P.PRITHAM
RA2411004010493

SRM Institute of Science and Technology

College of Engineering and Technology

Department of Electronics and Communication Engineering

Academic Year:2025-2026 (ODD)

Test: LLT-1 (Case Study & Simulation) **Date:27.10.2025**
Course Code & Title: 21ECC205T-Electromagnetic Theory and Interference
Year & Sem & Dept: II/III/ECE **Max. Marks:** 40

Course Articulation Matrix:

	21ECC205T- Electromagnetic Theory and Interference	Program Outcomes (PO)														
		Graduate Attributes												PSO		
CO	Course Outcomes (CO)	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	*Apply the concepts and knowledge to solve problems related to electric field.*	-	2	3	-	-	-	-	-	-	-	-	-	-	-	-
2	*Analyse the concepts of Magnetic field and Maxwell's equations in the real-world application.*	-	3	2	-	-	-	-	-	-	-	-	-	-	-	-
3	*Translate the phenomenon of guided wave propagation and its mode of propagation.*	-	3	2	-	-	-	-	-	-	-	-	-	-	-	-
4	*Describe the importance of transmission line theory applicable to low frequency transmission lines.*	-	2	3	-	-	-	-	-	-	-	-	-	-	-	-
5	*Solve transmission line parameter and impedance matching through analytical and graphical methods.*	-	2	3	-	-	-	-	-	-	-	-	-	-	-	-

40/40

(Answer ALL Questions) (4*10=40 Marks)

Q. No	Case study Questions	Marks	BL	CO	PO
1	An electric vehicle is facing electronic control system malfunctions caused by EMI from its high-power drive systems. What measures can be taken to enhance electromagnetic compatibility among the vehicle's electronic components?	10	4	4	3
2	A satellite communication system is encountering interference from solar radiation and other electromagnetic disturbances in space. What design strategies can be adopted to ensure reliable communication and reduce EMI?	10	4	4	3
	Simulation Questions				
3	A lossless 30 m long transmission line with Z_o=50Ω is established between two ground stations which operate at 2 MHz. The line is terminated with a load $Z_L = 60+j40\Omega$. If $u = 0.6c$ on the line. Write a **Scilab code** to plot the reflection coefficient (Γ), standing wave ratio (S) and input impedance in smith chart.	10	4	5	3
4	In a satellite base station, a load of 100 + j150 Ω is connected to a 75Ω lossless line. Write a **Scilab code** to plot the Reflection coefficient (Γ), SWR value and input impedance (Z_{in}) at 0.4λ from the load.	10	4	5	3

LLT-I [CASESTUDY & SIMULATION]
21ECC205T - EMT

nhance electromagnetic compatibility (EMC) in electric
les (EV's), engineers must implement a multi-layered
tegy that address both the sources and
ways of electromagnetic interference. This
ures reliable operation of electronic control
ems despite the presence of high-power drive
tems.

e disturbances can couple into sensitive electronic
rol units (ECUs), sensors and communication lines,
ding to malfunctions or degraded performance.
mitigate these effects and improve EMC, several
ign and system level measures are essential:

20/20

System shielding Techniques:
one of the most
ective methods is the use of electromagnetic
ielding which involves enclosing critical components
cables in conductive materials. This prevents
adiated EMI from reaching sensitive electronics.

Filtering components:
EMI filters, such as common-mode
hokes and ferrite beads are placed at power
put/output terminals to suppress conducted noise.
ese filters block high frequency interference while
llowing desired signals to pass. Designers often
e multistage filters in DC/DC converters and
nboard chargers to meet automotive EMC standards

PCB layout optimization:
printed circuit board
design plays a crucial role in minimizing EMI. Techniques
include short traces lengths, proper grounding and
segregation of high and low voltage domains.
round planes should be continous and connected to
chassis ground where appropriate. Differential signal
routing and impedance matching also help reduce

suspectibility to noise.

④ component selection and packaging:
using components with low parasitic inductance and capacitance and selecting IC's with build in EMI suppression features, can reduce emissions. Additionally, metal enclosures or EMI gaskets in module packaging help contain radiated noise.

By integrating these measures across the vehicle's architecture - from component level design to system-wide layout - engineers can significantly enhance electromagnetic compatibility. This ensures robust operation of control systems, safety features even in the presence of high power drive systems.

2) Satellite communication systems operate in a uniquely challenging electromagnetic environment. Solar flares, cosmic radiation and charged particles can introduce significant EMI, disrupting signal integrity, degrading onboard electronics and threatening mission success. To combat these threats, designers implement a combination of Physical, electrical and architectural safeguards:

① Radiation - Hardened components:
Satellites use radiation-hardened electronics that can withstand ionizing radiation without performance degradation. These components are build with specialized materia and manufacturing process to resist single-event upsets, latch ups and total ionizing dose effects

② Shielding and Enclosures:
Electromagnetic shielding is essential to protect sensitive subsystems. Conductive enclosures made of aluminium or composite materia

ith embedded metal layers block external EMI. ulti-layered shielding is often used around transponders, RF front ends and digital processors to isolate them from radiated noise.

③ Grounding and Bonding:

proper grounding and bonding techniques ensure that all satellite structures and electronic modules share a common electrical reference. This minimizes potential differences that could lead to EMI coupling. conductive paths are carefully designed to avoid ground loops and ensure low impedance return paths for high frequency currents.

④ Redundancy and Fault Tolerance:

critical communication paths are often duplicated with redundant transceivers, antennas and signal processors. this ensures continued operation even if one path is compromised by EMI or radiation damage.

