DISEÑO IMPLEMENTACIOÓN Y DESARROLLO DE UNA INTEFAZ 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 codigo ya se parte de un equivalente de dichas pruebas realizadas y del modelo del circiuto 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

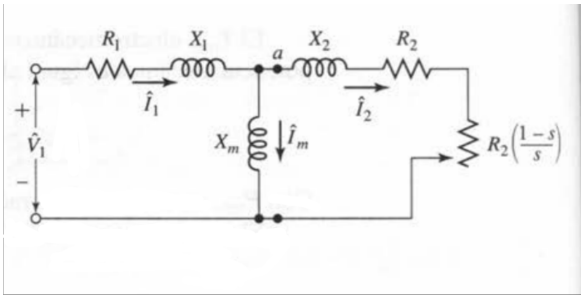


Figura 1: Modelo del circuito equivalente del motor de inducción.

III-B. Aspectos inciales 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.

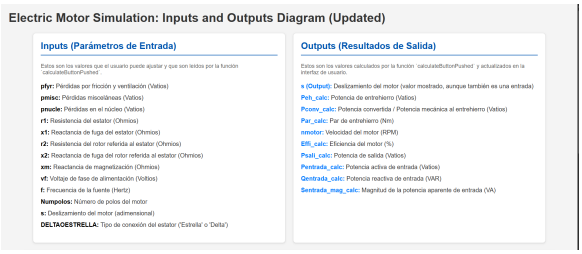


Figura 2: Entradas y Salidas.

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.

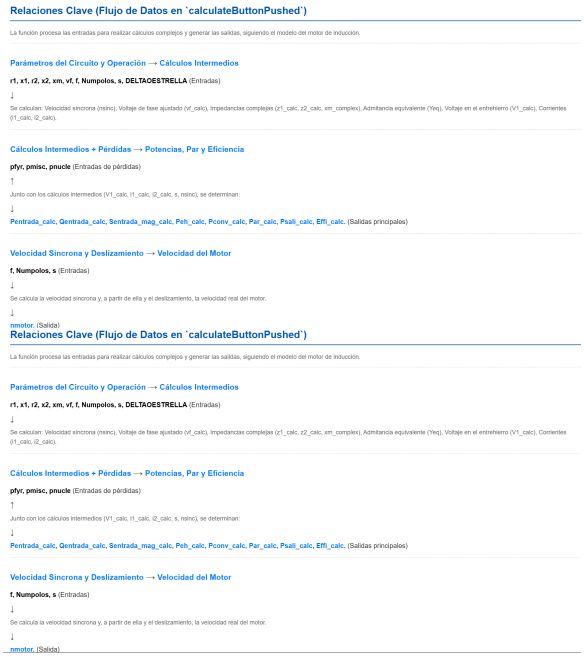


Figura 3: Flujo de datos de las entradas para obetener las salidas.

- Se muestran los resultados en la interfaz.
- Se grafican los resultados.

III-C. Aspectos relacionados con la intefaz

En la construccion de la interfaz primero se partio del diseño de la misma, teniendo en cuenta los parametros ademas de elementos como 2 sliders y editores de texto numericos tanto para las variables de entrada como de salida.

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

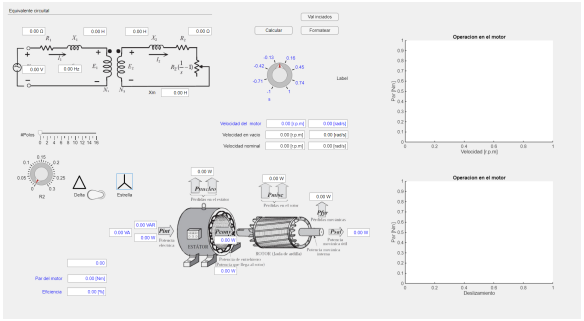


Figura 4: Interfaz de usuario.

