

# CONTROLE DE ALTITUDE DE UM MOTOR-HÉLICE UTILIZANDO O ALGORITMO DE GOLUB-HOUSEHOLDER PARA MODELAGEM NÃO-LINEAR E ALGORITMOS GENÉTICOS PARA SINTONIA DO CONTROLADOR

Manuel A. Rendón<sup>1</sup>, Exuperry B. Costa<sup>2</sup>, Ana S. C. Alves<sup>3</sup>, Elias Vilas Boas<sup>4</sup>,  
Leonardo M. Honório<sup>5</sup> e André Marcato<sup>6</sup>

Grupo de Inteligência e Mobilidade - GrIM; Universidade Federal de Juiz de Fora - UFJF

1, 5 e 6 Professores do Departamento de Energia Elétrica (DENE) - UFJF

2, 3 e 4 Pós-Graduação em Engenharia Elétrica (PPEE) - UFJF

**Resumo** – O presente trabalho descreve o desenvolvimento, implementação e validação de um controle de altitude para um sistema motor-hélice, que faz parte de um projeto do GrIM: Veículo autônomo não-tripulado (UAV) quadrotor para monitoramento remoto. Para sintonia dos parâmetros do controlador foram utilizados *algoritmos genéticos* (GA) e para modelagem não-linear foi empregado o algoritmo de *Golub-Householder* [1]. Um banco de testes foi montado para operar o motor-hélice, com instrumentação apropriada para monitorar a altitude e comandar o motor através de uma placa Arduino [2] e um inversor eletrônico. Excitando o sistema em malha aberta foram obtidos dados válidos para modelagem. Utilizando estruturas ARX [3] foram calculados modelos lineares de primeira e segunda ordem, e utilizou-se o Toolbox *sisotool* de *MatLab* para sintonia dos controladores Proporcional-Integral (PI) e Proporcional-Integral-Derivativo (PID). Através do algoritmo de *Golub-Householder* [1] foi obtido um modelo não-linear com regressores de até terceira ordem. Utilizando o modelo não-linear para calcular o *Fitness*, os parâmetros do controle PID foram calculados utilizando GA. Serão apresentados os resultados do controle calculado a partir dos modelos lineares, e o controle obtido a partir do modelo não-linear e GA.

**Palavras Chave** – Algoritmos genéticos, identificação de sistemas, Golub-Householder, sintonia de controlador PID, quadrotor.

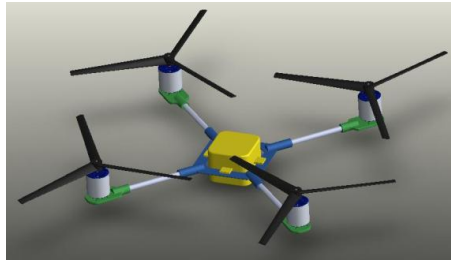
**Abstract** – Present work describes an altitude control development, implementation and validation, applied in motor-helix altitude control system, as part of a bigger GrIM Project: Quadrotor operating as unmanned autonomous vehicle (UAV) for remote monitoring. For control law's parameters tuning it was employed *genetic algorithms* (GA), and for non-linear modeling it was used the *Golub-Householder* algorithm [1]. A test bench was installed to operate the motor-helix, with proper instrumentation for monitoring altitude and command the helix motor through an Arduino [2] system and electronic inverter. With the system excited in open loop, valid data for modeling was obtained. Using ARX [3] structures first and second order linear models were calculated, using a *MatLab* Toolbox called *sisotool* for adjusting Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) compensators. A non-linear model was calculated employing the *Golub-Householder* [1] algorithm, limiting the regressors order in three. The PID compensator parameters were tuned with GA, and the non-linear model previously identified was used to calculate the GA *Fitness*. Performance of the compensator calculated using linear models will be related, as well as the compensator obtained using the non-linear models and GA.

