TAREA 5. ESTIMACIÓN DE PARÁMETROS DE LA MÁQUINA DE INDUCCIÓN

Abstract. En el presente trabajo se utilizará un método dinámico lineal para la estimación de los parámetros propios de la Máquina de Inducción (MI) que incluyen la resistencia de rotor, las inductancias de rotor y estator y la inductancia mutua. La resistencia del estator se tiene como dato. El método a utilizar incluye el cálculo de parámetros que minimizan la función costo asociada a los errores cuadráticos, utillizando para ello tres puntos de operación con cargas de rotor distintas. Los resultados obtenidos coinciden con los valores reales salvo en el caso de la inductancia mutua, en donde existe cierto margen de error.

1. Estimación con Método Dinámico y Modelo Lineal

Este método parte del modelo transitorio en coordenadas estatóricas, el cual se expresa de la siguiente manera:

$$\begin{bmatrix} \mathbf{v_e} \\ \mathbf{v_r^e} \end{bmatrix} = \begin{bmatrix} R_e & 0 \\ 0 & R_r \end{bmatrix} \begin{bmatrix} \mathbf{i}_e \\ \mathbf{i}_r^e \end{bmatrix} + \begin{bmatrix} L_e & L_{er} \\ L_{er} & L_r \end{bmatrix} p \begin{bmatrix} \mathbf{i}_e \\ \mathbf{i}_r^e \end{bmatrix} - j\omega \begin{bmatrix} 0 & 0 \\ L_{er} & L_r \end{bmatrix} \begin{bmatrix} \mathbf{i}_e \\ \mathbf{i}_r^e \end{bmatrix}$$

$$T_e - T_m = L_{er}\Im m\{\mathbf{i_e}.(\mathbf{i}_r^e)^*\}$$

Si la velocidad mecánica es constante, esta ecuación es lineal y al tratarse de un rotor jaula de ardilla, la tensión del circuito rotórico es nula. Se procede a sustituir la derivada de la corriente de rotor en la ecuación del estator luego de derivarla una vez con respecto al tiempo y finalmente resulta:

(0.1)
$$p\mathbf{v_e} = \left(L_e - \frac{L_{er}^2}{L_r}\right)p^2\mathbf{i_e} + \left(R_e + R_r\frac{L_e}{L_r} - jw\left(L_e - \frac{L_{er}^2}{L_r}\right)\right)p\mathbf{i_e} + \left(j\omega - \frac{R_r}{L_r}\right)(\mathbf{v_e} - R_e\mathbf{i_e})$$

Se reescribe la ecuación0.1 de forma tal de dejar a los miembros que no dependan de los parámetros a estimar de un lado y del otro los que son dependientes, agrupando los coeficientes en las constantes k_1, k_2, k_3. De esta forma, tomando en cuenta que la resistencia de estator es un parámetro medible de forma directa y por ende es conocida, la ecuación queda de la siguiente forma:

$$p\mathbf{v_e} - j\omega\mathbf{v_e} + R_e j\omega\mathbf{i_r} = k_1(p^2\mathbf{i_e} - j\omega\mathbf{i_e}) + k_2 p\mathbf{i_e} + k_3(R_e\mathbf{i_e} - \mathbf{v_e}) \tag{0.2}$$

En donde:

$$k_1 = \left(L_e - \frac{L_{er}^2}{L_r}\right); \quad k_2 = R_e + R_r\frac{L_e}{L_r}; \quad k_3 = \frac{R_r}{L_r};$$

TABLE 1. Resultados Obtenidos

Parámetro	Valor Real	Valor Estimado
R_e	0.03	-
R_r	0.03	0.0299
L_e	0.13	0.13
L_r	0.13	0.13
L_{er}	0.02	0.03

La ecuación requiere de tres medidas linealmente independientes para poder determinar por regresión lineal estos coeficientes. Es necesario usar al menos tres puntos de operación con diferente carga en el eje del rotor y con estas medidas se procederá a construir la función de costos. La función con los parámetros que minimizan este 'error' tiene la siguiente forma:

$$[k] = \left[\sum_{i=1}^{3} [w_i]^t [w_i]\right]^{-1} \sum_{i=1}^{3} [w_i]^t [h_i]$$

Donde:

$$[h_i] = p\mathbf{v_e} - j\omega\mathbf{v_e} + R_e j\omega\mathbf{i_e}$$

$$[w_i] = \left[(p^2\mathbf{i_e} - j\omega p\mathbf{i_e}),\ \ p\mathbf{i_e},\ \ (R_e\mathbf{i_e} - \mathbf{v_e})\right]$$

