ISSN: 2584-0495 Vol. 3, Issue 9, pp. 1747-1753

International Journal of Microsystems and IoT

ISSN: (Online) Journal homepage: https://www.ijmit.org

A Digital Twin-Enabled Parking Assist System Using Raspberry Pi and AWS Cloud Services

Shweta Chaudhary, Vinaya Gohokar

Cite as: Chaudhary, S., & Gohokar, V. (2025). A Digital Twin-Enabled Parking Assist System Using Raspberry Pi and AWS Cloud Services. International Journal of Microsystems and IoT, 3(9), 1747–1753. https://doi.org/10.5281/zenodo.18222420

International Journal of Microsystems and IoT
Vol.3, Issue 9, pp.1747-1753; DOI: https://doi.org/10.5281/zenodo.18222420

A Digital Twin-Enabled Parking Assist System Using Raspberry Pi and AWS Cloud Services

Shweta Chaudhary, Vinaya Gohokar
Dept. of Computer Science and Engineering Faculty of Engineering and Technology, Datta Meghe Institute of Higher Education and Research, Wardha, Maharashtra, India
E-mail: dikshanishane6@gmail.com, chetanp.feat@dmiher.edu.in

ABSTRACT
The concept of the Digital Twin (DT) has emerged as a transformative paradigm in intelligent system design, offering the ability to mirror and interact with physical assets in real time. This paper presents the development and demonstration of a DT-enabled IoT-based parking assist framework that closely integrates local sensing, mechanical actuation, and cloud-based analytics and visualization. The core implementation employs a Raspberry Pi 3B+ coupled with an ultrasonic motion sensor, DC and servo motor actuators, and AWS IoT services to enable seamless communication between the edge and cloud. Sensor data is transmitted to AWS IoT Core using the MQTT protocol for secure, low-latency exchanges, while AWS IoT TwinMaker is used to construct an interactive and simplified digital twin representation of the complete parking assist system. A closed feedback loop allows the cloud to transmit operational commands back to the edge device, enabling responsive and adaptive control. The implementation serves as a proof-of-concept, demonstrating the viability of cloud-linked DT architectures for autonomous and semi-autonomous parking assistance applications and highlighting the benefits of integrating scalable edge computing with advanced cloud ecosystems.

KEYWORDS
Digital Twin, IoT, Raspberry Pi, AWS IoT Core, AWS IoT TwinMaker, MQTT, Ultrasonic Sensor, DC Motor, Servo Motor.

1. INTRODUCTION

In recent years, digital twin technology—commonly described as a comprehensive, data-driven virtual mirror of a real-world entity—has emerged as a powerful method for improving system performance, safety, and predictive decision-making [1]. Unlike static models, a DT continuously synchronizes with its physical counterpart, updating its parameters and behaviors in real time using live data from sensors, control systems, and environmental inputs [2,3]. These synchronized models are applied across a wide spectrum of industries, including aerospace, healthcare, manufacturing, energy systems, urban infrastructure, and mobility services [4,5]. By enabling predictive analytics, what-if simulations, and closed-loop operational control, DTs offer significant pathways to improving efficiency, reducing maintenance costs, and optimizing resource allocation [6,7]. Within transportation systems, particularly in the realm of advanced parking assist technologies, DT integration provides direct benefits such as improved spatial optimization, enhanced driver safety, and reduced search times for available parking spaces [8]. Urban parking management often demands continuous monitoring of multiple sensor streams, robust communication channels, and intelligent control algorithms—requirements that map naturally onto Digital Twin architectures [9].

Recent advances in edge computing hardware, low-latency communication protocols, cloud platform capabilities, and AI-based decision engines have made it feasible to deploy cost-efficient DT-based parking assist solutions at scale [9,10]. By modeling vehicles, surrounding infrastructure, and environmental influences simultaneously, such systems can enable precise maneuvering and space occupancy prediction [11]. Nevertheless, the deployment of such architectures still presents challenges, including ensuring strict real-time responsiveness, preserving user privacy, achieving accurate sensor fusion, and ensuring interoperability across diverse IoT and vehicular platforms [12,13,14,15]

The primary objective of this work is to design and implement a Digital Twin–enabled IoT-based parking assist framework that integrates local sensing, mechanical actuation, and cloud-based analytics to facilitate real-time monitoring, decision-making, and control for autonomous and semi-autonomous parking.The proposed framework leverages a Raspberry Pi 3B+ with ultrasonic sensors, DC and servo motors for local perception and actuation, and AWS IoT services—including IoT Core, TwinMaker, Timestream DB, Lambda, and SageMaker—for secure communication, data processing, visualization, and decision intelligence. The system demonstrates a closed feedback loop between edge and cloud, highlighting the potential of scalable DT architectures for future intelligent transportation and smart parking applications.

