

# PHYSICS-AWARE IOT SIMULATION FRAMEWORK FOR REAL-TIME MONITORING AND CONTROL OF ELECTRONIC LABORATORY SYSTEM

H. Delibašić-Marković<sup>1\*</sup>, V. Petrović<sup>1</sup>, I. Petrović<sup>2</sup>

<sup>1</sup> University of Kragujevac,  
Faculty of Science,  
Radoja Domanovića 12, 34000 Kragujevac, SERBIA

<sup>2</sup>Academy of Professional Studies Šumadija,  
Department in Kragujevac,  
Kosovska 8, 34000 Kragujevac, SERBIA  
\*e-mail: hristina.delibasic@pmf.kg.ac.rs

The study presents a physics-aware simulation framework developed in Python to emulate Internet of Things (IoT)-based monitoring systems for electronic laboratory environments. The framework generates synthetic sensor data for temperature, humidity, and light intensity, representing environmental parameters that often influence the performance, stability, and safety of laboratory electronics and optoelectronic components. By implementing programmable threshold logic and real-time alert dispatch through RESTful APIs, the system enables automated detection of parameter exceedances and notification delivery without the need for physical hardware. The simulation supports both batch and real-time modes, allowing flexible emulation of thermal fluctuations, ambient light shifts, and humidity drift in controlled settings. Application scenarios include overheating prevention in laser-driven devices, monitoring of ambient light in optical benches, and detecting environmental drift in sensitive experiments such as photoemission or cryogenic setups. The results demonstrate reliable threshold crossing detection with response times under 2 seconds and support multivariate data visualisation to identify compound stress conditions. This framework is particularly suited for early-stage prototyping, instructional use, and safety validation in physics-oriented laboratories. By combining domain-specific sensor modelling with IoT-based control logic, the proposed method offers a low-cost and extensible backend for responsive laboratory monitoring and smart alert systems.

**Keywords:** *Electronic laboratory systems, Internet of Things (IoT), physics-based monitoring, Python, real-time alerting, sensor simulation.*

## 1. INTRODUCTION

---

The proliferation of Internet of Things (IoT) technologies has significantly transformed various domains, including residential, industrial, and scientific environments. In experimental physics and electronic laboratories, IoT systems increasingly support environmental monitoring, data acquisition, and automated control. These systems typically consist of interconnected sensors and microcontrollers that capture and transmit data for parameters such as temperature, humidity, luminosity, air quality, and motion, among others, forming the foundation for intelligent decision-making [2]. However, the initial development, prototyping, and educational exploration of IoT applications often face limitations due to the costs, complexities, and logistical issues associated with obtaining and deploying physical hardware. Moreover, physical sensor systems often require specialised knowledge, maintenance, and considerable resources for long-term operation [3]. To address these challenges, software-based IoT simulation frameworks have emerged as essential tools, providing efficient and cost-effective alternatives for preliminary testing and validation [4]. Such simulation platforms enable developers and researchers to rapidly evaluate IoT system designs, communication protocols, data analytics methods, and event-detection algorithms without the constraints and overhead of managing physical devices [5]. From the perspective of experimental physics, these tools also allow safe and repeatable testing of monitoring logic, alarm systems, and environmental responses in virtual laboratory conditions. This is particularly valuable in fields where thermal drift, humidity control, and light intensity directly impact the performance of sensitive electronics

or optical instrumentation. Furthermore, software-based simulations offer controlled environments for educational purposes, allowing students and trainees to gain practical experience in IoT technologies and system integration without significant investment in hardware. Python, a widely adopted programming language, has become particularly popular in the IoT domain due to its readability, extensive library, and ease of integration with external web services [6]. It supports efficient data analysis, visualisation, and rapid prototyping, making it suitable for building scalable backend applications for IoT systems. Numerous libraries, such as Pandas, NumPy, Matplotlib, and Requests, enable developers to perform advanced data processing, visualisation, and seamless integration with RESTful web APIs [7].

A crucial functionality of many IoT systems is real-time event detection, where sensor data streams are continuously monitored to trigger alerts or automated responses when certain conditions or thresholds are met [8]. Traditional IoT implementations commonly rely on physical sensor nodes and middleware platforms to manage these functionalities. However, fully software-driven approaches remain relatively underexplored, despite their potential for simplifying early-stage development, educational activities, and scenario analysis. The present paper introduces a fully software-based framework that simulates sensor data streams such as temperature, humidity, and luminosity through configurable data generators. The simulated data are continuously evaluated against predefined thresholds using Python-based logic, and upon detecting threshold-breach events, notifications are automatically dispatched through

external web services such as Telegram and IFTTT APIs. This approach facilitates rapid prototyping, educational demonstrations, and preliminary validations in resource-limited environments without reliance on physical hardware. It also serves as a surrogate environment for testing safety logic and environmental control in physics-related electronic laboratories.

The remainder of this paper is structured as follows. Section 2 reviews relevant literature and related work on IoT simulation frameworks and Python-based applications.

## 2. BACKGROUND AND SCIENTIFIC FOUNDATIONS

---

The advancement of IoT simulation frameworks has resulted in a variety of software-based methodologies aimed at replicating sensor behaviour, testing event-detection logic, and validating control responses, often without requiring physical hardware. In physics-focused laboratories and electronic experimental systems, this trend has opened new possibilities for modelling environmental monitoring systems and feedback mechanisms under controlled and reproducible conditions.

