Relatório de Atividade

Sintonia de Controladores e Estimação de Parâmetros

Débora Oliveira
Prof Antonio Marcus, Automação Inteligente 20.3

10 de novembro de 2020

Esse documento tem por objetivo descrever a sintonia do controlador PID de um robô de tração diferencial (RTD) a partir de algoritmos genéticos. Para aprender essa técnica de busca, primeiramente foi otimizado um problema de localização das raízes de uma função. Em seguida, empregou-se o código desenvolvido para a sintonia do controlador do RTD a partir da simulação do espaço de estados no ambiente MATLAB. Por fim, esses ganhos foram implementados em um controlador discretizado pela aproximação de Tustin,*backward-euler* e *foward-euler* utilizando *Zero-order hold* (ZOH) para um modelo simulado de um Pioneer P3DX na plataforma CoppeliaSim.

1 FUNDAMENTAÇÃO TEÓRICA

Algoritmo genético é uma técnica de busca de soluções ótimas. Esse método se fundamenta no processo biológico da seleção natural, no qual os indivíduos mais adequados são selecionados para gerar a próxima geração de soluções candidatas.

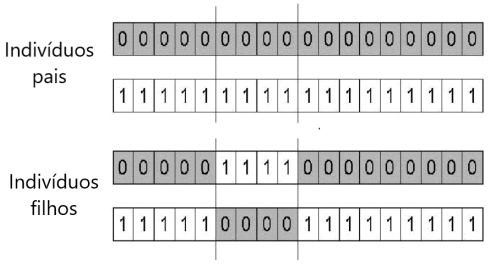
Na construção de algoritmos genéticos, o termo indivíduo se refere à solução candidata. Os genes do individuo equivalem aos bits da solução. Uma solução de N bits pode representar M variáveis.

Analogamente ao processo de reprodução dos seres vivos, os bits das soluções candidatas mais adequadas serão trocados e unidos, conforme ilustrado na Fig. 1. Esse cruzamento pode ser realizado uma ou mais vezes sobre a cadeia de bits. Uma iteração desse cruzamento é nomeado geração.

Assim como a teoria da seleção natural, é proposto que os indivíduos-filho das soluções ótimas de uma geração são mais adequados a sobrevivência. O número de indivíduos por geração é chamado população.

A função $f(x)$ que indica o quão próximo o indivíduo x se localiza da solução ótima do problema é denominada função de aptidão (*fitness*). Essa função pode ser determinada como

Fig. 1. Diagrama do cruzamento entre dois indivíduos pais para a geração de duas cadeias de bit.



Fonte: (RAHMANI et al., 2011).

$$f(x) = \frac{1}{J(x)} \tag{1}$$

para $J(x)$ a função de custo.

A função $J(x)$ deve ser construída conforme o problema de otimização analisado. Quanto menor $J(x)$, maior será $f(x)$ e a proximidade do indivíduo x da solução ótima.

Os códigos utilizados para esse trabalho são baseados no algoritmo genético "speedyGA" (MITCHELL, 1996). Para cada problema proposto, foram modificados o tamanho da população, a quantidade de gerações, número de bits N da solução candidata e a função de custo.

2 LOCALIZAÇÃO DAS RAÍZES

O primeiro problema de otimização é a busca pelas raízes da função $y(x)$ dada por

$$y(x) = 2e^{-0.1x} \sin x \tag{2}$$

Observando a equação (2), fica claro que $y(x) = 0$ para $x = n\pi$, $\forall n \in \mathbb{Z}$.