**Keywords** – Genetic algorithms, systems identification, Golub-Householder, PID compensator tuning, quadrotor.

## 1 Introdução

Um helicóptero é uma aeronave movimentada por hélice que para orientar o deslocamento utiliza acionamentos e engrenagens mecânicos que mudam a inclinação da hélice em relação ao resto do corpo [4]. Para manter operação confiável é necessária manutenção rigorosa e contínua nesses sistemas, o que limita o uso de tal aeronave para aplicações que priorizam simplicidade e confiabilidade tais como UAV. Os helicópteros não tripulados sofrem por causa disto, pois os acoplamentos mecânicos aumentam a complexidade diminuindo a confiabilidade do sistema do sistema e aumentando o custo final. Uma alternativa em termos práticos são os quadrotores, que são aeronaves com formato em X com quatro motores com hélices independentes montados fixos nos extremos (Figura 1). As manobras não dependem de peças móveis, mas da diferença de rotação nos motores em suas extremidades. Devido à configuração robusta e simples, enxerga-se nos quadrotores grande potencial para aplicações autônomas, e objetiva-se torná-los mais competitivos não apenas em custo, mas em desempenho dinâmico e capacidade aerodinâmica. Existem atualmente no mercado quadrotores comerciais que conseguem curtos tempos de voo e pouca capacidade de carga [5].

O estudo de quadrotores para aplicações em robótica teve seu início no ano 2000, e o controle de estabilidade, posição, e dinâmica tem requerido alguns estudos [6]. O quadrotor tem o comportamento dinâmico de um robô em 6 graus de liberdade (distância nos eixos X, Y e Z com relação a algum ponto de referência e os três ângulos de orientação com relação a esses mesmos eixos), podendo se mover holonomicamente, isto é, sem restrições nas direções e velocidades [7]. O presente estudo pretende desenvolver o controle de altitude de um motor-hélice do quadrotor, limitado pela estrutura do banco de ensaios a um grau de liberdade (pode-se movimentar só no eixo Z). A Figura 1 apresenta um desenho de um quadrotor.



**Figura 1** – Estrutura do Quadrotor

Para projetar o controlador do motor-hélice o estudo foi dividido em dois: Obter o modelo do motor hélice operando no banco de testes e calcular os parâmetros do controlador.

Um dos principais desafios em projeto de sistemas de controle é ajustar os parâmetros da lei de controle, especialmente quando a planta apresenta uma dinâmica não-linear. Utilizar algoritmos genéticos para sintonizar o controlador PID, em alguns casos on-line, tem apresentado bons resultados [8, 9, 10]. Algoritmos genéticos são conhecidos por ter bom desempenho em otimizar sistemas complexos. São métodos de procura global baseados numa metáfora da evolução natural, e partem de um conjunto de soluções candidatas a um determinado problema que evoluem utilizando operadores genéticos artificiais tais como reprodução, cruzamento e mutação. Trabalham combinando o princípio Darwiniano da ‘supervivência do mais forte’ com a estratégia da troca de informação probabilística inspirada no processo da genética natural. Essa combinação forma um algoritmo de procura estruturado ainda que randômico, que promete ser capaz de identificar soluções ótimas ou perto do ótimo a problemas de otimização [8]. Algoritmos genéticos têm sido utilizados para ajustar os parâmetros de um controlador PID, quando opera como servo e como regulador minimizando a degradação no desempenho, pois cada tipo de operação requer um ajuste diferente [9]. Outra forma de utilizar algoritmos genéticos em projeto de controladores PID é para resolver o problema de otimização multiobjetivo, aplicável a um sistema com várias malhas de controle [10].

Estudos de identificação de sistemas têm sido desenvolvidos estruturas do tipo ARX, ARMAX e Box-Jenkins [11], e não-lineares com algoritmos recursivos como o de Golub-Householder [1,12]. Essas ferramentas complementadas com procedimentos apropriados de análise, testes, amostragem, cálculos e validação apresentam-se muito úteis para o desenvolvimento de sistemas de controle. No presente estudo foi utilizada a ferramenta de Ziegler-Nichols para sintonia dos controladores PI e PID na abordagem linear, e algoritmos genéticos para sintonia do controle PID no modelo não-linear.

## 2 Detalhamento do Estudo

Para desenvolver o controle de altitude do motor-hélice foram seguidos os seguintes passos:

- Projeto e montagem do banco de ensaios, definição dos testes em malha aberta e amostragem dos dados.
- Obtenção da característica estática do modelo, e análise da linearidade.
- Cálculo e validação dos modelos lineares ARX de primeira e segunda ordem.
- Cálculo do controle PI e PID baseados no modelo linear. Implementação, testes e validação.
- Definição de testes malha aberta em subida e descida, amostragem dos dados.
- Obtenção da característica estática do modelo e análise de linearidade.
- Cálculo do modelo não-linear utilizando o algoritmo de Golub-Householder [3,10].
- Cálculo do controle PID com o modelo não-linear e algoritmos genéticos. Implementação, testes e validação.

