The GRITSBot in its Natural Habitat - A Multi-Robot Testbed

Daniel Pickem[1], Myron Lee[2], and Magnus Egerstedt[3]

***Abstract*— Current multi-agent robotic testbeds are prohibitively expensive or highly specialized and as such their use is limited to a small number of research laboratories. Given the high price tag, what is needed to scale multi-agent testbeds down both in price and size to make them accessible to a larger community? One answer is the GRITSBot, an inexpensive differential drive microrobot designed specifically to lower the entrance barrier to multi-agent robotics. The robot allows for a straightforward transition from current ground-based systems to the GRITSBot testbed because it closely resembles expensive platforms in capabilities and architecture. Additionally, the GRITSBot's support system allows a single user to easily operate and maintain a large collective of robots. These features include automatic sensor calibration, autonomous recharging, wireless reprogramming of the robot, as well as collective control.**

I. INTRODUCTION

Multi-agent robotics focuses on controlling large numbers of robots to accomplish collective tasks that go beyond individual agents' capabilities. Recent advances in multi-agent robotics have resulted in a vast body of work in the theoretical domain, on algorithms, and on control methods for such collectives of robots. To highlight the diversity of these collective tasks, consider collective transport, collective construction, collective SLAM, coverage control, vehicle routing problems, self-assembly, self-disassembly, or self-reconfiguration. With these advances arises the need for experimental verification. Oftentimes, the verification step is done in simulation for reasons of complexity, time or cost constraints. Nonetheless, an implementation of theoretical results on actual hardware is ultimately necessary. Current multi-agent testbeds come at a high price, both in financial terms as well as time to maintain and operate such a testbed. State-of-the-art experimental setups can cost tens of thousands of dollars in hardware alone. This begs the question *"Does multi-agent robotics have to be this prohibitively expensive?"* A number of recent low-cost hardware implementations suggest that there exists an interest in affordable testbeds in the multi-agent systems community.

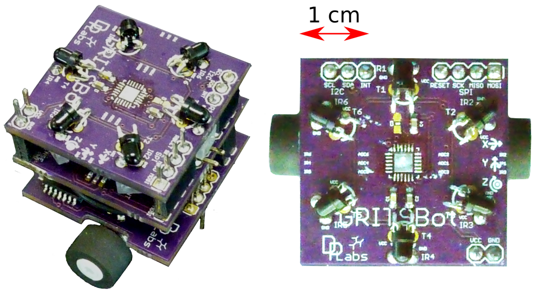

Fig. 1. Isometric and top view of the GRITSBot.

Several hardware implementations have been proposed to serve as inexpensive testbeds for multi-agent experiments and experimental verification of algorithms for collective tasks. The M-blocks [13] have been used for self-reconfiguration, 2D self-assembly and collective transport have been implemented on the Kilobots [14][15], and collective construction has been verified on the TERMES robotic system [12]. Most of these hardware platforms, however, have been developed for use in a specific setting.

Note, however, that a number of collective tasks can be implemented using wheeled ground robots, for example vehicle routing [1], coverage control [2], or collective exploration [11]. Therefore, as varied as these research domains and results are, the systems used for implementation and verification of theoretic results are similar in most research labs - wheeled ground robots and optionally a motion capture system to measure position and orientation of robots.

With that in mind, we have developed the GRITSBot (see Fig. 1), a low-cost differential drive microrobot aimed at closely replicating capabilities of current systems and enabling researchers to set up a multi-agent testbed that resembles current state-of-the-art platforms at a fraction of the cost. The goal for the GRITSBot is to remove the barrier to entry by showing that it is possible to design a low-cost, high-performance multi-robot system. As such, the primary design focus was on a low price, high usability, and a straightforward transition from current experimental setups to the GRITSBot.

