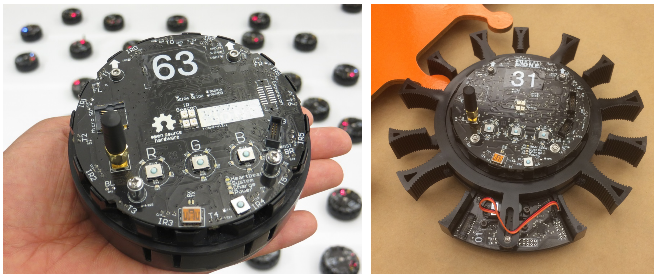
A Robot System Design for Low-Cost Multi-Robot Manipulation

James McLurkin[1], Adam McMullen[2], Nick Robbins[2], Golnaz Habibi[1], Aaron Becker[1], Alvin Chou[2], Hao Li[2], Meagan John[1], Nnena Okeke[3], Joshua Rykowski[4], Sunny Kim[1], William Xie[3], Taylor Vaughn[3], Yu Zhou[1], Jennifer Shen[1], Nelson Chen[1], Quillan Kaseman[1], Lindsay Langford[3], Jeremy Hunt[3], Amanda Boone[2], Kevin Koch[2]

***Abstract*—Multi-robot manipulation allows for scalable environmental interaction, which is critical for multi-robot systems to have an impact on our world. A successful manipulation model requires cost-effective robots, robust hardware, and proper system feedback and control. This paper details key sensing and manipulator capabilities of the r-one robot. The r-one robot is an advanced, open source, low-cost platform for multi-robot manipulation and sensing that meets all of these requirements. The parts cost is around $250 per robot. The r-one has a rich sensor suite, including a flexible IR communication/localization/obstacle detection system, high-precision quadrature encoders, gyroscope, accelerometer, integrated bump sensor, and light sensors. Two years of working with these robots inspired the development of an external manipulator that gives the robots the ability to interact with their environment. This paper presents an overview of the r-one, the r-one manipulator, and basic manipulation experiments to illustrate the efficacy our design. The advanced design, low cost, and small size can support university research with large populations of robots and multi-robot curriculum in computer science, electrical engineering, and mechanical engineering. We conclude with remarks on the future implementation of the manipulators and expected work to follow.**

I. Introduction and Related Work

Multi-robot systems have great potential for many practical applications. Exploration, mapping, search and rescue, surveillance, manipulation, and construction are all applications where multi-robot systems can have a large impact. In particular, *large populations* (100—10,000) can produce breakthrough solutions for multi-robot manipulation, with the ability to transport large numbers of objects of varying size. In this paper, we describe our approach to sensing and actuation for multi-robot manipulation. We identify three key components: a localization sensor to determine the pose of neighboring robots, a tactile sensor to sense impacts and exogenous forces, and a manipulator (gripper) to exert force on the object or other robots. These components need to be available at a low-cost, so that researchers can evaluate multi-robot manipulation with large populations of robots.



(a) The r-one robot **(b)** Manipulator attachment

Fig. 1: **(a)** The r-one robot is an advanced, low-cost, open-source platform designed for research, education, and outreach. **(b)** The r-one robot with a gripper designed for multi-robot manipulation. See experiments in video attachment http://youtu.be/FHlSOYLSe2E.

Figure 1 shows our multi-robot platform and manipulator. The platform is based on the r-one robot (Fig. 1a) [1]. The design satisfies three requirements: **1. Advanced Design:** The basic sensor suite is comprehensive, complete, and useful for many multi-robot manipulation tasks. In particular, specific multi-robot features—inter-robot communication, neighbor localization, and collision detection—are built-in. **2. Low Cost:** Low cost is required for research to scale to large populations. **3. Usability:** Efficient development requires that basic operations, like programming, charging, and data collection, be automated with a hands-free, centralized user interface.

The r-one has a parts cost of around $250 per robot, and requires around twenty minutes to assemble. The base robot has many sensors; including an accelerometer, gyro, wheel encoders, light sensors, bump sensors, and an infrared inter-robot communication and localization system. The centralized user interface features a radio, an infrared beacon for ground-truth localization, and an autonomous docking and charging system. The robot can be programmed in C/C++ or run an embedded Python interpreter to make programming more accessible to younger, less experienced students [1]. In addition, a custom hands-free bootloader simplifies updating and reprogramming large numbers of robots. The design is mature: we have had excellent success over the past three years in undergraduate, graduate, and high school courses, as well as outreach activities and demos. Figure 1b shows a r-one with a gripper. Each gripper costs approximately $70. The design allows robots to grab objects and other robots from any direction. The design lends itself to algorithmic simplicity because the entire assembly is free to rotate around the r-one's chassis.

