Mobile and Inflatable Interface for Human Robot Interaction

Jonathan Jaramillo, Andrew Lin, Emma Sung, Isabel Jane Hunt Richter, and Kirstin Petersen

Abstract— **Researchers hypothesize that networked physical robots may efficiently and safely interact with individuals in large crowds ranging from festivals and conventions to building evacuation. These scenarios pose a unique combination of hardware constraints involving safety for human contact, durability to withstand physical interaction, agility to keep up with moving crowds in non-trivial terrains, versatility to adapt to the situation and audience, and low cost to permit mass deployment. Here, we present a new mobile robot platform composed of a small rover base and a soft human-scale inflatable interface, capable of visual, tactile, and audible interaction with a variety of layman users. The inflatable interface allows the robot to maneuver discretely or in confined spaces when deflated, yet grow to encourage interaction; it combines an internal projector, a camera, speakers, and a microphone to emit and receive user information. The rover base is designed to keep up with humans at jogging speeds over relatively uneven terrain. Low weight further permits easy handling and transport. The entire robot costs less than 1.2K USD, and can serve as a general purpose, open source test platform for a range of future human-robot interaction research.**

I. INTRODUCTION

Populated settings such as festivals and museums call for efficient crowd management and information distribution. Researchers in the human-robot interaction (HRI) domain hypothesize that such a job may be performed well by autonomous mobile robots providing guidance or simply a situational overview. Such robots would hold special advantage in emergency situations, where deployment of human rescue workers poses added risks. We already see a similar trend in the commercial space, where human informants are replaced with robot receptionists, concierges, and waiters [1].

The field of HRI has focused mostly on the design of personable and intuitive user interfaces as well as high level autonomy and intent interpretation for such service robots [2], [3]. The design of the actual robot platforms is left largely to industry, and correspondingly there are a wide variety of robot informants available on the market, spanning applications from health care proxies to social companions, and a wide range of autonomy, price, size, and capability (Fig. 1). However, the majority of these platforms are rigid, expensive, slow or immobile, of fixed morphology, and restricted to indoor locomotion limiting their use to researchers who wish to explore interaction patterns in less structured settings. We argue that deployment of autonomous mobile robots for crowd management poses a unique combination of hardware requirements beyond the reach of existing robot platforms, including:

- **Multi-modal interaction.** Research has shown that active engagement of humans benefit significantly from more than standard visual displays [4], [5].
- **Safe interaction.** A simple way to ensure safe interaction is to rely on soft interfaces, as inspired by the field of soft robotics [6].
- **Adaptability.** To engage with a diverse audience, the platform must have an easily adaptable interface and be capable of changing its form factor [7].
- **Agility.** The robot must be able to move rapidly alongside people over uneven terrains and potentially in confined spaces [8].
- **Durability.** The robot must have sufficient battery life and require little maintenance, despite harsh environments or treatment.
- **Low cost and customization.** To permit mass deployment, customization, and easy adoption in research labs the robot must remain low cost, and easy to adapt and modify.

The closest related product reported in literature is the Puffy robot [9], an inflatable robot of similar appearance to the main character in the Big Hero 6 Disney movie, mounted on an iRobot® Create, designed for Children with neuro developmental disorders. However, due to our focus on a more general audience and deployment in crowds, the robot introduced here has a range of additional advantages in terms of interface adaptability, agility, and cost.

Specifically, we introduce a new HRI platform consisting

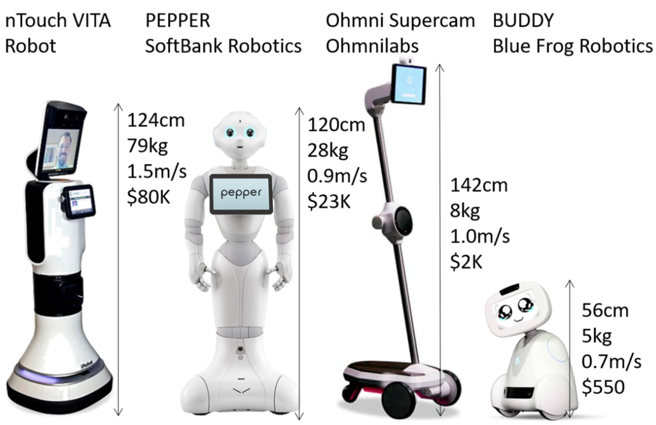

Fig. 1: Service sector robots span a wide range of applications, prices, sizes, and capabilities.

