Leveraging ROS to Support LLM-Based Human-Robot Interaction

Walleed Khan[1][a], Deeksha Chandola[1][b], Enas AlTarawenah[1][c],
Baran Parsai[1][d], Ishan Mangrota[2][e] and Michael Jenkin[1][f]
[1]*Electrical Engineering and Computer Science, Lassonde School of Engineering, York University, Toronto, Canada*
[2]*Information Technology, Netaji Subhas University of Technology, New Delhi, India*
walleedk@yorku.ca, enaskt2@yorku.ca, baranprs@my.yorku.ca, ishan.mangotra.ug22@nsut.ac.in, jenkin@yorku.ca

Keywords: Robot Control, Avatar-Based Interface.

Abstract: Large Language Model (LLM)-based systems have found wide application in providing an interface between complex systems and human users. It is thus not surprising to see interfaces between autonomous robots also adopting this strategy. Many modern robot systems utilize ROS as a middleware between hardware devices, standard software tools, and the higher level system requirements. Here we describe efforts to leverage LLM and ROS to provide not only this traditional middleware infrastructure but also to provide the audio- and text-based interface that users are beginning to expect from intelligent systems. A proof of concept implementation is described as well as an available set of tools to support the deployment of LLM-based interfaces to ROS-enabled robots and stationary interactive systems.

1 INTRODUCTION

As robots move out of the lab and into the world there is an increasing need to focus on developing robots that humans are willing to engage and interact with. Supporting this interaction may involve developing robots that provide a face, either realistic or cartoonish, to provide a focus for interaction. Adding a face to a robot can be beneficial (Altarawneh et al., 2020) but incorporating a visual display such as that shown on the robot in Figure 1 requires providing a software infrastructure that supports the integration of the visual appearance (an avatar) with the robot. Adding such a display also introduces the need to animate the avatar and provide mechanisms to drive the avatar with realistic speech. Early efforts, such as the one shown in Figure 1 relied on pattern-based chatbot technology to provide responses to queries of the robot. The development of Large Language models (LLMs) provides a more effective mechanism to drive the interaction.

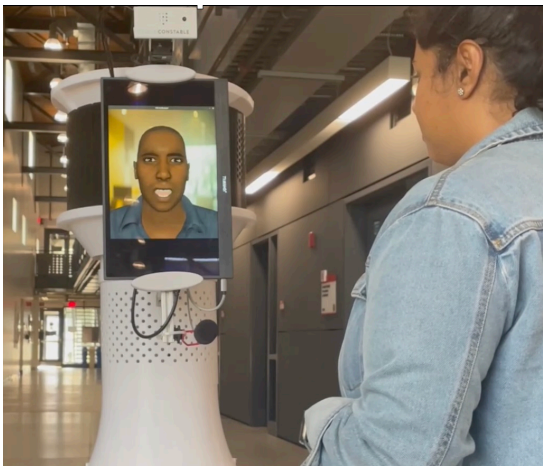

Figure 1: Interacting with a robot equipped with a visual avatar. Developing a system that provides such interaction mechanisms requires a software infrastructure to support generation and interaction with the embedded visual display.

There have been a number of efforts to leverage the capabilities of Large Language and Foundational Models to support the process of developing useful and user friendly software for robot control. For example, the `ROScribe` package (Technologies, 2005) can be used to assist in the development of novel ROS packages while Mower et al. (2024) describes a system in which an LLM constructs plans from a set of atomic actions and standard mechanisms to assemble them. But LLMs also find application in terms of transforming standard user interaction mechanisms

[a] https://orcid.org/0000-0002-8945-4329
[b] https://orcid.org/0000-0003-3654-0351
[c] https://orcid.org/0009-0004-8744-1139
[d] https://orcid.org/0009-0004-1329-7429
[e] https://orcid.org/0009-0008-0766-8966
[f] https://orcid.org/0000-0002-2969-0012

Khan, W., Chandola, D., AlTarawenah, E., Parsai, B., Mangrota, I. and Jenkin, M.
Leveraging ROS to Support LLM-Based Human-Robot Interaction.

