A Multimodal Interface for Real-Time Soldier-Robot Teaming

Daniel J. Barber[a], Thomas M. Howard[b], and Matthew R. Walter[c]

[a]University of Central Florida, Orlando, FL, USA
[b]University of Rochester, Rochester, NY, USA
[c]Toyota Technological Institute, Chicago, IL, USA

ABSTRACT

Recent research and advances in robotics have led to the development of novel platforms leveraging new sensing capabilities for semantic navigation. As these systems become increasingly more robust, they support highly complex commands beyond direct teleoperation and waypoint finding facilitating a transition away from robots as tools to robots as teammates. Supporting future Soldier-Robot teaming requires communication capabilities on par with human-human teams for successful integration of robots. Therefore, as robots increase in functionality, it is equally important that the interface between the Soldier and robot advances as well. Multimodal communication (MMC) enables human-robot teaming through redundancy and levels of communications more robust than single mode interaction. Commercial-off-the-shelf (COTS) technologies released in recent years for smart-phones and gaming provide tools for the creation of portable interfaces incorporating MMC through the use of speech, gestures, and visual displays. However, for multimodal interfaces to be successfully used in the military domain, they must be able to classify speech, gestures, and process natural language in real-time with high accuracy. For the present study, a prototype multimodal interface supporting real-time interactions with an autonomous robot was developed. This device integrated COTS Automated Speech Recognition (ASR), a custom gesture recognition glove, and natural language understanding on a tablet. This paper presents performance results (e.g. response times, accuracy) of the integrated device when commanding an autonomous robot to perform reconnaissance and surveillance activities in an unknown outdoor environment.

Keywords: Human Robot Interaction, Multimodal Communication, Automated Speech Recognition, Natural Language Understanding

1. INTRODUCTION

For several years now, the Department of Defense (DoD) has funded research to further the capabilities of robots to support advanced Soldier-Robot (SR) teaming as depicted in much of present day science fiction. Programs from the US Army Research Laboratory (ARL) such as the Robotics Collaborative Technology Alliance (RCTA) describe concepts where robots are no longer used as tools, but rather collaborators within mixed-initiative teams.[1,2] Recent advances fueled under these efforts have led to the development of novel robotic platforms incorporating new sensing capabilities enabling semantic navigation. With semantic navigation and mission planning capabilities, robots are now able to execute more complex and sometimes abstract commands such as "move quickly to the car behind the building".[3–5] To enable Soldiers to use these new platforms and their capabilities, communication interfaces must be enhanced to facilitate seamless integration and effective SR teaming.

One approach for advancing SR communication highlighted in literature is the use of multimodal communication (MMC). MMC is a model where six common themes emerge: meaning, context, natural, efficiency, effectiveness, and flexibility. With MMC, more complex information can be conveyed over multiple modes compared to single mode,[6] with ideas conveyed redundantly (back up signals) or non-redundantly (multiple messages).[7,8] Ultimately, MMC supports multiple levels of complexity.[6] Similar is the case for the natural theme,

Further author information: (Send correspondence to Daniel J. Barber)
Daniel J. Barber: E-mail: dbarber@ist.ucf.edu, Telephone: 1 407 882 1128
Thomas M. Howard: E-mail: thomas.howard@rochester.edu, Telephone: 1 585 275 3755
Matthew R. Walter: E-mail: mwalter@ttic.edu, Telephone: 1 773 834 3637

such that MMC results in more robust, natural, and efficient communication.[9] Leveraging natural forms of communication (e.g. speech and gestures) modeled after human-to-human communication is therefore most likely to support seamless integration of robots with their human counterparts without adding additional cognitive or task demands.

2. MULTIMODAL INTERFACE

In order to investigate the effectiveness of MMC based-on human-to-human communication for SR teaming, a prototype multimodal interface (MMI) was developed. The goal for this device was to enable real-time bi-directional communication with a robot teammate through natural and intuitive forms of communication (e.g. speech, gestures). For the present study, real-time is defined as how long it takes from the end of issuing a command until the interface completes input classification followed by transmission and receipt of the command with a robot. For efficient bi-directional communication in SR teaming, processing time for issued commands must be kept to a minimum, as delays can greatly impact system usability.[10]

Two modalities were selected in the design of the MMI, auditory (speech, audio cues, text-to-speech) and visual (gestures, tablet display), Figure 1. With different combinations of these modalities, the MMI enables a user to interact with a robot using single, dual, or redundant channels of communication. For example, a speech command to the robot with confirmation on a visual display, gesture command with response via text-to-speech, or speech with combined audio and visual feedback. Through flexible selection of communication channels, a user can maximize heads-up time without a requirement to look at a screen, but still have this information available for cases where the robot is no longer in line-of-sight or the data cannot easily be conveyed otherwise (e.g. image of unknown object).

