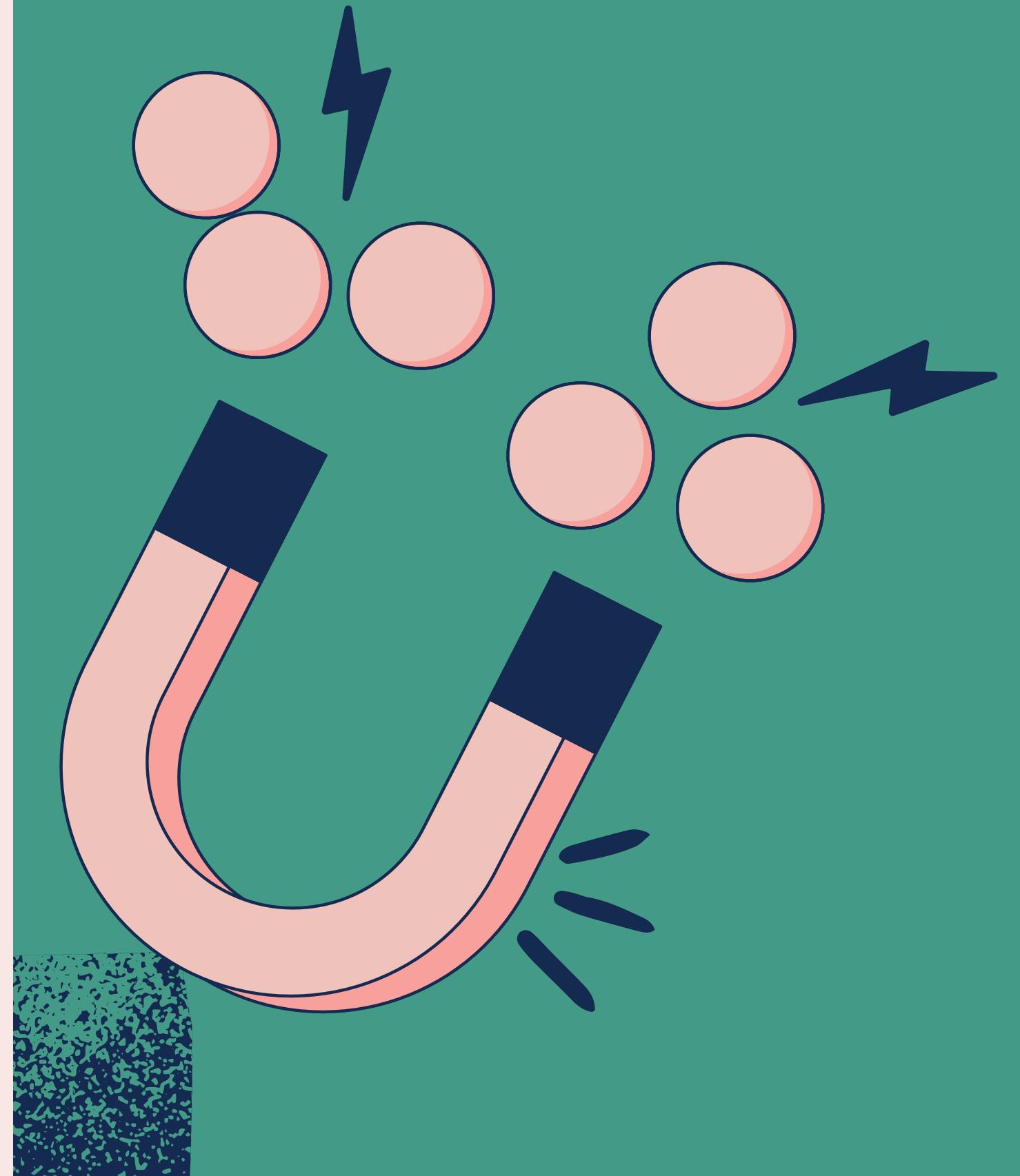


Reto: Frenado Magnético

Aldo Emiliano Galván Gómez A07106914
Edgar Arechiga Nuño A01647054
Leon Daniel Vilchis Arceo A01641413
Jorge Cárdenas Blanco A01647246



Frenos electromagnéticos

Un freno magnético es un dispositivo que usa fuerzas magnéticas para frenar o detener el movimiento de un componente mecánico sin contacto físico. Es común en maquinaria, trenes, ascensores y sistemas robóticos por su precisión, confiabilidad y bajo desgaste.



Freno de corrientes de Foucault

El freno lineal de corriente de Foucault es un sistema electromagnético usado en trenes de alta velocidad y montañas rusas. Funciona mediante la inducción de corrientes de Foucault.

Aplicaciones del Frenado magnético



Este tipo de frenado es valorado por su bajo mantenimiento, alta durabilidad y capacidad para funcionar en ambientes donde el contacto mecánico podría ser problemático.

- Maquinaria industrial: Para controlar y detener movimientos en equipos con alta precisión y sin desgaste mecánico.
- Trenes de alta velocidad: Uso de frenos de corrientes de Foucault para garantizar un frenado suave y seguro en sistemas ferroviarios modernos.
- Montañas rusas y torres de caída: Frenado lineal electromagnético que desacelera góndolas o carros de forma controlada, mejorando la seguridad y confort de los usuarios.
- Ascensores y sistemas robóticos: Para un control eficiente y sin contacto del movimiento, aumentando la vida útil del sistema.