## 3 Descrição do Banco de Ensaios

O banco de ensaios foi projetado com um dos quatro motores-hélice, para permitir somente o movimento vertical, limitando os outros cinco graus de liberdade (Figura 2).



(a)



(b)

**Figura 2** – (a) Banco de ensaios; (b) inversor eletrônico, motor, sensor laser, hélice, bateria e sistema Arduino

O banco de ensaios utilizado foi montado com os seguintes componentes:

- Motor brushless acoplado a uma hélice de passo 9x4,7.
- Circuito inversor eletrônico para alimentação do motor com ESC (electronic speed controller).
- Bateria de Li-Po (polímero de lítio).
- Estrutura metálica para o carrinho.
- Sensor de altitude por laser.
- Placa de desenvolvimento ATMEL Arduino 2560.
- Computador com software de configuração do Arduino e conexão USB.

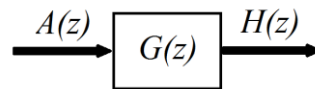
O comando de velocidade do motor é enviado na forma de um número inteiro desde o computador ao sistema Arduino através de uma porta USB. A plataforma Arduino [2] gera um sinal PWM (modulação por largura de pulso) proporcional ao comando. O circuito inversor ESC, alimentado pela bateria, recebe o sinal PWM e gera uma tensão alternada trifásica com frequência proporcional ao ciclo ativo do PWM. A variável de entrada do sistema é um número inteiro enviado do computador, e a saída é a altitude do motor-hélice medida com um sensor laser. Esse sinal é amostrado por um conversor A/D na placa Arduino. No software de configuração do Arduino é configurado o armazenamento dos dados.

## 4 Desenvolvimento do Controlador Baseado em Modelos Lineares

Para desenvolver o controlador foram definidos testes em malha aberta para identificar o modelo e em malha fechada para validar o controle. Testes em malha aberta são efetuados enviando comandos de aceleração ao sistema e medindo a saída (altitude) que varia livremente sem estar sujeita a nenhuma malha de realimentação. Testes em malha fechada são realizados enviando um valor de altitude de referência ao sistema, e observando como o sistema manipula o comando de aceleração para atingir a referência atuando de forma automática sujeito à realimentação pela lei de controle nele implementada.

### 4.1 Definição dos Testes em Malha Aberta

Utilizando o banco de ensaios foram executados testes em malha aberta e os dados foram amostrados com tempo de amostragem de 300 ms. Os testes em malha aberta excitam o conjunto motor-hélice com valores do comando de velocidade  $A(z)$ , enquanto que a variação da altitude  $H(z)$  é amostrada. A Figura 3 apresenta um esquema do sistema em malha aberta, onde  $G(z)$  representa a função de transferência em tempo discreto que deve ser identificada.



**Figura 3:** Representação do sistema motor-hélice em malha aberta

A variação no comando de velocidade foi em degraus ascendentes, para identificar a região de operação no banco de ensaios. O comando e a altitude são amostrados através de um software que se comunica com a placa Arduino.

Foi identificado que a faixa na qual gira o motor varia de 55 a 130, e que com valores menores a 87 o empuxo da hélice é insuficiente para erguer o sistema. Foram executados quatro ensaios conforme indica a Tabela 1, onde cada coluna (**a**, **b**, **c** e **d**) corresponde a um ensaio.

**Tabela 1** – Tempos e valores de comando de velocidade nos testes em malha aberta

Tempo (s)	Comando (a)	Comando (b)	Comando (c)	Comando (d)
0	65	76	87	87
8	70	79	88	88
16	75	82	89	89
24	80	85	90	90
32	85	88	91	91
40	90	91	92	92
48	95	94	93	93
56				94

### 4.2 Cálculo dos Modelos Lineares ARX

Foram utilizados comandos do MatLab para calcular modelos ARX de primeira e segunda ordem utilizando os dados amostrados [14].

O modelo de primeira ordem calculado como função de transferência em  $z$  foi:

$$G(z) = \frac{H(z)}{A(z)} = \frac{0,941}{1 - 0,9288z^{-1}} \quad (1)$$

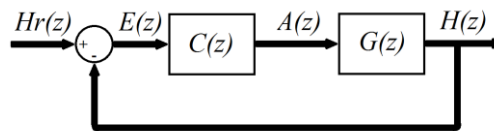
O modelo de segunda ordem calculado como função de transferência em  $z$  foi:

$$G(z) = \frac{H(z)}{A(z)} = \frac{-0,03508 + 0,7313z^{-1}}{1 - 1,371z^{-1} + 0,4249z^{-2}} \quad (2)$$

Os modelos correspondentes às equações (1) e (2) foram simulados e os valores da simulação foram comparados com dados reais para validação. A validação em ambos os casos foi pouco satisfatória. Para validação foi calculada a raiz quadrada do erro quadrático médio (*RMSE*) [15]. A validação *Fit RMSE* foi respectivamente de 71,56% e 71,21%.