In *Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025) - Volume 2*, pages 402-409
ISBN: 978-989-758-770-2; ISSN: 2184-2809

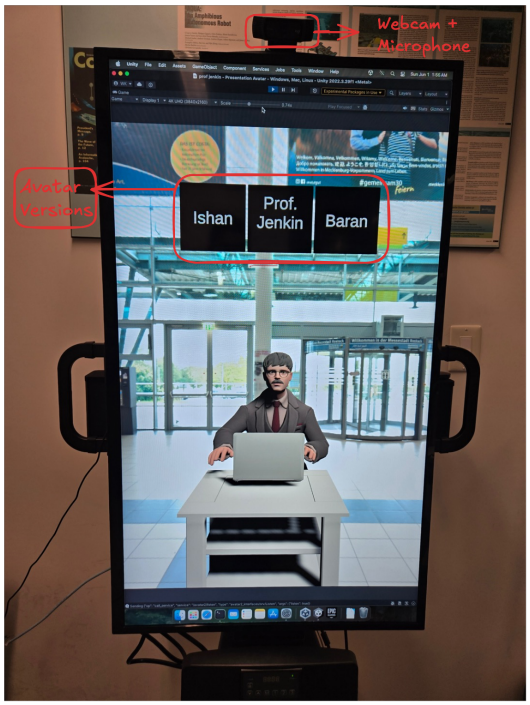

Figure 2: LLM-based interaction technology can be used to support a stationary user interface display as shown here. The software shown here supports multiple visual avatars. User input is via visual and audio channels and the user can choose between different animated avatars.

into specific robot actions. For example, Wang et al. (2024) describes a system that monitors the user and then uses an LLM to transduce multi-modal communication into commands to the robot. Here we are interested in leveraging an LLM to provide a more natural interface to a user interacting with a mobile robot or with a stationary avatar-based system such as the one shown in Figure 2.

Merging LLM-based systems with robot control software involves integrating two very different middleware software architectures. A common strategy in LLM-based systems is the use of a retrieval augmented generation (RAG) approach (Gao et al., 2024). In contrast, many modern robot systems utilize ROS (Macenski et al., 2022) as a middleware to structure the software infrastructure. ROS (the current common version is ROS 2), is a message passing paradigm in which messages are strongly typed and individual nodes operate in parallel. Integrating these two architectures can be challenging. ROS involves a parallel message passing architecture while RAG-informed LLM systems can be modelled as a query-response architecture. Here we explore the integration of these two architectures to develop a system that enables a robot or an avatar system using ROS to leverage advances in LLM-based interaction.

The remainder of this paper is organized as follows. Section 2 describes previous efforts that integrate LLMs in robot systems, with particular emphasis on leveraging LLMs to support human-robot interaction (HRI). Section 3 describes how ROS and RAG-LLM systems can be integrated together. Section 4 provides a simple example of how this combined architecture supports personalized HRI while retaining a standard ROS environment for robot control. Finally, Section 5 summarizes the work and describes ongoing work on RAG-LLM-ROS integration. The Avatar2 software package described here is available on GitHub at https://github.com/YorkCFR/Avatar2.

2 PREVIOUS WORK

There have been a number of efforts to leverage LLMs to support robot-related tasks from path-planning to learning from demonstration. See Wang et al. (2025) and Jeong et al. (2024) for recent reviews. Here we concentrate on the use of LLMs to support human-machine interaction and human-robot interaction in particular.

