

Periodo de oscilación de un muelle.

MAGALY POLETH QUISHPE CHIMARRO

E-mail: magaly.poleth.quishpe@estudiantupc.edu

RESUMEN

El periodo de oscilación que experimenta un muelle al aplicarse una fuerza en uno de sus extremos se puede determinar considerando el MAS. Para determinar el tiempo que tarda el muelle en completar un ciclo mientras experimenta la deformación se presenta el siguiente documento. Este trabajo buscará sintetizar como desarrollar un dispositivo de medición que permita obtener el periodo de oscilación. Además, considerando su ecuación fundamental se determinará la relación directa que existe entre la rigidez del muelle y la masa aplicada en los resultados obtenidos. Para la experimentación se utilizó la plataforma ArduinoR3, sensor de ultrasonido y software de procesamiento de datos que permita generar un código de programación para acumular las lecturas; en este caso se optó por la plataforma Matlab que permite desarrollar algoritmos de forma autónoma. Finalmente se puntualiza en la mejora de la sensibilidad del dispositivo y la detección de datos erróneos en las lecturas.

Palabras clave: instrumentación, programación, periodo, muelle.

1. INTRODUCCIÓN

En los últimos años los aparatos de medición, transmisión y automatización se hacen cada vez más rápidos, exactos y durables; deja los procesos de medición y control a costes reducidos. Los aparatos suelen medir las características físicas o químicas, controlar y monitorear todas las variables con profundidad y gran exactitud garantizando la repetibilidad de las medidas y sus resultados.

Seguramente le será fácil imaginar que, si se ejercen fuerzas sobre un muelle, tal que uno de los extremos permanezca fijo, se observará que el muelle se dilata o se comprime, según el sentido de la fuerza. Pero cómo medir el tiempo que tarda en realizar ese ciclo repetitivo, a continuación, se describe el proceso para desarrollar un dispositivo de medición.

2. MARCO TEÓRICO

Si consideramos un sistema conformado por un muelle de rigidez (k) con una partícula de masa (m) y lo soltamos

veremos que el muelle empieza a oscilar. Este se deforma alargándose o acortándose en una magnitud " x " llamada "deformación". Sabiendo que el periodo es lapso que separa dos instantes en los que el sistema se encuentra exactamente en el mismo estado: mismas posiciones, mismas velocidades, mismas amplitudes. Así el periodo es el tiempo que dura un ciclo en volver a comenzar (ver Fig. 1). [1]

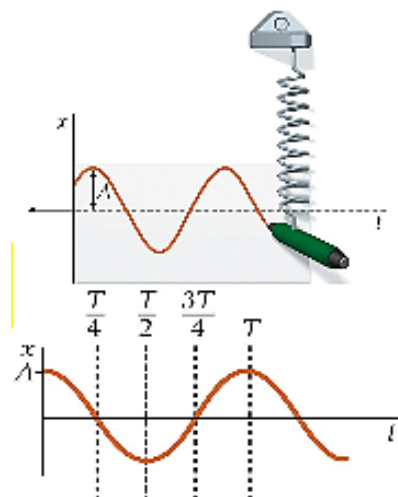


Fig.1: Periodo de un muelle.

Aplicamos la segunda ley de Newton al sistema formado por la partícula de masa (m) y el muelle de constante (k), relacionando las fuerzas actuantes con la aceleración Ec [1]. Considerando un sistema de movimiento armónico simple MAS la frecuencia angular (ω), frecuencia (f) y el periodo (T) se puede determinar mediante las fórmulas Ec [2], Ec [3] y Ec [4] respectivamente. [2][3]

$$-k * x(t) = m * a(t) \quad (1)$$

3. DESARROLLO

Para obtener el periodo de oscilación de un muelle se ha desarrollado un dispositivo de medición que consta de elementos como: placa Arduino UNOR3, cables de circuitos eléctricos, sensor de ultrasonido HC-SR04 y para procesar los datos obtenidos por estos elementos nos hemos apoyado en el software de Matlab.

