*Sensors and Materials*, Vol. 35, No. 4 (2023) 1487–1495
MYU Tokyo


S & M 3262

# Quick Prototyping of Companion Bots for Elderly People

Thanapat Lappromrattana and Pitikhate Sooraksa*

School of Engineering, King Mongkut's Institute of Technology Ladkrabang,
1, Soi Chalongkrung 1, Ladkrabang, Bangkok 10520, Thailand

(Received November 30, 2022; accepted February 22, 2023)

***Keywords:*** chatbot, voice command, companion bot, robot for elderly

Companion bots such as chatbots in cyberspace or robots in real space gained popularity during the COVID-19 pandemic as a means of comforting humans and reducing their loneliness. These bots can also help enhance the lives of elderly people. In this paper, we present how to design and implement a quick prototype of companion bots for elderly people. A companion bot named "Hello Steve" that is able to send emails, open YouTube to provide entertainment, and remember the times an elderly person must take medicine and remind them is designed and implemented as a quick prototype. In addition, the bot combines the features of a mobile robot and a chatbot. The experimental results show the effectiveness of the design through its very high accuracy when navigating mobile-robot-like tasks and responding to chatbot-like tasks via voice commands.

## 1. Introduction

The lockdown and isolation due to the COVID-19 pandemic have drastically affected the social life of elderly people, increasing the demand for robots and chatbots that can help them.[1–6] Isabet *et al.*[7] carried out a literature review of such robots using the PubMed/Medline, Web of Science, Scopus, EMBASE, and PsychINFO search engines with three categories of keywords: elderly or older or senior or elder; telepresence robot or assistive robot or social robot; and COVID-19.

As an example of using a multirobot architecture during the pandemic, Bao *et al.*[8] designed autonomous robots for cabin hospitals in Wuhan, China, with special focus on the sensors used by service robots. The functions of the robots include autonomous disinfection, delivery, cleaning, temperature measurement, physical interaction, and conversation assistance. Utilizing multiple sensors, these robots, controlled by cloud controllers, can collaborate with each other in large and complicated scenes.

The aim of our paper is to provide guidelines for constructing a quick prototype of a companion robot to assist elderly people in their daily life and during a lockdown. It focuses on the design and development of a user-friendly system that integrates chatbot components and mobile robotic parts. Available communication technologies such as WiFi, Zigbee, radio-

*Corresponding author: e-mail: pitikhate.so@kmitl.ac.th
https://doi.org/10.18494/SAM4268

ISSN 0914-4935 

frequency identification, and Bluetooth Low Energy enable assistive health robots and elderly people to communicate.(9–14) In addition, these sensors and communication technologies provide signal parameters applicable to localization techniques,(15,16) and can be used in combination with the navigation algorithm using ultrasonic sensors, as described in Sect. 2, along with other robotic information. The results and discussion of our study are presented in Sect. 3. Finally, Sect. 4 concludes the current work and suggests future work.

## 2. Materials and Methods

This section explains related materials and methods and is organized as follows. Section 2.1 describes voice assistant functions and the chatbot function of the robot named "Hello Steve". The physical appearance of "Hello Steve" is shown in Fig. 1. These voice command functions will be used to command the robots via the voices of elderly people. Section 2.2 presents a localization and map-making algorithm for our robot. These functions will be used by the robot to locate its position and orient itself inside the house of an elderly person while acting as a companion.