By integrating these strategies, satellite designers can significantly enhance electromagnetic compatibility and ensure robust communication links in space. As satellite constellations grow and missions become more complex, EMI mitigation with remain a cornerstone of reliable aerospace engineering.

Simulation Questions

1. A lossless 30 m long transmission line with Z_o=50Ω is established between two ground stations which operate at 2 MHz. The line is terminated with a load ZL = 60+j40Ω. If u = 0.6c on the line. Write a **Scilab code** to plot the reflection coefficient (Γ), standing wave ratio (S) and input impedance in smith chart.

CODE:

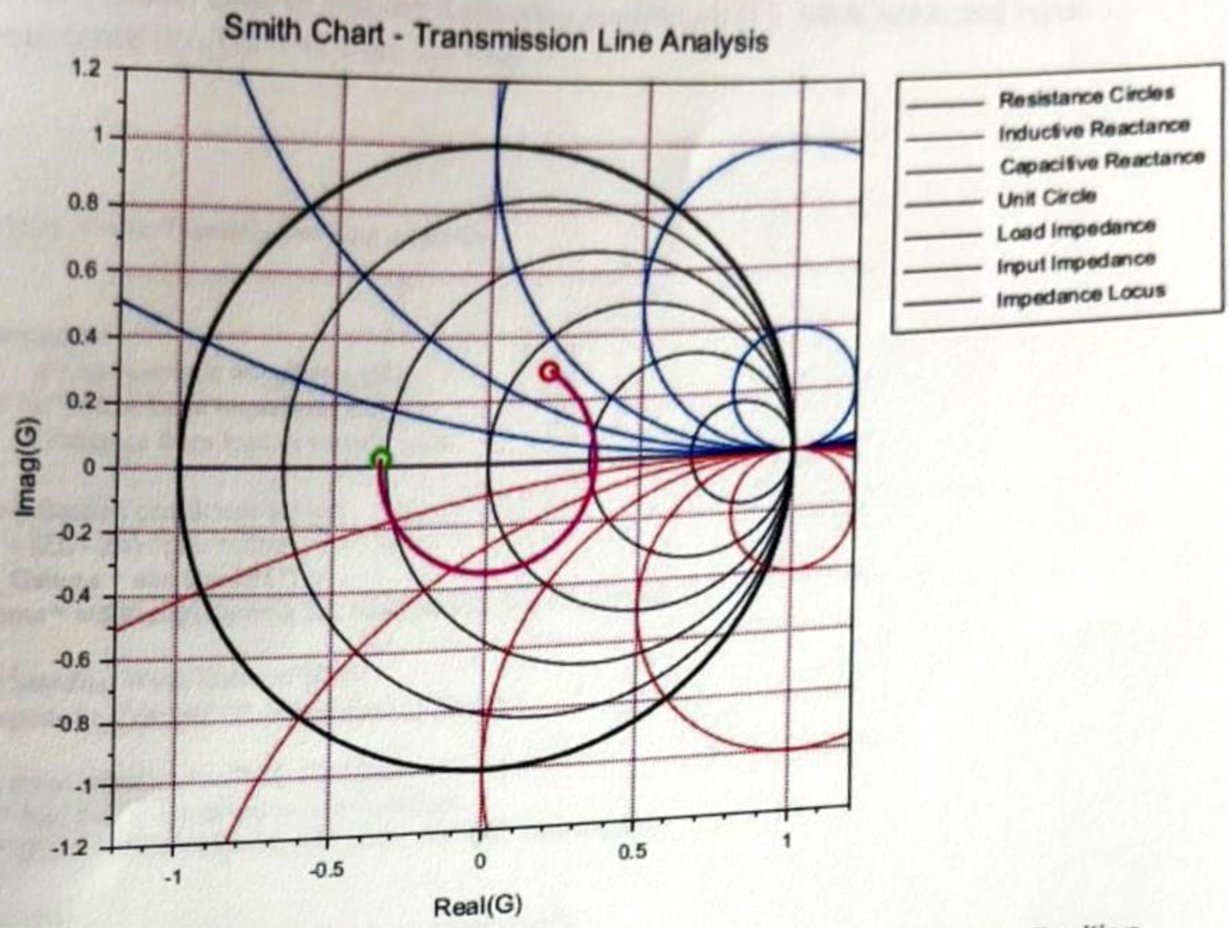
```
// Transmission Line Analysis and Smith Chart Plotting
clear; clc;

// Given parameters
f = 2e6;       // Frequency (Hz)
L = 30;        // Line length (m)
Zo = 50;        // Characteristic impedance (Ω)
ZL = 60 + %i*40; // Load impedance (Ω)
c = 3e8;       // Speed of light (m/s)
u = 0.6*c;     // Phase velocity (m/s)

// Calculate derived parameters
lambda = u/f;  // Wavelength (m)
beta = 2*%pi/lambda; // Phase constant (rad/m)
electrical_length = L/lambda; // Electrical length

// Calculate reflection coefficient at load
Gamma_L = (ZL - Zo)/(ZL + Zo);
magnitude_Gamma = abs(Gamma_L);
phase_Gamma = atan(imag(Gamma_L), real(Gamma_L));

// Calculate Standing Wave Ratio (SWR)
S = (1 + magnitude_Gamma)/(1 - magnitude_Gamma);

// Calculate input impedance
theta = beta * L; // Electrical length in radians
Z_in = Zo * (ZL + %i*Zo*tan(theta))/(Zo + %i*ZL*tan(theta));

// Display results
printf("Transmission Line Analysis Results:\n");
printf("================================\n");
printf("Frequency: %.2f MHz\n", f/1e6);
printf("Line Length: %.1f m\n", L);
printf("Wavelength: %.2f m\n", lambda);
printf("Electrical Length: %.3f wavelengths\n", electrical_length);
printf("Characteristic Impedance: %.1f Ω\n", Zo);
printf("Load Impedance: %.1f + j%.1f Ω\n", real(ZL), imag(ZL));
printf("\n");
printf("Reflection Coefficient (Γ):\n");
printf("  Magnitude: %.4f\n", magnitude_Gamma);
printf("  Phase: %.2f degrees\n", phase_Gamma*180/%pi);
printf("  Complex: %.4f + j%.4f\n", real(Gamma_L), imag(Gamma_L));
printf("\n");
printf("Standing Wave Ratio (S): %.4f\n", S);
printf("\n");
printf("Input Impedance:\n");
printf("  Z_in = %.2f + j%.2f Ω\n", real(Z_in), imag(Z_in));
printf("  Magnitude: %.2f Ω\n", abs(Z_in));
```