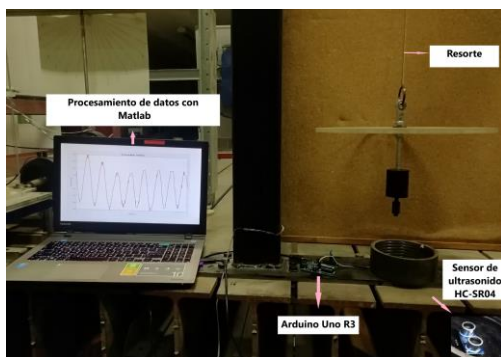


Fig.2: Dispositivo de medición de periodo de un muelle.

A continuación, en la Fig. 3 se indica el esquema de conexión del circuito; posteriormente se detalla las características de los elementos empleados.

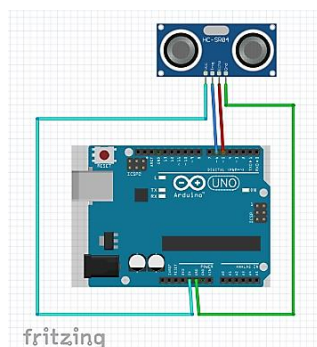


Fig.3: Esquema de conexión del sensor de ultrasonido HC-SR04 con Arduino UNOR3. [4]

$$\omega = \sqrt{\frac{k}{m}} \quad (2)$$

$$f = \frac{1}{T} = \frac{\omega}{2\pi} \quad (3)$$

$$T = 2\pi * \sqrt{\frac{m}{k}} \quad (4)$$

3.1. Plataforma Arduino UNO R3

Microcontrolador con componentes complementarios y puertos de entrada – salida que facilita la programación y la incorporación en otros diseños (hardware de código abierto) ver Fig. 4. Está preprogramado con un administrador de arranque que simplifica la carga de programas en la memoria flash del chip, evitando requerir un hardware adicional. [5]

Entre sus características técnicas principales se puede mencionar:

- Microcontrolador: ATmega328P
- Tensión de funcionamiento: 5V
- Voltaje de entrada: 7-12V
- PWM Digital I / O Pins: 6
- Clavijas de entrada analógica: 6
- Corriente DC por Pin E / S: 20 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria flash: 32KB (ATmega328P) 0.5 KB para carga de arranque.
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Velocidad de reloj: 16 MHz

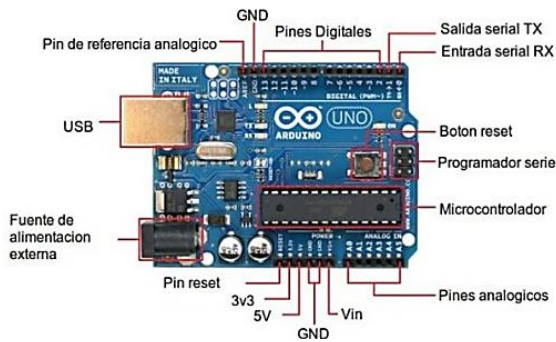


Fig. 4: Arduino UnoR3 [5]

3.2. Sensores de ultrasonido HC-SR04

El sensor utiliza las propiedades de propagación del sonido para medir distancias con un objeto colocado enfrente de este. Este tipo de ondas sonoras se encuentran por encima del espectro audible por los seres humanos como se indica en la Fig. 5.

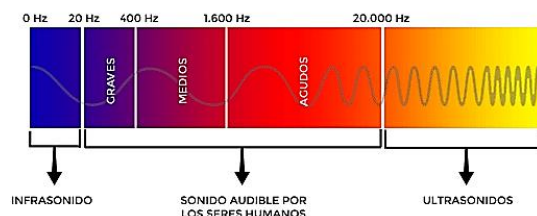


Fig. 5: Tipo de ondas sonoras.

Este sensor consta de 4 pines, alimentación (Vcc), un disparador (Trig), el receptor (Echo) y masa (GND) como se ilustra en la Fig. 6. Estas se conectan a entradas/salidas digitales; por un lado, Trig va a funcionar en modo salida y por otro lado Echo va a funcionar en modo entrada Fig. 7. Para el correcto funcionamiento del sensor es necesario el conectar estos 4 pines.