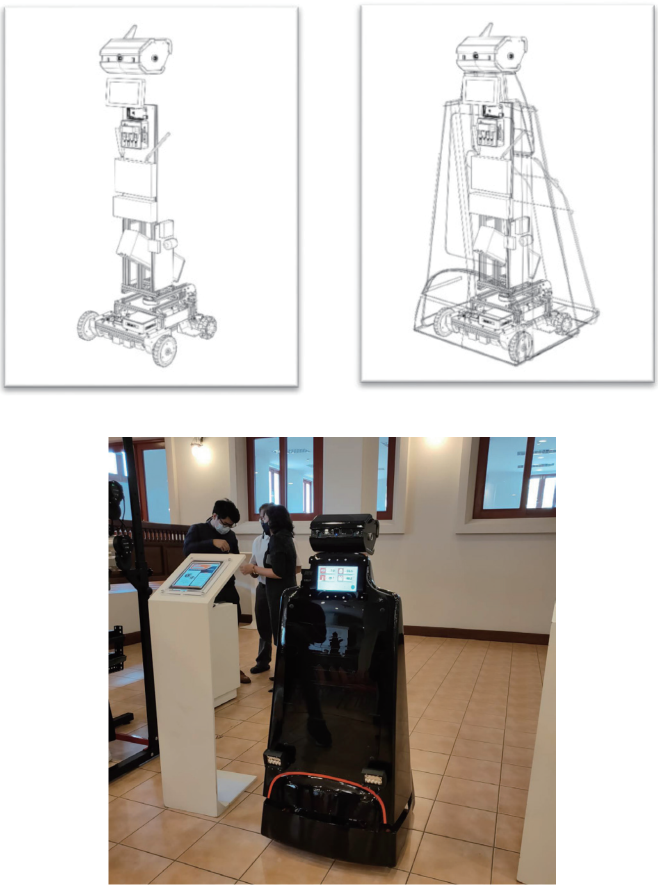

Fig. 1. (Color online) "Hello Steve", a companion robot.

### 2.1 Design of voice assistant functions

Voice assistant functions are designed on the basis of neuro-linguistic programming (NLP). In general, NLP frameworks can be divided into seven elements: segmentation, tokenizing, stop words, stemming, lemmatization, speech tagging, and name entity tagging.[17] The purpose of NLP is generally to mimic human languages. In other words, our goal is to create a personal assistant that is on standby at all times. Therefore, the system must be able to understand human speech and identify a given condition.

Once the bot receives a human voice command, it must execute that command as required. Therefore, the chatbot execution can be summarized into the four simple steps shown in Fig. 2. Firstly, the speech recognition step converts human speech into text. Secondly, the bot compares the text with conditions. Thirdly, the bot processes the conditions. Finally, the bot responds with an action or speech by transforming text into speech.

The overall design of the chatbot-like functions include the following.

(1) Voice commands for using YouTube.
- Voice command to open application.
- Voice command to search.
- Voice command to play video.

(2) Voice commands for using Google.
- Voice command to open application.
- Voice command to search.

(3) Voice commands for using Gmail.
- Voice command to send email.
- Voice command to name subject.
- Voice command to select who to send email to (based on name).
- Voice command to give details of email.

(4) Voice commands for reminder system.
- Voice command to initiate reminder system.
- Can freely choose what to be reminded of.
- Can set timer.
- Work in parallel with main program.

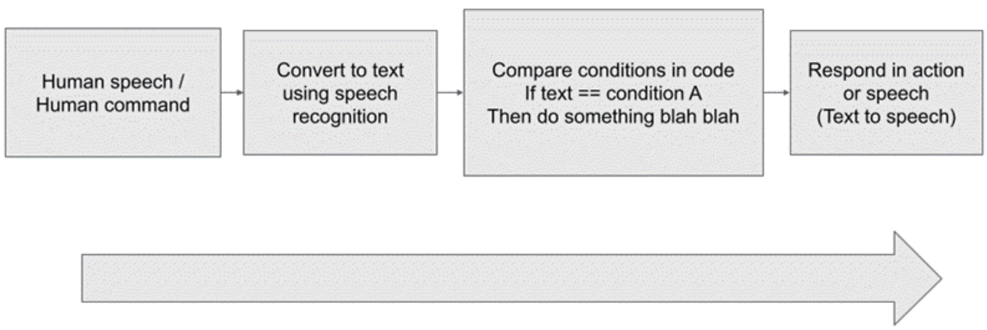


Fig. 2. Working concept of a voice chatbot.

(5) Voice commands for taking notes.
- Voice command to initiate note taking.
- Voice command to write details of note.

