CSE3063 LAB-06

Name: Preyash
Registration Number: 20BPS1022

Use Case Specifications

a.) Rover Navigation Control

Brief Description:
This use case outlines the process for controlling the navigation of a rover vehicle on a planetary surface. The rover's navigation control system allows users to input commands to steer the rover, set waypoints, and monitor its movement.

Basic Flow:

1) **User Input for Navigation:**
- User accesses the navigation control interface.
- User provides input commands, such as direction (forward, backward, left, right), speed, and waypoints.

2) **Navigation Command Execution:**
- The system processes the user's input and translates it into rover navigation commands.
- The rover executes the commands, adjusting its position and orientation accordingly.

3) **Monitoring and Feedback:**
- Users receive real-time feedback on the rover's position, orientation, and status.
- Users can view a live video feed or telemetry data from the rover.

4) **Waypoint Setting:**
- Users have the option to set waypoints for the rover to follow.
- The system records and tracks the set waypoints.

Alternative Flow:

1) **Unfulfilled Command:**
   If the system receives an invalid or unrecognized command, it notifies the user and requests a valid input.
2) **Prerequisites Not Met:**
   If the rover's navigation hardware or sensors are not functioning correctly, the system alerts the user and prevents navigation until the issues are resolved.
3) **Course Full (Obstacle Detected):**

If the rover encounters an obstacle in its path, it automatically stops and notifies the user. Users can then issue new navigation commands or adjust the path.

4) **Schedule Conflicts (Simultaneous Commands):**
If the system receives conflicting commands that cannot be executed simultaneously, it prioritizes and executes them in a sequential manner.

Pre-Conditions:

- The rover hardware and sensors are functional and connected.
- The rover is in a state ready for navigation.
- Users have access to the navigation control interface.
- Users have a basic understanding of the rover's navigation capabilities.

Post-Conditions:

- The rover has executed the user's navigation commands successfully.
- If waypoints were set, the rover follows the specified path.
- Users have received real-time feedback on the rover's status and position.
- Any encountered issues or obstacles have been appropriately addressed and communicated to the user.

b.) Rover Obstacle and Hazard Avoidance

Brief Description:

This use case outlines the process for the rover vehicle to autonomously detect and avoid obstacles and hazards on a planetary surface during navigation. The rover's obstacle and hazard avoidance system ensures safe traversal by detecting potential obstacles and adjusting its path accordingly.

Basic Flow:

1) **Automatic Hazard Detection:**

- The rover's sensors continuously scan the terrain for obstacles and hazards.
- If an obstacle or hazard is detected within the rover's path, the system proceeds to avoidance actions.

2) **Obstacle Avoidance Algorithm:**

- The system employs an obstacle avoidance algorithm to determine a safe path.
- The rover adjusts its navigation commands (e.g., steering, speed) to deviate from the detected obstacle while continuing toward its goal.

3) **Real-time Feedback:**

- Users receive real-time feedback on the rover's obstacle detection and avoidance.
- The system may display obstacle positions on a graphical interface for user awareness.

Alternative Flow:

1) **Unfulfilled Command:** If the system receives an invalid or unrecognized command related to obstacle avoidance, it notifies the user and requests a valid input.
2) **Prerequisites Not Met:** If the rover's sensors or avoidance algorithms are not functioning correctly, the system alerts the user and prevents navigation until the issues are resolved.
3) **Multiple Hazards Detected:** In scenarios where multiple obstacles or hazards are detected simultaneously, the system prioritizes avoidance actions based on proximity, size, or user-defined parameters.

Pre-Conditions:

- The rover hardware and sensors, including obstacle detection equipment, are functional and connected.
- The rover is in a state ready for navigation.
- Users have access to the navigation control interface, including options for obstacle and hazard avoidance.
- Users have a basic understanding of the rover's navigation and avoidance capabilities.

Post-Conditions:

- The rover has autonomously detected and successfully avoided obstacles or hazards.
- Users have received real-time feedback on obstacle detection and avoidance.
- Any encountered issues or obstacles have been appropriately addressed and communicated to the user.

c) Rover Sample Collection and Analysis

Brief Description:

This use case outlines the process for the rover vehicle to collect samples from a planetary surface and perform analysis on those samples. The rover's sample collection and analysis system allows users to specify target locations for sample collection and retrieve valuable data from those samples.

