

International Journal of Critical Computer-Based Systems

ISSN online: 1757-8787 - ISSN print: 1757-8779

<https://www.inderscience.com/ijccbs>

Mathematical processing units: a prototype for enabling next-generation ambient AI

Rohan Sinha, Aadi Jain, Dev Syam, Shivam Kundan

DOI: [10.1504/IJCCBS.2026.10081732](https://doi.org/10.1504/IJCCBS.2026.10081732)

Article History:

Received:	02 July 2025
Last revised:	04 September 2025
Accepted:	07 August 2026
Published online:	27 October 2026

Mathematical processing units: a prototype for enabling next-generation ambient AI

Rohan Sinha

Woodside Priory School,
302 Portola Rd., Portola Valley, CA 94028, USA
Email: rohansinha19@gmail.com

Aadi Jain

School of Computer Science,
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
Email: aadijain@andrew.cmu.edu

Dev Syam

GEMS Modern Academy,
Nad al Sheba, Dubai, United Arab Emirates
Email: devsyam101@gmail.com

Shivam Kundan*

School of Electrical, Computer and Biomedical Engineering,
Southern Illinois University,
1230 Lincoln Drive, Carbondale, IL 62901, USA
Email: shivamkundan@siu.edu
*Corresponding author

Abstract: The next generation of AI spanning from large language models (LLMs) to joint embedding predictive architectures (JEPAs) depend heavily on high-quality ambient data acquisition. However, the effectiveness of these future models is directly tied to the fidelity and efficiency of their training data pipelines. We present a proof-of-concept for a new type of processing hardware called mathematical processing units (MPUs), built upon commercially available FPGAs and designed to acquire and pre-process ambient sensor data with high fidelity, low power, and low cost, while also aiding in sensor fusion and fault tolerance. We demonstrate the effectiveness of the proposed approach by performing Kalman filtering on environmental data. Across six sensing scenarios, the MPU-based Kalman filter implementation reduced measurement noise by an average of 77.2%, demonstrating substantial improvement in signal fidelity. These results highlight the potential of MPUs as core infrastructure for future AI systems that interact with the physical world.

Keywords: FPGA; Kalman filtering; ambient intelligence; signal processing; VLSI design; large language models; LLMs; mathematical processing units; MPUs.

Reference to this paper should be made as follows: Sinha, R., Jain, A., Syam, D. and Kundan, S. (2026) ‘Mathematical processing units: a prototype for enabling next-generation ambient AI’, *Int. J. Critical Computer-Based Systems*, Vol. 12, No. 5, pp.1–21.

Biographical notes: Rohan Sinha is an undergraduate at the University of California, Berkeley, studying applied mathematics with interests in artificial intelligence, technology, and entrepreneurship. He is a Co-founder of NANDAShapers, a student-led initiative that has reached over 5,000 students across universities and aims to give students a voice in shaping the future of AI. He is particularly interested in applying mathematics and technology to real-world problems and understanding the broader societal impact of AI.

Aadi Jain is an undergraduate in Computer Science at Carnegie Mellon University, where he is doing research at the Microdynamic Systems Lab under Professor Ralph Hollis, using a magnetic-levitation haptic device to simulate force feedback for realistic, virtual surgical training. He is interested in problems at the hardware-software boundary: robotics, haptics, machine learning, and autonomous systems, including current work on vision-language-action models for small-scale autonomous racing.

Dev Syam is an undergraduate at the University of Warwick, studying Accounting and Finance with interests in equity research and application of AI in financial analysis. He worked as an Equity Research Intern at Mirae Asset Capital Markets in the TMT and Energy & Natural Resources sectors, and is focused on how AI-driven tools can improve forecasting accuracy and fundamental research. He founded Tetricks, a circular economy startup that converted surplus food waste into low-carbon construction materials. He is interested in the intersection of AI, finance, and sustainability, and how intelligent automation can reshape investment research and capital allocation toward climate-positive businesses.

Shivam Kundan received his PhD in Electrical and Computer Engineering from Southern Illinois University, Carbondale. His research focuses on AI-guided, resource-aware scheduling for improving performance-per-Watt in modern heterogeneous multicore processors, with broader interests in computer architecture and operating systems. He led the research presented in this study as part of a research project at Opengrowth Ventures.

1 Introduction

The next frontier of artificial intelligence lies in its seamless integration with the physical world. This work proposes a novel hardware implementation termed *mathematical processing units (MPUs)*, which aim to accelerate the process of developing and deploying the next generation of AI-powered systems in real-world applications, while also contributing to the advancement of fundamental research through low-cost and high-fidelity sensor data acquisition.

The primary goal of MPUs is to perform real-time mathematical pre-processing and signal conditioning to improve the quality of acquired ambient data. They also facilitate sensor fusion directly at the edge by combining multiple sensor streams and ensuring resiliency through fault tolerance and error detection methods. This also has the effect of

improving latency and reducing the workload of power-hungry compute units. Consequently, MPUs function as a power-efficient, low-cost, and high-quality source of ambient data that will facilitate responsive and adaptive behaviour in future AI systems.

1.1 Technological benefits of MPUs

Emerging applications such as humanoid robotics, autonomous vehicles (AVs), augmented reality (AR), and smart environments demand the real-time interpretation of complex multimodal sensor data. These systems must process vast amounts of ambient information both efficiently and accurately, often under stringent power consumption and latency constraints (Hazlewood and Coyle, 2009). MPUs represent a hardware innovation specifically tailored to address these challenges. Several applications can benefit from the integration of MPUs:

- **Humanoid robotics:** as per Google’s DeepMind division, progress beyond LLM powered technologies into robotics and other advanced agentic applications requires new sources of data (Silver and Sutton, 2025). This can be achieved by enabling AI systems to learn continually from their own experiences, with data being generated autonomously as the system interacts with its environment. Static, synthetically generated data will quickly become insufficient in such settings (Silver and Sutton, 2025). Since MPUs improve the quality of acquired ambient data through mathematical pre-processing, they offer a promising solution by empowering humanoid robots to autonomously and continually acquire information through physical interaction. MPUs can also offload pre-processing tasks from more expensive compute units, allowing those to focus on inferencing, communication, or other computationally intensive functions that enhance humanoid capabilities. Moreover, integrating multiple MPUs can provide redundancy and improve dependability which are essential features for mission-critical deployments.

Humanoid systems also face inherent sensing limitations such as self-occlusion, narrow sensor field-of-view, and calibration challenges (Zhang et al., 2025). These constraints can be mitigated through advanced sensor fusion facilitated by MPUs. Prior work on multimodal perception frameworks has shown the potential of embedded co-processors in aligning diverse sensor streams despite the problem of missing data and transformation inconsistencies (Zambelli et al., 2020; Roychoudhury et al., 2023), a role that MPUs are well-suited to fulfil.

Ultimately, the integration of MPUs in humanoid robotics leads to enhanced proprioception and environmental awareness, driven by higher quality and more diverse data inputs.

