Instituto Politécnico de Beja

School of Technology and Management

Master's in Informatics Engineering and Internet of Things

Motorcycle Motion Monitoring

IOPT net-based technical-scientific report

Darya Martsinouskaya, 26610

Beja, 30 June 2025

INSTITUTO POLITÉCNICO DE BEJA

School of Technology and Management

Master's in Informatics Engineering and Internet of Things

Motorcycle Motion Monitoring

IOPT net-based technical-scientific report

Darya Martsinouskaya, 26610

Orientado por :

Professor João Paulo Barros, IPBeja

Model-Based Development

2025

Resumo

Motorcycle Motion Monitoring

IOPT net-based technical-scientific report

Este projeto apresenta o design e a implementação de um sistema reativo de monitoramento de segurança para motociclistas, utilizando Redes de Petri do tipo Input/Output Place/Transition (IOPT). O sistema avalia dados de entrada que podem ser recebidos de sensores como o DHT11 (sensor de temperatura e humidade), o MPU6050 (giroscópio e acelerómetro) e o GPS (para condições de tráfego/estrada), com o objetivo de detectar situações potencialmente perigosas durante a condução. O modelo incorpora condições de guarda para a tomada de decisões inteligentes, ações de saída para acionar alertas e processamento paralelo para lidar simultaneamente com múltiplos fatores ambientais e rodoviários. A sincronização dos dados é realizada por meio de um mecanismo centralizado de registo, garantindo o registo completo e consistente de todos os parâmetros monitorizados. A arquitetura modular permite escalabilidade e integração com outros componentes de transporte inteligente. Esta abordagem demonstra a eficácia da modelação formal com Redes de Petri na melhoria da segurança rodoviária e da consciência do condutor em aplicações com suporte à Internet das Coisas (IoT).

Palavras-chave: *Rede IOPT, Monitorização da Temperatura e Humidade, Monitorização das Condições das Estradas e do Trânsito, Monitorização de Movimento, Registo de Dados, Modelação Formal.*

Abstract

Motorcycle Motion Monitoring

IOPT net-based technical-scientific report

This project presents the design and implementation of a reactive safety monitoring system for motorcycle riders using Input/Output Place/Transition (IOPT) Nets. The system evaluates input data which can be received from such sensors as the DHT11 (temperature and humidity sensor), MPU6050 (gyroscope and accelerometer), GPS (traffic/road condition inputs) to detect potentially dangerous riding conditions. The model incorporates guard conditions for intelligent decision-making, output actions to trigger alerts, and parallel processing to handle multiple environmental and road factors simultaneously. Data synchronization is achieved through a centralized logging mechanism, ensuring complete and consistent recording of all monitored parameters. The modular architecture supports scalability and integration with other smart transportation components. This approach demonstrates the effectiveness of formal Petri Net modeling in enhancing road safety and rider awareness in Internet of Things (IoT)-enabled applications.

Keywords: *IOPT net, Temperature and Humidity Monitoring, Road And Traffic Conditions Monitoring, Motion Monitoring, Data Logging, Formal Modeling.*

Conteúdo

Lista de Figuras

Capítulo 1

Introduction

Motorcycle riders face a range of safety challenges due to their increased exposure to environmental hazards, limited protection compared to enclosed vehicles, and the inherently dynamic nature of two-wheeled travel. Key risk factors include abrupt acceleration or braking, excessive tilt during cornering, slippery road conditions, and unpredictable traffic scenarios. These factors demand constant rider attention and adaptive behavior to prevent accidents and ensure safe riding practices. Motorcyclists are significantly more likely to be seriously injured or killed in road accidents than car occupants, especially in developing regions and rural areas where road infrastructure and emergency services may be limited.

While several commercial solutions exist for motorcycle monitoring, they are typically integrated into high-end motorcycles and remain inaccessible to the majority of riders due to cost and hardware limitations. For example, Harley-Davidson motorcycles incorporate embedded sensors and accelerometers to detect unsafe motion patterns and predict potential emergency situations [Har24]. Similarly, BMW Motorrad's ConnectedRide system features advanced sensor fusion (accelerometer, gyroscope, GPS) combined with cellular or Wi-Fi connectivity to monitor and enhance stability, braking, and suspension control in real time [Fas24]. However, these solutions rely heavily on stable internet connections and high-end embedded hardware, making them unsuitable for regions with poor connectivity or for use on older or lower-cost motorcycles.

Smartphone-based applications offer a more affordable alternative, such as those employed in the Kawasaki Ninja H2, which use built-in gyroscopes and accelerometers to track motion and performance metrics [Kaw24]. However, these systems are limited by the precision and reliability of mobile sensors, battery life constraints, and the inability to process large volumes of data locally without performance degradation. As a result, they often lack the robustness and responsiveness needed for accurate real-time safety feedback.

To bridge this gap, Internet of Things (IoT)-based motorcycle monitoring systems have emerged as a promising solution. Motion sensors—primarily accelerometers, gyroscopes, and GPS modules—play a critical role in capturing real-time data on speed, acceleration, tilt angles, and trajectories [TY24]. This sensor data, when analyzed effectively, enables detection of hazardous riding behavior and environmental risks, offering timely feedback to riders and promoting safer navigation strategies.

For instance, Rahman and Manoj (2024) proposed an IoT-based crash detection system using gyro and GPS sensors that alerts emergency services upon detecting accidents, potentially reducing emergency response times and fatalities [JM24]. Micko et al. (2021) reviewed integrated sensor solutions that monitor both road and vehicle dynamics to guide riders toward safer speeds and routes based on current riding conditions [I Z23]. These studies emphasize the vital role of multi-sensor integration and context-aware feedback mechanisms in reducing motorcycle accident rates.

Beyond motion parameters, researchers are now exploring the integration of environmental sensors to capture road surface conditions, weather data, and temperature, further enriching the context for risk assessment and proactive intervention [Zai+24]. The fusion of such heterogeneous data streams can significantly improve the precision and scope of rider safety systems, particularly when operating in unfamiliar or hazardous environments.

Equally important is the system architecture for processing and reacting to incoming sensor data. Three main computational models dominate this domain: edge computing, cloud computing, and hybrid approaches. Edge computing, which processes data locally on the device or nearby hardware, is highly effective for applications requiring low latency and real-time decision-making. Daraghmi (2022) demonstrated a vehicle speed monitoring system that utilizes edge computing to analyze traffic behavior in real time, ensuring fast response without relying on cloud infrastructure [Dar22]. Similarly, Akhtar et al. (2024) applied lightweight AI models like SSD Mobilenet on microcontrollers to detect dangerous motorbike maneuvers under resource-constrained conditions [AV24].

Cloud computing, on the other hand, excels at large-scale analytics and behavioral modeling. Kristiani et al. (2022) integrated cloud services with Narrowband IoT for data aggregation across multiple motorcycle sensors, enabling long-term pattern recognition and predictive maintenance [EY24]. Mukhopadhyay et al. (2024) emphasized the role of cloud computing in smart transportation systems, where aggregated data from various vehicles help optimize traffic flow and prevent accidents on a systemic level [Muk+24].

The hybrid model offers the best of both worlds, combining local responsiveness with cloud-based intelligence. Future systems can dynamically balance tasks between edge and cloud to optimize energy efficiency, cost, and performance. Research into hybrid models can also unlock scalable deployments, allowing personalized feedback and coordinated insights across entire fleets or regions.