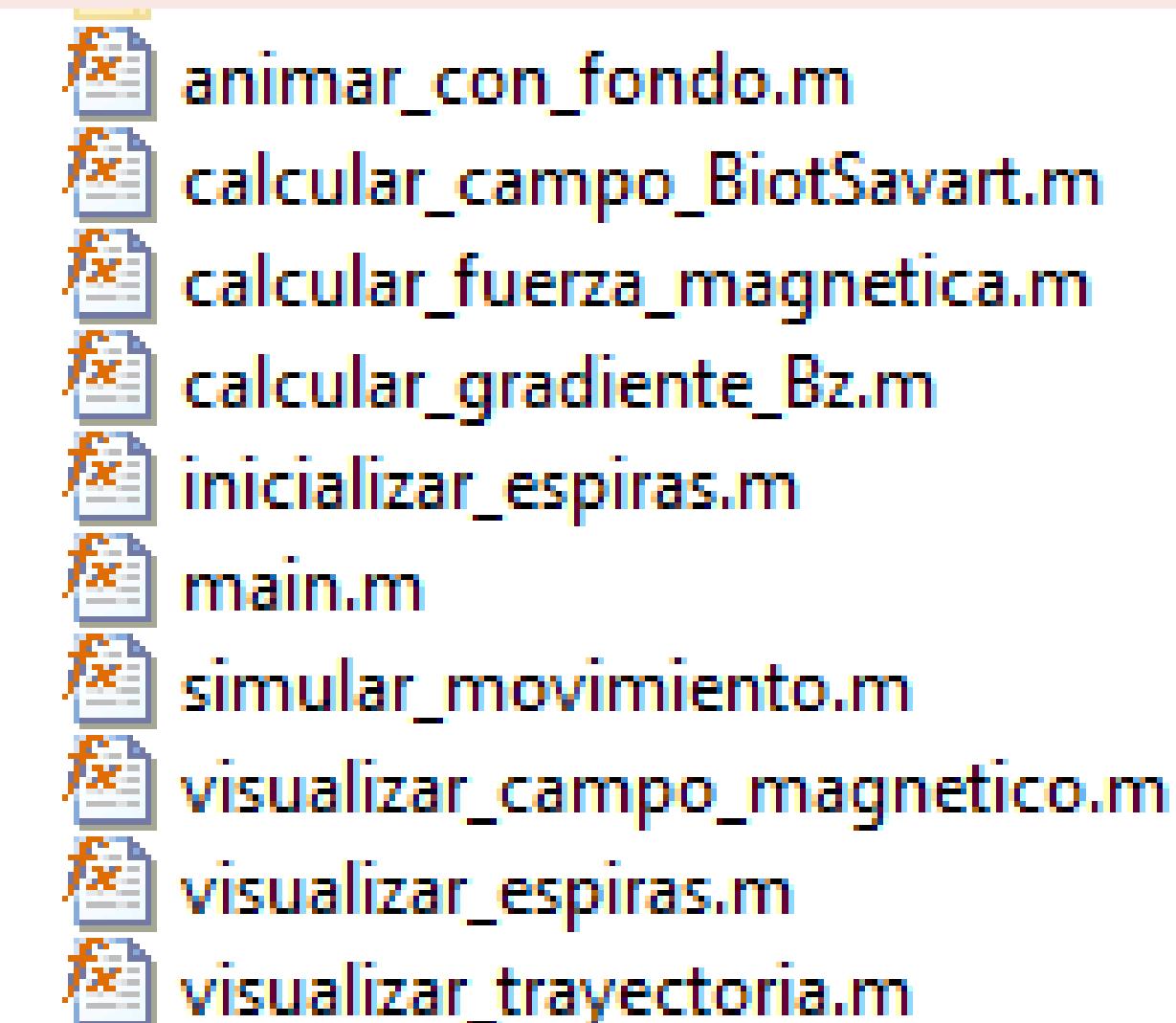
Reto: Frenado Magnético



Modelar y simular computacionalmente el frenado magnético en una torre de caída, analizando cómo las fuerzas magnéticas desaceleran una góndola en movimiento. El objetivo es comprender y predecir, a través de un modelo físico y su implementación en MATLAB, la evolución de la velocidad y aceleración bajo distintos parámetros de diseño, asegurando la seguridad y el confort de los usuarios.

Simulación en MATLAB

La simulación se estructura en MATLAB utilizando un enfoque modular. Cada función corresponde a una etapa física o computacional del modelo: desde la inicialización de espiras conductoras, pasando por el cálculo del campo magnético y el gradiente, hasta la simulación dinámica y visualización de resultados. El archivo principal, main.m, organiza la ejecución secuencial de cada bloque funcional, facilitando el análisis y la validación del sistema.