### 4.3 Sintonia de Parâmetros dos Controladores PI e PID

A Figura 4 representa o sistema de controle em malha fechada, onde a entrada é a altitude desejada ou de referência  $Hr(z)$ . A diferença entre a altitude de referência e a altitude real medida pelo sensor laser é o erro  $E(z)$ , entrada no bloco controlador expresso na forma de função de transferência em tempo discreto  $C(z)$ .



**Figura 4** – Esquema do sistema do controle em malha fechada

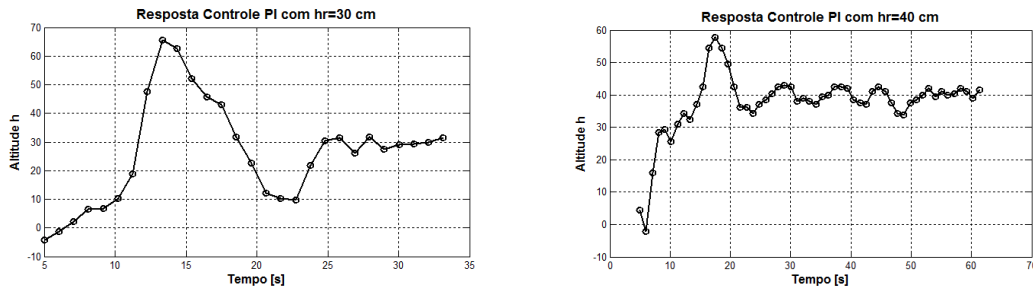
Utilizando o modelo linear calculado e o Toolbox *sisotool* do MATLAB foram calculados os parâmetros dos Controladores PI e PID, apresentados nas equações (3) e (5). Para sintonia de parâmetros foi escolhido Ziegler-Nichols de malha aberta [16,17]. A função de transferência no tempo discreto calculada para o controlador PI foi:

$$C(z) = \frac{A(z)}{E(z)} = \frac{0,0387 + 0,005144z^{-1}}{1 - z^{-1}} \quad (3)$$

Para implementar a lei de controle na placa Arduino a equação (3) é expressa como equação de diferenças:

$$a(k) = 0,0387 \cdot e(k) + 0,005144 \cdot e(k - 1) + a(k - 1) \quad (4)$$

O sistema foi operado em malha fechada no banco de ensaios com a lei de controle calculada, e os dados de altitude foram amostrados com tempo de amostragem de 300 ms. A Figura 5 mostra os gráficos da resposta do sistema com controle PI, desde o repouso, com altitude de referência  $Hr(z)$  de 30 e 40 cm.



**Figura 5** – Resposta do sistema operando com controle PI com  $Hr$  de 30 e 40 cm

Pode-se notar um valor significativo de máximo sobre-sinal, e um longo tempo de acomodação.

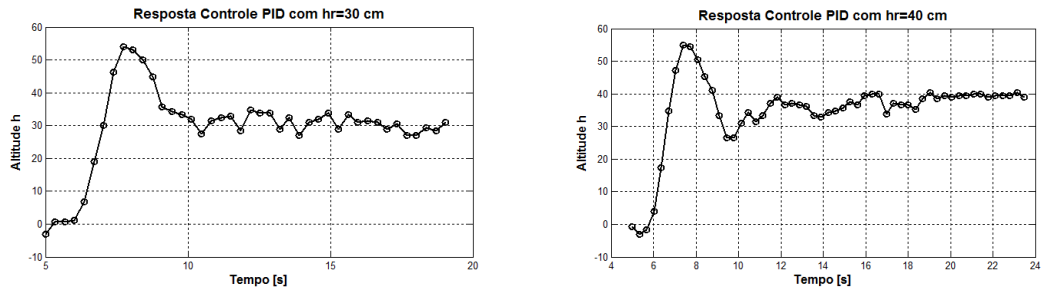
O cálculo foi feito novamente com *sisotool* de MatLab e foi obtido o controlador PID no tempo discreto.

$$C(z) = \frac{A(z)}{E(z)} = \frac{0,119 - 0,132z^{-1} + 0,028z^{-2}}{1 - z^{-1}} \quad (5)$$

A representação da equação (5) na forma de equação de diferenças:

$$a(k) = 0,119 \cdot e(k) - 0,132 \cdot e(k - 1) + 0,028 \cdot e(k - 2) + a(k - 1) \quad (6)$$

A Figura 6 mostra as respostas do sistema com controlador PID para altitude  $Hr$  de 30 e 40 cm.