Despite these advancements, most commercial and academic implementations remain either too expensive, too complex, or insufficiently adaptable for low-budget deployments. In response to these limitations, our research proposes a cost-effective and locally operated motorcycle motion monitoring system, built around the ESP32 microcontroller. The system integrates accelerometers and gyroscopes to track acceleration, tilt, and movement patterns in real time. It also incorporates environmental data inputs, such as temperature, humidity, traffic conditions, and road surface danger alerts, using additional sensors and user-reported information.

When potentially dangerous conditions are detected—based on a combination of sensor readings and contextual data—the system provides contextual suggestions to the rider, such as reducing speed, turning on lights, drinking water, changing the route, or adjusting the departure time. If no hazard is identified or once it is resolved, the system logs the event and resumes continuous monitoring.

The motivation for formal modeling using IOPT (Input-Output Place Transition nets) lies in the complexity of coordinating input-driven behavior, asynchronous sensor readings, and conditional feedback in real time. A formal model enables:

- Structured representation of logic behind sensor input processing and safety response generation;

- Simulation of event sequences such as danger detection → suggestion issuance → rider response → event logging;

- Verification of correct behavior under simultaneous or conflicting input conditions;

- Modular extension for new sensor types, rider behaviors, or risk categories.

Through the IOPT tools, the model can be validated, simulated, and visualized, ensuring behavioral correctness and facilitating scalability testing. This approach guarantees that the system reacts appropriately in various environmental and riding scenarios, offering a reliable, extensible, and affordable safety enhancement for motorcyclists—especially in contexts where commercial alternatives are unavailable or unsuitable.

The report is structured as follows: Section 2 provides a brief overview of the IOPT modeling and simulation tool, outlining its relevance to embedded and reactive system design. Section 3 introduces the key IOPT Petri net concepts applied in this project. Section 4 describes the construction of the model, including its structural diagram, token flow, and defined input/output signals. Section 5 presents the simulation and verification of the model, including detailed simulation scenarios that test different environmental and behavioral conditions, as well as a formal reachability and deadlock analysis to ensure model correctness and completeness. Section 6 discusses the results obtained during simulation and validates the model's expected behavior. Finally, Section 7 concludes the report by summarizing the main achievements and potential improvements for future iterations of the system.

Capítulo 2

Brief Overview of the IOPT Tools

IOPT-Tools is well-suited for the development of embedded control systems like a motorcycle motion and event monitoring platform due to its model-driven and automation-oriented approach. It enables the formal specification of system behavior using Input-Output Place-Transition nets, which are ideal for describing reactive systems with multiple input signals (e.g., from sensors like accelerometers or motion detectors) and output actions (e.g., triggering data logging or alerts).

The platform offers visual modeling and simulation tools that help in designing and validating system logic before hardware deployment. This reduces the likelihood of errors and speeds up development. Its support for automatic code generation in both C and VHDL formats means that models can be directly translated into executable code for deployment on embedded platforms such as Arduino, Raspberry Pi, or FPGAs—commonly used in IoT and vehicle monitoring applications.

Moreover, IOPT-Tools supports distributed systems through model composition and modularization. This makes it easier to design scalable and maintainable control logic, an essential feature when dealing with systems that may evolve to include additional sensors or components over time.

Overall, the tool's capabilities in simulation, verification, and code generation make it a reliable and efficient choice for developing and deploying embedded controller models in real-world applications [Per+22], [F P14].

Capítulo 3

IOPT Net Concepts Used

This chapter elaborates on the key modeling concepts employed in the developed IOPT (Input/Output Place/Transition) Petri Net, which was designed to monitor motorcycle riding conditions and enhance safety through real-time decision-making. The net models a reactive system capable of detecting hazardous environmental and traffic conditions, issuing alerts, and logging data for post-analysis. The core concepts implemented are discussed below.

Input/Output Place/Transition (IOPT) Formalism

IOPT Petri Nets extend traditional Petri Nets by incorporating input/output capabilities, guard conditions, and actions—making them particularly suitable for modeling embedded and real-time systems. In this project, IOPT enabled the integration of real-time sensor signals (temperature, humidity, motion, road, and traffic data) into the decision-making workflow of the safety monitoring system.

Guard Conditions

Guard conditions are logical expressions attached to transitions, which determine whether a transition can fire based on current input values. This allows the system to react only to specific situations. For example:

- CheckTempDanger Transition: Fires only if the "in temp"input signal is less then 10 or more than 25, indicating unsafe temperature levels.

- CheckTrafficDanger Transition: Fires if the "in traffic heavy"input signal == true, identifying hazardous traffic conditions.

These conditions allow for deterministic behavior and precise control flow based on sensor input.

Output Actions

Transitions in the net can perform output actions when fired. These are used to trigger system responses, such as issuing warnings or logging events.

Examples:

- If an unsafe temperature is detected, the transition sets the "alert temp = 1".
- If road conditions are satisfactory, the transition may set the "safe road = 1".

These output actions ensure that safety-critical information is conveyed to the rider or external monitoring system promptly.

Parallelism

A critical feature of the model is its ability to process multiple safety checks in parallel. Once motion is detected and data is collected and processed, the net enables concurrent evaluation of:

- Temperature.
- Humidity.
- Road Conditions.
- Traffic Conditions.
- Motion dynamics.

Each check is modeled as an independent transition path, enabling simultaneous execution. This ensures timely identification of multiple hazards without requiring sequential evaluation.

Token Synchronization

While each safety check operates independently, the results must be synchronized to maintain consistency. This is accomplished via a central transition, LogData Transition, which collects the output of all checks. This transition is only enabled once all relevant checks have been completed, ensuring that:

- No premature logging occurs.
- All available data is consolidated into a single log entry.

Synchronization guarantees that the system logs the complete context of a ride event, which is essential for diagnostics and performance analysis.

Modularity

The net's structure is modular, with each environmental, road condition or behavior factor encapsulated in its own subnetwork:

- Each check (e.g., CheckTempSafe/Danger Transition, CheckHumiditySafe/Danger Transition, CheckRoadSafe/Danger Transition, CheckTrafficSafe/Danger, CheckMotion/Danger) is a self-contained logic unit.

- Additional modules (e.g., for vibration monitoring) can be added without altering the existing structure.

Modularity enhances readability, reusability, and scalability of the net, making it suitable for expansion as more sensor types or decision rules are integrated.

Reactive System Design

The IOPT net behaves as a reactive system:

- Trigger: Starts when motion is detected (MotionDetected Transition).

- Response: Collects data, evaluates safety, issues alerts, and logs the result.

- Start State: Waits for the next trigger event.

This event-driven architecture aligns well with real-world use in embedded systems, where sensor events initiate specific control workflows.

Real-Time Signal Integration

Input signals to the net represent real-time data from hardware components such as:

- DHT11: for temperature and humidity.

- MPU6050: for motion detection.

- GPS modules or external APIs: for road and traffic conditions.

Using IOPT's ability to directly associate input places with signals and output transitions with actuators or logs ensures seamless communication with physical hardware or simulation environments

Thus, the use of IOPT Nets in this system provides a robust framework for modeling real-time, reactive behavior in a modular and parallel fashion. The combination of guards, actions, parallel execution, and synchronization results in a logically consistent and operationally effective safety monitoring system. These concepts form the foundation for the intelligent assessment of motorcycle riding conditions and demonstrate the suitability of IOPT for cyber-physical and IoT-enabled applications [Per+22].