II. RELATED WORK

Various sensor technologies have been explored for parking digital twins (DTs), each with distinct operating principles and applications. Ultrasonic sensors operate on sound wave reflection to measure proximity and occupancy, offering low cost and easy installation, though with limited range and environmental noise sensitivity [27]. Magnetic sensors detect disturbances in magnetic fields caused by vehicle metals, enabling low-power space detection but facing challenges with non-metallic objects [28]. Camera-based vision systems, coupled with AI processing, provide rich data for vehicle classification and behavior analysis, yet require computationally intensive image processing and raise privacy concerns [29]. Radar sensors utilize radio wave reflection and Doppler effect to measure speed, distance, and obstacles, functioning reliably in low visibility but at higher cost and integration complexity [30]. Pressure sensors detect vehicle presence through weight measurement, offering a simple binary output but requiring careful calibration [31]. LiDAR systems employ laser scanning to generate high-accuracy 3D spatial maps, supporting advanced mapping and autonomous navigation, though their high cost limits widespread adoption [32].

III. PARKING ASSIST SYSTEMS

A. Steering Assisted Parking

A steering assisted parking system displays helpful driving instructions on the dashboard as you park. It will give you cues about the best time to speed up, slow down, and shift gears. The car will steer itself and maneuver into the parking space [16].

B. Park-and-Exit Assistance

A park-and-exit assistance system adds the capability to guide you out of the parking space, making it easier to leave the slot if your vehicle got boxed in while you were away. Otherwise, it works the same as steering assisted parking [16].

C. Fully Automated Parking

This type of park assist can park your car for you. It can even control the accelerator and brake pedals. While fully automated park assist comes with many safety features, you might prefer to stay in control of your vehicle [16].

D. Automated Parallel Parking Systems

Automated parallel parking systems use sensors and cameras to detect available parking spaces and guide the vehicle into a parallel parking spot. The system takes control of the steering while the driver operates the accelerator and brake pedals. This technology eliminates the need for complex manoeuvres and reduces the chances of scraping the vehicle against curbs or other parked cars [17].

E. Perpendicular Parking Systems

Perpendicular parking systems assist drivers in parking their vehicles perpendicularly. Similar to automated parallel parking systems, these systems utilise sensors and cameras to detect obstacles and guide the driver into the parking spot. Perpendicular parking systems are particularly useful in tight spaces where manoeuvring can be challenging [17].

F. 360-Degree Camera Systems

360-degree camera systems provide drivers with a comprehensive view of the vehicle and its surroundings, aiding in parking and manoeuvring in crowded areas . By

IV. SYSTEM OVERVIEW

A DT consists of three core components: the physical entity, the virtual representation, and the data communication interface between them. [18]. In parking systems, these components map directly onto the physical vehicle and sensors (e.g., cameras, radar, LiDAR), the virtual parking model, and the communication systems (e.g., IoT and edge computing) that maintain synchronization [19].

1. Physical System

The Physical System represents the tangible hardware components responsible for sensing, data acquisition, and communication.

a) Sensors

The sensor layer comprises measurement devices such as ultrasonic rangefinders and imaging modules (e.g., USB camera) for environmental perception. These sensors capture physical parameters, including obstacle distance and spatial imagery, which are essential for parking assistance functionalities.

b) Data Acquisition Layer

This layer interfaces directly with the sensors to digitize and preprocess raw signals. For ultrasonic sensors, analog echo pulse timings are converted into precise distance measurements. For vision-based inputs, image frames are acquired in digital format. The objective of this layer is to transform low-level physical signals into structured digital datasets.

c) Data Communication Layer

The communication layer, implemented on the Raspberry Pi, packages the acquired sensor data into structured messages and transmits them to the cloud using secure protocols such