Para a localização das raízes de y, foram propostos indivíduos de 16 bits com sinal codificados em Q2.13. Dessa forma, x possui 1 bit de sinal, 2 bits de expoente e 13 bits de mantissa. A conversão do binário x é equivalente ao decimal $(-1)^s m(2^{-13})$, para

s o valor do bit de sinal e m o hexadecimal composto pelos bits 2 a 16. Logo, $x \in [-2^2 + 2^{-13}, 2^2 - 2^{-13}]$

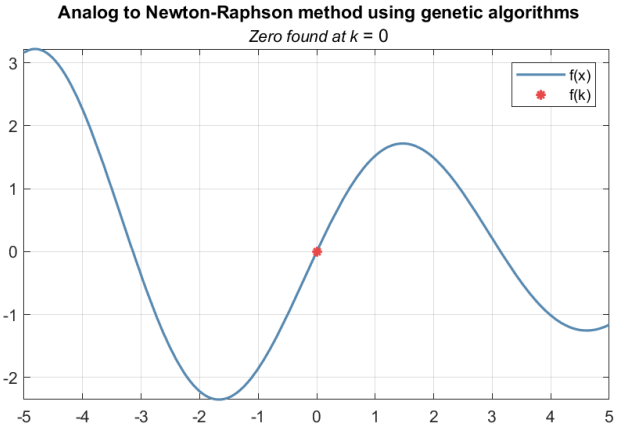
A função de custo utilizada foi

$$J(x) = |y(x)| \tag{3}$$

uma vez que esse custo é mínimo para valores próximos a $y(x) = 0$.

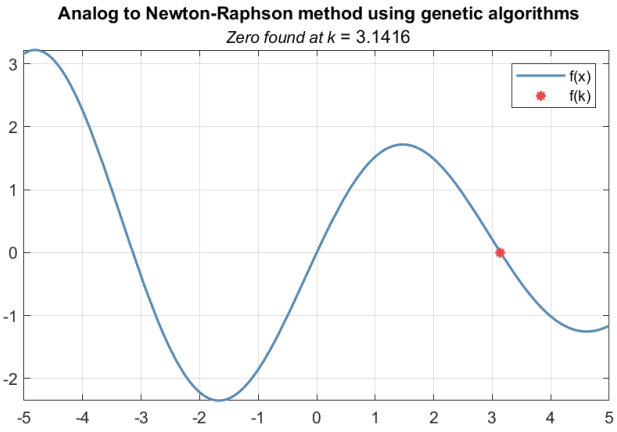
Tendo em vista que a primeira população é gerada aleatoriamente, em duas execuções foi possível encontrar dois zeros da função $y(x)$. Os zeros encontrados foram $x = 0$ e $x = \pi$, conforme ilustrado na Fig. 2 e Fig. 3.

Fig. 2. Otimização do método de Newton-Raphson utilizando algoritmos genéticos para a curva $y(x)$ com a solução ótima dada por $x = 0$.



Fonte: Autoria própria.

Fig. 3. Otimização do método de Newton-Raphson utilizando algoritmos genéticos para a curva $y(x)$ com a solução ótima dada por $x = \pi$



Fonte: Autoria própria.

3 Ajuste de ganhos de um controle PID

Conforme documentado nas atividades anteriores, o espaço de estados para um RTD cuja velocidade linear ν é constante e observável e a velocidade angular w é controlável é

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \nu \cos\phi \\ \nu \sin\phi \\ \omega \end{bmatrix} \tag{4}$$

O erro de entrada do sistema de controle é

$$e = \phi - \arctan\left(\frac{y_g - y}{x_g - x}\right) \tag{5}$$

para $[x_g\ y_g]$ as coordenadas do objetivo e $[x\ y]$ a localização atual do RTD no sistema de coordenadas universal.

A função de transferência do controlador PID é

$$G(s) = \frac{\Omega(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d \frac{p_d s}{s + p_d} \tag{6}$$

para p_d o polo do filtro derivativo. A função $G(s)$ determinada pela equação (6) pode ser fatorada em duas funções de transferência de modo que $G(s) = G_1(s)G_2(s)$. Logo, obtém-se

$$G_1(s) = \frac{\Omega(s)}{Z(s)} = (k_p s + k_i)(s + p_d) + k_d p_d s^2 \tag{7}$$

$$G_2(s) = \frac{Z(s)}{E(s)} = \frac{1}{s^2 + p_d s} \tag{8}$$

A representação em espaço de estados de $G_2(s)$ definida na equação (7) é

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -p_d & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e \tag{9}$$

