

Universidad de Alcalá
Escuela Politécnica Superior

Grado en Ingeniería Electrónica y Automática Industrial



Trabajo Fin de Grado

Emulación de líneas de transmisión mediante microcontrolador
F28377D de Texas Instruments

ESCUELA POLITECNICA

Autor: Alejandro Paniagua Tineo

Tutor/es: Santiago Cóbreces Álvarez

2017

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

Emulación de líneas de transmisión mediante microcontrolador
F28377D de Texas Instruments

Autor: Alejandro Paniagua Tineo

Tutor: Santiago Cóbreces Álvarez

TRIBUNAL:

Presidente:

Vocal 1º:

Vocal 2º:

Calificación:

Fecha:

*Este trabajo y todo el camino recorrido no habría sido posible sin la ayuda de mi madre,
mi padre, mi novia Laura, y Bea.
En especial, dar más que las gracias a mi hermano Arturo, por su apoyo y ayuda desde
que comencé en 2012.*

RESUMEN

El siguiente TFG trata de evaluar el comportamiento de corrientes y tensiones presentes en las líneas de transmisión. Para ello se empleará un microcontrolador de Texas Instruments, exactamente el F28377D de la serie C2000. Se llevará a cabo su programación mediante el compilador del microprocesador, *Code Composer Studio* y el entorno *Simulink* de MATLAB. Una vez realizadas simulaciones a nivel teórico, se hará uso de una fuente de tensión trifásica programable que permitirá conectar el microcontrolador a dicha fuente, pudiendo realizar la emulación de un escenario real

Fuente generadora - Línea de transmisión – Equipo bajo test

PALABRAS CLAVE

Línea de transmisión, tensión, carga, simulación, microprocesador, *Simulink*, función de transferencia, discretización

ABSTRACT

The following TFG is going to evaluate the behaviour of the current and voltage in a transmission line. To face this task it will be used a Texas Instruments micro controller, the F28377D C2000 series. It will be coded with the *Code Composer Studio* and with *Simulink* from MATLAB. Once the theoretical simulations are finished, a real 3-phase power voltage source will be used. The micro will be connected to the 3-phase source making possible the emulation of a 3-phase transmission line.

Voltage Source – Transmission line – System under test

KEY WORDS

Transmission line, voltage, load, simulation, microcontroller, *Simulink*, transfer function, discrete time

INDICE

1. Introducción	14
2. Base teórica	18
2.1. Resistencia en cables.....	18
2.2. Inductancia en cables.....	28
2.3. Capacidad en cables.....	31
2.4. Estructuras en pi.....	34
2.5. Ecuaciones del telegrafista.....	35
2.5.1. Líneas cortas.....	36
2.5.2. Líneas medias.....	39
2.5.3. Líneas largas.....	41
2.6. Sistema en variables de estado	47
2.6.1. Con carga conocida	47
2.6.2. En función de la carga	52
2.7. Discretización del modelo y comparación.....	54
3. Soporte físico	66
3.1. Microprocesador F28377D C2000.....	66
3.2. Matlab, entorno <i>Simulink</i>	69
3.3. <i>Code Composer Studio</i>	76
3.4. DAC externo.....	79
3.5. Etapas de adaptación y conexionado.....	81
3.6. Fuente trifásica.....	82
4. Diseño aplicación C	84
4.1. Estructura de la aplicación.	84
4.2. Configuración general.	84
4.2.1. ADC.....	85
4.2.2. SPI.....	86
4.2.3. PIE	87
4.3. Rutina de atención a la interrupción.....	87
4.3.1. Diagrama de flujo.....	88
4.3.2. Calibración de sensores y seguridad.....	89

5. Análisis experimental. Caso de estudio	91
5.1. Análisis del caso.....	91
5.2. Equipo experimental	105
5.3. Análisis en tiempo real. Medida de tiempos.....	105
5.4. Simulaciones en vacío.	106
5.5. Simulación con carga de $I_L=cte$	106
6. Conclusiones	109
6.1. Problemas encontrados.....	109
6.2. Futuras líneas de desarrollo.....	110
7. Bibliografía	112
8. Anexos	114

RESUMEN EXTENDIDO

La electricidad es un recurso completamente necesario hoy en día. Su generación, transporte y almacenamiento son temas muy complejos, por esto, el tema a desarrollar tratará simplemente algunas partes de este entorno complicado de ingeniería. En las siguientes páginas se tratará de mostrar el comportamiento de corrientes y tensiones presentes en una línea de transmisión. ¿Qué entendemos por línea de transmisión? Del inglés, *transmission line*, entendemos que la electricidad comienza a ser transportada desde el momento que sale de una fuente generadora (Central Nuclear, campo de aerogeneradores, central térmica...), hasta que llega a un punto de consumo, sea del tipo que sea (industria, abastecimiento de hogares, alumbrado...). No todas las líneas de transmisión son iguales, sino que dependen de muchos factores, por lo que su clasificación es amplia según diferentes criterios:

- Longitud:
 - o Cortas
 - o Medias
 - o Largas

- Tensión a soportar:
 - o Alta tensión
 - o Media tensión
 - o Baja tensión

- Entorno
 - o Aéreas
 - o Subterráneas
 - o Subacuáticas

En las siguientes líneas se tratará de representar algunas de las líneas anteriores de la forma más fiel posible, tanto con valores extraídos de tablas facilitadas por fabricantes de líneas de transmisión, como con ecuaciones teóricas que permitan a un usuario estimar los valores de una línea de transmisión diseñada por él mismo. Como es lógico, estas ecuaciones no serán exactas ya que el abanico de entornos, longitudes, tensiones, materiales empleados y demás factores es muy amplio. Por lo tanto, estas ecuaciones solo se deberían usar para una primera valoración de la línea a diseñar.

Una vez conocidos los parámetros de la línea que se desea simular, se introducirán en el modelo equivalente. Gracias a este modelo equivalente se realizará la primera simulación a nivel teórico mediante MATLAB. Estas simulaciones simplemente serán una aproximación muy básica de lo que se debe esperar y un primer análisis a nivel estructural de lo que se realizará más adelante.

Existen varios circuitos aceptados para la simulación de líneas de transmisión, pero en este proyecto se empleará el diagrama en pi. Se trata de un diagrama simple que permitirá emular tanto con valores reales como con los estimados mediante ecuaciones, el comportamiento de tensiones y corrientes.

El diseño de líneas de transmisión lleva consigo una reglamentación muy estricta que dictamina los parámetros constructivos que debe tener la línea. Estos parámetros constructivos hablan desde el tipo de cable empleado, aislante necesario, tipo de torretas de soporte, anclaje de las torretas, distancia que debe haber entre las torretas de soporte... Por eso, insistir en que este proyecto se centra en un análisis teórico de las corrientes y tensiones de las líneas desde un punto de vista más teórico, que se ciñe a la ingeniería electrónica y eléctrica. Con ello, se pretende dar una base de apoyo para posibles opciones de diseño de líneas, que en el caso de obtener valores adecuados a nivel eléctrico, fuese posible entrar a realizar una valoración constructiva de la línea.

Actualmente, las líneas de transmisión aéreas son muy superiores en número a las líneas de transmisión de otro tipo, como pueden ser subterráneas o subacuáticas. Esto tiene una explicación muy lógica, y es que hasta hace relativamente poco tiempo, las fuentes generadoras de energía se encontraban en tierra firme, y no kilómetros dentro del mar. Con el auge de las fuentes de energía renovables, sobre todo la generada por aerogeneradores y con la colocación de estos en superficie marina, nace la necesidad de valorar nuevas formas de transportar la electricidad, en entornos que hasta ahora no había sido necesario.

En la actualidad existe un macro proyecto, llevado a cabo frente a las costas de Dinamarca en el que está usando tecnología HVDC (*High Voltage Direct Current*). Esta tecnología, como su nombre indica, consiste en la transmisión de la electricidad en corriente continua. Obviamente, en estas líneas no se va a cuestionar la calidad de estas transmisiones, pero llevan un coste añadido muy elevado ya que los aerogeneradores entregan una tensión en corriente alterna, que deberá ser transformada a corriente continua mediante el uso de convertidores AC-DC. Posteriormente se realizará la transmisión, y por último, se deberá deshacer la conversión, ya que una vez en tierra se emplea la transmisión en corriente alterna, por lo que será necesario otro convertidor DC-AC.

Estas líneas tratan de cuantificar, y en un futuro corregir, las pérdidas y problemas asociados, que hacen que hoy en día la tecnología empleada para tan grandes proyectos sea HVDC y no HVAC.

Desde el punto de vista técnico, se quiere dar distintos niveles de abstracción. En un principio se estudiarán las ecuaciones del telegrafista, que explican el comportamiento de tensiones y corrientes en la red en función de su longitud. Con esto y la información extraída de unas líneas de transmisión subacuáticas, se tratará de establecer unas ecuaciones que representen fielmente los parámetros de este tipo de cables (resistencia equivalente, inductancia equivalente, capacidad equivalente...). Esto será útil para los usuarios que deseen realizar una simulación de una línea de transmisión a su medida.

Seguido a esto, se realizarán simulaciones en el entorno *Simulink* que permitirán obtener una primera impresión del comportamiento de tensiones y corriente a lo largo de la línea. Esto se trata de simples simulaciones de las estructuras en "Pi" que modelan la red, pero en este nivel de abstracción no es algo que se pueda implementar de forma directa en el microprocesador que se va a emplear.

El siguiente paso para que el modelo pueda ser implementado en el microprocesador es obtener su modelo equivalente en variables de estado (VVEE). Esto será explicado más adelante en profundidad, permitiendo al usuario elegir el número de segmentos en los que desea dividir la red. La obtención de las matrices de la función de transferencia se realizará mediante MATLAB y posteriormente será traducido a código en C para poder ser volcado al microprocesador mediante *Code Composer Studio*.

Este modelo en variables de estado debe discretizarse, para lo que se emplearan distintos métodos, discutiendo cual es el más conveniente. Una vez discretizado el modelo, ya es válido para el microprocesador de Texas Instruments.

El flujo de trabajo es algo más complejo que simplemente el volcado al microprocesador, ya que primero se desea controlar la tensión que genera la fuente trifásica, y esto se realizará mediante la generación de una interrupción periódica generada por un *Timer* interno de la CPU, esto despertará la ISR del ADC. Dentro de su interrupción, se convertirá el dato, ejecutarán las matrices de transferencia apropiadas del sistema, calculado la salida total y se enviará al DAC externo por SPI. Aclarar que la tensión medida por el ADC es una tensión entregada por los sensores de *Efecto Hall* presentes en la fuente trifásica, encargados de medir la corriente en la línea.

1. Introducción

El siguiente TFG se basa en la emulación de líneas de transmisión mediante el empleo de una fuente trifásica controlable y un microprocesador de la compañía Texas Instruments, exactamente de la serie C2000 el F28377D.

Para llegar a poder emular una línea de transmisión, se sentarán las bases teóricas de los parámetros que afectan a este tipo de líneas. Por lo tanto, en primer lugar, se plantearán unas ecuaciones que modulen, de la manera más precisa posible estos parámetros.

Por otro lado, también se tomarán líneas de transmisión existentes, que permitirán la simulación de situaciones reales. Con esto, lo que se consigue es saber si las ecuaciones empleadas para la estimación de parámetros son correctas.

Para concluir, se pasará de la abstracción teórica y simulada mediante MATLAB, al empleo del microprocesador, conectado a una fuente trifásica programable, con lo que se tratará de recoger datos reales.

La justificación de este trabajo tiene varias bases de soporte. Principalmente la generación y transporte de energía eléctrica parece algo indispensable hoy en día, tanto para nuevas instalaciones a construir como en la mejora y renovación de instalaciones ya existentes. Pero este trabajo se centra en una parte más concreta y problemática, que es la valoración de los parámetros que aparecen en estas líneas de transmisión, viendo qué sucede en función del tipo de conductor empleado, condiciones de entorno, tipo de aislante empleado, número de cables en la línea y distancia entre ellas.

Más concretamente tiene especial importancia los parámetros presentes en los materiales empleados en la construcción de la línea y las condiciones de entorno, ya que repercuten de manera directa en las inductancias y capacidades que se emplean en el modelo equivalente del circuito. En las transmisiones aéreas no tiene una repercusión muy importante debido a que no es necesario llevar un aislante muy denso, ya que éstas no suelen ser accesibles. En estas líneas se supondrán muchos escenarios en entornos subacuáticos, donde la importancia del aislante es máxima. Actualmente, para las líneas de transmisión subacuáticas se está empleado la tecnología HVDC (*High voltage direct current*), que como su nombre indica, se emplea una transmisión en corriente continua.

Esto presenta una problemática en la conversión, ya que las fuentes generadoras, suelen ser aerogeneradores, que entregan una corriente alterna con un alto voltaje. El uso de los convertidores AC-DC incrementa mucho los costes constructivos de las líneas, así que, en las siguientes líneas se observará la evolución de las tensiones a lo largo de la línea, lo que puede ser una introducción al análisis de una transmisión en corriente alterna.

Los parámetros mencionados anteriormente presentes en las líneas de transmisión se fundamentan en una base teórica, para la que se ha realizado un estudio de cómo varían

los elementos principales (resistencias, inductancias, capacidades) de un modelo representativo de una línea de transmisión, en función de sus características.

Por lo tanto, este trabajo se apoya en una base teórica muy importante, en la que no se profundizará en estas páginas, pero se mencionará de dónde son obtenidas muchas de las expresiones que se emplearán. Como es lógico, no se podrá conseguir una estimación fiel de todas las líneas de transmisión con una única ecuación. En algunos casos se comentará qué componentes de la ecuación se deben modificar para conseguir variar el rango de funcionamiento de las ecuaciones.

A partir de esta base teórica, se programará el microprocesador F28377D con la ayuda de *Code Composer Studio*, y MATLAB para la obtención de las matrices discretizadas de la línea equivalente. En un inicio, se consideró como herramienta principal el entorno *Simulink* de MATLAB, vía que tuvo que ser descartada ante la imposibilidad de realizar una conexión SPI entre el microprocesador y el DAC externo que se va a emplear. MATLAB en sus últimas versiones (R2016b), no contempla la activación del SPI para el micro F28377D en el entorno *Simulink*. Esta vía será factible cuando MATLAB habilite esta función, quedando la emulación más visible. Por este motivo, finalmente se optará por emplear *Code Composer Studio* para el volcado en placa y emulación, pero usando MATLAB en algunos tramos del desarrollo.

Tratando de resumir el diagrama de trabajo de los bloques físicos que se emplearán, podemos distinguir dos flujos: El envío de información hacia la fuente de tensión trifásica y los datos que serán recogidos de esta.

Para poder llevar a cabo los análisis de tensiones y corrientes presentes en la línea, se han debido de realizar una serie de tareas previamente. Principalmente se tiene que obtener el modelo equivalente de la línea que se quiere analizar. Para esto, se usará MATLAB y posteriormente CCS.

La metodología tendrá de forma aproximada la siguiente estructura de trabajo.

- PC en el que será ejecutado *Code Composer Studio (CCS)*.
- CCS será el encargado de volcar el código en el micro F28377D que enviará mediante SPI un dato en binario al DAC externo, con la que se conseguirá fijar una tensión, que será enviada a la fuente trifásica.
- La tensión que se introduce en la fuente de tensión trifásica proveniente del DAC deberá encontrarse entre $\pm 3v$, siendo esta tensión multiplicada por un factor, que finalmente será la tensión de trabajo en la salida de la fuente de tensión trifásica.

En el flujo de “retorno”, es decir la comunicación de vuelta entre la fuente trifásica y el microprocesador:

- Se tomarán medidas de las tensiones y corrientes en varios puntos de interés y que sea posible. Estas tensiones y corrientes no se encuentran en magnitudes que el microcontrolador sea capaz de trabajar con ellas, por lo que tendrán que ser adaptadas con una etapa previa a la recepción de los datos.
- Se modularán las señales mencionadas anteriormente tanto en módulo como en frecuencia, haciéndolas pasar por algún filtro, adaptándolas a los valores aceptables de trabajo y sin componentes frecuenciales no deseadas.
- Las señales adaptadas se harán pasar por el ADC integrado en el microcontrolador. Consiguiendo cerrar el círculo de trabajo.

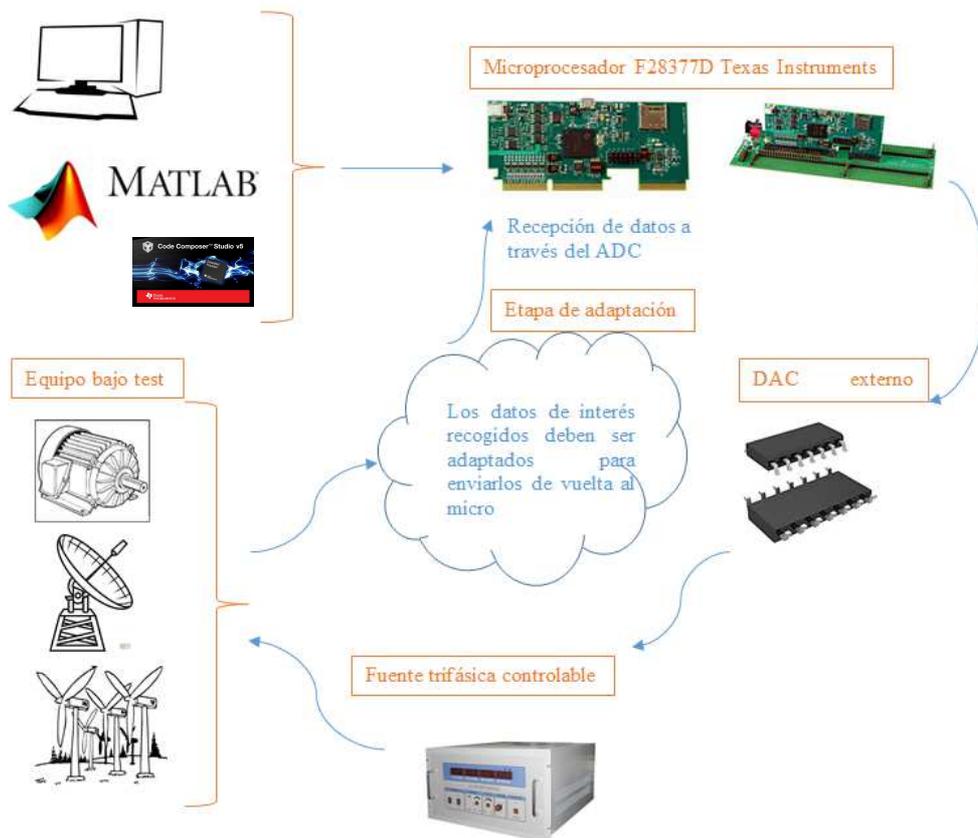


Ilustración 1. Esquema de trabajo del sistema completo

2. Base teórica.

Si queremos emular una línea de transmisión deberemos conocer los componentes del circuito equivalente que ésta presenta. Se emplea una estructura en π , compuesta por una resistencia serie, inductancia serie y dos capacidades. Seguidamente procederemos a valorar los parámetros que modifican los valores de estos componentes del circuito.

Como premisa, tomaremos como línea de transmisión, hilos conductores puntuales, recubiertos con un aislante, en los que se valorará su radio y la distancia a la que se encuentran de otros hilos conductores. Por lo tanto, estas ecuaciones no son únicas para modular los parámetros de las líneas de transmisión, en caso de que otra estructura de cable sea la empleada.

2.1 Resistencia en cables

La resistencia es el componente más básico. Su efecto principal lo podemos encontrar en corriente continua, al contrario de las inductancias y capacidades, que necesitan de una componente frecuencial para hacer su aparición. Más adelante se verá, que la resistencia también presenta diferencias cuando circula corriente continua a corriente alterna.

La resistencia tiene un componente físico muy claro. Representa la oposición del paso de electrones (corriente) entre una diferencia de potencial entre 2 puntos.

Esta resistencia al paso de electrones parece claro pensar que depende del material empleado, por lo que se usan materiales de alta conductividad. Estos son generalmente metales, en los que se analizan otros factores que se nos escapan como peso, variación de temperatura, precio...

También es importante hablar de la sección que presenta el cable conductor, ya que una mayor sección transversal de cable presenta un menor impedimento a la circulación de electrones, es decir, una menor resistencia.

Por último, la longitud. A mayor longitud, mayor resistencia presentará el cable.

En resumen, la resistencia equivalente en serie de un cable depende de los parámetros: resistividad del conductor, longitud del conductor y sección del conductor.

Esto lo podemos ver en la ecuación:

$$R_o = \frac{\rho l}{A} \Omega \quad (2.1.0)$$

El parámetro ρ representa la resistividad. Su dimensión es de $[\Omega \cdot m]$ aunque en muchas tablas se mostrará como $[\Omega \cdot mm^2/m]$. Esto facilita la dimensión de la expresión de R_0 ya que la longitud se suele expresar en metros $[m]$ mientras que la sección del cable, es comúnmente expresada en milímetros cuadrados.

Tabla de resistividades en materiales comúnmente empleados:

Material	$\rho_{20^{\circ}C} (\Omega \cdot mm^2/m)$	Material	$\rho_{20^{\circ}C} (\Omega \cdot mm^2/m)$
Plata	0.016	Níquel	0.08-0.11
Cobre	0.018	Hierro	0.10-0.15
Bronce	0.018-0.056	Estaño	0.11
Oro	0.023	Mercurio	0.96
Aluminio	0.028	Cromo-níquel	1.1
Magnesio	0.045	Bismuto	1.2
Grafito	0.046	PVC	10^{20}
Tungsteno	0.055	Vidrio	10^{20}
Cinc	0.063	Polipropileno	10^{21}
Latón	0.07-0.09	Cuarzo	$4 \cdot 10^{23}$

Tabla 1. Tabla de Resistividades. “Manual práctico electricidad ingenieros”

De esta tabla únicamente destacar los materiales conductores. Son aquellos que presentan una baja resistividad. Se puede apreciar una horquilla de valores muy amplia entre los aislantes y los conductores.

El parámetro “L”, representa la longitud del cable conductor $[m]$. La longitud del cable será muy importante, y se le aplicarán parámetros de corrección cuando el cable conductor este compuesto por más de 1 hilo, ya que el trenzado que presentan los cables, aumentará su longitud.

Por último, A, representa el área de la sección transversal del cable $[m^2]$. Más adelante se verá que la sección transversal del cable es importante en otros efectos cuando se trata de una corriente AC.

Esta expresión es “básica”. A partir de ésta, empezaremos a añadir factores condicionantes de la resistencia. En primer lugar añadiremos el efecto de la temperatura.

$$R_0 = \frac{\rho_{20} \cdot l}{A} [1 + \alpha(\theta - 20)] \quad (2.1.1)$$

Se trata de una expresión muy similar a la anterior, pero se ha añadido el factor $[1 + \alpha(\theta - 20)]$ donde el parámetro α , representa el coeficiente térmico del material empleado y el parámetro θ , que representa la temperatura. Podemos ver que si el

parámetro α fuese 0, es decir, no tuviese dependencia de la temperatura del material, la expresión sería la principal. Esto también sucede si la temperatura ya es 20 grados centígrados.

Tabla de coeficientes térmicos:

CONDUCTORES	RESISTIVIDAD[ρ] a 20°C en mm ² /m	Coefficiente de temperatura [α]
Cobre	$1.724138 \cdot 10^{-2}$	$3.93 \cdot 10^{-3}$
Aluminio	$2.857 \cdot 10^{-2}$	$4.07 \cdot 10^{-3}$
Plata	$1.47 \cdot 10^{-2}$	$3.8 \cdot 10^{-3}$
Hierro	$9.71 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$
Plomo	$20.65 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$
Acero	$20 \cdot 10^{-2}$	$5 \cdot 10^{-3}$
Wolframio	$5.65 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$
Oro	$2.44 \cdot 10^{-2}$	$3.4 \cdot 10^{-3}$
Platino	$10.6 \cdot 10^{-2}$	$3.93 \cdot 10^{-3}$

Tabla 2. Tabla de coeficientes térmicos. “Manual práctico electricidad ingenieros”

Como ya se ha mencionado anteriormente el número de hilos que presenta el cable conductor, hace variar la longitud efectiva del cable. Por esto, se valoran una serie de situaciones.

Conductor de 1 hilo: no necesita corrección.

Conductor de 3 hilos: aumenta un 1%, es decir, la longitud del cable deberá ser multiplicada por 1,01.

Conductor de 7 hilos: aumenta su longitud un 2%.

Conductores de 11 hilos o más: aumenta un 3%.

Todos los cálculos anteriores son aplicables para lo que se conoce como resistencia en DC. Pero en AC existen otros factores generadores de resistencia que se deben añadir a los mencionados anteriormente.

$$R_{AC} = R_{DC} \cdot (1 + y_s + y_p) \quad (2.1.2)$$

En estas páginas solo valoraremos el efecto skin y los efectos de proximidad, al considerar tubería y ductos magnéticos.

Por lo que la ecuación que modela el comportamiento de la resistencia quedará de la siguiente forma:

$$R_{AC} = R_{DC} \cdot [1 + 1,5 \cdot (y_s + y_p)] \quad (2.1.3)$$

El efecto skin viene producido por la frecuencia de la corriente donde ésta deja de utilizar toda la sección de cable disponible para concentrarse en las zonas más externas del hilo conductor, lo que supone una reducción de lo que se podría entender como sección útil.

El término Y_s se obtiene con la siguiente expresión.

$$y_s = \frac{x_s^4}{192 + 0,8 \cdot x_s^2} \quad (2.1.4)$$

Donde:

$$x_s^2 = F_k \cdot k_s \quad (2.1.5)$$

$$F_k = \frac{8 \cdot \omega \cdot f \cdot 10^{-7}}{R_{DC}} \quad (2.1.6)$$

Y el término y_p se obtiene por medio de la ecuación que se muestra a continuación

$$y_p = F_p \cdot \left(\frac{d_c}{s}\right)^2 \cdot \left[0,312 \cdot \left(\frac{d_c}{s}\right)^2 + \frac{1,18}{F_p + 0,27}\right] \quad (2.1.7)$$

$$x_p^2 = F_k \cdot k_p \quad (2.1.8)$$

$$F_p = \frac{x_p^4}{192 + 0,8 \cdot x_p^4} \quad (2.1.9)$$

Donde:

- f: frecuencia
- d_c : diámetro del conductor
- s, distancia entre conductores.
- K_s, k_p : Coeficientes que dependen de las características constructivas del conductor.
- Las expresiones anteriormente expuestas son válidas cuando se tiene la condición de que $x_p \leq 2.8$, lo que ocurre en la mayoría de los casos.

Tipo de conductor	Aislante seco e impregnado	k _s	k _p
COBRE			
Round, stranded	Si	1	1
Round, stranded	No	1	0.8
Round, compact	Si	1	1
Round, compact	No	1	0.8
Round, segmental		0.435	0.37
Sector shaped	Si	1	0.8
Sector shaped	No	1	1
ALUMINIO			
Round, stranded	Ambos	1	
Round, 4 segments	Ambos	0.28	
Round, 5 segments	Ambos	0.19	
Round, 6 segments	Ambos	0.12	

Tabla 3. Tabla de parámetros de resistencias en AC

Más adelante, se procederá a analizar cómo de buena es la ecuación que se menciona en las líneas anteriores para la emulación de valores de resistencia en corriente continua. Se analizará comparando valores de tensiones y corrientes que circulan por un circuito con valores de resistencias “reales”, extraídos de un catálogo de líneas de transmisión, con las tensiones y corrientes que se recogen con una resistencia “creada” por el usuario.

Comenzamos analizando una línea subacuática de Nexans. Sus parámetros constructivos son los siguientes:

(F)2XS(FL)2Y>c<RAA 6/10(12) kV

Constructional Data

1	2	3	4	5	6	7	8	9	10	11
Nominal cross sectional area of conductor (mm ²)	Conductor copper round stranded diameter over conductor (mm)	Insulation XLPE wall thickness (mm)	Screen copper wires and counter helix cross sectional area (mm ²)	Metallic tape aluminium wall thickness (mm)	Core sheath PE black wall thickness diameter (mm) (mm)	Bedding wall thickness (mm)	Armour steel wires round galvanized diameter (mm)	Serving bitumen fib. material incl. colour strip wall thickness (mm)	Outer diameter of cable (mm)	Cable weight (t/km)
35	7.0	3.4	16	0.2	1.6 24	2	4.0	3.5	71	7.4
50	8.2	3.4	16	0.2	1.7 25	2	4.0	3.5	73	8.0
70	9.9	3.4	16	0.2	1.7 28	2	4.0	3.5	79	9.4
95	11.5	3.4	16	0.2	1.8 30	2	4.0	3.5	83	10.5
120	13.0	3.4	16	0.2	1.8 31	2	4.0	3.5	86	11.6
150	14.5	3.4	25	0.2	1.9 33	2	4.0	3.5	89	12.9
185	16.1	3.4	25	0.2	1.9 34	2	4.0	3.5	93	14.4
240	18.6	3.4	25	0.2	2.0 37	2	4.0	3.5	98	16.6

Ilustración 2. Parámetros constructivos línea NEXANS. “Manual de líneas subacuáticas NEXANS”

Estos parámetros son importantes a la hora de diseñar físicamente la línea, ya que se deben tener en cuenta datos como el peso, la sección, los materiales, en función de las condiciones de entorno, las necesidades del usuario o los requisitos que marque la normativa. En el tema que concierne a este proyecto se tendrá en cuenta la sección nominal del cable conductor.

Hablando de la sección, se puede ver que el fabricante ofrece 8 tamaños de sección. Tendremos en cuenta esto a la hora de poder comparar simulación y cable real.