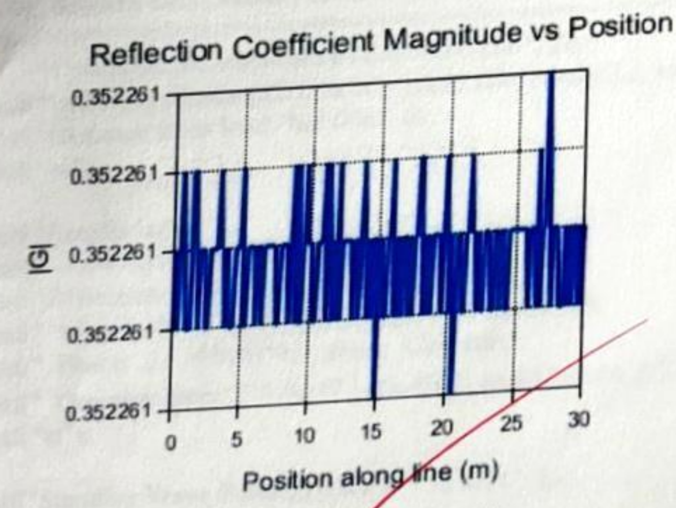
20/20

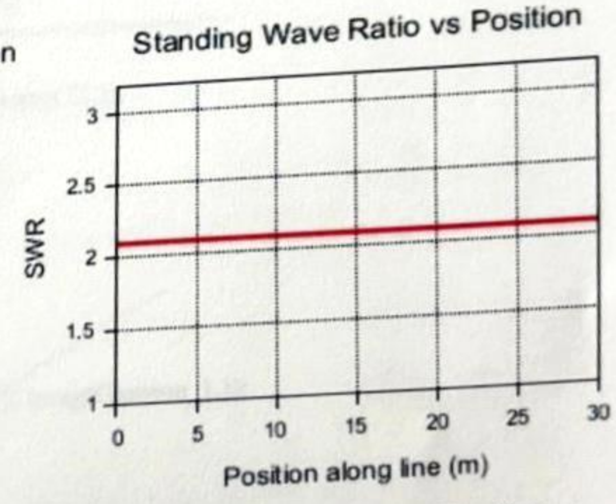
```
// Create Smith Chart visualization
// We'll create a custom Smith chart and plot the impedance trajectory

// Generate points for Smith chart circles
theta_circle = linspace(0, 2*%pi, 200);

// Resistance circles
r_values = [0.2, 0.5, 1, 2, 5]; // Normalized resistance values
// Reactance circles
x_values = [0.2, 0.5, 1, 2, 5]; // Normalized reactance values

// Create figure for Smith chart
scf(0);
clf();
xset("colormap", jetcolormap(64));

// Plot Smith chart background (constant resistance circles)
for i = 1:length(r_values)
   r = r_values(i);
   center_x = r/(1+r);
   radius = 1/(1+r);

   x_circle = center_x + radius*cos(theta_circle);
   y_circle = radius*sin(theta_circle);

   plot(x_circle, y_circle, 'k-', 'linewidth', 0.5);

   // Plot negative y for symmetry
   plot(x_circle, -y_circle, 'k-', 'linewidth', 0.5);
end

// Plot constant reactance circles
for i = 1:length(x_values)
   x = x_values(i);
   center_x = 1;
   center_y = 1/x;
   radius = abs(1/x);

   x_circle = center_x + radius*cos(theta_circle);
   y_circle = center_y + radius*sin(theta_circle);

   plot(x_circle, y_circle, 'b-', 'linewidth', 0.5);

   // Negative reactance circles
   center_y_neg = -1/x;
   y_circle_neg = center_y_neg + radius*sin(theta_circle);
   plot(x_circle, y_circle_neg, 'r-', 'linewidth', 0.5);
end

// Plot the real axis
plot([-1, 1], [0, 0], 'k-', 'linewidth', 1);

// Plot unit circle
plot(cos(theta_circle), sin(theta_circle), 'k-', 'linewidth', 2);

// Calculate normalized impedances
ZL_normalized = ZL/Zo;
Z_in_normalized = Z_in/Zo;
```