Funding for this research was provided by the NSF award number 1830471, GETTYLABS, and a Packard Fellowship for Science and Engineering.

[1] Cornell Engineering, Ithaca, NY 14853, USA. jdj78, yl656, hs699, ijr25, kirstin @cornell.edu

of an interactive inflatable interface mounted on top of a rover base capable of wireless remote control (Fig. 2). We nicknamed this robot Martha, after the Cornell President, Martha Pollock, who guided the university safely through the pandemic. The robot can facilitate engagement and safe human interaction through mobility, image projections, audio, gesture-based, and tactile feedback. The size of the inflatable interface can be also adjusted to fit the user. The robot weighs just 4.5 kg and can move at a maximum speed of 1.71 m/s corresponding to the lower end of an average human jogging speed. It measures 0.4 m in diameter and has a height between 0.3-1m depending on interface inflation. The robot can maneuver relatively rough terrains, such as carpets, tiles, and lawns; and easily traverse small obstacles such as uneven door frames and ramps. Because of its low weight, it is also easy for a user to pick up and carry past e.g. stairs or debris and big red handles provides an intuitive place to grasp the robot. The hardware is very modular and can easily be reconfigured. All software is written in Python running on a Raspberry Pi 4 embedded computer, with ample online support forums. To control the robot as is, we developed a multi threaded control architecture that enables users to control and stream data from the robot over a wireless network through a simple command line interface. Finally, the robot is made mostly from off-the-shelf parts and 3D printed components from low-end filament printers. It is at the lower end of the cost spectrum at just 1.2K USD. Upon acceptance of this article we intend to open source all CAD files and scripts to help other people replicate the design.

II. ROVER PLATFORM

The rover combines several modules which we briefly describe in the following section and in Fig. 3. Our design focus was on price, weight, durability, battery consumption, and ease of fabrication. With the exception of 3D printed parts and a foam core base plate, all components are off-the-shelf and easily obtainable. The hardware architecture is highly modular and extendable, and additional sensors/actuators can easily be added through SPI or UART interfaces, or simple digital I/O pins.

Chassis. The rover chassis is based on a simple open aluminum box measuring $0.2{\times}0.25{\times}0.08$ m^3, alongside several custom inserts printed in PLA on low-end filament printers (we used the Prusa mk II). These pieces are colored in turquoise, green, and red in Fig. 3 and include: motor clamps (6), a raised base plate to hold the electronic components and wires (11), stand-offs to mount the inflatable interface (12), a mount for the projector (17), a mount for three Time of Flight (TOF) sensors (10), and two large handles for transport (8). The robot weighs 4.5 kg, and all of the weight is located low to the ground ($<$ 0.2m) for more stable navigation over different terrains at relatively high speeds (Fig. 4).

Propulsion and Power. The robot locomotion is based on skid steering: four 0.12 m diameter rubber wheels (4) are mounted at each corner, driven by four 12 W motors (7) with a no load speed of 330 rpm and a stall torque of 14