Perhaps the most commonly encountered use of LLMs for human-machine interaction is via a chatbot, a program designed to simulate conversation with a human. Very early chatbots (e.g., Eliza – Weizenbaum 1966) were based on simple pattern matching. However, since the introduction of LLMs, LLMs have found wide application in the development of chatbots (see Dam et al., 2024 for a review). Fundamentally, LLMs are trained on an extremely large corpus of textual data and develop a model that given a portion of a text stream can predict the next textual token that should appear. Starting with an initial text prompt, this process can be applied recursively to generate a response to a given prompt. Internally, LLMs utilize a transformer architecture and an attention mechanism. The resulting trained architecture has typically been trained on a large and general corpus of data. This results in an effective text-based chatbot that can be used to generate a realistic response in a conversation or provide an answer to a given question. It is critical to recognize the limitations of the approach, however. An LLM trained on a general corpus of data, e.g., by scraping the internet, will not necessarily contain only truths, and the process of generalizing a response to a given token sequence can result in hallucination in the response. Detecting and dealing with hallucinations can be a challenging task. See Luo et al. (2024) for details.

Given their ability to engage in conversations,

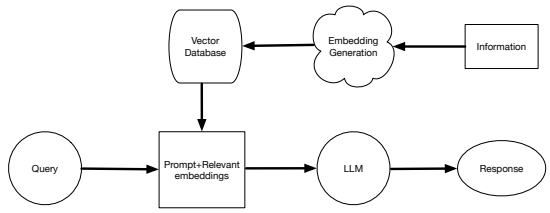

Figure 3: RAG-based interaction.

LLM-based systems have found application in systems that wish to engage the user in conversation. For example, Shoa and Friedman (2025) leverages LLM to deploy virtual humans in XR environments. In terms of robots, Kim et al. (2024) explores the impact of LLM-powered robots to engage in conversation with users, and reports that LLM-powered systems can enhance the user expectations in terms of robot communication strategies. While Ghamati et al. (2025) explore the use of personalized LLM-based communication with robots using EEG data.

2.1 Retrieval-Augmented Generation (RAG)

Tuning an LLM to a specific task can be accomplished in a number of different ways. Fine tuning the network on data specific to a given domain is a popular approach, however it can be difficult and computationally very expensive to perform this fine tuning. Another, less computationally expensive approach, is to provide within the prompt given to the LLM specific textual information that is relevant to the query. The basic concept is sketched in Figure 3. Prior to interaction with the chatbot local information is embedded in some representation and stored in a vector database. This process typically requires the local documentation to be chunked into manageable pieces related to the chatbot's expected response length. When a query is received by the chatbot, the vector database is searched and relevant chunks are retried from the database and integrated into the LLM query. This focuses the LLM's response on the chunks recovered from the database. A number of different software libraries, including LangChain, have been developed to support this process and provide tools to encode and recover data chunks from the vector database.

3 INTEGRATING LLMS WITHIN THE ROS ECOSYSTEM

A key difference between the RAG-LLM and ROS architectures is the asynchronous message-passing approach of ROS and the synchronous query-response nature of RAG-LLMs. Furthermore, LLMs are known for their latency in generating a response. Most commercial remote (cloud-based) LLMs must deal with communication latency and potential computational delays. Locally hosted LLMs avoid this communication latency but must deal with more severe local computational issues. In either case, dealing with this latency involves structuring LLM query responses within the ROS framework to retain liveliness in the ROS environment.

The basic structure of the approach is shown in the ROS computation graph in Figure 4. This graph shows only those nodes and messages related to HRI. The core process takes input from the user, including audio and visual information. This information is processed asynchronously to monitor the user's interaction with the system. (Here we assume only a single user communicates with the robot at a time.) This information is then processed by a RAG-LLM implemented in LangChain. Output from the RAG-LLM's is then used to provide textual output which is converted to an audio signal which is rendered to the user. While this rendering is taking place, the audio input process is suppressed so that the robot does not respond to its own utterances.

The prompt for the LLM is informed by a RAG system that is tuned by the user to whom the system is communicating as well as being informed by the information contained within the ROS messaging system.

