[TEST SUITE
MODULATION OF SUSCEPTIBILITY SIGNALS TESTING]

024/7M

CONTEXT

Electromagnetic interference (EMI) is a critical aspect of spacecraft design and testing, as it can affect the performance and safety of the spacecraft and its components. In the context of NASA, EMI refers to the unwanted electromagnetic energy that can interfere with the operation of electronic systems.

NASA has established various standards and guidelines for EMI testing and mitigation, including the NASA electromagnetic interference/compatibility (EMI/EMC) test facility at NASA Johnson Space Center. This facility provides a controlled environment for testing the EMI characteristics of spacecraft and their components.

In terms of software, NASA uses various tools and simulations to analyze and mitigate EMI in spacecraft design. For example, NASA's Spacecraft Electromagnetic Compatibility (EMC) Analysis Tool (SEMCAT) is a software tool used to analyze the EMC of spacecraft systems.

For my specific test plan I decided to consider the following EMI-related tests and procedures:

- **Radiated EMI** testing to ensure that your spacecraft or component does not emit excessive electromagnetic energy
- **Conducted EMI** testing to ensure that your spacecraft or component is not susceptible to electromagnetic interference
- **Electromagnetic susceptibility** testing to ensure that your spacecraft or component can operate safely in the presence of electromagnetic interference
- **Electromagnetic emission** testing to ensure that your spacecraft or component does not emit excessive electromagnetic energy

to consider the following best practices for EMI mitigation in your spacecraft design:

- Proper shielding and grounding of electronic components
- Use of EMI filters and absorbers
- Designing components with EMI mitigation in mind
- Using EMI-resistant materials and coatings

SCOPE

- The purpose of this test is to evaluate the susceptibility of the Equipment Under Test (EUT) to modulation of susceptibility signals.
- The test will be conducted in accordance with NASA's electromagnetic interference (EMI) testing requirements.

DEMO SCOPE

- Only test cases design

TEST OBJECTIVES

- To determine the susceptibility of the EUT to modulation of susceptibility signals.
- To identify the most susceptible operating mode of the EUT.
- To evaluate the effectiveness of EMI mitigation measures implemented in the EUT.

TEST CASES

Test Case Categories

1. Environmental Testing:
    - Temperature testing
    - Vibration testing
    - Acoustic testing
    - **Electromagnetic interference (EMI) testing**
2. System Simulation:
    - Use system simulation software to model and test the on-Orbit System's behavior in various scenarios
3. Subsystem and Unit Testing:
    - Test each subsystem and unit individually to ensure they meet the requirements outlined in SMC-S-016
4. Integration Testing:
    - Test the integration of all subsystems and units to ensure seamless communication and functionality

[TEST SUITE]

Test Suite: Electromagnetic Interference (EMI) Testing

Test Case 1: Radiated Emissions

- Objective: Verify that the on-Orbit System's radiated emissions do not exceed the limits specified in NASA's SMC-S-016.
- Test Method:
    - Measure the radiated emissions from the on-Orbit System using a spectrum analyzer and antennas.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test distances: 1 meter, 3 meters, and 10 meters.
- Acceptance Criteria:
    - Radiated emissions do not exceed the limits specified in SMC-S-016.

Test Case 2: Conducted Emissions

- Objective: Verify that the on-Orbit System's conducted emissions do not exceed the limits specified in NASA's SMC-S-016.
- Test Method:
    - Measure the conducted emissions from the on-Orbit System using a line impedance stabilization network (LISN) and a spectrum analyzer.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test points: Power input, data interfaces, and other conductive interfaces.
- Acceptance Criteria:
    - Conducted emissions do not exceed the limits specified in SMC-S-016.