[1]Computer Science Department, Rice University, Houston, TX {jmclurkin, maj5, sck3, Jennifer.Shen, Yu.Zhou, nc13, qtk1, ab55}@rice.edu

[2]Mechanical Engineering Department, Rice University, Houston, TX {ajm6, njr3, ahc1, hl11, arb9, kjk2}@rice.edu

[3]Electrical Engineering Department, Rice University, Houston, TX {nvo1, lkl6, jrh6}@rice.edu

[4]Computer Science Department, United States Military Academy, West Point, NY Joshua.Rykowski@usma.edu

Robot	Source	Wheel Encoders	Radio	Neighbor Pose Angles	Neighbor Pose Range	Robot ID Beacon	Visible Light sensor	IR Obstacle Detection	Ultrasonic Range	Accelerometer	Gyro	Bump Sensor	Cliff Detector	Temperature Sensor	Camera	Microphone	Remote Programming	Gang/Self Charging	Notes	Retail Price ($)	Parts Cost ($)
Khepera III	K-Team	•	•					•	•				•							2000	-
Create	iRobot	•										•	•							220	-
Scribbler	Parallax		•				•	•												198	-
Finch	Finch						•	•		•				•						99	-
robomote	USC	•	•					•				•							compass sensor	-	150
3pi	Pololu																		IR line sensor(x5)	99	-
CostBots	Berkeley		•							•									NEST sensor boards	-	200
Mindstorms	LEGO	•	•				•		•			•								249	-
kilobot	Harvard				•		•										•	•		-	14
e-puck	EPFL	•	•					•		•					•	•				979	-
e-puck + IR	EPFL	•	•	•	•			•		•					•	•				1388	-
r-one	Rice	•	•	•	•	•	•	•		•	•	•		•			•1	•	remote programming in development	-	250

TABLE I: A comparison of available low-cost robots suitable for multi-robot research.

A. *Existing Platforms*

There are many existing robot platforms designed for research and education, shown in Table I. Educational and hobby platforms such as the Pololu 3pi [2], Scribbler [3], robomote [4], and costbots [5] are inexpensive, but they lack basic sensors, such as wheel encoders. LEGO Mindstorms [6] is the leader in educational robotics, but as the system places a heavy emphasis on *building* the robot, there is often little time left for *programming* the robot. The iRobot Create [2] is a popular platform for medium-sized robots, but the size, cost, and limited sensor suite require a large test area and many add-on components for multi-robot work. Most of these platforms utilize either C, which is too difficult for younger students, or graphical programming languages, like the LEGO Bricks used on the NXT, which limit the software complexity. None of the available platforms have hardware for multi-robot communication and localization.

The Khepera robot [7] and EPFL e-puck [8] are research platforms, have a larger sensor suite, and are a practical size for indoor experiments. However, their cost makes fielding large populations difficult. The e-puck requires an optional inter-robot communication turret to localize neighbors. None of the existing platforms are uniquely identifiable from a global localization system.

The r-one's main contribution to the multi-robotics community is an integrated, low-cost platform with a rich sensor suite, including inter-robot communication and localization, scalable user interface for efficient operation with large populations, and support for C/C++ for research and Python for education. While the processing power is adequate for many applications, it is too limited for others, notably vision and SLAM. An expansion connector enables access to computer or cell phone processing power if needed.

II. Hardware Design

The main components of the robot are two printed circuit boards (PCBs) and three plastic parts that compose the chassis and the two-piece bump skirt. The additional hardware are components available from Digikey and Pololu Robotics. The exploded CAD diagram is shown in Fig. 2a. The robot is 11 cm in diameter, 4.2 cm tall, and weighs 308 grams.