Capítulo 4

Model Construction

4.1 Model Diagram and Description

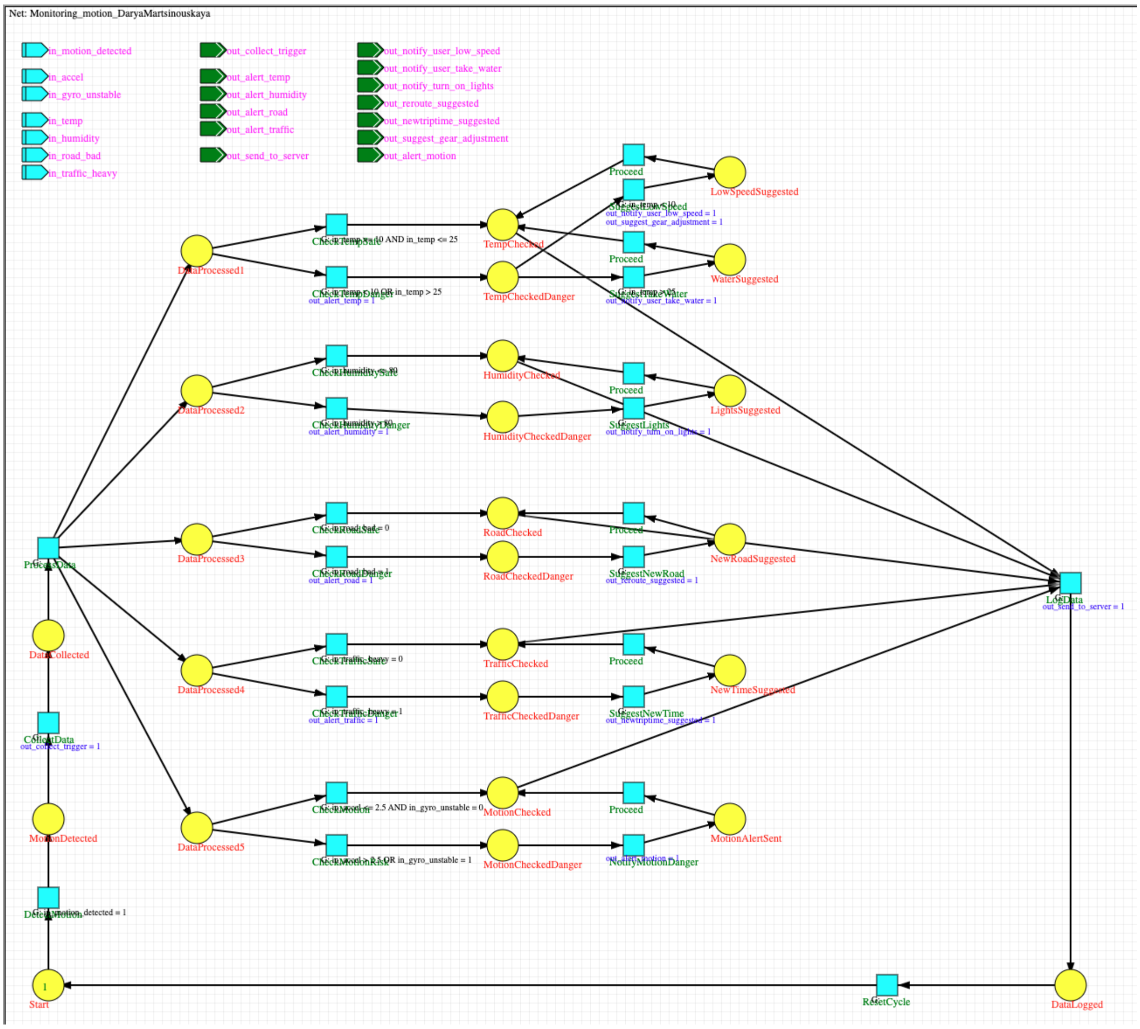

Figura 4.1: Motorcycle motion monitoring IOPT Model.

The designed system as shown on Figure 4.1 uses IOPT (Input/Output Place-Transition) Petri net concepts to model the behavior of an intelligent monitoring system for motorcycles. This system observes environmental parameters, road and traffic conditions, as well as driver behavior in order to analyze risks and provide real-time recommendations. The model structure utilizes places, transitions, guard conditions, output actions to represent and control the system's logic.

Initialization and Motion Detection:

The model begins at the Start Place, initialized with a single token. When motion is detected ("in motion detected = 1"), the DetectMotion Transition fires and places a token in MotionDetected Place. Then, CollectData Transition is triggered, initiating data collection through the "out collect trigger = 1"output action.

Data Processing:

After data is collected (DataCollected Place), the ProcessData Transition distributes the token into five parallel branches, each responsible for analyzing a specific input parameter.

1. Ambient Temperature:

- Safe Temperature: If the "in temp"input signal is between 10°C and 25°C, the temperature is considered safe for riding. Thus, CheckTempSafe Transition fires and places a token in TempChecked Place, which then leads to data logging via LogData Transition.

- Dangerous Temperature: If the temperature is outside this range, CheckTempDanger Transition fires and triggers the "out alert temp = 1"output action, indicating potentially hazardous conditions. A token is placed in TempCheckedDanger Place. Two scenarios follow based on temperature value:

a) Temperature is low (the "in temp"input signal is less than 10°C): SuggestLowSpeed Transition is triggered sending the "out notify user low speed = 1"and the "out suggest gear adjustment = 1"output actions, advising the rider to reduce speed and wear warmer protective gear.

b) Temperature is high (the "in temp"input signal is more than 25°C): SuggestTakeWater Transition fires with the "out notify user take water = 1"output action, recommending the rider to carry water.

2. Humidity:

- Safe Humidity: If the "in humidity"input signal is less than or equal to 80 percents, humidity is considered safe, thus, CheckHumiditySafe Transition leads to data logging.

- High Humidity: If humidity exceeds 80 percents, CheckHumidityDanger triggers the "out alert humidity = 1"output action. Then, SuggestLights Transition is fired with the "out notify user turn on lights = 1"output action, recommending the use of low-beam headlights.

3. Road Condition:

- Good Road Conditions: If the "in road bad"input signal is equal to 0, road is considered safe. CheckRoadSafe Transition is executed, and data is logged.

- Poor Road Condition: If the "in road bad"input signal is equal to 1, road is considered dangerous, thus, CheckRoadDanger Transition is activated with the "out alert road = 1"output action. Then, SuggestNewRoad Transition proposes an alternative safer route using the "out reroute suggested"output action equal to 1.

4. Traffic Conditions:

- Low Traffic: If the "in traffic heavy"input signal is equal to 0, traffic is considered normal, thus CheckTrafficSafe Transition leads to data logging.

- Heavy Traffic: If the "in traffic heavy"input signal is equal to 1, traffic is considered to be heavy and dangerous for motorcycle riding, thus, CheckTrafficDanger Transition is triggered with the "out alert traffic"output action equal to 1. Then, SuggestNewTime Transition is fired with the "out new trip time suggested"output action equal to 1, suggesting a better departure time.

5. Motion and Driving Behavior:

- Safe Behavior: If the "in accel"input signal is equal to to less then 2.5 and the "in gyro unstable"input signal is equal to 0, motorcycle motion is considered safe and CheckMotion Transition leads to data logging.

- Risky Motion: If exceeds the threshold or gyroscope readings indicate instability, CheckMotionRisk Transition is triggered. Then, NotifyMotionDanger Transition sends the "out alert motion"output action to warn the rider about potentially dangerous maneuvers.

Synchronization and Cycle Reset:

All parallel flows merge into the LogData Transition, which logs the gathered data to the server via the "out send to server"output action. Finally, ResetCycle Transition fires and returns the token to the Start Place, completing the monitoring cycle.