Se ha marcado la columna que debemos tener en cuenta, ya que el primer elemento que se estimará es la resistencia que presenta el cable en corriente continua, aunque no será la utilizada en las simulaciones, ya que para las simulaciones se empleará la resistencia en corriente alterna a 90°C. El cálculo de la resistencia en corriente continua es necesario ya que las ecuaciones empleadas para el cálculo de la resistencia en AC, se basa en un valor previo de la resistencia en corriente continua.

```
Seccion_Cable = [35 50 70 95 120 150 185 240]; % EN MILIMETROS CUADRADOS
```

```
Rdc_20C_NEXANS_matriz = [0.524 0.387 0.268 0.193 0.153 0.124 0.0991 0.0754]; % RESISTENCIA EN DC 20°C EN FUNCION DE LA SECCION Y LONGITUD
```

```
Resistencia_funcion_seccion = [0.67 0.49 0.34 0.25 0.20 0.16 0.13 0.10]; % RESISTENCIA EN AC 90°C EN FUNCION DE LA SECCION Y LA LONGITUD
```

(F)2XS(FL)2Y<c<RAA 6/10(12) kV

Electrical Data

1		2	3	4	5	6	7	8	9	
Nominal cross sectional area		Conductor resistance DC 20 °C	Conductor resistance AC 90 °C	Screen resistance 20 °C	Capacitance	Inductance	Current rating	Losses	1s short circuit current after full load at 90 °C conductor temperature	
conductor (mm ²)	screen (mm ²)	(Ω/km)	(Ω/km)	(Ω/km)	(μF/km)	(mH/km)	(A)	(W/m)	conductor (kA)	screen (kA)
35	16	0.524	0.67	1.15	0.24	0.43	181	67	5.0	3.3
50	16	0.387	0.49	1.15	0.27	0.41	213	69	7.1	3.3
70	16	0.268	0.34	1.15	0.33	0.39	259	71	10.0	3.3
95	16	0.193	0.25	1.15	0.36	0.37	307	73	13.6	3.3
120	16	0.153	0.20	1.15	0.39	0.36	347	75	17.1	3.3
150	25	0.124	0.16	0.73	0.42	0.35	385	77	21.4	5.1
185	25	0.0991	0.13	0.73	0.46	0.34	430	78	26.5	5.1
240	25	0.0754	0.10	0.73	0.51	0.32	491	81	34.3	5.1

Ilustración 3. Valores de impedancia por unidad de longitud. “Manual de líneas subacuáticas NEXANS”

Se han agrupado los valores necesarios en un script de Matlab, con la siguiente estructura:

Para la simulación de la resistencia en DC, se empleará la siguiente ecuación explicada en las líneas anteriores:

```
Rdc = (Resistividad_20C*Longitud/Seccion) * ( 1 + Coeficiente_Termico * (Temperatura-20));
```

Por lo que en las siguientes simulaciones se va a proceder a comparar la resistencia equivalente por unidad de longitud, en cables de la misma sección que se encuentran a la misma temperatura. Como se ve en las tablas, la resistencia en DC se ofrece a una temperatura de 20°C. Se van a emplear datos como la resistividad del conductor, en este caso cobre, a 20°C y el coeficiente térmico del cobre.

$$\rho_{20}(\text{cobre}) = 0.01786 \Omega \cdot \text{mm}^2/\text{m} \quad (2.1.10)$$

$$\alpha(\text{cobre}) = 3,93 \cdot 10^{-3} \quad (2.1.11)$$

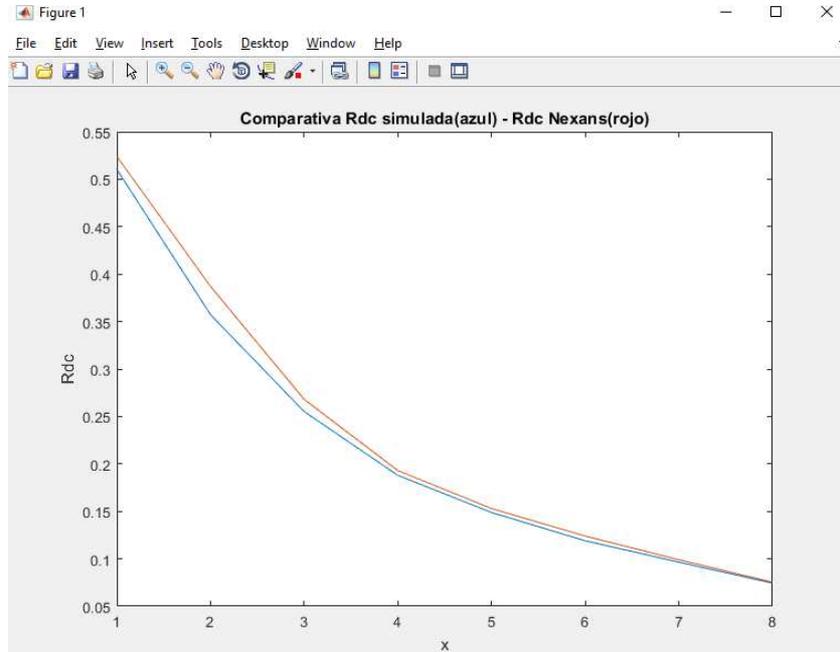


Ilustración 4. Comparativa Resistencia estimada - Resistencia NEXANS en DC a 20°C

Se puede afirmar que es una simulación bastante correcta. La resistencia en corriente continua está expresada en ohmios/km. El mayor error se comete con la sección de 50mm². Numéricamente, los errores cometidos en función de la sección son los siguientes:

	1	2	3	4	5	6	7	8
Seccion [mm²]	35	50	70	95	120	150	185	240
R_{DC} Nexans [Ω/Km]	0.5240	0.3870	0.2680	0.1930	0.1530	0.1240	0.0991	0.0754
R_{DC} estimada [Ω/Km]	0.5103	0.3572	0.2551	0.1880	0.1488	0.1191	0.0965	0.0744
Error de estimación [Ω/Km]	-0.0137	-0.0298	-0.0129	-0.0050	-0.0042	-0.0049	-0.0026	-0.0010

Tabla 4. Resultados estimación resistencia DC

Se ve que los errores no son significativos ya que se encuentran en la horquilla de valores que comprende desde [-0.0010,-0.0298]. Destacar que la resistencia estimada siempre es ligeramente menor que la real. Esto puede deberse a algún tipo de pérdidas presentes en la construcción física de la línea que en las ecuaciones no se están valorando.

Por el mismo procedimiento se puede calcular la resistencia en AC. Como ya hemos mencionado anteriormente, esta resistencia toma como base la resistencia en DC y se complementa con otros parámetros.

Estos parámetros complementarios se pueden separar en 2. Una componente complementaria es debida a la frecuencia y otros a la proximidad de otro cable conductor.

En el código empleado se encontrarán con el subíndice “K” en el caso que sean dependientes de la frecuencia, mientras que, para los elementos de proximidad, se empleará el subíndice “P”.

(F)2XS(FL)2Y>c<<RAA 6/10(12) kV

Electrical Data

1		2	3	4	5	6	7	8	9	
Nominal cross sectional area		Conductor resistance DC 20 °C	Conductor resistance AC 90 °C	Screen resistance 20 °C	Capacitance	Inductance	Current rating	Losses	1s short circuit current after full load at 90 °C conductor temperature	
conductor (mm ²)	screen (mm ²)	(Ω/km)	(Ω/km)	(Ω/km)	(μF/km)	(mH/km)	(A)	(W/m)	conductor (kA)	screen (kA)
35	16	0.524	0.67	1.15	0.24	0.43	181	67	5.0	3.3
50	16	0.387	0.49	1.15	0.27	0.41	213	69	7.1	3.3
70	16	0.268	0.34	1.15	0.33	0.39	259	71	10.0	3.3
95	16	0.193	0.25	1.15	0.36	0.37	307	73	13.6	3.3
120	16	0.153	0.20	1.15	0.39	0.36	347	75	17.1	3.3
150	25	0.124	0.16	0.73	0.42	0.35	385	77	21.4	5.1
185	25	0.0991	0.13	0.73	0.46	0.34	430	78	26.5	5.1
240	25	0.0754	0.10	0.73	0.51	0.32	491	81	34.3	5.1

Tabla 5. Valores de impedancia de la línea en función de la longitud. “Manual de líneas subacuáticas NEXANS”

En la tabla se ofrece esta resistencia a 90°C, por lo que primero se debe calcular la resistencia en DC a 90°C. También se muestran los valores en función de la sección, por lo que se seguirá una metodología similar, empleando en este caso la siguiente ecuación para la estimación de la resistencia en AC.

```

Ks = 1; %TABLAS
Fk = 8 * Frecuencia_rad_s * Frecuencia * 10^-7 / Rdc_estimada(i);
Xs = sqrt (Fk * Ks);
Ys = Xs^4 / (192 + 0.8 * Xs^2);

Radio_conductor(i) = sqrt(Seccion_Cable(i)/3.1415)/1000; % Seccion en mm por lo tanto radio en mm
Diametro_conductor(i) = 2 * Radio_conductor(i)/1000; % en metros

Kp = 1; %TABLAS TFG
Xp = sqrt (Fk * Kp);
Fp = Xp^4 / (192 + 0.8 * Xp^4);
Yp = Fp * (Diametro_conductor(i)/Distancia_entre_conductores)^2 * (0.312 * (Diametro_conductor(i)/Distancia_entre_conductores)^2 + 1.18 / (Fp + 0.27));

Rac_estimada = Rdc_estimada * ( 1 + 1.5*(Ys+Yp));

```

Comparativa entre los valores de las tablas de NEXANS y un cable simulado valorando los parámetros teóricos mencionados durante el desarrollo:

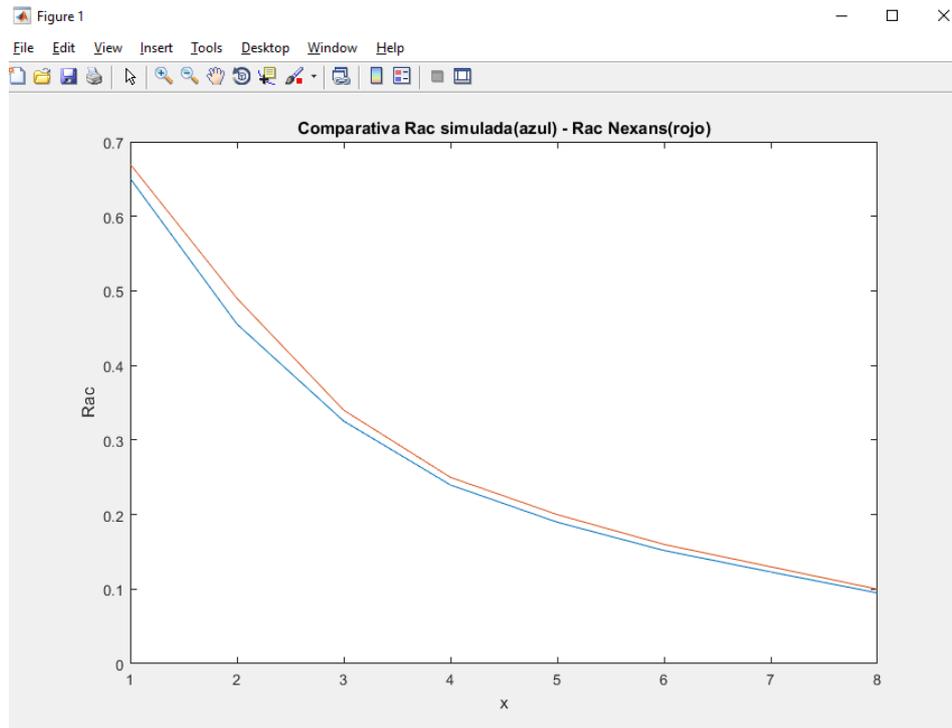


Ilustración 5. Comparación resistencia estimada - Resistencia NEXANS en AC 90°C

En este caso el error de estimación es el siguiente:

	1	2	3	4	5	6	7	8
Sección [mm²]	35	50	70	95	120	150	185	240
R_{AC} Nexans [Ω/Km]	0.6700	0.4900	0.3400	0.2500	0.2000	0.1600	0.1300	0.1000
R_{AC} estimada [Ω/Km]	0.6508	0.4555	0.3254	0.2398	0.1898	0.1518	0.1231	0.0949
Error de estimación [Ω/Km]	-0.0193	-0.0345	-0.0147	-0.0103	-0.0102	-0.0082	-0.0069	-0.0051

Tabla 6. Resultados de la estimación R_{AC}.

Igual que en el caso de la resistencia en corriente continua, el mínimo error se produce para la sección máxima de 240mm². Son valores muy pequeños, por lo que se puede afirmar que se está realizando una buena estimación de la parte resistiva de la línea de transmisión.

Para determinar los parámetros en alterna de la línea de transmisión es necesario saber cómo se comporta el campo eléctrico en el cable.

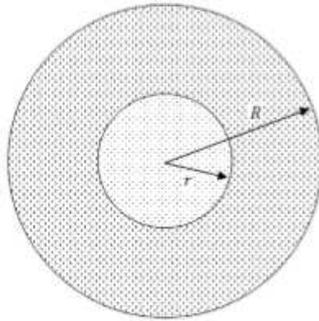


Ilustración 6. Estructura de cable

Partiendo de una sección transversal como la de la imagen, entendemos como conductor el radio “r” y como aislante la diferencia de radios “R-r”.

Tomando V_0 la tensión de la fase respecto al neutro, se puede expresar el campo eléctrico como:

$$E = \frac{V_0}{x \cdot \ln \frac{R}{r}} \quad (2.1.12)$$

Tomando como punto “x” uno perteneciente a la distancia entre r y R.

Para obtener una intensidad mínima del campo eléctrico máximo, se realiza:

$$\frac{d}{dx} \left(r \cdot \ln \frac{R}{r} \right) = 0 \quad (2.1.13)$$

Donde se obtiene que la relación entre R y r es $R/r = e$.

$$\Delta = R - r = r \cdot \left[e^{\frac{V_0}{r \cdot E}} - 1 \right] \quad (2.1.14)$$

Considerando esta relación entre radios obtenemos lo siguiente

$$R - r = 1,718 \cdot r \quad (2.1.15)$$

Por lo que el espesor de aislante siempre será proporcional a la sección del hilo conductor.

2.2 Inductancia en cables

Con lo mencionado en las líneas anteriores, se podrá realizar una buena estimación de las inductancias y capacidades presentes en el cable.

La inductancia de una línea de transmisión se calcula como el flujo enlazado por amperio. Siendo la permeabilidad μ constante, una corriente sinusoidal produce una variación sinusoidal de flujo en fase con la corriente.

$$L = \frac{\lambda}{I} \quad (2.2.0)$$

Hablando de la inductancia los parámetros que vamos a tener en cuenta son el número de líneas, el tamaño de los conductores, y la distancia entre ellos.

Inductancia debida al flujo interno por unidad de longitud:

$$L_{int} = \frac{1}{2} 10^{-7} \left[\frac{H}{m} \right] \quad (2.2.1)$$

La inductancia que aparece por la interacción de 2 cables:

$$L_{12} = 2 \times 10^{-7} \ln \left(\frac{D_2}{D_1} \right) \left[\frac{H}{m} \right] \quad (2.2.2)$$

Por lo que la suma de ambas se puede expresar como:

$$L_1 = \left(\frac{1}{2} + 2 \ln \left(\frac{D}{r_1} \right) \right) \times 10^{-7} \left[\frac{H}{m} \right] \quad (2.2.3)$$

Donde "D", representa la distancia entre cables y "r1" el radio de la línea a calcular su inductancia.

Esta expresión es más favorable escrita de la siguiente forma:

$$L_1 = 2 \times 10^{-7} \left(\ln(\varepsilon^{-\frac{1}{4}}) + \ln \left(\frac{D}{r_1} \right) \right) \left[\frac{H}{m} \right] \quad (2.2.4)$$

$$L_1 = 2 \times 10^{-7} \ln \left(\frac{D}{r_1 \cdot \varepsilon^{\frac{1}{4}}} \right) \left[\frac{H}{m} \right] \quad (2.2.5)$$

Donde se pasa a llamar:

$$r'_1 = r_1 \cdot \varepsilon^{-\frac{1}{4}} \quad (2.2.6)$$

$$L_1 = 2 \times 10^{-7} \ln \left(\frac{D}{r'_1} \right) \left[\frac{H}{m} \right] \quad (2.2.7)$$

Esta misma será la expresión para la línea 2:

$$L_1 = 2 \times 10^{-7} \ln \left(\frac{D}{r_2'} \right) \left[\frac{H}{m} \right] \quad (2.2.8)$$

$$L = L_1 + L_2 \quad (2.2.9)$$

Por lo tanto, la inductancia completa que simula una línea, se puede emular mediante la suma de 2 inductancias simples. Como se ha mencionado, se podrá comprobar como de acertada es esta simulación comparando tensiones y corrientes en el circuito con una inductancia en la que el usuario crea la línea de transmisión y sus condiciones de entorno, respecto a una línea de transmisión real de la que se conocen sus parámetros.

Pasando a valorar la estimación de los parámetros en corriente continua, se trabajará con una metodología similar a la empleada con las resistencias, tanto en continua como en alterna.

Comenzando por la inductancia, se recuerda la ecuación mencionada anteriormente con la que se trata de emular una línea:

$$L_1 = \left(\frac{1}{2} + 2 \ln \left(\frac{D}{r_1} \right) \right) \times 10^{-7} \left[\frac{H}{m} \right] \quad (2.2.10)$$

Mediante diversos ensayos, se ha determinado que la distancia a la que se considera que debe encontrarse un cable de otro en la línea de transmisión trifásica es a 4cm. El radio del conductor se determina de forma muy simple conociendo la sección.

Con estas bases, se determina el error de estimación:

	1	2	3	4	5	6	7	8
Sección [mm²]	35	50	70	95	120	150	185	240
L_{Nexans} [mH/Km]	0.4300	0.4100	0.3800	0.3700	0.3500	0.3400	0.3300	0.3200
L_{estimada} [mH/Km]	0.4581	0.4224	0.3888	0.3582	0.3349	0.3126	0.2916	0.2656
Error de estimación [mH/Km]	0.0280	0.0124	0.0087	-0.0117	-0.0151	-0.0274	-0.0384	-0.054

Tabla 7. Resultados estimación inductancia

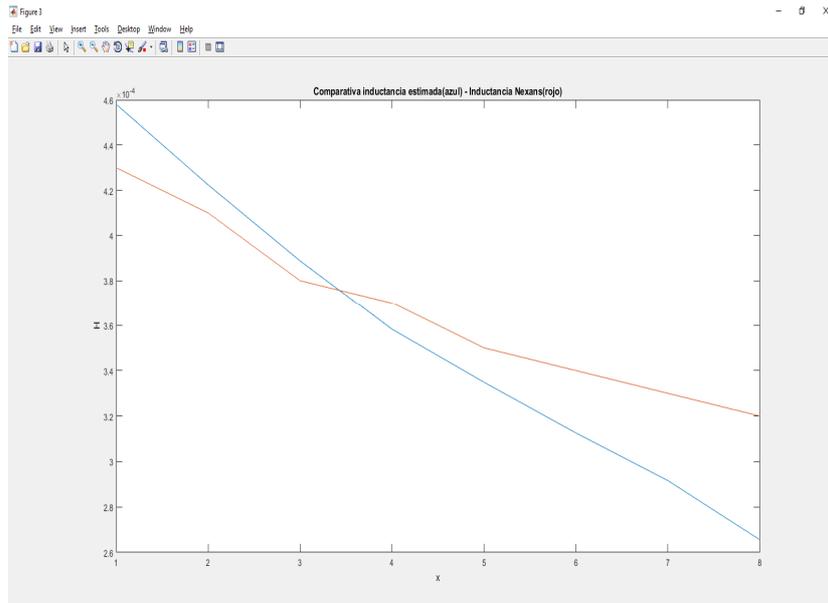


Ilustración 7. Diagrama de inductancia estimada- Inductancia NEXANS

En este caso los resultados obtenidos no son tan buenos como los mostrados anteriormente en la resistencia en corriente alterna.

Se ve que el error máximo se produce cuando la sección del cable es la más grande posible. Esto tiene sentido ya que como se ha mencionado, la inductancia presente en una línea de transmisión tiene una dependencia de la propia línea y de líneas externas, por lo que el grosor y el aislante que está presente, son muy importantes. Pese a esto, los resultados obtenidos son válidos para realizar una primera estimación de la inductancia presente.

2.3 Capacidad en cables

Por último, para poder definir de forma completa una línea de transmisión, se necesita modular mediante una ecuación la capacidad presente en estas líneas.

La capacidad entre los conductores de una línea de 2 hilos es definida como:

$$C = \frac{q}{v} [F/m] \quad (2.3.0)$$

Donde “q” es la carga de la línea expresada en culombios por metro [C/m] y “v” es la diferencia de potencial entre los conductores en voltios.

$$C = \frac{2 \cdot \pi \cdot \epsilon_r \cdot \epsilon_0}{\ln\left(\frac{D_{eq}}{r}\right)} \quad (2.3.1)$$

Analizando los parámetros presentes en la ecuación anterior, vemos que la capacidad de la línea depende de otros parámetros diferentes a la inductancia. En este caso, la permitividad del medio tiene mucha importancia. Los parámetros ϵ_0 y ϵ_r se corresponden con la permitividad del vacío y permitividad del medio respectivamente. Este producto se puede expresar como:

$$\epsilon = \epsilon_r \cdot \epsilon_0 \quad (2.3.2)$$

Para poder conocer la capacidad de una línea se necesita saber en qué medio se va a encontrar envuelta. Se adjunta una tabla de valores típicos de materiales o medios de entorno:

Material	ϵ (pF/m)	Material	ϵ (pF/m)
Aceite mineral	19,5	Caucho	de 20 a 50
Acetona	191	Madera	de 10 a 60
Aire	8,84	Papel duro	49,5
Agua destilada	81	PVC	de 30 a 40
Baquelita	de 50 a 80	Vidrio	de 40 a 60

Tabla 8. Valores permitividad. “Constantes físicas y tablas”

También se debe conocer la permitividad en el vacío. El valor de esta constante es:

$$\varepsilon_0 = 8.85 \cdot 10^{-12} \left[\frac{F}{m} \right] = 8.85 \cdot 10^{-3} \left[\frac{\mu F}{Km} \right] \quad (2.3.3)$$

Por último, se comprueba la fidelidad con la que se está estimando el valor de la capacidad de la línea de transmisión. Se recuerda que la ecuación empleada para estimar un valor de la capacidad es la siguiente:

$$C = \frac{2 \cdot \pi \cdot \varepsilon}{\ln\left(\frac{D_{eq}}{r}\right)} \quad (2.3.4)$$

Mediante simulaciones, se ha podido comprobar que el valor de " ε " que presentan estos conductores está en torno a los 72 pF/m, como se puede observar en la siguiente gráfica (ilustración 8).

Esto depende tanto del aislante del conductor como del cable como del entorno en el que este se encuentra.

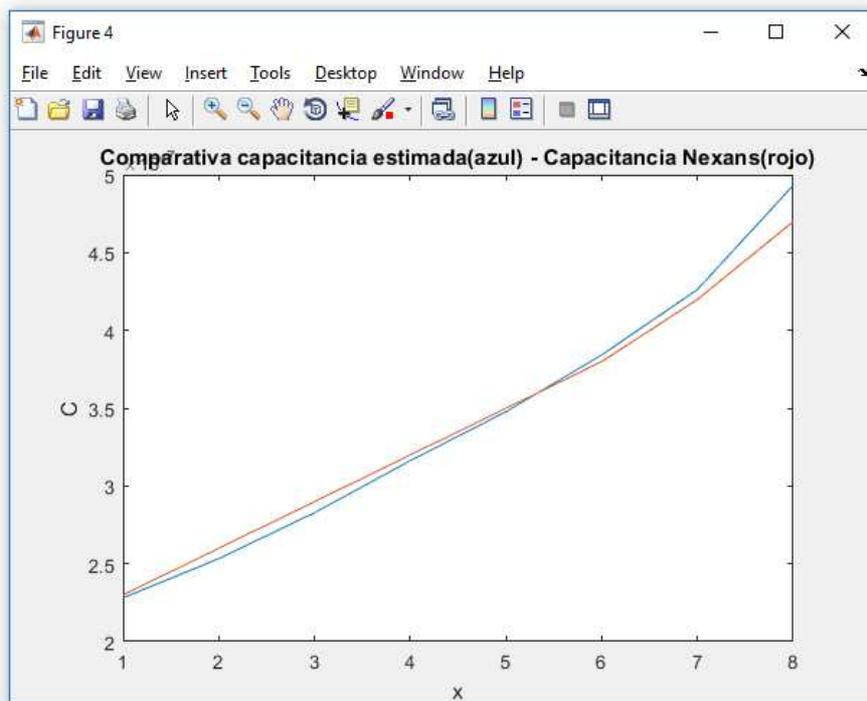


Ilustración 8. Comparación Capacidad estimada - Capacidad NEXANS

	1	2	3	4	5	6	7	8
Sección [mm²]	35	50	70	95	120	150	185	240
C^{Nexans} [μF/Km]	0.2400	0.2700	0.3300	0.3600	0.3900	0.4200	0.4600	0.5100
C^{estimada} [μF/Km]	0.2527	0.2806	0.3133	0.3504	0.3852	0.4257	0.4723	0.5465
Error de estimación [mF/Km]	0.0127	0.0106	-0.0166	-0.0096	-0.0047	0.0056	0.0122	0.0365

Tabla 9. Resultados estimación capacidad

2.4 Estructuras en π

A partir de todos los componentes calculados en los apéndices anteriores, se va a emplear una estructura en π , para la emulación de la línea de transmisión.

El circuito equivalente que se emplea para la simulación de líneas de transmisión es el siguiente:

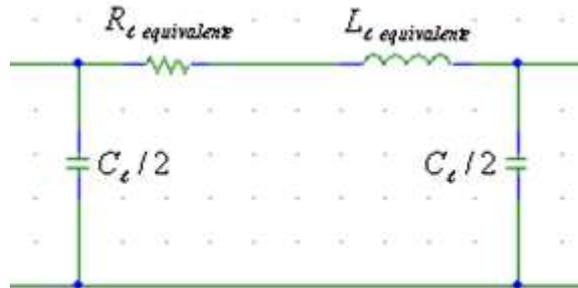


Ilustración 9. Esquema estructura en π

Su nombre viene dado por el parecido que presenta la distribución de los componentes con el propio símbolo.

Estas estructuras se pueden enlazar con otras idénticas, consiguiendo tener acceso a puntos intermedios de interés para el usuario que analiza la línea.

Más adelante se tratarán las diferentes líneas de transmisión que se pueden diferenciar en función de su longitud. Con las simulaciones que se llevarán a cabo, se tratará de comprobar las ecuaciones del telegrafista, que definen el comportamiento de corrientes y tensiones a lo largo de la línea en función de su longitud.

A continuación, se va a realizar una comparativa de datos reales de líneas de transmisión, aéreas y subacuáticas, gracias a la información que se ha podido extraer de las tablas mostradas por NEXANS. Con esto, se puede hacer una idea de cómo de aproximada es la emulación de líneas de transmisión que se está realizando y la diferencia que existe entre líneas subacuáticas y líneas aéreas.

2.5 Ecuaciones del telegrafista

Después de analizar los parámetros que van a afectar a las líneas de transmisión, se pasará a unir estos componentes, para observar el comportamiento de las corrientes y tensiones.

Para hablar de las corrientes y voltajes, primero se sentarán unas bases de cómo se suelen abordar este tiempo de problemas.

Primero de todo, se suele trabajar con parámetros agrupados, lo que significa que se considera que la componente resistiva, capacitiva e inductiva son uniformes a lo largo de toda la línea. Esto suficiente para las líneas que entendemos como cortas y medias.

Para una mayor precisión en la aproximación de las corrientes y voltajes, se pasa a trabajar con parámetros distribuidos, hablando de valores de resistencia, inductancia y capacidad por unidad de longitud.

Más adelante se mostrarán cuáles son los circuitos equivalentes que representan cada una de estas líneas.

A finales de 2016, se cuantificó que la suma de kilómetros de las líneas de transmisión eléctrica había superado los 43.000Km de longitud, siendo en su mayoría líneas de transmisión aéreas.

Red de transporte peninsular y no peninsular de Red Eléctrica (Datos acumulados a 31 de diciembre del 2016)

Km de circuito	2012	2013	2014	2015	2016
400kV	20.109	20.639	21.094	21.184	21.620
220kV	18.779	19.053	19.192	19.386	19.496
150 - 132 - 110kV	272	272	272	398	523
< 110kV	2.014	2.014	2.014	2.022	2.025
Total	41.174	41.978	42.572	42.989	43.664

Tabla 10. Datos de la Red Eléctrica Española

Para sentar unas bases de referencia, se podría decir que hablamos de líneas cortas cuando éstas no llegan a los 80km. Líneas medias cuando se encuentran entre 80km y 240km y por último hablaremos de líneas de transmisión largas cuando estas superan los 240km. Hay que decir que, para determinados cálculos, se puede trabajar en una línea de transmisión larga con parámetros agrupados.

Esta distribución está basada para las líneas aéreas, pero en las simulaciones se podrá observar que, usando un cable de transmisión subacuático, las distancias se reducen.

a. Líneas de transmisión cortas.

En este tipo de líneas, solo se tendrán en cuenta la componente resistiva e inductiva. En líneas de mayor longitud, habrá que tener en cuenta sus capacidades.

Se trata de un circuito muy simple, con una fuente generadora, una carga, y la línea de transmisión, que será representada con una impedancia $Z = R + j\omega L$

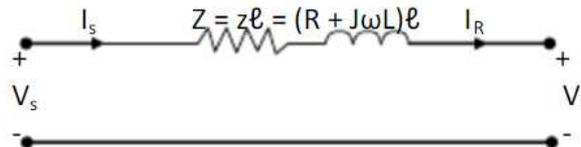


Ilustración 10. Circuito equivalente línea corta

Se puede separar el análisis en voltajes y corrientes aportados (fuentes generadoras), y voltajes y corrientes consumidos (carga, línea de transmisión).

Atendiendo a la figura, podemos ver que el generador tiene unas bornas una tensión de V_s , entregando una corriente I_s . La carga recibe una tensión V_r con un corriente I_r . En una situación ideal debería coincidir la tensión generada con la consumida, pero no es así ya que hay pérdida de voltaje en la línea. Las corrientes sí que son coincidentes $I_s = I_r$, por lo que el voltaje que cae en la línea será $I_r * Z$, el voltaje que cae en la carga será $I_r * \text{Carga}$ y la suma de éstos debe coincidir con la tensión generadora V_s .

Tomando como base los valores que proporciona la compañía NEXANS, se va a proceder a emular el funcionamiento de una línea corta.

En las siguientes líneas se adjunta el esquema empleado para la emulación, los parámetros pertenecientes a la RLC de la línea, la RLC de la carga, tensiones e intensidades medidas...

El esquema empleado es el siguiente:

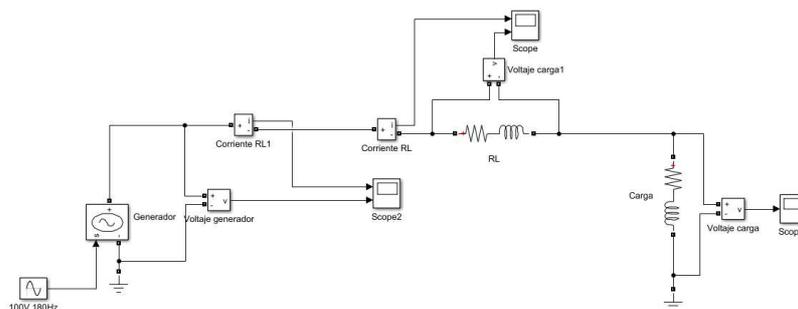


Ilustración 11. Esquema línea corta en Simulink

Se van a simular distintas líneas de transmisión de longitud corta. Se variará la longitud, la sección del cable y la magnitud de la carga. La metodología será similar para ilustrar todas las líneas y los parámetros sobre los que se tiene libertad a la hora de diseñar.

La tensión de línea que se va a valorar en las simulaciones es de 11000V a una frecuencia de 50Hz.

Se mostrarán las tensiones presentes en el generador y en el punto que se encuentra la carga. Así se podrán ver las pérdidas y el desfase a lo largo de la línea.

Antes de mostrar las capturas de las simulaciones se adjuntará una tabla que expondrá los datos de la línea: longitud, sección, resistencia, inductancia y valores de la carga.

Después, se hará una pequeña valoración de cada línea, mostrando la tensión perdida y desfase.

En las simulaciones a continuación (figura 12), se ve que en las líneas cortas no es posibles distinguir caidas de tensión considerables, ni con un segmento ni con tres.

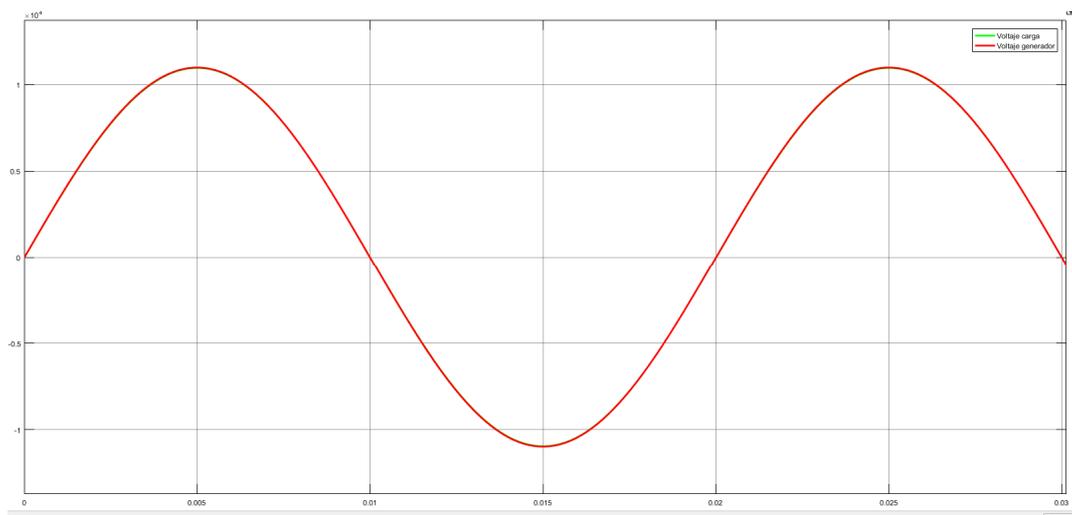


Ilustración 12. Simulación 1 segmento línea corta

Pese a no apreciarse en la simulación, la línea está dividida en 3 segmentos, en los que no hay caídas de tensión significativas.

