

Análisis del Movimiento Armónico Simple en un Péndulo

Repositorio: <https://github.com/AndresGUIO2/SHMAnalysis.git>

Héctor Andrés Güiza Ortegon hector.guiza@udea.edu.co
Daniel Cano Hernández daniel.canoh@udea.edu.co
Natalia Andrea García Ríos natalia.garcia9@udea.edu.co

Resumen— Este trabajo presenta el análisis del Movimiento Armónico Simple (MAS) aplicado a un sistema físico real: un péndulo simple. A través del uso de técnicas de procesamiento digital de imágenes, se capturó, segmentó y analizó cuadro a cuadro el movimiento del péndulo mediante un video experimental. Empleando transformaciones de color, operaciones morfológicas y seguimiento del centroide, se reconstruyó su trayectoria, lo que permitió calcular parámetros dinámicos como el ángulo, la velocidad y la aceleración angular. Estos datos fueron comparados con los resultados teóricos del MAS, mostrando una alta concordancia. El enfoque utilizado evidencia la capacidad del procesamiento de imágenes como herramienta precisa y didáctica para el estudio de sistemas físicos oscilatorios.

Palabras claves— *Movimiento Armónico Simple, procesamiento digital de imágenes, péndulo simple, análisis de video, segmentación en HSV, visión por computador, velocidad angular, aceleración angular, comparación experimental-teórica.*

I. INTRODUCCIÓN

El estudio del Movimiento Armónico Simple (MAS) consiste en comprender fenómenos físicos que involucran oscilaciones periódicas, como es el caso de un péndulo simple. En este laboratorio, el MAS se aborda desde una perspectiva computacional basada en el procesamiento digital de imágenes, lo que permite obtener parámetros físicos relevantes a partir del análisis cuadro a cuadro de un video real del sistema en movimiento.

Mediante algoritmos de detección y seguimiento, se identifica la posición del péndulo en cada fotograma, lo que posibilita el cálculo de su velocidad y aceleración angular. Estos resultados experimentales se contrastan con los valores teóricos derivados del modelo matemático del MAS, con el fin de evaluar la validez de este bajo condiciones controladas. El procesamiento incluye técnicas de filtrado, segmentación por umbral, transformaciones de espacio de color y operaciones morfológicas, las cuales mejoran la precisión en la detección del objeto de interés.

II. OBJETIVOS

A. Objetivo General

Aplicar técnicas fundamentales de procesamiento digital de imágenes para analizar el comportamiento de un péndulo en movimiento, y reforzar conceptos teóricos a través de su contraste con un sistema físico real.

B. Objetivos Específicos

- Desarrollar habilidades en el preprocesamiento de imágenes mediante la conversión de color, reducción de ruido y operaciones morfológicas.
- Implementar métodos de segmentación y detección de contornos para identificar el objeto de interés en un video.
- Extraer información temporal del sistema físico a partir de secuencias de imágenes, incluyendo la posición cuadro a cuadro del péndulo.
- Calcular parámetros dinámicos como velocidad y aceleración angular, a partir de los datos obtenidos por visión por computador.
- Validar experimentalmente el modelo del Movimiento Armónico Simple, comparando las mediciones obtenidas con las fórmulas analíticas.

III. METODOLOGÍA

El desarrollo del laboratorio se llevó a cabo siguiendo una serie de pasos que permitieron aplicar técnicas de procesamiento digital de imágenes para el análisis del movimiento de un péndulo simple.

A. Entorno de Desarrollo

El análisis se implementó en Python 3.11 utilizando un cuaderno de Jupyter como entorno de trabajo. Se creó un entorno virtual mediante Conda para garantizar la reproducibilidad. Las bibliotecas principales utilizadas fueron:

- 1) *OpenCV (cv2)*: lectura del video, conversión de espacios de color, operaciones morfológicas y detección de contornos.

