

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

**PROPOSTA DE GPC AUTO-TUNING USANDO ALGORITMO
GENÉTICO**

DISSERTAÇÃO DE MESTRADO

ADEMIR JOSÉ PIONTKOVSKY FILHO

VITÓRIA

2018

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA
ELÉTRICA

**PROPOSTA DE GPC AUTO-TUNING USANDO ALGORITMO
GENÉTICO**

ADEMIR JOSÉ PIONTKOVSKY FILHO

Dissertação de Mestrado
apresentada como requisito parcial para
obtenção do título de Mestre em
Engenharia Elétrica.

Orientador:
Prof. Dr. José Leandro Félix Salles.

VITÓRIA
2018

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

P662p Piontkovsky Filho, Ademir José, 1990-
Proposta de GPC *auto-tuning* usando algoritmo genético /
Ademir José Piontkovsky Filho. – 2018.
95 f. : il.

Orientador: José Leandro Félix Salles.
Dissertação (Mestrado em Engenharia Elétrica) –
Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Controle preditivo. 2. Algoritmos genéticos. 3. Sistemas de controle ajustável. 4. Controlador de auto-ajuste. 5. Controle adaptativo (Sistema). 6. Sistemas lineares variantes no tempo. I. Salles, José Leandro Félix. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

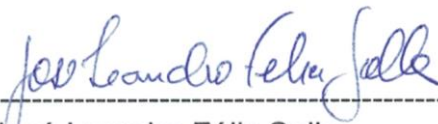
CDU: 621.3

ADEMIR JOSÉ PIONTKOVSKY FILHO

PROPOSTA DE GPC AUTO-TUNING USANDO ALGORITMO GENÉTICO

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

COMISSÃO EXAMINADORA



Prof. Dr. José Leandro Félix Salles
Universidade Federal do Espírito Santo
Orientador



Prof. Gustavo Maia de Almeida
Instituto Federal de Tecnologia - Serra (ES)



Prof. Dr. Klaus Fabian Côco
Universidade Federal do Espírito Santo
Dep. de Eng. Elétrica

Que seja feita a vontade de Deus.

Agradecimentos

A Deus, por todas as bênçãos que recebo nessa vida. A minha mãe, por todo o apoio e cuidado. A toda minha família, pela união. A todos os meus amigos, especialmente a Ana Paula Miranda Diniz, pelos auxílios. Ao meu orientador, pela confiança, compreensão e direção. À Capes, pelo apoio financeiro por meio da concessão de bolsas de apoio à pesquisa.

Muito Obrigado!

Resumo

Esta dissertação propõe uma técnica automática de sintonia para Controlador Preditivo Generalizado (GPC) com horizonte de controle igual a um para ser aplicado em qualquer planta, com a intenção de se projetar GPC adaptativos no modo *auto-tuning*. O algoritmo de sintonia é inicializado através de uma pré-análise do conjunto de parâmetros de sintonia que garantam a estabilidade da planta em malha fechada. Tal região de estabilidade é utilizada pelo Algoritmo Genético (AG) para buscar os parâmetros que fornecem o melhor sobressinal e tempo de acomodação da resposta ao degrau. Na pré-análise deve-se escolher o valor do parâmetro horizonte de previsão (h_p) e o intervalo desejado dos parâmetros λ e δ ao se avaliar o Lugar das Raízes da equação característica do sistema em malha fechada gerado através de uma interface gráfica. Através de simulações do GPC Adaptativo no modo *auto-tuning* utilizando o AG para plantas estáveis, instáveis, de fase mínima e não-mínima, mostra-se que o algoritmo proposto torna a convergência do AG mais rápida. Mostra-se nesta dissertação que em algumas plantas a região de estabilidade do sistema em malha fechada com o controlador GPC não se altera ao aumentar o valor de h_c , o que permite neste caso aplicar a técnicas de sintonia automática proposta para o GPC com horizonte de controle maior que 1. Além disto, é proposta uma regra de sintonia para o GPC *auto-tuning*, gerada através de uma sintonia off-line usando algoritmo genético e a interface gráfica, para uma planta de primeira ordem.

Palavras-Chave: Controle Preditivo Generalizado, Algoritmo Genético, Controle Adaptativo, *auto-tuning*, Sistemas Lineares Variantes no Tempo.

Abstract

This dissertation proposes an automatic tuning technique for Generalized Predictive Controller (GPC) with control horizon equal to one to be applied in any plant, with the intention of designing adaptive GPC in auto-tuning mode. The tuning algorithm is initialized through a pre-analysis of the set of tuning parameters that guarantee the stability of the plant in closed loop. Such a region of stability is used by the Genetic Algorithm (GA) to search for the parameters that provide the best overshoot and settling time of the step response. In the pre-analysis, one must choose the value of the prediction horizon parameter (h_p) and the desired interval of the λ and δ parameters when evaluating the Root Locus of the characteristic equation of the closed-loop system generated through a graphical interface. Through simulations of auto-tuning mode Adaptive GPC using the GA for stable, unstable, minimum and non-minimal phase plants, it is shown that the proposed algorithm makes AG convergence faster. It is shown in this dissertation that in some plants the stability region of the closed-loop system with the GPC controller does not change when increasing the value of h_c , which allows in this case to apply the proposed automatic tuning techniques to the GPC with horizon of control greater than 1. In addition, a tuning rule is proposed for GPC auto-tuning, generated through an offline tuning using genetic algorithm and the graphic interface, for a first-order plant.

Keywords: Generalized Predictive Control, Genetic Algorithm, Adaptive Control, auto-tuning, Time-Variants Linear Systems.

Lista de Figuras

Figura 2.1: Visualizando de forma gráfica o aumento de robustez do sistema ao se aumentar o valor de hp , a) $hp = 2$, b) $hp = 5$	32
Figura 2.2: Visualizando de forma gráfica o aumento de robustez do sistema ao se aumentar o valor de hp , a) $hp = 6$, b) $hp = 17$	32
Figura 2.3: Forma da roleta para o método seleção por roleta.....	34
Figura 2.4: Gráfico do Lugar das Raízes de função discreta para escolha do intervalo de $\frac{\lambda}{\delta}$, a) valor mínimo de $\frac{\lambda}{\delta}$, b) valor máximo de $\frac{\lambda}{\delta}$	36
Figura 2.5: Gráfico do Lugar das Raízes da função discreta mostrando a distribuição da primeira população para 50 indivíduos para o caso dos indivíduos criados utilizando a lógica de pequenos intervalos da região de $\frac{\lambda}{\delta}$ escolhida. 37	37
Figura 2.6: Gráfico do Lugar das Raízes da função discreta mostrando a distribuição da primeira população para 50 indivíduos para o caso dos indivíduos criados utilizando a lógica simples $a + \text{rand}(1,1)b$	38
Figura 3.1: Diagrama de blocos de <i>auto-tuning controller</i>	40
Figura 3.2: Diagrama de blocos de <i>self-tuning controller</i>	40
Figura 3.3: Diagrama de blocos para o sistema de Controle <i>Auto-tuning</i> utilizado neste trabalho.....	41
Figura 3.4: Diagrama de blocos do algoritmo de identificação recursiva utilizado neste trabalho.....	45
Figura 3.5: Diagrama de blocos para o sistema de Controle Adaptativo <i>auto-tuning</i> com controlador GPC utilizando AG na resintonia.....	48
Figura 4.1: Gráfico mostrando o padrão das simulações deste trabalho	53
Figura 4.2: Convergência dos parâmetros identificados por Identificação Recursiva em malha aberta com entrada PRBS da planta $G1z - 1$	57
Figura 4.3: Convergência dos parâmetros identificados por Identificação Recursiva em malha fechada com entrada sendo sinal de controle da planta $G1z - 1$	58
Figura 4.4: Convergência dos parâmetros identificados por Identificação Recursiva em malha aberta com entrada PRBS da planta $G2z - 1$	59

Figura 4.5: Convergência dos parâmetros identificados por Identificação Recursiva em malha fechada com entrada sendo sinal de controle da planta $G2z - 1$	59
Figura 4.6: Sinais de Saída, referência e controle da Simulação 1	61
Figura 4.7: Sinais de Saída, Referência e Controle da Simulação 3	62
Figura 4.8: Sinais de Saída, Referência e Controle da Simulação 5	65
Figura 4.9: Sinais de Saída, Referência e Controle da Simulação 8	67
Figura 4.10: Sinais de Saída, Referência e Controle da Simulação 9	70
Figura 4.11: Sinais de Saída, Referência e Controle da simulação da Planta $G3$	72
Figura 4.12: Sinais de Saída, Referência e Controle da simulação da Planta $G4$	73
Figura 4.13: Resposta encontrada pelo AG para a Planta $G5z - 1$	76
Figura 4.14: Lugar das Raízes da planta $G6z - 1$.em malha fechada com GPC com a) $hp = d + 1 = 2$ e b) $hp = d + 1 + 3 = 5$	77
Figura B.1: Interface Gráfica, dados de entrada da planta	88
Figura B.2: Interface Gráfica, dados de entrada do GPC	88
Figura B.3: Interface Gráfica, dados de entrada do Algoritmo Genético	89
Figura C.1: Gráfico do Lugar das Raízes de $G(z-1)$ para os casos de a) $hp = 5$ e b) $hp = 8$	91
Figura C.2: Resposta encontrada pelo AG ao se escolher o intervalo de λ/δ de 0,0022 até 0,191 (Caso 1)	92
Figura C.3: Resposta encontrada pelo AG ao se escolher o intervalo de λ/δ de 0 até infinito (Caso 2)	93
Figura C.4: Gráfico do Lugar das Raízes da função discreta mostrando a evolução do Algoritmo Genético para o intervalo de λ/δ de 0,0022 até 0,191	94
Figura C.5: Gráfico do Lugar das Raízes da função discreta mostrando a evolução do Algoritmo Genético para o intervalo de λ/δ de 0 até infinito	95

Lista de Tabelas

Tabela 1: Parâmetros de $G_1(z - 1)$ reais e identificados através da Simulação 1	61
Tabela 2: Sintonia de parâmetros do PID na Simulação 1	61
Tabela 3: Parâmetros de $G_2(z - 1)$ reais e identificados através da Simulação 3	63
Tabela 4: Sintonia de parâmetros do PID na Simulação 3	63
Tabela 5: Resumo dos resultados obtidos através das Simulações 1, 2, 3 e 4.	64
Tabela 6: Parâmetros de $G_1(z - 1)$ reais e identificados através da Simulação 5	66
Tabela 7: Sintonia de parâmetros do GPC na Simulação 5	66
Tabela 8: Parâmetros de $G_2(z - 1)$ reais e identificados através da Simulação 8.	68
Tabela 9: Sintonia de parâmetros do GPC na Simulação 8.	68
Tabela 10: Resumo dos resultados obtidos através das Simulações 5, 6, 7 e 8.	69
Tabela 11: Parâmetros de $G_1(z - 1)$ reais e identificados através da Simulação 9.	70
Tabela 12: Sintonia de parâmetros do GPC na Simulação 9.	70
Tabela 13: Resumo dos resultados obtidos através das Simulações 9 e 10. ..	71
Tabela 14: Sintonia de parâmetros do GPC na simulação da Planta G_3	73
Tabela 15: Sintonia de parâmetros do GPC na simulação da Planta G_4	74
Tabela 16: Resumo dos resultados das simulações G_3 e G_4	74
Tabela 17: Resultados de testes para $\lambda/\delta = 0,104$, ME para $h_p = 5$ com a Planta $G_5z - 1$	76
Tabela 18: Rotinas associadas ao código do Algoritmo Genético utilizado neste trabalho	85

Lista de Siglas

AG	Algoritmo Genético
CARIMA	<i>Controlled Auto-Regressive Integrated Moving Average</i>
DMC	<i>Dynamic Matrix Control</i>
GPC	<i>Generalized Predictive Control</i>
ISE	<i>Integral Squared Error</i>
LQG	<i>Linear Quadratic Gaussian Control</i>
LR	Lugar das Raízes
MPC	<i>Model Based Predictive Control</i>
MQR	Mínimos Quadrados Recursivos
PID	Controlador Proporcional, Integral e Derivativo
PRBS	<i>Pseudo-Random Binary Signal</i>
STC	<i>Self-Tuning Controller</i>

Lista de Símbolos

C	matriz de covariância da identificação recursiva
d	atraso de transporte discreto
$e(k)$	ruído gaussiano branco de média zero
$\hat{e}(k)$	erro de previsão, entre a saída real e a saída estimada da identificação recursiva
e_c, e_r	erro de realimentação de um sistema em malha fechada
F	matriz de respostas livres do GPC
$G_i(s)$	Função de Transferência de Plantas no Dominio de Laplace
$G_i(z^{-1})$	Função de Transferência de Plantas Discretas
h_c	horizonte de controle
h_m	valor inicial do horizonte de previsão
h_p	horizonte de previsão
I	matriz identidade
j	instantes a frente para previsão
J	Função custo do MPC
K	ganho estático
k	Instante de tempo discreto
k_d	ganho derivativo do controlador PID
k_i	ganho integral do controlador PID
K_p, k_r	ganho proporcional do controlador PID
K_{pu}	ganho crítico do controlador PID
$\max(u(k)), \max(u_k)$	valor máximo do sinal de controle
$\max(\Delta u(k))$	valor máximo da variação do sinal de controle
N	valor máximo o qual j atinge
na	ordem do polinômio do denominador da planta
nb	ordem do polinômio do numerador da planta
n_r	ruído na saída do sistema
OV	sobressinal - overshoot
q^{-i}	operador de atraso
q_c	parametros do controlador

Q_i	critério de identificação
Q_s	critério de síntese do controlador para o caso de controlador self-tuning
$ref, r(k)$	signal de referência do sistema
T_0	período de amostragem
T_d	tempo derivativo do controlador PID
T_i	tempo integral do controlador PID
t_s	tempo de acomodação
T_u	período crítico do controlador PID
$u(k)$	signal de controle
v	distúrbio no processo
$w(k)$	<i>referência futura</i>
$W(z^{-1})$	transformada Z da referência futura
$y(k)$	saída do sistema
$Y(z^{-1})$	transformada Z da saída do sistema
\mathbf{Y}	matriz de saídas previstas
$y(k)$	saídas previstas
δ	Ponderação de Erro de Previsão
$\Delta \mathbf{U}$	matriz de sequencia de sinais de controle
$\Delta u(k)$	variação do signal de controle
$\Delta U(z^{-1})$	transformada Z da variação do signal de controle
$\zeta, \varepsilon, \mu, \nu, \rho \text{ e } \eta$	variáveis auxiliares para cálculos em identificação recursiva
Θ	modelo identificado
$\hat{\theta}$	parâmetros estimados da planta pela identificação recursiva
λ	Ponderação de Esforço de Controle
τ	constante de tempo
φ	fator de esquecimento para identificação recursiva
$\Phi(t-1)$	vetor de regressores da identificação recursiva

Sumário

1	Introdução.....	16
	1.1 Objetivos e estrutura da dissertação.....	19
2	Controle Preditivo GPC.....	21
	2.1 Controlador Preditivo GPC.....	21
	2.1.1 GPC com $hc = 1$	27
	2.2 Sintonia do GPC.....	31
	2.2.1 Métodos heurísticos.....	31
	2.2.2 Método do lugar das Raízes para GPC com $hc = 1$	31
	2.2.3 Algoritmo Genético.....	33
	2.2.4 Sintonia de λ e δ usando AG.....	35
3	Controle Adaptativo.....	39
	3.1 Identificação Recursiva.....	42
	3.2 PID <i>Self-tuning</i> e <i>Auto-tuning</i>	46
	3.3 GPC adaptativo.....	47
	3.3.1 Com <i>auto-tuning</i> usando AG.....	47
	3.3.2 Com Regra de Sintonia gerada pelo próprio autor.....	49
	3.4 Sintonia dos parâmetros λ e δ GPC utilizando AG.....	49
	3.4.1 Sintonia de GPC estático.....	50
	3.4.2 AG na sintonia de GPC Adaptativo.....	50
	3.4.3 Regra de Sintonia na sintonia de GPC Adaptativo.....	51
4	Resultados.....	52
	4.1 Plantas e Simulações.....	52
	4.2 Identificação recursiva em malha aberta e em malha fechada.....	56
	4.3 PID adaptativo.....	60
	4.3.1 Análise da simulação1: planta G1(s) com PID e identificação Recursiva.....	60
	4.3.2 Análise da simulação2: planta G1(s) com PID e sem identificação recursiva.....	62
	4.3.3 Análise da simulação3: planta G2(s) com PID e identificação Recursiva.....	62
	4.3.4 Análise da simulação4: planta G2(s) com PID e sem identificação recursiva.....	63
	4.3.5 Resumo das análises do PID para as plantas G1(s) e G2(s).....	64
	4.4 GPC adaptativo com sintonia usando AG.....	64

4.4.1	Análise da simulação5: planta $G1(s)$ com GPC com AG e identificação Recursiva.....	65
4.4.2	Análise da simulação6: planta $G1(s)$ com GPC com AG e identificação Forçada.....	66
4.4.3	Análise da simulação7: planta $G2(s)$ com GPC com AG e identificação Recursiva.....	66
4.4.4	Análise da simulação8: planta $G2(s)$ com GPC com AG e sem identificação recursiva.....	67
4.4.5	Resumo das Análise do GPC com AG para plantas $G1(s)$ e $G2(s)$	68
4.5	GPC adaptativo com sintonia usando regra de sintonia	69
4.5.1	Análise da simulação9: planta $G1(s)$ com GPC com Regra de Sintonia e identificação Recursiva	69
4.5.2	Análise da simulação10: planta $G1(s)$ com GPC com Regra de Sintonia e sem identificação recursiva.....	71
4.5.3	Resumo das análises do GPC com Regra de Sintonia para a planta $G1(s)$..	71
4.6	Simulações para as plantas $G3(s)$ e $G4(s)$	71
4.7	Simulações para as plantas $G5(s)$ e $G6(s)$	75
4.8	Conclusão dos Resultados das simulações.....	77
5	Conclusões	79
	Referências Bibliográficas	81
	Apêndice A – Funções do código do Algoritmo Genético	85
	Apêndice B – Interface Gráfica	87
	Apêndice C – Vantagem da escolha de uma boa população inicial	91

1 Introdução

O Controle Preditivo Baseado em Modelo (MPC) vem sendo aplicado cada vez mais no ambiente industrial nestas ultimas décadas por se tratar de técnicas que são adequadas para controlar processos mono e multivariáveis, com e sem restrições, que apresentam elevado tempo morto e parâmetros variantes no tempo (CAMACHO e BORDONS, 2000). O MPC engloba uma diversidade de algoritmos de controle que utilizam o modelo do processo para realizar previsão da saída futura, obtendo o sinal de controle através da otimização de uma função do erro de previsão denominada função custo (ver Equação (1)). (ALMEIDA e SALLES, 2016)

Incluso em MPC tem-se as classes baseadas em modelos lineares como Controle por Matriz Dinâmica (DMC), o Controle Preditivo Generalizado (GPC), o Controlador Linear Quadrático Gaussiano (LQG), e as classes baseadas em modelos não lineares como o Modelo Não Linear Hammerstein, o Modelo Neural, dentre outros (ALMEIDA e SALLES, 2016).

A função custo a ser minimizada é constituída pelo erro médio entre a saída do sistema previsto e a referência prevista, bem como pelo quadrado da variação do controle. Como mostra a Equação (1).

$$J(hm, hp, hc) = \sum_{j=hm}^{hp} \delta \left(y(k+j) - w(k+j) \right)^2 + \sum_{j=hm}^{hc} \lambda \left(\Delta u(k+j-1) \right)^2 \quad (1)$$

Onde J é a função custo, hm é o valor inicial do horizonte de previsão que deve ser igual ao atraso de transporte discreto da planta d , j são as previsões passos a frente e k representa o instante de tempo discreto. Sendo que hm , hp , hc , k e j assumem valores inteiros e δ , λ , y , w e Δu assumem valores reais.

O parametro hp é o horizonte de previsão, que sugerido por Clarke, Mohtadi e Tuffs (1987), Clarke e Mohtadi (1989) e Banerjee e Shah (1992) deve ser sintonizado igual a 10 e entre valores $2na-1$ e $\frac{t_s}{T_0}$, sendo na a ordem do polinômio do denominador da planta, t_s o tempo de acomodação e T_0 o período

de amostragem do sistema. Ja Rani e Unbehauen (1997) sugerem sintonizar h_p entre o máximo tempo morto e o tempo de acomodação do sistema. Além disso, quanto maior o valor deste parâmetro mais robusto será o sistema, porém aumentará a complexidade dos calculos, aumentando, assim, o esforço computacional para se implementar o MPC.

O parametro h_c é o horizonte de controle, que é considerado $h_c < h_p$ e, geralmente, escolhido de forma arbitrária como realizado pelos autores Kanjilal (1995), Almeida e Salles (2016) e Tran *et al.* (2014). Este parâmetro tem como característica suavizar o sinal de controle e também aumentar o esforço computacional exigido para se implementar o sistema.

As variáveis $y(k+j)$ e $w(k+j)$ são as sequências de saídas previstas e a trajetória de referência futura a j passos à frente do sistema. Já $\Delta u(k+j-1)$ é a variação do sinal de controle obtida ao se minimizar a função custo J . É importante mencionar que o sinal de controle que é aplicado ao sistema é $u(k+j) = \Delta u(k+j) + u(k+j-1)$.

Quanto aos parametros δ e λ , estes chamam-se Ponderação de Erro de Previsão e Ponderação de Esforço de Controle, respectivamente. A maioria dos autores assumem $\delta=1$ na sintonia, pois este parametro se torna dispensável durante a manipulação das equações do GPC. Já o λ foi sintonizado por tentativa e erro em Karacan e Hapoglu (2000) e por um valor fixo, caso o horizonte de previsão seja sintonizado corretamente Yoshitani e Hasegawa (1998). Uma maneira mais clara de sintonizar este parâmetro é utilizar o método do Lugar das Raízes do sistema, como foi proposto por Mazoco e Salles (2015) e será utilizada nesta dissertação.

Familiarizado com a função custo, os parâmetros dos MPC e com a classe de MPC Lineares, atenta-se aos tópicos DMC e GPC apresentados por Camacho e Bordons (2000), e experimentados nos artigos Kokate, Waghmare e Deshmukh (2010) e Sato (2010). O DMC é baseado na resposta ao degrau do sistema, em dados de saída e entrada, e não é cabível quando a planta é variante no tempo. Já o GPC é especialmente interessante no caso de plantas variantes

no tempo por ser baseado em um modelo linear de Função de Transferência Discreta da Planta, que é o modelo CARIMA.

Logo, o GPC pode ser utilizado em Controle Adaptativo e na literatura é geralmente utilizado no modo “*self-tuning*”, como abordam Bitmead, Gevers e Wertz (1990), Sato (2010) e Bobál, Kubalcík e Dostál (2013).

Mais um detalhe que torna o GPC interessante é quando se é escolhido o valor do horizonte de controle igual a um ($h_c=1$), visto que isto reduz as dimensões das matrizes presentes no cálculo do GPC e permite explicitar equações do sistema no domínio Z, simplificando consideravelmente os cálculos e, com isso, diminuindo a demanda computacional facilitando a implementação em *hardwares*. Além do que, ainda se torna possível avaliar a estabilidade do sistema em malha fechada utilizando o Lugar das Raízes, como proposto por Mazoco e Salles (2015).

Apesar disso, ainda não se observa na literatura o uso do GPC com regras de sintonia, diferentemente dos controladores PID que já se encontra regras de sintonia para plantas de primeira, segunda e até terceira ordem, Bobál e Böhm *et al.* (2005), Sinthipsomboon, Pongaeen e Pratumswan (2011) e Talib e Darus (2013). Tais regras são aplicadas em controladores PID *self-tuning* (BOBÁL, BÖHM, *et al.*, 2005).

Todavia, pensando na sintonia dos parâmetros do GPC, esta pode ser feita da mesma forma com que foi mencionado anteriormente para qualquer outro MPC, mas, se a planta variar, o desempenho da resposta será prejudicado e então será necessário a realização de uma nova sintonia dos parâmetros do controlador GPC.

Outra opção para se sintonizar um MPC é utilizar algoritmos evolutivos que buscam a sintonia ótima, como o Algoritmo Genético (AG) (SARIMVEIS e BAFAS, 2002) e (AL-DUWAISH e NAEEM, 2001), que é mais eficiente. Porém, demanda um maior tempo devido ao esforço computacional exigido e, dessa forma, o AG é utilizado na pré-sintonia do MPC sendo aplicável também na sintonia *online* quando o sistema tiver resposta lenta, sendo que, métodos de

sintonia *online* são utilizados por Al-Ghazzawai *et al.* (2001), Mahmoudi, Lesani e Khabouri (2013) e Ziogou *et al.* (2013).

