Buckybot: Preliminary Control and Mapping Algorithms for a Robot Geometrically Based on a Truncated Icosahedron

Robert C. Grande, Michael D. M. Kutzer, Christopher Y. Brown, and Mehran Armand

In this article, we present preliminary work on motion planning and mapping algorithms for the Buckybot mobile robotic platform. We investigated implementation of wall-following algorithms and mapping unknown indoor environments by relying on rudimentary dead-reckoning and ultrasonic range finders. Buckybot is a ground-based platform whose geometry is based on a truncated icosahedron (a soccer ball shape with flattened sides). This platform has 20 passive hexagonal faces on which it can stably rest and 12 rounded pentagonal faces that can be extended linearly, allowing Buckybot to move. Because the robot is operational in any configuration, it is ideal for a variety of deployment scenarios, including throwing or dropping. Simulations grounded in experimental results show preliminary feasibility of Buckybot for indoor mapping applications.

INTRODUCTION

Buckybot is a new mobile robotic platform based on a truncated icosahedron (i.e., a soccer ball shape with flattened sides). It can rest stably on any of its 20 hexagonal faces and has 12 linearly actuated pentagonal faces that can be used to tip from hexagonal face to hexagonal face (Fig. 1). Each hexagonal face is adjacent to three actuators, allowing reliable movement in three directions. Pentagonal faces are rounded to prevent Buckybot from resting on a single actuator. In its current configuration, Buckybot moves by extending a single pentagonal face until the center of mass shifts sufficiently to incite a passive tip onto an adjacent hexagonal face.

The isotropic geometry of Buckybot makes platform locomotion independent of orientation. This can be advantageous when traversing unknown environments and in deployment scenarios in which there is a tumbling motion. As a result, one can deploy Buckybot in a variety of unconventional ways; Buckybot can be conceivably thrown, kicked, rolled, dropped, launched, etc., without compromising postdeployment locomotion. Additionally, the nearly spherical shape of Buckybot provides the possibility of both passive and active rolling, which can be ideal for fast locomotion scenarios and descending steep slopes.

Figure 1. Image of Buckybot with all 12 of its actuators extended.

Other groups have investigated and developed spherical rolling platforms.[1,2] Unlike these platforms, Buckybot is unique in that it currently uses a configuration with flat faces and relies on linear extensions of its sides to locomote. Although polyhedrons cannot roll as quickly and require more energy for rolling because of impact with the ground, they have several benefits. For example, polyhedrons can rest on modest slopes without any actuation and can move in low-gravity or low-friction environments in which traditional wheel-based robots cannot operate.[3]

Given the current tipping locomotion strategy, Buckybot is constrained to walk on what can be described as a honeycomb grid. Although this does not impose profound operational constraints, it does prohibit the use of many traditional trajectory and wall-following algorithms. For example, for systems with continuous dynamics and continuous sensing, transfer functions can describe the relationship between user input and wall distance. With a continuous transfer function defined, feedback controllers can be designed to stabilize the systems. With Buckybot, we have a quantized input space, which is a function of both the current position and orientation. Some groups have performed research on motion planning subject to kinodynamic constraints, with lattices using linear integer programming techniques, and with the A* algorithm.[4–6] However, these algorithms work better for motion planning and obstacle avoidance than for wall following. As a result, we propose new algorithms for wall following, with the goal of incorporating them into these more established algorithms in future work.

Given the geometry of Buckybot, to allow for equal sensing capabilities in all valid orientations, we propose the addition of sensors on all 20 hexagonal faces. For the proposed wall-following and mapping application, we will investigate the use of inexpensive range finders placed at the center of each passive hexagonal face and pointing radially outward. For the purposes of this preliminary work, we created an experimentally grounded simulation of the Buckybot platform with integrated ultrasonic range finders. To do so, we tested Buckybot and our proposed range finders independently. The purpose of these tests was to realistically define the pose uncertainty of Buckybot and the sensor noise associated with the range finder. Using this simulation, we evaluate the possibility of using a range finder-integrated Buckybot as a platform for autonomous navigation and mapping of unknown indoor environments. As part of this work, we also develop algorithms for identification of walls and wall following for Buckybot.