- **Autonomous vehicles:** one of the primary constraints preventing AVs from achieving higher levels of autonomy, especially SAE level 5 (Chen et al., 2023), is the high-speed acquisition and processing of environmental data under imperfect and adverse weather conditions such as rain, fog, and snow (Zhang et al., 2023). These conditions can degrade the quality and reliability of sensor data, requiring resilient fusion and abstraction techniques. MPUs can perform perception enhancement methods in real time to improve sensor fusion and quality monitoring. This is especially important in the context of low-level fusion where raw data from multiple modalities must be combined, as described in joint sensor data abstraction (Reichertt

et al., 2021). MPUs also enable a more distributed architecture in which individual vehicle subsystems process their own ambient information, enabling faster, localised decision-making while freeing the central compute units to focus on global tasks such as a-priori map integration, cloud communication, route-planning, and other computationally intensive operations.

Sensor failure is another major challenge in AVs (Matos et al., 2024), which can be ameliorated by integrating one or more MPUs acting as fault detectors and signal conditioners. MPUs can support calibration, metadata-informed abstraction, and detection of edge-case sensor anomalies, all of which are critical for dependable perception.

The end result is enhanced ability, dependability, and safety in autonomous driving systems, made possible by embedded intelligence that improves the adaptability and robustness of perception pipelines across variable sensor setups.

- **Augmented reality:** AR applications demand both low latency and high-bandwidth processing of multi-modal sensor data such as cameras, inertial measurements, and depth streams such as LiDAR, often under varying lighting and occlusion conditions. Embedded MPUs can perform real-time perception enhancement such as visual-inertial sensor fusion directly on AR headsets or mobile devices, reducing latency and network dependency (Łysakowski et al., 2023).

MPUs also provide resilience against tracking failures by integrating inertial measurements with optical data using Kalman-filtered fusion to maintain stable overlay alignment during fast motion or occlusion events (Fang et al., 2017). Moreover, previous research has shown that sensor fusion in industrial environments backed by embedded processors, such as the proposed MPUs, can enhance geometric registration and overlay accuracy while preserving real-time performance (Kumar et al., 2019). Finally, leveraging edge-embedded compute units like MPUs aligns with merged AR/edge architectures have been shown in previous works to balance latency, energy use, and scalability in both mobile and headset-based AR systems (Liu et al., 2019). In summary, integrating MPUs can enable delivery of robust, responsive, and autonomous AR experiences.

- **Smart environments:** smart environments, such as context-aware homes, offices, and public spaces, rely on a dense network of embedded IoT and edge systems that sense human presence and ambient conditions to adapt automatically (Perera et al., 2013; Ben Sada et al., 2023). These systems must process multimodal sensor data (e.g., motion, temperature, audio, light) in real-time to accurately infer occupant behaviour and environmental context, despite noisy signals and varying data rates.

MPUs can perform localised, low-latency sensor fusion and context reasoning at the edge, significantly reducing the need to offload raw data to central servers. This aligns with the selective sensing paradigm, where context-aware edge logic decides which data to sample, pre-process, or transmit, which are effects demonstrated in frameworks like CONTESS for smart cities (Ben Sada et al., 2023). By embedding automated sensor fusion pipelines such as dynamic Bayesian networks for multi-sensor integration MPUs can combine heterogeneous IoT inputs while accounting for uncertainty and noise (De Paola et al., 2016).

Moreover, MPUs enable secure and adaptive context handling, supporting fine grained encryption and policy enforcement through attribute-based encryption in smart environments (Ben Sada et al., 2023). This ensures that data streams (e.g., occupancy, biometrics) are filtered, reasoned over, and shared according to privacy and energy constraints. As a result, systems can proactively adjust lighting, HVAC, access control, or queue management, thus achieving both responsiveness and resilience.

By leveraging localised MPUs for real-time inference, selective sensing, and secure context adaptation, smart environments can achieve higher energy efficiency, user comfort, and autonomy, ultimately becoming truly intelligent, responsive spaces.

By improving the quality of input data, augmenting compute capacity, and simplifying computational workload distribution for emerging technologies such as humanoid robotics, augmented reality, autonomous vehicles, and smart environments, the proposed MPU implementation can play a pivotal role in the evolution of AI systems that utilise ambient intelligence.

1.2 Benefits to fundamental research

Beyond emergent technological applications, MPUs offer significant potential to advance fundamental scientific research, particularly in fields requiring high volume and diverse information from experimentation such as *infodynamics* and *cosmology*.

Infodynamics, an emerging discipline, explores the interplay between information and physical systems (Vopson, 2019). Recent theoretical developments, such as the proposed *Second Law of Infodynamics*, suggest that information entropy may decrease over time in certain systems, challenging the conventional thermodynamic principle that entropy always increases (Vopson, 2025). This novel idea, as highlighted in recent research and science communication literature (Orf, 2025), implies that information-centric processes could follow distinct physical laws. MPUs, with their capacity for real-time, low-power processing of complex data streams, are ideally suited to capture and analyse the subtle informational patterns that underpin such phenomena. By enabling detailed observation and modelling of information dynamics in physical systems, MPUs can provide empirical support for theoretical frameworks in infodynamics.

Cosmological experiments attempting to study aspects of the universe's origin, structure, and evolution require both large-scale and precise data acquisition for obtaining meaningful results. MPUs can enhance the capabilities of observational instruments by performing on-site data processing, reducing noise, and extracting meaningful patterns from largescale datasets. This facilitates more accurate measurements of cosmic phenomena, such as background radiation, gravitational waves, and galactic formations, thus facilitating a deeper understanding of the fundamental properties of the universe. Previous studies have demonstrated that onboard data processing via embedded processors enables real-time calibration and compression, effective noise reduction, and extraction of scientifically meaningful patterns in cosmological observations, ranging from the earliest NASA spacecraft missions to more modern implementations in missions like Euclid (Euclid Collaboration et al., 2025), SOPHI (Albert et al., 2018), MUSE (Weilbacher et al., 2020), and CMB surveys (Gerakakis et al., 2023).

By bridging the gap between theoretical physics and empirical observation, MPUs stand to accelerate discoveries in fundamental research such as infodynamics and

cosmology, offering new insights into both the informational and physical fabric of reality.

1.3 Towards offline ambient intelligence and edge-first compute architectures

As physical spaces increasingly adopt ambient intelligence, integrating sensors and actuators to sense and respond to human presence, they rely heavily on embedded computing to perform context-aware reasoning (Aarts and Encarnação, 2006; Perera et al., 2013).

While cloud-based compute offers convenience, it also introduces critical drawbacks such as latency, network unreliability, bandwidth limitations, privacy concerns, and vendor lock-in (Wikipedia, 2026a). These limitations not only degrade performance in real-time applications but also constrain deployment in privacy-sensitive or remote environments. To address these challenges, untethered compute ecosystems augmented by edge-native MPUs are essential. They enable low-latency local processing, privacy-preserving computation, and resilient operation without relying on external networks. For example:

- 1 Smart healthcare applications like remote monitoring or emergency alerts greatly benefit from edge systems, which offer faster response times and ensure compliance with privacy standards like HIPAA (Modupe et al., 2024).
- 2 Industrial automation where safety-critical processes depend on millisecond-level control loops and cannot tolerate cloud delays.
- 3 Disaster response and military operations where connectivity may be disrupted or insecure.

