

Análise de desempenho de técnicas de controle de ganhos variáveis na tarefa de controle de posição de manipuladores robóticos

Phelipe W. Oliveira

Universidade Federal do Ceará
Depto. de Eng. de Teleinformática
Campus do Pici s/n, Fortaleza-CE

Email: phelipewesleydeoliveira@gmail.com

George A.P. Thé

Universidade Federal do Ceará
Depto. de Eng. de Teleinformática
Campus do Pici s/n, Fortaleza-CE

Email: geothe@hotmail.com

Guilherme A. Barreto

Universidade Federal do Ceará
Depto. de Eng. de Teleinformática
Campus do Pici s/n, Fortaleza-CE

Email: guialenbar@gmail.com

Resumo - Este trabalho apresenta uma comparação entre controladores proporcionais-derivativos com compensação de gravidade (PD), controladores proporcionais-derivativos não lineares (NPD) e controladores proporcionais-derivativos fuzzy na tarefa de controle de posição e trajetória em manipuladores robóticos, buscando assim, verificar o desempenho destas técnicas bem como averiguar a capacidade destes controladores em manter este desempenho ao serem impostos a trajetórias para as quais não foram projetados, além disso, busca apresentar a capacidade da linguagem Julia em tarefas de simulações de sistemas dinâmicos e algoritmos de inteligência computacional.

Palavras chaves - Controle Inteligente, Controle Fuzzy, Robótica, Linguagem Julia.

I. INTRODUÇÃO

Robôs manipuladores desempenham funções importantes em processos de automação e sistemas industriais. O controle de posição é uma das tarefas fundamentais nesses sistemas, porém os manipuladores robóticos possuem uma dinâmica acoplada e são sistemas dinâmicos não lineares o que torna o controle de posição de braços robóticos uma tarefa difícil.

Técnicas para contornar este problema foram procuradas e uma delas é o controle independente das juntas, neste método cada junta recebe um controlador próprio independente das outras juntas, controladores clássicos como o proporcional-derivativo (PD) e o proporcional-integrativo-derivativo (PID) normalmente são utilizados neste tipo de controle independente. Estas técnicas ainda são amplamente utilizadas em aplicações industriais devido à sua simplicidade e ao seu desempenho, particularmente em aplicações nas quais os parâmetros do processo não são bem conhecidos [1], [2]. Porém esses controladores clássicos possuem ganhos ajustados para valores constantes o que pode fazer que em certas operações eles não obtenham o desempenho desejado.

Buscando superar esta limitação e assim obter melhor desempenho ou lidar com restrições reais, é necessário realizar modificações nos ganhos dos controladores durante a realização do processo [2]. Algumas técnicas costumam ser utilizadas para procurar atingir esse objetivo, tais como lógica

fuzzy [2] [3], redes neurais [4] [5] [6], algoritmos genéticos e controladores não lineares [7] [8] [9]. Controladores não lineares procuram modificar os valores dos ganhos através de funções não lineares que são, normalmente, dependentes de uma variável do sistema [9], [7], já os do tipo fuzzy utilizam funções de pertinência para fazer as mudanças necessárias nos ganhos.

Neste contexto o presente trabalho procura fazer uma comparação entre o desempenho de um controlador do tipo proporcional-derivativo com compensação de gravidade (PD), um controlador proporcional-derivativo fuzzy do tipo Mandani, um controlador proporcional-derivativo fuzzy do tipo Sugeno e um controlador do tipo proporcional-derivativo não linear (NPD). O motivo da escolha destas técnicas de ganhos dinâmico foi o fato de não necessitarem de um conjunto prévio de dados para a realização de uma etapa de treinamento do algoritmo, dados estes que nem sempre estão disponíveis facilmente o que torna estas técnicas mais complicadas de serem aplicadas do ponto de vista industrial.