Por sua vez, a função de transferência $G_1(s)$ definida na equação (7) pode ser reescrita como

$$\Omega(s) = Z(s)[s^2 k_p(1 + p_d) + s(k_p p_d + k_i) + k_i p_d] \tag{10}$$

Transformando equação (10) para o domínio do tempo e substituindo a equação para $\ddot{z}$ apresentada na equação (9), encontra-se $\omega(z, \dot{z}, e)$.

$$\omega = (k_i - k_p p_d^2)\dot{z} + k_i p_d z + (k_p + k_d p_d)e \tag{11}$$

Por fim, pode-se unir a equação (9), a equação (11) e a equação (4) em um único espaço de estados:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \nu \cos\phi \\ \nu \sin\phi \\ (k_i - k_p p_d^2)\dot{z} + k_i p_d z + (k_p + k_d p_d)e \\ \dot{z} \\ -p_d \dot{z} + e \end{bmatrix} \tag{12}$$

O espaço de estados apresentado na equação (12) foi implementado e solucionado no MATLAB para $k_p = 0.1$, $k_i = 0.1$, $k_d = 0.01$ e $p_d = 50$ utilizando

o método de Runge-Kutta (`ode45`). Essa sintonia foi otimizada utilizando algoritmos genéticos.

O indivíduo foi composto por 60 bits. Essa cadeia binária representa três variáveis (k_p, k_i e k_d), cada uma com 20 bits. Como os três ganhos devem ser positivos e menores que a unidade, a conversão de hexadecimal (k_{hex}) para decimal (k_{dec}) foi realizada conforme a formulação a seguir:

$$k_{dec} = \frac{k_{hex}}{2^{20}}$$

Para buscar os ganhos que produzissem o menor erro $e(t)$ ao longo de toda a trajetória, para a sintonia do PID foram testadas quatro funções de custo.

1. Integral do erro absoluto

$$J(x) = IAE = \int_0^{\inf} |e(t)|dt$$

2. Integral do erro médio quadrático

$$J(x) = ISE = \int_0^{\inf} e^2(t)dt$$

3. Integral do tempo multiplicado pelo erro absoluto

$$J(x) = ITAE = \int_0^{\inf} t|e(t)|dt$$

4. Integral do tempo multiplicado pelo erro absoluto

$$J(x) = ITSE = \int_0^{\inf} te^2(t)dt$$

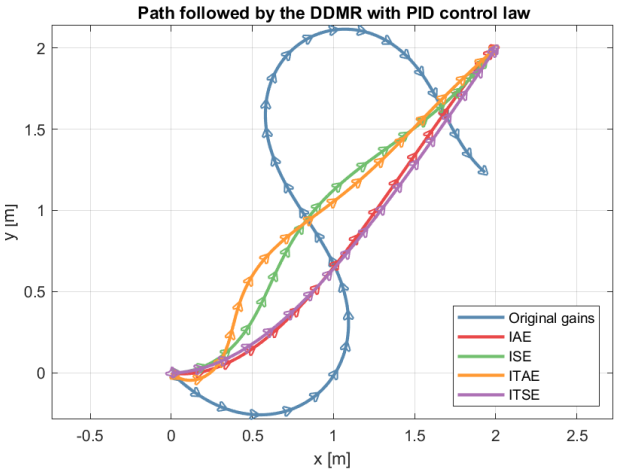
A população foi definida em 40 indivíduos cruzados em 500 gerações. A sintonia foi realizada para as condições iniciais $[x, y, \theta] = [-1.5, -1.5, 0]$ e a velocidade linear foi determinada constante $\nu = 0.25$m/s. A posição alvo foi definida em $[x_g, y_g] = [2, 2]$.

Os resultados das trajetórias simuladas a partir da equação (12) com os ganhos com a maior função de adaptação para a última geração de cada função de custo estão ilustradas nas Fig. 4. Os ganhos sintonizados para cada função de custo estão apresentados na Tab. 1.

