

TAREA 4.B: MÁQUINA DE INDUCCIÓN EN COORDENADAS ARBITRARIAS

ORLANDO TREJO - 12-88982

ABSTRACT. En el presente trabajo se muestran los resultados de la simulación en MATLAB de una Máquina de Inducción (MI) en coordenadas de arbitrarias. En este modelo se refieren todos los vectores espaciales a una posición arbitraria δ , realizando la rotación respectiva. Se muestran los resultados referentes al Par Eléctrico, la Velocidad Angular del Rotor y las Corrientes de Fase del Estator y del Rotor, trabajando todas las unidades físicas en el sistema por unidad.

1. Modelo de la Máquina de Inducción en Coordenadas Arbitrarias

El modelo de la MI utilizando una referencia arbitraria implica hacer rotaciones de los vectores espaciales del modelo de la MI para alinearlos con una referencia que puede estar variando en el tiempo. Se denomina a esta referencia δ y los vectores espaciales son transformados para seguir esta referencia. La deducción del modelo comienza con hacer las rotaciones a los vectores espaciales de voltaje de estator y rotor:

$$\mathbf{v}_e^\delta = \mathbf{v}_e e^{-j\delta}$$

Al tratarse de un rotor jaula de ardilla, $\mathbf{v}_r = 0$.

La derivada del ángulo δ será la velocidad de movimiento del sistema de referencia. A modo ilustrativo, se puede seleccionar $p\delta = 0$ si se trabajará con el sistema de referencia orientado con la fase a del estator (fijo); o $p\delta = \omega_r$ si el sistema de referencia estará orientado con la fase a del rotor, moviéndose a velocidad ω_r . El modelo completo en coordenadas arbitrarias queda de la siguiente forma:

$$\begin{bmatrix} \mathbf{v}_e^\delta \\ \mathbf{v}_r^\delta \end{bmatrix} = \begin{bmatrix} R_e & 0 \\ 0 & R_r \end{bmatrix} \begin{bmatrix} \mathbf{i}_e^\delta \\ \mathbf{i}_r^\delta \end{bmatrix} + \begin{bmatrix} L_e & M_{er} \\ M_{er} & L_r \end{bmatrix} p \begin{bmatrix} \mathbf{i}_e^\delta \\ \mathbf{i}_r^\delta \end{bmatrix} + j \begin{bmatrix} \dot{\delta} L_e & \dot{\delta} M_{er} \\ (\dot{\delta} - \dot{\theta}) M_{er} & (\dot{\delta} - \dot{\theta}) L_r \end{bmatrix} \begin{bmatrix} \mathbf{i}_e^\delta \\ \mathbf{i}_r^\delta \end{bmatrix}$$

Y la ecuación de la velocidad:

$$M_{er} \Im \{ \mathbf{i}_e^\delta (\mathbf{i}_r^\delta)^* \} - T_m = J.p\omega_r$$

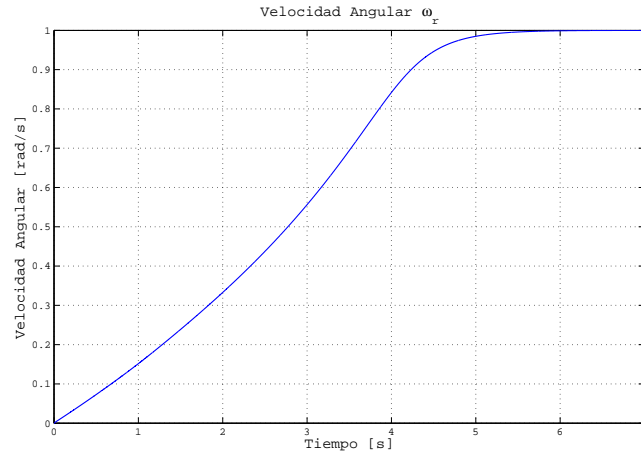
En este caso, se tomará como referencia la fase a del rotor, por lo que $p\delta = \omega_r$.

2. Simulación de la MI en Coordenadas Arbitrarias

En la tabla 1 se muestran los parámetros intrínsecos de la MI, los cuales serán utilizados en la simulación junto con el modelo en coordenadas arbitrarias mostrado anteriormente.