To demonstrate how the concept shown in Fig. 2 is executed, as an example, the Play YouTube command is discussed. Giving the command word "play" will allow the bot to play a specific video on the application. As shown in Fig. 3, using a voice command to name the video that the user wants the bot to play will result in the desired video being played. A block diagram corresponding to the event in Fig. 3 is given in Fig. 4.

As the name implies, the "wake word" to prepare the robot to receive a command is "Hello". Since the former name of the robot before integrating the chatbot feature was Steve, we call the upgraded robot "Hello Steve". As shown in Fig. 4, the user can give commands using voice assistant functions to open YouTube, play any video on YouTube based on a voice input, and search for any video on YouTube via voice input. Section 3.1 provides test results.

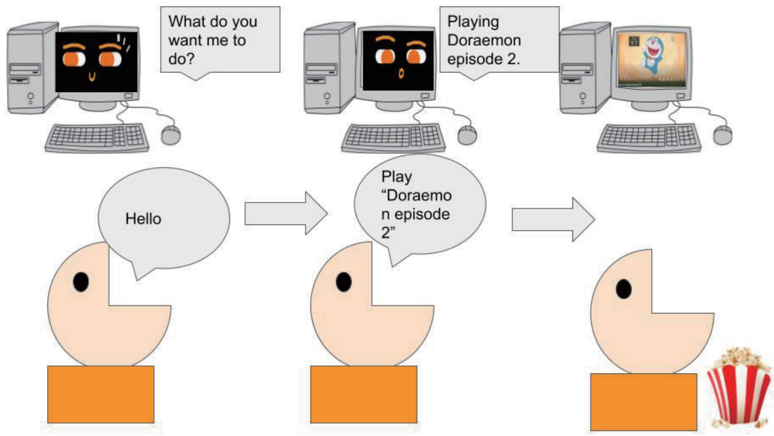


Fig. 3. (Color online) Example of using YouTube play command.

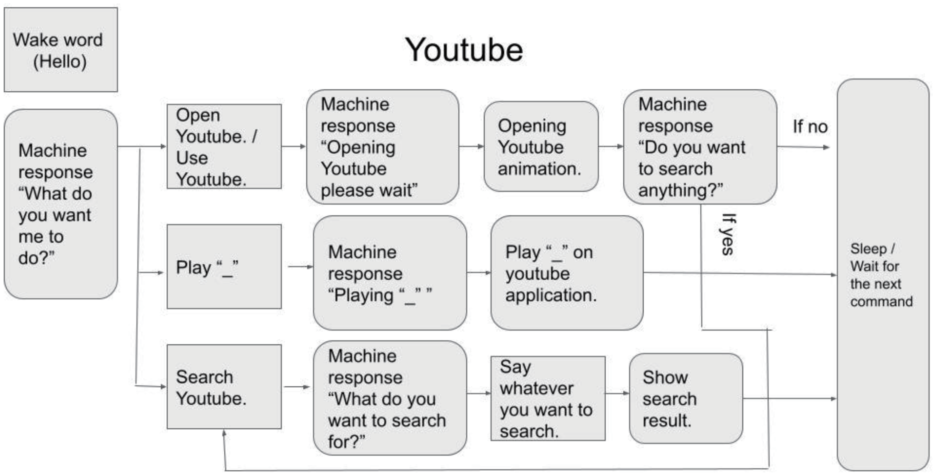


Fig. 4. Block diagram of the command functions using YouTube.

### 2.2 Localization and map-making algorithm

A localization and map-making algorithm for "Hello Steve" is quickly implemented by using classic Bayesian conditional probability.(18) A simple navigation concept for detecting the gateway to the next room is divided into two steps: Step 1, move the robot toward the goal; Step 2, follow the wall until it finds the next gateway. To reduce uncertainty, an empirical method for generating a sensor model is employed to collect data, in which the frequency of a correct reading of the sensor must agree with a set of beliefs from all possible observations.(18) Sensors such as sonar and LiDAR sensors can be used to observe whether an element *Grid*($i$, $j$) on a 2D space map is occupied or empty.