Hybrid and hardware-in-the-loop (HIL) systems, originally developed in structural engineering, have since been applied in laboratory automation and embedded electronics for real-time testing and simulation. Foundational work by Takanashi et al. [9] and further refinements in numerical stability and control delay compensation [10]–[13] highlight the importance of integrated simulation-feedback environments. Recent studies in laboratory robotics and radiation-prone environments [21], [22] adapted HIL methods for sensor calibration and control testing, though such systems require complex infrastructure and are not easily scalable for instructional or early-stage

development. Lightweight, software-only emulators such as NS-3 and CupCarbon [15], [16] are commonly used to simulate wireless sensor networks (WSNs), focusing largely on communication protocols. More recent efforts include virtual sensors in cloud-based platforms [17], and digital twin systems for laboratory automation [23], [24]. These frameworks, while powerful, often require access to real sensor nodes, cloud APIs, or predefined hardware templates, and tend to prioritize infrastructure modelling over fine-grained sensor emulation or application-layer threshold logic. Emerging platforms also emphasise container-based deployment and federated control architectures, as seen in Hamzaoui et al. [25] and Morcillo-Jimenez et al. [26], but are typically aimed at smart-city or industrial IoT rather than physics-oriented laboratory use cases. Meanwhile, Python has emerged as a leading language for scientific and embedded computing due to its readability, modularity, and strong ecosystem of libraries such as NumPy, pandas, matplotlib, and requests. Although Python-based frameworks are widely used in smart home, biomedical, and industrial IoT con-

texts, their application in laboratory physics, where environmental fluctuations can affect optoelectronic behaviour or measurement precision, remains relatively underexplored. Scientific projects like PySCF [18] demonstrate Python’s capacity for complex system modelling, while recent studies emphasise its effectiveness for constructing interactive sensor emulators and automated decision systems [27], [28].

Despite the availability of domain-specific tools for communication-layer emulation or digital twin creation, there remains

a noticeable lack of compact, open-source platforms that integrate real-time sensor simulation, event-driven alerting, and RESTful output dispatch in a single, hardware-independent architecture. This paper addresses that gap by proposing a modular Python-based framework suitable for environmental monitoring tasks in electronic laboratories and physics-oriented experimental setups, with applications in overheating detection, light fluctuation monitoring, and humidity drift analysis.

### 3. SYSTEM ARCHITECTURE

---

The framework proposed in this study is designed to emulate core sensing, evaluation, and alerting functionalities commonly found in IoT systems, relying exclusively on software components implemented in Python. Its structure reflects typical workflows in experimental physics laboratories, where sensitive equipment and samples must be maintained within well-defined environmental limits to avoid systematic errors, signal distortion, or even physical damage. The architecture consists of four sequential stages: generation of synthetic observables, threshold-based evaluation, event-triggered response, and data logging and visualisation.

At the first stage, virtual sensors produce synthetic time-dependent data streams for temperature,  $T$ , illumination,  $L$ , and humidity,  $H$ . These are key environmental variables in a wide range of physical experiments. For example, temperature fluctuations can lead to expansion or contraction of optical mounts, drift of laser cavity modes, or variation of semiconductor band gaps. Illumination levels directly affect photodiodes, charge-coupled devices (CCDs), and other light-sensitive detectors,

where insufficient or fluctuating light intensity introduces noise or measurement bias. Humidity changes can alter dielectric properties of gases, affect charge accumulation on insulating surfaces, or influence plasma initiation thresholds. In this framework, NumPy-based deterministic and stochastic functions replicate such physical variability: sinusoidal terms mimic daily thermal cycles, abrupt steps reproduce transient disturbances, and slow drifts emulate cumulative processes such as evaporation.

In the second stage, the threshold evaluation engine represents the monitoring of physical conditions relative to predefined critical limits ( $T_t$ ,  $L_t$ ,  $H_t$ ). These limits have direct physical meaning:  $T_t$  can be set at the maximum safe operating temperature of electronic components,  $L_t$  corresponds to the minimum illumination required to maintain detector linearity, while  $H_t$  defines the humidity above which condensation or dielectric breakdown risks arise. Whenever measured values satisfy conditions such as  $T > T_t$ ,  $L > L_t$ , or  $H > H_t$ , the system registers a critical event. The rule engine, implemented as Python conditional logic, thus emulates the supervisory role of inter-

locks and feedback systems commonly found in high-precision experiments.

The third stage is the decision layer. If any threshold is violated, the system follows the YES branch and issues an alert. In experimental physics practice, this corresponds to protective responses, i.e., activating shutters to prevent optical damage, cooling systems to limit thermal drift, or alarms to warn the operator. In the framework, such alerts are dispatched via RESTful APIs, demonstrating integration with real-time notification services (e.g., Telegram, IFTTT). The NO branch corresponds to stable operation: values remain within acceptable bounds, and the system continues with passive monitoring and documentation.

Finally, the logging and visualisation stage ensures reproducibility, a cornerstone of physics. All simulated observables and triggered events are stored in structured CSV files using pandas, while matplotlib visualisations provide diagnostic plots showing the temporal relation between environmental variations and threshold crossings. This is analogous to laboratory practice, where systematic records and post-experiment analysis are critical for verifying reproducibility and interpreting anomalies.

Figure 1 summarises the framework:

synthetic signals are compared to critical thresholds ( $T_t$ ,  $L_t$ ,  $H_t$ ), and depending on the result, either trigger real-time notifications or feed into data storage and visualisation. In this way, the architecture encapsulates the sensing-decision-response loop characteristic of real laboratory systems, providing a physics-aware simulation environment for prototyping, safety studies, and educational demonstrations.

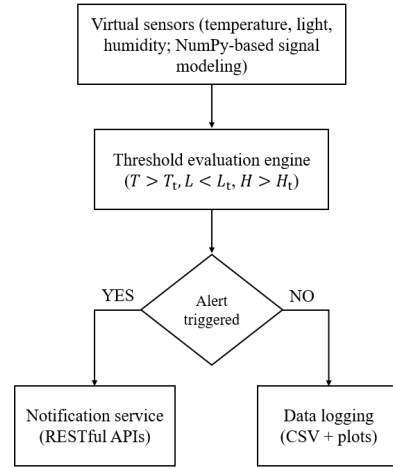


Fig. 1. Logical flow of the proposed physics-aware IoT simulation framework. Virtual sensors generate synthetic environmental data, which are evaluated by a Python-based threshold engine. If a threshold is exceeded (YES), an alert is triggered and dispatched via RESTful APIs; if not (NO), the data are stored and visualised for further analysis.