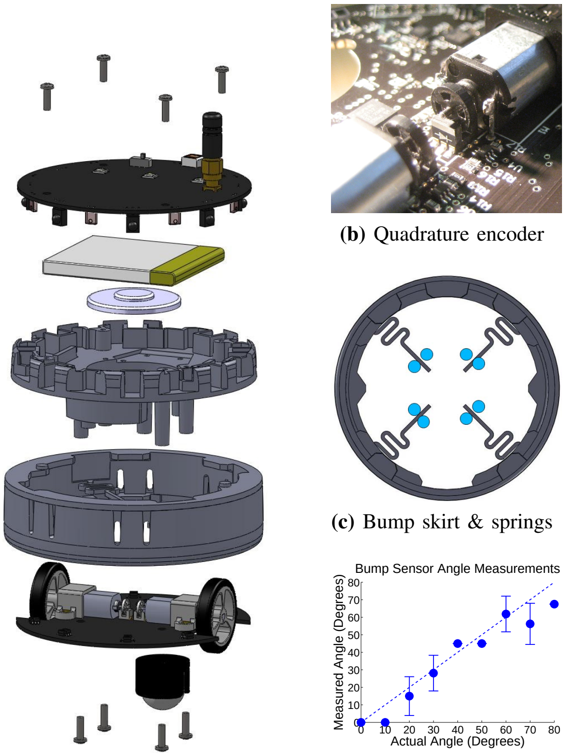
The total parts cost $247. The electrical/mechanical components are $130, PCB fabrication is $40, and PCB assembly is $60. The final mechanical assembly is simple, and takes about 20 minutes per robot. Total system current ranges from 140 mA to 650 mA, with 510 mA typical. With a 3.7 V 2000 mAh lithium-polymer battery, the robot runs for four hours. The battery can be charged from the USB port or from a docking connecter.

The sensor suite consists of a 3-axis gyro, 3-axis accelerometer, four visible-light photoresistors, and two quadrature encoders. Communication is handled by eight IR transmitters, eight IR receivers, a 2.4 GHz radio with 2 Mbps data rate, and a USB port. To interact with the user, the robot has a VLSI1053 audio chip for MIDI playback, three push buttons, and three arrays of five LEDs—each in red, green, and blue.

The robot is controlled by a Texas Instruments Stellaris LM3S8962 microcontroller. The CPU core is an ARM Cortex-M3 running at 50 MHz with 256 KB of flash memory and 64 KB of SRAM. The 2.4 GHz radio on the robot is used for inter-robot communication and for the scalable UI. Inter-robot localization is provided by the IR communication system described in Section II-C.

A. *Chassis and Encoders*

The exploded diagram in Figure 2a shows the main robot components. The chassis is composed of three injection-molded plastic parts. We used acetal homopolymer with Teflon (Delrin 500 AF), which is fairly inexpensive, strong, and very low friction. The motors and encoders mount directly to the bottom circuit board, as shown in Figure 2b. These low-cost quadrature encoders are a novel design, and they use an optical interruption sensor to detect gaps in an encoder wheel attached to the rear motor shaft. The



(a) Exploded view

(b) Quadrature encoder

(c) Bump skirt & springs

(d) Bump sensor accuracy

Fig. 2: **(a)** Exploded CAD view of the robot assembly. The robot is composed of two circuit boards and a plastic chassis. **(b)** The motors and encoders mount directly to the circuit board. Each quadrature encoder costs $0.75 and has a resolution of 0.0625 mm/tick. **(c)** Top view of the bump skirt. The light blue circles are posts from the chassis. The flexure springs are molded into the skirt and have a serpentine path to reduce their effective spring constant. **(d)** Bump sensor experiment data. We tested angles from 0° to 90° over 12 trials. The sensor has an average error of 8.1°.

encoder wheels are made from plastic on a laser cutter, and manufacturing tolerances limit the design to four slots, producing a 0.0625 mm/tick linear resolution at the wheel. The 32 mm wheels coupled with a 100:1 gearbox give the robots a maximum speed of 300 mm/sec, and the system is controllable down to 2 mm/sec.

B. Bump Sensor

The bump sensor encircles the entire perimeter of the robot. The sensor parts are shown in Figure 2a: the bottom circuit board and the plastic bump skirt directly above it. The bump skirt uses four integrated flexure springs shown in Figure 2c to center it when it is not contacting an obstacle. Deflections are detected with eight optoreflective sensors mounted facing up on the bottom circuit board. The multiple sensors provide the direction of the impact with an average error of 8.1°(Fig. 2d). We have used the sensor for obstacle avoidance, angle of impact estimation, subsequent "reflection" back into the environment, and, currently, for multi-robot manipulation.

