

DISEÑO IMPLEMENTACIÓN Y DESARROLLO DE UNA INTERFAZ PARA MOTORES DE INDUCCIÓN

Daniel Fernando Aranda Contreras
Escuela E3T, Universidad Industrial de Santander
Correo electrónico: {daniel2221648}@correo.uis.edu.co

Index Terms—Motor de Inducción, Interfaz Gráfica de Usuario (GUI), MATLAB App Designer, Cálculo de Parámetros, Máquinas Eléctricas, Modelado de Motores, Rendimiento Energético, Deslizamiento, Potencia (Eléctrica), Par Motor.

Resumen—This report details the design, implementation, and development of a graphical user interface (GUI), created using MATLAB App Designer, for the simulation and calculation of induction motor parameters. The interface facilitates the input of crucial data such as iron and stray losses, stator and rotor resistances and reactances, source voltage, frequency, number of poles, and slip.

I. INTRODUCCIÓN

La interfaz busca implementar un método numérico mediante App Designer de MATLAB que ayude a identificar aspectos de funcionamiento y emule el comportamiento de un motor de inducción en estado estacionario. Para unas entradas y salidas ya definidas. El alcance del proyecto no busca ser un método más preciso al empleado en la literatura y diversos libros de máquinas eléctricas; al contrario, toma como apoyo dichas ecuaciones ya definidas.

II. RESUMEN

Las máquinas de inducción, particularmente los motores trifásicos, son fundamentales en la industria debido a su robustez, eficiencia y bajo mantenimiento. Para comprender su comportamiento y características operativas, es esencial realizar ensayos experimentales que permitan analizar su rendimiento en distintas condiciones. Entre las pruebas más comunes se encuentran la prueba de vacío y la prueba de rotor bloqueado, las cuales proporcionan información clave para la obtención de los parámetros del equivalente eléctrico del motor. Para la creación del código ya se parte de un equivalente de dichas pruebas realizadas y del modelo del circuito se tienen las entradas y a partir de estas entradas hallar las salidas se tomó un modelo de motor ya definido en la literatura —con su significado físico—. Se concatenaron las ecuaciones del modelo en función de las entradas y salidas definidas.

III. PRINCIPIO DE FUNCIONAMIENTO ENTORNO MATLAB Y ASPECTOS RELACIONADOS CON LAS ECUACIONES

III-A. modelo del circuito empleado

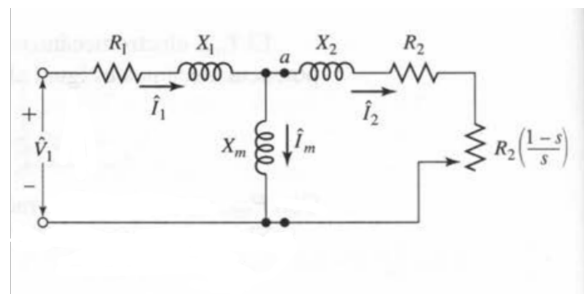


Figura 1: Modelo del circuito equivalente del motor de inducción.

III-B. Aspectos iniciales con el desarrollo de la interfaz

Si bien para el entorno matlab App Designer es un poco tedioso al principio de entender se parte de identificar las entradas y salidas.

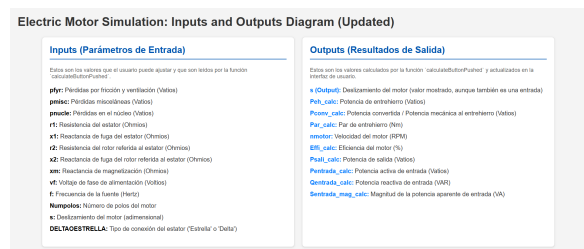
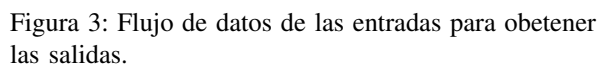


Figura 2: Entradas y Salidas.

En la figura ?? se menciona calculateButtonPushed, dicho botón que iniciará el funcionamiento de la interfaz.

De la figura ?? se puede observar que el flujo de datos de manera muy simplificada es el siguiente:

- Se ingresan los valores de las entradas.
- Se presiona el botón de calcular.
- Se ejecuta la función que calcula los valores de salida.



- ### III-C. Aspectos relacionados con la interfaz

Finalmente se puede observar el diseño de la interfaz con las entradas y salidas con sus respectivas unidades.