### 3. IMPLEMENTATION

The software framework has been implemented in Python as a modular simulation environment reproducing the sensing-evaluation-response cycle typically encountered in laboratory physics. The modules include synthetic signal generation, threshold evaluation, event classification, and systematic recording with visualisation. The implementation is physics-aware: each observable is modelled as

a dynamical process with well-defined statistical properties, and threshold conditions are treated as critical boundaries beyond which experimental stability or measurement validity is compromised.

The virtual sensor generation module reproduces three observables (see previous Section):  $T$ ,  $L$  and  $H$ . These quantities are central in laboratory environments, where thermal drift, fluctuations in illumination,

and uncontrolled humidity can strongly influence outcomes. For example, a slow increase in  $T$  may shift the refractive index of optical components,  $L$  fluctuations may degrade signal-to-noise ratios in detectors, and excessive  $H$  may lead to dielectric

breakdown or condensation on optical surfaces. To emulate such processes, a generalized sensor class was implemented, combining deterministic functions with autoregressive noise to capture correlated fluctuations:

```
import numpy as np

class VirtualSensor:
    def __init__(self, base, trend_func, noise_sigma=0.2, ar_coeff=0.7, step=1):
        self.base = base
        self.trend_func = trend_func
        self.noise_sigma = noise_sigma
        self.ar_coeff = ar_coeff
        self.prev_noise = 0
        self.t = 0
        self.step = step

    def read(self):
        deterministic = self.base + self.trend_func(self.t)
        # colored noise (AR(1))
        noise = self.ar_coeff * self.prev_noise + np.random.normal(0, self.noise_sigma)
        self.prev_noise = noise
        self.t += self.step
        return deterministic + noise
```

This implementation ensures that the simulated signals exhibit both slow deterministic trends and correlated stochastic components, reproducing features such as diurnal oscillations, step-like illumination drops, and gradual humidity decay.

The threshold evaluation engine compares observables with critical limits ( $T_t$ ,  $L_t$ ,  $H_t$ ) corresponding to physically mean-

ingful boundaries: the safe operational temperature of electronic components, the illumination required to maintain linear detector response, and the maximum humidity tolerable before condensation or breakdown. To reflect the fact that short-lived excursions are often irrelevant in practice, the evaluation logic incorporates persistence requirements, as shown in the following excerpt:

```
rules = [
    {"name": "Overheating", "condition": lambda T, L, H: T > Tt, "persistence": 5},
    {"name": "LowLight", "condition": lambda T, L, H: L < Lt, "persistence": 3},
    {"name": "HighHumidity", "condition": lambda T, L, H: H > Ht, "persistence": 10}
]

def evaluate_rules(T, L, H, history):
    events = []
    for rule in rules:
        condition_met = rule["condition"](T, L, H)
        if condition_met:
            history[rule["name"]] += 1
            if history[rule["name"]] >= rule["persistence"]:
                events.append(rule["name"])
        else:
            history[rule["name"]] = 0
    return events
```

This rule-based structure mirrors laboratory interlocks, where a condition must

persist beyond a finite duration  $\Delta t$  before being considered a genuine instability.

The event classification module creates structured records for each confirmed threshold crossing. Compound events (such as simultaneous overheating and low illumination) are identified and classified separately, since in physical setups such correla-

tions may indicate coupled processes (e.g., power supply failure leading to both heating and loss of illumination). The following implementation excerpt demonstrates how events are recorded with time stamps and annotated for compound behaviour:

```
import time

def classify_event(event_names, values):
    timestamp = time.strftime("%H:%M:%S")
    record = {
        "time": timestamp,
        "events": event_names,
        "values": values,
        "compound": len(event_names) > 1
    }
    log.append(record)
    return record
```

This procedure parallels experimental practice, where anomalous events are logged with precise timing information and correlations across multiple channels are interpreted as possible systemic faults.

Finally, the logging and visualisation module ensures that all data and events are archived for subsequent analysis. Beyond simple CSV export, the framework overlays threshold lines, marks event occurrences,

and computes basic statistics (e.g., mean excursions, event durations). Such diagnostic outputs parallel the inspection of oscilloscope traces or environmental monitors in experimental laboratories. The following excerpt shows how measurement traces are exported and plotted, providing visual diagnostics analogous to oscilloscope records or environmental monitors in physics laboratories:

```
import pandas as pd
import matplotlib.pyplot as plt

def export_and_plot(log, thresholds):
    df = pd.DataFrame(log)
    df.to_csv("sensor_log.csv", index=False)
    for var in ["temperature", "light", "humidity"]:
        plt.plot(df["time"], df[var], label=var)
        plt.axhline(y=thresholds[var], linestyle="--", label=f"{var} threshold")
    plt.legend()
    plt.show()
```

In this way, the implementation produces reproducible datasets and visual traces that allow verification of system stability, identification of anomalies, and quantitative analysis of threshold crossings.

The implementation presented in this section establishes a physics-aware testbed,

in which virtual observables evolve under controlled stochastic dynamics and are continuously evaluated against critical limits. Having outlined the methodological structure, we now turn to simulated scenarios that demonstrate the system's behaviour under representative experimental conditions.

## 4. RESULTS AND DISCUSSION

The temporal evolution of environmental parameters was examined under conditions similar to those of a controlled physics laboratory, where temperature, illumination, and humidity act as mutually coupled observables that influence the reproducibility of measurements. In such systems, even slow drifts in temperature or intensity can alter refractive indices, detector gain, or the stability of resonant components. The simulated data therefore represent not only computational constructs but physically interpretable states that would, in a real experimental setting, mark transitions between stable and unstable regimes.