IV. Diseno y aspectos relacionados con la programacion

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

```
function calculateButtonPushed(app, event)
    try
        % 1. Limpiar mensajes anteriores
        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. Validaci n 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, aseg rese de que todas las entradas sean valores num ricos v lidos y positivos. El n mero de polos no puede ser cero.', 'Error de Entrada', 'Icon', 'error');
            % Limpiar campos de salida en caso de error
            clearOutputFields(app);
            return;
        end
        % Validaci n espec fica 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 s ncrona. 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 expl citamente pone s=0, se asume que no hay carga.
            % MATLAB puede manejar inf, pero los resultados f sicos podr an no ser los esperados.
            % Para evitar NaN en los c lculos, si s es muy peque o, podemos forzar un m nimo.
            % Pero si el usuario pone 0, se deja que MATLAB maneje el inf/nan si ocurre.
```

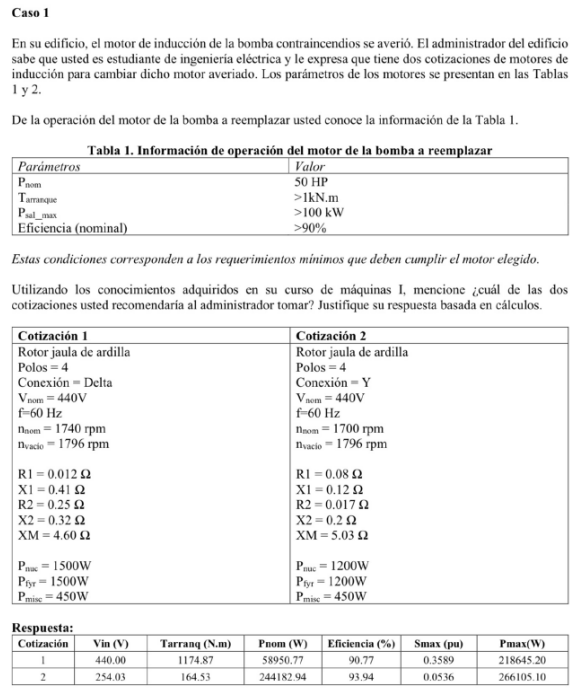
```
        % Aqu , se permite s=0 y se calcula, pero se advierte.
    elseif s < 0
        uialert(app.UIFigure, 'Advertencia: El deslizamiento (s) es negativo. Esto indica operaci n de frenado o generaci n. Los resultados pueden no ser t picos de operaci n motora.', 'Advertencia de Deslizamiento', 'Icon', 'warning');
    end
    % 4. C lculos principales del motor
    % Velocidad s ncrona (RPM)
    nsinc = 120 * f / Numpolos;
    % Velocidad del motor (RPM)
    % Si s=1, nmotor ser  0, como se especifica en el requisito.
    nmotor = nsinc * (1 - s);
    % Voltaje de fase (vf_calc) seg n la conexi n
    if DELTAOESTRELLA == 0 % Cuando es cero es estrella
        vf_calc = vf / sqrt(3);
    else % Conexi n Delta
        vf_calc = vf;
    end
    % Impedancias complejas
    z1_calc = r1 + 1j * x1;
    % z2_calc: Si s es 0, r2/s ser  Inf, MATLAB lo maneja.
    z2_calc = (r2 / s) + 1j * x2;
    xm_complex = 1j * xm;
    % Calcular V1, I1, I2 num ricamente
    % Ecuaci n del nodo V1: (V1 - vf) / z1 + V1 / (j*xm) + V1 / z2 = 0
    % V1 * (1/z1 + 1/(j*xm) + 1/z2) = vf / z1
    % V1 = (vf / z1) / (1/z1 + 1/(j*xm) + 1/z2)
    % Admitancia equivalente vista desde V1
    Yeq = (1 / z1_calc) + (1 / xm_complex) + (1 / z2_calc);
    % Voltaje V1 en el entrehierro
    V1_calc = (vf_calc / z1_calc) / Yeq;
    % Corrientes
    i1_calc = (V1_calc-vf_calc)/z1_calc; % Corriente del estator
    i2_calc = V1_calc/z2_calc; % Corriente del rotor referida al estator
    % Calcular Potencia de Entrada (Pentrada) y Potencia de Entrehierro (Peh)
    % Potencia compleja de entrada (fuente de voltaje)
    % La f rmula original del usuario tiene un signo negativo: Sentrada_calc = -3 * vf_calc * conj(i1_calc);
    Sentrada_calc = -3 * vf_calc * conj(i1_calc);
    Pentrada_calc = real(Sentrada_calc);
        % Potencia activa de entrada
    Qentrada_calc = imag(Sentrada_calc);
        % Potencia reactiva de entrada
    Sentrada_mag_calc = abs(Sentrada_calc);      % Magnitud de la potencia aparente de entrada
    % Peh se puede calcular de dos formas: Pentrada - Pcu1 - Pnucle O 3*abs(i2)^2*r2/s
    % Usaremos la segunda forma ya que i2 y s son conocidos