Compreendendo isto, torna-se então importante buscar formas de tornar a convergência do AG mais rápida para que este possa ser utilizado na sintonia do GPC em tempo real. Uma das formas de alcançar este objetivo é fornecer uma população inicial do AG que esteja próximo do ótimo local da função que se deseja minimizar. Este tipo de solução foi proposta em Fan e Joo (2009) o qual utiliza o AG para desenvolver um controlador PID *auto-tuning*. Neste artigo, utiliza-se o método de Ziegler-Nichols para definir os intervalos que contém os parâmetros do controlador PID que garantem a estabilidade, os quais constituirão a população inicial do AG.

1.1 Objetivos e estrutura da dissertação

O objetivo geral é desenvolver um GPC Adaptativo com Horizonte de controle igual a um, de maneira que seja resintonizável através do Algoritmo Genético.

Objetivos Específicos:

1. Desenvolver algoritmo GPC adaptativo com horizonte igual a um com sintonia *on-line* usando AG para sistemas mono-variáveis com plantas de primeira e segunda ordem com atraso de transporte de fase mínima ou não;
2. Criar uma interface gráfica para auxiliar na escolha dos parâmetros de sintonia do GPC através do *Lugar das Raízes da Equação Característica* dos sistemas mono-variáveis quando o parâmetros λ e horizonte de previsão h_p variam;
3. Desenvolver uma regra de sintonia do GPC *auto-tuning* com $h_c=1$ para aplicação em controle adaptativo;
4. Desenvolver algoritmos PID adaptativos de plantas de primeira e segunda ordem para comparar com o GPC adaptativo;

Parte dos resultados obtidos nesta dissertação foi aceito para publicação na Conferência Brasileira de Dinâmica, Controle e Aplicações (DINCON2017), outubro 2017, São José do Rio Preto, São Paulo. (PIONTKOVSKY FILHO e SALLES, 2017)

Esta dissertação é composta de cinco Capítulos. O primeiro Capítulo, que é a introdução, contém uma contextualização do problema de sintonia dos controladores baseado em modelo (MPC).

O Capítulo 2 trata das variáveis e matrizes das relações matemáticas, leis de controle, funções de transferências e parâmetros do GPC, bem como detalha a metodologia e algoritmos aplicados na sintonia do GPC com o auxílio de uma interface gráfica produzida neste trabalho utilizando ou não o Algoritmo Genético (AG).

No Capítulo 3, aborda-se o tema de Controle Adaptativo, a diferença entre controlador adaptativo no modo "*self-tuning*" e "*auto-tuning*", a importância da Identificação Recursiva e de um módulo de supervisão. Além disso, mostra-se as regras de sintonias utilizadas nos algoritmos desenvolvidos neste trabalho para se sintonizar os controladores PID Adaptativos no modo *auto-tuning*, e a configuração do Algoritmo Genético para se sintonizar controladores GPC Adaptativos no modo *auto-tuning*. Ainda sobre GPC Adaptativo, é proposto uma regra de sintonia para uma planta de primeira ordem para um pequeno intervalo de ganho K e constante de tempo τ .

A par do conteúdo dos capítulos anteriores, o Capítulo 4 expõe todos os testes realizados e resultados obtidos. Inicialmente, apresenta-se as plantas utilizadas nas simulações e posteriormente, mostra-se que o algoritmo de Identificação Recursiva tem êxito em identificar os parâmetros das plantas para o caso de malha aberta, mas, diverge quando em malha fechada. No entanto, em ambos os casos se mantem o erro entre a saída real e a estimada no intervalo $[-0,01, +0,01]$. Por ultimo, mostra-se e explica-se os gráficos e tabelas referentes a todas simulações realizadas para controladores PID e GPC.

Finalmente, apresenta-se no Capítulo 5 as conclusões deste estudo, juntamente de comentários adicionais e sugestões para trabalhos futuros.

2 Controle Preditivo GPC

Como definição do GPC pode-se dizer que é um controle preditivo baseado em modelo representado na forma de função de transferência. Este tópico apresenta a solução do GPC através de equações recursivas na forma matricial e suas dimensões dependem dos horizontes de previsão e controle. Neste contexto, observa-se que essas equações dependem também dos parâmetros do modelo, o que permite a implementação em controladores adaptativos. As equações do GPC serão desenvolvidas para sistemas monovariáveis sem distúrbio e sem restrições.

2.1 Controlador Preditivo GPC

Teoricamente o Controlador GPC se apresenta nos modelos monovariável e multivariável, ambos com ou sem restrições. Para fins de realização deste trabalho tomou-se como base o monovariável irrestrito, com uma forma de buscar parâmetros que atendam as especificações do usuário, causando uma restrição indireta. O GPC é baseado no modelo CARIMA de equação à diferenças, conforme apresenta a Equação (2).

$$A(q^{-1})y(kT_0) = q^{-d}B(q^{-1})u(kT_0 - 1) + C(q^{-1})\frac{e(kT_0)}{\Delta} \quad (2)$$

Sendo q^{-1} o operador de atraso, de forma que $q^{-1}y(k) = y(k-1)$ (AGUIRRE, 2007). Também é encontrado na literatura a utilização de z^{-1} como operador de atraso, isto deve-se ao fato de que no domínio Z o operador z^{-1} causar o mesmo efeito que o operador de atraso q^{-1} . A variável k representa o tempo discreto e assume valores inteiros, o período de amostragem T_0 é considerado igual a um. Os termos $y(k)$, $u(k-1)$ e $e(k)$ são a saída do sistema, a sequencia de controle e um ruído branco gaussiano de média zero, respectivamente. A variável d representa o atraso de transporte discreto da planta e $\Delta = 1 - q^{-1}$. $A(q^{-1})$, $B(q^{-1})$ e $C(q^{-1})$ os polinomios da forma mostrada nas Equações (3), (4) e (5).

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_{na}q^{-na} \quad (3)$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + \dots + b_{nb}q^{-nb} \quad (4)$$

$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_{nc}q^{-nc} \quad (5)$$

O conceito básico do Controlador GPC constitui em calcular uma sequência de sinais de controle futuros de modo a minimizar uma função custo definida através de um horizonte de previsão e de controle. Esta função custo foi mencionada anteriormente na Equação (1).

Ao se multiplicar os dois lados da Equação (2) por Δ , e por motivo de simplificação, considerando $C(q^{-1})=1$ para que o ruído seja apenas do tipo gaussiano branco e o período de amostragem $T_0 = 1$ [s], chega-se na Equação (6).

$$\Delta A(q^{-1})y(kT_0) = \Delta q^{-d}B(q^{-1})u(kT_0-1) + \Delta C(q^{-1})\frac{e(kT_0)}{\Delta}$$

$$\tilde{A}(q^{-1})y(k) = q^{-d}B(q^{-1})\Delta u(k-1) + e(k) \quad (6)$$

Sendo $\tilde{A}(q^{-1}) = \Delta A(q^{-1}) = (1-q^{-1})A(q^{-1}) = 1 + \tilde{a}_1q^{-1} + \tilde{a}_2q^{-2} + \dots + \tilde{a}_{na+1}q^{-(na+1)}$. O próximo passo é avançar a Equação (6) a j instantes a frente, como mostra a Equação (7).

$$\tilde{A}(q^{-1})y(k+j) = q^{-d}B(q^{-1})\Delta u(k+j-1) + e(k+j) \quad (7)$$

Nesta etapa é importante que sejam ressaltados dois detalhes: os valores que j assume e que a saída prevista dependerá, entre outros, das saídas nos instantes futuros.

Ao se explicar os valores assumidos por j nesta etapa, já fica visível a influência do parâmetro hp no sistema, pois: $j = d+1, \dots, d+N$ e nesta dissertação é considerado $N = hp - d$. Dessa forma, mostra-se que a escolha do valor de hp afeta a quantidade de previsões que serão calculadas.

Quanto a saída prevista, basta avaliar o lado esquerdo da igualdade da Equação (7) para perceber que esta dependerá de saídas futuras, e isto não é desejado. Para contornar esta situação pode-se utilizar um antigo problema matemático associado a matemáticos como *Diophantine*, *Bezout* e *Aryabhata*

(CHEN, 1999) chamado de Equação de Diofantina (ALMEIDA e SALLES, 2016) que tem o formato da Equação (8).

$$1 = E_j(q^{-1})\tilde{A}(q^{-1}) + F_j(q^{-1})q^{-j} \quad (8)$$

Sendo que o polinômio $E_j(q^{-1})$ é o quociente da divisão de 1 por $\tilde{A}(q^{-1})$, de grau $j-1$, e, o polinômio $F_j(q^{-1})$ é o resto desta divisão de grau na . Multiplicando os dois lados da Equação (7) pelo polinômio $E_j(q^{-1})$ e percebendo que a Equação (8) é equivalente a $E_j(q^{-1})\tilde{A}(q^{-1}) = 1 - F_j(q^{-1})q^{-j}$, chega-se na Equação (9).

$$\begin{aligned} E_j(q^{-1})\tilde{A}(q^{-1})y(k+j) &= E_j(q^{-1})q^{-d}B(q^{-1})\Delta u(k+j-1) + E_j(q^{-1})e(k+j) \\ (1 - F_j(q^{-1})q^{-j})y(k+j) &= E_j(q^{-1})q^{-d}B(q^{-1})\Delta u(k+j-1) + E_j(q^{-1})e(k+j) \\ y(k+j) - F_j(q^{-1})q^{-j}y(k+j) &= E_j(q^{-1})q^{-d}B(q^{-1})\Delta u(k+j-1) + E_j(q^{-1})e(k+j) \\ y(k+j) - F_j(q^{-1})y(k+j-j) &= E_j(q^{-1})B(q^{-1})\Delta u(k+j-1-d) + E_j(q^{-1})e(k+j) \\ y(k+j) &= E_j(q^{-1})B(q^{-1})\Delta u(k+j-d-1) + E_j(q^{-1})e(k+j) + F_j(q^{-1})y(k) \end{aligned} \quad (9)$$

Ao se aplicar o valor esperado na Equação (9) o termo $E_j(q^{-1})e(k+j)$ é eliminado por estar no futuro, pois o grau de $E_j(q^{-1})$ é $j-1$, resultando na Equação (10). (ALMEIDA e SALLES, 2016)

$$\begin{aligned} y(k+j) &= E_j(q^{-1})B(q^{-1})\Delta u(k+j-d-1) + F_j(q^{-1})y(k) \\ &= G_j(q^{-1})\Delta u(k+j-d-1) + F_j(q^{-1})y(k) \end{aligned} \quad (10)$$

Declarando $G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) = g_0 + g_1q^{-1} + g_2q^{-2} + \dots + g_{nb+j-1}q^{-(nb+j-1)}$, o termo $G_j(q^{-1})\Delta u(k+j-d-1)$ conterà ações de controle passadas, presente e futuras. No entanto, pode-se definir o polinômio $G_j(q^{-1})$ no formato de dois outros polinômios, $G_j(q^{-1}) = H_j(q^{-1}) + \overline{H}_j(q^{-1})$, onde os elementos de $H_j(q^{-1})$ abrangem o presente e futuro, e os de $\overline{H}_j(q^{-1})$ abrangem apenas o passado, como mostra as Equações (11) e (12).

$$H_j(q^{-1}) = g_0 + g_1q^{-1} + \dots + g_{j-d-1}q^{-(j-d-1)} \quad (11)$$

$$\bar{H}_j(q^{-1}) = g_{j-d} q^{-(j-d)} + \dots + g_{nb+j-1} q^{-(nb+j-1)} \quad (12)$$

Dessa forma, chama-se a atenção para a variável j : tal que $j = d+1, \dots, d+N$ e nesta dissertação é considerado $N = hp - d$. Então, o valor de hp também influencia na ordem dos polinômios presentes no equacionamento do GPC, e, conseqüentemente, no custo computacional para implementação em *hardware*.

Conforme Almeida e Salles (2016), empregando o Lema 1 de Almeida (2006) chega-se na Equação (13) mostrada a abaixo.

Lema 1 de Almeida (2006) :

$$\text{Para } j = d+l: \bar{H}_j(q^{-1})\Delta u(k+j-d-1) = q^l \bar{H}_{d+l}(q^{-1})\Delta u(k-1)$$

Prova: Para $j = d+l$, $\bar{H}_{d+l}(q^{-1}) = (g_l q^{-l} + \dots + g_{nb+d+l-1} q^{-(nb+d+l-1)})$ e constatar que:

$$\begin{aligned} \bar{H}_j(q^{-1})\Delta u(k+j-d-1) &= (g_{j-d} q^{-(j-d)} + \dots + g_{nb+j-1} q^{-(nb+j-1)})\Delta u(k+j-d-1) \\ &= (g_{d+l-d} q^{-(d+l-d)} + \dots + g_{nb+d+l-1} q^{-(nb+d+l-1)})\Delta u(k+d+l-d-1) \\ &= (g_l q^{-l} + \dots + g_{nb+d+l-1} q^{-(nb+d+l-1)})\Delta u(k+l-1) \\ &= \bar{H}_{d+l}(q^{-1})\Delta u(k+l-1) \\ &= \bar{H}_{d+l}(q^{-1})q^l \Delta u(k-1) \\ &= q^l \bar{H}_{d+l}(q^{-1})\Delta u(k-1) \quad \square. \end{aligned}$$

$$y(k+d+l) = H_{d+l}(q^{-1})\Delta u(k+l-1) + q^l \bar{H}_{d+l}(q^{-1})\Delta u(k-1) + F_{d+l}(q^{-1})y(k) \quad (13)$$

Considerando $1 \leq l \leq N$ na Equação (13) chega-se ao sistema de equações mostrada abaixo:

$$y(k+d+1) = H_{d+1}(q^{-1})\Delta u(k) + q^1 \bar{H}_{d+1}(q^{-1})\Delta u(k-1) + F_{d+1}(q^{-1})y(k)$$

$$y(k+d+2) = H_{d+2}(q^{-1})\Delta u(k+1) + q^2 \bar{H}_{d+2}(q^{-1})\Delta u(k-1) + F_{d+2}(q^{-1})y(k)$$

$$\vdots \quad \quad \quad \vdots$$

$$y(k+d+N) = H_{d+N}(q^{-1})\Delta u(k+N-1) + q^N \bar{H}_{d+N}(q^{-1})\Delta u(k-1) + F_{d+N}(q^{-1})y(k)$$

Aplicando o Lema 2 de Almeida (2006) neste sistema de equações chega-se na expressão matricial mostrada pela Equação (18).

Lema 2 de Almeida (2006) :

Seja um horizonte de controle hc e $l=1, \dots, N$. Então o termo $H_{d+l}(q^{-1})\Delta u(k+l-1)$ é representado através da Equação (14) para $l \leq hc$ e através da Equação (15) para $l \geq hc+1$.

$$H_{d+l}(q^{-1})\Delta u(k+l-1) = \begin{bmatrix} g_{l-1} & g_{l-2} & \dots & g_1 & g_0 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+j-d-2) \\ \Delta u(k+j-d-1) \end{bmatrix} \quad (14)$$

$$H_{d+l}(q^{-1})\Delta u(k+l-1) = \begin{bmatrix} g_{l-1} & g_{l-2} & \dots & g_{l+1-hc} & g_{l-hc} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+hc-2) \\ \Delta u(k+hc-1) \end{bmatrix} \quad (15)$$

Prova: Para $j=d+l$ e $l \leq hc$:

$$\begin{aligned} H_{d+l}(q^{-1})\Delta u(k+l-1) &= \left(g_0 + g_1 q^{-1} + \dots + g_{l-1} q^{-(l-1)} \right) \Delta u(k+l-1) \\ &= g_0 \Delta u(k+l-1) + g_1 \Delta u(k+l-2) + \dots + g_{l-1} \Delta u(k) \end{aligned} \quad (16)$$

sendo a Equação (16) equivalente a Equação (14). Para $l \geq hc+1$ tem-se que $\Delta u(k+l-1) = \Delta u(k+l-2) = \dots = \Delta u(k+hc+1) = \Delta u(k+hc) = 0$, então:

$$H_{d+l}(q^{-1})\Delta u(k+l-1) = g_{l-hc} \Delta u(k+hc-1) + g_{l-hc+1} \Delta u(k+hc-2) + \dots + g_{l-1} \Delta u(k) \quad (17)$$

sendo a Equação (17) equivalente a Equação (15). \square

$$\mathbf{Y} = \mathbf{G}\Delta\mathbf{u} + \bar{\mathbf{H}}(q^{-1})\Delta u(k-1) + \mathbf{F}(q^{-1})y(k) \quad (18)$$

sendo:

$$\mathbf{Y} = \begin{bmatrix} y(t+d+1) \\ y(t+d+2) \\ \vdots \\ y(t+d+N) \end{bmatrix}_{Nx1}, \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+hc-1) \end{bmatrix}_{hc \times 1}$$

$$\bar{\mathbf{H}}(q^{-1}) = \begin{bmatrix} q^1 \bar{H}_{d+1}(q^{-1}) \\ q^2 \bar{H}_{d+2}(q^{-1}) \\ \vdots \\ q^N \bar{H}_{d+N}(q^{-1}) \end{bmatrix}_{Nx1}, \quad \mathbf{F}(q^{-1}) = \begin{bmatrix} F_{d+1}(q^{-1}) \\ F_{d+2}(q^{-1}) \\ \vdots \\ F_{d+N}(q^{-1}) \end{bmatrix}_{Nx1}$$

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{hc-1} & g_{hc-2} & \dots & g_0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \dots & g_{N-hc} \end{bmatrix}_{N \times hc}$$

Nesta perspectiva, atenta-se para o detalhe que $N = hp - d$, reforçando mais uma vez a influência que o parâmetro hp tem no projeto do GPC, pois neste passo fica claro que este altera a quantidade de linhas das matrizes presentes no equacionamento do GPC. Já o parâmetro hc altera a quantidade de colunas da matriz \mathbf{G} , e, dessa forma, intervindo também no esforço computacional exigido na implementação.

Isto posto, chega-se na expressão de previsão na forma matricial descrita pela Equação (19).

$$\mathbf{Y} = \mathbf{G}\Delta\mathbf{U} + \mathbf{f} \quad (19)$$

sendo o termo $\mathbf{f} = \overline{\mathbf{H}}(q^{-1})\Delta u(k-1) + \mathbf{F}(q^{-1})y(k)$ intitulado por resposta livre.

2.1.1 GPC com $h_c=1$

É importante o $h_c=1$ visto que, facilita a implantação em *hardware* embarcado e, além disso, conforme menciona Mazoco e Salles (2015) pode-se desenhar o lugar das raízes.

O maior benefício de se manter o $h_c=1$ é o de facilitar a implementação e desta forma, possibilita a utilização de algoritmos propostos para simplificar ainda mais os cálculos, afim de que o sistema embarcado consiga realizar com rapidez. Outro benefício é que podem ficar explícitas as funções de transferência em malha fechada do GPC com a planta no domínio Z. Desta forma, têm-se as equações abaixo necessárias para se desenhar o lugar das raízes, conforme Mazoco e Salles (2015).

Partindo da Equação (20), que trata do vetor de variações do sinal de controle no tempo discreto, chega-se às equações (21), (22), (23), (24) e (25) conforme demonstra Mazoco e Salles (2015) sendo que, a Equação (22) é proveniente da transformada Z da Equação (21) enquanto que, as outras sofreram outras manipulações.

$$\begin{aligned} \Delta\mathbf{u} &= (\delta\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\delta\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \\ &= (\mathbf{G}^T\mathbf{G} + \frac{\lambda}{\delta}\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \end{aligned} \quad (20)$$

Na expressão (20) $\Delta\mathbf{u}$ é o vetor variação do sinal de controle de dimensão $h_c \times 1$; δ é o escalar Ponderação de Erro de Previsão; A matriz \mathbf{G}^T é transposta da matriz \mathbf{G} de dimensão $h_c \times N$; A matriz \mathbf{G} é uma das matrizes do controlador GPC de dimensão $N \times h_c$; λ é o escalar Ponderação de Esforço de Controle; \mathbf{I} é uma matriz identidade de dimensão $h_c \times h_c$; \mathbf{w} é o vetor de referências futuras de dimensão $N \times 1$; \mathbf{f} é o vetor resposta livre do GPC de dimensão $N \times 1$. É

importante lembrar que dentro do vetor resposta livre contem o sinal de saída $y(k)$ e que no desenvolvimento de Mazoco e Salles (2015) foi usado $N = hp - d$.

Sabendo disso, fica claro que o valor de hc altera a quantidade de linhas de $\Delta \mathbf{u}$. Abaixo, a Equação (21) mostra a variação do sinal de controle no instante atual k considerando $hc=1$, o que torna o vetor $\Delta \mathbf{u}$ em um escalar $\Delta u(k)$ (MAZOCO e SALLES, 2015). Note que o sinal de controle agora é um vetor 1×1 .

$$\Delta u(k) = \frac{\sum_{l=0}^{N-1} (g_l w_{l+1}) - \sum_{l=0}^{N-1} (g_l q^{l+1} \bar{H}_{d+l+1}(q^{-1}) \Delta u(k-1)) - \sum_{l=0}^{N-1} (g_l F_{d+l+1}(q^{-1}) y(k))}{\sum_{l=0}^{N-1} (g_l^2) + \frac{\lambda}{\delta}} \quad (21)$$

A escolha de $hc=1$ já reduz o custo computacional para implementação pelo fato de tornar o vetor $\Delta \mathbf{u}$ em um escalar. Além disso, recordando que $N = hp - d$ e avaliando a Equação (21), que é a solução explícita do controlador, note-se que o valor hp alterará o valor do índice final dos somatórios e, com isso, também afetando o custo computacional para implementação. De outra forma, escolher valores pequenos de hp também facilita a implementação do sistema.

Aplicando a Transformada Z na Equação (21), e com poucas manipulações, chega-se na Equação (22). O operador de atraso $q^{\pm i}$ passa para o operador $z^{\pm i}$.

$$\Delta U(z^{-1}) = \frac{W(z^{-1}) - \frac{\sum_{l=0}^{N-1} [g_l F_{d+l+1}(z^{-1})]}{\sum_{l=0}^{N-1} [g_l]} Y(z^{-1})}{\frac{\sum_{l=0}^{N-1} [g_l^2] + \sum_{l=0}^{N-1} [g_l z^{l+1} \bar{H}_{d+l+1}(z^{-1}) \Delta u(t-1)] + \frac{\lambda}{\delta}}{\sum_{l=0}^{N-1} [g_l]}} \quad (22)$$

Assim, para simplificar a análise, declara-se $S(z^{-1}) = \frac{\sum_{l=0}^{N-1} [g_l F_{d+l+1}(z^{-1})]}{\sum_{l=0}^{N-1} [g_l]}$ e

$$R(z^{-1}) = \frac{\sum_{l=0}^{N-1} [g_l^2] + \sum_{l=0}^{N-1} [g_l z^{l+1} \bar{H}_{d+l+1}(z^{-1}) \Delta u(t-1)] + \frac{\lambda}{\delta}}{\sum_{l=0}^{N-1} [g_l]}, \text{ e então a Equação (22) pode}$$

ser exibida no formato da Equação (23).

$$\Delta U(z^{-1}) = \frac{W(z^{-1}) - S(z^{-1})Y(z^{-1})}{R(z^{-1})} \quad (23)$$

onde $\Delta U(z^{-1})$ é a transformada Z do sinal $\Delta u(k)$. $W(z^{-1})$ é transformada Z do vetor w , que neste caso é um sinal de entrada constante tipo degrau, $Y(z^{-1})$ é a transformada Z do sinal de saída $y(k)$, $R(z^{-1})$ e $S(z^{-1})$ são polinômios em Z resultantes das manipulações.

A Equação (24) é o modelo CARIMA da planta no domínio de Z utilizado por Mazoco e Salles (2015):

$$\frac{Y(z^{-1})}{\Delta U(z^{-1})} = \frac{B(z^{-1})z^{-(d+1)}}{\tilde{A}(z^{-1})} \quad (24)$$

onde $B(z^{-1})$ é o numerador da planta no domínio de Z; $\tilde{A}(z^{-1})$ é o polinômio $A(z^{-1})$ multiplicado por Δ no domínio Z, sendo $\Delta = 1 - z^{-1}$.

Ao se utilizar a Equação (24) na Equação (23), chega-se na seguinte Equação (25) que é a Função de Transferência do Sistema em malha fechada:

$$\frac{Y(z^{-1})}{W(z^{-1})} = \frac{\gamma B(z^{-1})z^{-(d+1)}}{P(z^{-1}) + \frac{\lambda}{\delta} \tilde{A}(z^{-1})} \quad (25)$$

$$\text{Sendo } \gamma = \sum_{l=0}^{N-1} g_l$$

onde, g_l são os elementos da matriz \mathbf{G} do GPC, portanto γ é um escalar. $P(z^{-1})$ um polinômio em Z resultante das manipulações no desenvolvimento de Mazoco e Salles (2015) e tem o formato da Equação (26).

$$P(z^{-1}) = \left(\sum_{l=0}^{N-1} [g_l^2] + \sum_{l=0}^{N-1} [g_l z^l \bar{H}_{d+l+1}(z^{-1})] \right) \tilde{A}(z^{-1}) + B(z^{-1}) z^{-(d+1)} \left(\sum_{l=0}^{N-1} [g_l F_{d+l+1}(z^{-1})] \right) \quad (26)$$

Sendo que, g_l são os elementos da matriz \mathbf{G} do GPC; $\bar{H}_{d+l+1}(z^{-1})$ são polinômios criados a partir da matriz \mathbf{G} do GPC e $F_{d+l+1}(z^{-1})$ polinômios resultantes da solução da Equação de Diofantina.