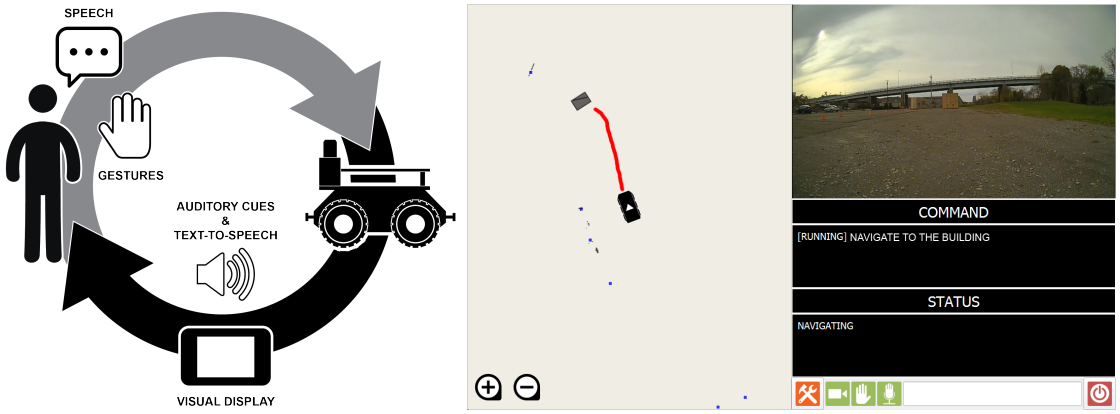

Figure 1. Illustration of bi-directional communication between a Soldier and robot (Left), and image of visual display from prototype MMI (right). For bi-directional communication, the Soldier issues commands using speech, gestures, or combination thereof, with responses from the robot delivered via auditory cues, text-to-speech, and visual display. The visual display of the MMI contains a top down interactive map, live video feed, current command, and robot status information.

Following the selection of speech and gestures for human-to-robot communication, the need for automated speech recognition (ASR), gesture recognition (GR), and natural language understanding (NLU) capabilities were further identified resulting in the system diagram illustrated in Figure 2.

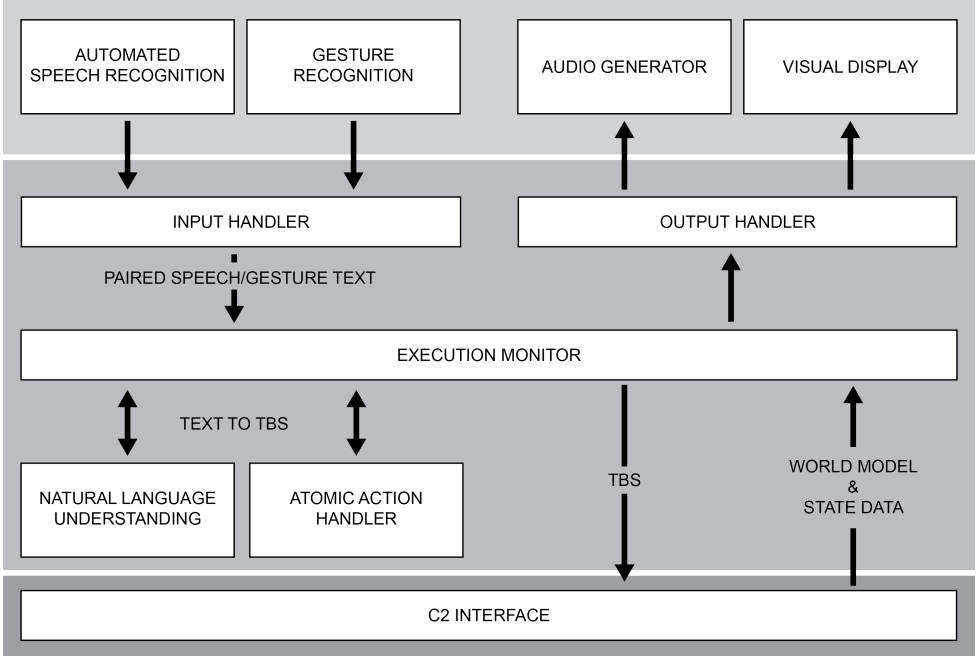

Figure 2. High-level diagram of multimodal interface components. Automated speech and gesture recognition modules convert user inputs to text. The input handler then sequences and/or pairs the text together and passes to the execution monitor. The execution monitor converts text to a robot command using natural language or atomic handler modules before sending to the robot using a command and control (C2) interface. Robot world model and state data from the C2 interface is updated within the execution monitor and passed to the output handler which triggers audio and updates the visual display.

