence ao domínio de estabilidade do sistema. Esta frequência é mostrada em (2), além de também mostrar se, com tal configuração, o sistema é estável ou não.

Caso alguma configuração seja útil ao usuário, em (4) é possível fazer uma captura de tal gráfico e escolher o local onde se deseja salvar a imagem gerada. São também disponibilizados alguns recursos do próprio MATLAB para que o usuário possa analisar o gráfico gerado (5), como aplicar ou remover zoom, ver os valores em determinados pontos das curvas e realizar translação do gráfico. Ainda, em (6), é possível ajustar os parâmetros dos *sliders* (passo e limites máximo e mínimo) e das caixas de textos disponíveis para escolha dos valores (resolução para arredondamento) - Figura 18(a) -, e também configurar os limites mínimos e máximos dos eixos do gráfico - Figura 18(b).

Para garantir nenhum comportamento inesperado, nesta seção, também foram implementados alguns tratamentos de erros. Para o caso onde o usuário altera os valores a partir dos *sliders*, nenhum tratamento é necessário, visto que a garantia de que os parâmetros do *slider* estão corretos garante o bom funcionamento do componente. Já para o caso onde o usuário digita os valores, testa-se se o que foi digitado é um número real de dimensão 1x1. Caso não seja, este é ignorado até que um número devidamente formatado seja digitado.

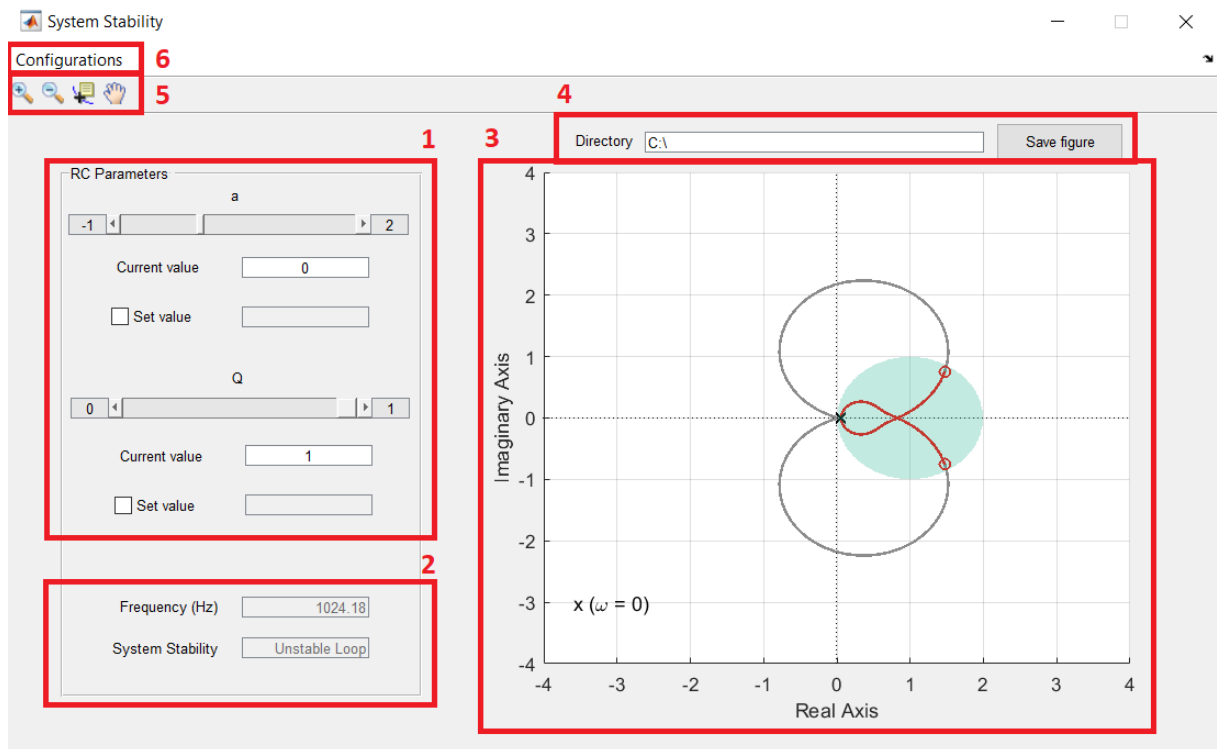
Nas configurações dos parâmetros do controlador e do gráfico, testa-se:

- Se os valores digitados são números reais de dimensão 1x1;
- Se o valor definido para o limite mínimo é menor do que o valor definido para o limite máximo;
- Se, para os *sliders*, o valor definido para o *step* é coerente (se $step < \frac{limite_max + limite_min}{2}$);
- Se o valor escolhido para precisão é um número maior do que 0.

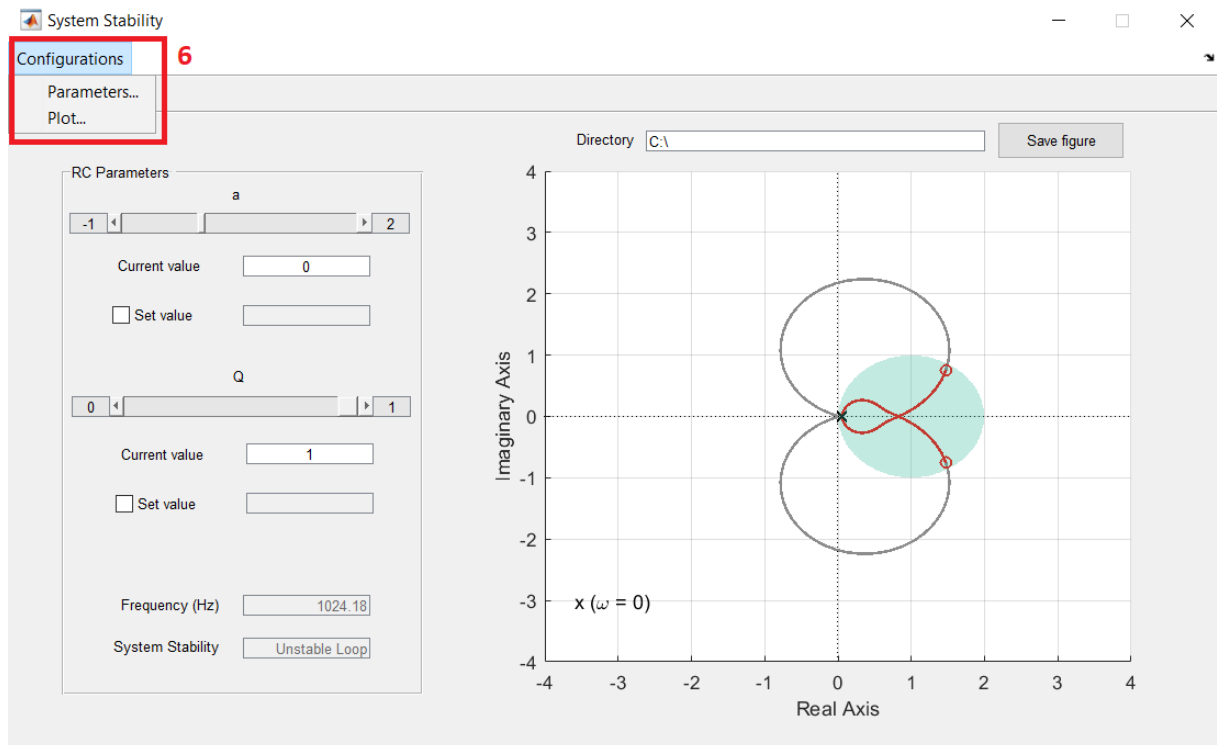
Além disso, no momento em que o usuário deseja salvar a imagem do gráfico, é testado se existe o caminho escolhido para tal. Se não existir, uma mensagem de erro aparece e o comando não é executado, até que seja alterado para uma pasta existente.

Figura 17: Seção para análise de estabilidade de acordo com os parâmetros do controlador.

(a)



(b)



Fonte: Próprio Autor.

Figura 18: Seções de configuração.

(a) Configurações dos parâmetros do controlador.

(b) Configurações dos eixos do gráfico.

Fonte: Próprio Autor.

3.2.3 Dimensionamento do Filtro

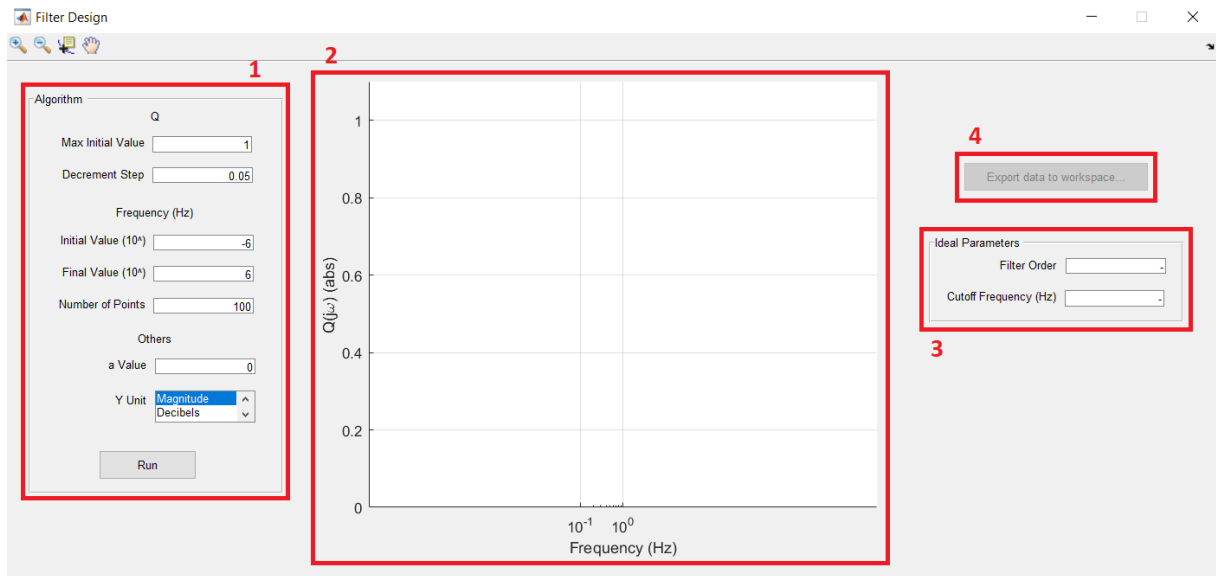
A outra opção após a definição da planta é a de dimensionamento de um filtro ideal para o parâmetro $Q(s)$. Nesta seção, o objetivo é encontrar a ordem e a frequência de corte para a criação de um filtro FIR ideal.

