

GPF-CONTROL: UM MODELO FUZZY-GENÉTICO PARA PROBLEMAS DE CONTROLE

ADRIANO S. KOSHIYAMA, MARLEY M.B.R. VELLASCO, RICARDO TANSCHKEIT.

*Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, n.225, Gávea, Rio de Janeiro, CEP.: 22451-900
E-mails: [adriano,marley,ricardo]@ele.puc-rio.br*

Abstract— This work presents a Genetic-Fuzzy Controller, called Genetic Programming Fuzzy for Control (GPF-Control), since it is based on Multi-Gene Genetic Programming. The main characteristics and distinctions of this approach are displayed when compared to traditional Fuzzy Logic Controllers. Two benchmarks application of GPF-Control are considered: the Cart-Centering Problem and the Inverted Pendulum. The results, when compared to other Genetic-Fuzzy Controllers in the literature, demonstrate the superiority and potentialities of GPF-Control.

Keywords— Fuzzy Control, Evolvable Systems, Hybrid Systems, Muti-Gene Genetic Programming.

Resumo— Este trabalho apresenta um Controlador Fuzzy-Genético, denominado *Genetic Programming Fuzzy for Control* (GPF-Control), devido a este ser baseado em Programação Genética Multigênica. São apresentadas as principais características e distinções desse controlador em relação ao Controlador Fuzzy tradicional. São exibidas duas aplicações benchmarks, a dizer: o problema da parada ótima (cart-centering) e o pêndulo invertido. Os resultados, quando comparados a outros Controladores Fuzzy-Genéticos elaborados na literatura, demonstram a superioridade e potencialidades do GPF-Control.

Palavras-chave— Controle Fuzzy, Sistemas Evolutivos, Sistemas Híbridos, Programação Genética Multigênica.

1 Introdução

Controladores elaborados por Lógica Fuzzy (Mendel, 1995; Tanscheit et al., 2007) tem sido usados como uma alternativa viável para descrever e manipular sistemas complexos, cuja solução tradicional é difícil ou a precisão resultante desta é abaixo do tolerável. Estes controladores se caracterizam pela capacidade de modelar processos a partir de regras fuzzy do tipo “se-então”, extraídas comumente do operador da planta. As abordagens clássicas evocam o uso de Sistemas de Inferência Fuzzy (SIF) do tipo Mamdani (Elmas et al., 2009; Córdon, 2011). Contudo, com o objetivo de atingir maior precisão e agilidade, controladores fuzzy do tipo Takagi-Sugeno-Kang tem sido recentemente explorados com maior ênfase (Jang et al., 1997; Precup e Hellendoorn, 2011).

No entanto, um especialista, que pode não existir, é necessário para a estimação dos parâmetros do SIF (Jang et al., 1997). Uma solução é fazer uso de um método de aprendizagem automático, que, ao entrar em contato com a planta, efetue a extração dos parâmetros para a formação da base de conhecimento do SIF. Dentre diferentes abordagens, destacam-se os Sistemas Fuzzy-Genéticos (Córdon et al., 2004; Córdon, 2011).

Sistemas Fuzzy-Genéticos unem a capacidade de inferência linguística dos SIF com a aprendizagem automática da meta-heurística genética. A principal abordagem encontrada na literatura faz uso de Algoritmos Genéticos para elaborar controladores fuzzy, seja ajustando as funções de pertinência (Thrft, 1991; Liu et al., 2001), como elaborando a base de regras fuzzy (Herrera et al., 1998; Pal e Pal, 2003). Trabalhos recentes têm explorado o uso da Programação Genética para a elaboração de controladores fuzzy, a partir de metodologias e objetivos semelhan-

tes aos já usados em Algoritmos Genéticos (Tunstel e Jamshidi, 1996; Hoffmann e Nelles, 2001; Tsakonas, 2013).

Em geral, é um aspecto positivo o emprego da meta-heurística genética apenas para buscar os parâmetros do SIF, ou seja, ela é usada como um instrumento do SIF, e não como um meio que possa alterar seu modo de raciocínio. Por outro lado, em arquiteturas com maior hibridização, em que a meta-heurística genética possui maior participação, é possível obter ganhos em acurácia e gerar variações na interpretabilidade que podem ser significativos ao projetista do SIF. Um exemplo são os modelos Neuro-Fuzzy (Kasabov e Song, 2002; Contreras et al., 2011), onde a Rede Neural possibilita auferir ganhos em precisão por possuir maior espaço na estrutura híbrida.