Similar to the GRITSBot, the Kilobot [14] was a step in the direction of a standard multi-robot research platform. However, its locomotion type complicates the transition from current platforms and experimental setups. Several other popular microrobotic platforms have been developed, an overview of which is given in Table I.[1] Few of these

This research was sponsored by AFOSR/MURI Project #FA9550-09-1-0538

[1]Daniel Pickem is with the department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA daniel.pickem@gatech.edu

[2]Myron Lee is with the department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA mlee350@gatech.edu

[3]Magnus Egerstedt is a professor at the department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA magnus@ece.gatech.edu

[1] We are aware of a number of other swarm robots, such as the marXbot, the WolfBot, the Garcia robots, etc. However, these platforms are significantly larger in size.

are commercially available (only the Kilobot, the Khepera robot, and the e-puck). Two other platforms claim to have their designs freely available online (the Alice and the Jasmine robot). However, the corresponding research projects no longer maintain their websites actively. That leaves a choice between the very inexpensive Kilobots with limited locomotive capabilities and costly differential drive robots such as the e-puck or Khepera. The GRITSBot was designed to fill that void and furthermore serve as an easily accessible, inexpensive, and space-efficient platform for multi-agent research. The rest of this paper introduces the design rationale and requirements in Section II, describes the individual functional blocks of the GRITSBot in Section III, demonstrates the capabilities of the GRITSBot in Section IV, and concludes in Section V.

II. DESIGN REQUIREMENTS

A common system setup based on differential drive ground robots and some form of motion capture system is used in a number of research labs. A new microrobot testbed should therefore closely resemble this setup. Additionally, the required capabilities of a testbed are determined by the algorithms the microrobots are tasked to execute. The decentralized nature of a variety of multi-agent algorithms (such as rendezvous, formation control, vehicle routing) at a minimum require local sensing and accurate locomotion. Furthermore, to broaden the scope of such a robotic testbed, it should offer the capability of remotely operating robots or closing the feedback loop remotely (for example for global position control). As such, some form of global communication and positioning system is also required. In summary, a multi-robotic testbed should have at least the following capabilities.

- High resolution and accuracy locomotion
- Range and bearing measurements
- Global positioning system
- Wireless communication with a global host

Furthermore, as robots are scaled down in size and scaled up in numbers additional maintenance and usability features become indispensable. These features allow a single user to easily handle large numbers of robots without the need to individually operate, program, charge, or calibrate them. These convenience features significantly speed up the experimental process and simplify the maintenance of a large collection of robots.

- Automatic sensor calibration
- Automatic battery charging
- Wireless programming
- Local communication between robots

In the development of the GRITSBot, the main focus lied on low-cost, small size, scalability, and simplicity of design, assembly, and usage.

A. Simplicity

Commercial availability is an advantage for those labs not equipped to assemble microrobots. However, it introduces a significant markup cost over the cost of parts alone. The GRITSBot was designed with ease of assembly in mind. Therefore, the total number of SMD (surface mount devices) components per board was kept to a minimum. The motor board contains 14 parts, the main board 22, and the sensor board 13 SMD parts in addition to 12 through-hole components. Therefore, not counting the header pins that connect the individual boards the total part count comes to just 61 components.[2] Manual assembly can be accomplished in one to two hours.

B. Modularity

Multi-robot systems can be used in a variety of settings each with specific requirements regarding sensing, actuation, and processing. Adaptability to environmental constraints and functional requirements dictates a modular design. For example, certain experiments might not require sensing, in which case, it should be simple to reduce the robots functionality. The layered design of the GRITSBot allows for fine-grained adaptability of its functionality by removing or adding layers.

C. Scalability

In simulation, multi-robot experiments can contain hundreds or thousands of robots. However, typical hardware implementations are limited to at most hundreds of robots [4][6] or in case of the Kilobot [14] to 1024. Depending on the required capabilities, certain limitations are imposed on the number of concurrently operating GRITSBots as well. On one hand, the overhead camera imposes limits on the total size of the environment (if absolute positioning is required). On the other hand, the bandwidth of the RF channel limits the number of robots (if global communication is required).

D. Low Cost

A major barrier for the widespread adoption of multi-agent testbeds is their prohibitive cost and to a lesser extent their size (both the size of individual robot as well as the full testbed setup). Whereas commercially available, wheeled robots are being sold at prices as low as $99 (e.g. the 3pi robot by Pololu, see also Table 1 in [**?**]) few of these low-cost platforms are viable research platforms (mostly due to a lack of required sensors). Robots such as the e-Puck and the Khepera III are fully capable and assembled research platforms, however their price limits their use to well funded labs (see Table I). On the lower end of the price spectrum one finds the Jasmine robot (see [6]), the Alice robot (see [4], [4]), the R-One (see [8]), and the newest addition, the Kilobot ($14 to $50 in parts depending on order quantities or $115 fully assembled, see [14]). With the exception of the Kilobot, these robots are neither commercially available nor are their designs available for replication anymore. The GRITSBot is fully open-source and available online and a single robot can be built for under $50.