inicializar_espiras.m

```
1 function [xg, yg, zg, const_BS, x, y, z, dx, dy, dz] = inicializar_espiras()
2     num_espiras = 5;
3     pts_espira = 80;
4     radio = 1.5;
5     altura_total = 6;
6     corriente = 1000;
7     mu_0 = 4*pi*1e-7;
8     const_BS = mu_0 * corriente / (4*pi);
9
10    espacioAmpleado = 2.5;
11    lx = 50; ly = 50; lz = 60;
12    xg = linspace(-espacioAmpleado*radio, espacioAmpleado*radio, lx);
13    yg = linspace(-espacioAmpleado*radio, espacioAmpleado*radio, ly);
14    zg = linspace(-0.5*altura_total, 1.5*altura_total, lz);
15
16    x = []; y = []; z = [];
17    dx = []; dy = []; dz = [];
18    altura_espira = altura_total / (num_espiras - 1);
19
20    for n = 0:num_espiras-1
21        t = linspace(0, 2*pi, pts_espira);
22        x_espira = radio * cos(t);
23        y_espira = radio * sin(t);
24        z_espira = ones(1, pts_espira) * (n * altura_espira);
25
26        x = [x, x_espira];
27        y = [y, y_espira];
28        z = [z, z_espira];
29
30        dt = mean(diff(t));
31        dx = [dx, -radio * sin(t) * dt];
32        dy = [dy, radio * cos(t) * dt];
33        dz = [dz, zeros(1, pts_espira)];
```

La función `inicializar_espiras.m` configura la base física del modelo generando cinco espiras circulares apiladas verticalmente, simulando la región conductora responsable del frenado magnético.

En el código se definen:

- El número de espiras, el radio y la altura total del sistema.
- Las posiciones espaciales de cada punto sobre las espiras, parametrizadas mediante ángulos.
- Vectores de posición y dirección de corriente para cada segmento de espira.
- Una malla tridimensional que permitirá evaluar el campo magnético inducido.

calcular_campo_BiotSavart.m

La función `calcular_campo_BiotSavart.m` implementa la ley de Biot-Savart para obtener el campo magnético generado por las espiras en toda la malla tridimensional del sistema.

En el código:

- Se recorre cada punto de la malla y cada segmento de espira.
- Se calcula el vector distancia y su módulo respecto a cada punto.
- Se evalúa la contribución diferencial de cada elemento de corriente a las componentes B_x , B_y , B_z
- Se suman todas las contribuciones para obtener el campo total en cada punto.

Esto es fundamental para determinar las regiones de mayor influencia del frenado magnético y para calcular la fuerza que actúa sobre la góndola en movimiento.

```
function [Bx, By, Bz] = calcular_campo_BiotSavart(xg, yg, zg, x, y, z, dx, dy, dz, const_BS)
    % Initialize arrays for the magnetic field components
    lx = length(xg);
    ly = length(yg);
    lz = length(zg);
    Bx = zeros(lx, ly, lz);
    By = zeros(lx, ly, lz);
    Bz = zeros(lx, ly, lz);

    % Loop over the grid points
    for i = 1:lx
        for j = 1:ly
            for k = 1:lz
                % Calculate the vector distance and its norm
                rx = xg(i) - x(j);
                ry = yg(j) - y(k);
                rz = zg(k) - z(i);
                r_norm = sqrt(rx^2 + ry^2 + rz^2);
                if r_norm == 0, r_norm = 1e-6; end
                r3 = r_norm^3;

                % Calculate differential contributions
                dBx = const_BS * (dy(j)*rz - dz(j)*ry) / r3;
                dBy = const_BS * (dz(j)*rx - dx(j)*rz) / r3;
                dBz = const_BS * (dx(j)*ry - dy(j)*rx) / r3;

                % Sum contributions to total field
                Bx(i,j,k) = Bx(i,j,k) + dBx;
                By(i,j,k) = By(i,j,k) + dBy;
                Bz(i,j,k) = Bz(i,j,k) + dBz;
            end
        end
    end
end
```

calcular_gradiente_Bz.m

```
1 function dBz_dz = calcular_gradiente_Bz(zg, Bz, xg, yg)
2     centro_x = round(length(xg)/2);
3     centro_y = round(length(yg)/2);
4     Bz_centro = squeeze(Bz(centro_x, centro_y, :));
5
6     delta = 0.005;
7     Bz_forward = interp1(zg, Bz_centro, zg + delta, 'linear', 'extrap');
8     Bz_backward = interp1(zg, Bz_centro, zg - delta, 'linear', 'extrap');
9     dBz_dz = (Bz_forward - Bz_backward) / (2 * delta);
10
11    figure;
12    plot(zg, dBz_dz, 'r');
13    xlabel('z (m)');
14    ylabel('dBz/dz (T/m)');
15    title('Gradiente del campo magnético');
16    grid on;
end
```

La función calcular_gradiente_Bz.m determina la variación del componente vertical del campo magnético a lo largo del eje de caída.

Se utiliza una aproximación por diferencias finitas para obtener dBz/dz en cada punto central de la malla.

El resultado es un perfil detallado de cómo cambia el campo magnético en la dirección de movimiento, información esencial para cuantificar el efecto del frenado magnético a lo largo de la trayectoria de la góndola.

calcular_fuerza_magetica.m

```
1 [-] function Fz = calcular_fuerza_magetica(dBz_dz)
2             mz = 3000;
3             Fz = mz * dBz_dz;
4         end
```

La fuerza magnética de frenado se evalúa en la función `calcular_fuerza_magetica.m` como el producto entre el momento magnético del dipolo y el gradiente del campo.

$$F_z = m_z \frac{dB_z}{dz}$$

Este valor representa la fuerza neta que actúa en sentido opuesto al movimiento, y es la responsable de desacelerar la góndola durante su recorrido en la región conductora.