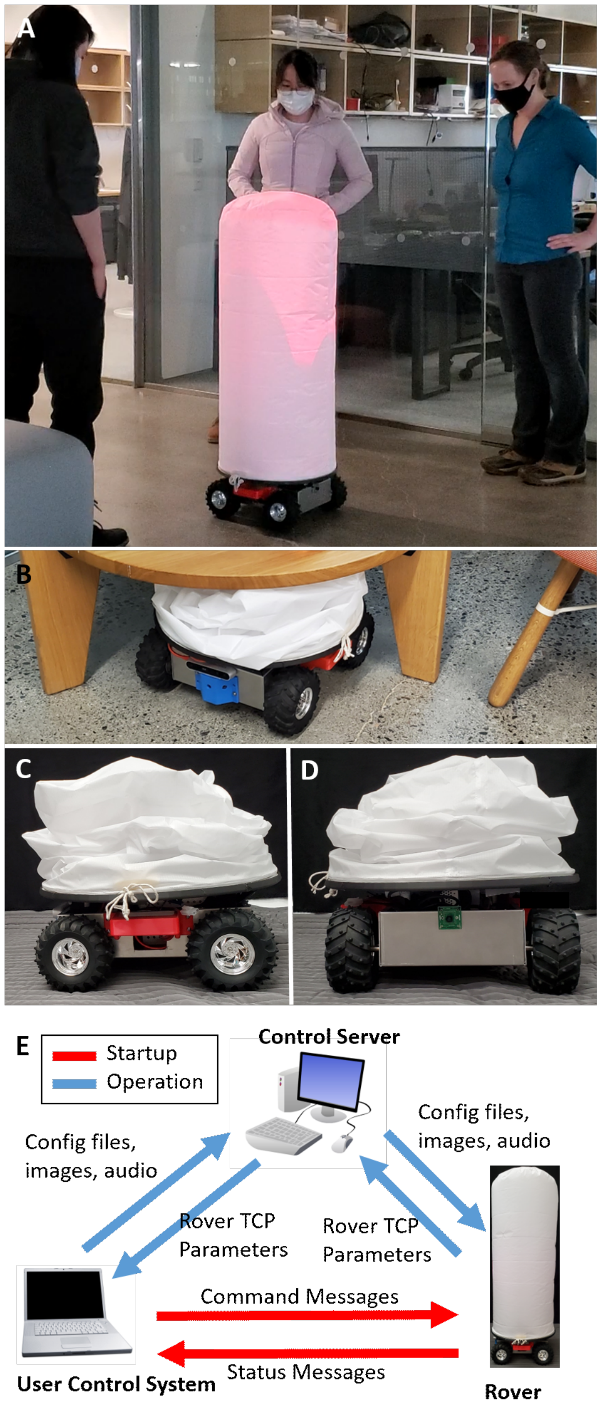

Fig. 2: We present an inexpensive, light weight, and agile remote-controlled mobile platform for visual, audible, and safe tactile human-robot interaction. A-B) The robot inflates to interact comfortably with people and deflates to fit under low clearance obstacles. C-D) Side and rear view. E) We created a custom control framework for easy control. Upon startup, the robot transmits TCP parameters to the control server and retrieves configurable files needed during runtime. The user control system retrieves the TCP parameters and creates a direct connection with the rover to control it.

Component	**Supplier**	**Unit Price**
Rover drive train and power		
Aluminum 8×10×3” open box [1]	Hammond	28.37 USD
LiPo battery, 11.1V/5200mAh [2]	Zeee	64.99 USD
DC-DC step-down regulator, 12-5V/3A [3]	Drok	15.50 USD
4×30:1 Metal gearmotor 37x52mm [7]	Pololu	24.95 USD
2×Simple HP motor controller 18v15 [13]	Pololu	46.95 USD
4×Wheel 120x60 mm, 4mm hub [4]	Dagu Thumper	7.48 USD
Computation and peripherals		
RPi 4 model B, 4GB [16]	Raspberrypi	61.30 USD
RPi camera v2 [2]2	Raspberrypi	9.99 USD
VGA super mini USB camera, 100d lens [5]	ELP	24.69 USD
RealSense Depth camera D435 [9]	Intel	169.99 USD
Zero4U 4 port USB hub [15]	Adafruit	9.95 USD
3× VL53L0X TOF sensors [10]	ACEIRMC	6.50 USD
3-axis acc/gyr (MPU-6050) [14]	Sparkfun	29.95 USD
P300 Neo projector [18]	AAXA	279.00 USD
Mini USB 2.0 Microphone Mic	Youmi	8.65 USD
Inflatable bladder		
Air fan (JDH7530S), 12v/0.5A, 80C [21]	Gemmy	16.24 USD
White ripstop nylon, 60×36”	Emmakites	9.00 USD
2×Dual TB6612FNG motor driver [23]	Sparkfun	4.95 USD
3×1000:1 micro-metal gearmotor, 6V [24]	Pololu	22.95 USD
3×Magnetic encoder, 12CPR	Pololu	3.98 USD
3×608-2RS Ball bearing [25]	Proigtz	1.20 USD
Total (incl. wires, PLA filament, tendons, and misc.)		∼ 1.200 USD

