

Comparação Experimental de Desempenho entre Controladores PI e PI-RLS na Regulação de Velocidade de um Motor Brushless

Fernando da Fonseca Schneider, Paulo Jefferson Dias de Oliveira Evald,
César Bastos da Silva, Eduardo do Amaral Leivas, Rodrigo Zelir Azzolin

Centro de Ciências Computacionais, Universidade Federal do Rio Grande, Rio Grande, 96201-900
Email: (fernando.f.schneider, paulo.evald, cesinha2612, eduardo.doal, rodrigoazzolin@gmail.com)

Resumo—Neste trabalho é apresentada a modelagem matemática de um motor *brushless* e seu controle de velocidade por duas técnicas de controle. A primeira estratégia de controle, comumente utilizada, é um controle PI (Proporcional-Integral) tradicional. Como alternativa à esse controlador de ganhos fixos, foi integrado ao controlador PI um algoritmo RLS (*Root-Least-Squares*), tornando possível uma adaptação dos ganhos do controlador e com isso melhorar seu desempenho. Resultados de simulação e experimentais são apresentados e discutidos, para comparação de desempenho entre as estratégias de controle implementadas.

Palavras-Chave – Controle PI, Controle PI-RLS, Motor *brushless*

I. INTRODUÇÃO

Nos dias atuais, a popularidade dos motores *brushless* tem aumentando significativamente, devido a suas vantagens em relação aos motores de corrente contínua e motores de indução [1]. Dentre as vantagens, destaca-se: alta eficiência, baixos níveis de ruído, melhor característica de velocidade×torque, dentre outras. Esse motor vem sendo empregado em diversas aplicações, tais como: mãos robóticas [2], propulsão de veículos aquáticos [3] e robôs de futebol *small size* [4].

Graças as melhorias nos *drivers* de semicondutores, o controle de motores *brushless* se tornou mais eficiente, possibilitando que esses motores operem com velocidades precisas [5]. Existem diferentes lógicas de controle que podem ser utilizados, tais como: Controle Proporcional-Integral (PI)[6], controle de velocidade baseado em lógica *fuzzy* Proporcional-Derivativa (PD)[7] e Proporcional-Integral-Derivativo de ordem fracionária[8].

Neste trabalho, é dado enfoque aos controladores PI. Essa estratégia de controle é atualmente a mais utilizada, principalmente em indústrias, como em controle de avião[9] e controle de nível de um tanque[10], embora tenha seu desempenho prejudicado frente à variações paramétricas, influência de distúrbios externos e até mesmo dinâmicas não-modeladas, como não-linearidades do sistema, características que o próprio motor *brushless* apresenta, em perdas no núcleo[11]. Essas limitações do

Controle PI são devido à sua estrutura de ganhos fixos [12]. Para tornar esse controle mais robusto, algoritmos para adaptação dos ganhos são integrados em sua estrutura, tais como: técnicas *fuzzy* [13], redes neurais artificiais [14], algoritmo RLS (*Recursive Least Square*), entre outros. Em especial, o algoritmo RLS demonstra bom desempenho para estimação de parâmetros com baixo custo computacional, podendo obter uma solução de alto desempenho para adaptação dos ganhos do controle PI [4]. Portanto, neste trabalho, este é o algoritmo escolhido para agregar adaptação aos ganhos do controle PI.

Portanto, o objetivo deste trabalho é projetar e implementar os controles PI tradicional e PI-RLS nos quatro motores *brushless* de um robô de futebol *small size*. Assim, a principal contribuição deste trabalho é a comparação de desempenho entre esses dois controladores, mostrando as vantagens da utilização de um controle PI-RLS em relação à um controle PI tradicional, experimentalmente.

A organização deste trabalho é dada a seguir. Na Seção 2 é descrita a modelagem do motor *brushless*, bem como o método utilizado para medir velocidade do mesmo. Na Seção 3 é demonstrado o projeto do Controle PI. Na Seção 4 é visto o equacionamento do Algoritmo RLS aplicando o mesmo no Controle PI. Na Seção 5 são destacados os resultados experimentais obtidos. E, por fim, na Seção 6 é abordada a conclusão.