TABLE 1. Parámetros de la Máquina de Inducción a Simular

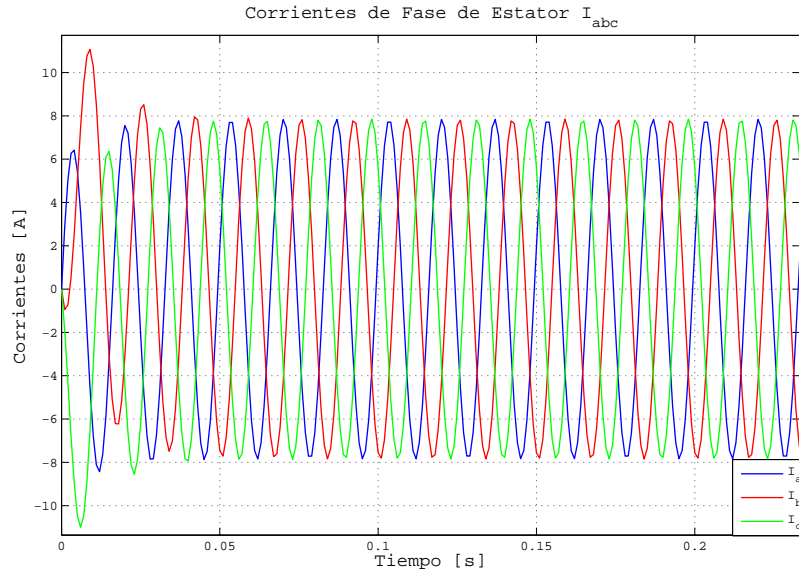
Parámetro	Valor en Por Unidad
$L_{\gamma e}$	0.1
$L_{\gamma r}$	0.1
L_{er}	0.02
L_e	$L_{\gamma e} + \frac{3}{2}L_{er} = 0.13$
L_r	$L_{\gamma r} + \frac{3}{2}L_{er} = 0.13$
M	$\frac{3}{2}L_{er} = 0.03$
R_e	0.03
R_r	0.03
J	$2H\omega_B = 377$

FIGURE 0.1. Velocidad Angular ω_r de la MI

El algoritmo de la simulación se inicia con la carga de parámetros y con la expresión del voltaje de estator en vectores espaciales. Luego se realiza la transformación a coordenadas δ y se sigue el alineamiento del ángulo δ con la referencia seleccionada. Luego se resuelven las ecuaciones del modelo mostrado anteriormente. Las condiciones iniciales son cero para las cinco variables diferenciales del modelo. Este modelo es presentado frecuentemente como una mejora del modelo de campo orientado ya que no presenta las limitantes de valor inicial de la corriente de magnetización modificada. El código en MATLAB se muestra en el apéndice A.

En la **figura 0.1** se puede observar la curva de la velocidad angular del rotor, se aprecia que la MI alcanza la velocidad sincrónica en un tiempo de 6 segundos.

En la **figura 0.2** se puede apreciar en detalle las formas de onda de la corriente de estator en el arranque, en coordenadas primitivas luego de aplicar la antitransformada δ (multiplicar por $e^{j\delta}$). Luego de estabilizarse, estas corrientes tienen la misma forma de onda hasta el final de la simulación.