O objetivo deste trabalho é desenvolver um controlador fuzzy-genético, denominado de *Genetic Programming Fuzzy for Control* (GPF-Control), usando a Programação Genética Multigênica (Hinchliffe et al., 1996; Searson et al., 2007) como método de extração de conhecimento da planta. Busca-se com esse modelo uma arquitetura distinta de um SIF tradicional, mas com o objetivo de: (1) ajuste automático do controlador; (2) possibilitar a planta atingir a referência de forma célere; (3) compreender em nível linguístico as ações do controlador; (4) ser de fácil implementação quando concebido.

Este trabalho está organizado da seguinte forma: a próxima seção exhibe os principais conceitos da meta-heurística genética usado no GPF-Control, a Programação Genética Multigênica. A seção 3 apresenta os principais aspectos teóricos para a elaboração do modelo GPF-Control. A seção 4 apresenta duas aplicações do GPF-Control: o problema de parada ótima (Alba et al., 1999) (*cart-centering problem*) e o pêndulo invertido (Tsakonas, 2013), sendo os resultados do GPF-Control comparados com os de

outros controladores fuzzy-genéticos. Por fim, a seção 5 dispõe as considerações finais e trabalhos futuros.

2 Programação Genética Multigênica

Programação Genética (PG) (Koza, 1992; Langdon e Poli, 2002) é um método pertencente à Computação Evolutiva, biologicamente inspirado e que sintetiza programas de computador para desenvolver alguma tarefa. Tipicamente, a PG usa uma população de indivíduos em que cada um deles é denotado por uma estrutura em árvores onde é codificada uma equação matemática que descreve um vetor de saída Y a partir de um conjunto de J variáveis de entradas X_j ($j=1, \dots, J$). Com base nessa estrutura realiza-se a PG tradicional. Já a Programação Genética Multigênica (PGMG) (Hinchliffe et al., 1996; Searson et al., 2007) denota um indivíduo como um complexo de estruturas em árvores (equações), que da mesma forma recebem X_j , buscando predizer Y .

Cada uma das D equações (G_d , $d=1, \dots, D$) é uma solução parcial para o problema, sendo que a saída final \hat{Y} pode ser resultado de uma combinação linear dos G_d . A partir disto é possível avaliar a qualidade global do indivíduo. É fácil verificar que, quando $D=1$, a PGMG reduz-se à PG clássica.

Em relação aos operadores genéticos, a operação de mutação na PGMG é similar à efetuada na PG clássica. No caso da operação de cruzamento, é necessário efetuar uma distinção no nível em que a operação é realizada: é possível aplicar o cruzamento no baixo e no alto nível. O baixo nível é o espaço onde é possível manipular as estruturas (terminais e funções) das equações presentes em um indivíduo. No caso, tanto a mutação quanto o cruzamento de baixo nível na PGMG são semelhantes ao que é efetuado na PG clássica.

Um exemplo de cruzamento de alto nível para um indivíduo com cinco equações é apresentado na Figura 1. O alto nível é o espaço em que se manipulam de forma macro as equações presentes no indivíduo. Logo, verifica-se que, a partir de dois pontos aleatórios, são permutadas equações (no caso três) de um indivíduo para o outro. Este ponto de corte aleatório pode ser simétrico - quando troca-se o mesmo número de equações entre indivíduos - ou assimétrico - caso contrário. A saída resultante é mais afetada pelo cruzamento de alto nível do que pela operação de cruzamento de baixo nível e pela mutação.

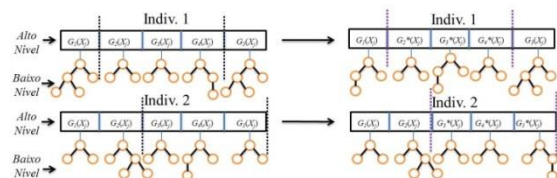


Figura 1. Exemplo de cruzamento de alto-nível entre indivíduos.

A próxima seção aborda o modelo GPF-Control, que, a partir das bases apresentadas da PGMG, utiliza essa meta-heurística em consórcio com a Lógica Fuzzy para controle, constituindo-se, assim, em um Sistema Fuzzy-Genético.

3 Controlador GPF-Control

Considere por Y_t o sinal de controle emitido à planta no instante de tempo t ($t=0,1, \dots, T$), proporcionando como resposta Z_{tk} que é o conjunto de k respostas, ou variáveis manipuladas da planta. Seja $X_{tk} = Z_{tk} - \text{Ref}_k$ o erro entre Z_{tk} e a k -ésima referência. Portanto, a partir de X_{tk} o controlador tenta enviar um sinal de controle Y_t em um determinado nível, de forma a tornar X_{tk} o menor possível.