II. MOTOR BRUSHLESS

O motor *brushless* utilizado nesse trabalho foi o motor **EC 45 flat 30w** [15]. Esse modelo contém três sensores *hall*, que possuem 24 pulsos por revolução (PPR). Esses pulsos são convertidos para para radianos (24 pulsos = 2π), para obter a posição angular ($\theta(s)$) do motor.

Para atenuação dos ruídos de leitura do sensor, foi implementado um Filtro de Variáveis de Estado [16], com frequência de corte de 10Hz e ordem 1. A função de transferência desse filtro é mostrada em (1), onde F , F_f , ω_c e n são a função à filtrar, função filtrada, banda passante do filtro e grau desejado da derivada, respectivamente. A função que será filtrada é o $\theta(s)$, e como o filtro é de primeira ordem, obtém-se sua derivada, a velocidade angular ($\omega(s)$).

$$\frac{F_f}{F} = \frac{\omega_c^{n+1}}{(s + \omega_c)^{n+1}} \quad (1)$$

Segundo *Gieras*, a parte mecânica do motor *brushless* possui uma constante tempo muito maior que a constante de tempo elétrica [17]. Portanto, com viés de simplificar a modelagem, pode-se modelar esse motor considerando apenas as partes elétricas, que apresentam as dinâmicas dominantes, resultando em um modelo descrito por uma função de transferência (FT) primeira ordem.

Uma FT de primeira ordem pode ser descrita por um ganho (K_0), que é a relação entre valor em regime permanente e a entrada aplicada, e por uma constante de tempo (τ), a qual corresponde ao tempo necessário para a resposta do sistema alcançar 63,2% do valor em regime permanente [18]. Para identificar a FT que descreve o comportamento dinâmico do motor, foram aplicados diferentes valores de PWM (*Pulse Width Modulation*) no *driver* de acionamento do motor, medindo os pulsos e estimando as respectivas velocidades com o filtro de variáveis de estado. Após diversos testes, foi feita uma média aritmética dos valores de K_0 e τ , conforme mostrado na Tabela I. A FT correspondente está descrita em (2).

Tabela I
PARÂMETROS IDENTIFICADOS DO MOTOR *brushless*

PWM	$\omega(s)$ (RPM)	K_0	τ (s)
16	305,5	19,0937	0,3194
32	775,4	24,23125	0,3196
48	1239	25,8125	0,3575
64	1693	26,4531	0,3475
80	2075	25,9375	0,3228
$\Sigma/5$		24,30	0,333

$$G(s) = \frac{K_0}{\tau s + 1} = \frac{24,30}{0,333s + 1} \quad (2)$$

Na seção a seguir será discutida a metodologia abordada para a sintonia do controlador Proporcional-Integral (PI).

III. CONTROLE PI

O Controle PI utiliza duas das três contribuições do Controle PID (Proporcional-Integral-Derivativo) [18]. A junção dessas ações de controle é benéfica em muitos aspectos. Especificamente, para o controle PI, tem-se a garantia de erro nulo em regime permanente, devido a ação integral e uma rápida convergência no rastreamento de referência, quando o sistema é linear e invariante no tempo [19]. O fator derivativo será evitado nos projetos de controle propostos neste trabalho pois a contribuição derivativa pode amplificar os ruídos do sistema [20].

A sintonia do controle PI utilizada neste trabalho é fundamentada sobre o comportamento dinâmico desejado. Isto é, deseja-se que a resposta apresente, ultrapassagem percentual (U.P.) de 1% no máximo e um tempo de acomodação t_s de 1 segundo (critério de 2%) [18], [19].