Table I: Comparison of existing Digital Twin based Parking systems

References	Sensor Types	Principle of Operations	Data Collected	Advantages	Disadvantages	Typical Use Cases in Parking DTs
[27]	Ultrasonic	Sound wave reflection	Proximity, space occupancy	Low-cost, short-range, easy to install	Limited range, sensitivity to environment noise	Spot-level detection of vehicle presence
[28]	Magnetic	Magnetic field disturbance from vehicle metals	Vehicle presence	Small size, low power	Sensitivity issues with non-metallic objects	Embedded in ground for space occupancy detection
[29]	Camera (Vision)	Image capture + AI processing	Vehicle type, space status, user behavior	Rich data, enables AI-based analytics	Requires image processing, privacy concerns	License plate recognition, spot monitoring
[30]	Radar	Radio wave reflection and Doppler	Speed, distance, obstacle detection	Works in low visibility	More expensive, data integration challenges	Vehicle tracking, movement detection in lots
[31]	Pressure Sensor	Measures weight/pressure applied on surface	Space occupancy	Simple binary signal	Sensitive to calibration	Detects car presence in individual parking bays
[32]	LiDAR	Laser-based scanning to generate 3D map	Detailed spatial layout	High accuracy	High cost	Advanced mapping, autonomous parking navigation

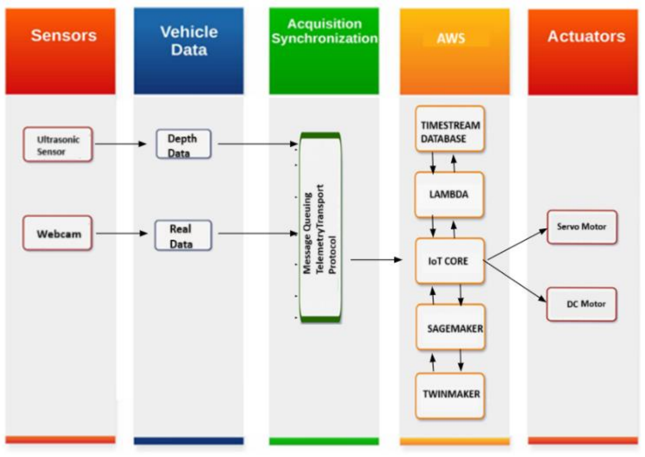


Fig 1 Framework for the proposed DT-enabled parking assist system

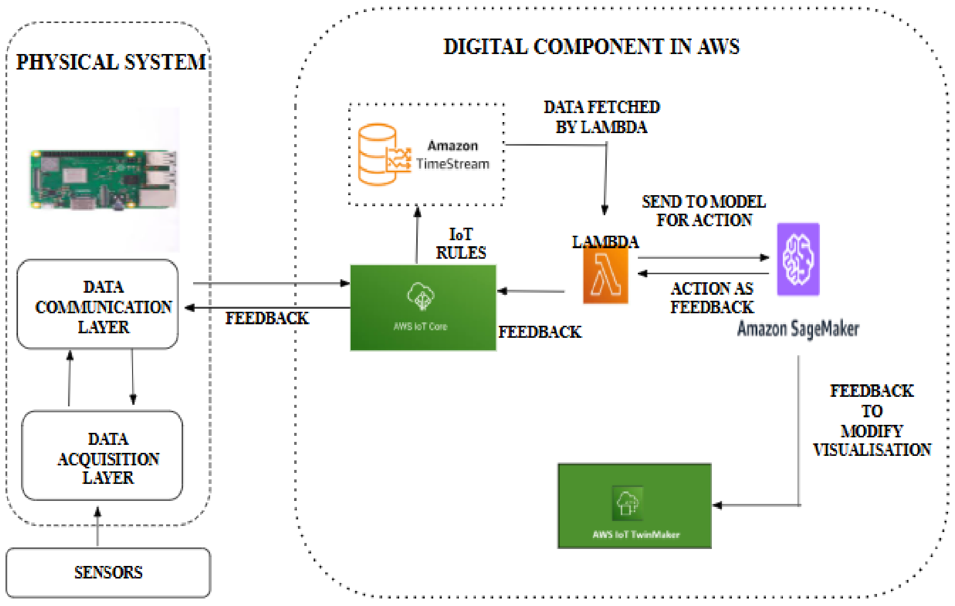


Fig 2 System architecture for proposed DT-enabled parking assist system

as MQTT over TLS. This layer also manages the reverse communication channel, whereby actuation commands from the cloud are relayed to the physical actuators (e.g., motors controlling vehicle movement)

2. Digital Component in AWS

The digital component is implemented within the AWS ecosystem, hosting the virtual representation of the physical system, AI inference modules, and historical data storage mechanisms.

a) AWS IoT Core

AWS IoT Core serves as the primary cloud gateway for bidirectional communication between the Physical System and the cloud infrastructure. It provides secure authentication, device identity management, and message routing capabilities. The service ensures low-latency delivery of sensor telemetry and feedback commands.

b) IoT Rules

IoT Rules define the conditional logic for routing incoming data streams to the appropriate AWS services. For example, real-time sensor data is forwarded to Amazon Timestream for archival, whereas selected datasets are transmitted to AWS Lambda for processing and inference triggering.

c) Amazon Timestream

Amazon Timestream is employed as a time-series database optimized for storing high-volume, timestamped sensor readings. This enables temporal analysis, trend identification, and performance monitoring of the physical system over extended periods.

d) AWS Lambda

AWS Lambda functions act as serverless computational units for on-demand data processing. Upon receiving new sensor updates, Lambda retrieves relevant datasets from Timestream, performs necessary preprocessing (e.g., normalization, filtering), and forwards the structured input to Amazon SageMaker for inference. Lambda also post-processes inference results for subsequent feedback transmission.