4.2 Input and Output Signals

The IOPT Petri Net model developed in this project simulates a reactive monitoring system for motorcycle safety. It operates based on a defined set of input and output signals, which imitate real-world sensor readings and trigger system actions in response to detected conditions.

Input Signals

The model receives multiple input signals that represent the data collected from various environmental and situational sensors:

- in motion detected

This signal simulates the detection of motion or ignition. It acts as the system trigger, initiating the data collection and evaluation process.

- in accel

A numerical input (range: 0–5) representing acceleration values measured by the MPU6050 accelerometer. High acceleration may indicate abrupt movement or aggressive riding behavior.

- in gyro unstable

A boolean input indicating gyroscope instability based on readings from the MPU6050 gyroscope. Instability may reflect erratic movement, leaning, or potential balance issues.

- in temp

Represents the ambient temperature reading obtained from, for example, the DHT11 sensor. This value is evaluated against predefined thresholds (e.g., 10°C to 25°C) to determine if the riding temperature is within a safe range.

- in humidity

Simulates the humidity reading from, for example, the DHT11 sensor. Humidity values are checked to identify potential riding discomfort or safety issues related to high (more than 80 percents) humidity.

- in road bad

A boolean signal that indicates poor road conditions. This may reflect potholes, uneven surfaces, or unpaved roads, typically derived from GPS-based data or manually reported inputs.

- in traffic heavy

A boolean input indicating the presence of heavy traffic conditions, usually gathered from map services or external traffic monitoring systems.

These input signals are used as guard conditions within transitions to control whether a particular transition can fire, allowing the system to selectively react based on the incoming data.

Output Signals

The model produces a set of output signals to notify the system or rider about detected hazards or statuses. These signals are triggered as a result of output actions defined in transitions:

- out collect trigger

Signals the successful completion of data collection after motion has been detected. It marks the beginning of the evaluation phase.

- out alert temp

Activated when the ambient temperature is outside the safe operational range. This alert can be used to notify the rider about extreme temperatures that may impact comfort or safety.

- out alert humidity

Triggered if the humidity level is considered potentially harmful or uncomfortable for riding, such as excessive moisture which may lead to fogging or reduced visibility.

- out alert road

Raised when bad road conditions are detected. This output is critical for alerting riders to potentially dangerous terrain.

- out alert traffic

Activated when the traffic conditions are classified as heavy. This allows the system to notify the rider of potential delays or increased risk in congested areas.

- out send to server

Sends the full set of evaluated data to a remote server for logging or further analysis. Supports centralized data storage and integration with other smart systems.

- out notify user low speed

Informs the rider to reduce speed based on detected hazardous conditions such as poor visibility.

- out notify user take water

A proactive health recommendation to the rider, typically triggered under hot weather, reminding them to stay hydrated.

- out notify user turn on lights

Suggests turning on lights in low-visibility scenarios, such as during fog, rain, enhancing rider visibility.

- out reroute suggested

Indicates that the system recommends an alternative route due to detected dangerous road conditions.

- out new trip time

Suggests a revised departure time to avoid peak traffic, enhancing travel efficiency and safety.

- out suggest gear adjustment

Based on temperature data, advises the rider to adjust equipment.

- out alert motion

Triggered when acceleration or gyroscope data indicates unsafe motion (e.g., in accel > 2.5 or in gyro unstable $= 1$). It reflects unstable or abrupt rider behavior and potential danger.

Thus, upon detecting motion (in motion detected), the system collects and processes input data from all sensors. The model checks each parameter (temperature, humidity, road condition, traffic, motion stability) in parallel branches, enabling real-time evaluation. Depending on the outcomes, appropriate alerts are issued. Once all evaluations are completed, a synchronization transition merges the results and triggers out log triggered to ensure that a consistent snapshot of the system state is stored.

This reactive mechanism allows timely warnings and creates a historical data log for further analysis, promoting proactive rider safety and system traceability.

Capítulo 5

Model Simulation and Verification

5.1 Simulation Scenarios

To validate the behavior of the developed intelligent monitoring system modeled using the IOPT Petri net formalism, a set of simulation scenarios were created and executed using the IOPT Tool. These simulations aim to verify that the system correctly responds to various environmental and behavioral conditions, as defined in the model. The simulations were conducted in step-by-step mode, with specific input signal values applied and the resulting transitions, output actions, and final token distributions observed and analyzed.

5.1.1 Scenario 1: Safe Riding Conditions

This scenario tests the system's behavior under optimal riding conditions. All monitored parameters are within safe thresholds.

Input Values:

in motion detected = 1

in temp = 20

in humidity = 50

in road bad = 0

in traffic heavy = 0

in accel = 1.5

in gyro unstable = 0

Expected Transitions:

- DetectMotion, CollectData, ProcessData
- CheckTempSafe, CheckHumiditySafe, CheckRoadSafe, CheckTrafficSafe, CheckMotion
- LogData, ResetCycle

Expected Output Actions:

out collect trigger = 1

out send to server = 1

Result: The system performs a full cycle, confirming safe conditions (as shown on Figure 5.1), logging the data, and returning to the initial state. All safe condition transitions are verified as reachable and functioning correctly.

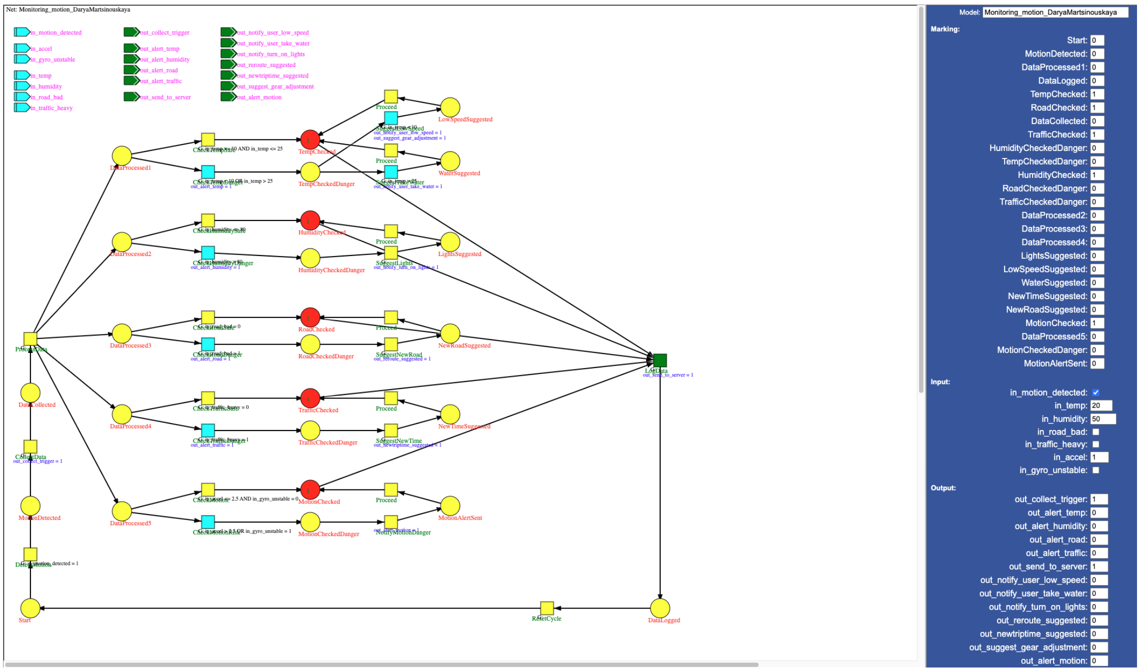

Figura 5.1: Scenario 1: Safe Riding Conditions

5.1.2 Scenario 2: Environmental Alerts (High Temperature and Humidity with Bad Road Conditions)

This scenario simulates an environment with multiple risk factors: high temperature, high humidity, and bad road conditions.