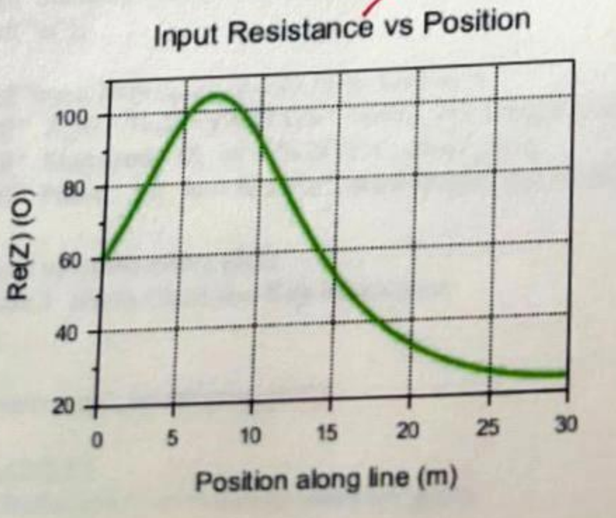
```
// Convert impedances to reflection coefficients
Gamma_ZL = (ZL_normalized - 1)/(ZL_normalized + 1);
Gamma_Zin = (Z_in_normalized - 1)/(Z_in_normalized + 1);

// Plot impedance points on Smith chart
plot(real(Gamma_ZL), imag(Gamma_ZL), 'ro', 'markersize', 8, 'linewidth', 2);
plot(real(Gamma_Zin), imag(Gamma_Zin), 'go', 'markersize', 8, 'linewidth', 2);

// Plot the impedance transformation along the line
positions = linspace(0, L, 100);
Gamma_positions = zeros(1, length(positions));

for i = 1:length(positions)
  l = positions(i);
  Gamma_positions(i) = Gamma_L * exp(-%i*2*beta*l);
end

plot(real(Gamma_positions), imag(Gamma_positions), 'm-', 'linewidth', 2);

// Add labels and legend
xlabel('Real(Γ)');
ylabel('Imag(Γ)');
title('Smith Chart - Transmission Line Analysis');
legend(['Resistance Circles'; 'Inductive Reactance'; 'Capacitive Reactance'; 'Unit Circle'; ...
    'Load Impedance'; 'Input Impedance'; 'Impedance Locus'], -1);

// Set equal aspect ratio and grid
a = gca();
a.isoview = "on";
a.grid = [1 1]*color('gray')*0.7;
a.data_bounds = [-1.2, -1.2; 1.2, 1.2];

// Create additional plots for detailed analysis
scf(1);
clf();
subplot(2,2,1);

// Plot magnitude of reflection coefficient vs position
positions_plot = linspace(0, L, 200);
Gamma_mag_plot = zeros(1, length(positions_plot));

for i = 1:length(positions_plot)
  l = positions_plot(i);
  Gamma_current = Gamma_L * exp(-%i*2*beta*l);
  Gamma_mag_plot(i) = abs(Gamma_current);
end

plot(positions_plot, Gamma_mag_plot, 'b-', 'linewidth', 2);
xlabel('Position along line (m)');
ylabel('|Γ|');
title('Reflection Coefficient Magnitude vs Position');
xgrid(1);

subplot(2,2,2);
// Plot SWR vs position (constant for lossless line)
SWR_plot = S * ones(1, length(positions_plot));
plot(positions_plot, SWR_plot, 'r-', 'linewidth', 2);
xlabel('Position along line (m)');
ylabel('SWR');
title('Standing Wave Ratio vs Position');
```

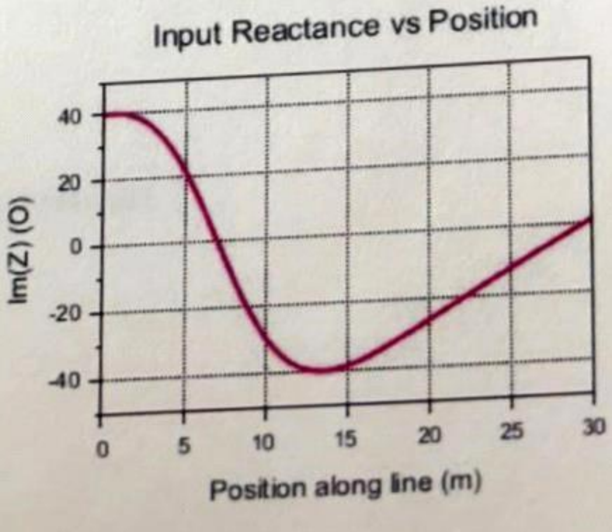
```
xgrid(1);

subplot(2,2,3);
// Plot input impedance real part vs position
Z_real_plot = zeros(1, length(positions_plot));
for i = 1:length(positions_plot)
   l = positions_plot(i);
   theta_current = beta * l;
   Z_current = Zo * (ZL + %i*Zo*tan(theta_current))/(Zo + %i*ZL*tan(theta_current));
   Z_real_plot(i) = real(Z_current);
end
plot(positions_plot, Z_real_plot, 'g-', 'linewidth', 2);
xlabel('Position along line (m)');
ylabel('Re(Z) (Ω)');
title('Input Resistance vs Position');
xgrid(1);

subplot(2,2,4);
// Plot input impedance imaginary part vs position
Z_imag_plot = zeros(1, length(positions_plot));
for i = 1:length(positions_plot)
   l = positions_plot(i);
   theta_current = beta * l;
   Z_current = Zo * (ZL + %i*Zo*tan(theta_current))/(Zo + %i*ZL*tan(theta_current));
   Z_imag_plot(i) = imag(Z_current);
end
plot(positions_plot, Z_imag_plot, 'm-', 'linewidth', 2);
xlabel('Position along line (m)');
ylabel('Im(Z) (Ω)');
title('Input Reactance vs Position');
xgrid(1);

// Display summary in console
printf("\nSummary:\n");
printf("========\n");
printf("The Smith chart shows:\n");
printf("- Load impedance (red): Z_L = %.1f + j%.1f Ω\n", real(ZL), imag(ZL));
printf("- Input impedance (green): Z_in = %.1f + j%.1f Ω\n", real(Z_in), imag(Z_in));
printf("- The purple curve shows impedance transformation along the line\n");
printf("- SWR remains constant at %.4f (lossless line)\n", S);
```