Este resultado alimenta la simulación dinámica, permitiendo estudiar el comportamiento de la góndola bajo diferentes condiciones de frenado.

simular_movimiento.m (1/2)

```
+4 calcular_campo_BiotSavart.m x calcular_gradiente_Bz.m x calcular_fuerza_magnetica.m x simular_movimiento.m
1 function [t, z_dipolo, z_free] = simular_movimiento(zg, Fz)
2 % Parámetros físicos
3 m = 0.001; % Masa del dipolo (kg)
4 g = 9.81; % Gravedad (m/s2)
5 w = -m * g; % Peso (negativo porque z crece hacia arriba)
6
7 % Tiempo
8 dt = 0.01;
9 t = 0:dt:7;
10
11 % Estado inicial [z; v]
12 z_dipolo = zeros(size(t));
13 v = zeros(size(t));
14 z_dipolo(1) = 10; % Posición inicial
15
16 % Caída libre para comparación
17 z_free = zeros(size(t));
18 v_free = zeros(size(t));
19 z_free(1) = 10;
20
21 % Función de aceleración total
22 a_total = @(z, v) (interp1(zg, Fz, z, 'linear', 'extrap') + w * v) / m;
```

En esta sección, el código define los parámetros físicos del sistema: masa del dipolo, gravedad, y peso.

Se establece el paso temporal y los vectores para la posición y velocidad iniciales tanto de la góndola bajo frenado como en caída libre.

La función de aceleración total evalúa, en cada instante, la suma de las fuerzas de gravedad, y frenado magnético interpolando los datos calculados previamente.

No se considera la fricción viscosa en nuestro problema ya que se simula todo en un caso ideal de vacío.

simular_movimiento.m (2/2)

```
25 for i = 1:length(t)-1
26     % RK4 para z_dipolo y v
27     z_i = z_dipolo(i);
28     v_i = v(i);
29
30     k1_v = a_total(z_i, v_i);
31     k1_z = v_i;
32
33     k2_v = a_total(z_i + 0.5*dt*k1_z, v_i + 0.5*dt*k1_v);
34     k2_z = v_i + 0.5*dt*k1_v;
35
36     k3_v = a_total(z_i + 0.5*dt*k2_z, v_i + 0.5*dt*k2_v);
37     k3_z = v_i + 0.5*dt*k2_v;
38
39     k4_v = a_total(z_i + dt*k3_z, v_i + dt*k3_v);
40     k4_z = v_i + dt*k3_v;
41
42     v(i+1) = v_i + dt/6 * (k1_v + 2*k2_v + 2*k3_v + k4_v);
43     z_dipolo(i+1) = z_i + dt/6 * (k1_z + 2*k2_z + 2*k3_z + k4_z);
44
45     % Caída libre con Euler (sin campo magnético)
46     a_free = -g;
47     v_free(i+1) = v_free(i) + a_free * dt;
48     z_free(i+1) = z_free(i) + v_free(i+1) * dt;
49 end
```

El bucle principal implementa el método de Runge-Kutta de cuarto orden (RK4) para integrar numéricamente la ecuación de movimiento de la góndola bajo el efecto de todas las fuerzas consideradas.

Simultáneamente, se calcula la caída libre mediante el método de Euler, permitiendo comparar la evolución de ambas trayectorias.

Es que finalmente buscamos la simulación más cercana a las condiciones reales.

visualizar_espiras.m

```
1 function visualizar_espiras(x, y, z, dx, dy, dz)
2     figure;
3     quiver3(x, y, z, dx, dy, dz, 'r');
4     grid on; view(3);
5     xlabel('X'); ylabel('Y'); zlabel('Z');
6     title('Espiras');
7 end
```

La función `visualizar_espiras.m` genera una representación tridimensional de las espiras conductoras, mostrando tanto su disposición espacial como la dirección de la corriente. Esta visualización valida la correcta inicialización geométrica del sistema y facilita el análisis previo al cálculo de campos magnéticos.

visualizar_campo_magntico.m

visualizar_campo_magntico.m permite analizar el perfil espacial del campo generado por las espiras. El archivo produce dos gráficas en el plano XZ: una con barra de color para interpretar la magnitud del campo, y otra simplificada que puede emplearse como fondo en la animación de la trayectoria. Esta función es clave para validar la distribución del campo antes de simular el movimiento.

```
main.m    visualizar_campo_magntico.m +  
1 Function visualizar_campo_magntico(xg, zg, Bx, Bz)  
2 [Xm, Zm] = meshgrid(xg, zg);  
3 centro_y = round(size(Bx, 2)/2);  
4 Bx_XZ = squeeze(Bx(:, centro_y, :));  
5 Bz_XZ = squeeze(Bz(:, centro_y, :));  
6  
7 figure;  
8 hold on;  
9 pcolor(Xm, Zm, sqrt(Bx_XZ.^2 + Bz_XZ.^2).^(1/3));  
10 shading interp; colormap jet; colorbar;  
11 streamslice(Xm, Zm, Bx_XZ', Bz_XZ', 3);  
12 xlabel('X'); ylabel('Z');  
13 title('Campo magnético');  
14  
15  
16 figure;  
17 hold on;  
18 pcolor(Xm, Zm, sqrt(Bx_XZ.^2 + Bz_XZ.^2).^(1/3));  
19 shading interp;  
20 colormap jet;  
21 colorbar;  
22 streamslice(Xm, Zm, Bx_XZ', Bz_XZ', 3);  
23 % Ocultar elementos no deseados  
24 axis off; % Oculta los ejes  
25 delete(findobj(gcf, 'Type', 'ColorBar')); % Elimina la barra de color  
26 set(get(gca, 'Title'), 'Visible', 'off'); % Oculta el título  
27 set(get(gca, 'XLLabel'), 'Visible', 'off'); % Oculta la etiqueta del eje X  
28 set(get(gca, 'YLLabel'), 'Visible', 'off'); % Oculta la etiqueta del eje Y  
29  
30 % Guardar solo el plot del campo magnético  
31 exportgraphics(gcf, 'campo_magntico.png', 'Resolution', 1000);  
32  
33 end
```