2.1 Gesture Recognition (GR)

Arm and hand gestures are a natural form of communication for Soldiers, with many signals codified in the U.S. Army Field Manual for Visual Signaling.[11] For classification of arm and hand gestures, the MMI incorporated a glove embedded with an inertial measurement unit (IMU) and flex resistors, Figure 3.

Figure 3. MMI gesture recognition glove. A 3D printed box on the back of the glove houses electronics and power, with flex resisters sewn inside the glove along the finger tips.

The instrumented "gesture glove" was based off of a previous design described in Barber et al.,[12] and was selected for its high classification accuracy across many unique gestures, hands-free nature, and previous integration with robots. The gesture glove included much of the same hardware from the previous version, including a 9 degrees of freedom Razor IMU[13] containing single axis gyro, dual axis gyro, triple axis accelerometer, triple axis magnetometer, and flex resistors. The Razor IMU was selected for its low cost, ability to measure ± 16 g of force, magnetometers for pointing gestures, and expandability to support analog inputs from the flex resistors. The flex resistors sewn into the glove along the wearers fingers enable detection of finger positions (open or closed). Other hardware changes include a Bluetooth 4.0 module for wireless communication and an open-fingered glove with shorter flex resistors. The latter change was made to allow use of finger tips on touch screen devices, better fit the user, and to ensure flex resistors ended just past the wearers knuckles. In previous implementations of the gesture glove, depending on the wearers hand-size, the resistors could extend past finger tips resulting in the resistors becoming bent or pinched when a closed fist was made.[?] The final version of the glove, pictured in Figure 3, integrated all electronics within a custom 3D printed box on the back of the glove and supports charging using a Micro-USB cable.

For the current effort, the GR system supported a total of nine gestures: Forward, Backward, Left, Right, Clockwise, Counter Clockwise, Resume, Pause, and Pointing, Table 1. To perform each gesture, the user created a fist (closed hand) to signal the start, and when finished released the fist (open hand). The flex resistors in the glove detected when the wearer made and release a fist, passing the window of data collected during that time to the statistical classifier described in Barber et al.[12] The only exception to this process is for pointing gestures which are detected using static pose heuristics which monitor for combinations of flex resistor values indicating only the index finger is extended and lack of motion of the hand from IMU sensors. Although classified using the gesture glove, pointing gestures were not incorporated into robot commands at the time of this writing, but are planned for use in future efforts to resolve objects/points of interest in speech commands.

Table 1. Arm and hand gestures, resulting robot action, and description of how each gesture was performed.

Gesture	**Action**	**Gesture Motion**
Forward	Move forward	Elbow starts tucked into side, with forearm and bicep at 90 degrees. Arm extends forward (pushing out) until extended keeping hand parallel to ground.
Backward	Move backward	Arm starts extended out from body, parallel with ground, move elbow inward until tucked next to body (pulling towards body), with forearm and bicep at 90 degrees and hand still parallel to ground.
Left	Step left	Starting with arm extended out from body with hand parallel to the ground, swing left towards chest 90 degrees while keeping hand parallel to ground.
Right	Step right	Staring with arm to your left (against body) with fist palm-down down and parallel to ground, swing right 90 degrees while keeping hand parallel to ground.
Rotate Right	Rotate clockwise	Starting with arm extended out from the body with fist palm-down, perform a circle rotating clockwise with a diameter of approximately 1 foot.
Rotate Left	Rotate counter-clockwise	Starting with arm extended out from the body with fist palm-down, perform a circle rotating counter-clockwise with a diameter of approximately 1 foot.
Resume	Resume previous command/action	Hold the arm extended to the rear behind your head and swing the arm overhead and forward in the direction of movement. Finish with the palm facing down when arm is horizontal. Matches ADVANCE or MOVE OUT Visual Signal from Army Field Manual.
Pause	Pause execution of current command/action	Starting with arm at rest or or extended, rotate up until arm is bent at 90 degrees with fist palm-out away from body to the right of your head (hold gesture).
Pointing	N/A	Pointing gesture with index finger out and all other fingers closed. User points index finger at a location or point of interest.

2.2 Automated Speech Recognition (ASR)

More than gestures, speech is the primary method of human-to-human communication, enabling transmission of complex ideas and instructions. Speech is increasingly being incorporated into commercial-off-the-shelf products from gaming, mobile-devices, and personal computers.[14–16] With commercial applications driving requirements and capabilities, automated speech recognition (ASR) technologies have rapidly increased in their performance.