Em linhas gerais o modelo GPF-Control consiste em quatro etapas: fuzzificação, inferência, defuzzificação e avaliação. O procedimento de inferência tem início na fuzzificação dos erros X_{tk} e do sinal emitido Y_t . Em seguida, são sintetizadas as funções que mapeiam cada estado linguístico X_{tk} a outro estado de Y_t , efetuando-se após a defuzzificação por alguma heurística. Esta solução é então avaliada e o processo evolutivo é repetido o número de vezes que for necessário para que a solução atenda a critérios estabelecidos pelo usuário. O diagrama em blocos da Figura 2 ilustra o modelo GPF-Control.

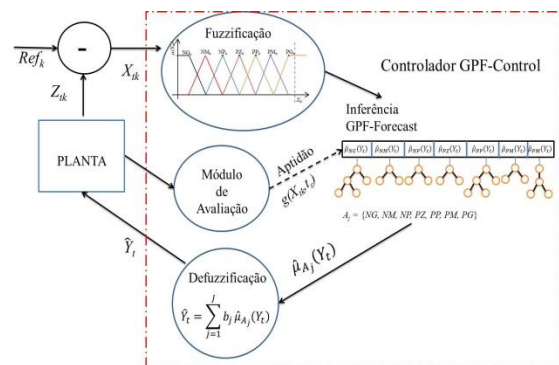


Figura 2. Diagrama do modelo GPF-Control

3.1. Fuzzificação

No controlador GPF-Control é considerado o grau de pertinência a um determinado estado linguístico de X_{tk} para assim definir o nível da ação de Y_t . Então, considera-se que tanto X_{tk} , quanto Y_t possam ter j diferentes estados linguísticos, manifestados matematicamente por funções de pertinência. Na literatura de controle fuzzy (Mendel, 1995; Pedrycz e Gomide, 2007) costuma-se atribuir sete diferentes estados linguísticos, tidos por: negativo grande (NG), negativo médio (NM), negativo pequeno (NP), próximo à zero (PZ), positivo pequeno (PP), positivo médio (PM) e positivo grande (PG).

Quando não se dispõe de um especialista, os conjuntos fuzzy de X_{tk} e de Y_t costumam ser dispostos de maneira uniforme e normalizada no universo de discurso de cada variável. A Figura 3 apresenta um

perfil típico de disposição. Seja A_{jk} o j -ésimo conjunto fuzzy para o k -ésimo erro X_{tk} . É possível definir $A_{jk} = \{NG_k, NM_k, NP_k, PZ_k, PP_k, PM_k, PG_k\}$. Por exemplo, A_{11} significa o conjunto fuzzy MP de X_{t1} , ou MP₁. Então, quando usado os rótulos linguísticos definidos anteriormente, tem-se $j=1,2,...,7$. Quando $k=0$ considera-se que o j -ésimo conjunto fuzzy A_j está relacionado à Y_t . A partir da fuzzificação das entradas inicia-se o processo de inferência GPF-Control.

3.2. Inferência GPF-Control

Considere que Y_t e X_{tk} tenham sido fuzzificados de forma uniforme e normalizada. Logo, tem-se as funções de pertinência $\mu_{NG}(Y_t), ..., \mu_{PG}(Y_t)$ para Y_t e $\mu_{NG_k}(X_{tk}), ..., \mu_{PG_k}(X_{tk})$ para X_{tk} . Deseja-se inferir quais e como os conjuntos fuzzy de X_{tk} relacionam-se com os conjuntos fuzzy de Y_t . Em outras palavras, deseja-se saber como os conceitos linguísticos das variáveis manipuladas definem os sinais emitidos pelo controlador.

A partir dos $\mu_{A_{jk}}(X_{tk})$ será possível extrair conclusões sobre a dinâmica de cada $\mu_{A_j}(Y_t)$ e, indiretamente, sobre Y_t . Logo, busca-se um mapeamento tal que:

$$\mu_{NG}(Y_t) = f_1(\mu_{A_{jk}}(X_{tk})) \quad (1)$$

$$\mu_{NM}(Y_t) = f_2(\mu_{A_{jk}}(X_{tk})) \quad (2)$$

$$\mu_{NP}(Y_t) = f_3(\mu_{A_{jk}}(X_{tk})) \quad (3)$$

$$\mu_{PZ}(Y_t) = f_4(\mu_{A_{jk}}(X_{tk})) \quad (4)$$

$$\mu_{PP}(Y_t) = f_5(\mu_{A_{jk}}(X_{tk})) \quad (5)$$

$$\mu_{PM}(Y_t) = f_6(\mu_{A_{jk}}(X_{tk})) \quad (6)$$

$$\mu_{PG}(Y_t) = f_7(\mu_{A_{jk}}(X_{tk})) \quad (7)$$

de tal forma que as ações de controle emitidas à planta atinjam o objetivo o mais rapidamente possível. Se a ação for minimizar X_{tk} , o similar é maximizar $\mu_{PZ}(X_{tk})$, ou seja, tornar o mais próximo de 1 o grau de pertinência do erro próximo de zero.

