

Implementação de Algoritmo Genético em Controlador de Motor de Relutância para Diminuição de *Ripple* de Torque

Gabrielle Magalhães Veras Braga¹, Renato Giacomini¹, André Perin¹, Milene Galeti¹

¹ Departamento de Elétrica, Centro Universitário FEI
uniebrag@fei.edu.br

Resumo: O presente artigo científico propõe a aplicação de Algoritmos Genéticos baseados em Inteligência Artificial para a mitigação das ondulações de torque em um Motor de Relutância síncrono (SyRM). A pesquisa explora a utilização dessas técnicas em conjunto com a linguagem de programação Python, visando a otimização do *Ripple* de torque do motor.

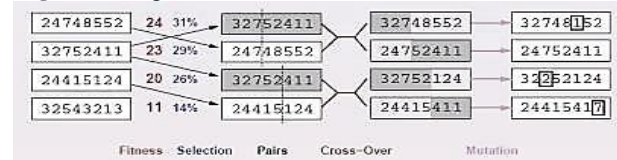
1. Introdução

O Motor de Relutância é uma máquina elétrica que provoca polos magnéticos temporários em seu rotor e gera torque por variação da relutância magnética, sendo esta definida como o critério que mensura a dificuldade causada pelo núcleo para conter o fluxo magnético [1]. São motores de construção simples, alta resistência e baixo custo de produção, porém tendem a gerar torque de baixa velocidade com ondulação, causando ruídos excessivos à operação [2]. Um Motor de Relutância síncrono possui quatro ou seis polos entre o estator e o rotor. A igualdade entre polos se mantém viável devido a aberturas nas áreas do componente móvel do equipamento. A fim de diminuir a relutância do circuito, o eixo direto do rotor alinha e acompanha o campo magnético girante [3].

O Algoritmo Genético (GAs: “Genetic Algorithms”) é um modelo computacional de otimização inspirado no princípio de Seleção Natural proposto por Charles Darwin em sua obra “A Origem”, publicada em 1859, em que o organismo mais apto tende a sobreviver e fixar suas características para as próximas gerações, considerando fatores de hereditariedade, variabilidade e reprodução [4]. O processo de otimização é uma técnica que visa o aumento de desempenho e eficiência, em que consiste na codificação do problema, análise da função objetivo a qual se deseja maximizar ou minimizar e a inserção de um espaço de soluções.

Um Algoritmo Genético inicia o processo com a criação de populações de cromossomos. Assim, são selecionados a partir do cálculo da função objetivo os que obtiveram alta aptidão, e conseqüentemente são gerados os cromossomos filhos (variantes) pelos métodos de Cruzamento e Mutação. As populações intermediárias agrupam os cromossomos pais que foram selecionados anteriormente com as probabilidades equivalentes às aptidões [5]. A Figura 1 retrata a seqüência de execução de um GA nos processos de Seleção, Cruzamento e Mutação.

Figura 1. Algoritmos Genéticos



Fonte: DECSAI [6]

Ripple de Torque é uma reação causada no motor que produz ondulações formadas por modificações no campo magnético e pelas rotações no rotor quando energizado. O Torque de Cogging, que promove trepidações/trancos no motor, contribui para essas alterações [7], embora o principal fator seja a corrente desmagnetizante nos enrolamentos de campo [8].

2. Métodos e Materiais

Com o propósito de investigar a otimização dos parâmetros da forma de onda visando à minimização da oscilação de torque, foi empregada uma implementação do Algoritmo Genético (AG) por meio da linguagem de programação Python. O procedimento se inicia com a primeira fase de simulação, na qual um conjunto inicial de coeficientes de Fourier é selecionado para aproximar uma forma de onda senoidal que representa o perfil de corrente. Esta onda é submetida a uma truncagem até o enésimo harmônico. É válido observar que a composição genética de cada solução guarda correspondência com os coeficientes componentes da série supramencionada. Nesse contexto, uma variante desses cinco coeficientes é gerada, e sua performance é comparada com resultados preexistentes, com o fito de identificar a solução ótima. Tal processo introduz variações na configuração da forma de onda. A subsequente etapa envolve a execução da simulação por um número pré-definido de iterações, viabilizando a avaliação de desempenho de cada conformação da forma de onda, sob o prisma da oscilação de torque.

Ao longo da abordagem do Algoritmo Genético (AG), utilizou-se um percentual de cruzamento de 50%, permitindo uma ponderação equilibrada entre os pais durante a reprodução, e mutações aleatórias para promover uma exploração abrangente do espaço de soluções. A população inicial consistiu em 25 indivíduos gerados aleatoriamente, e o critério de parada foi definido em 5 gerações. Os desfechos de cada simulação nas diversas iterações são retratados em um gráfico, conferindo assim a possibilidade de análise visual da oscilação de torque. Por fim, ao término do processo conduzido pelo AG, a solução de forma de onda caracterizada pela oscilação de torque minimizada é eleita como seleção final.

Através da utilização do Algoritmo Genético, torna-se um método que facilita a exploração de várias opções de formas de onda, ao mesmo tempo em que aprimora gradualmente os coeficientes subjacentes.