- 2) *NumPy*: manejo de arreglos numéricos y cálculo de derivadas.
- 3) *Matplotlib*: generación de gráficos.
- 4) *TQDM*: visualización del progreso durante el procesamiento.
- 5) *IPython.display*: incrustación del video en el notebook.
- 6) *Seaborn y Pandas*: análisis y visualización de datos.
- 7) *SciPy*: suavizado de señales con filtro de Savitzky-Golay.

```
import cv2
import numpy as np
from tqdm.notebook import tqdm
from IPython.display import Video, clear_output, display
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from scipy.signal import savgol_filter
```

B. Captura del Sistema Físico

Se grabó un video de 30 segundos de un péndulo oscilando libremente. El video se almacenó en formato .mp4 y se visualizó con:

```
Video("pendulito.mp4", embed=True, width=640, height=360)
```



Fig. 1. captura del sistema físico. Se observa el péndulo oscilando con una esfera azul como masa, suspendida por un hilo delgado frente a un fondo claro. Esta configuración fue diseñada para facilitar la segmentación por color en el análisis posterior mediante visión por computador.

C. Preprocesamiento y Segmentación

Cada fotograma fue convertido a HSV y se aplicó apertura morfológica para reducir ruido. Luego se segmentó el color azul en dos rangos (brillante y oscuro) y se aplicaron operaciones de dilatación y erosión:

```
def to_hsv(frame):
    return cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

def reduce_noise(channel1):
    kernel = np.ones((3, 3), np.uint8)
    return cv2.morphologyEx(channel1, cv2.MORPH_OPEN, kernel)

def threshold_hsv(hsv_frame):
    lower_bright = np.array([100, 150, 100])
    upper_bright = np.array([130, 255, 255])
    lower_dark = np.array([100, 100, 30])
    upper_dark = np.array([130, 255, 90])

    mask_bright = cv2.inRange(hsv_frame, lower_bright, upper_bright)
    mask_dark = cv2.inRange(hsv_frame, lower_dark, upper_dark)
    combined_mask = cv2.bitwise_or(mask_bright, mask_dark)

    kernel = np.ones((3, 3), np.uint8)
    mask_dilated = cv2.dilate(combined_mask, kernel, iterations=20)
    mask_eroded = cv2.erode(mask_dilated, kernel, iterations=15)
```

```
return mask_eroded
```

D. Detección del Péndulo y Cálculo del Centroide

Se identificó el contorno más grande y se calculó su centroide mediante momentos:

```
def find_largest_contour(binary_frame, min_area=100):
    contours, _ = cv2.findContours(binary_frame, cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    filtered = [cnt for cnt in contours if cv2.contourArea(cnt) >
        min_area]
    return max(filtered, key=cv2.contourArea) if filtered else None

def compute_centroid(contour):
    M = cv2.moments(contour)
    if M["m00"] != 0:
        return int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"])
    return None

def draw_centroid(frame, centroid):
    if centroid:
        cv2.circle(frame, centroid, 5, (0, 0, 255), -1)
    return frame
```

Para validar visualmente:

```
def draw_centroid(frame, centroid):
    if centroid:
        cv2.circle(frame, centroid, 5, (0, 0, 255), -1)
    return frame

frame_id = 358
cap = cv2.VideoCapture("pendulito.mp4")
cap.set(cv2.CAP_PROP_POS_FRAMES, frame_id)
ret, frame = cap.read()
cap.release()

if not ret:
    print("No se pudo leer el frame.")
else:
    hsv = to_hsv(frame)
    cleaned = reduce_noise(hsv[:, :, 2])
    binary = threshold_hsv(hsv)
    kernel = np.ones((3, 3), np.uint8)
    binary = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel,
        iterations=2)
    contour = find_largest_contour(binary, min_area=50)
    centroid = compute_centroid(contour) if contour is not None else None
    debug_frame = cv2.cvtColor(cleaned, cv2.COLOR_GRAY2BGR)
    if contour is not None:
        cv2.drawContours(debug_frame, [contour], -1, (0, 255, 0), 2)
    debug_frame = draw_centroid(debug_frame, centroid)
    fig, axs = plt.subplots(1, 3, figsize=(15, 5))
    axs[0].imshow(hsv[:, :, 2], cmap='gray'); axs[0].set_title('Canal V (Valor)')
    axs[1].imshow(binary, cmap='gray'); axs[1].set_title('Threshold')
    axs[2].imshow(cv2.cvtColor(debug_frame, cv2.COLOR_BGR2RGB));
    axs[2].set_title(f'Contorno + Centroide: {centroid}')
    for ax in axs: ax.axis('off')
    plt.show()
```

