



Introducción al Control Digital de Convertidores¹

Pablo A. Lezana Illesca

v1.0 Valparaíso, 19 de marzo 2019
v.1.2 : Valparaíso, 17 de mayo de 2019

¹Basado en el documento “Manejo de la Plataforma Digital DSP+FPGA para el control de Accionamientos”

Índice de Contenidos

Índice de Contenidos	I
Índice de Figuras	II
Índice de Tablas	II
Índice de Ejemplos	II
1 Introducción a Controladores Digitales.	1
1.1 Objetivos.	1
1.2 Conceptos básicos del control lineal	1
1.2.1 Sobre el modelo del sistema	1
1.2.2 Objetivo del controlador	1
1.2.3 Sobre el controlador y la idea de realimentación	2
1.2.4 Preguntas básicas que el diseñador debería hacerse	2
1.3 Limitaciones fundamentales del control	4
1.4 El controlador PI	5
1.5 Breve reseña al operador z	6
1.6 Diseño de controladores con Matlab	9
1.6.1 Root locus (lugar geométrico de raíces)	9
1.7 Saturación y estructuras anti-enrollamiento	12
2 Control Lineal de Corriente en Convertidores.	15
2.1 La planta	15
2.2 Diseño del controlador	16
2.3 Implementación	18
2.3.1 Generación de referencias	18
2.3.2 Fibra óptica	18
2.3.3 Tarjeta de circuitos de disparo	18
3 Actividades evaluadas.	20

Índice de Figuras

1.1	(a) Relación deseada entre la referencia y la salida; (b) Esquema general de un controlador.	2
1.2	Sistema de control basado en realimentación.	2
1.3	Esquema de control general, considerando perturbaciones.	4
1.4	Ejemplos de espectro de los diferentes ruidos que afectan al sistema.	5
1.5	Mapecto de puntos del plano s al plano z	8
1.6	Mapecto de múltiples franjas del plano s en el mismo plano z	8
1.7	Efecto del muestreo en la forma en que se observan las señales muestreadas con respecto a las señales reales.	9
1.8	Lugar geométrico de raíces para el sistema considerando: (a) la planta y una ganancia pura como controlador; (b) la planta y un controlador PI ; (c) la planta y un controlador PI con un cero más lento que en el caso anterior.	10
1.9	Respuesta de un sistema controlado ante un escalón de referencia de -1 a 1, sin esquema de antienrollamiento ($y(t)$) y con él ($y_{aw}(t)$).	12
1.10	Estructura de controlador para evitar el fenómeno del enrollamiento.	13
2.1	Diagrama general del esquema del sistema.	16
2.2	Diagrama de bloques de una carga $R - L$ trifásica balanceada en el eje rotatorio dq	17
2.3	Tarjeta de fibra óptica.	18
2.4	Tarjeta de disparo con opción de generación local de tiempo muerto.	19

Índice de Tablas

Índice de Ejemplos

1.1	Código C para saturación	12
1.2	Código C para la un PI con una estructura anti-enrollamiento	13

Capítulo 1

Introducción a Controladores Digitales.

1.1. Objetivos.

- Conocer los conceptos básicos del control basado en dispositivos digitales.
- Conocer aspectos de implementación de controladores digitales.

1.2. Conceptos básicos del control lineal

Antes de entrar al detalle del diseño de controladores en tiempo discreto, es importante comentar algunos aspectos básicos del control lineal, que en su mayoría son comunes tanto para el diseño en tiempo continuo como para el diseño en tiempo discreto.

1.2.1. Sobre el modelo del sistema

El primer paso para poder diseñar adecuadamente un controlador lineal es contar con un modelo lineal del sistema, o planta, que se desea controlar (en adelante G). Este modelo debe relacionar linealmente la(s) variable(s) de salida (en adelante y) que interesa controlar con la(s) variable(s) de entrada (en adelante u) que pueda(n) ser manejada(s) por el controlador. Sin este modelo es poco probable que el controlador logre controlar la variable de entrada, y si llega a hacerlo, probablemente el desempeño será pobre. Por otra parte, mientras más preciso sea el modelo, el desempeño del sistema controlado será mejor.

Al tener un modelo lineal del sistema será posible representarlo a través de una función de transferencia en el plano de Laplace o equivalentemente a través de variables de estado.

1.2.2. Objetivo del controlador

El controlador debe lograr que y siga a una señal de referencia r de la manera más parecida posible, incluso si el sistema se encuentra sometido a la acción de elementos externos, por lo tanto el controlador ideal debiese hacer que la función de transferencia entre la referencia y la salida fuese una ganancia unitaria, tal como muestra la Fig. 1.1(a).

En las siguientes secciones se verá como esta simple idea no es tan sencilla de llevar a cabo debido a, entre otros, incertidumbres en el modelo de la planta y la existencia de perturbaciones en el sistema real.

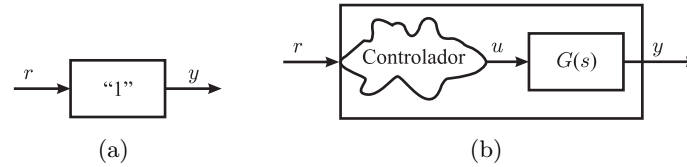


Figura 1.1. (a) Relación deseada entre la referencia y la salida; (b) Esquema general de un controlador.

1.2.3. Sobre el controlador y la idea de realimentación

Al tener un modelo que relaciona y con u a través de G , entonces será posible diseñar un controlador C , con alguna función de transferencia aún por determinar, que a partir de la diferencia (o error) entre la salida de referencia r y la salida real y determine de manera automática una cierta actuación u que minimice tal error. La Fig. 1.2 muestra estas relaciones a través de un diagrama de bloques. Note que de esto se desprende que para que un sistema de control basado en realimentación funcione, al menos durante un instante debe existir un error entre lo deseado y lo realmente obtenido. Los controladores que logran que este error se haga permanentemente cero son conocidos como *controladores con error estacionario cero* y es el tipo de controladores a los que siempre se debiera aspirar.

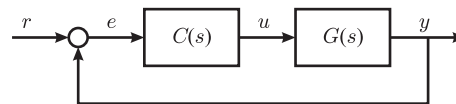


Figura 1.2. Sistema de control basado en realimentación.

1.2.4. Preguntas básicas que el diseñador debería hacerse

Antes de diseñar un controlador lineal, ya sea análogo o digital, el diseñador debería plantearse algunas preguntas sencillas, pero esenciales. Estas preguntas son:

1. ¿Cuál(es) es(son) la(s) variable(s) que quiero controlar?
Generalmente en nuestro caso estas variables resultan ser corrientes, voltajes, velocidad, torque o temperatura, entre otros.
2. ¿Qué variable(s) puedo manejar?
En estricto rigor, en electrónica de potencia, la mayoría de las veces lo que podemos controlar es el instante en que los diferentes semiconductores comienzan (y en algunos casos terminan) de conducir.
3. ¿Qué variable(s) **manejaables** puedo relacionar con la(s) variable(es) que quiero controlar a través de **una función lineal**?
En particular en el caso de semiconductores controlados, y desde la óptica de un controlador lineal, esto puede traducirse en valores medio de actuación de voltaje ($\bar{v}_o(t)$) sobre la carga de interés. Dado que en general las cargas que utilizamos pueden aproximarse por cargas lineales, la real relación que se establece es del tipo:

$$\frac{I(s)}{\bar{V}_o(s)} = H(s) \quad (1.1)$$

4. Establecida la relación lineal entre variables manejables y variables a controlar, ¿Qué características tienen la función obtenida?