Input Values:

in motion detected = 1

in temp = 30

in humidity = 90

in road bad = 1

in traffic heavy = 0

in accel = 1.2

in gyro unstable = 0

Expected Transitions:

- DetectMotion, CollectData, ProcessData

- CheckTempDanger, SuggestTakeWater, Proceed

- CheckHumidityDanger, SuggestLights, Proceed

- CheckRoadDanger, SuggestNewRoad, Proceed
- CheckMotion (safe branch)
- LogData, ResetCycle

Expected Output Actions:

out collect trigger = 1

out alert temp = 1, out notify user take water = 1

out alert humidity = 1, out notify user turn on lights = 1

out alert road = 1, out reroute suggested = 1

out send to server = 1

Result: All risk-related transitions and notifications, as shown on Figure 5.2, were triggered correctly. The simulation confirms that the system responds appropriately to dangerous environmental conditions.

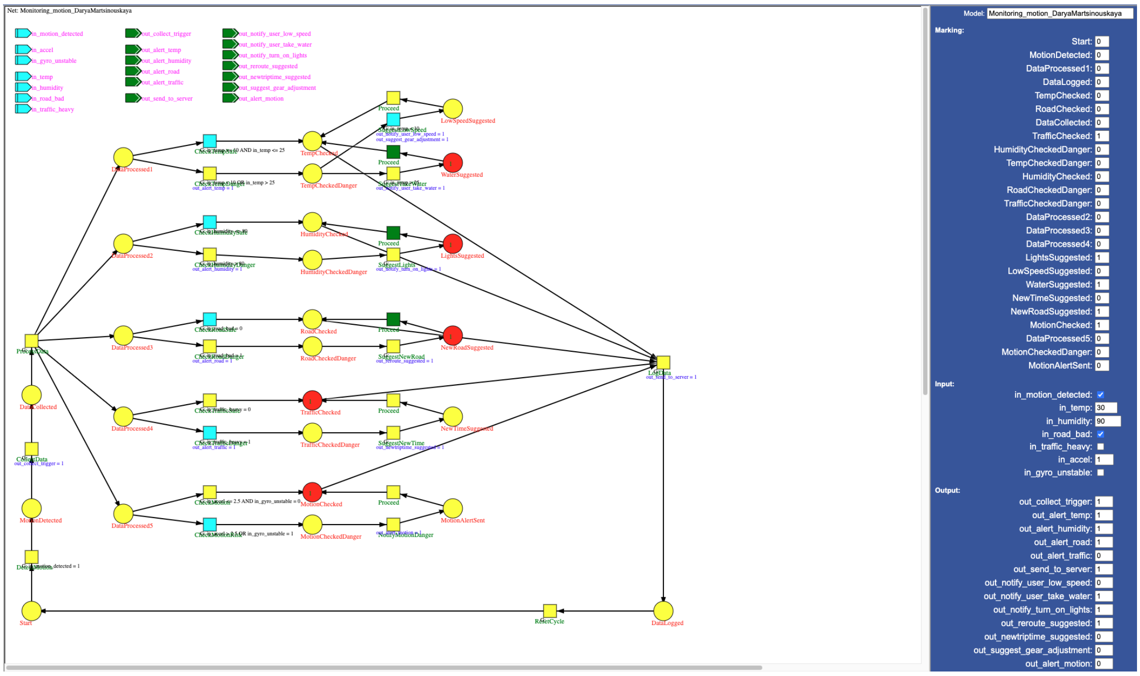

Figura 5.2: Scenario 2: Environmental Alerts (High Temperature and Humidity with Bad Road Conditions)

5.1.3 Scenario 3: Risky Driving Behavior with Environmental Alert (Low Temperature)

This scenario evaluates how the system reacts when dangerous driving behavior is detected through acceleration and gyroscope inputs as well as simulates an environment with one risk factors: low temperature.

Input Values:

in motion detected = 1

in temp = 0

in humidity = 60

in road bad = 0

in traffic heavy = 0

in accel = 3.0

in gyro unstable = 1

Expected Transitions:

- DetectMotion, CollectData, ProcessData
- CheckTempDanger, SuggestLowSpeed, Proceed
- CheckHumiditySafe, CheckRoadSafe, CheckTrafficSafe
- CheckMotionRisk, NotifyMotionDanger, Proceed
- LogData, ResetCycle

Expected Output Actions:

out collect trigger = 1

out alert temp = 1, out notify user low speed = 1, out suggest gear adjustment = 1

out alert motion = 1

out send to server = 1

Result: The model successfully detected risky motion and generated the appropriate warning output. This confirms the correct function of behavior monitoring logic. The simulation also confirms that the system responds appropriately to dangerous environmental condition such as low temperature. Results as shown on Figure 5.3.

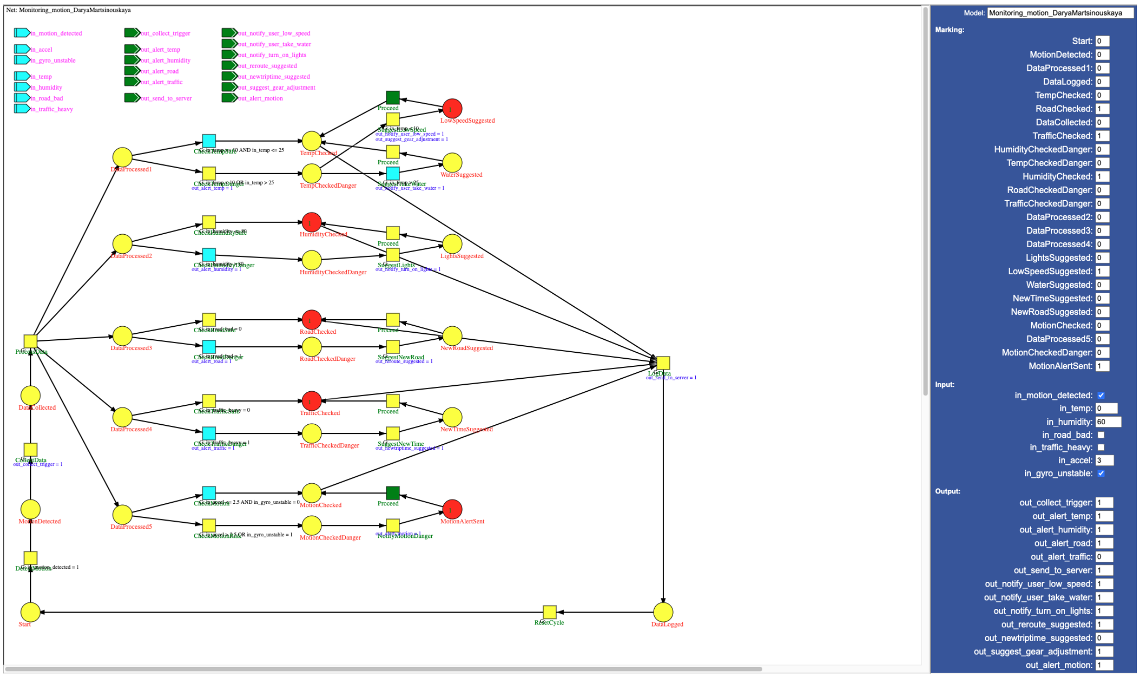

Figura 5.3: Scenario 3: Risky Driving Behavior with Environmental Alert (Low Temperature)

5.1.4 Scenario 4: Multiple Alerts Including Traffic and Motion

This complex scenario tests how the system handles combined risks of heavy traffic and dangerous rider behavior.