Beyond terrestrial uses, untethered compute is vital for robotic exploration in extreme environments:

- 1 Planetary aerobots such as those designed for Titan, rely on full onboard autonomy due to delays of up to 100 minutes for Earth communications. JPL studies outline architectures with onboard perception, flight control, planning, and science target selection, which are all necessary for robust operations (Nock et al., 1996; Elfes et al., 2005, 2008).
- 2 Underwater and deep-sea robots face similar constraints, working in high-pressure, noisy, and communication-limited environments. Modern deep-sea soft robots use embedded AI to perform navigation, sensing, and mission adaptation without tethered support (Li et al., 2023; Aracri et al., 2021).
- 3 Unmanned surface vehicles (USVs) like Wave Gliders execute long-term environmental monitoring by crossing oceans, under ice, or within critical infrastructure zones using onboard intelligence and secure edge processing to manage data collection, power, and communication continuity (Wikipedia, 2026b).

These examples show that both privacy-sensitive terrestrial applications and frontier robotic missions share a requirement for ambient, untethered intelligence. Embedded compute units such as the proposed MPUs have been shown to reduce the need for cloud offload by enabling adaptive sensing, real-time inference, local encryption, and meta data aware computation (Qu et al., 2024; Aracri et al., 2021). Additionally, decentralised edge

systems support federated learning and confidential computing, further safeguarding sensitive data and distributing compute loads across devices without relinquishing local control or privacy (Forster, 2025).

In conclusion, as we push the boundaries of intelligent environments, from smart hospitals to robotic explorers on alien moons, the need for edge-native, MPU-powered compute ecosystems becomes foundational. They offer faster response, defensible privacy, and mission resilience, traits that can only be guaranteed by immediate local processing of ambient data. This untethered approach paves the way for transformative applications in healthcare, infrastructure, extreme environments, and beyond.

2 Mathematical background

MPUs are capable of implementing several advanced mathematical techniques in real-time. They will act as a middle layer between commodity sensors and recording/compute devices with the exact type of computation and its parameters being user selectable, depending upon the use case. This allows flexibility to work with multiple sensor and data types.

2.1 Overview of various mathematical techniques

In data science, signal processing, and dynamic system modelling, a wide range of mathematical techniques are used to tackle different types of problems. Each of these techniques brings unique strengths and is tailored to address specific challenges like noise reduction, decision-making under uncertainty, dimensionality reduction, and optimisation. Below are some key mathematical techniques capable of being implemented by MPUs along with brief descriptions and examples of their applications.

- 1 Kalman filtering: the Kalman filter is an optimal recursive algorithm that estimates the true state of a dynamic system by combining noisy sensor measurements with a predictive model. It minimises the variance of estimation error over time and is widely used in tracking and navigation systems. Example: estimating the position of a drone by fusing data from GPS and an IMU (accelerometer + gyroscope).
- 2 Fast Fourier transform (FFT): FFT converts time-domain signals into their frequency-domain representation, revealing hidden periodicities and dominant frequencies. It is crucial for analysing vibration, audio, and RF signals. Example: Detecting mechanical faults in an industrial motor by analysing vibration frequencies from accelerometer data.
- 3 Moving average filter: this is a simple low-pass filter that smooths out high frequency noise by averaging a set number of recent data points. It is computationally lightweight and ideal for real-time denoising. Example: smoothing fluctuations in temperature readings from a digital thermometer.
- 4 Discrete wavelet transform (DWT): the discrete wavelet transform breaks down a signal into components at multiple scales, preserving both time and frequency information. It's particularly effective for detecting transient events in non-stationary signals. Example: identifying crack formation in a bridge structure from sudden changes in acoustic emission data.

- 5 Principal component analysis (PCA): PCA reduces the dimensionality of a dataset by transforming correlated variables into a smaller set of uncorrelated principal components, capturing the most variance. It helps extract key features from high dimensional sensor data. Example: condensing data from multiple air quality sensors (CO₂, VOC, PM_{2.5}, etc.) to a few meaningful indices for urban pollution monitoring.
- 6 Median filter: the median filter replaces each data point with the median of its surrounding values, making it robust against outliers and impulsive noise. It preserves edges better than linear filters. Example: cleaning up salt-and-pepper noise from a depth sensor or infrared camera in a robotics vision system.
- 7 Goertzel algorithm: the Goertzel algorithm is an efficient method for detecting the presence of specific frequency components in a signal without computing a full FFT. It is particularly effective when only a few frequencies are of interest, such as in tone detection or spectral analysis of narrow bands. Example: identifying DTMF tones in telecommunication signals or detecting the RPM of a motor by isolating its characteristic frequency.
- 8 Cumulative sum control chart (CUSUM): CUSUM is a sequential change detection algorithm that identifies shifts in the mean level of a measured process. It accumulates small deviations from a reference value over time, making it highly sensitive to subtle changes that might be missed by thresholding. Example: detecting early-stage sensor drift or component degradation in predictive maintenance systems by monitoring environmental or vibration data streams.

2.2 *In-depth exploration of Kalman filtering*

Kalman filtering stands out among the mathematical techniques because of its ability to handle noisy data and provide accurate, real-time state estimation in dynamic systems. Unlike many other filtering techniques, Kalman filtering is optimal in the sense that it minimises the mean squared error of the estimates. It is particularly effective when the system's state evolves over time, and measurements are corrupted by noise. To demonstrate the effectiveness of MPUs, we show the implementation a linear Kalman filter. This section provides detailed practical and mathematical background for this technique.

2.2.1 *Advantages of Kalman filtering*

Kalman filtering is a widely used mathematical framework for estimating the internal state of a dynamic system from noisy, incomplete, or indirect observations. The system 'state' comprises a set of variables such as the position and velocity of a mobile robot or the pressure and temperature within a physical environment that collectively describe the system at a given moment. In many real-world scenarios, these variables cannot be directly measured, and available sensor data are often corrupted by random noise or measurement errors. The Kalman filter addresses this challenge by generating a recursive estimate: it predicts the next system state based on a mathematical model, then corrects that prediction using the most recent sensor measurements. The fusion of model-based prediction and data-driven correction enables the filter to produce estimates that are

statistically closer to the true state than would be possible using either method independently.

A central strength of the Kalman filter lies in its explicit handling of uncertainty. Both the system model and the sensor measurements are treated probabilistically, allowing the filter to assign weights to each based on their respective confidence levels. This results in a weighted averaging mechanism in which the more reliable source, either the model or the measurement, has greater influence on the final estimate. This mechanism not only reduces the impact of noise but also enhances the filter's responsiveness to meaningful changes in the environment. Moreover, the Kalman filter operates recursively, requiring only the current state estimate and the most recent data for each update. This low memory footprint and computational efficiency make it particularly suitable for real-time and resource-constrained applications.

