In [ ]:

In [ ]:

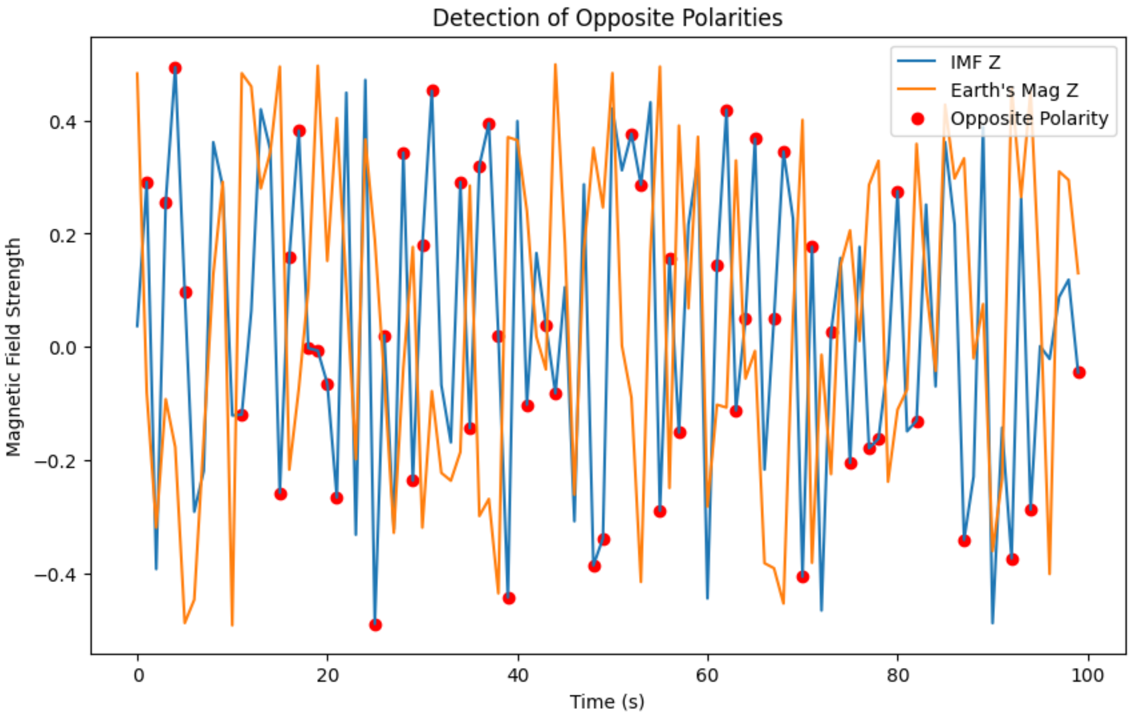
In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt

# Sample data (replace with your actual data)
timestamps = np.arange(0, 100, 1)  # Time in seconds
IMF_Z = np.random.rand(len(timestamps)) - 0.5  # Random IMF Z component (-0.5 to 0.5)
Earth_Mag_Z = np.random.rand(len(timestamps)) - 0.5  # Random Earth's magnetic field Z compone

# Detect intervals with opposite polarities
opposite_polarity_intervals = np.where(np.sign(IMF_Z) != np.sign(Earth_Mag_Z))