O algoritmo para esse dimensionamento se resume em, dado um valor de a constante, incrementar o valor de frequência e observar se, com o atual valor de $Q(s)$, a Inequação (2.13) é verdadeira. Quando não for, o valor de $Q(s)$ é decrementado numa taxa constante até que a inequação volte a ser verdadeira. Isso é feito até o último valor de frequência escolhido. Por isso, além de ser necessário conhecer a planta que se deseja controlar, outros parâmetros são requeridos, como mostra a Figura 19.

Para executar o algoritmo (1), é requerido:

- O valor inicial e o decremento desejado para o parâmetro $Q(s)$;
- Os valores inicial e final para a frequência e quantidade de pontos entre os extremos;

Figura 19: Seção para dimensionamento do filtro ideal para uma dada aplicação.



Fonte: Próprio Autor.

- O valor para o parâmetro a ;
- Modo de exibição do eixo y: magnitude ou decibels.

A curva resultante do algoritmo é apresentada em (2) e se refere ao comportamento do filtro ideal a ser utilizado em tal aplicação. Em (3), são mostradas a ordem e a frequência de corte correspondentes. Por fim, ainda é possível extrair os pontos que geraram o gráfico, além dos valores de ordem e frequência de corte do filtro, ao clicar no botão *Export data to workspace...* (4).

3.3 DETALHES DE IMPLEMENTAÇÃO

Após mostrar as interfaces da ferramenta criada e explicar como a mesma opera, serão dados alguns detalhes a respeito da sua implementação no MATLAB.

3.3.1 Estrutura da Ferramenta

A criação de toda a ferramenta em um único arquivo se tornou inviável em um determinado momento, por isso, ela foi dividida em diversos arquivos interligados. São eles:

- `main.m`: responsável pela primeira janela, onde é definida a planta a ser controlada;

- `about.m`: responsável pela janela que contém informações sobre a ferramenta;
- `controlador.m`: responsável pela janela da análise de estabilidade em função dos parâmetros do controlador;
 - `config_plot_control.m`: responsável pela janela de configuração dos eixos do gráfico;
 - `config_slider.m`: responsável pela janela de configuração dos parâmetros dos *sliders*;
 - `controlador_salvar.m`: responsável pela rotina que salva uma captura do gráfico;
 - `plotar.m`: responsável pela rotina de criação do gráfico (domínio de estabilidade em conjunto com diagrama de Nyquist);
- `filtro.m`: responsável pela janela do dimensionamento de um filtro ideal;
- `inequacao.m`: função criada com a implementação da Inequação (2.13).

3.3.2 Compartilhamento de Dados entre Arquivos

Sabendo da estrutura da ferramenta, por vezes uma informação adquirida em uma seção é necessária em uma outra. Por exemplo, a planta que se deseja controlar é definida logo na primeira janela do aplicativo, mas é necessária em todas as outras.

A estratégia utilizada para compartilhar dados foi a de associar dados a um componente específico usando a função `setappdata`, onde mais tarde é possível acessá-los usando a função `getappdata` (MATHWORKS, s.d.[c]). Segue abaixo um exemplo do que foi feito:

```
% Em main.m é criada uma estrutura que armazenará os dados da planta escolhida
data = struct('ganho', 0, 'num', [], 'den', [], 'ts', 0);

% Associa-se a estrutura criada à figura que compõe a janela principal (fig_main), onde
'PlantData' é o nome identificador escolhido
setappdata(handles.fig_main, 'PlantData', data);

%%%%

% Ainda em main.m, após a definição da planta, os valores de 'ganho', 'num', 'den' e 'ts'
definidos pelo usuário são salvos na estrutura criada

%%%%
```

```

% Primeiro recupera-se os dados salvos no objeto fig_main com nome 'PlantData'

    PlantData = getappdata(handles.fig_main, 'PlantData');

% Salva por cima os dados que foram obtidos através do usuário

    PlantData.ganho = ganho;

    PlantData.num = num;

    PlantData.den = den;

    PlantData.ts = ts;

% Armazena o conteúdo no objeto fig_main com nome 'PlantData'

    setappdata(handles.fig_main, 'PlantData', PlantData);

%%%%

% Agora, os parâmetros da planta são acessíveis em qualquer arquivo, desde que as
    propriedades da figura que compõe a janela principal (fig_main) sejam carregadas.
    Segue script utilizado em filtro.m, onde é necessário conhecer os dados da planta
    para executar o seu algoritmo

%%%%

% Localiza-se objeto de nome 'fig_main' e recupera-se os seus dados

    GUI = findobj(allchild(groot), 'flat', 'Tag', 'fig_main');

    handlesGUI = guidata(GUI);

% Salva os parâmetros da planta em outras variáveis para que se possa usar ao longo do
    algoritmo

    PlantData = getappdata(handlesGUI.fig_main, 'PlantData');

    ganho = PlantData.ganho;

    num = PlantData.num;

    den = PlantData.den;

    ts = PlantData.ts;

```

3.3.3 Gráfico de Estabilidade

O gráfico apresentado na Figura 17 é a composição do domínio de estabilidade do controlador com o diagrama de Nyquist da planta, além de apresentar até que frequência o diagrama está contido no domínio de estabilidade. Segue o *script* desta funcionalidade:

```

% Em controlador.m são obtidos os valores dos parâmetros 'a' e 'Q', seja por meio dos
    sliders ou por meio das caixas de texto. Com isso, a função 'plotar' é chamada para
    gerar o gráfico do diagrama de Nyquist em conjunto com a região de estabilidade do
    controlador

```

```

function plotar(a_value, q_value, handles, salvar)

% Localiza-se objeto de nome 'fig_main' e recupera-se os seus dados

GUI1 = findobj(allchild(groot), 'flat', 'Tag', 'fig_main');

handlesGUI1 = guidata(GUI1);

% Dados enviados quando a função foi chamada

a = a_value;

q = q_value;

% Recupera dados referentes ao diagrama de Nyquist a partir do compartilhamento de dados
entre arquivos

NyquistData = getappdata(handlesGUI1.fig_main, 'NyquistData');

re = NyquistData.real;

im = NyquistData.imag;

freq = NyquistData.freq;

% Definição dos limites do gráfico a partir das configurações definidas em
config_plot_control.m

lim_x = get(handles.plot_estabilidade, 'XLim');

lim_y = get(handles.plot_estabilidade, 'YLim');

r = min(lim_x(1), lim_y(1)):0.01:max(lim_x(2), lim_y(2));

[X, Y] = meshgrid(r);

% Definição da região de estabilidade

a = real(a);

q = abs(q)^2;

cond = inequacao(a, q, X, Y);

% Rotina para descobrir até onde o diagrama de Nyquist pertence ao domínio de
estabilidade. No final, os vetores 're_stable', 'im_stable' e 'last_w' contêm este
limite

re_stable = [re(1)];

im_stable = [im(1)];

last_w = [freq(1)];

```



```

for i = 2:1:length(freq)

    cond = inequacao(a, q, re(i), im(i));

    if cond

        re_stable(i) = re(i);

        im_stable(i) = im(i);

        last_w(i) = freq(i);

    else

        break

    end

end

end

%%%%
% O restante da função se resume em utilizar o que foi adquirido acima para criar o
% gráfico
%%%%
end % end function

```

3.3.4 Algoritmo de Dimensionamento do Filtro Ideal

Como introduzido na Seção 3.2.3, o filtro ideal para uma determinada aplicação é dimensionado a partir da variação da frequência associada à avaliação da Inequação (2.13), dados os valores dos parâmetros a e $Q(s)$. Agora, segue código escrito para executar esta tarefa, além da maneira utilizada para encontrar as referentes ordem e frequência de corte. Este algoritmo pode ser resumido no fluxograma mostrado na Figura 20.

```

% Localiza-se objeto de nome 'fig_main' e recupera os seus dados

GUI1 = findobj(allchild(groot), 'flat', 'Tag', 'fig_main');

handlesGUI1 = guidata(GUI1);

% Recupera dados referentes à planta a partir do compartilhamento de dados entre arquivos

PlantData = getappdata(handlesGUI1.fig_main, 'PlantData');

ganho = PlantData.ganho;

num = PlantData.num;

den = PlantData.den;

ts = PlantData.ts;

% Obtêm pontos do diagrama de Nyquist da função de transferência da planta

ft = ganho*tf(num, den, ts);

```

```

[re, im] = nyquist(ft, 2*pi*freq);
% Variáveis para executar o algoritmo

len = length(freq);      % Comprimento do vetor frequência

q = q_init*ones(1, len); % O vetor referente ao parâmetro Q(s) é inicializado como
                        % todos os valores igual ao escolhido pelo usuário ('q_init')

flag = 1;                %
mag_corte1 = 0;          % Variáveis
mag_corte2 = 0;          % de
f_c1 = 10;               % suporte
f_c2 = 0;                %

% Laço responsável pelo algoritmo de dimensionamento do filtro
for i = 1:1:len
    aux = abs(q(i))^2;
    cond = inequacao(a, aux, re(i), im(i));

    % Caso, para os atuais valores de frequência, 'a' e 'Q', a inequação não seja
    % válida, o valor de 'Q' é diminuído, no passo escolhido, até que a mesma se
    % torne válida. Só assim, dando prosseguimento ao algoritmo

    while(~cond)
        q(i:end) = q(i) - q_step;
        aux = abs(q(i))^2;
        cond = inequacao(a, aux, re(i), im(i));
    end

    % Obtêm-se os pontos pelas redondezas da frequência de corte (-3 dB)

    if (flag == 0 && mag2db(q(i)) < -3)
        mag_corte1 = mag2db(q(i-1));
        mag_corte2 = mag2db(q(i));
        f_c1 = freq(i-1);
        f_c2 = freq(i);
        flag = 1;
    end

    % Se entrar na condição anterior com i=1 vai dar problema no i-1. Tratamento
    % utilizado para o caso em que o primeiro ponto testado já é menor do que -3dB

    if (i==1)
        flag=0;
    end
end % end for