Y una vez obtenidos los coeficientes $[k]$, se pueden obtener los parámetros de la máquina con las siguientes relaciones:

$$L_r = \frac{k_2 - R_e}{k_3}; \quad R_r = k_2 - R_r; \quad L_{er} = \sqrt{(k_1 - L_e)(\frac{k_2 - R_e}{k_3})}$$

2. Implementación y Resultados

El modelo de máquina a utilizar para este método de estimación será el de coordenadas arbitrarias, con el caso de referencia al estator ya que es una de las premisas al deducir el modelo lineal. El tiempo de corrida de la simulación será suficiente para permitir que la máquina alcance velocidad constante con tres cargas distintas, las cuales incluyen vacío, rotor bloqueado y carga nominal. La velocidad alcanzada en cada medida de carga puede apreciarse en la figura 0.1. Una vez alcanzada velocidad constante en estos tres puntos de operación, se toma una muestra de las variables de estado: el voltaje de estator y su derivada; la corriente de estator y sus derivadas primera y segunda; y a partir de ellas se hace el calculo de $[h_i]$ y $[w_i]$ para hallar los tres coeficientes k. Los resultados obtenidos en la estimación de parámetros se muestran en la tabla 1.

El código en lenguaje MATLAB puede ser visualizado en el apéndice A.

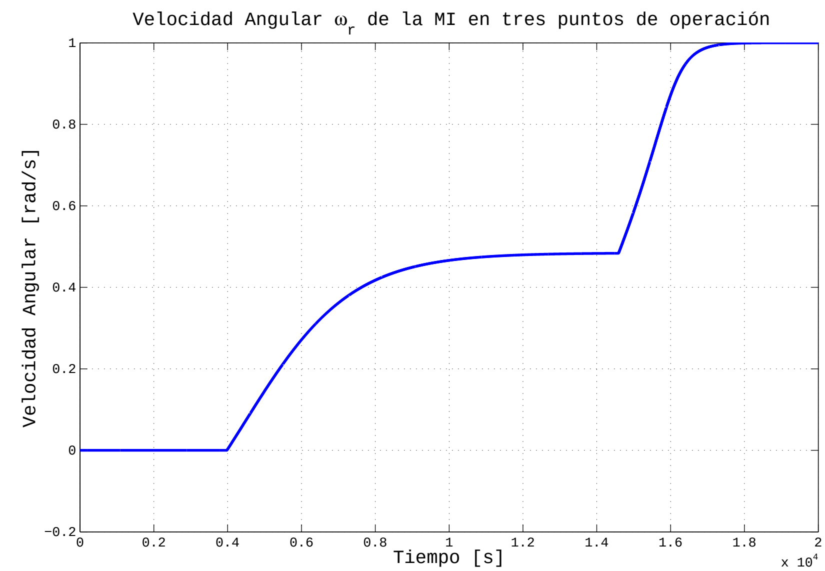

FIGURE 0.1. Velocidad Angular ω_r de la MI en tres puntos de operación

Apendice A. Códigos Fuente de las simulaciones en Lenguaje MATLAB

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PROGRAMA PRINCIPAL ESTIMACION DE PARAMETROS DE LA MI %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global Re Rr Le Lr VS Ler M Tr R L_1 J kl

%Parametros de la MI en por unidad
Re=0.03; Rr=0.03; Le=0.13; Lr=0.13; Ler=0.02; M=3*Ler/2; Tr=Lr/Rr; H=1;
kl=1; p=2; f=60; ws=2*pi*f/p; J=2*H*ws;

VS=sqrt(2/3)*[1 exp(1i*2*pi/3) exp(1i*4*pi/3)];

%Matrices para el modelo
R = [Re 0;0 Rr]; L=[Le M;M Lr]; L_1=inv(L);
x0 = [0,0,0,0,0]; t0=0; t=0:0.001*377:20*377;

% Llamada a la función de integración
[T,X]=ode45(@maquina_ep_ca,t,x0);

ie_delta = X(:,1); ir_delta = X(:,2);
wr = X(:,3); theta = X(:,4); delta = X(:,5);

%Cálculo de Par
Te=M*imag(ie_delta.*conj(ir_delta));