Atribuindo o valor de ultrapassagem percentual em (3), encontra-se o coeficiente de amortecimento $\zeta = 0,83$, e após isso, aplicando o valor do tempo de acomodação e o ζ em (4) obtém-se a frequência natural $\omega_n = 4,84 \text{ rad/s}$.

$$U.P. = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (3)$$

$$t_s = \frac{4}{\zeta\omega_n} \quad (4)$$

A estratégia de controle implementada, pode ser analisada na Figura 1, onde os blocos $C(s)$ e $G(s)$, são o controlador PI e o sistema, respectivamente. Os sinais $R(s)$, $E(s)$, $U(s)$, $U'(s)$ e $\omega(s)$, são a referência de velocidade desejada, o erro de rastreamento, a ação de controle, a ação de controle após a saturação e a velocidade angular, respectivamente. Ressalta-se que é imprescindível o uso de um saturador no sistema para que a simulação fique mais verossímil, pois o sistema real apresenta limitações de valores para a ação de controle.

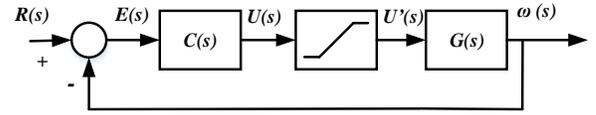


Figura 1. Sistema em malha fechada com controle PI

Sendo o ganho proporcional como K_P e o ganho integral por K_I , o sistema em malha fechada é descrito em (5).

$$\frac{\omega(s)}{R(s)} = \frac{\frac{K_P K_0}{\tau} s + \frac{K_I K_0}{\tau}}{s^2 + \frac{1+K_P K_0}{\tau} s + \frac{K_I K_0}{\tau}} \quad (5)$$

Comparando o resultado de (5) com a FT padrão de segundo grau [18], mostrada em (6) com os valores identificados de ζ e ω_n , pode-se calcular os ganhos, que resultaram em $K_P = 0,063$ e $K_I = 0,238$.

$$W(s) = \frac{\omega_n^2}{s^2 + 2\omega_n\zeta s + \omega_n^2} \quad (6)$$

O resultado da simulação do sistema de controle PI com os ganhos calculados, é mostrada na Figura 2. Como pode ser observado, a ultrapassagem percentual apresentada é próxima a 1% e o tempo de acomodação ficou em 1 segundo, atendendo o critério de 2% do projeto. Após 2 segundos se aplicou um distúrbio de 200 RPM, valor similar ao que ocorre na prática, percebeu-se que o controlador conseguiu corrigir a ação do distúrbio retornando para a referência desejada.

Na próxima seção é apresentado o algoritmo RLS, bem como sua integração ao controle PI.

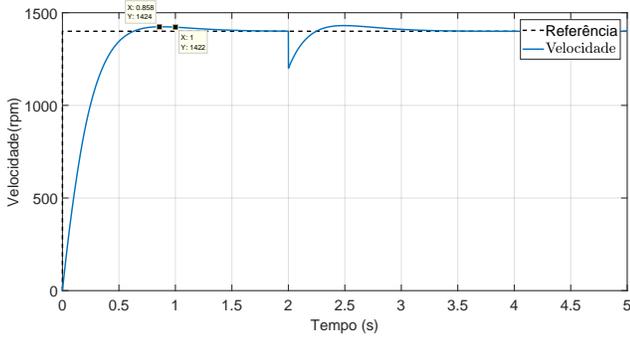


Figura 2. Simulação do Controle PI para um degrau de 1400 RPM

IV. ALGORITMO RLS E CONTROLE PI-RLS

Esse método é composto por um vetor predição ($\hat{\omega}(k)$), que irá variar até que fique idêntico a variável de interesse, neste caso $\omega(k)$, a matriz de regressão linear ($C(k)$), que contém as medições do sistema físico, e por último, o vetor paramétrico ($\hat{\theta}(k)$), que é onde está contido a estimação dos parâmetros do sistema [4]. Esse último vetor será calculado com o auxílio de outras duas matrizes, matriz de ganhos ($K(k)$) e a matriz de covariância ($P(k)$) [21].