**Figura 6** – Resposta do sistema operando com controle PID para degraus de altura  $Hr$  de 30 e 40  $cm$

Pode-se verificar uma diminuição no tempo de acomodação e no máximo sobre-sinal com relação ao anterior, indicando um melhor desempenho do controlador PID comparado com o controlador PI.

## 5 Desenvolvimento do Controlador Baseado em Modelos Não-Lineares

### 5.1 Definição dos Testes em Subida e Descida

Um segundo conjunto de testes foi executado para obter dados dinâmicos tanto em subida como em descida do sistema motor-hélice, e obter dados que permitam calcular um modelo que represente a completa dinâmica do processo (Tabela 2). Nessa Tabela cada coluna (a e b) corresponde a um ensaio.

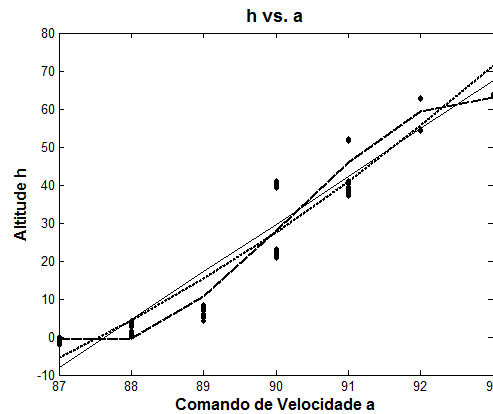
**Tabela 2** – Valores de comando de velocidade utilizados no segundo grupo de testes

Tempo (s)	Comando (a)	Comando (b)	Tempo (s)	Comando (a)	Comando (b)
0	84	84			
8	85	85	80	92	92
16	86	86	88	91	91
24	87	87	96	90	90
32	88	88	104	89	89
40	89	89	112	88	88
48	90	90	120	87	87
56	91	91	128	86	86
64	92	92	136	85	85
72	93	93	144	84	84

Novamente observou-se que o motor-hélice só decolar com comando de velocidade maior do que 87.

### 5.2 Cálculo da Característica Estática do Sistema

Utilizando os dados em regime foi calculada a característica estática aproximada por funções de primeira, segunda e terceira ordem. A Figura 7 apresenta graficamente as características estáticas do sistema.



**Figura 7** Característica estática do sistema

No gráfico os pontos representam os dados reais em regime, obtidos nos testes em subida e descida do motor-hélice. A linha contínua corresponde à característica linear, a linha de pontos à de segunda ordem e a tracejada à de terceira ordem. Os valores de validação foram: 70,49 %, 71,56 % e 78,65 %. Observou-se uma característica estática de terceira ordem.

### 5.3 Cálculo do Modelo Não-Linear Utilizando o Algoritmo de Golub-Householder

Para calcular o modelo não-linear foi utilizada uma rotina desenvolvida em um estudo anterior [1]. A ferramenta implementa o algoritmo de Golub-Householder em uma rotina de MatLab [12]. A matriz de regressores candidatos é montada baseada nos resultados da característica estática. Para validação foram utilizados duas técnicas: RMSE [15] e critério de Akaike (AIC) [18]. A Figura 8 mostra os gráficos dos modelos com melhor validação (7, 13 e 15 regressores).

