MARIANA LUISA DE LIMA TORQUATO

# DESENVOLVIMENTO DE FERRAMENTA PARA ESTUDO DA ESTABILIDADE DE SISTEMAS COM ATRASO PELA APROXIMAÇÃO DE ZEROS INSTÁVEIS DE FUNÇÕES TRANSCENDENTES

São Paulo
2008

MARIANA LUISA DE LIMA TORQUATO

# DESENVOLVIMENTO DE FERRAMENTA PARA ESTUDO DA ESTABILIDADE DE SISTEMAS COM ATRASO PELA APROXIMAÇÃO DE ZEROS INSTÁVEIS DE FUNÇÕES TRANSCENDENTES

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para conclusão do curso de Engenharia Mecânica – Habilitação em Automação e Sistemas

Orientador: Prof. Dr. Newton Maruyama

São Paulo
2008

## FICHA CATALOGRÁFICA

**Torquato, Mariana Luisa de Lima**

**Desenvolvimento de ferramenta para estudo da estabilidade de sistemas com atraso pela aproximação de zeros instáveis de funções transcendentes / M.L.L.Torquato. – São Paulo, 2008.**

**p.**

**Trabalho de Formatura – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.**

**1.Sistemas com atraso. 2.Estabilidade. I.Torquato, Mariana Luisa de Lima II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.**

# DEDICATÓRIA

*Dedico este trabalho aos meus pais, pelo apoio oferecido durante a elaboração deste e de tantos outros trabalhos.*

*Mariana*

# AGRADECIMENTOS

Ao Professor Newton Maruyama, pela estimulante orientação e apoio na realização desta Monografia, pela amizade e pelos constantes ensinamentos e conselhos.

Aos Professores Doutores Lucas Antonio Moscato e Edson Gomes, pelo constante estimulo e pelas valiosas considerações que muito me auxiliaram na realização desta Monografia.

À minha família pela compreensão e à todos que colaboraram direta ou indiretamente na execução deste trabalho.

# RESUMO

A presença de atrasos em diversos casos reais de sistemas faz com que a teoria de controle se torne não trivial, requerendo técnicas específicas de análise e comando. A comunidade que estuda o comportamento de sistemas com atraso se interessa pela determinação da estabilidade desses sistemas, a qual pode ser determinada pela localização de seus zeros e pólos. Alguns potentes *softwares* de cálculo numérico existem; entretanto, não contam com rotinas especializadas e aproximações a sistemas de ordens inferiores, que devem ser realizadas antes de se aplicar às rotinas disponíveis. Parte deste trabalho visa o estudo da determinação de raízes instáveis pela aproximação de funções transcendentes, para posterior estudo da estabilidade de sistemas com atraso. Outra parte consiste no desenvolvimento de um programa flexível de cálculo das raízes, para servir de ferramenta ao estudo da estabilidade de um sistema com atraso qualquer.

**Palavras-chave:** Controle, sistemas com atraso, estabilidade, aproximações de funções transcendentes.

# ABSTRACT

Delays occur in diversified types of real systems, so that the control theory becomes not trivial, requiring specific techniques for the analysis and command. The delay-community, which studies the behaviour of delay-systems, is interested in determining these systems' stability. This stability can be determined by the placement of zeros and poles. Some powerful softwares of numerical computing exist. However, they do not contain specialized routines and approximations to low-order systems should be done before using the available routines. Part of this work intends to study the determination of unstable roots by approximating transcendental functions, for further study of the delay-systems' stability. Another part of this work consists in developing a flexible program to compute roots, in order to serve as a tool for the study of the stability of any delayed system.



**Keywords:** Control, delay-systems, stability, transcendental functions.

# LISTA DE ILUSTRAÇÕES

# LISTA DE TABELAS

# Sumário

# 1 Introdução

Diversos sistemas reais apresentam evolução dependente do estado atual, assim como dos estados precedentes. A presença de transporte de matéria, de energia ou de informação, fenômeno que se constitui como uma fonte de atraso, leva a este comportamento. Estes sistemas são modelados com o uso de equações diferenciais com atrasos, as quais apresentam uma dimensão real do estado infinito. Dessa forma, a presença de atrasos faz com que a teoria de controle se torne não trivial, requerendo técnicas específicas de análise e comando.

A comunidade que estuda o comportamento desses sistemas se interessa pela determinação da estabilidade dos mesmos, através da determinação da localização de seus zeros e pólos. Alguns potentes *softwares* de cálculo científico, como *Matlab* e *Scilab*, estão disponíveis. Entretanto, não existem rotinas adaptadas aos sistemas supracitados. Devem-se incorporar aproximações de sistemas lineares invariantes com o tempo de dimensão infinita por sistemas de dimensão finita de ordens inferiores, para se aplicar às rotinas disponíveis.

Este trabalho tem como metodologia a revisão da literatura disponível sobre sistemas com atraso, bem como o estudo da determinação de raízes instáveis pela aproximação de funções transcendentes, para a obtenção de zeros e pólos, os quais servirão ao estudo da estabilidade desses sistemas. O resultado deste estudo será utilizado no desenvolvimento de um programa flexível de cálculo de raízes para servir de ferramenta ao estudo da estabilidade de um sistema com atraso qualquer.

Dessa forma, este trabalho tem como principal motivação atender às necessidades de uma comunidade de pesquisadores especializados no estudo destes sistemas, através da integração de conhecimentos teóricos a uma aplicação prática flexível.

## 2 Cenário

Inicialmente, as pesquisas conduzidas sobre este tema eram de natureza puramente tecnológica, mas os componentes metodológicos rapidamente dominaram a cena. A razão essencial foi, acima de tudo, a necessidade da formalização de problemas de base, tais como a estabilização e o recurso aos atrasos distribuídos, seguidos pela necessidade de se desenvolver módulos de ferramentas específicas, passo incontornável a toda e qualquer implementação de aplicação.

Esse importante eixo de pesquisa foi desenvolvido desde o início dos anos 90, motivado a princípio pelas demandas do setor sócio-econômico e preocupava-se com a estrutura de sistemas lineares com atraso. Esses sistemas podem representar fenômenos que intervêm em numerosos processos, tais como as telecomunicações, os transportes de energia ou de informação. São estudados em diversos programas de pesquisa internacionais, como no *National Science Foundation* (NSF) e no *Centre national de la recherche scientifique* (CNRS). Entre eles, o *Institut national de la recherche em informatique et automatique* (INRIA) está fortemente presente no desenvolvimento da "estrutura" e "estabilização".

Para a modelagem e análise de sistemas com atrasos, diferentes propriedades matemáticas são utilizadas. Algumas ferramentas matemáticas, como o uso do anel, denominado Épsilo*n,* para a localização de pólos para sistemas de equações lineares, já foram apresentadas, mas as principais conseqüências para o comando e estabilização restam ainda como objeto de estudo.

Alguns fenômenos ainda não foram dominados, nem explicados. Por exemplo, diferentes esquemas de discretização de integrais mostraram não ser equivalentes em relação à conservação da estabilidade. É necessário encontrar uma aproximação que não introduza instabilidades estruturais. Algumas configurações já foram exibidas, entretanto, na globalidade, existem problemas que continuam em aberto, cuja temática é atualmente abordada em trabalhos.

As leis de comando que operam na localização dos pólos são complexas, definidas como equações integrais implícitas. Diversos métodos foram propostos para se avaliar a margem de estabilidade sobre o atraso. Ao senso entrada limitada - saída limitada ([18]), as noções de estabilidade e estabilização se formulam no contexto geral da classe de *Callier-Desoer* e de funções de transferência $H_\infty$. Continuou-se a estudar o problema com a realização de uma escala constituída por diferentes tipos de sistemas com atraso.

Atualmente, os formalismos são bem conhecidos e são utilizados pela comunidade de estudo

em todo o mundo. As pesquisas se voltam em direção à implementação desses métodos. Isso conduz, de um lado, ao estudo de dificuldades numéricas que foram sublinhadas por vários autores e, de outro lado, ao aprofundamento da teoria, para levar em consideração a robusteza, resolver questões específicas, generalizar o método para classes mais amplas de sistemas dinâmicos ou, ao contrário, especializá-la para importantes modelos simples em prática.

Alguns algoritmos desenvolvidos já foram implementados em ferramentas de cálculo simbólico, como CoCoA e MapleV, permitindo, assim, abordar os aspectos da estabilidade. São ferramentas metodológicas que não existiam antes e que atualmente são continuados por alguns pesquisadores italianos das Universidades de Ancona e de Roma.

Entretanto, para se continuar o estudo da estabilidade de sistemas com atraso, existe uma necessidade de se incluir os diversos formalismos e algoritmos desenvolvidos, utilizando da manipulação simbólica como meio para se resolver um sistema específico. A satisfação a esta necessidade será o objeto deste trabalho.

## 3 Discussão do projeto

Este projeto foi dividido nas seguintes etapas:

a. Levantamento de dados;
b. Revisão bibliográfica;
c. Implementação;
d. Testes e Correções;
e. Confecção de documentação.

Ajudando a constituir o cronograma de atividades, estas etapas, previstas para uma realização lógica, atendem, primeiramente, a uma boa definição do problema, dado pelo Levantamento de dados e pelas bases lançadas com a Revisão bibliográfica. Em seguida, parte para a implementação do programa, para a verificação através de testes e correções e para a confecção da documentação.

Atividades a serem realizadas para o levantamento de dados:

a.1 A determinação da necessidade da comunidade de pesquisa, a partir do levantamento dos interesses de uma pesquisadora do grupo;

a.2 A busca por um professor interessado, para assumir a orientação junto à Escola Politécnica da USP;

a.3 Levantamento de referências bibliográficas.

Atividades previstas para a Revisão bibliográfica:

b.1 Leitura e análise de referências apontadas no levantamento de dados;

b.2 Organização de informações de modo a favorecer a operacionalização pelo desenvolvimento do programa.

Divisão das atividades para a implementação:

c.1 Determinação de funcionalidades desejadas para a ferramenta desenvolvida;

c.2 Determinação do *software* a ser utilizado para a implementação do programa;

c.3 Determinação de procedimento realizado no programa; e

c.4 Redação das rotinas que compõem o programa.

Descrição de atividades contidas na verificação do programa:

d.1 Teste com exemplo estudado na literatura

d.2 Comparação com resultados fornecidos por outros *softwares* livres

d.3 Determinação de limitações do programa

Nesta etapa, as correções são efetuadas pontualmente, conforme se identifiquem as necessidades de modificar algo ou *bugs* no programa.

Por último, visa à confecção de documentação necessária para a disciplina PMR2550, através da determinação de modelos a serem utilizados, na forma de um relatório final e de um artigo, para descrição de atividades realizadas no período.

## 4 Atividades realizadas

Grande dificuldade foi colocada no Levantamento de dados, desde o início do projeto, o que necessitou a consagração de grande parte do tempo a sua execução. Após a realização da primeira versão da ferramenta, três possíveis desdobramentos do trabalho foram discutidos com a pesquisadora interessada no tema do trabalho. Entretanto, as alterações foram balanceadas internamente com as outras atividades previstas para o período. Todas as atividades previstas foram realizadas.

Através do estudo bibliográfico, puderam-se verificar as formas de sistemas com atraso. Verificou-se, também, que não existem rotinas adequadas disponíveis nos *softwares Matlab* e *Scilab* para o cálculo dos pólos, o que justifica a motivação inicial de se incorporar aproximações de dimensão finita para sistemas de dimensão infinita, desenvolvendo-se, em seguida, uma ferramenta que execute essas aproximações, para se utilizar as rotinas disponíveis. Foi possível, também, a derivação de equações, apoiada sobre a extensa bibliografia, as quais são utilizadas no programa. Mesmo que todos os pólos possam ser identificados, a preocupação se volta somente à localização de pólos instáveis para a determinação da estabilidade do sistema.

A seguir, é apresentada a abordagem teórica, que permitiu a derivação de condições e expressões utilizadas no cálculo dos zeros, assim como a definição de funcionalidades a serem apresentadas pelo programa. Apresenta-se, também, a rotina implementada pelo programa escrito em código *Matlab*. Finalmente, apresentam-se os resultados dos testes realizados sobre um exemplo fornecido pela literatura e com resultados de outros *softwares* livres.

# 5 Abordagem teórica

## 5.1 Sistemas com atraso

### 5.1.1 Generalidades

Numerosos sistemas são modelados por equações diferenciais com atrasos (ver [2], [14], [13] para numerosos exemplos): a evolução do processo não depende somente do estado presente, mas também dos estados anteriores. Esses sistemas, cuja dimensão real do estado é infinita, requerem técnicas específicas de análise e comando (ver [1], [13], [16], [4] e [7]).

Consideram-se aqui os sistemas lineares que têm um número finito de atrasos (discretos positivos) sobre o estado, a entrada e a saída, e que são descritos pelas equações diferenciais lineares da forma

$$(S)\begin{cases} \dot{x}(t)+\sum_{i=0}^{n} E_i \dot{x}(t-\mu_i) = \sum_{i=0}^{k} A_i x(t-t_i) + \sum_{i=0}^{m} B_i u(t-\tau_i) \\ y(t) = \sum_{i=0}^{l} C_i x(t-\sigma_i) + \sum_{i=0}^{p} d_i u(t-\nu_i) \end{cases} \tag{1}$$

onde $x(t) \in R^n, u(t), y(t) \in R, A_i, B_i, C_i$ são matrizes $n \times n$, $n \times 1$ e $1 \times n$ e $d_i \in R$.

A função de transferência G de (S) é uma função transcendente dada por

$$G(s) = \left( \sum_{i=0}^{m} B_i e^{-s\tau_i} \right) \left( sI + s\sum_{i=0}^{n} E_i e^{-s\mu_i} - \sum_{i=0}^{k} A_i e^{-st_i} \right)^{-1} \left( \sum_{i=0}^{l} C_i e^{-s\sigma_i} \right) + \sum_{i=0}^{p} d_i e^{-s\nu_i} \tag{2}$$

que pode ser reescrita como

$$G(s) = \frac{h_2(s)}{h_1(s)} \tag{3}$$

onde

$$h_1(s) = \sum_{i=0}^{n_1} p_i(s) e^{-\gamma_i s} \quad e \quad h_2(s) = \sum_{i=0}^{n_2} q_i(s) e^{-\beta_i s} \tag{4}$$

com $0 = \gamma_0 < \gamma_1 \ldots < \gamma_{n_1}$, $0 \le \beta_0 < \beta_1 \ldots < \beta_{n_2}$, $p_i$ sendo polinômios de grau $\delta_i$, tal que $\delta_i \le \delta_0$ para todo $i \neq 0$, e $q_i$ sendo polinômios de grau $d_i < \delta_0$ para todos os $i$ (sistema próprio).

