Pheeno, A Versatile Swarm Robotic Research and Education Platform

Sean Wilson[1], Ruben Gameros[1], Michael Sheely[2], Matthew Lin[2], Kathryn Dover[2], Robert Gevorkyan[3], Matt Haberland[3], Andrea Bertozzi[3], and Spring Berman[1]

Abstract—Swarms of low-cost autonomous robots can potentially be used to collectively perform tasks over very large domains and time scales. Novel robots for swarm applications are currently being developed as a result of recent advances in sensing, actuation, processing, power, and manufacturing. These platforms can be used by researchers to conduct experiments with robot collectives and by educators to include robotic hardware in their curricula. However, existing low-cost robots are specialized and can lack desired sensing, navigation, control, and manipulation capabilities. This paper presents a new mobile robot platform, Pheeno, that is affordable, versatile, and suitable for multi-robot research, education, and outreach activities. Users can modify Pheeno for their applications by designing custom modules that attach to its core module. We describe the design of the Pheeno core and a three degree-of-freedom gripper module, which enables unprecedented manipulation capabilities for a robot of Pheeno's size and cost. We experimentally demonstrate Pheeno's ability to fuse measurements from its onboard odometry for global position estimation and use its camera for object identification in real time. We also show that groups of two and three Pheenos can act on commands from a central controller and consistently transport a payload in a desired direction.

Index Terms—Cooperative Manipulators; Distributed Robot Systems; Education Robotics

I. INTRODUCTION

CONTROL strategies for large-scale robot collectives, called *robotic swarms*, have recently begun to be implemented on physical testbeds [1]–[3]. However, many swarm applications continue to be simulated rather than tested on hardware due to the high cost of existing robot platforms or their inability to perform the required tasks. Hence, there is a need for an affordable yet capable robot platform that can be readily customized by users for desired swarm applications.
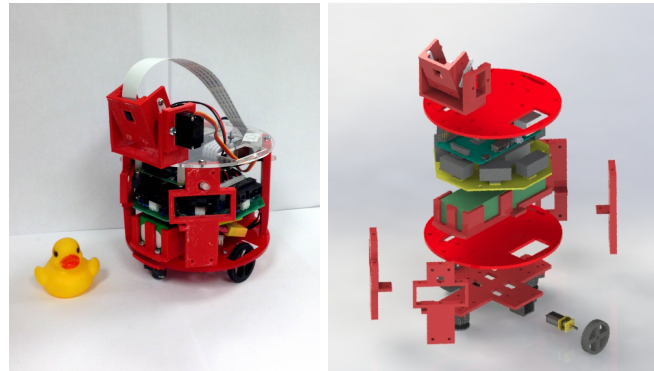
Fig. 1: *Left:* The Pheeno core module with the ICRA 2016 *duckie*, a standard size reference approximately 5 cm wide [4], for scale. *Right:* Exploded SolidWorks rendering of Pheeno's core module.
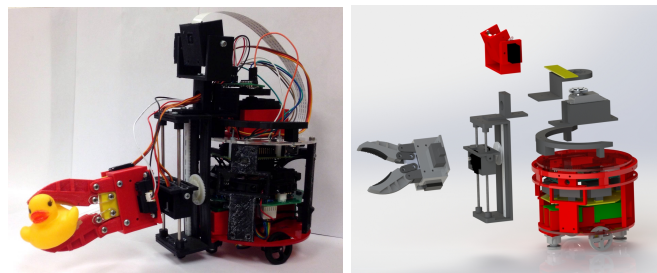

Fig. 2: *Left:* The Pheeno robot platform with gripper module holding an ICRA 2016 *duckie* [4] for scale. *Right:* Exploded SolidWorks rendering of the Pheeno gripper module (gray) and core module (red).

Robotic swarms have many potential uses, ranging from environmental monitoring, mapping, and surveillance to tasks that require manipulation as well as sensing, including construction, manufacturing, and disaster response. Since swarm robotics is a fairly young field, new algorithms and control strategies for swarms are constantly being developed with different requirements on the robots' sensing, actuation, control, and communication abilities. Existing platforms that are suitable for swarm robotic research have typically been developed for specific tasks and are not easy to modify.

Inexpensive, reliable robots are also valuable tools outside of a robotics research setting. Biologists can use simple robots to help elucidate the mechanisms that underlie collective

animal behaviors [5]–[7]. Robots can also be used in education and outreach activities to spark students' interest in STEM fields and demonstrate physical applications of mathematics and physics. Robots that are constructed from components with intuitive interfaces and large user communities are highly beneficial in these contexts.

To address these applications, we developed a new mobile robot platform *Pheeno*, shown in Fig. 1, with several design criteria in mind. Pheeno is small and affordable, sophisticated enough for multi-robot research experiments, accessible to students and others who are new to robotics, and modular to suit the requirements of different robotic tasks. We developed an inexpensive gripper module, shown in Fig. 2, that allows Pheeno to grasp, rotate, and lift objects in its environment for individual or cooperative manipulation tasks. We first briefly described the design of the Pheeno core and gripper module in [15]. In this paper, we compare Pheeno to existing platforms for multi-robot research and education (Section II), elaborate on the design of the core and gripper modules (Section III), and demonstrate Pheeno's sensing, navigation, processing, communication, and cooperative manipulation capabilities with a series of experiments (Section IV). We conclude with a discussion of future work, including approaches to full decentralization of the collective transport strategy presented here, which uses a control computer for path planning, localization, and task assignment (Section V).