visualizar_trayectoria.m

```
1 function visualizar_trayectoria(t, z_dipolo, z_free)
2     figure;
3     plot(t, z_dipolo, 'r-', 'DisplayName', 'Caida sobre un anillo');
4     hold on;
5     plot(t, z_free, 'b--', 'DisplayName', 'Caida Libre');
6     xlabel('Tiempo (s)');
7     ylabel('Posicion (m)');
8     title('Trayectoria del dipolo');
9     grid on;
10    legend('show');
11 end
```

La función `visualizar_trayectoria.m` muestra gráficamente la evolución temporal de la posición de la góndola. Se comparan directamente la trayectoria bajo frenado magnético y la de caída libre, facilitando el análisis cuantitativo y cualitativo de la efectividad del sistema de frenado.

animar_con_fondo.m

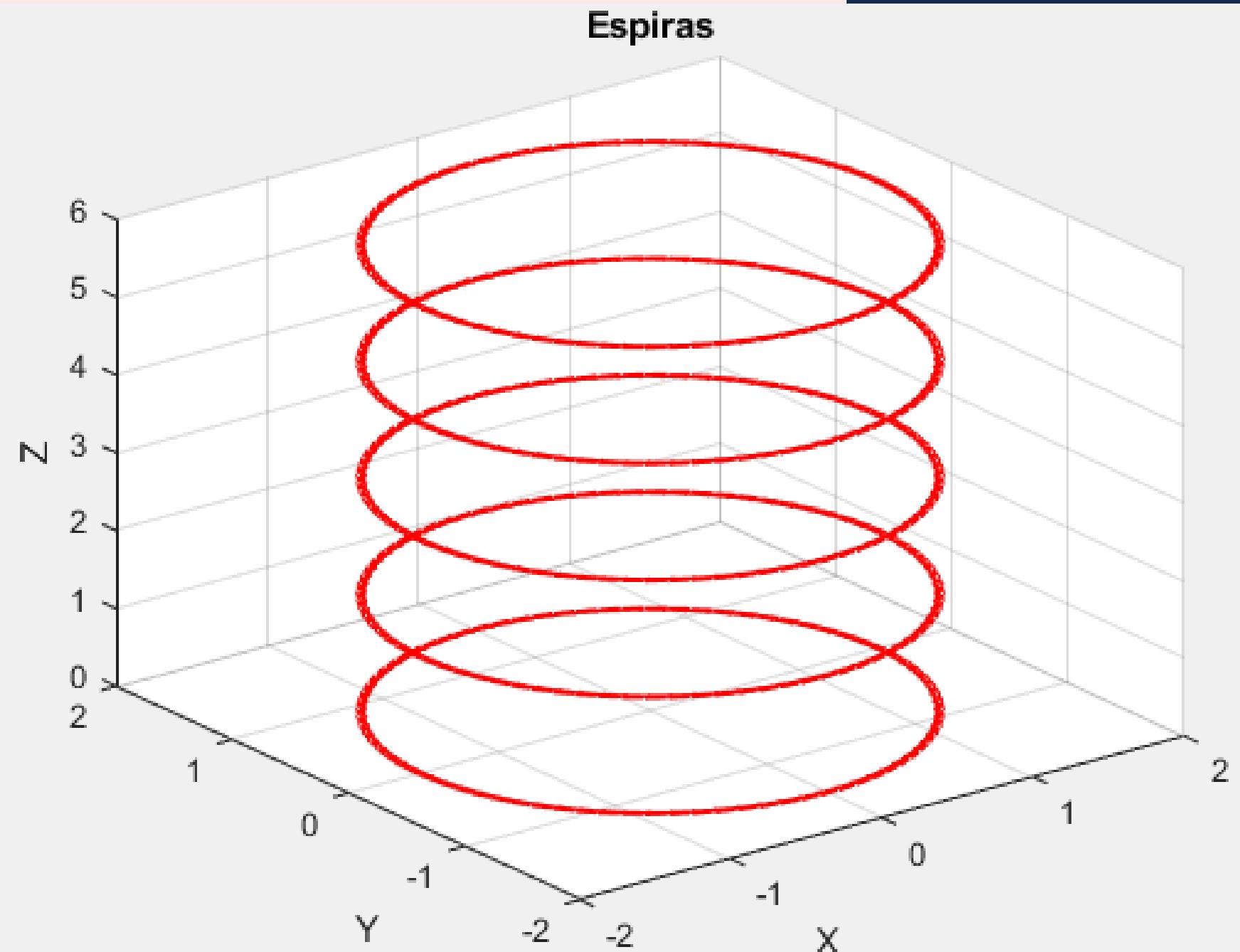
```
1 function animar_con_fondo(zm, imagen, nombre_video, zg)
2     x_range = [-0.1, 0.1]; % Pequeño rango en x
3     z_range = [min(zg), max(zg)];
4     writerObj = VideoWriter(nombre_video, 'Motion JPEG AVI');
5     writerObj.FrameRate = 30;
6     open(writerObj);
7     img = imread(imagen);
8     figure(5); clf;
9     imagesc(x_range, z_range, flipud(img));
10    axis xy;
11    hold on;
12    xlabel('x (m)'); ylabel('z (m)');
13    title('Trayectoria sobre campo magnético');
14    h = plot(0, zm(1), 'ro', 'MarkerSize', 10); % Punto inicial
15    for i = 1:length(zm)
16        set(h, 'YData', zm(i)); % Actualizar posición
17        drawnow;
18        frame = getframe(gcf);
19        writeVideo(writerObj, frame);
20    end
21    close(writerObj);
22 end
```

animar_con_fondo.m crea una animación que ilustra la trayectoria del dipolo magnético sobre el fondo del campo calculado previamente.

