Sensors and Materials, Vol. 38, No. 9 (2026) 5185–5198
MYU Tokyo

S & M 4617

Performance Analysis for Ultrasonic Sensors Using Arduino UNO on Controller Area Network System

Sheng-Jen Wen,[1,2] Fuh-Liang Wen,[3*] and Ching-Hsu Chan[3]

[1]Engineering & Maintenance Division, STARLUX Airlines Flight Operations Center,
1, Hangxiang Rd., Dayuan Dist., Taoyuan City 337052, Taiwan
[2]Department of Transportation Science, National Taiwan Ocean University,
2, Beining Rd., Zhongzheng Dist., Keelung City 202301, Taiwan
[3]Department of Electrical Engineering, St. John's University,
499, Sec. 4, Danjin Rd., Tamsui Dist., New Taipei City 251309, Taiwan

(Received March 4, 2026; accepted August 20, 2026)

Keywords: performance analysis, ultrasonic sensor, open loop, close loop, Arduino UNO, linear relationship

In this paper, we propose a simplified performance analysis for ultrasonic sensors, in which an open-loop design is compared with a close-loop controller area network (CAN) system. Using Arduino UNO microcontroller boards on Controller Area Network (CAN) System, the linearity and accuracy of HC-SR04 detecting modules are evaluated efficiently. Experimental results showed that the close-loop CAN system achieves faster response and higher accuracy under varying environmental conditions than the open-loop design. The method offers a practical solution for improving long-distance and wide-range ultrasonic measurements.

1. Introduction

Ultrasonic engineering has advanced considerably in recent decades,(1–4) finding applications in actuators,(5–10) sensing technologies,(11–19) industrial machining, and sonar systems. With progress in material engineering, ultrasonic waves are commonly generated by AC-powered natural crystals and piezoelectric ceramics operating above 20 kHz.(1,2) Despite their wide use, practical performance analysis for ultrasonic sensors remains limited,(20–32) particularly in detection system design. In this study, we present an open-loop performance analysis and propose a close-loop design based on Arduino UNO microcontroller boards and a controller area network (CAN) bus(33–36) for vehicle parking applications (Fig. 1). Ultrasonic sensing enables accurate distance and range measurements, which are widely applied in automotive parking detectors, with an effective detection range of 0.3–3.0 m.

The HC-SR04 detector integrates a horn, metal plate, piezoelectric ceramic, and AC input leads. The piezoelectric ceramic generates ultrasonic waves when energized and produces return signals upon echo reception. This cost-effective sensor is widely used for obstacle avoidance,

*Corresponding author: e-mail: ericwen@mail.sju.edu.tw
https://doi.org/10.18494/SAM6320

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

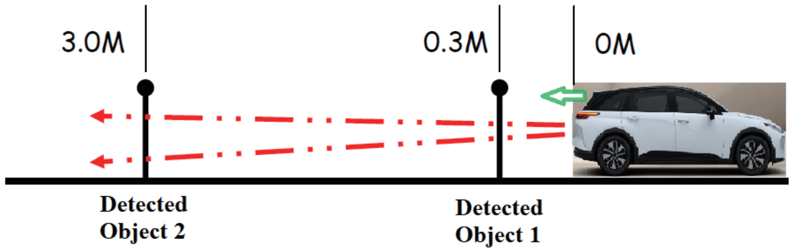


Fig. 1. (Color online) Distance measurement of ultrasonic detection and ranging.

water-level sensing, and parking assistance. Practical testing showed that it covers a 45° scanning sector, with optimal performance within 30°.

In modern automotive systems, reliable communication between devices is essential for safety and efficiency. To address this, various automotive networks have been developed, as follows: CAN bus with data rates up to 1 Mbps, the Local Interconnect Network (LIN) bus limited to 20 kbps, and FlexRay bus, which supports up to 10 Mbps for advanced onboard computing. In this work, the HC-SR04 detector is implemented for distance measurement and data transmission via Arduino UNO for open-loop performance analysis, while close-loop design is achieved using MCP2515 CAN bus modules. Additionally, a ZeroPlus logic analyzer is employed to evaluate signal integrity and communication stability on the CAN bus. In this study, we compare the open- and close-loop systems under constraints of quick response and low power consumption, offering insights for future optimization.

2. Experimental Setup

In this paper, we propose a simplified performance analysis for ultrasonic sensors using an Arduino UNO circuit design and measurement technique. The operating principle of ultrasonic ranging is illustrated in Fig. 2. The HC-SR04 supports a wide input voltage range (3.3–5 V), making it compatible with low-cost microcontrollers such as Arduino UNO. It applies the sonar principle by transmitting a 40 kHz ultrasonic pulse and calculating the distance from the echo return time. The module provides stable readings with good accuracy and integrates a four-pin 2.54 mm header, enabling easy prototyping on solderless breadboards.