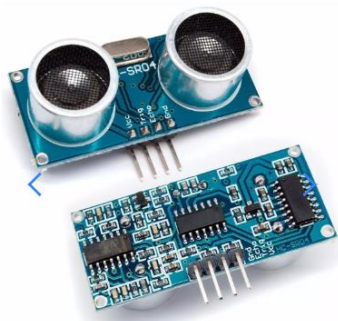


Fig. 6: Sensor de ultrasonido HC-SR04 [6]

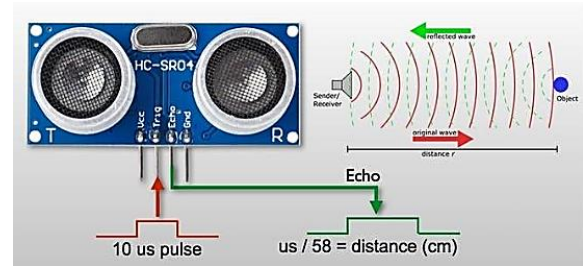


Fig. 7: Funcionamiento de sensor de ultrasonido HC-SR04

Las características de este sensor las indica a continuación:

- Alimentación: +5V DC
- Frecuencia ultrasónica: 40 KHz
- Salida: High 5V/Low 0V
- Consumo (suspendido): <2mA
- Consumo (trabajando): 15mA
- Ángulo efectivo: <15°
- Distancia: 2 – 400 cm (>250 cm resolución no es buena)
- Resolución: 0.3 cm

La distancia (x) entre el sensor y el obstáculo es determinada empleando la Ecu 5. Si se conoce el tiempo de viaje del sonido (t) y considerando que la onda sonora viaja a una velocidad (V) de propagación de 340 m/s a una temperatura de 20°C. Se debe considerar que la velocidad aumenta o disminuye 0,6 m/s por grado centígrado. La ecuación general de la velocidad se divide entre 2 Ec 5, debido a que la señal recorre el doble de la distancia entre la emisión y la recepción.

$$x = \frac{V \cdot t}{2} \quad (5)$$

3.3. Software de procesamiento de datos

Para el almacenamiento, lectura y análisis de datos del sensor Arduino se empleó el software matemático de Matlab. Un sistema interactivo que presenta como elemento básico de datos una matriz que no requiere

dimensionamiento. Este software es utilizado para investigación y para resolver problemas de ingeniería y matemáticas; gran énfasis en aplicaciones de control y procesamiento de señales. Una de las características más importantes es su capacidad de permitir al usuario en un autor contribuyente al crear sus propias aplicaciones. [7]

Previo a conectada la placa Arduino Uno al ordenador a través del puerto USB se debe incorporar el paquete de soporte de Simulink MATLAB para Arduino, permitirá la ejecución de forma autónoma con el dispositivo Arduino. Con el paquete de soporte incorporado, se desarrolla el código de programación para procesamiento de las variables para la obtención del periodo del muelle y la representación gráfica de sus resultados en tiempo real. El código de programación en MATLAB se puede observar en el Anexo 1.

4. ERRORES

En la toma de datos pueden ocasionar errores de lectura, mismos que se detectan en la fase de procesamiento de datos; estos errores se ocasionan se generan por no considerar las características del sensor utilizado. Para discretizar estos valores se debe incorporar en el código de programación condicionantes que evalúen los datos previos a su utilización en cálculos y evitar resultados erróneos. La Fig. 8 indica una interpretación gráfica de los datos leídos.

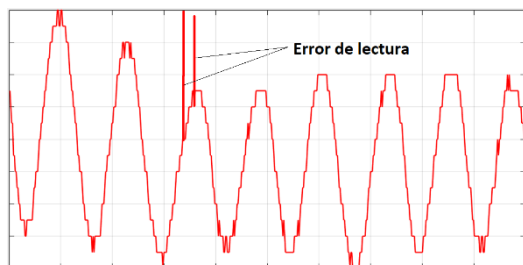


Fig. 8: Identificación de datos con anomalías.