Several commercial-off-the-shelf software development kits facilitate capture and conversion of speech to text for use in custom applications. For the MMI, several products were reviewed, resulting in the selection of the Microsoft Speech Platform SDK Version 11.[17]

The Microsoft Speech Platform SDK was selected based on previous performance analyses comparing it to other commercial-off-the-shelf products. These results demonstrated a classification accuracy of 98.96% on a command set developed for a robot spatial navigation-task using a squad-level vocabulary (SLV).[18,19] The Microsoft Speech Platform SDK is a grammar-based classifier, requiring creation of a dictionary defining all combinations of speech the target system supports. Although grammar-based classifiers limit the use of full natural language when using speech to command a robot, the resulting performance is higher due to the smaller search space for grounding and reduced chance of misclassifying random speech utterances not intended for the robot.[18] One additional feature of the selected ASR is the ability to run off-line without a requirement for a connection to a remote server for classification. This off-line support enables the MMI to function in outdoor areas where a network connection is not possible, and without any added latency from communication with remote servers. To further reduce the chance of speech misclassification, all speech commands added to the grammar-dictionary required use of a robots' call-sign when giving an order. For example, "husky, navigate to the car near the building." Upon completion of classification, the ASR provided text and classification confidence are sent to the MMI Input Handler.

2.3 Input Handler and Command Generation

As illustrated in Figure 2, the Input Handler module receives all classified data from the GR and ASR software components. During classification, both components generate events indicating that a gesture or speech utterance has started which the Input Handler stores. Upon completion of classification, the Input Handler first verifies that the confidence value for a given classified input/text exceeds pre-defined thresholds. If this test is passed, the resulting text is sent to the Execution Monitor. In then event that more than one input modality is active at the same time, (e.g. user finishes gesture but is still speaking), the Input Handler will suspend full processing until both inputs complete execution. Once finished, the resulting gesture and speech inputs are validated and the pair is then sent to the Execution Monitor. This feature supports redundant gesture and speech input from the user (e.g. pointing while issuing navigate command with speech). If the gesture and speech outputs are equivalent, only the speech text is shared.

The Execution Monitor determines if a given command should be sent to the robot and what mechanism to use for conversion from text to a tactical behavior specification (TBS) message the robot understands (sent via the command and control (C2) interface). For example, if the robot is already in a "pause/hold" state, and a "pause" command is given, then no further action is required and the input is ignored. The Execution Monitor, using the robot's world model and state information, therefore determines when it is valid to send new commands to the robot and ensures command procedures the robot supports are followed. For conversion of user input to a TBS, the Execution Monitor first determines if it is an Atomic Command or natural language. An Atomic Command represents an execution state transition (i.e. pause, resume, abort) or basic movement operation (i.e. forward, backward, left, right, rotate left, rotate right). Due to the simple nature of Atomic Commands, a specialized handler (Figure 2) generated the corresponding TBS. A Natural Language Understanding component converted all other text to a TBS.

2.4 Natural Language Understanding (NLU)

The task of translating natural language commands into the TBS lexicon can be viewed as a probabilistic *grounding* problem that involves inferring a mapping between linguistic elements from the command and their corresponding TBS clause constituents. Many methods used by robots for natural language understanding, such as the Generalized Grounding Graph,[20] exhibit complexity that is proportional to the number of unique groundings for a phrase. Thus, most methods do not scale to structured languages of the size that we consider. The Distributed Correspondence Graph (DCG)[21] improves the tractability of probabilistic inference by assuming conditional independence across constituents (e.g., objects, constraints) of a grounding, thereby improving scalability with the number of constituents. The more recent Hierarchical Distributed Correspondence Graph (HDCG)[22] further improves the efficiency of inference by searching over a pair of graphical models to find physical meaning of an expression. The first model considers the set of rules that govern the structure of the second

model that is used to ground the utterance. The rules inferred according to the first model are then used to construct a compact representation of the grounding model that allows for efficient inference. The MMI used the HDCG for translating natural language into TBS instructions for the experiments described in Section 3. The symbolic representation in the first model consisted of 44 constituents per phrase that were used to represent the rules for permitting object, region, action, mode, and constraint types in the second model. The symbolic representation in the second model was composed to up to 2,341,684 constituents that represent all possible objects, regions, actions, modes, and constraints that could be expressed by this implementation of the TBS. We observed an average of 4,528 expressed TBS constituents per phrase in the second model over the 43 annotated natural language instructions in the training set, which produced the 1,320,768 examples that were used to train the log-linear model in the HDCG.