3. Resultados

Neste estudo, procedeu-se à implementação do Algoritmo Genético (AG) utilizando o ambiente de programação Python. O propósito foi a determinação do conjunto ótimo de cinco coeficientes de Fourier. A etapa inicial da simulação envolveu a modelagem de uma onda senoidal por meio da aplicação dos coeficientes [1, 0, 0, 0, 0].

Posteriormente, o segundo coeficiente foi incorporado como uma variável aleatória, a fim de gerar o coeficiente resultante (filho). A introdução de mutações nos coeficientes teve como finalidade a identificação da solução mais vantajosa. Esta simulação foi executada ao longo de um número pré-determinado de iterações cíclicas e após a conclusão deste processo, os dados resultantes foram visualizados graficamente, sendo a solução final determinada com base nos resultados acumulados.

A sequência de codificação teve início com a simulação inicial da oscilação de referência, utilizando os coeficientes padrão, chamados de “pai_1”. A dinâmica do torque resultante foi representada graficamente em função do deslocamento angular. Essa representação contemplou as métricas de torque mínimo, médio e máximo, considerando simultaneamente a manifestação do Ripple de torque. A Tabela 1 apresenta os valores obtidos.

Tabela 1. Resultados das Simulações

Coefficientes	Torque máximo [N.m]	Torque mínimo [N.m]	Torque médio [N.m]	Ripple
[1, 0, 0, 0, 0]	0.216960	-0.001055	0.117173	0.218015
[0.89004607 0.07435444 0.05898471 0.26628612 0.23614292]	0.201633	0.002845	0.099736	0.198788
[0.73663591 0.1406138 0.03093387 0.01840131 0.38755424]	0.176180	-0.021900	0.060764	0.198080
[0.70952028 0.23553254 0.36835548 0.1894619 0.08275165]	0.145699	-0.002684	0.074679	0.148384

Fonte: Autor

A partir desta, pôde-se calcular as diferenças percentuais entre o ripple de referência [1,0,0,0,0] e o ripple dos melhores coeficientes encontrados pela equação:

$$Ripple\% = \frac{Ripple_{Ref} - Ripple_{encontrado}}{Ripple_{Ref}} * 100\% \quad (1),$$

cujas soluções corresponderam a 8,82%, 9,14% e 31,94%, respectivamente.

Cumprе ressaltar que as conclusões alcançadas constituem um processo evolutivo dentro desta pesquisa. A próxima etapa envolverá o refinamento do código de

simulação de otimização. Isso será realizado por meio da implementação de uma estrutura precisa de *Cruzamento*, visando a redução da aleatoriedade nos resultados e a convergência em direção à solução final.

4. Conclusões

Em síntese, trata-se como objetivo do projeto a redução de oscilações indesejadas de torque em Motores de Relutância síncronos (SyRM). Através deste pressuposto, realizou a aplicação de Algoritmos Genéticos e da linguagem de programação Python, como uma ferramenta para aprimorar a qualidade do torque gerado pelo motor.

Ao utilizar a modelagem da forma de onda usando coeficientes de Fourier, o estudo explora como o uso do Algoritmo Genético pode melhorar a configuração dos cinco coeficientes da série de Fourier, com o objetivo de minimizar as oscilações de torque conhecidas como Ripple de torque. O processo inclui iterações em que diferentes combinações de coeficientes foram testadas e avaliadas quanto ao impacto no comportamento do motor.

Os resultados obtidos após a execução das simulações e iterações do Algoritmo Genético revelaram uma solução de forma de onda que demonstra a redução nas oscilações de torque.

5. Referências

- [1] JF-PAREDE. Motor de Relutância: Funcionamento e Aplicações. Disponível em: <https://jf-parede.pt/what-is-reluctance-motor>. Acesso em: 10 agosto 2023.
- [2] GPAR. Controle de Motor de Relutância Variável. Disponível em: https://gpar.ufc.br/?page_id=990 Acesso em: 10 agosto 2023.
- [3] Marques S., Alexandre M. Motor Síncrono de Relutância. p. 20-21.
- [4] Charles Darwin: biografia e seleção natural. Disponível em: <https://brasilecola.uol.com.br/biologia/charles-darwin.htm> Acesso em: 10 agosto 2023.
- [5] AURÉLIO, M.; PACHECO, C.; PACHECO, C. Algoritmos Genéticos: princípios e aplicações. Disponível em: https://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_apost.pdf Acesso em: 10 agosto 2023.
- [6] BARZAL, F. Genetic Algorithms. Disponível em: <https://elvex.ugr.es/decsai/iaio/slides/G2%20Genetic%20Algorithms.pdf> Acesso em: 10 agosto 2023.
- [7] Emetor. Torque ripple. Disponível em: <https://www.emetor.com/glossary/torque-ripple/> Acesso em: 10 agosto 2023.
- [8] Kollmorgen. Torque de cogging e Ripple de Torque. Disponível em: <https://www.kollmorgen.com/pt-br/blogs/tudo-sobre-torque-de-cogging-e-ripple-de-torque-o-que-voc%C3%AA-precisa-saber> Acesso em: 10 agosto 2023.

Agradecimentos

À instituição Centro Universitário FEI pelo suporte financeiro.

¹ Aluno de IC do Centro Universitário FEI (PIBIC-FEI). Projeto com vigência de 09/2023 a 09/2024.