Tab. 1. Ganhos do controlador PID sintonizados para as funções de custo IAE, ISE, ITAE, ITSE utilizando algoritmo genético *speedyGA*.

	k_p	k_i	k_d
IAE	0.9652	0.0268	0.6572
ISE	1.0	0.9992	1.0
ITAE	1.0	0.9999	0.0
ITSE	1.0	0.0039	1.0

Fig. 4. Otimização da sintonia dos ganhos de um controlador PID empregando as funções de custo IAE, ISE, ITAE, ITSE.



Fonte: Autoria própria.

Analisando as trajetórias ilustradas na Fig. 4 e os ganhos sintonizados descritos na Tab. 1, é evidente que a suavidade da curva percorrida pelo RTD está associada ao termo k_i. Quanto maior a constante do termo integral, mais acentuadas são as curvas executadas pelo RTD. Essa conclusão independe da magnitude do termo derivativo k_d, uma vez que os comportamentos são semelhantes para a sintonia com a função de custo ITAE e ISE, as quais possuem k_d nos limites opostos do intervalo de ganho $k_d \in [0, 1]$.

Observando a Fig. 4, fica claro que as trajetórias mais suaves são encontradas para as funções de custo IAE e ITSE, as quais sintonizaram ganhos k_i pequenos. Já as funções de custo ISE e ITAE sintonizaram ganhos k_i próximos ao limite da unidade. O termo derivativo k_d é responsável pelo amortecimento da trajetória para a função ISE em relação à função de custo ITAE.

4 Discretização do controlador PID

Para o teste dos ganhos sintonizados apresentados na Tab. 1, foi implementado no modelo simulado do P3DX no CoppeliaSim a versão discretizada da equação (6). De modo geral, uma lei de controle discretizada é dada no domínio da transformada Z por

$$G(z) = \frac{\Omega(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \qquad (13)$$

A equação (13) pode ser transferida para o domínio do tempo, resultando a seguinte saída $\omega(h)$ do controlador:

$$\begin{aligned}\omega(h) = &-\frac{a_1}{a_0}\omega(h-1) - \frac{a_2}{a_0}\omega(h-2) \\ &+ \frac{b_0}{a_0}e(h) + \frac{b_1}{a_0}e(h-1) \\ &+ \frac{b_2}{a_0}e(h-2)\end{aligned} \tag{14}$$

para h o número do passo de simulação. O polo do filtro derivativo p_d foi renomeado N.

Para a aproximação de Tustin, considera-se que

$$s = \frac{2(z-1)}{T_s(z+1)} \tag{15}$$

para T_s o passo da simulação. No caso da cena simulada no CoppeliaSim, o passo foi determinado em 50ms.

Substituindo a equação (15) na equação (6), obtém se as seguintes constantes quando $G(z)$ é posta no formato da equação (13):

$$\begin{aligned}b_0 &= (2k_p + k_iT_s)\left(1 + \frac{NT_s}{2}\right) + 2k_dN \\ b_1 &= -4k_p + k_iT_s^2N - 4k_dN \\ b_2 &= (2k_p - k_iT_s)\left(1 - \frac{NT_s}{2}\right) + 2k_dN \\ a_0 &= 2 + NTs \qquad a_1 = -4 \qquad a_2 = 2 - NTs\end{aligned} \tag{16}$$

Já para a aproximação em *foward-euler*, tem-se

$$s = \frac{(z-1)}{T_s} \tag{17}$$

Substituindo a equação (17) na equação (6), obtém se as seguintes constantes quando $G(z)$ é posta no formato da equação (13):

$$\begin{aligned}b_0 &= k_p + Nk_d \\ b_1 &= k_p(NT_s - 2) + k_iT_s - 2K_dN \\ b_2 &= (k_p - k_iT_s)(1 - NT_s) + k_dN \\ a_0 &= 1 \qquad a_1 = NT_s - 2 \qquad a_2 = 1 - NTs\end{aligned} \tag{18}$$