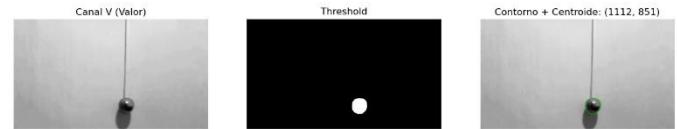


Fig. 2. resultado del preprocesamiento y segmentación. Se muestra el canal de valor (V) extraído del espacio HSV, la máscara binaria generada mediante umbralización, y la detección final del contorno junto al centroide del péndulo. Esta secuencia permitió validar visualmente la eficacia del procesamiento aplicado para el seguimiento del objeto.

E. Análisis del Movimiento

Se estimó la escala espacial y se extrajo la trayectoria del centroide:

```
scale = 0.37 / centroid[1]
pendulum_length = 0.52
centroids = []
cap = cv2.VideoCapture("pendulito.mp4")
total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
for frame_id in tqdm(range(total_frames)):
```

```

ret, frame = cap.read()
if not ret:
    centroids.append(None)
    continue
hsv = to_hsv(frame)
binary = threshold_hsv(hsv)
contour = find_largest_contour(binary, min_area=100)
centroid = compute_centroid(contour) if contour is not None else None
centroids.append(centroid)
cap.release()

```

Datos convertidos a DataFrame y graficados:

```

x_vals = [c[0] if c else None for c in centroids]
y_vals = [c[1] if c else None for c in centroids]
t_vals = list(range(len(centroids)))
df = pd.DataFrame({ 'Frame': t_vals, 'x(t)': x_vals, 'y(t)': y_vals })

# Visualización del movimiento del centroide
theme = sns.set_theme(style="whitegrid")
plt.figure(figsize=(10, 4))
sns.lineplot(data=df[['Frame', 'x(t)']].dropna(), x='Frame', y='x(t)', label='x(t)')
sns.lineplot(data=df[['Frame', 'y(t)']].dropna(), x='Frame', y='y(t)', label='y(t)')
plt.xlabel('Frame')
plt.ylabel('Coordenada')
plt.title('Movimiento del centroide del péndulo')
plt.legend()
plt.tight_layout()
plt.show()

```

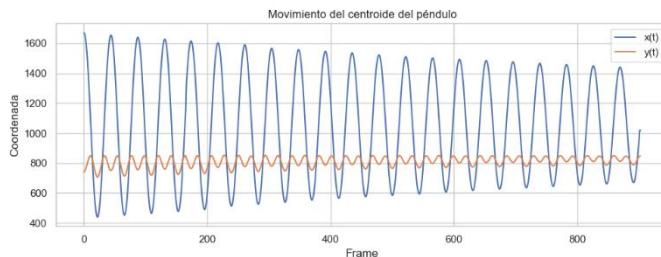


Fig. 3. movimiento del centroide del péndulo. Se observa una oscilación periódica bien definida en la componente horizontal $x(t)$, mientras que $y(t)$ permanece casi constante. Esta representación confirma que el péndulo se mueve principalmente en un plano, como es característico en sistemas que siguen el Movimiento Armónico Simple.

F. Cálculo de Ángulo, Velocidad y Aceleración Angular

Con la secuencia completa de centroides, se reconstruyó la trayectoria del péndulo en unidades físicas (metros), usando la escala obtenida previamente. A partir de esto se calcularon el ángulo angular θ , su derivada (velocidad angular ω) y su segunda derivada (aceleración angular α), comparando con los valores teóricos esperados del Movimiento Armónico Simple.

