

Programación en Lenguaje Java

Práctica 2.4. Movimiento oscilatorio forzado



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

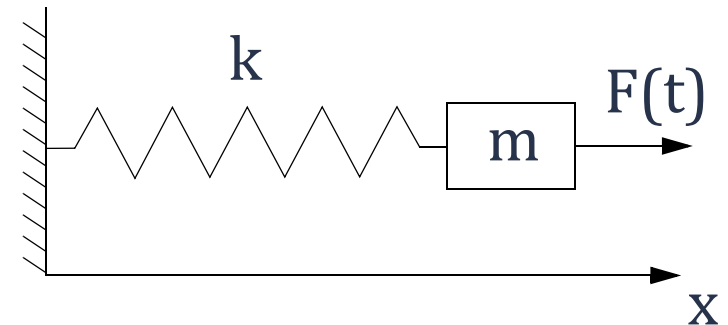
[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 2-4: Movimiento oscilatorio forzado

Objetivo: crear una clase con atributos y métodos sencillos, y un programa principal de prueba

Descripción: Queremos hacer un programa para calcular los movimientos de un muelle sometido a una fuerza externa, sin rozamiento

Es un movimiento oscilatorio forzado de una masa m , un muelle de constante k con coeficiente de fricción nulo y sometido a una fuerza externa $F=F_0\cos(\omega t)$



Diseño: clases

- `SistemaMasaMuelle`: representa el sistema masa-muelle
- `PruebaMasaMuelle`: es un programa con un `main` para probar la clase anterior
- `SimulaMuelle`: es otro programa que hace una gráfica del movimiento; se da ya hecho



Clase SistemaMasaMuelle

Atributos:

- **k**: constante del muelle en N/m
- **m**: masa en Kg
- **f0**: amplitud de la fuerza externa, en N
- **w**: frecuencia angular de la fuerza externa, en rad/s

Métodos

- Constructor: copia los parámetros en los atributos del mismo nombre

SistemaMasaMuelle

-double k, m
-double f0, w

+SistemaMasaMuelle
(double k, double m
double f0, double w)
+double posicion(double t,
double fase, double c)

Clase SistemaMasaMuelle (cont.)

- `posicion()`: calcula y retorna la posición de la masa, x , en metros, dados el tiempo t en segundos, el ángulo `fase` (γ), en grados y la constante c , en metros, que determina las condiciones iniciales; se utiliza esta fórmula:

$$x = C \cdot \cos(\omega_0 t - \gamma) + \frac{F_0}{m(\omega_0^2 - \omega^2)} \cos(\omega t)$$

- observar que los valores C y F_0 de la fórmula son las variables `c` y `f0`
- la frecuencia propia, ω_0 se obtiene con la fórmula:

$$\omega_0 = \sqrt{k/m}$$

- la `fase` γ viene en grados y debe convertirse a radianes

Probar la clase ejecutando el `main` de la clase `SimuladorMuelle` que se proporciona

Clase PruebaMasaMuelle

Crear esta clase con un método `main` que hace lo siguiente:

- Crea un par de objetos de la clase `SistemaMasaMuelle` con estos datos:

<code>k</code>	<code>m</code>	<code>f0</code>	<code>w</code>
75.0	0.5	12.0	5.0
120.5	1.2	15.5	6.0

- Muestra en pantalla las posiciones de cada objeto con estos datos (son 4 posiciones en total):

<code>t</code>	<code>fase</code>	<code>c</code>
1	45.0	0.3
2	60.0	0.3

Parte avanzada

a) Añadir a la clase `SistemaMasaMuelle` el siguiente método:

- `frecuenciaPropia()`: calcula y retorna la frecuencia angular propia del muelle, ω_0 , en radianes por segundo, según:

$$\omega_0 = \sqrt{k/m}$$

b) Modificar el método `posicion()` para que utilice el nuevo método para calcular la frecuencia propia

c) Modificar el programa principal añadiéndole instrucciones que muestren en pantalla las frecuencias propias de los dos objetos creados

Entrega

- Código de la clase `SistemaMasaMuelle` (5 puntos)
- Captura de pantalla de la ejecución de `SimuladorMuelle` (1 punto)
- Código de la clase `PruebaMasaMuelle` (1 punto)
- Captura de pantalla de la ejecución de `PruebaMasaMuelle` (1 punto)
- Código de la parte avanzada (clases `SistemaMasaMuelle` y `PruebaMasaMuelle` modificadas) (1 punto)
- Captura de pantalla de la ejecución del programa de la parte avanzada (1 punto)

Observaciones

- Recordar poner comentarios de documentación en cada clase y método, documentando lo que son sus parámetros y valores retornados si los hay
- Comentar las unidades en que se expresa cada variable (atributos, parámetros), los valores retornados, y los valores que se saquen por pantalla