[2] By comparison, the Kilobot uses 78 parts (based on the public bill of materials)

TABLE I

AN OVERVIEW OF MULTI-ROBOT PLATFORMS

Robot	Cost	Scalability	Odometry	Sensors	Locomotion	Size [cm]	Battery life [h]
GRITSBot	$∼ 50[1]	charge, program calibrate	stepper motors	distance, bearing, 3D accel., 3D gyro	wheel, 25cm/s	3	1-5
Kilobot [14]	$50[1,2,4]	charge, power program	other agents	distance, ambient light	vibration, 1cm/s	3.3	3-24
Jasmine [6]	$130[1]	charging	wheel encoders	distance, bearing, color	wheel, 50cm/s	3	1-2
Alice [4]	N/A	none	wheel encoders	distance, bearing, cliff	wheel, 2cm/s	2.1	1-10
r-one [8]	$220[1]	none	wheel encoders	visible light, 3D accel., 2D gyro, bump, IR	wheel, 30cm/s	10	6
SwarmBot [**?**]	N/A	charge, program, power, calibrate	wheel encoders	range, bearing, camera, bump	wheel, 50cm/s	12.7	3
e-puck [3]	$979	none	wheel encoders	range, bearing, 3D accel. microphones	wheel, 13cm/s	7.5	1-10
Khepera III[3]	$2750	none	wheel encoders	distance, bearing, IR ground sensors	wheel, 50cm/s	13	1-8

[1] Cost of parts

[2] Note that this price refers to order quantities of 100 or fewer

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

[4] Available for purchase at http://www.k-team.com/mobile-robotics-products/kilobot for $1150 for 10 robots

E. Small Form Factor

Available space in terms of room size often restricts the total number of robots in multi-agent experiments to few tens of robots (see for example [3], [**?**]). Recently a lot of work has been dedicated to miniaturizing robots to the extent where a testbed fits on a table (see [4] or [14]). That enables a much larger audience to participate in multi-robot experiments at a fraction of the cost and space requirements of previous hardware implementations. The GRITSBot features a footprint of 31 × 30 millimeters, which is approximately the size of a Kilobot.

F. A Support System - Usability

Since ease of use was a main design requirement for the GRITSBot, tools for setting up and maintaining a collective of microrobots is required. As indicated in [14] such convenience features include collective programming, powering and charging, as well as collective control. An additional tool we developed was automatic sensor calibration (see Section III-G). All these tools aim at automating the menial tasks of maintaining a large collective of robots by minimizing physical interaction with the robots. For example, an EEPROM chip on the GRITSBot enables wireless programming of both the motor board and the main board. In addition to individually reprogramming a robot based on its unique wireless ID, it is also possible to broadcast reprogram all available robots or groups of robots.

III. THE GRITSBOT

The GRITSBot features a layered design, where each layer fulfills a specific purpose (see Fig. 2) and can be swapped in case of up-/downgrades or replacements. This modular design was adopted for two main reasons: flexibility in adjusting the required capabilities of the robot to specific experiments and simplicity in design. Each layer was designed with a specific function in mind. This section describes each of the five functional blocks of the robot that are distributed across three circuit boards or layers.

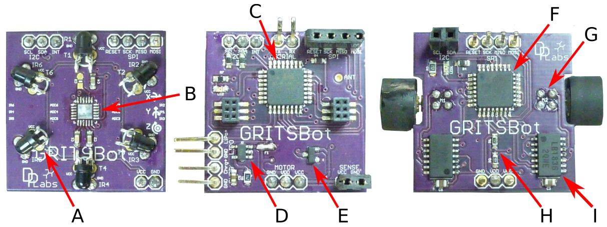

Fig. 2. The layers of the GRITSBot from left to right - sensor board, main board, motor board. The robots features include: (A) IR distance sensors, (B) accelerometer and gyro (currently not equipped), (C) main microcontroller, (D) battery charger, (E) voltage regulator, (F) motor board microcontroller, (G) stepper motor (on the bottom of the board), (H) battery voltage measurement, (I) motor controller.