3. METHOD

The primary objective for the prototype MMI was to demonstrate successful real-time MMC with a robot, with average total computation time no greater than two seconds. To ensure this ability, each of the primary components (GR, ASR, NLU, and C2) of the MMI were tested individually with sample data to capture processing time per component. The robot used for the study was a modified Husky UGV[23] from Clearpath Robotics running the RCTA software architecture.[3,24,25] The accuracy of the GR, ASR, and overall ability to successfully execute HRI the MMI is not reported here as it has been covered in previous publications.[?,4,12,18]

For the GR module processing time for each of eight gestures mapping to Atomic Commands was measured, with 200 samples per gesture for a total of 1600. The ASR, NLU, and C2 components were tested using a pre-defined list of speech commands at varying levels of complexity. Each of the speech inputs mapped to a semantic navigation command. Semantic navigation commands describe a goal location to navigate to, and may include spatial relation constraints and modifiers related to how the goal should be approached. For the present study, seven semantic navigation command archetypes were tested, each with increasing complexity. The simplest semantic navigation command instructs the robot to navigate to an object, while the most complex does so while driving in a specific mode (e.g. quickly) and using multiple spatial constraints. A full list of the semantic navigation command archetypes with examples is shown in Table 2. Eight speech commands for each command archetype were used for a total of 56 inputs. For the ASR the time from end of a speech utterance until classification completed was measured for each commands ten times. Processing time to convert each command over ten iterations was also measured for both the NLU and C2 systems.

Table 2. Semantic navigation command archetypes and corresponding examples.

Number	Command Archetype	Example(s)
1	navigate to the [OBJECT]	navigate to the building. navigate to the traffic barrel.
2	navigate to the [SPATIAL REL] of the [OBJECT]	navigate to the front of the traffic barrel. navigate to the right of the fire hydrant.
3	navigate to the [SPATIAL REL] of the [OBJECT] [REL DIST] the [OBJECT]	navigate to the front of the traffic barrel near the building. navigate to the left of the traffic barrel near the fire hydrant.
4	navigate [DRIVE MODIFIER] to the [SPATIAL REL] of the [OBJECT] [SPATIAL REL] the [OBJECT]	navigate quickly to the front of the traffic barrel near the building. navigate quickly to the right of the traffic barrel behind the traffic barrel.
5	[SPATIAL REL APPROACH] to the [SPATIAL REL] of the [OBJECT] and navigate to the [OBJECT]	stay to the left of the building and navigate to the traffic barrel. keep to the right of the traffic barrel and navigate to the fire hydrant.
6	[SPATIAL REL APPROACH] to the [SPATIAL REL] of the [OBJECT] and navigate to the [SPATIAL REL] of the [OBJECT]	stay to the left of the building and navigate to the traffic barrel to the left of the building. keep to the right of the traffic barrel and navigate to the fire hydrant behind the building.
7	[SPATIAL REL APPROACH] to the [SPATIAL REL] of the [OBJECT] and navigate [DRIVE MODIFIER] to the [SPATIAL REL] of the [OBJECT]	stay to the left of the building and navigate quickly to a traffic barrel to the left of the building. keep to the left of the traffic barrel and navigate quickly to a fire hydrant behind the fire hydrant.
OBJECT: building, traffic barrel, fire hydrant, car SPATIAL REL: front, back, left, right, behind REL DIST: near DRIVE MODIFIER: quickly, covertly SPATIAL REL APPROACH: stay to the, keep to the		

4. RESULTS

4.1 Gesture Recognition (GR)

Analyses of gesture recognition processing revealed a median time of 0.0068 milliseconds (Minimum = 0.0056, Maximum = 0.00587, $SD = 0.0016$) to convert from raw sensor data to a classified gesture. A repeated measures

ANOVA found no significant differences in processing time between gestures ($p > .05$). This finding clearly demonstrates the advantage of a statistical classifier for fast on-the-fly calculations. Classification times for all gestures are shown in Figure 4.