Kalman filtering has found extensive adoption in domains where accurate, real-time state estimation is critical. In GPS navigation, for instance, it combines noisy satellite data with vehicle kinematic models to yield precise estimates of position and velocity. In robotics and autonomous systems, Kalman filters integrate data from diverse sensors including lidar, radar, and inertial measurement units, to maintain accurate estimates of pose and motion. In aerospace applications, they are integral to navigation and control systems for aircraft and spacecraft, where they reconcile discrepancies between accelerometer and gyroscope readings. Similarly, in wearable health monitoring devices, Kalman filtering improves the reliability of physiological measurements such as heart rate and movement by filtering out sensor artefacts and noise.

Perhaps the most significant advantage of Kalman filtering is its adaptability to dynamic, time-varying systems. As new data become available, the filter incrementally refines its state estimate, enabling rapid response to changes while maintaining statistical rigor. This makes the Kalman filter a robust solution for embedded systems and autonomous agents that must operate reliably in uncertain and evolving environments. By continuously balancing prior knowledge against new observations, it achieves an optimal trade-off between precision and robustness.

Although this paper focuses on the standard linear Kalman filter, it is important to note that several extensions exist to accommodate nonlinear systems. Among the most widely used are the extended Kalman filter (EKF), which linearises the system around the current estimate, and the unscented Kalman filter (UKF), which uses deterministic sampling to capture nonlinear effects more accurately.

2.2.2 The Kalman filter cycle

The Kalman filter operates through a recursive two-step process consisting of a prediction phase followed by a measurement update. This cycle enables the filter to refine its estimate of the system state over time by combining a mathematical model of the system's dynamics with noisy observations.

- 1 Prediction (time update): in this step, the current state estimate and the system's dynamic model are used to project the state forward in time. Along with this predicted state, the associated uncertainty (covariance) is also propagated, typically increasing due to process noise. For example, consider a mobile robot estimated to be at 10 metres with a velocity of 2 m/s. After one second, the predicted position

becomes 12 metres, but the uncertainty increases as the robot may have deviated due to noise or unmodelled dynamics.

- 2 Measurement update (correction): upon receiving a new measurement, the filter compares it to the predicted value to compute the innovation, i.e., the difference between observed and predicted measurements. The state estimate is then updated by adding a weighted portion of this innovation to the predicted state. The weighting factor is the Kalman gain, which is determined based on the relative uncertainties of the prediction and the measurement. The uncertainty of the updated estimate is also revised, typically decreasing due to the integration of new information. Continuing the previous example, suppose the GPS reading at the next time step is 11.5 metres. The innovation is -0.5 metres. With a Kalman gain of 0.8 (indicating high trust in the measurement), the correction is -0.4 , yielding an updated position estimate of 11.6 metres. This estimate is more precise than either the prediction or measurement alone, as it synthesises information from both sources.

The recursive update can be summarised by the following expression:

$$\text{New Estimate} = \text{Predicted Estimate} + \text{Kalman Gain} \times \text{Prediction Error} \quad (1)$$

The Kalman gain is derived to minimise the a posteriori error covariance, making the Kalman filter an optimal estimator under the assumptions of linear system dynamics and Gaussian noise. When the prediction is uncertain but the measurement is reliable, the filter favours the measurement; conversely, if the model is highly accurate and the sensor data is noisy, greater weight is given to the prediction.

The standard Kalman filter relies on a set of key assumptions that define the scope within which it can operate optimally. First and foremost, it assumes that the system dynamics are linear, meaning the evolution of the system state over time can be expressed through linear equations involving matrices that act on the current state and any control inputs. In cases where the system exhibits nonlinear behaviour, linear approximations such as those used in the EKF may be employed; however, such extensions are beyond the scope of this introductory discussion.

A second core assumption concerns the statistical nature of the noise affecting the system. The Kalman filter presumes that both process noise (arising from system dynamics) and measurement noise (arising from sensor inaccuracies) are normally distributed with zero mean and known covariance. Gaussian noise is a common and practical assumption, not only because it approximates many real-world random processes, but also due to its favourable mathematical properties, most notably the fact that the Gaussian distribution is fully characterised by its mean and covariance.

Under the dual assumptions of linear system dynamics and Gaussian noise, the Kalman filter ensures that the posterior distribution of the state estimate remains Gaussian at all times. Consequently, the filter need only maintain and update the mean and covariance of the estimate, quantities that are propagated and corrected using compact matrix equations. This mathematical efficiency allows the Kalman filter to operate in real time while accurately capturing both the estimate and its associated uncertainty.

When these assumptions are violated such as in systems with strongly nonlinear behaviour or noise that deviates significantly from a Gaussian distribution, the standard Kalman filter may yield suboptimal results. In such scenarios, more sophisticated

approaches like the EKF, UKF, or Particle Filter may be appropriate. Nevertheless, in a wide range of practical applications, linear models with approximately Gaussian noise provide sufficiently accurate representations, making the Kalman filter a robust and effective solution.

In summary, Kalman filtering is a principled method for estimating the state of dynamic systems in the presence of uncertainty. By integrating model-based predictions with real-time measurements, it provides a statistically optimal estimate under its foundational assumptions. The filter is computationally efficient, conceptually elegant, and grounded in basic linear algebra and probability theory. This introduction has outlined the conceptual framework of the Kalman filter its prediction-correction cycle, underlying assumptions, and relevance to dynamic system estimation. With this foundation, we motivate its use in demonstrating our proof-of-concept MPUs, which are designed for real-time, high-fidelity sensor data fusion.

3 Experimental setup

This section describes the experimental setup for our proposed Kalman filtering methodology.

3.1 Hardware description and modular structure

To evaluate the feasibility of real-time Kalman filtering on digital hardware, we developed a modular hardware implementation of a one-dimensional discrete Kalman filter using Verilog HDL. The design was partitioned into small, reusable arithmetic components to support clarity, synthesis efficiency, and design flexibility. These modules were then composed into a single top-level unit that performed the Kalman filter operations per clock cycle. Figure 1 shows a visual representation of the entire data path.

The implementation includes the following core components:

- Addition module: computes signed addition of two fixed-point numbers; used in updating predicted error covariance and the state estimate.
- Subtraction module: computes signed subtraction; used in calculating the innovation term ($z - \hat{x}$) and error propagation.
- Multiplication module: multiplies two signed fixed-point values; used for applying the Kalman gain and updating the covariance.
- Division module: performs signed fixed-point division; used for computing the Kalman gain $K = P/(P + R)$.
- Top-level filter module: instantiates the above primitives to implement the Kalman prediction and correction steps.

Detailed specifications of each module, including interface definitions and internal logic behaviour, are provided in Tables 1 through 5. All datapaths are parameterisable by bit width and are synchronised via a global clock. Control logic handles reset conditions, measurement input, and state output capture.