A. Locomotion

One of the novelties of the GRITSBot is its locomotion system. Unlike previous microrobots, the GRITSBot does not use conventional DC motors and therefore does not require encoders to estimate their velocities. Instead, locomotion is based on miniature stepper motors. By their very nature, stepper motors completely obviate the need for velocity estimation since the target velocity of each motor can just be set through regulating the delay between individual steps. Odometry therefore is reduced to merely counting steps, which can be used to compute the velocities of the robot and estimate its position.

Since encoders can introduce significant estimation inaccuracies, others have attempted to circumvent their use. A recent approach to encoder-free odometry has been proposed in [5]. In that implementation, however, complex signal processing is required to compute motor velocities. The Kilobot (see [14] and [15]) addresses odometry in a different way. Since its vibration motors do not allow for encoder-based odometry, the Kilobot estimates its position based on measured distances to stationary neighbors. A drawback of this approach is the dependence on other agents. In the design of the Kilobot, vibration-based actuation was chosen for cost reasons. However, the costs of the encoder-free stepper motor design of the GRITSBot are comparable[3], yet offers high-accuracy locomotion at a top linear velocity of up to 25 cm/sec and rotational velocity of up to 820 degrees/sec.

B. Sensing

A primary requirement of a microrobot used in a multi-robot testbed is the measurement of distances and bearings to neighboring agents as well as obstacles. For reasons of both sensor size and cost, the GRITSBot, like most other microrobots, relies on infrared-based distance sensing. Six IR transmitters and receivers are arranged in 60° increments around the circumference (see Fig. 2(A)). In Section IV we will show how the consensus algorithm can be executed on the GRITSBot.

In addition to IR sensing, the sensor board also houses an accelerometer and gyroscope whose data can be fused into the velocity and position estimation to account for example for slip. One more sensor is mounted on the motor board, a battery voltage sensor (see Fig. 2(H)), whose data informs the control of the robot's recharge behavior (see IV-B).

C. Communication

The GRITSBot is equipped with an RF transceiver operating at 2.4 GHz. These low-power transceivers were chosen over WiFi for the main reason of reduced power consumption. Whereas a typical WiFi chip consumes approximately 250 mA, the integrated RF transceiver can operate at currents as low as 16 mA, thus drastically increasing battery life. The drawback of these low-power transceivers, however, is their lower data rate, which is limited to 2 Mbit/s.

A desirable feature of a multi-robot testbed is certainly local communication. Whereas in principle the GRITSBot is capable of local IR-based communication, no such communication protocol is currently implemented. Note however, that no local communication is required for most multi-robot experiments where distance and bearing measurements are available. If the need for local communication arises, it can be simulated through global communication to a host system.

D. Processing

The GRITSBot is equipped with two microcontrollers running at 8 MHz, an Atmega 168 chip on the motor board and an Atmega 328 chip on the main board. Whereas the Atmega 168 chip is solely responsible for motor velocity control of the stepper motors, the Atmega 328 is tasked with wireless communication, sensor data processing and user-defined high-level tasks such as obstacle avoidance or other behaviors.

E. Power System

The GRITSBot is powered by a 150 mAh single-cell lithium polymer (LiPo) battery that supplies a nominal voltage of 3.7V to the robot, which is then regulated down to 3.3V - the system operating voltage. Both the power regulation and the battery charging circuitry are embedded into the main board, which supplies power to the motor and sensor board through header pins. The charging circuitry of the robot operates at 5V input voltage. When connected to a power supply, it charges the battery through a single-cell LiPo charging chip (see Fig. 2(D)).