C. Inter-Robot Communication and Localization

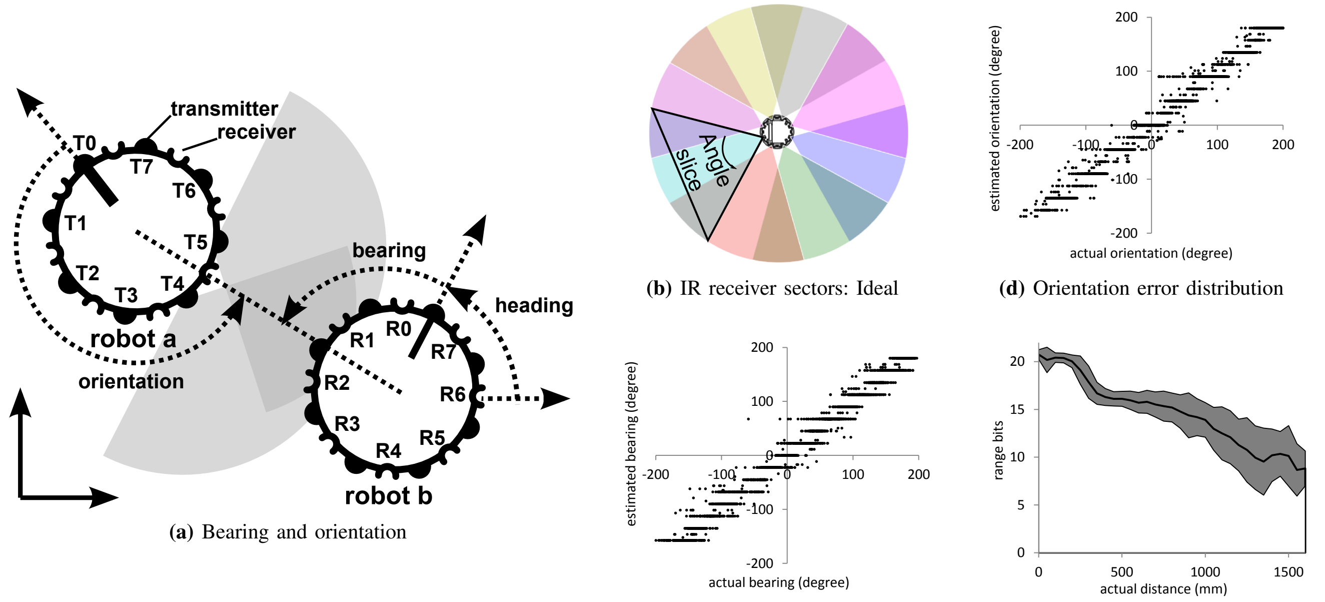
Multi-robot manipulation requires the robots to measure the pose of neighbors in order to control their physical configuration. There are many approaches to measuring the relative position of a neighbor robot, including vision-based systems [9], [10], and infrared light [11]–[14]. Because of our size and cost goals, the r-one robot uses an infrared (IR) communication system to localize their neighbors.

The IR communication system is composed of eight IR transmitters and eight IR receivers. These sixteen devices alternate around the perimeter of the robot, spaced 22.5° apart, starting with the first transmitter facing 0°, shown in Figure 3a. The chassis is designed so that transmitter emission patterns cross at the half-power point, which produce a fairly uniform radial emission pattern when the eight transmitters transmit in unison. Calculations from the emitter's data sheet predict a power variation of 4%, which is consistent with our qualitative observations. The receivers are standard Sharp IR remote control devices, with 38khz modulation and a maximum bit rate of 1250bps. Because the communication bandwidth is limited, the maximum communication range is short in order to limit the number of neighbors, and therefore messages, that are received by each robot.

The user creates "neighbor data" structures to store public data and share the data with neighboring robots, and the operating system combines this data and broadcasts one unified packet. The first datum in a message is always the transmitting robot's ID. We use the CCIT-16 polynomial for error detection and additional heuristics to filter out neighbors with weak communication links. Since IR bandwidth is limited, longer messages are broadcast via radio. The two separate transmissions are merged into a unified message by the receiving robot, using the IR message to determine if the radio message is from a neighboring robot. The software API abstracts away from the marshalling, transmission, reception, and decoding of the data.

Each robot transmits its neighbor message at periodic intervals, but with a random offset, similar to the ALOHA protocol [15]. This simple TDMA scheme limits our effective channel capacity to 50% of the available bandwidth but does not require a centralized scheduler, making it a practical design. The bandwidth limitations place restrictions on robot density, but increasing communication bandwidth would require signifigantly more expensive receivers, undermining our low-cost goal.

The eight IR receivers allow a robot to measure the bearing of a neighboring robot when a message is received. Refer to Figure 3a for definitions of bearing and orientation. The chassis is designed to limit the detection sector of each receiver to 68°, and these sectors overlap to form 16 distinct regions that each cover ≈22.5°, as shown in Figure 3b. By noting which receiver(s) get a message from a neighbor, the robot can calculate the bearing of the transmitting robot with a resolution of ≈22.5°. The error distribution for bearing calculation is shown in Figure 3c.