Table 1 Inputs and outputs of the addition module

<i>Port</i>	<i>Direction</i>	<i>Width</i>	<i>Description</i>
a	Input	Signed [W-1:0]	1st operand (fixed-point)
b	Input	Signed [W-1:0]	2nd operand (fixed-point)
res	Output	Signed [W-1:0]	Sum of a + b

Table 2 Inputs and outputs of the subtraction module

<i>Port</i>	<i>Direction</i>	<i>Width</i>	<i>Description</i>
a	Input	Signed [W-1:0]	Minuend
b	Input	Signed [W-1:0]	Subtrahend
result	Output	Signed [W-1:0]	Result of a-b

Table 3 Inputs and outputs of the multiplication module

<i>Port</i>	<i>Direction</i>	<i>Width</i>	<i>Description</i>
a	Input	Signed [W-1:0]	First multiplicand
b	Input	Signed [W-1:0]	Second multiplicand
res	Output	Signed [2W-1:0]	Product of a * b

3.2 Fixed-point number representation

The system operates entirely on fixed-point signed binary numbers to enable efficient synthesis on hardware platforms such as FPGAs. Each value is represented using a two's complement format with a fixed binary point.

Table 4 Inputs and outputs of the division module

<i>Port</i>	<i>Direction</i>	<i>Width</i>	<i>Description</i>
Numerator	Input	Signed [W-1:0]	Dividend
Denominator	Input	Signed [W-1:0]	Divisor
res	Output	Signed [W-1:0]	Result of division

Table 5 Inputs and outputs of the Kalman filter top module

<i>Port</i>	<i>Direction</i>	<i>Width</i>	<i>Description</i>
z	Input	Signed [W-1:0]	Measurement input
Q	Input	Signed [W-1:0]	Process noise covariance
R	Input	Signed [W-1:0]	Measurement noise covariance
clk	Input	1 bit	Clock signal
rst	Input	1 bit	Active-high synchronous reset
x	Output	Signed [W-1:0]	Estimated state
P	Output	Signed [W-1:0]	Updated error covariance

Let W denote the total bit width of a value. The representation allocates:

- 1 bit for sign

- I bits for integer part
- $F = W - I - 1$ bits for fractional part.

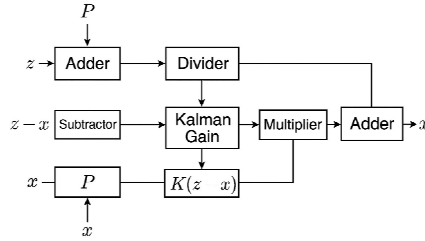
Typical configurations used in this work allocated $W = 16$ bits with $I = 5$ and $F = 10$, allowing for a dynamic range of approximately $[-16, +16]$ with a resolution of $2^{-10} \approx 0.00098$. Overflow conditions were not saturated and were monitored in simulation. Proper scaling of operands was performed before multiplication and division to maintain numerical accuracy.

3.3 Simulation and verification

Each arithmetic component was independently tested using dedicated testbenches that applied exhaustive or corner-case input values. Simulation waveforms were examined to validate correctness, signed arithmetic behaviour, and precision of fixed-point computations.

The full Kalman filter was tested using a top-level testbench that injected a sequence of noisy measurements and verified the convergence of the estimated state and the reduction of the error covariance over time. The testbench also generated clock and reset signals and printed the output estimates after each update step.

Figure 1 Datapath for Kalman filter



3.4 Input data generation

The true input values used in our experiments were generated through a combination of controlled synthetic simulations and empirical data trends derived from public datasets. For each application domain, the data was prepared as follows:

- Displacement and distance: a linear motion profile was synthesised in Python to simulate constant-velocity movement. This reflects standard conditions found in robotics, AVs, or inertial navigation systems.
- Temperature and humidity: true values were modelled on real-world climatological patterns sourced from the US National Weather Service. Synthetic data was smoothed and interpolated to mimic daily environmental fluctuations.
- Wind speed: time-varying wind profiles were simulated using reference curves from the NOAA Wind Resource Atlas. Gaussian noise was added to emulate fluctuations commonly observed in outdoor sensing conditions.

- LiDAR distance: a steadily increasing range profile was generated to mimic the approach of an object toward a LiDAR sensor. The profile simulates motion typical in SLAM and obstacle-avoidance applications.
- Battery state-of-charge (SOC): the SOC curve was created by modelling a constant current discharge based on normalised lithium-ion battery specifications. Manufacturer data and academic literature were referenced to match typical decline rates.

3.5 *Tools and environment*

- Simulator: Icarus Verilog (iverilog) for RTL simulation.
- Waveform viewer: GTKWave for analysing signal behaviour.
- Development environment: macOS with Makefile-based simulation flow.
- Target deployment: RTL design is synthesisable and intended for FPGA platforms such as Xilinx or Intel (Altera).

4 **Experimental results**

To evaluate the performance of the Kalman filter implementation, we conducted six simulated experiments across different domains. Each experiment consisted of a known true value profile with artificially added Gaussian noise, simulating sensor errors. The Kalman filter was then applied to these noisy measurements to generate corrected values. The experiments include motion tracking (displacement and distance), environmental monitoring (temperature and humidity), and embedded system scenarios (wind speed, LiDAR, and battery SOC).

For all experiments, Gaussian noise with zero mean and application-specific standard deviation was added to the true values to simulate sensor noise. These noisy signals were then passed through the Kalman filter for correction and smoothing. This process allows reproducible, domain-relevant validation of the filtering framework.

4.1 *Displacement and distance tracking*

This experiment modelled an object moving linearly from 0 to 100 metres. The raw displacement measurements were corrupted with Gaussian noise ($\sigma = 5$ m). The Kalman filter produced corrected values closely matching the true distance ($\sigma_n \approx 1$ m). Figure 2 shows a graph of the raw and corrected displacement values versus time.

4.2 *Temperature and humidity estimation*

True temperature and humidity values were linearly increased over time. Raw measurements were affected by Gaussian noise (standard deviations: 2.5°C and 3.5% respectively). The filter significantly reduced this noise while preserving the overall trends, resulting in temperature $\sigma_n = 0.5^\circ\text{C}$ and humidity ($\sigma_n = 0.8\%$). Figure 3 plots the

raw and corrected temperature vs. time and Figure 4 shows the plot of raw and corrected humidity vs. time.

4.3 Wind speed estimation

We simulated increasing wind speeds from 5 to 15 m/s with added noise ($\sigma = 2.5$ m/s). The Kalman filter reduced this to $\sigma \approx 0.6$ m/s, producing more stable results appropriate for weather stations or turbine control systems. Figure 5 shows the plot for raw and corrected wind speed vs. time.