The map-making process of the robot involves transforming a set of data obtained from local sensor observations into a global map. The process uses a robot-centered approach and is independent of the robot position. An occupancy grid, which is a 2D array defined by coordinates ($i$, $j$) in a rectangular space, is normally used to represent metric maps. An occupancy grid with higher accuracy can be obtained by fusing multiple sensor readings. Update rules and sensor models are required to combine the uncertainty for a pair of observations. This can be performed by using Bayesian methods, where a probabilistic sensor model or an approximated belief function is used when a statistical average is difficult to obtain.

In the Bayesian approach, the conditional probability $P(B|s)$ that a particular event $\boldsymbol{B}$ (approximated by a belief) actually occurs given a particular sensor reading $s$ can be expressed as

$$P\left(B\,|\,s\right)=\frac{P\left(s\,|\,B\right)P\left(B\right)}{P\left(s\,|\,B\right)P\left(B\right)+P\left(s\,|\,\neg B\right)P\left(\neg B\right)}. \tag{1}$$

Since $\boldsymbol{B} = \{B, \neg B\}$, where $B$ is occupied and $\neg B$ is empty, and $P(B|s)\ P(s) = P(s|B)\ P(B)$, Eq. (1) can be rewritten using $m$ multiple observations, $s_i$ ($i$ = 1, 2, 3, …, $m$), as

$$P\left(B\,|\,s_m\right)=\frac{P\left(s_m\,|\,B\right)P(B\,|\,s_{m-1})}{P\left(s_m\,|\,B\right)P\left(B\,|\,s_{m-1}\right)+P\left(s_m\,|\,\neg B\right)P(\neg B\,|\,s_{m-1})}. \tag{2}$$

Note that Eq. (2) will be updated each time a new observation is made, and the new value will be stored at *Grid*($i$, $j$). Since the commutative law is applied, two or more simultaneous readings yield the same result. Section 3.2 provides test results.

## 3. Results and Discussion

### 3.1 Voice command tests

Our experiment simulates the designed features of voice assistant functions to test the interaction between a human and the robot, as shown in Fig. 5. The voice command tests were simply carried out by commanding the robot and recording the response. Each test command in Table 1 was given 100 times for each item by using different words.