2.1 Open-loop measurement

As shown in Fig. 3(a), a 10 µs pulse applied to the "trigger" pin initiates ranging. The HC-SR04 sensor emits an 8-cycle 40 kHz burst, and the echo pulse width, proportional to object distance, is measured by an Arduino UNO. The sensor linearity and accuracy are evaluated with this open-loop design. Figure 3(b) shows the wiring, where the trigger transducer emits pulses that reflect off objects and are received by the echo transducer. Travel time is converted into distance using the standard formula.[36]

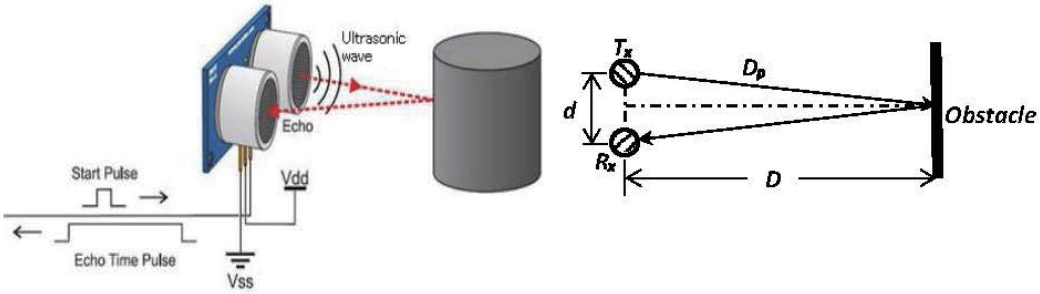
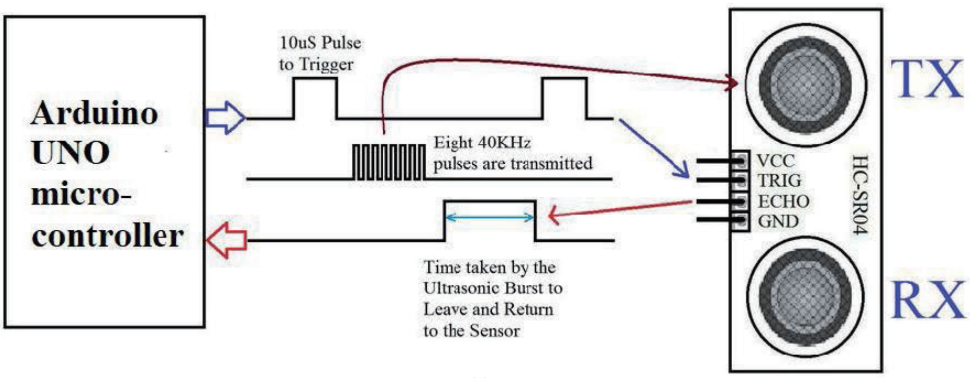
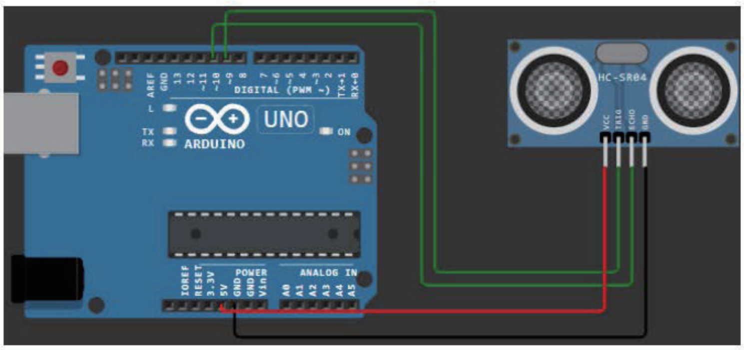


Fig. 2. (Color online) Ultrasonic wave sending and receiving on pulse signal reflections.



(a)



(b)

Fig. 3. (Color online) (a) Open-loop measurement and (b) wiring diagram for Arduino UNO board with ultrasonic detector HC-SR04.

$$D = D_p \sqrt{1 - \left(\frac{d}{2D_p} \right)^2} = \sqrt{\left(\frac{t}{2} \bullet v \right)^2 - \left(\frac{d}{2} \right)^2}, \tag{1}$$

where

D is the physical distance between an object and the ultrasonic sensor (cm),

D_p is half the traveling distance of ultrasonic waves (cm),

t is the signal travel time (s),

v is the speed of sound in air (~343 m/s) at room temperature of about 25 ℃, and

d is the pitch of the emitter (T_x) and receiver (R_x) for a pair of ultrasonic sensors, as shown in Fig. 2. If $D_p/d > 10$, sensing errors would be less than 1% for D and D_p

However, the measurement environment conditions—such as temperature, humidity, and atmospheric pressure—directly affect sensing accuracy. Errors of centimeter scale can occur within a 30 cm range, and larger errors arise at distances beyond 100 cm. Increasing the driving power of the emitter and enlarging the receiver amplitude can extend the measurement range. However, this also increases the number of reflected waves, which reduces efficiency and introduces additional errors, motivating the need for a close-loop approach.

The distance measured by the HC-SR04 ultrasonic sensor accounts for the round-trip travel of the signal, as shown in Fig. 4. The x-axis represents the sensor reading, whereas the y-axis shows the manually measured actual distance. The measurement error is calculated as

$$Error = Actual\ Distance - Sensor\ Reading,\ Error\% = Error/Actual\ Distance \times 100\%. \tag{2}$$