Para utilizar o método RLS, deve-se ter o sistema para na forma matricial, para identificar qual parcela do sistema corresponde as matrizes citadas anteriormente. Assim, o sistema na forma matricial é apresentado em (7). Portanto, $\omega(k)$ equivale ao vetor de predição, a matriz de regressão linear corresponde a parcela que contém o valor de PWM e a aceleração do sistema ($\dot{\omega}(k)$), por fim o vetor paramétrico corresponde a parcela dos parâmetros a serem estimados, K_0 e τ .

$$\omega(k) = [PWM(k) \quad -\dot{\omega}(k)] \begin{bmatrix} K_0 \\ \tau \end{bmatrix} \quad (7)$$

Para encontrar a velocidade estimada ($\hat{\omega}(k)$), utiliza-se a (8).

$$\hat{\omega}(k) = C(k)\hat{\theta}(k) \quad (8)$$

O erro de estimação é estabelecido pela diferença entre a velocidade real ($\omega(k)$) e a velocidade estimada ($\hat{\omega}(k)$), conforme mostrado a seguir.

$$\tilde{\omega}(k) = \omega(k) - \hat{\omega}(k) \quad (9)$$

As matrizes de ganhos ($K(k)$) e de covariância ($P(k)$), são mostradas em (10) e (11), respectivamente. Por fim, $\hat{\theta}(k)$ é calculado utilizando (12). Vale a pena ressaltar que quanto mais próximos dos valores ótimos, forem os valores projetados de $P(k)$, melhor será o desempenho do algoritmo. Portanto, como são desconhecidos os valores ótimos, esta matriz foi inicializada como uma matriz identidade (I).

$$K(k) = \frac{P(k-1)C^T(k)}{1 + C(k)P(k)C^T(k)} \quad (10)$$

$$P(k) = (I - K(k)C(k))P(k-1) \quad (11)$$

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\tilde{\omega}(k) \quad (12)$$

Para utilizar a matriz $C(k)$, deve-se medir a aceleração da resposta de saída, para isso, passará a velocidade pelo filtro de variáveis de estado, obtendo sua derivada, a aceleração ($\dot{\omega}(k)$), e a velocidade real filtrada ($\omega_f(k)$). É necessário utilizar a velocidade filtrada ($\omega_f(k)$) no lugar da velocidade real ($\omega(k)$) no cálculo do erro estimado, destacado em (9), pois dessa forma tanto a aceleração quanto a velocidade sofrem o atraso em fase [18], que é promovido pelo filtro de variáveis de estado. Além disso, a ação de controle também é passada pelo filtro de variáveis de estado, para que fique com mesmo atraso dos demais sinais. O diagrama do sistema controlador PI-RLS pode ser visto na Figura 3.

A saída do algoritmo RLS são os parâmetros \hat{K}_0 e $\hat{\tau}$, isto é, seus valores estimados. Esses valores são utilizados para cálculo dos ganhos K_P e K_I , conforme mostrado na Seção III.

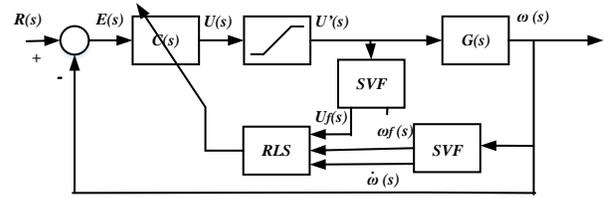


Figura 3. Simulação do Controle PI-RLS

Na Figura 4, é mostrada a simulação do rastreamento de referência de 1400RPM com um distúrbio de 200RPM sendo aplicado após 2 segundos de ensaio. Como pode-se observar, o sistema respondeu mais rapidamente ao PI-RLS do que ao PI puramente. Entretanto, a resposta com controle PI-RLS apresentou um maior sobressinal.