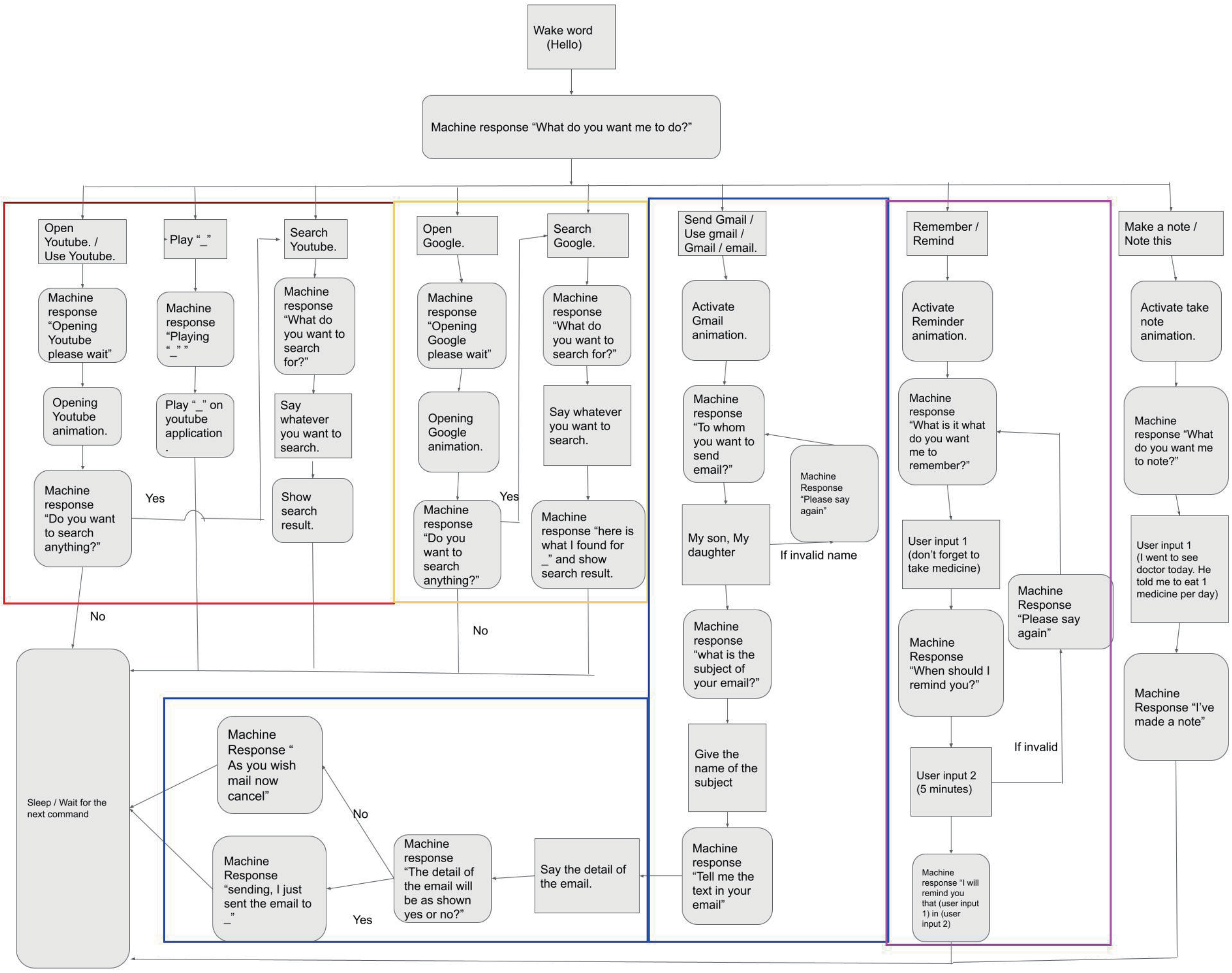


Fig. 5. (Color online) Block diagram of testing the command functions.

Table 1
Accuracy of voice command tests.

| Function | Specific tasks | Accuracy (%) |
|---|---|---|
| (1) Voice commands for using YouTube. | 1.1 Voice command to open application. | 95 |
| | 1.2 Voice command to search. | 92 |
| | 1.3 Voice command to play video. | 91 |
| (2) Voice commands for using Google. | 2.1 Voice command to open application. | 94 |
| | 2.2 Voice command to search. | 91 |
| (3) Voice commands for using Gmail. | 3.1 Voice command to send email. | 93 |
| | 3.2 Voice command to name subject. | 89 |
| | 3.3 Voice command to select who to send email to (based on name). | 86 |
| | 3.4 Voice command to give details of email. | 85 |
| (4) Voice commands for reminder system. | 4.1 Voice command to initiate reminder system. | 93 |
| | 4.2 Can freely choose what to be reminded of. | 85 |
| | 4.3 Can set timer. | 90 |
| | 4.4 Work in parallel with main program | 85 |
| (5) Voice commands for taking notes. | 5.1 Voice command to initiate note taking. | 92 |
| | 5.2 Voice command to write details of note. | 88 |

As shown in Table 1, voice commands were given to use YouTube, Google, Gmail, the reminder system, and the note-taking system in this order. The highest accuracy was obtained for the voice commands to open YouTube (95%), open Google (94%), and send an email to a specified address (93%). However, when using voice commands to give details when writing subjects, writing contents, and taking notes, the accuracy was reduced to around 85%.