    Peh_calc = 3*abs(i2_calc)^2 * r2 / s;
    % Calcular Potencia Convertida (Pconv)
    Pconv_calc = 3*abs(i2_calc)^2*r2*(1-s)/s;
    % Esto es equivalente a: Pconv_calc = Peh_calc * (1 - s);
    % Calcular Par de Entrehierro (Par)
    Par_calc = Peh_calc / (nsinc * 2 * pi / 60);
    % Calcular Potencia de Salida (Psali) y Eficiencia (Effi)
    % Psali = Pconv - P rdidas por fricci n y ventilaci n - P rdidas miscel neas - P rdidas en el n cleo
    Psali_calc = Pconv_calc - pfyr - pmisc - pnucle;
    % Eficiencia
    Effi_calc = (Psali_calc / Pentrada_calc) * 100;
    if Pentrada_calc <= 0 || isnan(Effi_calc) || isinf(Effi_calc)
        Effi_calc = 0; % Evitar divisi n por cero o resultados no num ricos
        if Pentrada_calc <= 0
            app.MessageBox.Text = 'Advertencia: La potencia activa de entrada es cero o negativa. La eficiencia no se puede calcular o no es aplicable.';
            app.MessageBox.FontColor = [0.85 0.33 0.1]; % Naranja
            app.MessageBox.Visible = 'on';
        end
    end
    % 5. Actualizar campos de salida de la UI
    app.sOutputEditField.Value = s; % Mostrar el deslizamiento de entrada
    app.pehEditField.Value = Peh_calc;
    app.pconvEditField.Value = Pconv_calc;
    app.parEditField.Value = Par_calc;
    app.nmotorEditField.Value = nmotor;
    app.effiEditField.Value = Effi_calc;
    app.psalEditField.Value = Psali_calc;
    app.PentradaEditField.Value = Pentrada_calc;
    app.QentradaEditField.Value = Qentrada_calc;
    app.SentradaEditField.Value = Sentrada_mag_calc;
    % Mostrar mensaje de  xito
    app.MessageBox.Text = 'C lculos completados exitosamente.';
    app.MessageBox.FontColor = [0 0.5 0]; % Verde
    app.MessageBox.Visible = 'on';
    % --- NUEVOS C LCULOS PARA GRAFICAR LA CURVA COMPLETA ---
    sgrafica = -1:0.001:1; % Rango de deslizamiento, por ejemplo
    nmotorgraf = nsinc .* (1 - sgrafica);
    % ... (el resto de tus c lculos vectoriales para z2_graf, V1_calcgraf, i2_calc_graf, Peh_graf, Par_calcgraf) ...
    z2_graf = (r2 ./ sgrafica) + 1j * x2;
    Yeqgraf = (1 ./ z1_calc) + (1 ./ xm_complex) + (1 ./ z2_graf);
    V1_calcgraf = (vf_calc ./ z1_calc)./ Yeqgraf;
    i2_calc_graf = V1_calcgraf ./ z2_graf;
    Peh_graf = 3 * abs(i2_calc_graf).^2 * r2 ./ sgrafica;
    Par_calcgraf = Peh_graf ./ (nsinc .* 2 * pi / 60);
    % --- C DIGO DE GRAFICACI N DIRECTO ---
    cla(app.updateMotorPlots); % Limpia la gr fica anterior
```

```
        plot(app.updateMotorPlots, nmotorgraf, Par_calcgraf, '-', 'LineWidth', 0.8);
        xlabel(app.updateMotorPlots, 'Velocidad del Motor (RPM)');
        ylabel(app.updateMotorPlots, 'Par del Motor (Nm)');
        title(app.updateMotorPlots, 'Curva Caracterstica: Par vs. Velocidad');
        grid(app.updateMotorPlots, 'on');
        hold(app.updateMotorPlots, 'on');
        % --- Segunda grfica
        plot(app.updateMotorPlots, nmotor, Par_calc, 'r-o', 'DisplayName', 'Potencia de Salida', 'LineWidth', 1.5); % Lnea roja con guiones y 'x'
        % 3. Restaurar el comportamiento predeterminado de los ejes
        hold(app.updateMotorPlots, 'off');
        cla(app.desli); % Limpia la grfica anterior
        plot(app.desli, sgrafica, Par_calcgraf, '-', 'LineWidth', 0.8);
        xlabel(app.desli, 'Deslizamiento');