Os sistemas com atraso se dividem em duas classes:

- Os sistemas de tipo retardado: $\delta_i < \delta_0$ para todo $i \neq 0$. Essa classe de sistemas foi analisada pela primeira vez por Bellman e Cooke [2]. Eles mostraram que esses sistemas possuem um número finito de pólos $(s_n)$ no semi-plano direito. Os pólos $(s_n)$ satisfazem $\mathrm{Re}\, s_n \to -\infty$;
- Sistemas do tipo neutro: $\delta_i = \delta_0$ para ao menos um $i \neq 0$. Os pólos se localizam em uma banda centrada em torno do eixo imaginário.

Os atrasos que são todos múltiplos de um mesmo inteiro são chamados de atrasos comensuráveis. Os outros atrasos são chamados de atrasos gerais. Para os sistemas do tipo neutro, trabalha-se com retardos do tipo comensurável, uma vez que, no caso geral, o comportamento assintótico das cadeias de pólos não é sempre determinável.

### 5.1.2 Estabilidade de sistemas com atraso

Para o caso geral de sistemas do tipo (1), aos quais se adiciona as condições iniciais $x(0) = x_{00} \in R^n$ e $x = x_0$ sobre $[-T, 0]$, dado T o maior dos atrasos físicos, ou seja, $T = \max\left(\max_{i\in[0,n]} \mu_i, \max_{i\in[0,k]} t_i\right)$, diferentes noções de estabilidade são consideradas, dado que se tem uma abordagem temporal ou entrada-saída (em freqüência) do problema.

O sistema é dito assintoticamente estável se

$$\forall (x_{00}, x_0) \in R^n \times L^2(-T, 0), \lim_{t\to+\infty} |x(t)|_{R^n} = 0. \tag{5}$$

O sistema é dito exponencialmente estável se existe $M \geq 0$ e $\omega > 0$ tais que

$$\forall (x_{00}, x_0) \in R^n \times L^2(-T, 0), \forall t \geq 0, |x(t)|_{R^n} \leq M e^{-\omega t}\left(|x_{00}| + \|x_0\|_{L^2}\right). \tag{6}$$

O sistema é dito $H_\infty$-estável se

$$\|G\|_{H_\infty} < \infty. \tag{7}$$

Se o sistema for $H_\infty$-estável, toda entrada $u$ no $L_2$ fornece uma saída no $L_2$. Os sistemas com atraso de tipo retardado $H_\infty$-estável são se e somente se eles não apresentam pólos no

$\{\mathrm{Re}\, s \geq 0\}$ (essa última condição sempre é necessária, mas raramente suficiente para um sistema qualquer). Tem-se, portanto, nesse caso, equivalência entre estabilidade assintótica, estabilidade $H_\infty$ e estabilidade exponencial. Para sistemas de tipo neutro, a estabilidade exponencial implica a estabilidade $H_\infty$.

Tem-se a estabilidade em malha fechada, mostrada na figura 1, se as funções de transferência $(I+PC)^{-1}, P(I+PC)^{-1}$ *e* $C(I+PC)^{-1}$ são em $H_\infty$. No âmbito desse estudo, interessa-se pela estabilidade $H_\infty$.

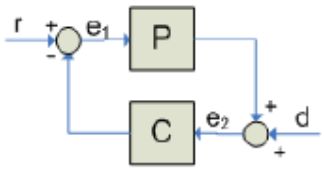


Figura 1: Malha fechada padrão.

Uma característica desses sistemas é que eles têm um número infinito de pólos, cuja localização depende continuamente dos atrasos (unicamente para valores estritamente positivos de atrasos no caso dos sistemas do tipo neutro).

O método de Walton e Marshall (ver [10] e [6]) pode ser utilizado para concluir sobre a presença de pólos instáveis de sistemas, cuja função de transferência tem um denominador do tipo $F(s,h) = A(s) + C(s)e^{-sh}$. Ele permite determinar os valores dos atrasos, que desestabilizam o sistema e, então, que eventualmente o re-estabilizam etc. O procedimento consiste em 3 etapas:

1. Análise da estabilidade para $h = 0$, ou seja, determinar o número de zeros no semi-plano direito do sistema sem atrasos, $F(s,0)$;
2. Análise dos $h$ positivos infinitamente pequenos e localização de novas raízes (em número infinito), que aparecem no plano complexo;
3. Localização dos $h$ positivos para os quais existem zeros da função $F(s,h)$, que se encontram sobre o eixo imaginário.

Nesse caso, procuram-se os valores de $h$ e $\omega$ tal que $F(i\omega,h) = 0$, e se continua determinando se esses zeros tocam no eixo ou se eles o cruzam. Continua-se estudando o movimento dos zeros, de modo a se determinar as regiões de instabilidade, ou seja, quando os zeros não se encontram todos no semi-plano esquerdo aberto.

Determinam-se os pontos potenciais, cruzando o eixo imaginário e analisando o polinômio $W(\omega^2) = A(i\omega)A(-i\omega) - C(i\omega)C(-i\omega)$, de mesmo grau que $A(s)$. Quando não se tem raízes positivas de $W(\omega^2) = 0$, não existe valor de $h$ para o qual $F(i\omega, h) = 0$, e, então, não existe mudança de estabilidade. Para todo $\omega \neq 0$ que satisfaz $W(\omega^2) = 0$, existe um $h$ real positivo tal que $F(i\omega, h) = 0$ dado por

$$\cos \omega h = \mathrm{Re}\left\{-\frac{A(i\omega)}{C(i\omega)}\right\}, \quad \sin \omega h = \mathrm{Im}\left\{\frac{A(i\omega)}{i\omega}\right\}. \tag{8}$$

Tem-se, portanto, que se $h_0(\omega)$ designa o menor valor de $h$, para um valor particular de $\omega$, tem-se um número infinito de valores de $h$ que satisfazem $W(\omega^2) = 0$, para cada $\omega$, dados por

$$h = h_0(\omega) + \frac{2\pi q}{\omega}, \quad q = 0, 1, 2, \ldots \tag{9}$$

O caso de sistemas do tipo neutro é mais complexo e será introduzido na subseção seguinte.

### 5.1.3 Sistemas com atraso do tipo neutro

Encontra-se na literatura ([5]) subclasses de sistemas neutros:

I. Somente um atraso no denominador:

Quando (3) é representado por

$$G(s) = \frac{q_0(s)}{p_0(s) + p_1(s)e^{-\gamma_0 s}}, \tag{10}$$

seja o real não nulo $\alpha = \lim_{|s|\to\infty} \frac{p_0(s)}{p_1(s)}$, encontram-se os seguintes casos:

a. $|\alpha| < 1$, para o qual G(s) tem um número infinito de pólos instáveis, assintóticos a uma linha vertical $\mathrm{Re}\, s \approx -\log \frac{|\alpha|}{h}$ no semi-plano direito, e então G(s) não pode ser $H_\infty(C_\infty)$;

b. $|\alpha| > 1$, para o qual os pólos de grande módulo de G(s) são assintóticos a uma linha vertical estritamente no semi-plano esquerdo e, então, G(s) tem um número finito de pólos instáveis, e se ela não tem nenhum, então G(s) é $H_\infty$;

c. $|\alpha| = 1$, para o qual os pólos de G(s) são assintóticos ao eixo imaginário.

No caso que $|\alpha| = 1$, supondo

$$\frac{p_0(s)}{p_1(s)} = \alpha + \frac{\beta}{s} + \frac{\gamma}{s^2} + O\left(\frac{1}{s^3}\right) \text{ para } |s| \to \infty \tag{11}$$

para $\alpha$, $\beta$ e $\gamma$ constantes, o sistema apresenta o seguinte comportamento:

a. Se $\frac{\gamma}{\alpha} > \frac{\beta^2}{2}$, o sistema tem um número infinito de pólos instáveis;

b. Se $\frac{\gamma}{\alpha} < \frac{\beta^2}{2}$, o sistema tem um número infinito de pólos estáveis. Ele não pode ter um número finito de pólos instáveis, e se ele não tem nenhum, então G(s) é $H_\infty$ se e somente se $\deg(p_0) \geq \deg(q_0) + 2$;

c. Se $\frac{\gamma}{\alpha} = \frac{\beta^2}{2}$, a condição $\deg(p_0) \geq \deg(q_0) + 2$ continua necessária à estabilidade do sistema.

II. Caso geral

Para o caso de vários atrasos, (3), para $\deg(p_0(s)) = m$, reagrupam-se os coeficientes do denominador em $s^m$, tendo o polinômio $p_0^m + p_1^m e^{-s_1 T} + \ldots + p_n^m e^{-s_n T}$. Efetuando uma mudança de variáveis ($e^{-T} = X$), se a função não tem zeros em $\text{Re}\, s \geq 0$ (equivalente ao fato que X não tenha pólos no circulo unitário), então G(s) tem um número finito de pólos instáveis.

### 5.2 Fatoração coprima e fatores de Bézout para sistemas com atraso

Dado (3) e (4), para sistemas do tipo retardado, existe uma função racional r(s), tal que

$\left(\frac{h_2(s)}{r(s)}, \frac{h_1(s)}{r(s)}\right)$ é uma fatoração coprima de G sobre $H_\infty$ (veja introdução em [17]). No caso que $h_1(s)$ e $h_2(s)$ não contenham mais que $\delta_0$ zeros instáveis comuns, então r(s) pode ser tomado como um polinômio, como $r(s) = (s+1)^{\delta_0}$. Isso também é válido para sistemas neutros que têm um número finito de pólos em $\{\mathrm{Re}\, s > -a\}$, onde $a > 0$.

A aproximação pela fatoração coprima dos sistemas com atraso é uma boa aproximação segundo $H_\infty$ (ver Corolário 5.3 de [15]).

### 5.3 Aproximações de sistemas com atrasos do tipo "*Dead-time*"

O operador atraso é um operador de decalagem fundamental em $H_2$, designado por $S_h$ e definido como $(S_h f)(s) = e^{-hs} f(s),\ f \in H_2$. Em [12], encontram-se os operadores de Kautz, Laguerre e Padé-2. Utiliza-se a aproximação de dimensão finita Padé-2, a qual é uma decalagem múltipla do tipo Kautz de multiplicidade 2*n,* dada por

$$S_P^{(n)} f = \left( \frac{1 - \frac{hs}{2n} + \frac{1}{3}\left(\frac{hs}{2n}\right)^2}{1 + \frac{hs}{2n} + \frac{1}{3}\left(\frac{hs}{2n}\right)^2} \right)^n f, \quad f \in H_2, \tag{12}$$

Seja o sistema com atrasos do tipo "*dead-time*" $G(s) = e^{-hs} R(s)$, onde $h > 0$ e $R(s) \neq 0$ é uma função de transferência racional estável e estritamente própria com grau relativo m, [12] dá

$$\left\| G - S_P^{(n)} R \right\|_\infty = \max\left\{ O\left( n^{\frac{-4m}{5}} \right), O\left(n^{-4}\right) \right\}. \tag{13}$$

Padé-2 apresenta, então, uma aproximação $H_\infty$ de boa ordem (em comparação a uma aproximação racional de ordem $O(n^{-m})$) e de fácil implementação.

O teorema 4.2 (ver [15]) fornece os erros da aproximação dos pólos. Supondo que $G(a) = 0$, onde G é analítico com $|G(s)| \le M$ para $|s - a| \le R$, e que $G^m(a)$ é a primeira derivada que não se

anula em a, e que $\beta > 0$ e $0 < \delta < R$, são tais que

$$\left\| G - S_P^{(k)} R \right\|_\infty \leq \beta < \frac{\delta^m \left| E^{(m)}(a) \right|}{m!} - \frac{M\delta^{m+1}}{R(R-\delta)}. \tag{14}$$

$S_P^{(k)} R$ tem, portanto, m zeros $a_{k,1}, \ldots, a_{k,m}$ com $\left| a_{k,i} - a \right| < \delta$ para cada i.

# 6 Definição da solução

## 6.1 Definição das funcionalidades

As qualidades definidas como essenciais, durante o desenvolvimento do programa, são a portabilidade, a flexibilidade, a facilidade de utilização e a validade dos resultados.

Para assegurar a portabilidade, todo o programa foi escrito em código *Matlab*, escolhido em detrimento do *Scilab*, pela maior riqueza das rotinas de interface.

Para a facilidade de utilização, de um lado, pensou-se em um programa *user friendly.* Por essa razão, foi introduzida uma interface intuitiva, convivial e ricamente comentada, que guia o usuário através da listagem de mensagens, que ilustram a natureza do sistema. De outro lado, para a facilidade à entrada de dados e para a recuperação de resultados de uma instância do programa, integrou-se a leitura e escrita em um arquivo texto, *results.* Esse arquivo mantém os dados – atrasos $\gamma_i$ e $\beta_i$, polinômios $p_i(s)$ e $q_i(s)$, o parâmetro $\varepsilon$ e as normas $H_\infty$ das aproximações $\frac{h_1(s)}{(s+1)^{\delta_0}}$ e $\frac{h_2(s)}{(s+1)^{\delta_0}}$, assim como os pólos instáveis – de uma instância anterior do programa, utilizados na próxima instância.

Para a flexibilidade, foi determinado que o usuário tivesse a liberdade de entrar em qualquer sistema do tipo (4), desde que com um número finito de $\gamma_i$, $\beta_i$, e de graus de $p_i(s)$ e $q_i(s)$. Também, escolheu-se o parâmetro $\varepsilon$, o qual é utilizado como parâmetro de parada aos cálculos de convergência feitos durante todo o programa. Dado $x_i$ o valor de uma variável $x$ na iteração $i$,

$$|x_i - x_{i-1}| < \varepsilon. \tag{15}$$

Finalmente, para a validade dos resultados obtidos, trabalha-se com o caso onde existe somente um número finito de pólos instáveis, pois se utilizam aproximações de dimensão finita para aproximar os zeros instáveis de (3). Nesses casos, a execução do programa é interrompida e o motivo é exposto ao usuário. Por sinal, não existe literatura disponível sobre a dinâmica da localização de todos os pólos possíveis, portanto, aplicou-se esse procedimento em todos os casos de cadeias infinitas de pólos instáveis.

Além disso, devido à perda de precisão de *Matlab*, à medida que se aumenta a ordem dos polinômios aproximados para a convergência da aproximação dos pólos instáveis, insere-se o cálculo da norma $H_\infty$ da aproximação obtida como parâmetro de parada, assim como para a informação do usuário quanto à qualidade da aproximação utilizada.

## 6.2 Rotina

Sistemas de dimensão infinita do tipo (3) são aproximados por sistemas de dimensão finita de ordem inferior. A aproximação ótima para esses sistemas, segundo $H_\infty$, é a aproximação Padé-2, apresentada na seção 6.3, a qual fornece um erro máximo dado por (13).