En el caso de cargas de tipo R_L se obtienen relaciones de primer orden y “bien comportadas”, es decir sin ceros ni polos en el semiplano derecho del plano s (o fuera del círculo unitario en el plano z). Dependiendo del tipo de actuador del que se disponga podría tener que considerarse un retardo de un período de muestreo, pero en general es un problema menor.

5. ¿Qué tipo de referencia quiero que siga la variable a controlar?

Esta es una pregunta que en general se ignore, pero resulta clave. En general las referencias en los sistemas eléctricos son de dos tipos: constantes o sinusoidales de alguna frecuencia bien determinada. En caso de ser una referencia sinusoidal de frecuencia variable, como discutiremos más adelante, nos enfrentamos a un problema mayor, que requiere de otras herramientas para ser tratado de manera adecuada.

6. De acuerdo a todo lo anterior ¿Qué características **debe** tener la función de transferencia del controlador?

Para responder a esta pregunta es necesario analizar el esquema basado en realimentación de la Fig. 1.2. En base a ella se puede establecer la relación:

$$\frac{y(s)}{r(s)} = \frac{C(s) \cdot G(s)}{1 + C(s) \cdot G(s)} \quad (1.2)$$

y lo que deseamos es que

$$y(s) = r(s). \quad (1.3)$$

Claramente esto se logra sólo si $C(s) \cdot G(s) \rightarrow \infty$. Una forma natural de alcanzar esto es haciendo que $C(s)$ sea lo más grande posible, pero esto genera problemas de rechazo a ruido (ver 1.3) además de actuaciones muy violentas, que pueden dañar tanto a los actuadores como a la planta. Es por esto que lo que se busca es lograr es (1.3) sólo a aquellas frecuencias que se encuentran presentes en la señal de referencia.

Por ejemplo, en el caso de que la referencia sea una **señal continua** y la planta **no tenga ceros en el origen**, entonces el requerimiento que se tiene es que:

$$\left| C(s) \cdot G(s) \right|_{s=0} \rightarrow \infty \quad (1.4)$$

lo que se logra de manera simple si $C(s)$ incluye un término $\frac{K_i}{s}$, es decir, un integrador. Por otro lado, se puede demostrar con mediana simpleza, que un controlador bipropio (o también llamado de paso directo, o simplemente proporcional), permite tener más grados de libertad y así obtener un mejor desempeño, lo que lleva a una estructura del tipo

$$C(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \quad (1.5)$$

conocida típicamente como *PI*. Note que si la referencia tiene una componente alterna,

$$\left| C(s) \cdot G(s) \right|_{s=j\omega} < \infty, \quad (1.6)$$

por lo que **necesariamente** existirá error de seguimiento.

En el caso de que la referencia tenga una componente de frecuencia ω_c entonces se requiere que

$$\left| C(s) \cdot G(s) \right|_{s=j\omega_c} \rightarrow \infty, \quad (1.7)$$

lo que se logra incluyendo un término del tipo $\frac{K_r}{s^2 + \omega_c^2}$ en $C(s)$, lo que se conoce como un *controlador resonante*. Al igual que en el caso del *PI* es recomendable agregar un término proporcional, obteniéndose estructuras del tipo:

$$C(s) = K_p + \frac{K_r}{s^2 + \omega_c^2} = \frac{K_p s^2 + K_p \omega_c^2 + K_r}{s^2 + \omega_c^2} \quad (1.8)$$

Incluso, para tener aún más grados de libertad para ubicar los polos y ceros del sistema, se pueden utilizar estructuras del tipo:

$$C(s) = K_p \frac{s^2 + K_1 s + K_2}{s^2 + \omega_c^2} \quad (1.9)$$

1.3. Limitaciones fundamentales del control

Se tiende a pensar que al igual que los computadores, los controladores son entes inteligentes que logran manejar variables o procesos. En ambos casos esta apreciación es errónea, ya que en general se limitan a aplicar de manera precisa una serie de reglas bien definidas. Es por esto que al diseñar un sistema de control es necesario tener en cuenta que existen ciertas limitaciones y consecuencias de implementar un tipo de diseño u otro, lo que siempre conlleva un compromiso entre diferentes variables.

Un esquema de control más completo que el visto anteriormente incluye la existencia de perturbaciones y ruidos en diferentes partes del sistema, tal como muestra la Fig. 1.3. Aquí se han incluido tres fuentes de ruido:

- Perturbación de entrada a la planta, δu , algunos ejemplos son:
 - Perturbaciones propias del sistema, como el torque de carga en una máquina, desagües en un estanque, etc...
 - Ruido en el actuador, como es el caso de los moduladores PWM, que si bien en promedio dentro de un período generan la referencia deseada, también generan ruido de alta frecuencia que excita a la planta.
- Ruido de salida, δy , como por ejemplo: efectos parásitos no modelados o perturbaciones externas.
- Ruido de medición, δm , debido principalmente a sensores ruidosos o que presentan algún tipo de no linealidad. En electrónica de potencia es frecuente que las mediciones se vean contaminadas por las emisiones electromagnéticas producto de la conmutación de los semiconductores de potencia.

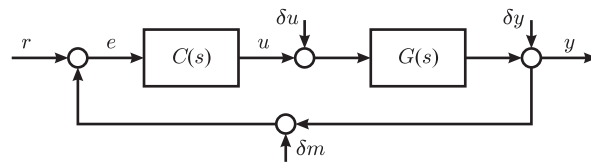


Figura 1.3. Esquema de control general, considerando perturbaciones.

Note que, por ejemplo, el controlador no tiene manera de identificar si una señal de ruido es generada por δy o por δm , sin embargo es claro que aquel generado por δy debiese ser compensado, ya que se trata de un ruido “real”, mientras que el generado por δm debiese ser ignorado, ya que no es un ruido que se encuentre verdaderamente presente en y . Ya puede verse que este tipo de problemas no es trivial de solucionar (de hecho puede no ser posible) y requiere que el diseñador tenga cierta información del sistema real. En un caso “bien comportado” los espectros de δy y δm se encontrarán suficientemente separados entre sí [tal como muestra la Fig. 1.4(a)], de manera que el diseñador será capaz de encontrar un controlador que compense adecuadamente el ruido correspondiente a δy e ignore el generado por δm , por otra parte un caso mucho más complicado es el que muestra la Fig. 1.4(b), donde los espectros de ambos ruidos se traslapan, en este caso el diseñador deberá decidir si dejará parte del ruido de δy sin compensar o si tratará de compensar el ruido “fantasma” generado por δm .

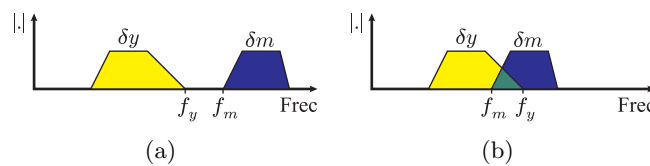


Figura 1.4. Ejemplos de espectro de los diferentes ruidos que afectan al sistema.