```

```

% Cálculo do coeficiente angular (decaimento)

coef_angular = (mag_corte1-mag_corte2)/log10(f_c1/f_c2);

% O coeficiente angular será igual a zero apenas no caso em que o primeiro ponto
% testado já é menor do que -3dB. Já que 'flag' sempre foi igual a 1, as variáveis
% ainda terão os seus valores iniciais. Nesse caso, é necessário aumentar o
% intervalo da frequência para que o algoritmo seja executado corretamente

if coef_angular == 0

    errordlg('You must increase the frequency range.','Invalid Input','modal');

    return

else

    % Calcula-se a frequência de corte do filtro

    
$$Coef.Angular = \frac{x_1+3dB}{\log(f_1-Freq.Corte)}$$


    freq_corte = 10^(log10(f_c1) - (mag_corte1+3)/coef_angular);

    % A ordem do filtro é encontrada considerando que:

    % Primeira ordem: decaimento de -20dB/década

    % Segunda ordem: decaimento de -40dB/década

    % Terceira ordem: decaimento de -60dB/década

    % ...

    % Por fim, arredonda-se pra cima o valor encontrado e garante-se que o mesmo seja
    % múltiplo de 2.

    filtro_ordem = ceil(coef_angular/-20);

    if mod(filtro_ordem, 2) ~= 0

        filtro_ordem = filtro_ordem + 1;

    end

    %%%%%

% O restante do código se resume em utilizar o que foi adquirido para criar o gráfico

    %%%%%

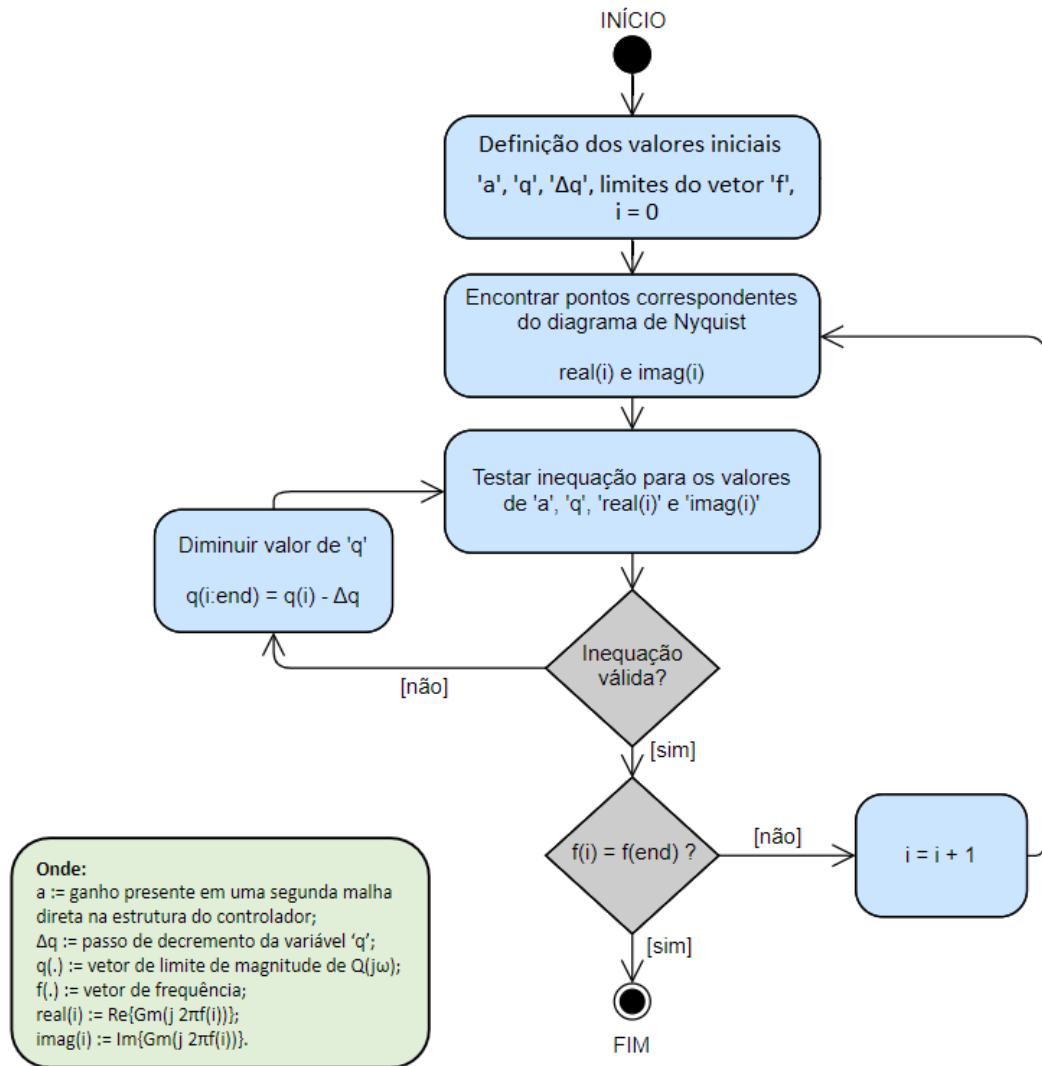
end % end if

```

3.4 CONCLUSÃO

A ferramenta GUIDE se mostrou suficiente para a implementação das equações correspondentes ao controlador objeto de estudo, além de proporcionar alguns recursos que contribuíram para o desenvolvimento do aplicativo desse trabalho, como: a possibilidade de utilizar as funções nativas do MATLAB e também a disponibilidade de elementos

Figura 20: Fluxograma referente ao algoritmo de dimensionamento do filtro ideal.



Fonte: Próprio Autor.

gráficos que contribuem para uma melhor experiência do usuário. Com isso, foi possível criar um aplicativo que oferece ao projetista um aporte visual para a escolha dos valores das variáveis do controlador.

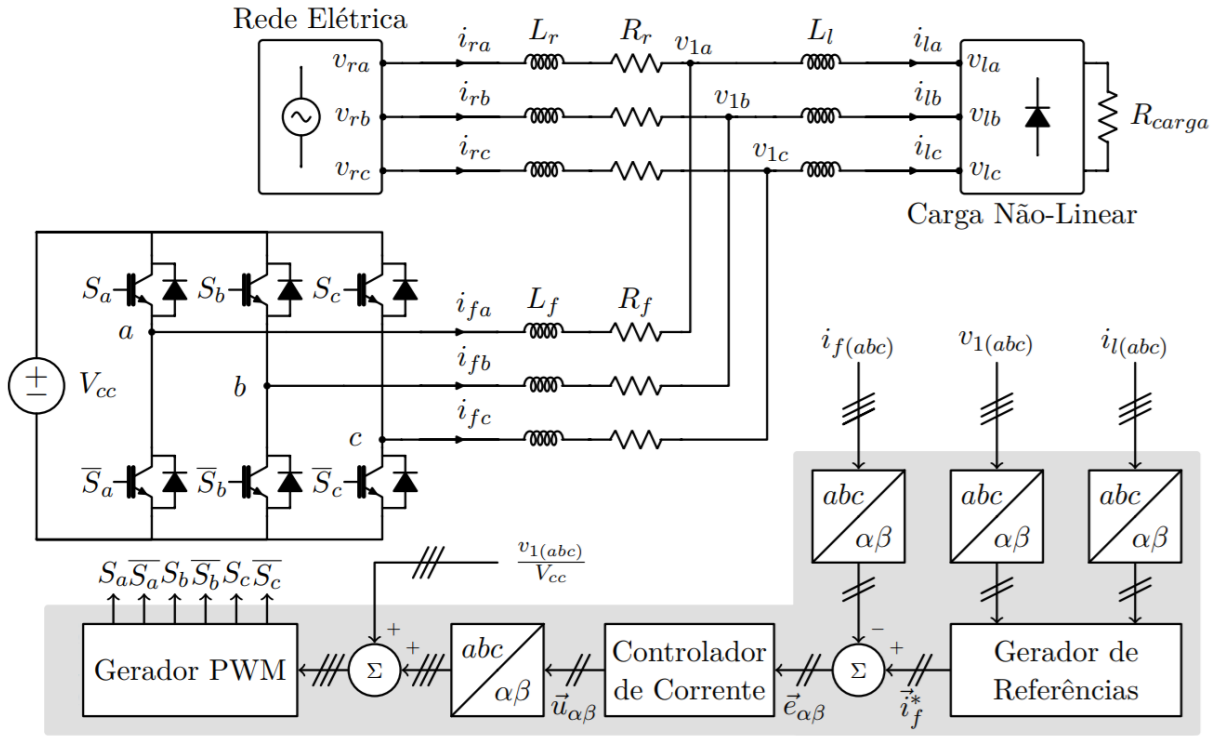
Foi necessário elaborar todas as rotinas de teste referentes às informações definidas pelo usuário, para confirmar que essas estão coerentes, garantindo que cada seção do aplicativo será executada corretamente. O desenvolvimento de um algoritmo para dimensionar um filtro ideal conseguiu dar ainda mais importância a este trabalho, o que será demonstrado no próximo capítulo.

4 VALIDAÇÃO DA FERRAMENTA NUMA APLICAÇÃO REAL

Este capítulo tem por finalidade a validação e, ainda, a comprovação da relevância da ferramenta criada. Isso se dará a partir da sua utilização em um sistema real, sendo apresentados tanto resultados de simulação usando o Simulink do MATLAB, como também resultados experimentais num protótipo em laboratório.

O sistema escolhido se refere ao controle de um filtro ativo trifásico paralelo a três fios (FAP), apresentado na Figura 21. A Tabela 2 apresenta os valores dos parâmetros do sistema, estes se referem aos valores reais da planta, obtidos em (NETO, 2020).

Figura 21: Diagrama do sistema escolhido para validação da ferramenta.



Fonte: (NETO, 2020).

Tabela 2: Parâmetros da aplicação.

Tensões (V)		Indutâncias (mH)			Resistências (Ω)			Amostragem (kHz)
$V_{r,rms}$	V_{cc}	L_r	L_f	L_l	R_r	R_f	R_{carga}	f_s
380	600	0,18617	2,5635	1,4830	0,0317	0,3075	48,4	17,28

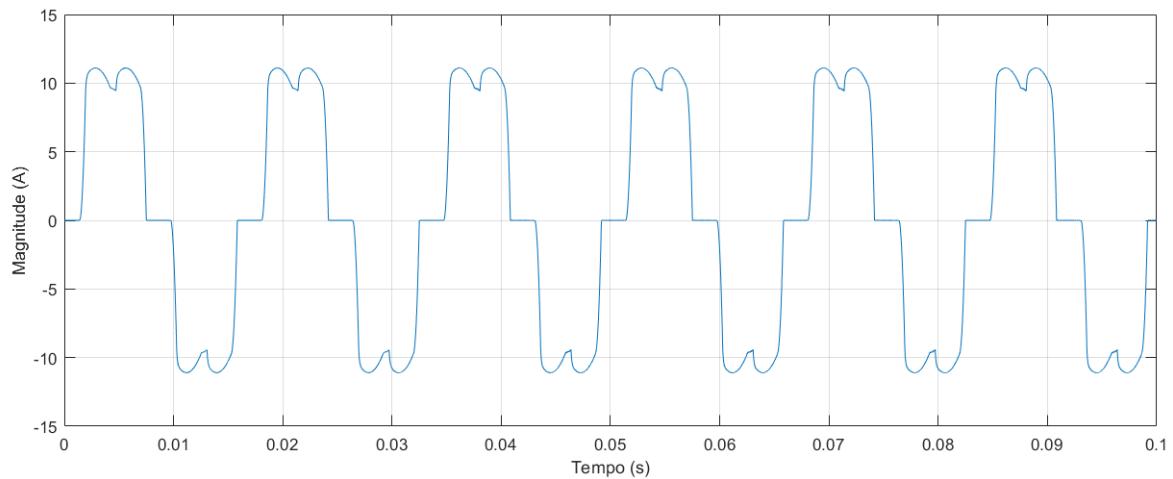
Fonte: Próprio Autor.