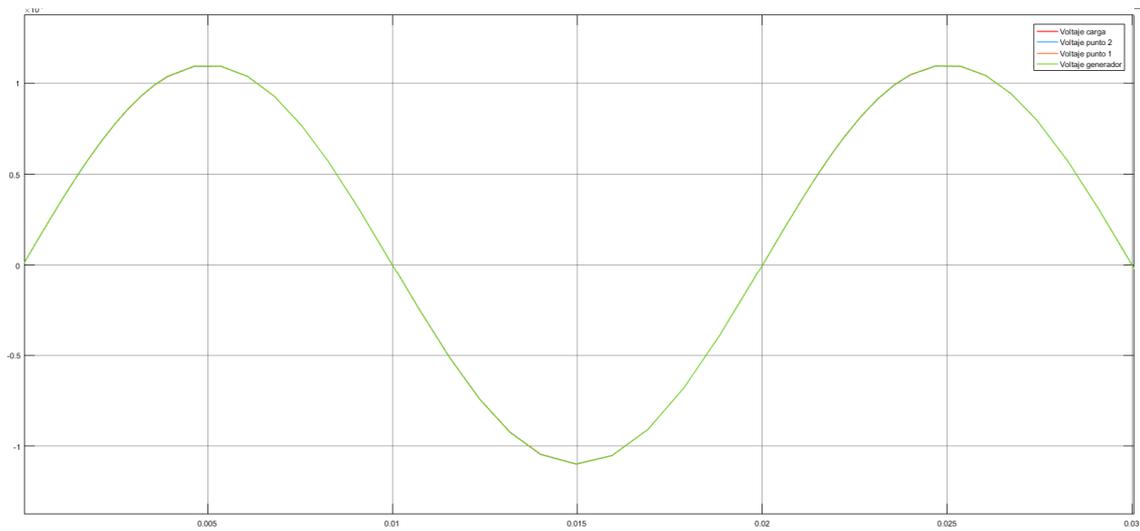


Ilustración 13. Simulación 3 segmentos línea corta

b. Línea de transmisión media.

Pasando a las líneas de transmisión de media distancia, se empieza a valorar la capacidad mencionada en las líneas anteriores. Como podemos ver en la figura, esta se divide en 2, formando una estructura en π .

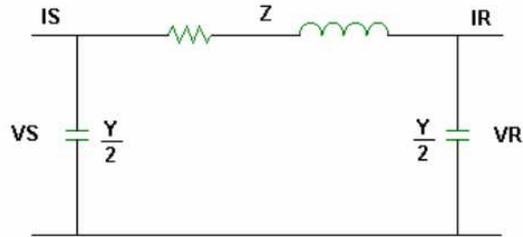


Ilustración 14. Segmento esquemático diagrama en pi

En MATLAB:

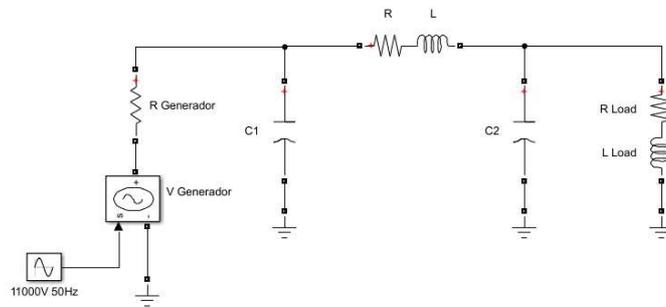


Ilustración 15. Esquema en MATLAB 1 segmento en pi

En este caso, se cree mejor emplear un esquema que permita observar y tener datos de varios puntos de la línea de transmisión por lo que el esquema empleado será el siguiente.

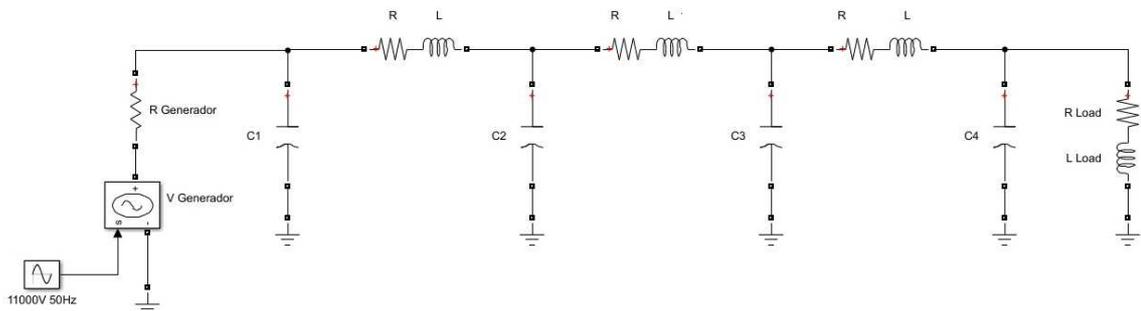


Ilustración 16. Esquema en MATLAB 3 segmentos en pi

Al tratar los parámetros de la línea como distribuidos, se considera que esta es uniforme en su composición a lo largo de ella. Por esto, se puede calcular la resistencia equivalente de la línea total, y dividirla en el número de segmentos que se desee. Cuantas más estructuras en π se coloquen en serie, más complejo será el circuito, pero se conocerá con mayor precisión que está sucediendo en la línea. Esto cobra mayor importancia en las líneas de transmisión de longitud larga, pero también será empleado para analizar líneas medias.

La metodología y estructura de simulación que se seguirá, será similar a la empleada en las líneas de corta distancia.

Se van a mostrar posibles figuras de las tensiones presentes en una línea media.

Línea media con 1 solo segmento. Comienza a haber algún tipo de diferencias entre generador y carga.

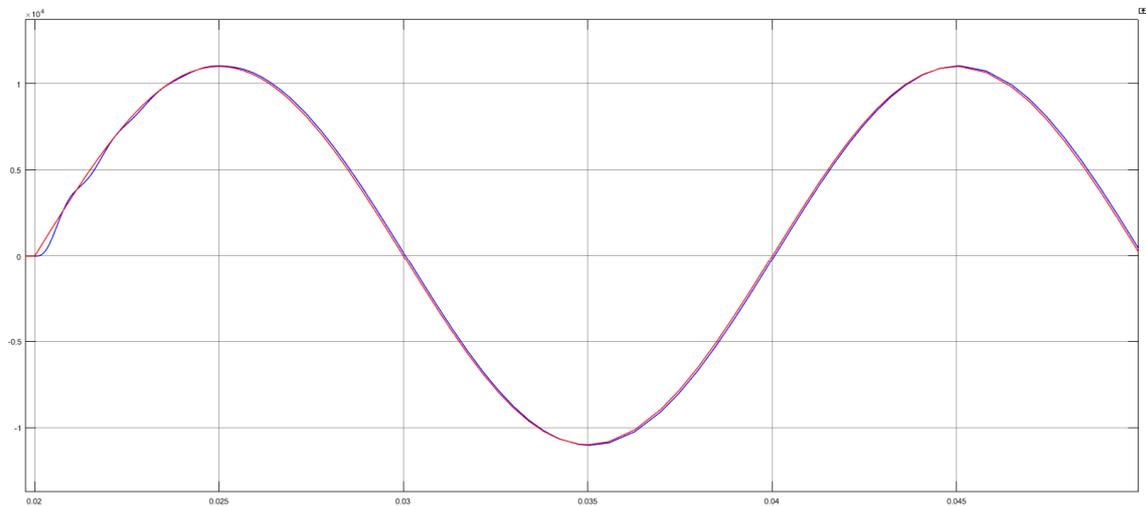


Ilustración 17. Simulación 1 segmento línea media

Línea media con 3 segmentos.

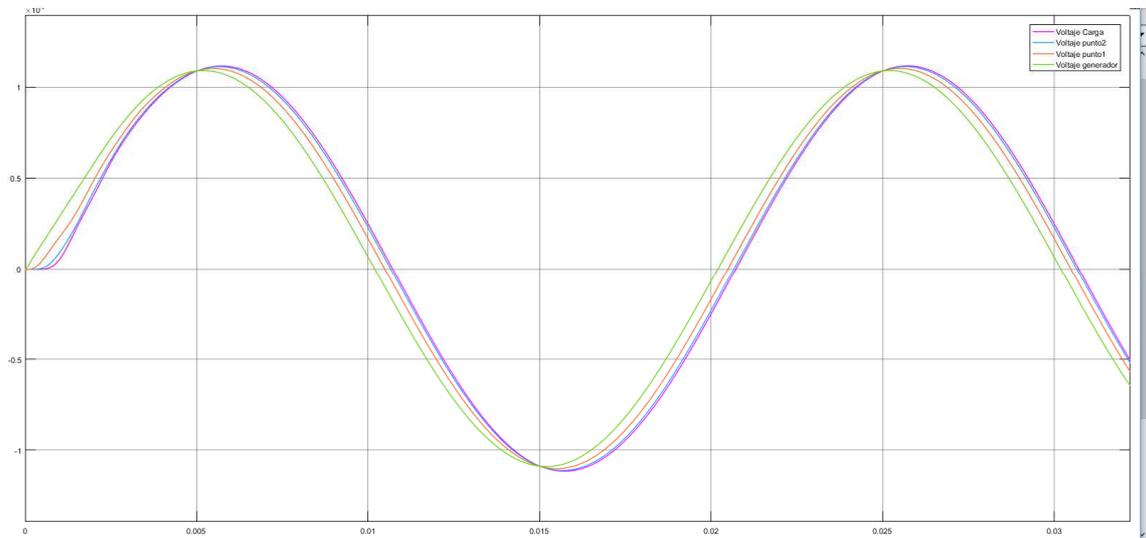


Ilustración 18. Simulación 3 segmentos línea media

c. Línea de transmisión larga.

La solución ideal para cualquier línea de transmisión se consigue considerando los parámetros distribuidos uniformemente a lo largo de la línea de transmisión, incluido para las líneas que hemos mencionado anteriormente. Pero se suelen tomar como parámetros agrupados debido a la simplicidad de los cálculos.

El esquema de una línea de transmisión larga es el siguiente:

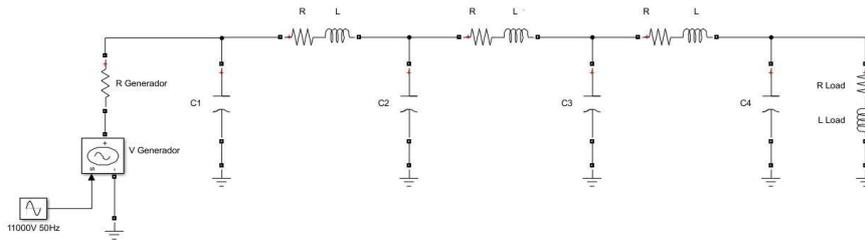


Ilustración 19. Esquema línea de transmisión larga 3 segmentos

Al tratarse de parámetros uniformemente distribuidos, las ecuaciones cambian notablemente.

Atendiendo al esquema mostrado, podemos ver que la corriente es $(I + I \cdot dl)/2$ por lo tanto:

$$dV = \frac{I + I + dI}{2} \cdot z \cdot dx \quad (2.5.1)$$

$$dI = \frac{V + V + dV}{2} \cdot y \cdot dx \quad (2.5.2)$$

Donde la “z” representa la impedancia de la línea por unidad de longitud y la “y” representa la admitancia de la línea por unidad de longitud.

Ambas se pueden reescribir como:

$$dV = I \cdot z \cdot dx \quad (2.5.3)$$

$$\frac{dV}{dx} = I \cdot z \quad (2.5.4)$$

$$dI = V \cdot y \cdot dx \quad (2.5.5)$$

$$\frac{dI}{dx} = V \cdot y \quad (2.5.6)$$

Si se deriva otra vez respecto de la distancia “x” obtenemos las ecuaciones.

$$\frac{d^2V}{dx^2} = z \cdot \frac{dI}{dx} \quad (2.5.7)$$

$$\frac{d^2I}{dx^2} = y \cdot \frac{dV}{dx} \quad (2.5.8)$$

Como podemos ver, en estas 2 últimas ecuaciones parecen los términos dI/dx y dV/dx , que anteriormente ya estaban despejadas, por lo que podemos componer 2 ecuaciones más completas.

$$\frac{d^2V}{dx^2} = z \cdot V \cdot y \quad (2.5.9)$$

$$\frac{d^2I}{dx^2} = y \cdot I \cdot z \quad (2.5.10)$$

Estas ecuaciones presentan una solución exponencial de la siguiente forma:

$$V = A_1 \cdot \varepsilon^{\sqrt{yz} \cdot x} - A_2 \cdot \varepsilon^{-\sqrt{yz} \cdot x} \quad (2.5.11)$$

$$\frac{d^2V}{dx^2} = z \cdot y \cdot [A_1 \cdot \varepsilon^{\sqrt{yz} \cdot x} - A_2 \cdot \varepsilon^{-\sqrt{yz} \cdot x}] \quad (2.5.12)$$

Por otro lado, en la ecuación de la corriente, el resultado obtenido es el siguiente:

$$I = \frac{1}{\sqrt{z}} \cdot A_1 \cdot \varepsilon^{\sqrt{yz} \cdot x} - \frac{1}{\sqrt{z}} \cdot A_2 \cdot \varepsilon^{-\sqrt{yz} \cdot x} \quad (2.5.13)$$

Solo falta calcular las constantes A_1 y A_2 , para las que se debe emplear algún punto de la línea en el cual se conozca su voltaje o corriente.

Un punto claro es para $x = 0$, es decir, el usuario conoce la tensión exacta a la salida del generador, que será V_R y la corriente será I_R

$$V \text{ para } x = 0; V = A_1 + A_2 = V_R \quad (2.5.14)$$

Calculando la corriente en el punto de $x = 0$, se obtiene lo siguiente:

$$I \text{ para } x = 0; I = \frac{1}{\sqrt{\frac{Z}{y}}}(A_1 - A_2) = I_R \quad (2.5.15)$$

De las dos últimas ecuaciones obtenemos los valores de A_1 y A_2 .

$$A_1 = \frac{V_R + I_R \cdot Z_C}{2}; \quad (2.5.16)$$

$$\text{Tomando } Z_c = \sqrt{\frac{Z}{y}} \quad (2.5.17)$$

$$A_2 = \frac{V_R - I_R \cdot Z_c}{2} \quad (2.5.18)$$

Por último, se pueden escribir unas ecuaciones únicas de tensión y corriente en función de V_R , I_R y la impedancia de línea como constantes, dependientes únicamente del punto de la línea del que se trate.

Para simplificar la ecuación, se expresa como:

$$\gamma = \sqrt{Z \cdot y} \quad (2.5.19)$$

$$V = \frac{V_R + I_R \cdot Z_c}{2} \cdot \varepsilon^{\gamma \cdot x} + \frac{V_R - I_R \cdot Z_c}{2} \cdot \varepsilon^{-\gamma \cdot x} \quad (2.5.20)$$

$$I = \frac{\frac{V_R}{Z_c} + I_R}{2} \cdot \varepsilon^{\gamma \cdot x} - \frac{\frac{V_R}{Z_c} - I_R}{2} \cdot \varepsilon^{-\gamma \cdot x} \quad (2.5.21)$$

Después de esta tediosa base matemática, se va a tratar de dar una interpretación física que simplifique el entendimiento de las líneas anteriores.

$$\gamma = \alpha + \beta j \quad (2.5.22)$$

Se trata de un número complejo, donde *Alpha* es la constante de atenuación y *Betta* es la constante de fase.

Se va a analizar el primer término del voltaje, para ver y explicar cómo afecta cada parámetro. Esto será extrapolable al resto.

$$\gamma = \alpha + \beta j \quad (2.5.22)$$

$$\begin{aligned} V^+ &= \frac{V_R + I_R \cdot Z_c}{2} \cdot \varepsilon^{\gamma \cdot x} = \frac{V_R + I_R \cdot Z_c}{2} \cdot \varepsilon^{(\alpha + \beta j) \cdot x} \\ &= \frac{V_R + I_R \cdot Z_c}{2} \cdot \varepsilon^{\alpha x} \cdot \varepsilon^{\beta j x} \end{aligned}$$

Se puede ver que la primera parte es constante, ya que no es dependiente de x .

El término central, es el encargado de modular la amplitud en función del valor de x , es decir, en función de la posición de la línea de transmisión que se quiera analizar.

Por último, se tiene una exponencial de exponente complejo, que se puede reescribir de la siguiente manera:

$$\varepsilon^{\beta j x} = \cos(\beta x) + \text{sen}(\beta x) \cdot j \quad (2.5.23)$$

Analizando la ecuación vemos que únicamente va a variar la fase, ya que su módulo es siempre igual a 1.

Volviendo al conjunto de la ecuación que define el voltaje, se puede interpretar como el primer término contribuye de forma positiva al módulo y avanza en fase. Esto es conocido como *voltaje incidente*.

Por el contrario, el segundo término, disminuye la magnitud de la onda y realiza un retardo de la fase. Esto es conocido como *voltaje reflejado*.

Resumiendo, se puede entender que el voltaje avanza de forma positiva desde la salida del generador, pero una vez llega a la carga comienza a reflejarse, por lo que el voltaje en la línea se interpreta como la superposición de 2 ondas. La que avanza en sentido positivo (generador->carga) y en sentido negativo (carga->generador).

En las líneas anteriores se afirma que tanto la tensión como la corriente en un punto de la línea están compuesta por la suma de dos componentes, la onda incidente y la reflejada. Parece lógico pensar que la velocidad de propagación es importante, y esta dependerá del medio.

La longitud de onda es:

$$\lambda = \frac{2 \cdot \pi}{\beta} \quad (2.5.24)$$

Donde β es la variación de fase por unidad de longitud en rad/m.

La velocidad de propagación es el producto de la longitud de onda por la frecuencia, por lo que se llega a la siguiente expresión:

$$\lambda = \frac{2 \cdot \pi \cdot f}{\beta} \quad (2.5.25)$$

Para una línea sin pérdidas β es:

$$\beta = \frac{2 \cdot \pi \cdot f \cdot \sqrt{L \cdot C}}{l} \quad (2.5.26)$$

$$\lambda = \frac{l}{f \cdot \sqrt{L \cdot C}} \quad (2.5.27)$$

Por lo tanto, la velocidad será:

$$Velocidad = \frac{l}{\sqrt{L \cdot C}} \quad (2.5.28)$$

El esquema empleado para las emulaciones será el siguiente:

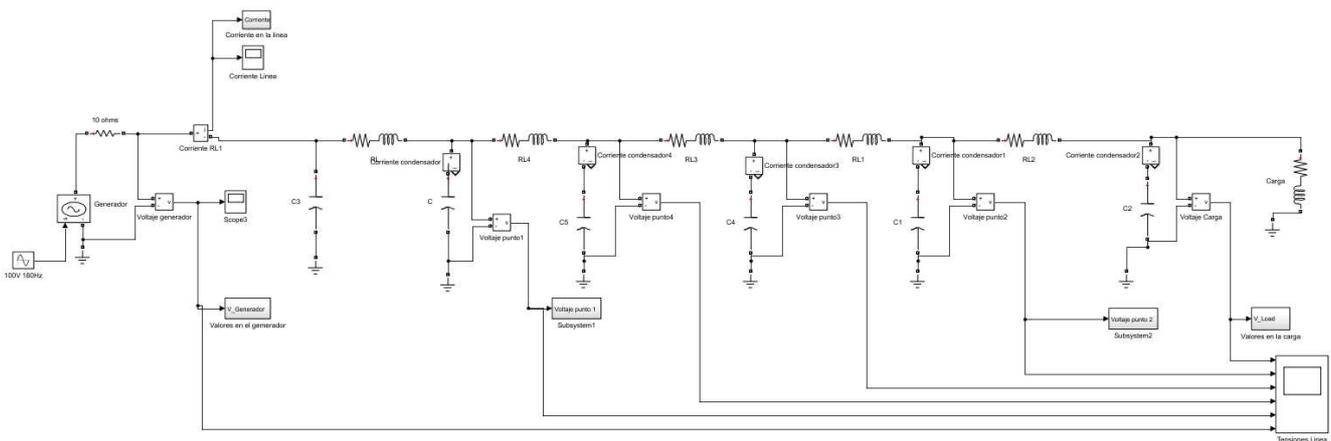


Ilustración 20. Esquema línea larga 5 segmentos

Al tratarse de una línea larga, se ha decidido tomar un esquema con 5 segmentos a nivel de simulación, lo que a nivel de implementación digital tendría un alto grado de complejidad, pero ayuda a hacer una idea de lo que sucede a lo largo de la línea.

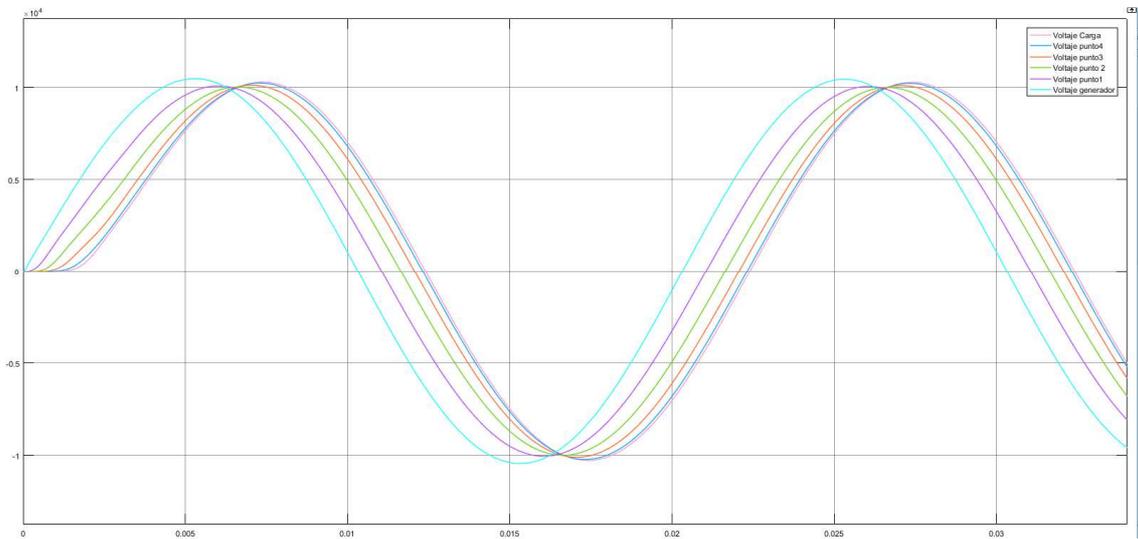


Ilustración 21. Simulación línea larga 5 segmentos

2.6 Sistema en variables de estado.

A continuación, se van a exponer distintos sistemas en variables de estado. En primer lugar, se valorará que la carga sea conocida. En segundo lugar, no será conocida, por lo que se deberá realizar mediante superposición. Se calculará el sistema en VVEE con el sistema en vacío y otro sistema en VVEE con el generador de tensión desconectado, colocando un generador de corriente en el lugar de la carga.

2.6.1 Conocida la carga.

Para poder volcar los parámetros de la línea de transmisión al microprocesador, es necesario convertir el sistema a variables de estado. Hasta ahora, se han mostrado y simulado las líneas en el entorno *Simulink* de MATLAB, pero esto carece de utilidad para trabajar con el F28377D de *Texas Instruments*. Por esto, en las siguientes líneas se va a explicar el paso a variables de estado de la estructura en π que empleamos para modelar las líneas.

Se comenzará mostrando un diagrama en π de un único segmento. Después, se ampliará a 3 segmentos y por último se mostrarán las matrices en función de los N segmentos que se deseen, dependiendo de las necesidades y capacidades de procesamiento de cada usuario.

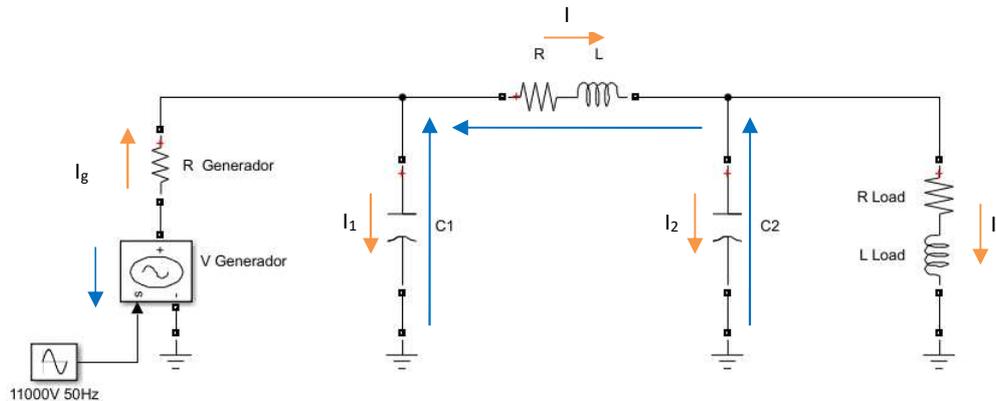


Ilustración 22. Esquema diagrama en π 1 segmento para VVEE

Las ecuaciones de un sistema en variables de estado, permiten a través de los estados definidos y las entradas del sistema, obtener la derivada de los estados y la salida del sistema. En estas líneas no se profundizará en la base teoría de los sistemas en variables de estado por lo que se adjuntará en los anexos.

$$\dot{x} = Ax + Bu \quad (2.6.0)$$

$$y = Cx + Du \quad (2.6.1)$$

En el caso que nos ocupa, se deben calcular las matrices A, B, C y D. Se trata de ecuaciones matriciales por lo que las dimensiones son muy importantes. La matriz de

estados estará formada por las corrientes en las bobinas y las tensiones en los condensadores. Como salida se tomará la tensión en la carga.

Analizando el circuito las ecuaciones eléctricas son las siguientes:

$$V_{generador} = I_g * R_g + I * R + \dot{I} * L + I_L * R_L + \dot{I}_L * L_L \quad (2.6.2)$$

La matriz de estados será:

$$x = \begin{pmatrix} I \\ I_L \\ V_{c1} \\ V_{c2} \end{pmatrix} \quad (2.6.3)$$

Y la matriz de entrada es unitaria, siendo la entrada $u = V_{generador}$

Se comenzará calculando las matrices A y B del sistema.

Tensión en el condensador número 1:

$$V_{c1} = \frac{I_1}{C_1} \quad (2.6.4)$$

El cálculo de I_1 , se puede expresar como:

$$I_1 = I_g - I \quad (2.6.5)$$

Donde I_g puede ser expresada en función de los estados como:

$$I_g = \frac{V - V_{c1}}{R_g} = \frac{1}{R_g} * V - \frac{1}{R_g} * V_{c1} \quad (2.6.6)$$

Con estas dos últimas ecuaciones, se puede definir perfectamente la fila 3 de las matrices A y B de estados.

$$\dot{V}_{c1} = \frac{\frac{V - V_{c1}}{R_g} - I}{C_1} = \frac{1}{R_g * C_1} * V - \frac{1}{R_g C_1} * V_{c1} - \frac{1}{C_1} * I \quad (2.6.7)$$

$$A = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ -\frac{1}{C_1} & 0 & -\frac{1}{R_g C_1} & 0 \\ x & x & x & x \end{pmatrix} ; B = \begin{pmatrix} x \\ x \\ \frac{1}{R_g * C_1} \\ x \end{pmatrix} \quad (2.6.8)$$

Se procederá con el resto de elementos de las matrices del mismo modo.

$$\dot{V}_{c2} = \frac{I_2}{C_2} \quad (2.6.9)$$

$$I_2 = I - I_L \quad (2.6.10)$$

$$\dot{V}_{c2} = \frac{1}{C_2} * I - \frac{1}{C_2} * I_L \quad (2.6.11)$$

En este caso no hay dependencia de la entrada V.

$$A = \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ -\frac{1}{C_1} & 0 & -\frac{1}{R_g C_1} & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 \end{pmatrix} ; B = \begin{pmatrix} x \\ x \\ \frac{1}{R_g * C_1} \\ 0 \end{pmatrix} \quad (2.6.12)$$

A continuación, se va a calcular los elementos de las matrices que afectan a las corrientes:

$$V_{c1} = I * R + \dot{I} * L + V_{c2} \quad (2.6.12)$$

$$I = \frac{V_{c1} - V_{c2} - I * R}{L} \quad (2.6.13)$$

$$V_{c2} = I_L * R_L + \dot{I}_L * L_L \quad (2.6.14)$$

$$\dot{I}_L = \frac{V_{c2} - I_L * R_L}{L_L} \quad (2.6.15)$$

Por lo tanto, quedando las matrices A y B:

$$\dot{x} = Ax + Bu \quad (2.6.16)$$

$$\begin{pmatrix} \dot{i} \\ \dot{I}_L \\ \dot{V}_{c1} \\ \dot{V}_{c2} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & \frac{1}{L} & -\frac{1}{L} \\ 0 & -\frac{R_L}{L_L} & 0 & \frac{1}{L_L} \\ -\frac{1}{C_1} & 0 & -\frac{1}{R_g C_1} & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 \end{pmatrix} \begin{pmatrix} I \\ I_L \\ V_{c1} \\ V_{c2} \end{pmatrix} + \begin{pmatrix} x \\ x \\ 1 \\ 0 \end{pmatrix} V \quad (2.6.17)$$

Para terminar de definir las ecuaciones del sistema, ya hemos mencionado que la salida es la tensión que cae en la carga. Esta tensión coincide con el voltaje del condensador C_2 , por lo que la salida coincide con un estado, haciendo que la matriz D valga 0.

$$y = Cx + Du \quad (2.6.18)$$

$$y = (0 \ 0 \ 0 \ 1) \begin{pmatrix} I \\ I_L \\ V_{c1} \\ V_{c2} \end{pmatrix} \quad (2.6.19)$$

A continuación, se van a mostrar las matrices del sistema en variables de estado para un circuito con 3 segmentos en π . Esto permite evaluar más puntos de la línea, pudiendo ver la evolución de corrientes y tensiones.

Las matrices no se justificarán como las mostradas anteriormente ya que el procedimiento es idéntico al llevado a cabo anteriormente.

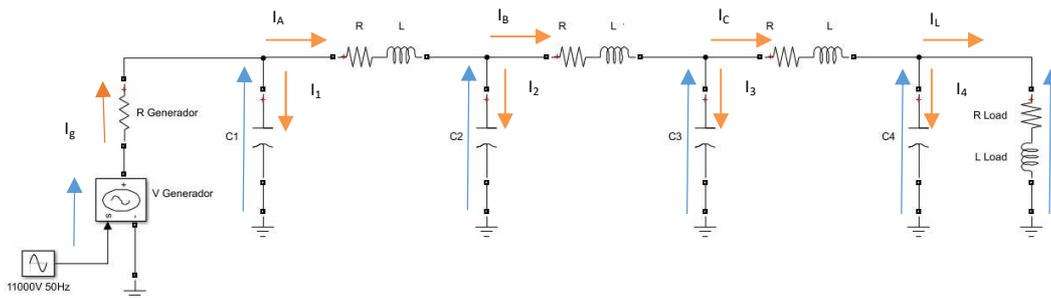


Ilustración 23. Esquema diagrama en pi 3 segmentos para VVEE

$$\begin{pmatrix} \dot{I}_A \\ \dot{I}_B \\ \dot{I}_C \\ \dot{I}_L \\ \dot{V}_{c1} \\ \dot{V}_{c2} \\ \dot{V}_{c3} \\ \dot{V}_{c4} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & 0 & 0 & 0 & \frac{1}{L} & -\frac{1}{L} \\ 0 & 0 & 0 & -\frac{R_L}{L_L} & 0 & 0 & 0 & \frac{1}{L_L} \\ -\frac{1}{C_1} & 0 & 0 & 0 & -\frac{1}{C_1 * R_g} & 0 & 0 & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_3} & -\frac{1}{C_3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_4} & -\frac{1}{C_4} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_A \\ I_B \\ I_C \\ I_L \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{R_g * C_1} \\ 0 \\ 0 \\ 0 \end{pmatrix} v \quad (2.6.20)$$

$$y = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \begin{pmatrix} I_A \\ I_B \\ I_C \\ I_L \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} \quad (2.6.21)$$

Por lo tanto, una matriz genérica será la siguiente:

$$\begin{pmatrix} \dot{I}_A \\ \vdots \\ \dot{I}_x \\ \dot{V}_{c1} \\ \vdots \\ \dot{V}_{cn} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & \dots & \dots & \frac{1}{L} & -\frac{1}{L} & \dots & 0 \\ 0 & -\frac{R}{L} & 0 & \dots & 0 & \frac{1}{L} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ddots & -\frac{1}{L} \\ 0 & 0 & 0 & -\frac{R_L}{L_L} & 0 & 0 & 0 & \frac{1}{L_L} \\ -\frac{1}{C_1} & 0 & \dots & \dots & -\frac{1}{C_1 * R_g} & 0 & \dots & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \frac{1}{C_n} & -\frac{1}{C_n} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_A \\ \vdots \\ I_x \\ V_{c1} \\ \vdots \\ V_{cn} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{R_g * C_1} \\ \vdots \\ 0 \end{pmatrix} v \quad (2.6.22)$$

$$y = (0 \ \dots \ 0 \ 1) \begin{pmatrix} I_A \\ \vdots \\ I_x \\ V_{c1} \\ \vdots \\ V_{cn} \end{pmatrix} \quad (2.6.23)$$

Comparando las funciones de transferencia que se obtendrán en cada caso, podemos ver que el orden aumenta de manera notable con el número de segmentos que se desea introducir en la línea. Lo que convertirá al sistema en un sistema más complejo y lento a la hora de procesar. En este punto, el usuario deberá valorar sus necesidades y las capacidades de procesamiento.

$$\frac{1.615e13 s + 1.615e19}{s^4 + 1.085e06 s^3 + 8.542e10 s^2 + 5.343e14 s + 1.615e19}$$

Ilustración 24. FT un segmento

$$\frac{1.516e29 s + 1.516e35}{s^8 + 1.088e06 s^7 + 8.919e10 s^6 + 1.224e15 s^5 + 5.182e19 s^4 + 3.393e23 s^3 + 7.461e27 s^2 + 2.206e31 s + 1.517e35}$$

Ilustración 25. FT tres segmentos

2.6.2 En función de la carga.

El esquema que se ha discretizado es el siguiente (figura 26). Como se puede observar, se trata del circuito empleado anteriormente, pero en vacío. Es decir, no hay una carga conectada en paralelo con el último condensador. Esto reduce el número de estados de las matrices en variables de estado, de 8 estados a 7 estados.