A comparação realizada representa uma contribuição para preencher lacunas deixadas nas análises apresentadas em [7] que compara o controlador NPD com o controlador PD e [2] que realizou uma comparação entre um controlador fuzzy e um controlador PID, ambos comparando técnicas de ganhos dinâmicos somente com técnicas clássicas, bem como para verificar a capacidade destes controladores em manter o desempenho em trajetórias para as quais não foram projetados, além disso, busca-se averiguar o desempenho da linguagem Julia com relação a simulações de sistemas dinâmicos e em algoritmos de inteligência computacional.

Este artigo está organizado da seguinte forma. Na seção II é apresentado o modelo dinâmico de manipuladores robóticos. Nas seções III, IV e V descrevem-se os controladores utilizados. Os resultados são discutidos na seção VI. A seção VII traz as conclusões do trabalho.

II. DINÂMICA DO MANIPULADOR

A dinâmica de um manipulador de n graus de liberdade pode ser escrita de uma forma padrão conforme mostrado em

(1) [10], [11], [7]

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau \quad (1)$$

Porém do ponto de vista prático os manipuladores sofrem distúrbios e restrições tais como o atrito e a capacidade de torque dos motores [10]. Desta forma é possível escrever a equação (1) de uma forma generalizada conforme

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + F(\dot{q}) + \tau_d = \tau \quad (2)$$

onde q é um vetor $n \times 1$ das posições das juntas, \dot{q} é um vetor $n \times 1$ das velocidades das juntas, \ddot{q} é um vetor $n \times 1$ das acelerações das juntas, τ é um vetor $n \times 1$ dos torques aplicados, $M(q)$ é uma matriz $n \times n$ simétrica e definida positiva da inércia do manipulador, $C(q, \dot{q})$ é um vetor $n \times 1$ de torques de Coriolis e centrífugos, $G(q)$ é o $n \times 1$ vetor de torque gravitacional, $F(\dot{q})$ é o vetor $n \times 1$ de fricção e τ_d é o vetor $n \times 1$ de perturbação.

O fenômeno da fricção é complexo de ser descrito, porém as características mais importantes podem ser retiradas dos efeitos de Coulomb e viscoso, assim a fricção pode ser modelada por

$$F(\dot{q}) = B\dot{q} + F_c \text{sign}(\dot{q}) \quad (3)$$

onde B e F_c são matrizes diagonais $n \times n$ de constantes positivas, que são relacionadas respectivamente com os coeficientes de atrito viscoso e dinâmico, e $\text{sign}()$ é uma função da forma

$$\text{sign}(x) = \begin{cases} 0 & \text{se } x = 0 \\ 1 & \text{se } x > 0 \\ -1 & \text{se } x < 0 \end{cases} \quad (4)$$

A perturbação τ_d pode ser usada, por exemplo, para representar qualquer dinâmica que possa ter sido modelada com pouca acurácia. Neste caso foi modelada da mesma forma que em [7] por

$$\tau_d = \begin{bmatrix} 5\cos(q_1) \\ \vdots \\ 5\cos(q_n) \end{bmatrix} \quad (5)$$

Para evitar o problema do torque máximo dos motores foi inserido no sistema um saturador, de forma que evite os picos de corrente e danifique o sistema. Assim o torque τ_i da junta i não pode ultrapassar o torque máximo do motor τ_i^{max} . Esse saturador pode ser modelado de forma [10]

$$\text{sat}(\tau_i) = \begin{cases} \tau_i & \text{se } |\tau_i| < \tau_i^{max} \\ \tau_i^{max} & \text{se } \tau_i \geq \tau_i^{max} \\ -\tau_i^{max} & \text{se } \tau_i \leq -\tau_i^{max} \end{cases} \quad (6)$$

Então o sistema pode ser escrito de forma a englobar o saturador conforme mostrado abaixo

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + F(\dot{q}) + \tau_d = \text{sat}(\tau) \quad (7)$$

III. CONTROLADOR PD COM COMPENSAÇÃO DA GRAVIDADE

Segundo [11] o controlador PD com compensação de gravidade é capaz de satisfazer o controle de posição para robôs com n graus de liberdade. Sua vantagem em relação ao PD tradicional está no fato dele garantir a realização do objetivo do controle de posição para modelos que contêm torque gravitacional. A lei de controle deste controlador é descrita por

$$\tau = K_p e - K_d \dot{e} + G(q) \quad (8)$$

onde e é o vetor $n \times 1$ do erro de posição e K_p e K_d são matrizes $n \times n$ diagonais e definidas positivas dos ganhos proporcionais e derivativos.