3.1 Tailoring the Response to the Individual

The system employs facial recognition capabilities to identify known individuals and personalize interactions based on their profiles. See Figure 5. This process also provides information about the user interacting with the robot that can be used to enhance the interaction process, and even to assist in ignoring users who are at some distance from the robot or avatar.

A standard face recognition system (`dlib`) based on HOG and SVM is used to recognize faces. This approach, introduced in Dalal and Triggs (2005) for body detection has been successfully adapted to face detection. (See Singh et al. 2020 for a review.) The largest identified face is then compared against previously captured snapshots of participant faces associ-

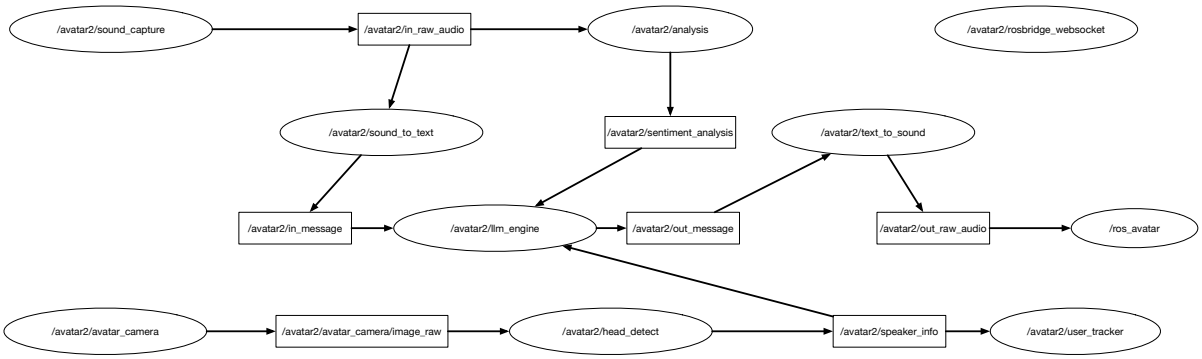

Figure 4: Basic structure of the approach. Standard ROS nodes are used to capture audio and visual interactions with the robot and either a 3d avatar (as shown in Figure 6) or 2d animation (as shown in Figure 7) is used to present audio responses and visual cues to the user. Shown here is a simple version of the approach using a 2D animation for presentation. The 3d version uses rosbridge to connect to the Unity-based avatar which also has access to the output of the user_tracker node which provides information to the animation system so that the avatar can focus on the individual who is interacting with the robot.

ated with this task to enable the identification of the individual who is communicating with the robot or avatar. Identified faces which are not identified as being members of this set of faces are labelled as 'unknown'.

As the bounding box of the face is known, it is possible to estimate a number of properties related to the user's visual interaction with the robot, including an estimate of the distance from the robot to the user. Based on the estimated distance, the nature of the interaction is defined in terms of proxemics (Hall, 1966). Specifically, the distance to the user is characterized as one of intimate, personal, social or public. Face tracking data is aggregated across time to characterize the current visual interaction between the robot/avatar and the user as being one of starting, continuing, or terminated. Conversations can be disrupted (another individual has been identified as the current speaker), idle (there is no one in the field of view of the camera), or looking (the robot was communicating with an individual who has not been detected for a short period of time). This information enables the LLM to incorporate information about the speaker (is this an ongoing conversation, for example) when formatting the LLM prompt.