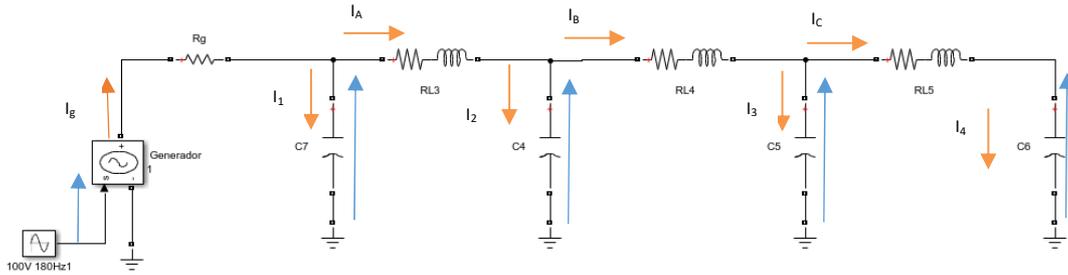
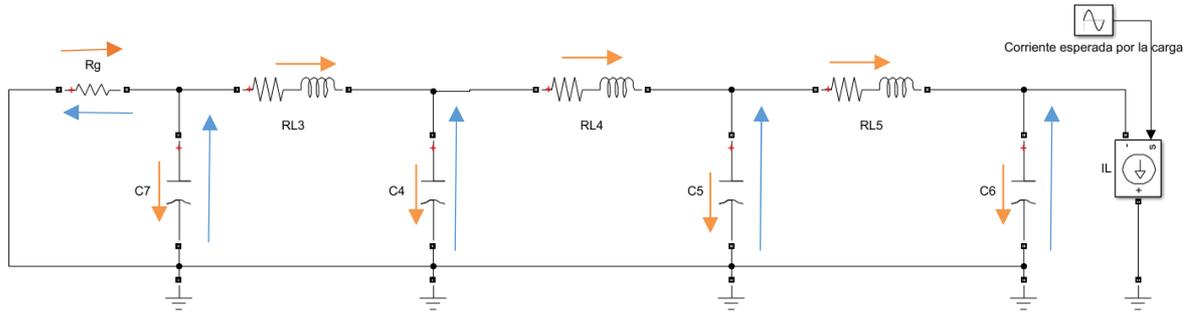


Ilustración 26. Diagrama sistema en VVEE en vacío.

$$\begin{pmatrix} \dot{I}_A \\ \dot{I}_B \\ \dot{I}_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} \\ -\frac{1}{C_1} & 0 & 0 & -\frac{1}{R_g * C_1} & 0 & 0 & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_3} & -\frac{1}{C_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_4} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{R_g * C_1} \\ 0 \\ 0 \\ 0 \end{pmatrix} I_L \quad (2.6.24)$$

$$y = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} \quad (2.6.25)$$

Para complementar este apartado, se trabajará con un generador de corriente, simulando un consumo por la carga desconocida. El esquema será el siguiente:



El resultado de sus matrices en variables de estado:

$$\begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} \\ -\frac{1}{C_1} & 0 & 0 & -\frac{1}{R_g * C_1} & 0 & 0 & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & \frac{1}{C_3} & 0 & 0 & 0 \\ 0 & \frac{1}{C_3} & -\frac{1}{C_3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_3} & \frac{1}{C_4} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{C_4} \end{pmatrix} I_L \quad (2.6.26)$$

$$y = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} \quad (2.6.27)$$

2.7 Discretización del modelo y comparación.

En el punto anterior se han mostrado las funciones de transferencia en espacio de estados en su forma continua. Para el trabajo con el microprocesador se discretizarán estas funciones. Esto se hará mediante código en MATLAB, empleando varios métodos de paso a forma discreta, viendo cual presenta una mejor respuesta.

El código empleado para llevar a cabo esto es el siguiente.

```
Ts=10e-6;

linea_lseg = ss(A,B,Cm,D)
[num_lseg,den_lseg] = ss2tf(A,B,Cm,D)
linea_FT_lseg = tf(num_lseg,den_lseg)

linea = ss(A3,B3,C3,D)
[num,den] = ss2tf(A3,B3,C3,D)
linea_FT = tf(num,den)

linea_discreta_impulse=c2d(linea,Ts,'impulse')
linea_discreta_zoh=c2d(linea,Ts,'zoh')
linea_discreta_foh=c2d(linea,Ts,'foh')
linea_discreta_tustin=c2d(linea,Ts,'tustin')
linea_discreta_matched=c2d(linea,Ts,'matched')

figure(1)
step(linea_discreta_impulse)
hold on
step(linea_FT)

figure(2)
step(linea_discreta_zoh)
hold on
step(linea_FT)

figure(3)
step(linea_discreta_foh)
hold on
step(linea_FT)

figure(4)
step(linea_discreta_tustin)
hold on
step(linea_FT)

figure(5)
step(linea_discreta_matched)
hold on
step(linea_FT)
```

Como se puede ver en el código, los métodos de discretización empleados son: *impulse*, *zoh*, *foh*, *Tustin* y *Matched*.

IMPULSE

Como se puede apreciar en su figura de respuesta ante una entrada escalón no se aprecian grandes diferencias entre el sistema continuo y el sistema en tiempo discreto.

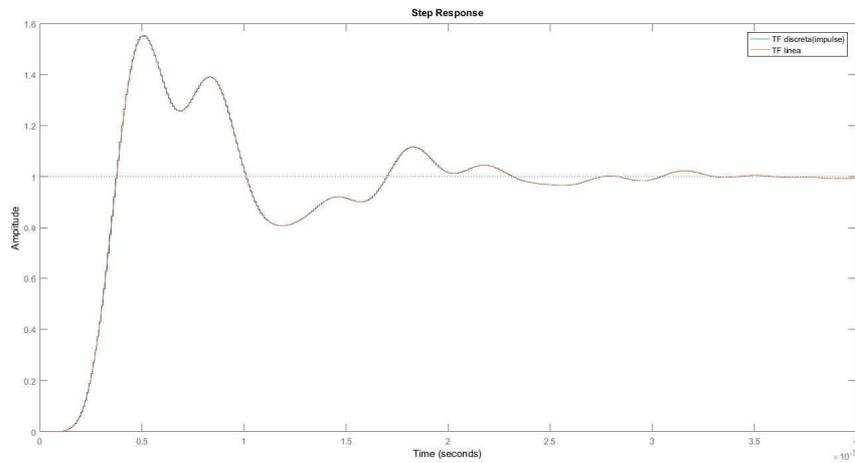


Ilustración 27. Respuesta del sistema discretizado con impulse ante entrada escalón

En detalle se puede apreciar lo siguiente:

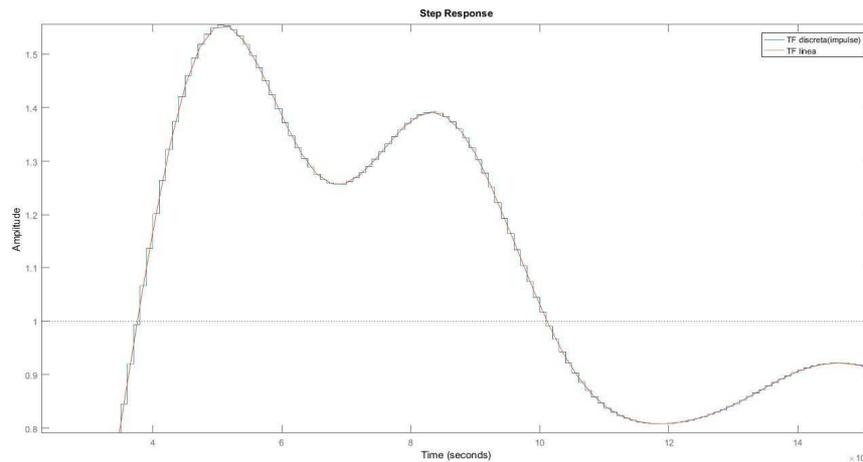


Ilustración 28. Detalle de la respuesta

Como se ve la respuesta en continua pasa justamente por el punto medio del paso de simulación. También se puede observar su estabilidad en la comparativa de diagramas de *Bode*.

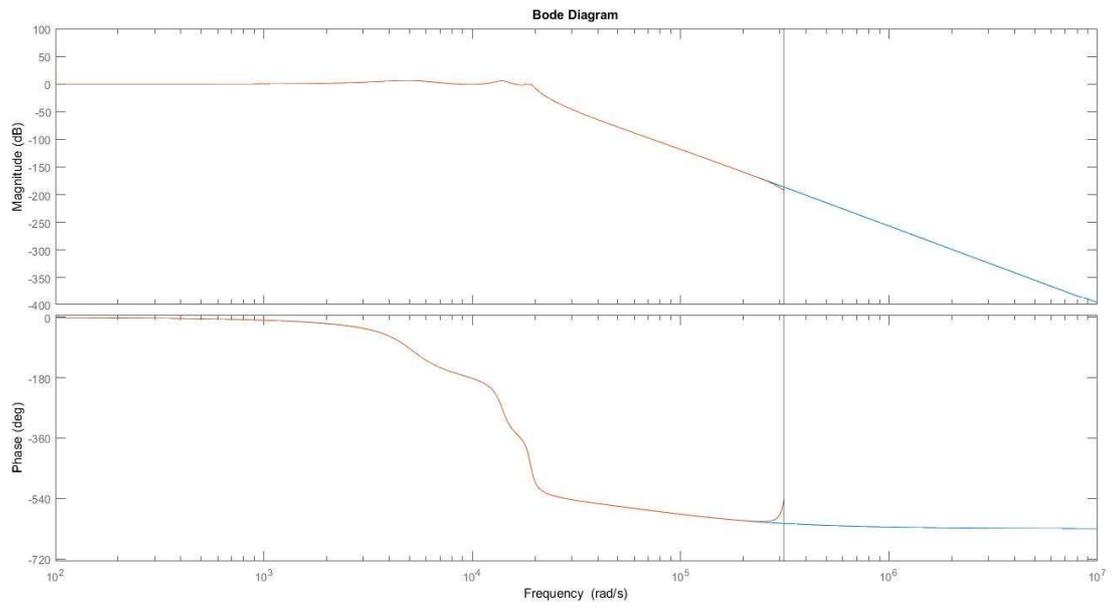


Ilustración 29. Diagrama de Bode sistema con Impulso

ZOH

Como se puede apreciar en su figura de respuesta ante una entrada escalón no se aprecian grandes diferencias entre el sistema continuo y el sistema en tiempo discreto.

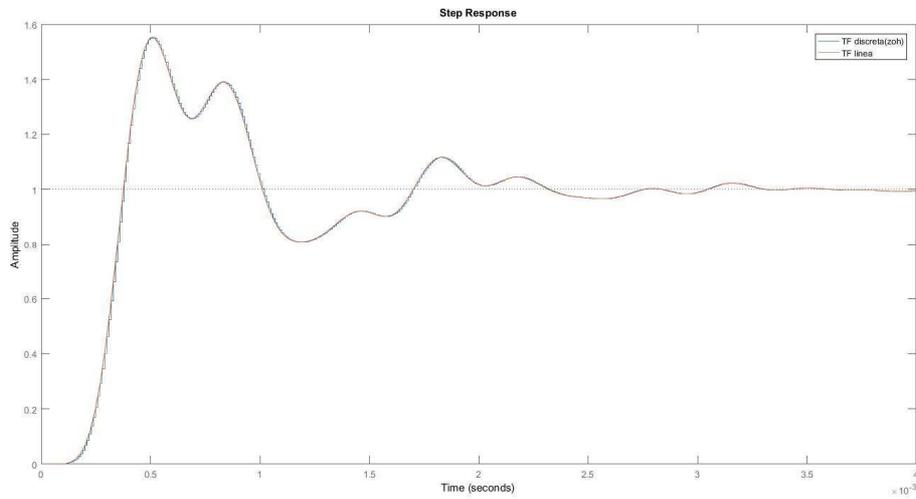


Ilustración 30. Respuesta del sistema ante escalón discretizado con ZOH

En detalle se puede apreciar lo siguiente:

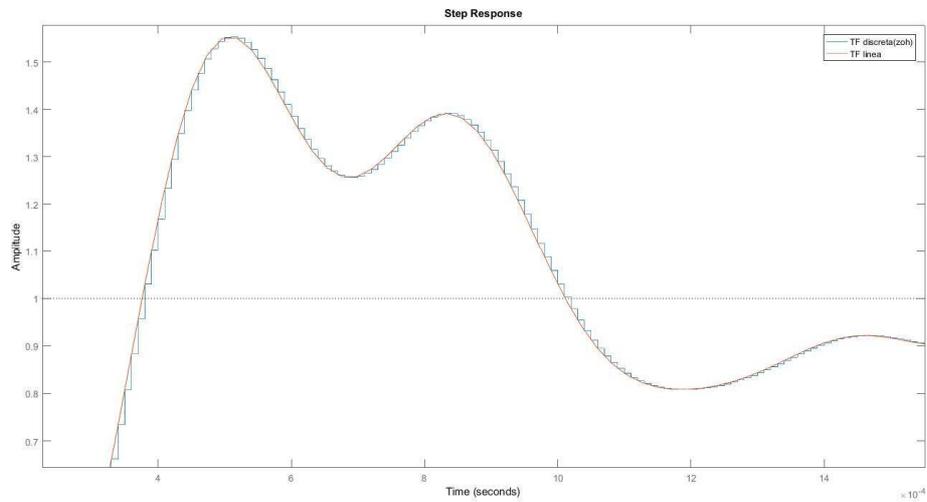


Ilustración 31. Detalle de la respuesta

A continuación, se van a mostrar el diagrama de Bode del sistema y la respuesta ante la entrada real de tipo senoidal.

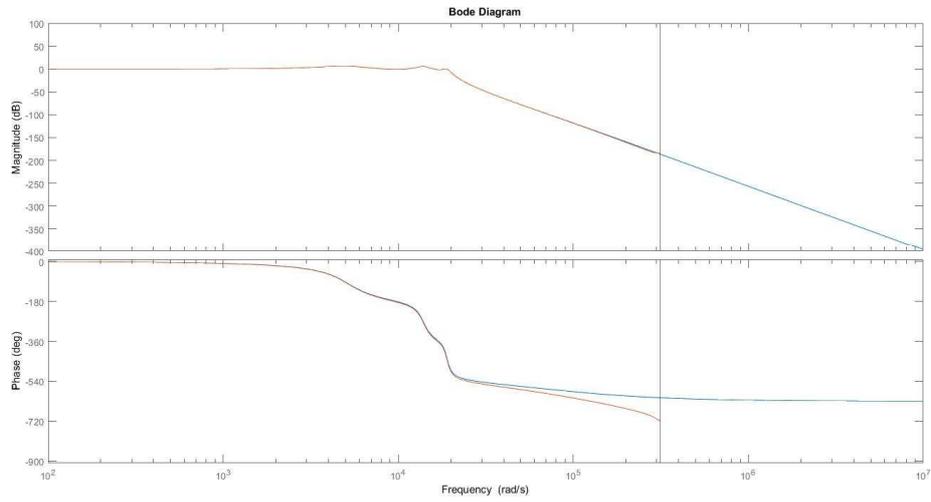


Ilustración 32. Diagrama Bode del sistema discretizado con ZOH

FOH

Como se puede apreciar en su figura de respuesta ante una entrada escalón no se aprecian grandes diferencias entre el sistema continuo y el sistema en tiempo discreto.

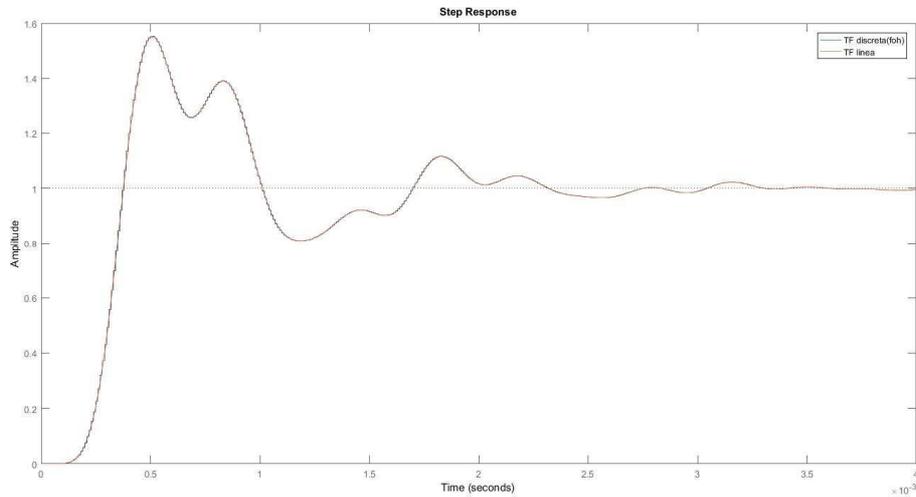


Ilustración 33. Respuesta ante escalón discretizado con FOH

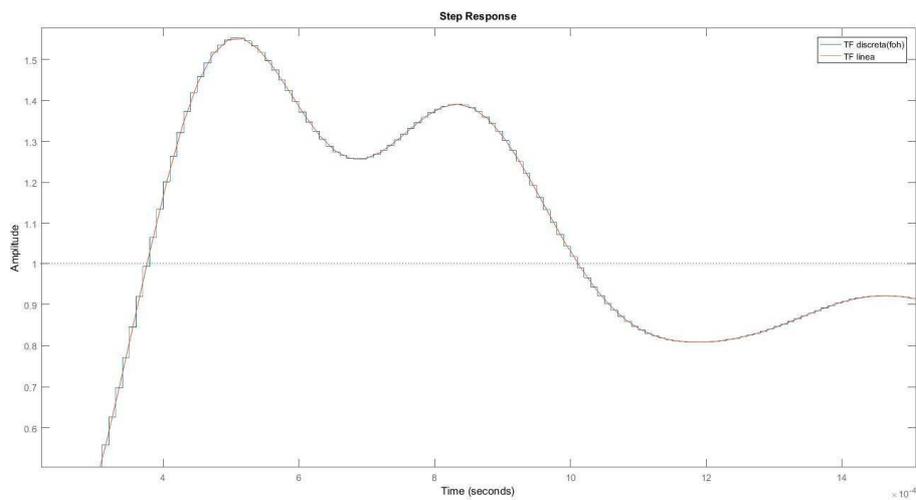


Ilustración 34. Detalle de respuesta

El diagrama Bode del sistema es el siguiente. Como en los casos anteriores, no existe diferencia entre el sistema continuo y el sistema discreto.

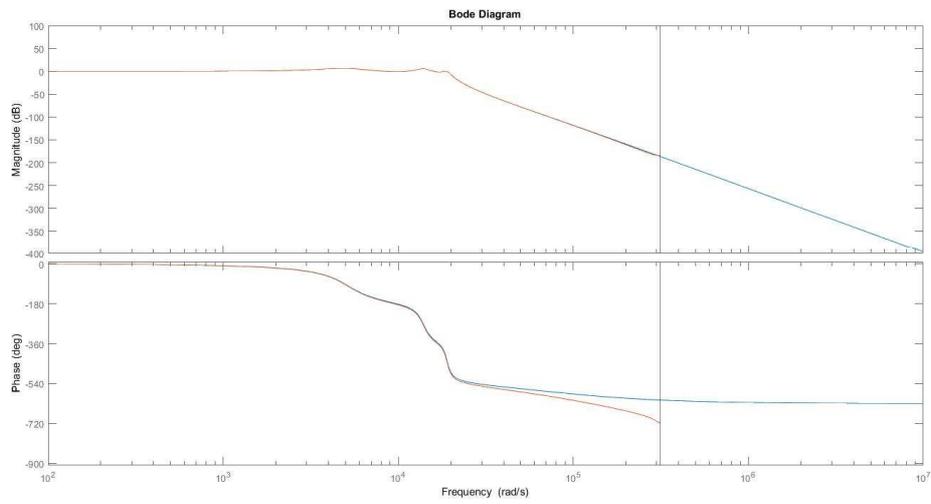


Ilustración 35. Bode del sistema FOH

TUSTIN

Como se puede apreciar en su figura de respuesta ante una entrada escalón no se aprecian grandes diferencias entre el sistema continuo y el sistema en tiempo discreto.

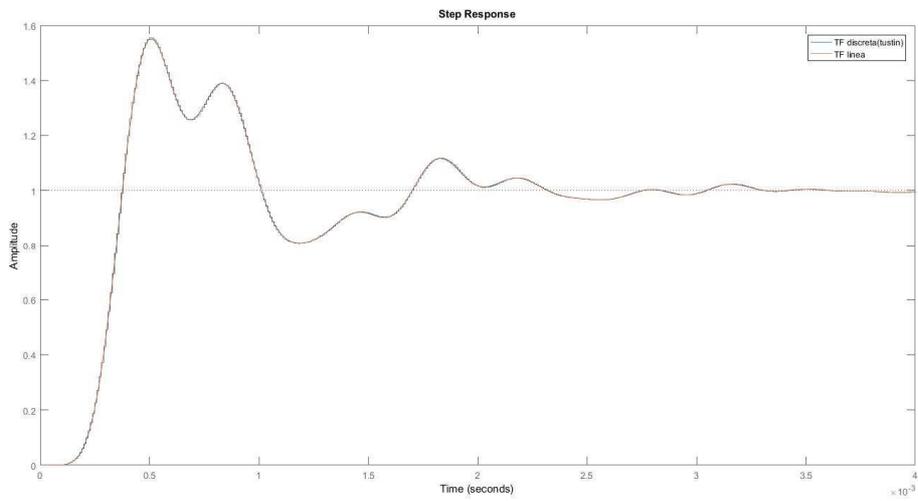


Ilustración 36. Respuesta escalón sistema discretizado con Tustin

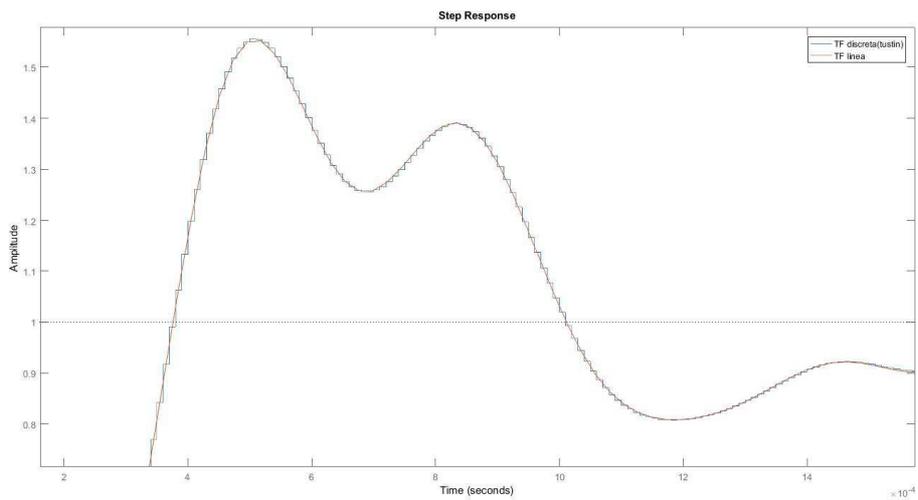


Ilustración 37. Detalle de la respuesta

En el siguiente diagrama de Bode se pueden apreciar diferencias en frecuencias altas, para el uso que se le quiere dar en este trabajo, estas diferencias no son representativas. Por otra parte, aparece una diferencia de fase de 720° . Esto se debe a problemas de la simulación con MATLAB, pero la fase es la misma al ser un múltiplo de 360° .

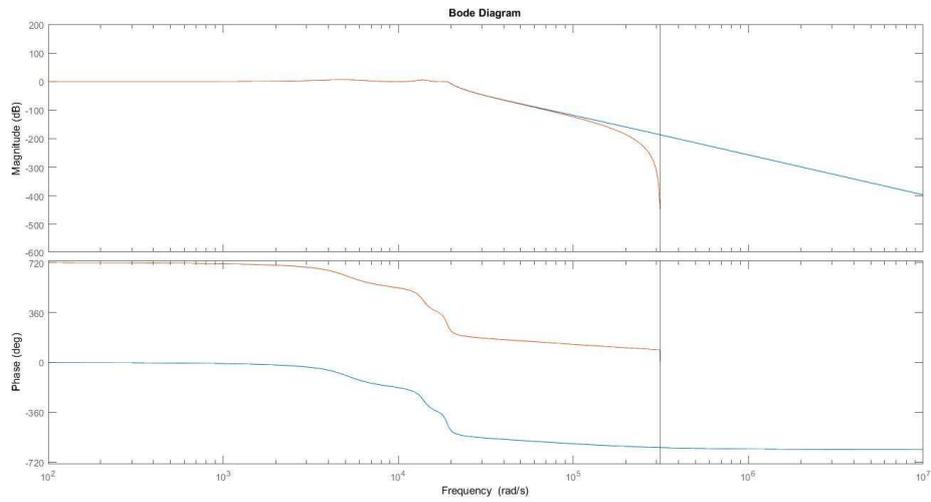


Ilustración 38. Bode sistema discretizado Tustin

MATCHED

Como se puede apreciar en su figura de respuesta ante una entrada escalón no se aprecian grandes diferencias entre el sistema continuo y el sistema en tiempo discreto.

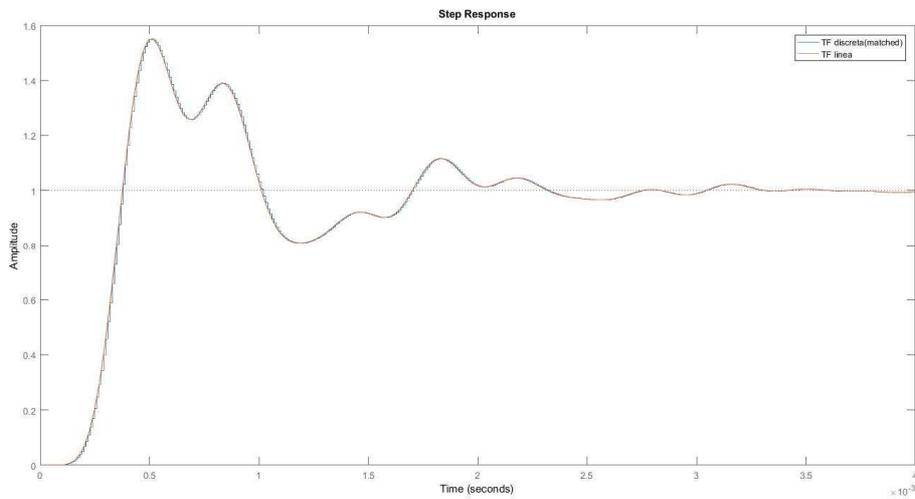


Ilustración 39. Respuesta ante escalón sistema discretizado Matched

En esta simulación se puede apreciar el tipo de aproximación que se está empleando. Al contrario que en Tustin, el punto de cruce entre funciones es al comienzo del escalón de cuantificación.

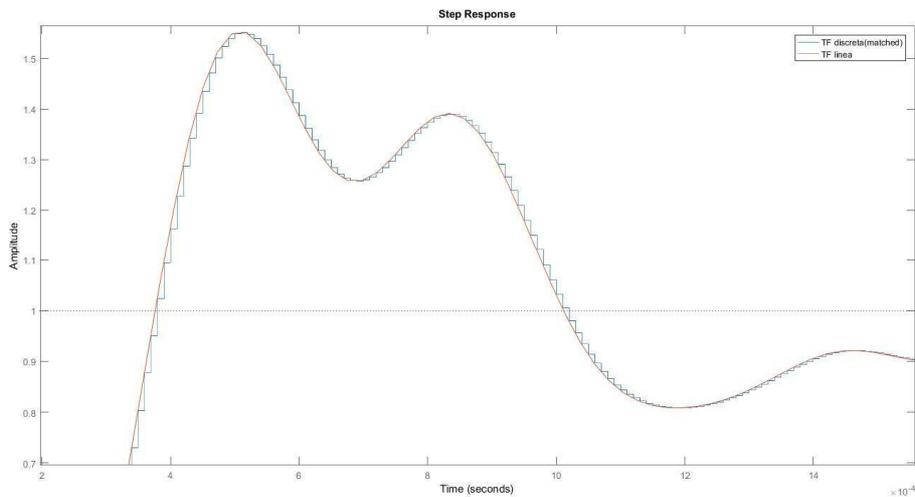


Ilustración 40. Detalle de la respuesta

En el siguiente diagrama de Bode sucede algo parecido al sistema discretizado con Tustin. Muestra una discrepancia de fase de 360°

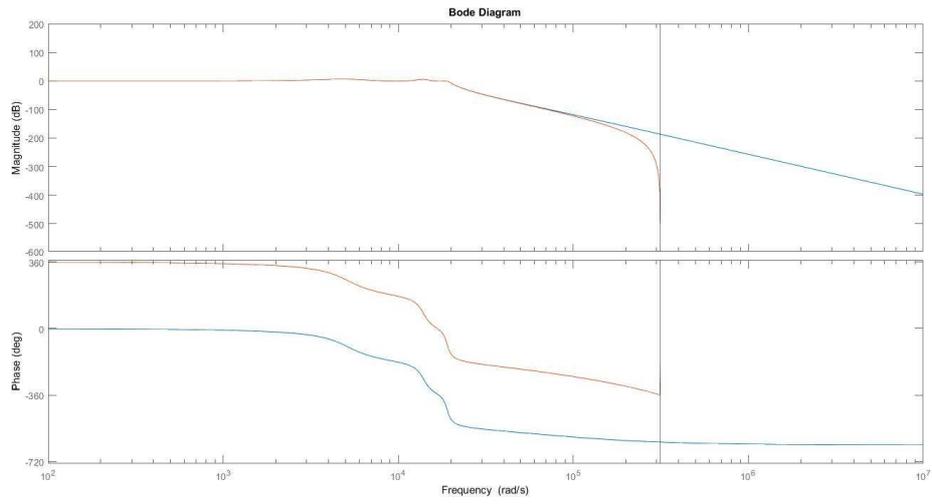


Ilustración 41. Bode del sistema discreto con Matched

3. Soporte físico

3.1 Micro C2000 F28377D

Como se ha mencionado con anterioridad, para desarrollar este proyecto se va a emplear un microcontrolador de la serie C2000 de *Texas Instruments*, concretamente el F28377D. En las siguientes líneas se va a proceder a realizar una pequeña introducción sobre las principales características de este microcontrolador.