Pode-se aplicar aos sistemas do tipo "*dead-time*", $e^{-sh}R(s)$, onde $R(s)$ é estritamente próprio. Aproximam-se, dessa forma, os polinômios $h_1(s)$ e $h_2(s)$, utilizando a aproximação de fatores coprimos (seção 6.2), com $\delta_0 = \max(\deg p_0, \deg q_0)+1$. Dessa forma, cada termo $R_{pi}e^{-\gamma_i s}$ e $R_{qi}e^{-\beta_i s}$ é do tipo "*dead-time*", com $R_{pi} = \frac{p_i(s)}{(s+1)^{\delta_0}}$ e $R_{qi} = \frac{q_i(s)}{(s+1)^{\delta_0}}$ estritamente próprios. Sejam $\frac{h_1(s)}{(s+1)^{\delta_0}}$ e $\frac{h_2(s)}{(s+1)^{\delta_0}}$ respectivamente D e *N,* reescrevem-se (3) como

$$G(s) = \frac{N}{D}. \tag{16}$$

Aplicando a aproximação Padé-2 sobre os sistemas do tipo "*dead-time*" de (16), obtém-se

$$G(s) = \frac{N_k}{D_k}, \tag{17}$$

que é a aproximação ótima de dimensão finita, para o caso que se tem um número finito de pólos instáveis. Pode-se, dessa forma, utilizar as rotinas disponíveis em *Matlab* e *Scilab* para os sistemas do tipo (17).

Analisam-se as raízes do sistema e a norma $H_\infty$ para cada $n$ utilizado na aproximação Padé-2. Calculam-se as raízes de $N_k$ e $D_k$, para o caso em que não se prevê, na literatura, uma situação com número infinito de raízes. Para o $N_k$, quando se tem um número infinito de raízes, aconselha-

se ao usuário efetuar o cálculo desses zeros à mão, enquanto se continua com o cálculo das raízes de $D_k$. Esse procedimento é encorajado, de forma a evitar raízes comuns entre o numerador e o denominador, uma vez que o sistema pode apresentar uma dinâmica não prevista na literatura.

Apresenta-se, aqui, somente o procedimento para $D$ e $D_k$, pois o mesmo procedimento é utilizado para $N$ e $N_k$. Desse modo, calculam-se as raízes de $D_k$ para a iteração $n$. Os cálculos são interrompidos uma vez que se verifica que a norma relativa (18) entre duas iterações consecutivas ou a diferença máxima entre as raízes (19) é inferior à $\varepsilon$, dado $k$ o número de raízes.

$$\frac{\left|D - D_k\right|_{H_\infty}}{\left|D\right|_{H_\infty}} < \varepsilon \tag{18}$$

$$\max_k \left| x_{n,k} - x_{n-1,k} \right| < \varepsilon \tag{19}$$

O procedimento ideal é de parar pela convergência da norma $H_\infty$, que contabiliza a convergência da aproximação em todo o semi-plano direito. Entretanto, o critério de convergência dos pólos é, também, adequado, pois se tem uma interdependência entre as localizações dos pólos e a aproximação ótima, segundo $H_\infty$ (veja seção 6.3, teorema 4.2 de [15]).

### 6.3 Descrição do programa

O programa foi estruturado em uma organização recursiva, para obter benefícios da semelhança entre o denominador e o numerador, quanto à entrada de dados e ao seu tratamento na leitura e escrita no arquivo *results*, a aproximação por Padé-2 e a fatoração coprima. O programa apresenta uma rotina principal recursiva, que faz a interface gráfica e a sincronização entre as diversas partes do programa.

A partir da inicialização, tem-se a leitura dos dados da iteração anterior, estocados no arquivo *results.* O programa segue pedindo valores utilizados na iteração presente. Começa-se com o pedido de épsilo*n,* seguido da inicialização do denominador $h_2(s)$ pela entrada dos atrasos $\gamma_i$ e dos polinômios $p_i(s)$. Verifica-se a presença de $\gamma_0 = 0$, sem a qual se reinicializa o procedimento. Em seguida, realiza-se a similar inicialização do numerador, pela entrada dos atrasos $\beta_i$ e dos

polinômios $q_i(s)$.

Após a entrada completa de dados do sistema na inicialização, segue-se com a rotina *calcul.m*, onde se começa a identificação do tipo de sistema, entre sistemas do tipo retardado ou sistema do tipo neutro. O programa é, dessa forma, subdividido em dois grandes grupos, o que é utilizado para efetuar a triagem de casos particulares, já estudados na literatura disponível (veja capítulo 6). Definem-se, assim, os intervalos para os quais o programa agirá – funções de transferência com um número finito de pólos instáveis.

Para sistemas do tipo neutro, o programa é guiado pelas considerações da seção 6.1.3. Assim, distingue-se a presença de um ou vários atrasos e realiza-se o cálculo de $\alpha, \beta$ e $\gamma$ (pela rotina *calcul-beta-gamma.m*). Sempre expondo a situação do sistema ao usuário, o programa é reinicializado quando se encontra no caso de infinitas raízes instáveis, ou quando não se tem exponencial no denominador ou, ainda, se o sistema não é próprio.

Realiza-se, enfim, o cálculo dos pólos, o que é feito seguindo o procedimento descrito na seção 7.2. O procedimento, descrito a seguir, é executado sempre para o denominador e é, também, executado para o numerador, quando este não apresenta infinitas raízes instáveis.

O parâmetro $\delta_0$ da fatoração coprima é encontrado e segue-se com a aproximação Padé-2 dos sistemas do tipo (16), com ajuda da função *approx-pade.m*. Encontra-se o grau *n* da aproximação, tendo em conta a convergência, seja das raízes, seja da norma relativa, conforme descrito na seção 7.2. No caso em que se calculam as raízes de $N_k$, interrompem-se os cálculos quando $N_k$ e $D_k$ apresentam raízes comuns ou muito próximas (de distância inferior a $\varepsilon$), uma vez que não se pode prever o comportamento do sistema.

Existem casos de infinitas raízes instáveis, não previstos na literatura disponível. Nesses casos, a aproximação por raízes ou pela norma relativa não converge. De fato, quando se aumenta o grau *n*, faz-se convergir às raízes de baixo módulo absoluto. Entretanto, nem todas as raízes obedecem à natureza assintótica vertical da localização esperada.

*Matlab* apresenta alguns erros também, uma vez que se trabalha com graus superiores a 65, quando se utiliza um único atraso igual a 1. Esse valor diminui quando se aumenta a magnitude e o número de atrasos presentes. Desse modo, limita-se o *loop* de convergência quando $n > 50$ e adotam-se raízes com módulo absoluto inferior a 100.

Algumas rotinas auxiliares são utilizadas para o tratamento e apresentação de polinômios, para o fornecimento da aproximação Padé-2, uma vez que a função disponível em *Matlab* não é

programada para qualquer que seja a entrada *n* e h. Também, escreveu-se uma rotina que realiza a manipulação e transformação de variáveis do tipo 'tf' em uma variável do tipo 'sym'. A rotina *csort.m* foi obtida no *site* oficial de *Matlab* e efetua a organização de vetores por um parâmetro escolhido ('real', 'complexo' ou 'abs').

A rotina *norma.m* foi concebida para calcular a norma $H_\infty$ em *Matlab*, para sistemas do tipo (3), fornecendo uma ajuda ao usuário, através de gráficos. A rotina divide o intervalo de cálculo e auto-refina seu passo sobre o eixo imaginário, enquanto o passo é maior que $\varepsilon$. É, portanto, uma rotina de custo em tempo proporcional ao $\varepsilon$ utilizado.

Testes foram realizados e a rotina mostrou-se confiável. Por [12], foram realizadas as aproximações de Laguerre, Kautz e Padé-2 propostas e os erros $H_\infty$ foram calculados. Verificou-se a correspondência entre o artigo e os resultados obtidos por Laguerre e Kautz. Devido às diferenças encontradas por Padé-2, os cálculos foram refeitos, auxiliados por métodos gráficos, e consultou-se o autor do artigo. Pela análise gráfica e pelo cálculo da norma $H_\infty$, os resultados obtidos pela rotina, aqui desenvolvida, mostraram-se corretos.

Finalmente, retornam-se os pólos encontrados à rotina principal e procede-se escrevendo os pólos no arquivo *results* e apresentando duas janelas com os pólos do sistema e com o gráfico dos pólos, efetuado por uma rotina específica. O programa espera que o usuário toque uma das janelas para terminar a execução.

# 7 Validação da ferramenta

## 7.1 Comparação com resultados da literatura

O artigo [15] calcula um exemplo numérico que provém na verdade de [9] e [8]. Adotou-se o procedimento descrito no capítulo 7, testando esse exemplo no programa realizado. Verificou-se uma boa correlação entre os resultados obtidos e aqueles fornecidos na literatura. Apresenta-se, a seguir, toda a execução para esse caso específico.

Os pólos instáveis "exatos", calculados em [8], são 5.002224 e 5.999994. Em [9], estimou-se a parte instável de G, utilizando uma transformada rápida de Fourier a 2048 pontos e aproximando $G_s$ por um sistema de 15ª ordem, obtendo-se os pólos aproximados 5.0035 e 5.9981. Os melhores resultados de [15] são os pólos aproximados 5.0026 e 6.0000, obtidos para grau $n = 4$ de Padé-2.

A função G(s) em questão é a função a seguir.

$$G(s) = \frac{20\left(6e^{-2s} + 2e^{-s} - 6\right)}{6s^2 + \left(6e^{-2s} - 2e^{-s} - 66\right)s - \left(2e^{-3s} + 30e^{-2s} - 12e^{-s} - 180\right)} \tag{20}$$

Após a escolha de $\varepsilon$, entra-se com os atrasos $\gamma_i$ e os polinômios $p_i(s)$ do denominador, seguidos dos atrasos $\beta_i$ e os polinômios $q_i(s)$ do numerador, como presente na tabela 1.

| | Atrasos | Polinômios |
|---|---|---|
| Denominados | $\gamma_0 = 0$<br>$\gamma_1 = 1$<br>$\gamma_2 = 2$<br>$\gamma_3 = 3$ | $p_0(s) = 6s^2 - 66s + 180$<br>$p_1(s) = -2s + 12$<br>$p_2(s) = 6s - 30$<br>$p_3 = -2$ |
| Numerador | $\beta_0 = 0$<br>$\beta_1 = 1$<br>$\beta_2 = 2$ | $q_0(s) = -120$<br>$q_1(s) = 40$<br>$q_2(s) = 120$ |

Tabela 1: Atrasos e polinômios de G(s).

Escolhendo $\varepsilon$ = 0.001, o programa esclarece que se trata de um sistema do tipo retardado e,

através de uma mensagem de erro, que o numerador tem infinitas raízes instáveis. O numerador constitui um sistema neutro com vários atrasos. Efetuando a mudança de variável $e^{-sh} = X^h$, obtém-se a equação

$$h_2(s) = 20\left(6e^{-2s} + 2e^{-s} - 6\right) \quad \rightarrow \quad h_2(X) = 120X^2 + 40X - 120 \tag{21}$$

que apresenta as raízes 0.8471270883 e -1.180460422, sendo uma contida no círculo unitário, o que confirma a presença de um número infinito de raízes instáveis (ver seção 6.1.3). O numerador apresenta esse comportamento para qualquer que seja o valor de $\varepsilon \in R$.

O programa não continua com o cálculo das raízes do numerador, mas somente com o cálculo das raízes do denominador. Entretanto, o usuário é aconselhado a efetuar à mão o cálculo das raízes do numerador, de modo a evitar o caso de raízes comuns ou muito próximas entre o numerador e o denominador.

Utilizando os métodos de resolução das funções analíticas ([11]), seja $e^s = w$, a solução da equação é

$$s = \log|w| + i \arg w. \tag{22}$$

Dessa forma, as raízes do numerador são -0.165905 e 0.165905 + (3.141593 + 2k)i, $k \in Z$. Verifica-se a presença de um número infinito de raízes, como previsto na literatura. Prossegue-se comparando as raízes do denominador, após a execução completa do programa.

O programa continua com o cálculo das raízes do denominador. Após verificar que o denominador tem um número finito de raízes, o programa fornece D(s) e $D_k$(s). A seguir, a aproximação D(s) é apresentada. Entretanto, não se apresenta a aproximação $D_k$(s), uma vez que se constitui como uma aproximação com numerador e denominador de ordem 35, com coeficientes de ordem 30.

$$D(s) = \frac{6s^2 - 66s + 180}{(s+1)^3} + \frac{(-2s+12)e^{-s}}{(s+1)^3} + \frac{(6s-30)e^{-2s}}{(s+1)^3} - \frac{2e^{-3s}}{(s+1)^3}. \tag{23}$$

O *loop* é interrompido pela convergência dos pólos, quando $n$ = 4. A norma $H_\infty$ relativa da aproximação do denominador é 0.0012, sendo a norma $\left|D - Dk\right|_{H_\infty} = 0.1850$. Os pólos aproximados do sistema são 5.002273 e 5.999986, que são aproximações melhores que aquelas fornecidas por [15] e [9]. O tempo de cálculo é de 501.40s. Verifica-se, dessa forma, que não se tem

raízes em comum ou muito próximas entre o numerador e o denominador, o que não invalida as aproximações feitas.

O gráfico dos pólos do sistema, fornecido pelo programa no final da sua execução, assim como as outras janelas abertas pela interface durante a execução, encontram-se no Anexo B.

O parâmetro de parada foi a convergência dos pólos e não a convergência da norma $H_\infty$ relativa. Com algumas pequenas mudanças no programa, é possível realizar simulações utilizando um único parâmetro de parada, quer seja quanto à norma relativa, quer seja quanto à convergência das raízes, sempre com $\varepsilon = 0.001$. A comparação com o caso anterior é dada na tabela 2. Utilizando somente as raízes como parâmetro de parada, duas possibilidades apresentam-se: com (*) e sem (**) a comparação final da norma $H_\infty$ relativa. Verifica-se, com essa separação, que o principal responsável pelo tempo de cálculo é o cálculo da norma $H_\infty$. Este cálculo, dado por um *loop* que auto-refina seu passo enquanto este for superior a $\varepsilon$, é incorporado ao *loop* de convergência de *n,* aumentando o custo total em tempo de cálculo do programa.

| Critérios de Parada | Raízes | Tempo de cálculo | n | $\frac{\lvert D - Dk\rvert_{H_\infty}}{\lvert D\rvert_{H_\infty}}$ |
|---|---|---|---|---|
| Norma e raízes | 5.002273<br>5.999986 | 501.40s | 4 | 0.0012 |
| Somente norma | 5.002257<br>5.999988 | 666.34s | 5 | $8.3564*10^{-4}$ |
| Somente raízes | 5.002273<br>5.999986 | 186.36s (*)<br>10.14s (**) | 4 | 0.0012 |

Tabela 2: Comparação dos resultados para diferentes critérios de parada.