(a) Bearing and orientation

(b) IR receiver sectors: Ideal

(d) Orientation error distribution

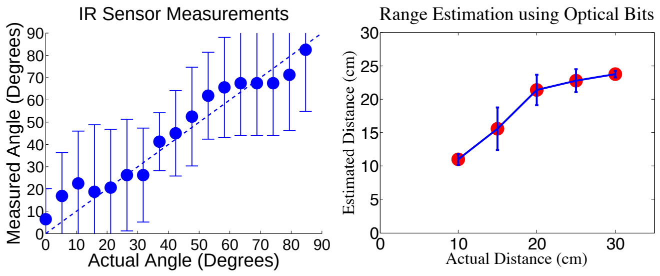
(c) Bearing error distribution

(e) IR comms. range

Fig. 3: **(a)** We define *bearing* and *orientation* as relative measurements between neighboring robots and *heading* as the direction a robot faces from an external reference frame. In this example, robot **a** is transmitting a message to robot **b**. The grey sectors illustrate the emission patterns from T4 and T5 of robot **a**, which are received on R1 and R2 on robot **b**. This allows robot **b** to calculate the relative bearing and orientation of the transmitting robot. See text for details. **(b)** Top view of each IR receiver's detection region. Each receiver detects signals in a 68° arc, which overlap to form 16 distinct sectors. A message from a neighboring robot will be received on one or two receivers and can be processed to determine the bearing. **(c)** Bearing measurement error distribution. **(d)** Orientation measurement error distribution. **(e)** Range between robots is measured by transmitting start bits at variable power. The lowest-power bit that is received is used to estimate the distance to the neighboring robot. The resulting estimate has a linear region from 200-600 mm, with some sensitivity out to 1200 mm. The maximum range has a sharp cutoff, falling from nearly 100% reception at 1600 mm to 0% at 1700 mm.

The eight IR transmitters can be enabled individually, which allows a receiving robot to measure the orientation of a transmitting robot. A set of "orientation bits" is sent from each transmitter sequentially, starting from transmitter 0. In the example shown in Figure 3a, the receiving robot will get the start bit and then nothing as the orientation bit is transmitted from transmitters 0-4. It then receives the orientation bits from transmitters 4 and 5 and then nothing again as the orientation bit is transmitted from transmitters 6 and 7. When complete, the receiver robot accepts orientation bits 4 and 5, indicating that these transmitters were facing the receiver when the message was sent. The receiving robot can now compute the orientation of the transmitting robot. The sector overlaps and error distribution of the orientation measurements are almost identical to that of bearing, but with a $\frac{\pi}{16}$ offset, as seen in Figure 3d. Range between robots is measured by ending each message with a set of "range bits" that are transmitted at variable power. By measuring which range bits are received, the robot can estimate range from 0 mm to 1600 mm with a varying degree of accuracy, as shown in Figure 3e.

Nearby obstacles will reflect a robot's own IR messages back to itself. With eight transmitters and receivers, this can produce a good sense of nearby obstacles. The bearing to the obstacle is measured the same way bearing to neighbors is measured and produces similar accuracy shown in Figure 4a.



(a) IR obstacle bearing accuracy.

(b) IR obstacle distance accuracy.

Fig. 4: **(a)** IR obstacle angular accuracy measured from a pine board, a common surface that reflects IR light well. The angle to the obstacle can be measured by noting the receivers that receive the reflected signal. The system has an average error of 19.4°. **(b)** IR communications system can detect obstacles by processing reflected messages. Range to an obstacle can be estimated by counting the number of bits in the reflected signal. This experiment used the same pine board.

The rone estimates range by counting the number of signal bits the receivers detect (Fig. 4b). Fewer optical signal bits are detected as distance increases. Since this is a reflected signal, the detection range will vary with surface albedo, and thin objects (*e.g.* chair legs) will not be detected at all.

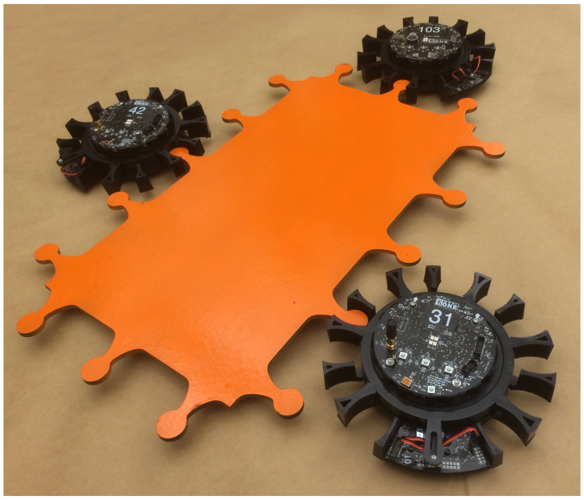


Fig. 5: Multiple robots with engaged manipulators can translate and rotate an object.