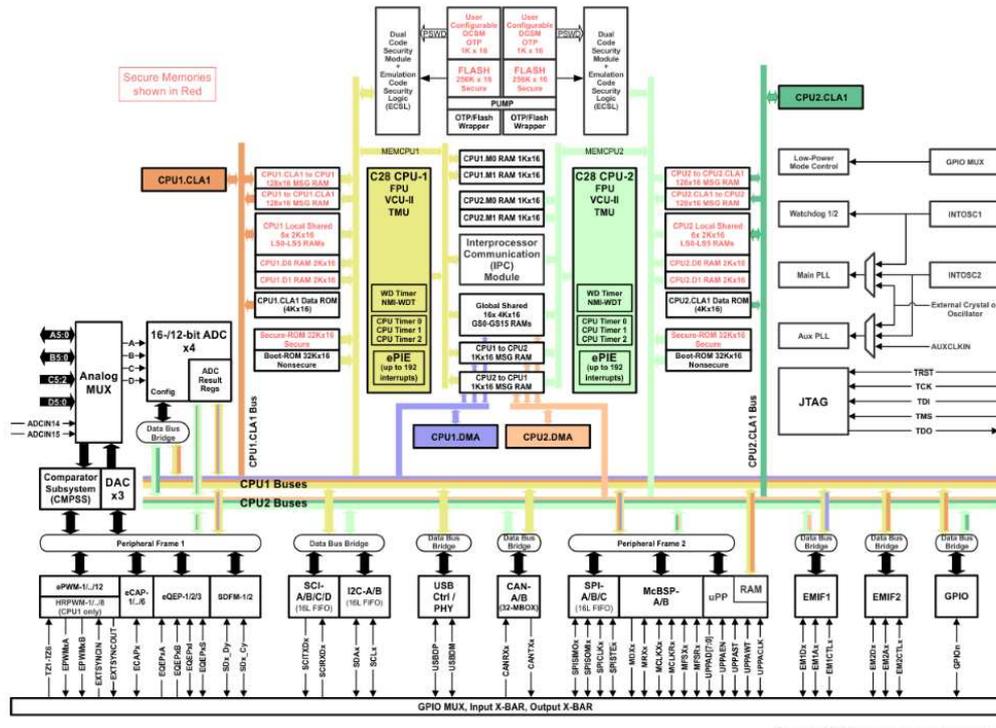
Se trata de un potente microcontrolador de 32bit de coma flotante. Su velocidad de funcionamiento es de 200MHz en sus dos núcleos. Tiene memoria flash de 1Mb Su principal aplicación es en sistemas de control en lazo cerrado como los que pueden ser necesarios en el control de un servo motor, convertidores e inversores solares, líneas de potencia.

Sus principales características desglosadas son las siguientes:

- Arquitectura de doble núcleo
 - 2 CPUs de 32-bit TMS320C28x.
 - Velocidad de proceso de hasta 200MHz.
 - Unidad de coma flotante precisión simple.
 - Unidad de cálculos trigonométricos (TMU).
 - Unidad Viterbil Cálculo matemático complejo (VCU-H).
- Dos Programmable Control Law Accelerators (CLAs): Co-procesadores que trabajan en paralelo con la CPU.
 - Velocidad de proceso de hasta 200MHz.
 - Unidad de coma flotante precisión simple.
 - Funcionamiento autónomo independiente de la CPU.
- Memoria integrada
 - 512KB o 1MB Flash con protección ECC o por paridad.
 - 172 o 204KB RAM con protección ECC o por paridad.
- Reloj y control del sistema
 - Dos osciladores internos 10MHz tipo zero-spin.
 - Oscilador interno cristal.
 - Circuito de detección de fallo en reloj.
- Diseño de núcleo 1.2V, 3.3V Entradas/Salidas.
- Periféricos del sistema
 - Dos interfaces de memoria externa (EMIF) con soporte ASRAM y SDRAM.
 - Doble DMA (Direct Memory Access) de 6 canales.
 - Hasta 169 GPIOs multiplexados individualmente con filtro de entrada.

- Controladores de Interrupciones de periféricos PIE.
 - Soporte para modo de baja potencia (LPM) con wake-up externo.
 - Periféricos de comunicaciones
 - USB 2.0 (MAC+PHY)
 - Soporte para interfaz puerto paralelo universal (uPP) 12-pin 3.3V.
 - 2 módulos CAN.
 - 3 puertos SPI de alta velocidad (hasta 50MHz).
 - Dos MCBPs (Multichannel Buffered Serial Ports).
 - 4 Interfaces de comunicación SCI/UART.
 - 2 interfaces I2C.
 - Subsistemas Analógicos.
 - 4 módulos ADC
 - Modo 16bit.
 - MSPS cada uno hasta 4.4 MSPS de salida total del sistema.
 - Entradas diferenciales.
 - Hasta 12 canales externos.
 - Modo 12bit.
 - 3.5 MSPS cada uno, hasta 14 MSPS de salida total del sistema.
 - Entradas Single-Ended.
 - Hasta 24 canales externos.
 - Sample and hold (S/H) único para cada ADC.
 - Post-procesado integrado en hardware de las conversiones ADC.
 - Calibración de offset.
 - Errores por cálculo de setpoints.
 - High, Low and Zero-crossing Compare, con capacidad de generación de interrupciones.
 - Captura del desfase disparo-muestreado.
 - 8 Comparadores con referencias DAC (Digital to analog converter) de 12bit.
 - 3 salidas DAC de 12 bit con buffer.
- Periféricos de control mejorados.
 - 24 canales PWM.
 - 16 canales PWM de alta resolución (HRPWM)
 - Alta resolución en los canales A y B de 8 módulos PWM.
 - Soporte DeadBand en PWM t HRPWM.
 - 6 Módulos de captura de eventos eCAP
 - 3 Módulos de lectura de encoders en cuadratura

- 9 canales de entrada para modulo filtro Sigma-Delta, dos filtros paralelos por canal.
- Rango de temperaturas desde -40 hasta 105°C



Atendiendo a estas características, se puede afirmar que se trata de un procesador correcto para el proyecto que se quiere desarrollar.

3.2 MATLAB

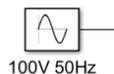
MATLAB será una de las principales herramientas para desarrollar este proyecto. Como se ha ido desglosando en páginas anteriores, se puede diferenciar un primer análisis, para el que se podrán realizar las simulaciones que se desee en *Simulink*, la parte de discretización del sistema, en la que también se emplearán archivos *m* de MATLAB, aunque posteriormente deban ser convertidos a código en C para el volcado en el microprocesador.

En un comienzo se trató de realizar una mayor parte de este trabajo en entorno *Simulink* y MATLAB, tratando de dotar al sistema con una interfaz más cómoda y amigable para el usuario, pero aparecieron problemas de Software que no pudieron ser resueltos. En próximas versiones de MATLAB, se cree que podrá ser fácilmente la adaptación de este trabajo a ese entorno, lo que podrá ser un buen cambio.

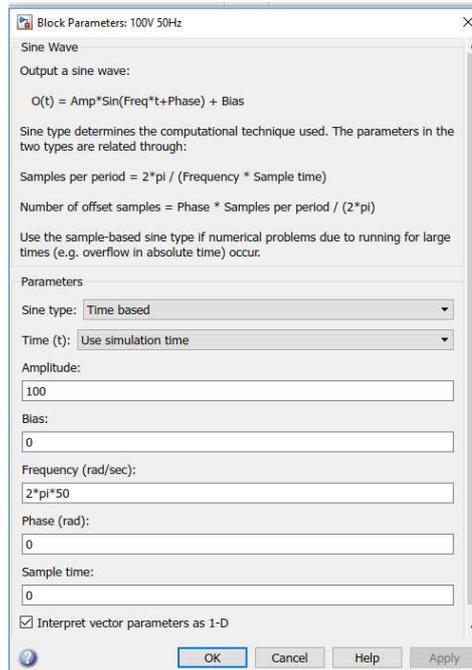
En las siguientes líneas se detallan algunos de los bloques y esquemas usados para la emulación de las líneas de transmisión, tanto a nivel teórico (tomando valores reales de cables) como realizando emulaciones de posibles líneas.

Los bloques que se van a utilizar son los siguientes:

En primer lugar, una fuente generadora de señal. En este caso se elige una señal sinusoidal, simulando la forma de onda de la corriente alterna. Fácilmente sustituible por cualquier otra forma de onda para observar que sucede en el circuito:

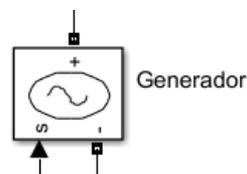


De este bloque se pueden variar los siguientes parámetros:



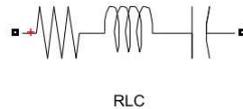
Se puede observar que en el encabezado del bloque se muestra un ejemplo explicando que forma de onda se va a reproducir. Se trata de una senoidal en la que se puede variar su amplitud; su “Offset”, es decir donde queremos que se encuentre el valor medio de la señal (Bias); su frecuencia, requerida en radianes segundo por lo que es necesario multiplicarla por $2 * \pi$; el desfase que se desea que tenga el seno expresado en radianes.

Esta señal no se puede conectar de forma directa al circuito, ya que Matlab la entiende simplemente como una forma de onda matemática. Para darle un sentido eléctrico de tensión es necesario hacerla pasar por el siguiente convertidor:

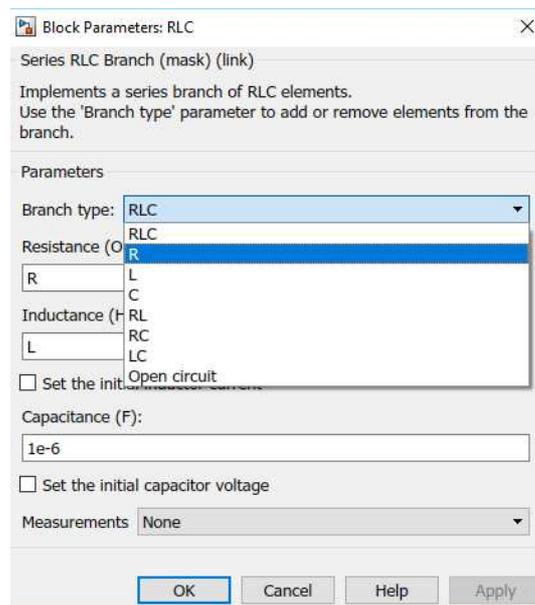


Este bloque no tiene mayor importancia que la mencionada. En la patilla “S” se introduce la señal que se quiere convertir a tensión, quedando libres los terminales (+) y (-) como se espera de un generador de tensión.

El siguiente bloque a comentar es el empleado para simular resistencias, bobinas y condensadores. El bloque en las librerías Matlab se llama "Series RLC Branch".

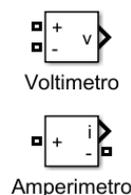


Que aparezcan los 3 componentes en serie no significa que tenga que ser así siempre, como se puede ver en su bloque de parámetros:



En la pestaña que se despliega, se puede elegir la combinación que se necesite. Una vez elegida la combinación, se introducen los valores que se desea que tengan estos parámetros. No es necesario poner un valor numérico, lo que sería bastante tedioso a la hora de tener que cambiar muchos valores. Se puede dejar en función de una variable a la que se puede cambiar su valor mediante un script de código en Matlab. Esto último es lo que se ha hecho realizando este proyecto.

Por último, mostrar los bloques que se han empleado para la toma de datos.



En la imagen anterior se puede ver un medidor de tensión y un medidor de corriente.

Como se sabe, el medidor de tensión se debe colocar en paralelo con la tensión que se desea medir. Para ello se emplearán las patillas (+) y (-). En la patilla “V” se puede colocar un bloque para almacenar la variable o un *Scope* para visualizar la forma de onda de la tensión.

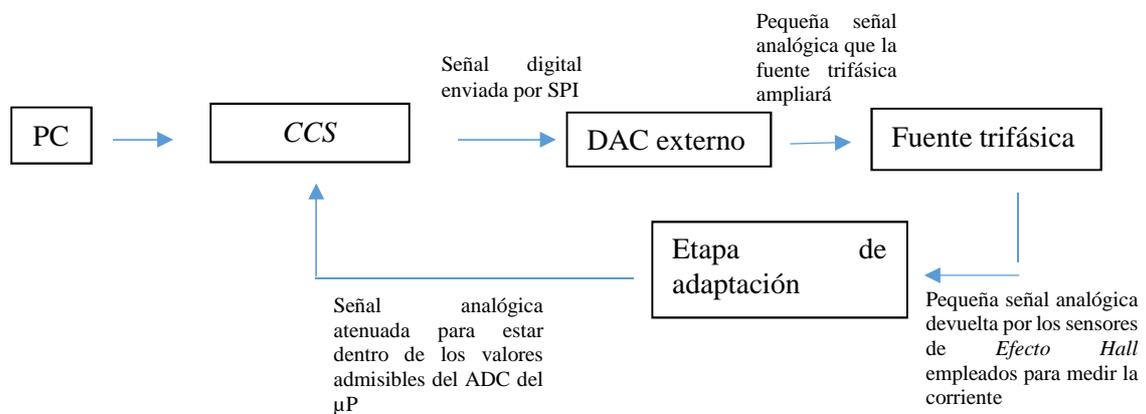
El funcionamiento del amperímetro es muy similar, pero no se debe colocar en paralelo, sino en serie en la rama que se desea medir la corriente. En la patilla “I” se puede ver su forma de onda.

Todos estos bloques son usados en un entorno de simulación *Simulink*, pero no sirven para la emulación mediante el microprocesador.

Para poder ser descargado al microprocesador, es necesaria la realización de un archivo en C que va a permitir enviar una señal *SPI* al DAC2420 elegido para regular la tensión que genere la fuente y también la programación del ADC del F28377D para la recogida de datos de la fuente trifásica. En este archivo también se encontrará la conversión del sistema discreto, obtenido en MATLAB.

A continuación, se mostrarán los bloques empleados para poder volcar el contenido al micro.

Para entender los bloques que se van a usar, se va a mostrar un esquema del flujo de trabajo:



A continuación, se va a hablar del código en C empleado para obtener el modelo en variables de estado y su paso a tiempo discreto.

Se van a facilitar las matrices en variables de estado para 1 segmento en *pi* y para 3 segmentos.

Matrices para un único segmento:

```
A = [-R/L 0 1/L -1/L;
      0 -R_Load/L_Load 0 1/L_Load;
      -1/(C/2) 0 -1/(Rg*C/2) 0;
      1/(C/2) -1/(C/2) 0 0];

B = [0;0;1/(Rg*C/2);0];
Cm = [0 0 0 1];
D = [0];
```

Se ha llamado "Cm" a la matriz, por problemas de nomenclatura con la "C" referente al valor del condensador.

Matrices para 3 segmentos:

```
A3 = [-R/L 0 0 0 1/L -1/L 0 0;
       0 -R/L 0 0 0 1/L -1/L 0;
       0 0 -R/L 0 0 0 1/L -1/L;
       0 0 0 -R_Load/L_Load 0 0 0 1/L_Load;
       -1/(C/2) 0 0 0 -1/(Rg*(C/2)) 0 0 0;
       1/C -1/C 0 0 0 0 0 0;
       0 1/C -1/C 0 0 0 0 0;
       0 0 1/(C/2) -1/(C/2) 0 0 0 0];

B3 = [0;0;0;0;1/(Rg*(C/2));0;0;0];

C3 = [0 0 0 0 0 0 0 1];
```

En este caso falta la matriz D, pero su valor volverá a ser [0], ya que la salida no depende de forma directa de la entrada.

Las matrices mostradas pertenecen a un sistema de carga conocida. A continuación, se mostrarán las 2 matrices restantes, que serán las empleadas para simular en el microprocesador.

Matrices de la función de transferencia V_T en VVEE.

```
A3_sis_vacio = [-R/L 0 0 1/L -1/L 0 0;
                 0 -R/L 0 0 1/L -1/L 0;
                 0 0 -R/L 0 0 1/L -1/L;
                 -1/(C/2) 0 0 -1/(Rg*(C/2)) 0 0 0;
                 1/C -1/C 0 0 0 0 0;
                 0 1/C -1/C 0 0 0 0;
                 0 0 1/(C/2) 0 0 0 0];

B3_sis_vacio = [0;0;0;1/(Rg*(C/2));0;0;0];

C3_sis_vacio = [0 0 0 0 0 0 1];
```

Ilustración 42. Matrices sistema en vacío VVEE

Matrices de la función de transferencia V_z en VVEE.

```
A3_sis_cargado = [-R/L 0 0 1/L -1/L 0 0;  
0 -R/L 0 0 1/L -1/L 0;  
0 0 -R/L 0 0 1/L -1/L;  
-1/(C/2) 0 0 1/(Rg*(C/2)) 0 0 0;  
1/C -1/C 0 0 0 0 0;  
0 1/C -1/C 0 0 0 0;  
0 0 1/(C/2) 0 0 0 0];  
  
B3_sis_cargado = [0;0;0;0;0;0;-1/(C/2)];  
  
C3_sis_cargado = [0 0 0 0 0 0 1];
```

Ilustración 43.Matrices del sistema cargado en VVEE

Con este archivo, el usuario simplemente tiene que introducir los valores de su línea, ya sea calculada desde 0 o de un manual técnico con valores reales, lo que puede ser realmente interesante. Estas matrices pueden sufrir ligeras variaciones, si se quieren dar salidas de puntos medios de la línea, lo que afectaría directamente a la matriz C.

El paso previo a poder discretizar el sistema es la obtención de una matriz de transferencia que permita aplicar una entrada y obtener la salida. Para lo que se utilizarán las siguientes líneas.

Discretización de líneas con 1 y 3 segmentos conocida la entrada y la carga, para obtener la salida.

```
linea_lseg = ss(A,B,Cm,D);  
[num_lseg,den_lseg] = ss2tf(A,B,Cm,D);  
linea_TF_lseg = tf(num_lseg,den_lseg);  
  
linea=ss(A3,B3,C3,D);  
[num,den] = ss2tf(A3,B3,C3,D);  
linea_TF = tf(num,den);
```

Discretización de funciones de transferencia en vacío y con carga desconocida.

```
linea_vacio=ss(A3_sis_vacio,B3_sis_vacio,C3_sis_vacio,D);  
[num_vacio,den_vacio] = ss2tf(A3_sis_vacio,B3_sis_vacio,C3_sis_vacio,D);  
linea_TF_vacio = tf(num_vacio,den_vacio);  
  
linea_vacio_tustin=c2d(linea_vacio,Ts,'tustin');  
linea_TF_vacio_tustin=c2d(linea_TF_vacio,Ts,'tustin');
```

Ilustración 44. Discretización de la función de transferencia en vacío.

```

linea_TF_vacio_tustin =
2.238e-11 z^7 + 1.567e-10 z^6 + 4.7e-10 z^5 + 7.833e-10 z^4 + 7.833e-10 z^3 + 4.7e-10 z^2 + 1.567e-10 z + 2.238e-11
-----
z^7 - 6.714 z^6 + 19.3 z^5 - 30.81 z^4 + 29.47 z^3 - 16.9 z^2 + 5.374 z - 0.7314

```

Ilustración 45. Función de transferencia en vacío

```

linea_cargada=ss(A3_sis_cargado,B3_sis_cargado,C3_sis_cargado,D);
[num_cargada,den_cargada] = ss2tf(A3_sis_cargado,B3_sis_cargado,C3_sis_cargado,D);
linea_TF_cargada = tf(num_cargada,den_cargada);

linea_cargada_tustin=c2d(linea_cargada,Ts,'tustin');
linea_TF_cargada_tustin=c2d(linea_TF_cargada,Ts,'tustin');

```

Ilustración 46. Discretización del sistema con carga

```

linea_TF_cargada_tustin =
-1.483 z^7 + 6.996 z^6 - 11.69 z^5 + 5.388 z^4 + 7.373 z^3 - 11.3 z^2 + 5.804 z - 1.085
-----
z^7 - 6.714 z^6 + 19.3 z^5 - 30.81 z^4 + 29.47 z^3 - 16.9 z^2 + 5.374 z - 0.7314

```

Ilustración 47. Función de transferencia cargada.

Se muestran las funciones de transferencia discretizadas únicamente de los 2 últimos casos valorados, ya que estas son con las que se trabajará dentro del procesador.

Estos sistemas han sido discretizados por varios metodos. Todos ellos han resultado satisfactorios. Finalmente se ha optado por el empleo del método de *Tustin*.

Cualquiera de los métodos empleados, deberá ser convertido a C para que sea de utilidad con el micro F28377D. Este paso se detallará más adelante.

3.3 Code Composer Studio (CCS)

En un primer desarrollo se trató de evitar esta plataforma para el desarrollo de este proyecto, pero debido a las prestaciones finales que se obtuvieron del entorno *Simulink* ha sido totalmente necesaria la incorporación de este Software al desarrollo.

En este entorno se realizará la parte de la emulación de la línea trifásica, es decir la última etapa de este proyecto. Los factores que han hecho completamente necesario el uso de *CCS*, han sido la problemática configuración del *SPI* y la compleja recepción de los datos que debe convertir el ADC en la plataforma *Simulink*.

Se comenzará explicando la configuración del *SPI*. Para su configuración, se ha tomado como base uno de los múltiples ejemplos que facilita *Texas Instruments*, dentro de *ControlSUITE*. Conceptualmente, lo que se quiere realizar es enviar un número, comprendido entre 0x0000 y 0x0FFF al DAC externo. Como se expondrá en el siguiente punto de este trabajo, el DAC cuenta con un tamaño de palabra de 12 bits y otros 2 bits para direccionar su salida.

Para la transmisión del dato únicamente será necesaria la llamada a las funciones definidas para el microcontrolador.

```
//  
// Transmit data  
//  
spi_xmit(sdata);
```

Ilustración 48. Transmisión del dato que se desea

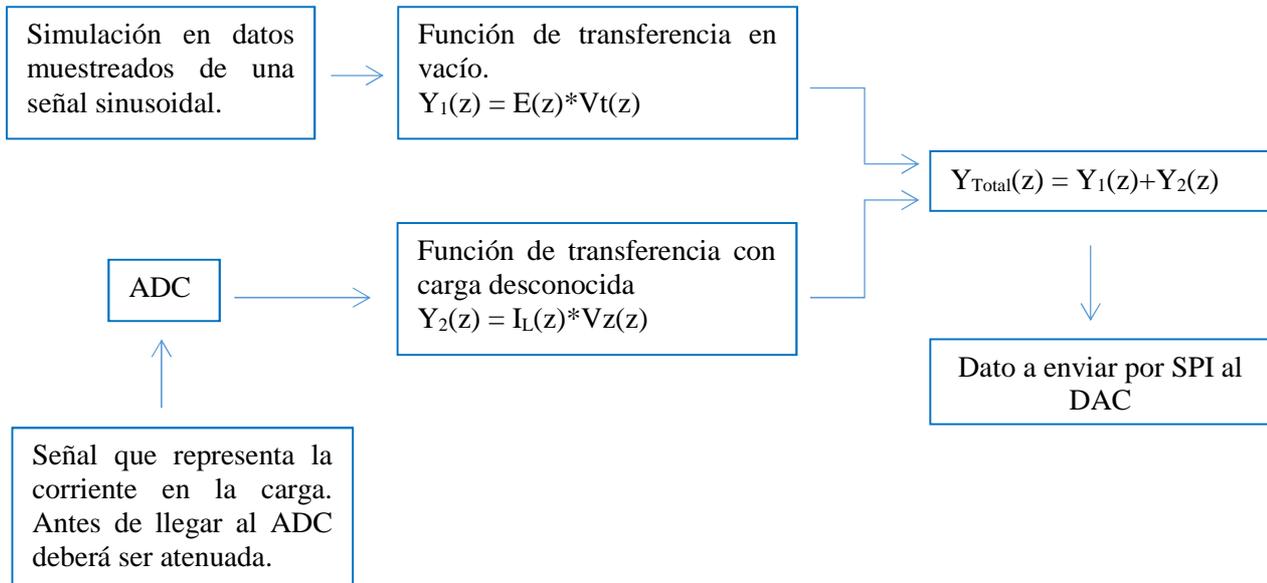
Con esta información, diremos que el dato DDXX 0000 0000 0000 en binario dará como salida del DAC su referencia inferior de voltaje que será fijada de forma externa. Mientras que el dato DDXX 1111 1111 1111 dará como salida la referencia de voltaje superior, también fijada de forma externa. La transmisión de estos datos es gobernada por el reloj encargado de controlar la transmisión de datos del *SPI* y será conectado físicamente al DAC externo.

Para la configuración del ADC, también se ha tomado como base un ejemplo de configuración proporcionado por *Texas Instruments*. En este caso, se trata de un ADC de 12 bits, interno en el micro F28377D. La señal que le debe llegar a este ADC ya ha debido ser modulada por la etapa de adaptación externa, para trabajar con valores admisibles por el micro. El ADC del F28377D permite variar la $V_{ref\ superior}$, mientras que la $V_{ref\ inferior}$ se encuentra fijada a masa.

La lectura de datos se realiza físicamente por el puerto 09 de la *Docking Station*, correspondiente a la entrada 0 del módulo A del ADC. Toda esta nomenclatura puede ser complicada por lo que se facilitarán los manuales tanto del microprocesador como de la estación R4.1 con la que se ha trabajado.

El programa CCS es muy completo y específico para este tipo de procesadores, por lo que permite trazar muchas de las variables con las que se está trabajando. En la pantalla se pueden ver los valores que recoge el ADC y se trabajará posteriormente.

Finalmente, al decidir emular el caso de una conexión en la que se desconocen los valores de la carga, por lo que el esquema que se va a emplear es el siguiente:



Por último, se ha tenido que realizar la mencionada conversión de las funciones de transferencia a código C. Esto puede ser bastante simple, si se tiene clara la estructura de una función de transferencia en tiempo discreto. La estructura es la siguiente.

$$\frac{Y(z)}{E(z)} = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4} + a_5z^{-5} + a_6z^{-6} + a_7z^{-7}}{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + b_5z^{-5} + b_6z^{-6} + b_7z^{-7}}$$

Con $b_0 = 1$:

$$Y(z) = E(z)[a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4} + a_5z^{-5} + a_6z^{-6} + a_7z^{-7}] - Y(z)[b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + b_5z^{-5} + b_6z^{-6} + b_7z^{-7}]$$

Por lo tanto, el código empleado es el siguiente en ambas funciones de transferencia. Como se puede observar, se tienen que ir desplazando las variables de entrada y salida en función del momento en el que se encuentre el sistema.

Todas estas tareas serán llevadas a cabo dentro de la ISR del ADC, que será generada periódicamente por un *Timer* integrado en la CPU del micro.

Función de transferencia Vt y su desplazamiento de valores en función del estado.

```
////FUNCION DE TRANSFERENCIA VT
y_vt_k0 = num_Vt[0]*e_vt_k0 + num_Vt[1]*e_vt_k1 + num_Vt[2]*e_vt_k2 + num_Vt[3]*e_vt_k3 + num_Vt[4]*e_vt_k4 + num_Vt[5]*e_vt_k5 + num_Vt[6]*e_vt_k6
- den_Vt[1]*y_vt_k1 - den_Vt[2]*y_vt_k2 -den_Vt[3]*y_vt_k3 - den_Vt[4]*y_vt_k4 - den_Vt[5]*y_vt_k5 - den_Vt[6]*y_vt_k6;

e_vt_k6 = e_vt_k5;
e_vt_k5 = e_vt_k4;
e_vt_k4 = e_vt_k3;
e_vt_k3 = e_vt_k2;
e_vt_k2 = e_vt_k1;
e_vt_k1 = e_vt_k0;

y_vt_k6 = y_vt_k5;
y_vt_k5 = y_vt_k4;
y_vt_k4 = y_vt_k3;
y_vt_k3 = y_vt_k2;
y_vt_k2 = y_vt_k1;
y_vt_k1 = y_vt_k0;
```

Ilustración 49. FT de Vt y desplazamiento

Función de transferencia Vt y su desplazamiento de valores en función del estado. En este caso el valor de la entrada se actualiza con el dato tomado en el ADC

```
// FUNCION DE TRANSFERENCIA VZ
e_vz_k0 = AdcaResultRegs.ADCRESULT0;

y_vz_k0 = num_Vz[0]*e_vz_k0 + num_Vz[1]*e_vz_k1 + num_Vz[2]*e_vz_k2 + num_Vz[3]*e_vz_k3 + num_Vz[4]*e_vz_k4 + num_Vz[5]*e_vz_k5 + num_Vz[6]*e_vz_k6
- den_Vz[1]*y_vz_k1 - den_Vz[2]*y_vz_k2 -den_Vz[3]*y_vz_k3 - den_Vz[4]*y_vz_k4 - den_Vz[5]*y_vz_k5 - den_Vz[6]*y_vz_k6;

e_vz_k6 = e_vz_k5;
e_vz_k5 = e_vz_k4;
e_vz_k4 = e_vz_k3;
e_vz_k3 = e_vz_k2;
e_vz_k2 = e_vz_k1;
e_vz_k1 = e_vz_k0;

y_vz_k6 = y_vz_k5;
y_vz_k5 = y_vz_k4;
y_vz_k4 = y_vz_k3;
y_vz_k3 = y_vz_k2;
y_vz_k2 = y_vz_k1;
y_vz_k1 = y_vz_k0;
```

Ilustración 50.FT de Vz, desplazamiento y toma de dato

3.4 DAC externo (8420).

Ya hemos hablado de este componente en varias páginas. Se trata de un DAC de 16 bits, 12 para el tamaño del dato a convertir, 2 de modo de funcionamiento y 2 de transición. Esto se encuentra explicado de forma más detallada en la documentación perteneciente a este componente, que aparece en la bibliografía.

Se trata de un componente físico, integrado en un chip de 16 patillas. Estas patillas fijan las alimentaciones del componente, el rango de los valores de los datos de salida, el reloj que controla las conversiones, puerto de salida...



De este integrado externo deberemos tener en cuenta las siguientes conexiones.

- **V_{DD}**: Alimentación positiva. Desde los 5v hasta los 15v.
- **V_{OUT A}**: Se tomará esta salida. Deberá ser elegida mediante los 2 primeros bits en la cadena de transmisión del SPI con los valores 00.
- **V_{REF LO}**: Marcará el valor mínimo de salida. En nuestro caso próximo a los -3v. Puede aproximarse hasta a 2.5v del valor de alimentación negativa.
- **V_{REF HI}**: Marcará el valor máximo de salida. En nuestro caso próximo a los 3v. Puede aproximarse hasta a 2.5v del valor de alimentación positiva.
- **V_{SS}**: Alimentación negativa. Desde los -5v hasta los -15v.
- **GND**: Conectar a masa.
- **SDI**: Canal encargado de recibir los datos desde el micro.
- **CLK**: Señal de reloj, también procedente desde el F28377D

En función del modo de trabajo que se desee, se consultará la siguiente tabla:

Table 6. Control Function Logic Table

CLK ¹	\overline{CS} ¹	\overline{LD}	\overline{CLR}	CLSEL	Serial Input Shift Register	DAC Register A to DAC Register D
NC ²	High	High	Low	High	No change	Loads midscale value (0x800)
NC ²	High	High	Low	Low	No change	Loads zero-scale value (0x000)
NC ²	High	High	↑	High/Low	No change	Latches value
↑	Low	High	High	NC ²	Shifts register one bit	No change
Low	↑	High	High	NC ²	Shifts register one bit	No change
High	NC (↑) ²	↓	High	NC ²	No change	Loads the serial data-word ³
High	NC ²	Low	High	NC ²	No change	Transparent ⁴
NC ²	High	High	High	NC ²	No change	No change

¹ CLK and \overline{CS} are interchangeable.

² NC = Don't Care.

³ Returning \overline{CS} high while CLK is high avoids an additional false clock of serial input data. CLK and \overline{CS} are interchangeable.

⁴ Do not clock in serial data while \overline{LD} is low.

En el caso desarrollado se ha elegido la siguiente configuración:

- **CS:** Conexión a masa
- **LD:** Conexión a Vcc
- **CLSEL:** Sin conexión

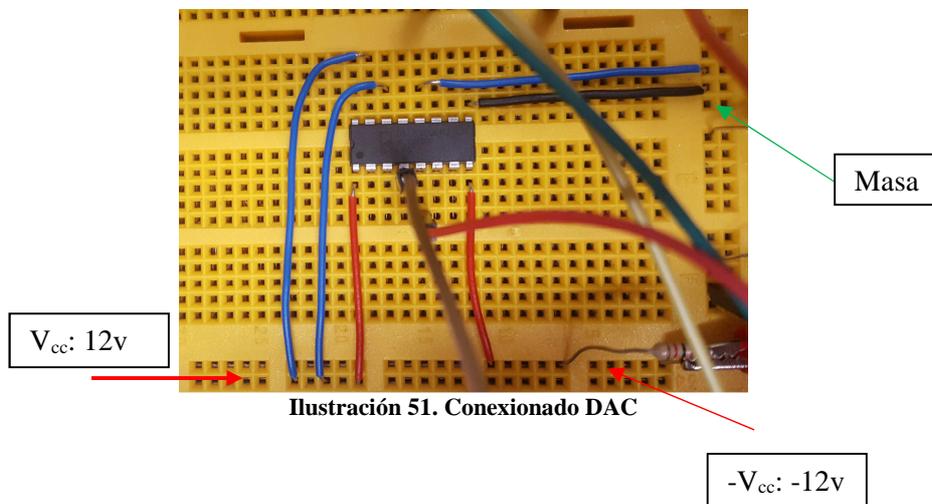


Ilustración 51. Conexión DAC

3.5 Etapa de adaptación y conexionado.

Esta etapa ha sido diseñada y fabricada por integrantes del grupo GEISER de la Universidad de Alcalá de Henares. Simplemente se trata de un prototipo, previo a la fabricación de una placa a medida para el procesador que contará con esta etapa de adaptación integrada.