Este es sólo una de las limitaciones fundamentales del control que se enumeran a continuación:

1. Mientras más rápido sea un controlador (posea un ancho de banda mayor), más sensible será al ruido de medición.
2. Un controlador óptimo para un tipo de referencia es también óptimo para compensar perturbaciones de entrada de ese tipo, pero es pésimo para compensar ruido de medición de ese mismo tipo.

1.4. El controlador PI

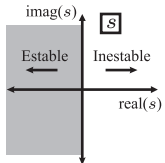
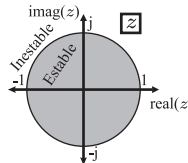
Uno de los esquemas más comunes para el control de sistemas es el conocido *PI* (Proporcional Integrativo), esta estructura es bastante simple y útil, sin embargo es importante tener en cuenta que tiene una importante limitación:

*Un PI logra error estacionario cero SÓLO para el seguimiento de referencias **constantes**. Si la referencia tienen componentes alternas, SIEMPRE habrá error en la fase y en la magnitud de la señal controlada.*

Es por esto que en general un PI se utiliza para controladores de torque o velocidad (que en general son manejadas por referencias constantes), mientras que no puede aplicarse directamente a corrientes de fase en, por ejemplo, una máquina asincrónica².

Dependiendo de como se implementará el controlador, la representación matemática del *PI* cambia un poco, pero en general se mantienen la mayoría de sus propiedades:

²Para solucionar este problema las corrientes **deben** ser pre-tratadas matemáticamente para transformarlas de senoidales a señales continuas a través de una transformación de coordenadas.

Implementación	
Amplificadores operacionales	Microcontroladores
Espacio de trabajo	
Plano s (tiempo continuo)	Plano z (tiempo discreto)
Expresión matemática	
$PI(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s}$	$PI(z) = K_p + \frac{K_i}{z-1} = \frac{K_p z + K_i - K_p}{z-1}$
Características de la función de transferencia	
1.- Un cero que depende de K_p y K_i . 2.- Un polo en el origen, $s = 0$ (integrador).	1.- Un cero que depende de K_p y K_i . 2.- Un polo en $z = 1$ (integrador).
Para estabilidad, los polos de lazo cerrado deben ubicarse	
En el semiplano izquierdo: 	En el interior de una circunferencia unitaria: 

Ya que el objetivo principal de este documento es el uso de la plataforma digital disponible en el laboratorio, en lo sucesivo el análisis se centrará en la implementación de un controlador en tiempo discreto, sin embargo la mayoría de los conceptos se pueden extender al caso en tiempo continuo.

1.5. Breve reseña al operador z

La definición más simple del operador z es la de un paso adelante en el tiempo, de esta manera:

$$zY(z) \Longleftrightarrow y(t+h) \quad (1.10)$$

o el valor que tendrá la función $y(t)$ en el siguiente instante de muestreo (donde h es el período de muestreo), lo que de paso deja de manifiesto la idea de que se está trabajando con un sistema que es visto en instantes discretos de tiempo.

A partir de esta simple definición se obtienen los siguientes resultados útiles:

$$z^2 Y(z) \Longleftrightarrow y(t+2h) \quad (1.11)$$

$$z^n Y(z) \Longleftrightarrow y(t+nh) \quad (1.12)$$

$$z^{-1} Y(z) \Longleftrightarrow y(t-h) \quad (1.13)$$

De la teoría de los sistemas lineales, se sabe que un sistema lineal puede representarse a través de una ecuación diferencial, y por ende a través de una fracción polinomial en función del operador s . Pues bien, también es posible representar al sistema lineal como una *ecuación en diferencias* y

por ende a través de una fracción polinomial en función del operador z :

$$\begin{aligned} b_n y(t + nh) + b_{n-1} y(t + [n-1]h) + \dots + b_0 y(t) &= a_m u(t + mh) + a_{m-1} u(t + [m-1]h) + \dots + a_0 u(t) \\ \Rightarrow b_n z^n Y(z) + b_{n-1} z^{n-1} Y(z) + \dots + b_0 Y(z) &= a_m z^m U(z) + a_{m-1} z^{m-1} U(z) + \dots + a_0 U(z) \\ \Rightarrow H(z) = \frac{Y(z)}{U(z)} &= \frac{a_m z^m + a_{m-1} z^{m-1} + \dots + a_0}{b_n z^n + b_{n-1} z^{n-1} + \dots + b_0} \end{aligned} \quad (1.14)$$

Al igual que en el caso continuo, para que la función tenga sentido m **debe ser menor** que n de otro modo querría decir que las entradas futuras del sistema tienen efecto en las salidas actuales.

Otra similitud al caso continuo es que al poder expresarse el sistema en forma de una fracción polinomial, se mantiene la idea de que la estabilidad del sistema depende de la ubicación de los *polos* (o raíces del denominador) de $H(z)$, los que pueden ser reales y/o imaginarios complejos conjugados. Sin embargo es importante tener en cuenta que un polo ubicado en un lugar específico tiene distintas interpretaciones en el plano $[s]$ y en el plano $[z]$. Para entender esto de mejor manera es útil la siguiente relación:

$$z = e^{sh} = e^{h \operatorname{Re}(s)} \left[\cos(h \operatorname{Im}(s)) \pm j \sin(h \operatorname{Im}(s)) \right] = e^{h\sigma} \left[\cos(h\omega) \pm j \sin(h\omega) \right] \quad (1.15)$$

Aplicando (1.15) para los diferentes valores de s mostrados en la Fig. 1.5 tenemos que si el(los) polo(s) se encuentra(n) en:

a: Polo real en s en el semiplano izquierdo $\Rightarrow z_a = e^a$ con $a < 0 \Rightarrow |z_a| < 1$.

b: Polo real en s en el semiplano izquierdo $\Rightarrow z_b = e^b$ con $b < a < 0 \Rightarrow |z_b| < |z_a| < 1$.

c: Polo real en s en el semiplano derecho $\Rightarrow z_c = e^c$ con $c > 0 \Rightarrow |z_c| > 1$.

d: Polo real en s en el semiplano derecho $\Rightarrow z_d = e^d$ con $d > c > 0 \Rightarrow |z_d| > |z_c| > 1$.

e: Polo real en s en el origen (integrador) $\Rightarrow z_e = e^0 \Rightarrow |z_e| = 1$.

f: Polos complejos conjugados en s sin parte real \Rightarrow

$$z_f = \cos(h\omega_f) \pm j \sin(h\omega_f) \Rightarrow |z_f| = 1, \angle(h\omega_f)$$

g: Polos complejos conjugados en s con parte real negativa \Rightarrow

$$z_g = e^{h\sigma_g} \left[\cos(h\omega_g) \pm j \sin(h\omega_g) \right] \Rightarrow |z_g| < 1, \angle(h\omega_g)$$

h: Polos complejos conjugados en s con parte real positiva \Rightarrow

$$z_h = e^{h\sigma_h} \left[\cos(h\omega_h) \pm j \sin(h\omega_h) \right] \Rightarrow |z_h| > 1, \angle(h\omega_h)$$

i: Polo real en $s \rightarrow \infty$ (retardo puro) $\Rightarrow z_i = e^{-\infty} = 0$, un polo en el origen del plano z .

El polo en i es particularmente interesante, ya que trabajar en el plano s con funciones de transferencia con retardos es en general engorroso (debido a que el polo se encuentra en $-\infty$), mientras que en el caso del plano z se encuentra en el origen.