The temporal behaviour of temperature in confined experimental systems can be described as a coupled deterministic-stochastic process governed by the balance between internal power dissipation and external heat exchange. In practical laboratory conditions, even small variations in driving current or environmental convec-

tion can produce measurable temperature oscillations with a characteristic relaxation time comparable to the thermal response of the apparatus. This behaviour can be approximated by a first-order differential relation:

$$C \frac{dT}{dt} = P_{in}(t) - \kappa [T(t) - T_{env}], \quad (1)$$

where  $C$  is the effective thermal capacity,  $P_{in}(t)$  – the instantaneous dissipated power, and  $\kappa$  – the heat-transfer coefficient to the surroundings. For a periodically modulated power input, the steady-state solution of this equation naturally exhibits a harmonic term with amplitude  $A$  and period  $\tau$  superimposed on a slow monotonic drift that arises from incomplete heat removal.

Following this reasoning, the simulated temperature evolution was represented phenomenologically as follows:

$$T(t) = T_0 + A \sin\left(\frac{2\pi t}{\tau}\right) + \gamma t + \varepsilon_t, \quad \varepsilon_t = \phi \varepsilon_{t-1} + \sigma \eta_t. \quad (2)$$

Here  $T_0 = 26^\circ\text{C}$  represents the mean equilibrium temperature,  $A = 3^\circ\text{C}$  – the amplitude of thermal modulation, and  $\gamma = 3 \times 10^{-4}^\circ\text{C/s}$  – a slow monotonic drift associated with imperfect heat removal. The correlated noise term  $\varepsilon_t$  reproduces the typical signature of thermal inertia with  $\phi = 0.85$  and  $\sigma = 0.18^\circ\text{C}$ . The deterministic part of this expression reflects the classical solution of the relaxation equation, while the additive autoregressive term  $\varepsilon_t$  introduces correlated thermal noise. Unlike uncorrelated Gaussian fluctuations, this term captures the finite thermal memory of

real systems, in which temperature deviations persist due to slow conduction and convection processes. Similar approaches have been successfully used in the modelling of micro-heater arrays [29], diode-laser thermal feedback loops [30], and photothermal sensing [31]. The correlated noise process, parameterized by the coefficient  $\phi$ , reproduces the experimentally observed  $1/f^\alpha$  spectral dependence of temperature fluctuations in massive optical and electronic assemblies, typically with  $0.8 < \alpha < 1.2$  [30], [32].

The statistical formulation introduced

above was implemented numerically to reproduce the thermal evolution of a laboratory-scale enclosure under cyclic heating and correlated noise excitation. The model parameters were selected to emulate conditions typical for stabilized electronic or optical assemblies, where the relaxation time of the structure is comparable to the external modulation period. The synthetic signal was generated using the discrete formulation of Eq. (2), with an autoregressive component ensuring temporal correlation of the noise sequence. The persistence criterion  $\Delta t_p$  was introduced to identify intervals in which  $T(t)$  remained above the critical threshold  $T_t$  for several consecutive samples, representing sustained departures from thermal equilibrium rather

than momentary transients. Such treatment allows the simulation to capture the essential thermodynamic asymmetry of heating and cooling cycles: while the sinusoidal term defines the nominal response to modulated power, the drift and correlated noise determine whether localized overheating persists long enough to induce measurable physical consequences. In practice, this corresponds to situations where residual heat in semiconductor mounts or laser diodes accumulates faster than it can dissipate, leading to delayed recovery and oscillatory instability around the equilibrium point. The code segment below numerically integrates these relations and produces the time trace shown in Fig. 2.

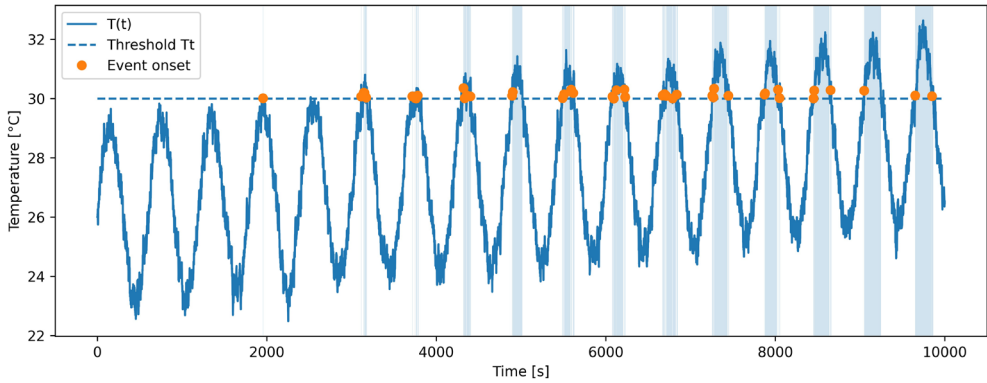


Fig. 2. Simulated time dependence of temperature  $T(t)$  (solid line) with the critical limit  $T_t = 30^\circ\text{C}$  (dashed). Shaded intervals denote confirmed overheating events satisfying the persistence condition  $\Delta t_p = 5\text{ s}$ , while open circles mark the onset of each instability.

The temperature evolution depicted in Fig. 2 exhibits the interplay between deterministic modulation and correlated fluctuations expected for systems with finite heat capacity. The oscillatory background arises from periodic modulation of the dissipated power  $P_{in}(t)$ , while the slow drift  $\gamma t$  represents incomplete compensation between input and output heat fluxes. The correlated noise component  $\varepsilon_t$  introduces

long-memory behaviour that mimics real conduction and convection processes within the apparatus. Consequently, temperature excursions above  $T_t$  are not isolated impulses but extended plateaus that satisfy the persistence condition, indicating quasi-stationary overheating states. From a physical standpoint, these intervals correspond to the regime where the power supplied to the system temporarily exceeds

its radiative and convective loss rate, producing local thermal accumulation. The mean duration of such intervals ( $\approx 11$  s) is consistent with relaxation times reported for aluminium or steel optical mounts of similar volume under comparable heat loads [33]. Each sustained crossing of  $T_l$  marks a state in which material parameters undergo nonlinear variation—thermal expansion of order  $10^{-6}$ , refractive-index shifts  $\Delta n/n \approx 10^{-5}$ , and resistivity changes that can alter bias conditions in semiconductor junctions. In optical experiments, even sub-degree changes at this level are sufficient to cause phase drifts in interferometers and to destabilize laser cavity alignment. The persistence-based event identification, as shown by shaded regions, thus provides a direct quantitative analogue of experimental interlock logic, distinguishing thermodynamically significant overheating from random jitter. The resulting time series reflects the coupled nature of thermal noise and slow feedback typical of open physical systems: heating and cooling are dynami-

cally imbalanced, producing self-sustained fluctuations whose statistical properties follow a  $1/f^\alpha$  spectrum, in agreement with Refs. [30], [32].