Todas as três modalidades citadas acima são rotinas com implementação possível e fornecem, para o caso das equações com um número finito de raízes instáveis, melhores aproximações que aquelas fornecidas na literatura.

Outros testes foram realizados para as três modalidades, com o objetivo de avaliar a influência do parâmetro $\varepsilon$ na performance do programa.

O parâmetro *dif* é o parâmetro considerado quando se comparam as raízes. É dado por

$\max(|p_{1,n} - p_{1,n-1}|, |p_{2,n} - p_{2,n-1}|)$. Além disso, conhecendo os pólos exatos, dados por [8], tem-se o parâmetro *dist*, dado por $\sqrt{(p_{1,n} - 5.002224)^2 + (p_{2,n} - 5.999994)^2}$. Tem-se, também, a norma relativa $H_\infty$ (18), que é representada por *dif2*.

Para a terceira modalidade, onde testes de convergência foram feitos somente com as raízes de iterações consecutivas, efetuaram-se testes para os dois casos explicados acima (com (*) e sem (**) a comparação da norma no final da execução). Os resultados encontram-se na tabela 3.

| $\varepsilon$ | dist | Raízes | Tempo(*) | Tempo(**) | n | *dif2* | *dif* |
|---|---|---|---|---|---|---|---|
| $10^{-1}$ | 5.671868E-04 | 5.002273<br>5.999986 | 145.57 | 8.10 | 3 | 1.7624E-03 | 1.2723E-02 |
| $10^{-2}$ | 5.009629E-05 | 5.002004454<br>6.000516973 | 192.37 | 9.44 | 4 | 1.1563E-03 | 5.3145E-04 |
| 10-3 | 5.009629E-05 | 5.002273374<br>5.999985522 | 239.90 | 12.87 | 4 | 1.1564E-03 | 5.3145E-04 |
| 10-4 | 3.324945E-05 | 5,002256739<br>5,999988194 | 372.13 | 11.37 | 5 | 8.3564E-04 | 1.6635E-05 |
| 10-5 | 2.472935E-05 | 5,002248609<br>5,999991562 | 606.18 | 11.86 | 6 | 6.3366E-04 | 8.1297E-06 |
| 10-6 | 1.845368E-05 | 5,002242447<br>5,999993491 | 1484.02 | 16.50 | 9 | 3.3770E-04 | 9.0667E-07 |
| 10-7 | 1.720850E-05 | 5,002241207<br>5,999993798 | 4060.62 | 31.24 | 14 | 1.0740E-04 | 8.7484E-08 |
| 10-8 | 1.710352E-05 | 5,002241103<br>5,999993822 | 4144.23 | 44.86 | 16 | 0.0000E+00 | 4.3661E-08 |

Tabela 3: Resultados de simulações para a modalidade “Somente raízes”.

Verifica-se, pela análise da tabela 3, o que já se esperava: as raízes dependem unicamente do grau *n*, entretanto, o tempo (*) não é somente dependente de *n*, mas, principalmente, de $\varepsilon$, utilizado como valor de parada nos testes efetuados em todo o programa.

Verifica-se que, conforme se refina $\varepsilon$, obtêm-se raízes mais próximas às raízes exatas (veja

raízes e *dist*), assim como a diminuição de *dif2* e *dif*. Ocorre, portanto, o comportamento assintótico convergente assim que se aumenta o grau *n*.

Verifica-se, também, uma perda da precisão de *Matlab* para $\varepsilon$ superiores ou iguais a $10^{-8}$. Analisando *dist*, verifica-se que a norma *dif2* não poderia ser nula. Isso acontece uma vez que se trabalha com situações críticas limites, utilizando polinômios de grau muito alto (superior a 100), com coeficientes de alto módulo (ordem superior a $10^{180}$), ao mesmo tempo em que se trabalha com valores de $\varepsilon$ de baixo módulo.

Efetuando agora simulações para o primeiro caso, com a análise simultânea da convergência de raízes e da norma $H_\infty$ relativa, obtêm-se os resultados da tabela 4.

| $\varepsilon$ | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ |
|---|---|---|---|---|---|---|
| *dist* | 2.1221E-01 | 1.3284E-02 | 5.0096E-05 | 3.3249E-05 | 2.4729E-05 | 1.8454E-05 |
| Raízes | 4.869814<br>5.834158 | 4.989282<br>6.002986 | 5.002273374<br>5.999985522 | 5.002256739<br>5.999988194 | 5.002248609<br>5.999991562 | 5.002242447<br>5.999992491 |
| Tempo | 58.56 | 147.43 | 545.89 | 926.58 | 1632.67 | 3814.66 |
| *n* | 1 | 2 | 4 | 5 | 6 | 9 |
| *dif2* | 1.0908E-02 | 3.5261E-03 | 1.1564E-03 | 8.3564E-04 | 6.3366E-04 | 3.3770E-04 |
| *dif* | 1.6584E-01 | 1.6883E-01 | 5.3145E-04 | 1.6635E-05 | 8.1297E-06 | 9.0667E-07 |

Tabela 4: Resultados de simulações para a modalidade "Norma e raízes".

Analisando a tabela 4, verifica-se que o programa só foi suspenso, pelo critério da norma, quando $\varepsilon = 10^{-1}$ e $\varepsilon = 10^{-2}$. Para todos os outros casos, o critério empregado foi a distância máxima entre as raízes de iterações consecutivas. Nota-se, assim, que a localização das raízes converge mais rapidamente. Tem-se, dessa forma, um tempo de cálculo mais elevado que para o caso anterior, uma vez que se calcula a norma $H_\infty$ em cada iteração.

Verificando *dist* e as raízes, nota-se uma igualdade entre os valores das tabelas 3 e 4, salvo os casos $\varepsilon = 10^{-1}$ e $\varepsilon = 10^{-2}$. Nesse caso, o programa foi suspenso pelo critério da norma e foram obtidos valores mais elevados de *dist* e do tempo de cálculo que da tabela 3, devido às piores aproximações das raízes.

Considerando somente a norma relativa como parâmetro de parada, não é possível testar para $\varepsilon \geq 10^{-4}$, uma vez que *Matlab* apresenta erros nas rotinas do *Symbolic Math Toolbox*, devido à

presença de inteiros de módulo muito alto.

Para o valor $\varepsilon = 10^{-3}$, os resultados são apresentados na tabela 2. Os resultados obtidos não são excepcionais, pois o procedimento é suspenso no grau $n = 5$, mas se tem uma simulação com alto custo em tempo e recursos informáticos. Dessa forma, percebe-se que esse critério, não utilizado no programa, não é um bom critério de parada.

Simulações foram efetuadas para se verificar a relação entre a localização das raízes e o grau *n*. Os resultados encontram-se na tabela 5. Os outros critérios, empregados nas outras tabelas, não foram considerados, porque esse não era o objetivo dessas simulações. Além disso, o tempo máximo de cálculo é 42.94s, insignificante em comparação com os outros valores presentes nas tabelas 3 e 4.

A qualidade das raízes aproximadas está em relação direta com o grau *n* empregado na aproximação, o que já se tinha concluído anteriormente. Verificou-se uma convergência exponencial das raízes aproximadas, dado de forma bastante lenta para $n \geq 3$. Têm-se boas aproximações das raízes a partir de $n = 3$.

| *n* | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| *dif* | 1.6880E-01 | 1.2700E-02 | 5.3145E-04 | 1.6635E-05 | 8.1297E-06 |
| *dist* | 1.3284E-02 | 5.6719E-04 | 5.0096E-05 | 3.3249E-05 | 2.4729E-05 |
| *n* | 7 | 8 | 9 | 10 | 11 |
| *dif* | 3.5428E-06 | 1.7127E-06 | 9.0667E-07 | 5.1579E-07 | 3.1083E-07 |
| *dist* | 2.1103E-05 | 1.9368E-05 | 1.8454E-05 | 1.7935E-05 | 1.7623E-05 |
| *n* | 12 | 13 | 14 | 15 | 16 |
| *dif* | 1.9631E-07 | 1.2892E-07 | 8.7484E-08 | 6.1058E-08 | 4.3661E-08 |
| *dist* | 1.7426E-05 | 1.7296E-05 | 1.7209E-05 | 1.7141E-05 | 1.7104E-05 |

Tabela 5: Relação entre *n* e a localização das raízes.

Essa característica da convergência foi percebida desde as tabelas 3 e 4, pela análise de *dif*, *dif2* e *dist*. Percebeu-se uma queda significativa para os valores iniciais de *n,* seguido por uma convergência lenta para valores superiores. Existem bons compromissos entre a qualidade das raízes aproximadas, tempo de cálculo e custo em recursos informáticos para os valores $\varepsilon = 0.001$ e $\varepsilon = 0.0001$.

Finalmente, conclui-se que as aproximações e o procedimento, descritos no capítulo 7, fornecem as melhores aproximações de pólos, para o exemplo em questão, que aqueles da literatura. Esse é um exemplo genérico de um sistema com atrasos, com um número finito de raízes instáveis. Para esses sistemas, a qualidade das aproximações dos pólos é garantida pela utilização da aproximação Padé-2, ótima no sentido $H_\infty$.

## 7.2 Comparação com os *softwares* livres existentes

Três *softwares* de ajuda à localização de raízes de uma função transcendente foram estudados:

I. *DDE-Biftool*;
II. *Proj2D*;
III. *IntLab*.

*DDE-Biftool*, que resolve por bifurcação, é um *software* complexo à utilização. Foi abandonado na análise realizada, pois trabalha no domínio temporal, enquanto que esse trabalho utiliza o domínio das freqüências.

*Proj2D* e *IntLab* trabalham com o método de cálculo por intervalo. *IntLab* é um *Matlab toolbox* que suporta intervalos reais e complexos aplicável em vetores e matrizes. Na sua descrição, diz ser concebido para apresentar execução e implementação muito rápidas, com verificação do resultado.

Entretanto, para ser um *toolbox*, se configura como um ambiente de programação, no qual se deveriam escrever os algoritmos. Dever-se-ia, então, reescrever o programa, utilizando as funções desse *toolbox*, o que não era ideal e foi, então, abandonado.

*Proj2D* é um calculador desenvolvido para caracterizar a projeção, em duas dimensões, de um conjunto definido por um sistema de restrições. Esse *software*, desenvolvido por Massa Dao, Xavier Baguenard e Luc Jauli*n,* utiliza o cálculo por intervalos e a propagação de restrições, permitindo, segundo os autores, obterem-se resultados garantidos, com os tempos de cálculo muito sensíveis ao número de parâmetros. *Proj2D* já foi utilizado para a análise da estabilidade de sistemas com atrasos. Esse foi o *software* escolhido para efetuar as comparações com o programa desenvolvido.

Múltiplos testes foram realizados com *Proj2D* e *cp_2*, tendo como base o exemplo numérico de [9], para o qual se conhecia a localização exata dos pólos instáveis. As equações e os resultados encontram-se no Anexo C. A rotina para o pré-tratamento das equações, para a utilização dos

*softwares,* encontra-se no Anexo D.

Para o exemplo de [9], verifica-se que os dois métodos apresentam uma performance equivalente. Justificado pela análise dos resultados presentes no Anexo C, verifica-se uma superioridade do programa desenvolvido neste trabalho. Mesmo sofrendo uma pequena perda de precisão, conforme se aumenta o grau do polinômio analisado, este método sempre permitiu a localização de todas as raízes, o que não acontece com *Proj2D*.

Além disso, não analisando a norma $H_\infty$, o que é perfeitamente aceitável, uma vez que se trabalha com um número finito de raízes instáveis, *cp_2* apresenta um tempo de cálculo muito menor que aquele de *Proj2D*. Outra consideração é a dificuldade de se analisar os domínios fornecidos por *Proj2D* e a característica de se fornecer grandes domínios **Ambíguos**, onde se sabe, segundo os autores, que existe ao menos uma raiz, mas não se identifica o seu número.

Conclui-se que *Proj2D* não apresentou um bom desempenho nos seguintes casos:

I. Raízes longínquas da origem: a performance é perturbada, pois não se tem a localização de todas as raízes, chegando-se ao extremo de não se encontrar nenhuma raiz;
II. Raízes próximas ou raízes múltiplas: verifica-se uma explosão do tempo de cálculo e domínios resultantes muito grandes, contendo várias raízes, sendo, portanto, imprecisos e com um desempenho ruim;
III. Aumento do grau do polinômio ou dos atrasos: existe a possibilidade de perda de raízes.

Chega-se a essas conclusões pela comparação em relação ao custo de tempo, tamanho e natureza dos intervalos, precisão e número de raízes fornecidas. Descobriu-se que a implantação da aritmética por intervalos de *Proj2D* não utiliza arredondamentos dirigidos aos cálculos em *floating point*, ou seja, o arredondamento para baixo, para se calcular o limite inferior do intervalo, e o arredondamento para cima, para se calcular o limite superior.

Isso pode apresentar problemas de confiabilidade e tempo de cálculo, uma vez que se trabalha com números muito baixos ou muito altos (como $e^{-100}$ e $e^{100}$). Deveria se incluir a multi-precisão, com o cálculo pelos intervalos compreendendo o *outward rounding*, ou seja, o arredondamento exterior, não incluso em *Proj2D*.

Verificou-se, entretanto, uma superioridade de *Proj2D* quando se efetua cálculos com funções com infinitas raízes instáveis, pois a versão atual de *cp_2* não é capaz de efetuar esses cálculos e *Proj2D* fornece aproximações aceitáveis para altos valores absolutos de módulos.

## 7.3 Limitações

A principal limitação do programa é a capacidade de determinar bem as raízes instáveis, desde que essas se apresentem em um número finito. Essa limitação é devida à aproximação Padé-2 utilizada para aproximar o sistema que, mesmo sendo ótima ao senso $H_\infty$ para o semi-plano direito, é uma aproximação de dimensão finita e, então, incapaz de aproximar as infinitas raízes instáveis.

Em seguida, verifica-se que podem existir casos de infinitas raízes instáveis não previstos na literatura disponível. O critério de parada, nesses casos, não é otimizado e deveria ser reformulado. Dessa forma, podem-se ter raízes instáveis que não sejam bem aproximadas, ainda que se encontrem, apesar de tudo, boas aproximações de raízes de baixo módulo. Esse ponto não faz parte do âmbito do programa, o qual é de fornecer aproximações de boa qualidade, o que não é possível para o caso de infinitas raízes instáveis, utilizando uma aproximação de dimensão finita. É, portanto, um suplemento, tendo em vista a incapacidade de se identificar no começo da execução, devido à ausência de literatura disponível.

A incorporação do cálculo da norma aumenta o tempo necessário. Deve-se admitir, *a priori*, que isto é uma ferramenta auxiliar na identificação da qualidade das aproximações para o caso de um número finito de pólos instáveis, pois a aproximação é já ótima ao senso $H_\infty$. A utilização de *Matlab,* para o cálculo por *loops,* não é a linguagem mais rápida a disposição. Poder-se-ia reescrever em linguagem C. Entretanto, isso necessitaria do triplo do tempo para se escrever o programa em *Matlab* por um programador *expert*, sendo impossível de efetuá-lo neste trabalho, em função da disponibilidade de tempo.