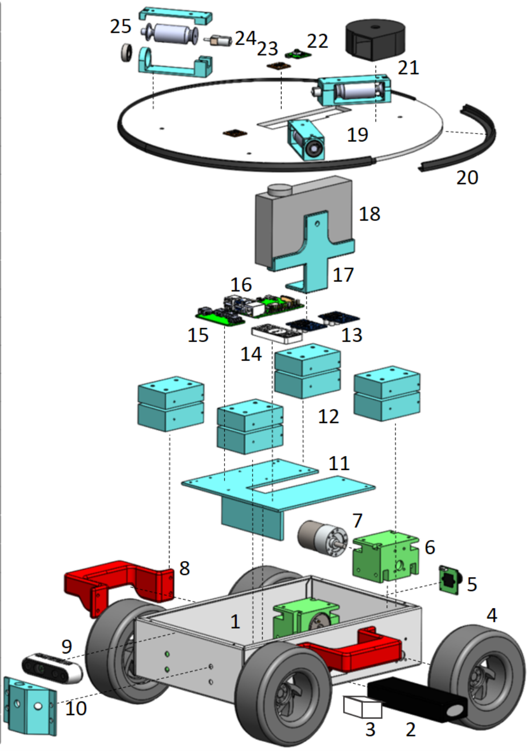

Fig. 3: Left: Table of major robot components. Right: Partially exploded view of the robot without the soft inflatable interface installed. Commercially available components are shown by superscript numbers in the table; items (6, 8, 10-12, 17, 25) were printed on low-end 3D filament printers; (19) is manually cut 6.35 mm foam board.

kgcm. The two motors on each side are driven in parallel by high power motor controllers (13). The robot is powered by a LiPo battery (2) which slides in underneath the base plate. The battery power is passed through a DC-DC step down regulator (3) before reaching the electronics. This enables the rover to operate continuously for over an hour while inflated and moving continuously at walking pace. Realistically, the rover will last much longer in experiments where it will often be interacting with users while stationary. It should be noted that the chassis has amble room for a second battery should the operational time need to be doubled.

Computation and Remote Control. All components on the robot are driven from a Raspberry Pi (RPi) 4 model B single board computer (16), featuring a 64-bit quad-core processor, dual-band 2.4-5.0 GHz wireless LAN, USB 2.0 and 3.0, and lots of digital I/O. Although the user may program the robot to operate autonomously, we focused on enabling remote control as is common in Wizard-of-Oz human-robot experiments or when more compute-intensive processing needs to be done off board. Upon initialization, the rover transmits its network parameters (IP address and port) to the Control Server which maintains a static IP address. This allows the user's control platform to retrieve these parameters and establish a direct TCP connection to the rover, even while operating with a wireless network that dynamically assigns a IP addresses as is common at many academic institutions (Fig. 2.B). In the event that the IP address of the rover changes during operation, the new address is automaticallly sent to the Control Server. This also permits simultaneous operation of multiple rovers with dynamic IP addresses within the same wireless network. The Control Server also acts as a platform for the user to upload custom configuration files, and any data files needed by the rover during operation. As detailed in Sec. IV, we set up the rover to be controlled by Python script running on a laptop, or e.g. ROS for higher level control.

Sensors for navigation. To help the robot navigate it has a RealSense RGB-D (9) camera mounted on the front and a wide-angle camera (5) mounted on the back (Fig. 5). By pointing the RealSense upwards at a slight angle, we expect to be able to detect the presence and gaze direction of nearby potential audience. The user can stream 640x480 pixel 3D/RGB data from the RealSense at 16.5 fps for off board processing or request individual images if continuous streaming is not needed. We have also included space for future versions to incorporate an optional Coral hardware Accelerator to do onboard image processing via deep learning. We further added three TOF sensors with a range up to 2 m in the front of the robot in case rapid obstacle avoidance is needed, which is likely in scenarios with large crowds, and an optional 3-axis accelerometer and gyroscope (14) to help the robot detect when it is driving on steep inclines or if it has fallen over.

Fig. 4: The Dagu Wild Thumber gives the rover mobility over relatively rough terrain as well as conventional HRI lab spaces such as carpet and tile.

III. INFLATABLE INTERFACE

The interface consists of a soft, inflatable tubular bladder with an internal camera-projector pair; beyond the audible feedback from the rover, such camera-projector systems have been shown to effectively gain the attention of a human audience [10]. We chose this technology because the field of soft robotics has shown that soft components have many advantages over rigid components in robots which engage in human interaction [6]. The inherent compliance of soft robots ensure safety during interaction; fluid-driven interfaces such as this one furthermore permit visual and morphological changes that can be adapted to the user in question; finally, soft interfaces, like soft robots, may be cheaper and simpler to manufacture than their rigid counterparts. The current inflatable interface with driver circuits costs $\sim$120 USD. We should note that human-scale inflatables have been deployed previously [11], however the need for them to act on an agile mobile base and as an adaptive interface is new.