Test Case 3: Radiated Immunity

- Objective: Verify that the on-Orbit System can operate correctly in the presence of radiated electromagnetic fields.
- Test Method:

    - Expose the on-Orbit System to radiated electromagnetic fields using a transverse electromagnetic (TEM) cell or a reverberation chamber.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test levels: Up to 200 V/m.
- **Acceptance Criteria:**
    - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 4: Conducted Immunity

- Objective: Verify that the on-Orbit System can operate correctly in the presence of conducted electromagnetic disturbances.
- Test Method:
    - Inject conducted electromagnetic disturbances into the on-Orbit System's power input and data interfaces using a direct injection probe.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test levels: Up to 10 V.
- Acceptance Criteria:
    - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 5: Electromagnetic Pulse (EMP) Testing

- Objective: Verify that the on-Orbit System can withstand electromagnetic pulses (EMPs) without malfunction or degradation.
- Test Method:
    - Expose the on-Orbit System to EMPs using a EMP simulator.
    - Test levels: Up to 50 kV/m.
- Acceptance Criteria:
    - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 6: Lightning Strike Testing

- Objective: Verify that the on-Orbit System can withstand lightning strikes without malfunction or degradation.
- Test Method:
  - Simulate lightning strikes using a lightning strike simulator.
  - Test levels: Up to 200 kA.
- Acceptance Criteria:
  - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

[DEMO TEST AUTOMATION DESIGN]

Test Suite: Modulation of Susceptibility Signals Testing

Test Case 1: Amplitude Modulation

- Objective: Verify that the on-Orbit System can operate correctly in the presence of amplitude-modulated susceptibility signals.
- Test Method:
  - Generate an amplitude-modulated signal using a signal generator.
  - Inject the signal into the on-Orbit System's power input, data interfaces, or other susceptible points.
  - Test frequencies: 10 kHz to 40 GHz.
  - Test modulation depths: 10% to 90%.
- Acceptance Criteria:
  - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 2: Frequency Modulation

- Objective: Verify that the on-Orbit System can operate correctly in the presence of frequency-modulated susceptibility signals.
- Test Method:
  - Generate a frequency-modulated signal using a signal generator.
  - Inject the signal into the on-Orbit System's power input, data interfaces, or other susceptible points.

    - Test frequencies: 10 kHz to 40 GHz.
    - Test modulation frequencies: 10 Hz to 10 kHz.
- Acceptance Criteria:
    - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 3: Pulse Modulation

- Objective: Verify that the on-Orbit System can operate correctly in the presence of pulse-modulated susceptibility signals.
- Test Method:
    - Generate a pulse-modulated signal using a signal generator.
    - Inject the signal into the on-Orbit System's power input, data interfaces, or other susceptible points.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test pulse widths: 10 ns to 10 ms.
    - Test pulse repetition rates: 10 Hz to 10 kHz.
- Acceptance Criteria:
    - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 4: Mixed Modulation

- Objective: Verify that the on-Orbit System can operate correctly in the presence of mixed-modulated susceptibility signals (e.g., amplitude and frequency modulation).
- Test Method:
    - Generate a mixed-modulated signal using a signal generator.
    - Inject the signal into the on-Orbit System's power input, data interfaces, or other susceptible points.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test modulation depths and frequencies: vary according to the specific modulation type.
- Acceptance Criteria:
    - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Test Case 5: Modulation of Multiple Signals

- Objective: Verify that the on-Orbit System can operate correctly in the presence of multiple modulated susceptibility signals.
- Test Method:
  - Generate multiple modulated signals using multiple signal generators.
  - Inject the signals into the on-Orbit System's power input, data interfaces, or other susceptible points.
  - Test frequencies: 10 kHz to 40 GHz.
  - Test modulation depths and frequencies: vary according to the specific modulation type.
- Acceptance Criteria:
  - The on-Orbit System operates correctly and does not exhibit any malfunction or degradation.

Remember to consult [NASA's SMC-S-016] and other relevant standards and guidelines for specific testing requirements and procedures. Additionally, ensure that your test suite is tailored to the specific needs and requirements of your on-Orbit System.

Test Suite: [`modulation-susceptibility-tests`]

Test Case 1: Amplitude Modulation

- Test File: `amplitude-modulation.test.js`
- Test Method:
  - Use Playwright to automate the signal generator (e.g., Agilent N5182A) to generate an amplitude-modulated signal.
  - Use Playwright to automate the data acquisition system (e.g., National Instruments DAQmx) to measure the on-Orbit System's response to the modulated signal.
  - Test frequencies: 10 kHz to 40 GHz.