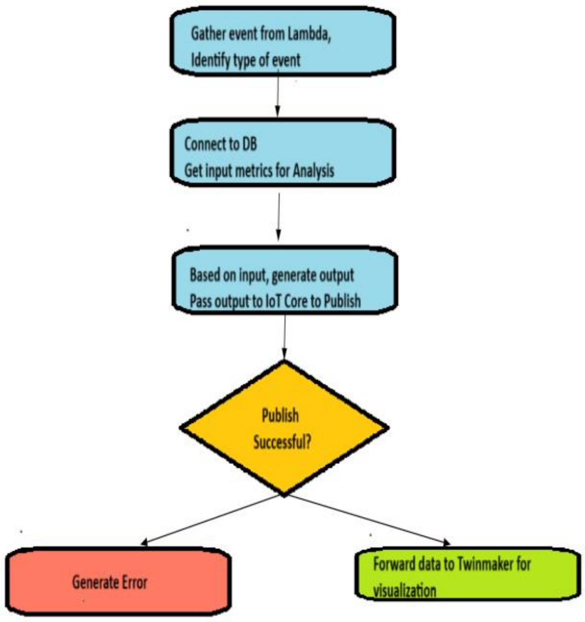

Gather event from Lambda, Identify type of event
Connect to DB Get input metrics for Analysis
Based on input, generate output Pass output to IoT Core to Publish
Publish Successful?
Generate Error
Forward data to Twinmaker for visualization

Fig 3 Flowchart for Lambda function

e) Amazon SageMaker

Amazon SageMaker hosts the machine learning inference model responsible for decision-making in the parking assistance task. The model analyzes the incoming sensory data and outputs recommended control actions (e.g., move forward, halt, adjust steering angle) to facilitate optimal parking maneuvers. These inferences are returned to AWS Lambda for communication back to the Physical System.

f) AWS IoT TwinMaker

AWS IoT TwinMaker constructs and maintains the virtual representation of the Physical System. It integrates real-time telemetry with the 3D visualization of the parking environment, thereby enabling remote monitoring and simulation. This module dynamically updates based on both the physical sensor data and inferred control actions, ensuring accurate synchronization between the physical and digital entities.

3. Feedback Mechanism

The architecture is underpinned by a continuous feedback loop. Sensor data flows from the Physical System to AWS IoT Core, through processing and inference pipelines, and returns as control commands to the Raspberry Pi. Simultaneously, visualization updates in IoT TwinMaker reflect the latest system state, ensuring the digital twin mirrors the physical environment in real time.

V. DIGITAL TWIN ARCHITECTURE FOR PARKING ASSISTANCE

The proposed solution integrates ultrasonic sensors, servo motors, DC motors, a Raspberry Pi, and a suite of AWS services to implement a smart parking system. The AWS services used include IoT Core, Timestream DB, SageMaker, Lambda, and TwinMaker, each contributing to data collection, processing, decision-making, and visualization. Precision of obstacle detection plays a critical role in ensuring the safety and accuracy of parking maneuvers.

The servo motor in the solution represents the steering mechanism of a vehicle. It receives feedback through the AWS IoT Core using an MQTT connection. Based on the feedback, the servo motor adjusts its rotation angle to steer the vehicle accordingly. Similarly, the DC motor represents the vehicle's engine. It also receives commands from IoT Core via MQTT, which direct the motor to switch on, switch off, or keep moving, effectively simulating engine operation.

AWS SageMaker hosts the decision-making model that governs the vehicle's actions. This model is deployed as an endpoint, allowing other services to make inference requests. The current model primarily accepts distance data from the ultrasonic sensor as input, though integration of spatial data captured via a webcam is underway to enhance performance. The output of the model includes operational commands such as "start," "stop," "keep moving," and steering angles for the servo motor.