II. EXISTING PLATFORMS

In recent years, various robot platforms have been developed for multi-robot research and education. Table I compares several existing platforms that have capabilities similar to Pheeno's.

The r-one robot [9] is an open source platform designed for multi-robot experiments and education. It has a large sensor array for communication and localization. A gripper attachment [8] allows the robot to drag payloads along the ground, although it does not enable 3D manipulation. The platform is not readily expandable to users who lack significant experience with electronics.

The WolfBot [10] is an open source platform designed for distributed sensing and education. It has a sensor suite for communication, localization, and on-board image processing. The platform incorporates the BeagleBone Black computer, which allows the robot to be modular but has less community support than Pheeno's processors, the Raspberry Pi 2 and the Arduino Pro Mini. The cost of parts for the WolfBot is $550, which is twice the cost of the Pheeno base.

The Khepera IV is an expandable, commercially available research and educational platform. Additional modules have been developed for the robot, including a two degree-of-freedom revolute, revolute (RR) serial linked manipulator capable of lifting 50 g. The robot and gripper attachment retail for approximately $6,000, making them expensive to use in educational curricula and multi-robot research.

The marXbot [11] is a highly capable, modular, open source platform designed for multi-robot experiments. It has a large array of sensors for communication, localization, and onboard image processing. It also has a three-pronged attachment mechanism that can connect to a *hand-bot* [2] for manipulation tasks. Its large sensor suite makes it an expensive platform for multi-robot applications.

The e-puck [12] is a commercially available robot designed for education at the university level. The platform is equipped with sensors for odometry and communication, and its capabilities can be increased with various extensions. However, the robot is fairly expensive for multi-robot applications, retailing for about $1,000.

The Thymio II [13] is an open source platform designed for users with little or no previous experience in robotics. It has a variety of sensors but lacks basic odometry sensors like wheel encoders, a gyroscope, or a magnetometer, although its capabilities can be expanded with accessories such as LEGO components.

The Scribbler 2 [14] is a commercially available, open source educational robot that is suitable for users with a range of programming experience. Additional devices, sensors, and servos can be attached to the robot through a hacker port, and an add-on board can be plugged in to provide a camera and wireless communication with a computer.

There are other robot platforms that can be used for multi-robot research and education, although they are too dissimilar to Pheeno to be included in the comparison table. The Kilobot [16] and GRITSbot [17] are very small, affordable robots that have been developed for swarm robotic experiments. The Pololu 3pi [18] is a small hobby platform with very limited sensing. The LEGO Mindstorms [19] and iRobot Create 2 [18] are designed for education and can be given augmented capabilities through accessories and expansion packs. The LEGO Mindstorms kit has a specific set of sensors and chassis components, allowing up to four sensors and four motors to be driven at once. This inherently limits its sensing and actuation abilities. The iRobot Create 2 is suitable for users with more experience in robotics. Its main chassis contains bump sensors, drop sensors, and encoders, which enable the robot to perceive its environment through collisions but do not allow it to predict and avoid encounters.

III. DESIGN

Our objective was to create a low-cost robot that can be customized by users to conduct a variety of multi-robot research experiments. Toward this end, we designed Pheeno to be a modular platform that is constructed from commercially available components, including low-cost processors, an array of basic sensors, and 3D printing plastic filament. The robot is composed of a *core module*, described in Section III-A, that users can interface to their own custom-designed modules for desired applications. Section III-B describes one such module, a three degree-of-freedom *gripper module* that enables the robot to manipulate and transport objects either individually or in cooperation with other Pheeno platforms. The list of robot components, schematics for PCB boards, CAD designs for the core and gripper modules, and guides to assembly, calibration, and programming of the platform are publicly available in an online repository [20].

Robot	Processor	Sensing	Communication	Manipulator	Cost
Pheeno	ATmega328P and ARM Cortex-A7	3D accelerometer, 3D magnetometer, wheel encoders, IR, camera	Serial, WiFi, Bluetooth	RPR serial linkage	$270 for base (core), $80 for gripper[†]
r-one [8], [9]	ARM Cortex-M3	3D accelerometer, 2D gyroscope, wheel encoders, IR, bump, ambient light	Radio, IR	Omni-directional gripper	$220 for base, $70 for gripper[†]
WolfBot [10]	ARM Cortex-A8	3D accelerometer, 3D magnetometer, IR, camera, microphone, ambient light	WiFi, Zigbee	None	$550[†]
Khepera IV[*1]	ARM Cortex-A8	3D accelerometer, 3D gyroscope, wheel encoders, IR, ultrasonic, camera, ambient light	WiFi, Bluetooth	RR serial linkage	$3,180 for base, $2,900 for gripper[*1]
marXbot [11]	ARM11	3D accelerometer, 3D gyroscope, IR, omni-directional camera, front camera, RFID reader, 2D force	WiFi, Bluetooth	Three-fingered attachment device	N/A
e-puck [12]	dsPIC 30F6014A	3D accelerometer, IR, camera, microphones, range and bearing turret[⊕], three-camera turret[⊕], omni-directional camera[⊕]	Zigbee	None	$1,000 for base, battery, and charger[*2]
Thymio II [13]	PIC24FJ128GB106	3D accelerometer, IR, microphone, temperature, touch	IR	None	$199 for base[*3]
Scribbler 2 [14]	P8X32A-Q44	wheel encoders, microphone, IR, ambient light, camera[⊕]	Serial, Bluetooth[⊕]	None	$150 for base[*4], $100 for IPRE Fluke2 Board[*5]

[†] Cost of parts.   [*] Retail price.   [⊕] Additional extensions not included with base robot.