No começo, idealizou-se incluir o cálculo dos fatores de Bézout e de um controlador capaz de estabilizar o sistema. Entretanto, não houve tempo para programá-lo.

Trabalha-se no programa com polinômios e grau proporcional a $2n$. Tem-se, assim, que, conforme se aumenta o grau $n$ visando fazer convergir a aproximação, trabalha-se com polinômios de ordem crescente, que atingem altos valores. *Matlab* perde sua precisão, conforme se aumentam os graus do polinômio. É, portanto, uma limitação do *software* utilizado no cálculo.

Durante a construção das aproximações, não houve simplificação entre o numerador e o denominador da aproximação. Assim, perde-se precisão de resultados de *Matlab,* trabalhando-se com polinômios de alto grau, sem ser necessário, uma vez que se poderia conseguir fazer as

simplificações. Isso não é possível utilizando as variáveis do tipo ‘tf’, mas somente as variáveis do tipo ‘sym’. Dever-se-ia estudar os resultados de rotinas análogas, como *zero* para ‘tf’ e *solve* para ‘sym’. Assegurando a mesma eficácia, o programa reteria as melhores aproximações de raízes, passando todo o código em ‘tf’ para ‘sym’ e efetuando as simplificações de polinômios.

## 8 Avaliação dos resultados obtidos

Obteve-se, em uma primeira etapa, a estruturação do trabalho, com o extensivo estudo da literatura e esquematização da rotina a ser implementada. Da mesma forma, definiram-se as características do programa a ser escrito, assim como se efetuou a redação das rotinas que compõem o programa. Pequenos testes foram efetuados com equações bastante simples e verificou-se que o programa respondia de forma adequada.

Em uma segunda etapa, com o programa concebido, realizou-se a validação e análise do mesmo, via realização de testes. Algumas correções pontuais foram realizadas sobre os erros apresentados, através da comparação dos resultados oriundos do programa com resultados advindos da literatura, assim como *bugs* próprios ao programa, que foram corrigidos. Identificou-se, também, que alguns *softwares* livres podem ser utilizados, com certa adaptação, para se obter resultados necessários ao estudo da estabilidade desses sistemas. Esses resultados foram usados para comparação com os resultados fornecidos pelo programa realizado neste projeto.

Um estudo sobre as limitações do programa a ser realizado foi também realizado.

# 9 Avaliação de atividades futuras

Este trabalho conta com uma algumas simplificações para que a determinação de raízes seja possível. Essa era a definição inicial do escopo do projeto. Entretanto, três possíveis desdobramentos do projeto foram apresentados conforme se avançou no plano inicial:

I. Ampliar o acesso a toda a cadeia de pólos para sistemas com dois atrasos, uma vez que o programa se limita aos pólos de baixo módulo;
II. Ampliar o acesso a toda a cadeia de pólos para sistemas com *n* atrasos;
III. Realizar um *toolbox* em *Scilab*.

Entretanto, para a realização do projeto da matéria PMR2550, considera-se como encerradas as atividades com o projeto já realizado e testado. Essas atividades listadas acima se constituem como atividades futuras, a serem realizadas fora do escopo desta disciplina.

## 10 Cronograma

O cronograma de atividades inicial foi determinado como na tabela 6, levando em conta as etapas que constituem o projeto. O cronograma foi respeitado, mas a etapa "Levantamento de Dados" foi amplificada, graças à necessidade para a definição do escopo do projeto, dada à identificação dos três desdobramentos possíveis do projeto. O novo cronograma de atividades se encontra na tabela 7.

| Semana | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Atividade / Data | 25/ 09 | 02/ 10 | 09/ 10 | 16/ 10 | 23/ 10 | 30/ 10 | 06/ 11 | 13/ 11 | 20/ 11 | 27/ 11 | 04/ 12 |
| 1) Levantamento de dados: discussão de informações sobre o projeto. | XX | XX | XX | XX | XX | XX | | | | | |
| 2) Revisão bibliográfica. | | XX | XX | XX | XX | XX | X | | | | |
| 3) Implementação. | | | XX | XX | XX | XX | XX | XX | | | |
| 4) Testes e correções. | | | | | | XX | XX | XX | X | | |
| 7) Confecção do relatório parcial, minuta de artigo, monografia final, artigo e material de apresentação. | | | | XX | | | XX | XX | XX | XX | XX |

**Tabela 6: Cronograma inicial de atividades.**

| Semana | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Atividade / Data | 25/ 09 | 02/ 10 | 09/ 10 | 16/ 10 | 23/ 10 | 30/ 10 | 06/ 11 | 13/ 11 | 20/ 11 | 27/ 11 | 04/ 12 |
| 1) Levantamento de dados: discussão de informações sobre o projeto. | XX | XX | XX | XX | XX | XX | XX | XX | | | |
| 2) Revisão bibliográfica. | | XX | XX | XX | XX | XX | X | | | | |
| 3) Implementação. | | | XX | XX | XX | XX | XX | XX | | | |
| 4) Testes e correções. | | | | | | XX | XX | XX | X | | |
| 7) Confecção do relatório parcial, minuta de artigo, monografia final, artigo e material de apresentação. | | | | XX | | | XX | XX | XX | XX | XX |

**Tabela 7: Novo cronograma de atividades.**

# 11 Conclusão

Em comparação com os resultados dos *softwares* e da literatura disponível, o programa *cp_2* é o responsável pelas melhores aproximações de raízes instáveis, uma vez que elas estejam presentes em número finito. Além disso, é o programa que apresenta o melhor desempenho em termos de tempo de cálculo, facilidade de interpretação de resultados e identificação do número exato de raízes.

Isso foi possível, uma vez que escolhida uma rotina de resolução, tendo como base numerosos resultados da literatura. Efetuou-se pela combinação das aproximações por fatores coprimos e Padé-2, os quais fornecem uma aproximação de dimensão finita ótima ao senso $H_\infty$ para os sistemas analisados, com um sistema correto de avaliação de raízes, que analisa a convergência da localização das mesmas, assim que da norma $H_\infty$ relativa.

O programa obedece as funcionalidades essenciais definidas na seção 7.1. É um programa *user friendly*, flexível e com resultados válidos, o que foi verificado por numerosos testes, representados por um conjunto significativo, contido no Anexo C. Um ponto extra para a flexibilidade é que, mesmo que o programa tenha sido desenvolvido visando, principalmente, a utilização para a determinação da estabilidade dos sistemas com atraso, fica, também, uma ferramenta genérica de análise matemática, capaz de resolver funções transcendentes de complexidades distintas no semi-plano direito.

Dessa forma, desenvolveu-se um programa que se constitui como uma importante ferramenta à análise e à identificação de sistemas com atrasos, sendo de grande importância aos estudos de problemas de controle robusto. O programa será colocado em linha no mais curto intervalo de tempo possível para servir à comunidade de sistemas com atrasos e a todos aqueles que desejarem resolver funções transcendentes.

## Referências bibliográficas


[1] M.C. Delfour A. Bensoussa*n,* G. Da Prato and S.K. Mitter. *Representataion and control of infinite dimensional systems*. Birkhaüser, Bosto*n,* 1992, Tome 1.

[2] R. Bellman and K.L. Cooke. *Differential-Difference Equations*. Academic Press, New York, 1963.

[3] C. Bonnet and J.R. Partington. *Bézout factors and I1-optimal controllers for delay systems using a two-parameter compensator scheme*. IEEE Trans. Automat. Control, (44):1512-1521, 1999.

[4] C. Bonnet and J.R. Partington. *Stabilization of some fractional delay systems of neutral type*, Automatica, 2007.

[5] C. Bonnet and J.R. Partington. *H-∞ and bibo stabilization of delay systems of neutral type*, 2004. 283-288.

[6] C. Bonnet and J.R. Partington. *Stablization of a class of delay systems using pi methods.* Rapport de recherche, INRIA Rocquencourt, 2005.

[7] R.F. Curtain and H. Zwart. *An introduction to infinite-dimensional linear systems theory*, volume 21 of Texts in Applied Mathematics. Springer-Verlag, New York, 1995.

[8] E.B. Lee G. Gu, P.P. Khargonekar and P. Misra. *Finite-dimensional approximations of unstable infinite dimensional systems*. Proc. 1990 C.D.C.

[9] E.B. Lee G. Gu, P.P. Khargonekar and P. Misra. *Approximation of infinite dimensional systems*. IEEE Trans. Automat. Control, (34) :610–618, 1989.

[10] K. Walton A. Korytowski J. E. Marshall, H. Górecki. *Time-Delay Systems: Stability and Performace Criteria with Applications*. Ellis Horwood, Londo*n,* 1992.

[11] Marc Lenoir. *Fonctions d'une variable complexe*, 2006/2007.

[12] P. M. Mäkilä and J. R. Partington. *Shift operator induced approximations of delay systems*. SIAM J. Control Optim, 37(6) :1897–1912.

[13] P.M. Ndiaye. *Perturbation des systèmes linéaires par des retards : Sensibilité des Performances*. PhD thesis, Université Paris-IX Dauphine, 1997.

[14] S.-I. Niculescu. *Delay Effect on Stability: A Robust Control Approach*. Springer-Verlag, Berli*n,* 2001.

[15] J. R. Partington. *Approximation of unstable infinite-dimensional systems using coprime factors*. Systems and Control Letters, (16) :86–96, 1991.

[16] M. Vidyasagar. *Control System Synthesis.* MIT Press, 1985.

[17] M. Vidyasagar. *A brief history of the graph topology*. European Journal of Control, 2 :80–87, 1996.

[18] C.A. Desoer, M. Vidyasagar. *Feedback Systems: Input-Output Properties*, Academic Press, USA, 1975.

# Anexos

## Anexo A: Readme

PC_2 : Approximation des zéros instables de fonctions transcendantes pour l'étude de la stabilité de systèmes à retards.

---

This is the README file which you get when you unwrap our distribution file. This program enables the roots computation of transcendent functions, for the study of delay systems.

Handling the Distribution File
The distribution file contains the source code for the program. All distribution files are available as compressed tar files, gzipped tar files, or as zip files. For the compressed tar files, first uncompress and untar the file <dist>.tar.Z where <dist> is the name of the specific distribution:

uncompress <dist>.tar.Z
gunzip <dist>.tar.gz

Then untar the file by typing
tar xf <dist>.tar

For the zip files, type the following:
unzip <dist>.zip

Generated Directory structure
The distribution files are all designed to be unwrapped in the directory "pc_2". A distribution file also contains a few informative files like this README file and the file *'results'*. The structure of the entire unpacked distribution file is as follows (directories):

pc_2 Top-node
*cp_2*/*cp_2*.m Main routine
*cp_2*/*calcul.m* Auxiliary routine
*cp_2*/approx_pade.m Auxiliary routine
*cp_2*/tf2sym.m Auxiliary routine
*cp_2*/calcul_beta_gamma.m Auxiliary routine
*cp_2*/csort.m Auxiliary routine
*cp_2*/norma.m Auxiliary routine
*cp_2*/pade2_meu.m Auxiliary routine
*cp_2*/plot_poles.m Auxiliary routine
*cp_2*/print_n_d.m Auxiliary routine
*cp_2*/*results* Auxiliary file
*cp_2*/readme.txt Documentation

All the source code is written in *Matlab* code to assure portability.

Run

Open *Matlab* and place in the "*cp_2*" directory. Write the routine call '*cp_2*' in the Command Window. It starts to show the Interface figures:

1. Enter the value of epsilon
2. Enter the delays of the denominator
3. Polynoms Pi(s) = _____ * exp(-iT)
4. Enter the delays of the numerator
5. Polynoms Pi(s) = _____ * exp(-iT)

Write in a numeric type, with '.' instead of ','. Don't leave any other type, such as characters. You should write the arithmetic operators (+, -, *, /, ^).
General values of epsilon are 0.001 and 0.0001. (Interface figure n°1)
You should write all the polynoms (Interface figure n°3) for the delays written (Interface figure n°2).

The program shows error messages on account of instability. The program displays too its states and features in the Command Window, such as the degree *n* of the approximation Padé-2 and the polynoms D and Dk (as well as *n* and Nk).
The program opens another figure with the graph of the poles.
The *results* and simulation features (epsilo*n*, denominator delays, denominator polynoms, numerator delays, numerator polynoms, H-infinity relative norme and poles) are stocked in the file '*results*'.

Got Problems or Comments?
If you encounter problems, would like to feed back suggestions good ideas etc. then please send a mail explaining your problem to mlltorquato@gmail.com

## Anexo B: Gráfico de pólos e interfaces do programa para o exemplo numérico de [15]

Apresenta-se, aqui, a execução do programa para o exemplo da seção 8.2, de modo a introduzir as interfaces encontradas durante a execução para um exemplo qualquer. Dessa forma, apresenta-se o caráter *user friendly* do programa.

A execução do programa começa pela abertura de *Matlab* e pela indicação em *Current Directory* da pasta onde existem as rotinas que compõem o programa *cp_2*. Digita-se *cp_2* para executar o programa.

A próxima janela apresentada pede pelo valor de $\varepsilon$, que será utilizado em todos os *loops* do programa, como se vê na figura B.1. O valor habitual é $\varepsilon = 0.001$.