Compreendendo estes termos e avaliando a Equação Característica do sistema em malha fechada, que é o denominador da Equação (25), nota-se que esta pode ser igualada a zero e escrita na forma da Equação (27):

$$1 + \frac{\lambda \tilde{A}(z^{-1})}{\delta P(z^{-1})} = 0 \quad (27)$$

Assim, pode-se observar que é semelhante a uma das estruturas de se desenhar o Lugar das Raízes, mais especificamente a estrutura de se desenhar o Lugar das Raízes para um sistema em malha fechada com um controlador do tipo Proporcional, como mostra a Equação (28).

$$1 + K \frac{N(z^{-1})}{D(z^{-1})} = 0 \quad (28)$$

Dessa forma, ao se desenhar o Lugar das Raízes para a Equação (27), demonstra-se que o ganho será $\frac{\lambda}{\delta}$ será equivalente ao ganho K .

2.2 Sintonia do GPC

Nesta seção aborda-se alguns métodos para sintonizar um GPC de forma heurística, além de um método para avaliar a estabilidade do sistema em malha fechada com GPC antes de se fazer a sintonia.

A sintonia dos parâmetros do GPC muitas vezes é feita por tentativa e erro por não haver outra forma de se analisar a estabilidade do mesmo em malha fechada.

Porém, como mencionado anteriormente quando $hc=1$ pode-se escrever de forma explícita a Equação (25), que é a função de transferência do sistema em malha fechada da planta com o GPC no domínio de Z, e com isso se torna possível analisar a estabilidade do sistema em malha fechada do GPC com uma planta escolhida.

2.2.1 Metodos heurísticos

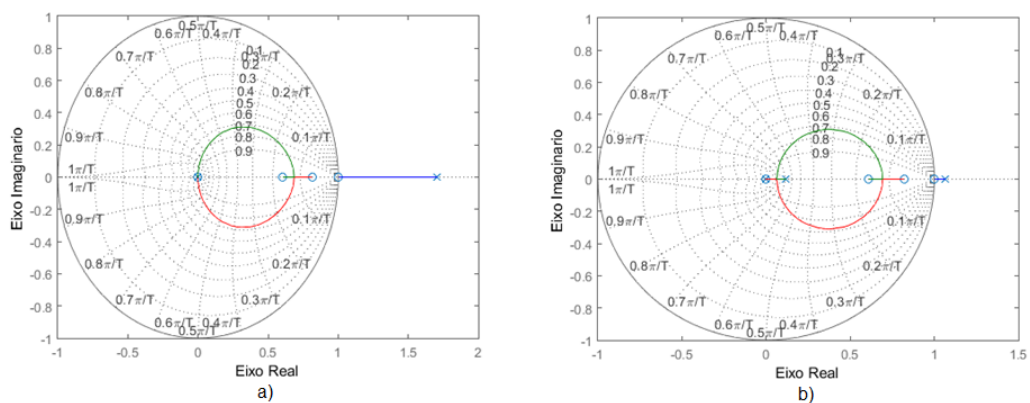
De acordo com o estudo realizado por Garriga e Soroush (2010), há uma diversidade de regras de sintonia de MPC utilizando métodos não automáticos, os quais são obtidos por meio de regras heurísticas que dependem das características do processo a ser controlado, e são desenvolvidas mediante a experiência do operador. Nos métodos não automáticos, a sintonia é realizada de maneira *offline*, que consiste em realizar uma pré-sintonia dos parâmetros iniciais do MPC simulando-se o modelo dinâmico que representa o processo a ser controlado. Em seguida testa-se o controlador MPC no processo real em malha fechada, e caso não sejam atendidas as especificações do desempenho do sistema de controle, executa-se uma ressintonia dos parâmetros iniciais.

2.2.2 Método do lugar das Raízes para GPC com $hc=1$

O Método Lugar das Raízes para GPC com $hc=1$ foi utilizado em Mazoco e Salles (2015). Este método pode ser utilizado para avaliar a estabilidade de controladores de GPC mesmo quando $hc>1$ para algumas plantas, e também para mostrar graficamente que o sistema vai se tornando mais robusto conforme o valor de hp aumenta, como é fundamentado na teoria de GPC.

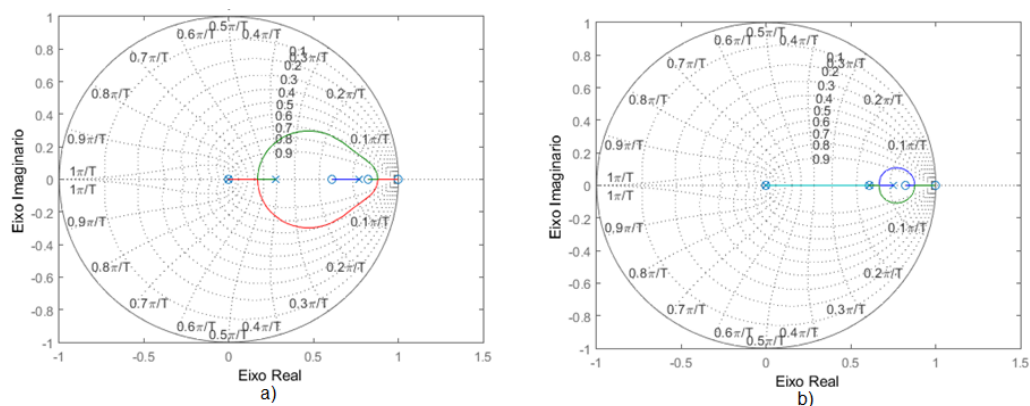
Com isso, a primeira informação que o usuário deve decidir é com relação ao hp , e isto deve ser feito ao avaliar os vários lugares das raízes, para cada valor do mesmo. A Figura 2.1 e Figura 2.2 abaixo mostram os lugares das raízes gerados para uma planta de fase não mínima e estável para vários casos de hp 2, 5, 6 e 17 mostrando que o gráfico tende a se concentrar dentro da circunferência de raio 1, a qual se refere a região de estabilidade do sistema discreto.

Figura 2.1: Visualizando de forma gráfica o aumento de robustez do sistema ao se aumentar o valor de hp , a) $hp = 2$, b) $hp = 5$



Fonte: próprio autor 2017.

Figura 2.2: Visualizando de forma gráfica o aumento de robustez do sistema ao se aumentar o valor de hp , a) $hp = 6$, b) $hp = 17$



Fonte: próprio autor 2017.

Para esta planta, nos casos de $hp=2$ até $hp=5$, o sistema em malha fechada é sempre instável para qualquer valor de $\frac{\lambda}{\delta}$. A partir de $hp=6$ o sistema se torna estável para qualquer valor de $\frac{\lambda}{\delta}$. Vale lembrar que, para cada planta e tempo de amostragem o formato do lugar das raízes muda e, assim, deve ser sempre avaliado de forma individual além do que, quanto maior o valor de hp , maior capacidade de processamento é exigida do sistema, dificultando a implementação da Equação (21).

Dessa forma, visualiza-se graficamente qual o menor valor de hp e quais valores de $\frac{\lambda}{\delta}$ para se gerar sistema estáveis em malha fechada. Caso se utilize algum algoritmo de busca para encontrar o melhor caso, basta escolher um intervalo de $\frac{\lambda}{\delta}$.

No presente trabalho foi utilizado o AG como algoritmo de busca para encontrar o melhor caso. Desta forma, o intervalo de $\frac{\lambda}{\delta}$ define o tempo de processamento de busca do mesmo.

2.2.3 Algoritmo Genético

Algoritmo Genético (AG) é um algoritmo de busca inspirado na teoria de seleção natural de Darwin e na genética de Mendel (ALMEIDA e SALLES, 2016).

O AG é composto por genes, indivíduo, função *fitness*, população, geração, método de seleção, técnica Elitista, operações genéticas e taxa de operadores genéticos. Os genes são as variáveis do problema. O indivíduo é qualificado pela função *fitness* e é composto por genes. A população é composta de indivíduos, sendo a primeira geração composta pela primeira população. Já a segunda geração em diante é composta por novos indivíduos resultantes das operações genéticas com os indivíduos selecionados da geração passada.

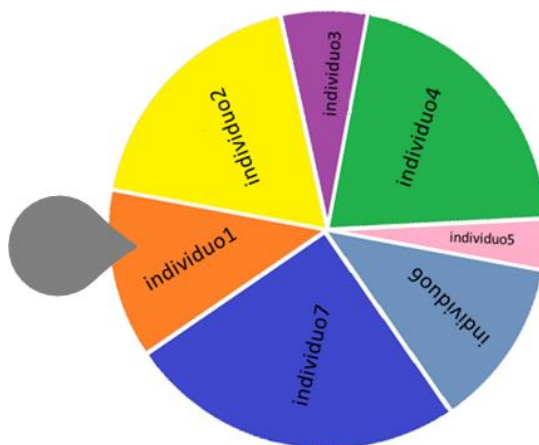
Na literatura de AG encontram-se os métodos de seleção por roleta, por torneio, o ordenamento, biclassista e por sorteio Universal (ALMEIDA e SALLES,

2016). Estes métodos de seleção são apenas para escolher quais indivíduos vão trocar informação genética, ou seja, os indivíduos que vão gerar herdeiros.

O método de Seleção por Roleta é o mais clássico e foi o utilizado neste trabalho. Este método seleciona os indivíduos de forma aleatória ao se girar a roleta que contém os indivíduos. O detalhe é o espaço que cada indivíduo ocupa nesta roleta, que depende da sua capacidade de solucionar o problema. No contexto de AG esta capacidade se chama função *fitness*, com isso, os melhores indivíduos tem maior probabilidade de serem selecionados para gerar descendentes, como mostra a Figura 2.3. Nos contextos teóricos de AG e de programação, pode-se garantir que os melhores indivíduos continuaram nas próximas gerações e isso é denominada Técnica Elitista, a qual também foi utilizada no presente trabalho.

A capacidade do indivíduo de resolver o problema depende de seus genes (MAN, TANG e KWONG, 1999). Na natureza, o indivíduo considerado o melhor, é o que tem o maior valor nos genes que determinam a dominancia. No contexto do GPC, os genes dos indivíduos são os parâmetros do controlador.

Figura 2.3: Forma da roleta para o método seleção por roleta



Fonte: próprio autor, 2018

A partir dos indivíduos selecionados para gerar herdeiros resta saber como serão gerados os genes desses herdeiros, que de acordo com a genética de Mendel existem as operações genéticas (ALMEIDA e SALLES, 2016): *Crossover* (recombinação) que consiste na troca aleatória de genes entre os indivíduos; *Mutação* que consiste em um gene se alterar aleatoriamente; *Reprodução*

(clonagem) que consiste em copiar um indivíduo para próxima geração, que no contexto de programação é o mesmo que o que a técnica elitista faz.

Com relação ao ambiente de programação pode-se escolher a primeira população, tamanho da população, quantidade de gerações e probabilidade de mutação e probabilidade de *crossover*. De acordo com Almeida e Salles (2016), quando existe algum conhecimento sobre a solução do problema é interessante que os indivíduos da população inicial sejam representados por esses conhecimentos heurísticos. Assim, diminuirá o tamanho da população e a quantidade de gerações necessárias para se encontrar o melhor caso e conseqüentemente diminuirá o tempo de processamento demandado.

No Apêndice A são explicadas as funções do código do AG utilizado neste trabalho.

2.2.4 Sintonia de λ e δ usando AG

Os indivíduos do AG serão compostos pelos genes λ e δ e a extensão do intervalo escolhido é que determinará o tamanho da população para que o AG encontre um bom resultado.

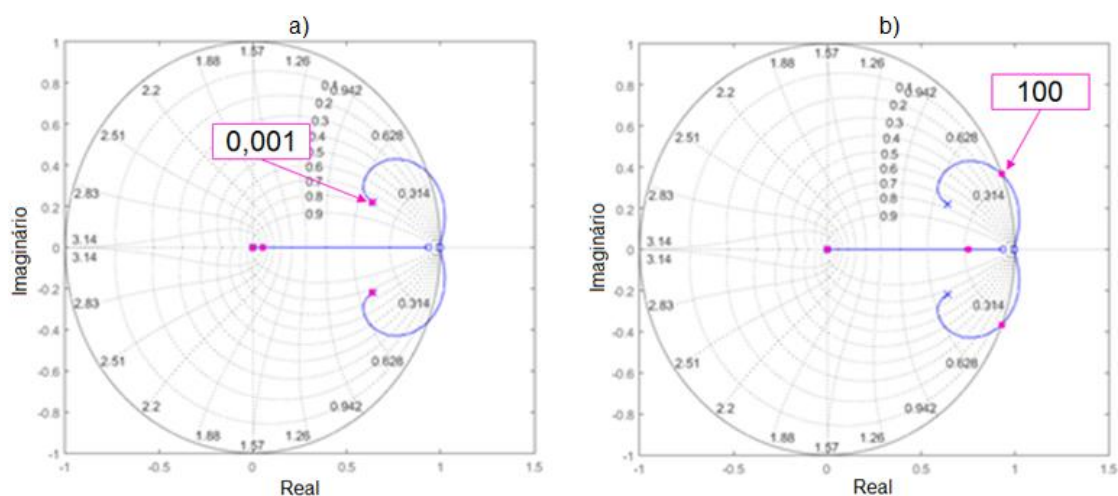
Neste trabalho os parâmetros do GPC que serão buscados pelo AG são apenas o λ e o δ pois, o valor de hp é fixado ao se avaliar o lugar das raízes e $hc=1$. A função *fitness* escolhida é a Integral do Erro Quadrático (*Integral Squared Error – ISE*), é importante ressaltar que foram incluídas manipulações no código da rotina 4, explicada no Apêndice A, de forma a avaliar também os valores de sobressinal e máxima variação do sinal de controle. Assim sendo, se estes valores ultrapassarem os desejados pelo usuário, o *fitness* é alterado para valores que garantem que estes indivíduos sejam descartados na etapa de seleção do AG. Além disto, foi definido que quanto menor o valor do *fitness* melhor classificado será o indivíduo.

Como mencionado por Almeida e Salles (2016) inserir conhecimento sobre o problema na criação dos indivíduos da população inicial reduz o tempo necessário de processamento do AG. Neste caso, o conhecimento é a região de

λ e δ escolhida, um exemplo é mostrado no Apêndice C. Neste trabalho foi desenvolvida uma interface gráfica com o intuito de facilitar e agilizar este processo (Apêndice B). A Figura 2.4 mostra um exemplo de escolha da região de $\frac{\lambda}{\delta}$.

Figura 2.4: Gráfico do Lugar das Raízes de função discreta para escolha do intervalo de $\frac{\lambda}{\delta}$

, a) valor mínimo de $\frac{\lambda}{\delta}$, b) valor máximo de $\frac{\lambda}{\delta}$



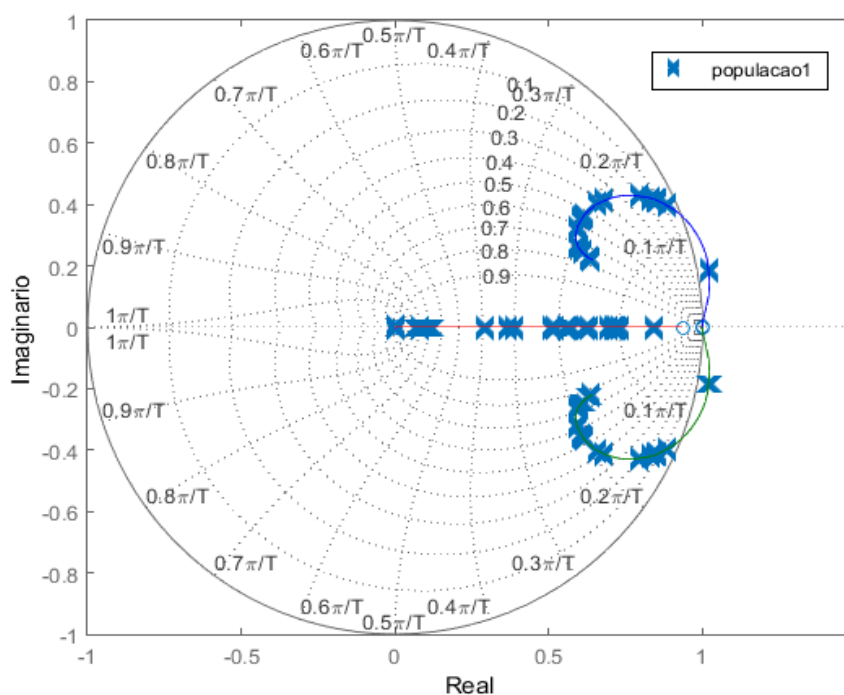
Fonte: próprio autor, 2017.

Além disso, a lógica de criação dos indivíduos da primeira população também reduz o tempo necessário de processamento do AG, desde que a população inicial contenha indivíduos melhores distribuídos. A forma como se criou os indivíduos da primeira população neste trabalho foi por pequenos intervalos de casas decimais, como por exemplo, se o intervalo escolhido for de 0,001 até 100, o algoritmo cria a parte dos indivíduos em cada casa decimal, e entre esse intervalo tem-se 10^{-3} e 10^2 , o que significa 5 casas decimais. Desta forma, parte dos indivíduos são criados entre 10^{-3} e 10^{-2} , outra parte criados entre 10^{-2} e 10^{-1} , e assim sucessivamente até finalmente serem criados entre 10^1 e 10^2 . Com isso as próximas gerações terão indivíduos melhores distribuídos ajudando a encontrar a melhor resposta de forma mais rápida. Como mostra a Figura 2.5.

Os indivíduos criados fora da faixa desejada são em consequência da divisão do tamanho da população pela quantidade de casas decimais que não resultam em número inteiro, fazendo assim com que os indivíduos restantes sejam criados também fora da faixa. Porém, são poucos os indivíduos criados fora da faixa desejada, e são descartados nas gerações futuras.

Figura 2.5: Gráfico do Lugar das Raízes da função discreta mostrando a distribuição da primeira população para 50 indivíduos para o caso dos indivíduos criados utilizando a lógica de

pequenos intervalos da região de $\frac{\lambda}{\delta}$ escolhida

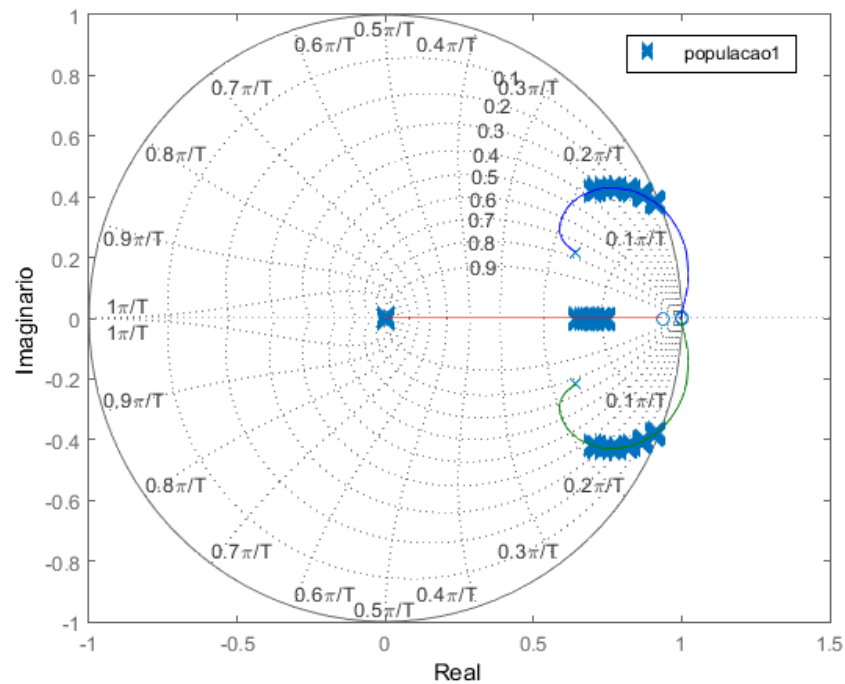


Fonte: próprio autor, 2017.

Na forma de um contra-exemplo, a Figura 2.6, que ilustra como os indivíduos da população inicial ficam distribuídos para o caso de uma lógica simples no formato $a + \text{rand}(1,1)b$, sendo $\text{rand}(1,1)$ um valor aleatório entre zero (0) e um (1), e a e b os valores mínimo e máximo do intervalo de $\frac{\lambda}{\delta}$, respectivamente.

Comparando a Figura 2.5 com a Figura 2.6, fica claro que os indivíduos da primeira população criados pela lógica de pequenos intervalos estão melhores distribuídos na região escolhida.

Figura 2.6: Gráfico do Lugar das Raízes da função discreta mostrando a distribuição da primeira população para 50 indivíduos para o caso dos indivíduos criados utilizando a lógica simples $a + \text{rand}(1,1)b$



Fonte: próprio autor, 2018.

3 Controle Adaptativo

A classe de controladores adaptativos pode controlar *online* sistemas desconhecidos usando dados medidos em tempo real ao longo da trajetória do sistema. Enquanto assimilam as leis de controle, eles são capazes de garantir a estabilidade e a eficiência do sistema. Os controladores adaptativos usam técnicas de Identificação de Sistemas que, conforme Aguirre (2007) e Coelho (2006), servem para identificar os parâmetros do sistema e, posteriormente, usam o modelo obtido para resolver equações de projeto ótimo. (LEWIS, VRABIL e SYRMOS, 2012)

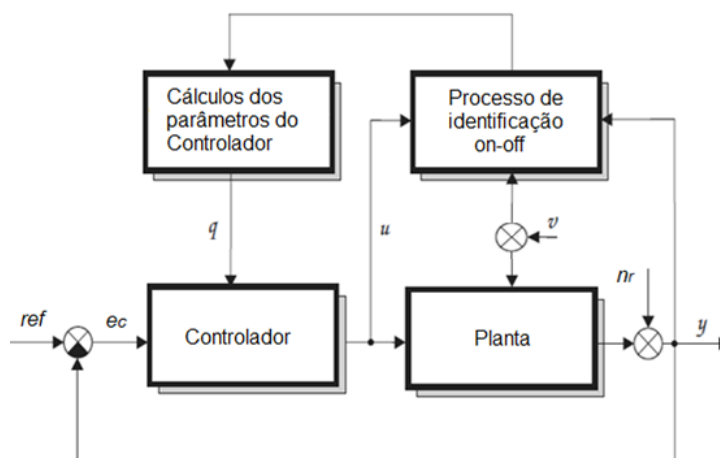
Das várias vertentes de controle adaptativo, uma que mostrou grande potencial e sucesso foi o chamada “*self-tuning controller*” (STC). Também existe o “*auto-tuning controller*” sendo que, essas nomenclaturas se confundem entre si. A diferença do “*self-tuning*” para o “*auto-tuning*” encontra-se na identificação. O “*auto-tuning*” identifica um modelo e o controlador é ajustado apenas quando a identificação se encerra. Já no “*self-tuning*”, a planta é identificada e o controlador é ajustado ambos a cada amostra. (BOBÁL, BÖHM, *et al.*, 2005)

No modo “*auto-tuning*”, uma vez que a identificação é interrompida, o controlador é ajustado e permanece com os parâmetros fixos até que então valores de índices de desempenho sejam violados e, com essa ocorrência, o ciclo é repetido. A Figura 3.1 ilustra o Diagrama de Blocos de um sistema em malha fechada com *auto-tuning*, sendo: a entrada referência *ref* escolhida pelo operador, o erro entre a entrada e a saída e_c , o sinal de controle u , distúrbio no processo v , ruído na saída n_r , a saída y , e os parâmetros do controlador q . (BOBÁL, BÖHM, *et al.*, 2005)

A racionalidade básica por trás do *auto-tuning* e do *self-tuning* é a identificação recursiva, ainda em STC, tem-se, assim, os implícitos e os explícitos. Os STC explícitos usam o modelo do processo adquirido com identificação de sistemas, e os parâmetros do controlador são então calculados dependendo do modelo. Já os STC implícitos, não usam o modelo e sim apenas a entrada e saída do processo para ajustar os parâmetros do controlador. A Figura 3.2 ilustra o Diagrama de Blocos de um sistema em malha fechada com *self-tuning* explícito, onde as novas variáveis são: critério de identificação Q_i ,

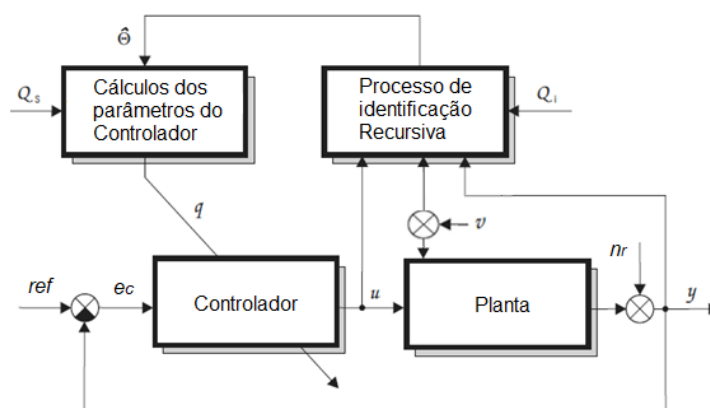
parâmetros do modelo identificado Θ , critério de síntese do controlador Q_s . (BOBÁL, BÖHM, *et al.*, 2005)

Figura 3.1: Diagrama de blocos de *auto-tuning controller*



Fonte: (BOBÁL, BÖHM, *et al.*, 2005).