[1] Available for purchase at http://www.k-team.com/mobile-robotics-products/khepera-iv

[2] Available for purchase at http://www.gctronic.com/shop.php

[3] Available for purchase at http://www.techykids.com

[4] Available for purchase at https://www.parallax.com/product/28136

[5] Available for purchase at http://www.betterbots.com

TABLE I: Comparison of currently available multi-robot platforms.

A. Core Module

Figure 1 shows the fully assembled core module of Pheeno with an exploded SolidWorks rendering. The cylindrical core has a diameter of 12.7 cm and a height of 11.1 cm. Most of the components are 3D printed using standard ABS plastic, which allows for easy replication and modification of the core. The only components that are not 3D printed are the motor mounts as well as the circular base and cap of the housing. These are standard robotic chassis parts sold by Pololu Robotics & Electronics [21], used to reduce the printing time of the robot. Currently, a Pheeno core takes about 5 hours to print with a MakerBot Replicator 2X 3D printer (MakerBot Industries) using an infill of 12%.

The core module is a differential drive platform that is actuated by two standard micro-metal gear motors with extended back shafts and supported by two caster wheels to maintain the balance of the robot. Motors with a 51.45:1 gear ratio are currently used, but standard micro-metal motors with higher or lower gear ratios can be substituted, allowing Pheeno to be quicker or more powerful to suit the user's needs. With the current gear ratio and 32-mm-diameter wheels, Pheeno can move at controllable speeds between 4 cm/s and 42 cm/s. The extended back shafts enable the attachment of magnetic quadrature wheel encoders, which have a linear resolution of 0.163 mm/tick. The robot can measure its acceleration and heading using an STMicroelectronics LSM303D e-compass

containing a 3D accelerometer and a 3D magnetometer.

The main processors onboard Pheeno are the Raspberry Pi 2 Model B microprocessor [22] and the Arduino Pro Mini microcontroller (3.3V model) [23]. These boards were chosen for their accessibility to new users and the large user communities supporting them. The Raspberry Pi is a credit-card sized Linux computer that enables users to program the robot in a range of languages. The Arduino Pro Mini is a small (1.8 cm × 3.3 cm) member of the widely used Arduino microcontroller family. These two boards interact through serial communication. Pheeno uses the Raspberry Pi for high-level control and image processing and the Arduino Pro Mini for control of low-level actuation and sensor data processing for accurate navigation. All of the GPIO pins on the Arduino Pro Mini are occupied by connections to the accelerometer, magnetometer, wheel encoders, H-bridge motor driver, and infrared (IR) sensors on the core module. The Raspberry Pi has 26 open GPIO pins that can be used for sensor inputs and actuator control, as well as 3 open USB ports that allow the use of various USB adaptors such as WiFi and Bluetooth for communication.

The core module is equipped with six IR proximity sensors for enabling collision avoidance, a Raspberry Pi camera for vision-based object detection, and four RGB LEDs for displaying the robot's state. Five of the IR sensors are evenly spaced along the front perimeter of the robot, and one sensor is
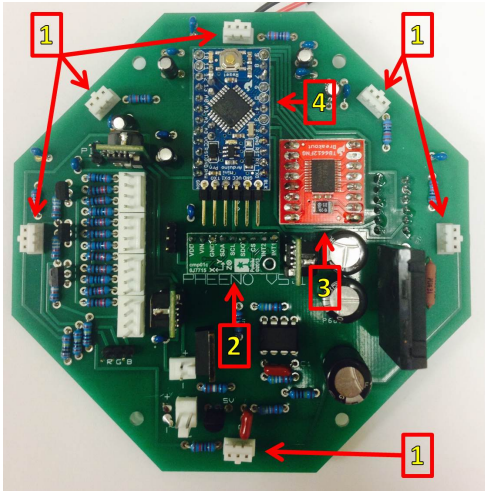
Fig. 3: Top-down view of the main Pheeno circuit board with the major components numbered. **1)** Infrared (IR) sensor mounts able to interface with any 3-pin JST connector. **2)** 3D accelerometer and magnetometer. **3)** Motor control board (H-Bridge). **4)** Arduino Pro Mini microprocessor.

Fig. 4: A SolidWorks rendering of the gripper module with degrees of freedom and components shown. **1)** Yaw servo, which enables $180°$ rotation about the central vertical axis of the Pheeno core module. **2)** Gear rack servo, which enables 6.2 cm of prismatic motion. **3)** Wrist servo, which enables $180°$ rotation about the radial axis of the Pheeno core. **4)** Underactuated gripper with potentiometer feedback. **5)** Core module camera. **6)** IR distance sensor.

placed on the back to detect nearby objects. Currently, Pheeno is equipped with six Sharp GP2Y0A41SK0F IR sensors with a range of $4-30$ cm. IR sensors with different ranges could be substituted if they interface with 3-pin JST PH connectors on our custom printed circuit board (PCB), shown in Fig. 3. The Raspberry Pi camera, a 5MP Omnivision 5647 sensor in a fixed focus module, is mounted on top of the core. A servomechanism tilts the pitch angle of the camera within a $180°$ range. The LEDs are evenly spaced around the perimeter of the core.