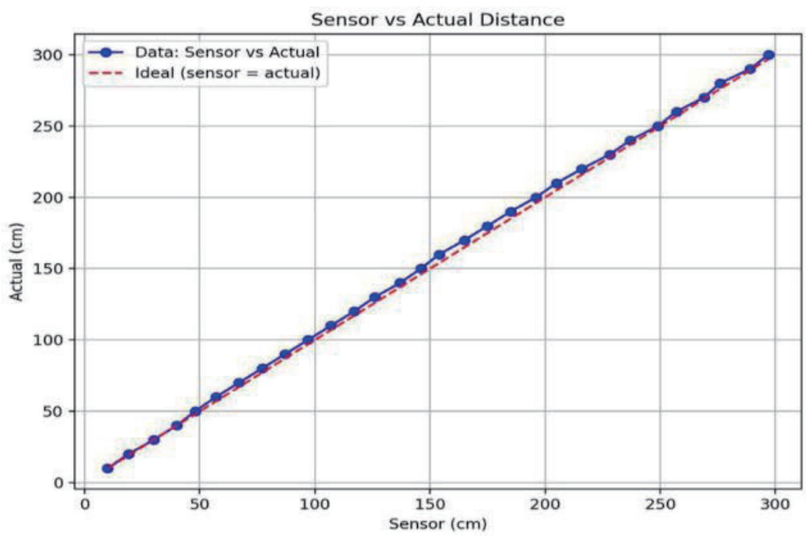


Fig. 4. (Color online) Linear relationship between the distance measured by the HC-SR04 ultrasonic sensor (x-axis) and the actual distance measured manually (y-axis).

In Fig. 4, the blue dotted line represents the sensor readings, and the red dashed line indicates the ideal case where the sensor readings perfectly match the actual distance ($y = x$). A strong linear correlation is observed between the sensor readings and the actual distances. However, because Arduino UNO provides no feedback for double-checking the measurements, this lack of verification remains a notable limitation. As Fig. 4 shows, the results of the open-loop measurement indicate that the HC-SR04 sensor has high accuracy at short distances (≤50 cm) with an error of <5%. In contrast, at distances >150 cm, deviation increases, reaching approximately ~3–4%.

2.2 Close-loop measurement on CAN bus

To extend the original open-loop design, a corresponding close-loop measuring system was implemented on a CAN bus. In this system, multiple devices communicate via CAN message frames, forming a simplified distributed embedded network. Reliable and real-time interaction between sensors and controllers is crucial for performance. The system integrates HC-SR04 ultrasonic detectors and DHT11 temperature/humidity sensors on the CAN bus to transmit distance and environmental data across nodes. System behavior under varying thermal conditions was analyzed using the USBCAN II Pro device and GCAN Tools software.

The close-loop CAN system (Fig. 5) was designed to (1) integrate the HC-SR04 ultrasonic sensor with CAN bus communication, (2) transmit data frames between two Arduino UNO nodes (transmitter and receiver), (3) display measured distances on a 16 × 2 I2C Liquid-Crystal Display (LCD) via the CAN bus, (4) analyze CAN signals using a ZeroPlus logic analyzer, and (5) validate received data with GCAN Tools and assess accuracy against actual distances and ambient temperature.

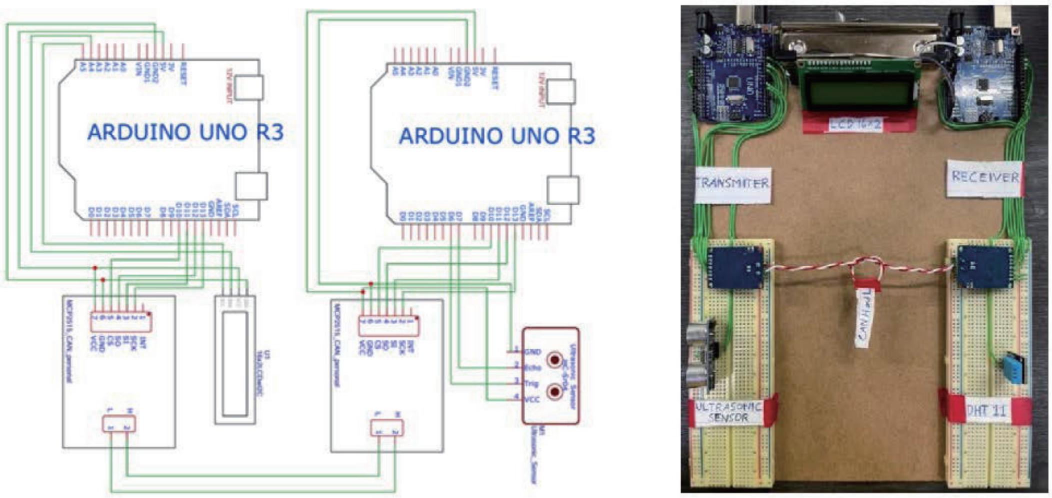


Fig. 5. (Color online) Circuit schematic of Arduino UNOs connected with MCP2515 modules (photo) interfaced with ultrasonic sensor.

The hardware and software configuration of the system includes the following.

- Arduino UNO (Transmitter & Receiver)—The transmitter reads ultrasonic data, whereas the receiver receives frames and displays distance on the LCD.
- HC-SR04 Ultrasonic Sensor & DHT11 Sensor—The HC-SR04 measures distance, and the DHT11 monitors temperature and humidity to evaluate CAN data stability under thermal variations.
- MCP2515 CAN Module—This connects the Arduinos and the USBCAN interface to implement CAN communication.
- ZeroPlus Logic Analyzer & USBCAN II Pro—The logic analyzer captures CAN signals, whereas USBCAN II Pro and GCAN Tools validate data frames and assess measurement accuracy relative to actual distances and ambient conditions.

3. Measuring Instruments and Experimental Techniques

The innovative CAN bus experimental setup, illustrated in Fig. 6, incorporates an XDS-series oscilloscope, a USBCAN-II Pro interface with GCAN software, and a PC-based logic analyzer (LA). CAN dynamic protocol waveforms are captured and decoded in hexadecimal format via the oscilloscope and/or USBCAN-II Pro device with its associated software. The LA plays a critical role in protocol analysis, debugging, and system verification of the digital system, as well as in monitoring timing violations and transient events.