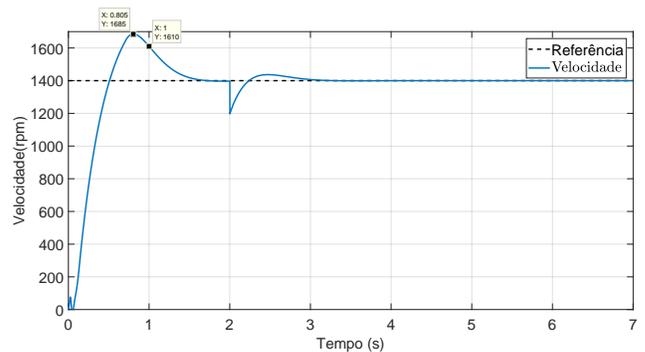


Figura 4. Simulação do Controle PI-RLS para um degrau de 1400 RPM

Nas Figuras 5, 6 e 7 são apresentados a estimação de \hat{K}_0 , $\hat{\tau}$ e dos ganhos K_P e K_I , respectivamente, para o mesmo ensaio demonstrado na Figura 4. Como pode ser observado nessas figuras, os parâmetros foram estimados em cerca de 1 segundo, nota-se também que após a aplicação do distúrbio, os parâmetros \hat{K}_0 , $\hat{\tau}$ variaram e a partir dessa variação, novos ganhos K_P e K_I se estabeleceram para que o sistema se mantesse com as mesmas características de desempenho.

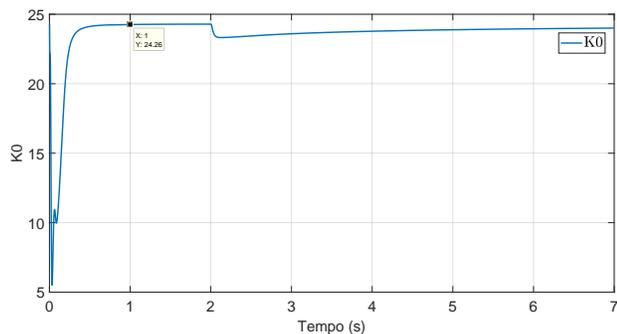


Figura 5. Estimação do parâmetro \hat{K}_0

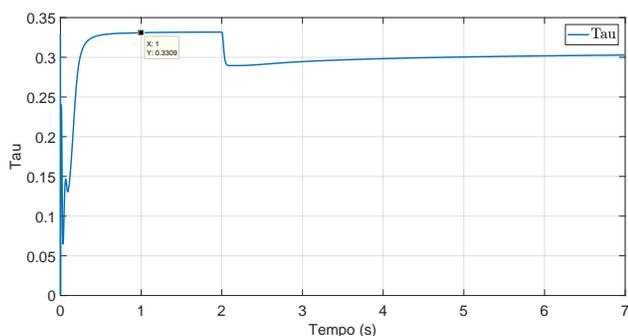


Figura 6. Estimação do parâmetro $\hat{\tau}$

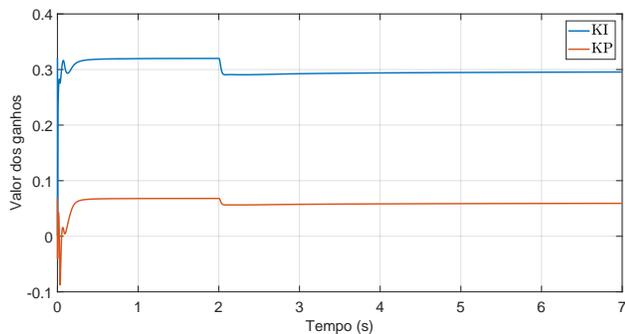


Figura 7. Estimação dos ganhos K_P e K_I

Na seção a seguir serão mostrados os resultados experimentais obtidos.

V. RESULTADOS EXPERIMENTAIS

A bancada montada para os testes experimentais é mostrada na Figura 8, onde está presente o motor *brushless* e um motor corrente contínua, que é utilizado como gerador. Além disso, há um botão para acionar os terminais do motor cc em curto-circuito, fazendo com que o mesmo gere um distúrbio para o motor *brushless*.