5. SENSIBILIDAD

Se relacionada directamente con la calidad (precisión) de las medidas que se realicen, es la menor división de la escala en

que está graduado el instrumento. cuanto menos dispersos estén los resultados de las medidas la precisión de un aparato de medida será mayor. En la codificación de lectura de datos en Matlab se consideró la función "rate" que establece la tasa de cooperaciones en escaneo de por segundo. Para nuestro código se determinó un escaneo de 57.8 por segundo; este valor puede variar de acuerdo a que tan sensible se desee considerar la medición. La Fig. 9 indica la sensibilidad de los datos procesados en esta experimentación.

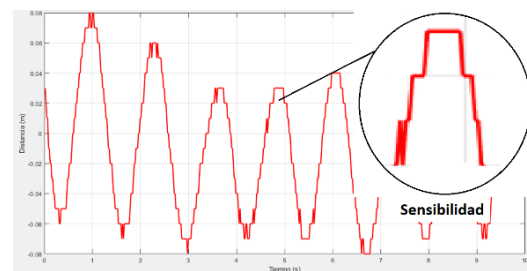


Fig. 9: Sensibilidad de datos.

6. RESULTADOS

Para la obtención del periodo del muelle por este método experimental, no es necesario incluir el valor de la rigidez de cada muelle. Pero por motivos de caracterización para poder relacionar los resultados, se decidió obtener la rigidez para cada muelle aplicando la ley de Hooke como se Ec [1]. Los resultados se visualizan en la tabla 1.

Muelle	M (kg)	ΔL (m)	k (N/m)
1	4,50	0,20	226,38
2	4,50	0,09	485,11
3	4,50	0,13	334,43

Tabla 1: Caracterización de la rigidez de los muelles.

La Fig. 10 representa los desplazamientos experimentados por el muelle 1 en un tiempo de 10 segundos al aplicar distintas cargas. Se puede observar que el periodo de la onda a medida que se incrementa la carga sobre el muelle disminuye notablemente para cada caso.

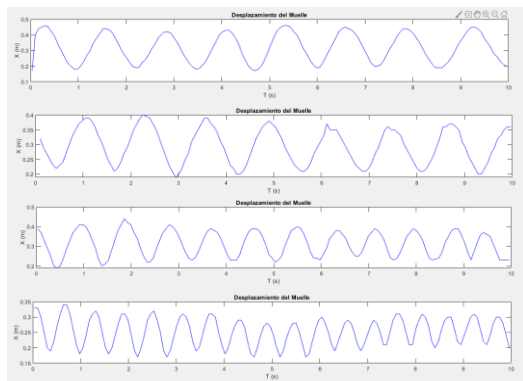


Fig. 10: Variación del periodo experimentado por el muelle 1 al aplicar distintas cargas.

A continuación, la tabla 2 indica un resumen de los resultados obtenidos al aplicar a cada tipo de muelle distintas cargas.

Muelle	Caso	k (N/m)	M (kg)	T (s)
1	A1	226,38	1,50	0,67
	B1	226,38	3,00	0,88
	C1	226,38	4,50	1,09
	D1	226,38	6,00	1,29
2	A2	485,11	1,50	0,54
	B2	485,11	3,00	0,68
	C2	485,11	4,50	0,83
	D2	485,11	6,00	0,97
3	A3	334,43	1,50	0,60
	B3	334,43	3,00	0,78
	C3	334,43	4,50	0,94
	D3	334,43	6,00	1,10

Tabla 2: Resumen del periodo para cada tipo de muelle al variar la masa de aplicación.

En la Fig. 11 representa la relación que experimenta periodo-rigidez y de igual manera la Fig. 12 se observa la relación periodo-masa para cada uno de los muelles experimentados.

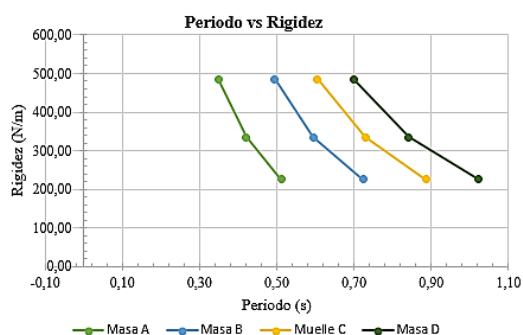


Fig. 11: Relación Periodo-rigidez

Como se puede observar esta relación es directamente proporcional, a medida que los muelles son más rígidos y se aplica cargas

ligeras, el periodo es muy largo. De forma opuesta se comporta los muelles de poca rigidez expuestos a cargas altas, con lo cual se verificó que este sistema cumple con la Ec [4].