Considerando uma carga não-linear conectada à rede elétrica, a corrente consumida por esta carga distorce as correntes da rede i_{ra} , i_{rb} e i_{rc} . Por isso, de modo que a corrente

resultante fornecida pela rede seja idealmente senoidal, as correntes i_{fa} , i_{fb} e i_{fc} são adequadamente injetadas na carga a partir do chaveamento de um conversor CC-CA.

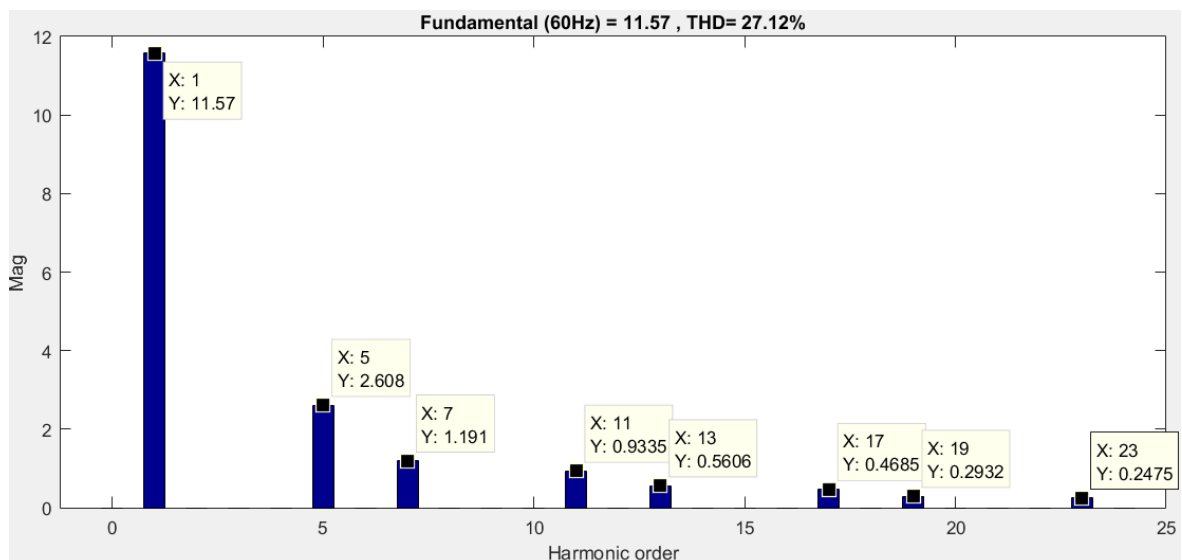
Nesta aplicação, a carga não-linear se refere a um retificador trifásico a diodos, cuja corrente (utilizando a fase a como referência) tem o comportamento no tempo apresentado na Figura 22 e pode ser representada no domínio da frequência conforme a Figura 23, com mais detalhes na Tabela 3.

Figura 22: Comportamento no tempo da corrente da fase a da carga não-linear i_{la} .



Fonte: Próprio Autor.

Figura 23: Representação da corrente da fase a da carga não-linear i_{la} no domínio da frequência.



Fonte: Próprio Autor.

Como a ideia do FAP é injetar corrente na carga de modo que a corrente da rede não seja distorcida, então, é coerente pensar que a referência do controle seja apenas os

Tabela 3: Componentes harmônicas da corrente da fase a da carga não-linear i_{la} .

Ordem Harmônica	Amplitude	Fase	Sequência
1	11,57A	-7,8°	Positiva
5	2,61A	143,9°	Negativa
7	1,19A	114,3°	Positiva
11	0,93A	-89,8°	Negativa
13	0,56A	245°	Positiva
17	0,47A	33,6°	Negativa
19	0,29A	15,6°	Positiva
23	0,25A	151,6°	Negativa

Fonte: Próprio Autor.

harmônicos da corrente da carga, sem a parte fundamental. Assim, fazendo com que a parte fundamental seja entregue apenas pela rede e que os harmônicos sejam entregues apenas pelo FAP. Na prática, aplica-se a transformada de Clarke (abc para $\alpha\beta$) nas tensões v_l e correntes i_l consumidas pela carga a partir da seguinte relação:

$$\begin{bmatrix} x_0(t) \\ x_\alpha(t) \\ x_\beta(t) \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 1 & \frac{-1}{2} & \frac{-1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{-3}}{2} \end{bmatrix} \begin{bmatrix} x_a(t) \\ x_b(t) \\ x_c(t) \end{bmatrix} \quad (4.1)$$

e então, ambas as variáveis em α e em β passam por um filtro passa-baixa com frequência de corte próxima à frequência da rede, para que sejam extraídas as suas partes fundamentais. Em seguida, o sinal da saída do filtro é subtraído do sinal completo, obtendo-se um novo sinal que possui apenas as componentes harmônicas. Em resumo, a referência do controle \vec{i}_f^* deve levar em conta $i_l - i_{l1}$, ou seja, as harmônicas da corrente da carga. Mais detalhes de implementação são encontrados em (NETO, 2020), onde o FAP é representado pela seguinte função de transferência:

$$G(s) = \frac{V_{cc}/L_f}{s + R_f/L_f}. \quad (4.2)$$

A modelagem do sistema em diagrama de blocos no tempo discreto pode ser observada na Figura 24, onde o ganho repetitivo K_{rc} se encontra agora após o controlador, sem perda do que já foi desenvolvido. Os parâmetros $n = 6$ e $m = 1$ são obtidos a partir da análise do espectro harmônico das correntes que alimentam o retificador a diodos (i_{la} , i_{lb} e i_{lc}) e o número de amostras por período da fundamental N é calculado a partir da frequência fundamental da rede f_1 e da frequência de amostragem f_s , sendo $N = f_s/f_1 = 288$. A planta $G(s)$ é discretizada a partir de um retentor de ordem zero

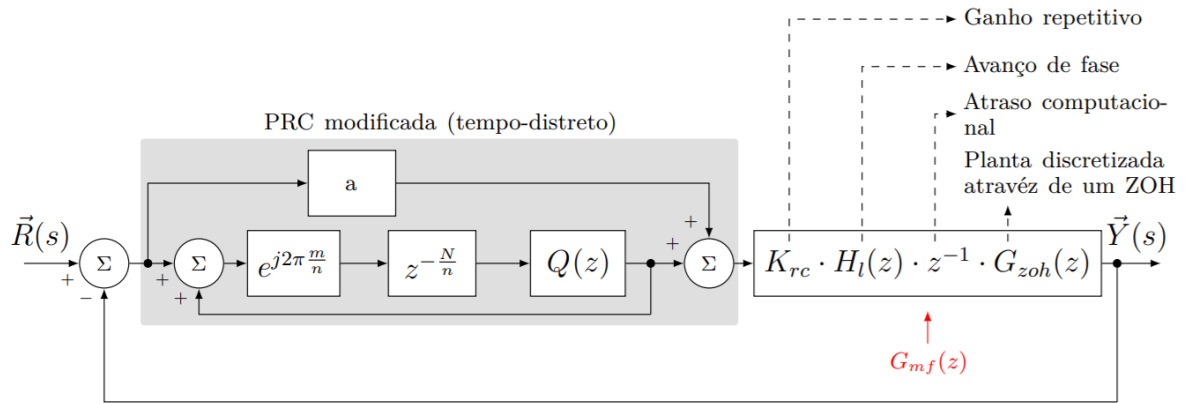
(ZOH - *zero-order holder*), sendo chamada agora de $G_{zoh}(z)$ que, após substituição dos valores, se torna:

$$G_{zoh}(z) = \frac{13,5}{z - 0,9931}. \quad (4.3)$$

O atraso unitário z^{-1} é utilizado para levar em conta o atraso computacional relacionado à implementação digital do sistema. Ainda, o termo $H_l(z)$ refere-se a um avanço de fase, adicionado por (NETO, 2020) para tornar o sistema estável, ao atenuar o efeito do atraso computacional, que é dado por:

$$H_l(z) = \frac{0,6526 - 0,4301z^{-1}}{1 - 0,08271 \cdot z^{-1}} \quad (4.4)$$

Figura 24: Diagrama de blocos do sistema.



Fonte: (NETO, 2020).

A validação e comprovação da relevância da ferramenta se darão a partir dos seguintes testes:

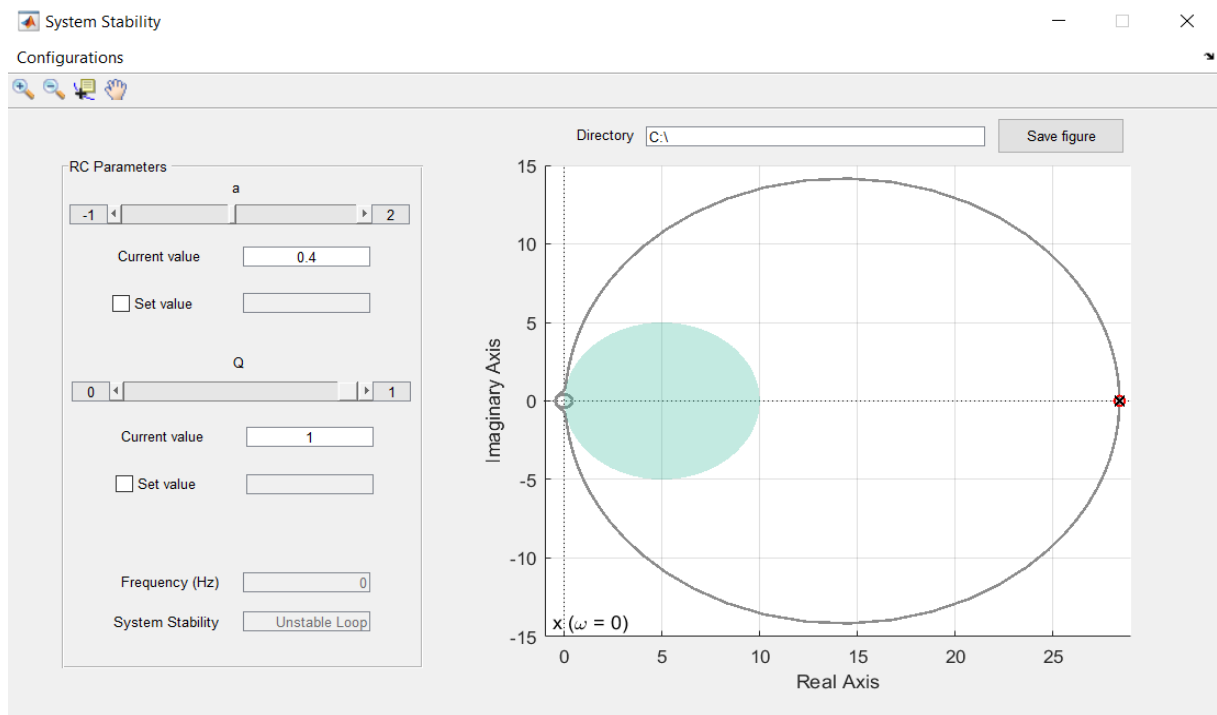
1. Análise da estabilidade do sistema variando apenas o parâmetro a (com $Q(z) = 1$);
2. Análise da estabilidade do sistema variando os parâmetros a e $Q(z)$;
3. Dimensionamento de um filtro para o sistema;
4. Análise do erro do sistema de acordo com a escolha das características do filtro.