The robot is powered by a 11.1V 3000 mAh LiPo battery that is secured to the bottom of the chassis. The current draw of the robot is 120 mA when idle and 410 mA during typical use. When the motors are stalled and image processing is performed onboard, the current draw can spike to 670 mA. During demonstrations at outreach events, where visitors remotely controlled the robot through a graphical user interface on a laptop computer, the robot operated continuously for 5 hours while retrieving small objects and streaming 1080×720p video to the laptop.

B. Gripper Module

In order to give Pheeno manipulation capabilities, we designed a gripper module that consists of a standard 3-degree-of-freedom (DOF) revolute, prismatic, revolute (RPR) serial arm with an end-effector capable of grasping an object, shown in Fig. 4. The joints of the arm are driven by three standard servos, which provide the arm with the ability to lift an object up to 6.2 cm, roll it up to $180°$ about the core's radial axis, and rotate it up to $180°$ about the core's central vertical axis.

Most of the gripper components are 3D printed using standard ABS plastic, allowing for easy modification and replication. The rack gearing that provides the prismatic motion and two shafts that stabilize the lift are available commercially from [24], [25]. The gripper jaws are composed of a rigid
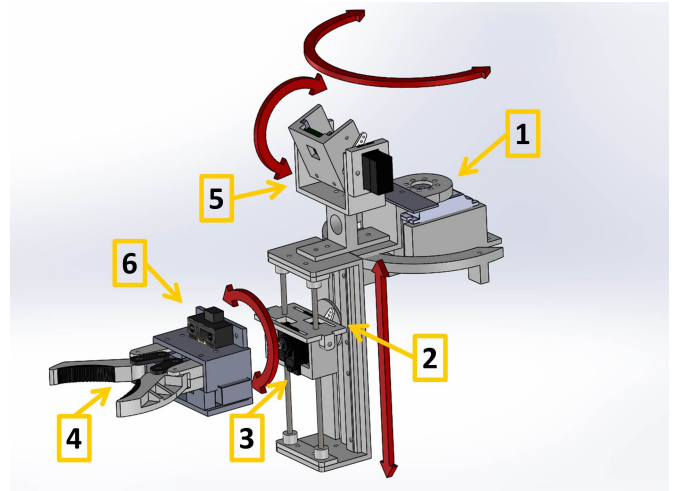
structure with molded urethane rubber pads that deform while grasping an object. The gripper is underactuated, driven by a single servo through a yoke mechanism. The Raspberry Pi camera from the core module is affixed to the top of the gripper module, allowing visual servoing of the gripper if desired. Additional sensors included in the gripper module are a front-mounted IR sensor and a potentiometer to give feedback on grasping.

Another PCB is included in the module to provide power to the servos and take in sensor inputs. An 8-channel 10-bit ADC allows these analog sensors to be connected to the Raspberry Pi, enabling closed-loop control of the gripper through image processing and sensor feedback. The PCB has been designed with two open channels for the ADC, which allows additional analog sensors to be interfaced with the gripper. When Pheeno is driving and moving its gripper during typical use, its current draw is 640 mA. In situations where the gripper servo and drive motors are stalled and the lift servo is strained while image processing is being performed onboard, the current draw jumps to 1.08 A. A large battery capacity was chosen to accommodate these high power demands.

The arm is capable of lifting and manipulating a weight of about 400 g. This allows Pheeno to manipulate light objects independently or cooperate in a team to transport heavier objects. When a robot manipulates an object by itself, its yaw servo can be actively controlled to rotate the object about the core module's central vertical axis independently of the rotation of the wheel base. During collective transport tasks, each robot can turn off its yaw servo, which allows its drive train to backdrive the yaw servo and rotate to a desired heading within a $180°$ range. This enables the robots

to simultaneously grasp a load while driving in a common direction at different angles relative to their manipulator arms. Once a robot detaches from the load, its yaw servo can become active again to return the gripper to its forward configuration.

IV. EXPERIMENTS

Three types of experiments were conducted to evaluate Pheeno's ability to localize using its onboard odometry, identify objects by color from its camera images, communicate through WiFi, and cooperatively manipulate an object as a team. We discuss the experimental results in this section.

A. Dead Reckoning

A path-following algorithm was implemented in two experiments in order to test Pheeno's ability to determine its (x, y) coordinates in a global reference frame using only onboard odometry. We define Pheeno's heading as θ, its angular speed as ω, and its translational speed as v. The robot's motion can be modeled as unicycle dynamics:

$$\dot{x} = v \cos \theta, \qquad \dot{y} = v \sin \theta, \qquad \dot{\theta} = \omega$$

A standard PID controller was used to drive the robot to a desired position (x_d, y_d) at a constant speed while tracking a desired heading,

$$\theta_d(t) = \arctan \frac{y_d - \hat{y}(t)}{x_d - \hat{x}(t)},$$

where $(\hat{x}(t), \hat{y}(t))$ is the robot's estimate of its position at time t. This state estimate was updated at a rate of 15 Hz. In the first experiment, Pheeno estimated its global position and orientation using only measurements from its encoders. In the second experiment, a set of complementary filters was applied to the platform's encoder, accelerometer, and magnetometer measurements in order to correct for errors due to wheel slipping, which can occur when the robot accelerates and makes fast turns. A general block diagram of a complementary filter is shown in Fig. 5. For this sensor fusion, the encoder measurements were high-pass filtered and the accelerometer and magnetometer readings were low-pass filtered using a first-order filter.