Basic Flow:

1) **User Input for Sample Collection:**

- Users access the sample collection and analysis interface.
- Users specify the target location(s) for sample collection.

2) **Rover Navigation to Collection Site:**

- The system processes the user's input and plans a navigation route to the specified collection site.
- The rover autonomously navigates to the designated location.

3) **Sample Collection:**

- The rover's robotic arm or collection mechanism retrieves samples from the planetary surface.
- Collected samples are securely stored on board.

-

4) **Analysis Request:**

- Users may request specific analyses to be performed on the collected samples.
- Users input analysis parameters and preferences.

5) **Sample Analysis:**

- The rover's onboard analysis instruments are deployed to perform the requested analyses.
- Data and results are generated from the sample analysis.

Alternative Flow:

1) **Unfulfilled Command:** If the system receives an invalid or unrecognized command related to sample collection and analysis, it notifies the user and requests a valid input.

2) **Prerequisites Not Met:** If the rover's sample collection or analysis equipment is not functioning correctly, the system alerts the user and prevents sample collection and analysis until the issues are resolved.

3) **Sample Collection Failure:** If the rover encounters difficulties in collecting samples (e.g., unable to reach the target location or retrieve samples), it notifies the user and provides options for resolution.

4) **Analysis Limitations:** In cases where the requested sample analysis is not possible due to instrument limitations, the system informs the user and suggests alternative analyses if available.

Pre-Conditions:

- The rover hardware and instruments, including sample collection and analysis tools, are functional and connected.
- The rover is in a state ready for navigation and sample collection.
- Users have access to the sample collection and analysis interface.
- Users have a basic understanding of the rover's sample collection and analysis capabilities.

Post-Conditions:

- The rover has successfully collected samples from the specified location(s).
- Analysis results are available for user review.
- Users have received real-time feedback on sample collection and analysis progress.
- Any encountered issues or analysis limitations have been appropriately addressed and communicated to the user.

d) Rover Health Monitoring and System Updates

Brief Description:

This use case outlines the process for monitoring the health and status of a rover vehicle deployed on a planetary surface. It also covers the procedure for updating the rover's software and system components to ensure optimal performance.

Basic Flow:

1) **Health Monitoring:**
- The rover's onboard sensors continuously monitor its various systems, including power, temperature, communication, and hardware components.
- Health data is collected and analyzed in real-time.

2) **Alert Generation:**
- If any anomalies or issues are detected during health monitoring, the system generates alerts.
- Alerts can include warnings related to temperature, power levels, or hardware malfunctions.

3) **Alert Notification:**
- Users and mission control receive notifications of generated alerts.
- Notifications can be in the form of messages, emails, or other communication channels.

4) **Diagnosis and Troubleshooting:**
- Mission control or rover operators analyze the received alerts and diagnose the specific issues.
- Troubleshooting steps may be initiated remotely or planned for rover intervention.

5) **System Updates Request:**
- When necessary, users or mission control can request system updates, including software patches or firmware upgrades.

6) **System Update Preparation:**
- Before performing system updates, the rover prepares by storing current system configurations and backup data.

7) **System Update Deployment:**
- New software or firmware updates are sent to the rover.
- The rover installs and verifies the updates, ensuring compatibility and stability.

Alternative Flow:

1) **No Health Issues Detected:** If no health issues are detected during monitoring, no alerts are generated, and the process proceeds without troubleshooting.

2) **Update Request Denied:** Mission control may deny a system update request if it poses risks or if the rover is not in a stable state. Users receive a notification explaining the denial and potential alternative actions.

Pre-Conditions:

- The rover's health monitoring sensors and systems are operational.
- The rover is in communication with mission control or operators.
- Users have access to the system update interface and authorization to request updates.
- Appropriate backup mechanisms are in place to preserve critical data during updates.

Post-Conditions:

- Health status and issues, if any, have been assessed and addressed.
- System updates have been successfully applied, verified, and validated.
- Users and mission control have received notifications and reports on health status and updates.