

CONTROL DIGITAL - MAESTRÍA EN SISTEMAS EMBEBIDOS

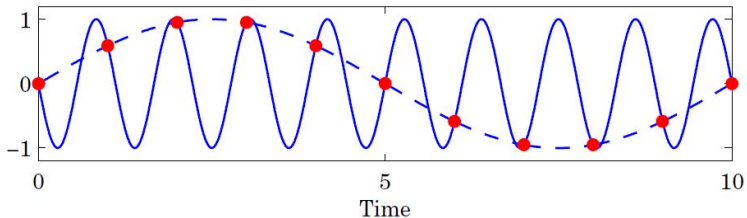
ASPECTOS DE LA IMPLEMENTACIÓN

20 DE MARZO 2023

- 1 Muestreo y Aliasing
- 2 Selección del periodo de muestreo
- 3 Retardo computacional
- 4 Consideraciones Numéricas

MUESTREO Y ALIASING

MUESTREO Y ALIASING



$$y_1(t) = \sin(1,8\pi t - \pi)$$

$$y_2(t) = \sin(0,2\pi t)$$

$$h=1, \omega_s = 2\pi \implies$$

$$\sin(0,2\pi kh) = \sin(1,8\pi kh - \pi) = \sin(2,2\pi kh) = \sin(3,8\pi kh - \pi) \dots$$

Muestrear una señal con frecuencia ω crea nuevas componentes de frecuencia

$$\omega_{muestreada} = \pm\omega + n\omega_s$$

donde $\omega_s = 2\pi/h$ es la frecuencia de muestreo y n es un número natural.

Entonces, la frecuencia ω es el alias de

$\omega_s - \omega, \omega_s + \omega, 2\omega_s - \omega, 2\omega_s + \omega \dots$

La frecuencia de Nyquist es:

$$\omega_N = \omega_s/2$$

Es un pasa-bajo que atenúa todas las frecuencias por sobre la frecuencia de Nyquist antes de muestrear. ¡Debe contener parte analógica!

Opciones:

- Filtro Analógico puro

- ▶ Por ejemplo, Filtros Bessel o Butterworth de 2-6 orden.
- ▶ En estos casos es difícil cambiar el intervalo de muestreo.

- Filtro analógico + digital

1. Se tiene un filtro analógico con frecuencia fija, la salida de ese filtro se muestrea *rápido*.
2. Se hace un submuestreo con un filtro PB digital.
3. El algoritmo de control trabaja a la tasa más baja de muestreo. Esto permite cambiar fácilmente el intervalo de muestreo.

Como regla general, considerar que las frecuencias por encima de ω_N deben ser atenuadas por un factor 10:

$$|G_f(j\omega_N)| \leq 0,1$$

Es probable que el filtro afecte el margen de fase del sistema (es importante analizar ese punto).

Si lo modifica significativamente \rightarrow incluir su dinámica en el modelo del proceso.

SELECCIÓN DEL PERIODO DE MUESTREO

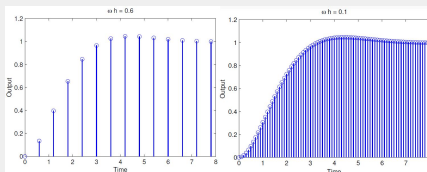
PERÍODO DE MUESTREO - DISEÑO DIGITAL

Regla general:

$$f_{LC}h \approx 0,1 \text{ a } 0,6$$

donde ω es la frecuencia deseada del sistema a lazo cerrado.

Esta elección nos da de 4 a 20 muestras por tiempo de subida (permite una correcta visualización).



El límite inferior de h está dado por el *hardware* utilizado y su tiempo de cómputo asociado (τ_C):

$$\tau_C = T_{ADC} + T_{PROC. ALGORITMO} + T_{DAC}$$

$$h \geq 20\tau_C$$

Vamos a considerar que la frecuencia de operación del sistema, en lazo cerrado, está relacionada a un ancho de banda $f_{LC} = BW$.

Entonces, para hacer los cálculos del TP, una vez que conozcamos el tiempo de subida que deseamos a lazo cerrado $t_{r_{LC}}$, se calcula BW cómo:

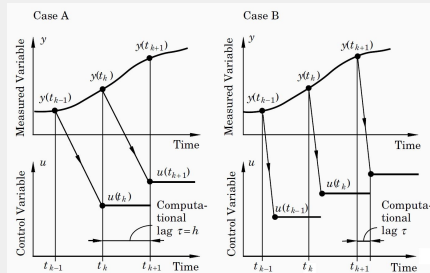
$$BW = 0,35/t_{r_{LC}}$$

RETARDO COMPUTACIONAL

RETARDO COMPUTACIONAL

Problema: $u[k]$ no puede ser generada instantáneamente en el tiempo k cuando $y[k]$ es muestreada. Opciones de solución:

- A. Se miden las variables en t_k y se aplica el control en t_{k+1}
- B. Se miden las variables en t_k y se aplica el control lo más rápido posible



Caso A: Retardo de una muestra.

Caso B: Minimizar el retardo computacional.

- El retardo de A es determinístico, aunque puede ser un poco grande.
- El retardo en B es variable y aleatorio, puede causar *jitter*.

Algoritmo

En el caso B, es deseable hacer el mínimo número de operaciones entre la lectura de las variables y la salida de control:

1. Lectura del ADC $y[k]$; $r[k]$
2. Cálculo de $u[k]$
3. Conversión DA de $u[k]$
4. Actualización de las demás variables ...