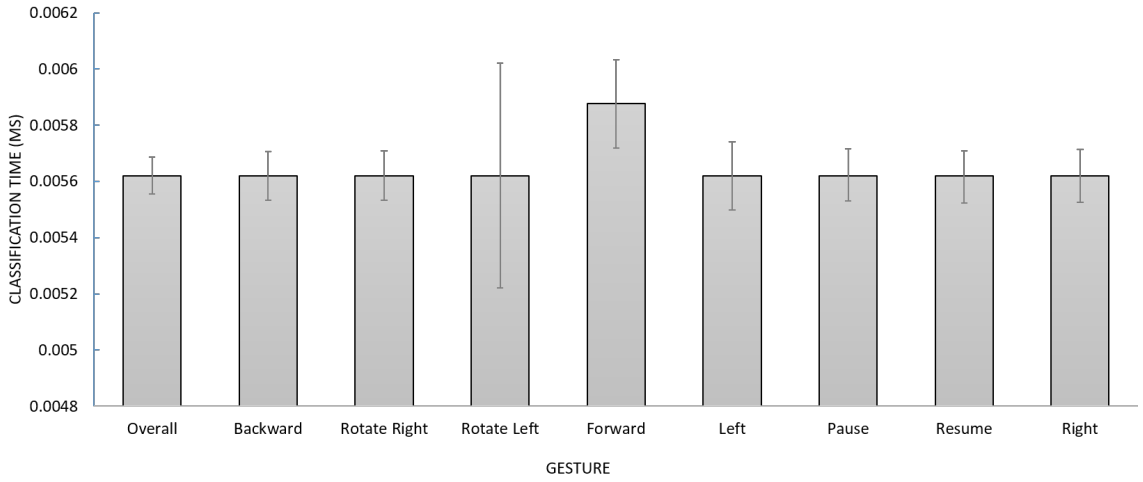

Figure 4. Median classification time in milliseconds for each over the eight gestures and overall. Error bars represent standard error.

4.2 Automated Speech Recognition (ASR)

Analyses of ASR processing revealed a median time of 6.5178 milliseconds (Minimum = 2.0046, Maximum = 12.0594, $SD = 1.16$) to classify speech utterances (regardless of classification accuracy). A repeated measures ANOVA determined there was no significant difference in classification time between command archetypes ($p > .05$). Speech recognition times for each command archetype are shown in Figure 5. Further evaluation of the accuracy of the Microsoft Speech Platform SDK 11 for ASR can be found in.[?, 18]

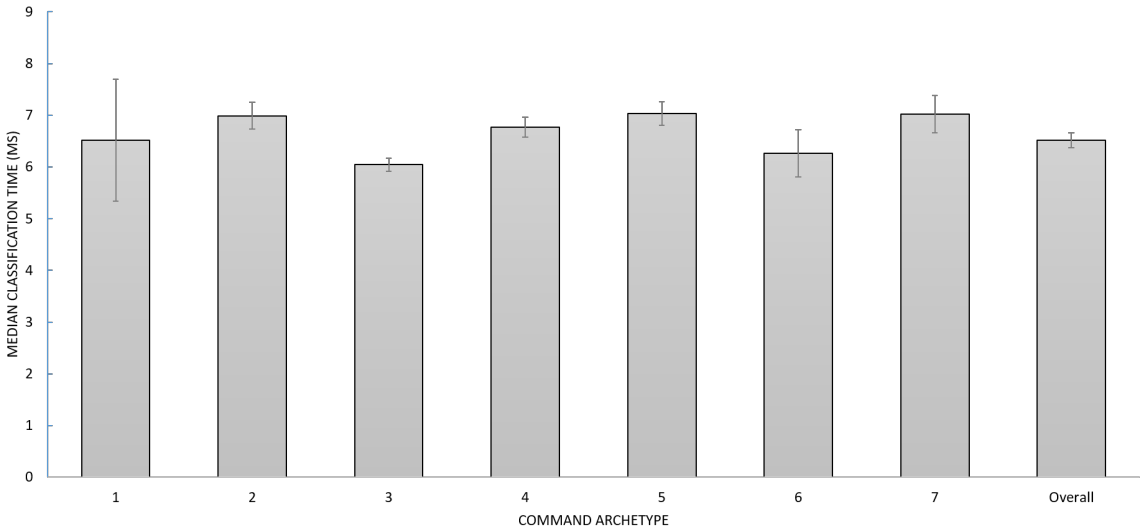

Figure 5. Median classification time in milliseconds for ASR across each command archetype.

4.3 Natural Language Understanding (NLU)

Analyses of NLU processing revealed a median time of 559.90 milliseconds (Minimum = 287.28, Maximum = 1692.01, SD = 275.81) to convert from text to TBS. A repeated measures ANOVA revealed a significant main effect for command archetype, $F(6, 18) = 19.82$ for $p \leq .0$. As illustrated in Figure 6, a trend in the data shows as command complexity increases (1 = lowest, 7 highest), so does processing time. Note, for all text inputs evaluated, a valid and correct TBS was generated.