The target trajectory and actual robot trajectory in the first and second experiments are compared in Fig. 6a and Fig. 6b, respectively. In both experiments, drift in the robot's position from the target trajectory is unavoidable. In the second experiment, the inclusion of the accelerometer and magnetometer measurements produces a more accurate estimate of the robot's global position compared to the first experiment, in which only encoder measurements are used. However, the resulting improvement in tracking performance is relatively small; moreover, the localization approach in the first experiment is easier to implement and explain in an educational setting.

B. Image Processing

To evaluate Pheeno's onboard image processing capabilities, we measured the time required for the robot to (1) acquire image frames from its camera for processing, and (2) perform
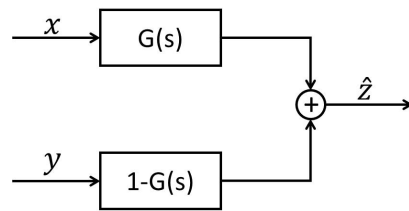

Fig. 5: A block diagram of a complementary filter. Here, x and y are measurements of a state z, and $\hat{z}$ is the estimate of this state by the filter.
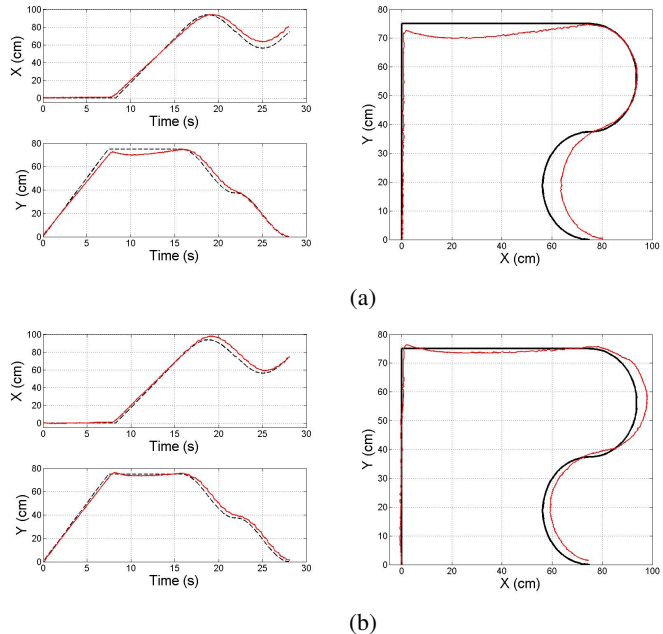

(a)

(b)

Fig. 6: Pheeno's actual trajectory (red line) as it tracks a predefined trajectory (black line) while localizing using (a) only encoder measurements, or (b) a fusion of encoder, accelerometer, and magnetometer measurements.

contouring to identify the centers of mass of shapes with different colors in the image. These tests were run for the image in Fig. 7 at three different resolutions and used image processing algorithms from the OpenCV library [26]. The images were transformed to the HSV color space, thresholded for red, yellow, and blue, and contoured using the standard transform, thresholding, and contouring functions from OpenCV 2.7 in Python 2.7.10. Figure 7 shows an output of the color tracking algorithm with the color blob centers of mass identified. Table II lists the minimum, average, and maximum times to acquire and process 200 frames at different image resolutions. The data shows the expected tradeoff between resolution and processing speed and demonstrates that the image processing routines are performed with reasonable sampling times at lower resolutions.

C. Collective Transport

We implemented Algorithm 1 on several Pheenos equipped with gripper modules in order to test the platform's ability to
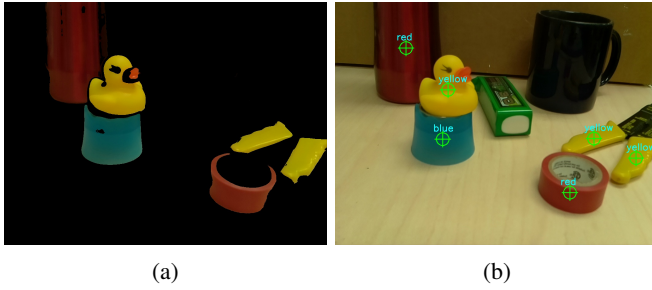
(a)   (b)

Fig. 7: (a) A masked image from Pheeno's camera after contouring for red, yellow, and blue. (b) Identification of each color blob's center of mass.