Se creo una función global que lee cada una de las entradas y a partir de las ecuaciones haya los parametros de la salida:

```
app.MessageBox.Text = '';  
app.MessageBox.Visible = 'off';  
% Asegurate de que los nombres de las  
propiedades (ej. .Value)
```

```
% y los nombres de los componentes (
ej. .pfyrEditField) sean correctos.
pfyr = app.pfyrEditField.Value;
pmisc = app.pmiscEditField.Value;
pnucle = app.pnucleEditField.Value;
r1 = app.r1EditField.Value;
x1 = app.x1EditField.Value;
r2 = app.r2EditField.Value;
x2 = app.x2EditField.Value;
xm = app.XmEditField.Value;
vf = app.vfEditField.Value;
f = app.fEditField.Value;
Numpolos = app.NumpolosEditField.
```

```
Value;
    s = app.sEditField.Value; %
Deslizamiento (input)
    % Determinar el tipo de conexi n (
Estrella o Delta)
```

```

    % Asume que DELTAOESTRELLADropDown es
    un DropDown con 'Estrella' y 'Delta'
    if strcmp(app.DELTAOESTRELLAEditField
.Value, 'Estrella')

```

```
DELTAOESTRELLA = 0; % Estrella
else % 'Delta'
    DELTAOESTRELLA = 1; % Delta
```

```
end
% 3. Validacin de entradas
% Se valida que los valores sean
positivos y que Numpolos y s no sean cero
.
```

```

    if any([pfyr, pmisc, pnucle, r1, x1,
r2, x2, xm, vf, f, Numpolos] < 0) ||
Numpolos == 0

```

```
    % Si alguna entrada es negativa o
    Numpolos es cero.
```

```
        UIAlert(app.UIFigure, 'Por favor, asegurese de que todas las entradas sean valores numricos vlidos y positivos. El nmero de polos no puede ser cero.', 'Error de Entrada', 'Icon', 'error');
```

```

        % Limpiar campos de salida en
    caso de error
    clearOutputFields(app);
    return;
end

```

```
end
% Validación específica para el
deslizamiento 's'
```

```

        if s == 0
            uialert(app.UIFigure, '
Advertencia: El deslizamiento (s) es cero

```

. Esto implica que el motor est
funcionando a velocidad sncrona.

Algunas ecuaciones de potencia y par pueden resultar en cero.', 'Advertencia de Deslizamiento', 'Icon', 'warning');

```

% Si s es 0, r2/s ser a infinito
. En un motor real, s nunca es
exactamente 0 con carga.

```

```

% Sin embargo, si el usuario
explicitamente pone s=0, se asume que no
hay carga.

```

```
% MATLAB puede manejar inf, pero
los resultados físicos podran no ser
los esperados.
```

```

% Para evitar NaN en los
cálculos, si s es muy pequeño, podemos
forzar un mínimo.

```

```

% Pero si el usuario pone 0, se
deja que MATLAB maneje el inf/nan si
ocurre.