4.1 RESULTADOS

4.1.1 Variação no Parâmetro a (com $Q(z) = 1$)

Primeiramente, foi observado o comportamento do sistema quando $Q(z) = 1$ e $a = 0.4$, $a = 0.5$ ou $a = 0.8$, utilizando $K_{rc} = 0,06$. A Figura 25 mostra o aplicativo, na seção de análise de estabilidade do sistema, após carregar os parâmetros do FAP e com $Q(z) = 1$ e $a = 0.4$. Para as próximas análises, apenas o gráfico correspondente à uma determinada configuração será mostrado. De acordo com a Figura 26, é esperado que o sistema não seja estável para nenhuma das combinações propostas. O que pode ser comprovado ao observar a corrente da rede (i_{ra}) correspondente à simulação de cada combinação.

Figura 25: Seção de análise de estabilidade da ferramenta criada com $Q(z) = 1$ e $a = 0.4$, após carregar os parâmetros do FAP.

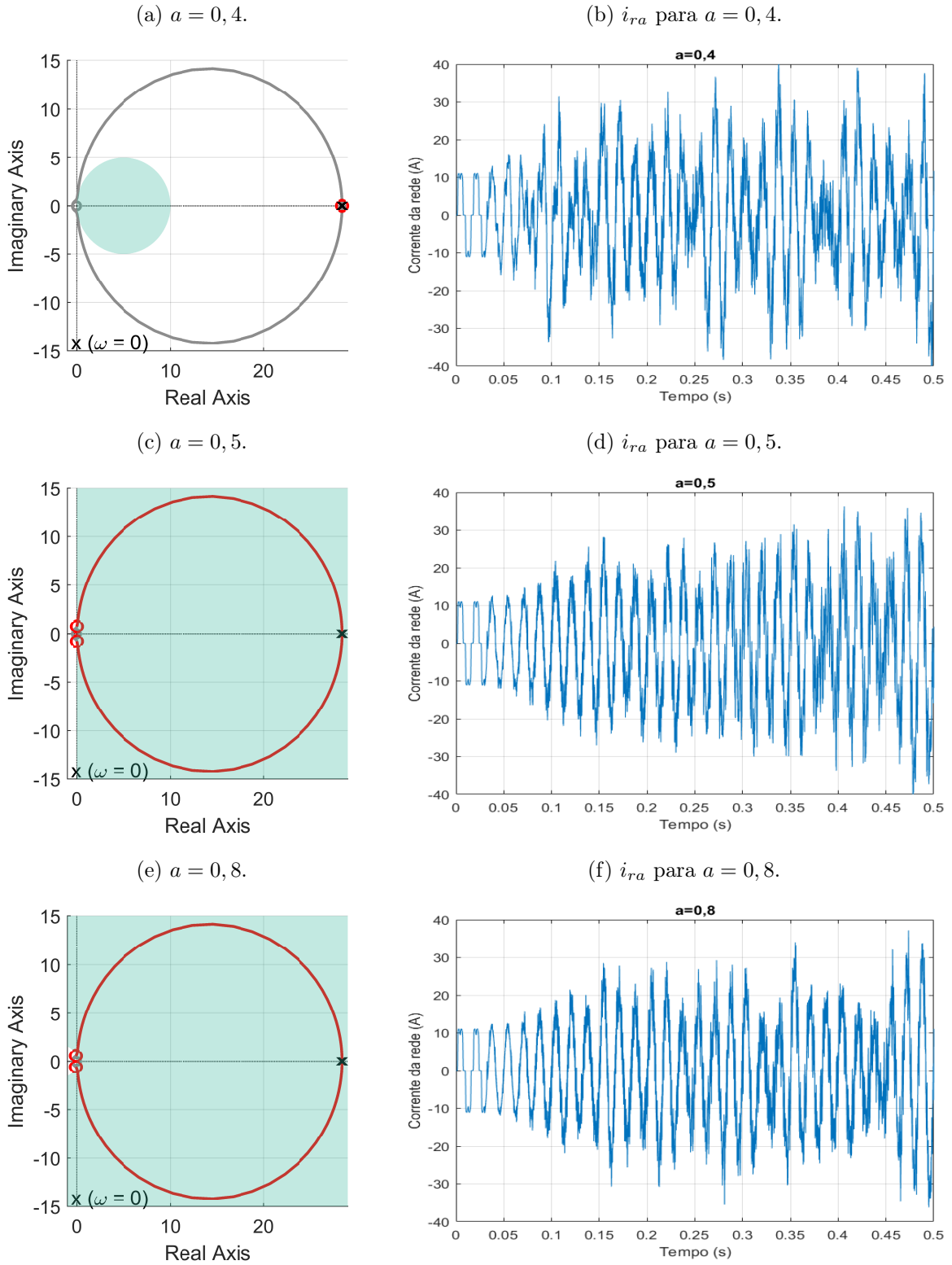


Fonte: Próprio Autor.

4.1.2 Variação nos Parâmetros a e $Q(z)$

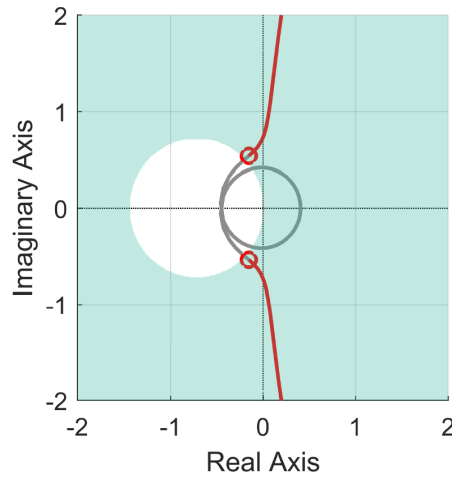
Agora, para efeito de comparação, será mantido o valor de $a = 0,5$, mas serão testados dois outros valores para $Q(z)$: $Q(z) = 0,9$ e $Q(z) = 0,6$. Pela Figura 28, é esperado que o sistema se mantenha estável apenas no segundo caso, o que é comprovado

Figura 26: Análise da estabilidade do sistema a partir da ferramenta criada para vários valores de a (com $Q(z) = 1$) e os resultados do sistema de controle simulado para cada caso, mostrando instabilidade em todas as situações.



Fonte: Próprio Autor.

Figura 27: Diagrama de Nyquist da planta e domínio de estabilidade do controlador em torno da origem, mostrando a impossibilidade de se garantir estabilidade com $Q(z) = 1$.



Fonte: Próprio Autor.

a partir da observação da corrente da rede (i_{ra}) correspondente. Nela, observa-se que para o primeiro caso, apesar de um comportamento visivelmente melhor do que quando $Q(z) = 1$, a corrente não possui um comportamento estável. Já para o segundo caso, apesar de ainda não ser uma senoide ideal, a corrente se mantém estável.

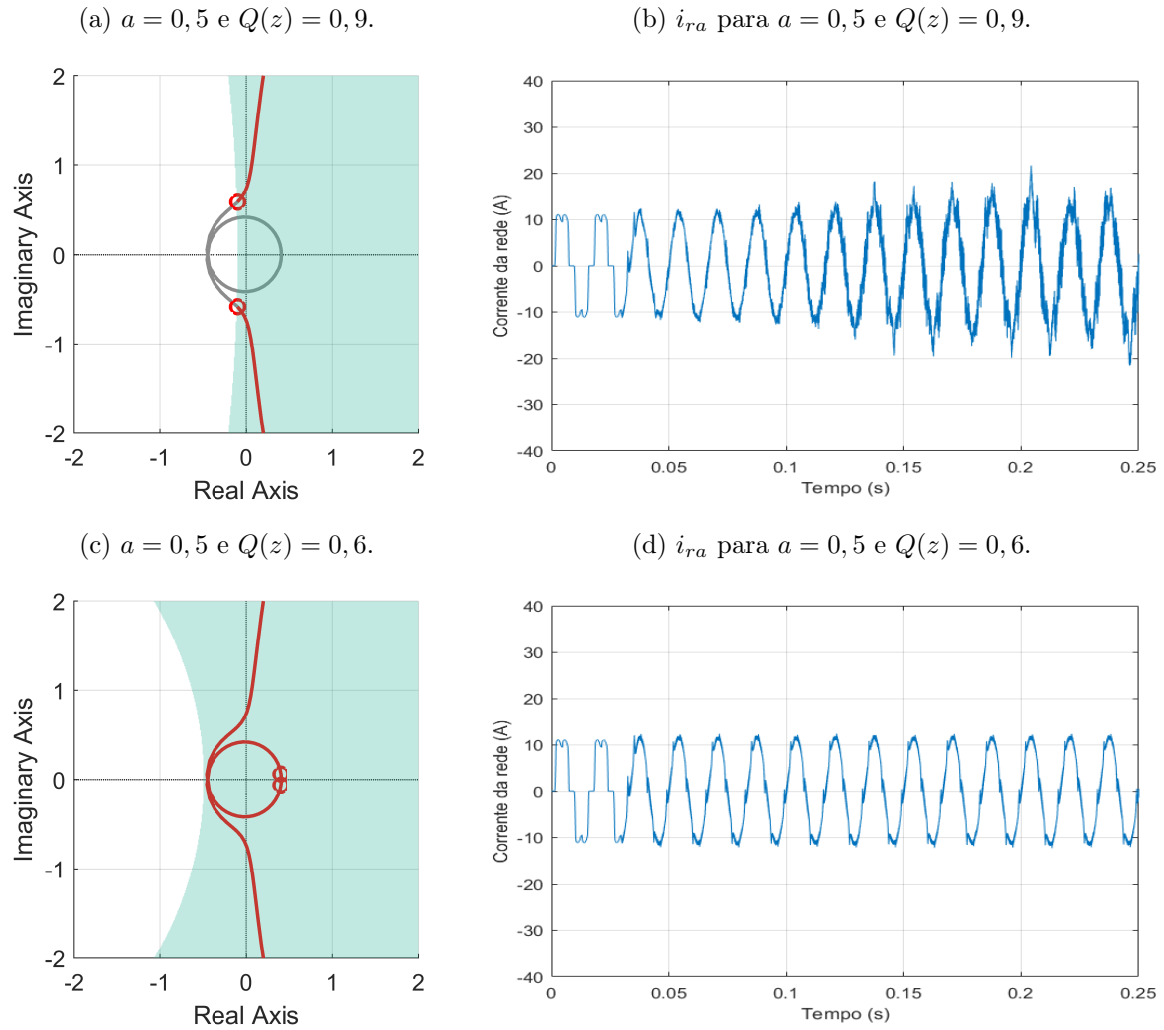
Como explicado na Seção 2.3, o erro não diverge, nem converge para zero quando utilizado um valor constante para o parâmetro $Q(z)$, o que pode ser observado na Figura 29. Por isso, a próxima seção trata do dimensionamento de um filtro passa-baixa para o sistema.