Figure 2 Raw vs. corrected displacement data over time (see online version for colours)

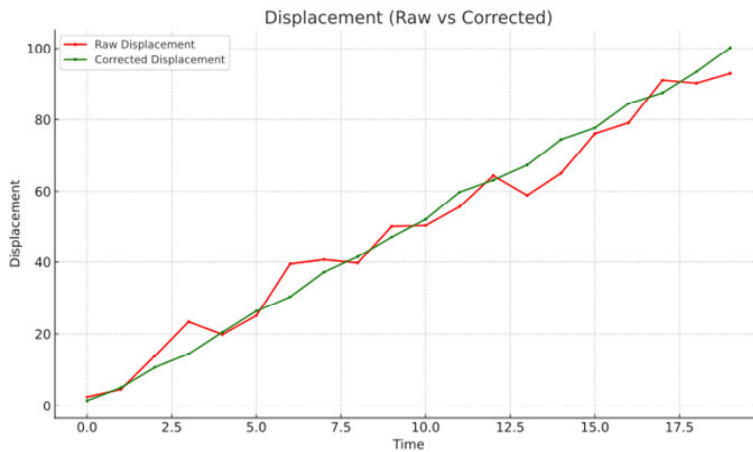
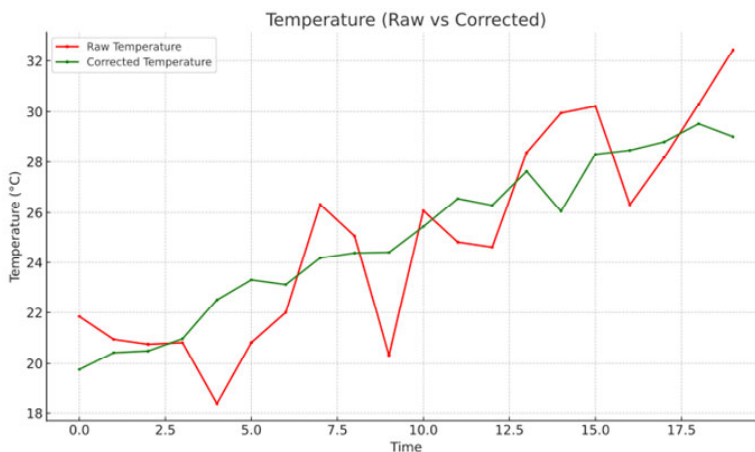


Figure 3 Raw vs. corrected temperature readings (see online version for colours)



4.4 LiDAR distance measurement

A linear distance profile from 1 to 10 m was used, with synthetic noise mimicking typical LiDAR variance ($\sigma = 0.8$ m). The filtered output showed excellent agreement with the

expected path, suitable for robotic applications. Figure 6 shows the plot for raw and corrected LiDAR distance vs. time.

4.5 Battery SOC estimation

Battery SOC was simulated to decline from 100% to 20% with noise ($\sigma = 4\%$) to reflect real-world sensing issues. The Kalman filter produced a smoothed profile, reducing the noise to $\sigma \approx 1\%$. Figure 7 shows the plot for raw and corrected battery SOC vs. time.

Figure 4 Raw vs. corrected humidity readings (see online version for colours)

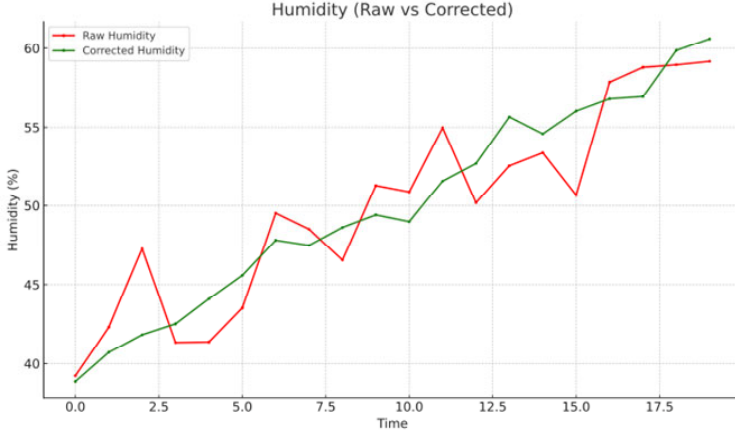
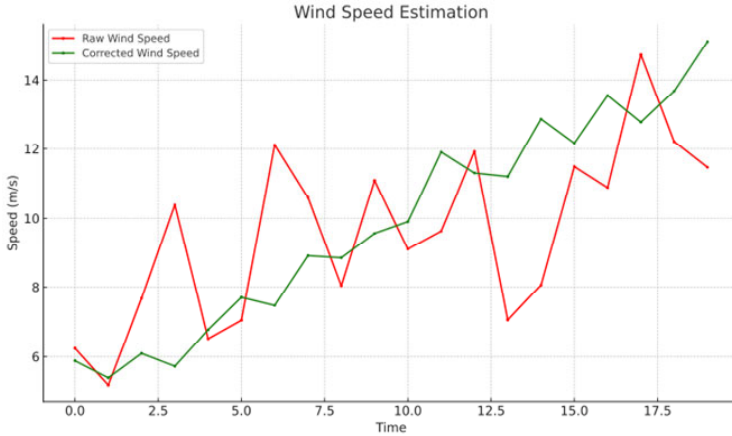


Figure 5 Raw vs. corrected wind speed estimates over time (see online version for colours)



4.6 Noise reduction calculation

To quantify the effectiveness of the Kalman filter, we define the percentage improvement in sensor accuracy based on the reduction in standard deviation (σ) between the raw and filtered signals. Equation (2) shows the calculation.

$$\text{Improvement}(\%) = \left(1 - \frac{\sigma_{\text{filtered}}}{\sigma_{\text{raw}}} \right) * 100 \quad (2)$$

For example, in the displacement tracking experiment, improvement % was calculated in the following manner:

$$\sigma_{\text{raw}} = 5.0 \text{ m}$$

$$\sigma_{\text{filtered}} = 1.0 \text{ m}$$

$$\text{Improvement} = \left(1 - \frac{1.0}{5.0} \right) * 100 = 80.0\%$$

This calculation was repeated across all six experiments, yielding an average improvement of 77.2%. Equation (3) shows the exact calculation.

$$\text{Avg. Improvement} = \frac{80 + 80 + 77.1 + 76 + 75 + 75}{6} \approx 77.2\% \quad (3)$$

Figure 6 Raw vs. corrected LiDAR distance readings (see online version for colours)

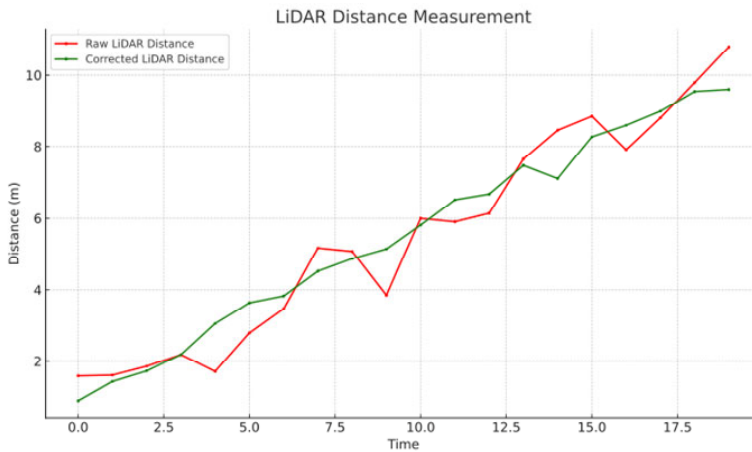
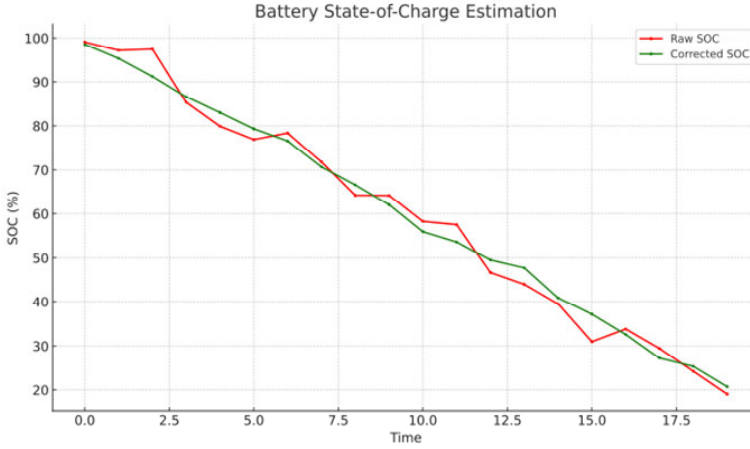