Figura 3.2: Diagrama de blocos de *self-tuning controller*



Fonte: (BOBÁL, BÖHM, *et al.*, 2005).

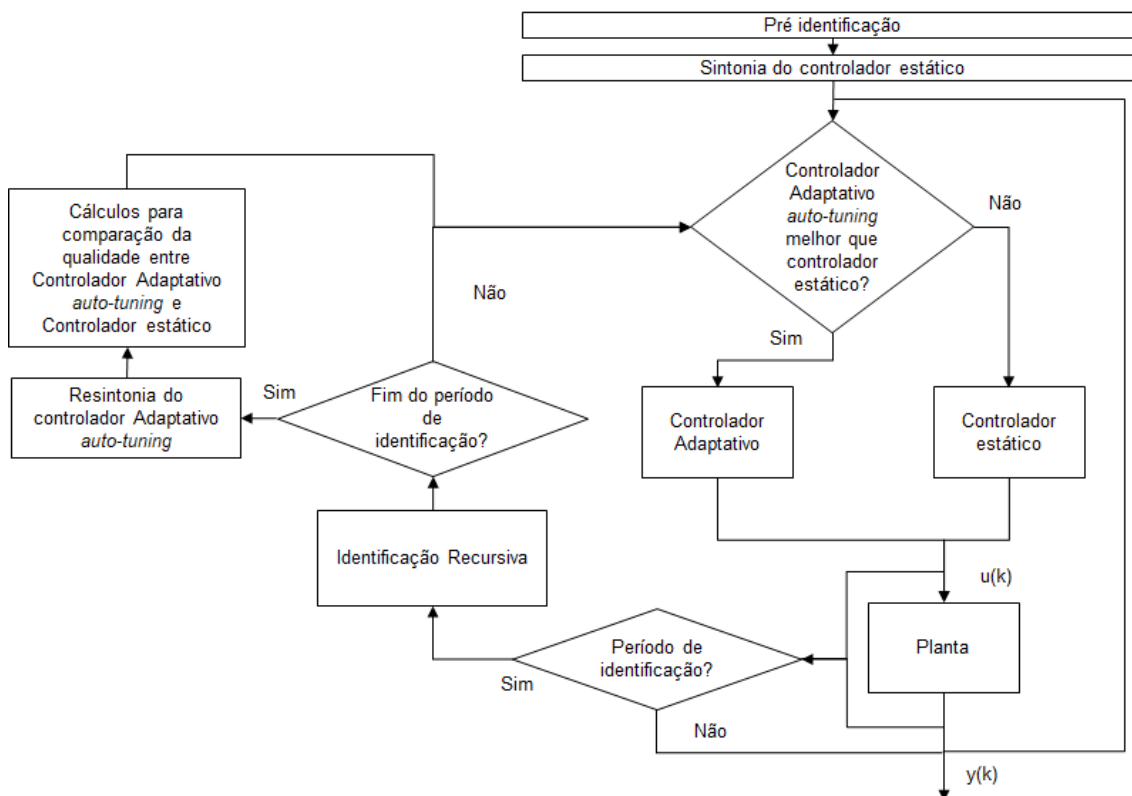
A auto sintonia do controlador depende das regras de controle que, por sua vez, dependem dos parâmetros do modelo identificado do processo a ser controlado. Em Bobál, Böhm, *et al.* (2005) é abordado, entre outros, controlador adaptativo “*self-tuning*” dos controladores Proporcional Integral Derivativo (PID).

As condições necessárias para identificação em controle adaptativo, tanto no modo “*auto-tuning*” quanto no modo “*self-tuning*”, não são ideais (BOBÁL, BÖHM, *et al.*, 2005). Dessa forma, utiliza-se um módulo de supervisão para

garantir que o processo continue estável em caso de falha na identificação e sintonia inadequada do controlador adaptativo (COELHO, 2006). Isso significa ter um controlador estático com parâmetros fixos que garanta a estabilidade do sistema em paralelo com o controlador adaptativo. Desta forma, mesmo que a resposta do sistema com este controlador estático tenha valores de sobressinal e tempo de acomodação fora do desejado, este ainda manterá o sistema estável.

Neste trabalho adotou-se controlador adaptativo modo “*auto-tuning*”. Dessa forma, quando a identificação completar seu ciclo e o controlador adaptativo *auto-tuning* for resintonizado ocorrerá o momento de decisão entre o controlador adaptativo *auto-tuning* e o controlador estático, momento este que será o de comparar a saída real com a saída estimada do controlador adaptativo *auto-tuning*. Se a sintonia do controlador adaptativo *auto-tuning* estiver dentro do valor desejado este será chaveado e o controlador estático ficará desativado. Com isso, o sistema em malha fechada com controlador adaptativo *auto-tuning* e o controlador estático foi construído como mostra a Figura 3.3.

Figura 3.3: Diagrama de blocos para o sistema de Controle *Auto-tuning* utilizado neste trabalho



3.1 Identificação Recursiva

Inicialmente, se deve ter em mente que a identificação não recursiva funciona com uma grande quantidade de dados, sendo realizada com a planta *off-line* enquanto que a identificação recursiva precisa de poucos dados (COELHO, 2006), podendo ser realizada com a planta *on-line*. Porém, para isso é preciso fazer uma formulação adequada para proporcionar procedimentos eficientes em tempo real para a identificação recursiva ser aplicada no controle adaptativo (WELLSTEAD e ZARROP, 1991).

Existem diferentes métodos para estimação de parâmetros de modelos lineares discretos e neste trabalho foi utilizado o método dos Mínimos Quadrados Recursivos (MQR) com fator de esquecimento variando conforme Coelho (2006), Aguirre (2007) e Bobál, Böhm, *et al.* (2005). Isto posto, é importante mencionar que o método MQR é a base para o desenvolvimento de outros métodos de estimação (LJUNG, 1999). A técnica de fator de esquecimento é para garantir que seja identificadas até mesmo pequenas variações nos parâmetros do processo (BOBÁL, BÖHM, *et al.*, 2005).

Deve-se ressaltar que, Coelho (2006) e Bobál, Böhm, *et al.* (2005) apresentam diferentes técnicas para o fator de esquecimento e neste trabalho foi utilizada a técnica proposta por Bobál, Böhm, *et al.* (2005) de fator de esquecimento exponencial. O problema de *Blow-Up* na matriz de covariância, que instabiliza a identificação, ocorre quando o fator de esquecimento é usado de forma inadequada. Isso acontece quando o fator de esquecimento se mantém constante por um longo período (COELHO, 2006). Mesmo usando as técnicas de fator de esquecimento variáveis, esse problema ainda ocorre quando o sistema fica muito tempo em regime. Com isso, neste trabalho, além de se usar o fator de esquecimento variando, para evitar este problema, foi utilizada uma lógica simples de manter o fator de esquecimento abaixo de 0,9. Esta lógica foi utilizada simplesmente no ambiente de programação quando o fator de esquecimento atingia o valor de 0,9, e, sendo utilizada essa lógica forçou a voltar para o valor 0,5.

A qualidade da estimação depende da natureza do ruído, da estrutura do modelo, do tipo de aplicação e da riqueza nas informações contidas nas medidas, conforme afirma Coelho (2006).

Neste trabalho, considera-se a priori que a ordem do modelo é conhecida e as amostras de medidas de entradas e saídas da planta estão disponíveis a cada período de amostragem durante o período de identificação, que foram de 600 amostras após a variação da planta.

O algoritmo de identificação recursiva, proposto por Bobál, Böhm, *et al.* (2005), é composto por: vetor de regressores Φ , como mostra a Equação (29).

$$\Phi(k-1) = \begin{bmatrix} -y(k-1) \\ -y(k-2) \\ \vdots \\ -y(k-n) \\ u(k-1) \\ u(k-2) \\ \vdots \\ u(k-n) \end{bmatrix} \quad (29)$$

sendo $y(k-1), \dots, y(k-n)$ e $u(k-1), \dots, u(k-n)$ as saídas e as entradas passadas da planta, respectivamente.

O erro de previsão $\hat{e}(k)$, como mostra a Equação (30).

$$\hat{e}(k) = y(k) - \theta\Phi \quad (30)$$

sendo $y(k)$ a saída atual da planta, o vetor de parâmetros identificados θ e o vetor de regressores Φ .

A variável auxiliar ζ , como mostra a Equação (31).

$$\zeta = \Phi^T C \Phi \quad (31)$$

sendo o vetor de regressores Φ e a Matriz de Covariância C .

A atualização do vetor de parâmetros identificados θ , Como mostra a Equação (32).

$$\theta = \theta + \frac{\mathbf{C}\Phi}{1+\zeta} \hat{e}(k) \quad (32)$$

sendo o vetor de parâmetros identificados $\theta = [a_1 \ a_2 \ \dots \ a_n \ b_1 \ b_2 \ \dots \ b_n]$, a_1, \dots, a_n e b_1, \dots, b_n os parâmetros identificados dos polinômios $A(q^{-1})$ e $B(q^{-1})$, a Matriz de Covariância \mathbf{C} , a variável auxiliar ζ e o erro de previsão $\hat{e}(k)$.

A variável auxiliar ε , como mostra a Equação (33).

$$\varepsilon = \varphi - \frac{(1-\varphi)}{\zeta} \quad (33)$$

sendo o fator de esquecimento φ e a variável auxiliar ζ .

A atualização da Matriz de Covariância \mathbf{C} , como mostra a Equação (34).

$$\mathbf{C} = \mathbf{C} - \left(\frac{\mathbf{C}\Phi\Phi^T\mathbf{C}}{\varepsilon^{-1} + \zeta} \right) \quad (34)$$

sendo o vetor de regressores Φ , o vetor de regressores transposto Φ^T , a variável auxiliar ε e a variável auxiliar ζ .

O cálculo das variáveis auxiliares μ , ν e η , como mostram as Equações (35), (36) e (37).

$$\mu = \varphi \left(\mu + \frac{\hat{e}(k)^2}{(1+\zeta)} \right) \quad (35)$$

sendo o fator de esquecimento φ , o erro de previsão $\hat{e}(k)$ e a variável auxiliar ζ .

$$\nu = \varphi(\nu+1) \quad (36)$$

sendo o fator de esquecimento φ .

$$\eta = \frac{\hat{e}(k)^2}{\mu} \quad (37)$$

sendo o erro de previsão $\hat{e}(k)$ e a variável auxiliar μ .

A atualização do fator de esquecimento φ , como mostra a Equação (38).

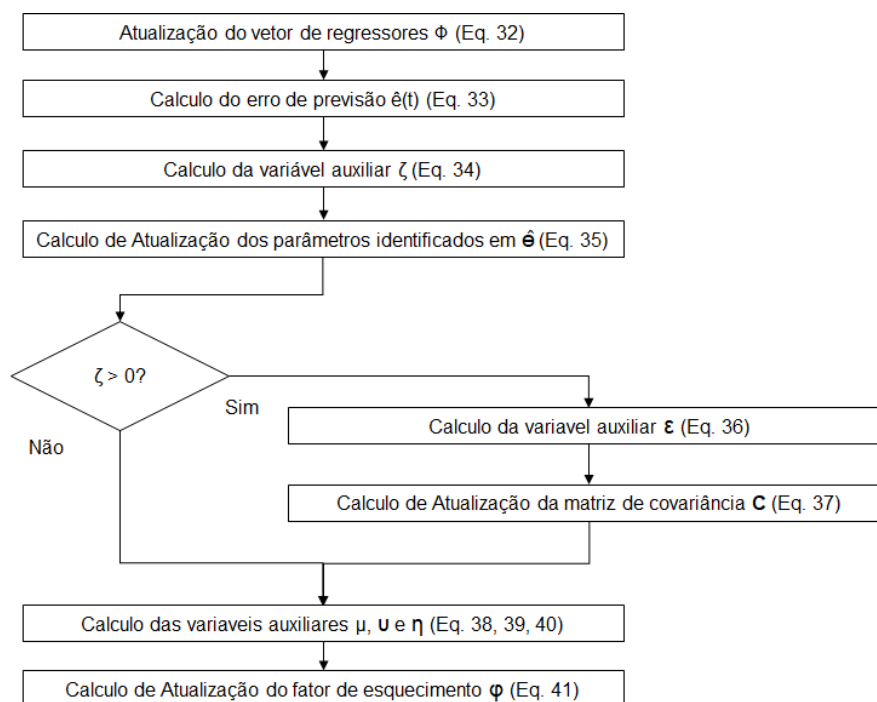
$$\varphi = \frac{1}{1+(1+\rho) * \ln \left(1 + \zeta - \frac{\zeta}{(1+\zeta)} + \zeta(\nu+1) \frac{\eta}{1+\zeta+\eta} \right)} \quad (38)$$

sendo as variáveis auxiliares ρ , ζ , ν , e η .

Desta forma, durante o período de identificação são atualizados o vetor de regressores, o erro de previsão, também chamado de erro de modelagem, que mede a qualidade do modelo estimado, a matriz de covariância e o fator de esquecimento a cada amostra. A Figura 3.4 exibe o diagrama de blocos do algoritmo de identificação recursiva utilizado neste trabalho.

Uma das premissas necessárias a se assumir para o método mínimos quadrados é que a ordem de $A(q^{-1})$ (na) e a ordem de $B(q^{-1})$ (nb) sejam iguais ($na=nb=n$) durante o período de identificação, conforme menciona Bobal, Böhm, et al. (2005).

Figura 3.4: Diagrama de blocos do algoritmo de identificação recursiva utilizado neste trabalho



Fonte: próprio autor, 2017.

Completando o t3pico de identifica33o recursiva, deve-se especificar que na inicializa33o da identifica33o recursiva foram utilizados os parâmetros pr3e-identificados em θ . Na matriz de covariânci3a \mathbf{C} escolheu-se uma matriz diagonal com valores de 10^3 e o fator de esquecimento φ igual a 1. Al3e3m disto, as variáveis auxiliares para a atualiza33o do fator de esquecimento ρ , η , μ e ν foram iniciadas com os valores 0,99, 1, 0,001 e 0,000001, respectivamente, como proposto por Bobál, B3ohm, *et al.* (2005).

3.2 PID Self-tuning e Auto-tuning

A literatura de controle j3a est3a bem constru3ida no tema de regras de sintonia para controladores PID.

Desta forma, criou-se algoritmos para PID *auto-tuning* baseado em proposto por Bobál, B3ohm, *et al.* (2005) que sugere algoritmos e c3odigos para PID *self-tuning*.

Sobre as regras de sintonia do PID foram usadas, para sistemas de primeira ordem, a regra de sintonia CHR (Chein, Hrones e Reswick) de respostas mais r3apidas e de 20% sobressinal. Sobre a equa33o do sinal de controle $u(k)$ foi utilizada a estrutura padr3o do PID em paralelo discreta como mostra a Equa33o (39).

$$u(k) = K_p \left\{ e_r(k) + \frac{T_0}{T_i} \sum_{i=1}^k e_r(i-1) + \frac{T_d}{T_0} (e_r(k) - e_r(k-1)) \right\}$$

$$\text{onde } K_p = \frac{0,95\tau}{K\theta}, T_i = 1,357\tau \text{ e } T_d = 0,473\theta. \quad (39)$$

sendo, K_p o ganho proporcional do controlador PID, T_i e T_d os tempos integral e derivativo e T_0 o tempo de amostragem. Os parâmetros τ , θ e K s3o3o a constante de tempo, o atraso de transporte e o ganho da planta, respectivamente. Quanto aos vetores, $u(k)$ 3e3o3o o sinal de controle e $e_r(k)$ 3e3o3o o erro de realimenta33o entre a refer3e3ncia e a sa3ida.

Para sistemas de segunda ordem foi utilizada a regra sintonia de Ziegler e Nichols modificada na estrutura de Takahashi como proposto por Bobál, Böhm, *et al.* (2005). A estrutura de Takahashi define a equação do sinal de controle discreta $u(k)$ como mostra a Equação (40).

$$u(k) = k_r (-y(k) + y(k-1)) + k_i (r(k) - y(k)) + k_d (2y(k-1) - y(k) - y(k-2)) + u(k-1) \quad (40)$$

onde $k_i = \frac{1,2K_{pu}T_0}{T_u}$, $k_r = 0,6K_{pu} - \frac{k_i}{2}$ e $k_d = 3K_{pu}T_u \frac{3K_{pu}T_u}{40T_0}$

sendo, k_r , k_i e k_d os ganhos proporcional, integral e derivativo do controlador PID, T_0 o tempo de amostragem, K_{pu} o ganho crítico proporcional e T_u o período crítico de oscilações em malha fechada. Em relação aos vetores, $u(k)$ é o sinal de controle, $y(k)$ a saída do sistema e $r(k)$ a referência do sistema.

3.3 GPC adaptativo

Na literatura encontra-se o GPC adaptativo usando o método LQG (Linear Quadratic Gaussian) para sintonizar os parâmetros do controlador, como se baseia Bitmead, Gevers e Wertz (1990). Já Kanjilal (1995) desenvolve o GPC Adaptativo mas, somente para o caso de hp com valores elevados.

A proposta deste trabalho foi adotar o GPC com pequenos valores hp , $hc=1$ e com uma pré-análise do sistema avaliando Lugar das Raízes para simplificar o projeto no que se refere a análise de estabilidade, facilidade de projetar e diminuir o custo de processamento. Assim, propõe-se duas formas de resintonizar o controlador de forma adaptativa: uma com o auxílio do AG e outra com regras sintonia geradas pelo próprio autor.

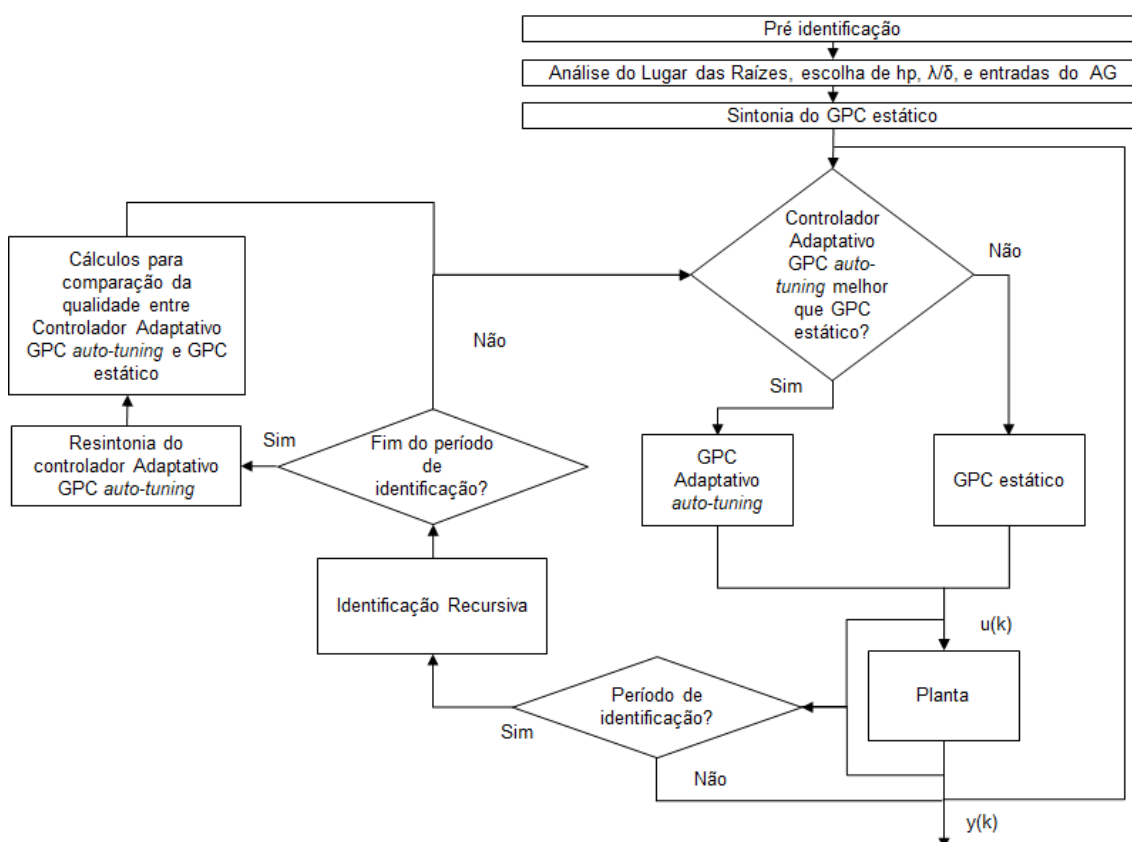
3.3.1 Com *auto-tuning* usando AG

A sintonia do GPC utilizando AG de forma adaptativa pode ser utilizada para os sistemas de qualquer ordem, desde que os parâmetros do modelo sejam identificados corretamente. Como neste trabalho foram fixados hp, hc e

analisado um intervalo de busca dos parâmetros λ e δ , o tamanho da população do AG pôde ser menor e com isso o tempo de processamento necessário diminuiu para aproximadamente 9 a 11 segundos. A sintonia do controlador é feita apenas uma vez após o período de identificação e este tempo fica incluso no período de identificação do controlador adaptativo GPC *auto-tuning* sendo que, a estabilidade é garantida pelo controlador estático como mostrado na seção 2.2.4.

Com isso, neste trabalho, o método de controlador adaptativo GPC *auto-tuning* utilizando AG depende da pré-análise do usuário na fase de pré-identificação para a escolha desejada dos parâmetros h_p e o intervalo $\frac{\lambda}{\delta}$ sendo que, o AG deve procurar o melhor caso. Quanto maior for este intervalo maior deverá ser o tamanho da população, o que também deve ser analisado durante a fase de pré-identificação, como mostra a Figura 3.5.

Figura 3.5: Diagrama de blocos para o sistema de Controle Adaptativo *auto-tuning* com controlador GPC utilizando AG na resintonia



3.3.2 Com Regra de Sintonia gerada pelo próprio autor

Foram gerados dados, utilizando a interface gráfica (Apêndice B), afim de se criar uma regra de sintonia dos parâmetros $\frac{\lambda}{\delta}$ do GPC em relação aos parâmetros de uma planta de primeira ordem no formato $G(s) = \frac{K}{\tau s + 1} e^{-\theta s}$. Variou-se o ganho K de 1 a 15, e a constante de tempo τ de 25 a 35 ambos com o passo de 1 em 1, totalizando de 165 testes. O parâmetro θ foi fixado em 7,5 [s]. Tais dados foram gerados ao se buscar o caso em que se obtivesse $OV < 10\%$ e $\max(|\Delta u(k)|) < 0,2$.

Aproximou-se este conjunto de dados em uma função multivariável $\frac{\lambda}{\delta}(K, \tau)$. Porém, após experimentar, obteve-se melhores resultados ao se usar várias funções de $\frac{\lambda}{\delta}(K, \tau)$ separadas por intervalo de K , como mostra a Equação (41).

$$\frac{\lambda}{\delta}(K, \tau) = \begin{cases} 1,2151K^2 - 0,0422K\tau + 1,9583K + 0,0207\tau^2 - 1,1978\tau + 16,6278 & \text{para } K \in [1,3] \\ 3,4663K^2 - 0,0411K\tau - 17,8189K + 0,0081\tau^2 + 0,0529\tau + 32,7343 & \text{para } K \in (3,7] \\ 8,3970K^2 - 1,8133K\tau - 53,4854K + 0,1305\tau^2 + 5,9500\tau + 112,1659 & \text{para } K \in (7,10] \\ 1,0716K^2 + 1,1103K\tau - 27,5679K + 0,4310\tau^2 - 35,6133\tau + 645,0560 & \text{para } K \in (10,15] \end{cases} \quad (41)$$

3.4 Sintonia dos parâmetros λ e δ GPC utilizando AG

A sintonia dos parâmetros λ e δ utilizando AG depende previamente da análise do lugar das raízes para $hc=1$ pois, ao visualizar o lugar das raízes o usuário decidirá o valor de hp como explicado na subseção 2.2.2.