Otro aspecto que es necesario destacar es que de acuerdo con (1.15), y debido a la periodicidad de las funciones $\cos(h\omega)$ y $\sin(h\omega)$, el mapeo desde el plano s al plano z es periódico, es decir,

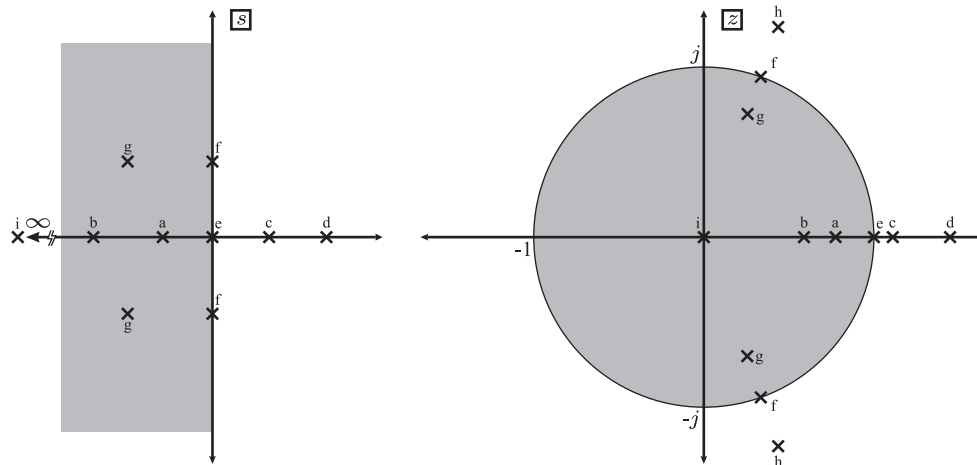


Figura 1.5. Mapeo de puntos del plano s al plano z .

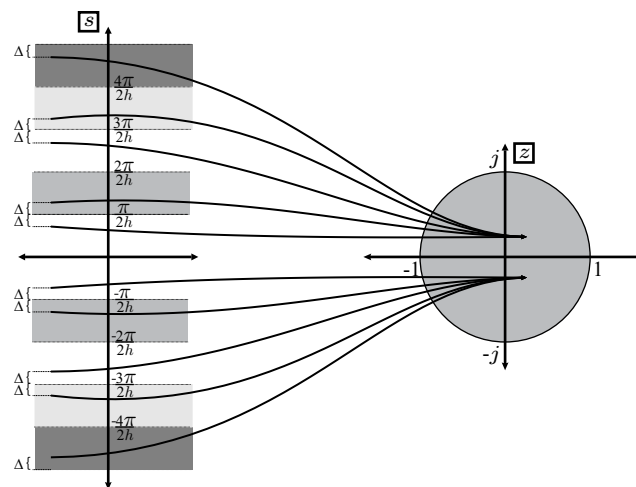


Figura 1.6. Mapeo de múltiples franjas del plano s en el mismo plano z .

infinitas franjas del plano s son mapeadas sobre el mismo plano z , y por ende, infinitos puntos del plano s son mapeados en un mismo punto del plano z , tal como se muestra en la Fig. 1.6.

Un ejemplo simple desde el punto de vista práctico es lo que se muestra en la Fig. 1.7, donde se ha muestreado una señal a una frecuencia fija $1/h$. La secuencia obtenida corresponde a una serie de puntos de magnitud constante [Fig. 1.7(a)]. Esta señal aparentemente continua, puede haber sido obtenida del muestreo de una señal análoga (plano s) de frecuencia $0, 1/h, 2/h$, etc.... Este fenómeno se conoce como doblaje de frecuencia y se relaciona fuertemente con la *frecuencia de Nyquist*³. Un ejemplo más detallado se puede revisar en el script disponible en la web de la asignatura.

Para transformar la relación entre el plano s y el plano z en una relación 1:1 y así evitar problemas relacionados con el doblaje de frecuencia, es indispensable que las señales análogas pasen por un filtro pasabajos con una frecuencia de corte menor a $\frac{2}{h}$ previo a la conversión A/D. De esta forma se garantiza que no se hayan muestreado señales cuya frecuencia sea superior a $2/h$

³Esta frecuencia se relaciona con el Teorema Fundamental del Muestreo o Teorema de Nyquist-Shannon que básicamente plantea que para poder reconstruir una señal muestreada se requiere muestrearla a una frecuencia mayor al doble de su máxima frecuencia.

y sólo la franja del plano s centrada en 0 será mapeada al plano z .

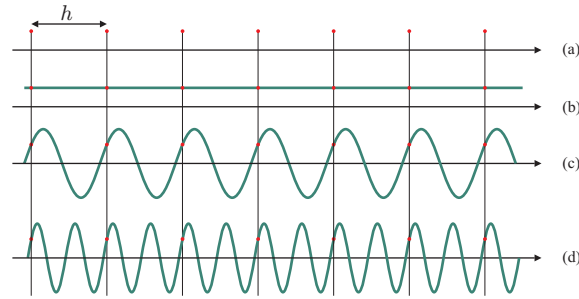


Figura 1.7. Efecto del muestreo en la forma en que se observan las señales muestreadas con respecto a las señales reales.

1.6. Diseño de controladores con Matlab

En general las estructuras de los controladores tienen 2 o más grados de libertad, es decir, parámetros que el diseñador puede ajustar, en principio, de manera completamente arbitraria. Dependiendo de la elección de estos parámetros, cambiarán las características de la respuesta del sistema controlado como por ejemplo: la velocidad del seguimiento, la magnitud y duración de las oscilaciones, y la capacidad de compensar perturbaciones.

Existen varias “recetas” para ajustar controladores simples como los PI , entre ellas se tiene:

- Ziegler y Nichols.
- Óptimo simétrico.
- Óptimo del módulo.

Estas recetas apelan a la simplicidad y a ser “óptimas” en algún sentido, pero no necesariamente son óptimas para la aplicación específica que el diseñador debe enfrentar.

El desarrollo de las herramientas computacionales (como por ejemplo `Matlab`) permiten ver de manera simple el efecto de la elección diferentes parámetros en el controlador, por lo que el uso de las recetas antes descritas no es tan necesaria.

1.6.1. Root locus (lugar geométrico de raíces)

Consideremos el lazo básico de realimentación mostrado en la Fig. 1.2, pero utilizando el operador z ⁴. Además, consideremos que el controlador $C(z)$ y la planta $G(z)$ son funciones polinomiales que pueden ser reescritas de la siguiente forma:

$$C(z) = K \frac{N_c(z)}{D_c(z)}$$

$$G(z) = \frac{N_g(z)}{D_g(z)}$$

con $N_x(z)$, $D_x(z)$ polinomios en z y K una constante.

⁴El análisis es idéntico para el caso del operador s .