Resolution (pixels)	Image Acquisition (sec)	Contouring (sec)
320×240	[0.010, 0.023, 0.040]	[0.170, 0.172, 0.220]
640×480	[0.050, 0.063, 0.090]	[0.660, 0.672, 0.680]
1024×768	[0.130, 0.139, 0.180]	[1.760, 1.781, 1.800]

TABLE II: Time to acquire an image and perform image processing routines at different resolutions. The results are from 200 captured frames. The data take the form *[minimum, average, maximum]* time to process a frame in seconds.

perform cooperative manipulation tasks. The experiments were performed in the UCLA Applied Mathematics Laboratory. Three robots were placed in a 1.5 m × 2.1 m arena with a single circular payload of height 8 cm, diameter 20 cm, and weight 150 g. The robots and payload are marked with 2D binary identification tags to enable real-time tracking of their positions and orientations. The tags are tracked using two overhead Imaging Source DMK 21F04 1/4" Monochrome CCD cameras with a resolution of 640×480 pixels at a frame rate of 30 FPS. The robots and load are identified from their tags using the thresholding, boxpoint, and contouring OpenCV libraries on a Windows computer. A *control computer* serves as a pseudo-GPS, path planner, and communication hub for the robots. The control computer and the robots communicate with each other using WiFi. The overall control architecture of each robot is shown in Fig. 8.

1) Robot path planning: The control computer selects a *goal position* a set distance away from the load perimeter for each robot (see Fig. 9). The first is designated as the position of a robot that will pull the load backward in the desired direction of transport. The control computer assigns this position to the robot that is closest to it. The remaining goal positions are evenly spaced around the load according to the number of robots in the team. Starting from the first goal position, the control computer sweeps the image in the counterclockwise direction and assigns the next detected robot to the next goal position around the load.

The control computer plans each robot's path as a series of points from its initial position to the goal position. A waypoint algorithm is used to design robot paths that circumvent the

---

Algorithm 1: Transport strategy for a single robot

while *not at goal position around load* **do**
    move to next waypoint
end
while *entire team not at goal positions around load* **do**
    wait
end
if *gripper IR sensor measurement* > 8 *cm* **then**
    move forward
else
    rotate core to target transport direction
    rotate gripper back to load
    grasp load
end
while *entire team not grasping load* **do**
    wait
end
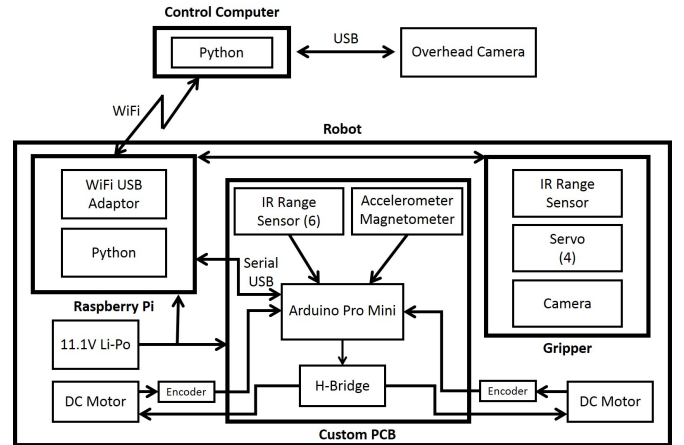lift load
carry load in target transport direction

Fig. 8: Robot control architecture for collective transport.

load, avoiding robot-load collisions. The control computer sends each robot its next waypoint and acts as a pseudo-GPS, updating the robot's global position and orientation at a rate of 3 Hz to correct for errors in the onboard state estimates described in Section IV-A. When a robot determines that it has reached its waypoint, it requests the next one from the control computer. If it has reached its final location around the load, it waits for all robots to communicate to the control computer that they have reached their goal locations.

2) Load grasping and transport: The control computer notifies the robots once they all have reached their assigned positions around the load. Next, the robots must orient themselves in the desired direction of transport and grasp the load, as shown in Fig. 10. In the *Approach* phase, each robot drives forward until the reading from the IR proximity sensor on its gripper drops below 8 cm, indicating that the load is within its gripper jaws. The robot then records its current orientation and enters the *Core Rotation* phase, during which its core module rotates so that its heading aligns with the direction of
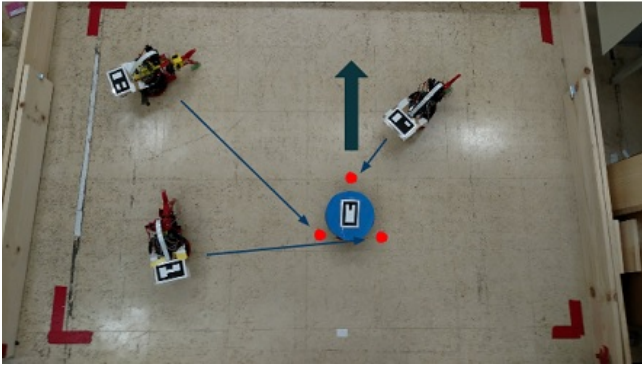
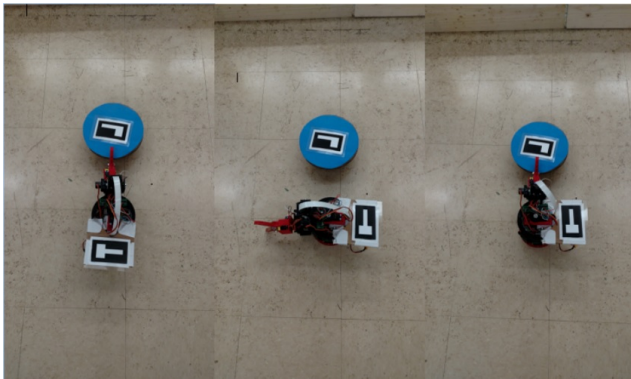Fig. 9: Goal position assignments for three Pheenos prior to transport in the direction of the large arrow.