AWS IoT Core manages secure communication between the Raspberry Pi and the AWS cloud. Security credentials include the certificates AmazonRootCA1.pem and certificate.pem.crt, as well as the private key private.pem.key. These credentials are embedded in the Raspberry Pi program to ensure encrypted and authenticated communication. Two IoT rules are defined for message routing: (i) SendToTimestream, which stores incoming data in the Timestream database along with a timestamp, and (ii) LambdaTrigger, which invokes a Lambda function for further processing. The Timestream DB component consists of a database named smartParkingDB with a table obstacle Distance. This table stores incoming distance readings and associated metadata, enabling both historical analysis and real-time inspection through queries. The Lambda function, named Interact with Model, is triggered by the LambdaTrigger IoT rule. This function connects to the SageMaker model endpoint, sends sensor input data for inference, and transmits the resulting decisions back to IoT Core. This enables near-real-time command generation for the actuators.

Finally, AWS TwinMaker provides a digital twin visualization of the parking lot environment. A workspace named Smart_Parking Workspace is created, containing a 3D scene of a parking lot and vehicle models. These assets are stored in an Amazon S3 bucket and linked within TwinMaker. The car model's rotation and translation properties are bound to real-time feedback from the SageMaker model, enabling the digital twin to mirror physical actions.

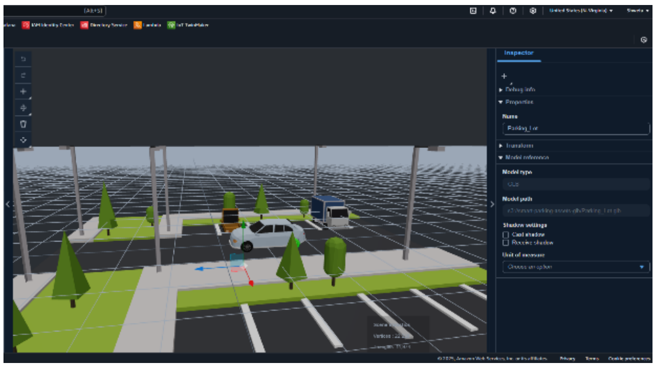


Fig 4:Parallel Parking representation in AWS

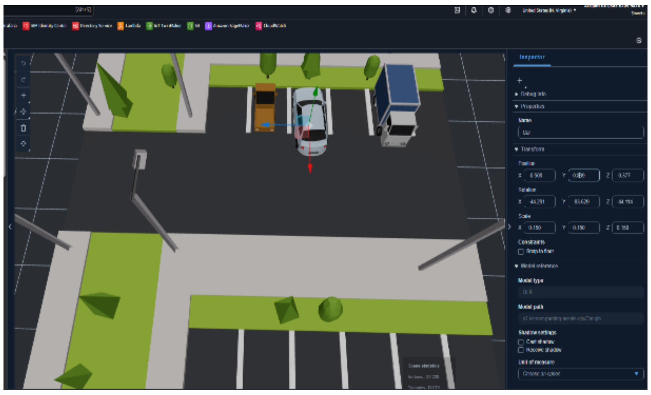


Fig5 Perpendicular Parking representation in AWS

VI. CHALLENGE AND FUTURE WORK

A. Integration with Smart Cities

Digital twins for parking assist systems are particularly relevant for smart cities, where they can support broader traffic management and resource allocation goals [24]. Integration with citywide data sources, such as traffic cameras and environmental sensors, will allow DTs to contribute to real-time urban mobility planning [24]. The adoption of DT-based parking assist systems in smart cities will require coordinated efforts to ensure compatibility and interoperability across systems [24].

B. Real-Time Decision Making and Predictive Analytics

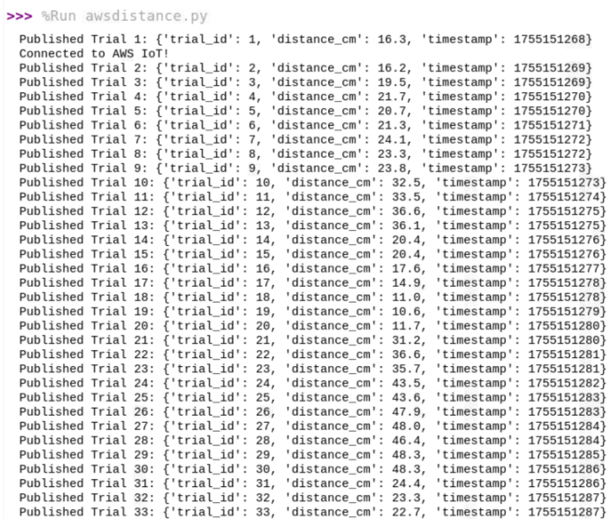
To make real-time decisions, DTs must process vast quantities of data efficiently. Future research should focus on developing more efficient AI models and edge-computing frameworks to meet these requirements. Additionally, predictive analytics can further optimize DT performance, allowing parking assist systems to anticipate future behaviors and respond proactively [25].