Smith Chart - Transmission Line Analysis
Resistance Circles
Inductive Reactance
Capacitive Reactance
Unit Circle
Load Impedance
Input Impedance
Impedance Locus
Imag(G)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1
-0.5
0
0.5
1
Real(G)


Reflection Coefficient Magnitude vs Position
|G|
0.352261
0.352261
0.352261
0.352261
0.352261
0
5
10
15
20
25
30
Position along line (m)


Standing Wave Ratio vs Position
SWR
3
2.5
2
1.5
1
0
5
10
15
20
25
30
Position along line (m)


Input Resistance vs Position
Re(Z) (O)
100
80
60
40
20
0
5
10
15
20
25
30
Position along line (m)


Input Reactance vs Position
Im(Z) (O)
40
20
0
-20
-40
0
5
10
15
20
25
30
Position along line (m)

2. In a satellite base station, a load of 100 + j150 Ω is connected to a 75Ω lossless line. Write a **Scilab code** to plot the Reflection coefficient (Γ), SWR value and input impedance (Zin) at 0.4λ from the load.

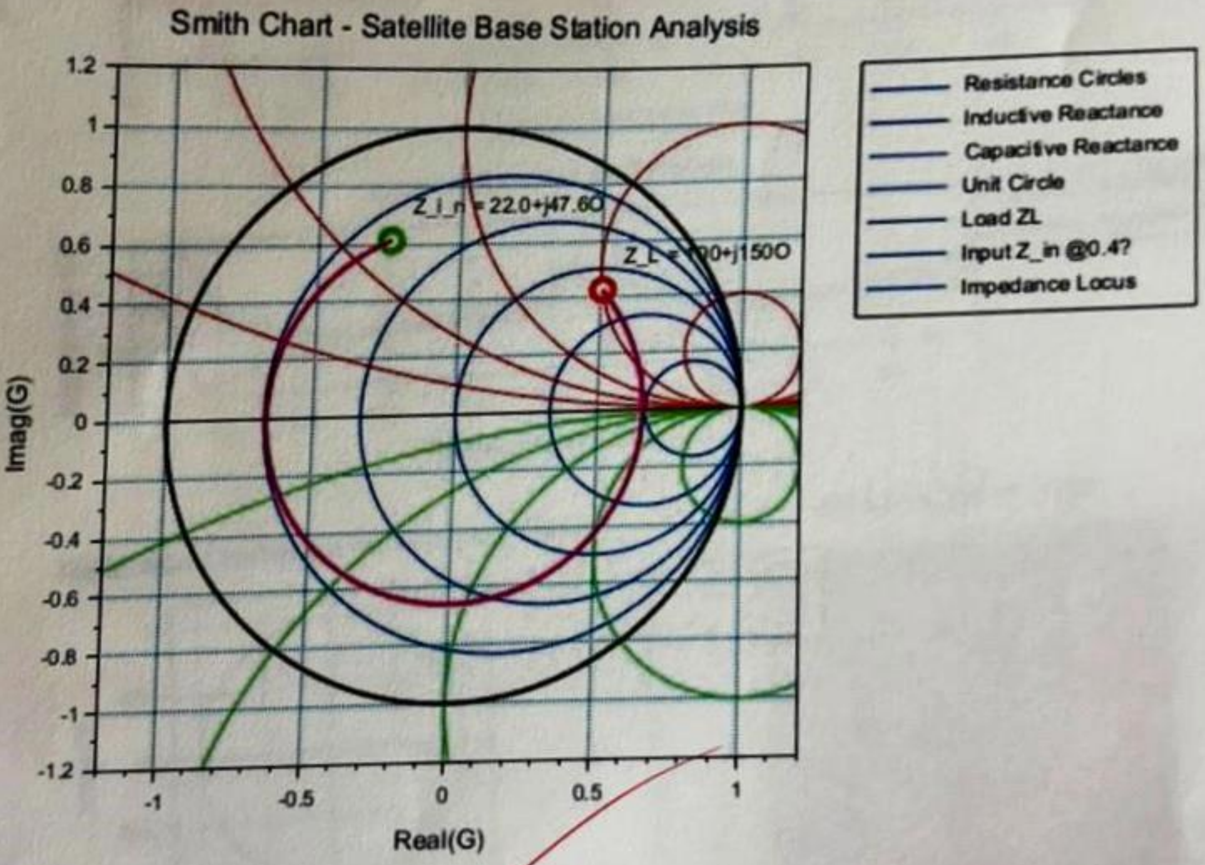
CODE:

```
// Satellite Base Station Transmission Line Analysis
clear; clc;

// Given parameters
Zo = 75;       // Characteristic impedance (Ω)
ZL = 100 + %i*150; // Load impedance (Ω)
d = 0.4;       // Distance from load in wavelengths

// Calculate reflection coefficient at load
Gamma_L = (ZL - Zo) / (ZL + Zo);
magnitude_Gamma = abs(Gamma_L);
phase_Gamma = atan(imag(Gamma_L), real(Gamma_L)) * 180/%pi;

// Calculate Standing Wave Ratio (SWR)
S = (1 + magnitude_Gamma) / (1 - magnitude_Gamma);

// Calculate input impedance at 0.4λ from load
beta_d = 2 * %pi * d; // Electrical length in radians
Z_in = Zo * (ZL + %i*Zo*tan(beta_d)) / (Zo + %i*ZL*tan(beta_d));

// Display results
printf("Satellite Base Station Transmission Line Analysis\n");
printf("==================================================\n");
printf("Characteristic Impedance (Zo): %.0f Ω\n", Zo);
printf("Load Impedance (ZL): %.0f + j%.0f Ω\n", real(ZL), imag(ZL));
printf("Distance from load: %.1fλ\n", d);
printf("\n");

printf("Results:\n");
printf("--------\n");
printf("Reflection Coefficient (Γ):\n");
printf("  Magnitude: |Γ| = %.4f\n", magnitude_Gamma);
printf("  Phase: ∠Γ = %.2f°\n", phase_Gamma);
printf("  Complex form: Γ = %.4f + j%.4f\n", real(Gamma_L), imag(Gamma_L));
printf("\n");

printf("Standing Wave Ratio (SWR): S = %.4f\n", S);
printf("\n");

printf("Input Impedance at 0.4λ from load:\n");
printf("  Z_in = %.2f + j%.2f Ω\n", real(Z_in), imag(Z_in));
printf("  Magnitude: |Z_in| = %.2f Ω\n", abs(Z_in));
printf("  Phase: ∠Z_in = %.2f°\n", atan(imag(Z_in), real(Z_in))*180/%pi);

// Create comprehensive plots
// Figure 1: Smith Chart and Key Parameters
scf(0);
clf();
xset("colormap", jetcolormap(64));

subplot(2,2,1);
// Plot Reflection Coefficient on complex plane
```