TABLE II
TOTAL COST PER ROBOT

Parameter	Value	Comment
Main board	12.60$	Power management, RF, main processing
Motor board	12.34$	Actuation and motor control
Sensor board	16.00$	IR sensing, accelerometer, gyro
Various	5.78$	Battery, chassis
Total	46.72$	

Currently, depending on the activity level of the robot, it can operate between 30 minutes to five hours. Note that for battery life measurements the obstacle avoidance behavior shown in Section IV-A was used at a duty cycle of 50% meaning that the robot was moving only half the time. Also, wireless communication was deactivated for this test, but no other power conservation measures were used. The robot remained operational for 63 minutes on a single battery charge. If longer battery life is required, the GRITSBot can be equipped with a battery up to 400 mAh thereby almost tripling its battery life. Note that with an autonomous charging behavior in place, the robot can recharge its battery within approximately 30 minutes thus extending its battery life indefinitely. As shown in Section IV-B, the charging station is embedded into the testbed walls and therefore the robot can recharge without operator intervention.

F. Cost

The costs per robot are based on order quantities of at least 25 robots, which results in parts cost of about 45$ per robot making the cost comparable to the Kilobot at low quantities. Table II summarizes costs by boards. Assembly is not factored into the cost shown in Table II. The GRITSBot can be assembled by SMD pick-and-place machines or, given the low number of parts, manually in approximately one to two hours.

G. The Testbed

1) Calibration: Since the robot measures a voltage with its IR sensors, one has to establish the mapping between distance sensor voltages and the actual distance values in meters. The calibration station (see Fig. 3) provides such a mechanism and allows for an automatic calibration of the robot's IR sensors. In case the calibration data is lost or corrupted, the robot can be recalibrated with minimal user intervention.

The current model of the calibration station uses two stepper motors - one that rotates a platform holding the robot and one that linearly moves an obstacle. The rotating stepper motor ensures that only one of the robot's distance sensors is active and pointing directly at the obstacle. The second motor varies the distance of the obstacle to the

[3] The locomotion system of the GRITSBot costs $4.32 for orders of 25 robots compared to $3.12 for the Kilobot at quantities of 1000 robots

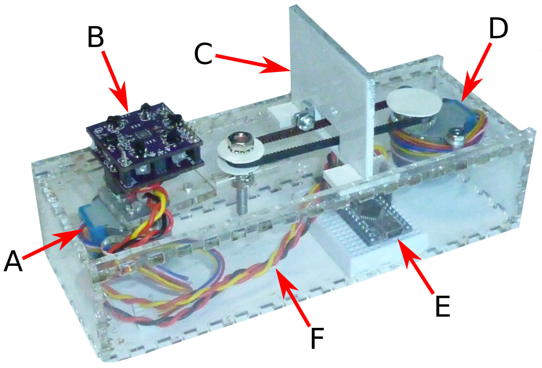

Fig. 3. Automatic sensor calibration with the following components: (A) stepper motor rotating the robot platform, (B) a GRITSBot being calibrated, (C) controlled obstacle, (D) stepper motor controlling linear distance of obstacle to robot, (E) microcontroller, (F) communication and power supply between robot and calibration station.

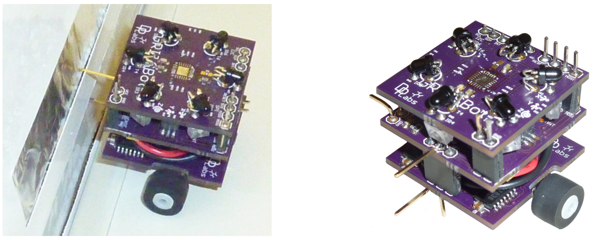

Fig. 4. The charging station for autonomous recharging of the GRITSBot's batteries.

robot in known increments which are then mapped to each of the corresponding sensor voltages. After this process is repeated for all six sensors, the information is written to the non-volatile EEPROM memory of the robot's main board. Therefore, the robot retains its calibration data and does not have to be recalibrated after a power cycle.

2) Charging: Along with the calibration station, autonomous charging provides a crucial mechanism for an ecosystem of self-sustaining robots. The GRITSBot is designed with two extending prongs at different heights that connect to two aluminum strips embedded in the arena walls. One of the metal strips supplies a 5V input voltage while the other serves as ground. This setup allows the GRITSBot autonomously drive up to the charging station (see Fig. 4).