C. Enhancing Human-Machine Collaboration

In semi-autonomous parking systems, the collaboration between human drivers and DTs remains a key focus area. Interface improvements and predictive alerts for drivers will enhance the user experience. Further work is needed to improve system transparency, allowing drivers to understand and trust the DT's decisions [26].

VII. CONCLUSION

Digital twins represent a transformative approach to advancing parking assist technology. By creating real-time, adaptable, and predictive models, DTs can enhance the efficiency, safety, and user experience of autonomous and semi-autonomous parking systems. However, as technology progresses, it will require ongoing attention to scalability, security, and compliance. Integrating DT-based parking solutions within smart city infrastructures holds promise for broader applications, making parking systems more intelligent and responsive.



```
>>> %Run awsdistance.py
  Published Trial 1: {'trial_id': 1, 'distance_cm': 16.3, 'timestamp': 1755151268}
  Connected to AWS IoT!
  Published Trial 2: {'trial_id': 2, 'distance_cm': 16.2, 'timestamp': 1755151269}
  Published Trial 3: {'trial_id': 3, 'distance_cm': 19.5, 'timestamp': 1755151269}
  Published Trial 4: {'trial_id': 4, 'distance_cm': 21.7, 'timestamp': 1755151270}
  Published Trial 5: {'trial_id': 5, 'distance_cm': 20.7, 'timestamp': 1755151270}
  Published Trial 6: {'trial_id': 6, 'distance_cm': 21.3, 'timestamp': 1755151271}
  Published Trial 7: {'trial_id': 7, 'distance_cm': 24.1, 'timestamp': 1755151272}
  Published Trial 8: {'trial_id': 8, 'distance_cm': 23.3, 'timestamp': 1755151272}
  Published Trial 9: {'trial_id': 9, 'distance_cm': 23.8, 'timestamp': 1755151273}
  Published Trial 10: {'trial_id': 10, 'distance_cm': 32.5, 'timestamp': 1755151273}
  Published Trial 11: {'trial_id': 11, 'distance_cm': 33.5, 'timestamp': 1755151274}
  Published Trial 12: {'trial_id': 12, 'distance_cm': 36.6, 'timestamp': 1755151275}
  Published Trial 13: {'trial_id': 13, 'distance_cm': 36.1, 'timestamp': 1755151275}
  Published Trial 14: {'trial_id': 14, 'distance_cm': 20.4, 'timestamp': 1755151276}
  Published Trial 15: {'trial_id': 15, 'distance_cm': 20.4, 'timestamp': 1755151276}
  Published Trial 16: {'trial_id': 16, 'distance_cm': 17.6, 'timestamp': 1755151277}
  Published Trial 17: {'trial_id': 17, 'distance_cm': 14.9, 'timestamp': 1755151278}
  Published Trial 18: {'trial_id': 18, 'distance_cm': 11.0, 'timestamp': 1755151278}
  Published Trial 19: {'trial_id': 19, 'distance_cm': 10.6, 'timestamp': 1755151279}
  Published Trial 20: {'trial_id': 20, 'distance_cm': 11.7, 'timestamp': 1755151280}
  Published Trial 21: {'trial_id': 21, 'distance_cm': 31.2, 'timestamp': 1755151280}
  Published Trial 22: {'trial_id': 22, 'distance_cm': 36.6, 'timestamp': 1755151281}
  Published Trial 23: {'trial_id': 23, 'distance_cm': 35.7, 'timestamp': 1755151281}
  Published Trial 24: {'trial_id': 24, 'distance_cm': 43.5, 'timestamp': 1755151282}
  Published Trial 25: {'trial_id': 25, 'distance_cm': 43.6, 'timestamp': 1755151283}
  Published Trial 26: {'trial_id': 26, 'distance_cm': 47.9, 'timestamp': 1755151283}
  Published Trial 27: {'trial_id': 27, 'distance_cm': 48.0, 'timestamp': 1755151284}
  Published Trial 28: {'trial_id': 28, 'distance_cm': 46.4, 'timestamp': 1755151284}
  Published Trial 29: {'trial_id': 29, 'distance_cm': 48.3, 'timestamp': 1755151285}
  Published Trial 30: {'trial_id': 30, 'distance_cm': 48.3, 'timestamp': 1755151286}
  Published Trial 31: {'trial_id': 31, 'distance_cm': 24.4, 'timestamp': 1755151286}
  Published Trial 32: {'trial_id': 32, 'distance_cm': 23.3, 'timestamp': 1755151287}
  Published Trial 33: {'trial_id': 33, 'distance_cm': 22.7, 'timestamp': 1755151287}
```