Our interface consists of a tubular fabric bladder made of light weight ripstop Nylon (48 g/m^2). The interface can be inflated to full size using a quiet fan (21) similar to those used with Halloween inflatables. Full inflation takes $\sim$40 s, deflation takes $\sim$ 50s (Fig. 6). To fold the inflatable interface in a stable manner, we added three internal winches (24-25) spaced 120 degrees apart. Each of these use a small, strongly geared motor and a 3D printed drums with two inlets (diameter 8 and 14 mm) to drive two cables tied to the middle and the top of the inflatable interface respectively. To contract the interface we simply run the fan at 50% duty cycle to keep the cables taught, and then drive in the winch. Each motor has magnetic encoders to help ensure that they run at equal speeds and prevent them from over or under-extruding. The double cables cause the lower half of the interface to be folded almost entirely before the top half. This means that we can operate the interface at partial heights to accommodate the height of the user. When the interface is fully contracted the robot measures just 0.3 m. In this state it can easily navigate under obstacles such as chairs and tables (Fig. 2.B), and it is easy to handle for anyone who wishes to pick it up.

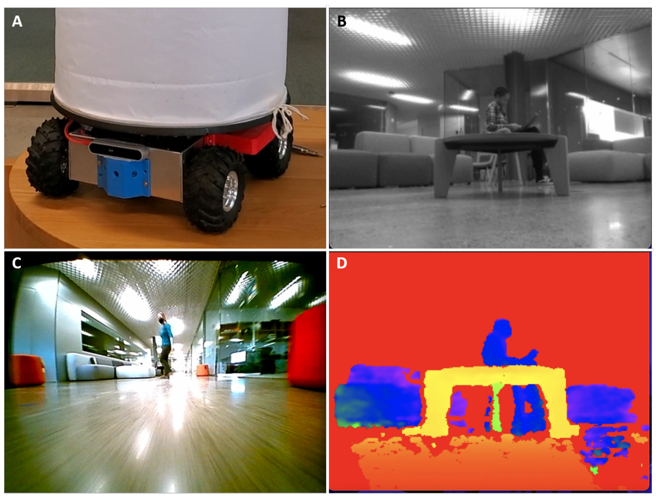

Fig. 5: Example output from cameras on the Marta robot. A) View of the front RealSense camera. B) View from the RealSense camera, here shown in monochrome. C) View from rear RGB camera. D) View from front RealSense depth data.

When the inflatable interface is fully extended it measures 1m creating a clear presence to gain audience attention and permitting comfortable interaction for an average height adult (Fig. 2.A).

The camera-projector pair is mounted inside the inflatable interface facing the front of the robot. The projector itself is countersunk into the Aluminum chassis to protect it in case the robot topples over. We chose the P300 Neo Pico projector (18) for its low weight, built in battery (lasts up to 2.5 hrs), built-in speaker, and high light intensity (420 lumens). Direct projection onto the Nylon interface produces a crisp image which is visible even in high light conditions. We automatically correct for distortion of the image onto the rounded surface in software, such that the user can simply upload the image they wish to display without further processing. We also experimented with the addition of a fish-eye lens such that the projector light covers the entirety of the inflatable interface. Additionally, to receive information from an audience interacting with the inflatable interface, we mounted a RPi camera (23) next to the projector. Following previous work with a past version of our robot [12], we demonstrate that images and videos from this camera can be used to classify shadow-based gestures and direct canvas touch. We also mounted a microphone to pick up on loud audible cues as could be relevant in disaster scenarios.