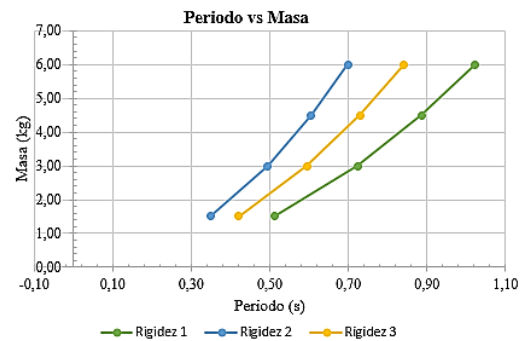


Fig. 12: Relación Periodo-Masa.

7. CONCLUSIONES

- Con la ayuda de elementos electrónicos y un software de procesamiento se desarrolló un dispositivo que permita determinar el periodo de un muelle. Lo que nos indica que, identificando las variables adecuadas, las técnicas de análisis y modelización es posible crear diversos dispositivos que nos ayuden en la medición de magnitudes.
- El periodo de un muelle está estrechamente vinculado con la masa y rigidez. A medida que los muelles sean más rígidos el periodo de oscilación será más pequeño. Por lo contrario, si el muelle presenta poca rigidez su periodo de oscilación será muy grande.
- Para controlar los errores en la toma de datos se debe tener muy en cuenta la resolución de los aparatos y sus limitaciones; no se debe abusar de sus características. De igual manera se logra mejorar la calidad de los resultados, trabajando en la sensibilidad que se desea dar al instrumento de medida. La sensibilidad y control de errores ayudan de forma directa a determinar la calidad de la medición.
- Se debe tener cuidado al experimentar con muelles de poca rigidez expuestos a altas cargas, debido a que su periodo de desplazamientos es muy corto

ocasionando en muchas ocasiones lecturas erróneas.

8. REFERENCIAS

- [1] A. Arrieta, E. Arrieta, and J. Tejeiros, "Masa Efectiva para un Sistema de Muelle Real," *Rev. Colomb. Física*, vol. 41, no. 2, pp. 2–4, 2009.
- [2] Landau & Lifshitz, *Mecánica*, Reverté. Barcelona, 1991.
- [3] J. B. Marion, *Dinámica clásica de las partículas y sistemas*. Barcelona, 1996.
- [4] "www.fritzing.org." .
- [5] J. C. Herrero Herranz and J. Sánchez Allende, "Una mirada al mundo Arduino," *Rev. Tecnol. y Desarro.*, vol. 13, p. 28, 2015.
- [6] "www.arduino.cc." .
- [7] "https://es.mathworks.com/." .

Anexo 1 (Código de programación)