The device under test (DUT) communicates via the CAN bus, transmitting real-time data frames from both an ultrasonic sensor and a DHT11 temperature and humidity sensor. Prior to calibrating various ultrasonic sensors, the ZDS2000 signal board was used as a standard signal source for the entire CAN bus measuring platform.

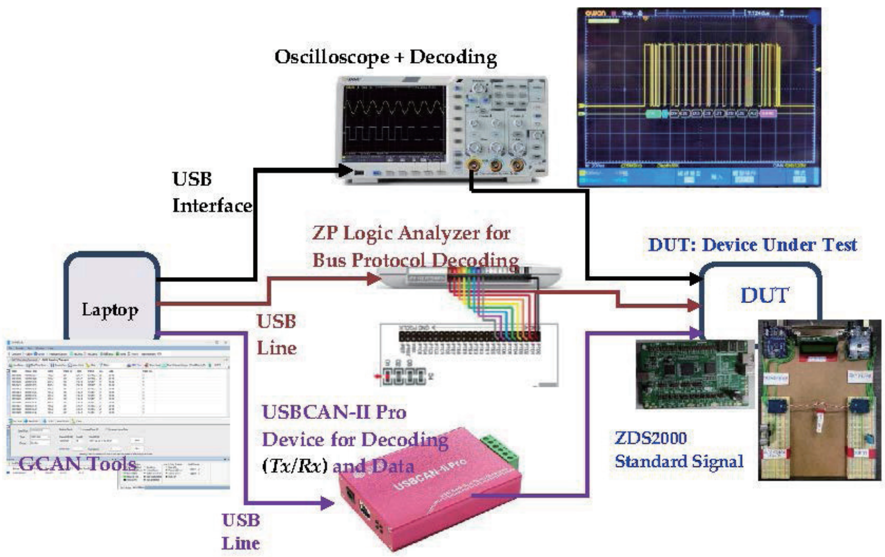


Fig. 6. (Color online) Experimental configuration of CAN bus communication system.

The experimental setup for CAN bus communication, shown in Fig. 7, integrates HC-SR04 and DHT11 sensors with Arduino UNOs and MCP2515 CAN modules. Sensor performance was evaluated under various ambient temperatures, and the stability of CAN data transmission under thermal disturbances was assessed. The received data were validated using GCAN Tools and a PC-based LA, by comparing CAN-transmitted distance and temperature values against actual measurements. Data frames were encoded with appropriate identifiers, and the USBCAN II Pro device reads the transmitted frames for cross-verification. Environmental temperature effects were occasionally simulated using a hair dryer to test system robustness.

There is a simple correction through Node 2: Receiver in Fig. 7 as feedback on the internal calculation of an Arduino-microprocessor-connected MCP2515 CAN module. Also, all accuracy readings are displayed on a 16 × 2 array LCD, and they are stored into the memory for double checking the data. Therefore, this concept is similar to close-loop measurement but not a close-loop control principle that offers the feedback action.

3.1 USBCAN device with GCAN software

To analyze CAN bus data frames using the USBCAN-II Pro device (USBCANII), connect USBCANII to the CAN bus system via a USB interface. The procedure is as follows. (1) After the driver is installed for USBCANII, the setting up of the device manager is required. Moreover, (2) through the secure connections with the USBCANII to the computer, the CAN bus network is verified. Finally, data collection and analysis can be executed using GCAN software after configuring the correct baud rate. These collected data cover the complete CAN messages (ID, data, timestamp) for further analysis. Thus, the USBCANII and its software allow effective data collection and analysis, as shown in Figs. 8(a) and 8(b), while contributing significantly to the project’s success.

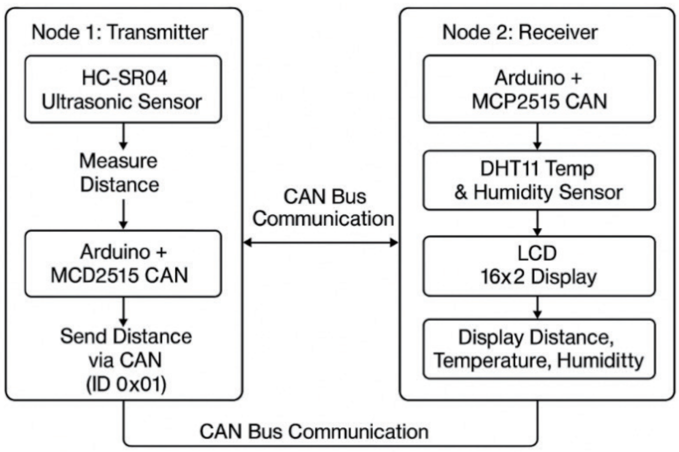
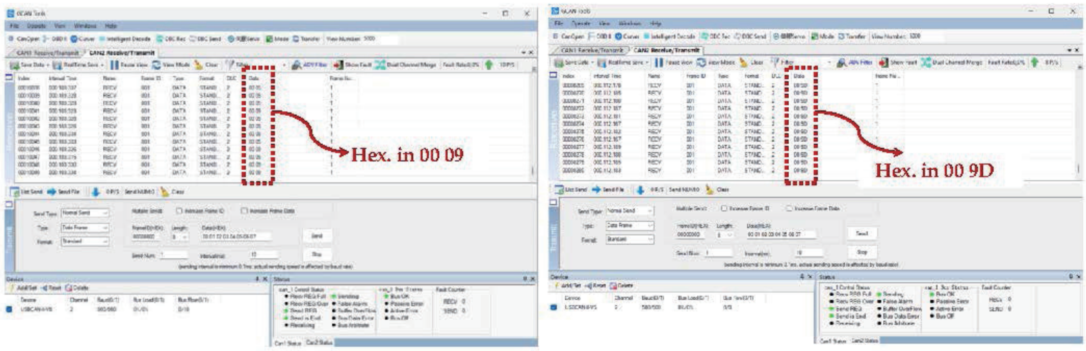


Fig. 7. Experimental setup and procedures of CAN bus communication.