3) Global Positioning: Each robot is equipped with an identification tag. This is currently implemented using the AprilTags C++ library[4] (see [10]) and 2D tags. Using an overhead camera, the absolute coordinates of each tag can be tracked and therefore global position data can be sent wirelessly to each robot. Additionally, the coordinates of the charging station are known to the system so that it can guide the robots to the charging station. In this work, we used a Microsoft LifeCam Studio at a resolution of 800x600 pixels whose data was processed using an Intel i7-4500U processor. In this setup, we achieved frame rates of up to 10 fps for up to 5 tracked tags and 5 - 7 fps for up to 25 tags.

IV. EXPERIMENTS

This section demonstrates the functionality of the GRITSBot for multi-agent experiments. The first behavior - obstacle avoidance - uses only local information from the robot's IR sensors, thus requiring no external input. The charging behavior shows how to close the position feedback loop through the overhead camera system and illustrates the robot's nonlinear controller. And lastly, we implemented the consensus algorithm to show the robot's suitability for multi-robot purposes.

A. Obstacle avoidance and random walk

The obstacle avoidance behavior is implemented as a simple finite state machine with three states: forward, reverse, random turn. The robot moves forward until it detects an obstacle, backs up for a few centimeters upon obstacle detection, turns left or right for a random amount of time, and proceeds moving forward. Note that this behavior is purely reactive and thus independent external input. Obstacle avoidance as a basic behavior can be combined with other behaviors such as consensus.

B. Charging behavior

This behavior can be implemented in two ways, a random walk in the environment that concludes once the robot finds the boundary of the environment and starts charging. Or it can be implemented as a behavior that has access to global position data from the overhead camera. In this section, to demonstrate closing the feedback loop through the overhead camera, we implemented the second variant of this behavior. In charging mode, the robot runs a nonlinear velocity controller and a linear position controller. More specifically, the GRITSBot can be modeled using unicycle dynamics.

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned}$$

By controlling a point d in front of the robot offset by length l (see [7]), we can feedback linearize the dynamics. Let $x' = x + l\,cos\theta$ and $y' = y + l\,sin\theta$. Then the dynamics can be rewritten as follows.

$$\begin{aligned} \dot{z} &= \begin{bmatrix}\dot{x}'\\ \dot{y}'\end{bmatrix} = \begin{bmatrix}v_x\\ v_y\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & l\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \\ &= R(\theta)S(l)\begin{bmatrix}v\\ \omega\end{bmatrix} = G(\theta,l)v \end{aligned}$$

Then we can rewrite $v = G^{-1}v_{lin}$. This transformed system now has linear dynamics, in which the linear velocities of point d (i.e. v_{lin}) are mapped to linear and rotational velocities of the unicycle model (i.e. v and ω).

$$\begin{aligned} v &= \cos(-\theta)v_x - \sin(-\theta)v_y \\ \omega &= \frac{1}{l}\left(\sin(-\theta)v_x + \cos(-\theta)v_y\right) \end{aligned}$$

The linear input velocities v_x and v_y in this equation are computed using a linear feedback position controller that

[4] http://people.csail.mit.edu/kaess/apriltags/

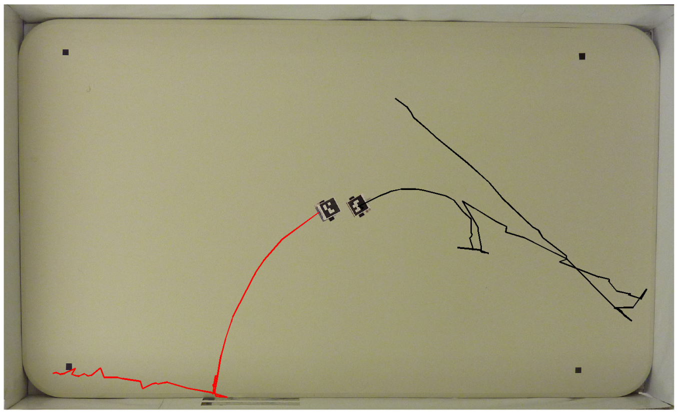

Fig. 5. The trajectories of the consensus behavior combined with obstacle avoidance and a random walk behavior in case the robots are farther than their sensing radius apart. Note that the test arena shown here measured approximately 122 x 72 centimeters.

receives its position feedback from the overhead camera system shown in Section III-G.3.