Con esto la función de transferencia entre la referencia y la salida puede reescribirse como:

$$\begin{aligned} \frac{Y(z)}{R(z)} &= \frac{C(z) G(z)}{1 + C(z) G(z)} = \frac{K \frac{N_c(z) N_g(z)}{D_c(z) D_g(z)}}{1 + K \frac{N_c(z) N_g(z)}{D_c(z) D_g(z)}} \\ &= \frac{K N_c(z) N_g(z)}{D_c(z) D_g(z) + K N_c(z) N_g(z)} \end{aligned} \quad (1.16)$$

Si analizamos el denominador de (1.16) (que corresponde a los polos del sistema en lazo cerrado), podemos hacernos una idea clara del comportamiento del sistema. Como un primer paso se puede considerar que la constante K es un parámetro libre que debe ser ajustado por el diseñador y que los polinomios $N_c(z)$ y $D_c(z)$ se encuentran fijos (aunque en realidad son grados de libertad adicionales sobre los que hablaremos más adelante). De manera sencilla podemos ver que:

- Si $K \rightarrow 0$ los polos en lazo cerrado del sistema tienden a las raíces de $D_c(z) D_g(z)$, es decir a los polos del sistema en lazo abierto.
- Si $K \rightarrow \infty$ los polos en lazo cerrado del sistema tienden a las raíces de $N_c(z) N_g(z)$, es decir a los ceros del sistema en lazo abierto.

En la práctica para diferentes valores de K , se encontrarán diferentes valores para los polos de (1.16), lo que desde el punto de vista gráfico define un lugar geométrico para estas raíces.

Matlab provee el entorno de trabajo *rltool* que permite obtener este lugar geométrico de manera muy sencilla. La sintaxis para utilizar esta función es la siguiente:

$$\text{rltool}(\text{Planta})$$

Donde *Planta* es el sistema a controlar, el que puede estar en el formato de: función de transferencia (tf), variable de estado (ss) o cero-polo-ganancia (zpk), y modelada en tiempo continuo (plano s) o tiempo discreto (plano z). Si la planta se encuentra en el plano discreto se desplegará un gráfico como el mostrado en la Fig. 1.8(a). El comando *rltool* asume como estructura inicial del controlador una ganancia pura, es decir, $C(z) = K$.

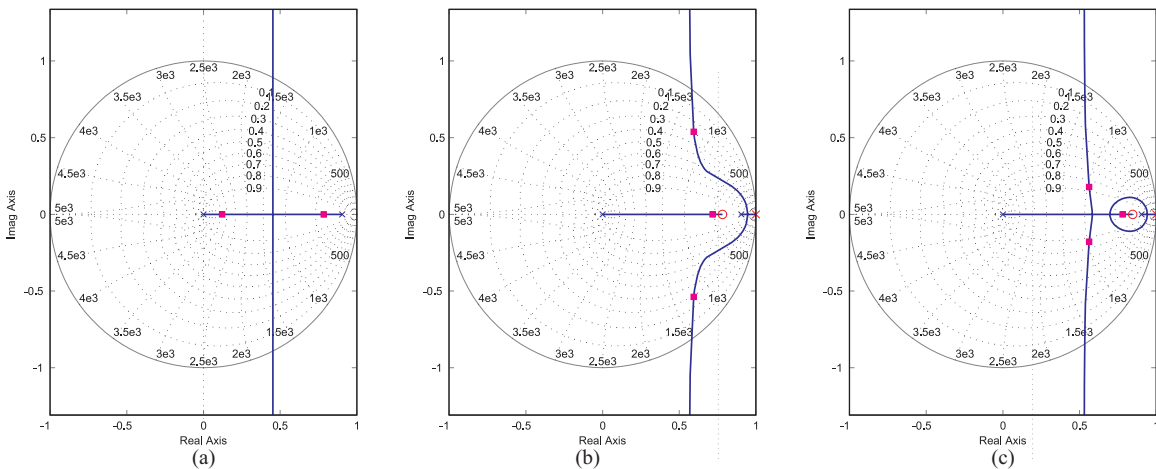


Figura 1.8. Lugar geométrico de raíces para el sistema considerando: (a) la planta y una ganancia pura como controlador; (b) la planta y un controlador PI; (c) la planta y un controlador PI con un cero más lento que en el caso anterior.

Los principales elementos de este gráfico son:

- Las líneas azules corresponden al lugar geométrico de todas las posibles raíces de lazo cerrado dada la planta y el controlador elegido.
- Los cuadrados lilas señalan la ubicación de los polos de lazo cerrado para un valor específico de K .
- Las \times y \circ azules indican los polos y ceros, respectivamente, de la planta, los que no se pueden mover dado que se considera que la planta no puede ser cambiada.
- Las \times y \circ rojas indican los polos y ceros, respectivamente, del controlador, los que pueden moverse, agregarse y/o eliminarse según se requiera, modificando así la estructura del controlador y por tanto el lugar geométrico de los polos de lazo cerrado.

De acuerdo a esto, la planta usada para la obtención de los gráficos de la Fig. 1.8 tiene dos polos: uno de ellos en el origen (retardo puro) y otro estable. La Fig. 1.8(b) muestra como cambia el lugar geométrico de raíces al cambiar la estructura del controlador a un PI , mientras que la Fig. 1.8(c) muestra el lugar geométrico obtenido al usar un PI cuyo cero es más lento que el del caso anterior.

La forma de agregar polos y ceros del controlador dependerá de la versión de **Matlab** de que se disponga. En particular en la versión 7.8, esto se realiza en la viñeta “Compensator Editor” de la ventana “Control and Estimation Tools Manager” que se abrirá al ejecutar el comando *rltool*. Posicione el cursor en el recuadro “Dynamics” y presione el botón derecho de su mouse. Se desplegará el menú *Add Pole/Zero* con las opciones para agregar este tipo de elementos.

Una vez encontrado el controlador deseado, su estructura final aparecerá en la parte superior de la ventana “Compensator Editor”.

Otra herramienta del entorno *rltool* es la posibilidad de ver “en línea” el comportamiento del sistema realimentado ante diferentes escenarios, en función del diseño realizado. Para esto basta con activar los ensayos deseado en la viñeta **Analysis**, de la ventana de lugares geométricos. Algunos de los ensayos disponibles son:

- Respuesta ante escalón de referencia (*Response to step command*), donde se desplegará en azul la variación de la salida y en color verde la variación de la actuación.
- Respuesta ante escalón de perturbación (*Response to step disturbance*), donde se desplegará en azul la variación de la salida y en color verde la variación de la actuación.
- Diagrama de Bode del lazo cerrado (*Closed-loop Bode*).
- Respuestas a impulso.

En líneas generales, los polos de lazo cerrado con parte imaginaria tienen respuestas oscilatorias, mientras más cerca del borde de la circunferencia unitaria, estas se mantienen por más tiempo. Por otra parte, considerando sólo la parte real de los polos, mientras se encuentren más cerca del origen, el sistema tenderá a la referencia de manera más rápida. Finalmente, es muy probable que la función de transferencia de lazo cerrado tenga varios polos, en estos casos los polos que determinan la velocidad del sistema son los más lentos, es decir aquellos con parte real más cercana a 1.

Ejemplo 1.1 (Código C para saturación).

```

if (act >= sat_pos)
    act = sat_pos;
else
    if (act <= sat_neg)
        act = sat_neg;

```

1.7. Saturación y estructuras anti-enrollamiento

En la vida real es inevitable que las variables de actuación tengan algún tipo de límite, ya sea de magnitud y/o de tasa de variación entre otras. Estas limitaciones reales deben aplicarse a la salida de los controladores con el fin de evitar problemas de implementación.

La limitación más recurrente es el de la saturación en magnitud. Esto se debe a que las actuaciones, como por el ejemplo el máximo voltaje que se puede aplicar a la carga, generalmente está limitado por motivos prácticos. Desarrollar un código que implemente la saturación es una tarea bastante trivial, tal como se muestra en el Ejemplo 1.1.