3.2 Dealing with User Sentiment

A wide range of methods exist for identifying a speaker's emotional state, including those based on visual cues, textual analysis, and audio signals, as well as more recent approaches that combine multiple data types. Prior studies, such as Soleymani et al. (2017) and Tripathi et al. (2019), provide extensive reviews of these techniques. The basic approach here is to assign a one-hot encoded vector over a set of emotion labels – (Sadness, Excitement, Anger, Neutral, Happy, Fear, or Surprise) – to each user utterance. Although it would be possible to integrate multiple cues to the perceived emotional content of an utterance, for example, to combine text-based and audio-based emotion detection, here we concentrate on an audio signal-only approach that is based on the work of Tripathi et al. (2019). Audio signals are decomposed into a collection of audio features including Fourier frequencies and Med-frequency Cepstral Coefficients. These features are used within a deep neural network involving stacked LSTMs to map the audio signal onto the one-hot vector described above. The audio-only system in Tripathi et al. (2019) relied on the IEMOCAP dataset (Busso et al., 2008) for training and a smaller set of sentiment classes. For the system used here, we expand the set of sentiment classes to the seven provided above and used both the IEMOCAP and MELD (Poria et al., 2018) datasets to increase the size of the training dataset.

3.3 Monitoring the Robot System's State

Having the LLM monitor the robot system is straightforward as the RAG infrastructure has complete access to the ROS ecosystem. To take but one simple example, to expose the current pose of the robot to the LLM it is straightforward to add a statement such as

```
The robot is currently at location
x=3.0m and y=2.0m.
```

to the prompt. This can be done either automatically or more efficiently to only include such information if the user's query appears to contain key words such as 'location' or sequences like 'where are you'. As the RAG process is also aware of the user with whom it

Figure 5: Facial recognition system recognizing the individual interacting with the system. The system is able to retrieve the user information and pass it on in the ROS network with their id, name, and their role.

is conversing, the nature of the response can be tuned so that a response that is appropriate for a system developer (such as based on the property above) can be replaced with a property such as

```
The robot is currently near the
kitchen.
```

assuming that the location (3.0,2.0) in the global coordinate frame is near the kitchen.

3.4 Giving a Face to the Interaction

The avatar architecture described here draws on the Extensible Cloud-based Avatar framework described in Altaraweneh et al. (2021). This system serves as a puppetry toolkit compatible with the Robot Operating System (ROS). An overview of the avatar system architecture is depicted in Figure 4. In response to human interaction, the avatar system integrates the generated response into the avatar display. The rendering system, as described in Altarawneh and Jenkin (2020), combines speech audio with synchronized lip motion and expressive facial animations to produce coherent avatar responses, which are integrated into the display using idle loop animations. Rather than defaulting to a static avatar pose between responses, the system employs a dynamic idle animation to maintain an animated presence and to assist in masking any latency associated with responding to a user query.

Early implementations of this avatar rendering required an in-house computational and rendering cluster composed of multicore CPU servers equipped with GPUs and ample memory and storage, capable of supporting intensive animation tasks. The current implementation leverages standard video game assets that support non-player-characters (NPCs) and the Unity

Figure 6: Sample avatars. Constructing and rigging avatars is simplified by the existence of a number of standard toolsets that enable construction. These toolsets also support lip/mouth synchronization with audio utterances and the introduction of animations that mimic human (and other avatar) mannerisms to provide a feeling of naturalness to the agent being simulated.

Game Engine to render the avatar. These libraries also simplify the deployment of the delay loops described in Altaraweneh et al. (2021). Communication between the ROS and Unity spaces is provided by the UnityRos library described in Codd-Downey et al. (2014). Individual animated avatars are built using the `Ready Player Me`[1] toolkit, and idle animations are generated using `Mixamo`[2]. A view of a sample avatars created in this manner is given in Figure 6.

The rendering process has complete access to the ROS environment, including sentiment and visual information captured of the user. This would enable, for example, the avatar to direct its gaze at the user interacting with the robot and its avatar. It is important to observe that the display does not have to resemble a human (or even biological) entity. Figure 7, for example, shows a alternative 'avatar' that consists of a simple animated textured sphere whose appearance changes with the intensity of the audio signal being uttered.

4 AN EXAMPLE INTERACTION

To demonstrate the integrated system's capabilities in a real-world deployment, we present an interaction with an avatar assistant operating at the welcome desk at the ACME Hearing Clinic. The system shown here is operating using a Llama-3.1-8B model deployed on a dedicated RTX 4090 GPU-powered system. While this smaller model provides near real-time

[1] See readyplayer.me
[2] See mixamo.com.