Para estabelecer a função de ligação $f_j(.)$ é necessário definir a forma funcional e escolher quais conjuntos fuzzy das variáveis manipuladas devem figurar em cada mapeamento dos conjuntos fuzzy da saída do controlador. Esta decisão é complexa, pois envolve escolhas adversas, entre as quais:

1. Forma funcional: linear ou não linear; idêntico para todas as equações ou diferente em algumas.
2. Seleção dos conjuntos fuzzy: quais, quanto e em quais mapeamentos $f_j(.)$ de saída eles devem ingressar.

Além destes dois fatores, deve-se ter em conta que uma função com muitos parâmetros e concebida de forma não linear torna a aceção do modelo mais obscura, apesar de possivelmente tornar a predição mais acurada. Logo, deve-se buscar ao máximo a preservação da inteligibilidade linguística do modelo, acompanhada de um mínimo de perdas em precisão. Uma forma de possibilitar o alcance de ambos os objetivos, visando a sintetizar automaticamente a forma funcional e simultaneamente escolher os conjuntos fuzzy das variáveis de entrada, é a partir do uso da PGMG.

Uma vez que na PGMG um indivíduo representa um complexo de programas, é possível definir para cada indivíduo um número de programas necessário para produzir um número de saídas necessárias. No caso de sete conjuntos fuzzy associados à variável de saída, é necessário manter sete programas por indivíduo, com cruzamento de alto nível simétrico e com inicialização estrita de sete programas por indivíduo.

Por exemplo, para a PGMG sintetizar a equação (1), seja o conjunto de terminais (funções de pertinência dos conjuntos associados às variáveis de entrada e constantes) e funções mostrados na Tabela 1 e ilustrados na Figura 3.

Tabela 1. Terminais e funções de entrada para o GPF-Control.

Terminais	Funções
$\mu_{A_{jk}}(X_{tk})$ e constantes geradas aleatoriamente no intervalo entre [0,1].	Soma, subtração, raiz-quadrada, potência quadrática e produto.

No conjunto de terminais estão presentes os sete conjuntos fuzzy ($j=1,...,7$) de cada k -ésimo erro X_{tk} , além de constantes geradas pela PGMG no intervalo [0,1].

Do conjunto de funções, as operações de soma e subtração denotam o sentido em que variações expressas pelos termos linguísticos das variáveis de entrada implicam em variações positivas (afirmação) ou negativas (negação) no grau de compatibilidade do termo linguístico da saída. As operações de raiz quadrada e de potência quadrática funcionam como os intensificadores e redutores de Zadeh (Liu et al., 2001). A função produto possui duas funcionalidades:

1. Produto tipo 1: Associar uma constante a um $\mu_{A_{jk}}(X_{tk})$;
2. Produto tipo 2: Combinar um $\mu_{A_{jk}}(X_{tk})$ com outro $\mu_{A_{jk}}(X_{tk})$ qualquer.

Sem perda de generalidade, considere-se $\hat{\mu}_{NG}(Y_t) = 0,8 \mu_{NG_1}(X_{t1})$ como exemplo da primeira característica do produto. Então, se $\mu_{NG_1}(X_{t1}) = 1$, segue-se que $\hat{\mu}_{NG}(Y_t) = 0,8$. Logo, a constante usada em conjunto com a função produto tipo 1 denota o nível da transmissão que a ativação específica de um

conjunto fuzzy da variável manipulada acarreta no nível do sinal emitido.

Para o segundo caso, seja a expressão abaixo:

$$\mu_{NG}(Y_t) = \mu_{PG_1}(X_{t1})\mu_{PM_1}(X_{t1}) + \mu_{PP_1}(X_{t1})\mu_{PG_1}(X_{t1}) + \mu_{PG_1}(X_{t1})\mu_{PP_2}(X_{t2}) \quad (8)$$

a primeira parte de (8), $\mu_{PG_1}(X_{t1})\mu_{PM_1}(X_{t1})$, considerando-se partições uniformes e normalizadas, pode ser interpretada linguisticamente como “aproximadamente entre positivo médio e grande”. Na segunda parte da expressão, $\mu_{PG_1}(X_{t1})\mu_{PP_1}(X_{t1}) = 0$, pois os conjuntos não são adjacentes e, portanto, o produto interno entre eles é nulo (ver Figura 3).

Para a terceira parte, $\mu_{PG_1}(X_{t1})\mu_{PP_2}(X_{t2})$, pode-se entender da forma clássica como “se X_{t1} é PG_1 e X_{t2} é PP_2 então Y_t é NG com grau de pertinência estimado em $\mu_{PG_1}(X_{t1})\mu_{PP_2}(X_{t2})$ ”.