- Test modulation depths: 10% to 90%.

Test Steps:

```
const playwright = require('playwright');
(async () => {
  // Launch the signal generator and data acquisition system
  const signalGenerator = await playwright.chromium.launch();
  const dataAcquisitionSystem = await playwright.chromium.launch();

  // Generate an amplitude-modulated signal
  await signalGenerator.goto('http://signal-generator-ip-address');
  await signalGenerator.fill('input[name="frequency"]', '10 kHz');
  await signalGenerator.fill('input[name="modulationDepth"]', '50%');
  await signalGenerator.click('button[type="submit"]');
  // Measure the on-Orbit System's response
                                                                     await
dataAcquisitionSystem.goto('http://data-acquisition-system-ip-address
');
  await dataAcquisitionSystem.fill('input[name="channel"]', '1');
  await dataAcquisitionSystem.click('button[type="start"]');

  // Verify the on-Orbit System's response
  const response = await dataAcquisitionSystem.$('div.response');
  expect(response.textContent).toBe('Within acceptable limits');

  // Close the browsers
  await signalGenerator.close();
  await dataAcquisitionSystem.close();
})();
```

Test Case 2: Frequency Modulation

- Test File: `frequency-modulation.test.js`
- Test Method:
    - Use Playwright to automate the signal generator (e.g., Agilent N5182A) to generate a frequency-modulated signal.
    - Use Playwright to automate the data acquisition system (e.g., National Instruments DAQmx) to measure the on-Orbit System's response to the modulated signal.
    - Test frequencies: 10 kHz to 40 GHz.
    - Test modulation frequencies: 10 Hz to 10 kHz.
- Test Steps:

```
const playwright = require('playwright');
```

```
(async () => {
  // Launch the signal generator and data acquisition system
  const signalGenerator = await playwright.chromium.launch();
  const dataAcquisitionSystem = await playwright.chromium.launch();

  // Generate a frequency-modulated signal
  await signalGenerator.goto('http://signal-generator-ip-address');
  await signalGenerator.fill('input[name="frequency"]', '20 kHz');
     await  signalGenerator.fill('input[name="modulationFrequency"]',
'100 Hz');
  await signalGenerator.click('button[type="submit"]');
  // Measure the on-Orbit System's response
                                                                       await
dataAcquisitionSystem.goto('http://data-acquisition-system-ip-address
');
  await dataAcquisitionSystem.fill('input[name="channel"]', '1');
  await dataAcquisitionSystem.click('button[type="start"]');

  // Verify the on-Orbit System's response
  const response = await dataAcquisitionSystem.$('div.response');
  expect(response.textContent).toBe('Within acceptable limits');

  // Close the browsers
  await signalGenerator.close();
  await dataAcquisitionSystem.close();
})();
```

Test Case 3: Pulse Modulation

- Test File: `pulse-modulation.test.js`
- Test Method:
    - Use Playwright to automate the signal generator (e.g., Agilent N5182A) to generate a pulse-modulated signal.

- Use Playwright to automate the data acquisition system (e.g., National Instruments DAQmx) to measure the on-Orbit System's response to the modulated signal.
- Test frequencies: 10 kHz to 40 GHz.
- Test pulse widths: 10 ns to 10 ms.
- Test pulse repetition rates: 10 Hz to 10 kHz.

- **Test Steps:**

```
const playwright = require('playwright');
(async () => {
  // Launch the signal generator and data acquisition system
  const signalGenerator = await playwright.chromium.launch();
  const dataAcquisitionSystem = await playwright.chromium.launch();
  // Generate a pulse-modulated signal
  await signalGenerator.goto('http://signal-generator-ip-address');
  await signalGenerator.fill('input[name="frequency"]', '30 kHz');
  await signalGenerator.fill('input[name="pulse');

)}
```

API TEST SUITE

API using the auth/login endpoint

This test suite includes five test cases that cover the most important functionality of the "Modulation of Susceptibility Signals" system:

1. **Signal Modulation:** Tests that a signal can be modulated with the correct parameters.
2. **Signal Demodulation:** Tests that a signal can be demodulated with the correct parameters.