Por fim, para a aproximação em *backward-euler*, considera-se

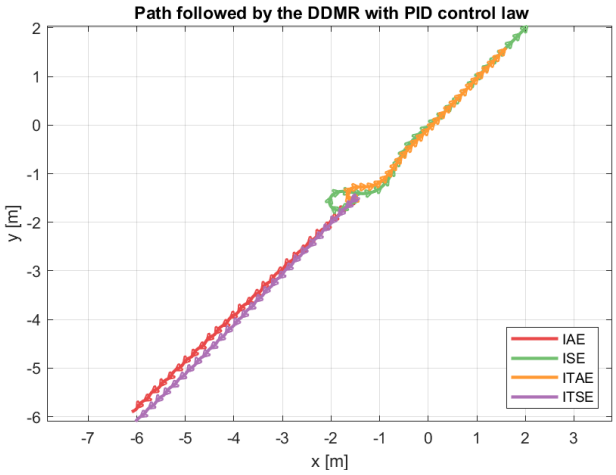
$$s = \frac{(z-1)}{zT_s} \tag{19}$$

Substituindo a equação (19) na equação (6), obtém se as seguintes constantes quando $G(z)$ é posta no formato da equação (13):

$$\begin{aligned}b_0 &= (k_p + k_iT_s)(1 + NT_s) + k_dN \\ b_1 &= -[k_p(2 + NT_s) + k_iT_s + 2k_dN] \\ b_2 &= k_p + k_dN \\ a_0 &= 1 + NT_s \qquad a_1 = -(2 + NT_s) \qquad a_2 = 1\end{aligned} \tag{20}$$

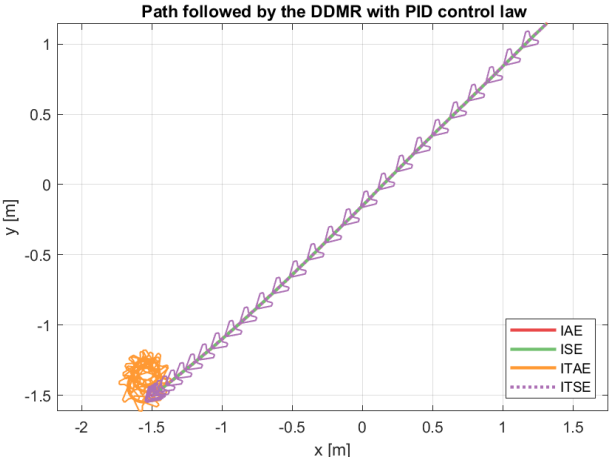
Os controladores discretos descritos conforme a equação (16), equação (18) e equação (20) foram construídos em linguagem Lua para o P3DX e simulados no CoppeliaSim. Na Fig. 5, Fig. 6 e Fig. 7 estão ilustradas as trajetórias percorridas pelo robô simulado com o controlador discreto respectivamente com a aproximação de Tustin, *foward-euler* e *backward-euler* para os ganhos apresentados na Tab. 1.

Fig. 5. Trajetória percorrida pelo P3DX simulado com o controlador discreto aproximado por Tustin do controlador PID sintonizado a partir da otimização com algoritmo genético empregando as funções de custo IAE, ISE, ITAE, ITSE.



Fonte: Autoria própria.

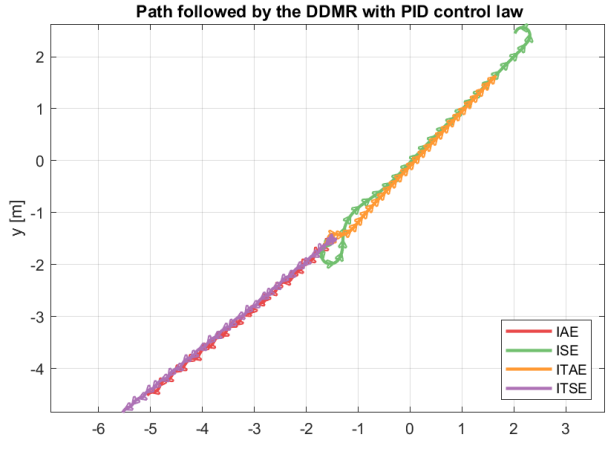
Fig. 6. Trajetória percorrida pelo P3DX simulado com o controlador discreto aproximado por *foward-euler* do controlador PID sintonizado a partir da otimização com algoritmo genético empregando as funções de custo IAE, ISE, ITAE, ITSE.