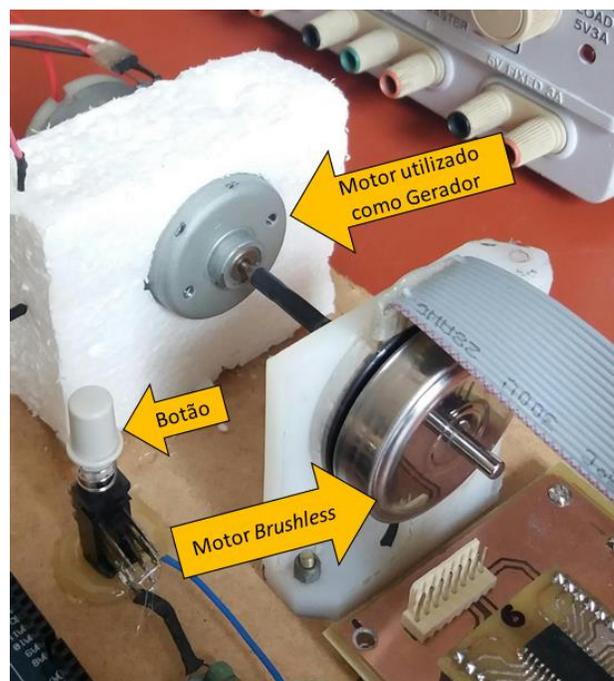


Figura 8. Bancada para testes experimentais

O resultados obtidos para o PI podem ser visto na Figura 9. Note que o sistema correspondeu ao tempo de acomodação desejado, bem como a máxima ultrapassagem percentual projetada, contudo seu comportamento foi diferente ao comportamento simulado. Isto é, a resposta experimental foi mais lenta e houveram pequenas oscilações devido a influência de dinâmicas não modelados. Já em 2,696 segundos, foi aplicado um distúrbio e o tempo para que o sistema se estabilizasse novamente foi de 1,136 segundo.

A resposta para o PI-RLS é mostrada na Figura 10, onde é possível perceber que houve um sobressinal, contudo, o sistema atinge a referência e se estabiliza mais rápido do que apenas com o controle PI. Em 0,827 segundo foi aplicado um distúrbio, e como pode ser visto, o sistema levou cerca de 0,451 segundo para retornar ao valor desejado, ou seja, praticamente um terço do tempo necessário para que o controle PI compensasse o distúrbio.

Assim como na simulação, será apresentado o comportamento da estimação dos parâmetros. Na Figura 11, é mostrado a estimação do parâmetro \hat{K}_0 , que estava convergindo para 24,06, próximo ao modelado ($K_0 = 24,30$),