(a) (b)

Fig. 8. (Color online) Results of two experiments with various CAN temperatures from 16 to 30 °C using GCAN Tools. (a) Measured at 16 °C and 9 cm (hexadecimal 00 09) sensing distance. (b) Measured at 30 °C and 157 cm (hexadecimal 00 9D) sensing distance.

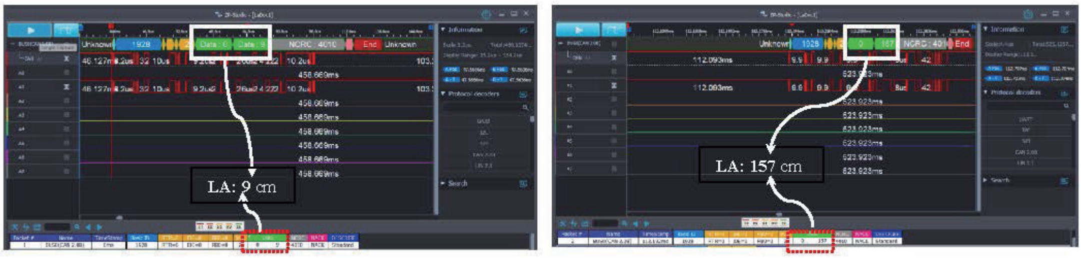
3.2 ZeroPlus PC-based LA with ZP Studio software

A PC-based LA can be used to observe, capture, and analyze the protocol signals of CAN bus in real time. It complements software-only CAN sniffers by providing precise timing and signal integrity analysis. The typical workflow involves the following. (1) Through software and driver installation and set up, the LA's probe pins are connected to the CAN_H and CAN_L lines using proper clip probes. (2) After connecting the ground clip to the common GND of the system, all probes are securely connected and insulated to avoid interference or short circuits. (3) By using the ZP Studio software and selecting the correct channel, the series signals are acquisitioned on the basis of the sampling rate (recommended: at least 10 times the CAN Bus baud rate; e.g., for 500 kbps, use ≥ 5 MHz). (4) By applying the protocol decoder setting to CAN 2.0, the CAN data are captured and analyzed. Thus, a PC-based LA demonstrates its engineering function for capturing signals and triggering specific frame IDs or patterns. It is highly suitable to observe the waveform and to decode CAN frame information (ID, data, CRC, etc.). In CAN engineering, a PC-based LA is always applied to waveform and timing data to detect anomalies, voltage issues, or bus arbitration behavior, as shown in Figs. 9(a) and 9(b).

Using the values measured taking into consideration the effect of temperature as an environmental factor, shown in Table 1, root mean squared error (*RMSE*), standard deviation, and error distribution versus distance obtained from ultrasonic detectors were determined and are shown in Fig. 10. The CAN system stability and reproducibility are confirmed.

4. Results and Discussion

After the implementation of close-loop measurement on CAN bus, the following results were obtained: (a) CAN data captured using GCAN Tools with the USBCAN-II Pro device (Fig. 8), and (b) data captured using ZP Studio with the Zeroplus LA (Fig. 9). Table 1 summarizes the measured distances under different ambient temperatures (~16 and ~30 °C). The temperature



(a) (b)

Fig. 9. (Color online) Results of LA measurements at temperatures and sensing distances of (a) 16 °C and 9 cm and (b) 30 °C and 157 cm, respectively.

Table 1
Measured distances with thermal effect considered as an environment factor in the close-loop measurement with ultrasonic sensors.

Actual distance Unit: cm	Low-temperature (~16 °C) Measured distance (cm)	High-temperature (~30 °C) Measured distance (cm)	Difference (heat) Unit: °C
10	9	8	−1
20	19	18	−1
30	29	28/27	−1/−2
40	38	37	−1
50	48	47	−1
60	57	56	−1
70	67	66	−1
80	77	76	−1
90	87	86	−1
100	97	96	−1
110	107	105/106	−2/−1
120	117	115	−2
130	127	125	−2
140	137	135	−2
150	147	145	−2
160	157	155	−2

selections of 16 and 30 °C represent the natural ambient range of a standard experimental workshop in Taiwan, covering the typical seasonal variations throughout one year. Here, 16 °C represents the typical indoor temperature during a Taiwanese winter, and 30 °C represents the typical indoor temperature during a Taiwanese summer afternoon. The experiment was conducted in a standard lab classroom under Taiwan's natural climate. Testing at 16 and 30 °C allows us to evaluate the system's stability across the most common environmental temperatures in Taiwan. This ensures that the CAN-bus data and sensor accuracy remain reliable under real-world ambient fluctuations.