Após decidido o valor de hp , é escolhida uma faixa de $\frac{\lambda}{\delta}$ a ser utilizada na criação dos indivíduos da primeira população do AG, como mencionado na seção 2.2.3.

Neste trabalho, foram inseridos no código do AG manipulações do *fitness* dos indivíduos que gerassem sistemas com valor de sobressinal e variação do sinal de controle acima do desejado de forma que estes fossem descartados, como mencionado na seção 2.2.4. Isto posto, a lógica para garantir esse descarte foi uma simples condição “se” da forma: se o valor de sobressinal ou variação do sinal de controle ultrapassarem o valor desejado, o *fitness* é multiplicado por 10^6 .

3.4.1 Sintonia de GPC estático

A sintonia do GPC estático é o projeto de apenas um controlador GPC para uma dada planta. O AG vai apenas buscar os valores de λ e δ para uma planta escolhida pelo usuário.

Nesse caso, é apenas como uma forma de encontrar o melhor caso para um projeto, sendo considerado um ótimo tipo de controle.

3.4.2 AG na sintonia de GPC Adaptativo

Ao se utilizar o AG na sintonia de GPC adaptativo, o AG vai buscar novos valores de λ e δ para cada vez que se variar a planta. A vantagem principal desde método é que pode ser utilizado para qualquer planta, pois o AG vai buscar os valores dos parâmetros para sintonizar qualquer caso. Porém, deve-se atentar ao tempo de processamento que o AG exige, o que pode ser inviável para implementar, mas, ainda assim, pode-se visualizar seus resultados em simulações.

A sintonia do controlador GPC utilizando o AG ocorre apenas quando o sistema foi reidentificado, após todas as amostras de identificação, sendo que, neste trabalho adotou-se como 600 amostras. Isso significa que o AG vai trabalhar apenas quando a identificação estiver finalizada, o que significa um modo *auto-tuning* de controladores adaptativos.

3.4.3 Regra de Sintonia na sintonia de GPC Adaptativo

As Regras de Sintonia geradas na seção 3.3.2 serão utilizadas nas simulações de GPC adaptativo para o caso de uma planta de primeira ordem. O detalhe deste método é que a qualidade da Regra de Sintonia dependerá da quantidade de dados utilizados ao se aproximar a função, que neste trabalho foi uma pequena faixa de K e τ .

4 Resultados

Neste capítulo serão apresentadas a análise dos resultados obtidos por meio de simulações no *software* MatLab, fazendo uso das plantas apresentadas na Seção 4.1.

Na Seção 4.2 serão apresentadas as plantas utilizadas para as simulações da Identificação Recursiva implementadas tanto em malha aberta, quanto em malha fechada, afim de se demonstrar a necessidade de uma regra de sintonia eficiente. Nas Seções 4.3, 4.4 e 4.5 serão realizadas as respectivas análises dos testes realizados dos controladores PID e GPC adaptativos para as plantas $G_1(s)$ e $G_2(s)$. Uma análise semelhante será feita para as plantas $G_3(s)$ e $G_4(s)$, nas Seção 4.6. Além disto, com o objetivo de reforçar a eficácia do método proposto por este trabalho, na Seção 4.7 serão testadas duas novas plantas, $G_5(s)$ e $G_6(s)$.

4.1 Plantas e Simulações

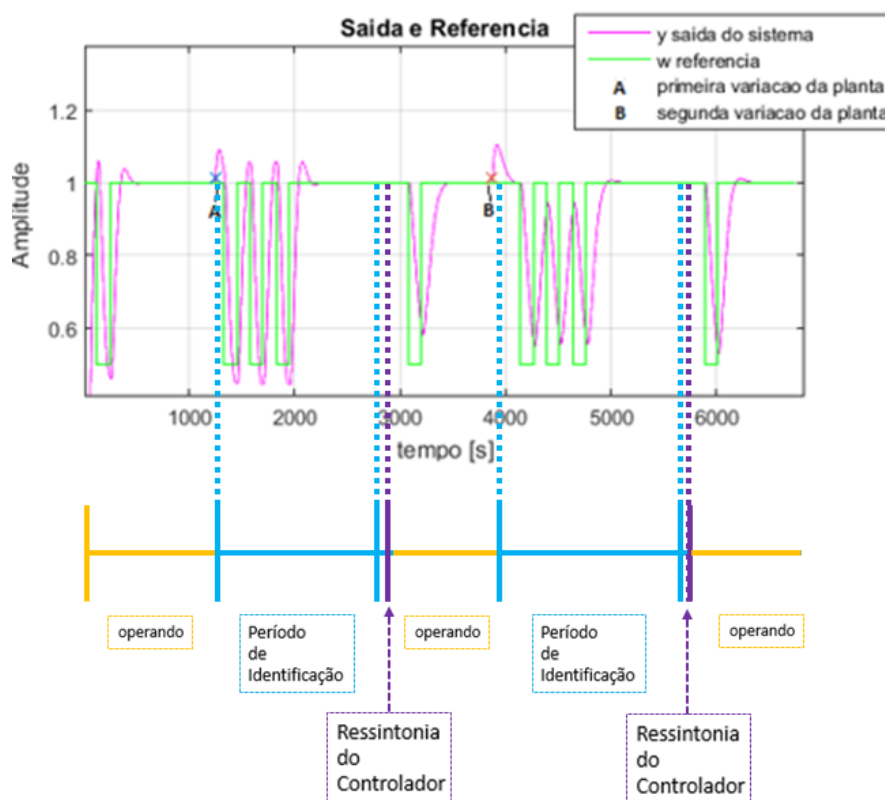
Para fins comparativos, serão propostas simulações para controladores PID adaptativos *auto-tuning* utilizando regras de CHR (Chein, Hrones e Reswick), para as plantas de primeira ordem; controladores PID adaptativos *auto-tuning* utilizando a regra de sintonia de Ziegler-Nichols modificado na estrutura de Takahashi baseado no que foi proposto por Bobál et al. (2005), para plantas de segunda ordem; controladores GPC adaptativos *auto-tuning*, utilizando AG na sintonia para plantas de qualquer ordem; e controladores GPC adaptativos *auto-tuning*, utilizando regra de sintonia desenvolvida neste estudo para plantas de primeira ordem.

Para as simulações, a planta e o seu respectivo controlador foram previamente identificados e sintonizados, de maneira a iniciar o sistema em operação normal. A partir de então, ocorre uma mudança de *setpoint* na entrada na forma de um ciclo de onda quadrada com período de 50 amostras e amplitude 0,5 para mostrar que, de fato, o sistema está bem sintonizado.

Após 500 amostras, ocorre a primeira variação na planta e, 25 amostras depois, é novamente inserida uma mudança no *setpoint*. Esta mudança é dada por quatro ciclos, atuando com as mesmas características da primeira variação. Ao todo, a Identificação Recursiva atua por 600 amostras, definido experimentalmente como o tempo necessário para os parâmetros sejam identificados.

Após a identificação ocorre uma nova sintonia do controlador e, 100 amostras depois, acontece uma nova mudança de *setpoint* na entrada, com o intuito de mostrar como o sistema está respondendo à esta nova sintonia. Depois de 500 amostras, ocorre uma segunda variação na planta e então repete-se os passos da identificação e sintonia do controlador descritos anteriormente (ver Figura 4.1).

Figura 4.1: Gráfico mostrando o padrão das simulações deste trabalho



Fonte: O próprio autor, 2017

Além disto, é importante ressaltar que nem todos as simulações utilizam a identificação recursiva, mas são considerados os parâmetros teóricos da planta, dado que a identificação recursiva em malha fechada diverge dos parâmetros reais da planta.

No Quadro 1, são apresentadas as plantas $G_1(s)$, $G_2(s)$, $G_3(s)$ e $G_4(s)$, seguidas das suas versões discretas ($G_1(z^{-1})$, $G_2(z^{-1})$, $G_3(z^{-1})$ e $G_4(z^{-1})$) e seus respectivos períodos de amostragem. Neste quadro também são evidenciadas as variações de cada uma das plantas utilizadas durante as simulações.

Quadro 1 – Características das Plantas $G_1(s)$, $G_2(s)$, $G_3(s)$ e $G_4(s)$, utilizadas nas simulações.

Planta G_1		
Função de Transferência	Contínua	$G_1(s) = \frac{2}{25s+1} e^{-7,5s}$
	Discreta	$G_1(z^{-1}) = z^{-4} \frac{0,1903}{1-0,9048z^{-1}}$
Período de Amostragem		2,5 segundos
1ª Variação da Planta	Contínua	$G_{1Mod1}(s) = \frac{2,3}{26,25s+1} e^{-7,5s}$
	Discreta	$G_{1Mod1}(z^{-1}) = z^{-4} \frac{0,2089}{1-0,9092z^{-1}}$
2ª Variação da Planta	Contínua	$G_{1Mod2}(s) = \frac{2,645}{27,56s+1} e^{-7,5s}$
	Discreta	$G_{1Mod2}(z^{-1}) = z^{-4} \frac{0,2294}{1-0,9133z^{-1}}$

Fonte: O próprio autor, 2017.

Quadro 1 – Características das Plantas $G_1(s)$, $G_2(s)$, $G_3(s)$ e $G_4(s)$, utilizadas nas simulações. (continuação)

Planta G_2		
Função de Transferência	Contínua	$G_2(s) = \frac{-4s+1}{40s^2+14s+1}$
	Discreta	$G_2(z^{-1}) = z^{-1} \frac{-0,1017+0,173z^{-1}}{1-1,425z^{-1}+0,4966z^{-2}}$
Período de Amostragem		2 segundos
1ª Variação da Planta	Contínua	$G_{2Mod1}(s) = \frac{-5,48s+1}{47,2s^2+14s+1}$
	Discreta	$G_{2Mod1}(z^{-1}) = z^{-1} \frac{-0,1378+0,2013z^{-1}}{1-1,489z^{-1}+0,5525z^{-2}}$
2ª Variação da Planta	Contínua	$G_{2Mod2}(s) = \frac{-7,508s+1}{55,7s^2+14s+1}$
	Discreta	$G_{2Mod2}(z^{-1}) = z^{-1} \frac{-0,179+0,2351z^{-1}}{1-1,549z^{-1}+0,6049z^{-2}}$
Planta G_3		
Função de Transferência	Contínua	$G_3(s) = \frac{s+1}{8s^2-2s-1}$
	Discreta	$G_3(z^{-1}) = z^{-1} \frac{0,6624+0,01368z^{-1}}{1-3,325z^{-1}+1,649z^{-2}}$
Período de Amostragem		2 segundos
1ª Variação da Planta	Contínua	$G_{3Mod1}(s) = \frac{1,15s+1}{8,4s^2-2s-1}$
	Discreta	$G_{3Mod1}(z^{-1}) = z^{-1} \frac{0,6716+0,0372z^{-1}}{1-3,244z^{-1}+1,61z^{-2}}$
2ª Variação da Planta	Contínua	$G_{3Mod2}(s) = \frac{1,322s+1}{8,82s^2-2s-1}$
	Discreta	$G_{3Mod2}(z^{-1}) = z^{-1} \frac{0,6845+0,08869z^{-1}}{1-3,17z^{-1}+1,574z^{-2}}$

Fonte: O próprio autor, 2017.

Quadro 1 – Características das Plantas $G_1(s)$, $G_2(s)$, $G_3(s)$ e $G_4(s)$, utilizadas nas simulações. (continuação)

Planta G_4		
Função de Transferência	Contínua	$G_4(s) = \frac{10}{s^3 + 4s^2 + 4s}$
	Discreta	$G_4(z^{-1}) = z^{-1} \frac{1,51 + 5,469z^{-1} + 1,236z^{-2}}{1 - 2,637z^{-1} + 2,308z^{-2} - 0,6703z^{-3}} 10^{-3}$
Período de Amostragem		0,1 segundos
1ª Variação da Planta	Contínua	$G_{4Mod1}(s) = \frac{13,7}{1,18s^3 + 4s^2 + 4s}$
	Discreta	$G_{4Mod1}(z^{-1}) = z^{-1} \frac{1,779 + 6,539z^{-1} + 1,501z^{-2}}{1 - 2,684z^{-1} + 2,396z^{-2} - 0,7125z^{-3}} 10^{-3}$
2ª Variação da Planta	Contínua	$G_{4Mod2}(s) = \frac{18,77}{1,392s^3 + 4s^2 + 4s}$
	Discreta	$G_{4Mod2}(z^{-1}) = z^{-1} \frac{2,091 + 7,786z^{-1} + 1,811z^{-2}}{1 - 2,725z^{-1} + 2,476z^{-2} - 0,7503z^{-3}} 10^{-3}$

Fonte: O próprio autor, 2017.

Para as plantas $G_1(s)$, $G_2(s)$, $G_3(s)$ e $G_4(s)$ a quantidade total de amostras para cada simulação foi de 2700, porém no final de cada simulação converte-se o eixo-x para segundos, multiplicando o total de amostras pelo período de amostragem de cada planta. Assim, os eixos em segundos correspondem a 6750 [s], 5400 [s], 5400 [s] e 270 [s] para as plantas $G_1(s)$, $G_2(s)$, $G_3(s)$ e $G_4(s)$, respectivamente.

4.2 Identificação recursiva em malha aberta e em malha fechada

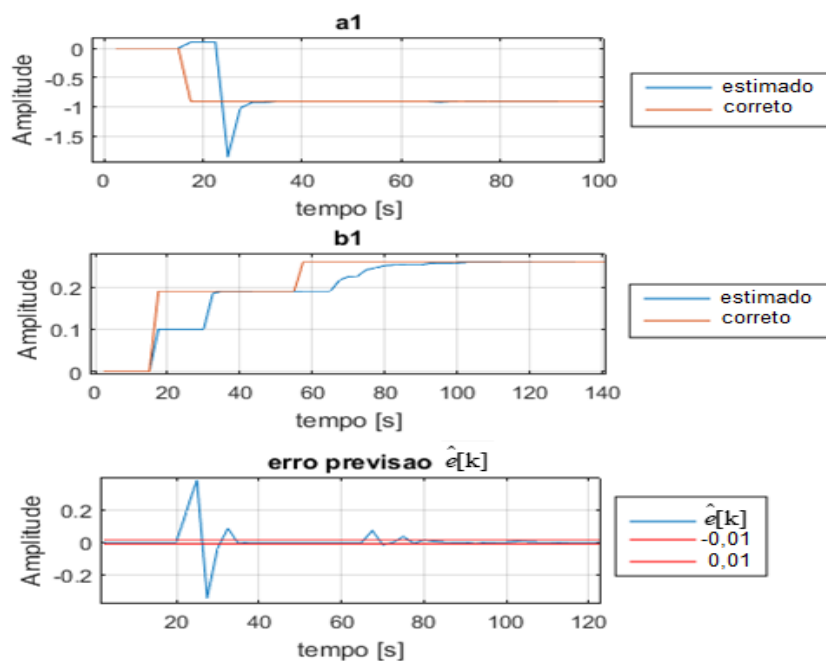
Nesta seção serão apresentados os resultados de identificação dos parâmetros das plantas $G_1(s)$ e $G_2(s)$, quando utilizados a Identificação Recursiva em malha aberta com entrada PRBS (Sinal Binário Pseudo-Aleatório - *Pseudo-Random Binary Signal*) e em malha fechada com PID e entrada sendo o sinal de controle. A implementação da Identificação Recursiva no modo *auto-*

tuning é baseada no código proposto por Bobál, Böhm, *et al.* (2005) de Identificação Recursiva no modo *self-tuning*.

O método de Identificação Recursiva utilizado foi o Mínimos Quadrados Recursivos com fator de esquecimento variável como apresentado na seção 3.1.

Conforme mostra a Figura 4.2, durante as simulações da planta G_1 , o parâmetro a_1 convergiu de maneira rápida, enquanto que para a convergência de b_1 foi necessário um tempo maior. Além disto, para a Identificação Recursiva, os erros de previsão estão, em geral, dentro da faixa de aceitação.

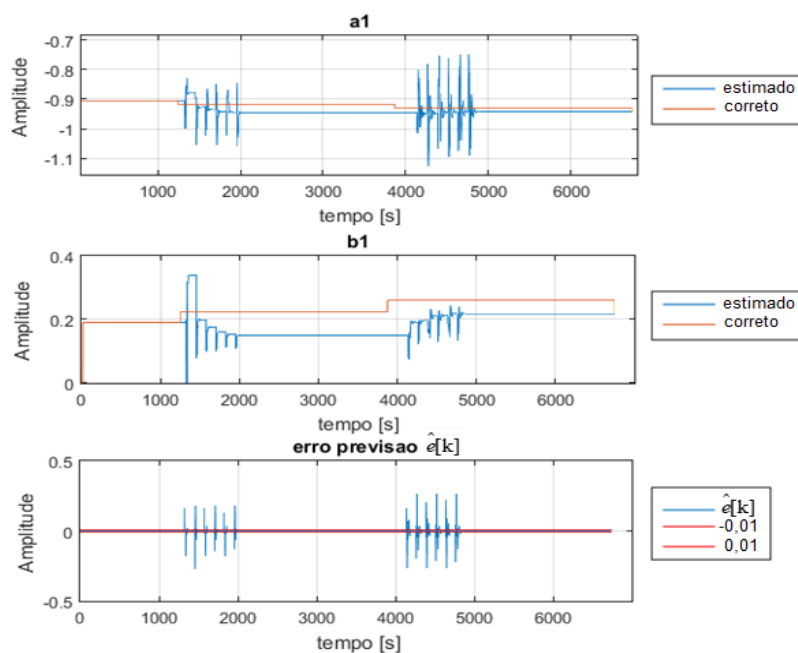
Figura 4.2: Convergência dos parâmetros identificados por Identificação Recursiva em malha aberta com entrada PRBS da planta $G_1(z^{-1})$



Fonte: O próprio autor, 2017.

No caso de malha fechada, a Identificação Recursiva não converge para os valores reais, conforme pode ser visualizado através da Figura 4.3.

Figura 4.3: Convergência dos parâmetros identificados por Identificação Recursiva em malha fechada com entrada sendo sinal de controle da planta $G_1(z^{-1})$

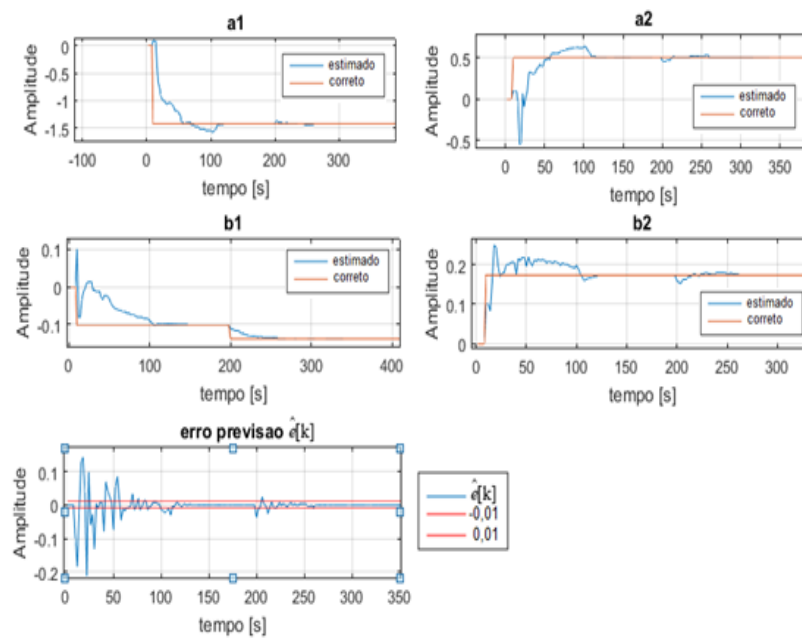


Fonte: O próprio autor, 2017.

Além disso, foi necessário uma quantidade maior de amostras para a simulação.

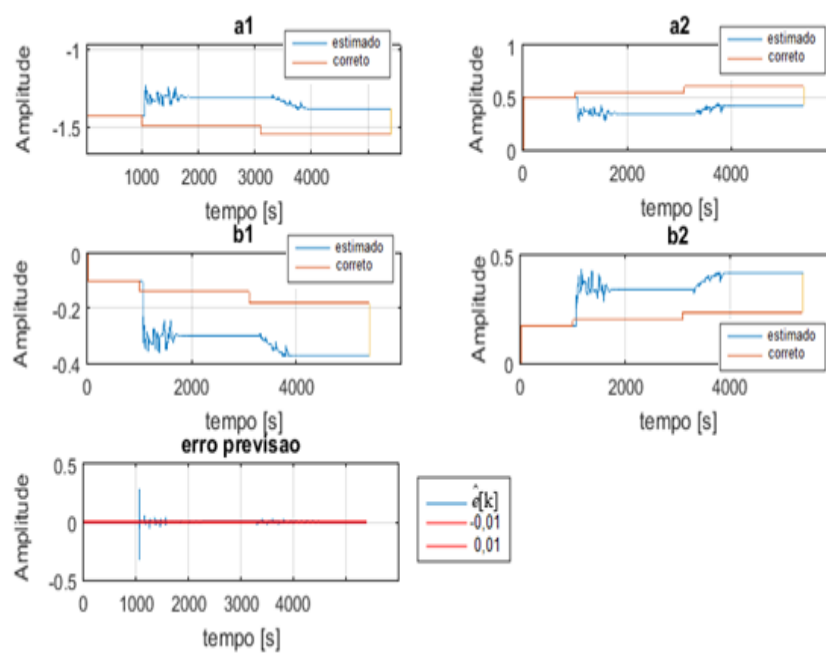
Um comportamento similar ao da planta G_1 ocorre durante as simulações da planta G_2 . No caso em malha aberta, os parâmetros a_1 e a_2 convergem de maneira rápida, enquanto que os demais necessitam de um tempo maior (ver Figura 4.4). Em malha fechada, os parâmetros também não convergem para os valores reais da planta, embora que o erro de previsão permaneça dentro da faixa $[-0,01, +0,01]$ (ver Figura 4.5).

Figura 4.4: Convergência dos parâmetros identificados por Identificação Recursiva em malha aberta com entrada PRBS da planta $G_2(z^{-1})$



Fonte: O próprio autor, 2017.

Figura 4.5: Convergência dos parâmetros identificados por Identificação Recursiva em malha fechada com entrada sendo sinal de controle da planta $G_2(z^{-1})$



Fonte: O próprio autor, 2017.

Portanto, o algoritmo desenvolvido para Identificação Recursiva faz com que os parâmetros da planta convirjam para os valores reais em malha aberta, mantendo o erro de previsão dentro da faixa $[-0,01, +0,01]$, enquanto que, em malha fechada, apenas se mantém o erro de previsão dentro da faixa de aceitação, sem que ocorra a convergência dos parâmetros da planta.

4.3 PID adaptativo

Nesta Seção serão abordadas as simulações referentes aos sistemas implementados através da planta G_1 , com o PID ressintonizado pelas regras de sintonia de CHR para respostas mais rápidas com sobressinal (*Overshoot – OV*) de até 20%, e pela planta G_2 , utilizando a regra de sintonia Ziegler-Nichols na estrutura de Takahasi. Em ambas as plantas serão utilizados um controlador PID paralelo clássico.

4.3.1 Análise da simulação1: planta $G_1(s)$ com PID e identificação Recursiva

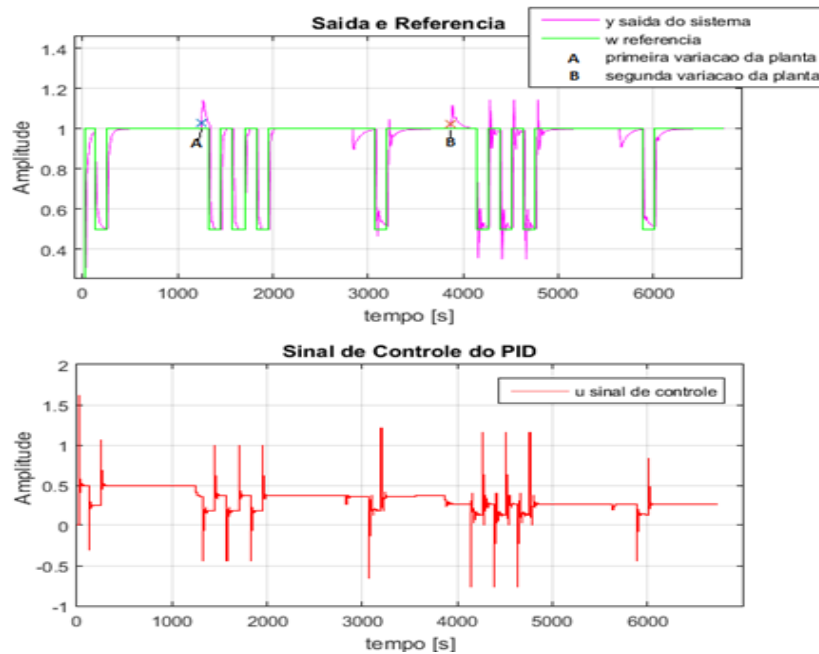
Analisando a Figura 4.6, é possível averiguar que a variação “A” ocorre em 1248 [s]. Os parâmetros da planta foram reidentificados em 2833 [s] e o PID foi ressintonizado em 2835 [s]. A mudança de *setpoint*, responsável por validar a nova sintonia, ocorre em 3085 [s], enquanto que um *OV* de 4,5% acontece em 3223 [s]. No instante de tempo de 3668 [s], a saída tornou-se idêntica à entrada, resultando-se em um tempo de acomodação de 583 [s].