Cabe destacar que el hecho de que un sistema aplique su valor máximo de actuación no es un problema en sí mismo, sólo quiere decir que el controlador está tratando de aplicar una gran actuación de manera de alcanzar los valores de referencia de acuerdo a los criterios del diseñador. Ahora bien, al saturarse el sistema, se hace evidente una no-linealidad del sistema, por lo que es de esperar que el comportamiento del sistema completo difiera del comportamiento esperado al considerar al sistema como un sistema lineal.

El problema surge al tratar de “informar” al controlador que se ha alcanzado el nivel de saturación y que, en términos coloquiales, no sirve de nada seguir intentando subiendo la actuación, ya que ésta ya ha alcanzado su máximo valor. Si esta acción no se realiza, el controlador seguirá subiendo su actuación y, más importante aún en el caso de controladores con algún tipo de integración, seguirá acumulando error. Este proceso se conoce con el nombre de *enrollamiento* y puede causar oscilaciones importantes en la actuación y en la salida del sistema, tal como se muestra en la Fig. 1.9. En esta figura, en azul se muestra la respuesta del sistema cuando no se considera un esquema de antienrollamiento, mientras que en verde se muestra la respuesta ante el mismo controlador, pero agregando un esquema de antienrollamiento.

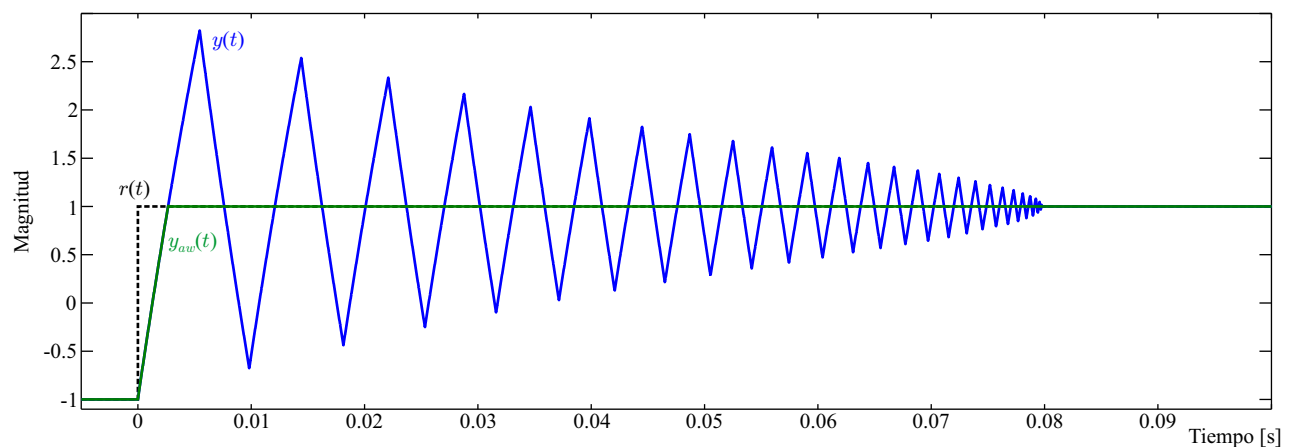


Figura 1.9. Respuesta de un sistema controlado ante un escalón de referencia de -1 a 1, sin esquema de antienrollamiento ($y(t)$) y con él ($y_{aw}(t)$).

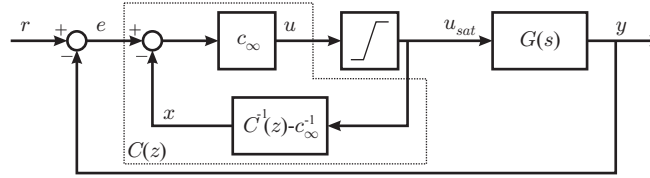


Figura 1.10. Estructura de controlador para evitar el fenómeno del enrollamiento.

A grandes rasgos, la explicación de estas oscilaciones debido al *enrollamiento* es la siguiente: Supongamos que debido a un aumento en la referencia se genera una señal de error positiva, luego, el controlador aumentará su actuación con el fin de llevar a la planta a su nueva referencia. Si la actuación del controlador llega a su nivel máximo, la salida de la planta seguirá aumentando, pero más lento que si se estuviera aplicando una actuación de mayor valor, por lo que se tardará más de lo diseñado en alcanzar el valor de referencia. Durante todo este tiempo, el controlador seguirá acumulando error positivo y por tanto aumentando su solicitud de actuación (la cual la planta real no puede alcanzar). Una vez alcanzada la nueva referencia, el controlador debería ser capaz de encontrar rápidamente el punto de equilibrio necesario para mantener la salida estable, sin embargo, producto del gran error acumulado tiene un nivel de inercia muy grande y necesita que el error sea negativo durante bastante tiempo para poder eliminar todo el error acumulado, esto provocará que la salida sobrepase largamente la referencia, creado un error negativo. Cuando el error acumulado en la acción integral llega a cero, la salida de la planta estará bastante por sobre el valor deseado, por lo que ahora comienza una acumulación negativa de error, que disminuye la actuación haciendo que la salida baje. Dependiendo de la planta y el controlador, esto puede llevar al controlador a solicitar una actuación muy negativa, generando una saturación por abajo del sistema. Esto puede ocasionar una serie de oscilaciones e incluso llevar al sistema a la inestabilidad. Las estructuras anti-enrollamiento “informan” al controlador sobre el estado real de la actuación aplicada de manera de evitar que la acción integral siga acumulando error si ya se ha alcanzado el nivel de saturación. De esta manera se corta el ciclo de acumulaciones excesivas de error y por ende las oscilaciones en la salida de la planta, tal como se muestra en la Fig. 1.9.

Una estructura de anti-enrollamiento que permite enfrentar exitosamente todo tipo de saturaciones, y valida para cualquier estructura de controlador bipropia y que no tenga ceros en el semiplano derecho, se muestra en la Fig. 1.10. Donde:

- $C(z)$: es la función de transferencia del controlador.
- c_∞ : Es la ganancia proporcional del controlador $C(z)$, o de manera más formal:

$$c_\infty = \lim_{z \rightarrow \infty} C(z). \quad (1.17)$$

Con esta estructura, los resultados cuando el sistema no se encuentra saturado, son idénticos, tal como muestra la Fig. 1.9, sin embargo las diferencias son claras una vez que la referencia es alcanzada. Es evidente que los resultados obtenidos con la estructura anti-enrollamiento son mejores que aquellos sin esta estructura.

A nivel de código, para un controlador tipo *PI*, la estructura de la Fig. 1.10 se muestra en el Ejemplo 1.2, donde $C1$ y $C2$ se determinan a partir de los coeficientes del controlador *PI*.

Ejemplo 1.2 (Código C para la un PI con una estructura anti-enrollamiento).

```
e_act = ref - med;
x_act = C1 * x_ant + C2 * u_ant;
u_act = Kp * (e_act - x_act);

if (u_act > sat_pos)
    u_act = sat_pos;
else
    if (u_act < sat_neg)
        u_act = sat_neg;

u_ant = u_act;
x_ant = x_act;
```

Capítulo 2

Control Lineal de Corriente en Convertidores.