The analysis was further extended to the illumination subsystem, which in most laboratory settings serves as the dominant feedback channel coupling optical power to thermal and electronic stability. In systems employing diode or solid-state lasers, the instantaneous intensity  $L(t)$  is subject to both continuous stochastic modulation and discrete discontinuities arising from cavity dynamics or active optical control. The intensity variations, although apparently random, are governed by well-defined physical processes such as relative intensity noise (RIN) originating from carrier-density fluctuations, mode hopping due to cavity length detuning, and blanking events imposed by acousto-optic or electro-optic modulators. To reproduce these features, the temporal evolution of  $L(t)$  was synthesized as the sum of three terms:

$$L(t) = L_0 [1 + \eta(t)] + \sum_i \Delta L_i e^{-(t-t_i)/\tau_r} + \sum_j \Theta_j(t), \quad (3)$$

where  $L_0$  is the nominal optical intensity, represents RIN with a  $1/f^\alpha$  power-spectral density ( $\alpha \approx 1$ ),  $\Delta L_i$  are mode-hop transients decaying exponentially with recovery time  $\tau_r$ , and  $\Theta_j(t)$  are deep attenuation windows corresponding to active optical blanks. The RIN term was implemented via spectral shaping of white noise in the Fourier domain, ensuring realistic low-frequency dominance consistent with experimental power-spectral measurements of diode and Ti:sapphire systems [34], [35]. The inclusion of negative exponential transients captures the cavity re-locking behaviour following mode hops, as observed in temperature- or current-swept laser diodes. The blanking windows emulate short interruptions of the optical beam—common during pulse-pick gating or acousto-optic modulator (AOM), typically producing transient relaxation of the detection electronics and measurable under-illumination at the sensor.

The trace in Fig. 3 illustrates the characteristic interplay between stochastic RIN fluctuations and discrete nonlinear events, analogous to the response of actively sta-

bilized laboratory systems. The  $1/f^\alpha$  noise background defines the broadband low-frequency modulation that limits power stability, while the exponentially recovering

steps represent transient cavity instabilities or mode changes. The sharp minima correspond to deliberate AOM blanking intervals, which in real experiments are used to synchronize illumination with acquisition windows or to prevent detector saturation. From a physical standpoint, these features map directly onto the operational regimes of laboratory systems. The relative noise amplitude of approximately 2 % *rms* is representative of mid-power diode lasers operated under moderate feedback conditions [34], [35]. The recovery constant  $\tau_r = 20$  s agrees with empirical values reported for thermally compensated extended-cavity lasers [36]. Each detected under-illumination event (shaded intervals) reflects a period where the local optical power drops below the sensor's linear response region,

potentially compromising the accuracy of intensity-dependent measurements. The persistence-based logic used to identify such events, analogous to that employed for temperature analysis, provides a quantitative tool for distinguishing genuine power degradation from high-frequency flicker. This approach thus bridges stochastic and event-driven phenomena in laboratory systems, enabling numerical reproduction of intensity instabilities that are otherwise observable only through dedicated diagnostic instrumentation. The physical realism of the model is supported by the agreement of fluctuation spectra with experimental RIN profiles and by the temporal scales consistent with known mode-hop and servo-recovery dynamics in diode and fibre laser systems [32]–[34].

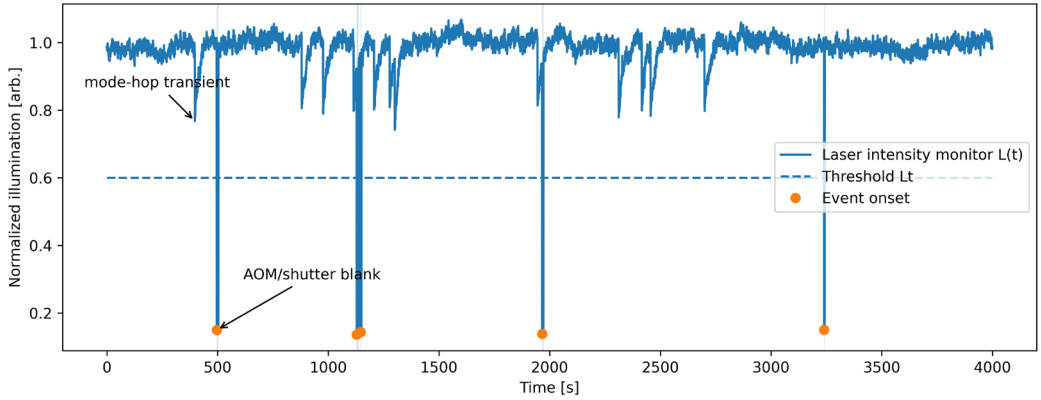


Fig. 3. Simulated illumination trace  $L(t)$  showing combined effects of RIN, discrete mode-hop transients with exponential recovery, and AOM blanking intervals. The dashed line indicates the critical illumination threshold  $L_t = 0.6$ . Shaded regions correspond to confirmed under-illumination events satisfying the persistence condition  $\Delta t_p = 3$  s.