3.3. Defuzzificação

O procedimento de defuzzificação associa a um conjunto fuzzy de saída um valor preciso, a partir de alguma heurística decisória (média dos máximos, centro de gravidade, etc.) (Roychowdhury e Pedrycz, 2001). No caso do modelo GPF-Forecast a defuzzificação pode ser realizada pelo método da altura:

$$Y_t = \frac{\sum_{j=1}^J b_j \mu_{A_j}}{\sum_{j=1}^J \mu_{A_j}} \quad (9)$$

em que $\sum_{j=1}^J b_j$ são os parâmetros de centro de cada função de pertinência triangular (vértice superior), e $\sum_{j=1}^J \mu_{A_j}(Y_t)$ são as saídas de cada equação gerada pela PGMG. Se a fuzzificação se der como na Figura 3, tem-se $b_1 = b_{NG}, \dots, b_7 = b_{PG}$ e $\hat{\mu}_{A_1}(Y_t) = \hat{\mu}_{NG}(Y_t), \dots, \hat{\mu}_{A_7}(Y_t) = \hat{\mu}_{PG}(Y_t)$. Ainda, a defuzzificação pode ser realizada pela altura média dos máximos, conforme a equação:

$$Y_t = \frac{\sum_{j=1}^J \phi_j b_j \mu_{A_j}(Y_t)}{\sum_{j=1}^J \phi_j \mu_{A_j}(Y_t)} \quad (10)$$

sendo ϕ_j uma função indicadora tal que $\phi_j = 1$, quando $\mu_{A_j}(Y_t) > \mu_{A_p}(Y_t)$, para todo $p=1, \dots, J$, e $p \neq j$, e $\phi_j = 0$, caso contrário. Para exemplificar o processo de cálculo e as diferenças em relação ao método da altura, seja a Figura 3, demonstrando um processo de defuzzificação.

Por exemplo, foram obtidos como resultado: $\mu_{NM}(Y_t) = 0,8$, $\mu_{PZ}(Y_t) = 0,6$, enquanto que $\mu_{NG}(Y_t) = \mu_{NP}(Y_t) = \mu_{PP}(Y_t) = \mu_{PM}(Y_t) = \mu_{PG}(Y_t) = 0$. Então, de acordo com a Eq. (9):

$$Y_t = \frac{-20 \cdot 0,8 + 0 \cdot 0,6}{0,8 + 0,6} = -11,43$$

Pelo método da altura média dos máximos, a resposta seria -20. Se houvesse dois ou mais valores máximos, a divisão forneceria uma média entre os máximos. Os demais termos nos cálculos eram iguais a zero, portanto foram omitidos. A presença destes dois métodos visa a proporcionar opções ao usuário, tanto pela seleção de uma defuzzificação mais “contínua” (método da altura), quanto uma mais “discreta” (método da altura média dos máximos).

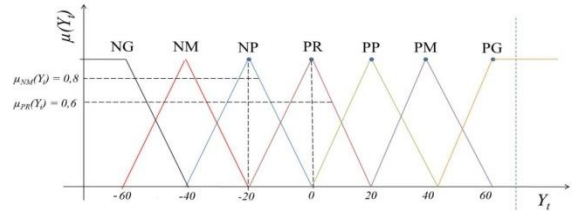


Figura 3. Exemplificação dos métodos de defuzzificação.

3.4. Avaliação

Obtida a saída Y_t , esta é enviada pelo controlador GPF-Control para a planta. A planta executa mais uma vez a operação, fornecendo um novo vetor de estados Z_{tk} , obtendo-se X_{tk} pela sua subtração da k -ésima referência. Este X_{tk} é novamente fuzzificado e realimentado ao GPF-Control (equações 1-7). Este procedimento é retomado o número de vezes necessário até atingir um critério de parada. Como o modo de avaliar a qualidade da solução fornecida pela PGMG é dependente do problema, usa-se uma função $g(X_{tk}, t_e)$. Em problemas onde T é fixado (problemas de trajetória, por exemplo), $g(X_{tk}, t_e)$ toma a forma de $\sum_{t=1}^T |X_{tk}|$, que deve ser minimizado. Em parada ótima, $g(X_{tk}, t_e)$ é igual a t_e , onde t_e é o tempo total para que $|X_{tk}| < \varepsilon_k$; portanto, $g(X_{tk}, t_e)$ deve ser minimizado.

4 Resultados e Discussões

4.1. Descrição dos Estudos de Casos

Foram realizados dois experimentos com o modelo GPF-Control, usando benchmarks conhecidos na literatura de controle: o problema da parada ótima (cart-centering) (Koza, 1992; Alba et al., 1999) e o pêndulo invertido (Tsakonas, 2013).