3. **Signal Analysis:** Tests that a signal can be analyzed with the correct parameters.
4. **Signal Visualization**: Tests that a signal can be visualized with the correct parameters.
5. **Error Handling:** Tests that the system returns an error response for an invalid request.

Test steps

```
// modulation-of-susceptibility-signals.test.js

const { test, expect } = require('@playwright/test');

test.describe('Modulation of Susceptibility Signals', () => {
  const API_ENDPOINT =
    'https://api.modulation-of-susceptibility-signals.nasa.gov';
  const USERNAME = 'nasa_user';
  const PASSWORD = 'nasa_password';

  test.beforeEach(async ({ request }) => {
    // Authenticate with the API
    const res = await request.post(`${API_ENDPOINT}/auth/login`, {
      data: {
        username: USERNAME,
        password: PASSWORD,
      },
    });
    expect(res.status()).toBe(200);
    const token = await res.json();
    expect(token.accessToken).not.toBeUndefined();
                         request.setHeader('Authorization',      `Bearer
${token.accessToken}`);
  });
```

```
  test('Test Case 1: Signal Modulation', async ({ request }) => {
    // Send a request to modulate a signal
    const signalId = 'ignal-123';
    const modulationParams = {
      frequency: 100,
      amplitude: 50,
    };
    const res = await request.post(
      `${API_ENDPOINT}/signals/${signalId}/modulate`,
      {
        data: modulationParams,
      }
    );
    expect(res.status()).toBe(200);
    const modulationResult = await res.json();
    expect(modulationResult.signalId).toBe(signalId);

expect(modulationResult.modulationParams).toEqual(modulationParams);
  });

  test('Test Case 2: Signal Demodulation', async ({ request }) => {
    // Send a request to demodulate a signal
    const signalId = 'ignal-123';
    const demodulationParams = {
      frequency: 100,
      amplitude: 50,
    };
    const res = await request.post(
      `${API_ENDPOINT}/signals/${signalId}/demodulate`,
      {
        data: demodulationParams,
      }
    );
    expect(res.status()).toBe(200);
    const demodulationResult = await res.json();
    expect(demodulationResult.signalId).toBe(signalId);

expect(demodulationResult.demodulationParams).toEqual(demodulationPar
```

```
ams);
  });

  test('Test Case 3: Signal Analysis', async ({ request }) => {
    // Send a request to analyze a signal
    const signalId = 'ignal-123';
    const analysisParams = {
      windowSize: 100,
      overlap: 50,
    };
    const res = await request.post(
      `${API_ENDPOINT}/signals/${signalId}/analyze`,
      {
        data: analysisParams,
      }
    );
    expect(res.status()).toBe(200);
    const analysisResult = await res.json();
    expect(analysisResult.signalId).toBe(signalId);
    expect(analysisResult.analysisParams).toEqual(analysisParams);
  });

  test('Test Case 4: Signal Visualization', async ({ request }) => {
    // Send a request to visualize a signal
    const signalId = 'ignal-123';
    const visualizationParams = {
      plotType: 'time-domain',
      xAxisLabel: 'Time (s)',
      yAxisLabel: 'Amplitude (V)',
    };
    const res = await request.post(
      `${API_ENDPOINT}/signals/${signalId}/visualize`,
      {
        data: visualizationParams,
      }
    );
    expect(res.status()).toBe(200);
    const visualizationResult = await res.json();
    expect(visualizationResult.signalId).toBe(signalId);
```

```
    expect(visualizationResult.visualizationParams).toEqual(
      visualizationParams
    );
  });

  test('Test Case 5: Error Handling', async ({ request }) => {
    // Send an invalid request to test error handling
    const signalId = 'invalid-signal';
    const res = await request.post(
      `${API_ENDPOINT}/signals/${signalId}/modulate`,
      {
        data: {
          frequency: 100,
          amplitude: 50,
        },
      }
    );
    expect(res.status()).toBe(404);
    const errorResponse = await res.json();
    expect(errorResponse.error).toBe('Signal not found');
  });
});
```

Made By me = [Juhan Tous](#) using Linux Add me I'm looking new opportunities contact me

[Saga/Ave Fenix]

@JUANSITOAUTH