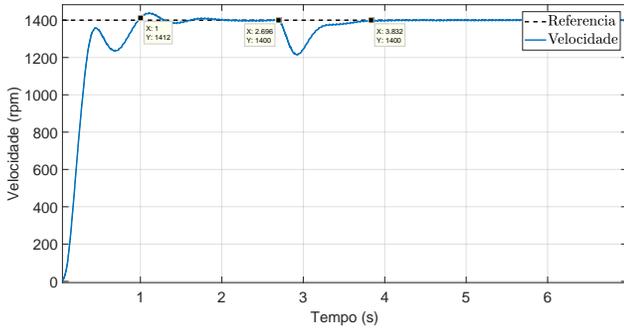


Figura 9. Resposta para o controlador PI para um degrau de 1400 RPM

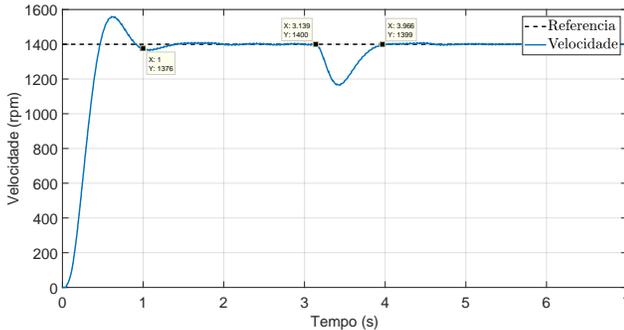


Figura 10. Resposta para o controlador PI-RLS para um degrau de 1400 RPM

entretanto, no momento em que houve aplicação do distúrbio, houve uma queda brusca no valor desse parâmetro, percebe-se que o mesmo tendeu ao valor de 20,53. De forma análoga, na Figura 12, é mostrada a estimação para o parâmetro $\hat{\tau}$, que inicialmente estava tendendo para 0,22, e com a aplicação do distúrbio alterou-se para 0,18.

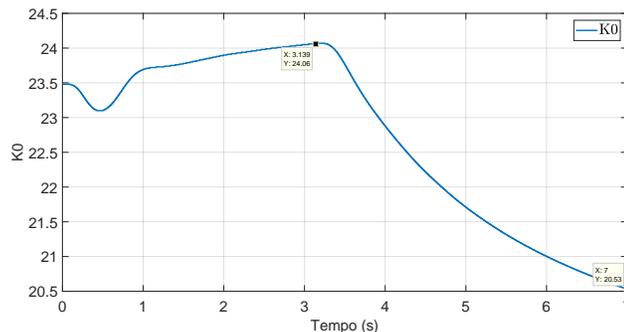


Figura 11. Estimação do parâmetro \hat{K}_0

VI. CONCLUSÃO

Neste trabalho foi apresentado uma comparação experimental de desempenho dos controles PI e PI-RLS aplicados à um motor *brushless*. Para aferir a robustez dos

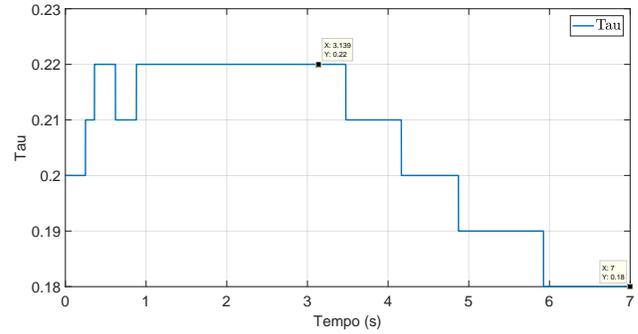


Figura 12. Estimação do parâmetro $\hat{\tau}$

algoritmos, um motor de corrente contínua foi utilizado para gerar distúrbios.

Primeiramente, pode-se concluir que a FT de primeira ordem representa o sistema adequadamente, pois, foi aferido experimentalmente que o controle PI projetado e sintonizado para essa planta simplificada, apresentou bons resultados. Entretanto, para aplicações que requerem grande precisão sugere-se expandir o modelo, incluindo as dinâmicas não modeladas nesse artigo, para o tornar mais verossímil.

Por fim, nota-se que o controle PI-RLS é vantajoso comparado ao PI, pois apresenta menor tempo de convergência da resposta para regime permanente, bem como estabilização do sistema mais rápida, quando aplicado um distúrbio. Além disso, não é necessário o conhecimento prévio da FT do sistema para execução do PI-RLS, pois este algoritmo responde apenas a ação de controle e a derivada do sinal da saída. Também destaca-se a maior agilidade no projeto de controladores, permitindo a utilização em várias máquinas com potências diferentes utilizando o mesmo critério de projeto.