4.1.3 Dimensionamento do Filtro

Para esta seção, será utilizado o algoritmo da ferramenta que projeta um filtro ideal para o sistema. A Figura 30 mostra a janela do aplicativo após a execução deste algoritmo, onde os valores definidos inicialmente e os resultados obtidos a partir dele são apresentados na Tabela 4.

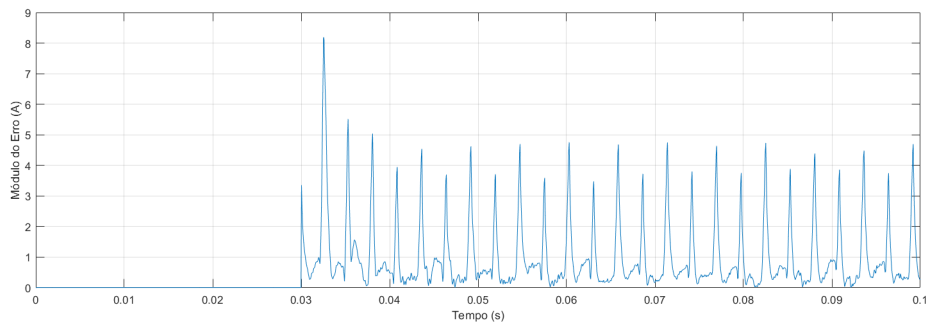
Vale destacar que a magnitude de $Q(z)$ da curva gerada pela ferramenta se refere à magnitude máxima que o parâmetro deve ter naquela determinada frequência para garantir a estabilidade. Portanto, o filtro ideal é aquele que está o mais próximo possível desta curva. Porém, por vezes, esse filtro ideal pode ser demasiado custoso a se implementar. Exatamente por isso a ferramenta permite que o usuário exporte os dados da curva gerada, a fim de deixá-lo livre para dimensionar um filtro menos dispendioso, mas

Figura 28: Análise da estabilidade do sistema a partir da ferramenta criada com $a = 0,5$ e $Q(z) \neq 1$.



Fonte: Próprio Autor.

Figura 29: Erro de corrente do sistema de controle para $a = 0,5$ e $Q(z) = 0,6$.

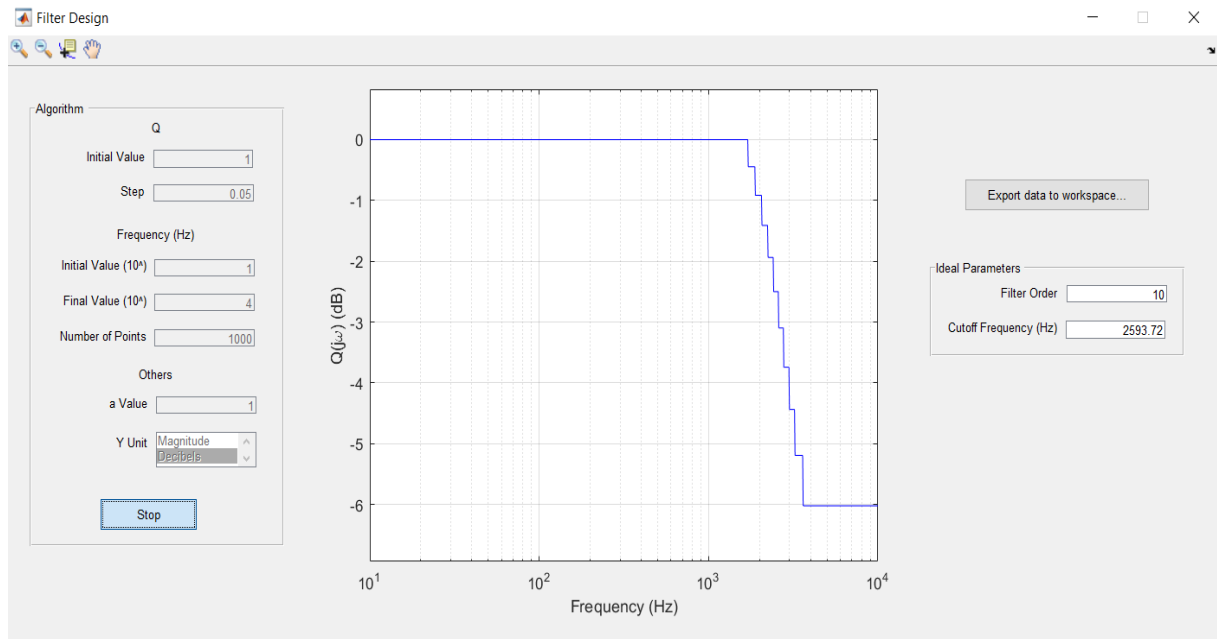


Fonte: Próprio Autor.

que continue garantindo a estabilidade do sistema.

O objetivo agora é mostrar que, a partir da curva gerada, é possível dimensionar um filtro tão bom quanto o ideal proposto, mas que seja menos custoso computacio-

Figura 30: Execução do algoritmo de dimensionamento do filtro $Q(z)$ para o FAP em análise.



Fonte: Próprio Autor.

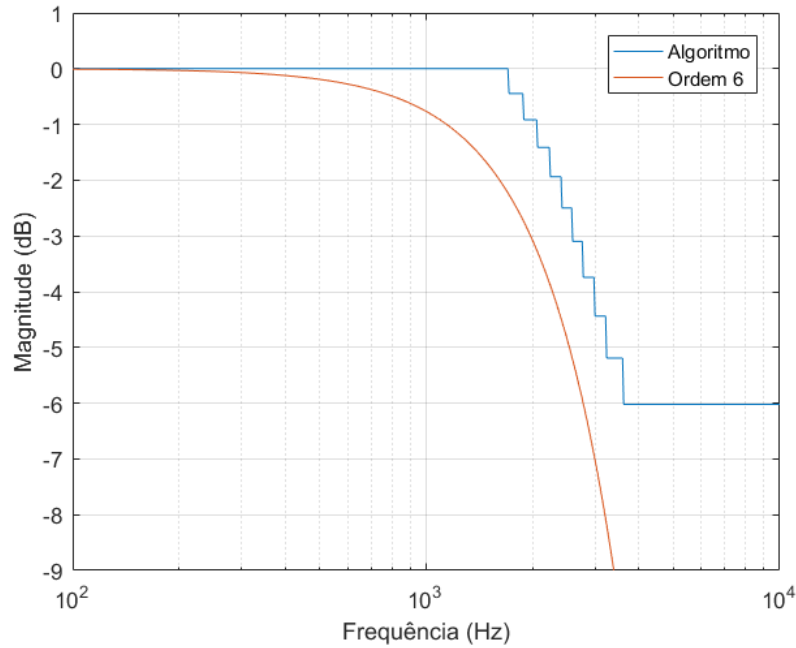
nalmente (menor ordem). Após alguns testes utilizando a função $\text{fir1}()$ do MATLAB, apropriada para gerar filtros FIR (*Finite Impulse Response*), foi observado que um filtro de ordem 6 e frequência de corte igual a 1800 Hz possui um comportamento no domínio da frequência que respeita a curva limite de estabilidade gerada pela ferramenta, como pode ser observado na Figura 31. Esse filtro foi utilizado no protótipo do FAP e o resultado experimental pode ser observado na Figura 32, onde i_r se refere à corrente da rede, i_l se refere à corrente da carga e i_f se refere à corrente do FAP.

Tabela 4: Parâmetros e resultados da execução do algoritmo de dimensionamento do filtro ideal para o caso real.

Definições Iniciais	
a	1
$Q(z)$ Inicial	1
Decremento Δq	0,05
Frequência Inicial	10 Hz
Frequência Final	10 kHz
Nº de Pontos (Vetor Frequência)	1000
Resultados	
Ordem do Filtro	10
Frequência de Corte	2593,72 Hz

Fonte: Próprio Autor.

Figura 31: Projeto de um filtro FIR de ordem 6 a partir da curva limite de estabilidade de $Q(z)$ gerada pela ferramenta desenvolvida, que é aplicada no sistema de controle do FAP em análise.