Input Values:

in motion detected = 1

in temp = 25

in humidity = 75

in road bad = 0

in traffic heavy = 1

in accel = 2.8

in gyro unstable = 1

Expected Transitions:

- DetectMotion, CollectData, ProcessData
- CheckTempSafe, CheckHumiditySafe, CheckRoadSafe (safe branches)
- CheckTrafficDanger, SuggestNewTime, Proceed
- CheckMotionRisk, NotifyMotionDanger, Proceed
- LogData, ResetCycle

Expected Output Actions:

out collect trigger = 1

out alert traffic = 1, out new trip time suggested = 1

out alert motion = 1

out send to server = 1

Result: The system reacted to both traffic and behavioral risks by triggering relevant notifications. As shown on Figure 5.4, the correct parallel branch behavior and synchronization at the logging transition were verified.

Thus, the simulation results confirm that the Petri net model behaves as expected under varying conditions. All transitions and output actions were reachable and correctly triggered according to the input values. The system returns to its initial state after each full monitoring cycle, ensuring continuous operation.

5.2 Reachability and Deadlock Analysis

To verify the correctness and robustness of the designed Petri net model, a formal reachability and deadlock analysis was performed using the IOPT Tool. The goal of this analysis is to ensure that:

- All critical states of the system are reachable under valid input conditions.
- There are no deadlocks, i.e., situations where no transitions can be fired and the system gets stuck.
- The model correctly returns to its initial state after each complete monitoring cycle.

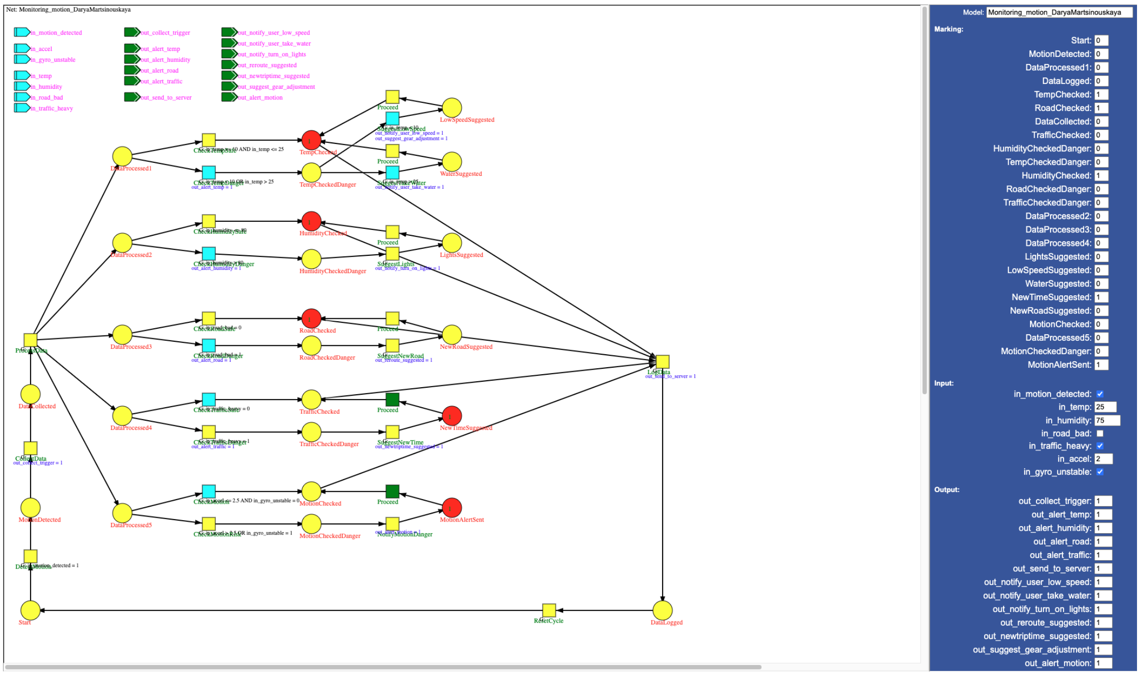

Figura 5.4: Scenario 4: Multiple Alerts Including Traffic and Motion

5.2.1 Reachability Analysis

Reachability analysis involves determining whether a specific marking (i.e., a specific distribution of tokens across places) can be achieved from the initial marking through a sequence of valid transitions.

Key reachability checks performed:

Motion Detected State: When in motion detected = 1 is provided as input, the token successfully moves from Start to MotionDetected, indicating the system can detect movement and begin its cycle.

Environmental Branches: For each environmental parameter (temperature, humidity, road condition, traffic), both safe and dangerous paths were triggered through conditional inputs. This verified the reachability of all the following places:

TempChecked, TempCheckedDanger, WaterSuggested, LowSpeedSuggested.

HumidityChecked, HumidityCheckedDanger, LightsSuggested.

RoadChecked, RoadCheckedDanger, NewRoadSuggested.

TrafficChecked, TrafficCheckedDanger, NewTimeSuggested.

Behavior Monitoring: Tokens reached either MotionChecked or MotionCheckedDanger together with MotionAlertSent paths depending on acceleration and gyroscope input signals.

Final States: All scenarios eventually led to the DataLogged Place and then to the Start place after ResetCycle Transition, proving that the model always returns to the initial marking.

No unreachable places or transitions were found during simulation and step-by-step analysis, confirming that the system is fully navigable under valid inputs.

5.2.2 Deadlock Analysis

A deadlock in a Petri net occurs when no transitions are enabled and the system cannot proceed further. To verify the absence of deadlocks:

- All simulation scenarios were examined for the possibility of ending in a marking where no transitions are enabled.

- The model structure includes a complete monitoring loop, ending with the ResetCycle transition that returns a token to the Start place. This ensures cyclic operation and re-enablement of the initial transition (DetectMotion) for the next round.

- Branch transitions include proper guard conditions based on inputs. If no data is collected (CollectData is not fired), analysis transitions are never reached, which is the intended behavior.

Additionally, the IOPT Tool's internal deadlock detection mechanism was used to confirm that:

- No deadlock markings were encountered in the reachability graph.

- For every reachable marking, at least one transition is enabled or can become enabled under proper input conditions.

To sum up, the model satisfies both liveness and safety properties:

- It is live, as all transitions can eventually be fired under the right conditions and the cycle repeats indefinitely.

- It is deadlock-free, as confirmed by reachability graph inspection and step-by-step simulations.

This analysis demonstrates that the IOPT Petri net model is structurally correct, functionally complete, and suitable for real-time monitoring tasks in motorcycle safety systems.

5.2.3 State-Space Analysis

To evaluate the complexity and scalability of our model, a full state-space generation, an automatic engine of IOPT Tool that builds the full reachability graph of the system, capturing every possible scenario that could arise from the defined transitions and inputs [Per+22], was executed.

Figure 5.5 illustrates initial settings for analysis used (one token is placed into Start State).

Figures 5.6 and 5.7 contain analysis results.