The entire process was monitored using CAN frames captured with the USBCAN-II Pro device and GCAN Tools software. Our device successfully received and decoded the CAN message frames, with all transmitted distance and temperature data being consistently received in the correct format. Data for selected example message frames are listed in Table 2.

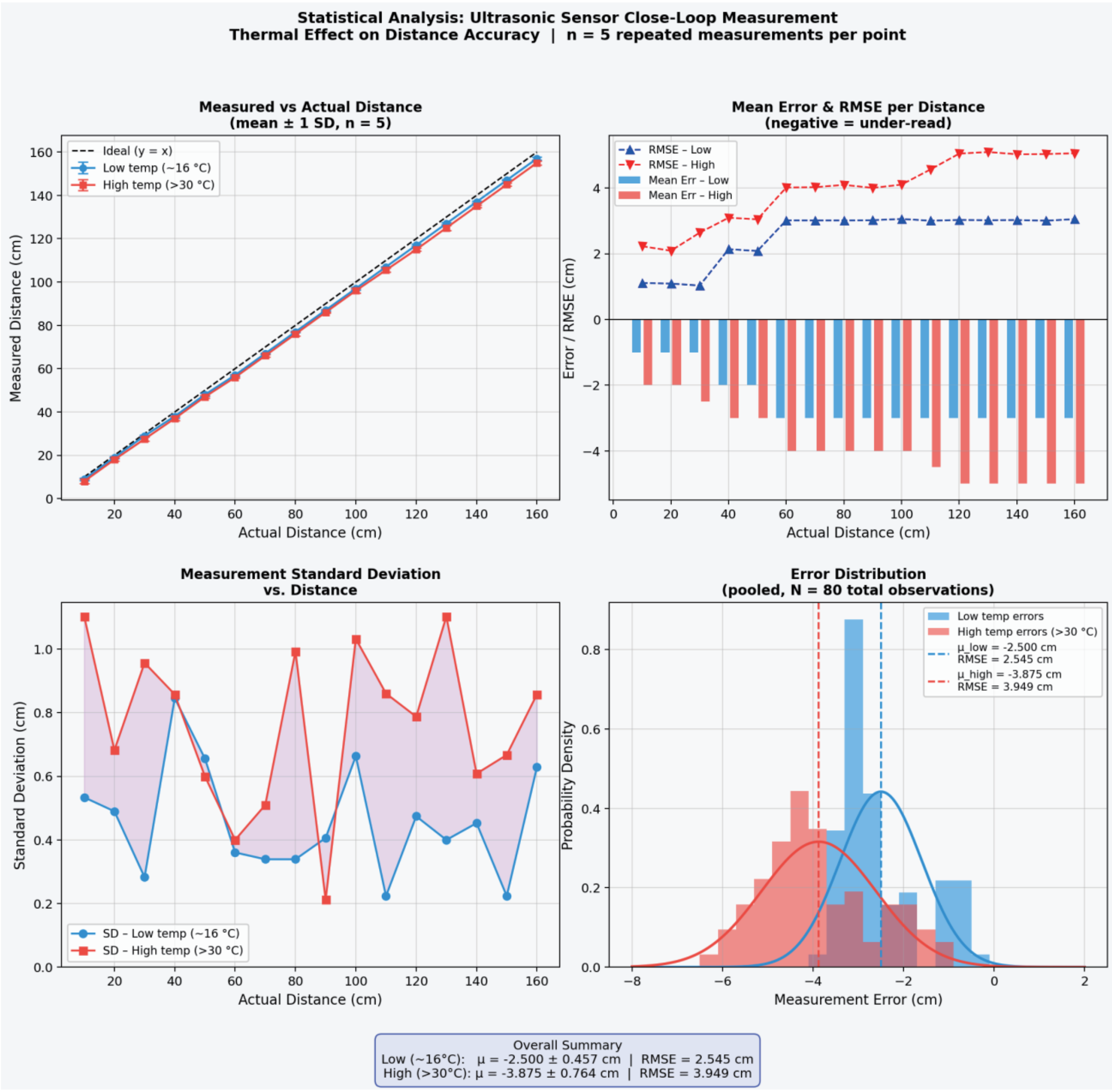


Fig. 10. (Color online) Characteristic curves and statistical analysis of data from thermal disturbance experiment using a hair dryer as a heater.

Table 2
Data of CAN protocol message frames from USBCAN-II Pro device at different temperatures.

Environment temperature at 16 ℃			Environment temperature at 30 ℃		
Frame ID	Data (Hex)	Interpreted data	Frame ID	Data (Hex)	Interpreted data
0x001	00 09	Distance = 9 cm	0x001	00 08	Distance = 8 cm
0x001	00 30	Distance = 48 cm	0x001	00 2F	Distance = 47 cm
0x001	00 61	Distance = 97 cm	0x001	00 60	Distance = 96 cm
0x001	00 9D	Distance = 157 cm	0x001	00 9B	Distance = 155 cm

Temperature and distance are critical and interrelated factors in ultrasonic sensing because sound velocity depends on air temperature, which directly affects time-of-flight distance calculations. As shown in Table 2, at higher temperatures, ultrasound travels faster, resulting in a shorter time-of-flight (shorter by about 1–2 cm). Thus, the speed of ultrasound in air is not constant, necessitating compensation to achieve accurate measurements for comparing system performance before and after CAN bus compensation. A 1.04–2.13% improvement in accuracy is attained after CAN bus compensation.