REFERÊNCIAS

- [1] P. Yedamale, “Brushless dc (bldc) motor fundamentals,” *Microchip Technology Inc*, vol. 20, pp. 3–15, 2003.
- [2] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, C. Melchiorri, and G. Palli, “Ubh 3: A biologically inspired robotic hand,” in *proceedings of the IEEE Int. Conf. on Intelligent Manipulation and Grasping*, 2004.
- [3] D. Roper, S. Sharma, R. Sutton, and P. Culverhouse, “A review of developments towards biologically inspired propulsion systems for autonomous underwater vehicles,” *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 225, no. 2, pp. 77–96, 2011.
- [4] E. B. Siqueira, J. L. Mor, R. Z. Azzolin, and V. M. de Oliveira, “Algorithm to identification of parameters and automatic re-project of speed controller of bldc motor,” *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 256–261, 2015.
- [5] J. F. Gieras, *Permanent magnet motor technology: design and applications*. CRC press, 2009.
- [6] A. M. Ahmed, A. Ali-Eldin, M. S. Elksasy, and F. F. Areed, “Brushless dc motor speed control using both pi controller and fuzzy pi controller,” *International Journal of Computer Applications*, vol. 109, no. 10, pp. 29–35, 2015.

- [7] K. Premkumar and B. Manikandan, "Bat algorithm optimized fuzzy pd based speed controller for brushless direct current motor," *Engineering Science and Technology, an International Journal*, vol. 19, no. 2, pp. 818–840, 2016.
- [8] S. W. Khubalkar, A. S. Chopade, A. S. Junghare, and M. V. Aware, "Design and tuning of fractional order pid controller for speed control of permanent magnet brushless dc motor," in *proceedings of the 2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, Kolkata, India, Jan 2016, pp. 326–320.
- [9] H. Du, X. Hu, C. Ma, and G. Zhang, "Numerical pi tuning method and its application in aircraft pitch control," in *proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Siem Reap, Cambodia, June 2017, pp. 1270–1274.
- [10] T. A. Rahmawati, N. . A. Fakhriy, G. N. P. Pratama, A. I. Cahyadi, and S. Herdjunto, "A remedy design of pi controller for liquid level control," in *proceedings of the 2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, March 2018, pp. 661–666.
- [11] M. Fasil, N. Mijatovic, B. B. Jensen, and J. Holboll, "Nonlinear dynamic model of pmbldc motor considering core losses," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 12, pp. 9282–9290, Dec 2017.
- [12] S. F. Toloue and M. Moallem, "Torque control of a brushless dc motor using multivariable sliding mode extremum seeking pi tuning," in *proceedings of the Industrial Electronics Society, IECON 2017-43rd Annual Conference of the IEEE*. Beijing, China: IEEE, 2017, pp. 2917–2922.
- [13] S. Tong, S. Sui, and Y. Li, "Fuzzy adaptive output feedback control of mimo nonlinear systems with partial tracking errors constrained," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 4, pp. 729–742, 2015.
- [14] Y.-J. Liu, L. Tang, S. Tong, and C. P. Chen, "Adaptive nn controller design for a class of nonlinear mimo discrete-time systems," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 5, pp. 1007–1018, 2015.
- [15] "Maxon ec motor," https://www.maxonmotor.com/medias/sys_master/root/8825434996766/17-EN-265.pdf, accessed: 2018-07-37.
- [16] P. J. D. O. Evald, J. L. Mór, and R. Z. Azzolin, "A pole placement control by states feedback based on state variable filter," in *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*. IEEE, 2016, pp. 25–30.
- [17] J. F. Gieras, R.-J. Wang, and M. J. Kamper, *Axial flux permanent magnet brushless machines*. Springer Science & Business Media, 2008.
- [18] K. Ogata, P. Á. Maya, and F. Leonardi, *Engenharia de controle moderno*. Prentice Hall, 2003.
- [19] R. C. Dorf and R. H. Bishop, "Sistemas de controle modernos," *12ª Edição, Editora LTC*, 2009.
- [20] Y. Li, K. H. Ang, and G. C. Y. Chong, "Pid control system analysis and design," *IEEE Control Systems*, vol. 26, no. 1, pp. 32–41, Feb 2006.
- [21] R. P. Vieira, R. Z. Azzolin, and H. A. Grindling, "Parameter identification of a single-phase induction motor using rls algorithm," in *proceedings of the 2009 Brazilian Power Electronics Conference*, Bonito-Mato Grosso do Sul, Brazil, Sept 2009, pp. 517–523.