IV. CONTROLADOR NPD COM COMPENSAÇÃO DA GRAVIDADE

O controlador NPD é um controlador do tipo PD porém onde os ganhos proporcionais e derivativos são funções não lineares do erro de posição, desta forma temos que os valores dos ganhos são modificados de forma dinâmica. Mais detalhes sobre esse tipo de controlador podem ser encontrados em [7] e [9]. Aplicando essa mesma ideia ao controlador PD com compensação de gravidade teremos uma lei de controle descrita por

$$\tau = K_p(e)e - K_d(e)\dot{e} + G(q) \quad (9)$$

Neste trabalho as funções não lineares usadas foram as mesmas usadas por [7] de forma que os ganhos proporcionais para a junta i são da forma

$$K_p^i(e) = a_i + b_i e_i^2 \quad (10)$$

com a_i e b_i sendo parâmetros que alteram a curva de K_p^i e e_i sendo o erro de posição da junta i . E os ganhos derivativos são da forma

$$K_d^i(e) = \frac{c_i}{0.001 + e_i^2} \quad (11)$$

onde c_i é um parâmetro que modifica a curva de K_d^i e e_i é o erro de posição da junta i .

V. CONTROLADOR PD FUZZY

Da mesma forma que o controlador NPD, o controlador fuzzy também modifica os ganhos de forma dinâmica. Segundo [2] a lógica fuzzy é uma abordagem adequada, um mecanismo para determinar os ganhos não lineares do controlador com base em especificações práticas anteriores.

Neste trabalho foram utilizados os modelos fuzzy Mandani, apresentado por [10], bem como o modelo fuzzy Sugeno, porém na literatura é possível encontrar artigos comparando diferentes abordagens fuzzy como no de [3]. Os modelos utilizados procuram ser simples para diminuir o tempo de processamento necessário em uma implementação real, desta forma apresentam partição em poucos conjuntos e uma única entrada x e sua respectiva saída y . No sistema Mandani foram definidos três conjuntos fuzzy trapezoidais, mostrados

na figura 1, chamados pequeno(P), médio(M) e grande(G), para cobrir a variável de entrada e três conjuntos, mostrados na figura 2, também chamados de pequeno(P), médio(M) e grande(G), para cobrir a variável de saída.

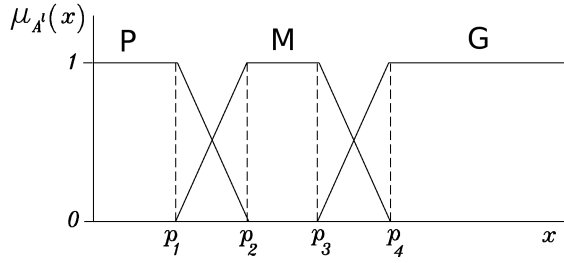


Fig. 1. Conjunto de entrada

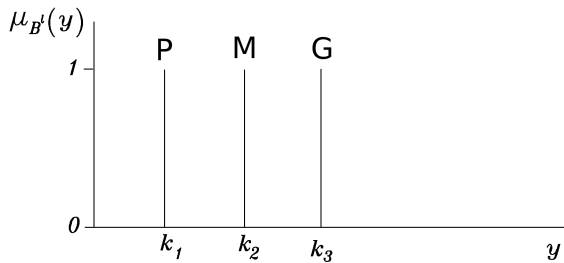


Fig. 2. Conjunto de saída

O modelo de regras fuzzy utilizadas no sistema Mandani é da seguinte forma

$$\begin{aligned} \text{IF } x \text{ é P então } y \text{ é G} \\ \text{IF } x \text{ é M então } y \text{ é M} \\ \text{IF } x \text{ é G então } y \text{ é P} \end{aligned} \quad (12)$$

onde x é dado por $|e|$ e y é dado como o valor de K_p ou K_d de cada junta, melhores explicações sobre o uso destas regras podem ser encontradas em [10] e [2]