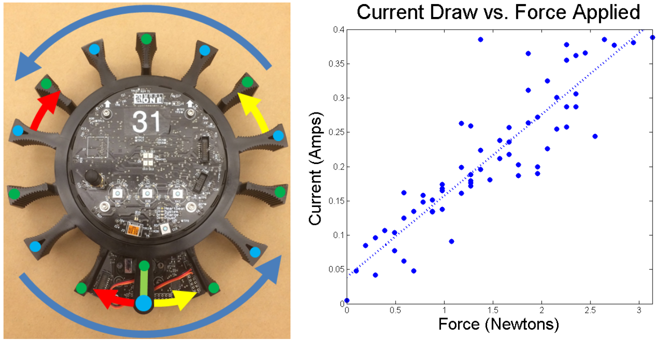
D. Bootloader

Our hands-free bootloader updates and reprograms all of the robots simultaneously via radio. The user selects a robot with the desired program and starts the update cycle. This robot becomes the “host robot”. All robots compute a 32-bit CRC (cyclic redundancy check) for each 4-KB segment of program in flash memory. The host broadcasts its CRC values and queries each robot individually in ID order. The queried robot compares the host’s CRCs to its own to find mismatched segments, and requests updated segments. As the host replies to the requests, robots not actively queried that require the same segments eavesdrop on the broadcast and update their program concurrently. The received packets are written directly to flash, minimizing the delay between segments. The protocol has been tested and optimized to provide an efficient way to distribute new software to a large number of r-ones. The current implementation of the bootloader allows filtering robots by ID range, subnetwork, and hardware revisions. The bootloader is easily extendable for additional features in the future.

III. Manipulation

To increase the applicability of the r-one robots to swarm-based object manipulation, they must interact with objects of various size and shape. We have designed a manipulator (gripper) attachment that is easily installed onto the robots for this purpose. Our manipulation model is mechanically and algorithmically simple, scalable for large numbers of robots, and cost-effective. Basic manipulation experiments ensure that the gripper design is practical. Going forward, this manipulator will extend the r-ones capabilities and allow for greater exploration into multi-robot manipulation techniques.

A. The Gripper



(a) **(b)**

Fig. 6: **(a)** The blue arrows indicate the free-rotation of the design. The red and yellow arrows show clockwise and counterclockwise motion of the top gripper (blue dots) via the motion of the servo; the bottom gripper (green dots) is “fixed” relative to the top gripper, and the light green line is the servo horn. **(b)** Current draw from the servo vs. tangential force applied by the gripper paddles

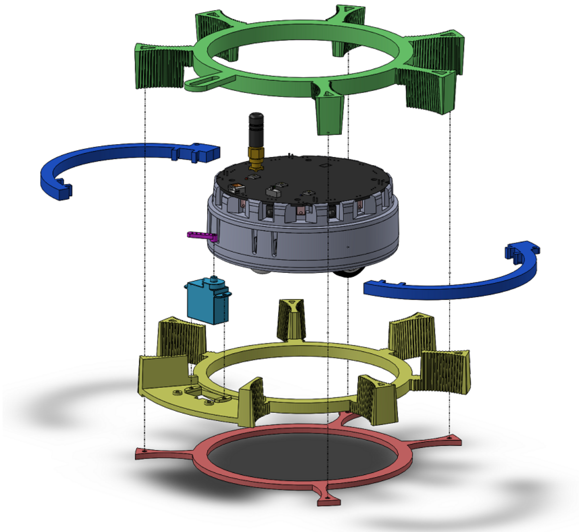
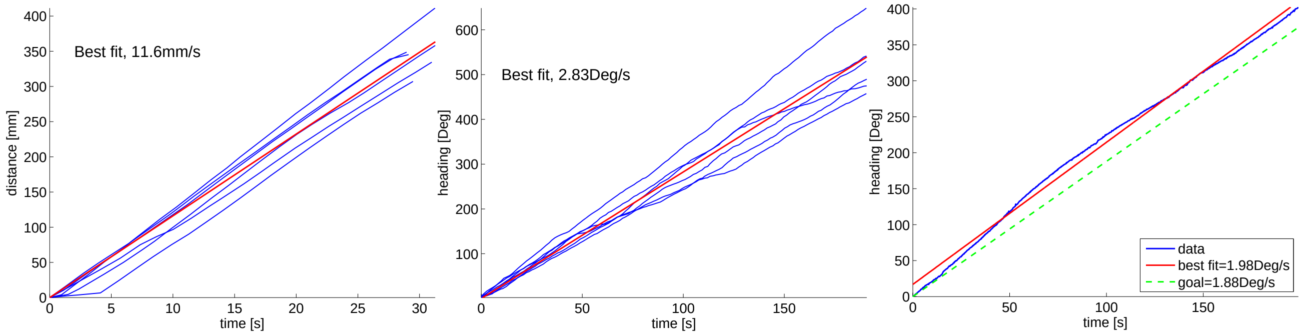


Fig. 7: A detailed exploded view of the gripper assembly.