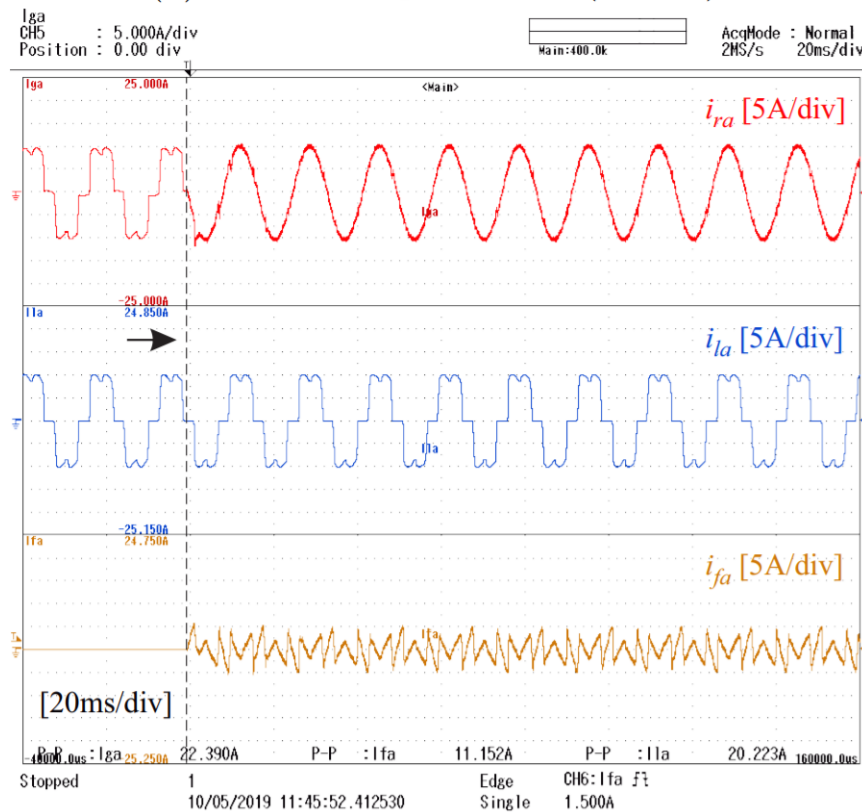
Fonte: Próprio Autor.

Ainda, é possível observar o impacto da variação do parâmetro a na estabilidade do sistema. Para tal, serão mantidos os valores de todos os outros parâmetros e o sistema será testado para os seguintes valores de a : $a = 0,8$, $a = 0,6$ e $a = 0,4$. Na Figura 33, é possível observar o resultado destes testes, onde nas Figuras (a), (c) e (e) são apresentados os limites da magnitude do parâmetro $Q(z)$ (em verde) e a curva do filtro utilizado (em vermelho) e nas Figuras (b), (d) e (f) são apresentadas as correntes da rede (vermelho), da carga (azul) e do FAP (laranja). Para os dois primeiros casos, o filtro utilizado ainda está dentro dos limites de magnitude, gerando correntes estáveis e bem controladas, diferentemente quando $a = 0,4$, no qual a corrente da rede i_{ra} apresenta oscilação crescente, indicativo de que o sistema vai tender à instabilidade.

4.1.4 Impacto da Escolha da Equação do Filtro

Apesar deste capítulo se referir à testes relacionados ao sistema apresentado na Figura 21, agora será aberta uma exceção e a ferramenta será utilizada considerando o controle da planta descrita pela Equação (2.14), pois esta permite uma melhor visualização dos sinais de erro em função do tamanho do filtro FIR utilizado e sua banda passante, elucidando melhor aquilo que se deseja demonstrar aqui.

Figura 32: Resultado experimental com filtro FIR dimensionado a partir da ferramenta, com $a = 1$.



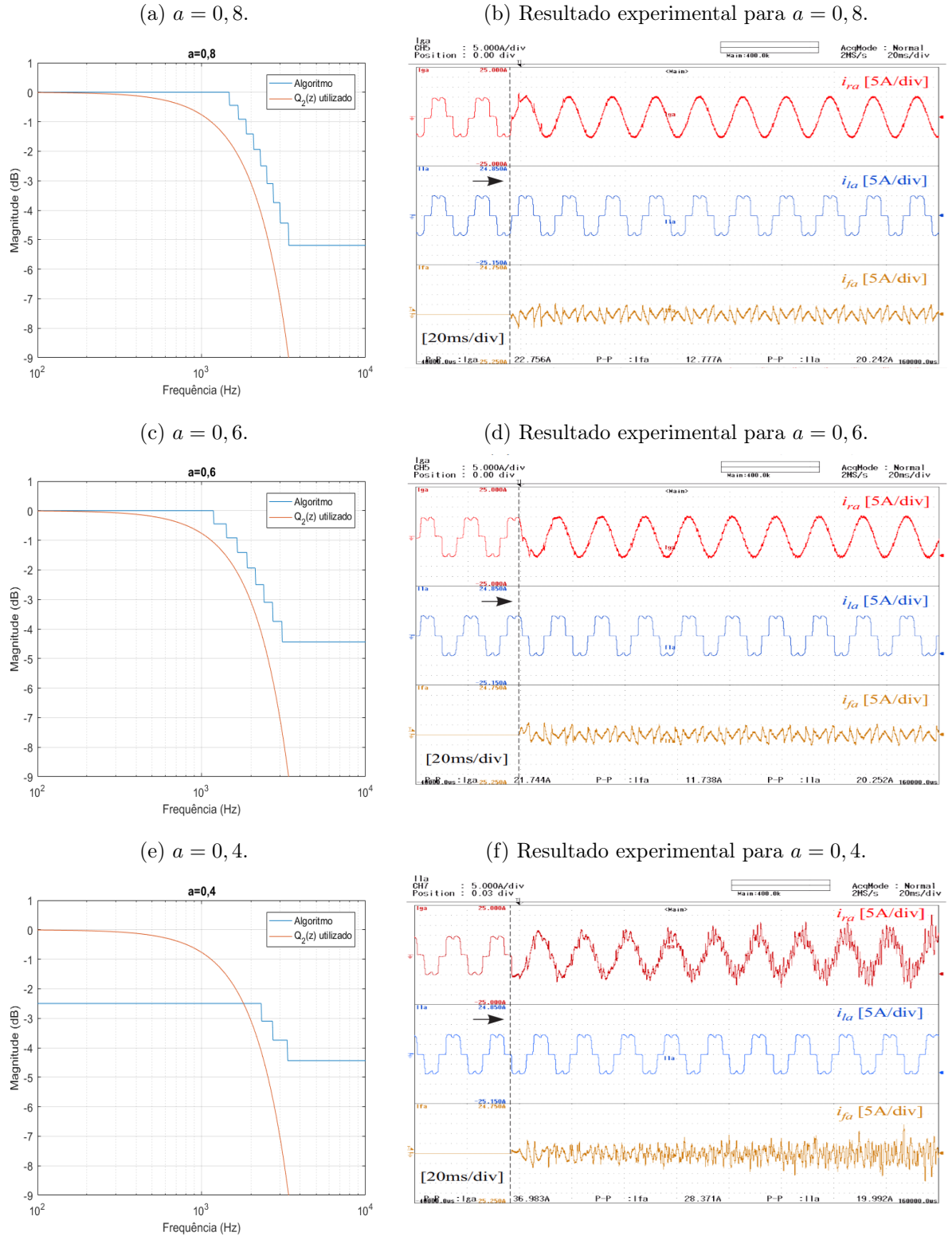
Fonte: (NETO, 2020).

Na seção anterior, o filtro utilizado no sistema foi escolhido a partir da curva com os limites da magnitude de $Q(z)$ gerada pela ferramenta. Foi escolhido um filtro que respeitasse os limites, mas não exatamente aquele proposto. Agora, o fato de o filtro dimensionado ser ou não custoso não será mais importante, o que será observado é o impacto dessas escolhas no controle do sistema, mais especificamente no comportamento do erro.

A Figura 34 mostra a janela do aplicativo após a execução do algoritmo de dimensionamento do filtro. Os valores definidos inicialmente e os resultados obtidos são apresentados na Tabela 5.

Então, segundo a ferramenta, o melhor filtro para esta aplicação é aquele de ordem 16 e com frequência de corte igual a 1088,2 Hz. O objetivo agora é encontrar um filtro menos custoso computacionalmente (menor ordem), mas que respeita os limites impostos pela ferramenta. Segundo (MICHELS, 2006), é muito comum que em aplicações com este tipo de controlador seja utilizado um filtro de ordem 2, porém, o autor não conseguiu chegar a um filtro de 2º ordem, utilizando a função $fir1()$ do MATLAB, que respeitasse os

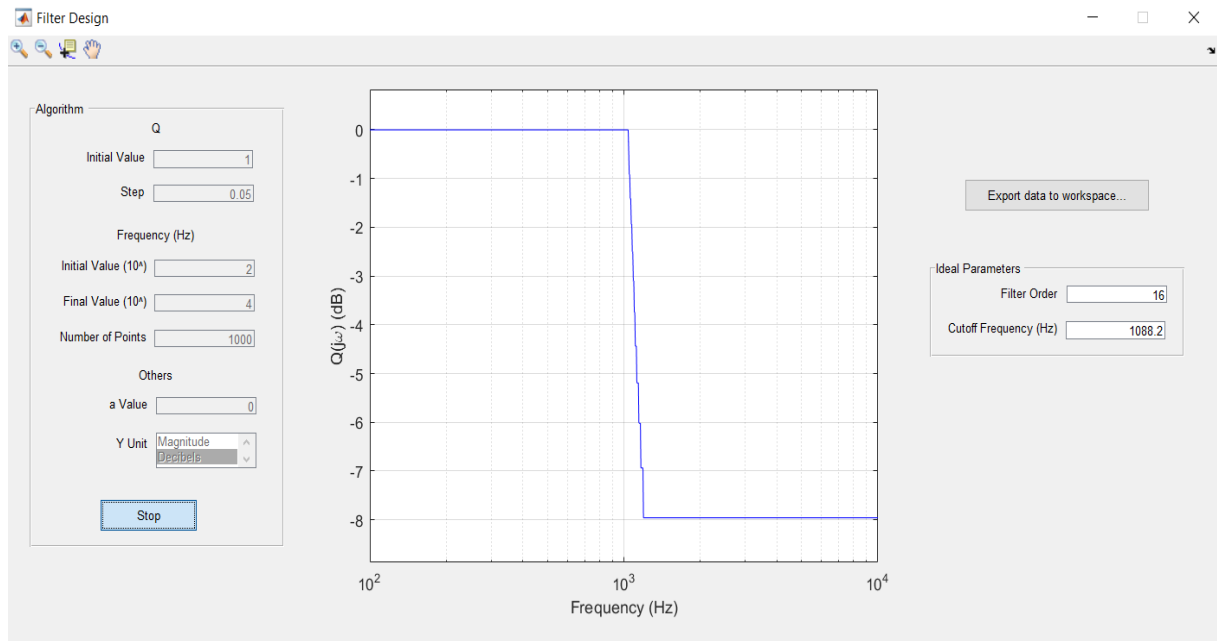
Figura 33: Resultado experimental com filtro FIR projetado e para diferentes valores do parâmetro a .



Fonte: (NETO, 2020), Adaptado.

limites impostos pela ferramenta. Por isso, a estratégia foi aumentar a ordem do filtro e diminuir a frequência de corte, gradativamente. Enfim, foi possível chegar em um filtro de

Figura 34: Execução do algoritmo de dimensionamento do filtro ideal para planta da Equação (2.14).