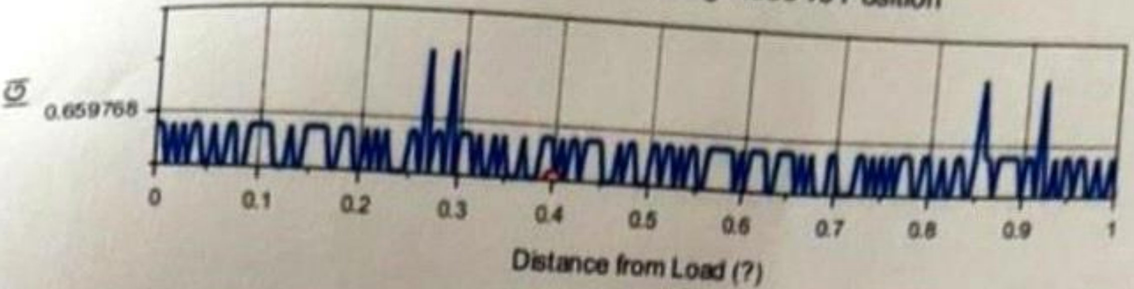
```
theta = linspace(0, 2**%pi, 100);
plot(cos(theta), sin(theta), 'k-', 'linewidth', 1); // Unit circle
plot([-1 1], [0 0], 'k:', 'linewidth', 0.5);
plot([0 0], [-1 1], 'k:', 'linewidth', 0.5);

// Plot Gamma point
plot(real(Gamma_L), imag(Gamma_L), 'ro', 'markersize', 8, 'linewidth', 2);
xlabel('Real(Γ)');
ylabel('Imag(Γ)');
title('Reflection Coefficient on Complex Plane');
a = gca();
a.isoview = "on";
a.data_bounds = [-1.1, -1.1; 1.1, 1.1];
xgrid(1);

// Add annotation for Gamma value
xstring(real(Gamma_L)+0.1, imag(Gamma_L)+0.1, sprintf('Γ = %.3f∠%.1f°', magnitude_Gamma,
phase_Gamma));

subplot(2,2,2);
// Display SWR value prominently
bar(1, S, 'r');
a = gca();
a.x_ticks = tlist(["ticks", "locations", "labels"], [1], ["SWR"]);
ylabel('Standing Wave Ratio');
title('Standing Wave Ratio');
ylim([1, 2.5]);
// Add value text
xstring(1, S+0.1, sprintf('S = %.4f', S));

subplot(2,2,3);
// Display impedance comparison
impedances = [real(ZL), imag(ZL), abs(ZL); real(Z_in), imag(Z_in), abs(Z_in)];
bar([1, 2, 3; 1, 2, 3]', impedances', 'grouped');
legend(['Load ZL'; 'Input Z_in'], -1);
a = gca();
a.x_ticks = tlist(["ticks", "locations", "labels"], [1,2,3], ["Real", "Imag", "Mag"]);
ylabel('Impedance (Ω)');
title('Impedance Comparison');

subplot(2,2,4);
// Polar plot of Gamma
theta_polar = linspace(0, 2**%pi, 100);
polarplot(theta_polar, ones(1,100), 'k-'); // Unit circle
polarplot(phase_Gamma**%pi/180, magnitude_Gamma, 'ro', 'markersize', 8, 'linewidth', 2);
title('Reflection Coefficient (Polar)');

// Figure 2: Smith Chart Implementation
scf(1);
clf();

// Create custom Smith Chart
theta = linspace(0, 2**%pi, 200);

// Resistance circles
r_values = [0.2, 0.5, 1, 2, 5];
for i = 1:length(r_values)
  r = r_values(i);
  center = r/(r+1);
  radius = 1/(r+1);
```

```
    x_circle = center + radius*cos(theta);
    y_circle = radius*sin(theta);
    plot(x_circle, y_circle, 'b-', 'linewidth', 0.7);
    plot(x_circle, -y_circle, 'b-', 'linewidth', 0.7);
end

// Reactance circles
x_values = [0.2, 0.5, 1, 2, 5];
for i = 1:length(x_values)
    x = x_values(i);
    center_x = 1;
    center_y = 1/x;
    radius = abs(1/x);
    x_circle = center_x + radius*cos(theta);
    y_circle = center_y + radius*sin(theta);
    plot(x_circle, y_circle, 'r-', 'linewidth', 0.7);

    // Negative reactance
    center_y_neg = -1/x;
    y_circle_neg = center_y_neg + radius*sin(theta);
    plot(x_circle, y_circle_neg, 'g-', 'linewidth', 0.7);
end

// Unit circle and axes
plot(cos(theta), sin(theta), 'k-', 'linewidth', 2);
plot([-1 1], [0 0], 'k-', 'linewidth', 1);

// Calculate normalized impedances and their Gamma points
ZL_norm = ZL/Zo;
Z_in_norm = Z_in/Zo;

Gamma_ZL = (ZL_norm - 1)/(ZL_norm + 1);
Gamma_Zin = (Z_in_norm - 1)/(Z_in_norm + 1);

// Plot impedance points
plot(real(Gamma_ZL), imag(Gamma_ZL), 'ro', 'markersize', 10, 'linewidth', 3);
plot(real(Gamma_Zin), imag(Gamma_Zin), 'go', 'markersize', 10, 'linewidth', 3);

// Plot impedance transformation along the line
positions = linspace(0, d, 50);
Gamma_trajectory = zeros(1, length(positions));
for i = 1:length(positions)
    dist = positions(i);
    beta_dist = 2 * %pi * dist;
    Gamma_trajectory(i) = Gamma_L * exp(-%i * 2 * beta_dist);
end
plot(real(Gamma_trajectory), imag(Gamma_trajectory), 'm-', 'linewidth', 2);

// Formatting
xlabel('Real(Γ)');
ylabel('Imag(Γ)');
title('Smith Chart - Satellite Base Station Analysis');
legend(['Resistance Circles'; 'Inductive Reactance'; 'Capacitive Reactance'; ...
      'Unit Circle'; 'Load ZL'; 'Input Z_in @0.4λ'; 'Impedance Locus'], -1);
a = gca();
a.isoview = "on";
a.data_bounds = [-1.1, -1.1; 1.1, 1.1];
a.grid = [1 1]*color('gray')*0.5;

// Add impedance values on the chart
```