```

```

45         % Aqu , se permite s=0 y se
calcula, pero se advierte.
46         elseif s < 0
47             uialert(app.UIFigure, '
Advertencia: El deslizamiento (s) es
negativo. Esto indica operaci n de
frenado o generaci n. Los resultados
pueden no ser t picos de operaci n
motora.', 'Advertencia de Deslizamiento',
'Icon', 'warning');
48         end
49         % 4. C lculos principales del motor
50         % Velocidad s ncrona (RPM)
51         nsinc = 120 * f / Numpolos;
52         % Velocidad del motor (RPM)
53         % Si s=1, nmotor ser 0, como se
especifica en el requisito.
54         nmotor = nsinc * (1 - s);
55         % Voltaje de fase (vf_calc) seg n la
conexi n
56         if DELTAOESTRELLA == 0 % Cuando es
cero es estrella
57             vf_calc = vf / sqrt(3);
58         else % Conexi n Delta
59             vf_calc = vf;
60         end
61         % Impedancias complejas
62         z1_calc = r1 + lj * x1;
63         % z2_calc: Si s es 0, r2/s ser Inf,
MATLAB lo maneja.
64         z2_calc = (r2 / s) + lj * x2;
65         xm_complex = lj * xm;
66         % Calcular V1, I1, I2 numricamente
67         % Ecuaci n del nodo V1: (V1 - vf) /
z1 + V1 / (j*xm) + V1 / z2 = 0
68         % V1 * (1/z1 + 1/(j*xm) + 1/z2) = vf
/ z1
69         % V1 = (vf / z1) / (1/z1 + 1/(j*xm) +
1/z2)
70         % Admitancia equivalente vista desde
V1
71         Yeq = (1 / z1_calc) + (1 / xm_complex
) + (1 / z2_calc);
72         % Voltaje V1 en el entrehierro
73         V1_calc = (vf_calc / z1_calc) / Yeq;
74         % Corrientes
75         i1_calc = (V1_calc - vf_calc) / z1_calc;
% Corriente del estator
76         i2_calc = V1_calc / z2_calc; %
Corriente del rotor referida al estator
77         % Calcular Potencia de Entrada (
Pentrada) y Potencia de Entrehierro (Peh)
78         % Potencia compleja de entrada (
fuente de voltaje)
79         % La frmula original del usuario
tiene un signo negativo: Sentrada_calc =
-3 * vf_calc * conj(i1_calc);
80         Sentrada_calc = -3 * vf_calc * conj(
i1_calc);
81         Pentrada_calc = real(Sentrada_calc);
% Potencia activa de entrada
82         Qentrada_calc = imag(Sentrada_calc);
% Potencia reactiva de entrada
83         Sentrada_mag_calc = abs(Sentrada_calc
); % Magnitud de la potencia
aparente de entrada
84         % Peh se puede calcular de dos formas
: Pentrada - Pcul - Pnucle O 3*abs(i2)^2*
r2/s
85         % Usaremos la segunda forma ya que i2
y s son conocidos
86         Peh_calc = 3*abs(i2_calc)^2 * r2 / s;
87         % Calcular Potencia Convertida (Pconv
)
88         Pconv_calc = 3*abs(i2_calc)^2*r2*(1-s
)/s;
89         % Esto es equivalente a: Pconv_calc =
Peh_calc * (1 - s);
90         % Calcular Par de Entrehierro (Par)
91         Par_calc = Peh_calc / (nsinc * 2 * pi
/ 60);
92         % Calcular Potencia de Salida (Psali)
y Eficiencia (Effi)
93         % Psali = Pconv - Prdidas por
fricci n y ventilaci n - Prdidas
miscelneas - Prdidas en el ncleo
94         Psali_calc = Pconv_calc - pfyr -
pmisc - pnucle;
95         % Eficiencia
96         Effi_calc = (Psali_calc /
Pentrada_calc) * 100;
97         if Pentrada_calc <= 0 || isnan(
Effi_calc) || isinf(Effi_calc)
98             Effi_calc = 0; % Evitar divisi n
por cero o resultados no numricos
99             if Pentrada_calc <= 0
100                 app.MessageBox.Text = '
Advertencia: La potencia activa de
entrada es cero o negativa. La eficiencia
no se puede calcular o no es aplicable.'
;
101                 app.MessageBox.FontColor =
[0.85 0.33 0.1]; % Naranja
102                 app.MessageBox.Visible = 'on'
;
103             end
104         end
105         % 5. Actualizar campos de salida de
la UI
106         app.sOutputEditField.Value = s; %
Mostrar el deslizamiento de entrada
107         app.pehEditField.Value = Peh_calc;
108         app.pconvEditField.Value = Pconv_calc
;
109         app.parEditField.Value = Par_calc;
110         app.nmotorEditField.Value = nmotor;
111         app.ffiEditField.Value = Effi_calc;
112         app.psalEditField.Value = Psali_calc;
113         app.PentradaEditField.Value =
Pentrada_calc;
114         app.QentradaEditField.Value =
Qentrada_calc;
115         app.SentradaEditField.Value =
Sentrada_mag_calc;
116         % Mostrar mensaje de xito
117         app.MessageBox.Text = 'C lculos
completados exitosamente.';
118         app.MessageBox.FontColor = [0 0.5 0];
% Verde
119         app.MessageBox.Visible = 'on';
120         % --- NUEVOS C LCULOS PARA GRAFICAR
LA CURVA COMPLETA ---
121         sgrafica = -1:0.001:1; % Rango de
deslizamiento, por ejemplo
122         nmotorgraf = nsinc .* (1 - sgrafica);
123         % ... (el resto de tus c lculos
vectoriales para z2_graf, V1_calcgraf,
i2_calc_graf, Peh_graf, Par_calcgraf) ...
124         z2_graf = (r2 ./ sgrafica) + lj * x2;
125         Yeqgraf = (1 ./ z1_calc) + (1 ./
xm_complex) + (1 ./ z2_graf);
126         V1_calcgraf = (vf_calc ./ z1_calc) ./
Yeqgraf;
127         i2_calc_graf = V1_calcgraf ./ z2_graf
;
128         Peh_graf = 3 * abs(i2_calc_graf).^2 *
r2 ./ sgrafica;
129         Par_calcgraf = Peh_graf ./ (nsinc .*
2 * pi / 60);
130         % --- C DIGO DE GRAFICACI N DIRECTO
---
131         cla(app.updateMotorPlots); % Limpia
la gr fica anterior