Fig 6: Distance Measurement output from raspberry pi to AWS

Table II shows the comparison between the actual and average measured distances from the ultrasonic sensor. The results show that the system maintained high accuracy, with deviations within ±1.3 cm for all tested distances. We observed both positive and negative error values, which corresponded to slight overestimation and underestimation. The largest deviation was −1.27 cm at an actual distance of 25 cm, and the smallest was −0.15 cm at 24 cm. These results indicate stable and reliable distance measurement performance, making it suitable for parking assist applications.

Table II: Comparison between Actual and measured Distance

Actual Distance (cm)	**Average Measured Distance (cm)**	**Average Error (cm)**
20	20.90	0.90
23	22.60	-0.40
24	23.85	-0.15
25	23.73	-1.27
31	31.20	0.20
35	34.67	-0.32
36	36.44	0.44

The time from capturing sensor data to its arrival at Raspberry Pi averaged 18 ms. The minimum was 14 ms and the maximum was 27 ms. This stage makes up a small part of the total loop latency. MQTT-over-TLS transfer from the Raspberry Pi to AWS IoT Core averaged 78 ms. The minimum was 65 ms and the maximum was 104 ms. Network conditions and TLS overheads caused some variation. The combined AWS Lambda invocation and SageMaker inference averaged 142 ms, ranging from 118 ms to 181 ms. This stage was the largest contributor to latency. The entire pipeline, from sensor acquisition to the inference result, averaged 238 ms. The minimum was 205 ms and the maximum was 291 ms.

Table III: Latency Analysis of the end-to-end Data Loop

Process Stage	**Avg (ms)**	**Min (ms)**	**Max (ms)**
Sensor to Raspberry Pi	18	14	27
Pi to AWS IoT Core	78	65	104
AWS Lambda + SageMaker Inference	142	118	181
Total End-to-End Loop	238	205	291

The results show response times under 300 ms, which are suitable for near-real-time applications. The inference stage (AWS Lambda + SageMaker) made up about 60% of the total latency. This indicates potential for improvement through model compression, serverless warm-start techniques, or edge deployment. The small standard deviation in the acquisition and network stages indicates stable hardware performance. Most of the variation came from the cloud inference stage.

REFERENCES

[1] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in Transdisciplinary Perspectives on Complex Systems. Cham: Springer, 2017, pp. 85-113.

[2] T. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US Air Force vehicles," in Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Dyn. Mater. Conf., 2012, pp. 1-14.

[3] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," J. Manuf. Syst., vol. 48, pp. 157–169, Jul. 2018.

[4] A. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," Procedia Manuf., vol. 11, pp. 939–948, 2017.

[5] X. Zhang, Y. Zhang, Y. Zhou, and F. Tao, "Digital twin-driven cyber-physical production system towards smart shop-floor," J. Ambient Intell. Hum. Comput., vol. 10, no. 11, pp. 4439–4453, Nov. 2019.

[6] Y. Song and C. Liao, "Analysis and review of state-of-the-art automatic parking assist system," 2016 IEEE Int. Conf. Vehicular Electron. Safety (ICVES), Beijing, China, 2016, pp. 1-6, doi: 10.1109/ICVES.2016.7548171.

[7] M. Liu, J. Naoum-Sawaya, Y. Gu, F. Lecue, and R. Shorten, "A distributed Markovian parking assist system," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2230-2240, Jun. 2019, doi: 10.1109/TITS.2018.2865648.

[8] P. Cheedalla and M. Karanam, "Parallel parking and parking assist system for autonomous and semi-autonomous vehicles," 2022 10th Int. Conf. Reliability, Infocom Technol. Optimization (Trends Future Directions) (ICRITO), Noida, India, 2022, pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964917.

[9] L. A. Curiel-Ramirez, R. A. Ramirez-Mendoza, G. Carrera, J. Izquierdo-Reyes, and M. R. Bustamante-Bello, "Towards a modular framework for semi-autonomous driving assistance systems," Int. J. Interact. Des. Manuf. (IJIDeM), vol. 12, pp. 1–9, 2018, doi: 10.1007/s12008-018-0465-9.