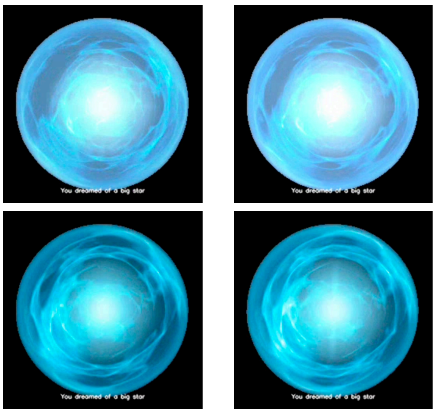

Figure 7: An alternative display 'face'. Rather than utilizing a 3d rendered avatar, other options are possible. Here an animated sphere is used to focus the user's attention. Four frames of the animation are shown.

performance, larger language models introduce more substantial delays in response generation, highlighting the critical trade-off between model sophistication and conversational responsiveness.

The system implements basic query caching during ongoing interactions, storing queries until a predefined cache size is full, to further reduce response times for subsequent queries. Upon response generation, the LLM output is broadcast as a ROS topic across the network infrastructure. The Unity-powered avatar system, described in Section 3.4, subscribes to these topic messages through the rosbridge websocket connection.

When the Unity system receives an incoming topic message containing the generated response, it initiates synchronized lip movements and contextually appropriate gestures throughout the duration of the utterance. These movements during speech represent a deliberate implementation of human conversational mannerisms, designed to enhance the perceived authenticity of the interaction. Upon completion of the spoken response, the avatar transitions to attentive listening behaviors, including subtle head movements mirroring the natural behavioral patterns observed in human-to-human conversation.

The following transcript demonstrates these integrated capabilities in action:

Mary (AI Assistant): `Welcome to the ACME Hearing Clinic! I'm Mary, one of our friendly and knowledgeable staff members. How can we help you today? Do you have an appointment scheduled or would you like to schedule one?`

Walk-in Patient: "I don't have an appointment there, but I wanted to ask you about your clinic hours and who will make it I might be able to see."

Mary (AI Assistant): `Hello! I'm Mary, a helpful assistant at the ACME Hearing Clinic. We're open from Monday to Friday at 9am till 5 pm. If you'd like to schedule an appointment, please feel free to call us at (555) 555-5555 or email us at ACMEhearingclinic@nowhere.na. We'll be happy to accommodate your needs! Please note that we may have limited availability on evenings and weekends, but we can try to work with you to find a time that suits you best.`

Walk-in Patient: "Can you tell me about who I might be able to see?"

Mary (AI Assistant): `Welcome to the ACME Hearing Clinic! I'm Mary, and I'll be happy to help you. As for who you might see here, we have a team of experienced audiologists and hearing specialists. They will assess your hearing and provide personalized recommendations for treatment. You may also meet with our speech-language pathologist (SLP) if you're experiencing difficulties with communication or understanding speech in noisy environments. We'll do our best to make sure you feel comfortable and supported throughout the process.`

Walk-in Patient: "Can you tell me about the resident doctor and the clinic?"

Mary (AI Assistant): `I'm Mary, a helpful assistant at the ACME Hearing Clinic. I'd be happy to tell you about our clinic and staff. Our clinic is located on the main floor of Life Labs at 99 Nowhere Road. We're a state-of-the-art facility with a team of experienced audiologists and hearing aid specialists. One of our resident doctors, Dr. Patel, is an expert in audiology and hearing health. She's been working with us for several years and has helped countless patients improve their hearing and quality of life. We also have a team of friendly and knowledgeable staff who are always happy to help answer any questions you may have about your hearing or our services.`

The system maintains consistent persona throughout the exchange, with Mary appropriately greeting the patient and maintaining a professional yet welcoming tone. The RAG system retrieves domain-specific information including precise operating hours, contact details, staff qualifications, and facility location, demonstrating knowledge base integration. However, the conversation also reveals ar-

eas for improvement in discourse management. The avatar exhibits redundant greeting behaviors, particularly noticeable in the second and third responses where Mary re-introduces herself despite already having established her identity. Currently, work is addressing this exact issue to reduce the redundant exchange and enhance the experience with a more sophisticated dialogue state tracking to maintain conversational coherence across multiple exchanges.