A variação “B” ocorreu em 3873 [s], sendo os parâmetros reidentificados em torno de 5600 [s] e o PID ressintonizado em 5645 [s]. Em 5895 [s] ocorreu uma nova mudança de *setpoint*, sem a ocorrência de *OV*. A saída iguala-se à entrada em 6465 [s], o que significa que um tempo de acomodação de 570 [s].

Embora a Identificação Recursiva não tenha apresentado resultados significativamente bons, foi possível controlar o sistema dentro dos valores

escolhidos de OV e tempo de acomodação (*Settling Time* – t_s). A Tabela 1 ilustra um comparativo entre os parâmetros reais e os identificados da planta $G_1(z^{-1})$.

Figura 4.6: Sinais de Saída, referência e controle da Simulação 1



Fonte: O próprio autor, 2017.

Tabela 1: Parâmetros de $G_1(z^{-1})$ reais e identificados através da Simulação 1

Tempo (s)	Parâmetros			
	a_1	\hat{a}_1	b_1	\hat{b}_1
0	-0,9048	-0,9048	0,1903	0,1903
2833	-0,9187	-0,9455	0,2226	0,1494
5600	-0,9307	-0,942	0,2601	0,2179

Fonte: O próprio autor, 2017

Os parâmetros do Controlador PID foram ressintonizados conforme apresenta a Tabela 2.

Tabela 2: Sintonia de parâmetros do PID na Simulação 1

Tempo (s)	k_p	k_i	k_d
0	0,6655	33,92	3,547
2835	0,8476	60,49	3,547
5645	0,5814	56,74	3,547

Fonte: O próprio autor, 2017.

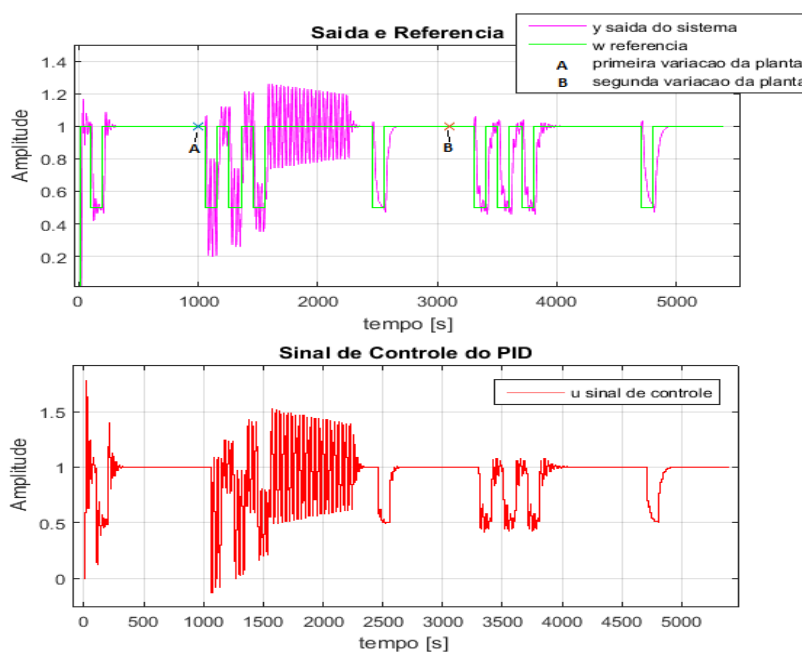
4.3.2 Análise da simulação2: planta G1(s) com PID e sem identificação recursiva.

A resposta do sistema com Identificação Forçada foi próxima ao obtido pelo sistema com Identificação Recursiva. Assim sendo, decidiu-se não apresentar os gráficos resultantes. No entanto, na seção 4.3.5 serão apresentados os valores de OV , t_s , $\max(|u(k)|)$ e $\max(|\Delta u(k)|)$.

4.3.3 Análise da simulação3: planta G2(s) com PID e identificação Recursiva

Seguindo a mesma análise realizada na Subseção 4.3.1, observa-se, por meio da Figura 4.7, a ocorrência da variação “A” em 1000 [s]. Os parâmetros do sistema foram reidentificados em 2400 [s] e o PID resintonizado no instante de 2420 [s]. Aos 2454 [s] ocorreu a mudança de *setpoint*, para a avaliação da sintonia, sem a ocorrência de sobressinal. O t_s foi de 264 [s], visto que a saída igualou-se à entrada em 2718 [s].

Figura 4.7: Sinais de Saída, Referência e Controle da Simulação 3



Fonte: O próprio autor, 2017.

Já a variação “B” ocorreu aos 3100 [s]. Os parâmetros foram reidentificados aos 4660 [s] e o PID resintonizado aos 4678 [s]. A mudança de *setpoint* ocorreu aos 4702 [s], sem a presença de *OV*. A saída igualou-se à entrada aos 5026 [s], correspondendo a um t_s de 324 [s].

Mesmo os parâmetros não tendo sido identificados corretamente através da Identificação Recursiva, o sistema foi controlado dentro dos valores escolhidos de *OV* e t_s . A Tabela 3 apresenta os parâmetros reais e identificados da planta $G_2(z^{-1})$.

Tabela 3: Parâmetros de $G_2(z^{-1})$ reais e identificados através da Simulação 3

Tempo (s)	Parâmetros							
	a_1	\hat{a}_1	a_2	\hat{a}_2	b_1	\hat{b}_1	b_2	\hat{b}_2
0	-1,425	-1,425	0,4966	0,4966	-0,1017	-0,1017	0,173	0,173
2400	-1,489	-1,307	0,5525	0,3479	-0,1378	-0,3003	0,2013	0,3415
4660	-1,549	-1,382	0,604	0,424	-0,179	-0,3751	0,2351	0,4167

Fonte: O próprio autor, 2017

Os parâmetros do Controlador PID foram resintonizados conforme apresenta a Tabela 4.

Tabela 4: Sintonia de parâmetros do PID na Simulação 3

Tempo (s)	k_p	k_i	k_d
0	1,598	0,297	2,566
2420	1,082	0,127	2,583
4678	0,7877	0,08346	2,061

Fonte: O próprio autor, 2017

4.3.4 Análise da simulação4: planta $G_2(s)$ com PID e sem identificação recursiva.

Novamente a resposta do sistema com Identificação Forçada foi próxima daquela obtida utilizando a Identificação Recursiva e, por isso, decidiu-se pela não apresentação dos gráficos. Contudo, serão apresentados, na seção 4.3.5, os valores de *OV*, t_s , $\max(|u(k)|)$ e $\max(|\Delta u(k)|)$.

4.3.5 Resumo das análises do PID para as plantas $G_1(s)$ e $G_2(s)$

A Tabela 5 apresenta um resumo das Simulações 1, 2, 3 e 4. Nota-se que, embora os resultados da Identificação Recursiva não tenham convergido corretamente para os valores desejados, o sistema ainda é controlado na faixa desejada. Isto ocorre pois a regra de sintonia foi bem estruturada, para qualquer valor de τ e K .

Como os parâmetros do controlador PID foram ressintonizados e que, para ambas as plantas, os parâmetros do controlador variam, mesmo com os parâmetros identificados divergindo dos valores reais. Este fato mostra que foi possível controlar o sistema utilizando um controlador PID, mesmo que a Identificação Recursiva convirja para parâmetros diferentes dos reais.

Tabela 5: Resumo dos resultados obtidos através das Simulações 1, 2, 3 e 4.

Simulação	1ª Variação da Planta			2ª Variação da Planta			Geral
	OV [%]	t_s [s]	$\max(u(k))$	OV [%]	t_s [s]	$\max(u(k))$	$\max(\Delta u(k))$
1	4,5	583	1,6	0	570	1,25	1,1
2	0	515	1,53	0	550	0,8	1,1
3	0	264	1,77	0	324	1,081	0,4
4	2,2	250	1,7	0	300	1,5	0,4

Fonte: O próprio autor, 2017.

4.4 GPC adaptativo com sintonia usando AG

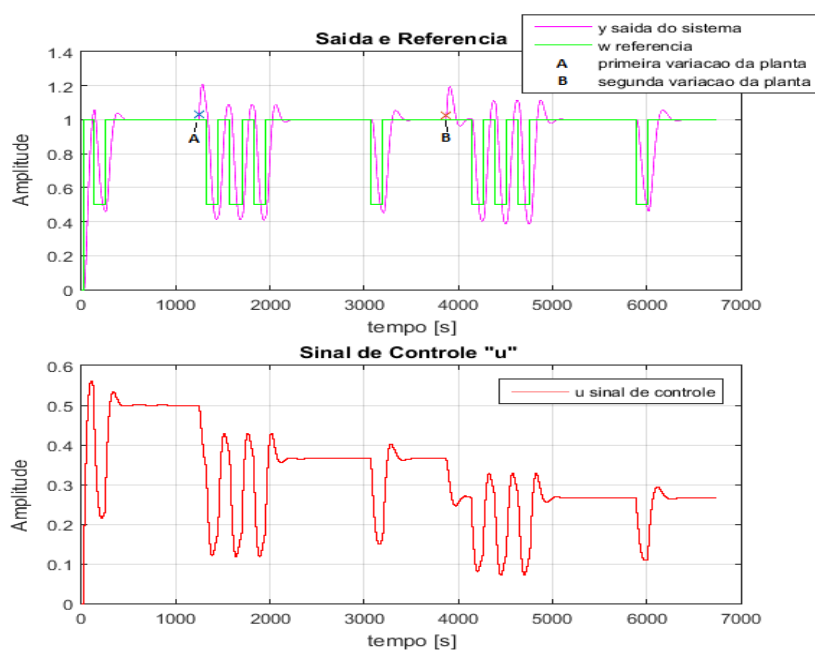
Nesta Seção serão avaliadas as simulações referentes ao sistema com a planta G_1 e G_2 utilizando um GPC ressintonizado por meio de Algoritmo Genético. Em relação ao GPC, para a planta G_1 , foi escolhido um $h_p = d + 1 = 5$ (com um tempo de atraso, d , de 4 segundos) e para a planta G_2 , um $h_p = d + 1 + 7 = 9$, com $d = 1$. Para o Algoritmo Genético foi selecionada uma população de 100 indivíduos, 5 gerações e apenas uma repetição do algoritmo. Isto demandou uma média de 10 segundos de processamento para cada ressintonia do GPC.

4.4.1 Análise da simulação5: planta $G_1(s)$ com GPC com AG e identificação Recursiva

Analisando a Figura 4.8 nota-se a ocorrência da variação “A” aos 1248 [s], sendo a reidentificação aos 2833 [s] e ressintonia do GPC aos 2835 [s]. No instante 3085 [s], ocorreu uma mudança de *setpoint* e aos 3340 [s], um sobressinal de 5,5%. O tempo de acomodação foi de 490 [s], visto que a saída tornou-se igual à entrada aos 3575 [s].

A variação “B” ocorreu aos 3873 [s]. A reidentificação e a ressintonia do GPC ocorreram respectivamente aos 5600 [s] e 5645 [s]. Aos 5895 [s] ocorreu a mudança de *setpoint* e aos 6173 [s], um sobressinal de 5,7%. Uma vez que a saída igualou-se à entrada aos 6505 [s], o t_s foi de 610 [s].

Figura 4.8: Sinais de Saída, Referência e Controle da Simulação 5.



Fonte: O próprio autor, 2017.

As Tabela 6 e Tabela 7 apresentam, respectivamente, os parâmetros reais e identificados da planta $G_1(z^{-1})$ e os parâmetros ressintonizados do Controlador GPC.

Semelhante ao que ocorreu nas simulações anteriores, mesmo com a Identificação Recursiva não acertando corretamente os parâmetros, o sistema foi controlado dentro dos valores escolhidos de $0V$ e t_s . Ao se comparar a Tabela

6 com a Tabela 1 nota-se que a identificação recursiva tem resultados diferentes na identificação dos parâmetros do sistema com controlador GPC e PID.

Tabela 6: Parâmetros de $G_1(z^{-1})$ reais e identificados através da Simulação 5

Tempo (s)	Parâmetros			
	a_1	\hat{a}_1	b_1	\hat{b}_1
0	-0,9048	-0,9048	0,1903	0,1903
2833	-0,9187	-0,9179	0,2226	0,2248
5600	-0,9307	-0,9245	0,2601	0,2836

Fonte: O próprio autor, 2017

Tabela 7: Sintonia de parâmetros do GPC na Simulação 5

Tempo (s)	λ/δ
0	5,557
2835	8,923
5645	18,08

Fonte: O próprio autor, 2017.

4.4.2 Análise da simulação6: planta G1(s) com GPC com AG e identificação Forçada

Novamente a resposta do sistema com Identificação Forçada foi próxima da do sistema com Identificação Recursiva, não sendo necessária a apresentação dos respectivos gráficos. Na seção 4.4.5, os valores de OV , t_s , $\max(|u(k)|)$ e $\max(|\Delta u(k)|)$, obtidos pela simulação em questão, poderão ser visualizados.

4.4.3 Análise da simulação7: planta G2(s) com GPC com AG e identificação Recursiva

Uma vez que a resposta do sistema com Identificação Recursiva não se aproximou do valor desejado de OV e t_s , decidiu-se pela não apresentação dos gráficos pertencentes a esta simulação. Na seção 4.4.5, os respectivos valores de OV , t_s , $\max(|u(k)|)$ e $\max(|\Delta u(k)|)$ poderão ser visualizados.

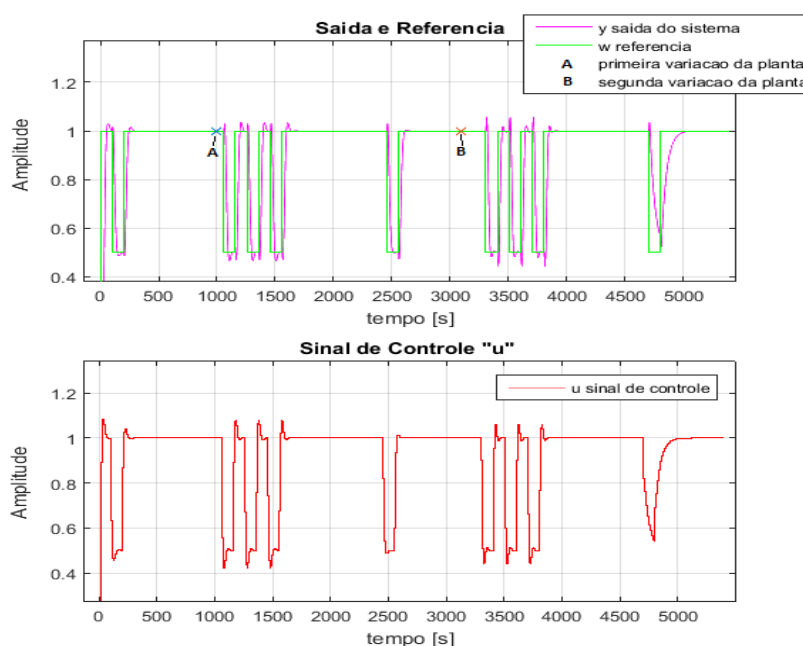
Nesta simulação o AG não conseguiu encontrar uma resposta dentro dos valores desejados de OV e t_s , pois, não existia tal resposta na planta

identificada. Isto ocorreu por em razão do detalhe da identificação recursiva em malha fechada divergir dos parâmetros reais, que, neste caso, a divergencia foi de valor elevado causando um sistema identificado que não possuía nenhuma resposta dentro dos valores desejados.

4.4.4 Análise da simulação8: planta G2(s) com GPC com AG e sem identificação recursiva.

Por meio da Figura 4.9, observa-se a ocorrência da variação “A” no instante 1000 [s], sendo os parâmetros reidentificados aos 2400 [s] e o GPC resintonizado no instante 2420 [s]. A avaliação da resintonia, através da mudança de *setpoint*, ocorreu aos 2454 [s]. No instante 2622 [s], ocorreu um *OV* de 0,2% e aos 2648 [s] a saída tornou-se idêntica à entrada, o que resulta em um t_s de 194 [s].

Figura 4.9: Sinais de Saída, Referência e Controle da Simulação 8.



Fonte: O próprio autor, 2017.

Já a variação “B” ocorreu no instante 3100 [s]. Os parâmetros foram reidentificados aos 4660 [s] e o GPC resintonizado o GPC aos 4678 [s]. Uma

outra mudança de *setpoint*, sem a ocorrência de sobressinal, ocorre aos 4702 [s]. O tempo de acomodação do sistema foi de 500 [s], visto que a saída igualou-se a entrada aos 5202 [s].

A Identificação Forçada empregada à planta $G_2(z^{-1})$, resultou a correta identificação de seus parâmetros, como mostra a Tabela 8.

Tabela 8: Parâmetros de $G_2(z^{-1})$ reais e identificados através da Simulação 8.

Tempo (s)	Parâmetros							
	a_1	\hat{a}_1	a_2	\hat{a}_2	b_1	\hat{b}_1	b_2	\hat{b}_2
0	-1,425	-1,425	0,4966	0,4966	-0,1017	-0,1017	0,173	0,173
2400	-1,489	-1,489	0,5525	0,5525	-0,1378	-0,1378	0,2013	0,2013
4660	-1,549	-1,549	0,604	0,604	-0,179	-0,179	0,2351	0,2351

Fonte: O próprio autor, 2017.

Os parâmetros do Controlador GPC λ/δ foram resintonizados, de acordo com os valores apresentados pela Tabela 9.

Tabela 9: Sintonia de parâmetros do GPC na Simulação 8.

Tempo (s)	λ/δ
0	7,876
2420	4,565
4678	0,6116

Fonte: O próprio autor, 2017

4.4.5 Resumo das Análise do GPC com AG para plantas $G_1(s)$ e $G_2(s)$

A Tabela 10 mostra um resumo dos resultados gerados através das Simulações 5, 6, 7 e 8. Observa-se que nas simulações 5 e 6 foram obtidos parâmetros com valores bastante semelhantes e, uma vez que, o AG foi configurado para procurar valores de OV menores que 10% e $\max(|\Delta u(k)|)$ menores que 0,2, os resultados ficaram dentro da faixa de valores desejada. Embora na Simulação 7, tenha sido verificada a ocorrência de um OV acima de 10%, justificada pela propriedade de aleatoriedade do AG, para as demais simulações o AG encontrou sintonias satisfatórias, isto é, valores OV menores que 10% e $\max(|\Delta u(k)|)$ menores que 0,2.

Tabela 10: Resumo dos resultados obtidos através das Simulações 5, 6, 7 e 8.

Simulação	1ª Variação da Planta			2ª Variação da Planta			Geral
	OV [%]	t_s [s]	$\max(u[k])$	OV [%]	t_s [s]	$\max(u[k])$	
5	5,5	490	0,6	5,7	610	0,3	0,04
6	5,4	455	0,6	4,2	650	0,4	0,035
7	20,1	300	1,276	2,2	450	1,428	0,1996
8	0,2	194	1,085	0	500	1,03	0,1862

Fonte: O próprio autor, 2017.

Portanto, é possível concluir que, para a planta G_1 , o controlador GPC com AG consegue controlar o sistema mesmo que a Identificação Recursiva convirja para parâmetros diferentes dos reais. Diferentemente do que ocorre com a planta G_2 , a qual a Identificação Recursiva não foi capaz de convergir para os parâmetros reais.

4.5 GPC adaptativo com sintonia usando regra de sintonia

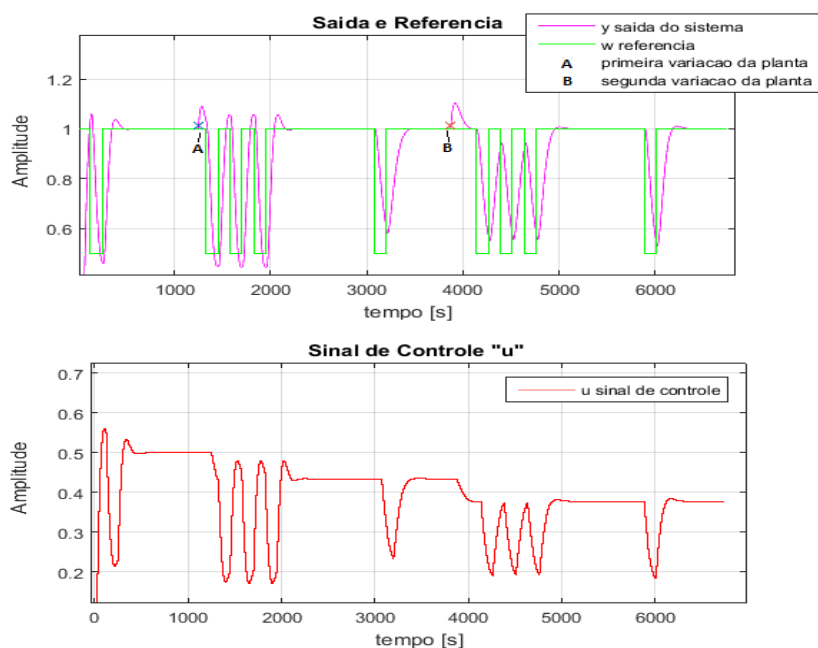
Nesta seção serão avaliadas as simulações referentes ao sistema novamente com a planta G_1 , porém utilizando o GPC ressintonizado por Regras de Sintonia desenvolvidas neste estudo.

4.5.1 Análise da simulação9: planta $G1(s)$ com GPC com Regra de Sintonia e identificação Recursiva

Avaliando a Figura 4.10, é possível constatar a ocorrência da variação “A” no instante 1248 [s]. Os parâmetros da planta foram reidentificados aos 2833 [s] e o GPC foi ressintonizado aos 2835 [s]. A mudança de *setpoint*, ocorreu aos 3085 [s] e no instante 3528 [s] verifica-se um sobressinal de 0,1%. O t_s foi de 513 [s], visto que a saída igualou-se à entrada no instante 3598 [s].

A variação “B” ocorreu no instante 3873 [s], o sistema foi reidentificado aos 5600 [s] e o controlador foi ressintonizado aos 5645 [s]. Aos 5895 [s] ocorreu outra mudança de *setpoint*, com um sobressinal de 1,1% aos no instante 6230 [s]. Aos 6360 [s] a saída tornou-se igual a entrada, resultando em um t_s de 465 [s].

Figura 4.10: Sinais de Saída, Referência e Controle da Simulação 9.



Fonte: O próprio autor, 2017.

De acordo com a Tabela 11, a Identificação Recursiva promoveu a correta identificação dos parâmetros da planta.

Tabela 11: Parâmetros de $G_1(z^{-1})$ reais e identificados através da Simulação 9.

Tempo (s)	Parâmetros			
	a_1	\hat{a}_1	b_1	\hat{b}_1
0	-0,9048	-0,9048	0,1903	0,1903
2833	-0,9187	-0,9042	0,2226	0,2204
5600	-0,9307	-0,9075	0,2601	0,2445

Fonte: O próprio autor, 2017.

Os parâmetros do Controlador GPC $\frac{\lambda}{\delta}$ foram ressintonizados, conforme ilustra a Tabela 12.

Tabela 12: Sintonia de parâmetros do GPC na Simulação 9.

Tempo (s)	λ/δ
0	5,56
2835	15,64
5645	15,66

Fonte: O próprio autor, 2017.

4.5.2 Análise da simulação10: planta G1(s) com GPC com Regra de Sintonia e sem identificação recursiva.

Uma vez que a resposta do sistema com Identificação Forçada foi próxima da resposta obtida pelo sistema com Identificação Recursiva, decidiu-se pela não apresentação dos gráficos gerados. No entanto, segue na Seção 4.5.3 os valores de OV , t_s , $\max(|u(k)|)$ e $\max(|\Delta u(k)|)$ resultantes.

4.5.3 Resumo das análises do GPC com Regra de Sintonia para a planta G1(s)

A Tabela 13 apresenta os valores OV , t_s , $\max(|u(k)|)$ e $\max(|\Delta u(k)|)$ gerados pelas Simulações 9 e 10. Deve-se mencionar que a regra de sintonia foi gerada apenas para um pequeno intervalo de K e τ .

Tabela 13: Resumo dos resultados obtidos através das Simulações 9 e 10.