En prácticamente cualquier convertidor de potencia el lazo de control más interno es el de la corriente del sistema. Este lazo es por tanto el de dinámicas más rápidas y el que puede tener efectos más complejos en caso de un mal funcionamiento.

En esta sección se discutirán algunos de los elementos más relevantes a tener en cuenta para el diseño de estos controladores, tomando como base el control de corrientes trifásicas alternas en una carga de tipo $R - L$.

2.1. La planta

Las ecuaciones generales de una carga trifásica balanceada son las siguientes:

$$v_{aN} = R i_a + L \frac{di_a}{dt} + v_{nN} \quad (2.1)$$

$$v_{bN} = R i_b + L \frac{di_b}{dt} + v_{nN} \quad (2.2)$$

$$v_{cN} = R i_c + L \frac{di_c}{dt} + v_{nN} \quad (2.3)$$

En estos sistemas las referencias a seguir son senoidales de magnitud variable y, en la mayoría de las ocasiones, de frecuencia variable⁵. Como se discutió en el capítulo anterior, no es posible diseñar un controlador lineal que cumpla con estas características, lo que obliga a realizar algunas transformaciones matemáticas a (2.1)-(2.3) de manera de que, **en estado estacionario**, este sistema pueda ser representado por un conjunto de variables continuas. Si esto se logra, entonces el sistema puede ser controlado a partir de un par de controladores PI .

El conjunto de transformaciones lineales que implementan este cambio de coordenadas son la transformada de Clarke y la de Park:

⁵La excepción más normal son las aplicaciones conectadas a la red.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (2.4)$$

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} \quad (2.5)$$

$$\theta(t) = \int_0^t \omega_r(\tau) d\tau + \theta(0) \quad (2.6)$$

Básicamente, (2.4) pasa de un sistema trifásico a bifásico en cuadratura. En el caso de que las corrientes correspondan a un sistema trifásico balanceado, i_α e i_β se combinan para formar un vector de corriente de acuerdo con:

$$\vec{i}_{\alpha\beta} = i_\alpha + j i_\beta, \quad (2.7)$$

entonces, $\vec{i}_{\alpha\beta}$ describe una trayectoria circular que gira con respecto al origen a frecuencia ω . Por otra parte, (2.5) rota el eje coordenado según el ángulo θ . Si θ varía con la misma velocidad y sentido con el que gira el vector de corriente ($\omega = \omega_r$), entonces $\vec{i}_{\alpha\beta}$ aparece estacionario frente a este nuevo eje coordenado rotatorio, transformándose en \vec{i}_{dq} .

2.2. Diseño del controlador

La Fig. 2.1 muestra el diagrama general de control para un convertidor.

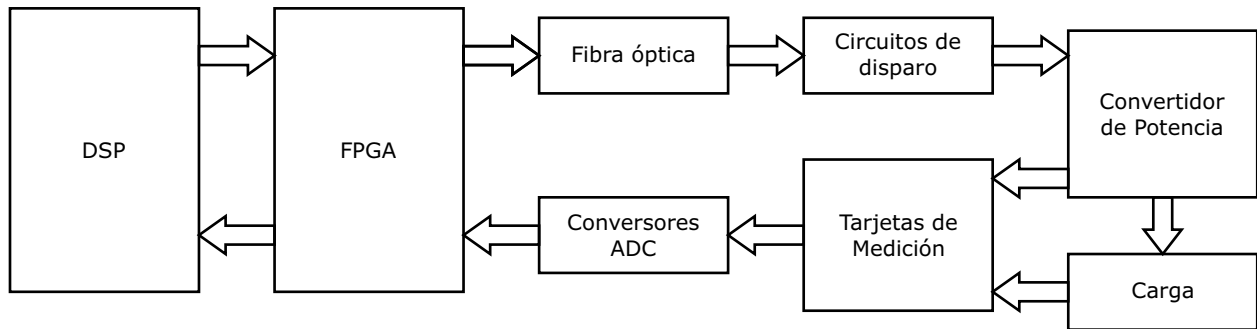


Figura 2.1. Diagrama general del esquema del sistema.

El DSP determina los voltajes que debe aplicar el convertidor, y envía los respectivos valores a la FPGA. Ésta toma estos valores y los transforma en pulsos de disparo a través de sus moduladores PWM. Con el fin de lograr una mayor inmunidad al ruido electromagnético y para aislar la etapa de control de la de potencia, es frecuente utilizar un enlace de fibra óptica entre la FPGA y los circuitos de disparo⁶. Estos últimos reciben la señal de la fibra óptica, y la transforman nuevamente en señales eléctricas que son enviadas a los respectivos terminales de gate de los semiconductores

⁶En algunos casos los pulsos se pueden enviar en corriente o en voltaje.

de potencia. Las señales relevantes de la carga y del convertidor son medidas a partir de tarjetas de medición específicas (voltaje, corriente, velocidad, torque, etc...), las que adaptan las señales a los rangos de operación de los conversores AD. Los datos de las conversiones son enviadas por la FPGA al DSP, quien recalcula las actuaciones en base a los algoritmos de control implementados y el ciclo se repite.

Tal como se mencionó en sección 1.2.4, para poder diseñar apropiadamente el controlador, es necesario tener un modelo lineal que relacione las variables de entrada (voltaje) y salida (corrientes) del sistema. En este caso, a partir de (2.1)-(2.3) y las transformaciones (2.4) y (2.5), las relaciones en el sistema de coordenadas rotatorio es:

$$v_d = R i_d + L \frac{di_d}{dt} + \omega L i_q \quad (2.8)$$

$$v_q = R i_q + L \frac{di_q}{dt} - \omega L i_d \quad (2.9)$$

Luego, el diagrama de bloques del sistema es el mostrado en la Fig. 2.2(a).

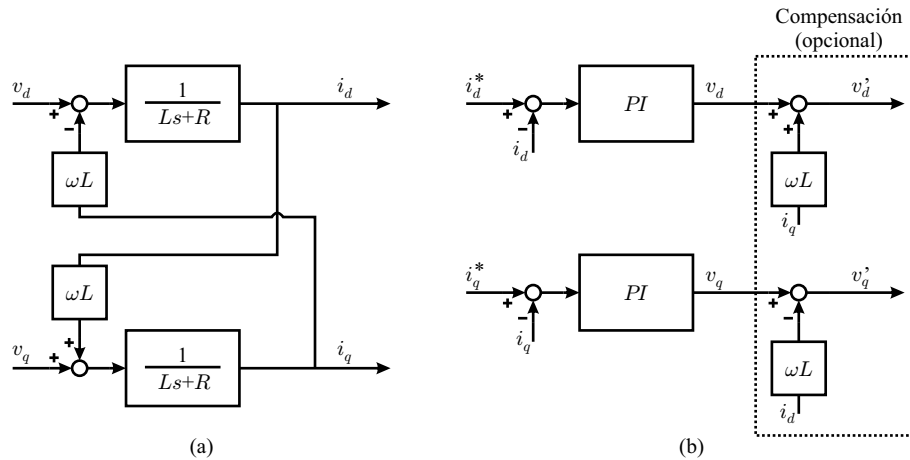


Figura 2.2. Diagrama de bloques de una carga $R - L$ trifásica balanceada en el eje rotatorio dq .

Se aprecia que existe un acople entre los canales d y q . En estado estacionario tanto i_d como i_q serán constantes, por lo que su efecto de *perturbación* será compensado por un controlador tipo PI . A pesar de esto, dado que la perturbación es conocida, también es posible realizar una compensación a la salida del controlador de voltaje [Fig. 2.2(b)], lo que mejora en algo el desempeño del sistema.