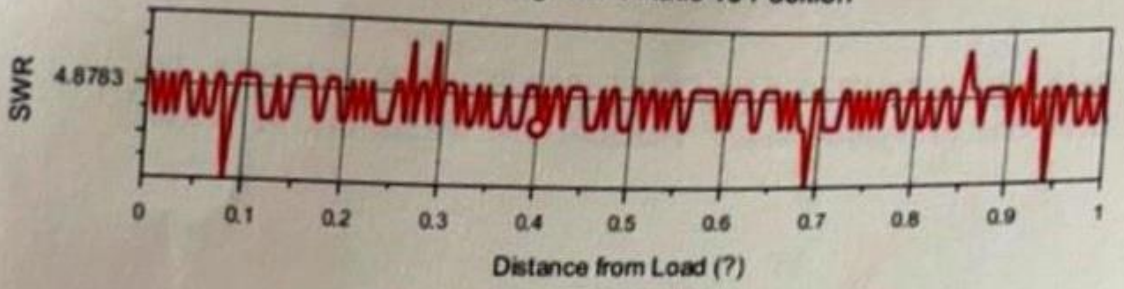
```
xstring(real(Gamma_ZL)+0.05, imag(Gamma_ZL)+0.05, ...
      sprintf('Z_L = %.0f+j%.0fΩ', real(ZL), imag(ZL)));
xstring(real(Gamma_Zin)+0.05, imag(Gamma_Zin)+0.05, ...
      sprintf('Z_i_n = %.1f+j%.1fΩ', real(Z_in), imag(Z_in)));

// Figure 3: Parameter variation along transmission line
scf(2);
clf();

// Analyze along entire wavelength
positions_full = linspace(0, 1, 200); // 0 to 1 wavelength
Gamma_full = zeros(1, length(positions_full));
Z_in_full = zeros(1, length(positions_full));
SWR_full = zeros(1, length(positions_full));

for i = 1:length(positions_full)
   dist = positions_full(i);
   beta_dist = 2 * %pi * dist;
   Gamma_current = Gamma_L * exp(-%i * 2 * beta_dist);
   Gamma_full(i) = Gamma_current;

   Z_current = Zo * (ZL + %i*Zo*tan(beta_dist)) / (Zo + %i*ZL*tan(beta_dist));
   Z_in_full(i) = Z_current;

   SWR_full(i) = (1 + abs(Gamma_current)) / (1 - abs(Gamma_current));
end

subplot(3,1,1);
plot(positions_full, abs(Gamma_full), 'b-', 'linewidth', 2);
xlabel('Distance from Load (λ)');
ylabel('|Γ|');
title('Reflection Coefficient Magnitude vs Position');
xgrid(1);
// Mark our point of interest
plot(d, abs(Gamma_L * exp(-%i * 2 * 2*%pi*d)), 'ro', 'markersize', 8, 'linewidth', 2);
xstring(d, abs(Gamma_L * exp(-%i * 2 * 2*%pi*d)) + 0.05, '0.4λ point');

subplot(3,1,2);
plot(positions_full, SWR_full, 'r-', 'linewidth', 2);
xlabel('Distance from Load (λ)');
ylabel('SWR');
title('Standing Wave Ratio vs Position');
xgrid(1);
plot(d, (1 + abs(Gamma_L * exp(-%i * 2 * 2*%pi*d))) / (1 - abs(Gamma_L * exp(-%i * 2 * 2*%pi*d))), ...
   'ro', 'markersize', 8, 'linewidth', 2);

subplot(3,1,3);
plot(positions_full, real(Z_in_full), 'g-', 'linewidth', 2, 'label', 'Real(Z_in)');
plot(positions_full, imag(Z_in_full), 'm-', 'linewidth', 2, 'label', 'Imag(Z_in)');
xlabel('Distance from Load (λ)');
ylabel('Impedance (Ω)');
title('Input Impedance vs Position');
legend(['Real Part'; 'Imaginary Part'], -1);
xgrid(1);
plot(d, real(Z_in), 'go', 'markersize', 8, 'linewidth', 2);
plot(d, imag(Z_in), 'mo', 'markersize', 8, 'linewidth', 2);

// Print detailed analysis
printf("\nDetailed Analysis:\n");
printf("====================\n");
printf("Normalized Load Impedance: Z_L/Zo = %.3f + j%.3f\n", real(ZL_norm), imag(ZL_norm));
printf("Normalized Input Impedance: Z_in/Zo = %.3f + j%.3f\n", real(Z_in_norm), imag(Z_in_norm));
printf("\n");
printf("At 0.4λ from load:\n");
printf("  Reflection Coefficient rotates by: %.1f° clockwise\n", d * 720);
printf("  New Gamma phase: %.1f°\n", atan(imag(Gamma_L * exp(-%i * 2 * beta_d)), ...
                        real(Gamma_L * exp(-%i * 2 * beta_d)))*180/%pi);
printf("\n");
printf("The Smith chart shows the impedance transformation from load to input.\n");
printf("The purple curve represents how the impedance appears to change along the line.\n");
```



