

**REPORT ON design and functional simulation of land rover FIGO FSM**

CREATED BY – VAIBHAV SINGH

TEAM NAME- SPARTANS

**Flow chart on design and functional simulation of land rover FIGO FSM**

END

FINALIZE DESIGN

INTEGRATION AND TESTING

DEFINE REQUIREMENTS

FUNCTIONALSIMULATION

DESIGN FSM

**START**

DEFINE REQUIREMENTS

* GATHER SPECIFICATIONS AND USER NEEDS
* ANALYZE SYSTEM CONSTRAINTS
* DOCUMENT REQUIREMENTS

**GATHER SPECIFICATIONS AND USER NEEDS**

* Conduct market research and user surveys
* Engage with stakeholders to identify requirements
* Define performance expectations (speed, power)
* Determine target cases and audiences
* Consider safety and regulatory requirements
* Document gathered specifications and user needs

**ANALYZE SYSTEM CONSTRAINTS**

* Evaluate available resources (budget, time, manpower)
* Assess technical capabilities and limitations
* Consider compatibility with existing systems
* Identify potential risks and challenges
* Document system constraints and limitations

**DOCUMENT REQUIREMENTS**

* Compile and review gathered specifications and user needs
* Define functional and non- functional requirements
* Seek feedback and validation from stakeholders
* Document finalized requirements

DESIGN FSM

🡪 Identify state and state variables

🡪Define transition between states

🡪Consider input and output signals

🡪Draw FSM diagram

**Identify state and state variables**

To identify states and state variables in the context of a Land Rover Figo FSM, we need a deeper understanding of the specific system and its behaviour. As an AI language model, I don't have access to real-time or model-specific data. However, I can provide you with some examples of possible states and state variables that could be relevant for a Land Rover Figo FSM:

**States:**

Engine Off

Engine On

Parked

Moving Forward

Moving Reverse

Idle

Braking

Turning

Emergency Stop

Maintenance Mode

**State Variables:**

Speed

Gear position

Accelerator pedal position

Brake pedal status

Steering angle

Engine temperature

Fuel level

Door status (open/closed)

Seatbelt status (fastened/unfastened)

System fault status

**DEFINE TRANSITION BETWEEN STATES**

Transitions between states in a Finite State Machine (FSM) define how the system moves from one state to another based on certain conditions or events. In the context of a Land Rover Figo FSM, here are some examples of possible transitions between states:

Transition: Engine Off to Engine On Condition: Ignition key turned to the "On" position Action: Start the engine

Transition: Engine On to Parked Condition: Vehicle speed is zero and gear position is in "Park" Action: Set the vehicle to a parked state

Transition: Parked to Moving Forward Condition: Accelerator pedal pressed and gear position is in "Drive" Action: Start moving forward

Transition: Moving Forward to Braking Condition: Brake pedal pressed Action: Apply brakes and slow down the vehicle

Transition: Braking to Parked Condition: Vehicle speed is zero and gear position is in "Park" Action: Bring the vehicle to a stop and set it to a parked state

Transition: Moving Forward to Moving Reverse Condition: Gear position is shifted from "Drive" to "Reverse" Action: Change the direction of vehicle movement

Transition: Moving Reverse to Moving Forward Condition: Gear position is shifted from "Reverse" to "Drive" Action: Change the direction of vehicle movement

Transition: Moving Forward to Turning Condition: Steering angle is beyond a threshold Action: Start the turning manoeuvre

Transition: Turning to Moving Forward Condition: Steering angle returns to a straight position Action: Continue moving forward after completing the turn

Transition: Any State to Emergency Stop Condition: Emergency stop button pressed or critical fault detected Action: Immediately halt all vehicle functions and activate emergency procedures

FUNCTIONAL SIMULATION

Functional simulation is a crucial step in the design and development of a system, such as a Land Rover Figo FSM. It involves implementing the FSM model in a simulation tool and testing its behaviour under various input scenarios. Here's an outline of the functional simulation process:

Implement FSM in a Simulation Tool: Choose a suitable simulation tool (e.g., Simulink) and use it to create a digital representation of the Land Rover Figo FSM based on the FSM design. This involves coding the logic, states, state variables, transitions, and input/output signals.

Define Input Scenarios and Test Cases: Identify a range of input scenarios that the system is expected to encounter during normal operation. This can include accelerator pedal inputs, brake inputs, steering inputs, and other relevant signals. Define test cases that cover different combinations and sequences of inputs.

Simulate the FSM Behaviour: Run the functional simulation by providing the defined input scenarios and test cases to the FSM model. The simulation tool will execute the FSM logic and generate the corresponding output signals based on the defined transitions and state changes.

Observe and Analyse Results: Monitor the simulation outputs and observe the behaviour of the Land Rover Figo FSM under different input scenarios. Pay attention to the transitions between states, the consistency of the system's response, and any unexpected behaviour that may arise.

Debug and Optimize FSM Design: If any issues or unexpected behaviour is observed during the simulation, debug the FSM model to identify and resolve the underlying problems. This may involve modifying the FSM logic, adjusting state transitions, or refining the input/output signals.