Fig. 10: Pheeno reorienting itself and grasping the load in preparation for transport. From left to right: the *Approach* phase, *Core Rotation* phase, and *Gripper Rotation* phase.
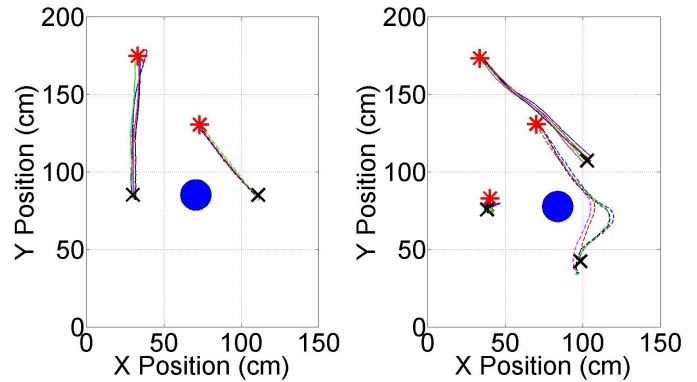

Fig. 11: The individual robot trajectories during the path planning phases of five transport trials with a team of (*left*) two robots and (*right*) three robots. Trajectories with the same line style correspond to the same robot, and trajectories with the same color correspond to the same transport trial. Robots begin at the red stars and move to the black ×'s, and the blue circle represents the load.

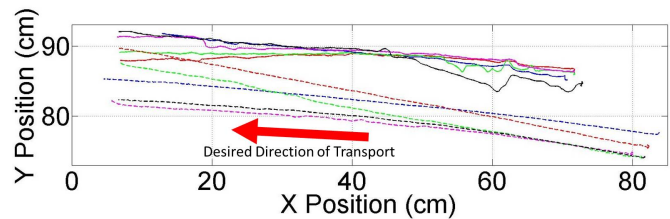Fig. 12: The load trajectory during five transport trials with a team of two (solid lines) and three (dashed lines) robots.

transport. If the robot ends up facing the desired direction, then it will drive forward during transport; if it is facing the opposite direction, then it will drive backward. The robot calculates the angular difference $\Delta\theta$ between its initial and final headings using its onboard magnetometer, and in the subsequent *Gripper Rotation* phase, its gripper rotates an angle $-\Delta\theta$ back toward the load.

A robot's onboard camera is used to correct any misalignment of its gripper with respect to the load due to noise in the magnetometer readings and error in the rotation of the core and gripper. The Raspberry Pi uses an algorithm similar to the one described in Section IV-B to determine the center of the load, which is colored blue. Two PI controllers regulate the yaw angle of the gripper and the pitch angle of the camera to align the center of the thresholded image with the center of the frame. Once this alignment is achieved and the load is gripped, the robot communicates to the control computer that it is ready to lift the load and waits for confirmation.

After all robots have communicated they are ready to lift, the control computer sends them a command to start the transport. At this point, the control computer no longer serves as a pseudo-GPS or communication hub. The robots simultaneously lift the load, turn off their yaw servos to allow passive yaw rotation of their grippers, and drive in the

direction of transport while maintaining their headings with a PI controller that acts on each robot's magnetometer readings. The robots continue transporting the load until they exit the arena defined by the overhead cameras' view.

3) Experimental results: Transport experiments were performed with teams of two and three robots, with the same transport task repeated five times for each team size. The initial positions and orientations of the load and robots were chosen using a random number generator. Fig. 11 plots the robot trajectories during the path planning phase of the experiments. The robots follow very similar trajectories during each trial, demonstrating that they can reliably communicate with a central computer hub to localize and receive commands. The slight discrepancies in robot paths across trials are likely due to errors by the control computer in reading the fiducial tags and variations in each robot's determination of whether or not it has reached a particular waypoint along its path. Fig. 12 plots the load trajectories during the transport phase of the experiments. For each team size, the load follows approximately the same trajectory and travels in the desired direction, indicating that the robots are able to consistently achieve stable transport of the load in a target direction without communication. Small variations in the initial load position

are due to human placement error, and discrepancies in the load trajectories are likely due to noise in the magnetometer readings of each robot.

V. CONCLUSION

Pheeno is a new mobile robot platform that is designed to be accessible to students for educational use, while still incorporating sensing and manipulation capabilities that are sophisticated enough for multi-robot research experiments. The robot's modular design allows users to develop custom attachments that suit their specific applications. This paper focuses on the design of the core robot module and a gripper module that enables Pheeno to manipulate objects. The robot's capabilities are demonstrated with proof-of-concept experiments on trajectory tracking, image processing, and collective transport.

In the future, we plan to use Pheeno for larger-scale swarm robotic experiments on mapping, coverage, manipulation, and construction. For these applications, we will develop fully decentralized control strategies that do not require a central computer. Toward this end, we will implement local wireless communication on the robots and the ability to autonomously recharge. In collective transport experiments, we will replace the centralized planning of robot trajectories with a boundary coverage scheme such as our stochastic strategy in [27] to achieve robot allocation around a load. The robots will use consensus algorithms to coordinate their grasping, lifting, and transport of the load, rather than relying on external commands to synchronize these actions. We are currently designing additional modules for the robot and developing a graphical user interface to remotely control Pheenos from a smartphone or laptop computer.