Primero, se obtuvo el intervalo de muestreo temporal dt a partir del fps del video:

```

g = 9.81 # aceleración de la gravedad
cap = cv2.VideoCapture("pendulito.mp4")
fps = cap.get(cv2.CAP_PROP_FPS)
cap.release()

if fps is None or fps <= 0:
    print("⚠️ No se pudo leer el FPS; usando valor por defecto de 30")
    fps = 30.0

dt = 1.0 / fps
print(f"Usando dt = {dt:.4f} s por frame")

>> Usando dt = 0.0333 s por frame

```

Se convirtieron las coordenadas de píxeles a metros y se calcularon los desplazamientos relativos al punto de suspensión estimado:

```

df['x (m)'] = df['x(t)'] * scale
df['y (m)'] = df['y(t)'] * scale

# Punto de suspensión del péndulo estimado (en pixeles)
pivot_px = (1112, -345.5)

# Desplazamientos relativos en metros

df['dx (m)'] = (df['x(t)'] - pivot_px[0]) * scale
df['dy (m)'] = (df['y(t)'] - pivot_px[1]) * scale

```

El ángulo de oscilación θ se calculó como:

$$\theta(t) = \arctan\left(\frac{dx(t)}{dy(t)}\right) \quad (1)$$

```
# Ángulo real θ en radianes
df['theta (rad)'] = np.arctan2(df['dx (m)'], df['dy (m)'])
```

1) Suavizado y Derivación Experimental: Se aplicó interpolación y un filtro de Savitzky-Golay para suavizar la señal angular antes de calcular sus derivadas:

```
# Interpolación de valores perdidos y filtrado de outliers
```

```
mask = df['theta (rad)'].diff().abs() > 0.3
df.loc[mask, 'theta (rad)'] = np.nan
df['theta (rad)'] = df['theta (rad)'].interpolate(limit=2)
```

```
# Suavizado y derivación
```

```
df['theta_smooth'] = savgol_filter(df['theta (rad)'], window_length=11,
polyorder=3, mode='interp')
df['omega (rad/s)'] = np.gradient(df['theta_smooth'], dt)
df['alpha experimental (rad/s²)'] = np.gradient(df['omega (rad/s)'], dt)
```

2) Comparación con el Modelo Teórico: Para comparar con un péndulo ideal, se calculó la aceleración angular teórica a partir del modelo del MAS:

$$\theta_{teórica}(t) = \theta_0 \cdot \cos(\omega_0 t) \quad (2)$$

$$\alpha_{teórica}(t) = -\omega_0^2 \cdot \theta_{teórica}(t) \quad (3)$$

```
omega0 = np.sqrt(g / pendulum_lenght)
time = df['Frame'] * dt
theta0 = df['theta (rad)'].abs().max()
df['theta teor pura'] = theta0 * np.cos(omega0 * time)
df['alpha teor pura'] = - omega0**2 * df['theta teor pura']
```

3) Visualización de Resultados: Se graficaron θ , ω y α , tanto experimental como teórica.

```

ang_df = df.copy()
ang_df['Time (s)'] = ang_df['Frame'] * dt

# Desplazamiento angular

plt.figure(figsize=(10, 4))
sns.lineplot(data=ang_df, x='Time (s)', y='theta (rad)')
plt.title('Desplazamiento Angular θ')
plt.xlabel('Tiempo (s)')
plt.ylabel('θ (rad)')
plt.tight_layout()
plt.show()

# Velocidad angular

plt.figure(figsize=(10, 4))
sns.lineplot(data=ang_df, x='Time (s)', y='omega (rad/s)')
plt.title('Velocidad Angular ω')
plt.xlabel('Tiempo (s)')
plt.ylabel('ω (rad/s)')
plt.tight_layout()

```

```

plt.show()

# Aceleración angular experimental vs. Teórica
plt.figure(figsize=(10, 4))
sns.lineplot(data=ang_df, x='Time (s)', y='alpha experimental (rad/s²)', label='experimental')
sns.lineplot(data=ang_df, x='Time (s)', y='alpha teor pura', label='teórica')
plt.title('Aceleración Angular α')
plt.xlabel('Tiempo (s)')
plt.ylabel('α (rad/s²)')
plt.legend()
plt.tight_layout()
plt.show()

