

Sistemas Inteligentes: Otimização Multi-Objetivo

Nelson Carlos de Sousa Campos

Maio de 2016

1 Introdução

Os motores de indução são amplamente utilizados para condução de cargas mecânicas no sistema comercial e industrial, devido ao seu baixo custo e confiabilidade. [6]

Os parâmetros do modelo de circuito equivalente são, em geral, determinados através dos ensaios de rotor bloqueado e sem carga, que são clássicos como está prescrito no padrão IEEE 112 (IEEE Standard Test Procedure for Polyphase Induction Motors and Generators, 2004). Os parâmetros obtidos a partir dos dados de testes clássicos não representam o comportamento da máquina em toda a faixa de escorregamento ($s \in [0, 1]$) [5]. Isto é porque os parâmetros da máquina de indução não são constantes ao longo de todo o intervalo de variação de escorregamento. Assim, é necessário modificar os parâmetros iniciais obtidos a partir dos testes clássicos, a fim de melhorar a descrição do modelo e reduzir a diferença entre o modelo real e os parâmetros estimados. [6].

2 Modelo da Máquina

O modelo de circuito equivalente por fase para uma de três fases fonte de tensão simétrico da máquina de indução é ilustrado na Figura 1. As descrições dos parâmetros da máquina são listadas na Tabela 1.

As equações de corrente e torque da máquina são descritas por (1) e (2), respectivamente, onde os A, B, C e D são definidas de (3) a (6).

$$I(s, \theta) = V \sqrt{\frac{C^2 + D^2}{A^2 + B^2}} \quad (1)$$

$$T(s, \theta) = 3V^2 \frac{p}{\omega} \frac{R_{fe}^2 R_r}{(A^2 + B^2)s} \quad (2)$$

$$A = R_s \left(\frac{R_r}{s} + R_{fe} \left(1 + \frac{X_r}{X_m} \right) \right) + X_s \left(\frac{R_{fe} R_r}{s X_m} - X_r \right) + \frac{R_{fe} R_r}{s} \quad (3)$$

$$B = X_s \left(\frac{R_r}{s} + R_{fe} \left(1 + \frac{X_r}{X_m} \right) \right) \quad (4)$$

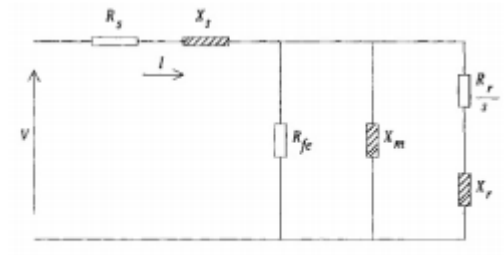


Figura 1: Circuito equivalente da máquina, fonte: [6]

Tabela 1: Parâmetros elétricos do motor de indução

Potência	1.5kW
Voltagem	220V
Corrente	5.8A
Frequência	60Hz
Número de pólos	2
Torque	8.0N
R_s	1.93
X_s	1.658
R_{fe}	310
X_m	38.7
R_r	3.84
X_r	6.789

$$C = \frac{R_r}{s} + R_{fe} \left(1 + \frac{X_r}{X_m}\right) \quad (5)$$

$$D = \frac{R_{fe} R_r}{s X_m} - X_r \quad (6)$$

3 Otimização Multi-Objetivo

Uma maneira de resolver o problema de estimação da máquina de indução é considerar uma coleção de funções objetivos e, portanto, adotar uma abordagem de otimização multi-objetivo (MOO). Neste caso, cada função objetivo pode ter uma solução ótima diferente e, se isso for verdade, as funções objetivo são conhecidas como conflitantes. Quando as funções objetivos são conflitantes, o problema resulta em um conjunto de soluções ao invés de uma solução única. As soluções no conjunto de soluções são definidas para serem ótimas. Elas são frequentemente chamadas de **Conjuntos de Pareto**. [6] [2]

Tabela 2: Parâmetros estimados

Rr	3.72
Xr	5.98
Xs	2.32
Rs	1.85
Xm	37.8

4 Algoritmos Genéticos

No domínio da inteligência artificial, um algoritmo genético (AG) é uma pesquisa heurística que imita o processo de selecção natural. Esta heurística (também chamada às vezes metaheurística) é rotineiramente usada para gerar soluções úteis em problemas de otimização e busca. [1] Algoritmos genéticos podem tratar qualquer tipo de problema cuja função tenha vários mínimos ou até mesmo quando a função não tem informação de gradiente ou não seja diferenciável. [6]. Eles também são algoritmos paralelos e são adequados para resolver problemas onde o espaço de soluções possíveis é grande. AG usam a linguagem da microbiologia e operações genéticas no desenvolvimento de novas soluções candidatas.

Uma população representa um grupo de potenciais pontos de solução, uma geração representa a iteração, um cromossomo é comparável a um design point (componente simples) e um gene é comparável a uma componente estrutura de design points.