ACKNOWLEDGMENTS

REFERENCES

[1] G. Habibi, K. Zachary, W. Xie, M. Jellins, and J. McLurkin, "Distributed centroid estimation and motion controllers for collective transport by multi-robot systems," in *Proc. IEEE Int'l. Conf. on Robotics and Automation (ICRA)*, 2015.

[2] M. Dorigo *et al.*, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, no. 4, pp. 60–71, Dec 2013.

[3] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014. [Online]. Available: http://www.sciencemag.org/content/345/6198/795.abstract

[4] ICRA 2016 trailer theme: ducks in the loop & duck-robot interaction. Accessed: 2015-01-12. [Online]. Available: http://trailer.icra2016.org

[5] S. Garnier, "From ants to robots and back: How robotics can contribute to the study of collective animal behavior," in *Bio-Inspired Self-Organizing Robotic Systems*, ser. Studies in Computational Intelligence, Y. Meng and Y. Jin, Eds. Springer Berlin Heidelberg, 2011, vol. 355, pp. 105–120. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20760-0_5

[6] S. Garnier, M. Combe, C. Jost, and G. Theraulaz, "Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? A swarm robotics test bed," *PLoS Comput. Biol.*, vol. 9, no. 3, p. e1002903, Mar 2013. [Online]. Available: http://dx.doi.org/10.1371/journal.pcbi.1002903

[7] J. Krause, A. F. Winfield, and J.-L. Deneubourg, "Interactive robots in experimental biology," *Trends in Ecology & Evolution*, vol. 26, no. 7, pp. 369–375, 2011.

[8] J. McLurkin, A. McMullen, N. Robbins, G. Habibi, A. Becker, A. Chou, H. Li, M. John, N. Okeke, J. Rykowski, S. Kim, W. Xie, T. Vaughn, Y. Zhou, J. Shen, N. Chen, Q. Kaseman, L. Langford, J. Hunt, A. Boone, and K. Koch, "A robot system design for low-cost multi-robot manipulation," in *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 912–918.

[9] J. McLurkin, A. J. Lynch, S. Rixner, T. W. Barr, A. Chou, K. Foster, and S. Bilstein, "A low-cost multi-robot system for research, teaching, and outreach," in *Distributed Autonomous Robotic Systems (DARS)*. Springer, 2013, pp. 597–609.

[10] J. Betthauser, D. Benavides, J. Schornick, N. O'Hara, J. Patel, J. Cole, and E. Lobaton, "WolfBot: A distributed mobile sensing platform for research and education," in *Zone 1 Conf. of the American Society for Engineering Education (ASEE Zone 1)*. IEEE, 2014, pp. 1–8.

[11] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada, "The marXbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 4187–4193.

[12] 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 *Proc. 9th Conf. on Autonomous Robot Systems and Competitions*, vol. 1, no. 1. IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.

[13] F. Riedo, M. Chevalier, S. Magnenat, and F. Mondada, "Thymio II, a robot that grows wiser with children," in *IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, Nov 2013, pp. 187–193.

[14] Scribbler 2. Accessed: 2016-01-12. [Online]. Available: https://www.parallax.com/product/28136

[15] S. Wilson, R. Gameros, and S. Berman, "Pheeno, a versatile swarm robotic research and education platform," in *Proc. 1st Int'l. Symp. on Swarm Behavior and Bio-Inspired Robotics (SWARM)*, 2015, poster abstract.

[16] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. IEEE Int'l. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 3293–3298.

[17] D. Pickem, M. Lee, and M. Egerstedt, "The GRITSBot in its natural habitat - a multi-robot testbed," in *Proc. IEEE Int'l. Conf. on Robotics and Automation (ICRA)*, 2015.

[18] L. Major, T. Kyriacou, and O. P. Brereton, "Systematic literature review: Teaching novices programming using robots," *IET Software*, vol. 6, no. 6, pp. 502–513, 2012.

[19] D. Baum, *Definitive Guide to Lego Mindstorms*, 2nd ed. Apress, Nov. 2002.

[20] Pheeno Robot GitHub Repository. Accessed: 2015-01-12. [Online]. Available: https://github.com/ACSLab/PheenoRobot

[21] Pololu Robotics & Electronics. Accessed: 2015-01-12. [Online]. Available: https://www.pololu.com/

[22] Raspberry Pi 2 Model B. Accessed: 2015-01-12. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

[23] Arduino Pro Mini. Accessed: 2015-01-12. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardProMini

[24] Gizmoszone. Accessed: 2015-01-12. [Online]. Available: http://www.gizmoszone.com

[25] ServoCity. Accessed: 2015-01-12. [Online]. Available: https://www.servocity.com/

[26] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.

[27] S. Wilson, T. P. Pavlic, G. P. Kumar, A. Buffin, S. C. Pratt, and S. Berman, "Design of ant-inspired stochastic control policies for collective transport by robotic swarms," *Swarm Intelligence*, vol. 8, no. 4, pp. 303–327, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11721-014-0100-8