The inflatable interface and driver circuitry is mounted on a foam core base plate (19). The base plate itself is mounted to the rover via snap-in magnets for easy servicing during the design phase, and securely mounted via screws during demonstrations. The winch and fan drivers are connected to the RPi using easily detachable headers. We used a layer of flexible foam to keep the holes around the cables and the fan inlet relatively airtight. The rim of the base plate consists of a triple layer of foam board, onto which the inflatable interface

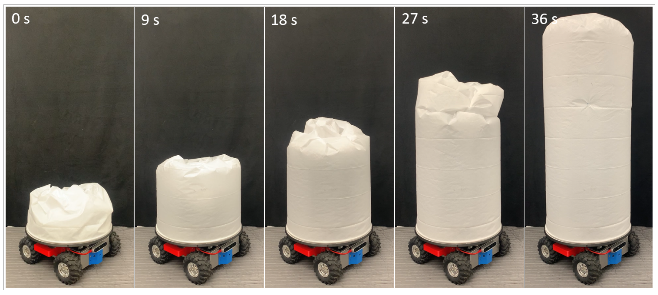

Fig. 6: Example sequence of the robot inflating its interface.

is secured via a simple slip-knot (Fig. 2.C). We further 3D printed a rim from Carbon filament (20) to protect the softer foam board in case the robot tips over.

IV. SOFTWARE ARCHITECTURE

The rover implements a highly modular multi-threaded software architecture (Fig. 7) written in Python3 that allows researchers and developers to program custom operational capabilities. The goal of this architecture is to be robust and efficient while minimizing the effort needed to develop new functionality. This section summarizes significant software components and capabilities of this architecture.

A Thread Manager spawns and monitors all computing threads on the rover. It ensures that a bug in one thread does not cause parallel threads to crash. The TCP Manager receives and enqueues all incoming Command Messages in the Command Priority Queue. All commands are encoded at the application layer using JavaScript Object Notation (JSON) and contain information about the sender and command priority level. In cases where incoming Command Messages exceed the computational ability of the RPi, the Command Priority Queue ensures that commands are executed in user specified priority order. Command Messages of equal priority are executed in the order they are received. This, in conjunction with TCP's out-of-order delivery prevention, makes for a robust messages transmission and execution protocol.

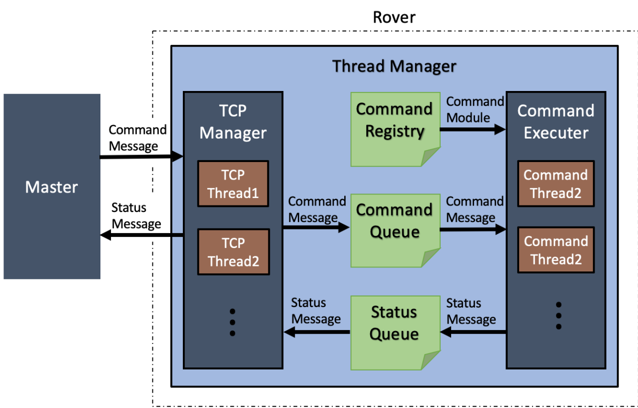

Fig. 7: A system diagram of the software architecture. The architecture is designed to be highly modular and adaptable, making it simple for users to operate the rover as is or to add new functionality by registering their own custom Command Modules.

After the TCP Manager places incoming Command Messages into the Command Priority Queue, The Command Executor processes each command by looking up the command's associated Command Module (callback function) in the Command Registry and executing it in a new Command Thread. Command Threads are executed in parallel, leveraging all four cores of the RPi processor. Basic controls, such as driving commands, projector images, camera feeds, etc., have pre-programmed Command Message and Command Modules (Table I). User-specific functionality, such as new sensors, maneuvers, interfaces, etc., can be easily implemented by creating a new Command Module and registering it in the Command Registry. Upon execution, each Command Module returns a Status Message which the Command Executor adds to the Status Queue to be processed by the TCP Manager. The TCP Manager sends these Status Messages back to the user control platform over the same socket as incoming messages. These Status Messages help to maintain robustness by informing the control platform if there is a problem during the execution of a Command Module. Lastly, the rover asserts a software interface to ensure that user defined Command Messages, Command Modules, and Status Messages are properly implemented. The remainder of this section highlights three of the key software functionalities of the rover.