Following the analysis of thermal and optical stability, the third observable, relative humidity  $H(t)$  is introduced to represent the slowest environmental variable governing charge transport, dielectric strength, and optical quality within

laboratory setups. Unlike temperature or illumination, humidity evolves primarily through diffusion and sorption kinetics, where molecular exchange occurs between air, surfaces, and porous materials. These processes are nonlinear and asymmetric:

moisture adsorption proceeds rapidly once the local vapour pressure exceeds equilibrium, whereas desorption is diffusion-limited, leading to prolonged relaxation tails. The long correlation times associated with humidity therefore make it a critical variable in experiments involving high-voltage

electronics, interferometric optics, or sensitive dielectric samples, where condensation or ionic conduction can destabilize measurements. Mathematically, this regime can be captured by a hysteretic relaxation model with diffusion-like stochastic forcing:

$$\frac{dH}{dt} = \frac{H_{eq}(t) - H}{\tau_{\uparrow}} \Theta[H_{eq}(t) - H] + \frac{H_{eq}(t) - H}{\tau_{\downarrow}} \Theta[H - H_{eq}(t)] + \xi(t), \quad (4)$$

where  $\tau_{\uparrow}$  and  $\tau_{\downarrow}$  denote the characteristic time constants of humidification and drying, respectively ( $\tau_{\downarrow} > \tau_{\uparrow}$ ),  $\Theta$  is the Heaviside step function, and  $\xi(t)$  represents small correlated perturbations associated with turbulent mixing and micro-leakage. Short-lived infiltration impulses (e.g., door openings or transient vapour loads) are introduced as additive exponential terms  $A_i e^{-(t-t_i)/\tau_i}$  superimposed on the slowly varying equilibrium  $H_{eq}(t)$ . This approach reflects the same formalism used in diffusion-reaction studies of confined atmospheres and moisture transport in dielectric composites [37], [38].

The simulated trace reproduces several physically distinct phenomena observable in closed laboratory environments. The initial fluctuations below 60 % RH correspond to a regime dominated by random exchange with the surroundings, yielding a low-amplitude, diffusion-like noise with a correlation time of a few minutes. Around  $t \approx 50$  min, an infiltration pulse drives a rapid increase in humidity, followed by a slow exponential relaxation, an archetypal signature of asymmetric sorption dynamics. As  $\tau_{\downarrow} > \tau_{\uparrow}$ , drying proceeds more sluggishly than humidification, producing extended plateaus and hysteretic cycles analogous to those reported in adsorption-desorption experiments on polymeric films [39]. The shaded intervals highlight periods where  $H(t)$  remains above the condensation threshold  $H_c$  for longer than  $\Delta t_p$ . Such events correspond to physically meaningful states where water films can form on optical mounts or PCB surfaces, leading

to increased dielectric losses, charge leakage, or phase instability. In this sense, the persistence criterion acts as a temporal filter that distinguishes statistically rare fluctuations from environmentally significant excursions, a distinction essential in the automation of safety interlocks and stability diagnostics. The cluster of events near  $t \approx 100$  min illustrates cumulative vapor accumulation under weak ventilation, after which the system gradually re-establishes equilibrium through slow diffusive exchange. Overall, the humidity response complements the temperature and illumination models by emphasising mass-transport-limited relaxation rather than thermal or radiative coupling. Its inclusion closes the triad of environmental observables necessary for full physics-aware IoT simulations, where each subsystem, thermal, optical, and hygroscopic, acts as a distinct yet interacting dissipative channel.

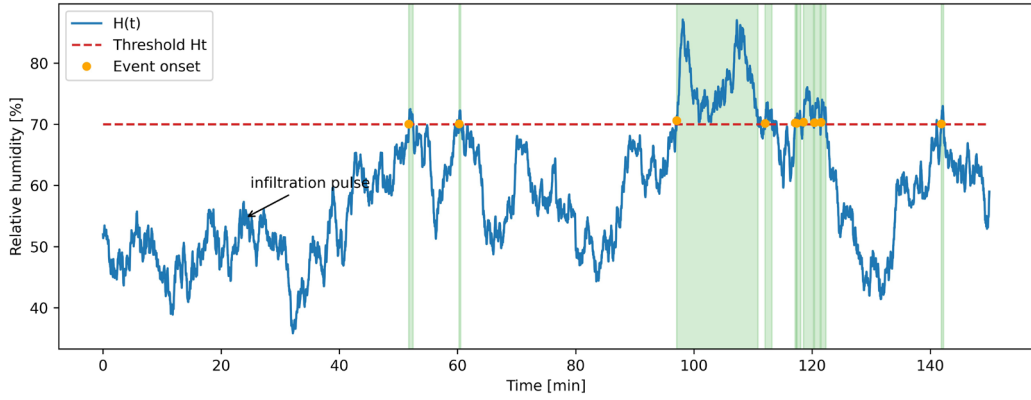


Fig. 4. Simulated relative humidity  $H(t)$  exhibiting diffusion-driven relaxation with hysteresis ( $\tau_{\uparrow} = 250$  s and  $\tau_{\downarrow} = 900$  s), superposed infiltration spikes, and small correlated fluctuations. The dashed line marks the critical humidity  $H_t = 70\%$ , above which condensation and surface-conductivity effects emerge. Violet markers denote confirmed event onsets based on a persistence window of  $\Delta t_p = 10$  s, and green-shaded intervals represent sustained over-humidity episodes.