% Paso a coordenadas primitivas
iae=sqrt(2/3)*real(ie_delta.*exp(1i*delta));
ibe=sqrt(2/3)*real(ie_delta.*exp(1i*delta)*exp(1i*4*pi/3));
ice=sqrt(2/3)*real(ie_delta.*exp(1i*delta)*exp(1i*2*pi/3));

iar=sqrt(2/3)*real(ir_delta.*exp(1i*(delta-theta)));
```

```
ibr=sqrt(2/3)*real(ir_delta.*exp(1i*(delta-theta)).*exp(1i*4*pi/3));
icr=sqrt(2/3)*real(ir_delta.*exp(1i*(delta-theta)).*exp(1i*2*pi/3));

% Variables de estado
Ve=exp(1i*T)/sqrt(2/3);
Ie=sqrt(2/3)*(iae+ibe*exp(1i*2*pi/3)+ice*exp(1i*4*pi/3));
Ir=sqrt(2/3)*(iar+ibr*exp(1i*2*pi/3)+icr*exp(1i*4*pi/3));

% Derivadas de las variables de estado
pVe = 1i*Ve;
pIe = 1i*Ie;
p2Ie = -Ie;

% Ora forma de derivar
% pVe = diff(Ve)/(0.001*377);
% pIe = diff(Ie)/(0.001*377);
% p2Ie = diff(diff(Ie))/(0.001*377)/(0.001*377);

% Tres medidas para distintas cargas (puntos)
punto_op = [ 0.2 0.7 0.9];

% Inicialización de Variables Intermedias para la Regresión
wi=zeros(length(punto_op),3);
hi=zeros(length(punto_op),1);
NUM = zeros(3,1); DEN = zeros(3);

% Calculo de coeficientes
for temp=1:length(punto_op)
    m=floor(punto_op(temp)*length(Ve));

    % Cuando son 5 coeficientes
    %wi(temp,:)=[(p2Ie(m)-1i*wr(m)*pIe(m)), pIe(m), -1i*wr(m)*Ir(m), -Ve(m), Ie
      (m)];
    %hi(temp)=pVe(m)-1i*wr(m)*Ve(m);

    % Cuando son 3 coeficientes
    wi(temp,:)=[(p2Ie(m)-1i*wr(m)*pIe(m)), pIe(m), (Re*Ie(m)-Ve(m))];
    hi(temp)= pVe(m)-1i*wr(m)*Ve(m)+1i*Re*wr(m)*Ie(m);

    NUM = wi(temp,:)'*hi(temp) + NUM;
    DEN = wi(temp,:)'*wi(temp,:) + DEN;

end

K = DEN\NUM;

Rr_est = real((K(2)-Re));
Lr_est = real((K(2)-Re)/K(3));
Le_est = Lr_est;
Ler_est= real(sqrt((Le_est-K(1))*Lr_est));

var = [Rr_est,Lr_est,Ler_est];
disp(var)
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MODELO MI EN COORD ARBITRARIAS ESTIMACION DE PARAMETROS  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function px=maquina_ep_ca(t,x)

global Re Rr Le Lr VS Ler M Tr R L_1 J kl

ie_delta=x(1); ir_delta=x(2); wr=x(3); theta=x(4); delta=x(5);

ve = VS*[cos(t);cos(t-2*pi/3);cos(t-4*pi/3)]; vr = 0;

% TRANSFORMACION A COORDENADAS DELTA
ve_delta = ve*exp(-1i*delta);
vr_delta = vr*exp(-1i*delta);

p_delta = 0; %Alineado con la fase a del estator
%p_delta = wr; %Alineado con la fase a del rotor

% PUNTOS DE CARGA
if t<1500
    Tm=M*imag(ie_delta*conj(ir_delta)); % Rotor bloqueado
elseif (t>=1500)&&(t<5500)
    Tm=kl*wr^2;                         % Carga Nominal
else
    Tm=0;                               % Vacio
end

% MODELO
G = [p_delta*Le p_delta*M; (p_delta-wr)*M (p_delta-wr)*Lr];

pi_delta = L_1 * ([ve_delta;vr_delta] - R*[ie_delta;ir_delta] - ...
           1i*G*[ie_delta;ir_delta]);

pie_delta = pi_delta(1,1);
pir_delta = pi_delta(2,1);

pwr=(M*imag(ie_delta*conj(ir_delta))-Tm)/J;
px = [pie_delta;pir_delta;pwr;wr;p_delta];

end
```