FIGURE 0.2. Corrientes de Fase de Estator I_{abc}

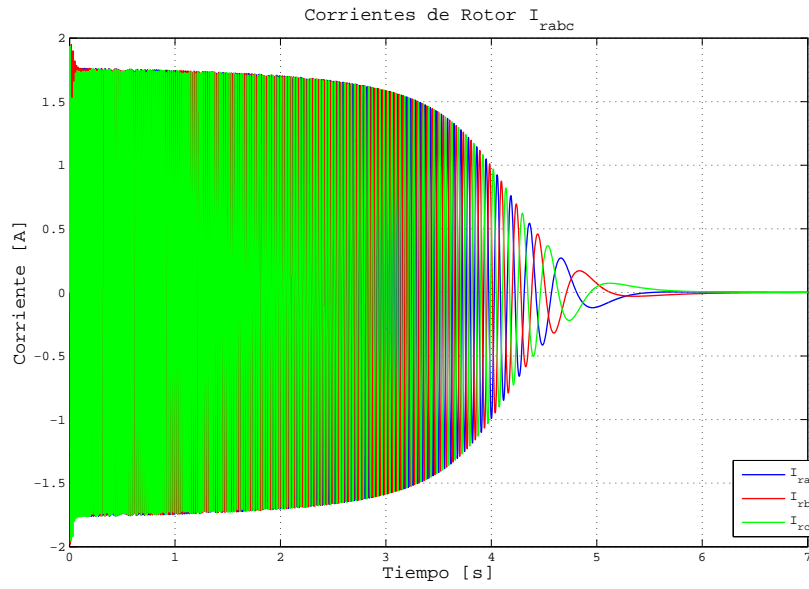
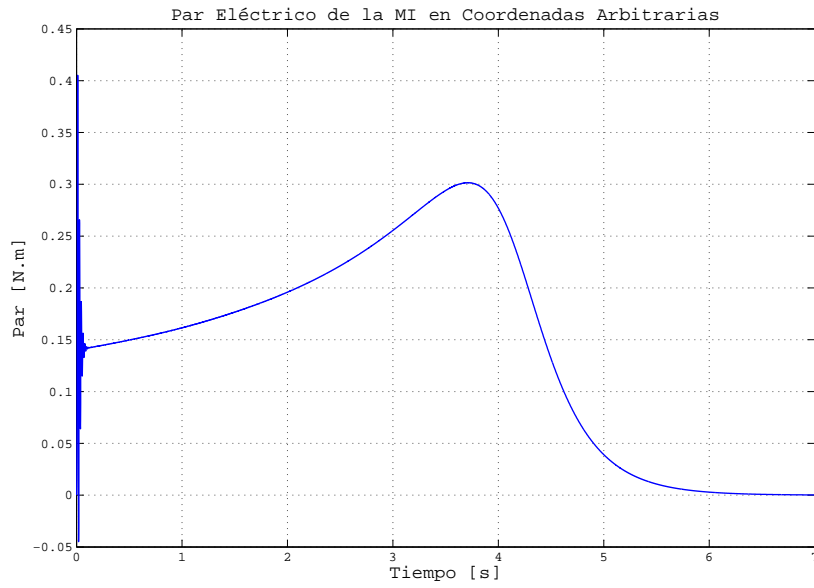
En la **figura 0.3** se pueden apreciar las formas de onda de la corriente de rotor, en coordenadas primitivas, luego de aplicar la antitransformada δ . Se puede observar que después de que la MI alcanza la velocidad sincrónica, las corrientes de rotor se hacen cero. También se observa que la frecuencia de las señales va disminuyendo a medida que se hacen cero, a diferencia de lo observado en el modelo de campo orientado.

En la **figura 0.4** se observa la forma resultante del par eléctrico T_e , en donde se puede observar que el máximo par se obtiene antes de llegar a la velocidad sincrónica y luego va disminuyendo hasta llegar a cero. En comparación con la curva de par eléctrico mostrada usando el modelo en campo orientado, el par máximo es menor y es coherente ya que la velocidad sincrónica se alcanza en un tiempo menor.

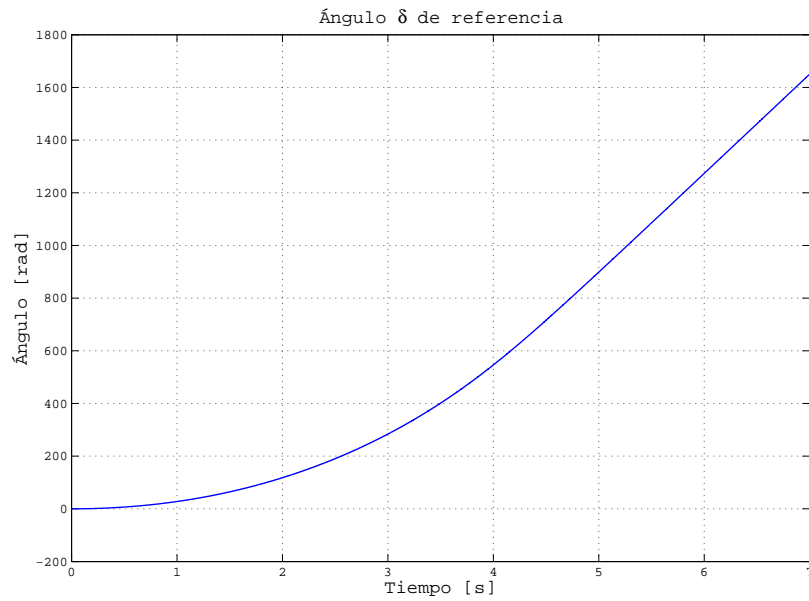
En la **figura 0.5** se aprecia el comportamiento del ángulo δ de referencia del modelo. Se observa que el crecimiento es lineal luego de alcanzar la velocidad sincrónica.