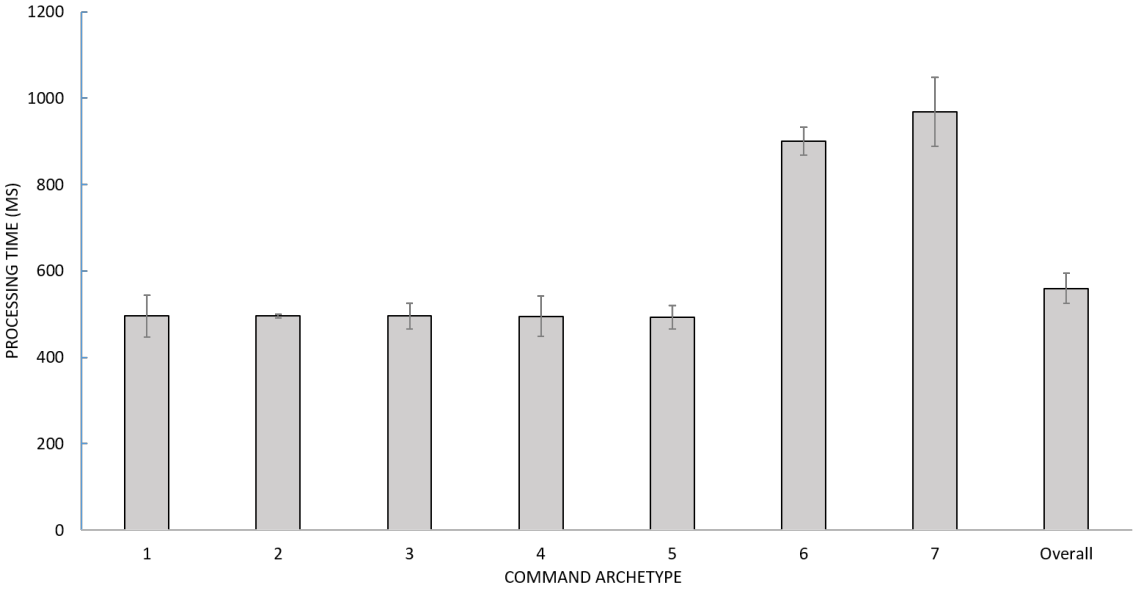

Figure 6. Median processing time in milliseconds for NLU across each command archetype.

4.4 Command and Control (C2)

Analyses of C2 processing revealed a median time of 2.92 milliseconds (Minimum = 2.33, Maximum = 7.26, SD = 0.953) to transmit a TBS to and receive a response from the robot over a standard wireless connection. A repeated measures ANOVA showed no main effect for transmission time and command archetype, (p >.05).

5. DISCUSSION

It is clear from the timing results presented that the integrated MMI is able to operate in real-time with minimal delay to the user when issuing commands. Combining processing time across analyzed components, times range from 291.62 ms to 1682.01 ms, with a median of 569.34 ms from start to finish. Through the use of a statistical classifier, the GR module has the shortest processing time of all the software components. However, the authors recognize that only 8 gestures were used for the present study, and therefore further testing should be performed to determine the cost in processing time (and possibly accuracy) with the addition of more gestures. This result would likely be affected from changes to the underlying classification method, especially one that is able to spot gesture start and end signals dynamically without the user making a fist. The Microsoft Speech Platform SDK selected for ASR also showed high performance across all tested speech and command archetypes. Moreover, it is interesting to note that regardless of the complexity of speech input, processing time was not impacted. This finding further indicates the maturity of modern ASR technologies and the benefits they have received from inclusion in consumer products. The NLU software component also showed high performance in terms of processing ability for real-time use. Although conversion from text to TBS was largest of all tested components, it is also the most complex and challenging aspect of the integrated MMI. As stated previously, for the current effort combined pointing gestures with speech were not included. It is possible that with the inclusion of gestures and

additional world model information, TBS grounding could be improved to address situations where it is unclear what object/point of interest a person is referring to. This enhanced functionality could result in different processing times for NLU, and should be addressed in future efforts.

ACKNOWLEDGMENTS

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0016. The views and conclusions contained in this document are those of the author's and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] U.S. Army Research Laboratory, "Robotics collaborative technology alliance." http://www.arl.army.mil/www/default.cfm?page=392 (2012). [Online; accessed 21-January-2016].

[2] Phillips, E., Ososky, S., Grove, J., and Jentsch, F., "From tools to teammates toward the development of appropriate mental models for intelligent robots," in [*Proceedings of the Human Factors and Ergonomics Society Annual Meeting*], **55**(1), 1491–1495, SAGE Publications (2011).

[3] Lennon, C., Bodt, B., Childers, M., Oh, J., Suppe, A., Navarro-Serment, L., Dean, R., Keegan, T., Diberardino, C., and Zhu, M., "An integrated assessment of progress in robotic perception and semantic navigation," tech. rep., DTIC Document (2015).