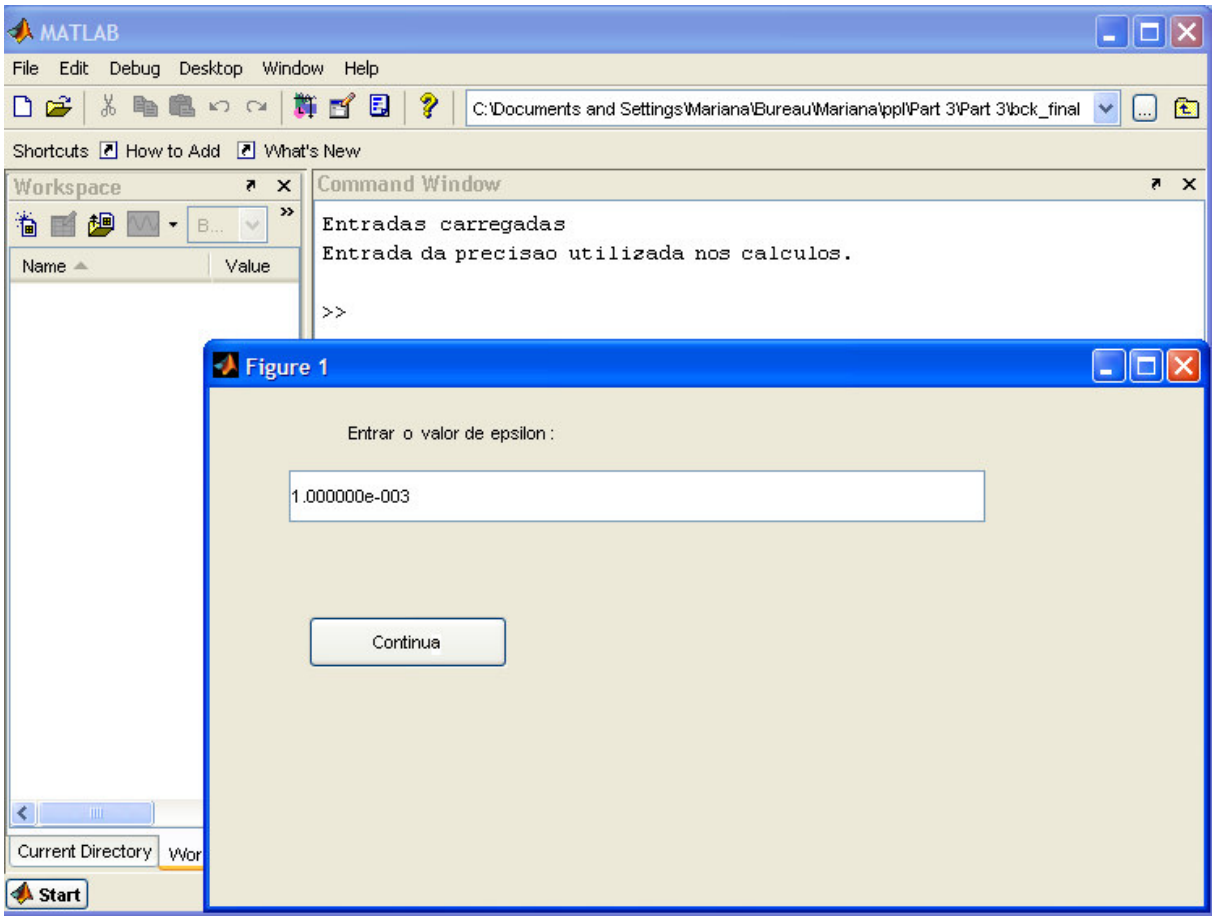


Figura B.1: Inserção do valor de $\varepsilon$.

Em seguida, começa a inicialização do denominador. Pode-se acompanhar o estado do sistema pela análise do *Command Window*. A próxima janela pede os atrasos do denominador

(figura B.2).

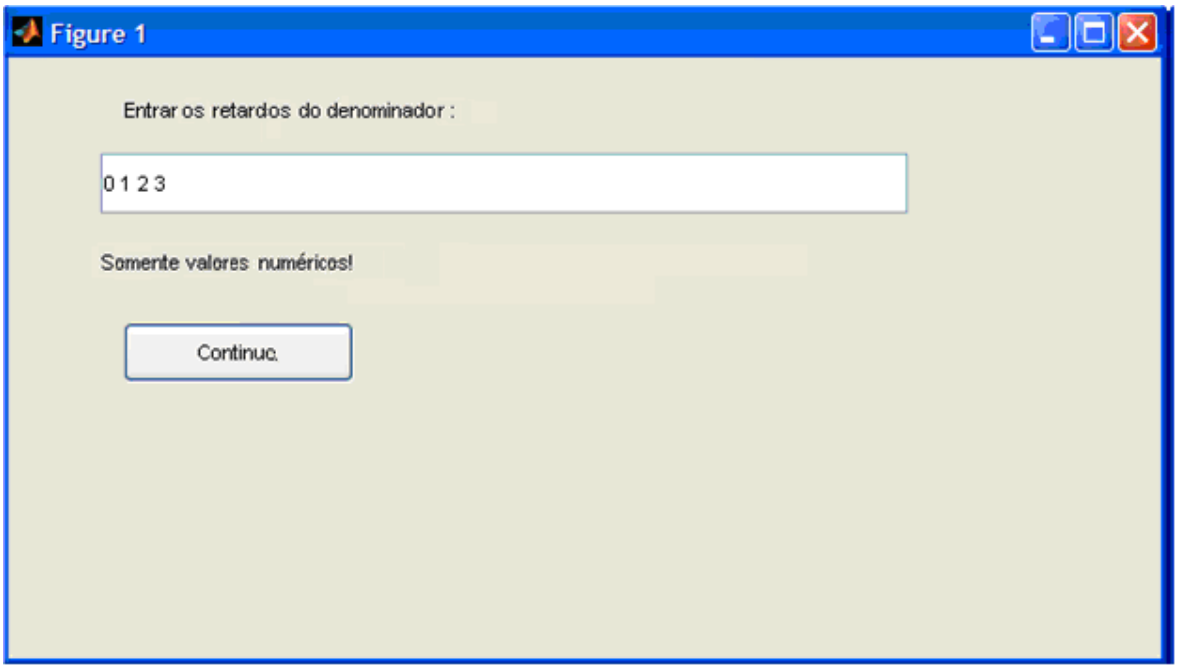


Figura B.2: Inserção dos atrasos $\gamma_i$ do denominador.

Para cada atraso $\gamma_i$ incluído, deve-se introduzir um polinômio $p_i(s)$ na janela mostrada pela figura B.3. Para o sistema incluído, no exemplo numérico de [15], o numerador apresenta infinitas raízes instáveis. Desse modo, o programa indica, por uma mensagem de erro, o estado do sistema e explica que o programa não continua o cálculo das raízes do numerador. Isso é somente um dos casos possíveis, onde o programa mostra mensagens de erros ou a explicação ao usuário. As outras mensagens possíveis já foram explicadas na descrição do programa.

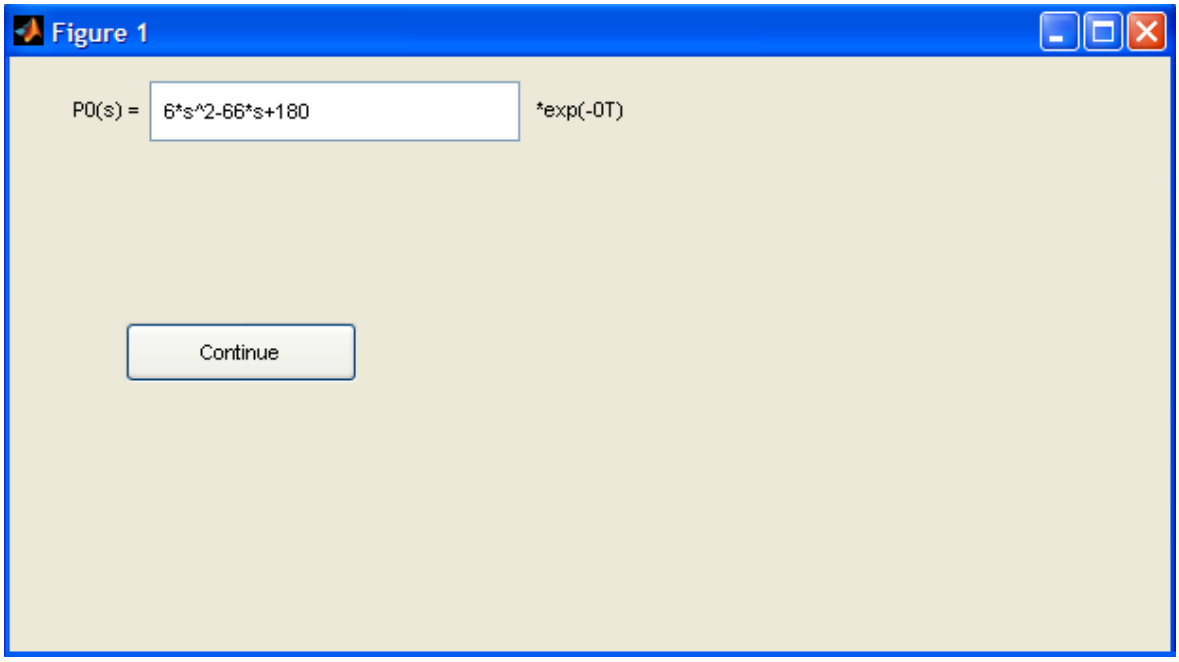


Figura B.3: Inserção dos polinômios $p_i(s)$ do denominador.

O programa continua com o cálculo das raízes do denominador. Como as aproximações de $n$ e $N_k$ não foram efetuadas, têm-se somente apresentados D e $D_k$. A apresentação dessas saídas, entre outras, é verificada na figura B.4.

Na finalização do programa, quando se tem raízes instáveis, mostram-se duas janelas: uma com o gráfico dos pólos instáveis (figura B.5) e outra com os valores numéricos calculados dos pólos, divididos em partes reais e imaginárias (figura B.6).

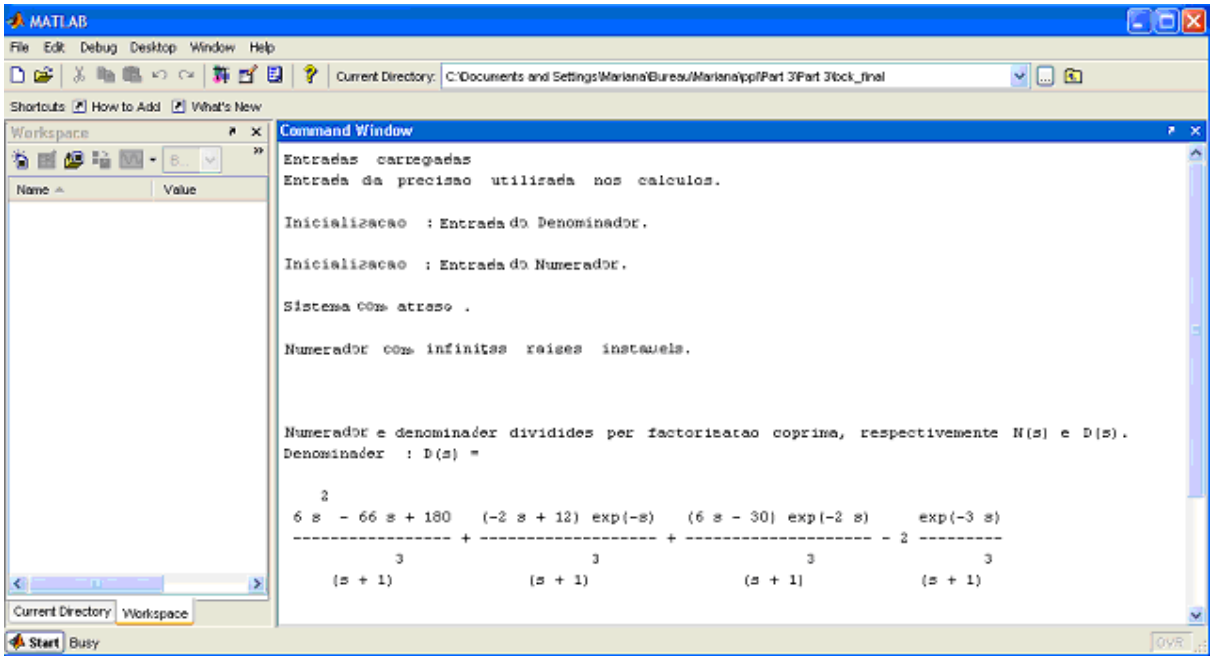


Figura B.4: Estado do sistema e aproximações.

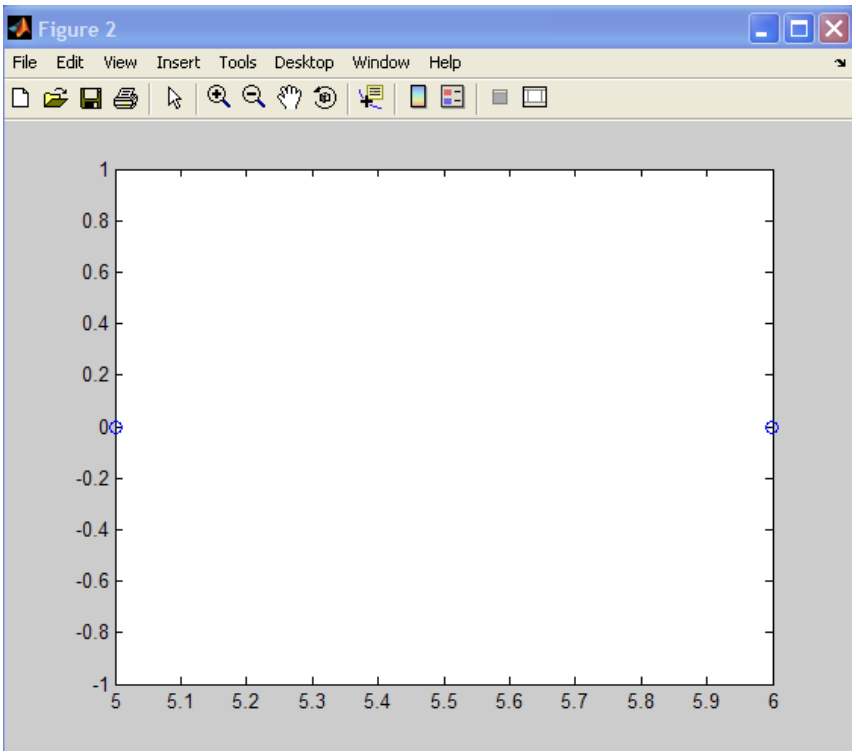


Figura B.5: Gráfico dos pólos instáveis.

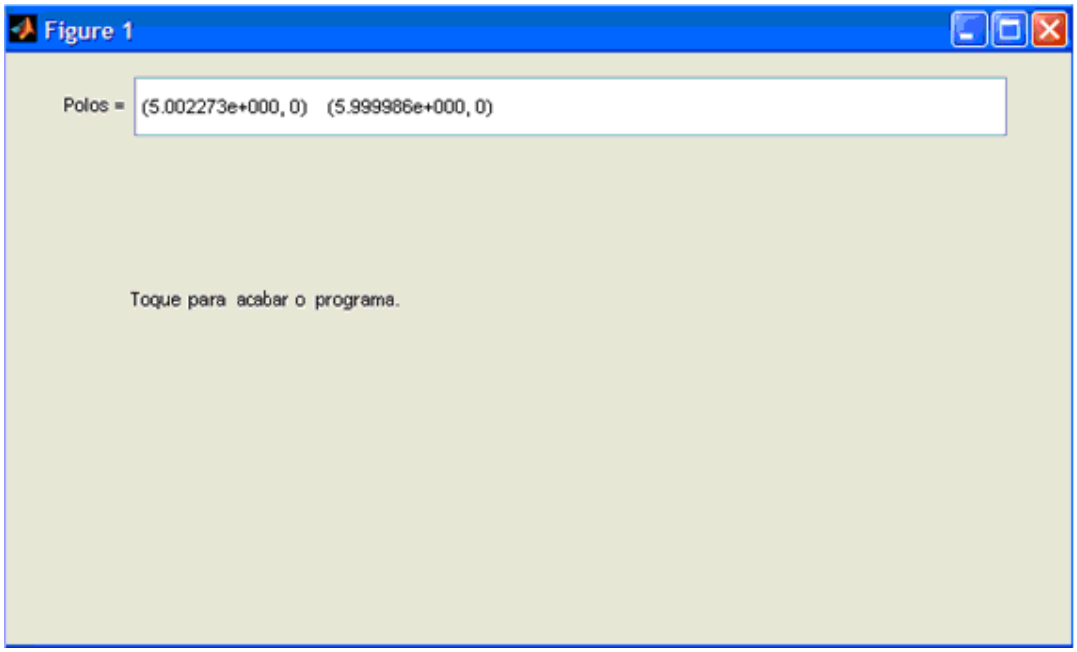


Figura B.6: Valores numéricos dos pólos.