O problema da parada ótima consiste em um carro com massa m , deslizando em um trilho com fricção desprezível. Em um momento t qualquer, o carro está em uma posição X_t , com velocidade V_t . Institui-se um ponto de referência no qual o carro deve ser parado, aplicando uma força F_t . A dinâmica do sistema pode ser encontrada nas referências fornecidas.

Como meio de comparação foram seguidas as configurações usadas pelos autores: $F_t \in [-2,5 \ 2,5]$ N, $V_t \in [-2,5 \ 2,5]$ m/s e $X_t \in [-2,5 \ 2,5]$ m, com tolerância em relação a referência $\varepsilon = 0,5$, $m = 2$ kg, taxa de amostragem $\tau=0,02$ s, e referência igual a 0.

Portanto, o sistema deve conduzir o carro até $|X_t - 0| < 0,5$ e $|V_t - 0| < 0,5$. São fornecidos 16 condições iniciais com valores uniformemente distribuídos nos domínios de X_t . Portanto, a função a ser minimizada pelo controlador GPF-Control é t_e (como descrito na seção 3.4), o tempo necessário até atender o critério de parada: $|X_t - 0| < 0,5$ e $|V_t - 0| < 0,5$. O tempo máximo é de 10 segundos. Após, são selecionadas 1000 posições aleatórias, visando avaliar qual o tempo médio que o GPF-Control tomou para estacionar estes 100 carros nos limites da referência.

Foi estipulado o mesmo número de avaliações (25000) para o treinamento do GPF-Control que Thrift (1991) e Alba et al. (1999). As variáveis X_t e V_t foram fuzzificadas conforme a Figura 3, e as funções usadas foram soma, subtração e produto. O método de defuzzificação escolhido foi o da altura média dos máximos.

O pêndulo invertido se configura pela presença de um carro com massa M , sobre um trilho com fricção desprezível, que possui em seu centro um pêndulo com massa m e altura l . O objetivo é manter o pêndulo o maior tempo possível equilibrado, aplicando uma força F_t , que ao aumentar (ou reduzir) a velocidade V_t do carro, influa também na velocidade angular ω_t e, respectivamente, no ângulo θ_t do pêndulo. As equações que descrevem a dinâmica do pêndulo são encontradas em Koza (1992), ou em Tsakonas (2013).

De forma a comparar os resultados com Tsakonas (2013), empregaram-se as seguintes configurações: $\omega_t \in [-0,87 \ 0,87]\text{rad/s}$, $\theta_t \in [-0,34 \ 0,34]\text{rad}$, $F_t \in [-50 \ 50]\text{N}$, X_t e V_t sem restrições, $M=1\text{kg}$, $m=0,1\text{kg}$, $l=0,5\text{m}$, $g=9,8\text{m/s}^2$ e $\tau=0,01\text{s}$. São dados dois ângulos iniciais: $\theta_0 = \{-0,18, 0,18\}\text{rad}$, com $\omega_0 = \{0,0\}\text{rad/s}$. O objetivo é manter $|\theta_t - 0| < 0,01$, o mais rápido possível em um intervalo de no máximo 1 segundo.

Utilizou-se o mesmo número de avaliações (100000) para o treinamento do GPF-Control que Tsakonas (2013). As variáveis X_t , V_t , ω_t , θ_t foram fuzzificadas conforme Figura 3, e as funções usadas foram soma, subtração e produto. O método de defuzzificação escolhido foi o da altura.

Demais parâmetros usados em ambos os problemas: tamanho da população: 100, altura média da árvore: 7, taxa de cruzamento de alto nível: 50%, cruzamento de baixo nível: 85%, mutação: 15%. Os experimentos foram realizado no MATLAB R2010a (MATLAB, 2010).

4.2. Cart-Centering

A Figura 4a apresenta a trajetória da posição nos 132 passos (2,64s), enquanto que a figura 4b apresenta a relação velocidade e posição do melhor indivíduo da última geração do GPF-Control. Verifica-se que todas as posições usadas para treino conseguiram atender ao critério de parada de $|X_t - 0| < 0,5$ e $|V_t - 0| < 0,5$. O tempo médio, nas 1000 posições aleatórias, foi de 2,72s. A tabela 2 apresenta a com-

paração com os demais métodos, sendo estes inferiores aos obtidos com o GPF-Control.

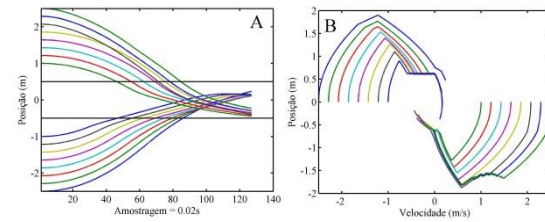


Figura 4. Trajetória da posição e relação posição-velocidade.