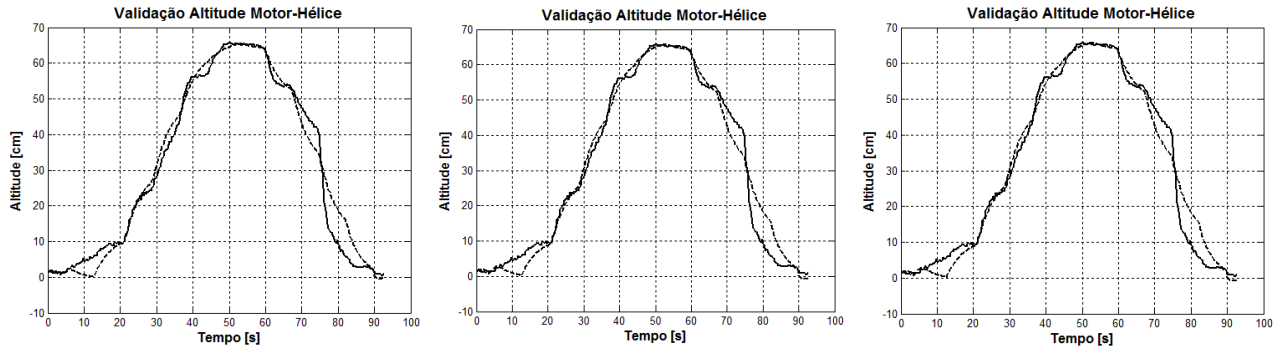


Figura 8 – Validação gráfica dos modelos não-lineares obtidos utilizando o algoritmo de Golub-Householder

Tabela 3 – Validação dos modelos não-lineares

No. de Regressores	RMSE [%]	AIC
7	84,85	2,65
13	85,39	2,62
15	85,41	2,63

Observa-se nos resultados que a melhora na validação com 13 e 15 regressores foi desprezível, não justificando um modelo mais complexo. Isto é verificado visualmente nos gráficos da Figura 8. A Tabela 4 apresenta os valores calculados do modelo com 7 regressores.

Tabela 4 – Parâmetros calculados no modelo com 7 regressores

Regressor	Parâmetro
$y(k-1)$	1,1640
$y(k-3)$	-0,2004
$u(k-1)$	0,3348
$u(k-1)u(k-3)^2$	-0,0374
$u(k-2)$	0,4013
$u(k-3)^2$	0,0829
$u(k-1)^2$	-0,0718

O modelo não-linear expresso na forma de equação de diferenças.

$$y(k) = 1,1640y(k-1) - 0,2004y(k-3) + 0,3348u(k-1) - 0,0374u(k-1)u(k-3)^2 + 0,4013u(k-2) + 0,0829u(k-3)^2 - 0,0718u(k-1)^2 \quad (7)$$

### 5.4 Sintonia do Controlador PID Utilizando Algoritmos Genéticos

Foi desenvolvida uma rotina em MatLab para estimar os parâmetros do controle PID utilizando algoritmos genéticos [19]. A função de transferência do controlador PID no tempo discreto é:

$$C(z) = K_p + \frac{K_i}{(1-z^{-1})} + K_d(1-z^{-1}) \quad (8)$$

Os três parâmetros do controlador ( $K_p$ ,  $K_i$  e  $K_d$ ) formam um indivíduo, (chamado de cromossomo), e são conhecidos como os genes desse indivíduo. Partindo de um indivíduo inicial, é criada uma primeira geração utilizando ferramentas de mutação e cruzamento. O indivíduo inicial é formado pelos parâmetros calculados anteriormente com o modelo linear. A função de transferência do controlador PID (5) é rearranjada para mostrar os valores dos parâmetros ( $K_p = 0,028$ ,  $K_i = 0,076$  e  $K_d = 0,015$ ).

$$C(z) = \frac{A(z)}{E(z)} = 0,028 + \frac{0,076}{1-z^{-1}} + 0,015(1-z^{-1}) \quad (9)$$

Para selecionar indivíduos e criar uma nova geração o algoritmo utiliza ferramentas de seleção natural, mutação e cruzamento [19]. Em cada geração os indivíduos da população são testados um a um pelo algoritmo utilizando uma função de

otimização (*Fitness*), que deverá minimizar a função objetivo. Como função de otimização foi usada a raiz quadrada do somatório dos erros quadráticos. O erro em questão é a diferença entre o degrau da altitude (*Hr*) e o valor de altitude calculado em cada instante (*H*). Para calcular a altitude foi utilizado o modelo não-linear estimado anteriormente (7).

No algoritmo genético cada nova geração é formada por uma população de indivíduos com menor valor de função de otimização. Quando o algoritmo pára, da última geração é escolhido o indivíduo de melhor validação (menor *Fitness*). Nesse estudo foi definida uma população de 50 indivíduos e foram calculadas 100 gerações. O gap de geração empregado foi de 0,1.

A Tabela 5 apresenta os valores do primeiro individuo e do individuo final escolhido utilizando algoritmos genéticos, junto com os valores de *Fitness*.