Simulação	1ª Variação da Planta			2ª Variação da Planta			Geral
	OV [%]	t_s [s]	$\max(u(k))$	OV [%]	t_s [s]	$\max(u(k))$	
9	0,1	513	0,44	1,1	465	0,38	0,034
10	1,1	490	0,55	0,9	500	0,3	0,035

Fonte: O próprio autor, 2017.

A sintonia do controlador GPC consegue controlar o sistema, desde que e os parâmetros variem dentro da faixa a qual a regra de sintonia foi criada. Desta forma, gerar uma regra de sintonia para, por exemplo, τ entre zero e infinito parece resolver o problema, uma vez que, o controle do sistema é mais importante do que a exata identificação dos parâmetros.

4.6 Simulações para as plantas G3(s) e G4(s)

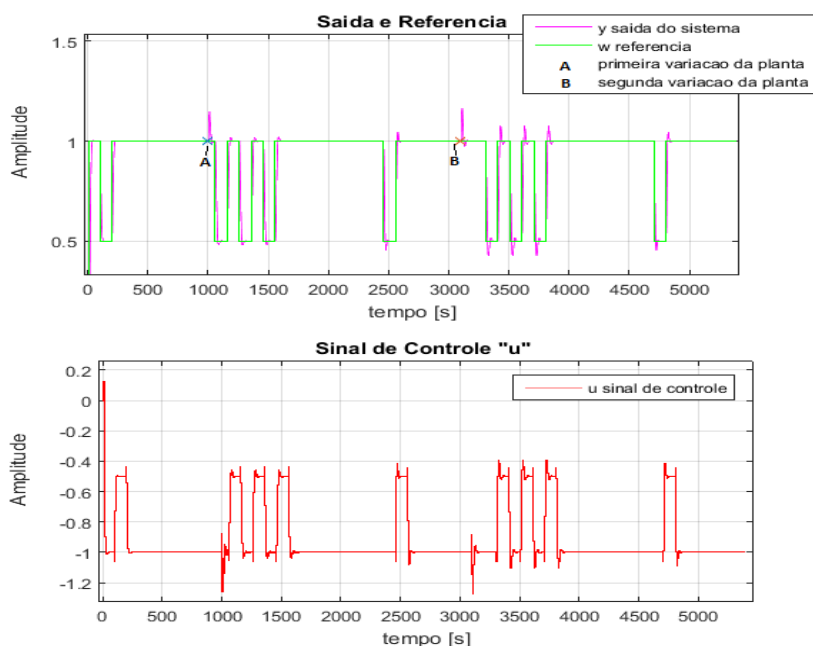
Nesta subseção serão avaliadas as simulações referentes às plantas G_3 e G_4 com GPC de forma ganho escalonado, encontrado através do Algoritmo Genético. Os valores de h_p e h_c foram fixados (sendo $h_c = 1$, para ambas as plantas e h_p igual a 4 e 5, respectivamente, para as plantas G_3 e G_4 , enquanto que variou-se a razão λ/δ).

Por se tratarem de plantas instáveis, os casos de PID não podem ser aplicados às mesmas. Além disto, em virtude da complexidade inerente às plantas, o AG foi sintonizado para buscar respostas com OV abaixo de 10% e uma variação do sinal de controle abaixo de 0,2 e 2, respectivamente, para as para as plantas G_3 e G_4 .

A Figura 4.11 ilustra os sinais de saída, referência e controle da planta G_3 durante a simulação. Através desta figura, observa-se a ocorrência da variação “A” no instante 1000 [s]. Os parâmetros foram reidentificados no instante 2300 [s] e o GPC ressintonizado aos 2360 [s]. A validação da sintonia, através da mudança de *setpoint*, se deu aos 2458 [s], com a ocorrência de um sobressinal de 4,5% aos 2576 [s]. No instante 2636 [s], a saída igualou-se à entrada, resultando em um t_s de 178 [s].

A variação “B” ocorreu aos 3100 [s], a reidentificação dos parâmetros ocorreu no instante 4630 [s] e o GPC foi ressintonizado aos 4672 [s]. No instante 4702 [s], ocorreu outra mudança de *setpoint*, com um sobressinal de 4,7% aos 4824 [s]. O t_s foi de 180 [s], visto que a saída igualou-se à entrada no instante 4882 [s].

Figura 4.11: Sinais de Saída, Referência e Controle da simulação da Planta G_3 .



Fonte: O próprio autor, 2017.

Sobre os parâmetros da identificação, estes foram forçados a serem os reais, descartando a necessidade de uma tabela para mostrar os mesmos. Já os parâmetros do Controlador GPC $\frac{\lambda}{\delta}$, estes foram resintonizados conforme apresenta a Tabela 14.

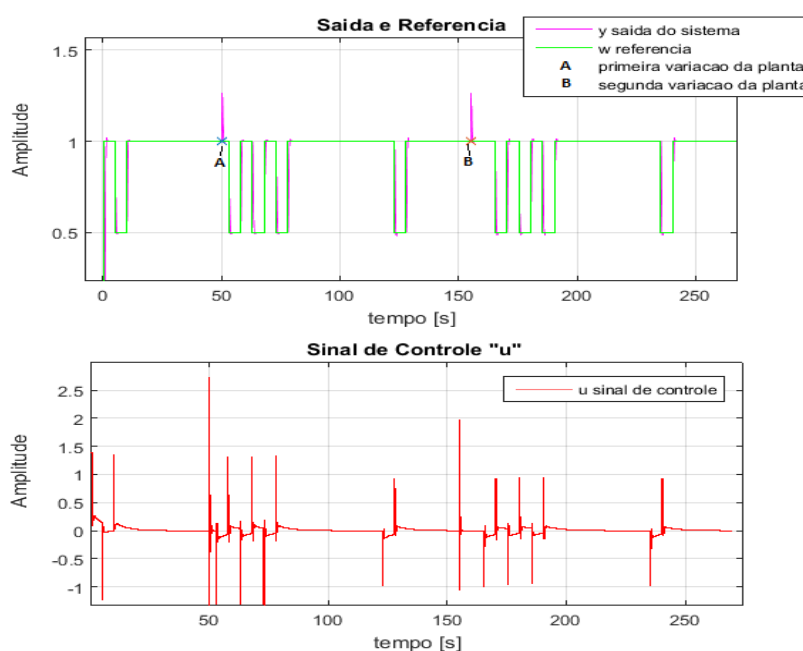
Tabela 14: Sintonia de parâmetros do GPC na simulação da Planta G_3 .

Tempo (s)	λ/δ
0	7,985
2360	21,14
4672	20,54

Fonte: O próprio autor, 2017.

A Figura 4.12 refere-se à simulação da planta G_4 . Nota-se que a variação “A” ocorreu no instante 50 [s] e a reidentificação dos parâmetros do sistema e a resintonia do GPC, deram-se respectivamente nos instantes 118 [s] e 120 [s]. Aos 122,8 [s], ocorreu outra mudança de *setpoint* e aos 128,8 [s], ocorreu um sobressinal de 1,9%. A saída igualou-se a entrada no instante 133,4 [s], o que significa um t_s de 10,6 [s].

Figura 4.12: Sinais de Saída, Referência e Controle da simulação da Planta G_4 .



Fonte: O próprio autor, 2017.

A variação “B” ocorreu em 155 [s] e foi reidentificado até 229 [s] e então resintonizado o GPC em 232 [s], em 235,2 [s] ocorreu outra mudança de *setpoint* para se avaliar a resintonia, em 241,2 [s] ocorreu o sobressinal de 1,7% e em 245,4 [s] ocorreu de a saída ficar igual a entrada, o que significa que o t_s é de 10,2 [s].

Assim como ocorreu com a planta G_3 , os parâmetros da identificação para a planta G_4 também foram forçados a serem os reais, descartando, assim, a necessidade de uma tabela para a apresentação dos mesmos. Os parâmetros do Controlador GPC λ/δ , foram resintonizados de acordo com os valores apresentados pela Tabela 15.

Tabela 15: Sintonia de parâmetros do GPC na simulação da Planta G_4 .

Tempo (s)	λ/δ
0	0,001481
120	0,1211
232	0,09398

Fonte: O próprio autor, 2017.

A Tabela 16 evidencia os resultados obtidos para as simulações das plantas G_3 e G_4 utilizando o GPC com AG e Identificação Forçada.

Tabela 16: Resumo dos resultados das simulações G_3 e G_4 .

Planta	1ª Variação da Planta			2ª Variação da Planta			Geral
	OV [%]	t_s [s]	$\max(u(k))$	OV [%]	t_s [s]	$\max(u(k))$	
G_3	4,5	178	-0,4146	4,7	180	-0,4115	0,19
G_4	1,9	10,6	0,923	1,7	10,2	0,9209	1,367

Fonte: O próprio autor, 2017.

Observa-se que as respostas encontradas estão dentro da faixa desejada de OV e da variação do sinal de controle. Ainda vale lembrar que, mesmo se não estivessem as respostas de acordo com os valores desejados, o AG retornaria a sintonia com o valor mais próximo do desejado.

4.7 Simulações para as plantas $G_5(s)$ e $G_6(s)$

Com o intuito de reforçar a precisão do método para avaliar a estabilidade do GPC utilizado neste trabalho, bem como da utilidade da interface gráfica, também foram testadas as plantas G_5 e G_6 , com o intuito de demonstrar a facilidade de, não só, avaliar a estabilidade, bem como de se projetar controlador GPC para as mesmas.

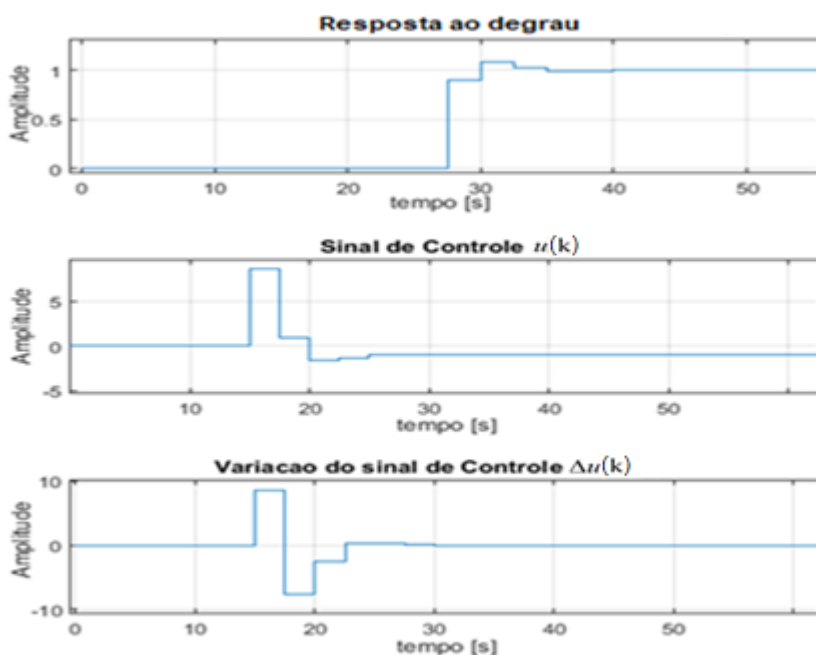
A primeira planta, G_5 é evidenciada pelas Equações (42) e (43), as quais apresentam, respectivamente, as funções de transferência contínua e discreta sendo o período de amostragem T_0 igual a 2,5 segundos.

$$G_5(s) = \frac{0,04001}{s - 0,03994} e^{-7,5s} \quad (42)$$

$$G_5(z^{-1}) = z^{-4} \frac{0,1052}{1 - 1,105z^{-1}} \quad (43)$$

A finalidade desta simulação é comprovar que o método CHR não consegue resintonizar os parâmetros, visto que este foi projetado apenas para plantas de primeira ordem e estáveis. Diferentemente do que ocorre com o GPC com AG, que é capaz de buscar os parâmetros $\frac{\lambda}{\delta}$ para quaisquer casos.

Foi escolhido um $h_p = d + 1 = 5$, com $d = 4$, sendo a única restrição do AG a busca por um OV abaixo de 10% (não houve restrição no que concerne à variação de sinal de controle). Os resultados da simulação da planta G_5 podem ser analisados através da Figura 4.13. Aos 30 [s] ocorreu o OV de 8,82%, o sinal de controle e a variação do sinal de controle atingiram valor máximo de 9,4 aos 15 [s], sendo encontrado um valor de $\frac{\lambda}{\delta}$ igual a 0,001.

Figura 4.13: Resposta encontrada pelo AG para a Planta $G_5(z^{-1})$.

Fonte: O próprio autor, 2017.

Além disso, foi simulado o GPC escolhendo-se o valor de $\lambda/\delta = 0,104$ visando gerar um sistema Marginalmente Estável (ME). Ao variar o valor de h_c de 1 até 4, foi possível verificar que a estabilidade de $h_c = 1$ e $h_c > 1$ é a mesma como mostra Tabela 17.

Tabela 17: Resultados de testes para $\lambda/\delta = 0,104$, ME para $h_p = 5$ com a Planta $G_5(z^{-1})$.

h_c	Estabilidade
1	ME
2	ME
3	ME
4	ME

Fonte: O próprio autor, 2017.

Ao simular valores de $\frac{\lambda}{\delta}$ na região de estabilidade, variando-se o $h_c = 1$ e $h_c > 1$, foi possível concluir que o aumento do h_c não surte em mudanças significativas quanto à estabilidade. As mudanças registradas foram na curva do sinal de controle, que passou a ter uma característica mais suave, acarretando em um aumento dos valores de t_s e OV . Assim, o aumento do h_c , resulta em um sistema que exige maior capacidade de processamento do *hardware* embarcado.

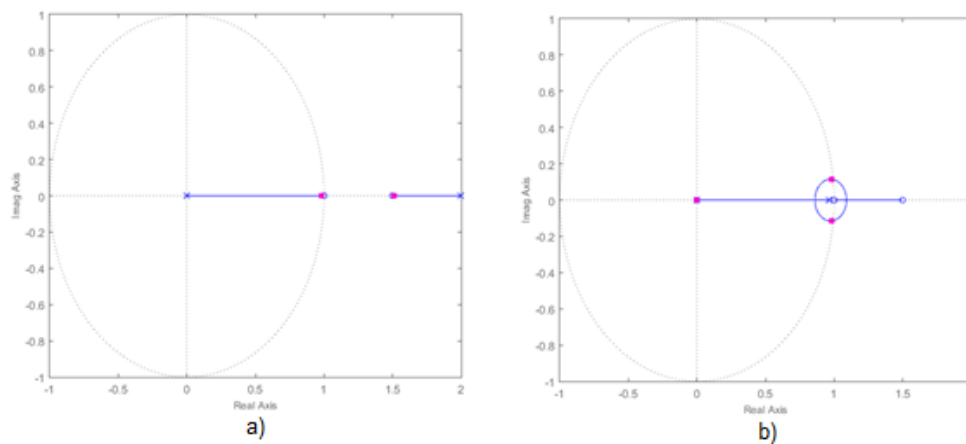
Outra planta utilizada nas simulações foi a G_6 , mencionada por ROSSITER (2004), descrita pelas Equações (44) e (45). Trata-se de uma planta de difícil sintonia através de GPC para $h_c = 1$. Além disto, quando $h_p = d + 1$, esta planta não possui região de estabilidade.

$$G_6(s) = \frac{0,1s - 0,8109}{s - 4,055} e^{-0,1s} \quad (44)$$

$$G_6(z^{-1}) = z^{-1} \frac{0,1 - 0,2z^{-1}}{1 - 1,5z^{-1}} \quad (45)$$

Ao se utilizar o método proposto neste trabalho, a sintonia desta planta é feita com facilidade. A Figura 4.14 apresenta o Lugar das Raízes para diferentes valores de h_p . Para $h_p = 2$, qualquer valor de λ/δ gera um sistema instável, já para um $h_p = 5$, observa-se uma região contendo valores de λ/δ que resultam em sistemas estáveis e uma outra região que gera sistemas instáveis.

Figura 4.14: Lugar das Raízes da planta $G_6(z^{-1})$. em malha fechada com GPC com a) $h_p = d + 1 = 2$ e b) $h_p = d + 1 + 3 = 5$.



Fonte: O próprio autor, 2017.

4.8 Conclusão dos Resultados das simulações

Para planta G_1 , tanto o controlador PID quanto o GPC *auto-tuning* sintonizado por AG, obtiveram respostas dentro da faixa de valores desejada. Ao se utilizar o controlador GPC *auto-tuning* sintonizado por regras de sintonia, foi necessário, primeiramente, que a Identificação Recursiva acertasse os parâmetros, uma vez que as regras de sintonia abordadas por este trabalho são para uma pequena faixa de valores de K e τ .

Ainda sobre plantas de primeira ordem, foi constatado que nem sempre o aumento do valor de h_c do GPC, resulta em melhores soluções. Desta forma, por questões de demanda de processamento, sugere-se o uso de $h_c = 1$.

Para a planta G_2 , através da sintonia de PID, que já é bem fundamentada, foram obtidos resultados bastante significativos. Ao se implementar o GPC *auto-tuning* com AG, tornou-se necessário que a Identificação Recursiva primeiramente acertasse os parâmetros.

Além disto, durante a execução dos testes, observou-se que quando o parâmetro h_p do GPC é escolhido como $h_p = d + 1$ e, se por algum motivo, o atraso da planta muda, o projeto fica comprometido. Isto se deve ao valor de h_p alterar significativamente a característica do Lugar das Raízes, podendo alterar a região de estabilidade do sistema. A solução é escolher um $h_p = d + 1 + c$, sendo c uma constante cujo valor está de acordo com o valor máximo da variação do atraso do sistema. No entanto, isto pode ocasionar um valor elevado de h_p .

5 Conclusões

O problema de sintonia do Controlador Preditivo Generalizado vem sendo alvo de estudos nas últimas décadas e apresenta atualmente grande evolução contínua por não existir uma solução geral do ponto de vista da teoria. A técnica aqui proposta, permite ao usuário avaliar a estabilidade do sistema em malha fechada, para plantas estáveis ou não, para o caso de $hc=1$. Porém, também foi mostrado que para a planta G_5 , ao se aumentar o valor de hc não foi alterada a região de estabilidade, sugerindo que a técnica aqui apresentada pode ser usada para avaliar a estabilidade de algumas plantas para qualquer valor de hc .

Ainda sobre a sintonia de GPC, mostrou-se que a técnica proposta neste trabalho, facilita o projeto para a planta G_6 , a qual foi considerada planta difícil para se projetar um GPC.

Quanto ao caso de GPC adaptativo com a utilização do AG, este mostrou-se promissor no caso do modo *auto-tuning*, pois, se o projeto for bem feito com a técnica mostrada neste trabalho, o AG levará apenas segundos para encontrar o melhor caso na etapa de ressintonia dos parâmetros λ e δ do GPC, desde que, a identificação recursiva convirja para os parâmetros da planta.

Para o caso de GPC *auto-tuning* com Regras de Sintonia, mostrou-se a necessidade de se gerar as funções com grande quantidade de dados, para que esta ressintonize os parâmetros λ e δ para qualquer planta, pois, as funções criadas neste trabalho foram apenas para um pequeno intervalo de ganho K e constante de tempo τ de uma planta de primeira ordem.

Como sugestão para futuras pesquisas, atenta-se a refinar as regras de sintonia aqui produzidas, estudar algum algoritmo que consiga buscar a melhor região de λ e δ do Lugar das Raízes sem precisar da análise de um operador, bem como, avaliar para quais tipos de plantas o valor de $hc>1$ altera a região de estabilidade, aplicar o que foi desenvolvido em experimento real utilizando o Arduíno e verificar a sintonia para plantas com variação do seu atraso de transporte garantindo que o parâmetro hp acompanhe esta evolução. Além disto, desenvolver e experimentar um algoritmo de identificação recursiva

baseada no método das Variáveis Instrumentais, como proposto por Coelho (2006).

Como sugerido pela banca, aplicar o que foi desenvolvido em experimento real utilizando o *hardware Raspberry*.

Referências Bibliográficas

- AGUIRRE, L. **INTRODUÇÃO À IDENTIFICAÇÃO DE SISTEMAS Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais**. 3rd. ed. Belo Horizonte - MG: Editora UFMG, 2007.
- AL-DUWAISH, H.; NAEEM, W. **Nonlinear Model Predictive Control of Hammerstein and Wiener Models Using Genetic Algorithms**. Mexico City: IEEE International Conference on Control Applications, 2001. 465-469 p.
- AL-GHAZZAWAI, A. et al. **On-line tuning strategy for model predictive controllers**. [S.l.]: Journal of process Control, v. 11, 2001. 265-284 p.
- ALMEIDA, G. M. **Sintonia de controle preditivo generalizado através de algoritmo genético**. Vitória/ES: Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica, PPGE/UFES, 2006.
- ALMEIDA, G.; SALLES, J. L. **CONTROLE PREDITIVO Sintonia e aplicações na siderurgia**. 1st. ed. Curitiba - PR: Appris, 2016.
- BANERJEE, P.; SHAH, S. **Tuning guidelines for robust generalized predictive control**. [S.l.]: Proceedings of the 31st IEEE Conference on Decision and Control, 1992. 3233-3234 p.
- BITMEAD, R. R.; GEVERS, M.; WERTZ, V. **Adaptive Optimal Control The Thinking Man's GPC**. Brunswick, Victoria: Prentice Hall, 1990.
- BOBÁL, D. et al. **Digital Self-tuning Controllers**. London: Springer, 2005.
- BOBÁL, V.; KUBALCÍK, M.; DOSTÁL, P. **Identification and Self-tuning Predictive Control of Heat Exchanger**. Štrbské Pleso, Slovakia: International Conference on Process Control (PC), 2013. 219-224 p.
- CAMACHO, E. F.; BORDONS, C. C. **Model Predictive Control**. 2nd. ed. New York: [s.n.], 2000.
- CHEN, C. T. **Linear System Theory and Design**. New York: Holt-Saunders International Editions, 1999.

CLARKE, D. W.; MOHTADI, C.; TUFFS, P. S. **Generalized predictive control. part i. the basic algorithm.** [S.l.]: Automatica, 1987.

CLARKE, D.; MOHTADI, C. **Properties of generalized predictive control.** [S.l.]: Automatica, v. 25, 1989.

COELHO, N. **Identificacao de Sistemas Lineares Dinâmicos.** Paraná: UFPR, 2006.

FAN, L.; JOO, E. M. **Design for Auto-tuning PID Controller Based on Genetic Algorithms.** [S.l.]: IEEE, 2009. 1924-1928 p.

GARRIGA, J. L.; SOROUSH, M. **Model predictive control tuning methods: a review.** [S.l.]: Ind. Eng. Chem. Res., v. 39, 2010. 3505-3515 p.

KANJILAL, P. P. **Adaptive Prediction and Predictive Control.** London: The Institution of Engineering and Technology, 1995.

KARACAN, H.; HAPOGLU, M. A. S. **Application of optimal adaptive generalized predictive control to a packed distillation column.** [S.l.]: Chemical Engineering Journal, v. 84, 2000. 389-393 p.

KOKATE, R. D.; WAGHMARE, L. M.; DESHMUKH, S. D. **Review of Tuning Methods of DMC and Performance Evaluation with PID Algorithms on a FOPDT Model.** Kottayam: International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 2010. 71-75 p.

LEWIS, F.; VRABIL, D.; SYRMOS, V. **OPTIMAL CONTROL.** New Jersey: John Wiley & Sons, Inc. Hoboken, v. 3a , 2012.

LJUNG, L. **System Identification: Theory for the User.** New Jersey: Prentice-Hall, Inc., v. 2a, 1999.

MAHMOUDI, H.; LESANI, M. J.; KHABOURI, D. A. **Online Fuzzy Tuning of Weighting Factor in Model Predictive Control of PMSM.** [S.l.]: Iranian Conference on Fuzzy Systems (IFSC), v. 13, 2013.

MAN, K. F.; TANG, K. S.; KWONG, S. **Genetic Algorithms.** London: Springer, 1999.

MAZOCO, B.; SALLES, J. L. **PROPOSTA DE UM ALGORITMO GPC ADAPTATIVO COM BAIXO CUSTO COMPUTACIONAL**, Dissertação de Mestrado em Engenharia Elétrica. Vitória - ES: UFES, 2015.

PIONTKOVSKY FILHO, A. J.; SALLES, J. L. F. **Sintonia Eficiente do Esforço de Controle do GPC com horizonte de controle unitário**. São José do Rio Preto, SP: Conferência Brasileira de Dinâmica, Controle e Aplicações (DINCON), 2017.

RANI, K. Y.; UNBEHAUEN, H. **Study of predictive controller tuning methods**. [S.l.]: Automatica, v. 33, 1997. 2243-2248 p.

ROSSITER, J. A. **Model-based predictive control: a practical approach**. Boca Raton, Fla.: CRC Press, 2004.

SARIMVEIS, H.; BAFAS, G. **Fuzzy model predictive control of non-linear processes using genetic algorithms**. [S.l.]: Elsevier B.V., 2002. 59-80 p.