The high accuracy of using voice commands in open applications may be due to the commands being in common use, meaning that the chatbot has learned and can easily recognize these commands. In addition, the commands contain messages of low complexity. In contrast, for the voice commands having contents with high complexity and less frequent use, the chatbot might not have learned the words before and might not recognize the commands. This might explain why the accuracy was reduced to around 85% for voice commands for recognizing tasks in specific detail. To improve the results, we recommend moving from the use of a voice command detector and a recognizer to the use of classifiers and custom training to handle more detailed commands; this should be addressed in future research.

### 3.2 Localization and map-making tests

Figure 6 shows the results of localization and map making using the algorithm described in Sect. 2.2 and a laser scan with a data sampling rate of 10 Hz. The sensor can scan in the range from 0.15 to 15 m. However, the actual maximum scan range is effectively around 11 m. There was around 5% error in the sensor range owing to reflection on shiny materials. The velocity of the robot during the test run was around 0.3 m/s. The naked "Hello Steve" robot without its cover shown in Fig. 6(b) made the map shown in Fig. 6(a). The red spot in the map in Fig. 6(b) was caused by a ray of sunlight penetrating from outside the glass windows. Such a problem is known as a cost map error, and this problem must be solved in future work.

To analyze the cost map error, the high intensity of the red spot perceived by the robot could be used as a key to address the sources of the errors. The error could be a limitation of Bayes'

(a) (b)

Fig. 6. (Color online) Performance of the proposed algorithm: (a) localization and map making and (b) cost map error.

rule in Eq. (2), which relies on evidence being represented by a probability function. A shortcoming of using a probability function is that if the robot is stationary and repeatedly returns the same range readings, the probability does not reflect the true events with respect to time. It is advantageous to collect data again at a different time to obtain two or more independent observations so that Bayes' theorem can be applied. This means that a better decision can be made from overlapping LiDAR or sensor readings at two different times.

However, in many real-world scenarios, results similar to Bayesian probabilities can be achieved using a possibilistic belief function. Even though the belief function provides only partial evidence, it offers useful information when combined with a supervisory rule for the robot to detect when the occupancy grid might be subject to errors. This may be a solution for the red-dot error observed in this study, which will be further investigated in future research.

## 4. Conclusions

A companion robot named "Hello Steve" was designed and implemented to serve as a case study for quick prototyping. According to the experimental results, for chatbot functions, a promising average accuracy of around 90% was achieved, indicating that the functions are usable at the laboratory level. Nevertheless, the accuracy must be improved before the robot can be used with elderly people. The integration of a new function such as ChatGPT[19] into the robot may also improve its performance related to real-time information, human-like interaction, answering questions on health, and other related topics. The implementation of a learning algorithm such as a deep learning algorithm should also be investigated in future work. Satisfactory results were achieved for the localization and map-making function, but the problem of the cost map error must be solved in future research, for example, by using algorithms or adding new fingerprint techniques.[20,21]

## Acknowledgments

This work was supported by MTEC under Grant 2564 and King Mongkut's Institute of Technology Ladkrabang under Grant 2566. The authors would like to thank Ms. Pravee Chanachavarin and her team for assisting in building "Hello Steve".

## References

1 S. Valtolina and L. Hu: Proc. Bianual Conf. (SIGCHI, 2021).
2 K. Sriwisathiyakun and C. Dhamanitayakul: Educ. Inf. Technol. **27** (2022) 6251. https://doi.org/10.1007/s10639-021-10862-z
3 C. Rodsawang, P. Thongkliang, T. Intawong, A. Sonong, Y. Thitiwatthana, and S. Chottanapund: OSIR (2020) 13.
4 E. Adamopoulou and L. Moussiades: An Overview of Chatbot Technology **584** (2020) 373. https://doi.org/10.1007/978-3-030-49186-4_31
5 S. Wollny, J. Schneider, D. D. Mitri, J. Weidlich, M. Rittberger, and H. Drachsler: Front. Artif. Intell. **4** (2021). https://doi.org/10.3389/frai.2021.654924
6 C. Getson and G. Nojat: Robotics **10** (2021) 106. https://doi.org/10.3390/robotics10030106