Esta visualización dinámica ayuda a comprender el efecto del frenado magnético en tiempo real y a identificar visualmente las regiones de máxima desaceleración.

Resultados e Interpretación

Espiras



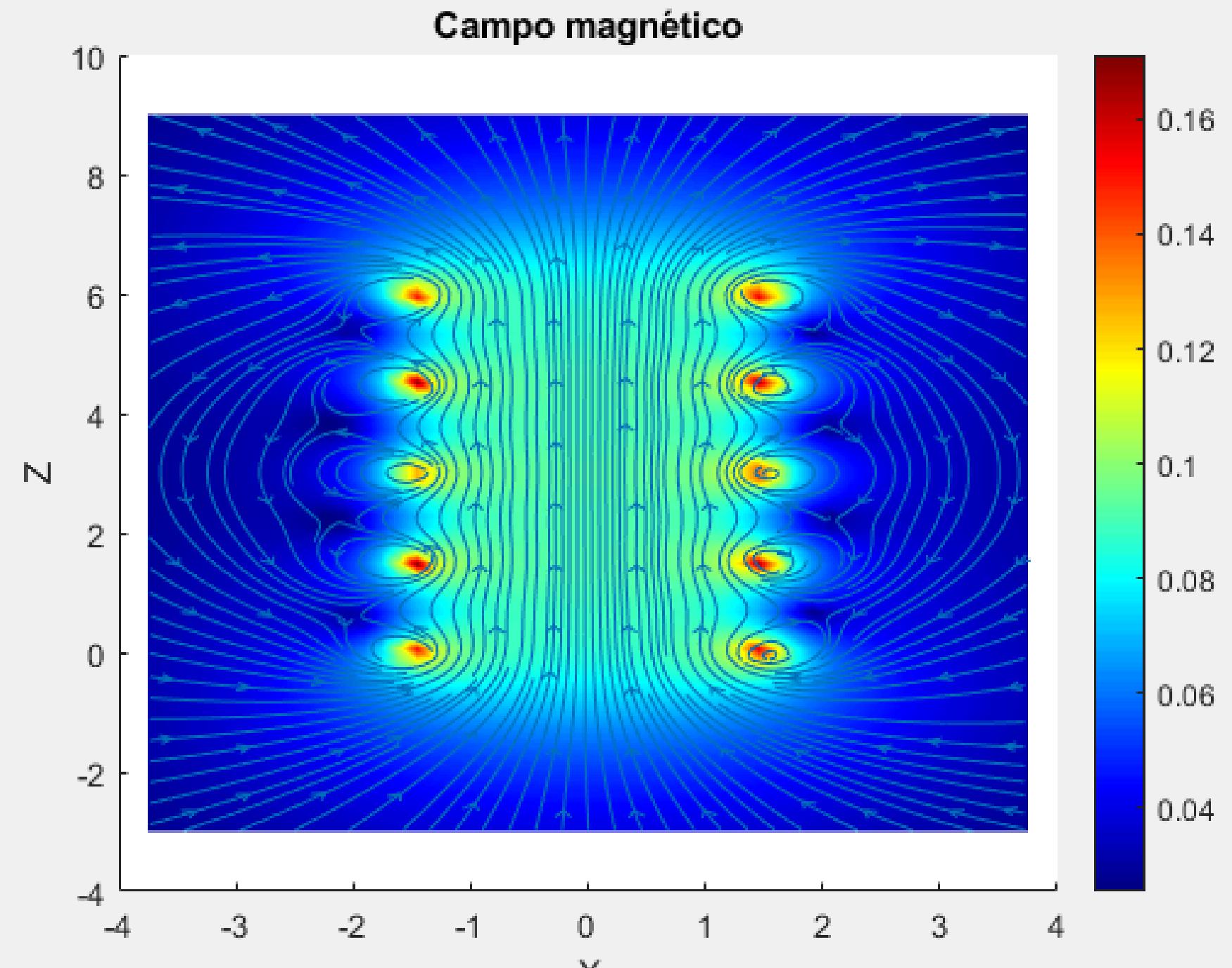
La imagen muestra la disposición tridimensional de las cinco espiras circulares utilizadas para modelar la región conductora responsable del frenado magnético en la torre de caída. Cada espira está apilada verticalmente a intervalos regulares sobre el eje Z, simulando de manera precisa la geometría real del sistema físico.