SATO, T. **Design of a GPC-based PID controller for controlling a weigh feeder**. [S.l.]: Control Engineering Practice Elsevier B.V., 2010. 105-113 p.

SINTHIPSOMBOON, K.; PONGAEN, W.; PRATUMSUWAN, P. **A Hybrid of Fuzzy and Fuzzy self-tuning PID Controller for Servo Electro-hydraulic System**. [S.l.]: IEEE Conference on Industrial Electronics and Applications, 2011. 220-225 p.

TALIB, M. H. A.; DARUS, I. Z. M. **Self-Tuning PID Controller for Active Suspension System with Hydraulic Actuator**. [S.l.]: IEEE Symposium on Computers & Informatics, 2013. 86-91 p.

TRAN, Q. N. et al. **Generalized Predictive Control tuning by controller mathing**. [S.l.]: American Control Conference, 2014. 4889-4894 p.

WELLSTEAD, P. E.; ZARROP, M. B. **Self-tuning Systems: Control and Signal Processing**. [S.l.]: John Wiley & Sons Ltd, 1991.

YOSHITANI, N.; HASEGAWA, A. **Model-based control of strip temperature for the heating furnace in continuous annealing**. [S.l.]: IEEE Transactions on Control Systems Technology, v. 6, 1998.

ZIOGOU, C. et al. **On-line nonlinear model predictive control of a PEM fuel cell system.** [S.l.]: Journal of Process Control, v. 23, 2013. 483-492 p.

Apêndice A – Funções do código do Algoritmo Genético

O código do AG utilizado neste trabalho foi desenvolvido por Gustavo Maia de Almeida e adaptado a este trabalho pelo autor, e é composto por doze rotinas, sendo estas apresentadas na Tabela 18.

Tabela 18: Rotinas associadas ao código do Algoritmo Genético utilizado neste trabalho

	rotina	explicação
1	run_genetic_algorithms.m	Função principal deste código de AG. Recebe como entradas o tamanho da população e a quantidade de gerações e então utiliza as outras rotinas para retornar apenas o melhor indivíduo e o fitness à este associado após realizar calculos de todas as populações de todas as gerações.
2	create_individual.m	Cria um indivíduo com os genes escolhidos. É nesta rotina que deve-se escrever como um indivíduo será criado, e então este será repetido pela rotina "create_population.m" para se criar toda a população.
3	create_population.m	Repete a função "create_individual.m" para criar uma população de tamanho escolhido como entrada na rotina 1.
4	standardized.m	Calcula o fitness de apenas um indivíduo (a rotina "juntos.m" repete a "standardized.m" para cada indivíduo). É nesta rotina que deve-se escrever os cálculos para realização do cálculo do fitness de um indivíduo, além de também ser possível de se adicionar lógicas de manipulação do fitness.

Tabela 1: Rotinas associadas ao código do Algoritmo Genético utilizado neste trabalho
(continuação)

5	juntos.m	Calcula o fitness de cada indivíduo para uma geração ao repetir a função "standardized.m", e organiza de forma decrescente os indivíduos da população em relação ao fitness.
6	execute_generation.m	Executa a funcao "juntos.m" para cada geração e usa a função "breed_new_population.m" para criar a população da próxima geração.
7	breed_new_population.m	Cria uma nova população selecionando indivíduos pelo método Seleção por Roleta e então realizando as operações genéticas.

Fonte: próprio autor, 2018.

Tabela 18: Rotinas associadas ao código do Algoritmo Genético utilizado neste trabalho
(continuação)

	rotina	explicação
8	operator.m	Aqui ocorrem as operações genéticas (a função "breed_new_population.m" que chama esta função)
9	variable.m	Cria variáveis globais que são usadas na execução do código.
10	random_integer.m	Apenas uma lógica para um número aleatório inteiro.
11	random_floating_point_number.m	Apenas uma lógica para um número aleatório real.
12	park_miller_randomizer.m	Apenas para auxiliar a aleatoriedade do código.

Fonte: próprio autor, 2018.

Sendo as taxas de *crossover*, mutação e reprodução inclusas na rotina 8 – operator.m – e assumem os valores de 90%, 1% e 1%, respectivamente.

Diante disso, deve-se utilizar a rotina 1 para iniciar a execução de todo o código. As rotinas 2 e 4, que são as rotinas que definem o problema, pois é onde cada indivíduo será criado e como este será avaliado (cálculo de *fitness*). Desta forma, para o problema de sintonia do GPC, a rotina 2 deve conter a lógica de como os indivíduos λ e δ serão criados obedecendo o intervalo escolhido pelo usuário, e a rotina 4 deve conter os cálculos das equações apresentadas na seção 2.1 e 2.1.1 além da função escolhida como *fitness*.

Apêndice B – Interface Gráfica

Com o objetivo de agilizar o projeto do Controlador GPC foi desenvolvida uma interface gráfica na qual o usuário poderá projetar um controlador GPC para uma planta e visualizar o sinal de saída, a entrada degrau unitário, sinal de controle e variação do sinal de controle em malha fechada. Além disso, pode-se mostrar a evolução do Algoritmo Genético, e, assim, serão gerados mais dois gráficos sendo: um mostrando os indivíduos e seus respectivos *fitness* e outro que mostra os pólos de cada indivíduo $\frac{\lambda}{\delta}$ sobrepondo o Lugar das Raízes.

Além disso, no final de cada simulação, são automaticamente armazenados os dados da simulação em um arquivo de Excel nomeado “ResultadoSimulações.xlsx” e a cada simulação é preenchida uma linha neste arquivo. Também mostra-se as funções de transferência dadas pelas equações (3) e (4), que são as funções do controlador GPC para o caso de $hc=1$, no *workspace* do *MATLAB*.

Com isso, o usuário deve inserir as informações sobre a Função de Transferência da Planta, o GPC, sobre o algoritmo genético ou se desejar poderá realizar a simulação para apenas um valor de $\frac{\lambda}{\delta}$ escolhido ao avaliar o Lugar das Raízes. Desta forma, o usuário pode realizar o projeto de um controlador GPC ou de vários de forma consecutiva. Se realizado o projeto para apenas um controlador GPC, o usuário poderá ir na tabela de Excel e visualizar o resultado em formato de dados extraíndo apenas os que lhes interessam como, por exemplo, os valores de hp , hc , λ e δ do controlador GPC. Se realizado vários testes de forma consecutiva o arquivo em Excel conterà uma tabela com todos os testes realizados pois, cada linha é resultado de um teste. Com os dados desta tabela de Excel pode-se aproximar os parâmetros λ e δ do GPC em relação aos parâmetros da planta no formato de uma função multivariável.

A Figura B.1 mostra a parte da interface gráfica onde o usuário deverá inserir os dados sobre a planta em domínio de *Laplace*, e esta interface respeita o formato que o *MATLAB* opera, o que significa que as entradas devem ser no formato de vetor linha, o tempo de amostragem para que esta seja discretizada

usando método segurador de ordem zero e também a quantidade de amostras desejadas para a simulação. Ao se discretizar, é inserido um atraso adicional na planta discretizada, pois este modelo foi baseado no modelo proposto por CAMACHO e BORDONS (2000).

Figura B.1: Interface Gráfica, dados de entrada da planta

Dados da Planta:

K = [1] theta = [7.5]

Numerador = [1]

Denominador = [25 -1]

Tempo de Amostragem: [2.5] Quantidade de Amostras da Simulação: [150]

Fonte: próprio autor, 2017.

A Figura B.2 mostra a parte da interface gráfica referente aos dados do controlador GPC, sendo que os parâmetros h_p e h_c já assumem um valor mínimo de $d+1$ e 1, respectivamente. Nesta etapa, o usuário pode preencher o campo do h_p com diferentes valores para então avaliar a característica de cada lugar das raízes e escolher o que melhor atende a resolução de seu problema. Este procedimento foi utilizado neste trabalho para mostrar que quando o valor de h_p aumenta o sistema se torna mais robusto e estável. Os valores de $\frac{\lambda}{\delta}$ mínimo e máximo são usados caso o usuário queira utilizar o Algoritmo Genético para buscar o melhor caso avaliando sobressinal (OV) e máxima variação de sinal de controle ($\max(\Delta u)$), sendo que, o valor “lambda/delta fixo” é para o caso de desejar simular o sistema para apenas um valor e avaliar a resposta.

Figura B.2: Interface Gráfica, dados de entrada do GPC

Dados do GPC:

$h_p = d + 1 +$ [0]

$h_c = 1 +$ [0]

lambda/delta = faixa a ser escolhida avaliando Lugar das Raízes

lambda/delta minimo: [0.176]

lambda/delta maximo: [4.3*10³]

lambda/delta fixo: [0.104]

Fonte: próprio autor, 2017.

A Figura B.3 mostra os campos de entrada para o Algoritmo Genético, sendo: Repetições do AG a quantidade de vezes que o AG será reiniciado, Gerações do AG a quantidade de evoluções, População do AG a quantidade de indivíduos, OV e $\max(\Delta u)$ desejados, os quais são filtros para se buscar soluções apenas abaixo dos valores escolhidos. A Evolução do AG mostra os gráficos de como os indivíduos foram convergindo no Lugar das Raízes e nos valores de fitness dos mesmos.

Figura B.3: Interface Gráfica, dados de entrada do Algoritmo Genético

Fonte: próprio autor, 2017.

Os passos para se utilizar esta interface gráfica são:

Passo 1 - Dados sobre a planta em questão: inserir os dados da função de transferência da planta no formato de numerador, denominador e atraso, além do ganho da mesma forma que se é declarado no *MATLAB*.

Passo 2 - Parte dos dados do controlador de GPC: inserir os dados de h_p e de h_c do GPC. Neste passo é importante lembrar que deve-se começar escolhendo $h_p = d + 1 + 0$ e que o valor de h_c só vai influenciar nas respostas finais visto que, o lugar das raízes é gerado apenas para $h_c = 1$. Após inserir estes dados do Passo 1 e Passo 2 deve-se clicar no botão “Gerar Lugar das Raízes”.

Passo 3 - Análise de estabilidade do GPC: nesta parte do processo o usuário avaliará o gráfico do lugar das raízes gerado e então escolherá a região de $\frac{\lambda}{\delta}$

que será inserido no algoritmo genético para busca. Caso a característica do lugar das raízes seja ruim, como o caso de sempre instável, o usuário deverá retornar ao passo 2 e aumentar o valor de h_p .

Deve-se ressaltar que, sempre que se alterarem os valores dos passos 1, 2 e 3 obrigatoriamente é preciso clicar novamente no botão “Gerar Lugar das Raízes” antes de continuar os passos abaixo.

Passo 4 - Entradas do Algoritmo Genético: escolhido um intervalo de $\frac{\lambda}{\delta}$ no Passo 3 deve-se inserir estes nos campos do Algoritmo Genético e escolher a quantidade de iterações desejadas, tamanho da população e quantidade de gerações, os filtros de *overshoot* e variação do sinal de controle desejados. Após inserir os dados das partes 1, 2, 3 e 4 deve-se clicar no botão “Gerar Resposta por AG”. Caso quiser gerar respostas para apenas um valor de $\frac{\lambda}{\delta}$, o campo “lambda/delta fixo” deve ser preenchido e então clicar no botão “ Gerar Respostas por lambda/delta fixo”.

Passo 5 - Avaliar se os resultados atendem o especificado, pois o AG busca a resposta dentro dos filtros desejados, porém, pode ocorrer de não existir tal resposta e ainda assim o AG vai retornar o melhor caso que encontrou. Se atendida as necessidades, basta abrir a tabela de Excel e retirar os dados desejados deste projeto. Em caso negativo, deve-se realizar novamente o teste aumentando os valores de h_p do GPC ou os valores de gerações e tamanho da população do AG.

Apêndice C – Vantagem da escolha de uma boa população inicial

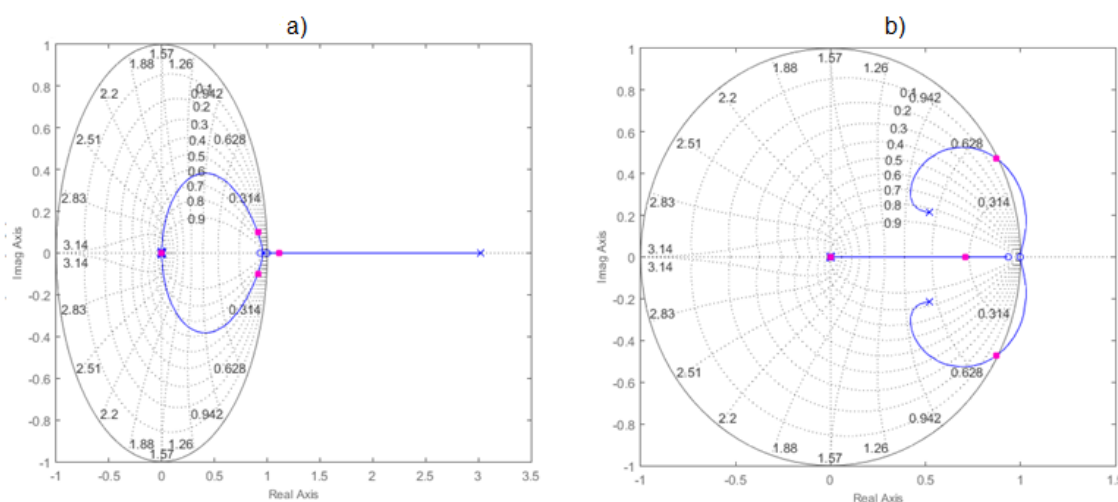
Como forma de exemplificar os benefícios da interface gráfica desenvolvida, nesta etapa será utilizada uma planta teórica instável com atraso e de fase não mínima (PIONTKOVSKY FILHO e SALLES, 2017). As funções de transferência contínua e discreta (utilizando um período de amostragem de 1 segundo e um segurador de ordem zero) são dadas, respectivamente, pelas Equações (46) e (47):

$$G(s) = \frac{s-1}{15s^2+s} e^{-3s} \quad (46)$$

$$G(z^{-1}) = z^{-4} \frac{0,03189 - 0,09638z^{-1}}{1 - 1,936z^{-1} + 0,9355z^{-2}} \quad (47)$$

Após avaliar a característica do lugar das raízes variando-se h_p , escolheu-se $h_p = 8$, por ter sido o menor valor no qual foi possível obter regiões que resultavam em sistemas estáveis e em uma boa região de amortecimento, conforme ilustra a Figura C.1 Para valores de h_p acima de 8 o lugar das raízes não apresentou regiões melhores, e, além disso, quanto maior o valor de h_p mais capacidade de processamento é exigida do *hardware* de implementação.

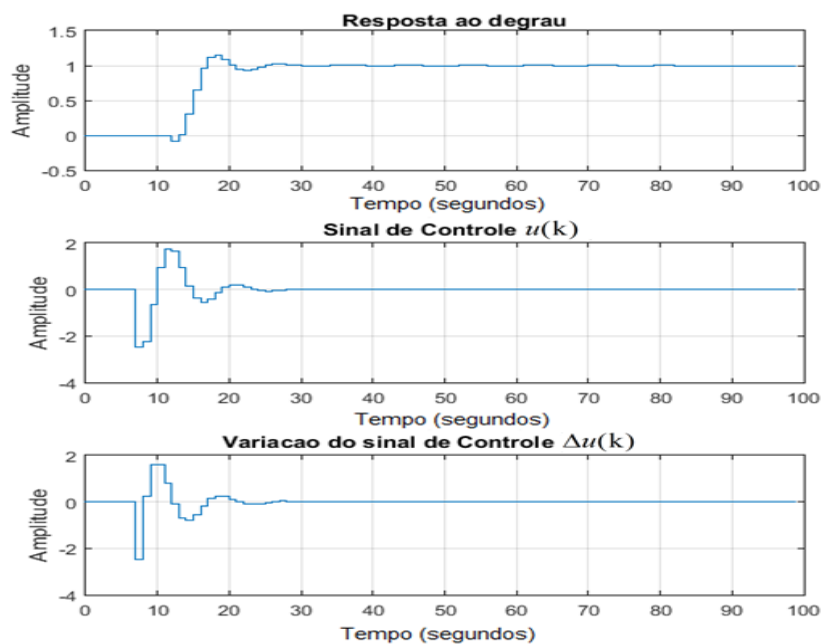
Figura C.1: Gráfico do Lugar das Raízes de $G(z^{-1})$ para os casos de a) $h_p = 5$ e b) $h_p = 8$



Fonte: próprio autor, 2017.

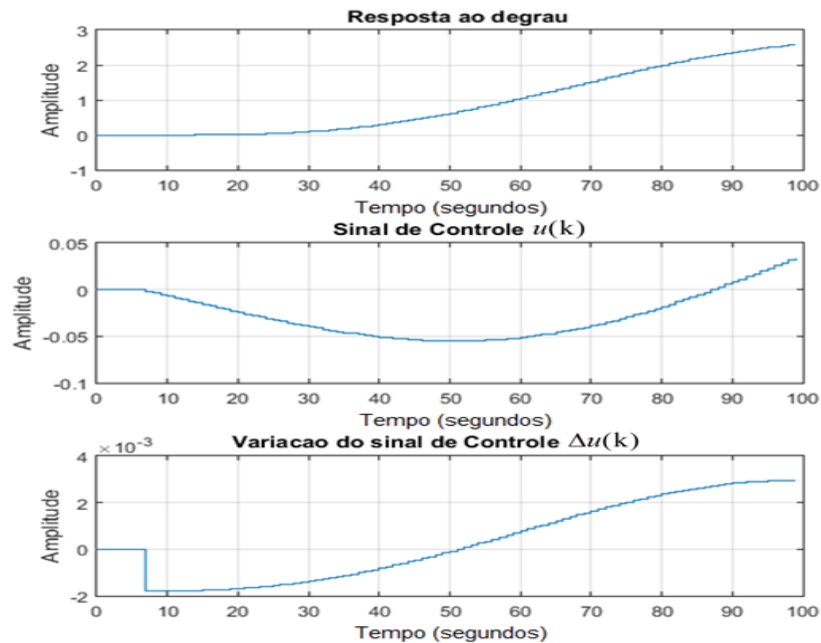
Foram realizados dois testes, variando-se os intervalos de $\frac{\lambda}{\delta}$. O Caso 1 abrangeu toda a região estável, com $\frac{\lambda}{\delta}$ de 0,0022 a 0,191. O Caso 2 utilizou a faixa máxima, isto é, de 0 até infinito (definido por 10^{20}). Estabeleceu-se, como valores desejados de OV e $\max(|\Delta u(k)|)$, 15% e 2,5 [V], respectivamente. Com relação ao AG, foram considerados uma população de 50 indivíduos, 5 gerações e 3 reinicializações. Esta seleção de valores demandou um tempo de processamento de aproximadamente 1 minuto para cada teste. Os resultados dos testes são apresentados, respectivamente, pela Figura C.2 e Figura C.3.

Figura C.2: Resposta encontrada pelo AG ao se escolher o intervalo de λ/δ de 0,0022 até 0,191 (Caso 1)



Fonte: próprio autor, 2017.

Figura C.3: Resposta encontrada pelo AG ao se escolher o intervalo de λ/δ de 0 até infinito (Caso 2)



Fonte: próprio autor, 2017.

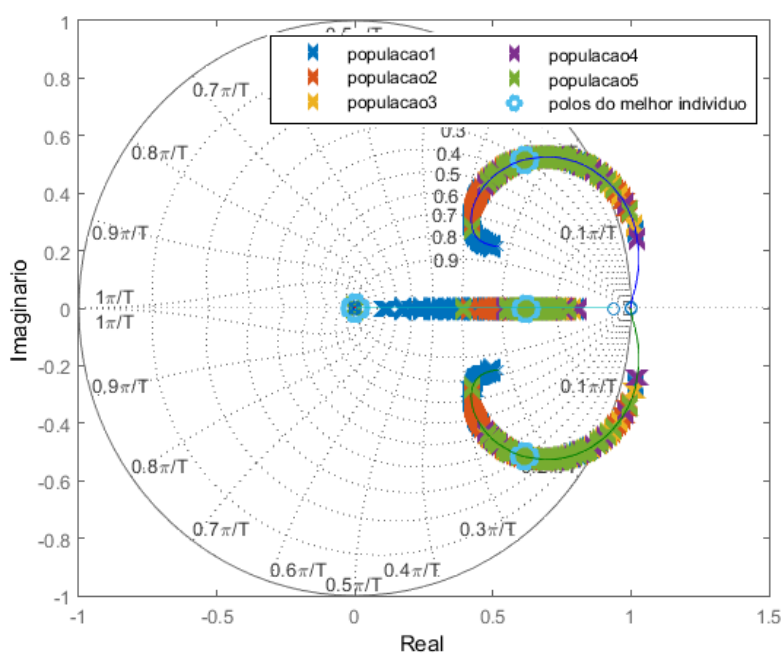
Observa-se que, para o Caso 1, foi obtida uma resposta com $OV= 14,97\%$, $t_s = 28$ [s] e $\max(|\Delta u(k)|) = 2,4$ [V], atendendo aos valores desejados de OV e $\max(|\Delta u(k)|)$. Já, para o Caso 2, foi obtida uma resposta com OV , t_s e $\max(|\Delta u(k)|)$ indeterminados, em razão de se tratar de um sistema instável.

Além disto, partindo-se da premissa de que é atribuída a população inicial do AG valores aleatórios que obedecem a uma determinada faixa de valores, ambos os testes foram repetidos cerca de 20 vezes. Os resultados do Caso 1 não apresentaram variações relevantes, ou seja, 100% dos testes realizados foram obtidas respostas próximas ao que mostra a Figura C.2, diferentemente do Caso 2, que em razão do intervalo de $\frac{\lambda}{\delta}$ ser amplo, apenas 10% dos testes apresentaram respostas aceitáveis.

A Figura C.4 mostra os pólos gerados pelos indivíduos $\frac{\lambda}{\delta}$ para o caso do intervalo escolhido apenas com a região estável. Nota-se que, cada geração é bem distribuída, mesmo que não esteja visível no gráfico já que, os polos das gerações vão se sobrepondo e o que ocorreu foi que a primeira geração abrange

boa parte da região estável e as próximas gerações vão convergindo para o melhor caso, até que um dos indivíduos da geração 5 se apresenta como o melhor caso. Alguns indivíduos ainda estão fora da faixa estável devido ao fato da propriedade de mutação do AG, além do detalhe mencionado no final da seção 2.2.4.

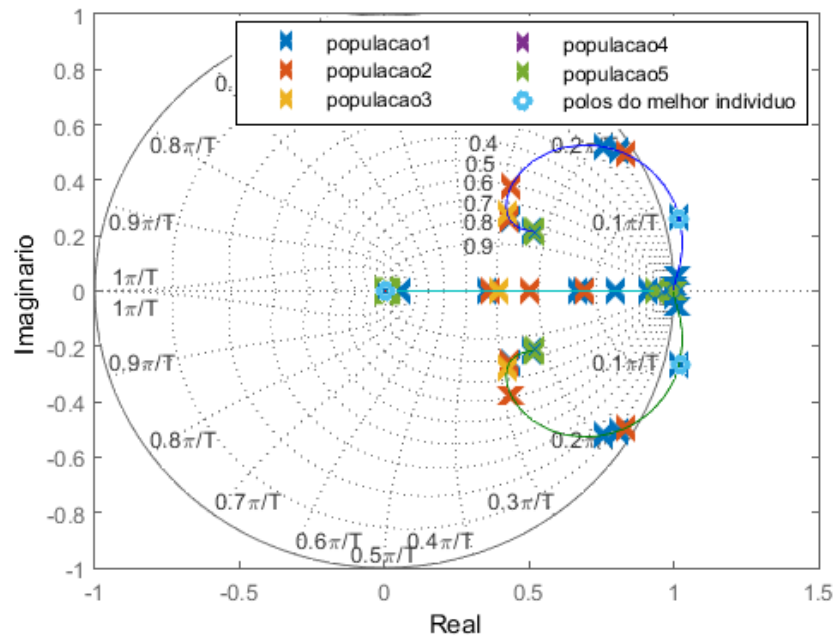
Figura C.4: Gráfico do Lugar das Raízes da função discreta mostrando a evolução do Algoritmo Genético para o intervalo de λ/δ de 0,0022 até 0,191



Fonte: próprio autor, 2017.

A Figura C.5 mostra os polos gerados pelos indivíduos $\frac{\lambda}{\delta}$ para o caso do intervalo escolhido de zero a infinito, onde os indivíduos estão concentrados em poucos pontos visto que, muitos dos indivíduos são criados próximos de um mesmo ponto, devido ao intervalo ser muito extenso. A escolha dos pólos instáveis como melhor indivíduo se deve ao fato de que nenhum outro pólo encontrado estar dentro das especificações e, assim, são considerados tão ruins quanto um polo instável.

Figura C.5: Gráfico do Lugar das Raízes da função discreta mostrando a evolução do Algoritmo Genético para o intervalo de λ/δ de 0 até infinito



Fonte: próprio autor, 2017.