Table 6 Summary of results

<i>Metric</i>	<i>Raw Std. Dev. (σ_n)</i>	<i>Filt. Std. Dev. (σ_k)</i>	<i>Improve %</i>
Displacement	5.0	1.0	80.0%
Temperature	2.5	0.5	80.0%
Humidity	3.5	0.8	77.1%
Wind speed	2.5	0.6	76.0%
LiDAR distance	0.8	0.2	75.0%
Battery charge	4.0	1.0	75.0%

Table 6 summarises the improvement achieved by our Kalman filter implementation for each individual case.

Figure 7 Raw vs. corrected battery SOC over time (see online version for colours)

4.7 Summary of improvements

Across all six domains, the Kalman filter consistently improved sensor accuracy by reducing measurement noise. The highest improvement was observed in displacement and temperature measurements (both 80%), where the noise profile was relatively high but smooth. The lowest improvements (around 75%) occurred in battery SOC and LiDAR distance, which likely reflects either a lower signal-to-noise ratio or more abrupt changes in the input signal.

Notably, the average improvement across all experiments was approximately 77.2%, underscoring the robustness and domain-independence of the implemented Kalman filter. This consistent performance across diverse sensing modalities suggests strong generalisability and supports its deployment in embedded and real-time systems.

5 Future work

To expand the capabilities of the proposed MPU framework, our future efforts will focus on three key areas: multi-stream sensor data filtering, dynamically reconfigurable mathematical pre-processing, and integrated fault tolerance mechanisms.

First, we aim to implement selective and adaptive filtering mechanisms across multiple simultaneous sensor streams. This would allow the MPU to handle heterogeneous sensor inputs (e.g., environmental, inertial, and visual data) and apply custom digital filters such as Kalman, low-pass, and moving-average filters in parallel pipelines. The system will incorporate context-aware filter switching based on sensor health, signal quality, or dynamic environmental changes.

Second, we will develop a reconfigurable pre-processing architecture within the FPGA logic. This will enable runtime switching of mathematical operations, such as coordinate transformations, normalisation, sensor fusion, and temporal smoothing based on configuration commands or adaptive agent logic. Such dynamic reconfiguration is essential for ambient intelligence systems operating under variable conditions and resource constraints.

Third, we plan to integrate fault detection and tolerance mechanisms, including signal validity checks, drift estimation, and fallback pathways. MPUs will be capable of identifying sensor anomalies through statistical models and injecting synthetic signals or reweighted sensor fusion models to preserve system stability and inference accuracy.

To validate these improvements, we will synthesise and deploy our Verilog design on a Digilent Arty A7 FPGA development board. This platform offers sufficient logic resources and I/O capability to interface with commonly available commercial off-the-shelf (COTS) sensors. Initial hardware testing will include: HTU31D for temperature and humidity, BMP388 for barometric pressure, SGP30 for VOC (volatile organic compounds), and additional I2C or SPI-based sensors for expanding test coverage.

By demonstrating the MPU's real-time performance, adaptability, and robustness with real-world sensor hardware, we aim to further validate its suitability as a core enabler of embedded ambient intelligence.

6 Conclusions

This work proposes a novel hardware implementation termed MPUs whose primary goal is to perform real-time mathematical pre-processing and signal conditioning to improve the quality of acquired ambient data. The concept is validated through hardware-based implementation of the Kalman filter using Verilog, targeting practical sensor noise reduction in real-time systems. Our modular design was validated through multiple simulated experiments across domains such as motion tracking, environmental monitoring, and power system management. In each case, the filter consistently reduced signal noise by over 77%, illustrating its versatility and effectiveness. The use of fixed-point arithmetic and hardware-friendly operations ensures suitability for deployment in resource-constrained edge environments. This aligns with the growing need for localised, intelligent computation in domains such as IoT, robotics, and autonomous systems. Future work may explore adaptive tuning of Kalman parameters in hardware, tighter integration with FPGA-based sensor arrays, and quantifying the energy-performance tradeoffs. Additionally, real-world validation using physical sensors and actuators will be an important next step.

The overall aim of the MPU is to accelerate the process of developing and deploying the next generation of AI-powered systems in emerging real-world applications such as humanoid robotics, autonomous driving, AR, and smart environments. They can also encourage new innovation in smart healthcare, distributed AI, and planetary aerobots. In addition, they also contribute to the advancement of fundamental research because of their low-cost and high-fidelity sensor data acquisition. By bridging the gap between theoretical physics and empirical observation, MPUs stand to accelerate discoveries in frontier research for fields such as cosmology and infodynamics, offering new insights into both the physical and informational fabric of reality.

Declarations

All authors declare that they have no conflicts of interest.

