SIMULATION OF STEADY-STATE BEHAVIOR OF SINUSOIDALLY EXCITED RLC SERIES CIRCUIT

A MATLAB APPROACH

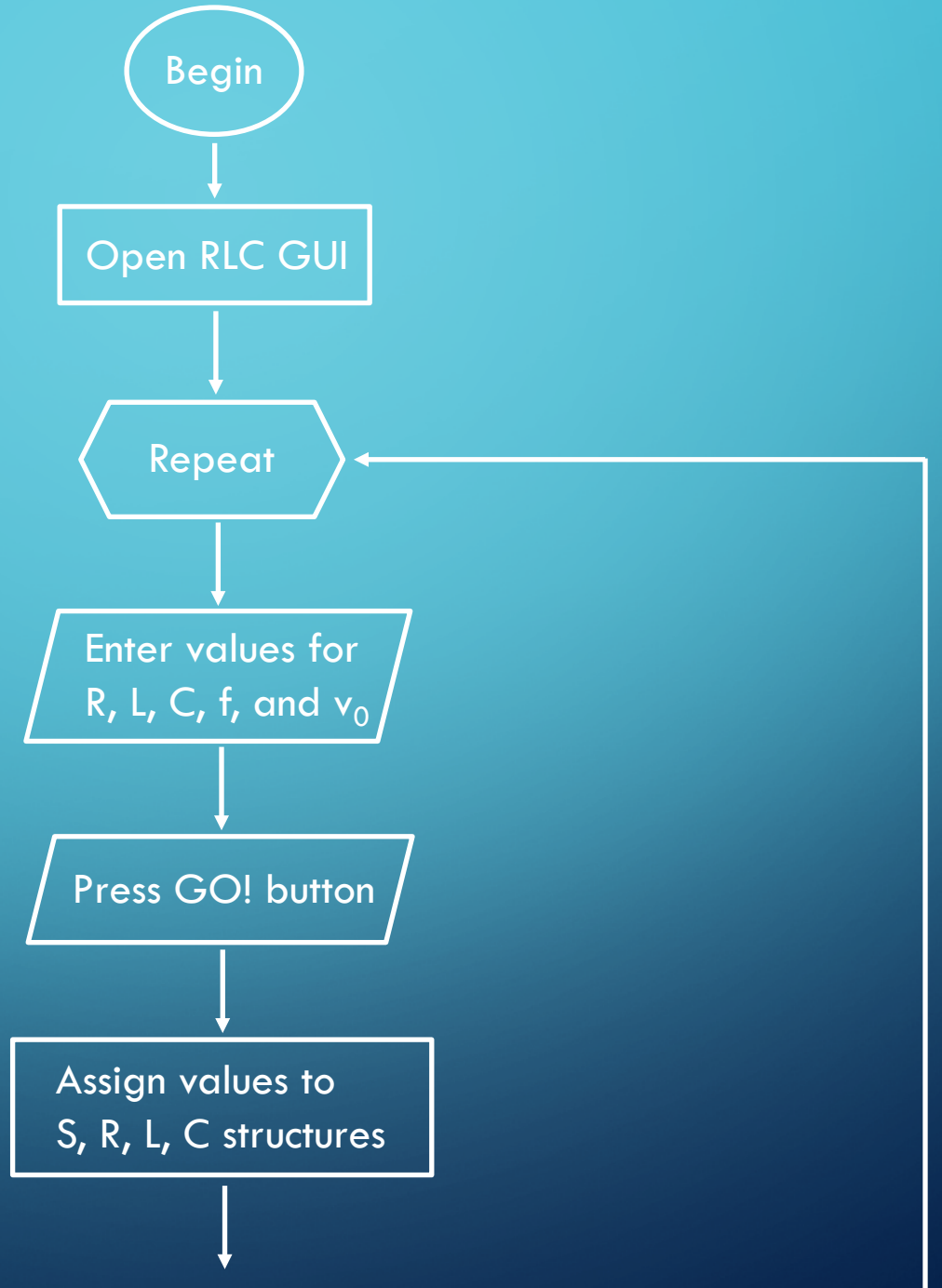
BY: OHMEGA EAGLES

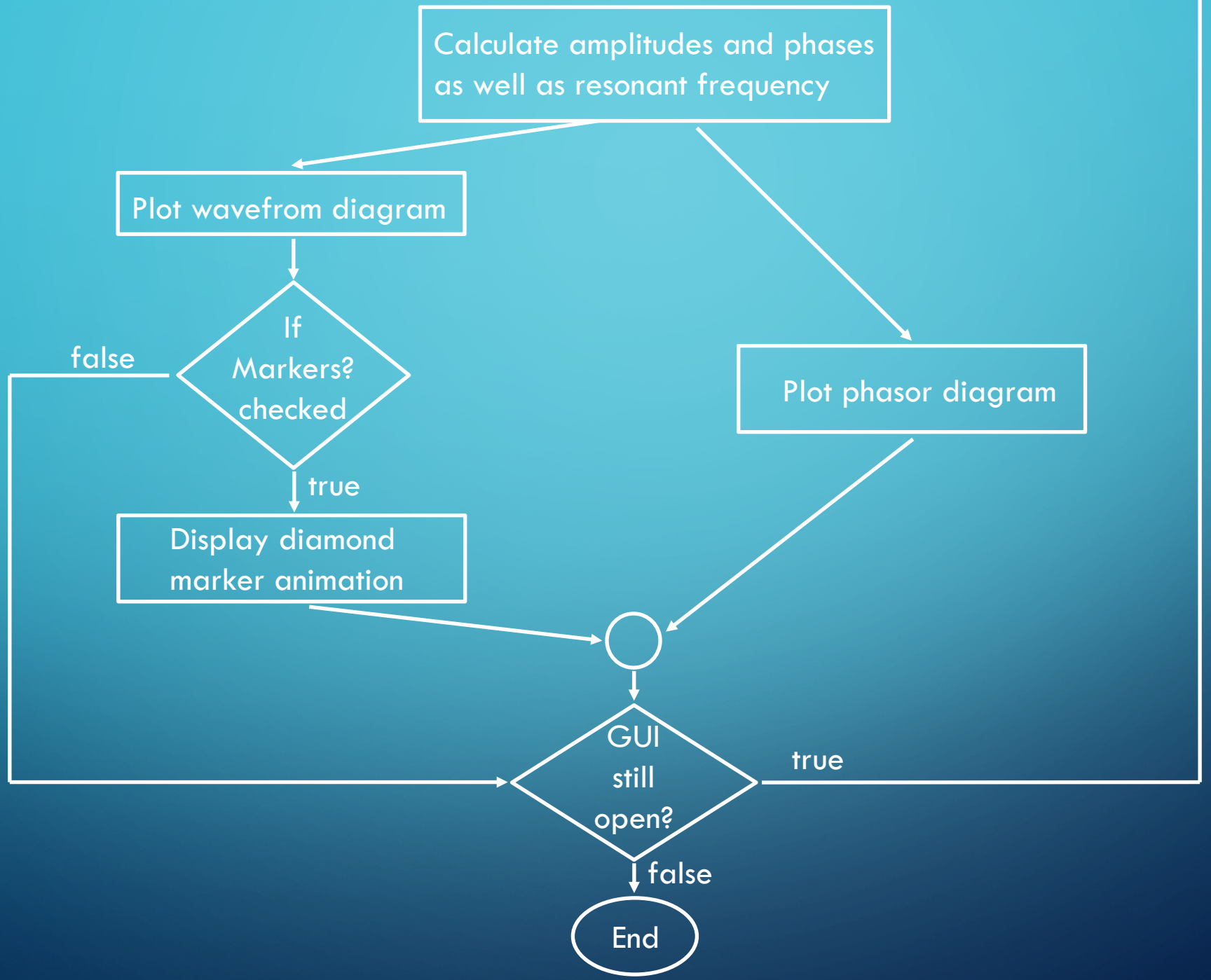
IMAN AYAZ, THOMAS TEMPLIN, DANIEL ILLOH

PROBLEM STATEMENT

- The concept of RLC circuits is learned in Physics courses at various levels. An RLC circuit is an electrical circuit containing a Resistor(R), an inductor(L) and a capacitor(C) connected in series or in parallel. They are used in radios, televisions, oscillation circuits etc. The goal of group is to create an application that provides the output graph and phasor diagram for an RLC circuit with a given input of frequency and amplitude. We are making it as an educational tool and as a visualization to demonstrate how RLC circuits work. It will help in understanding the mathematical equations associated with RLC circuits. The phasor diagram and the graph show the expected relationship between the amplitude and the phase of the resistor, capacitor and inductor.

APPROACH


Begin
Open RLC GUI
Repeat
Enter values for
R, L, C, f, and v_0
Press GO! button
Assign values to
S, R, L, C structures


Calculate amplitudes and phases as well as resonant frequency
Plot wavefrom diagram
If Markers? checked
false
true
Display diamond marker animation
Plot phasor diagram
GUI still open?
true
false
End