BUCKYBOT PLATFORM

Actuation and Control

Our current Buckybot prototype is approximately 26.0 cm (10.23 in.) in diameter with the pentagonal faces fully retracted. The distance between opposite hexagonal faces for this scale platform is approximately 23.9 cm (9.42 in.). Pentagonal faces are actuated using Haydon size-11 noncaptive stepper motor linear actuators. These actuators enable a reliable extension of up to 6.7 cm (2.64 in.) to enable tipping from face to face. Because of the slow speed of these actuators, tipping is currently the only feasible gait for the current Buckybot prototype. New actuators are currently under development for a smaller, faster, more agile Buckybot.[7] A review of this effort will be discussed in *Conclusions and Future Work*.

The Buckybot is controlled wirelessly via Bluetooth. Commands are received and echoed using Roving Networks FireFly (RN-240/422) serial adapters connected to a communication board containing a mixed-signal microcontroller (C8051F410, Silicon Laboratories Inc.) that interprets and relays commands to a motor communication bus. Each motor is controlled independently using a motor board also containing a mixed-signal microcontroller (C8051F410, Silicon Laboratories Inc.). Motor boards are given unique addresses and are connected to the communication bus in series. Buckybot also contains an inertial measurement unit leveraging a Honeywell HMC6343 tilt-compensated magnetometer and its own unique Bluetooth module (RN-41, Roving Networks). This independent wireless connection enables streaming inertial measurement unit data to a remote interface without affecting communication with Buckybot.

All electrical components are powered using six 3.7-V, 1050-mAh polymer lithium-ion cells (Powerizer PL-553562-10C) wired in two parallel sets. These batteries offer an operating life of greater than 4.7 h (assuming actuators are run continuously). In future iterations, additional sensors will be incorporated into the system to enable environmental sensing and data collection. For the purpose of this work, range finders were considered for indoor autonomous mapping applications.

Geometry and Sensing

As mentioned in the *Introduction*, Buckybot is geometrically based on a truncated icosahedron. However, because the robot can only rest stably on its 20 hexagonal faces, Buckybot can be considered a regular icosahedron for the sake of motion planning. While on a given face, Buckybot can tip in one of three directions. With each step, Buckybot's possible tipping directions shift by 60°. As a result, Buckybot is constrained to walk on a honeycomb lattice. To identify orientation, each hexagonal face is numbered from 1 to 20, and each actuator is labeled A through L.

Using the accelerometer available on the inertial measurement unit, Buckybot's current resting face can easily be determined. To sense the world outside the robot, we simulated range finders placed in the center of each hexagonal face and pointing radially outward. While sitting on a given face, we assume that the four bottom- and top-most range finder integrated faces will not provide useful information about surrounding obstacles. Of the 12 remaining range finders, the upper six are oriented 19.47° above the horizontal, and the bottom six are oriented 19.47° below the horizontal. Assuming that walls and obstacles are vertical and floors remain level relative to Buckybot, the bottom faces have a visibility of only about 23.0 cm (9 in.) before the sensor picks up the floor. As a result, we consider only the six sensors oriented 19.47° above the horizontal for our algorithms.

As shown in Fig. 2, these remaining sensors are not spread uniformly around the robot in the horizontal plane. Adjacent range finders are separated by 44.5°, and range finders separated by an actuator are separated by 75.5°. As Buckybot moves, the orientation of the sensing lines changes relative to the global frame. This must be considered and accounted for in control algorithms.

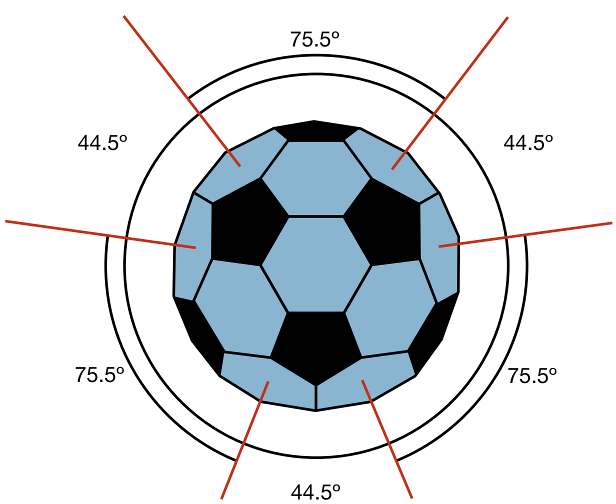

Figure 2. View of Buckybot from the top, with projected line of sight of range finders.

