

Programación en Lenguaje Java

Práctica 3.4. Cálculos con señales cuadradas



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

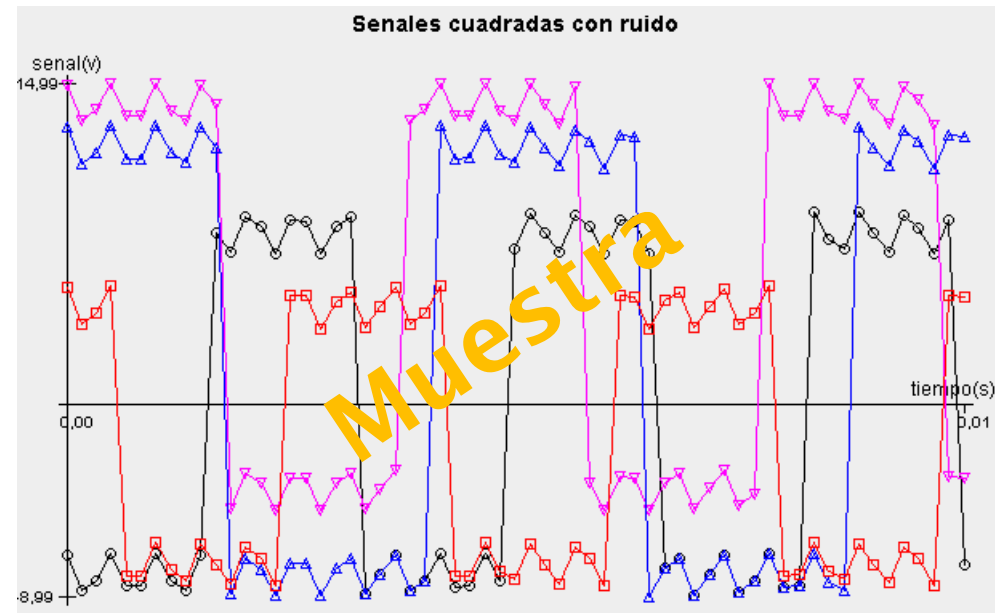
Práctica 3-4: Cálculos con señales cuadradas

Objetivos: Practicar con instrucciones condicionales y de bucle

Descripción: Se dispone de un sistema experimental que mide cuatro señales eléctricas periódicas, de tipo cuadrado con ruido

Se dispone de la clase **Sensores** que simula el funcionamiento de los cuatro sensores

- Las características de las cuatro medidas son:
 - Valores medios (+-1v, por las fluctuaciones): 0.0, 5.0, 2.0, -1.5 v
 - Periodos: 2.0,2.4,2.8,2.2 ms



Clase Sensores

Métodos:

- *Constructor*: crea el objeto, realiza las medidas y las almacena internamente
- `rebobina()`: hace que la medida actual recogida mediante el método `medida()` sea la primera realizada
- `medida()`: retorna en voltios el valor de la medida actual del sensor indicado por el parámetro `numSensor`, que debe estar entre 0 y 3
- `tiempo()`: retorna en segundos el tiempo actual al que corresponden las medidas obtenidas por el método `medida()`
- `avanza()`: si quedan más medidas avanza la medida actual a la siguiente y retorna `true`; en caso contrario, retorna `false`

Sensores
...
+Sensores() +void rebobina() +double medida(int numSensor) +double tiempo() +boolean avanza()

Recorrido de las medidas de la clase Sensores

Usando un objeto de esta clase llamado `sensores`, para leer las medidas del sensor número `i` usar este pseudocódigo:

```
sensores.rebobina()  
hacer  
    usar sensores.medida(i) y sensores.tiempo()  
mientras sensores.avanza()
```

Código a realizar

Se desea crear una clase llamada `Medidas` con métodos para obtener información de las medidas de estas señales eléctricas

También se creará otra clase llamada `MuestraMedidas` con un `main` para mostrar información sobre las medidas

Clase Medidas

Atributos:

- **sens**: objeto de la clase **Sensores**

Métodos:

- **Constructor**: copia en **sens** la referencia al objeto que se pasa como parámetro
- **pintaGraficas()**: muestra las medidas de los cuatro sensores en un objeto de la clase **Grafica** del paquete **fundamentos**
 - para cada sensor recorre todas sus medidas e inserta en la gráfica parejas tiempo-medida
 - cambia de gráfica (con **otraGrafica()**) al cambiar de un sensor a otro
 - al final de todo pinta las gráficas con el método **pinta()**

Medidas
-Sensores sens
+Medidas (Sensores sens) +void pintaGraficas() +double media(int numSensor)

Clase Medidas (cont)

- `media()`: retorna el valor medio de la señal del sensor indicado por `numSensor`; si este parámetro no está entre 0 y 3 retorna `Double.NaN` para indicar el error
 - para calcular la media se contabilizan todas las medidas del sensor indicado

$$media(sensor) = \frac{\sum_{\forall i} medida(sensor)}{num}$$

- donde `num` es el número de medidas
- `num` se puede calcular con un contador que se incrementa durante el recorrido de las medidas en una unidad a cada paso

Clase MuestraMedidas

Esta clase contiene el programa principal que hace:

- Crea un objeto de la clase `Sensores`
- Crea un objeto de la clase `Medidas` pasándole como parámetro el objeto anterior
- Dibuja las gráficas de las medidas invocando a `pintaGraficas()`
- Usando un bucle para cada sensor muestra en pantalla el resultado de invocar a su método `media()`

Parte avanzada

Añadir a la clase `Medidas` un método que calcule y retorne el periodo de la señal del sensor que se pasa en el parámetro `numSensor`; si este parámetro no está entre 0 y 3 retorna `Double.NaN` para indicar el error

Añadir también al main instrucciones para mostrar en pantalla los periodos de las señales de los cuatro sensores

Cálculo del periodo:

- el periodo se obtiene calculando el tiempo transcurrido entre dos flancos de subida de la señal
- un flanco de subida ocurre cuando la señal pasa de valer menos que la media a valer mas que la media
- se puede usar este pseudocódigo para calcular el periodo:

```
real med=media(numSensor)
sens.rebobina()
// buscar el primer flanco de subida
buscarFlanco(numSensor,med)
real tIni=sens.tiempo()
// buscar el segundo flanco de subida
buscarFlanco(numSensor,med)
real tFin=sens.tiempo()
retorna tFin-tIni
```

Cálculo del periodo (cont)

- donde para buscar un flanco se puede usar el método privado que aparece a continuación

```
método buscarFlanco(entero numSensor, real med)
  // buscar nivel bajo
  mientras sens.medida(numSensor)>med
    sens.avanza()
  fin mientras
  // ahora buscar un nivel alto
  hacer
    sens.avanza()
  mientras sens.medida(numSensor)<med
fin método
```

Entrega

Entregar un informe en pdf que contenga:

- Código de la clase `Medidas` (5 puntos)
- Código del `main` (1 punto)
- Captura de pantalla de los resultados de la ejecución del `main`, incluyendo las gráficas y las medias (2 puntos)
- Parte avanzada: código del método `periodo()` (1 punto)
- Parte avanzada: captura de pantalla de la ejecución del `main` que muestre los periodos obtenidos (1 punto)