Tabela 2. Comparações com demais métodos.

Método	Tempo médio
Solução ótima	2,58s
GPF-Control	2,72s
Thrift (1991)	2,98s
Alba et al. (1999)	3,16s

As equações (11-17) apresentam o módulo de inferência do sistema GPF-Control.

$$\mu_{NG}(F_t) = 0,62\mu_{PP}(V_t) + 0,32\mu_{PM}(V_t) + 0,72\mu_{NP}(X_t) + 0,84 * \mu_{PM}(X_t) + 0,01 \quad (11)$$

$$\mu_{NM}(F_t) = 0,51 * (0,33\mu_{NP}(X_t)\mu_{PZ}(V_t) + \mu_{PG}(X_t)\mu_{PZ}(V_t) + 0,33\mu_{NP}(X_t)^2 + 1,33\mu_{NP}(X_t)\mu_{NM}(V_t) + \mu_{PP}(X_t)\mu_{NM}(V_t) + \mu_{PG}(X_t)\mu_{NM}(V_t) + \mu_{NM}(V_t)^2) \quad (12)$$

$$\mu_{NP}(F_t) = 0,20\mu_{NP}(X_t)\mu_{NM}(X_t) - 0,20\mu_{NP}(X_t)^2 + 0,23\mu_{NP}(X_t) + \mu_{NP}(X_t)\mu_{PM}(V_t) \quad (13)$$

$$\mu_{PZ}(F_t) = 0,08(\mu_{NM}(X_t)\mu_{PP}(V_t) + \mu_{NM}(X_t)\mu_{NM}(V_t)) \quad (14)$$

$$\mu_{PP}(F_t) = \mu_{NP}(X_t) \quad (15)$$

$$\mu_{PM}(F_t) = \mu_{NP}(X_t) \quad (16)$$

$$\mu_{PG}(F_t) = \mu_{NM}(V_t) + 0,02 \quad (17)$$

Como exemplo de interpretabilidade da solução do GPF-Control, observa-se que a equação (15) indica que uma força positiva pequena deve ser aplicada quanto maior for o grau de pertinência da posição do carro no estado negativo pequeno.

4.3. Pêndulo Invertido

A Figura 5 apresenta o comportamento do melhor indivíduo, nas 100.000 avaliações, a partir das duas angulações fornecidas pelo problema. Observa-se que o GPF-Control obteve êxito em manter o ângulo perto da referência demandada, sendo que a partir do 0,24 segundos este manteve o ângulo do pêndulo no limiar $|\theta_t - 0| < 0,01$ estabelecido. O GPF-Control proporcionou um resultado superior ao do controlador Fuzzy-Genético descrito em Tsakonas (2013), no qual, para uma mesma tolerância, o tempo foi de 0,61 segundos.

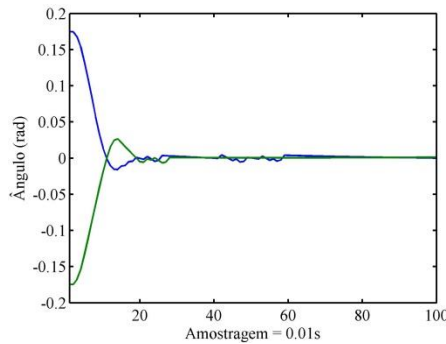


Figura 4. Trajetória da posição e relação posição-velocidade.

5 Conclusões

Este trabalho apresentou o controlador Fuzzy-Genético GPF-Control. Os estudos de casos efetuados demonstram as potencialidades desse controlador, tanto do ponto de vista das soluções encontradas, quanto da facilidade de implementação, pois a forma de defuzzificação e inferência são baseadas em equações algébricas. Trabalhos futuros devem incluir investigações sobre a sensibilidade do controlador a ruído, ou a variações em seus parâmetros, tais como as funções usadas e número de conjuntos fuzzy escolhidos para cada variável. Ainda, fazendo uso de novos benchmarks, procura-se verificar a abrangência do método.