Iterate the Simulation: Repeat the simulation process with different input scenarios, test cases, and potential edge cases to ensure the robustness and reliability of the Land Rover Figo FSM. Iterate on the FSM design, simulation, and debugging until the desired functionality and performance are achieved.

INTEGRATION AND TESTING

* Integrate FSM with overall control system
* Perform hardware in the loop testing
* Validate the system against requirements
* Iteratively improve and refine the design

**Integrate FSM with overall control system**

1. Identify System Components: Identify the various components of the overall control system that need to interact with the FSM. This can include sensors, actuators, controllers, human-machine interfaces (HMI), communication interfaces, and other relevant subsystems.
2. Define Communication Interfaces: Determine the communication interfaces and protocols required for the FSM to exchange data and commands with other system components. This could involve using standard protocols like CAN (Controller Area Network), Ethernet, or custom interfaces specific to the system.
3. Implement FSM Interface: Modify or enhance the FSM implementation to accommodate the communication interfaces and data exchange mechanisms required for integration. This may involve updating the FSM logic and state transitions to send and receive appropriate messages or signals.
4. Connect Data Inputs and Outputs: Establish connections between the FSM and the input/output signals of the overall control system. This includes connecting the FSM to sensors that provide input data (e.g., speed sensors, temperature sensors) and connecting the FSM to actuators that receive commands (e.g., engine control module, brake system).
5. Implement Message Parsing and Processing: Develop the necessary software or hardware modules to parse and process the messages exchanged between the FSM and other system components. This ensures that the FSM can interpret and respond to the incoming data and commands appropriately.
6. Test Communication and Compatibility: Conduct tests to verify the communication between the FSM and other system components. Validate the compatibility of data formats, message protocols, and the overall functionality of the integrated system.
7. Address Integration Issues: Identify and resolve any integration issues that arise during the integration process. This may involve debugging, modifying the FSM logic, adjusting interface configurations, or updating the communication protocols.
8. Verify Functional Interactions: Perform tests and simulations to ensure that the FSM interacts correctly with the other control system components. Validate that the FSM's behaviour aligns with the system's overall goals and requirements.
9. Validate System Performance: Evaluate the performance of the integrated system as a whole, considering factors such as response time, accuracy, stability, and overall system behaviour. Verify that the FSM functions reliably within the control system context.
10. Iterate and Refine: Iterate the integration process as necessary to address any identified issues, optimize performance, and improve the overall system integration. This may involve making iterative adjustments to the FSM or other system components.
11. Document Integration: Document the integration process, including the configurations, interfaces, and interactions between the FSM and the overall control system. Capture any lessons learned, recommendations, or modifications made during the integration process.

FINALIZE DESIGN

Finalizing the design of the Land Rover Figo FSM involves reviewing and validating the FSM's functionality, performance, and reliability, making necessary adjustments, and documenting the finalized design. Here's an outline of the steps involved:

1. Review FSM Requirements: Review the initial FSM requirements and ensure that they have been met. Verify that the FSM design aligns with the desired functionality and specifications.
2. Validate FSM Behavior: Conduct thorough testing and simulations to validate the FSM's behavior under various scenarios and conditions. Verify that the FSM correctly transitions between states, produces the expected outputs, and responds appropriately to input signals.
3. Assess Performance and Efficiency: Evaluate the performance of the FSM in terms of speed, accuracy, responsiveness, and resource utilization. Identify any areas where performance improvements can be made and optimize the FSM design accordingly.
4. Conduct Error Handling and Fault Analysis: Evaluate the FSM's error handling capabilities and its ability to handle exceptional or fault conditions. Identify any potential vulnerabilities or weaknesses in the FSM's fault management and recovery mechanisms, and make necessary improvements.
5. Validate Integration with System Components: Verify that the FSM integrates seamlessly with other system components, such as sensors, actuators, controllers, and communication interfaces. Test the FSM's interactions with these components to ensure compatibility and reliable operation.
6. Consider Safety and Security Requirements: Evaluate the FSM design with respect to safety and security requirements. Identify and address any potential risks, vulnerabilities, or cybersecurity concerns associated with the FSM's operation.
7. Incorporate Feedback and Lessons Learned: Take into account feedback and insights gained throughout the design and testing process. Consider any suggestions or recommendations from stakeholders, users, or testing teams to refine the FSM design.
8. Make Necessary Design Adjustments: Based on the validation results and feedback received, make any necessary adjustments to the FSM design. This may involve modifying the FSM logic, updating state transitions, improving error handling mechanisms, or enhancing performance.
9. Document the Finalized Design: Document the finalized FSM design, including its architecture, state diagram, state variables, transitions, and interfaces. Provide clear and comprehensive documentation that can be used for future reference, maintenance, and further development.
10. Conduct Design Review: Conduct a formal design review with relevant stakeholders to present the finalized FSM design. Seek feedback, address any remaining concerns, and obtain approval for the design before proceeding to the next phase of the project.
11. Communicate Design Changes: Ensure that all relevant teams, including development, testing, and implementation, are informed about the finalized FSM design. Communicate any design changes or updates to the appropriate parties to ensure a smooth transition to the implementation phase.