O modelo de desfuzzyficação utilizado foi o do valor médio, assim a saída y pode ser obtida por

$$y = \frac{\sum_{l=1}^m \bar{y} \mu_{A^l}(x)}{\sum_{l=1}^m \mu_{A^l}(x)} \quad (13)$$

onde m é o número de regras fuzzy utilizadas no modelo e $\bar{y} = \{k1, k2, k3\}$ denota o centro das funções de pertinência do conjunto de saída.

No modelo Sugeno o conjunto de entrada adotado foi o mesmo adotado no sistema Mandani porém as saídas de cada regra foram modeladas como equações da forma

$$ax + b \quad (14)$$

onde x é a varável de entrada e a e b são constantes, o valor que sai do sistema Sugeno é obtido também através da equação 13.

VI. RESULTADOS EXPERIMENTAIS

Para poder comparar o desempenho dos controladores na tarefa de posicionamento das juntas de manipuladores robóticos e acrescentar aos trabalhos [7] e [2] de forma a comparar as técnicas de ganhos dinâmicos entre si e não apenas com as técnicas de controle clássicas, foi realizada a simulação de um robô planar de dois graus de liberdade conforme mostrado na figura 3, o experimento foi totalmente implementado na linguagem Julia (mais detalhes sobre a linguagem podem ser encontrado em [12]).

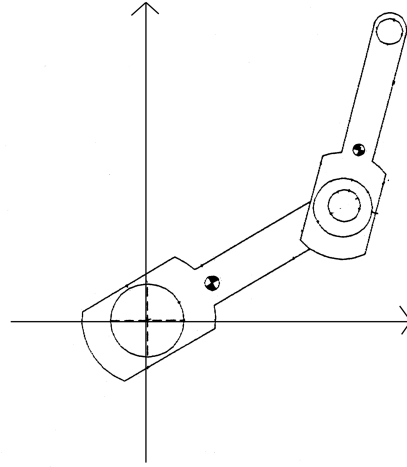


Fig. 3. Modelo de robô 2-GDL

Os dados do robô simulado neste trabalho podem ser encontrados em [13]. Os torques máximos permitidos por cada motor na simulação foram adotados como sendo de 200 N.m para o motor da junta 1 e de 15 N.m para o motor da junta 2, estes valores foram utilizados no saturador para evitar picos de energia que poderiam prejudicar o funcionamento real dos motores de um manipulador robótico.

Para a simulação dinâmica foi utilizado o método de Dormand-Prince, este é um método para resolução de equações diferenciais ordinárias e é um membro da família de métodos de Runge-Kutta. Todas as simulações implementadas utilizaram exatamente a mesma forma de resolver o conjunto de equações do sistema dinâmico, a única diferença existente nas simulações era o controlador empregado.

Para a comparação os controladores foram ajustados para seguir uma trajetória alvo da junta 1 de $\text{seno}(t)$ e na junta 2 de $\text{seno}(t)$, onde t é o tempo, foi considerada condição satisfatória de operação aquela onde o controlador tivesse o erro quadrático médio inferior ao valor de 0.06 rad^2 e o valor máximo do erro não passasse de 0.03 rad , após encontrado valores que satisfizessem esses requisitos para a trajetória desejada os ganhos não eram mais alterados. Os parâmetros utilizados nos controladores PD, NPD e fuzzy durante a simulação são mostrados nas tabelas I, II, III, IV, V,

VI e VII. Os resultados desta primeira simulação podem ser vistos nos gráficos apresentados nas figuras 4 e 5 e a tabela VIII apresenta informações sobre o erro máximo(EMAX), erro mínimo(EMIN) e erro quadrático médio(EQM).