## 5. CONCLUSIONS

The study presented a physics-aware simulation framework reproducing the core dynamics of sensing, evaluation, and feedback processes encountered in experimental laboratory environments. By emulating temperature, illumination, and humidity through deterministic-stochastic models, the system successfully bridges the conceptual gap between traditional IoT architectures and real physical observables governed by diffusion, conduction, and radiative mechanisms. Each simulated quantity, thermal drift, optical intensity fluctuation, and hygroscopic relaxation, was described using compact analytical forms derived from established physical laws, while stochastic components were introduced to reproduce the correlated noise and hysteresis typical of real instrumentation. The proposed persistence-based threshold logic further demonstrated how critical boundaries in physical systems can be treated analogously to safety or control limits in IoT signal processing. The temporal persistence

criterion effectively distinguished genuine instabilities, such as sustained overheating or prolonged over-humidity, from random noise-induced excursions. This feature is particularly valuable in the automation of monitoring systems for precision optical experiments, vacuum setups, or high-voltage electronics, where small environmental fluctuations can propagate into measurable signal degradation or irreversible failure. Beyond its computational simplicity, the framework's strength lies in its interpretability: all model parameters, heat capacity, relaxation times, radiative coupling coefficients, and diffusion constants, have direct physical analogues. This correspondence ensures that the simulated results remain quantitatively meaningful and scalable to experimental setups of different geometries or materials. Future developments may include coupling the simulated observables through energy-balance equations to capture inter-variable correlations (e.g., humidity-dependent thermal conductivity

or illumination-induced heating). Another direction involves extending the alerting architecture to include reinforcement-based adaptive thresholds, allowing the system to autonomously recalibrate in response to changing boundary conditions. These

advancements would further enhance the role of software-based, physics-grounded IoT systems as tools for both predictive diagnostics and pedagogical visualisation in applied physics laboratories.

## ACKNOWLEDGEMENT

---

The authors would like to acknowledge the support received from the Science Fund of the Republic of Serbia, Grant No. 6821, “Atoms and (bio) molecules: dynamics and collisional processes on short time scales – ATMOLCOL.” The authors also acknowledge the support provided by the Kronospan Foundation through the International Competition of Projects “skill.ED”, Edition 2025, within the project “Sensors, Physics and AI Lab for Industrial Training (SPARK-ED)”. Our appreciation also

goes to the Serbian Ministry of Education, Science and Technological Development (Agreement No. 451-03-136/2025-03/200122). H. Delibašić Marković would also like to express gratitude to COST Actions CA21159 – “Understanding interaction of light with biological surfaces: possibility for new electronic materials and devices” for their support and CA22148 – “An international network for Non-linear Extreme Ultraviolet to hard X-ray techniques (NEXT)”.

## REFERENCES

---

1. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>
2. Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440. <https://doi.org/10.1016/j.bushor.2015.03.008>
3. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
4. Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. *Computer Communications*, 89, 5–16. <https://doi.org/10.1016/j.comcom.2016.03.015>
5. Yassein, M. B., Mardini, W., & Khalil, A. (2016). Internet of Things architectures for small and medium enterprises. In *2016 International Conference on Engineering & MIS (ICEMIS)* (pp. 1–6). IEEE. <https://doi.org/10.1145/3234698.3234730>
6. McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the Python in Science Conference (SciPy)* (pp. 51–56). Austin, TX: SciPy.
7. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
8. Bröring, A., Seeger, J., Papoutsakis, M., Fysarakis, K., & Caracalli, A. (2020). Networking-Aware IoT Application Development. *Sensors*, 20(3), 897. <https://doi.org/10.3390/s20030897>

9. Takanashi, K., Udagawa, K., Seki, M., Okada, T., & Tanaka, H. (1975). Non-linear Earthquake Response Analysis of Structures by a Computer-Actuator On-line System. *Bulletin of the Earthquake Resistant Structure Research Center*, 8, 1–10.
10. Chen, C., Ricles, J. M., Marullo, T. M., & Mercan, O. (2009). Real-time hybrid testing using the unconditionally stable explicit CR integration algorithm. *Earthquake Engineering & Structural Dynamics*, 38(1), 23–47. <https://doi.org/10.1002/eqe.838>
11. Kolay, C., Ricles, J. M., Marullo, T. M., Mahvashmohammadi, A., & Sause, R. (2015). Implementation and application of the unconditionally stable explicit parametrically dissipative KR- $\alpha$  method for real-time hybrid simulation. *Earthquake Engineering & Structural Dynamics*, 44(5), 735–755. <https://doi.org/10.1002/eqe.2484>
12. Chae, Y., Kazemibidokhti, K., & Ricles, J. M. (2013). Adaptive time series compensator for delay compensation of servo-hydraulic actuator systems for real-time hybrid simulation. *Earthquake Engineering & Structural Dynamics*, 42(11), 1697–1717. <https://doi.org/10.1002/eqe.2294>
13. Bas, E. E., & Moustafa, M. A. (2020). Real-time hybrid simulation with deep learning computational substructures: System validation using linear specimens. *Machine Learning and Knowledge Extraction*, 2(4), 469–489. <https://doi.org/10.3390/make2040026>
14. Zhang, X., Xie, X., Tang, S., Zhao, H., Shi, X., Wang, L., ... & Xiang, P. (2024). High-speed railway seismic response prediction using CNN-LSTM hybrid neural network. *Journal of Civil Structural Health Monitoring*, 14(5), 1125–1148. <https://doi.org/10.1007/s13349-023-00758-6>
15. Alam, T. (2021). Cloud-based IoT applications and their roles in smart cities. *Smart Cities*, 4(3), 1196–1212. <https://doi.org/10.3390/smartcities4030064>
16. Chernyshev, M., Baig, Z., Bello, O., & Zeadally, S. (2017). Internet of Things (IoT): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3), 1637–1647. <https://doi.org/10.1109/JIOT.2017.2786639>
17. Li, S., Tryfonas, T., & Li, H. (2016). The Internet of Things: A security point of view. *Internet Research*, 26(2), 337–359. <https://doi.org/10.1108/IntR-07-2014-0173>
18. Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., ... & Wouters, S. (2018). PySCF: The Python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, 8(1), e1340. <https://doi.org/10.1002/wcms.1340>
19. Krishnamurthy, J., & Maheswaran, M. (2016). Programming frameworks for Internet of Things. *Internet of Things*, 79–98. <https://doi.org/10.1016/B978-0-12-805395-9.00005-8>
20. Li, S., Xu, L. D., & Zhao, S. (2015). The Internet of Things: A survey. *Information Systems Frontiers*, 17, 243–259. DOI:10.1007/s10796-014-9492-7
21. Kazanskiy, N. L., Khonina, S. N., & Butt, M. A. (2024). A review on flexible wearables—Recent developments in non-invasive continuous health monitoring. *Sensors and Actuators A: Physical*, 366, 114993. <https://doi.org/10.1016/j.sna.2023.114993>
22. Garg, A. (2024). *Analysis and Mitigation of Single Event Transients (SET) effect on RISC-V based FPGA* (Master's Thesis). Politecnico di Torino.
23. Coito, T., Martins, M. S., Firme, B., Figueiredo, J., Vieira, S. M., & Sousa, J. M. (2022). Assessing the impact of automation in pharmaceutical quality control labs using a digital twin. *Journal of Manufacturing Systems*, 62, 270–285. <https://doi.org/10.1016/j.jmsy.2021.11.014>
24. Chen, D., Shi, X., Zhang, H., Song, X., Zhang, D., Chen, Y., & Yan, J. (2024). A phone-based distributed ambient temperature measurement system with an efficient label-free automated training strategy. *IEEE Transactions on Mobile Computing*, 23(12), 11781–11796. <https://doi.org/10.1109/TMC.2024.3399843>