```

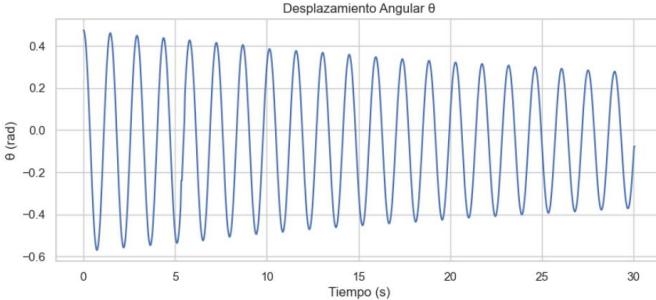


Fig. 4. desplazamiento angular del péndulo en función del tiempo. La señal presenta un comportamiento oscilatorio amortiguado, característico del Movimiento Armónico Simple en presencia de fuerzas disipativas como la fricción o la resistencia del aire.

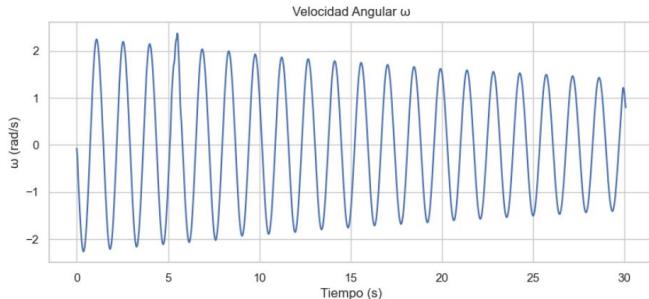


Fig. 5. velocidad angular del péndulo en función del tiempo. La gráfica muestra una oscilación sinusoidal atenuada, propia del Movimiento Armónico Simple amortiguado. La disminución progresiva en la amplitud de refleja la acción de fuerzas disipativas como la fricción del aire o las imperfecciones del sistema.

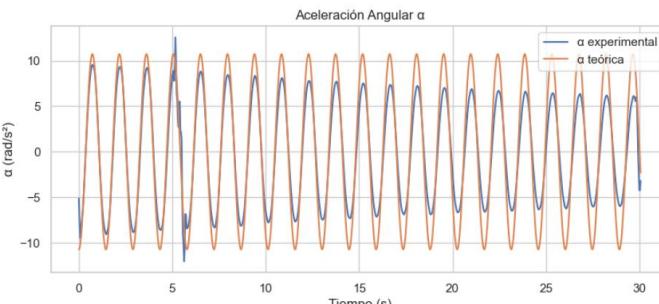


Fig. 6. comparación entre la aceleración angular experimental y la teórica del péndulo. Ambas curvas presentan un patrón oscilatorio en fase y con igual frecuencia. Las diferencias en amplitud evidencian efectos reales como pérdidas energéticas, ruido en la detección o simplificaciones del modelo teórico sin disipación.

Estas gráficas permitieron comparar el comportamiento angular del sistema real frente al ideal, destacando la validez del modelo MAS y las desviaciones atribuibles a condiciones experimentales reales.

IV. ANÁLISIS DE RESULTADOS

Durante el análisis se generaron diversas visualizaciones que permitieron observar el proceso completo de procesamiento digital de imágenes, desde la segmentación del objeto hasta la evaluación cuantitativa de su comportamiento dinámico. A continuación, se describen los principales hallazgos.

A. Detección y Segmentación Exitosa

En la Figura 2 se presenta el resultado de la segmentación del péndulo en un fotograma representativo. El canal de valor (V) del espacio HSV permitió resaltar la intensidad del objeto, facilitando su separación del fondo. Al aplicar umbralización y operaciones morfológicas (apertura, dilatación, erosión y cierre), se obtuvo una máscara binaria limpia del objeto. A partir de esta se identificó el contorno principal y se calculó su centroide, lo cual fue clave para el análisis temporal del movimiento.