7 B. Isabet, M. Pino, M. Lewis, S. Benveniste, and A.-S. Rigaud: Int. J. Environ. Res. Public Health **18** (2021) 3597. https://doi.org/10.3390/ijerph18073597
8 S. Bao, B. Huang, J. Yuan, B. Wang, L. Xia, M. Wang, and Y. Liu : Sens. Mater. **33** (2021) 3187.
9 S. Denis, R. Berkvens, and M. Weyn: Sensors **19** (2019) 5329. https://doi.org/10.3390/s19235329
10 W. Ruan, Q. Z. Sheng, L. Yao, T. Gu, M. Ruta, and L. Shangguan: Proc. IEEE 17th Int. Symp. a World Wireless, Mob Multimed Networks (WoWMoM, 2016) 1–9. https://doi.org/10.1109/WoWMoM.2016.7523524
11 W. Ruan, L. Yao, Q. Z. Sheng, N. J. G. Falkner, and X. Li: Proc. 11th Int. Conf. Mob. Ubiquitous Syst. Comput. Netw. Serv. (MobiQuitous, 2014) 80–89. https://doi.org/10.4108/icst.mobiquitous.2014.258004
12 M. Scherhäufl, M. Pichler, E. Schimbäck, D. J. Müller, A. Ziroff, and A. Stelzer: IEEE Trans. Microw. Theor. Tech. **61** (2013) 4724. https://doi.org/10.1109/TMTT.2013.2287183
13 J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica: Sensors **17** (2017) 1. https://doi.org/10.3390/s17122898
14 J. R. Jiang, H. Subakti, and H. S. Liang: Sensors **21** (2021) 5434. https://doi.org/10.3390/s21165434
15 F. Zafari, A. Gkelias, and K. K. Leung: IEEE Commun. Surv. Tutorials **21** (2019) 2568. https://doi.org/10.1109/COMST.2019.2911558
16 A. Yassin, Y. Nasser, and M. Awad: IEEE Commun. Surv. Tutorials **19** (2017) 1327. https://doi.org/10.1109/COMST.2016.2632427
15 A. S. A. Sukor, L. M. Kamarudin, A. Zakaria, N. A. Rahim, S. Sudin, and H. Nishizaki: Smart Cities **3** (2020) 444. https://doi.org/10.3390/smartcities3020024
16 J. Zheng, C. Wu, H. Chu, and Y. Xu: Procedia Eng. **15** (2011) 876. https://doi.org/10.1016/j.proeng.2011.08.162
17 D. Khurana, A. Koli, and A. Khatter: Multimed. Tools Appl. (2022). https://doi.org/10.1007/s11042-022-13428-4S
18 H. P. Moravec: AI Mag. **9** (2021).
19 E. Mollick: Harv. Bus. Rev. (2022). https://hbr.org/2022/12/chatgpt-is-a-tipping-point-for-ai (acccessed January 2023).
20 F. A. Monero, J. Monroy, J. R. Ruiz-Samiento, C. Calendo, and J. Gonzalez-Jaminez: Sensors **20** (2020) 240. https://doi.org/10.3390/s20010240
21 D. J. Suroso, P. Cherntanomworng, and P. Sooraksa: Sens. Mater. **33** (2021) 4331. https://doi.org/10.18494/SAM.2021.3632


## About the Authors

**Thanapat Lappromrattana** is pursuing a B.Eng. degree in robotics and artificial intelligence (RAI) at King Mongkut's Institute of Technology Ladkrabang. He was a trainee in the 5G Department at True Corp. His research interests are in NLP, human–robot interactions, chatbots, and voice commands in robotics. (62011272@kmitl.ac.th)

**Pitikhate Sooraksa** is a professor at the School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand. He received his B.Ed. (Hons) and M.Sc. degrees in physics from Srinakharinwirot University, Thailand, his M.S. degree from George Washington University, USA, in 1992, and his Ph.D. degree from the University of Houston, USA, in 1996, both in electrical engineering. His research interests include cyber-physical applications and rapid prototypes in robotics and AI. (pitikhate.so@kmitl.ac.th)