junta	K_p	K_d
1	27000	100
2	19610	20

TABLE I

VALORES UTILIZADOS NO CONTROLADOR PD

junta	a	b	c
1	35090	21000	0.1
2	39370	10650	0.03

TABLE II

VALORES UTILIZADOS NO CONTROLADOR NPD

junta	p_1	p_2	p_3	p_4
1	0.035	0.07	0.17	0.52
2	0.035	0.07	0.17	0.52

TABLE III

VALORES UTILIZADOS NO CONJUNTO DE ENTRADA DO CONTROLADOR FUZZY

junta	k_1	k_2	k_3
1	4400	16400	46900
2	890	1100	36000

TABLE IV

VALORES UTILIZADOS NO CONJUNTO DE SAÍDA DO CONTROLADOR FUZZY PARA GANHO K_p

junta	k_1	k_2	k_3
1	182	410	400
2	10	20	20

TABLE V

VALORES UTILIZADOS NO CONJUNTO DE SAÍDA DO CONTROLADOR FUZZY PARA GANHO K_d

junta	Pequeno	Médio	Grande
1	$1000x + 8400$	$1000x + 10400$	$1000x + 30000$
2	$1000x + 3890$	$1000x + 21100$	$1000x + 36000$

TABLE VI

VALORES UTILIZADOS NO CONJUNTO DE SAÍDA DE CADA REGRA DO CONTROLADOR FUZZY SUGENO PARA GANHO K_p

junta	Pequeno	Médio	Grande
1	$100x + 150$	$100x + 400$	$100x + 410$
2	$100x + 2$	$100x + 4$	$100x + 15$

TABLE VII

VALORES UTILIZADOS NO CONJUNTO DE SAÍDA DE CADA REGRA DO CONTROLADOR FUZZY SUGENO PARA GANHO K_d

É possível ver nas figuras 4 e 5 bem como na tabela VIII que todos os controladores obtiveram um bom desempenho, todos seguiram a trajetória desejada apresentando um erro baixo, o que já era esperado já que todos foram projetados para essa condição de operação. Porém, agora modificando a trajetória

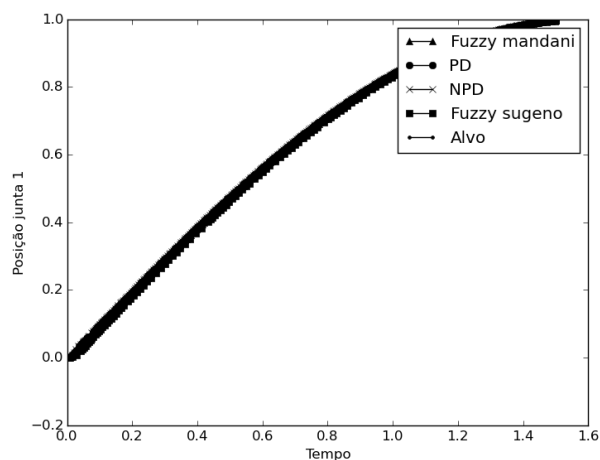


Fig. 4. Resultados da simulação 1 para junta 1

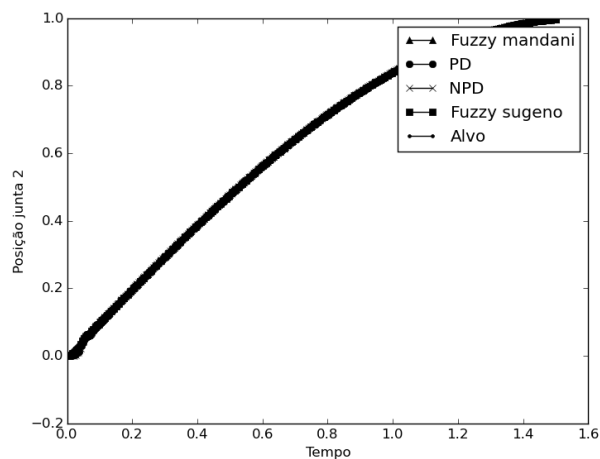


Fig. 5. Resultados da simulação 1 para junta 2

	junta	EMAX	EMIN	EQM
PD	1	0.009	0	$7.29 * 10^{-6}$
	2	0.023	0	$4.44 * 10^{-6}$
NPD	1	0.008	0	$4.39 * 10^{-6}$
	2	0.024	0	$3.51 * 10^{-6}$
FUZZY Mandani	1	0.009	0	$3.55 * 10^{-5}$
	2	0.018	0	$2.32 * 10^{-6}$
FUZZY SUGENO	1	0.02	0	0.0001
	2	0.011	0	$3.13 * 10^{-6}$

