

Usando Algoritmos Genéticos Para Otimizar um Controlador PID

1st Renan Medke

PPGEE Programa de Pós Graduação em Engenharia Elétrica

Universidade Federal de Santa Maria - UFSM

Santa Maria-RS, Brasil

renanmedke@gmail.com

Abstract—This article aims to show simulation results of a parallel PID controller, improved by genetic algorithm for the plant of a buck converter.

Index Terms—Buck, PidTune, PID, Algoritmo Genético

I. INTRODUÇÃO

A busca por otimização em produtos e projetos é um dos grandes desafios da engenharia, pois há necessidade de produzir um mesmo produto com o menor custo possível, tornando-o mais acessível as massas ou então simplesmente tornando-o mais eficiente. No caso de controladores busca-se torná-los estáveis com o menor custo computacional possível, buscando algoritmos que melhorem ou evitem empecilhos já conhecidos, como por exemplo, derivadas e descontinuidades das funções, buscando contornar esses problemas é possível utilizar métodos heurísticos e meta heurísticos na solução das equações de ordem elevada.

Esse trabalho da disciplina Processamento de Sinais terá como objeto de estudo e análise a otimização por meio de algoritmos genéticos de um controlador PID, cujos valores foram obtidos com o uso do comando PIDtune do matlab para um conversor Buck.

II. CONVERSOR E ESTRUTURA DE CONTROLE

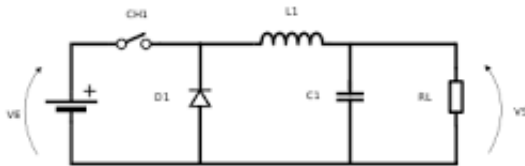


Fig. 1. Conversor Buck.

O conversor usado para análise é o Buck, conforme figura 1, que será controlado por uma estrutura PID. Controladores PID possuem três componentes para atuar sobre o erro medido em relação a um sinal de referência, uma Proporcional, uma Integrativa e outra Derivativa, que funcionam, nesse caso, de forma paralela, conforme figura 3.

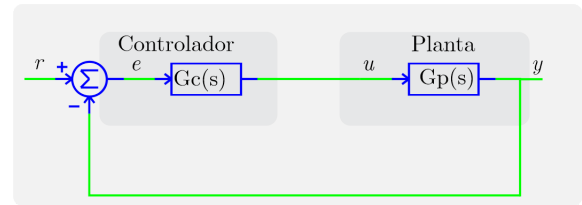


Fig. 2. Estrutura de controle usada.

Considerando a estrutura da Figura 2, o compensador pode ser obtido de várias maneiras, entre elas estão os métodos clássicos, nesse caso foi obtido através do PIDtune do matlab [3]. Foram considerados os seguintes parâmetros: $L = 1mH$, $C = 100\mu F$ e $R = 10ohms$, frequência de chaveamento de $20kHz$, $Vo = 5V$ e $Vin = 10V$. Considerando esses parâmetros e buscando controlar a tensão de saída por meio da razão cíclica, temos a seguinte função de transferência:

$$Gp(s) = \frac{1 * 10^8}{s^2 + 1000s + 1 * 10^7} \quad (1)$$

O retorno do PIDtune do matlab obteve a função de transferência do compensador:

$$Gc(s) = \frac{161.1s^2 + 9.131 * 10^5 s + 1.233 * 10^9}{s^2 + 1.43 * 10^6 s} \quad (2)$$

III. ALGORITMOS GENÉTICOS

Métodos heurísticos, conforme [1], buscam soluções para uma determinada equação sem um método clássico matemático, eles basicamente jogam valores e monitoram a saída, achando resultados bons. Algoritmos Genéticos são métodos meta heurísticos bioinspirados, que buscam satisfazer uma equação colocando valores para seus parâmetros. Monitorando seu resultado acham soluções possíveis dentro

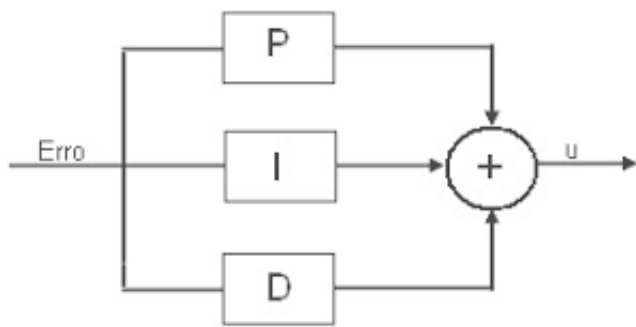


Fig. 3. Estrutura de controle usada.