```

```

132     plot(app.updateMotorPlots, nmotorgraf, Par_calcgraf, '-', 'LineWidth', 0.8);
133     xlabel(app.updateMotorPlots, 'Velocidad del Motor (RPM)');
134     ylabel(app.updateMotorPlots, 'Par del Motor (Nm)');
135     title(app.updateMotorPlots, 'Curva Característica: Par vs. Velocidad');
136     grid(app.updateMotorPlots, 'on');
137     hold(app.updateMotorPlots, 'on');
138     % --- Segunda gráfica
139     plot(app.updateMotorPlots, nmotor, Par_calc, 'r-o', 'DisplayName', 'Potencia de Salida', 'LineWidth', 1.5); % Línea roja con guiones y 'x'
140     % 3. Restaurar el comportamiento predeterminado de los ejes
141     hold(app.updateMotorPlots, 'off');
142     cla(app.desli); % Limpia la gráfica anterior
143     plot(app.desli, sgrafica, Par_calcgraf, '-', 'LineWidth', 0.8);
144     xlabel(app.desli, 'Deslizamiento');
145     ylabel(app.desli, 'Par del Motor (Nm)');
146     title(app.desli, 'Curva Característica: Par vs. Deslizamiento');
147     grid(app.desli, 'on');
148     hold(app.desli, 'on');
149     plot(app.desli, s, Par_calc, 'r-o', 'DisplayName', 'Deslizamiento', 'LineWidth', 1.5); % Línea roja con guiones y 'x'
150     hold(app.desli, 'off');
151     % --- Aquí es donde agregas la llamada a la función de graficación ---
152     % Los parámetros s_values, Psali_results, s_single, Par_at_s_single, % Psali_at_s_single, nmotor_at_s_single se pasan para que la función
153     % updateMotorPlots tenga todos los datos que espera, incluso si para
154     % esta gráfica específica solo usamos nmotor y Par_calc.
155     % Si updateMotorPlots espera vectores, puedes convertir los valores
156     % escalares a vectores de un solo elemento:
157     catch ME
158     % 6. Manejo de errores
159     % Mostrar mensaje de error al usuario
160     uialert(app.UIFigure, ['Error en el cálculo: ' ME.message], 'Error', 'Icon', 'error');
161     app.MessageBox.Text = ['Error: ' ME.message];
162     app.MessageBox.FontColor = [1 0 0]; % Rojo
163     app.MessageBox.Visible = 'on';
164     % Limpiar campos de salida en caso de error
165     clearOutputFields(app);
166 end
167 end
168 % --- Función auxiliar para limpiar campos de salida ---
169 function clearOutputFields(app)
170     app.sOutputEditField.Value = 0;
171     app.pegEditField.Value = 0;
172     app.pconvEditField.Value = 0;
173     app.parEditField.Value = 0;
174     app.nmotorEditField.Value = 0;
175     app.ffiEditField.Value = 0;
176     app.psalEditField.Value = 0;
177     app.PentradaEditField.Value = 0;
178     app.QentradaEditField.Value = 0;
179     app.SentradaEditField.Value = 0;
180 end
181 end
182 % --- Función de inicio de la aplicación (StartupFcn) ---
183 % (Opcional) Puedes usar esta función para establecer valores predeterminados
184 % al iniciar la aplicación.
185 function StartupFcn(app)
186     % Establecer valores predeterminados para los campos de entrada
187     app.pfyEditField.Value = 1500;
188     app.pmiscEditField.Value = 1500;
189     app.pnucleEditField.Value = 450;
190     app.r1EditField.Value = 0.012;
191     app.x1EditField.Value = 0.41;
192     app.r2EditField.Value = 0.25;
193     app.x2EditField.Value = 0.32;
194     app.XmEditField.Value = 4.6;
195     app.vfEditField.Value = 440;
196     app.fEditField.Value = 60;
197     app.NumpolosEditField.Value = 4;
198     app.sEditField.Value = (1800-1740)/1800;
199     % Deslizamiento inicial
200     % Establecer la opción predeterminada para el DropDown
201     app.DELTAOESTRELLAEditField.Items = {'Estrella', 'Delta'};
202     app.DELTAOESTRELLAEditField.Value = 'Delta';
203     % Inicializar campos de salida como vacíos o NaN
204     clearOutputFields(app);
205     % Ocultar el cuadro de mensajes al inicio
206     app.MessageBox.Visible = 'off';
207 end