TABLE VIII

RESULTADOS PARA SIMULAÇÃO 1

das duas juntas para uma função do tipo $\coseno(t)$ onde t é o tempo sem modificar os os ganhos e valores da funções de pertinência que já haviam sido projetados para funções do tipo $seno(t)$ teremos respostas diferentes dos controladores. As figuras 6 e 7 apresentam os gráficos desta simulação e a tabela IX mostra os erros obtidos em cada controlador.

Agora é possível observar uma grande diferença nas re-

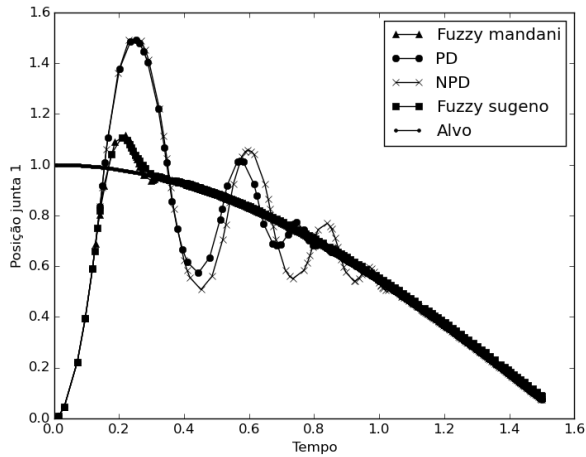


Fig. 6. Resultados da simulação 2 para junta 1

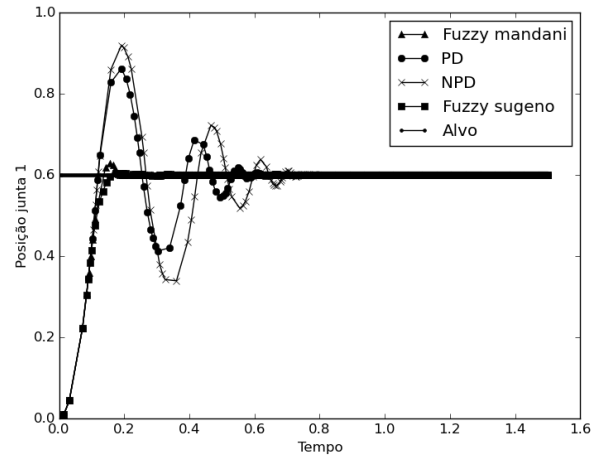


Fig. 8. Resultados da simulação 3 para junta 1

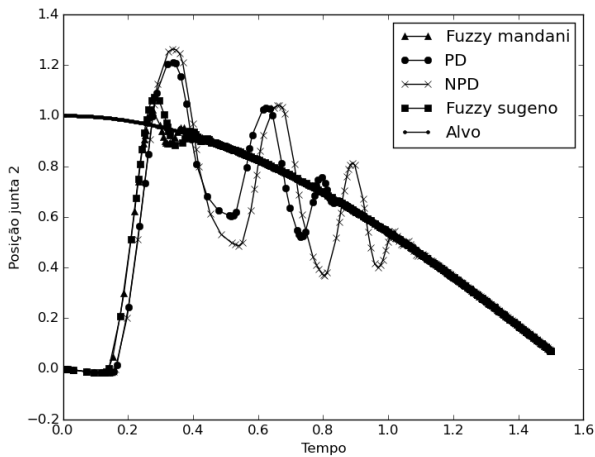


Fig. 7. Resultados da simulação 2 para junta 2

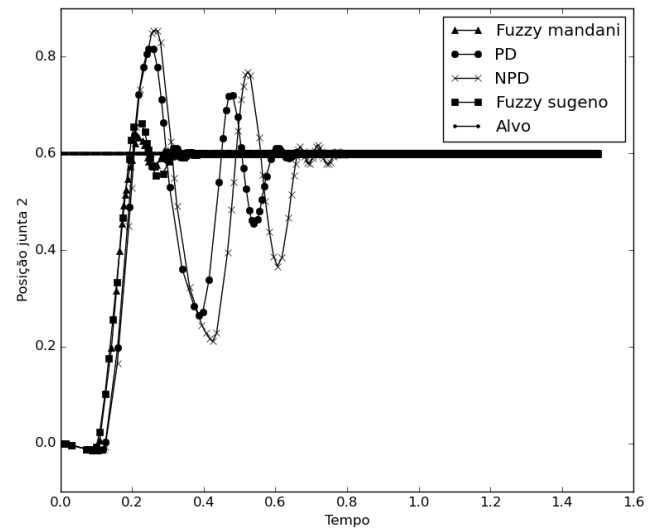


Fig. 9. Resultados da simulação 3 para junta 2

	junta	EMAX	EMIN	EQM
PD	1	1	$4.20 * 10^{-5}$	0.017
	2	1	$3.42 * 10^{-5}$	0.03
NPD	1	1	0.00017	0.015
	2	1	$5.33 * 10^{-5}$	0.027
FUZZY Mandani	1	1	0.0011	0.0098
	2	1	$1.8 * 10^{-5}$	0.018
FUZZY SUGENO	1	1	0.002	0.023
	2	1	$2.93 * 10^{-5}$	0.04

TABLE IX
RESULTADOS PARA SIMULAÇÃO 2

	junta	EMAX	EMIN	EQM
PD	1	0.6	$1.32 * 10^{-5}$	0.0015
	2	0.6	$1.42 * 10^{-5}$	0.003
NPD	1	0.6	$1.67 * 10^{-6}$	0.0012
	2	0.6	$5.24 * 10^{-5}$	0.0027
FUZZY Mandani	1	0.6	$4.92 * 10^{-5}$	0.0007
	2	0.6	$2.57 * 10^{-5}$	0.0017
FUZZY SUGENO	1	0.6	$4.13 * 10^{-5}$	0.001
	2	0.6	0.00017	0.002

TABLE X
RESULTADOS PARA SIMULAÇÃO 3

spostas dos controladores, os controladores do tipo fuzzy apresentaram desempenho superior aos demais quanto aos critérios de tempo de acomodação e sobressinais durante o percurso do caminho desejado, isto mostra que o controlador fuzzy foi mais robusto aos outros apresentados. Analisando agora o caso onde as juntas devem ir e permanecer em uma posição fixa, para isso foi escolhida como alvo a posição de