Communication and Data Streams. All Command Message and Status Messages are transmitted via TCP at the transport layer, ensuring in order message delivery and data re-transmission in the event of a failed or corrupted transmission, due to TCP's data packet transmission acknowledgement. Given the imperative nature of these messages, the network throughput overhead associated with TCP is justified. Likewise, if the user requests a single image or reading from any on board sensor, data integrity is essential, and therefore TCP transmission is used. At the application layer, these messages and data are encoded in JSON or binary. However, for network communication where periodic data corruption or transmission failure is acceptable, such as high frequency video/sensor data streams, we recourse to other communication protocols capable of higher data throughput. The Intel Realsense camera streams 3D and RGB images to the client via User Data Protocol (UPD) at the transport layer and encoded binary at the application layer. While UDP lacks data packet transmission acknowledgement, it excels in transmission speeds. This is acceptable because a dropped frame is not detrimental to system performance when streaming at 30 frames per second. Similarly, the time of flight sensors can stream in encoded binary via UDP. Given that the internal and external rear-facing cameras generate traditional RGB images, the rover uses Rapid Spanning Tree Protocol (SRTP) at the application layer and UPD at the transmission layer. RSTP offers the capability of being decoded by any popular off-the-shelf media player, such as VLC, offering a simple and easy way for users to view camera data.

TABLE I: Thread and Command Module Overview

Command Module	**Description**
Text-to-Speech	Rover dictates audible messages through speakers
External Camera	Returns single images or starts/stops an RSTP video stream of rear facing camera
Internal Camera	Returns single images or starts/stops an RSTP video stream of camera inside the inflatable interface
Image	Upload, download, and list image database for projector and specify what image to project on inflatable interface
Set speed	Adjusts speed of left and right wheel motors
ToF sensors	Returns single Time of Flight reading or starts/stops a ToF data stream over UDP
Read IMU	Returns single inertial measurement unit reading or starts/stops a IMU data stream over UDP
Inflate/deflate	Controls inflatable interface
Check for command	Always running; monitors and maintains Command Queue
TCP Thread	**Description**
Listen	Receive and enqueues incoming command messages
Check new IP	Monitors IP of rover and updates Pi Server
Check status queue	Monitors and maintains status queue

Text-to-speech. Traditional alarm tones, such as beeping, can be unclear and misinterpreted as there is no universally accepted interpretation for non-verbal alert sounds. This problem can exacerbated by the high stress conditions of evacuation and emergency scenarios. Further more, it has been shown that verbal material presented auditorily is better remembered than the same material presented visually [13]. We therefore implemented an offline TTS (text-to-speech) engine, called pyttsx3, to convey verbal information to the people in its vicinity.

Gesture Recognition. We use the projector to display information and visual buttons on the top surface of the inflatable interface. The internal camera and a contact shadow detection algorithm is used to identify when and where a user is touching this interface. The rover uses OpenCV [14], an open source Python-based image processing library, to perform a number of image transformations including thresholds, Gaussian blur, erosion, and dilation. It then identifies groups of pixels that are potential contact shadows. Lastly, these groups of pixels are fit into contours that are identified by an OpenCV implementation of an algorithm proposed in [15]. These contours are then paired down using empirically determined criteria to identify the location of a contact shadow on the canvas. The processing frame rate can be configured by the researcher to tailor computational resource consumption. While this algorithm only leverages deterministic image processing techniques, it could be made more robust in future work by implementing convolution neural networks (CNN) for contact shadow detection, such as [12]. For example, the use of a CNN could prove to increase robustness in cases of extreme lighting conditions, or help to better identify features of the human hand; filtering out contact shadows cast by other objects. In future work, we hope to implement further gesture recognition capabilities, such as swiping and scrolling using the dynamic position of the user's hand.

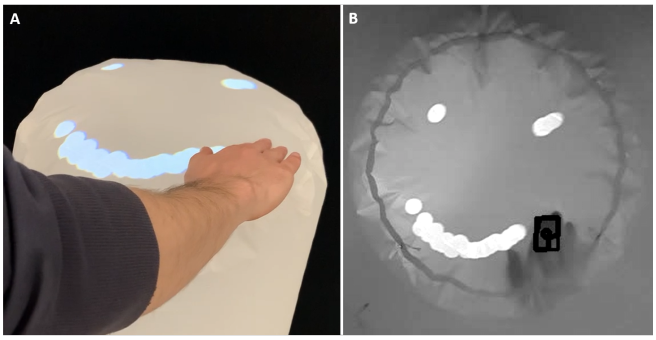

Fig. 8: The camera-projector pair inside the inflatable interface can be used as a versatile adaptable interface. Here a simple drawing program is created by recognizing touch through pixel value thresholds.