Una buena regla es leer las entradas antes de aplicar la salida.

CONSIDERACIONES NUMÉRICAS

En el análisis y diseño de control usualmente se asume precisión infinita en los cálculos aritméticos.

Todos los parámetros y variables se asumen como números reales.

Posibles fuentes de error en una implementación digital con largo de palabra finito:

1. Cuantización en los conversores A-D.
2. Cuantización en los parámetros (coeficientes del controlador).
3. Redondeo y *overflow* en las operaciones matemáticas.
4. Cuantización en el conversor D-A.

La magnitud de los problemas depende de:

1. El largo de la palabra.
2. El tipo de dato utilizado (punto fijo o flotante).
3. La realización del controlador.

- Es conveniente que la resolución del ADC sea mayor que la del DAC.
 - ▶ A-D: 10 - 16 bits.
 - ▶ D-A: 8 - 12 bits
- Debido a que es complicado probar como afectan al sistema, se realizan simulaciones para tenerlos en cuenta

La cuantización es un fenómeno no lineal, puede ocasionar ciclos límites. Se puede analizar con (fuera del alcance del curso):

- Análisis no lineal:
 - ▶ Aproximación por funciones descriptivas.
 - ▶ Teoría de oscilaciones con relé.
- Análisis Lineal
 - ▶ Tratar la cuantización como una perturbación estocástica.

EJEMPLO DEL EFECTO DE CUANTIZACIÓN

Proceso:

$$P(s) = 1/s^2$$

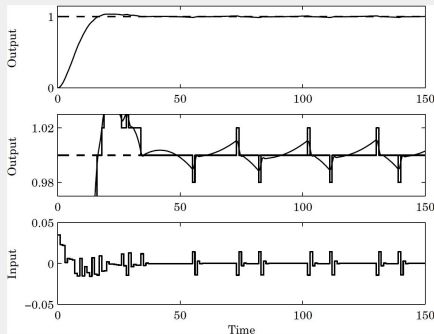
Intervalo de muestreo:

$$h = 1s$$

Controlador:

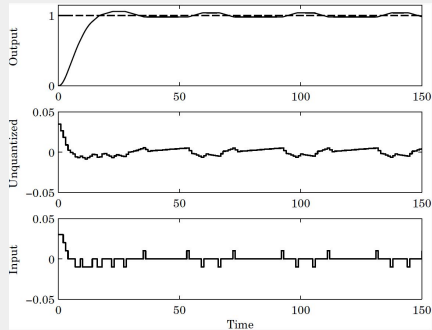
$$C(z) = \frac{0,715z^2 - 1,281z + 0,58}{(z - 1)(z + 0,188)}$$

SIMULACIÓN CON UNA CUANTIZACIÓN EN EL CONVERTOR A-D ($\delta y = 0,02$)



Se produce un ciclo límite con período 28 s y amplitud 0.01.
(El cual puede predecirse si se hace una descripción con función descriptiva).

SIMULACIÓN CON UNA CUANTIZACIÓN EN EL CONVERTOR A-D ($\delta u = 0,01$)



Se produce un ciclo límite de período 39 s y amplitud 0.01.

- El tipo de dato seleccionado puede inestabilizar al sistema
- Para analizar cómo se modifica el sistema, simular teniendo en cuenta el tipo de datos.
- Tener en cuenta que los errores de cuantización y redondeo generan pequeños errores, pero los errores de overflow pueden ser desastrosos.
- Las operaciones con largo de palabra finito no son asociativas ni distributivas.

Con punto flotante hay que ser cuidadoso y con punto fijo un poco más.

PUNTO FIJO

Conviene representar todos las variables utilizando **enteros**.

Utilizar una **escala binaria** para hacer que todos los números entren en uno de los tipos de datos enteros, ej.

- 8 bits (char, int8_t) $[-128, 127]$
- 16 bits (short, int16_t) $[-32768, 32767]$
- 32 bits (long, int32_t) $[-2147483648, 2147483647]$

Simular cómo se modifica el algoritmo cuando se cambia el tipo de datos

Punto Fijo + Octave

```
>> fixed(is,ds,f)
```

is: Cantidad de bits parte entera; ds: Cantidad de bits parte decimal; f: Valor que se desea convertir

- Se debe seleccionar el tipo de datos con suficiente precisión numérica.
- Se debe saber (o estimar) el mínimo y máximo valor de cada variable para poder seleccionar los factores de escala apropiados.
- Se debe llevar esa escala a todos los operadores aritméticos.
- Revisar si esa escala no provoca *overflow*.

Un controlador lineal

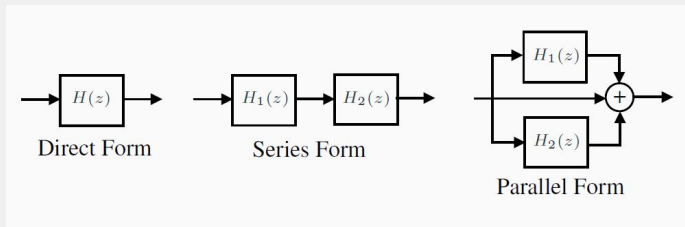
$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

puede ser realizado de diferentes maneras con un comportamiento equivalente de entrada - salida.

- Forma directa (así como viene).
- En serie.
- En paralelo.

REALIZACIONES

Si la forma directa genera complicaciones, se puede dividir la función del controlador en controladores de primer o segundo orden:



Consejo: Tratar de regular la ganancia de cada bloque así tiene cada uno la misma amplificación.