Para determinar los parámetros de los controladores PI se sugiere realizar el diseño utilizando la herramienta *rltool* explicada en el punto 1.6. Sin embargo, es necesario considerar lo siguiente:

1. Es necesario contar con un modelo discreto de la planta. Para esto utilice el comando $Gz=c2d(Gs,h)$ de *Matlab*, donde G_s es la planta en tiempo continuo y h el tiempo de muestreo de su controlador. Este comando entregará el modelo discreto de la planta, G_z , asumiendo que la actuación del sistema se mantiene constante durante el periodo h (retentor de orden cero).
2. En la experiencia 2 de este laboratorio, se planteó la ventaja de realizar la carga de las referencias de los moduladores de manera sincrónica con los máximos y/o mínimos de las señales triangulares. Para lograr esto se introdujo el concepto de los registros *shadow*. Una consecuencia de esta carga sincrónica es que las actuaciones del sistema calculadas a partir de las mediciones en el tiempo t , se aplicarán a la carga en el instante $t + h$, es decir, con **un**

retardo de un período de muestreo, por lo que el sistema real a controlar es de la forma:

$$G'_z = z^{-1}G_z \quad (2.10)$$

Una vez obtenidos los parámetros de los controladores, se recomienda utilizar el código del ejemplo 1.2. Tenga la precaución de que la saturación no es por componente individual (v_d o v_q), si no que de la magnitud del vector, correspondiente a $\sqrt{v_d^2 + v_q^2}$, por lo que deberá elegir entre saturar proporcionalmente o privilegiar un eje por sobre el otro.

2.3. Implementación

2.3.1. Generación de referencias

Una vez calculado el voltaje que se debe aplicar, es necesario escalarlo y dejarlo en un formato apropiado. Cabe recordar que las señales triangulares de los moduladores PWM van desde 0 a CMAX (definido por el usuario). Luego, si la actuación del controlador se encuentra en el rango

$$v_o^* \in [\check{v}, \hat{v}], \quad (2.11)$$

el valor a enviar al modulador debe ser ajustado de acuerdo con

$$\text{cuenta} = (\text{int}) \left(\text{CMAX} \frac{v_o^* - \check{v}}{\hat{v} - \check{v}} \right) \quad (2.12)$$

2.3.2. Fibra óptica

Para utilizar la tarjeta de fibra óptica (Fig. 2.3), es necesario que el firmware de la FPGA tenga redireccionado apropiadamente las salidas de los moduladores. Por otro lado, cada tarjeta de fibra óptica cuenta con hasta 12 salidas (D1 a D12), cada una de estas salidas puede conectarse a uno de entre 5 o 4 pines de la FPGA. Para seleccionar que pin (y por ende qué señal de modulador) será utilizado, se debe soldar una resistencia de 0Ω en el pad correspondiente al pin seleccionado.

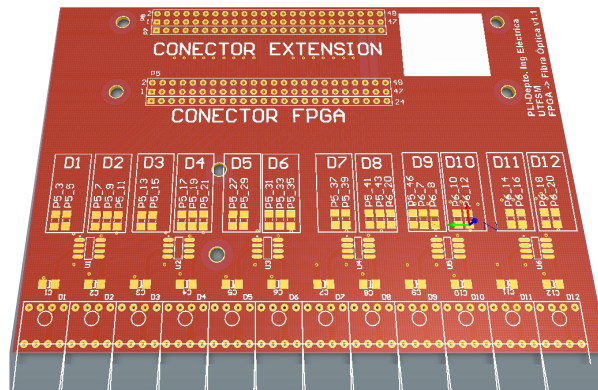


Figura 2.3. Tarjeta de fibra óptica.

2.3.3. Tarjeta de circuitos de disparo

La tarjeta de circuitos de disparo mostrada en la Fig. 2.4 cumple con varias funciones:

1. Transformar las señales de luz provenientes de la fibra óptica en señales de corriente/voltaje.
2. En caso de que se requiera, generar una señal cuasicomplementaria a la recibida en las entradas *Principal*, agregando un tiempo muerto tanto a la señal original como a la señal cuasicomplementaria. Para esto debe soldarse una resistencia de 0Ω entre los pines 3 y 1 de los elementos S_{i_tmx} .
3. Generar hasta cuatro pulsos de disparo de $\pm 12V$ aislados entre si entre los terminales de *Gate* y *Emit* respectivos, en función de las señales *Principal* (si se utiliza el generador de tiempo muerto de la placa) o de las señales de los cuatro receptores de fibra óptica en caso de que se usen estos pulsos de manera directa (resistencia de 0Ω entre los pines 2 y 1 de los elementos S_{i_tmx})
4. El tiempo muerto se configura a partir de los potenciómetros R_{i_tmx} y va desde unos pocos ns a los $2\mu s$ aproximadamente.

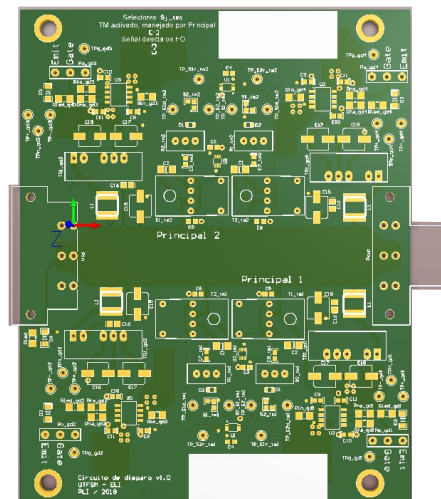


Figura 2.4. Tarjeta de disparo con opción de generación local de tiempo muerto.

Capítulo 3

Actividades evaluadas.

1. Utilizando Matlab Simulink simule un inversor de dos niveles, diseñe un esquema de control como el visto en esta guía y compruebe el correcto funcionamiento realizando:

- Escalones de magnitud y frecuencia (en el rango 20Hz-50Hz).
- Cambios en la resistencia de carga en torno al 10 %.

Para esto utilice un modelo simple del convertidor, que consista en una ganancia cuya salida valga V_{cc} si la salida del modulador es 1, y 0 si la salida del modulador es 0.

Los parámetros del sistema son:

Param	Valor
R	10Ω
L	10mH
V_{cc}	300V
f_T	5kHz

2. Diseñe un esquema de control basado en un controlador proporcional-resonante a 50Hz que controle directamente en $a - b - c$ y compruebe su funcionamiento realizando escalones de referencia de magnitud.
3. Implemente los dos controladores antes mencionados en la plataforma experimental disponible en el laboratorio. Para el primer caso, las referencias de magnitud y frecuencia serán enviados desde Matlab, a través del HPI. En el segundo caso, sólo la magnitud será enviada por la misma vía.

En ambos casos, la plataforma deberá enviar a Matlab, cada vez que se le requiera, 100ms de mediciones de la señal de referencia, la actuación y la medición de corriente de al menos una fase.

Además genere códigos que permitan realizar escalones de magnitud y frecuencia, según corresponda, teniendo especial cuidado en almacenar los datos de al menos dos ciclos de la señal **antes** de que se efectúe el escalon y suficientes ciclos **después** del escalón de manera de que se haya alcanzado el estado estacionario.