La placa empleada consta de 4 módulos para poder montar hasta 4 operacionales LM741, pudiendo atenuar 4 señales al mismo tiempo. No solo atenúan la señal recibida, sino que también la limpian de posible ruido.

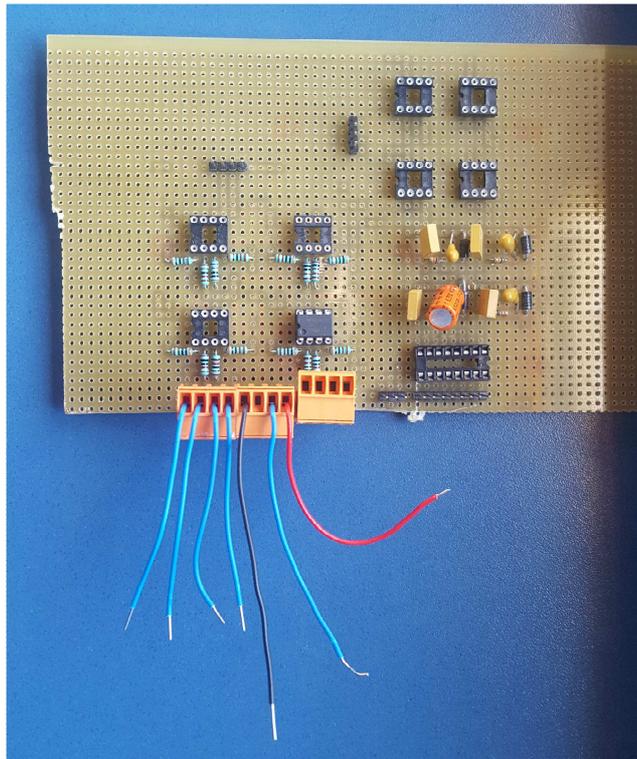


Ilustración 52. Etapa de adaptación

Esta etapa irá colocada justo después de los sensores que tiene la fuente trifásica, evitando cualquier problema de voltajes no aptos para el microprocesador. La salida del canal usado se conectará de forma directa al pin de entrada del ADC del F28377D. En el caso de que se desee trabajar con una señal trifásica, se deberán configurar varias entradas en el ADC.

3.6 Fuente trifásica controlable

Se va a usar una fuente trifásica programable. Concretamente se va a hacer uso de la fuente “*Pacific Smart Source 345-AMX*”. Esta fuente reúne las características necesarias para la implementación final del sistema. Presenta un puerto de entrada de tensión analógica que permite controlar la tensión de salida de la fuente. Por otro lado, es posible medir la corriente por la carga gracias a unos sensores de *efecto Hall*.

Esta fuente presenta las siguientes características:

Standard Features:

- Advanced Linear Amplifiers Provide Very Low Voltage Distortion, no Switching Noise, Fast Voltage and Current Slew Rates, Exceptionally Low Output Impedance and High Peak Current Capability
- 1, 2, or 3 Phase Output Form selectable from front panel or bus command
- 20 to 5,000 Hz. Full Power Bandwidth Operation – 5Hz to 50KHz small signal bandwidth, 3dB at 10% of full voltage
- Precision Voltage Programming – 0.05% with Continuous Self-Calibration (CSC) engaged
- True-RMS Metering of Volts, Amps, and Power
- GPIB (IEEE-488.2) or RS-232 Interface
- Waveform Library – Arbitrary Waveform Generator
- Up to 99 Programs with Associated Transients for Static and Dynamic Test Applications
- UPC Studio Software Suite

Ilustración 53. Características técnicas de la fuente

Se dejará en la bibliografía la referencia al manual completo para una mejor comprensión de las características.



Model 345AMX

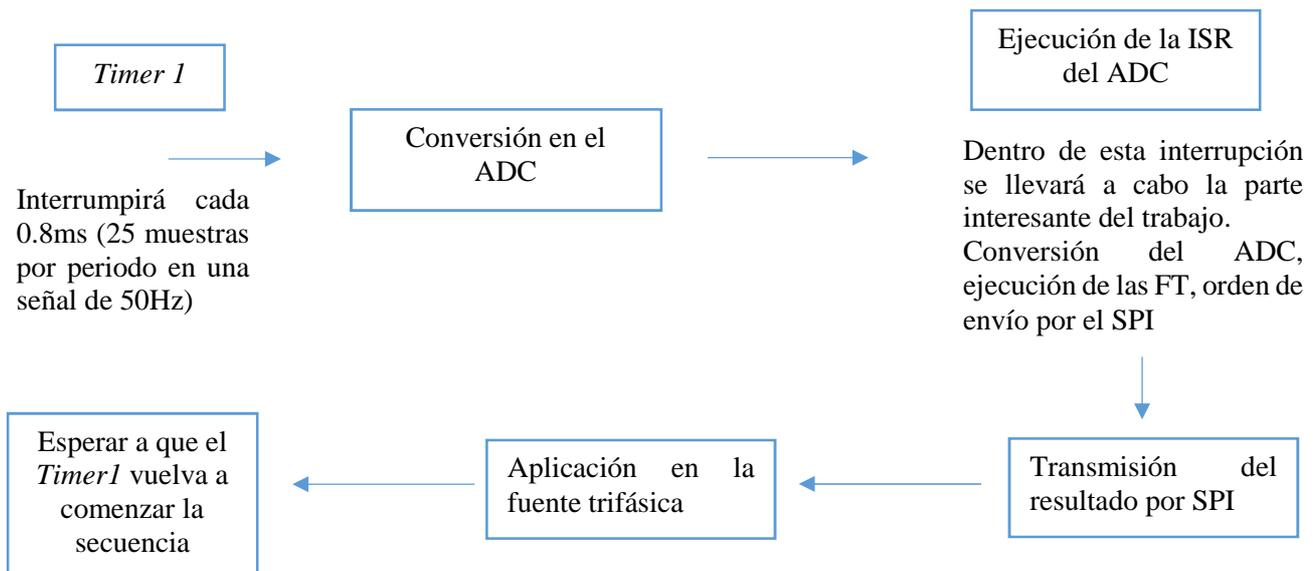
Ilustración 54. Fuente trifásica modelo

4. Diseño aplicación C.

4.1. Estructura de la aplicación

Para poder completar la implementación en el microprocesador, deberá ser necesario el uso de código C.

En este entorno se deberá programar una rutina de interrupción periódica que habilite la toma y conversión de datos en el ADC. Esto se realizará mediante un *Timer* interno de la CPU en concreto el *Timer1*. Para facilitar la tarea, se va a mostrar un pequeño esquema para entender el flujo de trabajo.



4.2. Configuración general

El trabajo en conjunto empleará la siguiente estructura.

Primero. El ADC es controlado por un *Timer* de la CPU, que le manda convertir el dato de vuelta de la CPU cada un cierto tiempo que se valorará más adelante. Para esto se deberán configurar tanto *Timer* como ADC.

Segundo. Una vez convertido el dato, saltará la rutina ISR del ADC. Dentro de esta rutina se debe emular transmisión de la línea. Esto se realizará mediante 2 funciones de transferencia. La primera perteneciente al sistema en vacío y la segunda al sistema cargado a una terminada corriente que se estará leyendo desde la fuente trifásica y siendo convertida por el ADC cada vez que salte la interrupción. Para poder realizar todo esto se ha debido de configurar correctamente el ADC y programar correctamente las funciones de transferencia como se ha mostrado en los apartados anteriores.

Tercero. Una vez concluido el paso 2, es decir, se han ejecutado correctamente las funciones de transferencia y se han sumado sus valores. El valor obtenido deberá transmitirse por el SPI. El SPI también deberá ser configurado previamente.

Importante que toda la rutina que debe realizar la interrupción se ejecute en menos de 0.8ms, ya que en ese tiempo volverá a recibir la llamada del *Timer1*.

4.2.1. ADC

Se configurará el ADC para que periódicamente convierta el dato que se esté recibiendo de vuelta desde la fuente de tensión trifásica. Esta configuración deberá ser a una frecuencia adecuada. No deberá ser una frecuencia muy rápida ya que se deberá dejar tiempo suficiente para que se pueda ejecutar la ISR. Para la conjuración de esto se realizará de la siguiente manera.

Se deberán enlazar el *Timer1* y el ADC y se configurarán los campos necesarios, como por ejemplo canal de recepción del ADC, ventana de adquisición, SOC... Por lo que se emplearán las siguientes líneas:

```
void ConfigureADC(void);
void SetupADCTimer1(Uint16 channel);
interrupt void adca1_isr(void);
```

Ilustración 55. Funciones de configuración

Dentro de estas funciones se encontrará la siguiente configuración.

Configuración del ADC:

```
id ConfigureADC(void)
    EALLOW;

    //
    //write configurations
    //
    AdcaRegs.ADCCTL2.bit.PRESCALE = 6; //set ADCCLK divider to /4
    AdcSetMode(ADC_ADCA, ADC_RESOLUTION_12BIT, ADC_SIGNALMODE_SINGLE);

    //
    //Set pulse positions to late
    //
    AdcaRegs.ADCCTL1.bit.INTPULSEPOS = 1;

    //
    //power up the ADC
    //
    AdcaRegs.ADCCTL1.bit.ADCPWDNZ = 1;

    //
    //delay for 1ms to allow ADC time to power up
    //
    DELAY_US(1000);

    EDIS;
```

Ilustración 56. Configuración de ADC

Configuración de la sincronización Timer1 – ADC:

```
void SetupADTimer1(Uint16 channel)
{
    Uint16 acqps;

    //
    //determine minimum acquisition window (in SYSCLKS) based on resolution
    //
    if(ADC_RESOLUTION_12BIT == AdcaRegs.ADCCTL2.bit.RESOLUTION)
    {
        acqps = 14; //75ns
    }
    else //resolution is 16-bit
    {
        acqps = 63; //320ns
    }

    //
    //Select the channels to convert and end of conversion flag
    //
    EALLOW;
    AdcaRegs.ADCSOC0CTL.bit.CHSEL = channel; //SOC0 will convert pin A0
    AdcaRegs.ADCSOC0CTL.bit.ACQPS = acqps; //sample window is 100 SYSCLK cycles
    AdcaRegs.ADCSOC0CTL.bit.TRIGSEL = 2; // TIMER1
    AdcaRegs.ADCINTSELIN2.bit.INT1SEL = 0; //end of SOC0 will set INT1 flag
    AdcaRegs.ADCINTSELIN2.bit.INT1E = 1; //enable INT1 flag
    AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //make sure INT1 flag is cleared
    EDIS;
}
```

Ilustración 57. Configuración ADC-Timer1

4.2.2. SPI

El SPI solamente enviará datos cuando sea conveniente. Es decir, dentro de la ISR, una vez se hayan realizado los cálculos necesarios.

```
}
void spi_xmit(Uint16 a)
{
    SpiaRegs.SPITXBUF = a;
}

void spi_fifo_init()
{
    //
    // Initialize SPI FIFO registers
    //
    SpiaRegs.SPIFFTX.all = 0xE040;
    SpiaRegs.SPIFFRX.all = 0x2044;
    SpiaRegs.SPIFFCT.all = 0x0;

    //
    // Initialize core SPI registers
    //
    InitSpi();
}
```

Ilustración 58. Configuración del SPI

4.2.3 PIE

La interrupción periódica ha sido configurada mediante un *Timer* que fuerza al ADC a la toma de un dato cada cierto tiempo.

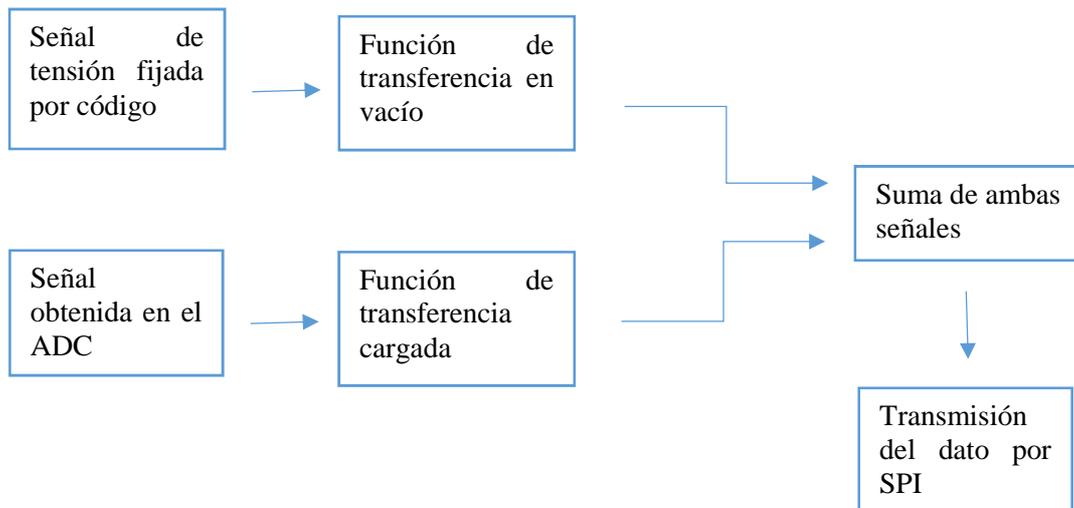
```
InitCpuTimers(); // For this example, only initialize the Cpu Timers
//60MHz CPU Freq, 50 millisecond Period (in uSeconds)
ConfigCpuTimer(&CpuTimer1, 60, 500000);
```

Ilustración 59. Interrupción periódica mediante Timer1

4.3. Rutina de atención a la interrupción

La rutina de atención a la interrupción va a ser controlada por el *Timer*. El *Timer* saltará periódicamente. La conversión del dato se realizará a través del puerto 9 de la *Docking Station*, correspondiente con la entrada 0 del ADC del bloque A. El ADC notificará cuando ha terminado la conversión, activando una señal que hará activar su función ISR.

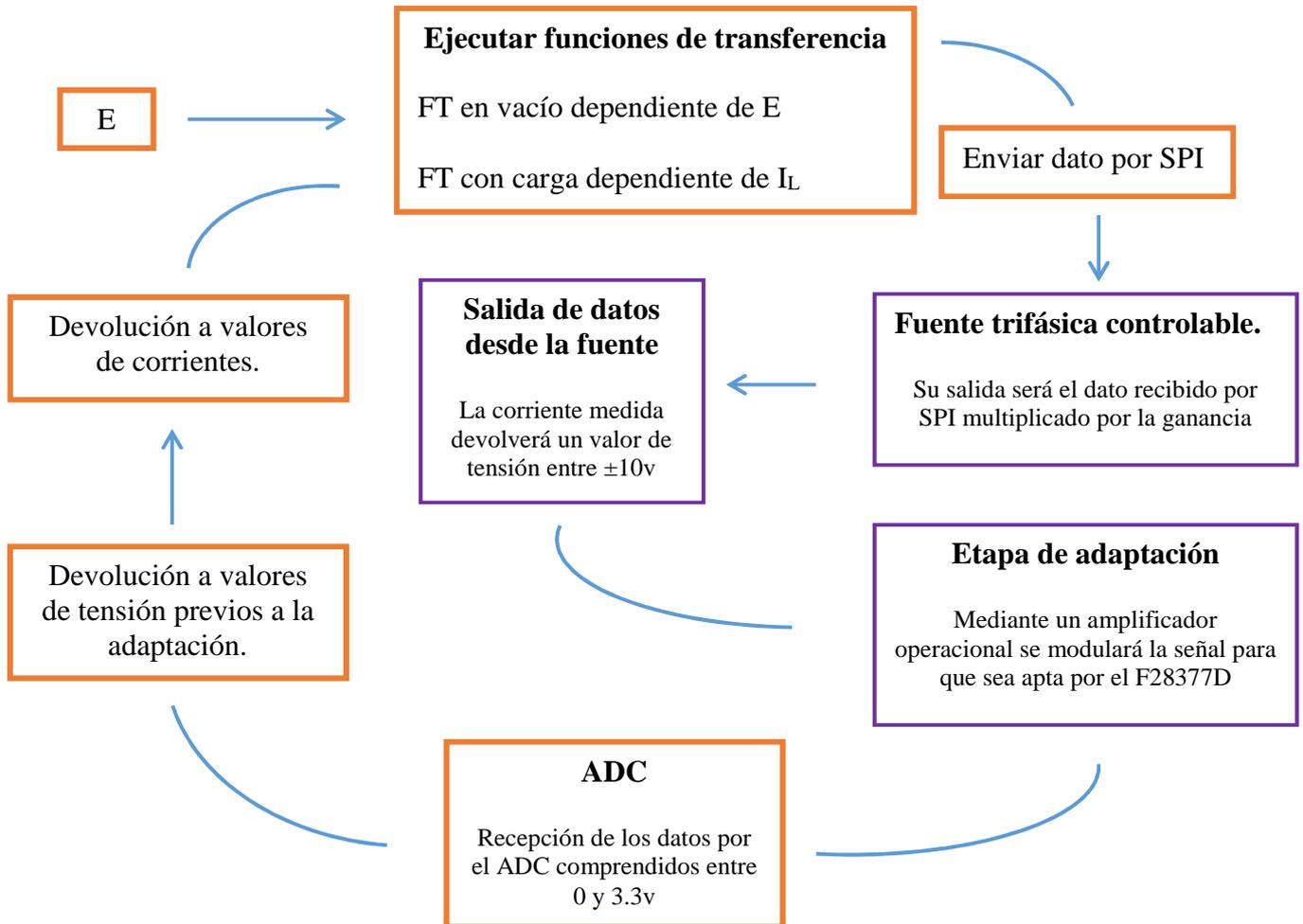
Dentro de esta función ISR se debe realizar la mayor parte del trabajo. En primer lugar empleando las funciones de transferencia, se deberán obtener los siguientes resultados.



El diagrama mostrado, únicamente representa la parte que deberá realizar la función de interrupción cuando el ADC ha terminado con la conversión del dato.

4.3.1. Diagrama de flujo

El diagrama completo de trabajo es el siguiente. Los recuadros en naranja representarán funciones realizadas dentro del microprocesador mientras que los recuadros morados son externos al micro.



A lo largo de estas páginas se han ido mostrando partes de este diagrama de conjunto, profundizando en algunas de las cajas presentes en él.

De este diagrama nos centraremos en varios datos. El encargado de dar comienzo a esta secuencia es un *Timer* que le pide al ADC la toma de datos.

El tiempo de ejecución no debe ser mayor que el periodo de llamada del *Timer*

4.3.2. Calibración de sensores y seguridad

Ante cualquier señal a recibir y cualquier señal a enviar se deberá testear que los valores sean adecuados y se encuentren dentro de unos márgenes admisibles por seguridad y correcto funcionamiento. Esto se deberá realizar en la lectura recibida por el ADC antes de pasar los datos a las funciones de transferencia y en el dato a enviar por *SPI*.

5. Análisis Experimental. Caso de estudio

5.1. Análisis del caso.

En este apartado se explicará paso a paso el proceso a seguir para poder implementar todo lo mencionado en las líneas anteriores.

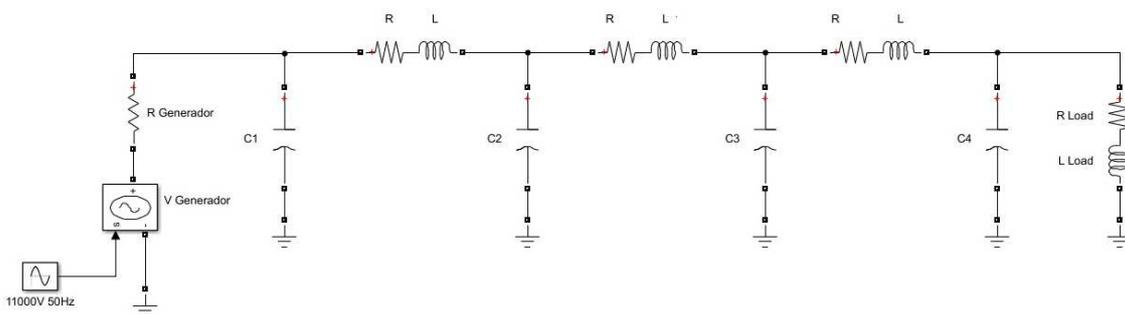
En primer lugar, se separará el entorno de simulación del entorno de emulación física en la fuente de alimentación real. En el primer caso se necesitaría únicamente un ordenador con las prestaciones adecuadas para poder usar MATLAB de forma correcta. En cambio, para el segundo caso, seguirá siendo necesario el uso de MATLAB en su entorno *Simulink*, más el entorno físico que se necesita, siendo este:

- Micro F28377D C2000
- DAC externo 8420
- Fuente trifásica controlable
- Etapas de adaptación de señal.

Entorno de simulación:

Como se ha hablado a lo largo de este trabajo la plataforma principal de trabajo es *Simulink*, sin embargo, será muy recomendable el trabajo con *Scripts* (ficheros de código MATLAB *m*) para poder cambiar de una forma más rápida y fácil los valores de los componentes presentes en el archivo *Simulink*.

Para comenzar se debe decidir cuantos segmentos en π se van a querer introducir en el análisis. En el ejemplo que se va a desarrollar se van a emplear 3 segmentos, pero este número puede ser el que se desee.



Los elementos empleados hasta este punto solo son:

- **Sine Wave.** Se ha elegido este bloque por comodidad. Se puede elegir cualquier otro bloque, para introducir la señal que se desee introducir en la simulación del circuito.

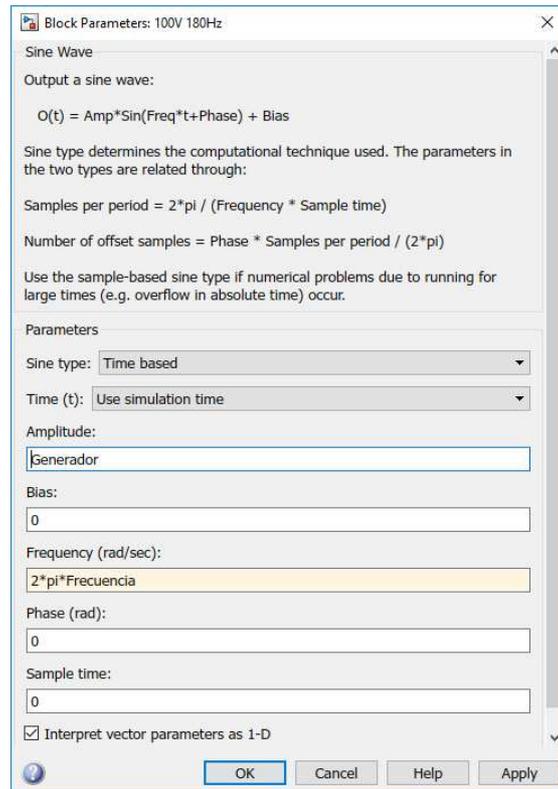
En el bloque se pueden fijar los siguientes valores. Como se ha dicho los valores son parametrizados.

“Generador” es la tensión de pico que se desea para la señal senoidal.

La frecuencia debe ser introducida en radianes/segundo, por lo que una forma de introducir dicho valor es $2 \cdot \pi \cdot \text{Frecuencia}$.

El valor de π se fijará con el número de decimales que se crea oportuno, y la frecuencia se encontrará fijada en hercios.

```
Generador = 11000; %V
pi = 3.1415;
Frecuencia = 50; % Hz
```



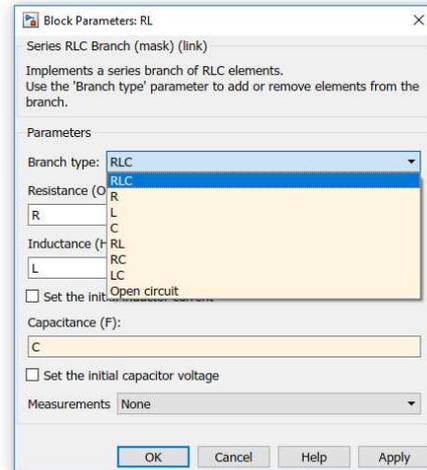
- **Controlled Voltage Source.** Este bloque convierte la señal, para ser interpretada como un voltaje. El bloque no tiene ningún parámetro que deba ser modificado, por lo que no se mostrará su bloque.
- **Series RLC Branch.** Es el bloque que simula resistencias, bobinas y condensadores. No es necesario que los 3 elementos se encuentren presentes, porque se seleccionaran los deseados.

Ya se ha mostrado anteriormente la estructura que se emplea para simular la línea. Por lo que se deberán fijar los parametros de la longitud, el material del cable, el material aislante, con lo que se obtendrán los valores de la R_{eq} , L_{eq} y C_{eq} de la línea.

Estos valores, tienen que ser divididos entre el numero de segmentos en π que se desea introducir en la línea.

En el caso de los condensadores, el primer condensador (en paralelo con el generador y la R_g) y el último (en paralelo con la carga), deben tener el el valor de $C/2$.

Las ecuaciones empleadas para calcular los valores equivalentes de la línea son las siguientes.



○ **Resistencia equivalente:**

La resistancia estimada en corriente alterna depende de la estimada en corriente continua.

La ecuacion ya mostrada en el apartado de “Base teorica”:

$$R_{DC\ eq} = \frac{\rho_{20} \cdot l}{A} [1 + \alpha(\theta - 20)]$$

La resistividad y el coeficiente de temperatura dependen del materia empleado por los que el usuario debiera informarse sobre estos valores. En el caso que se va a desarrollar se ha elegido como material el cobre.

$$\rho_{20^{\circ}C\ Cobre} = 4.01786 \left[\frac{\Omega}{m} \right]$$

$$\alpha_{Cobre} = 3.93 \cdot 10^{-3} \left[\frac{1}{^{\circ}C} \right]$$

$$l = 80\ Km$$

$$A = 35\ mm^2$$

Obteniendo unra $R_{DC\ eq} = 11.71\ \Omega$

Esta $R_{DC\ eq}$ se complementa con otros parametros para poder obtener la $R_{AC\ eq}$.

$$R_{AC\ eq} = R_{DC\ eq} \cdot [1 + 1.5 \cdot (Y_s + Y_p)]$$

Los valores de Y_s e Y_p , son dependientes de valores como la sección del cable. que también deberá ser fijada, junto con la distancia respecto a otros conductores cercanos...

En este caso no muestran valores significativos por lo que se puede considerar que $R_{AC\ eq} = R_{DC\ eq}$

Con todo lo mencionado, la resistencia que presentara cada segmento de la línea será la siguiente:

$$R = \frac{R_{AC\ eq}}{\text{Número de segmentos}} = \frac{11.71}{3} = 3.9034\Omega \text{ (por segmento)}$$

○ Inductancia equivalente

Los valores de la inductancia equivalente y capacidad equivalente son mucho más complicados de estimar que el valor de la resistencia equivalente. Esto se debe a que dependen de más factores y a su vez más difíciles de determinar con exactitud (permitividad del aislante, distancia y tamaño de otros conductores...)

Por esto, se ha aceptado como buena la siguiente expresión de L_{eq} ya que como se vió en el apartado número 2 de este trabajo, se aproximaba bastante a los valores que aportaba la compañía especializada en líneas de transmisión *Nexans*.

$$L_1 = \left(1 + 2 \ln\left(\frac{D}{r_1}\right)\right) \times 10^{-7} \left[\frac{H}{m}\right]$$

Con lo que se ha obtenido el siguiente resultado:

$$L_{eq} = 8.9\ mH$$
$$L_{eq\ por\ segmento} = 8.9\ mH$$

○ **Capacidad equivalente**

Para el condensador equivalente de la línea se ha dado por buena la siguiente expresión. Hay que decir que hay muchos libros que profundizan en los parámetros equivalentes de líneas de transmisión, obteniendo distintas expresiones en función del entorno de trabajo deseado. En nuestro caso se ha comprobado que el valor obtenido por unidad de longitud es muy similar al que proporciona el manual de NEXANS.

$$C = \frac{2 \cdot \pi \cdot \epsilon_r \cdot \epsilon_0}{\ln\left(\frac{D_{eq}}{r}\right)} \text{ (por unidad de longitud)}$$

En este caso, se deberá fijar la permitividad del aislante, la distancia a la que se encuentra el conductor de otras líneas respecto en relación a su propio radio.

En el caso que nos atiende, el producto de las permitividades (ϵ), se ha fijado en $72 \cdot 10^{-9} \mu\text{F}/\text{Km}$. Esto se ha hecho, otra vez, comparando con el valor presente en la tabla del fabricante. Se trata de un valor realmente bueno, ya que como se ve en las gráficas del apartado 2.3 ambas son muy similares.

Por lo tanto en nuestra línea tendrá una capacidad equivalente de:

$$C_{eq} = \frac{C \cdot l}{\text{Número de segmentos}}$$

En el caso que se está simulando el resultado es:

$$C_{eq} = 6.738 \mu\text{F}$$

En resumen, los valores finales de la línea que se va a simular son los siguientes:

$$\begin{aligned} \text{longitud} &= 80\text{Km} \\ \text{temperatura} &= 90^\circ\text{C} \\ \epsilon &= 72 \mu\text{F}/\text{Km} \end{aligned}$$

$$\begin{aligned} R_{eq} &= 3.9034\Omega \\ L_{eq} &= \\ C_{eq} &= 6.738 \mu\text{F} \end{aligned}$$

Con estos valores, las corrientes y tensiones presentes en el circuito son las siguientes, empleando el siguiente esquema.