C. Rendezvous or consensus

The rendezvous problem is a canonical problem in multi-agent robotics and involves N agents reaching an agreement of some scalar or vector quantity in a network (see [9]). In this experiment, the quantity to agree on is a two-dimensional position, i.e. we want all agents to meet at an unspecified common location. The rendezvous dynamics can be described with the consensus equation. Assuming the position of an agent i is x_i, its neighbors $\mathcal{N}_i$ are all agents withing a certain radius δ, and each agent can measure relative displacements, i.e. $x_i - x_j, \ \forall j \in \mathcal{N}_i$. Then the consensus equation results in agents moving towards the centroid of all its neighbors' positions.

$$\dot{x_i} = -\sum_{j \in \mathcal{N}_i} (x_i - x_j)$$

Since it is difficult to distinguish between other agents and obstacles based on IR distance measurements alone, in this implementation, robots receive displacement information from the overhead camera system of section III-G. Furthermore, note that $\delta < 10cm$ for the GRITSBot. If no other agent is in agent i's sensing range (i.e. $|\mathcal{N}_i| = 0$), then it executes a random walk behavior. Fig. 5 shows an example of consensus with two robots. The shown trajectories show both the random walk behavior (the jagged looking parts of the trajectory) as well as the eventual consensus approach.

V. CONCLUSIONS

In this paper, we have presented the GRITSBot, a low-cost differential drive microrobot designed to shrink the typical multi-agent testbed to fit on a table and make multi-agent robot experiments accessible to a large community. For this purpose, the GRITSBot offers features that minimize maintenance efforts and significantly increase usability - including automatic sensor calibration, autonomous recharging, and an inexpensive overhead positioning system using a simple webcam. The GRITSBot provides a low-cost robotics research platform and drastically lowers the entrance barrier into the world of multi-agent robotics.

REFERENCES

[1] A. Arsie, K. Savla, and E. Frazzoli. Efficient routing algorithms for multiple vehicles with no explicit communications. *Automatic Control, IEEE Transactions on*, 54(10):2302–2317, 2009.

[2] G. Arslan, J.R. Marden, and J.S. Shamma. Autonomous vehicle-target assignment: A game-theoretical formulation. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):584–596, 2007.

[3] M. Bonani, X. Raemy, J. Pugh, F. Mondana, C. Cianci, A. Klaptocz, S. Magnenat, J. C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. *Proceedings of the 9th Conference on Autnomous Robot Systems and Competitions*, 1:59–65, 2009.

[4] G. Caprari, P. Balmer, R. Piguet, and R. Siegwart. The autonomous micro robot alice: a platform for scientific and commercial applications. In *Micromechatronics and Human Science (MHS), 1998 IEEE International Symposium on*, pages 231–235, 1998.

[5] Serge Kernbach. Encoder-free odometric system for autonomous microrobots. *Mechatronics*, 22(6):870–880, 2012.

[6] Serge Kernbach, Ronald Thenius, Olga Kernbach, and Thomas Schmickl. Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adaptive Behaviour*, 17(3):237–259, 2009.

[7] Hassan K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition edition, December 2001.

[8] James McLurkin, Jennifer Smith, James Frankel, David Sotkowitz, David Blau, and Brian Schmidt. Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 72–75, 2006.

[9] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, July 2010.

[10] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407, 2011.

[11] L. E. Parker and A. Howard. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25:431–447, 2006.

[12] K. Petersen, R. Nagpal, and J. Werfel. Termes: An autonomous robotic system for three-dimensional collective construction. In *Robotics: Science and Systems (RSS), 2011 Conference on*, 2011.

[13] John Romanishin, Kyle Gilpin, and Daniela Rus. M-blocks: Momentum-driven, magnetic modular robots. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4288–4295, 2013.

[14] M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3293–3298, 2012.

[15] M. Rubenstein, A. Cabrera, J. Werfel, G. Habibi, J. McLurkin, and R. Nagpal. Collective transport of complex objects by simple robots: theory and experiments. In *Autonomous Agents and Multiagent Systems (AAMAS), 2013 International Conference on*, pages 47–54, 2013.