Visualización del campo magnético

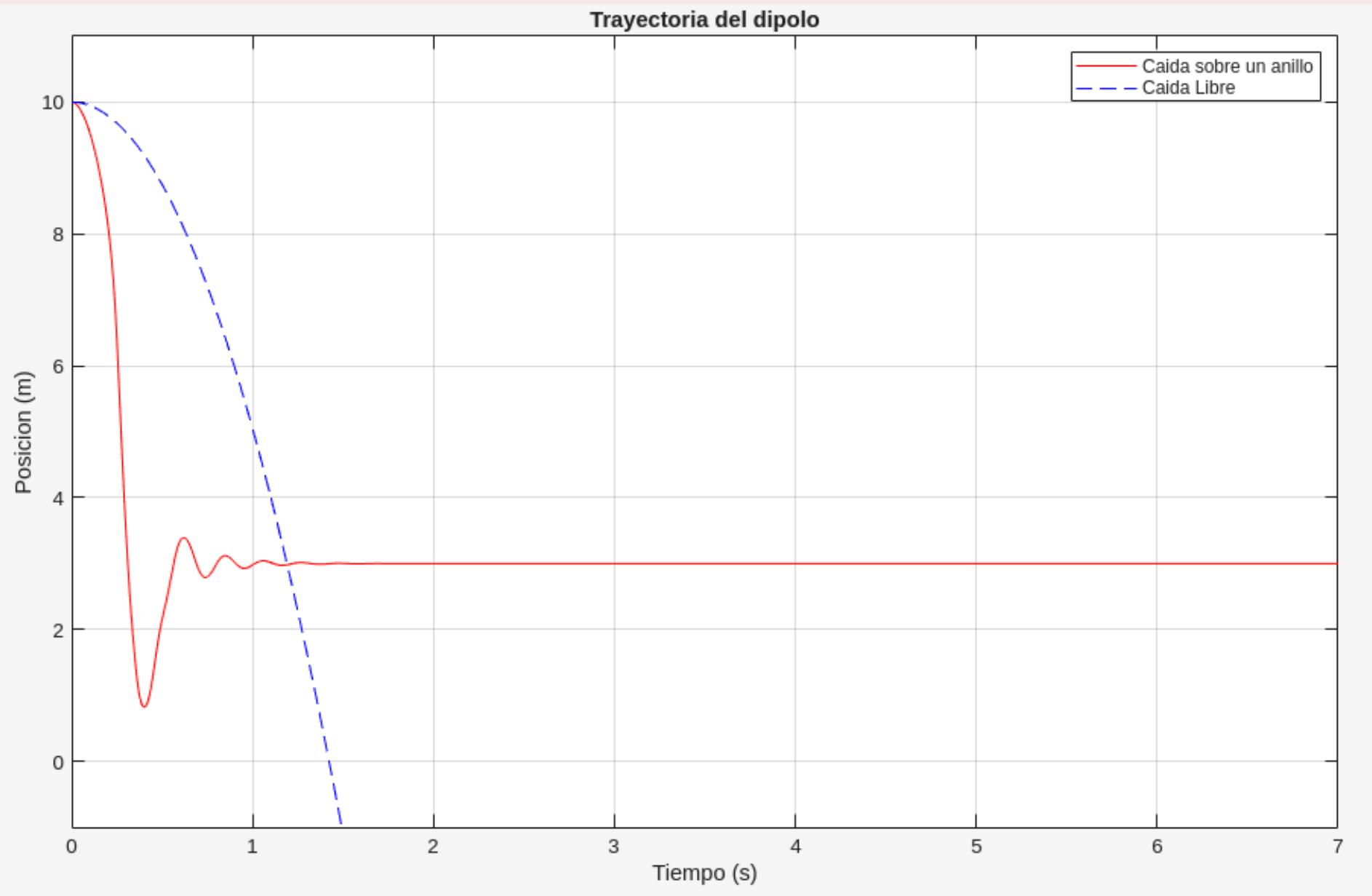
Esta gráfica muestra la intensidad y la dirección del campo magnético generado por las espiras en el plano XZ.

El color indica la magnitud del campo: rojo es más intenso, azul es menor.

Observamos que el campo es máximo y uniforme en el centro, justo donde ocurrirá el frenado de la góndola.