## Anexo C: Comparação com *Proj2D* – Equações e resultados

Testaram-se o programa e o *software Proj2D* sobre um conjunto de funções representativas, de modo a se avaliar as performances dos dois programas para casos diferentes: polinômios de graus altos, com grandes atrasos, com raízes múltiplas ou próximas, com raízes afastadas ou com raízes complexas.

Utilizou-se, como base, o exemplo numérico de [9], chamado a seguir por $f_{Gu}$, já plenamente estudado na seção 8.1. Mudanças foram feitas, multiplicando por diversos polinômios. Depois, efetuou-se uma mudança na função $f_{Gu}$, obtendo a função $f_{Gu}^{mod}$, a qual será utilizada como base para o estudo de outras funções compostas.

Sejam

$$f_{Gu} = 6s^2 + \left(6e^{-2s} - 2e^{-s} - 66\right)s - \left(2e^{-3s} + 30e^{-2s} - 12e^{-s} - 180\right) \quad \text{e} \tag{C.1}$$

$$f_{Gu}^{\text{mod}} = 6s^2 + \left(6e^{-2s} - 2e^{-s} - 66\right)s - \left(30e^{-2s} - 12e^{-s} - 180\right) \tag{C.2}$$

as funções testadas são:

$$f_1 = \left[(s-1)*(s-5)*f_{Gu}\right]^{-1} \tag{C.3}$$

$$f_2 = \left[(s-1)*f_{Gu}\right]^{-1} \tag{C.4}$$

$$f_3 = \left[(s-1)*(s-15)*f_{Gu}\right]^{-1} \tag{C.5}$$

$$f_4 = \left[(s-1)*(s-3)*f_{Gu}\right]^{-1} \tag{C.6}$$

$$f_5 = \left[(s-150)*(s-20000)*f_{Gu}\right]^{-1} \tag{C.7}$$

$$f_6 = \left[f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.8}$$

$$f_7 = \left[(s-1)*f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.9}$$

$$f_8 = \left[(s-1)*(s-15)*f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.10}$$

$$f_9 = \left[(s-1)*(s-5)*f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.11}$$

$$f_{10} = \left[(s-5)*f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.12}$$

$$f_{11} = \left[(s-3)*(s-20)*(s-150)*f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.13}$$

$$f_{12} = \left[(s-2000)*f_{Gu}^{\text{mod}}\right]^{-1} \tag{C.14}$$

$$f_{13} = \left[6s^2 + (6e^{-2s} - 2e^{-s} - 66)s - 2e^{-3s} + (3s-7)e^{-4s} - 2e^{-5s} + (34s-12)e^{-6s} + 13e^{-7s}s\right]^{-1} \quad \text{(C.15)}$$

$$f_{12} = \left[\left(s^2 - 6s + 13\right) * f_{Gu}\right]^{-1} \quad \text{(C.16)}$$

As equações foram escolhidas, no modo descrito acima, porque se conhecem os pólos exatos de $f_{Gu}$, dados por [8]. Inclui-se a multiplicação por fatores que fornecem facilmente raízes bem estabelecidas, a fim de verificar o desempenho dos programas para encontrá-las.

Para *cp_2*, os testes foram realizados com $\varepsilon$ = 0.001 e não incluem a verificação da norma $H_\infty$. Em *Proj2D*, os testes são realizados com $\varepsilon$ = 0.01 e $\varepsilon$ = 0.001. Trabalha-se com $\varepsilon$ = 0.001 quando se trata de um caso crítico (intervalo sem identificação de raízes ou intervalo com várias raízes).

Dessa forma, uma vez que se sabe a localização das raízes e que se têm os resultados das simulações anteriores com um valor maior de $\varepsilon$, tenta-se utilizar os domínios inicias menores em torno do valor das raízes com um valor menor de $\varepsilon$.

A comparação é realizada tendo em conta as raízes encontradas e o tempo de cálculo. No caso de *Proj2D*, apresentam-se as raízes encontradas pelo intervalo que as contém e pela natureza do intervalo (entre domínios **Solução** e **Ambíguo**).

As rotinas em *Maple,* para se encontrar as equações a serem introduzidas nos dois programas, encontram-se no Anexo D, para a função $f_1$.

Os intervalos representados acima foram reagrupados de modo a facilitar a interpretação dos resultados e a sua demonstração. Geralmente, o resultado de uma simulação em *Proj2D* é de interpretação muito mais difícil. Os intervalos foram reagrupados em favor da segurança.

Verifica-se, assim, que *cp_2* fornece resultados de interpretação muito mais fácil. Entretanto, *Proj2D* fornece um domínio ambíguo, mas é claro que existe ao menos uma raiz nesse intervalo (de acordo com os autores). Entretanto, não se explica o número de raízes dentro do intervalo. Isso não é desejado para a identificação de um sistema a ser controlado.

Analisando os pólos encontrados por *cp_2* para as funções de $f_1$ a $f_5$, verifica-se que o erro máximo foi de 0.000049. Além disso, o programa é capaz de encontrar todos os pólos, mesmo em casos extremos, como os pólos muito próximos ou os pólos longínquos. Esse comportamento é verificado junto aos tempos de cálculo muito baixos em relação àqueles apresentados por *Proj2D*.

A característica mais importante, em relação ao tempo de cálculo, é que *cp_2* apresenta um comportamento relativamente uniforme em relação à presença de graus diferentes e a casos

diferentes de distribuição dos pólos: pólos próximos, pólos distantes da origem. Isso não é observado em *Proj2D*, onde a presença de pólos múltiplos ou próximos aumenta muito o tempo de cálculo. Esse comportamento é acompanhado da localização de um domínio de grandes dimensões, não sendo, então, eficaz, quando o objetivo é determinar os pólos do sistema.

Tomou-se a liberdade de modificar a função da literatura, $f_{Gu}$, devido à qualidade dos pólos encontrados pela função $f_{Gu}$ e à correspondência entre os pólos encontrados pelos dois programas. Desse modo, considera-se que o programa tem um comportamento e um desempenho independente dos fatores críticos considerados nos primeiros testes. Deseja-se, agora, testar a influência do número de atrasos e dos seus módulos.

Dessa forma, as simulações foram realizadas com as funções $f_6$ a $f_{13}$. Considera-se que os resultados de *cp_2* são boas aproximações dos pólos do sistema, enquanto se tem um número finito de raízes instáveis. Verifica-se que os dois programas apresentaram, para $f_6$, o mesmo número de pólos, com as localizações respectivas coerentes. Isso foi pego como uma prova do desempenho de dois programas para essa função. Verifica-se, também, um tempo de cálculo bastante baixo para *Proj2D,* enquanto não se tem raízes múltiplas e se têm atrasos mais baixos.

Multiplica-se, então, $f_6$ por fatores com as raízes facilmente determinadas, para continuar a avaliação das performances dos programas. Pela análise de $f_8$, verifica-se que a presença de raízes longínquas penaliza o desempenho de *Proj2D*, uma vez que se perde a precisão e o tamanho dos intervalos fornecidos, unidos a uma explosão do tempo de cálculo.

Pela análise de $f_{11}$ e $f_{12}$, verifica-se a presença de pólos longínquos, não reconhecidos por *Proj2D,* e que podem até retirar o reconhecimento de outras raízes mais próximas à origem ($f_{12}$). Já para a função $f_{13}$, criada aleatoriamente com o objetivo de analisar o desempenho dos programas com presença de grandes atrasos, não se verifica nenhuma correlação entre os resultados dos dois programas, ao mesmo tempo em que se verifica um ligeiro aumento no tempo de cálculo de *cp_2*.

Não se podem emitir conclusões sobre a qualidade das performances dos sistemas, mesmo que se possa verificar que os pólos de *cp_2* são mais próximos àqueles de $f_6$ e $f_{Gu}$. Verificando as simulações anteriores, nota-se que os resultados de *cp_2* são geralmente mais precisos e confiáveis.

| | Programa | Tempo | Pólos | Natureza |
|---|---|---|---|---|
| f1 | *cp_2* | 8.75 | 1.000000; 5.000000; 5.002273; 5.999986 | |
| | *Proj2D* | 473.68 | [0.999999, 1.000000]x[0, 1.532455E-11] | Ambíguo |
| f2 | *cp_2* | 8.51 | 1.000000; 5.002273; 5.999986 | |
| | *Proj2D* | 13.44 | [0.999999, 1.000000]x[0,0]<br>[5.00224095480, 5.00224095485]x[0,2.913511E-11]<br>[5.999755, 6.005859]x[0,0.005346] | Ambíguo |
| f3 | *cp_2* | 8.99 | 1.000000; 5.002273; 5.999986; 1.500000e+001 | |
| | *Proj2D* | 262.87 | [0.999999,1.000000]x[0,0]<br>[4.965209, 6.079102]x[0, 0.079346]<br>[14.996983, 15.008545]x[0, 0.003052] | Ambíguo |
| f4 | *cp_2* | 8.35 | 1.000000; 3.000000; 5.002273; 5.999986 | |
| | *Proj2D* | 871.38 | [0.999048, 1.000562]x[0, 0.000781]<br>[2.993310, 3.006934]x[0, 0.007032]<br>[4.987573, 5.017090]x[0, 0.015625]<br>[5.991894, 6.008545]x[0, 0.008594] | Ambíguo |
| f5 | *cp_2* | 9.09 | 5.002273; 5.999986; 1.500000e+002; 2.00000e+004 | |
| | *Proj2D* | 28.85 | [5.001362, 5.003261]x[0, 0] | Ambíguo |
| f6 | *cp_2* | 6.86 | 5.002283; 5.999965 | |
| | *Proj2D* | 0.312 | [5.00224105610,5.00224105612]x[0,3.73E-12]<br>[5.99999385062,5.99999385063]x[0,4.32E-12] | Ambíguo |
| f7 | *cp_2* | 6.52 | 1.000000; 5.002283; 5.999965 | |
| | *Proj2D* | 4.56 | [0.999999,1.000000]x[0,0]<br>[5.0022410560,5.0022410561]x[0,2.88E-11]<br>[5.999366,6.000354]x[0,0.000976] | Ambíguo |
| f8 | *cp_2* | 6.80 | 1.000000; 5.002283; 5.999965; 1.500000e+001 | |
| | *Proj2D* | 143.30 | [0.999999, 1.000000]x[0, 3.004E-12]<br>[4.90625, 5.125]x[0, 0.15625]<br>[5.8671875, 6.117187]x[0, 0.1875]<br>[14.992187, 15.007812]x[0, 0.007812] | Ambíguo |
| f9 | *cp_2* | 6.28 | 1.000000; 5.000000; 5.002283; 5.999965 | |
| | *Proj2D* | 1814.30 | [0.999260, 1.003461]x[0, 0]<br>[4.121093, 6.425781]x[0, 1.425781] | Ambíguo |
| f10 | *cp_2* | 7.02 | 5.000000; 5.002283; 5.999965 | |
| | *Proj2D* | 3150.44 | [4,983642, 5.018798]x[0, 0.031494]<br>[5.999511, 6.000244]x[0, 0.000732] | Ambíguo |
| f11 | *cp_2* | 6.89 | 3; 5.002283; 5.999965; 20; 150 | |
| | *Proj2D* | 465.94 | [2.982146, 3.017397]x[0, 0.019531]<br>[4.808095, 5.259295]x[0, 0.449219]<br>[5.724594, 6.253344]x[0, 0.46875] | Ambíguo |
| f12 | *cp_2* | 6.57 | 5.002283; 5.999965; 2.000000e+003 | |
| | *Proj2D* | 0.52 | ø x ø | - |
| f13 | *cp_2* | 22.48 | 5.002246; 5.999984 | |
| | *Proj2D* | 0.83 | [0.046431,0.053062]x[0.234587, 0.251133] | Ambíguo |
| f14 | *cp_2* | 8.46 | 3 + 2i ; 3 - 2i ; 5.002273 ; 5.999986 | |
| | *Proj2D* | 948.63 | [2.802734, 3.199219]x[1.748046, 2.197266]<br>[4.525390, 6.234375]x[0, 0.888672] | Ambíguo |

Tabela C.1: Comparação dos resultados de *cp_2* e *Proj2D*.

A simulação com $f_{14}$ foi feita para se analisar o desempenho dos programas para se encontrar pólos complexos. Os dois programas encontraram os pólos procurados, entretanto *Proj2D* apresentou um tempo maior de cálculo e grandes intervalos.

Apresenta-se, na figura C.1, a interface inicial, onde se definem as restrições, o valor de $\varepsilon$ e as variáveis, para $f_1$. Apresenta-se, também, o resultado da simulação com a determinação progressiva dos intervalos, para $f_{14}$, na figura C.2.

Numerosos testes foram feitos para sistemas com infinitas raízes. Para a função $f_{15}$, por exemplo, testou-se em *cp_2* e verificaram-se, graças à literatura, que o caso apresenta infinitas raízes instáveis. Desse modo, o programa explica, através de uma mensagem de erro, e é reinicializado.

$$f_{15} = -s^2 + s^2 * e^{-s} \tag{C.17}$$

Para *Proj2D*, entretanto, utilizou-se $\varepsilon = 0.01$ com um intervalo inicial de [0, 100] x [0, 100], chegando após 0.21s às raízes mostradas na tabela C.2, a qual contém, também, as raízes calculadas, de acordo com os métodos da literatura.

Essas raízes foram comparadas aos valores propostos na literatura e constituem boas aproximações para as raízes de alto valor absoluto. Verifica-se, então, uma superioridade de *Proj2D* no caso de infinitas raízes, pois é possível calcular, com uma precisão relativa, certo número do conjunto de infinitas raízes instáveis.

Isso foi possível antes por *cp_2*. Notou-se uma perda do caráter assintótico da aproximação Padé-2 durante a aproximação do sistema com infinitas raízes instáveis, como se vê pela figura C.3.