The response latency achieved through local GPU processing enables natural conversational flow, while the cached query system ensures that common inquiries about clinic hours and services are delivered with minimal computational overhead. Throughout each response, the Unity-powered avatar system synchronizes speech with appropriate gestures and maintains attentive listening behaviors between utterances, creating a compelling demonstration of naturalistic human-robot interaction.

5 ONGOING WORK

The RAG-LLM-ROS system has been deployed for a collection of different applications, including as modelling a welcoming avatar for a medical clinic. The use of ROS as a middleware enables the straightforward integration of visual and other sensor cues to the chatbot, enabling a high level of personalization without requiring a significant investment in software to process sensor data.

Current development efforts are focused on leveraging the system's facial recognition capabilities to implement role-based access control and personalized interaction management. Figure 5 showcases the recognition capabilities of the system. The existing user identification system, which successfully recognizes known individuals and characterizes interaction dynamics through proxemics analysis, is being extended to support hierarchical user privileges and administrative functions.

The system described here assumes that the avatar/robot does not have to respond to commands through executed actions. As a consequence the current system assumes that the output of the LLM is a text string that includes only the text to be presented through the animated interface. Ongoing work is exploring including basic robot actions (e.g., move to a given location) through the use of a structured LLM response, e.g., by having the LLM respond using a json structure that includes both motion commands as well as text to respond with).

Using LLMs to power natural and responsive Human-Robot Interaction (HRI) systems involves managing the inherent latency of Large Language Models (LLMs). The system described here utilizes ROS to create a robust and asynchronous architecture, yet delays arising from LLM processing, network communication, or local computation remain inevitable. Such delays disrupt the flow of interaction and negatively impact user experience as mentioned in Schoenberg et al. (2014) and Zhang et al. (2024). Current work on addressing this challenge seeks to manage the user's perception of latency by leveraging avatar behaviors that mimic human conversational cues during cognitive processing. Research has demonstrated that conversational fillers and accompanying gestures can significantly improve human-robot interaction by making responses appear more natural and reducing perceived delays (Wigdor et al., 2016). Ongoing research is developing and validates techniques that make unavoidable waiting periods feel like a natural part of the interaction, thereby enhancing the avatar's perceived responsiveness and naturalness.

ACKNOWLEDGEMENTS

The financial support the NSERC Canadian Robotic Network, Mitacs Globalink and the IDEaS SentryNet project are gratefully acknowledged.

REFERENCES

Altarawneh, E. and jenkin, M. (2020). System and method for rendering of an animated avatar. US Patent 10580187B2.

Altarawneh, E., M., M. J., and MacKenzie, I. S. (2020). Is putting a face on a robot worthwhile? In *Proc. Workshop on Active Vision and Perception in Human(-Robot) Collaboration*. Held in conjunction with the 29th IEEE Int. Conf. on Robot and Human Interactive Communication.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335.

Codd-Downey, R., Forooshani, P. M., Speers, A., Wang, H., and Jenkin, M. (2014). From ROS to Unity: Leveraging robot and virtual environment middleware for immersive teleoperation. In *IEEE International Conference on Information and Automation (ICIA)*, pages 932–936, Hailar, China.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, San Diego, CA.

Dam, S. K., Hong, C. S., Qiao, Y., and Zhang, C. (2024). A complete survey on LLM-based AI Chatbots. arXiv 2406.16937.