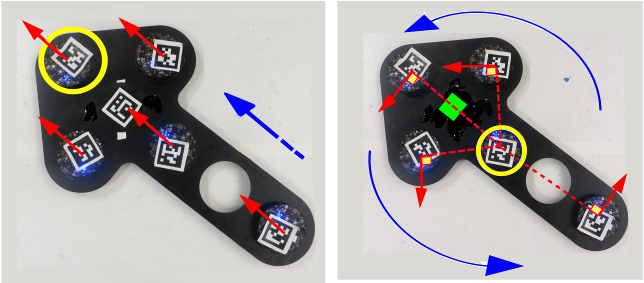
The r-one gripper is designed for multi-robot manipulation. It can attach to vertical features on objects of various size and shape independent of the robot’s orientation. The manipulator has five main components: a top and bottom gripper, chassis clips, a bottom stop, and an S-75 sub-micro servo as seen in Fig. 7. The outer diameter of the assembly is 19 cm, and the manipulator can grab object features up to 5 cm wide. The top and bottom grippers are injection molded from Teflon-infused acetal homopolymer (Delrin 500 AF), and the chassis clips and bottom stop are laser cut from ABS. Assembly requires only seven screws and about eight minutes. The estimated cost is $70 per robot for a set of sixty; $30 for plastic parts and $40 for electronic components (board and servo).

The gripper is mounted to the bump skirt and uses the bump sensors to measure the direction of applied radial forces. In addition, the gripper can sense both the paddles’ position and applied tangential force (“gripping” force) via servo feedback (Fig. 6b). The maximum gripping force is 4 N. Each robot can pull an object behind them with a peak force of 24 N before damaging the servo, which is well above the frictional force available from the tires.



(a) Absolute translation velocities **(b)** Angular velocities about a pivot **(c)** Angular velocities about an arbitrary point

Fig. 8: Collective transportation experiments.



(a) Translation **(b)** Pivoting around central robot

Fig. 9: Multi-robot manipulation for object transport. These images are from an early experiment.

The manipulator has two degrees of freedom: the entire assembly is free to rotate around the r-one's chassis, and the paddles can move relative to one another (Fig. 6a). This is an omni-directional planar gripper; the robots can approach objects from nearly any direction, and the paddles can grab an object from either side. Each chassis clip has a small rare-earth magnet embedded at one end. Hall-effect sensors mounted on the board sense the proximity of these magnets and act as limit stops; that is, when the magnets are directly above the sensors, the robot will initiate a counter-rotation to "unwind" itself, thereby preventing over-rotation. This allows the wires connecting the top and gripper boards together to remain relatively in place and unwound.

Free rotation about the chassis (blue arrows in Fig. 6a) allows modeling an engaged robot as a pin joint, *i.e.* an engaged object cannot apply a torque to the robot. Removing the torque simplifies the mathematical model of object rotation and translation. A single robot cannot effectively rotate an object without prior knowledge of the object's center of friction, but multiple robots can rotate an object precisely by exerting forces to create a net torque around *any* point. To translate a large object, multiple robots can attach at different locations around the object and subsequently align their headings (Fig. 5).

B. Transportation

Our gripper-enhanced robots will soon manipulate objects in unison. Currently, we use objects that restrain the robots in order to test their ability to translate and rotate objects (Fig. 9). The robots spin freely within these objects and behave as a pin-joint; this is similar to the gripper design, in which the robots spin freely inside the gripper. These objects can be translated, rotated, or any combination of the two. In this paper, we show basic experiments for translation and rotation. Future work aims to design algorithms to allow simultaneous translation and rotation to quickly move an object through obstacle-filled terrain. Our results thus far have shown the effectiveness of performing translation and rotation separately.

a) Translation: We assume that the robots are pre-attached to the object and that each robot can rotate freely relative to the object. To translate an object along a desired path, one robot is externally elected as a leader and holds a constant heading and speed. All of the other carrier robots follow that leader's heading and speed by reaching a consensus through distributed algorithms (Fig 9a) [16]. The robots can rotate freely to the proper heading without rotating the object and therefore translate in any direction. Figure 8a shows the nearly identical velocity profiles of each robot.