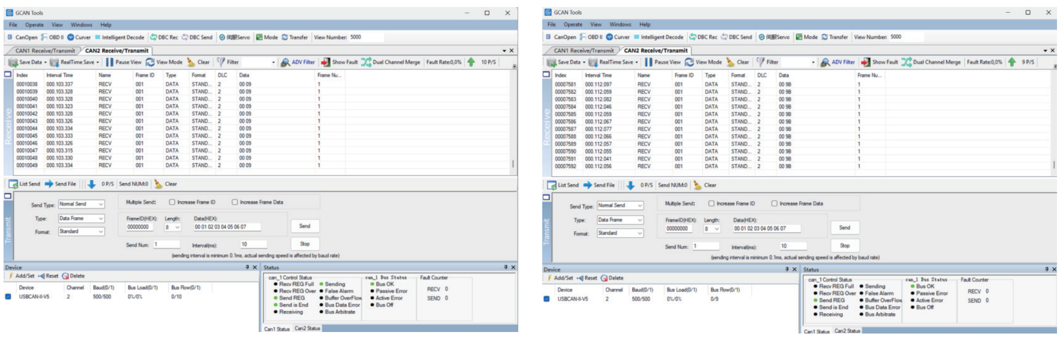
To evaluate system reproducibility, measurements were repeated five times at each distance point under the two temperature conditions (n = 5, total = 80 observations per condition). These results show mean errors of −2.500 ± 0.457 cm (*RMSE* = 2.545 cm) at low temperature (~16 °C) and −3.875 ± 0.764 cm (*RMSE* = 3.949 cm) at high temperature (~30 °C), confirming that thermal effects introduce a systematic negative bias consistent with increased acoustic propagation speed at elevated temperatures.

GCAN Tools software provided a real-time frame viewer and frame history log, enabling the visual verification of the ID, timestamp, DLC, and payload data. The representative screenshots under cold and warm conditions are shown in Fig. 11.

To investigate the effect of thermal disturbances on ultrasonic sensors, an experiment was conducted using a hair dryer to introduce warm airflow into the measurement environment. The hair dryer was positioned 10–15 cm from the sensor, directed sideways. Observations include the following.

- Sensor readings fluctuated around 46–47 cm despite the target object being fixed.
- When the object was moved far away (e.g., 160 cm), readings remained locked on the object.
- Eventually, the sensor reported distances of only 3–4 cm and became unresponsive.
- Resetting power restored functionality.

The disturbance to readings is primarily caused by thermal effects (increased sound speed), not acoustic noise, which resulted in shortened echo travel time and disrupted reflections. However, by addressing potential sources of error and measurement uncertainty, we strengthen the credibility and accuracy of the results obtained using the close-loop CAN system.



(a) (b)

Fig. 11. (Color online) GCAN Tools software display of the original data of CAN protocol message frames. (a) 9 cm distance at 16 °C and (b) 155 cm distance at 30 °C.

5. Conclusions

A simplified close-loop design for ultrasonic sensors was implemented on a CAN bus platform. The HC-SR04 successfully transmitted distance data via CAN bus to a 16 × 2 LCD, with correct transmission verified by GCAN Tools and USBCAN II Pro at 16 and 30 °C. Distance readings decreased at higher temperatures, reflecting the temperature dependence of sound speed. A PC-based LA confirmed stable, accurate CAN communication. These results indicate that environmental factors can be incorporated into the measurement and that HC-SR04 and DHT11 sensors can be effectively integrated with CAN bus for practical applications. After applying the close-loop sensor design in a CAN system, the ultrasonic sensing technique can be focused on the integration of 3D ultrasonic sensor arrays in the microprocessor of Arduino UNO R4. Further studies with AI-driven sensor fusion based on the advanced microprocessor of Arduino UNO Q can enhance obstacle recognition in unstructured environments, enabling current limitations in angular resolution and long-range accuracy to be addressed.

Acknowledgments

We sincerely thank the Taiwan Society of Professional Development in Vehicles for supporting this research under grant number TSPDV 20250615.

References

1 D. Ensminger: Ultrasonics: Fundamentals, Technology, Applications, 2nd ed., Revised and Expanded (New York: Marcel Dekker, 1988).
2 O. V. Abramov: Ultrasound in Liquid and Solid Metals (CRC Press, 1994).
3 S. Ueha, Y. Tomikawa, M. Kurosawa, and N. Nakamura: Ultrasonic Motors- Theory and Application (New York: Oxford University Press, 1993).
4 T. Sashida and T. Kenjo: An Introduction to Ultrasonic Motors (Oxford: Clarendon Press, 1993).
5 A. Carullo and M. Parvis: IEEE Sensors J. **1** (2001) 143. https://doi.org/10.1109/JSEN.2001.936931
6 F.-L. Wen, M.-H. Lai, and C.-P. Wen: Instrum. Exp. Tech. **67** (2024) 602.
7 C.-C. Wen, F.-L. Wen, and C.-H. Lin: Instrum. Exp. Tech. **56** (2013) 104.
8 M.-H. Lai, M.S. Ouyang, and F.-L. Wen: J. Mech. **28** (2012) 559.
9 F. L. Wen and I. Hsu: Instrum. Exp. Tech. **54** (2011) 488.
10 M.-H. Lai, F.-L. Wen, Y.-L. Shen, and M. Ouyang: Applied Mechanics and Materials **87** (2011) 1.
11 Y. Wei: Proc. 5th Int. Conf. Signal Processing and Machine Learning (2022).
12 L. Koval, J. Vaňuš, and P. Bilík: 14th IFAC Conf. Programmable Devices and Embedded Systems (PDES 2016) **49** (2016) 153. https://doi.org/10.1016/j.ifacol.2016.12.026
13 W. Prima and Y. C. Giap: Bit-Tech **3** (2020) 81.
14 A. L. Bowler, M. P. Pound, and N. J. Watson: Ultrasonic **124** (2022) 106776. https://doi.org/10.1016/j.ultras.2022.106776
15 J. Park, Y. Je, H. Lee, and W. Moon: Ultrasonics **50** (2010) 3406.
16 G. Mauris, E. Benoit, and L. Foulloy: IEEE Trans. Instrum. Meas. **49** (2000) 1. https://dblp.org/db/journals/tim/tim49.html
17 C. Nadee and K. Chamnongthai: J. Healthcare Eng. Article ID **2212020** (2022) 21. https://doi.org/10.1155/2022/2212020
18 T. S. R. Pereira, T. P. de Carvalho, T. A. Mendes, and K. T. M. Formiga: Sustainability **14** (2022) 5512. https://doi.org/10.3390/su14095512
19 F. P. Nurmaida, A. I. Gunawan, and R. S. Dewanto: Adv. Technol. Innovation **9** (2024) 186.
20 HC-SR04 Ultrasonic Sensor Module User Guide V2.0: www.handsontec.com (accessed December 2023).