MATLAB CODE

- For loop (marker animation)
- String manipulation [str2double(…), strcat(…)]
- Plots (GUI, waveform and phasor diagrams)
- Functions (10 regular + 18 GUI functions)
- Structures (one each for S, R, L, and C)

```
function amplitude = calculate_amplitude(w_in, v0, R, L, C, type)
% Calculates and returns magnitude of waveform of circuit element
%{
    Formal parameters:
    w_in: angular frequency of voltage source (rad/s)
    v0: amplitude of voltage source (V)
    R: series resistance (ohm)
    L: series inductance (H)
    C: series capacitance (F)
    type: a passive circuit element ('R', 'L', or 'C')
%}
    den = sqrt((1-w_in^2*L*C)^2 + (w_in*R*C)^2);    % denominator of expression
    switch type
        case 'R'    % resistor
            amplitude = v0*w_in*R*C/den;
        case 'L'    % inductor
            amplitude = v0*w_in^2*L*C/den;
        case 'C'    % capacitor
            amplitude = v0/den;
    end
end
```

```
% Toggle for animation
if anim_flag

    % Assignment of plot handles for markers
    S.p = assign_plots(t, S.amp, S.amp, 0, 'k');
    R.p = assign_plots(t, S.amp, R.amp, R.phi, 'r');
    L.p = assign_plots(t, S.amp, L.amp, L.phi, 'g');
    C.p = assign_plots(t, S.amp, C.amp, C.phi, 'b');

    % Add moving markers
    len_t = length(t);
    for k = 2:4:len_t
        animate_waveform(S.p, t, k, S.omg, S.amp, 0)        % voltage source
        animate_waveform(R.p, t, k, S.omg, R.amp, R.phi)    % resistor
        animate_waveform(L.p, t, k, S.omg, L.amp, L.phi)    % inductor
        animate_waveform(C.p, t, k, S.omg, C.amp, C.phi)    % capacitor
    end
```

SOME FUNCTIONS USED:

- Imread (for matlab to read the code of external images and load them into the folder)
- Imshow (to display the image)
- axes (to place the image on certain axes)

```
guidata(hObject, handles);
theImage=imread('Copy_of_circuit.jpg');      % load circuit image
axes(handles.Inputs);                        % assin axes handles
imshow(theImage);                            % display image
```

Editor - /Users/ekenedilichukwuilloh/Documents/MATLAB/phasorDiagram.m

phasorDiagram.m

```
function diagram = phasorDiagram(w_in, v0, R, L, C)

amplitudeR = calculate_amplitude(w_in, v0, R, L, C,'R');
amplitudeL = calculate_amplitude(w_in, v0, R, L, C,'L');
amplitudeC = calculate_amplitude(w_in, v0, R, L, C,'C');

phaseR = calculate_phase(w_in, R, L, C,'R');
phaseL = calculate_phase(w_in, R, L, C,'L');
phaseC = calculate_phase(w_in, R, L, C,'C');


x_start = 0;% Starting position
y_start = 0;

%PLOT THE AXES
angle = (0)*pi/180; % Bearing angle
[x,y] = pol2cart(angle,350);
% Draw line between start and target
davinci('arrow','X',[x_start x],'Y',[y_start y],'Color'...
    ,[0.5 0.5 0.5],'LineWidth',1,'Head.Length',10,'Head.Width',20);

hold on

angle = (90)*pi/180; % Bearing angle
[x,y] = pol2cart(angle,350);
% Draw line between start and target
davinci('arrow','X',[x_start x],'Y',[y_start y],'Color'...
    ,[0.5 0.5 0.5],'LineWidth',1,'Head.Length',10,'Head.Width',20);
```

Editor - /Users/ekenedilichukwuilloh/Documents/MATLAB/phasorDiagram.m

phasorDiagram.m

```
    [x,y] = pol2cart(angle,v0);
    % Draw line between start and target
    davinci('arrow','X',[x_start x],'Y',[y_start y],'Color'...
        ,'k','LineWidth',2,'Head.Length',30,'Head.Width',30);



    %code for Vr
    [x,y] = pol2cart(phaseR,amplitudeR);
    % Draw line between start and target
    davinci('arrow','X',[x_start x],'Y',[y_start y],'Color'...
        ,'red','LineWidth',2,'Head.Length',30,'Head.Width',30);

    %code for Vl
    [x,y] = pol2cart(phaseL,amplitudeL);
    % Draw line between start and target
    davinci('arrow','X',[x_start x],'Y',[y_start y],'Color'...
        ,'green','LineWidth',2,'Head.Length',30,'Head.Width',30);

    %code for Vc
    [x,y] = pol2cart(phaseC,amplitudeC);
    % Draw line between start and target
    davinci('arrow','X',[x_start x],'Y',[y_start y],'Color'...
        ,'blue','LineWidth',2,'Head.Length',30,'Head.Width',30);


    axis equal
    axis([-v0*1.5 v0*1.5 -v0*1.5 v0*1.5])

    end

```

FUNCTIONS USED FOR PHASOR DIAGRAM

- Pol2cart – Transform polar or cylindrical coordinates to Cartesian for plotting.
- Davinci – uses low level MATLAB graphics commands to draw midlevel shapes. Can be used instead of plot to add more graphics.

CHALLENGES

- Showing the RLC circuit in the GUI
- Integrating both circuit codes into the GUI code