```

Listing 1: MATLAB Code

V. PRUEBAS CON LA INTERFAZ

la interfaz fue probada con el siguiente ejemplo:

Caso 1

En su edificio, el motor de inducción de la bomba contraincendios se averió. El administrador del edificio sabe que usted es estudiante de ingeniería eléctrica y le expresa que tiene dos cotizaciones de motores de inducción para cambiar dicho motor averiado. Los parámetros de los motores se presentan en las Tablas 1 y 2.

De la operación del motor de la bomba a reemplazar usted conoce la información de la Tabla 1.

Tabla 1. Información de operación del motor de la bomba a reemplazar

Parámetros	Valor
P _{nom}	50 HP
T _{arranq}	>1kN.m
P _{sal_max}	>100 kW
Eficiencia (nominal)	>90%

Estas condiciones corresponden a los requerimientos mínimos que deben cumplir el motor elegido.

Utilizando los conocimientos adquiridos en su curso de máquinas I, mencione ¿cuál de las dos cotizaciones usted recomendaría al administrador tomar? Justifique su respuesta basada en cálculos.

Cotización 1	Cotización 2
Rotor jaula de ardilla	Rotor jaula de ardilla
Polos = 4	Polos = 4
Conexión = Delta	Conexión = Y
V _{nom} = 440V	V _{nom} = 440V
f = 60 Hz	f = 60 Hz
n _{nom} = 1740 rpm	n _{nom} = 1700 rpm
n _{vacio} = 1796 rpm	n _{vacio} = 1796 rpm
R1 = 0.012 Ω	R1 = 0.08 Ω
X1 = 0.41 Ω	X1 = 0.12 Ω
R2 = 0.25 Ω	R2 = 0.017 Ω
X2 = 0.32 Ω	X2 = 0.2 Ω
XM = 4.60 Ω	XM = 5.03 Ω
P _{mec} = 1500W	P _{mec} = 1200W
P _{gyr} = 1500W	P _{gyr} = 1200W
P _{misc} = 450W	P _{misc} = 450W

Respuesta:

Cotización	Vin (V)	Tarranq (N.m)	Pnom (W)	Eficiencia (%)	Smax (pu)	Pmax(W)
1	440.00	1174.87	58950.77	90.77	0.3589	218645.20
2	254.03	164.53	244182.94	93.94	0.0536	266105.10

Figura 5: Ejemplo para poner a prueba la interfaz.

En donde los resultados de la interfaz fueron los siguientes:

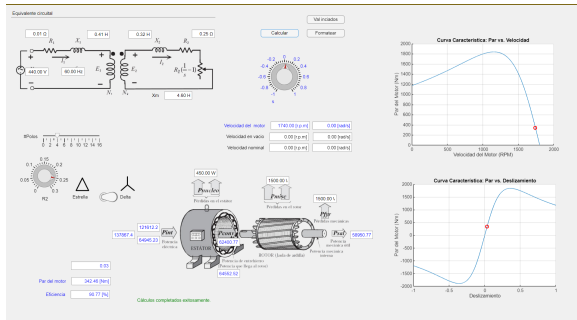


Figura 6: Resultados de la interfaz al ejemplo.

VI. CONCLUSIONES

La interfaz desarrollada mostro un resultado conciso y claro. Con relación a los resultados fueron los esperados, aunque como adicional se podria hacer que en las graficas se mostrara la velocidad cuando el motor opera como generador para la grafica Par del motor velocidad del motor no lo logre sin embargo para la grafica deslizamiento del motor y par del motor si.

REFERENCIAS

- [1] A. E. Fitzgerald, C. Kingsley, y A. Kusko, *Electric Machinery*, 7^a ed. Nueva York, EE. UU.: McGraw-Hill Education, 2020.
- [2] J. Fraile Mora, *Máquinas Eléctricas*, 6^a ed. Madrid, España: McGraw-Hill, 2008.