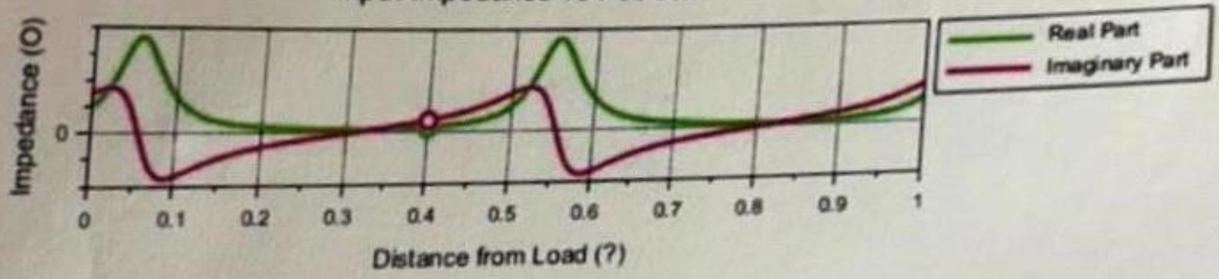
Reflection Coefficient Magnitude vs Position



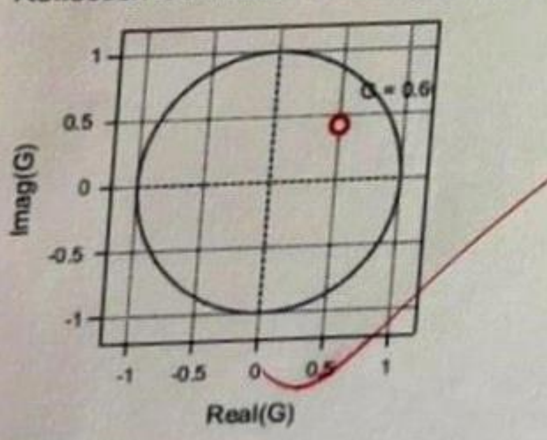
Standing Wave Ratio vs Position



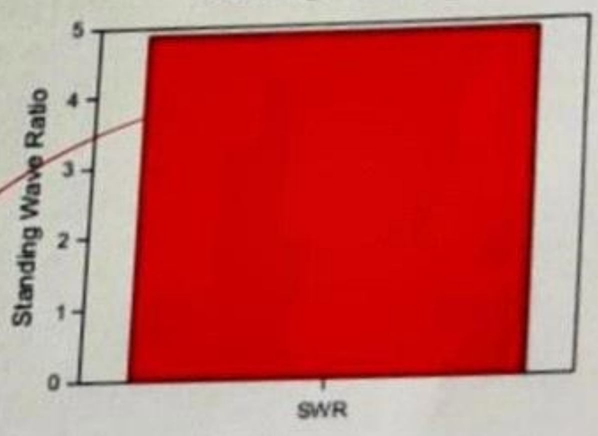
Input Impedance vs Position



Reflection Coefficient on Complex Plane



Standing Wave Ratio



Impedance Comparison

Impedance (O)