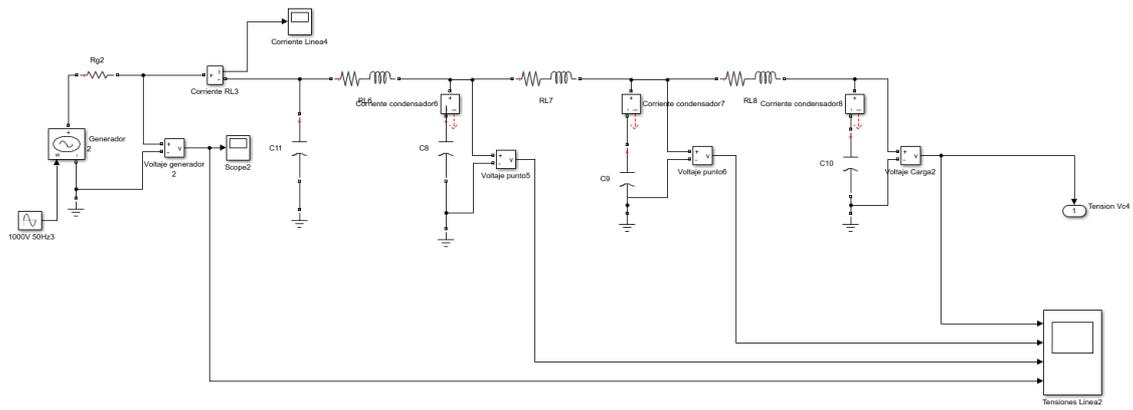


Ilustración 60. Diagrama en vacío

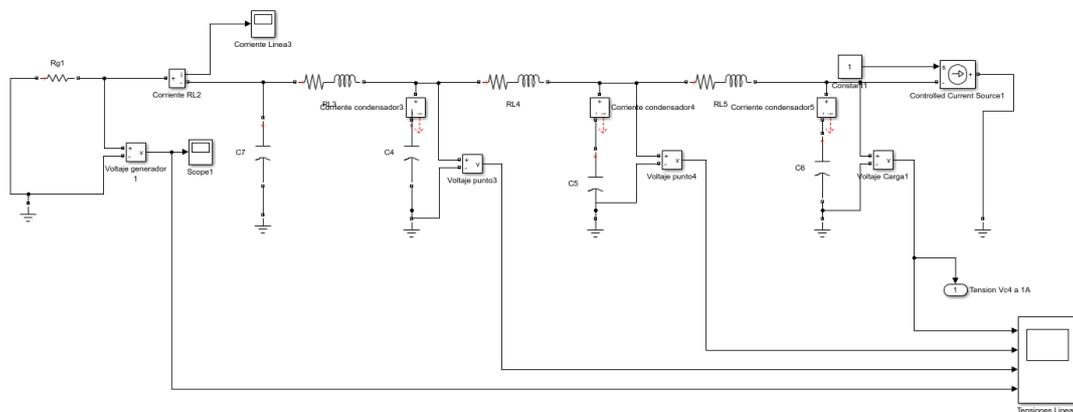
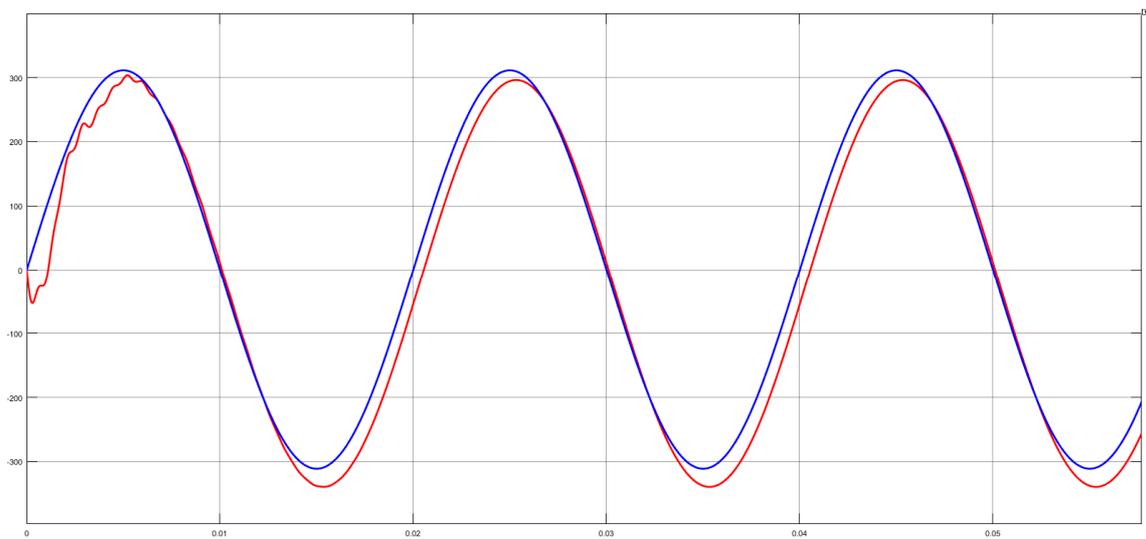


Ilustración 61. Diagrama consumiendo 1ª

Obteniendo los siguientes resultados en tiempo continuo.



Entorno de emulación:

- Función de transferencia

Para poder volcar el sistema en el micro, lo primero que se debe hacer es pasar el circuito equivalente a variables de estado. Esto ya ha sido explicado en puntos anteriores, por lo que se empleará la matriz de conversión ya mostrada.

Por lo tanto, se emplearán las matrices mostradas anteriormente.

En vacío:

$$\begin{pmatrix} \dot{I}_A \\ \dot{I}_B \\ \dot{I}_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} \\ -\frac{1}{C_1} & 0 & 0 & -\frac{1}{R_g * C_1} & 0 & 0 & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_3} & -\frac{1}{C_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_4} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} I_L$$

$$y = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix}$$

En carga:

$$\begin{pmatrix} \dot{I}_A \\ \dot{I}_B \\ \dot{I}_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} = \begin{pmatrix} -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & 0 & 0 & \frac{1}{L} & -\frac{1}{L} \\ -\frac{1}{C_1} & 0 & 0 & -\frac{1}{R_g * C_1} & 0 & 0 & 0 \\ \frac{1}{C_2} & -\frac{1}{C_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_3} & -\frac{1}{C_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_4} & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{C_4} \end{pmatrix} I_L$$

$$y = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1) \begin{pmatrix} I_A \\ I_B \\ I_C \\ V_{c1} \\ V_{c2} \\ V_{c3} \\ V_{c4} \end{pmatrix}$$

En este caso, no es necesario emplear la matriz genérica ya que se ha justificado explícitamente las matrices para un circuito equivalente de 3 segmentos.

Sustituyendo los valores de las resistencias (R_g , R_{eq}), bobinas y condensadores, se obtiene la siguiente matriz:

```
A3_sis_vacio =
1.0e+05 *
-0.0047      0      0      0.0012  -0.0012      0      0
      0  -0.0047      0      0      0.0012  -0.0012      0
      0      0  -0.0047      0      0      0.0012  -0.0012
-2.9683      0      0  -0.2968      0      0      0
 1.4842  -1.4842      0      0      0      0      0
      0  1.4842  -1.4842      0      0      0      0
      0      0  2.9683      0      0      0      0

>> B3_sis_vacio
B3_sis_vacio =
1.0e+04 *
      0
      0
      0
 2.9683
      0
      0
      0

>> C3_sis_vacio
C3_sis_vacio =
      0      0      0      0      0      0      1
```

Ilustración 62. Valores numéricos matrices sistema en vacío

```
A3_sis_cargado =
1.0e+05 *
-0.0047      0      0      0.0012  -0.0012      0      0
      0  -0.0047      0      0      0.0012  -0.0012      0
      0      0  -0.0047      0      0      0.0012  -0.0012
-2.9683      0      0  -0.2968      0      0      0
 1.4842  -1.4842      0      0      0      0      0
      0  1.4842  -1.4842      0      0      0      0
      0      0  2.9683      0      0      0      0

>> B3_sis_cargado
B3_sis_cargado =
1.0e+05 *
      0
      0
      0
      0
      0
      0
      0
-2.9683

>> C3_sis_cargado
C3_sis_cargado =
      0      0      0      0      0      0      1
```

Ilustración 63. Valores numéricos matrices sistema cargado

A partir de estas matrices, se puede sacar la función de transferencia del sistema. Esto se puede realizar mediante código MATLAB una vez han sido guardadas las matrices en variables de estado.

%% PASO A FUNCION DE TRANSFERENCIA

```
linea=ss(A3,B3,C3,D);
[num,den] = ss2tf(A3,B3,C3,D);
linea_FT = tf(num,den);
```

Una vez se tiene la función de transferencia, se debe elegir un método para pasar a tiempo discreto el sistema. Para esto existen varios métodos como se ha mostrado anteriormente. El usuario podrá usar el que más le convenga, eligiendo entre uno de los siguientes:

%% METODOS DE DISCRETIZACION

```
linea_discreta_impulse=c2d(linea,Ts,'impulse');
linea_discreta_zoh=c2d(linea,Ts,'zoh');
linea_discreta_foh=c2d(linea,Ts,'foh');
linea_discreta_tustin=c2d(linea,Ts,'tustin');
linea_discreta_matched=c2d(linea,Ts,'matched');
```

En el caso que se está desarrollando, se va a elegir el método de *Tustin*. Obteniendo las siguientes funciones de transferencia.

```
linea_FT_vacio_tustin =
  2.238e-11 z^7 + 1.567e-10 z^6 + 4.7e-10 z^5 + 7.833e-10 z^4 + 7.833e-10 z^3 + 4.7e-10 z^2 + 1.567e-10 z + 2.238e-11
  -----
  z^7 - 6.714 z^6 + 19.3 z^5 - 30.81 z^4 + 29.47 z^3 - 16.9 z^2 + 5.374 z - 0.7314

linea_FT_cargada_tustin =
  -1.483 z^7 + 6.996 z^6 - 11.69 z^5 + 5.388 z^4 + 7.373 z^3 - 11.3 z^2 + 5.804 z - 1.085
  -----
  z^7 - 6.714 z^6 + 19.3 z^5 - 30.81 z^4 + 29.47 z^3 - 16.9 z^2 + 5.374 z - 0.7314
```

Esta función de transferencia será fundamental para la emulación de la línea.

La función de transferencia tiene que ser convertida a código .C para poder ser volcada mediante CCS al microprocesador. Su código es el siguiente:

```

///FUNCION DE TRANSFERENCIA VT
y_vt_k0 = num_Vt[0]*e_vt_k0 + num_Vt[1]*e_vt_k1 + num_Vt[2]*e_vt_k2 + num_Vt[3]*e_vt_k3 + num_Vt[4]*e_vt_k4 + num_Vt[5]*e_vt_k5 + num_Vt[6]*e_vt_k6
          - den_Vt[1]*y_vt_k1 - den_Vt[2]*y_vt_k2 - den_Vt[3]*y_vt_k3 - den_Vt[4]*y_vt_k4 - den_Vt[5]*y_vt_k5 - den_Vt[6]*y_vt_k6;

e_vt_k6 = e_vt_k5;
e_vt_k5 = e_vt_k4;
e_vt_k4 = e_vt_k3;
e_vt_k3 = e_vt_k2;
e_vt_k2 = e_vt_k1;
e_vt_k1 = e_vt_k0;

y_vt_k6 = y_vt_k5;
y_vt_k5 = y_vt_k4;
y_vt_k4 = y_vt_k3;
y_vt_k3 = y_vt_k2;
y_vt_k2 = y_vt_k1;
y_vt_k1 = y_vt_k0;

```

Ilustración 64. Función de transferencia Vt en código.

```

// FUNCION DE TRANSFERENCIA VZ
e_vz_k0 = AdcaResultRegs.ADCRESULT0;

y_vz_k0 = num_Vz[0]*e_vz_k0 + num_Vz[1]*e_vz_k1 + num_Vz[2]*e_vz_k2 + num_Vz[3]*e_vz_k3 + num_Vz[4]*e_vz_k4 + num_Vz[5]*e_vz_k5 + num_Vz[6]*e_vz_k6
          - den_Vz[1]*y_vz_k1 - den_Vz[2]*y_vz_k2 - den_Vz[3]*y_vz_k3 - den_Vz[4]*y_vz_k4 - den_Vz[5]*y_vz_k5 - den_Vz[6]*y_vz_k6;

e_vz_k6 = e_vz_k5;
e_vz_k5 = e_vz_k4;
e_vz_k4 = e_vz_k3;
e_vz_k3 = e_vz_k2;
e_vz_k2 = e_vz_k1;
e_vz_k1 = e_vz_k0;

y_vz_k6 = y_vz_k5;
y_vz_k5 = y_vz_k4;
y_vz_k4 = y_vz_k3;
y_vz_k3 = y_vz_k2;
y_vz_k2 = y_vz_k1;
y_vz_k1 = y_vz_k0;

```

Ilustración 65. Función de transferencia Vz en código.

- Generación de la señal digital y envío por SPI

El DAC8420 trabaja con 16 bits recibidos por un puerto *SPI*. De estos 16 bits se dividen tres. Los primeros dos bits seleccionan cuál de los cuatro DAC internos que tiene el integrado, se desea que sea el que trabaje. Seguidos a estos, se tienen dos bits de transición, que darán paso al tamaño de palabra para convertir, que son 12 bits, por lo que podemos hablar de que solo tiene 14 bits útiles.

Estos bits van se transmiten en forma MSB, es decir el primer bit del tamaño de palabra es el de mayor peso en la transmisión.

Parte del código que es necesario es el siguiente:

```
// FUNCIONES DEL SPI

void spi_xmit(Uint16 a)
{
    SpiaRegs.SPITXBUF = a;
}

void spi_fifo_init()
{
    //
    // Initialize SPI FIFO registers
    //
    SpiaRegs.SPIFFTX.all = 0xE040;
    SpiaRegs.SPIFFRX.all = 0x2044;
    SpiaRegs.SPIFFCT.all = 0x0;

    //
    // Initialize core SPI registers
    //
    InitSpi();
}
```

Ilustración 66. Funciones de configuración SPI

Para transmitir el dato, únicamente será necesaria la siguiente línea:

```
sdata = y_total;
spi_xmit(sdata);
```

Ilustración 67. Función para enviar el dato por SPI

Algunas de las transmisiones realizadas comprobando el funcionamiento fueron las siguientes:

Visualización de la transmisión de un dato y el CLK del SPI.

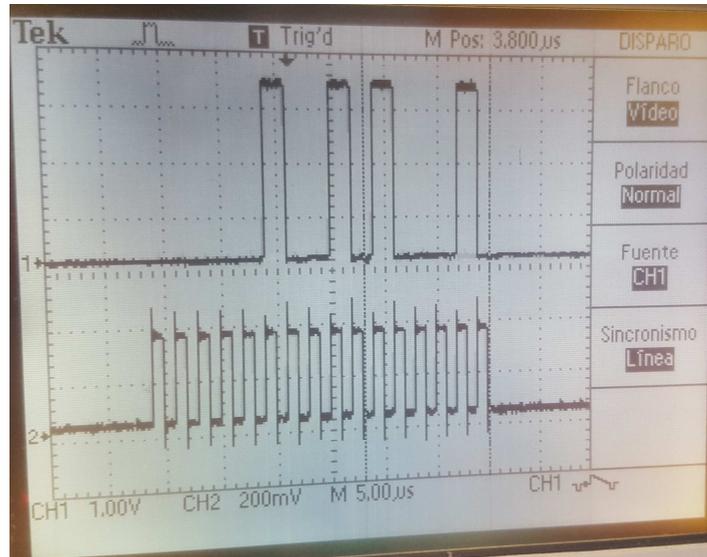


Ilustración 68. Transmisión de dato por SPI

Transmisión de un dato con la función del SPI ya integrada dentro de la rutina de interrupción generada por el *Timer1*.

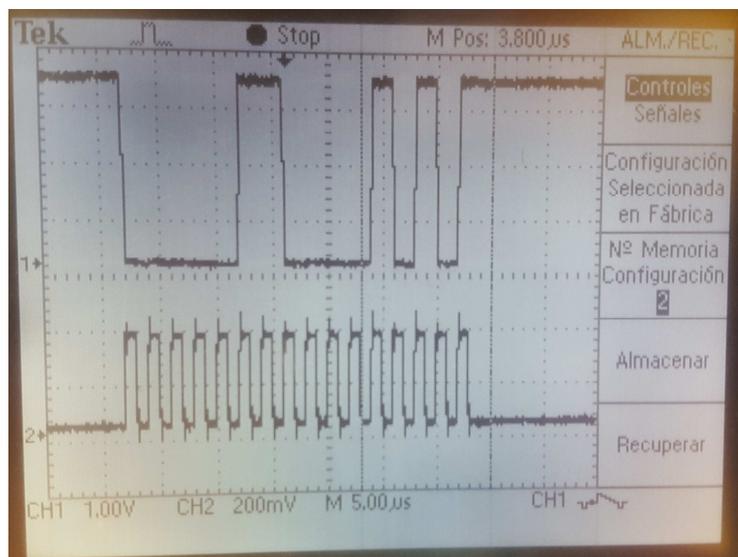


Ilustración 69. Dato por SPI

Por último mostrar una imagen del sistema completo montado:

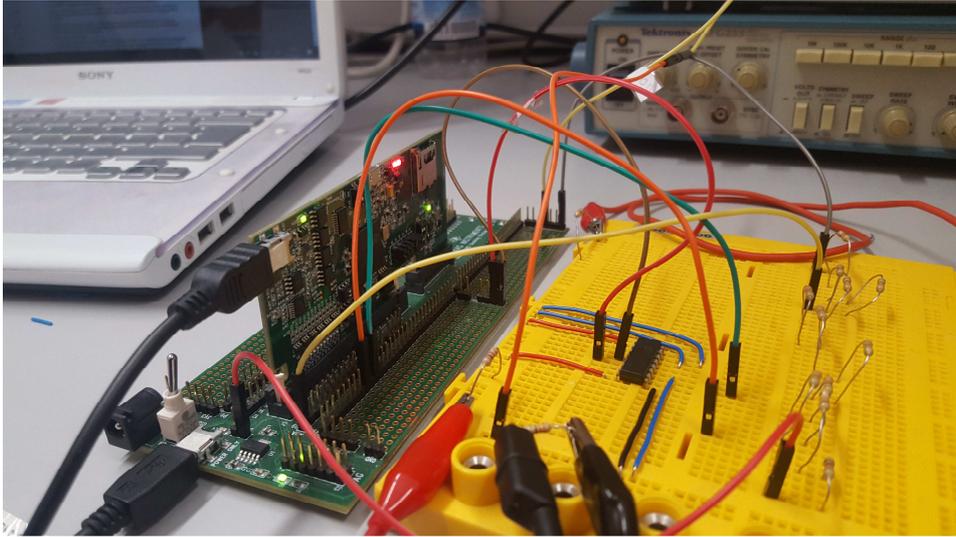


Ilustración 70. Sistema total montado.

- **Adaptación de la tensión medida y ADC**

Como se ha ido mencionando a lo largo de este trabajo, la señal que se recibe de vuelta desde la fuente trifásica no es una señal válida para el microprocesador que se está empleando, por lo que emplearemos una etapa de adaptación. Esta etapa de adaptación no será siempre necesaria.

Esta etapa, simplemente cuenta con un amplificador operacional, en concreto el LM741, del que se dejarán sus hojas características más adelante.

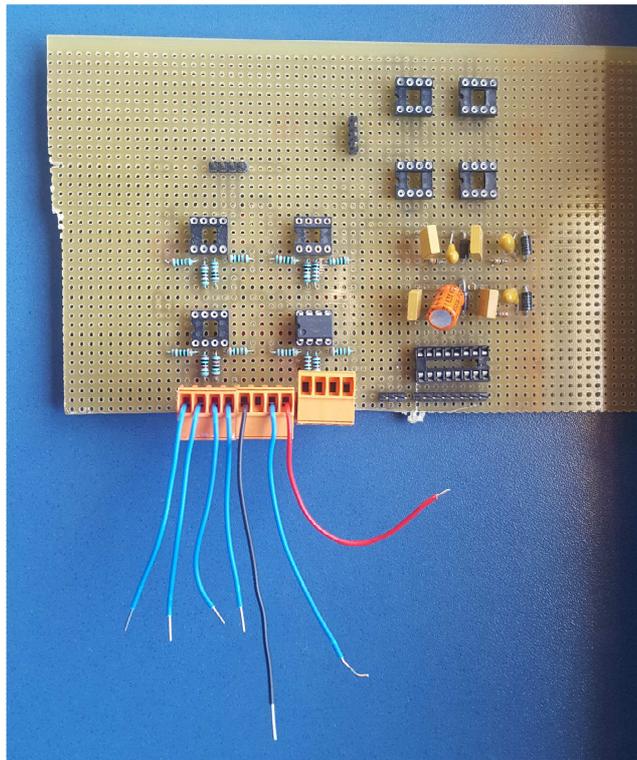


Ilustración 71. Etapa de adaptación provisional

Como se puede apreciar en la imagen, se ha diseñado esta placa provisional para poder trabajar hasta con 4 señales. En nuestro caso solo serán necesarias como mucho hasta 3 de ellas. También cuenta con una etapa de filtrado, para obtener una señal mucho más limpia de posible ruido.

5.2 Equipo experimental

El equipo necesario será el siguiente:

- Ordenador capaz de ejecutar *Code Composer Studio*.
- Microprocesador Texas Instruments F28377D.
- Placa externa de conexiones donde conectar el DAC8420.
- Osciloscopio para el seguimiento de señales.
- Fuente de alimentación capaz de entregar entre $\pm 12\text{v}$ para la alimentación del DAC externo.
- Fuente trifásica controlable.

5.3 Análisis en tiempo real. Medida de tiempos

Todas estas funciones descritas, deben ser realizables en tiempo real, es decir, no debe haber superposición de tareas. Para esto se ha debido elegir adecuadamente el periodo de muestreo con el cual se han discretizado las funciones de transferencia. Este tiempo de muestreo debe coincidir con el tiempo de llamada periódica que realiza el *Timer1* sobre el ADC dando comienzo a su ISR.

El periodo de muestreo elegido ha sido de 0.00008s , por lo que en una entrada senoidal de frecuencia 50Hz , se podrán obtener 250 muestras.

Con este paso, se ha podido comprobar que el sistema funciona de forma correcta.

A continuación, se mostrará cómo se está realizando la transmisión de datos por el *SPI*. Esto demuestra un correcto funcionamiento ya que la función del *SPI* es la última que se encuentra dentro de la *ISR* del ADC.

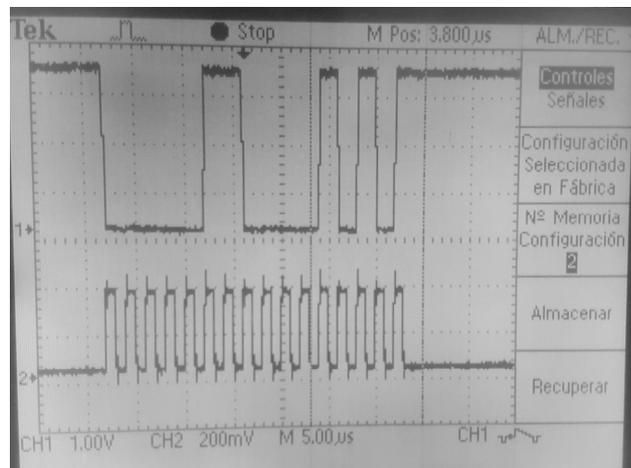
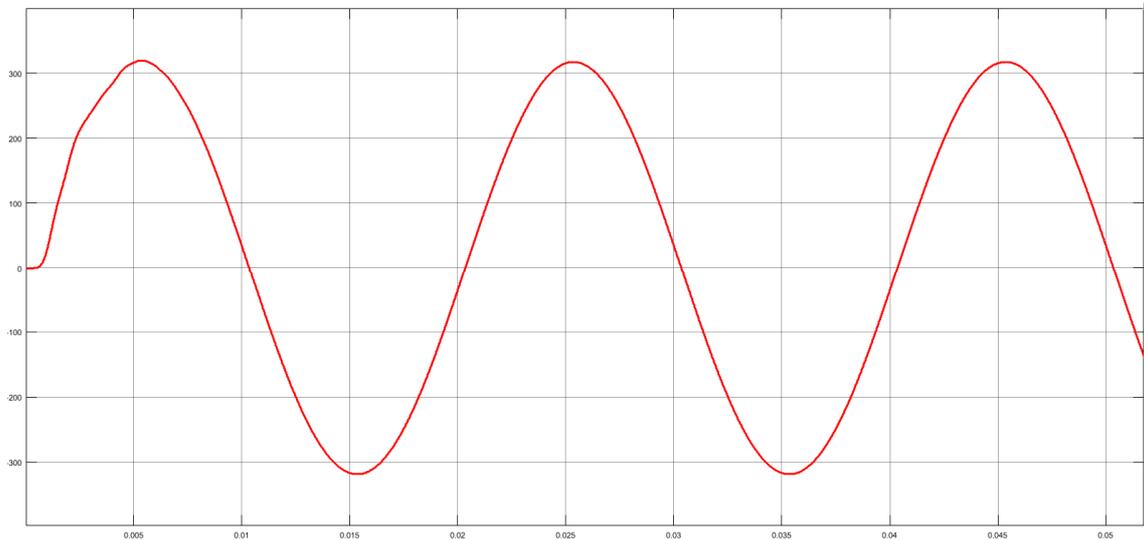


Ilustración 72. Transmisión del SPI

5.4 Simulaciones en vacío.

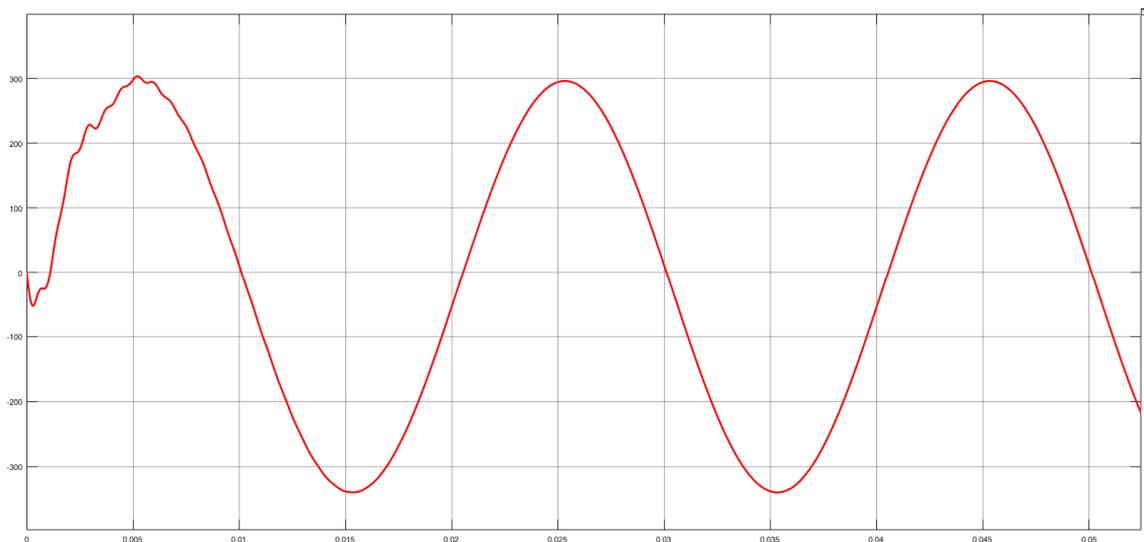
El sistema en vacío ha sido mostrado con anterioridad. En este caso, al no tener carga, la corriente I_L es cero. Por esto, la dependencia de la tensión en el condensador 4 depende únicamente de la senoidal aplicada a la entrada.



Se puede ver de forma clara y como era de esperar, que la diferencia de tensión que existe respecto a la senoidal aplicada a la entrada. Esto permite cuantificar las pérdidas que hay en la línea. Al tener un paso de simulación pequeño, parece que se trata de una señal continua.

5.5 Simulación con carga de $I_L=cte$

Siguiendo la metodología anterior. En este caso se supone una corriente constante, por lo que al efecto de vacío se le deben superponer los efectos de esta corriente emulando la carga.



Como se puede ver en los resultados, la caída de tensión es mayor que en la situación de vacío. Estos resultados son bastante interesantes, y con algunas variaciones en las matrices del sistema en variables de estados permitirán valorar valores de tensión y corriente en puntos medios de la línea. También podrán determinar el comportamiento de algunos sistemas electrónicos conectados a una red previamente estable.

6. Conclusiones

6.1 Problemas encontrados.

Conexión F28377D->PC

A lo largo del desarrollo de este trabajo han ido apareciendo pequeños problemas que han debido ser solucionados a lo largo del camino. En primer lugar, la conexión entre el microprocesador y el PC con el que se trabaja fue complicada. Después de verificar muchas conexiones y familiarizarse con el entorno, el problema que no permitía la conexión era al parecer un cable “defectuoso”. Me refiero a este cable como “defectuoso” ya que, el microprocesador tiene una función de testear si se encuentra conectado de forma correcta, y usando esta herramienta todo parecía funcionar. Finalmente probando varios cables se consiguió una buena conexión.

Base teórica

Hablando de la obtención de las ecuaciones de las líneas de transmisión, se han obtenido unas ecuaciones de aproximación. Hoy en día existen multitud de tipos de cables empleados para las transmisiones de potencia con distintos grosores, distintos aislantes, distinto número de conductores, distinto material conductor... Por todo esto se ha tomado como referencia el manual mostrado en el apartado de Base Teórica, perteneciente a la compañía NEXANS. Con las ecuaciones empleadas se consiguen valores realmente próximos, por lo que se han aceptado como válidas. Esto no quiere decir que sean unas ecuaciones que se puedan tomar como 100% fiables, se trata de una primera aproximación.

Entorno *Simulink*

Aquí se han encontrado grandes problemas que no deberían haberse producido, con lo que ha supuesto una demora de tiempo importante en el desarrollo de este trabajo. El micro de *Texas Instruments* F28377D puede ser programado mediante un diagrama de bloques en entorno *Simulink*. Esta fue la vía inicial que tomó este proyecto. Decir que este tipo de programación debería ser completa, es decir, con todas las funciones que tiene el micro, pero no es así, por lo que fue forzado el cambio a usar *Code Composer Studio*. Concretando el problema, el micro posee un sistema de conexión SPI, típicamente usado, pero completamente imposible de activar desde MATLAB. *Simulink* permite la configuración de la mayoría de periféricos presentes en la placa del F28377D menos la activación del sistema SPI.

Por otro lado, la visualización de los datos de conversión de ADC no resultó ser lo intuitiva para el usuario que se esperaba, por lo que junto con el inconveniente mayor que se encontró en la configuración del SPI se decidió cambiar al uso de CCS.

Conexión del F28337D.

Llegado un punto del desarrollo de este trabajo, el procesador sufrió algún tipo de problema que no pudo ser detectado ni solventado por lo que las simulaciones en vacío y a corriente constante han tenido que ser comentadas sobre supuestos resultados obtenidos de las situaciones reales emuladas en el entorno *Simulink* de MATLAB.

6.2 Futuras líneas de desarrollo.

Este trabajo fin de grado trata de abrir una vía de emulación de posibles escenarios en el diseño de líneas de transmisión. Se trata de un campo muy amplio y complejo por lo que las posibilidades de continuar hablando de este tema son muchas.

- **Mejora o especificación de las ecuaciones.** Se podría ampliar el tipo de líneas de transmisión a analizar. Es decir, se podrían añadir más parámetros de dependencia en las líneas en unas ecuaciones genéricas o la realización de unas ecuaciones particulares para cada tipo de línea.
- **Implementación en MATLAB.** Se cree que en futuras versiones de MATLAB (posteriores a la R2016b), los problemas que han aparecido en el entorno deberían desaparecer, por lo que este entorno podría ser mucho más visual y práctico a la hora de presentarlo ante un usuario que no ha estado durante la elaboración del programa. *Simulink* permitiría crear un interfaz en el que solo se puedan tocar los parámetros que se vayan a estudiar de la línea. Esto facilitaría también mucho la obtención de las funciones de transferencia y su posterior discretización.
- **Emulación de más escenarios.** Simplemente partiendo de toda la base que aportan estas páginas, se podrán analizar equipos y sistemas que este trabajo no ha tratado, pudiendo observar nuevos resultados y la elaboración de sistemas de control que sean de utilidad para unas condiciones deseadas.

