Simulación en SimsCape para el ejercicio 3, Técnicas de solución.

Para el circuito de la figura 1, encontrar las variables solicitadas en el mismo, empleando el software SimsCape.





En primer lugar, buscamos nuestras fuentes de tensión, así como vemos en la Figura 2.





En donde la configuración de esta fuente de voltaje, se ilustra en la Figura 3.

🔁 Block Parameters: Fuente de Voltaje	ĸ
AC Voltage Source (mask) (link)	
Ideal sinusoidal AC Voltage source.	
Parameters Load Flow	
Peak amplitude (V): 200	
Phase (deg): 0	
Frequency (Hz): 60	
Sample time: 0	
Measurements Voltage	•
OK Cancel Help Apply	



Como se ilustra en la Figura 3, en la configuración de la fuente de voltaje, nos solicita un voltaje pico, un ángulo de desfase (en grados), la frecuencia en que opera la fuente y el tiempo de muestreo, entonces nos dirigimos a la Figura 1, y completamos los datos solicitados.





Ahora escogemos la fuente de corriente que tenemos propuesta en el circuito, para ello, la encontramos como se ilustra en la Figura 4.

🔀 Block Parameters: Fuente de Corriente	×
AC Current Source (mask) (link)	
Ideal sinusoidal AC current source. The positive current direction is indicated by the arrow.	
Parameters	
Peak amplitude (A):	
10	
Phase (deg):	
90	
Frequency (Hz):	
60	
Sample time:	
0	
Measurements Current -	
OK Cancel Help Apply	

Figura 5

En la Figura 5, podemos evidenciar la forma de configurar nuestra fuente de corriente, que, de igual manera que la fuente de voltaje, los datos solicitados son los mismos.

Para completar nuestro circuito, nos falta los elementos pasivos del mismo, para eso, lo encontramos como se evidencia en la Figura 6.



Figura 6

Pero este bloque nos indica un elemento RLC, pero debemos configurarlo para adaptarlo a lo que necesite3mos, así como vemos en la Figura 7.

🔁 Block Parameters: 400 [Ω] X	🚹 Block Parameters: -j350 [Ω] 🛛 🗙		
Series RLC Branch (mask) (link)	Series RLC Branch (mask) (link)		
Implements a series branch of RLC elements. Use the 'Branch type' parameter to add or remove elements from the branch.	Implements a series branch of RLC elements. Use the 'Branch type' parameter to add or remove elements from the branch.		
Parameters	Parameters		
Branch type: R	Branch type: C		
Resistance (O R	Capacitance (I RLC		
400 L C	7.578806814 L C		
Measurement: RL RC	Set the init RL		
LC Open circuit	Measurements LC		
Open circuit cancel help Appiy			
Block Parameters: -j660 [Ω] X	OK Cancel Help Apply		
Series RLC Branch (mask) (iink) Implements a series branch of RLC elements	Block Parameters: j530 [Ω] ×		
Use the 'Branch type' parameter to add or remove elements from the	Series RLC Branch (mask) (link)		
branch.	Use the 'Branch type' parameter to add or remove elements from the		
Parameters	branch.		
Branch type: C	Parameters		
Capacitance (I R	Branch type: L		
	Inductance (H R		
RC			
Open circuit	RC		
OK Cancel Help Apply	Measurements LC Open circuit		
Block Parameters: 200 [O]	OK Cancel Help Apply		
Series RLC Branch (mask) (link)	Pa Block Parameters: 220 [O]		
Implements a series branch of RLC elements.	Series RLC Branch (mask) (link)		
Use the 'Branch type' parameter to add or remove elements from the branch.	Implements a series branch of RLC elements.		
Parameters	Use the 'Branch type' parameter to add or remove elements from the branch.		
Branch type: R	Parameters		
Resistance (O	Branch type: L		
200 L	Inductance (H RLC		
Measurement: RL	0.742723067 L		
LC	Set the initi RL		
Open circuit Cancer Help Appry	RC Measurements LC		
	Open circuit		
	OK Cancel Help Apply		

Figura 7

Para encontrar los valores de bobina y Condensador, simplemente nos devolvemos con las ecuaciones de impedancia inductiva y capacitiva, y encontramos los valores en el dominio del tiempo.





Ahora debemos colocar nuestros instrumentos de medición, voltímetro y amperímetro. En donde, los encontramos como se ilustra en la Figura 8.