O algoritmo genético multi-objetivo é um variante do Non-dominated Sorting Genetic Algorithm-II (NSGA-II), que utiliza um algoritmo genético elitista controlado. Em algoritmos não-elitistas, o operador genético pode destruir algumas das soluções não dominantes para explorar o espaço de soluções. O elitismo pode acelerar o desempenho do algoritmo genético de forma significativa, e também ajuda a evitar a perda de boas soluções uma vez que tenham sido encontradas. [4] [3]

5 Resultados e Conclusões

A estimação dos parametros da máquina foi realizada com a aplicação de de algoritmos genéticos em otimização multi-objetivo, onde uma função minimiza o erro médio quadrático da corrente e do torque, com o tamanho da população como sendo 200. Os resultados estão ilustrados na Tabela 2. Os códigos das simulações estão anexados nos códigos 1 e 2.

O resultado da simulação está ilustrado na Figura 2. Os resultados se mostraram satisfatórios e a aplicação de algoritmos genéticos com a otimização multi-objetivo validou as medições pseudo-experimentais, onde as incertezas das medições foram simuladas com adição de ruído branco.

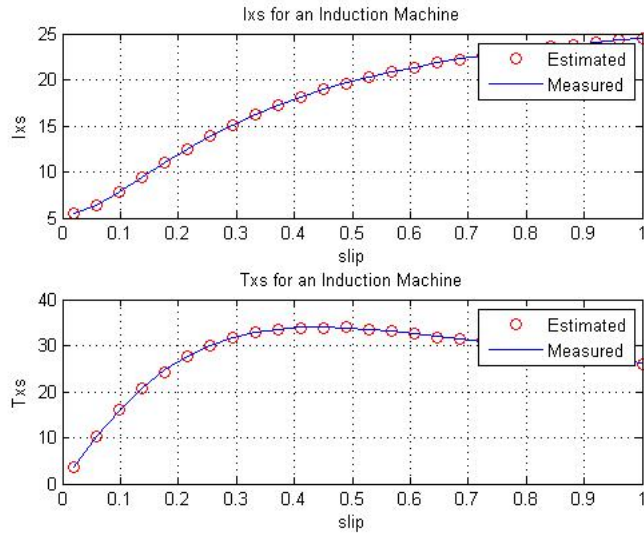


Figura 2: Resultados da estimação da corrente e torque

Referências

- [1] Genetic algorithm. https://en.wikipedia.org/wiki/Genetic_algorithm.
- [2] Multi-objective optimization. https://en.wikipedia.org/wiki/Multi-objective_optimization.
- [3] Arshad Afzal and Kwang-Yong Kim. Three-objective optimization of a staggered herringbone micromixer. *Sensors and Actuators B: Chemical*, 192:350–360, 2014.
- [4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
- [5] Antonio Marcus Nogueira Lima, Cursino Brandão Jacobina, and Eurico Bezerra de Souza Filho. Nonlinear parameter estimation of steady-state induction machine models. *Industrial Electronics, IEEE Transactions on*, 44(3):390–397, 1997.
- [6] Joao B. M. SANTOS and ANTONIO M. N. Lima. Multi-objective identification of induction machine models. *Simpósio Brasileiro de Automação Inteligente*, 2015.