7. Bibliografía

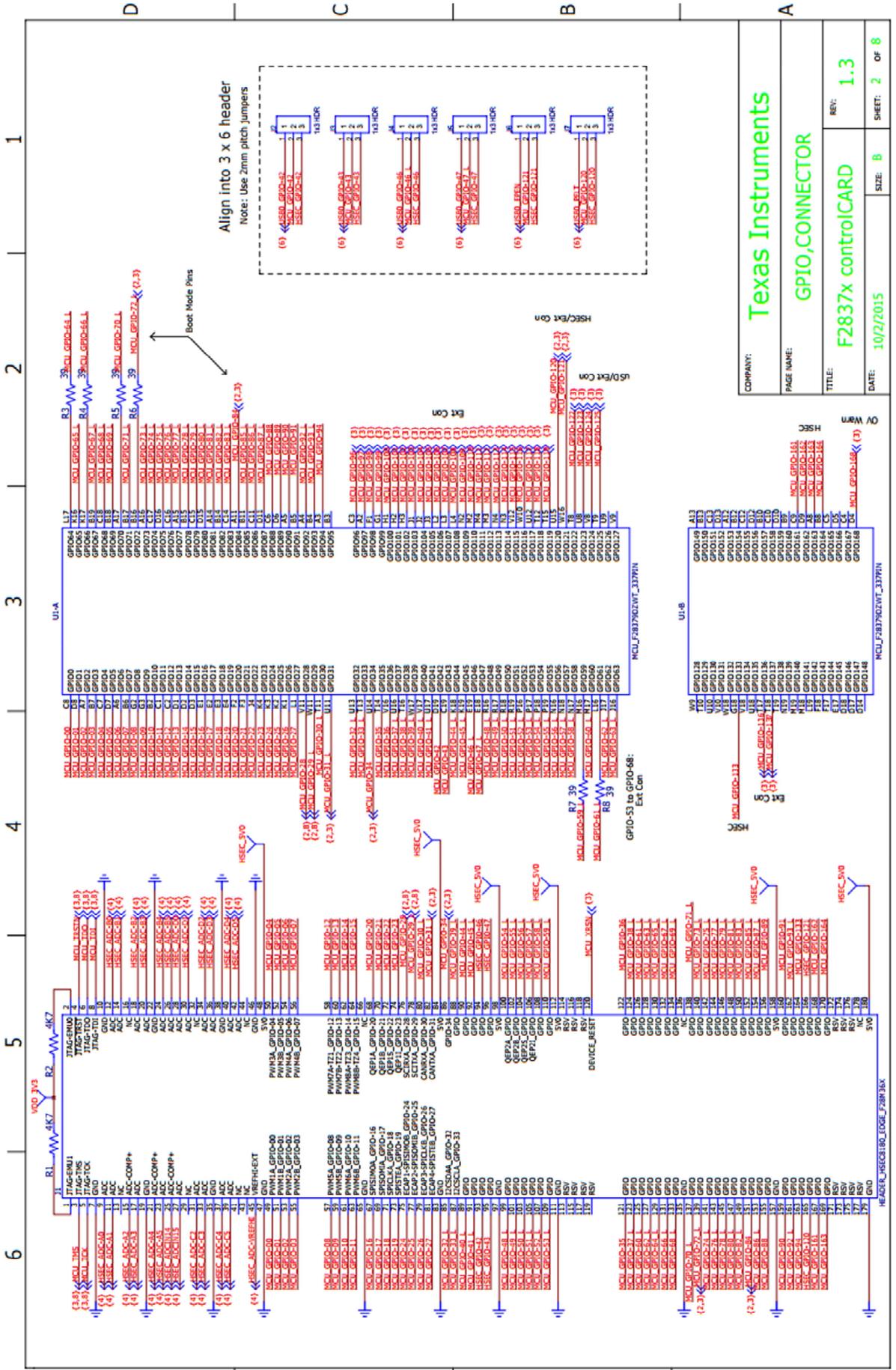
- Maoud Farzaneh, Ph D, Shahab Faroki, PH D, William A. Chisholm, Ph D: *Electrical Design of Overhead Power Transmission Lines*.
- Ebrahim A. Brandan, Mohammed E. Rizq, Mansour H abdel-Raham: *Investigation of submarine cable parameters impact on energitation transien in offshore wind farms*. Electrical Engineering Department, University of El-Mansoura, Mansoura, Egypt.
- John J. Grainger, William D. Stevenson Jr : *Power System Analysis*
- Reta-Hernandez, Manuel: *Power System Stability & Control, Cap 13: Transmission Line Parameters*, Universidad Autónoma de Zacatecas, México.
- Carlos K. C. Arruda, Anny A. Silveira, Fernando C. Dart: *Comparison of Power Transmission Lines Model with Finite Elements*.
- NEXANS, Submarine Power Cables.
- 345-AMX AC Power Source Datasheet.
- TMS320 F2837xD Dual-Core Delfino®; Microcontrollers datasheet.
- TMS320 F2837xD Dual-Core Delfino®; User manual.
- DAC8420 Datasheet.
- Estadísticas del Sistema eléctrico español.

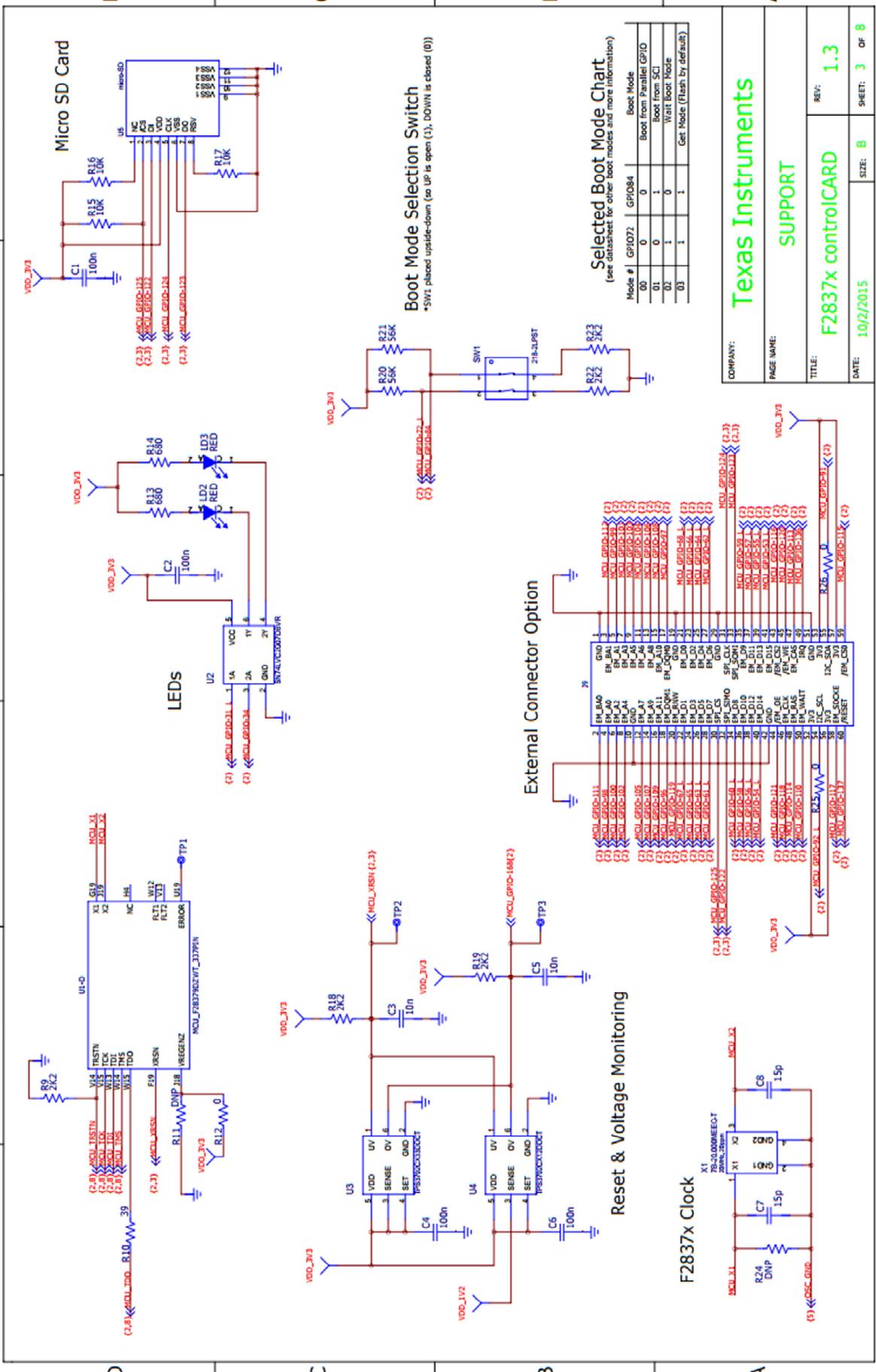
ANEXO I. Correspondencia de pines F28377D – Docking Station

1		2		3		4		5		6	
REVISION RECORD											
WNO:	SCH REV:	PCB REV:	DATE:	NOTES:							
TI=BL	RI=O	RI=O	RI=O	TI=BL	RI=O	RI=O	RI=O	TI=BL	RI=O	RI=O	RI=O
			05-Jun-2013								
			24-Oct-2013								
			09-May-2014								
			17-May-2015								

<p>NOTES:</p> <p>1) USB Differential Pairs - 90 Ohm (A) USB-DM & USB-DP (B) USB0_GPIO-42 & USB0_GPIO-43 (C) MCU_GPIO-42 & MCU_GPIO-43</p> <p>2) EMIF - External Memory Interface Impedance Matching from J1 to U1 (A) MCU_GPIO-39:41, MCU_GPIO-44:52, MCU_GPIO-86:94 - Address [0:21] (B) MCU_GPIO-85, MCU_GPIO-83:53 - Data [0:31] (C) MCU_GPIO-30 - Clock (D) MCU_GPIO-37, 31, 29 - Read/Write/ClkEn Pins (E) MCU_GPIO-32, 34, 35, 28 - Chip Select Pins</p> <p>3) ADC Differential Pair Impedance Matching (A) HSEC_ADC even pins should match with HSEC_ADC + 1 pin (ie HSEC_ADC-C2 should match with HSEC_ADC-C3) (B) MCU_ADC even pins should match with MCU_ADC + 1 pin (ie MCU_ADC-A0 should match with MCU_ADC-A1)</p>
--

<p>COMPANY: Texas Instruments</p> <p>PAGE NAME: TITLE</p> <p>TITLE: F2837x controlCARD</p> <p>DATE: 10/2/2015</p> <p>SIZE: B</p> <p>REV: 1.3</p> <p>SHEET: 1 OF 8</p>
--





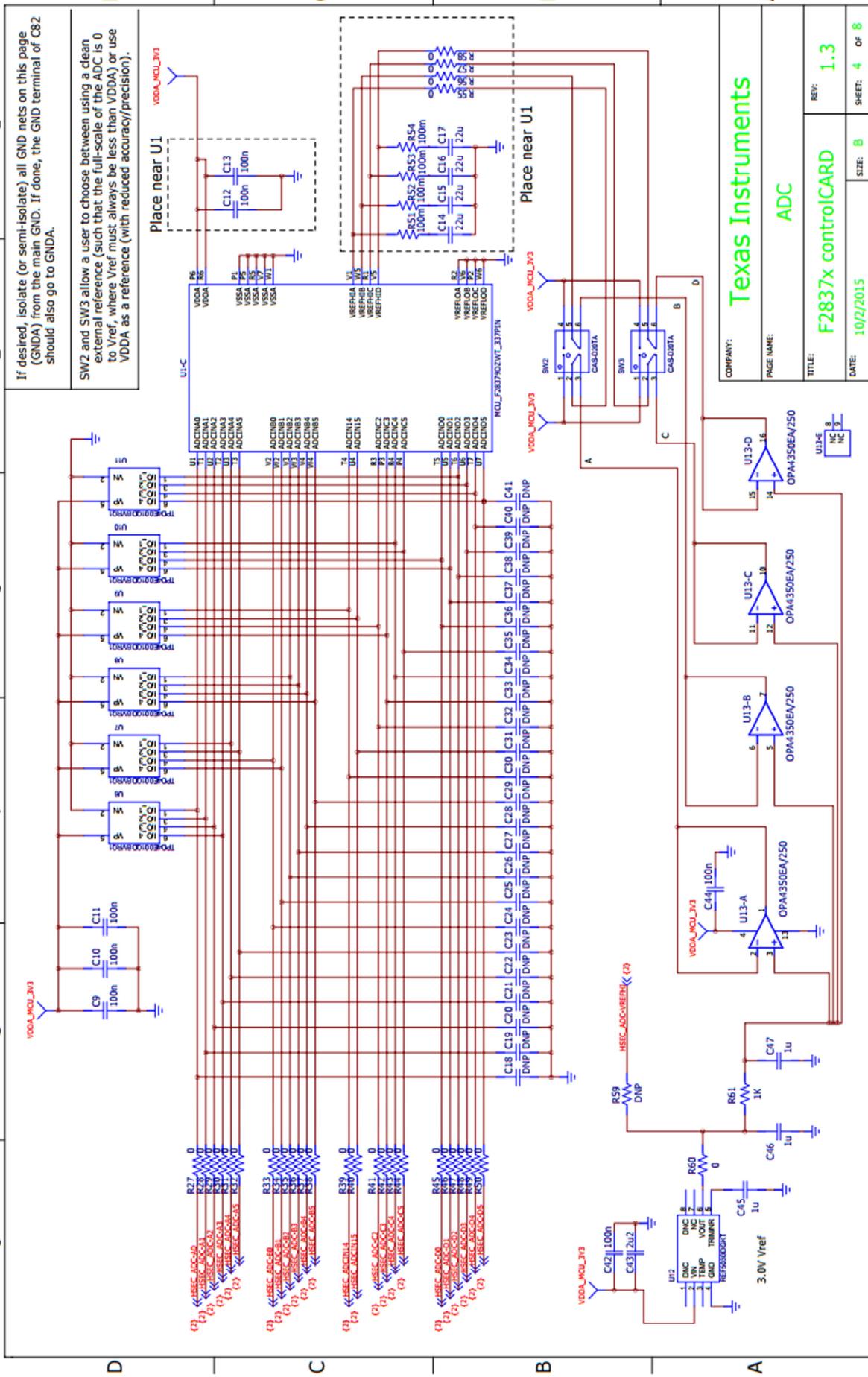
1 2 3 4 5 6

If desired, isolate (or semi-isolate) all GND nets on this page (GNDA) from the main GND. If done, the GND terminal of C82 should also go to GNDA.

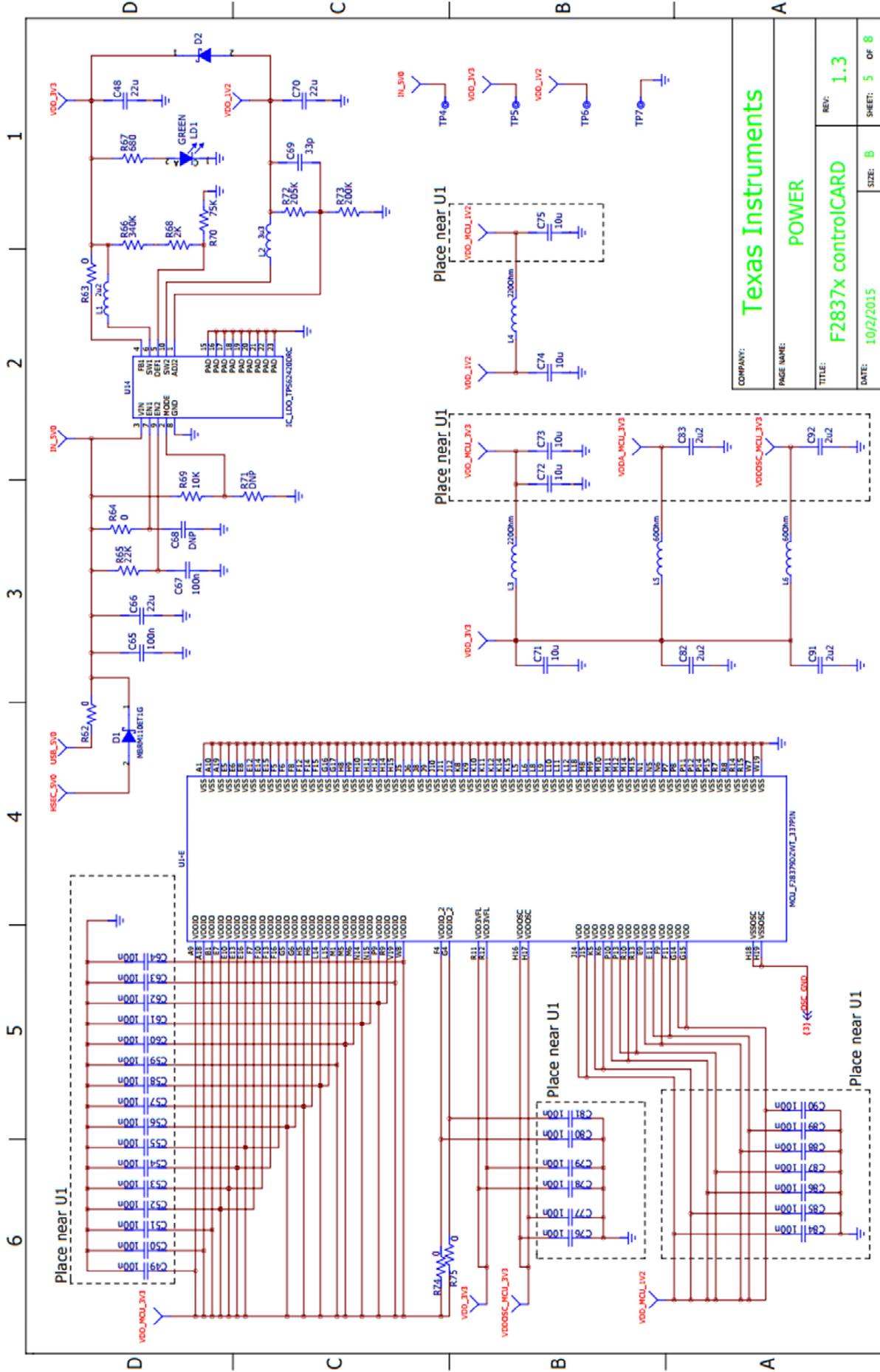
SW2 and SW3 allow a user to choose between using a clean external reference (such that the full-scale of the ADC is 0 to Vref, where Vref must always be less than VDDA) or use VDDA as a reference (with reduced accuracy/precision).

Place near U1

Place near U1

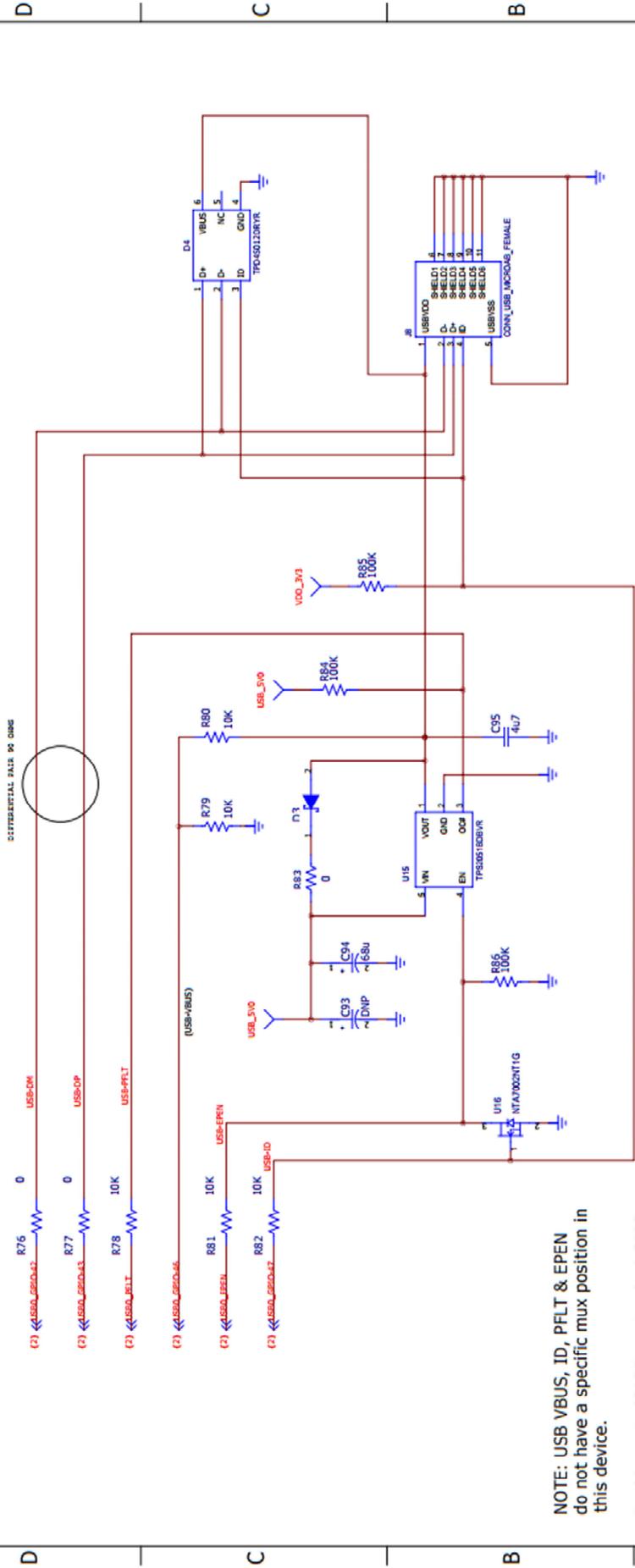


COMPANY: Texas Instruments	
PAGE NAME: ADC	
TITLE: F2837x controlCARD	REV: 1.3
DATE: 10/2/2015	SIZE: B
SHEET: 4 OF 8	



COMPANY: Texas Instruments	
PAGE NAME: POWER	
TITLE: F2837x controlCARD	REV: 1.3
DATE: 10/2/2015	SIZE: B
SHEET: 5 OF 8	

6 5 4 3 2 1



NOTE: USB VBUS, ID, PFLT & EPEN do not have a specific mux position in this device.

In this controlCARD, a standard GPIO is used to detect changes to these signals.

R83 is populated to allow a customer to evaluate the controlCARD without a baseboard if they desire. USB specifies the total bus capacitance on any device as 10uF max. R83 should be depopulated in most end applications.

COMPANY: Texas Instruments	
PAGE NAME: USB	
TITLE: F2837x controlCARD	REV: 1.3
DATE: 10/2/2015	SIZE: B
SHEET: 6 OF 8	

ANEXO II. Código integro MATLAB estimación de parámetros.

```

%Parametros fisicos
j = 1;

Distancia_entre_conductores = 0.02; % Metros(2cm)
Longitud = 1 * 10^3; % metros
Seccion_Cable = [35 50 70 95 120 150 185 240]; % EN MILIMETROS CUADRADOS

epsilon_0 = 8.854*10^-12; % [F/m]
epsilon_r = 1; % Permitividad en el medio (en el vacio 1)
epsilon = 72 * 10^-9; % Producto epsilon_0*epsilon_r (AGUA DESTILADA) [uF/Km] LA ESTIMACION MEJORA MUCHO CON UN VALOR DE 65 (YA QUE NO ES AGUA DESTILADA EN REALIDAD)

Temperatura = 90;
Coeficiente_Termico = 3.93 * 10^-3; % COBRE
Resistividad_20C = 0.01786; % COBRE

Seccion = Seccion_Cable(j)

%Parametros electricos
Frecuencia = 50; % 50Hz
Frecuencia_rad_s = 2*3.1415*Frecuencia;

%Tabla valores cable Nexans
Rdc_20C_NEXANS_matriz = [0.524 0.387 0.268 0.193 0.153 0.124 0.0991 0.0754]; % RESISTENCIA EN DC 20°C EN FUNCION DE LA SECCION Y LONGITUD
R_funcion_seccion = [0.67 0.49 0.34 0.25 0.20 0.16 0.13 0.10]; % RESISTENCIA EN AC 90°C EN FUNCION DE LA SECCION Y LA LONGITUD
L_funcion_seccion = [0.43*10^-3 0.41*10^-3 0.38*10^-3 0.37*10^-3 0.35*10^-3 0.34*10^-3 0.33*10^-3 0.32*10^-3];
C_funcion_seccion = [0.24*10^-6 0.27*10^-6 0.33*10^-6 0.36*10^-6 0.39*10^-6 0.42*10^-6 0.46*10^-6 0.51*10^-6];

R = R_funcion_seccion(j)*Longitud/1000
L = L_funcion_seccion(j)*Longitud/1000
C = C_funcion_seccion(j)*Longitud/1000

% SECCION Y LONGITUD SELECCIONADAS ANTERIORMENTE, MISMA QUE EL CABLE
% NEXANS PARA PODER COMPARAR
for i =1:8

    %RESISTENCIA

    Rdc_estimada(i) = (Resistividad_20C*Longitud/Seccion_Cable(i)) * ( 1 + Coeficiente_Termico * (Temperatura-20));

    Error_estimacion_Rdc(i) = Rdc_estimada(i) - Rdc_20C_NEXANS_matriz(i)

    figure(1)
    plot (Rdc_estimada)
    title('Comparativa Rdc estimada(azul) - Rdc Nexans(rojo)')
    xlabel('x')
    ylabel('Rdc')
    hold on
    plot(Rdc_20C_NEXANS_matriz)

    Ks = 1; %TABLAS
    Fk = 8 * Frecuencia_rad_s * Frecuencia * 10^-7 / Rdc_estimada(i);
    Xs = sqrt (Fk * Ks);
    Ys = Xs^4 / (192 + 0.8 * Xs^2);

    Radio_conductor(i) = sqrt(Seccion_Cable(i)/3.1415)/1000; % Seccion en mm por lo tanto radio en mm
    Diametro_conductor(i) = 2 * Radio_conductor(i); % en metros

    Kp = 1; %TABLAS TFG
    Xp = sqrt (Fk * Kp);
    Fp = Xp^4 / (192 + 0.8 * Xp^4);
    Yp = Fp * (Diametro_conductor(i)/Distancia_entre_conductores)^2 * (0.312 * (Diametro_conductor(i)/Distancia_entre_conductores)^2 + 1.18 / (Fp +0.27));

    figure(2);
    plot (Rac_estimada)
    title('Comparativa Rac estimada(azul) - Rac Nexans(rojo)')
    xlabel('x')
    ylabel('Rac')
    hold on
    plot (R_funcion_seccion)

    %INDUCTANCIA

    L_interna = (i * 10^-7) * Longitud/1000 ; % H/m
    L_externa = 2 * 10^-7 * log (Distancia_entre_conductores/Radio_conductor(i)); %H/m

    L_estimada(i) =( L_interna + L_externa) * 10^3; % L1+L2 mH/Km

    Error_estimacion_inductancia(i) = L_estimada(i) - L_funcion_seccion(i)

```

```

%CAPACIDAD
C_estimada(i) = (2 * pi * epsilon) / (log (Distancia_entre_conductores/Radio_conductor(i))); %[uF/Km]
Error_estimacion_capacitancia(i) = C_estimada(i) - C_funcion_seccion(i)

figure(4);
plot (C_estimada)
title('Comparativa capacitancia estimada(azul) - Capacitancia Nexans(rojo)')
xlabel('x')
ylabel('C')
hold on
plot (C_funcion_seccion)
end

```

ANEXO III. Código integro MATLAB: matrices VVEE, discretización, Funciones de transferencia.

```

Generador = 220*sqrt(2); %V
Rg = 10;
pi = 3.1415;
Frecuencia = 50; % Hz
Frecuencia_rad_s = 2 * pi * Frecuencia;
Numero_de_segmentos = 3;
Ts = 10e-6;

%% RESISTENCIA EQUIVALENTE
Seccion_Cable = 35; %mm2
Coeficiente_Termico = 3.93 * 10^-3; % COBRE
Resistividad_20C = 4.01786; % COBRE
Longitud = 80; %Km
Temperatura = 90;

Rdc_eq = (Resistividad_20C*Longitud/Seccion_Cable) * ( 1 + Coeficiente_Termico * (Temperatura-20));

Radio_conductor = sqrt(Seccion_Cable/3.1415)/1000; % Seccion en mm por lo tanto radio en mm
Diametro_conductor = 2 * Radio_conductor; % en metros

Ks = 1; %TABLAS
Fk = 8 * Frecuencia_rad_s * Frecuencia * 10^-7 / Rdc_eq;
Xs = sqrt (Fk * Ks);
Ys = Xs^4 / (192 + 0.8 * Xs^2);

Distancia_entre_conductores = 0.02;
Kp = 1; %TABLAS TFG
Xp = sqrt (Fk * Kp);
Fp = Xp^4 / (192 + 0.8 * Xp^4);
Yp = Fp * (Diametro_conductor/Distancia_entre_conductores)^2 * (0.312 * (Diametro_conductor/Distancia_entre_conductores)^2 + 1.18 / (Fp +0.27));

Rac_eq = Rdc_eq * ( 1 + 1.5*(Ys+Yp));
R = Rac_eq/Numero_de_segmentos; % R_AC equivalente por segmento

%% INDUCTANCIA EQUIVALENTE

L_interna = (1 * 10^-7) * Longitud ; % %H/m
L_externa = 2 * 10^-7 * log (Distancia_entre_conductores/Radio_conductor); %H/m

L_eq = ( L_interna + L_externa) * 10^3; % L1+L2 mH/Km

L = L_eq/3;

%% CAPACIDAD EQUIVALENTE
epsilon_0 = 8.854*10^-12; % [F/m]
epsilon_r = 1; % Permitividad en el medio (en el vacio 1)
epsilon = 72 * 10^-9; %

C_linea = (2 * pi * epsilon) / (log (Distancia_entre_conductores/Radio_conductor)); % [uF/Km]
C_eq = C_linea * Longitud;
C = C_eq / Numero_de_segmentos; % % C equivalente por segmento

%%MATRICES EN VVEE

%CONOCIDA LA CARGA
% Matrices para 1 segmento
A = [-R/L 0 1/L -1/L; 0 -R_Load/L_Load 0 1/L_Load; -1/(C/2) 0 -1/(Rg*C/2) 0; 1/(C/2) -1/(C/2) 0 0];
B = [0;0;1/(Rg*C/2);0];
Cm = [0 0 0 1];
D = [0];

```

```

% Matrices para 3 segmentos
A3 = [-R/L 0 0 0 1/L -1/L 0 0;
      0 -R/L 0 0 0 1/L -1/L 0;
      0 0 -R/L 0 0 0 1/L -1/L;
      0 0 0 -R_Load/L_Load 0 0 0 1/L_Load;
      -1/(C/2) 0 0 0 -1/(Rg*(C/2)) 0 0 0;
      1/C -1/C 0 0 0 0 0 0;
      0 1/C -1/C 0 0 0 0 0;
      0 0 1/(C/2) -1/(C/2) 0 0 0 0];

B3 = [0;0;0;0;1/(Rg*(C/2));0;0;0];

C3 = [0 0 0 0 0 0 0 1];

%DESCONOCIDA LA CARGA

% Sistema en vacio

A3_sis_vacio = [-R/L 0 0 1/L -1/L 0 0;
               0 -R/L 0 0 1/L -1/L 0;
               0 0 -R/L 0 0 1/L -1/L;
               -1/(C/2) 0 0 -1/(Rg*(C/2)) 0 0 0;
               1/C -1/C 0 0 0 0 0;
               0 1/C -1/C 0 0 0 0;
               0 0 1/(C/2) 0 0 0 0];

B3_sis_vacio = [0;0;0;1/(Rg*(C/2));0;0;0];

C3_sis_vacio = [0 0 0 0 0 0 0 1];

%Valor fijando corriente

A3_sis_cargado = [-R/L 0 0 1/L -1/L 0 0;
                 0 -R/L 0 0 1/L -1/L 0;
                 0 0 -R/L 0 0 1/L -1/L;
                 -1/(C/2) 0 0 -1/(Rg*(C/2)) 0 0 0;
                 1/C -1/C 0 0 0 0 0;
                 0 1/C -1/C 0 0 0 0;
                 0 0 1/(C/2) 0 0 0 0];

B3_sis_cargado = [0;0;0;0;0;0;-1/(C/2)];

C3_sis_cargado = [0 0 0 0 0 0 0 1];

%% Paso a FT y discretizacion

linea=ss(A3,B3,C3,D)
[num,den] = ss2tf(A3,B3,C3,D)
linea_TF = tf(num,den)
linea_discreta_tustin=c2d(linea,Ts,'tustin');
linea_TF_tustin=c2d(linea_TF,Ts,'tustin');

linea_vacio=ss(A3_sis_vacio,B3_sis_vacio,C3_sis_vacio,D);
[num_vacio,den_vacio] = ss2tf(A3_sis_vacio,B3_sis_vacio,C3_sis_vacio,D);
linea_TF_vacio = tf(num_vacio,den_vacio);

linea_vacio_tustin=c2d(linea_vacio,Ts,'tustin');
linea_TF_vacio_tustin=c2d(linea_TF_vacio,Ts,'tustin');

```

```

%% Paso a FT y discretizacion
linea=ss(A3,B3,C3,D)
[num,den] = ss2tf(A3,B3,C3,D)
linea_TF = tf(num,den)
linea_discreta_tustin=c2d(linea,Ts,'tustin');
linea_TF_tustin=c2d(linea_TF,Ts,'tustin');

linea_vacio=ss(A3_sis_vacio,B3_sis_vacio,C3_sis_vacio,D);
[num_vacio,den_vacio] = ss2tf(A3_sis_vacio,B3_sis_vacio,C3_sis_vacio,D);
linea_TF_vacio = tf(num_vacio,den_vacio);

linea_vacio_tustin=c2d(linea_vacio,Ts,'tustin');
linea_TF_vacio_tustin=c2d(linea_TF_vacio,Ts,'tustin');

figure(1)
step(linea_vacio_tustin)
hold on
step(linea_TF_vacio)
legend('TF discreta(tustin)','TF linea')
figure(2)
bode(linea_vacio,linea_vacio_tustin)

linea_cargada=ss(A3_sis_cargado,B3_sis_cargado,C3_sis_cargado,D);
[num_cargada,den_cargada] = ss2tf(A3_sis_cargado,B3_sis_cargado,C3_sis_cargado,D);
linea_TF_cargada = tf(num_cargada,den_cargada);

linea_cargada_tustin=c2d(linea_cargada,Ts,'tustin');
linea_TF_cargada_tustin=c2d(linea_TF_cargada,Ts,'tustin');

figure(3)
%step(linea_cargada_tustin)
hold on
step(linea_TF_cargada)
legend('TF discreta(tustin)','TF linea')
figure(4)
bode(linea_cargada,linea_cargada_tustin)

```


Universidad de Alcalá
Escuela Politécnica Superior



Universidad
de Alcalá