Summing up, the following results were obtained:

Total reachable states: 43,050

Total transition links: 1,382,800

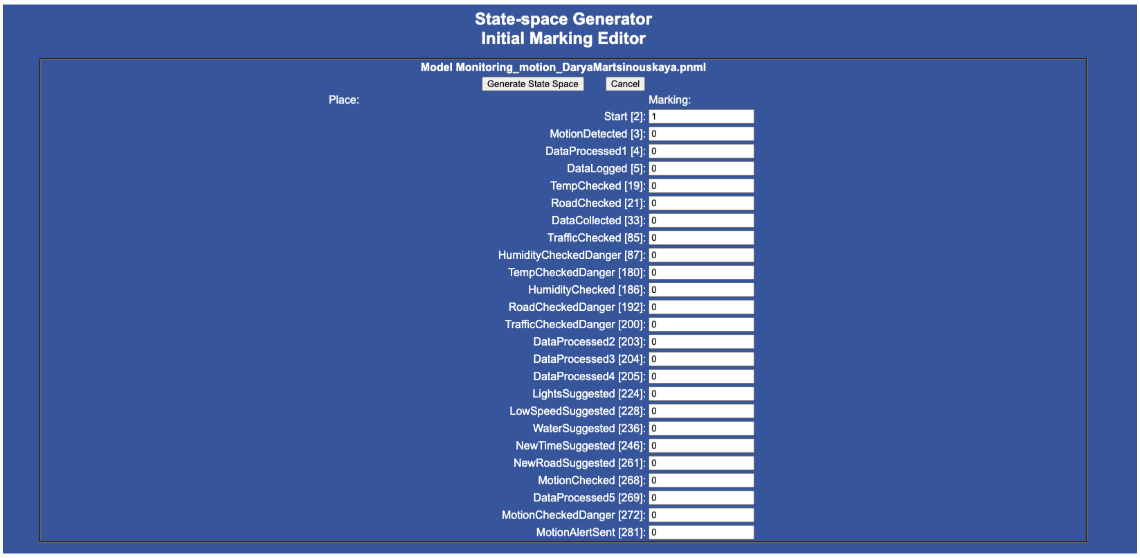

Figura 5.5: State-Space Analysis - Initial Set up

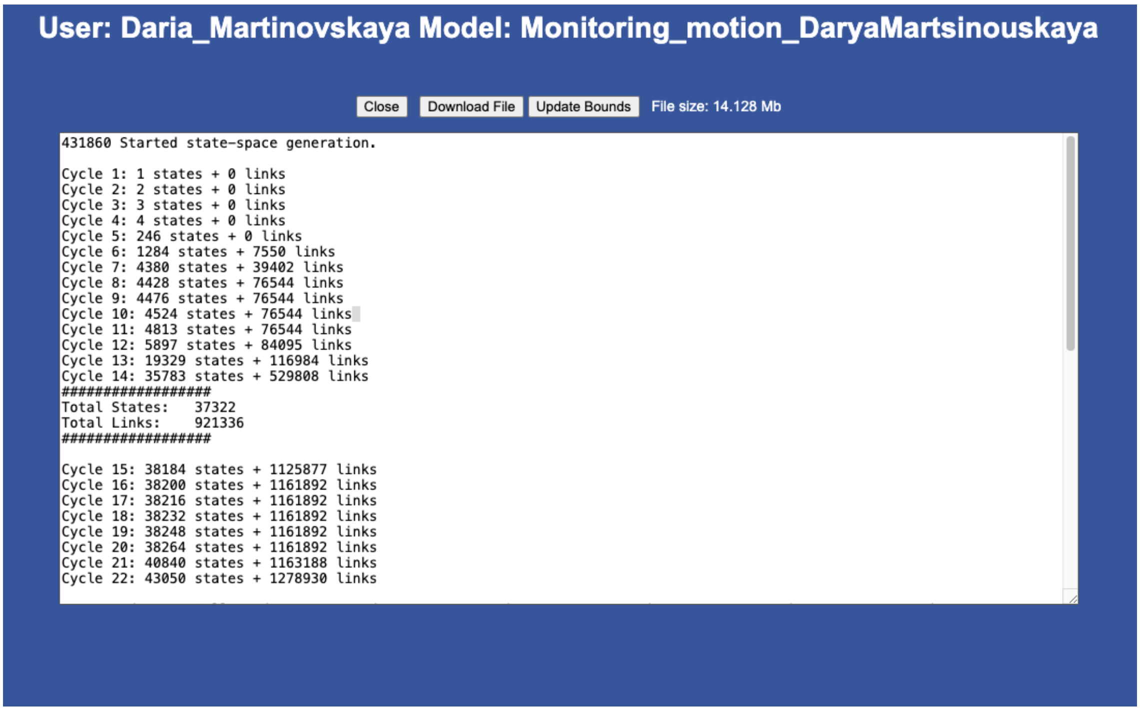

Figura 5.6: State-Space Analysis - Results (Part 1)

Total conflicts: 36,511

Deadlocks found: 0

Invalid markings: 0

These values confirm that:

- The model exhibits rich dynamic behavior and supports high concurrency due to the presence of multiple branches executing in parallel.

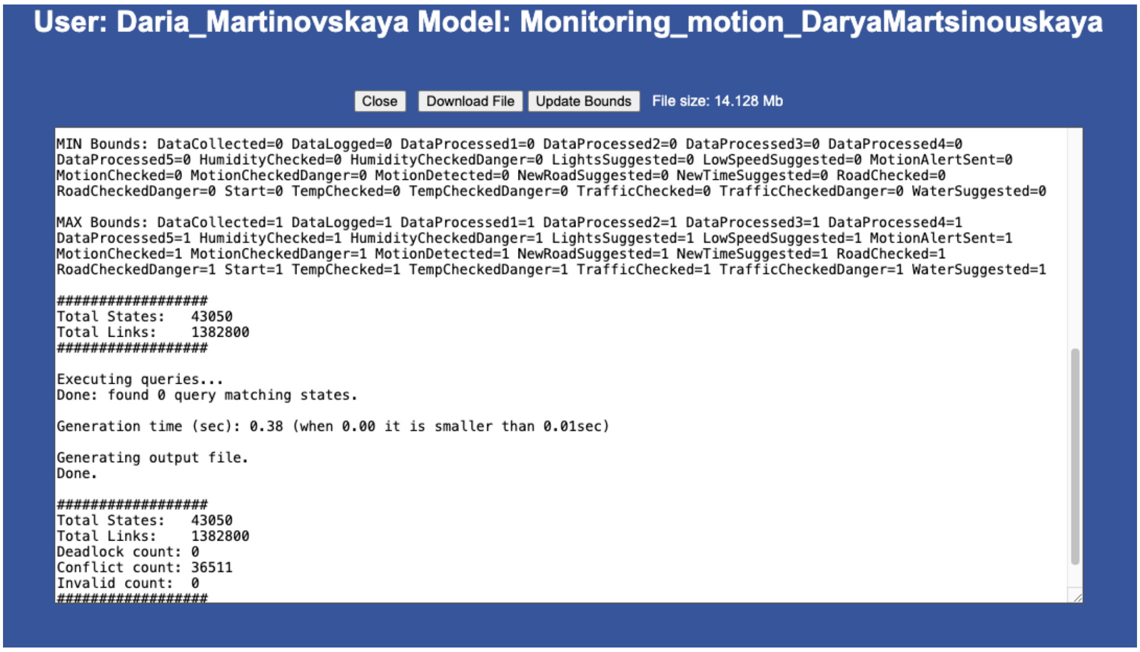

Figura 5.7: State-Space Analysis - Results (Part 2)

- The absence of deadlocks across all reachable states reinforces the earlier deadlock analysis, confirming that the system is capable of continuous operation.

- The high number of conflicts (36,511) corresponds to valid design behavior, where multiple transitions are enabled concurrently due to independent sensor branches, allowing flexible responses to real-time events.

- The model finishes each operational cycle by returning a token to the initial place (Start), enabling new monitoring cycles, which validates both structural liveness and reset capability.