<pre> %% PROGRAMA PARA OBTENER EL PERIODO DE UN MUELLE %% Vinculación de Arduino a Matlab delete(instrfind({'Port'}, {'COM1'})); s = serial('COM1','BaudRate',9600,'Terminator','CR/LF'); warning('off','MATLAB:serial:fscanf:unsuccessfulRead'); fopen(s); %% Condiciones iniciales de lectura tmax = 10; rate = 57.8; %% Figura: desplazamiento del muelle f = figure('Name','Captura'); a = axes('XLim',[0 tmax],'YLim',[0 6]); al = line(nan,nan,'Color','r','LineWidth',2); xlabel('Tiempo (s)') ylabel('Distancia (m)') title('Distancia en tiempo real con Arduino') grid on hold on % Acumulador v1 = zeros(1,tmax*rate); tiempo = zeros(1,tmax*rate); i = 1; t = 0; % ejecutar bucle cronometrado tic while t<tmax t = toc; tiempo(i)=t; % leer del puerto serie a = fscanf(s,'%d %d'); v1(i)=a(1)/100; % dibujar en la figura x = linspace(0,i/rate,i); set(al,'YData',v1(1:i),'XData',x); drawnow % seguir sumatoria=1; if i > 101 sumatoria=0; tamano=i-10; for j=i-1:tamano sumatoria=sumatoria+abs(v1(j)); end end if sumatoria < 0.2 t=tmax; end i = i+1; end % ejecutar bucle cronometrado tic while t<tmax t = toc; tiempo(i)=t; % leer del puerto serie a = fscanf(s,'%d %d'); v1(i)=a(1)/100; % dibujar en la figura x = linspace(0,i/rate,i); set(al,'YData',v1(1:i),'XData',x); drawnow % seguir sumatoria=1; if i > 101 sumatoria=0; tamano=i-10; for j=i-1:tamano sumatoria=sumatoria+abs(v1(j)); end end if sumatoria < 0.2 t=tmax; end i = i+1; end %% Obtención del periodo Dx=v1'; Tx=x'; B1=[]; C1=[]; D1=[]; E1=[]; Al=length (Tx); for i=1:Al if Tx(i)<=10 && Dx(i)<=0.4 B1(i)=Tx(i); D1(i)=Dx(i); C1=[C1;B1(i)]; E1=[E1;D1(i)]; end end C1; E1; F1=max(E1,[],'all'); G1=min(E1,[],'all'); H1=(E1-G1)*0.5; I1=H1-C1; J1=E1-I1; Dp=J1; A=Dp; </pre>	
	<pre> % Limpieza de datos previos % Crear objeto serie % Advertencia de tipo de datos % Apertura del puerto % Tiempo de captura en s % Rapidez de lecturas % Nombre de la figura % Delimitación de ejes x-y % Tener cuidado con el acumulador % Vector de cargas % Tamaño de la matriz tiempo % Límite rango de análisis [tiempo y desplazamiento], control de distorsión % Matriz de tiempo sin error % Matriz de desplazamiento sin error % Máximo desplazamiento % Mínimo desplazamiento % Altura a desplazar para cambia el ejes en gráfica % Matriz de desplazamientos cambiado su eje </pre>


```

C=length (B);

%Gráfica del Movimiento del muelle
%figure %Grafica sin mover el eje y
%hold on
%grid
%plot (C1,E1)
%title('Movimiento del Muelle')
%xlabel('T (s)')
%ylabel('X (m)')

E=[]; F=[]; G=[]; H=[]; K=[]; L=[]; M=[]; N=[]; O=[];
j=D(4)-D(3); %Determina el periodo de divisiones
for i=1:C
    if A(i)>=0
        E(i)=D(i);
        G=[G;E(i)];
    elseif A(i)<0
        F(i)=D(i);
        H=[H;F(i)];
    end
end
G;
Q=length (G);
H;
P=length (H);
for i=1:Q-1
    K(i)=G(i+1)-G(i);
    if K(i)>j
        L=G(i);
        O=[O;L];
    end
end
O
for i=1:P-1
    M(i)=H(i+1)-H(i);
    if M(i)>j
        N=H(i);
        N=[N;M];
    end
end
N
T=O(6)-O(5)
Ti=N(3)-N(2)

figure (2) %Grafica movido el eje y
plot (C1,J1,'Color','r','LineWidth',2)
title (['Periodo del Muelle T= ' num2str(T) 's'])
xlabel('Tiempo (s)')
ylabel('Distancia (m)')
hold on
grid on
%%Incrementto del programama
TT=T/(2*pi());

% Velocidad angular
VM=A/TT;

%Aceleración angular
AM=A/(TT^2);

figure (3)
hold on
grid
subplot (3,1,1)
plot(C1,E1,'-b')
title('Desplazamiento del Muelle')
xlabel('T (s)')
ylabel('X (m)')
subplot (3,1,2)
plot(C1,VM,'-r')
title('Velocidad del Muelle')
xlabel('T (s)')
ylabel('V (m/s)')
subplot (3,1,3)
plot(C1,AM,'-c')
title('Aceleración del Muelle')
xlabel('T (s)')
ylabel('a (m2/s^2)')

```