21 A. Carullo and M. Parvis: IEEE Sensors J. **1** (2001) 143. https://ieeexplore.ieee.org/document/936931
22 Y. Wei: Proc. 2024 5th Int. Conf. Signal Processing and Machine Learning (ICSPML, 2024) 144–148.
23 M. Sirumalla: Kakatiya Institute of Technology & Science, Warangal (2021) 1. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3918137
24 H. Z. Khaleel1, A. K. Ahmed, A. Sh. M. Al-Obaidi, S. Luckyardi, D. F. Husaeni, R. A. Mahmod, and A. J. Humaidi: Indonesian J. Sci. Technol. **9** (2024) 145.
25 J. Park, Y. Je, H. Lee, and W. Moon: Ultrasonics **50** (2010) 340. https://doi.org/10.1016/j.ultras.2009.10.013
26 S. S. Makadi, A. S. Mahulkar, N. R. Malave, S. S. Maind, A. A. Mali, N. Maindargi, and P. P. Dandavate: Int. Conf. Innovative Computing & Communication (ICICC, 2024). SSRN: https://ssrn.com/abstract=4746566 or http://dx.doi.org/10.2139/ssrn.4746566
27 T. Pereira, T. Carvalho, T. Mendes, and K. Formiga: Sustainability **14** (2022) 5512. https://doi.org/10.3390/su14095512
28 B. S. Bari, K. Yelamarthi, and S. Ghafoor: Sensors **23** (2023) 3610. https://doi.org/10.3390/s23073610
29 CAN in Automation: http://www.can-cia.org (accessed March 2023).
30 F.-H. Hsu, T.-C. Liu: Hua Kang J Engineering, Chinese Culture University **27** (2011) 148.
31 S.-S. Huang and Y.-H. Lin: J. Cheng Shiu University **26** (2013) 33.
32 CAN Specification Version 2.0: https://www.academia.edu/9998014/BOSCH_CAN_Specification (accessed September 1991).
33 CAN Primer: Creating Your Own Network: https://www.keil.com/appnotes/files/apnt_236_v2.9.pdf (accessed 2013).
34 Introduction to the Controller Area Network (CAN): https://www.ti.com/lit/an/sloa101b/sloa101b.pdf?ts=1786356992237&ref_url=https%253A%252F%252Fwww.google.com%252F (accessed May 2016).
35 Controller Area Network (CAN) Implementation Guide: https://www.analog.com/en/resources/app-notes/an-1123.html (accessed 2017).
36 G.-J. Jong, A.-S. Hou, and C.-H. Liao: Sensor Principle (Chuan Hwa Publishing Ltd., New Taipei City, 2012) Mandrin Version, Chap. 3.

About the Authors

Sheng-Jen Wen received his B.S. degree from National Formosa University, Taiwan, in 2020 and is currently pursuing his M.S. degree at the Department of Transportation Science, National Taiwan Ocean University, Taiwan. From 2020 to 2023, he was a maintenance technician at the Aerospace Industrial Development Corporation (AIDC). Since 2023, he has been a quality assurance engineer at STARLUX Airlines, Taiwan. His research interests are in avionics engineering, CAN bus, and sensors.

Fuh-Liang Wen received his M.S. degree in manufacturing technology (automation) from Minnesota State University at Mankato, Minnesota, USA, in 1992 and his Ph.D. degree in engineering and system science (electronic, instrument control & MEMS) from National Tsing Hua University, Taiwan in 2004. Since 2009, he has been a professor at St. John's University, Taiwan. His research interests include the field of CAN/LIN/FlexRay bus and the design and control of piezoceramic-driven ultrasonic sensors and actuators.

Ching-Hsu Chan received his B.S. degree from National Chiao Tung University, Taiwan, in 1992 and his M.S. and Ph.D. degrees from National Taiwan University, Taiwan, in 1994 and 1998, respectively. From 2002 to 2007, he was an assistant professor at St. John's University, Taiwan, where he has been an associate professor since 2007. His research interests include semiconductor optics, semiconductor manufacturing and processes, and sensors.