B. Movimiento del Centroide

La Figura 3 muestra la evolución de las coordenadas del centroide a lo largo del tiempo. La componente horizontal $x(t)$ evidencia un patrón oscilatorio claro, como se espera en un péndulo que se desplaza principalmente sobre un plano. Por su parte, la coordenada vertical $y(t)$ permanece casi constante, confirmando que el movimiento es mayormente angular.

C. Análisis Angular

En la Figura 4 se visualiza el desplazamiento angular $\theta(t)$. Su comportamiento periódico confirma la naturaleza oscilatoria del sistema. La Figura 5 complementa este análisis mostrando la evolución de la velocidad angular $\omega(t)$, la cual también sigue un patrón sinusoidal, con amortiguamiento progresivo debido a la fricción o pérdida de energía por resistencia del aire.

D. Comparación entre Aceleración Angular Experimental y Teórica

La Figura 6 compara la aceleración angular obtenida experimentalmente $\alpha_{exp}(t)$ con la aceleración teórica $\alpha_{teo}(t)$ calculada bajo el modelo de Movimiento Armónico Simple (MAS). Las dos curvas muestran un alto grado de concordancia en frecuencia y forma. Las diferencias observadas en amplitud pueden atribuirse al ruido en la detección, errores de segmentación y simplificaciones del modelo, como la ausencia de términos disipativos.

E. Discusión General

El sistema analizado se comporta, en términos generales, como un péndulo ideal bajo régimen de MAS. La relación entre aceleración y desplazamiento angulares sigue aproximadamente la ley $\alpha = -\omega_0^2 \cdot \theta$, lo que valida la precisión del modelo teórico. Las discrepancias menores detectadas reflejan los efectos reales del entorno experimental: fricción, imperfecciones ópticas, resolución limitada y ruido digital.

En conjunto, los resultados confirman que el uso de procesamiento digital de imágenes es una herramienta efectiva y precisa para el análisis dinámico de sistemas físicos oscilatorios.

V. CONCLUSIÓN

El presente laboratorio permitió integrar conceptos teóricos del Movimiento Armónico Simple (MAS) con herramientas prácticas de procesamiento digital de imágenes para el análisis del comportamiento dinámico de un péndulo real.

Mediante el uso de técnicas de segmentación en el espacio de color HSV, operaciones morfológicas y detección de contornos, fue posible identificar de manera precisa el péndulo en cada fotograma del video. A partir del seguimiento del centroide, se reconstruyó la trayectoria del sistema y se calculó el ángulo de oscilación. Posteriormente, se derivaron numéricamente la velocidad y la aceleración angular.

Los resultados obtenidos mostraron una evolución oscilatoria coherente con el modelo teórico del MAS. La comparación entre la aceleración angular teórica y la experimental evidenció una fuerte correspondencia en términos de frecuencia y forma, con discrepancias leves atribuibles a factores como la fricción, errores acumulados en el procesamiento de imágenes o simplificaciones propias del modelo ideal.

En conclusión, el enfoque computacional implementado no solo permitió validar experimentalmente el comportamiento armónico del sistema, sino que también demostró el potencial del procesamiento digital de imágenes como herramienta para el estudio cuantitativo de fenómenos físicos reales.

REFERENCIAS

- [1] D. C. Hernández, "Movimiento armónico simple y péndulo simple," *SlideShare*, 2013. [Online]. Disponible en: <https://es.slideshare.net/dianacarolinavelagarcia/movimiento-armnico-simple-y-pendulo-simple>.
- [2] "Movimiento armónico simple: péndulo simple," *Teachy*, [Online]. Disponible en: <https://www.teachy.app/es/resumenes/educacion-media/media-superior-3-grado/fisica-a-espanol/resumen-de-movimiento-armonico-simple-pendulo-simple>.
- [3] "Técnicas de procesamiento digital de imágenes," *Revista de Marina*, [Online]. Disponible en: <https://revistamarina.cl/articulo/tecnicas-de-procesamiento-digital-de-imagenes>.
- [4] "Análisis MAS – código para seguimiento de péndulo," *MyCompiler.io*, 2024. [Online]. Disponible en: <https://www.mycompiler.io/view/JK5ODvKQGWF>.