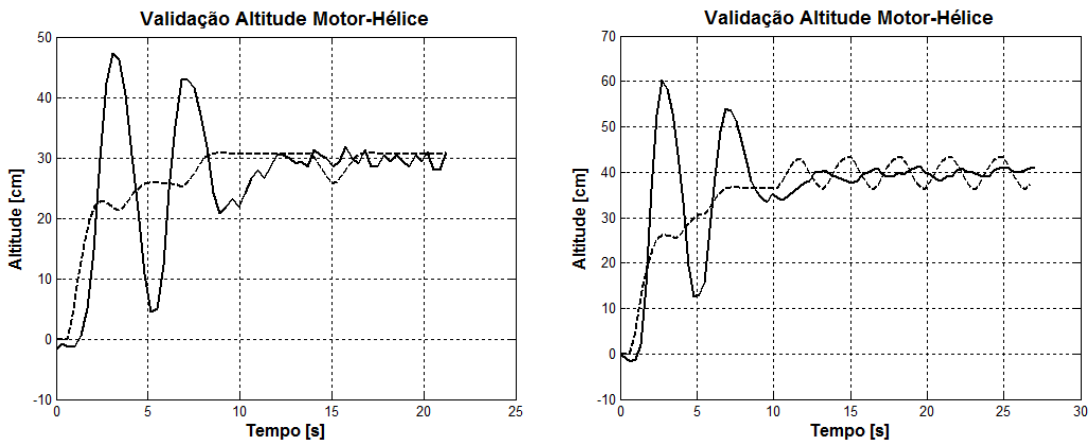
**Tabela 5 – Parâmetros do controlador PID**

	<b>Calculado com o Modelo Linear e o Critério de Ziegler-Nichols de Malha Aberta</b>	<b>Calculado com o Modelo Não-Linear e Algoritmos Genéticos</b>
$K_p$	0,028	0,1026
$K_i$	0,076	0,0453
$K_d$	0,015	0,1424
<b>Fitness</b>	115,67	104,49

Observa-se que utilizando algoritmos genéticos foi possível achar parâmetros com melhor desempenho em malha fechada.

### 5.5 Implementação da Lei de Controle e Validação do Desempenho do Controlador PID

Os parâmetros do controlador PID obtidos com algoritmos genéticos foram implementados no banco de ensaios e o sistema foi operado em malha fechada. Os valores de altitude foram amostrados, e na Figura 9 mostram-se os sinais reais (linha contínua), e simulados (linha tracejada), do sistema para dois degraus de altitude *Hr*: 30 e 40 cm.



**Figura 9 – Comparação da resposta real e simulada do sistema operando em malha fechada**

A Tabela 6 apresenta os valores de *Fitness* do sistema em malha fechada com controlador PID ajustado com algoritmos genéticos. A primeira coluna apresenta o valor calculado com simulação do modelo não-linear, a segunda o valor calculado com dados reais amostrados. Na mesma Tabela o Degrau de Altitude corresponde a um comando ingressado no computador, que indica ao sistema a altitude a ser atingida pelo motor-hélice.

**Tabela 6 – Função de otimização dos controladores PID**

<b>Degrau de Altitude [cm]</b>	<b>Fitness estimado com o modelo</b>	<b>Fitness calculado com dados reais</b>
30	75,66	97,31
40	110,38	116,23

## 6 Conclusões

O desenvolvimento de um sistema autônomo para um quadrotor não é trivial, e por isso foi montado um banco de ensaios para estudar a dinâmica de um motor-hélice limitado a um grau de liberdade, ou seja que se pode movimentar só verticalmente, na direção do eixo Z. Analisando com detalhe a dinâmica desse sistema facilita a compreensão e a análise de sistemas mais complexos como o quadrotor. O banco de ensaios mostrou enorme utilidade facilitando a tarefa de executar testes, amostrar dados, e validar leis de controle.



No presente trabalho o desempenho do modelo linear identificado da planta se mostrou pouco preciso, e o controle em malha fechada tentou compensar esse erro na aproximação.

Algumas características do banco de ensaios foram relevantes, afetando a linearidade do sistema e afastando-o de uma condição invariante no tempo: carga da bateria, efeitos de borda que fazem o empuxo variável com a altura, etc. Esse último é provavelmente a maior fonte de não linearidade. Isto explica a melhor validação do sistema identificado não-linear, e a necessidade de utilizar técnicas de inteligência artificial para sintonizar o controlador. O modelo não-linear demonstrou um melhor desempenho e o algoritmo de Golub-Householder mostrou ser uma ferramenta útil para obter modelos não-lineares em sinais amostrados, orientados a desenvolvimento de sistemas de controle.