🖥 🖥 Simulink Library Browser					– 🗆 ×
🗢 💠 solver 🗸 🗸 🗸 🔁 🗸 🔁 🗘	2				
Simulink/Math Operations					
 ✓ Simulink Commonly Used Blocks Continuous > Dashboard Discontinuities Discrete Logic and Bit Operations Lookup Tables Math Operations Messages & Events Model Verification Signal Attributes Signal Routing Sinks 	$\begin{array}{c} & & \\$	Add Complex to Real-Imag Magnitude-Angle to Complex Magnitude-Angle to Complex	$\begin{array}{c} & \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c}$	Y0 A Y Assignment Dot Product	Find Nonzero Elements MinMax
Sources String User-Defined Functions > Additional Math & Discrete > Quick Insert Fixed-Point Designer > Fixed-Point Designer HDL Support > HDL Code > Simsufink 3D Animation > Simulink SD Animation > Simulink Extras Stateflow > System Identification Toolbox Recently Used	Resettable	Reciprocal Sqrt Sine Wave Function	Reshape 1 Slider Gain V Sum of Elements	$\begin{array}{c} & & \\ \hline \\ \hline$	Sign Sign Squeeze
	Vector Concatenate	Veighted Sample Time Math			

Figura 9

En la Figura 9, encontramos un bloque que nos será muy útil para interpretar los resultados del voltímetro y amperímetro, este bloque se llama "complex to Magnitude-Angle", en donde sabemos que el amperímetro y voltímetro nos arroja el resultado en número rectangular, el cual este bloque nos ayuda a convertir este número a su forma polar.

Pero en la Figura 9, adicionalmente escogimos un bloque de producto y de división, estos los utilizaremos para crear un conversor de radianes a grados, porque el bloque anterior, nos suministra un dato de ángulo en radianes, y para mayor entendimiento, construimos un conversor, con ayuda también de un bloque de fuente constante, que lo encontramos en Simulink-sources. El conversor nos queda como se ilustra en la Figura 10.



Figura 10

Pero para simplificar nuestro esquema, seleccionamos todo nuestro conversor y automáticamente nos sale un recuadro azul, con tres puntos en la parte inferior derecha, en donde nos arroja un pequeño menú, y seleccionamos "create Subsystem", esto nos ayuda a crear un solo bloque con esta configuración, así como se ve en la Figura 11.



Figura 11

Ahora nos falta un instrumento para visualizar el resultado, el bloque que nos sirve, es el que se llama display, el cual encontramos como se ilustra en la Figura 12.

🖥 🖥 Simulink Library Browser		– 🗆 X						
	& ▼ 🖏 ▼ 🚍 ▼ 💣 😐 🧿							
Simulink/Sinks								
 Simulink Commonly Used Blocks Continuous Dashboard Discontinuities Discrete Logic and Bit Operations Lookup Tables Math Operations Messages & Events Model Verification Model Verification Model Verification Model Verification Model Verification Signal Attributes Signal Routing Sinks Sources String User-Defined Functions Additional Math & Discrete Quick Insert Fixed-Point Designer 	 Display Floating Scope I Out1 Record Stop Simulation Terminator Simout Simout To Workspace XY Graph 	> OutBus .signal1 Out Bus Element Scope ↓ untitled.mat To File						



Y por último, nos hace falta un bloque que nos ayude a correr el software, el cual se llama "Powergui", sin este bloque, no podemos ejecutar nuestro programa y nos generaría error, para este caso configuramos nuestro bloque powegui en phasor, en donde le ajustamos a una frecuencia de 60 [*H*], este bloque lo encontramos como se muestra en la Figura 13.



Figura 13



Figura 14

En la Figura 14, podemos ver el esquema completo de nuestro circuito en donde al seleccionar cada elemento, le cambiamos el nombre con los valores del circuito inicial para familiarizar mejor el esquema.

A su vez, podemos ver que el resultado de las variables que nos solicitan se tiene en su forma rectangular y con el conversor que creamos, tenemos la forma polar de nuestro resultado.

Cada valor está marcado con el nombre de la variable solicitada, para mejor interpretación, el cual se ve reflejado en los dispalys.

Con esto, damos por terminado nuestro esquema, esperamos que sea de gran ayuda para sus cursos de circuitos eléctricos.