Referências Bibliográficas

- Alba, E., Cotta, C. and Troya, J. (1999). Evolutionary design of fuzzy logic controllers using strongly-typed GP. *Mathware and Soft Computing*, Vol.6, pp. 109–124.
- Contreras, R.J., Vellasco, M.M.B.R. and Tanscheit, R. (2011). Hierarchical type-2 neuro-fuzzy BSP model. *Information Sciences*, Vol.181, No.15, pp.3210-3224. DOI: [10.1016/j.ins.2011.03.019](https://doi.org/10.1016/j.ins.2011.03.019)
- Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., and Magdalena, L. (2004). Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy sets and systems*, Vol.141, No.1, pp. 5-31. DOI: [10.1016/S0165-0114\(03\)00111-8](https://doi.org/10.1016/S0165-0114(03)00111-8)
- Cordón, O. (2011). A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems. *International Journal of Approximate Reasoning*, Vol.52, No.6, pp.894-913. DOI: [10.1016/j.ijar.2011.03.004](https://doi.org/10.1016/j.ijar.2011.03.004)
- Elmas, C., Deperlioglu, O., and Sayan, H. H. (2009). Adaptive fuzzy logic controller for DC–DC converters. *Expert Systems with Applications*, Vol.36, No.2, pp.1540-1548. DOI: [10.1016/j.eswa.2007.11.029](https://doi.org/10.1016/j.eswa.2007.11.029)
- Herrera, F., Lozano, M. and Verdegay, J. L. (1998). A learning process for fuzzy control rules using genetic algorithms. *Fuzzy sets and systems*, Vol.100, No.1, pp.143-158. DOI: [10.1016/S0165-0114\(97\)00043-2](https://doi.org/10.1016/S0165-0114(97)00043-2)
- Hinchliffe, M.P., Willis, M.J., Hiden, H., Tham, M.T., McKay, B. and G.W. Barton (1996). Modeling chemical process systems using a multi-gene genetic programming algorithm. In: *Proceedings of the First Annual Conference of Genetic Programming*. MIT Press, Massachusetts, pp. 56-65.
- Hoffmann, F. and Nelles, O. (2001). Genetic programming for model selection of TSK-fuzzy systems. *Information Sciences*, Vol.136, No.1, pp. 7-28. DOI: [10.1016/S0020-0255\(01\)00139-6](https://doi.org/10.1016/S0020-0255(01)00139-6)
- Jang, J.S.R., Sun, C.T. and Mizutani, E. (1997). *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*. Prentice-Hall, Englewood Cliffs – NJ.
- Kasabov, N. and Song, Q. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, Vol.10, N.2, pp.144-154, 2002. DOI: [10.1109/91.995117](https://doi.org/10.1109/91.995117)
- Langdon, W.B. and Poli, R. (2002). *Foundations of Genetic Programming*. Springer-Verlag, Heidelberg.
- Liu, B. D., Chen, C. Y. and Tsao, J. Y. (2001). Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol.31, No.1, pp.32-53. DOI: [10.1109/3477.907563](https://doi.org/10.1109/3477.907563)
- Luke, S. and Panait, L. (2002). Lexicographic parsimony pressure. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann Publishers, New York, pp. 829-836.
- MATLAB (2010). *MATLAB 7.10.0 (R2010a)*. Massachusetts: The MathWorks Inc.
- Mendel, J. M. (1995). Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, Vol.83, No.3, p.345-377. DOI: [10.1109/5.364485](https://doi.org/10.1109/5.364485)
- Pal, T., and Pal, N. R. (2003). SOGARG: A self-organized genetic algorithm-based rule generation scheme for fuzzy controllers. *IEEE Transactions on Evolutionary Computation*, Vol.7, No.4, pp.397-415. DOI: [10.1109/TEVC.2003.815377](https://doi.org/10.1109/TEVC.2003.815377)
- Precup, R. E. and Hellendoorn, H. (2011). A survey on industrial applications of fuzzy control. *Computers in Industry*, Vol.62 No.3, pp.213-226. DOI: [10.1016/j.compind.2010.10.001](https://doi.org/10.1016/j.compind.2010.10.001)
- Roychowdhury, S. and Pedrycz, W. (2001). A survey of defuzzification strategies. *International Journal of intelligent systems*, Vol.16, No.6 pp.679-695. DOI: [10.1002/int.1030](https://doi.org/10.1002/int.1030)
- Searson, D.P., Willis, M.J. and Montague, G.A. (2007). Co-evolution of non-linear PLS model components. *Journal of Chemometrics*, Vol. 2, pp. 592-603, 2007. DOI: [10.1002/cem.1084](https://doi.org/10.1002/cem.1084)
- Tanscheit, R., Gomide, F., Teixeira, M.C.M. and Aguirre, L.A. (2007). Modelagem e controle nebuloso. *Controle & Automação*, Vol.3, pp.283-324.
- Tsakonas, A. (2013). Local and global optimization for Takagi–Sugeno fuzzy system by memetic genetic programming. *Expert Systems with Applications*, Vol.40, No.8, pp.3282-3298. DOI: [10.1016/j.eswa.2012.12.099](https://doi.org/10.1016/j.eswa.2012.12.099)
- Thrift, P. (1991). Fuzzy logic synthesis with genetic algorithms. In: *Proceedings of the fourth international conference on genetic algorithms*. Morgan Kaufmann Publishers, pp. 509-513.