Mostra-se, na figura C.4, o gráfico, manipulado em *Excel* para a retirada de raízes que não obedecem ao caráter assintótico, as infinitas raízes calculadas da função $f_{15}$, antes da modificação do programa.

A localização das infinitas raízes instáveis foi estudada com o programa aqui produzido, mas isto foi excluído do programa, porque se quer respeitar a validade dos resultados fornecidos (veja seção 7.1).

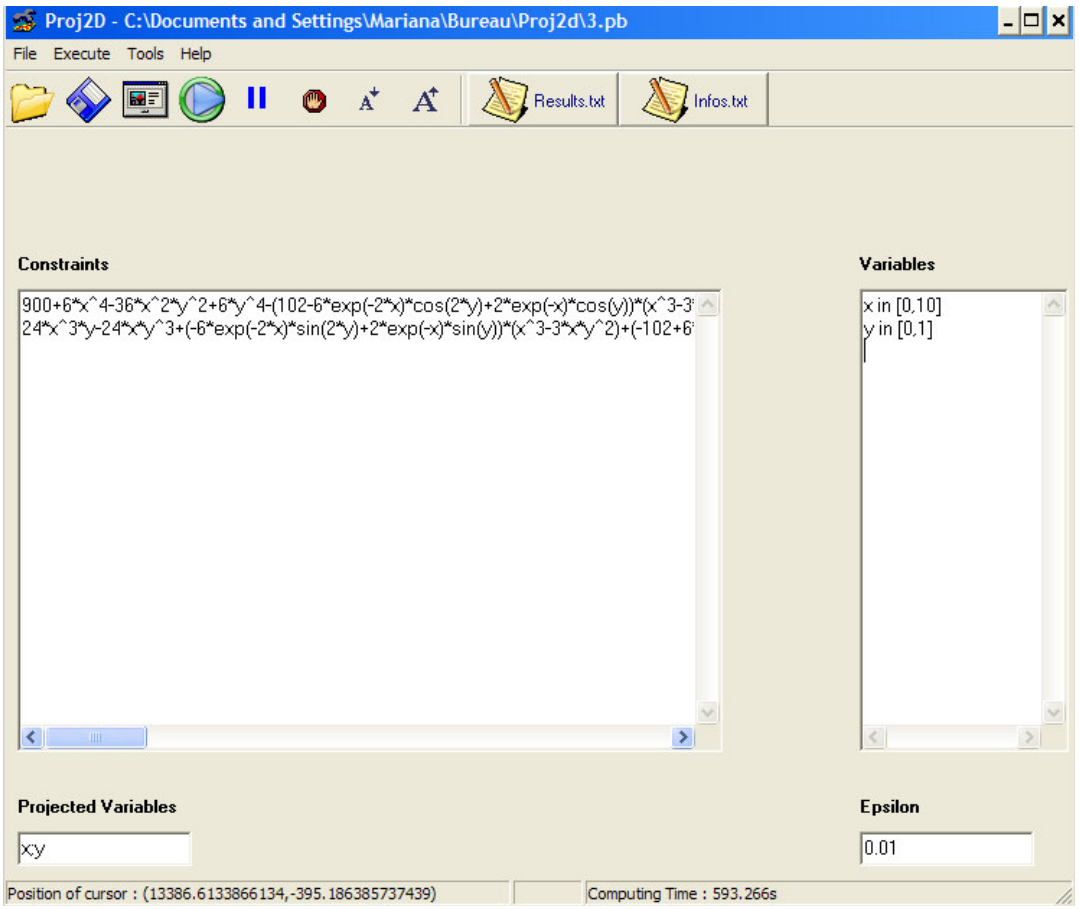


Figura C.1: Determinação dos parâmetros da simulação em *Proj2D* para $f_1$.

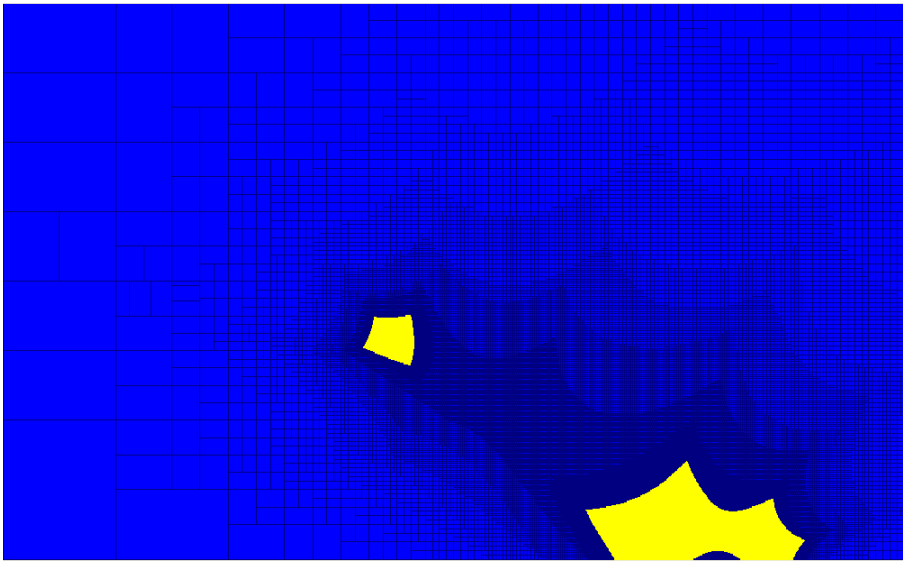

Figura C.2: Resultados da simulação de $f_{14}$ em *Proj2D*.

| *Proj2D* | Literatura |
|---|---|
| [0.000990,0.000991]x[50.245548,50.245570] | 0,000989 + 50,245588i |
| [0.000782,0.000783]x[56.530953,56.530974] | 0,000781 + 56,530983i |
| [0.000633,0.000634]x[62.815912,62.815933] | 0,000633 + 62,815937i |
| [0.000523,0.000524]x[69.100548,69.100569] | 0,000523 + 69,100569i |
| [0.000439,0.000440]x[75.384942,75.384962] | 0,000439 + 75,384960i |
| [0.000374,0.000375]x[81.669149,81.669169] | 0,000374 + 81,669166i |
| [0.000323,0.000323]x[87.953209,87.953230] | 0,000323 + 87,953226i |
| [0.000281,0.000281]x[94.237154,94.237174] | 0,000281 + 94,237169i |
| [0.003980,0.003981]x[25.092710,25.092731] | 0,003957 + 25,092952i |
| [0.002542,0.002542]x[31.383966,31.383987] | 0,002533 + 31,38409i |
| [0.001763,0.001763]x[37.672507,37.672527] | 0,001759 + 37,672586i |
| [0.001294,0.001295]x[43.959507,43.959527] | 0,001292 + 43,959561i |
| [0.007110,0.007111]x[18.795943,18.795964] | 0,007036 + 18,796504i |
| [0.016218,0.016220]x[12.484914,12.484935] | 0,015831 + 12,486793i |
| [0.070521,0.070527]x[6.1086384,6.1086641] | 0,063326 + 6,124030i |
| [0.775195,0.78125]x[1.395545,1.400976] | NaN – Infinito |

Tabela C.2: Raízes de $f_{15}$ por *Proj2D*.

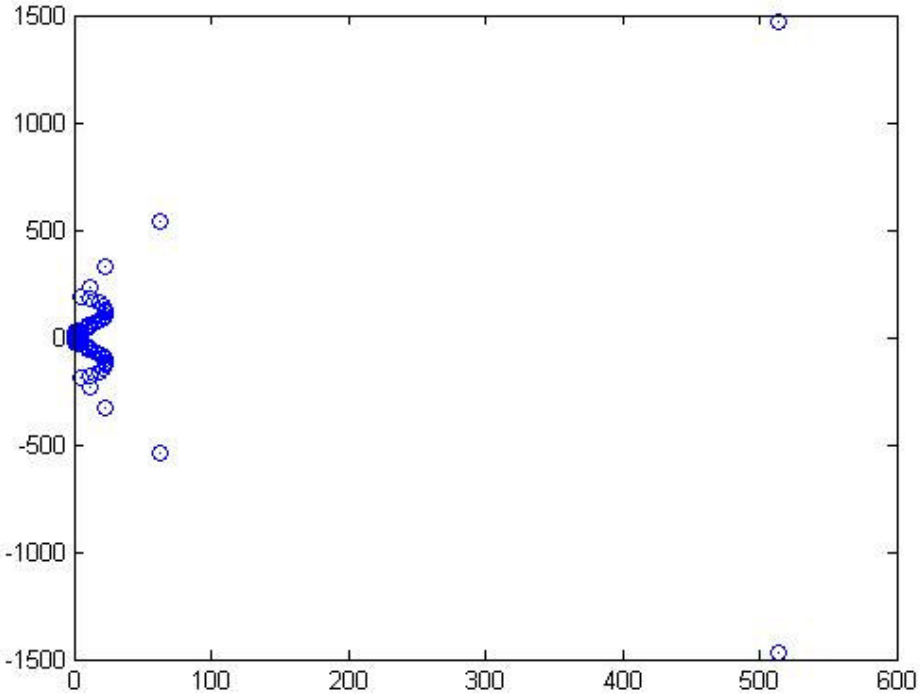


Figura C.3: Localização das infinitas raízes instáveis.

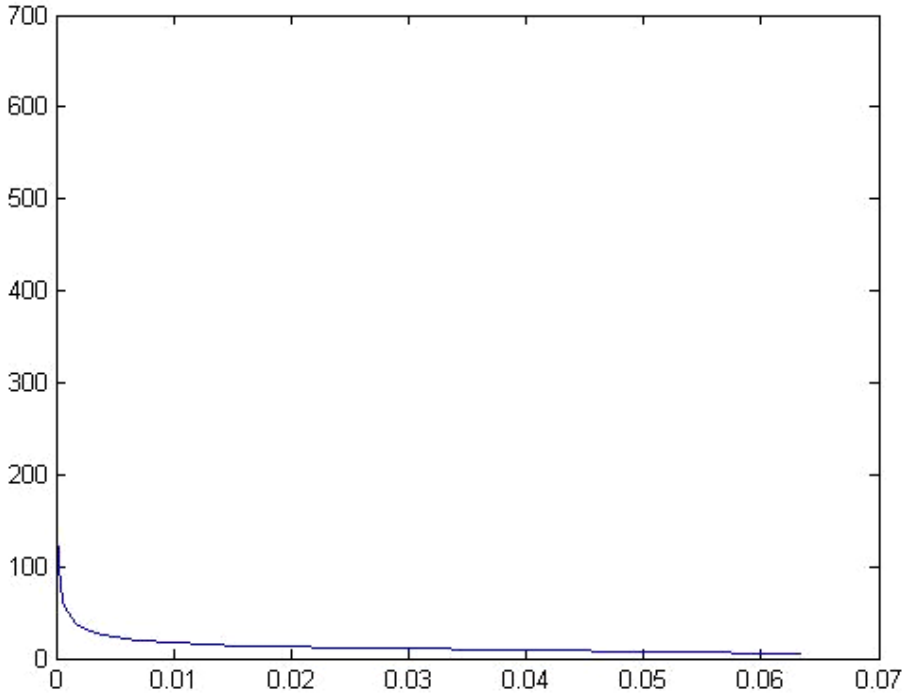


Figura C.4: Gráfico das infinitas raízes de $f_{15}$.

## Anexo D: Rotina em *Maple* para a comparação com *Proj2D*

As rotinas abaixo foram criadas para gerar de modo quase automático as entradas dos programas *cp_2* e *Proj2D*. As rotinas em *Maple* foram utilizadas, graças à facilidade de aplicação e a riqueza na gestão das variáveis simbólicas.

Apresentam-se, aqui, as rotinas para a função $f_1$ (C.3), uma vez que se aplica a mesma rotina para as outras funções. A fim de se produzir as entradas para o programa *Proj2D*, utiliza-se a rotina *1_fct.mw* dada abaixo.

```
G := (s-1)*(s-5)*(6*s^2+(6*exp(-2*s)-2*exp(-s)-66)*s-2*exp(-3*s))
G := collect(evala(Expand(G)), s)
G := evalc(subs(s = x+I*y, G))
Gre := convert(evalc(Re(G)), string)
Gim := convert(evalc(Im(G)), string)
```

Esses comandos geram as entradas, no campo "Restrições", do programa *Proj2D*, respectivamente *Gre* e *Gim,* igualadas a zero. A fim de se produzir as entradas para o programa *cp_2*, utiliza-se a rotina *1_fct_matlab.mw*, dada pelos comandos abaixo.

```
G := (s-1)*(s-5)*(6*s^2+(6*exp(-2*s)-2*exp(-s)-66)*s-2*exp(-3*s))
G := Expand(G)
G := evala(G)
G := collect(G, exp(-s))
G := collect(G, exp(-2*s))
G := collect(G, exp(-3*s))
G := convert(G, string)
```

Chega-se, então, à função G, onde se encontra a divisão dos polinômios para cada atraso. Esses polinômios são as entradas do programa *cp_2*.

## Anexo E: Resolução formal de uma equação de terceiro grau para a análise da estabilização de um sistema com atraso

A questão é saber se um controlador do tipo $C(s) = \dfrac{s^2 + bs + c}{\alpha s^2 + \beta s + \gamma}$ pode estabilizar (ao senso $H_\infty$) um sistema com atrasos do tipo $P(s) = \dfrac{e^{-sT}}{s - \sigma}$, para todo $T \geq 0$.

Busca-se, dessa forma, que $(I + PC)^{-1}$, $P(I + PC)^{-1}$ e $C(I + PC)^{-1}$ sejam em $H_\infty$. O denominador da malha fechada é do tipo $F(s,T) = A(s) + C(s)\exp(-sT)$, onde $\deg(A) > \deg(C)$. As técnicas de Walton e Marshall (seção 6.1.2) foram utilizadas para se determinar os valores do atraso que desestabilizam o sistema e que, então, eventualmente, o re-estabilize.

Começou-se a estudar o controlador $C(s) = \dfrac{s^2 + bs + c}{\alpha s^2 + \beta s + \gamma}$ capaz de estabilizar $P(s) = \dfrac{e^{-sT}}{s - \sigma}$.

Analisando $F(s,0)$, chega-se a um polinômio parametrizado de grau 3 (com 5 parâmetros). Deve-se verificar a ausência de raízes instáveis para $T = 0$, o que foi feito pelo critério de estabilidade de Routh-Hurwitz. O método de Walton e Marshall pede que se verifique se existem valores de parâmetros tais que um polinômio parametrizado de grau 3 não tenha nenhuma raiz real positiva. Tentou-se utilizar o método de Cardan para $T > 0$.

Chegou-se a um sistema não linear de equações parametrizadas, difícil de ser resolvido à mão ou com *Maple*. Essa parte do trabalho foi abandonada, por não se configurar como integrante do escopo inicial deste trabalho, mas como uma possível extensão do mesmo.