These results quantitatively validate the conclusions from the simulation-based reachability and deadlock analysis, while also providing insight into the internal complexity and robustness of the system under all possible execution paths.

Capítulo 6

Results and Discussion

The results of the simulation and structural verification reveal that the IOPT-based model effectively supports the design and analysis of a safety-enhancing system for motorcycle riding. The system's modular structure—with five parallel analysis branches—enables real-time and independent evaluation of multiple risk factors, which closely resembles the way real-world systems should handle concurrent sensor input.

One of the main strengths of this model is its clear logic and readable structure. Each transition is associated with a specific environmental or behavioral condition, and output actions are logically derived from guarded transitions. This clarity not only improves maintainability but also makes it easier to verify the correctness of the model.

Another strength lies in the model's scalability. Additional safety rules or environmental checks could be easily incorporated by adding new branches or transitions without disrupting the existing structure. For example, weather forecast data or road surface analysis via external APIs could be integrated in future extensions.

However, some limitations are acknowledged. The model operates based on discrete inputs and does not incorporate real-time execution or dynamic data sampling. It also assumes accurate and noise-free sensor inputs, which may not reflect the reality of on-road conditions. Moreover, user feedback and system adaptation based on rider history are not currently part of the model but could significantly enhance personalized risk evaluation.

In terms of applicability, the model was designed with potential hardware implementation in mind. It could be deployed using microcontrollers such as the ESP32, connected to sensors like accelerometers, temperature/humidity modules, GPS as well as road condition detectors. The output actions could then be transformed into real-world notifications via a mobile app or dashboard interface.

Overall, the IOPT model demonstrates a valid, structured, and efficient approach to modeling a smart motorcycle monitoring system. It highlights how formal modeling tools can contribute to the development of reliable, safety-critical systems and opens possibilities for future enhancements and real-world prototyping.

Capítulo 7

Conclusion

This report presented the design, implementation, and verification of an intelligent motorcycle monitoring system modeled using the IOPT tools. The objective was to develop a modular and scalable system capable of analyzing multiple factors that affect motorcycle safety, including environmental conditions, road quality, traffic levels, and rider behavior.

The model construction focused on separating each functional component into parallel branches, enabling concurrent analysis of diverse sensor inputs. Input and output signals were carefully defined to reflect real-world interactions, such as temperature alerts or traffic notifications. Simulation scenarios demonstrated that the model responds correctly to both normal and hazardous conditions, providing the appropriate alerts or actions in each case.

Formal verification techniques, including reachability and deadlock analysis as well as state-space analysis provided by IOPT Tools functionality, validated the model's correctness and robustness. All intended states were reachable, and no deadlocks were detected, confirming that the model maintains logical consistency throughout execution cycles.

The results show that the IOPT tools is highly effective for modeling safety-critical systems in the context of Internet of Things applications. The model developed is not only logically sound but also adaptable for future expansions or hardware implementations. Despite some limitations—such as the absence of real-time data handling and adaptive feedback—the system offers a solid foundation for further development.

In conclusion, this project demonstrates the practical use of Petri nets and the IOPT tools in designing intelligent systems that enhance road safety. It also highlights the potential of formal methods in supporting reliable system development in the domain of smart transportation.

Bibliografia

[AV24] M. H. Yousaf A. Akhtar R. Ahmen e S. A. Velastin. «Real-Time Motorbike Detection: AI on the Edge Perspective». Em: *Mathematics* (2024). URL: `https://doi.org/10.3390/math12071103` (ver p. 2).

[Dar22] Y.-A. Daraghmi. «Vehicle Speed Monitoring System Based on Edge Computing». Em: *Sensors* (2022). URL: `https://doi.org/10.1109/ICPET53277.2021.00009` (ver p. 2).

[EY24] T-H Yu E. Kristiani e Ch-T Yang. «On Construction of Real-Time Monitoring System for Sport Cruiser Motorcycles Using NB-IoT and Multi-Sensors». Em: *Sensors* (2024). URL: `https://doi.org/10.3390/s24237484` (ver p. 2).

[F P14] L. Gomes F. Pereira F. Moutinho. *IOPT Tools User Manual - Version 1.1.* GRES Research Group, 2014 (ver p. 5).

[Fas24] FasterCapital. «Motorcycle Internet of Things: Revolutionizing the Motorcycle Industry: How IoT is Driving Innovation». Em: *FasterCapital* (2024). URL: `https://fastercapital.com/content/Motorcycle-Internet-of-Things--Revolutionizing-the-Motorcycle-Industry--How-IoT-is-Driving-Innovation.html#What-is-Motorcycle-IoT-and-Why-is-it-Important-` (ver p. 1).

[Har24] Harley-Davidson. «HARLEY-DAVIDSON® RDRS SAFETY ENHANCEMENTS DESIGNED TO BOOST MOTORCYCLE RIDER CONFIDENCE». Em: *Harley-Davidson* (2024). URL: `https://h-dmediakit.com/eu/news-articles/harley-davidson%C2%AE-rdrs-safety-enhancements-designed-to-boost-motorcycle-rider-confidence.html` (ver p. 1).

[I Z23] P. Papcun I. Zolotova K. Micko. «Review of IoT Sensor Systems Used for Monitoring the Road Infrastructure». Em: *Sensors* (2023). URL: `https://doi.org/10.3390/s23094469` (ver p. 2).

[JM24] R. Rahman J. e K. Manoj. «Motorcycle Crash Detection And Alert System Using Iot». Em: *International Journal of Creative Research Thoughts (IJCRT)* (2024). URL: `https://ijcrt.org/papers/IJCRT2401295.pdf` (ver p. 2).

[Kaw24] Kawasaki. «Ninja H2 SX SE: overview». Em: *Kawasaki* (2024). URL: `https://www.kawasaki.eu/en/Motorcycles/Sport_Tourer/ninja_h2_sx_se_2025.html#:~:text=Using%20a%20rear%2Dfacing%20radar,in%20the%20rider's%20blind%20spot.` (ver p. 1).

[Muk+24] S. Mukhopadhyay et al. «A Review and Analysis of IoT Enabled Smart Transportation Using Machine Learning Techniques». Em: *International Journal of Transport Development and Integration* 8.1 (mar. de 2024), pp. 61–77. DOI: `10.18280/ijtdi.080106`. URL: `https://www.iieta.org/journals/ijtdi/paper/10.18280/ijtdi.080106` (ver p. 2).

[Per+22] Filipe Pereira et al. «IOPT-Tools – From Executable Models to Automatic Code Generation for Embedded Controllers Development». Em: *Applications and Theory of Petri Nets and Concurrency*. Vol. 13291. Lecture Notes in Computer Science. Springer, 2022, pp. 105–124. DOI: `10.1007/978-3-031-06653-5_7`. URL: `https://link.springer.com/chapter/10.1007/978-3-031-06653-5_7` (ver pp. 5, 9, 23).

[TY24] E. Kristiani T-H Yu e Ch-T Yang. «On Construction of Real-Time Monitoring System for Sport Cruiser Motorcycles Using NB-IoT and Multi-Sensors». Em: *Sensors* (2024). URL: `https://doi.org/10.3390/s24237484.` (ver p. 1).

[Zai+24] M. Zaid et al. «IoT based Truck Wheel Temperature Monitoring System». Em: *2024 Asian Conference on Intelligent Technologies (ACOIT)*. IEEE, 2024. DOI: `10.1109/ACOIT60719.2024.10939186`. URL: `https://ieeexplore.ieee.org/document/10939186` (ver p. 2).