3. Conclusiones

Se logró simular en MATLAB el modelo de la máquina de inducción usando una referencia arbitraria, en este caso la fase A del rotor. Se hizo la prueba colocando otra referencia (fase A del estator) y se obtuvieron resultados similares, lo cual confirma la validez del modelo. Como se dijo anteriormente, este modelo no tiene la inconsistencia inicial que podría tener la corriente de magnetización modificada en el modelo de campo orientado, es por ello que es usado en la práctica para realizar control de diversos tipos. El caso trivial de este modelo consistiría en dejar la referencia fija en el estator y se tendría la ventaja adicional que el resto de las variables de estado a las que se tiene acceso están en el estator.

FIGURE 0.3. Corrientes de Fase de Rotor I_{rabc} FIGURE 0.4. Par Eléctrico T_e

Si bien este modelo presenta ciertas discrepancias con el modelo de campo orientado, los resultados obtenidos son similares y concuerdan con lo descrito en la teoría [1].

FIGURE 0.5. Ángulo δ de referencia

Apéndice A: Programa de Simulación en MATLAB

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % PROGRAMA PRINCIPAL MI EN COORDENADAS ARBITRARIAS %
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  global Re Rr Le Lr VS Ler M Tr R L_1 Tm J
6
7  % Parametros de la MI en por unidad
8  Re=0.03; Rr=0.03; Le=0.13; Lr=0.13; Ler=0.02; M=3*Ler/2;
9  Tr=Lr/Rr; Tm=0; H=1; p=2; f=60; ws=2*pi*f/p; J=2*M*ws;
10
11 VS=sqrt(2/3)*[1 exp(1i*2*pi/3) exp(1i*4*pi/3)];
12
13 % Matrices para el modelo
14 R = [Re 0;0 Rr]; L=[Le M;M Lr]; L_1=inv(L);
15
16 % Tiempo y Valores iniciales de simulación
17 x0 = [0,0,0,0,0]; t0=0; t=0:0.001*377:7*377;
18
19 % Llamada a la función de integración
20 [T,X]=ode45(@maquina_ca,t,x0);
21
22 ie_delta = X(:,1);
23 ir_delta = X(:,2);
24 wr = X(:,3);
25 theta = X(:,4);
26 delta = X(:,5);
27
28 % Cálculo de Par

```

```
29 Te=M*imag(ie_delta.*conj(ir_delta));
30
31 % Paso a coordenadas primitivas
32 iae=sqrt(2/3)*real(ie_delta.*exp(1i*delta));
33 ibe=sqrt(2/3)*real(ie_delta.*exp(1i*delta)*exp(1i*4*pi/3));
34 ice=sqrt(2/3)*real(ie_delta.*exp(1i*delta)*exp(1i*2*pi/3));
35
36 iar=sqrt(2/3)*real(ir_delta.*exp(1i*(delta-theta)));
37 ibr=sqrt(2/3)*real(ir_delta.*exp(1i*(delta-theta)).*exp(1i*4*pi/3));
38 icr=sqrt(2/3)*real(ir_delta.*exp(1i*(delta-theta)).*exp(1i*2*pi/3));
```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % FUNCION MODELO DE LA MI EN COORDENADAS ARBITRARIAS %
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5  function px=maquina_ca(t,x)
6
7  global Re Rr Le Lr VS Ler M Tr R L_1 Tm J
8
9  ie_delta=x(1); ir_delta=x(2); wr=x(3); theta=x(4); delta=x(5);
10
11  % Transformacion de la fuente a Vectores Espaciales
12  ve = VS*[cos(t);cos(t-2*pi/3);cos(t-4*pi/3)]; vr = 0;
13
14  % Transformación a Coordenadas Delta
15  ve_delta = ve*exp(-1i*delta);
16  vr_delta = vr*exp(-1i*delta);
17
18  p_delta = wr; %Alineado con la fase a del rotor
19  %p_delta = 0; %Alineado con la fase a del estator
20
21  % MODELO
22  G = [p_delta*Le, p_delta*M; (p_delta-wr)*M, (p_delta-wr)*Lr];
23
24  pi_delta = L_1 * ([ve_delta;vr_delta] - R*[ie_delta;ir_delta] - ...
25                    1i*G*[ie_delta;ir_delta]);
26
27  pie_delta = pi_delta(1,1);
28  pir_delta = pi_delta(2,1);
29
30  pwr=(M*imag(ie_delta*conj(ir_delta))-Tm)/J;
31
32  px = [pie_delta;pir_delta;pwr;wr;p_delta];
33
34  end

```