O fato do comando de velocidade  $A(z)$  estar limitado a valores inteiros dificultou em alguns casos que o sistema atinja uma condição aceitável de equilíbrio. Nas Figuras 5, 6 e 9 pode-se observar isto com clareza. Os testes em malha aberta mostraram que uma variação mínima do comando  $A(z)$  produz uma variação média de altitude de 10 cm. No sistema operando em malha fechada isto influencia no tempo de acomodação do sistema, que demora em atingir a altura desejada. Essa característica foi calculada na simulação como se observa no valor oscilatório do sinal de altitude na Figura 11.

A sintonia de parâmetros do controlador PID baseado em métodos de otimização, neste caso Algoritmos Genéticos, tem demonstrado ser uma ferramenta útil, de fácil entendimento e implementação. O fator crítico mostrou ser a escolha do ponto inicial da busca, e a decisão do tamanho da região de busca do algoritmo. Outro ponto importante é a decisão do Fitness, que foi escolhido ser a raiz quadrada do somatório do erro ao quadrado, mas que poderia ser o tempo de acomodação, ou o máximo sobre-sinal, etc. A escolha depende do critério de desempenho desejado, focado na aplicação final do sistema. Foram testados vários parâmetros nos algoritmos genéticos, variando o gap de geração, o número de indivíduos e de gerações, mas não foram observadas melhorias significativas nos resultados.

O método computacional utilizado nesta otimização teve leve custo computacional, e boa convergência, e mostra-se um caminho promissor para análise dinâmica e desenvolvimento em controle de quadrotoros.

## 7 Referências

- [1] Rendón, M. “Desenvolvimento de Modelos Matemáticos Representativos do Comportamento Dinâmico de Micro-Turbinas para a Conexão na Rede Elétrica”, Tese de Doutorado, UNIFEI, 2010.
- [2] Open source electronic prototyping platform - Arduino home page, <http://www.arduino.cc/> último acesso Maio de 2011.
- [3] Aguirre, L.A. “Introdução à Identificação de Sistemas”, Editora UFMG, 2004.
- [4] Hull, David G., “Fundamentals of Airplane Flight Mechanics”, Editora Springer, 1ra Edição, 2007.
- [5] Draganfly Innovations Inc, (2009), <http://www.draganfly.com/>.
- [6] Pounds, P.; Mahony, R.; Corke, P.; “Modeling and control of a large quadrotor robot”; ELSEVIER Control Engineering Practice 18 (2010) pp. 691-699.
- [7] Spong, Mark W. “Robot Modeling and Control”. John Wiley and Sons, Inc. (2005).
- [8] Vlachos, C. “Solution to the Shell standard control problem using genetically tuned PID controllers”, ELSEVIER Control Engineering Practice 10 (2002) pp. 151-163.
- [9] Arrieta, O. “PID autotuning for weighted servo/regulation control operation”, ELSEVIER Journal of Process Control 20 (2010) pp. 472-480.
- [10] Herreros, A. “Design of PID-type controllers using multiobjective genetic algorithms”, ISA Transactions 41 (2002) pp. 457-472.
- [11] Rendón, Manuel A. “Modelagem e Simulação do Sistema de Controle de uma Micro-Turbina a Gás”, Dissertação de Mestrado, UNIFEI, 2005.
- [12] Rendón, Manuel A., Goes, Luiz C., Médici, Mairum. “Modelagem Dinâmica Não-Linear de uma Micro-Turbina a Gás Utilizando o Algoritmo de Golub-Householder”, V CONEM, 2008, Salvador, BA.
- [13] Nelles, O. “Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models”, Springer, Alemanha, 2001.
- [14] Hanselman, D. “MatLab 5 - Guia do Usuário”, Editora Makron 1999.
- [15] Stark, L. “Telerobotics: Display, control, and communication problems”, IEEE Journal of Robotics and Automation, Volume 3, Pages: 67-75, 1987.
- [16] Ogata, K., “Discrete-Time Control Systems”, Editora Prentice Hall, 2a. Edição, 1995.
- [17] Phillips, N; Nagle, H. “Digital Control System Analysis and Design”, Editora Prentice Hall, 3a. Edição, 1994.
- [18] Akaike, H. “Information theory as an extension of the maximum likelihood principle”, Petrov and F. Csaki, editors. Second International Symposium on Information Theory. Akademiai Kiado, 1973, Budapest, Hungary.
- [19] Porter, B. “Genetic tuning of digital PID controllers”. IET Electronics Letters, Volume 28, Pages: 843-844, 1992.