b) Rotation about a Pivot Robot: In this method, a robot is selected as a stationary pivot while the other robots rotate about that pivot (Fig 9b). Each robot attempts to move perpendicularly to the moment arm between the robot and the pivot. Ideally, the pivot does not move and the other robots rotate the object about the pivot by varying their velocity based on the calculated distance to the pivot robot. Robot performance for rotation about a pivot can be seen in Fig. 8b. Constant angular velocities are harder to achieve due to pivot robot jitter. Ultimately, there will be no pivot robot, and robots will calculate the center of mass of the object and rotate about that point; this algorithm is planned for future work.

c) Rotation about the Center of Mass: We assign a point on the object as the center of mass without involving a pivot robot. This kind of rotation is split into two parts: (1) estimation of the center of mass of the object and (2) rotation around an arbitrary point. Once the robots perform both of these functions independently, they will be integrated to rotate about the center of mass. We have only tested rotation about an arbitrary point (Fig. 8c). Center of mass estimation algorithms are currently under development, and we will combine the two in the near future.

Going forward, we will develop algorithms for simultaneous rotation and translation. This will require more complex control schemes and will be done using cycloidal motion paths on the robots. Second, we will integrate the gripper attachment, as it will enable the robots to attach to objects in desired locations and manipulate freely.

IV. Conclusion

Our robot combines an advanced sensor suite that allows multi-robot system behavior with a focus on manipulation, easily allowing rapid development of interesting systems. We hope the r-one robot and manipulator will grow to be a popular platform for multi-robot manipulation research. Our design is cost-effective, simple, and practical, allowing for many students in high school, undergraduate, and graduate settings to become involved with multi-robot manipulation.

References

[1] J. McLurkin, A. Lynch, S. Rixner, T. Barr, A. Chou, K. Foster, and S. Bilstein, "A low-cost multi-robot system for research, teaching, and outreach," *Proc. of the Tenth Int. Symp. on Distributed Autonomous Robotic Systems DARS-10, November*, pp. 597–609, 2010.

[2] R. Weiss and I. Overcast, "Finding your bot-mate: criteria for evaluating robot kits for use in undergraduate computer science education," *J. Comput. Sci. Coll.*, vol. 24, pp. 43–49, Dec. 2009.

[3] T. Balch, J. Summet, D. Blank, D. Kumar, M. Guzdial, K. O'Hara, D. Walker, M. Sweat, C. Gupta, S. Tansley, J. Jackson, M. Gupta, M. Muhammad, S. Prashad, N. Eilbert, and A. Gavin, "Designing personal robots for education: Hardware, software, and curriculum," in *Pervasive Computing, IEEE*, vol. 7, pp. 5–9, Apr. 2008.

[4] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale sensor networks," in *IEEE Int. Conf. Rob. Aut.*, 2002.

[5] S. Bergbreiter and K. Pister, "Costsbots: An off-the-shelf platform for distributed robotics," in *IEEE Int. Rob. and Sys.*, pp. 1632–1637, 2003.

[6] D. Baum and R. Zurcher, *Definitive Guide to Lego Mindstorms*, vol. 2. Apress, 2003.

[7] F. Mondada, E. Franzi, and A. Guignard, "The development of khepera," in *Proceedings of the 1st international Khepera workshop*, vol. 64, pp. 7–13, 1999.

[8] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, pp. 59–65, 2009.

[9] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *International Journal of Robotics Research*, vol. 25, pp. 431–447, May 2006.

[10] A. Howard, L. E. Parker, and G. S. Sukhatme, "The SDR experience: Experiments with a Large-Scale heterogenous mobile robot team," in *9th International Symposium on Experimental Robotics 2004*, (Singapore), June 2004.

[11] J. McLurkin, *Stupid Robot Tricks: A Behavior-Based Distributed Algorithm Library for Programming Swarms of Robots*. S.M. thesis, Massachusetts Institute of Technology, 2004.

[12] J. Pugh and A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 188–193, 2006.

[13] A. Gutierrez, A. Campo, M. Dorigo, J. Donate, F. Monasterio-Huelin, and L. Magdalena, "Open e-puck range and bearing miniaturized board for local communication in swarm robotics," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, (Kobe, Japan), pp. 1745–1750, IEEE Press, 2009.

[14] D. Payton, R. Estkowski, and M. Howard, "Compound behaviors in pheromone robotics," *Robotics and Autonomous Systems*, vol. 44, no. 3-4, pp. 229–240, 2003.

[15] N. Abramson, "The aloha system: Another alternative for computer communications," Technical Report B70-1, University of Hawaii, Honolulu, Hawaii, Apr. 1970.

[16] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked Multi-Agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.