6 Anexos

Código 1: Estimação de parâmetros de uma máquina indução

```
%
%clear all
close all
%
global N V p w medidas
%
Rr=3.84;
Xr=6.789;
Xs=1.658;
Rs=1.93;
Xm=38.7;
%
N=25;
V=220;
f=60;
w=2*pi*f;
p=2;
In=5.8/50;
Pn=1.5e3/50;
Tn=8.0/50;
s=[0.02:(1-0.02)/N:1.0]';
for i=1:length(s),
%
A=Rs*(1+Xr/Xm)+(1+Xs/Xm)*Rr/s(i);
B=Xr+Xs*(1+Xr/Xm)-Rs*Rr/Xm/s(i);
C=1+Xr/Xm;
D=Rr/Xm/s(i);
Im(i,1)=V*sqrt((C*C+D*D)/(A*A+B*B));
% Zeq(i,1) = sqrt(-1)*Xm*(Rr/s(i)+sqrt(-1)*Xr)/(Rr/s(i)+
    sqrt(-1)*(Xr+Xm)) + Rs+sqrt(-1)*Xs;
% theta = angle(Zeq(i,1));
% fpm(i,1) = cos(theta);
% Pam(i,1) = 3*V*V/abs(Zeq(i,1))*fpm(i,1);
% Prm(i,1) = 3*V*V/abs(Zeq(i,1))*sin(theta);
% %
% Im(i,1)=V*sqrt((C*C+D*D)/(A*A+B*B))+In*randn();
% Pm(i,1)=3*V*V*(A*C-B*D)/(A*A+B*B)+Pn*randn();
Tm(i,1)=3*V*V*(p/w)*(Rr/s(i))/(A*A+B*B)+Tn*randn();
end
%
medidas=[s Im Tm];
%
x0 = (1/1)*[3 6 2 1 30]';
lb = [0 0 0 0 0]';
```

```

ub = [10 10 5 5 50]';
%
options=optimset('Display','iter','TolCon',1e-12,'TolFun',
    1e-12,'TolX',1e-12);
%x=lsqnonlin(@fcnls,x0,lb,ub,options);
FitnessFunction = @fcnls;
numberOfVariables = 5;
options = gaoptimset('PopulationSize',200);
x = gamultiobj(FitnessFunction,numberOfVariables
   ,[],[],[],lb,ub,options);
msg = ['ideal _Rr=_ num2str(Rr) ',_Xr=_ num2str(Xr)
    ',_Xs=_ num2str(Xs) ',_Rs=_ num2str(Rs) ',_Xm=_
    num2str(Xm) ];
disp(msg)
%
linha=size(x,2);
Rr=x(linha,1);
Xr=x(linha,2);
Xs=x(linha,3);
Rs=x(linha,4);
Xm=x(linha,5);
%
msg = ['final _Rr=_ num2str(Rr) ',_Xr=_ num2str(Xr)
    ',_Xs=_ num2str(Xs) ',_Rs=_ num2str(Rs) ',_Xm=_
    num2str(Xm) ];
disp(msg)
%
for i=1:length(s)
A=Rs*(1+Xr/Xm)+(1+Xs/Xm)*Rr/s(i);
B=Xr+Xs*(1+Xr/Xm)-Rs*Rr/Xm/s(i);
C=1+Xr/Xm;
D=Rr/Xm/s(i);
% Ie(i,1)=V*sqrt((C*C+D*D)/(A*A+B*B));
% Zeq(i,1) = sqrt(-1)*Xm*(Rr/s(i)+sqrt(-1)*Xr)/(Rr/s(i)+
    sqrt(-1)*(Xr+Xm)) + Rs+sqrt(-1)*Xs;
% theta = angle(Zeq(i,1));
% fpe(i,1) = cos(theta);
% Pae(i,1) = 3*V*V/abs(Zeq(i,1))*fpe(i,1);
% Pre(i,1) = 3*V*V/abs(Zeq(i,1))*sin(theta);

%
Ie(i,1)=V*sqrt((C*C+D*D)/(A*A+B*B));
%Pe(i,1)=3*V*V*(A*C-B*D)/(A*A+B*B);
Te(i,1)=3*V*V*(p/w)*(Rr/s(i))/(A*A+B*B);
end

```

```

%
figure(1)
subplot(2,1,1)
plot(s,Im,'ro',s,Ie,'b-'), grid, title('I\texttimes s'),
      title('Ixs_for_an_Induction_Machine'), ylabel('Ixs'),
      xlabel('slip'), legend('Estimated', 'Measured')
subplot(2,1,2)
plot(s,Tm,'ro',s,Te,'b-'), grid, title('T\texttimes s'),
      title('Txs_for_an_Induction_Machine'), ylabel('Txs'),
      xlabel('slip'), legend('Estimated', 'Measured')
% subplot(6,1,3)
% plot(s,Pm,'ro',s,Pe,'k-'), grid, title('P \texttimes s')
% subplot(6,1,4)
% plot(s,fpm,'ro',s,fpe,'k-'), grid, title('fp \texttimes s')
% subplot(6,1,5)
% plot(s,Pam,'ro',s,Pae,'k-'), grid, title('Pa \texttimes s')
% subplot(6,1,6)
% plot(s,Prm,'ro',s,Pre,'k-'), grid, title('Q \texttimes s')
%

```

Código 2: Função multi-objetivo

```

function f = fcnl(x)
% global N V p w medidas
% %
% s=medidas(:,1);
% Im=medidas(:,2);
% %
% Rr=x(1);
% Xr=x(2);
% Xs=x(3);
% Rs=x(4);
% Xm=x(5);
% %
% for i=1:length(s),
%   A=Rs*(1+Xr/Xm)+(1+Xs/Xm)*Rr/s(i);
%   B=Xr+Xs*(1+Xr/Xm)-Rs*Rr/Xm/s(i);
%   C=1+Xr/Xm;
%   D=Rr/Xm/s(i);
%   Ic(i,1)=V*sqrt((C*C+D*D)/(A*A+B*B));
% end
% %
% f = Im-Ic;
global medidas

V = 220;

```

```

f=60;
w=2*pi*f;
p=2;
Rr=x(1);
Xr=x(2);
Xs=x(3);
Rs=x(4);
Xm=x(5);

s = medidas(:,1);
y=zeros(size(s,2),2);
for i=1:length(s)
    A=Rs*(1+Xr/Xm)+(1+Xs/Xm)*Rr/s(i);
    B=Xr+Xs*(1+Xr/Xm)-Rs*Rr/Xm/s(i);
    C=1+Xr/Xm;
    D=Rr/Xm/s(i);
    y(i,1) = V*sqrt((C*C+D*D)/(A*A+B*B)); %Ic
    y(i,2) = 3*V*V*(p/w)*(Rr/s(i))/(A*A+B*B); %Tc
end

f(1)=sum((y(:,1)-medidas(:,2)).^2);
f(2)=sum((y(:,2)-medidas(:,3)).^2);
end

```