References

- Aarts, E. and Encarnação, J.L. (2006) *True Visions: The Emergence of Ambient Intelligence*, Springer, Berlin.
- Albert, K., Hirzberger, J., Busse, D., Lange, T., Kolleck, M., Fiethe, B., Suárez, D.O., Woch, J., Schou, J., Rodriguez, J.B. et al. (2018) 'Autonomous on-board data processing and instrument calibration software for the SO/PHI', *Software and Cyberinfrastructure for Astronomy V*, Vol. 10707, pp.177–185.
- Aracri, S., Giorgio-Serchi, F., Suaria, G., Sayed, M.E., Nemitz, M.P., Mahon, S. and Stokes, A.A. (2021) 'Soft robots for ocean exploration and offshore operations: a perspective', *Soft Robotics*, Vol. 8, No. 6, pp.625–639.
- Ben Sada, A., Naouri, A., Khelloufi, A., Dhelim, S. and Ning, H. (2023) 'A context-aware edge computing framework for smart internet of things', *Future Internet*, Vol. 15, No. 5, p.154.
- Chen, S., Zong, S., Chen, T., Huang, Z., Chen, Y. and Labi, S. (2023) 'A taxonomy for autonomous vehicles considering ambient road infrastructure', *Sustainability*, Vol. 15, No. 14, p.11258.
- De Paola, A., Ferraro, P., Gaglio, S., Re, G.L. and Das, S.K. (2016) 'An adaptive Bayesian system for context-aware data fusion in smart environments', *IEEE Transactions on Mobile Computing*, Vol. 16, No. 6, pp.1502–1515.
- Elfes, A., Hall, J.L., Kulczycki, E.A., Clouse, D.S., Morfopoulos, A.C., Montgomery, J.F., Cameron, J.M., Ansar, A. and Machuzak, R.J. (2008) 'An autonomy architecture for aerobot exploration of the Saturnian moon Titan', *2008 IEEE Aerospace Conference*, pp.1–9.
- Elfes, A., Montgomery, J.F., Hall, J.L., Joshi, S., Payne, J. and Bergh, C.F. (2005) *Autonomous Flight Control for a Titan Exploration Aerobot*, Citeseer.
- Euclid Collaboration et al. (2025) 'Euclid: III. The NISP Instrument', *Astronomy & Astrophysics*, Vol. 697, Article A3, DOI: 10.1051/0004-6361/202450786.
- Fang, W., Zheng, L., Deng, H. and Zhang, H. (2017) 'Realtime motion tracking for mobile augmented/virtual reality using adaptive visualinertial fusion', *Sensors*, Vol. 17, No. 5, p.1037.
- Forster, J. (2025) [online] <https://www.apheris.com/resources/blog/federated-learning-anddata-mesh> (accessed 26 September 2026).
- Gerakakis, S., Brilenkov, M., Ieronymaki, M., San, M., Watts, D.J., Andersen, K.J., Aurlien, R., Banerji, R., Basyrov, A., Bersanelli, M., Bertocco, S., Carbone, M., Colombo, L.P.L., Eriksen, H.K., Eskilt, J.R., Foss, M.K., Franceschet, C., Fuskeland, U., Galeotta, S., Galloway, M., Gjerløw, E., Hensley, B., Herman, D., Iacobellis, M., Ihle, H.T., Jewell, J.B., Karakci, A., Keihänen, E., Keskitalo, R., Lunde, J.G.S., Maggio, G., Maino, D., Maris, M., Paradiso, S., Reinecke, M., Stutzer, N-O., Suur-Uski, A-S., Svalheim, T.L., Tavagnacco, D., Thommesen, H., Wehus, I.K. and Zacchei, A. (2023) *From BeyondPlanck to Cosmoglobe: Open Science, Reproducibility, and Data Longevity*, <https://arxiv.org/abs/2205.11262>.
- Hazlewood, W.R and Coyle, L. (2009) 'On ambient information systems: challenges of design and evaluation', *International Journal of Ambient Computing and Intelligence (IJACI)*, Vol. 1, No. 2, pp.1–12.
- Kumar, G.A., Patil, A.K., Kang, T.W. and Chai, Y.H. (2019) 'Sensor fusion based pipeline inspection for the augmented reality system', *Symmetry*, Vol. 11, No. 10, p.1325.
- Li, G., Wong, T-W., Shih, B., Guo, C., Wang, L., Liu, J., Wang, T., Liu, X., Yan, J., Wu, B. et al. (2023) 'Bioinspired soft robots for deep-sea exploration', *Nature Communications*, Vol. 14, No. 1, p.7097.
- Liu, L., Li, H. and Gruteser, M. (2019) 'Edge assisted realtime object detection for mobile augmented reality', *The 25th Annual International Conference on Mobile Computing and Networking*, pp.1–16.

- Lysakowski, M., Zywanowski, K., Banaszczyk, A., Nowicki, M.R., Skrzypczynski, P. and Tadeja, S.K. (2023) 'Real-time onboard object detection for augmented reality: enhancing head-mounted display with yolov8', *2023 IEEE International Conference on Edge Computing and Communications (EDGE)*, pp.364–371.
- Matos, F., Bernardino, J., Durães, J. and Cunha, J. (2024) 'A survey on sensor failures in autonomous vehicles: challenges and solutions', *Sensors*, Vol. 24, No. 16, p.5108.
- Modupe, O.T., Otitoola, A.A., Oladapo, O.J., Abiona, O.O., Oyeniran, O.C., Adewusi, A.O., Komolafe, A.M. and Obijuru, A. (2024) 'Reviewing the transformational impact of edge computing on real-time data processing and analytics', *Computer Science & IT Research Journal*, Vol. 5, No. 3, pp.603–702.
- Nock, K., Jones, J. and Rodriguez, G. (1996) 'Planetary aerobots – a program for robotic balloon exploration', *34th Aerospace Sciences Meeting and Exhibit*, p.355.
- Orf, D. (2025) [online] <https://www.popularmechanics.com/science/environment/a64378430/simulationtheory-new-physics-law> (accessed 26 September 2026).
- Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D. (2013) 'Context aware computing for the internet of things: a survey', *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, pp.414–454.
- Qu, J., Xu, Y., Li, Z., Yu, Z., Mao, B., Wang, Y., Wang, Z., Fan, Q., Qian, X. and Zhang, M. et al. (2024) 'Recent advances on underwater soft robots', *Advanced Intelligent Systems*, Vol. 6, No. 2, p.2300299.
- Reichert, H., Lang, L., Rosch, K., Bogdoll, D., Doll, K., Sick, B., Reuss, H-C., Stiller, C. and Zollner, J.M. (2021) 'Towards sensor data abstraction of autonomous vehicle perception systems', *2021 IEEE International Smart Cities Conference (ISC2)*, pp.1–4.
- Roychoudhury, A., Khorshidi, S., Agrawal, S. and Bennewitz, M. (2023) 'Perception for humanoid robots', *Current Robotics Reports*, Vol. 4, No. 4, pp.127–140.
- Silver, D. and Sutton, R. (2025) [online] <https://storage.googleapis.com/deepmind-media/Era-ofExperience%20The%20Era%20of%20Experience%20Paper.pdf> (accessed 26 September 2026).
- Vopson, M. (2025) 'On the second law of infodynamics from cosmological thermodynamics', *IPI Letters*, Vol. 3, No. 1, pp.N6–N9.
- Vopson, M.M. (2019) 'The mass-energy-information equivalence principle', *AIP Advances*, Vol. 9, No. 9, pp.095206-1–095206-4, Article 095206.
- Weilbacher, P.M., Palsa, R., Streicher, O., Bacon, R., Urrutia, T., Wisotzki, L., Conseil, S., Husemann, B., Jarno, A. and Kelz, A. et al. (2020) 'The data processing pipeline for the MUSE instrument', *Astronomy & Astrophysics*, Vol. 641, Article A28.
- Wikipedia (2026a) [online] https://en.wikipedia.org/wiki/Edge_computing (accessed 26 September 2026).
- Wikipedia (2026b) [online] https://en.wikipedia.org/wiki/Wave_Glider (accessed 26 September 2026).
- Zambelli, M., Cully, A. and Demiris, Y. (2020) 'Multimodal representation models for prediction and control from partial information', *Robotics and Autonomous Systems*, Vol. 123, Article 103312.
- Zhang, Q., Zhang, Z., Cui, W., Sun, J., Cao, J., Guo, Y., Han, G., Zhao, W., Wang, J. and Sun, C. (2025) *Humanoidpano: Hybrid Spherical Panoramic-lidar Cross-modal Perception for Humanoid Robots*, arXiv preprint arXiv:2503.09010.
- Zhang, Y., Carballo, A., Yang, H. and Takeda, K. (2023) 'Perception and sensing for autonomous vehicles under adverse weather conditions: a survey', *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 196, p.146177.