0.6 rad para cada junta do robô.

Temos novamente que os controladores do tipo fuzzy apresentaram um resultado superior aos demais, desta vez bem mas

visível graficamente através das figuras 8 e 9 onde é possível ver os controladores PD e NPD apresentando um tempo de acomodação bem maior do que os valores dos controladores fuzzy, bem como sobressinais muito elevados.

Do ponto de vista do desempenho da linguagem Julia para trabalho em simulações de sistemas dinâmicos e trabalhos com algoritmos de inteligência computacional, temos que ela é uma excelente escolha. O processo de simulação do robô com o controlador consistia na resolução de sistemas de equações diferenciais bem como na implementação dos controladores, todo o código foi implementado na linguagem Julia. O tempo necessário para executar os quatro controladores ao mesmo tempo e mostrar as respostas desejadas não ultrapassou nem uma vez de 3 segundos, isso considerando que o código não foi otimizado e nem usou todos os requisitos disponíveis na linguagem tais como programação paralela, o que tornaria a resposta ainda mais rápida.

VII. CONCLUSÕES

Neste trabalho, o problema de controle de posição de manipuladores robóticos foi considerado partindo de quatro estratégias de controle. Simulações foram realizadas para implementar uma técnica clássica (ganhos constantes) e outras três de ganhos variáveis, sendo estas: PD com compensação de gravidade, NPD, PD fuzzy Mandani e PD fuzzy Sugeno, respectivamente.