E. Altarawneh, E., Jenkin, M., and Scott MacKenzie, I. (2021). An extensible cloud based avatar: Implementation and evaluation. In Brooks, A. L., Brahman, S., Kapralos, B., Nakajima, A., Tyerman, J., and Jain, L. C., editors, *Recent Advances in Technologies for Inclusive Well-Being: Virtual Patients, Gamification and Simulation*, pages 503–522. Springer International Publishing, Cham.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. arXiv 2312.10997.

Ghamati, K., Banitalebi Dehkordi, M., and Zaraki, A. (2025). Towards AI-powered applications: The development of a personalised LLM for HRI and HCI. *Sensors*, 25.

Hall, E. T. (1966). *The Hidden Dimension*. Anchor Books.

Jeong, H., Lee, H., Kim, C., and Shin, S. (2024). A survey of robot intelligence with large language models. *Appl. Sci.*, 14:8868.

Kim, C. Y., Lee, C. P., and Mutlu, B. (2024). Understanding large-language model LLM-powered Human-Robot Interaction. In *Proc. ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, page 371–380.

Luo, J., Li, T., Wu, D., Jenkin, M., Liu, S., and Dudek, G. (2024). Hallucination detection and hallucination mitigation: An investigation. arXIV 2401.08358.

Macenski, S., Foote, T., Gerkey, B., Lalancette, C., and Woodall, W. (2022). Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7:eabm6074.

Mower, C. E., Wan, Y., Yu, H., Grosnit, A., Gonzalez-Billandon, J., Zimmer, M., Wang, J., Zhang, X., Zhao, Y., Zhai, A., Liu, P., Palenicek, D., Tateo, D., Cadena, C., Hutter, M., Peters, J., Tian, G., Zhuang, Y., Shao, K., Quan, X., Hao, J., Wang, J., and Bou-Ammar, H. (2024). ROS-LLM: A ROS framework for embodied AI with task feedback and structured reasoning. arXiv:2406.19741.

Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., and Mihalcea, R. (2018). Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Schoenenberg, K., Raake, A., and Koeppe, J. (2014). Why are you so slow? – misattribution of transmission delay to attributes of the conversation partner at the far-end. *International Journal of Human-Computer Studies*, 72(5):477–487.

Shoa, A. and Friedman, D. (2025). Milo: an LLM-based virtual human open-source platform for extended reality. *Frontiers in Virtual Reality*, 6.

Singh, S., Singh, D., and Yadav, V. (2020). Face recognition using HOG feature extraciton and SVM classifiere. *Int. J. of Emerging Trends in Engineering Research*, 8.

Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S.-F., and Pantic, M. (2017). A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14.

Technologies, R. C. (2005). Roscribe: Create ROS packages using LLMs. https://github.com/RoboCoachTechnologies/ROScribe, accessed Jun-30-2025.

Tripathi, S., Tripathi, S., and Beigi, H. (2019). Multi-modal emotion recognition on IEMOCAP dataset using deep learning. arXiv 1804.05788.

Wang, C., Hasler, S., Tanneberg, D., Ocker, F., F. Joublin, F., Ceravola, A., Deigmoeller, J., and Gienger, M. (2024). LaMI: Large Language Models for Multi-Modal Human-Robot Interaction. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, page 1–10. ACM.

Wang, J., Shi, E., Hu, H., Ma, C., Liu, Y., Wang, X., Yao, Y., Liu, X., Ge, B., and Zhang, S. (2025). Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence*, 4:52–64.

Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, page 36–45.

Wigdor, N., de Greeff, J., Looije, R., and Neerincx, M. A. (2016). How to improve human-robot interaction with conversational fillers. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 219–224.

Zhang, Z., Tsiakas, K., and Schneegass, C. (2024). Explaining the wait: How justifying chatbot response delays impact user trust. *ACM Conversational User Interfaces 2024*, page 1–16.