[4] Barber, D. J., Abich, J., Phillips, E., Talone, A. B., Jentsch, F., and Hill, S. G., "Field assessment of multimodal communication for dismounted human-robot teams," in [*Proceedings of the Human Factors and Ergonomics Society Annual Meeting*], **59**(1), 921–925, SAGE Publications (2015).

[5] Hemachandra, S., Duvallet, F., Howard, T. M., Roy, N., Stentz, A., and Walter, M. R., "Learning models for following natural language directions in unknown environments," in [*Robotics and Automation (ICRA), 2015 IEEE International Conference on*], (5 2015).

[6] Bischoff, R. and Graefe, V., "Dependable multimodal communication and interaction with robotic assistants," in [*Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*], 300–305, IEEE (2002).

[7] Partan, S. and Marler, P., "Communication goes multimodal," *Science* **283**(5406), 1272–1273 (1999).

[8] Parr, L. A., "Perceptual biases for multimodal cues in chimpanzee (pan troglodytes) affect recognition," *Animal cognition* **7**(3), 171–178 (2004).

[9] Mariani, J., "Spoken language processing and multimodal communication: A view from europe," in [*Plenary Talk, NSF Workshop on Human-centered Systems: Information, Interactivity, and Intelligence (HCS), Arlington, VA, USA*], (1997).

[10] Brooke, J. et al., "Sus-a quick and dirty usability scale," *Usability evaluation in industry* **189**(194), 4–7 (1996).

[11] U.S. Army, Washington, D.C., *FM 21-60 Visual Signals* (1987).

[12] Barber, D., Lackey, S., Reinerman-Jones, L., and Hudson, I., "Visual and tactile interfaces for bi-directional human robot communication," in [*SPIE Defense, Security, and Sensing*], 87410U–87410U, International Society for Optics and Photonics (2013).

[13] SparkFun Electronics, "9 degrees of freedom - razor imu." https://www.sparkfun.com/products/10736 (2016). [Online; accessed: 2016-03-16].

[14] Kattoju, R. K., Barber, D., Abich, J., and Harris, J., "Technological evaluation of gesture and speech interfaces for enabling dismounted soldier-robot dialogue," in [*SPIE Defense+ Security*], International Society for Optics and Photonics (2015).

[15] Microsoft, "Use kinect voice commands with xbox one." http://support.xbox.com/en-US/xbox-one/accessories/voice-commands (2016). [Online; accessed: 2016-03-20].

[16] Apple, "Ios siri." http://www.apple.com/ios/siri/ (2016). [Online; accessed: 2016-03-20].

[17] Microsoft, "Get started with windows 10: What is cortana." http://windows.microsoft.com/en-us/windows-10/getstarted-what-is-cortana (2016). [Online; accessed: 2016-03-20].

[18] Microsoft, "Software development kit (sdk) for the microsoft speech platform runtime 11." https://www.microsoft.com/en-us/download/details.aspx?id=27226 (2016). [Online; accessed: 2016-03-20].

[19] Harris, J. and Barber, D., "Speech and gesture interfaces for squad-level human-robot teaming," in [*SPIE Defense+ Security*], 90840B–90840B, International Society for Optics and Photonics (2014).

[20] Barber, D., Wohleber, R. W., Parchment, A., Jentsch, F., and Elliott, L., "Development of a squad level vocabulary for human-robot interaction," in [*Virtual, Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments*], 139–148, Springer (2014).

[21] Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S. J., and Roy, N., "Understanding natural language commands for robotic navigation and mobile manipulation," in [*AAAI Conference on Artificial Intelligence, North America*], AAAI Publications (2011).

[22] Howard, T. M., Tellex, S., and Roy, N., "A natural language planner interface for mobile manipulators," in [*Robotics and Automation (ICRA), 2014 IEEE International Conference on*], 6652–6659, IEEE (2014).

[23] Chung, I., Propp, O., Walter, M. R., and Howard, T. M., "On the performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions," in [*Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*], 5247–5252, IEEE (2015).

[24] Robotics, C., "Husky-ugv." clearpathrobotics.com/husky-unmanned-ground-vehicle-robot (2016). [Online; accessed: 2016-03-20].

[25] Dean, R. M. S. and DiBerardino, C. A., "Robotic collaborative technology alliance: an open architecture approach to integrated research," in [*SPIE Defense+ Security*], 90960M–90960M, International Society for Optics and Photonics (2014).

[26] Dean, R. M., Oh, J., and Vinokurov, J., "Common world model for unmanned systems: Phase 2," in [*SPIE Defense+ Security*], 90840I–90840I, International Society for Optics and Photonics (2014).