# Plot the data and highlight intervals with opposite polarities
plt.figure(figsize=(10, 6))
plt.plot(timestamps, IMF_Z, label='IMF Z')
plt.plot(timestamps, Earth_Mag_Z, label="Earth's Mag Z")
plt.scatter(timestamps[opposite_polarity_intervals], IMF_Z[opposite_polarity_intervals], color
plt.xlabel('Time (s)')
plt.ylabel('Magnetic Field Strength')
plt.legend()
plt.title('Detection of Opposite Polarities')
plt.show()
```



In [ ]:

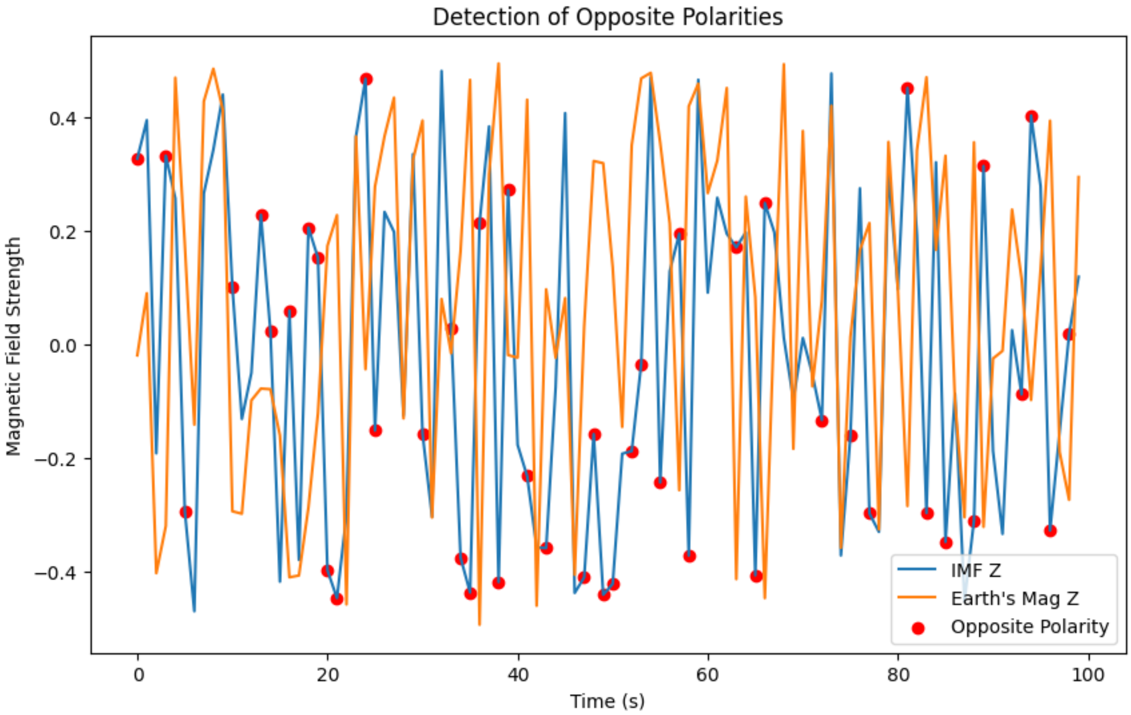
```
import numpy as np
import matplotlib.pyplot as plt

# Sample data (replace with your actual data)
timestamps = np.arange(0, 100, 1)  # Time in seconds
IMF_Z = np.random.rand(len(timestamps)) - 0.5  # Random IMF Z component (-0.5 to 0.5)
Earth_Mag_Z = np.random.rand(len(timestamps)) - 0.5  # Random Earth's magnetic field Z compone

# Detect intervals with opposite polarities
opposite_polarity_intervals = np.where(np.sign(IMF_Z) != np.sign(Earth_Mag_Z))

# Plot the data and highlight intervals with opposite polarities
```

```
plt.figure(figsize=(10, 6))
plt.plot(timestamps, IMF_Z, label='IMF Z')
plt.plot(timestamps, Earth_Mag_Z, label="Earth's Mag Z")
plt.scatter(timestamps[opposite_polarity_intervals], IMF_Z[opposite_polarity_intervals], color
plt.xlabel('Time (s)')
plt.ylabel('Magnetic Field Strength')
plt.legend()
plt.title('Detection of Opposite Polarities')
plt.show()
```



In [1]:

```
import pandas as pd

# Load data from a CSV file (replace 'your_data.csv' with your actual file path)
data = pd.read_csv('solar_wind.csv')

# Extract the columns for IMF and Earth's magnetic field components
X_IMF = data['bx_gsm']
Y_IMF = data['by_gsm']
Z_IMF = data['bz_gsm']

X_Earth = data['bx_gse']
Y_Earth = data['by_gse']
Z_Earth = data['bz_gse']

# Initialize counter for magnetic reconnection events
magnetic_reconnection_count = 0

# Time tolerance in seconds (can be adjusted based on your data)
time_tolerance = 1

# Loop through data points to find instances of opposite polarities in the Z component
num_data_points = len(data)

for i in range(num_data_points):

    # Check for opposite polarities in Z component
    if Z_IMF[i] * Z_Earth[i] < 0:

        # Check if the condition persists within the time_tolerance
        if i + time_tolerance < num_data_points and Z_IMF[i + time_tolerance] * Z_Earth[i + ti
```