V. CONCLUSION AND FUTURE WORK

In this paper, we argued for a set of requirements that applies to HRI robot platforms that can act in crowded, unstructured environments, and introduced a new open source platform, Martha, that complies with these requirements and is designed specifically to serve academic studies. The robot is capable of driving at jogging pace, perform on-axis turns, and has a low center of mass to accommodate locomotion on inclined surfaces. It has a small form factor and can easily maneuver under low clearance objects, such as coffee tables. The camera-projector pair coupled with the soft inflatable interface permits a range of interaction studies related to visual, audible, and haptic cues. The robot is inexpensive, compact, and light weight compared to its counterparts. It requires little know-how to implement and operate, and it is fast to replicate with mostly off-the-shelf components ($<$ 1 week). Both software and hardware is built in a modular fashion to ease adaptation to new academic studies.

In the future, we aim to explore three avenues: 1) Incorporation of control modules into ROS, to enable users to leverage the wealth of libraries avaialble, including SLAM for navigation. 2) We hope to explore the opportunities afforded by the soft interface further, including more interesting morphological changes, onboard image analysis to recognize user feedback such as tactile interactions and gestures. 3) Finally, we will explore added mechanical abilities, such as the ability of the robot to survive impacts and falls, self-righting if accidentally turned over, and whether multiple robots may collective lift debris or transport loads safely on the inflatable interface.

VI. Acknowledgements

We would like to thank the following participants for their efforts to design and implement the first version of this rover as part of their M.Eng. degrees: Paolo Arguelles, Labhansh Atriwal, Emilie Baker, Connor Hahn, Yunshan Liu, Ta Wei Yeh, and Caitlin Stanton.

References

[1] C. Breazeal, K. Dautenhahn, and T. Kanda, "Social robotics," in *Springer handbook of robotics*. Springer, 2016, pp. 1935–1972.

[2] D. Kragic, J. Gustafson, H. Karaoguz, P. Jensfelt, and R. Krug, "Interactive, Collaborative Robots: Challenges and Opportunities." in *IJCAI*, 2018, pp. 18–25.

[3] P. Tsarouchi, S. Makris, and G. Chryssolouris, "Human–robot interaction review and challenges on task planning and programming," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 8, pp. 916–931, 2016.

[4] J. Schulte, C. Rosenberg, and S. Thrun, "Spontaneous, short-term interaction with mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1. IEEE, 1999, pp. 658–663.

[5] T. Shibata and R. Irie, "Artificial emotional creature for human-robot interaction-a new direction for intelligent systems," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 1997, p. 47.

[6] P. Polygerinos, N. Correll, S. A. Morin, B. Mosadegh, C. D. Onal, K. Petersen, M. Cianchetti, M. T. Tolley, and R. F. Shepherd, "Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction," *Advanced Engineering Materials*, pp. 1 700 016—-n/a. [Online]. Available: http://dx.doi.org/10.1002/adem.201700016

[7] C. Bartneck and J. Forlizzi, "A design-centred framework for social human-robot interaction," in *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No. 04TH8759)*. IEEE, 2004, pp. 591–594.

[8] N. Rossini and A. Esposito, "Modeling Human Behaviour in Emergency: A Research Agenda for the Creation of a Rescue Robot." in *WIRN*, 2011, pp. 235–246.

[9] F. Garzotto, M. Gelsomini, and Y. Kinoe, "Puffy: a mobile inflatable interactive companion for children with neurodevelopmental disorder," in *IFIP Conference on Human-Computer Interaction*. Springer, 2017, pp. 467–492.

[10] S. Borkowski, S. Sabry, and J. L. Crowley, "Projector-camera pair: an universal io device for human machine interaction," in *Polish National Robotics Conference KKR VIII*, 2004.

[11] H. Sareen, U. Umapathi, P. Shin, Y. Kakehi, J. Ou, H. Ishii, and P. Maes, "Printflatables: printing human-scale, functional and dynamic inflatable objects," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 3669–3680.

[12] Y. Hu, S. M. Bejarano, and G. Hoffman, "Shadowsense: Detecting human touch in a social robot using shadow image classification," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–24, 2020.

[13] C. G. Penney, "Modality effects in short-term verbal memory." *Psychological bulletin*, vol. 82, no. 1, p. 68, 1975.

[14] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[15] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.