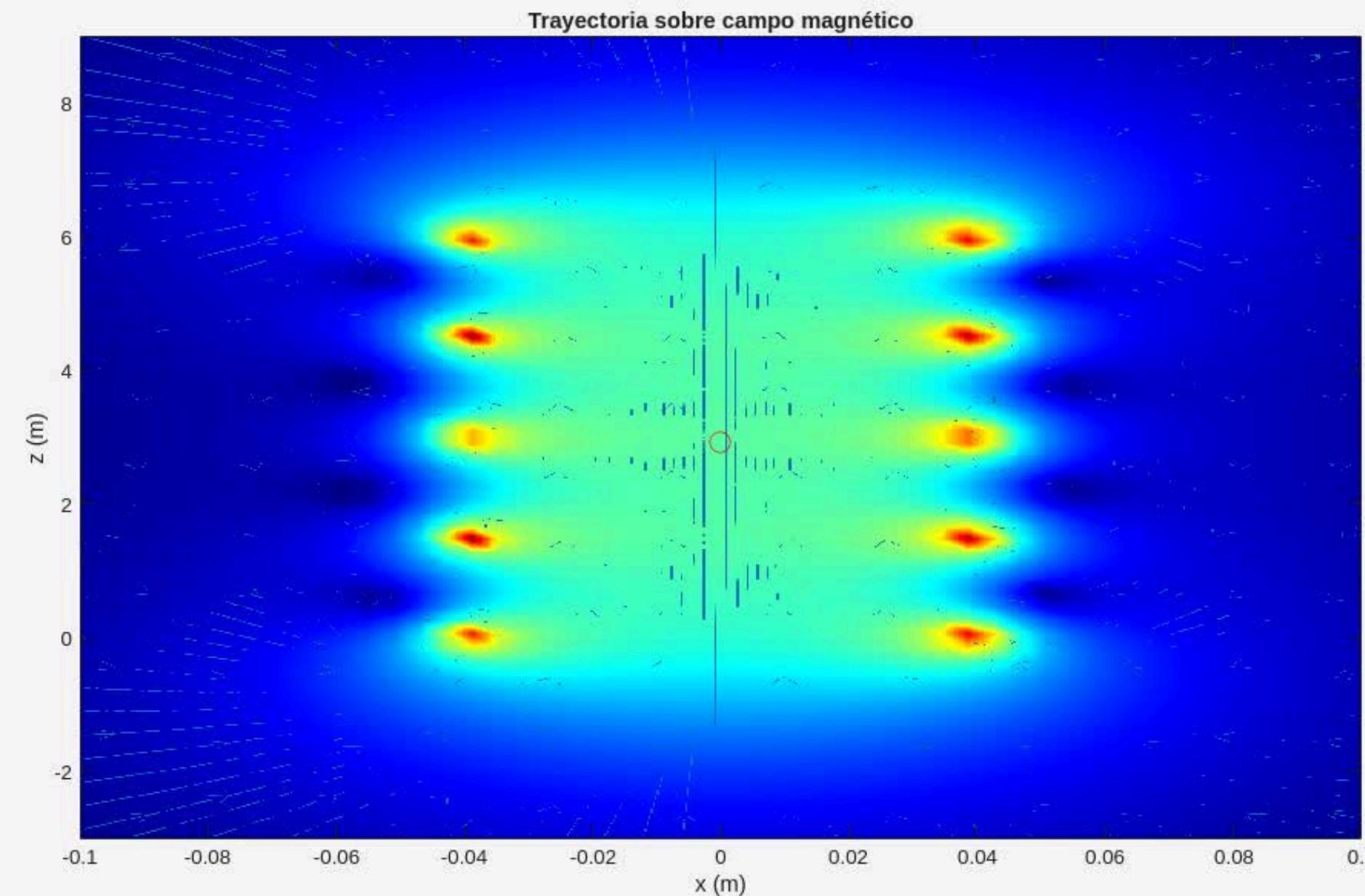


Trayectoria de góndolas



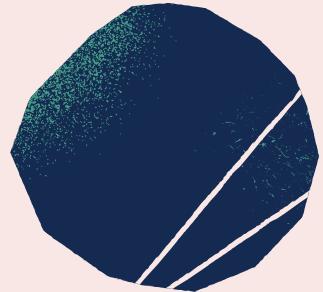
El frenado magnético detiene rápidamente la góndola, evitando que continúe cayendo como ocurriría en caída libre. Esto demuestra la efectividad del sistema.

Video



Video

El video que sacamos con animar_con_fondo.m muestra cómo se mueve el imán sobre el campo magnético en el plano XZ. El imán, representado por el círculo rojo, frena por cada espira que pasa, siendo la 3 espira en la que detiene por completo, comparado con la caída libre que simula el sistema apagado, se ve una clara diferencia, además que el imán se frena suavemente y no de golpe.



Conclusiones



Este proyecto fue muy interesante, usamos MATLAB para simular cómo un freno magnético para un iman en una torre de caída. El modelo, que armamos paso a paso, usa cosas como la ley de Biot-Savart y el método de Runge Kutta para calcular cómo se mueve el iman. Los resultados muestran que el freno magnético hace que el iman baje más lento, se nota la diferencia con la caída libre, demostrando un sistema funcional.

Referencias:

1. Bagno, E., & Eylon, B. S. (1997). From problem solving to a knowledge structure: An example from the domain of electromagnetism. *American Journal of Physics*, 65(8), 726-736. [24](#)
2. Bozzo, G., Michelini, M., & Santi, L. (2021). Atwood's machine and electromagnetic induction: A real quantitative experiment to analyze students' understanding. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(11), em2029. [24](#)
3. Ghayal, Y., Bhatt, R., & Patel, D. (2023). Research paper on eddy current braking system. *International Research Journal of Modernization in Engineering Technology and Science*, 5(5), 2755-2762. [2](#)
4. Härtel, H. (2018). Electromagnetic induction: An alternative for teaching and understanding. *European Journal of Physics Education*, 9(2), 1-15. [22](#)
5. Li, J., Zhang, K., & Wang, S. (2022). Simulation of the braking effects of permanent magnet eddy current brake and its effects on levitation characteristics of HTS maglev vehicles. *Actuators*, 11(10), 295. [11](#)
6. Michelini, M., & Vercellati, S. (2012). Modeling electromagnetic induction with analogies and experiments for conceptual change. *Latin-American Journal of Physics Education*, 6(1), 7-12. [24](#)
7. Neumann, E. (2023). Physics simulations: Interactive computational models for learning dynamics. *MyPhysicsLab Documentation*. Retrieved from <https://www.myphysicslab.com> [31](#)
8. Touzani, R., & Rappaz, J. (2014). Mathematical models for eddy currents and magnetostatics: With selected applications. Springer Science & Business Media. [9](#)
9. Wang, H., Chen, L., & Liu, Y. (2024). Eddy current mechanism model for dynamic magnetic field in ferromagnetic metal structures. *Electronics*, 13(18), 3772.