        ylabel(app.desli, 'Par del Motor (Nm)');
        title(app.desli, 'Curva Caracterstica: Par vs. Deslizamiento');
        grid(app.desli, 'on');
        hold(app.desli, 'on');
        plot(app.desli, s, Par_calc, 'r o', 'DisplayName', 'Deslizamiento', 'LineWidth', 1.5); % Lnea roja con guiones y 'x'
        hold(app.desli, 'off');
        % --- Aqu es donde agregas la llamada a la funcin de graficacin ---
        % Los parmetros s_values, Psali_results, s_single, Par_at_s_single,
        % Psali_at_s_single, nmotor_at_s_single se pasan para que la funcin
        % updateMotorPlots tenga todos los datos que espera, incluso si para
        % esta grfica especfica solo usamos nmotor y Par_calc.
        % Si updateMotorPlots espera vectores, puedes convertir los valores
        % escalares a vectores de un solo elemento:
    catch ME
        % 6. Manejo de errores
        % Mostrar mensaje de error al usuario
        uialert(app.UIFigure, ['Error en el clculo: ' ME.message], 'Error', 'Icon', 'error');
        app.MessageBox.Text = ['Error: ' ME.message];
        app.MessageBox.FontColor = [1 0 0]; % Rojo
        app.MessageBox.Visible = 'on';
        % Limpiar campos de salida en caso de error
        clearOutputFields(app);
    end
end
% --- Funcin auxiliar para limpiar campos de salida ---
function clearOutputFields(app)
    app.sOutputEditField.Value = 0;
    app.pehEditField.Value = 0;
    app.pconvEditField.Value = 0;
    app.parEditField.Value = 0;
    app.nmotorEditField.Value = 0;
    app.effiEditField.Value = 0;
    app.psalEditField.Value = 0;
    app.PentradaEditField.Value = 0;
    app.QentradaEditField.Value = 0;
    app.SentradaEditField.Value = 0;
end
% --- Funcin de inicio de la aplicacin (StartupFcn) ---
% (Opcional) Puedes usar esta funcin para establecer valores predeterminados
% al iniciar la aplicacin.
function StartupFcn(app)
    % Establecer valores predeterminados para los campos de entrada
    app.pfyrEditField.Value = 1500;
    app.pmiscEditField.Value = 1500;
    app.pnucleEditField.Value = 450;
    app.r1EditField.Value = 0.012;
    app.x1EditField.Value = 0.41;
    app.r2EditField.Value = 0.25;
    app.x2EditField.Value = 0.32;
    app.XmEditField.Value = 4.6;
    app.vfEditField.Value = 440;
    app.fEditField.Value = 60;
    app.NumpolosEditField.Value = 4;
    app.sEditField.Value = (1800-1740)/1800;
    % Deslizamiento inicial
    % Establecer la opcin predeterminada para el DropDown
    app.DELTAOESTRELLAEditField.Items = {'Estrella', 'Delta'};
    app.DELTAOESTRELLAEditField.Value = 'Delta';
    % Inicializar campos de salida como vacos o NaN
    clearOutputFields(app);
    % Ocultar el cuadro de mensajes al inicio
    app.MessageBox.Visible = 'off';
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
$T_{arranque}$	>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_{nuc} = 1500W	P_{nuc} = 1200W
P_{fyr} = 1500W	P_{fyr} = 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:

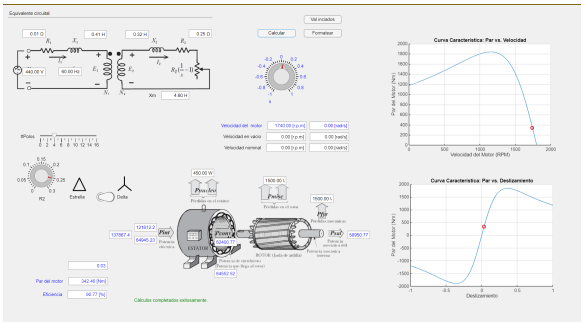


Figura 6: Resultados de la interfaz al ejemplo.

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*, 7ª ed. Nueva York, EE. UU.: McGraw-Hill Education, 2020.

[2] J. Fraile Mora, *Máquinas Eléctricas*, 6ª ed. Madrid, España: McGraw-Hill, 2008.