Nas simulações foi possível notar as melhores respostas dos controladores fuzzy em comparação ao controlador PD clássico e ao controlador PD não linear do ponto de vista de modificações nas trajetórias sem a necessidade de reajustes dos parâmetros do controlador, o que mostra que controladores do tipo fuzzy conseguem obter um desempenho elevado mesmo quando estão operando em condições para as quais não foram projetados. Os controladores fuzzy implementados foram modelados com poucos conjuntos e poucas regras e mesmo assim obtiveram bons resultados o que os torna uma opção viável do ponto de vista computacional e prático e mostra-se uma opção para aplicações onde pode haver mudança nas trajetórias desejadas para os elos.

Foi possível observar também que a linguagem Julia obtém bons resultados no tempo de resposta de simulações de sistemas dinâmicos o que faz com que ela seja uma boa opção para atividades que envolvam este tipo de tarefa, bem como em algoritmos que requerem um alto poder de processamento.

Para trabalhos futuros pretende-se verificar se ocorre melhora no desempenho dos controladores fuzzy quando é feito um ajuste fino dos atributos dos grupos fuzzy bem como modificando as funções do fuzzy Sugeno para polinômios de ordem superior para assim comparar o desempenho de diferentes técnicas fuzzy de controle em simulações e em manipuladores robóticos reais, além disso, verificar a estabilidade desses controladores.

AGRADECIMENTOS

Os autores agradecem ao CNPq pelo apoio ao desenvolvimento deste trabalho.

REFERENCES

- [1] V. Mummadi, "Design of robust digital PID controller for H-bridge softswitching boost converter," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 2883–2897, 2011.
- [2] J. L. Meza, V. Santibáñez, R. Soto, and M. A. Llama, "Fuzzy self-tuning PID semiglobal regulator for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 6, pp. 2709–2717, 2012.
- [3] B. B. Ghosh, B. K. Sarkar, and R. Saha, "Realtime performance analysis of different combinations of fuzzy-PID and bias controllers for a two degree of freedom electrohydraulic parallel manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 34, pp. 62–69, 2015.
- [4] T. D. Le, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro, "An online self-gain tuning method using neural networks for nonlinear PD computed torque controller of a 2-dof parallel manipulator," *Neurocomputing*, vol. 116, pp. 53–61, 2013.
- [5] S. Cong and Y. Liang, "PID-like neural network nonlinear adaptive control for uncertain multivariable motion control systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3872–3879, 2009.
- [6] J. C. Shen, "Fuzzy neural networks for tuning PID controller for plants with underdamped responses," *IEEE Transactions on Industrial Electronics*, vol. 9, no. 2, pp. 333–342, 2001.
- [7] J. Huang, C. Yang, and J. Ye, "Nonlinear PD controllers with gravity compensation for robot manipulators," *Cybernetics and Information Technologies*, vol. 14, no. 1, pp. 141–150, 2001.
- [8] C. Dey, R. K. Mudi, and D. Simhachalam, "A simple nonlinear PD controller for integrating processes," *ISA Transactions*, vol. 53, no. 1, pp. 162–172, 2014.
- [9] R. Kelly and R. Carelli, "A class of nonlinear PD-type controllers for robot manipulators," *Journal of Robotic Systems*, vol. 13, no. 2, pp. 793–802, 1996.
- [10] M. A. Llama, R. Kelly, and V. Santibáñez, "A stable motion control system for manipulators via fuzzy self-tuning," *Fuzzy Sets and Systems*, vol. 124, no. 2, pp. 133–154, 2001.
- [11] R. Kelly, V. Santibáñez, and A. Loria, *Control of Robot Manipulators in Joint Space*. New York: Springer-Verlag, 2005.
- [12] (2015) The julia language. [Online]. Available: <http://julialang.org/>
- [13] F. Reyes and R. Kelly, "Experimental evaluation of model-based controllers on a direct-drive robot arm," *Mechatronics*, vol. 11, pp. 267–282, 2001.