Fonte: Autoria própria.

Comparando a Fig. 5 e a Fig. 7, fica clara a semelhança entre as aproximações que consideram a existência de polos (Tustin e *backward-euler*), conforme a equação (15) e equação (19).

Fig. 7. Trajetória percorrida pelo P3DX simulado com o controlador discreto aproximado por *backward-euler* do controlador PID sintonizado a partir da otimização com algoritmo genético empregando as funções de custo IAE, ISE, ITAE, ITSE.



Fonte: Autoria própria.

Ambas as aproximações de Tustin e *backward-euler* do controlador discreto apresentam instabilidade para k_i aproximadamente nulos sintonizados para as funções de custo IAE e ITSE.

Já a aproximação por *foward-euler* apresenta a mesma trajetória para os ganhos sintonizados para as funções de custo IAE, ISE e ITSE. A instabilidade ocorre para o controlador sintonizado utilizando a função de custo ITAE, uma vez que o termo derivativo k_d é nulo.

Analisando a Fig. 5, Fig. 6 e Fig. 7, fica claro que o comportamento do controlador discreto não é equivalente à trajetória percorrida pelo modelo de equações diferencias da equação (12) apresentada na Fig. 4. Essa observação é justificada pela diferença entre os passos do controlador simulado no MATLAB pseudo-contínuo e o controlador discreto simulado em linguagem Lua no CoppeliaSim. Enquanto aquele é solucionado pelo método de Runge-Kutta (`ode45`) para passos de aproximadamente 17ms, esse último foi simulado com passo de 50ms.

5 Conclusão

Esse relatório descreve a implementação e análise de dois algoritmos genéticos para a busca da solução ótima de dois problemas analíticos: a localização das raízes de uma função e a sintonia de um controle PID para um robô de tração diferencial.

Comparando os resultados obtidos pelo algoritmo genético e o método clássico de localização das raízes por Newton-Raphson, é evidente que aquele é uma solução mais genérica que o último. Uma vez que a inicialização dos valores iniciais de busca são aleatórios a cada nova população de indivíduos, é ultrapassado o obstáculo do reconhecimento de apenas um ponto mínimo-local — o qual é característico da solução numérica de Newton-Raphson.

O uso de uma algoritmo genético para a sintonia de um controle PID apresentou resultados rápidos e tão eficientes quanto os ganhos obtidos a partir das estratégias de sintonia clássica, conforme apresentado na Fig. 4. O programa poderia ser otimizado considerando outras funções de custo, como o erro médio absoluto. Também poderiam ser modificados as fronteiras dos ganhos representados pelos 44 bits, os quais foram limitados à unidade, e as condições inicias de sintonia para $\theta = 0.25\pi$. Por fim, o algoritmo genético produzido poderia ser adaptado à sintonia de um controlador discreto, a partir da adaptação do espaço de estados apresentado na equação (12). Essa modificação poderia permitir a busca por uma solução ótima mais adequada a simulação implementada no CoppeliaSim.

6 Referências

[1] RAHMANI, Amir; SHIRGAHI, Hossein. **Grid resources selection optimization with quality of service guarantee by a hybrid algorithm of genetic and particle swarm optimization**. International Journal of the Physical Sciences, v. 6, nov 2011. DOI 10.5897/IJPS11.858.

[2] MITCHELL, Melanie. Genetic Algorithms: An Overview. In: **An Introduction to Genetic Algorithms**. 1. ed. Boston: MIT Press, 1996. v. 1, cap. 1, p. 1-31. ISBN 9780262133166.