[10] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," IEEE Intell. Transp. Syst. Mag., vol. 6, no. 4, pp. 6–22, Winter 2014, doi: 10.1109/MITS.2014.2336271.

[11] O. M. J. Carsten and L. Nilsson, "Safety assessment of driver assistance systems," Eur. J. Transp. Infrastruct. Res., vol. 1, no. 3, pp. 225–243, 2001.

[12] J. D. Paramasatya and F. R. Saputri, "Experimental study on the use of sharp infrared sensor for distance detection in parking assistance applications in automotive systems," Int. J. Inf. Syst. Comput. Sci. (IJISCS), vol. 7, no. 3, pp. 221–227, Dec. 2023.

[13] M. C. J. T. Manuel, "Design and development of algorithms and control strategies for automatic parking systems: Advanced driver assistance system," 2020.

[14] G. S. Gupta, A. Roy, S. K. Das, and R. J. V. Bosch, "Autonomous parking using hybrid techniques: A review," Veh. Commun., vol. 26, pp. 1–14, Jun. 2021.

[15] S. Thrun et al., "Stanley: The robot that won the DARPA Grand Challenge," J. Field Robot., vol. 23, no. 9, pp. 661–692, 2006.

[16] Y. Zhang, T. He, and X. Zhang, "Machine learning applications for next-generation autonomous vehicle parking systems: A survey," IEEE Access, vol. 10, pp. 85732–85743, Jul. 2022.

[17] A. L. Yuille, "Anomaly detection in autonomous vehicles with neural network-based sensor fusion," Auton. Veh. Conf., pp. 1–8, 2019.

[18] C. Qian, X. Liu, C. Ripley, M. Qian, F. Liang, and W. Yu, "Digital Twin—Cyber Replica of Physical Things:

Architecture, Applications and Future Research Directions," Future Internet, vol. 14, no. 2, p. 64, Feb. 2022.

[19] Wang, and M. Wu, "Intelligent Parking Service System Design Based on Digital Twin for Old Residential Areas," Electronics, vol. 13, no. 23, p. 4597, Nov. 2024.

[20] D. Wu, A. Zheng, W. Yu, H. Cao, Q. Ling, J. Liu, and D. Zhou, "Digital Twin Technology in Transportation Infrastructure: A Comprehensive Survey of Current Applications, Challenges, and Future Directions," Appl. Sci., vol. 15, no. 4, p. 1911, Feb. 2025.

[21] A. R. Broggi, M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco, and A. Piazzi, "Visual perception of obstacles and vehicles for platooning," IEEE Trans. Intell. Transp. Syst., vol. 1, no. 3, pp. 164–176, Sep. 2000.

[22] S. Rai and D. Sharma, "A review on artificial intelligence approaches for autonomous parking of intelligent vehicle," Eng. Sci. Technol. Int. J., vol. 25, pp. 1–7, 2022.

[23] R. Mohan, A. Srivastava, M. Arora, and A. Banerjee, "Design of path planning algorithms for parking assistance system in autonomous vehicles," Int. J. Veh. Des., vol. 23, no. 6, pp. 45–60, 2019.

[24] F. Fang and J. Zhou, "Digital twin-driven adaptive manufacturing," Annu. Rev. Control Robot. Auton. Syst., vol. 1, pp. 201–224, 2022.

[25] C. Lin, H. Ghenniwa, and W. Shen, "Agent-based approach to supporting conflict resolution in a multi-driver cooperative parking environment," Comput. Ind., vol. 79, pp. 1–14, 2016.

[26] J. Kim and M. Jeon, "Intelligent parking assist system using sensor fusion algorithm," Sensors, vol. 21, pp. 1–15, 2021.

[27] Geng, Y., Cassandras, C. G., & Wang, Y. (2012). A new smart parking system infrastructure and implementation. Procedia - Social and Behavioral Sciences, 54, 1278–1287.

[28] Lin, T., Rivano, H., & Le Mouël, F. (2017). A survey of smart parking solutions. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229–3253.

[29]Amato, G., Carrara, F., Falchi, F., Gennaro, C., & Vairo, C. (2016). Car parking occupancy detection using smart camera networks and deep learning. 2016 IEEE Symposium on Computers and Communication (ISCC).

[30] Liu, H., & Teng, J. (2018). Radar-based vehicle detection and tracking for parking management. Sensors, 18(2), 357.

[31] Rodriguez, R., & Martinez, J. (2015). Wireless sensor network for intelligent parking systems. Procedia Computer Science, 52, 103–110.

[32] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., ... & Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms. 2011 IEEE Intelligent Vehicles Symposium (IV).