25. Hamzaoui, I., Duthil, B., Courboulay, V., & Medromi, H. (2024). A topical review on container-based cloud revolution: Multi-directional challenges, and future trends. *SN Computer Science*, 5(4), 416. <https://doi.org/10.1007/s42979-024-02763-y>
26. Morcillo-Jimenez, R., Rivas, J. M., Ruiz, M. D., Martin-Bautista, M. J., & Fernandez-Basso, C. (2025). Privacy-preserving energy analytics in smart offices via container-based Federated Learning. *Internet of Things*, 34, 101782. <https://doi.org/10.1016/j.iot.2025.101782>
27. Fokina, E. A., Trofimov, A. A., Ponomarev, V. N., & Zdobnov, S. A. (2024). Design of a Simulation Model of a Temperature Sensor under Critical Temperature and Sinusoidal Vibrations. *Technical Physics*, 69(11), 2620–2630. <https://doi.org/10.1134/S1063784224701093>
28. Rut'kov, E. V., Beliaeva, O. A., & Gall, N. R. (2023). Physical Processes in the Pirani Type Low Vacuum Sensor. *Technical Physics*, 68(12), 766–773. <https://doi.org/10.1134/S1063784223080327>
29. Ju, L., Jiang, P., Ren, Y., Liu, R., Kong, Y., Yun, S., Ye, Y., Jiao, B., Hao, Q., & Sun, H. (2025). Overview of research progress and application prospects of thermal test chips. *Micromachines*, 16(6), 669. <https://doi.org/10.3390/mi16060669>
30. Geitner, M., Sandoval, F. A. S., Bertin, E., & Bellon, L. (2017). Low thermal fluctuations in a system heated out of equilibrium. *Physical Review E*, 95, 032138. <https://doi.org/10.1103/PhysRevE.95.032138>
31. Michaud-Belleau, V., Fokoua, E. R. N., Horak, P., Wheeler, N. V., Rikimi, S., Bradley, T. D., ... & Genest, J. (2022). Fundamental thermal noise in antiresonant hollow-core fibers. *Physical Review A*, 106, 023501. <https://doi.org/10.1103/PhysRevA.106.023501>
32. Varley, J. B., Lordi, V., He, X., & Rockett, A. (2016). First principles calculations of point defect diffusion in CdS buffer layers: Implications for Cu(In,Ga)(Se,S)<sub>2</sub> and Cu<sub>2</sub>ZnSn(Se,S)<sub>4</sub>-based thin-film photovoltaics. *Journal of Applied Physics*, 119(2), 025703. <https://doi.org/10.1063/1.4939656>
33. Saggini, B., Tarabini, M., & Scaccabarozzi, D. (2010). Infrared optical element mounting techniques for wide temperature ranges. *Applied Optics*, 49(3), 542–550. <https://doi.org/10.1364/AO.49.000542>
34. Zhu, X., Wang, J., Gao, X., Fan, J., & Hu, M. (2025). Stable cascaded femtosecond optical parametric amplifiers in the NIR-I region at 50 MHz. *Chinese Optics Letters*, 23(1), 011902. <https://doi.org/10.1364/COL.23.011902>
35. Zhang, M., Xie, Y., Zhang, J., Wang, W., Wu, C., Chen, T., ... & Chen, P. (2021). Estimation of the laser frequency noise spectrum by continuous dynamical decoupling. *Physical Review Applied*, 15(1), 014033. <https://doi.org/10.1103/PhysRevApplied.15.014033>
36. Jiang, X., Scott, J., Friesen, M., & Saffman, M. (2023). Sensitivity of quantum gate fidelity to laser phase and intensity noise. *Physical Review A*, 107(4), 042611. <https://doi.org/10.1103/PhysRevA.107.042611>
37. Burns, I. S., Hult, J., Hartung, G., & Kaminski, C. F. (2007). A thermometry technique based on atomic lineshapes using diode laser LIF in flames. *Proceedings of the Combustion Institute*, 31(1), 775–783. <https://doi.org/10.1016/j.proci.2006.07.243>
38. Tartakovsky, D. M., & Dentz, M. (2019). Diffusion in porous media: Phenomena and mechanisms. *Transport in Porous Media*, 130(1), 105–127. <https://doi.org/10.1007/s11242-019-01262-6>
39. Cartiailler, V. (2021). *Study of temperature and moisture impact on water diffusion in materials and on electronic circuits failure mechanisms* (Doctoral dissertation). Université de Bordeaux.
40. Jeromenok, J., & Weber, J. (2013). Restricted access: On the nature of adsorption/desorption hysteresis in amorphous, microporous polymeric materials. *Langmuir*, 29(42), 12982–12992. <https://doi.org/10.1021/la402630s>