Fonte: Próprio Autor.

Tabela 5: Parâmetros e resultados da execução do algoritmo de dimensionamento do filtro ideal para planta da Equação (2.14).

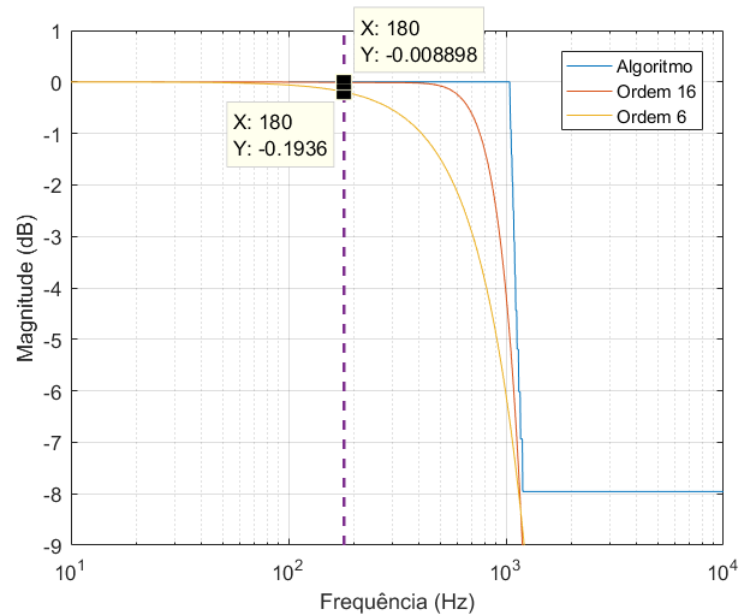
Definições Iniciais	
a	0
$Q(z)$ Inicial	1
Decremento Δq	0,05
Frequência Inicial	100 Hz
Frequência Final	10 kHz
Nº de Pontos (Vetor Frequência)	1000
Resultados	
Ordem do Filtro	16
Frequência de Corte	1088,2 Hz

Fonte: Próprio Autor.

ordem 6 e frequência de corte igual a 700 Hz. A Figura 35 mostra o comportamento dos referidos filtros, juntamente com o resultado obtido da ferramenta. Portanto, é esperado que ambos os filtros sejam capazes de controlar o sistema.

Porém, segundo (NETO, 2020), para todas as frequências nas quais o filtro possui um ganho não unitário, o sistema tende a um erro de regime permanente não nulo. Logo, o melhor filtro é aquele que possui uma faixa de passagem com ganho unitário mais abrangente possível. Sabendo disso, se considerado como referência um sinal trifásico de amplitude igual a 1 e de frequência igual a 180Hz, é esperado que o erro em regime

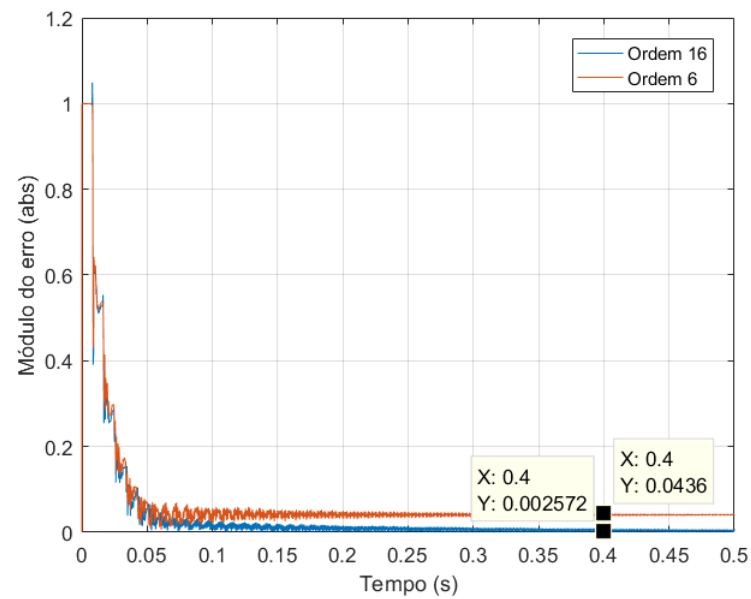
Figura 35: Projeto dos filtros baseado no limite definido pela ferramenta para a planta da Equação (2.14).



Fonte: Próprio Autor.

permanente para o sistema com o filtro de ordem 16 seja menor que com aquele de ordem 6. Esse exato comportamento pode ser observado na Figura 36, onde o erro de regime permanente para o filtro de ordem 6 é aproximadamente 17 vezes maior do que para o filtro de ordem 16.

Figura 36: Módulo do erro para diferentes filtros.



Fonte: Próprio Autor.

4.2 CONCLUSÃO

Enfim, foi possível comprovar que a implementação realizada no Capítulo 3 trata fielmente o controlador proposto por (NETO, 2020), visto que todas as premissas levantadas foram comprovadas por meio de testes experimentais. Ainda, observa-se dinamicamente o impacto das escolhas do projetista.

Além disso, foi demonstrada a importância do algoritmo de dimensionamento de um filtro ideal que, ao gerar uma curva que impõe a magnitude máxima de $Q(z)$ referente a determinadas frequências, proporciona ao usuário uma liberdade de projetar um filtro de acordo com as suas necessidades. É o usuário que definirá aquilo que possui maior relevância para a sua aplicação.

5 CONCLUSÃO

O propósito deste trabalho foi a criação de uma ferramenta no MATLAB, apresentada detalhadamente no Capítulo 3, cujo objetivo é fomentar e facilitar o projeto de controladores repetitivos. Para tal, conhecimentos sobre MATLAB, programação e desenvolvimento de software se fizeram essenciais. Além disso, por se referir à uma proposta de controlador repetitivo, foi necessário também a compreensão de conceitos relacionados à sistemas de controle e ao projeto de filtros.

O objetivo principal era que a ferramenta respeitasse todas as regras referentes ao controlador proposto por (NETO, 2020), o que foi possível comprovar a partir das análises e testes em uma aplicação real, realizadas no Capítulo 4. Além disso, ela se mostra útil por facilitar a visualização da influência dos parâmetros do controlador na estabilidade do sistema. Ainda, ela se mostra relevante ao proporcionar ao usuário uma liberdade nas escolhas dos parâmetros de acordo com a sua necessidade, como foi mostrado no teste de diferentes ordens para o filtro.

Para o autor, este trabalho contribui não só para projetos de controladores repetitivos, como também para o desenvolvimento de aplicativos no MATLAB, de maneira geral, visto que essa prática é uma forma de valorizar ainda mais o objeto de estudo. Isso impacta positivamente no desenvolvimento de trabalhos, em qualquer nível. Por isso, é esperado que futuramente outros trabalhos tomem este como referência e acrescentem esse tipo de recurso nas suas propostas.

A ferramenta nessa versão já disponibiliza diversos recursos ao usuário que deseja projetar um controlador repetitivo, porém, alguns outros recursos podem ser adicionados futuramente:

- Mudança em tempo real do gráfico do domínio de estabilidade ao deslocar os sliders;
- Possibilidade de alteração de mais parâmetros dos gráficos;
- No dimensionamento do filtro, apresentação dos coeficientes e da curva do filtro proposto.

O autor experimentou implementar todos esses recursos citados, porém, para cada um se chegou em um ponto onde era necessário um estudo mais aprofundado em algumas

áreas. Por não serem imprescindíveis e por não diminuírem a importância da ferramenta, foram deixados como sugestões para trabalhos futuros.

Sabendo que a ferramenta engloba uma grande quantidade de arquivos, os quais possuem várias linhas de código, referentes à configuração dos objetos presentes nas figuras e também aos algoritmos desenvolvidos, o autor considerou inconveniente anexar todos os códigos a este trabalho. Por isso, tanto o instalador da ferramenta quanto todos os arquivos de código podem ser acessados pelo *link*: <https://mathworks.com/matlabcentral/fileexchange/74759-repetitive-controller-designer>.

REFERÊNCIAS

- FRANCIS, Bruce. **A course in H-infinity control theory**. 1. ed. Berlim: Springer-Verlag, 1987, (Lecture notes in control and information sciences, vol. 88). DOI: 10.1007/BFb0007371.
- FRANCIS, Bruce; WONHAM, Walter. The internal model principle for linear multivariable regulators. **Applied Mathematics and Optimization**, vol. 2, 1975, pp. 170–194, DOI: 10.1007/BF01447855.
- HARA, Shinji; OMATA, Tohru; NAKANO, Michio. Synthesis of repetitive control systems and its application. **1985 24th IEEE Conference on Decision and Control**, pp. 1387–1392, DOI: 10.1109/CDC.1985.268739.
- KAZMIERKOWSKI, Marian P.; MALESANI, Luigi. Current control techniques for three-phase voltage-source PWM converters: a survey. **IEEE Trans. Industrial Electronics**, vol. 45, 1998, pp. 691–703, DOI: 10.1109/41.720325.
- MATHWORKS. **GUIDE Migration Strategies**. Disponível em: https://www.mathworks.com/help/matlab/creating_guis/differences-between-app-designer-and-guide.html. Acessado em: 15/02/2020.
- _____. **See how MATLAB and Simulink are used in applications and industry**. Disponível em: <https://www.mathworks.com/academia/educators.html>. Acessado em: 15/02/2020.
- _____. **Share Data Among Callbacks**. Disponível em: https://www.mathworks.com/help/matlab/creating_guis/share-data-among-callbacks.html. Acessado em: 23/02/2020.
- MICHELS, Leandro. **Metodologia de Projeto de Fontes Ininterruptas de Energia Monofásicas Empregando Controladores de Ação Repetitiva Auxiliar no Estágio de Saída**. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Santa Maria, Santa Maria, 2006.
- NETO, Rafael C. **Controle Repetitivo Complexo Baseado em GDSC Aplicado a Filtro Ativo de Potência**. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Pernambuco, Recife, 2018.

NETO, Rafael C. **Controle Repetitivo Implementado por Estruturas Real e Complexa: Uma Abordagem Unificada**. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal de Pernambuco, Recife, 2020.

NETO, Rafael C. et al. Controladores Complexos Aplicados a Vetores Espaciais: Características e Vantagens. **XXII Congresso Brasileiro de Automática**, João Pessoa, 2018, DOI: 10.20906/CPS/CBA2018-0926.

THIBAUT, Robert. **Contrôle de l’Energie Injectée dans un Réseau Electrique par un Convertisseur Triphasé Utilisant un Régulateur Basé sur un Modèle Sinusoïdal**. Maîtrise en Génie Électrique – Université du Québec, Montréal, 2006.