de um intervalo, e dentro desse intervalo buscam achar a melhor solução local. O matlab possui uma biblioteca que usa algoritmos genéticos [2] para melhorar a solução já existente de determinada equação, no caso será usada a função de transferência obtida na equação (2) para passar por esse melhoramento e assim obtemos a seguinte equação de compensador:

$$Gc(s) = \frac{1810s^2 + 2.368 * 10^6 s + 1.462 * 10^{10}}{s^2 + 3.533 * 10^6 s} \quad (3)$$

IV. RESULTADOS E SIMULAÇÕES

Dando um comando step no matlab para os compensadores obtidos nas equações (2) e (3) temos:

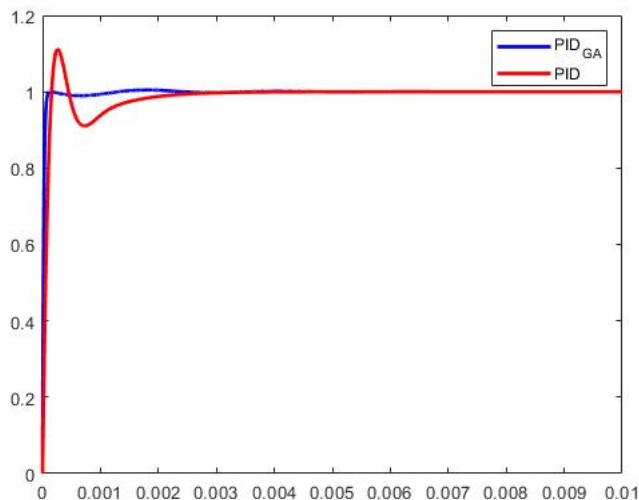


Fig. 4. Comparação entre as soluções.

Usando os compensadores obtidos realizou-se uma simulação no software psim e obtivemos o seguinte resultado da figura 6.

Observa-se que em regime permanente o compensador melhorado pelo algoritmo genético apresenta melhor resposta a distúrbios, no primeiro caso um degrau de 20% a menos na tensão de entrada, o compensador em verde que tem um

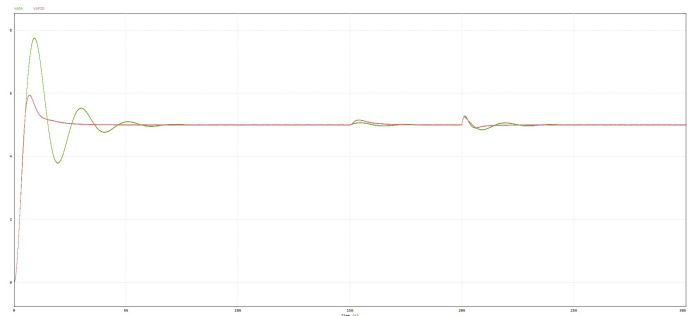


Fig. 5. Resultado da simulação.

menor overshoot e para o caso de um degrau de 50% a menos de carga também há um menor overshoot.

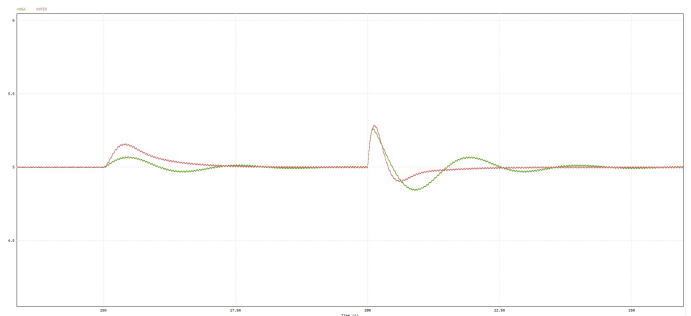


Fig. 6. Detalhe da simulação.

REFERENCES

- [1] COLIN Emerson Carlos, Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas, Rio de Janeiro:LTC, 2007.
- [2] MATLAB, Find minimum of function using genetic algorithm Matlab,Disponível em: <https://www.mathworks.com/help/gads/ga.html>. Acesso em: 01 de Julho 2023
- [3] MATLAB, PID tuning algorithm for linear plant model,Disponível em: <https://www.mathworks.com/help/control/ref/dynamicsystem.pidtune.html>. Acesso em: 25 de Junho 2023