WALL-FOLLOWING AND MAPPING ALGORITHMS

In this section, we discuss wall-following algorithms for Buckybot by using feedback from the six sensors mentioned in the preceding section. We assume that all walls are vertical and sufficiently tall that Buckybot can sense them. At the maximum sensing distance, 183 cm (72 in.), walls need only be 80 cm (31.5 in.) tall to be sensed.

Identifying and Locating Walls

An important and basic function of many ground-based mobile robots is the ability to locate and follow a wall at a prescribed distance by using feedback. For Buckybot, the first step to this is locating and properly identifying a wall by using the proposed range finder network. To complete this first step, we developed criteria for wall identification by using the following assumptions. First, we assumed that the walls are long enough to ensure that two adjacent sensors can pick up the same wall. If only one sensor registers an obstacle, then we assumed that the sensor is picking up an object and not a wall. Second, we defined a cutoff range for wall detection for all sensors. If an object is detected farther away than the cutoff range, in this case 183 cm (72 in.), then Buckybot will ignore it.

By using basic geometry, both the distance of the robot to the wall and the orientation of the wall can be determined. To make these determinations, we use a combination of law of cosines, Heron's formula, and law of sines.

Using Fig. 3, we first solve for z using the range finder readings (x, y) and the law of cosines. Once z is defined, we apply Heron's formula 1 to solve for the area (A) of the associated triangle,

$$A = \sqrt{s(s-x)(s-y)(s-z)}\,, \tag{1}$$

where $s = \frac{x+y+z}{2}$ is the semi-perimeter. Noting that the area of the associated triangle can be equivalently defined,

$$A = \frac{1}{2}dz, \tag{2}$$

where d is the orthogonal distance from the wall to the center of Buckybot (Fig. 3). Combining Eqs. 1 and 2, and solving for d, we find that the associated distance is defined:

$$d = \frac{2}{z}\sqrt{s(s-x)(s-y)(s-z)}\,. \tag{3}$$

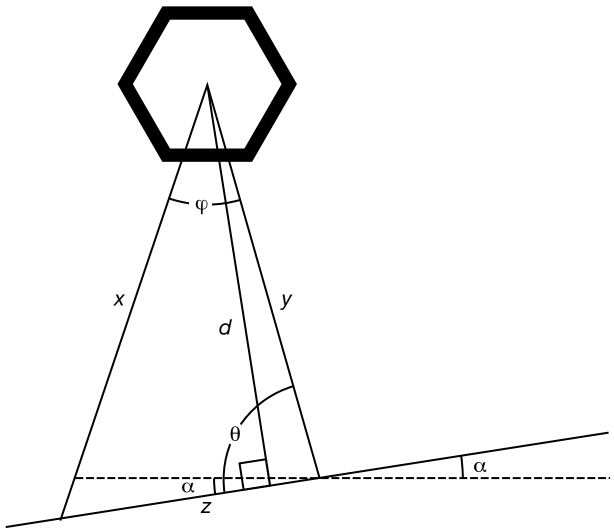

Figure 3. Buckybot wall-following geometry.

Before successful wall following, the wall's inclination, α (Fig. 3), relative to the closest range finder must be determined. Using law of sines to find θ (Fig. 3), and noting that the sum of angles in a triangle is 180°, we find:

$$\alpha = \arcsin\left(\frac{x}{z}\sin(\varphi)\right) - \frac{180 - \varphi}{2}. \quad (4)$$

Wall Following

With the distance to the wall d and inclination of the wall α determined, algorithms can be created to select a wall and follow it. It should be noted that with walls identified, wall-following algorithms can be chosen freely to suit the needs of the task at hand. For this preliminary work, we will follow the closest wall in a counterclockwise direction at a fixed distance of d^*. To accomplish this, we define a cost function (Eq. 5) weighing the importance of accurate wall following with the wall-following speed:

$$J((u(k)) = \sum_{i=1}^{k} \| (d(i) - d^*) \| + \| x(i) - (x(i-1) + x_d) \| . (5)$$

Here, x_d is a factor used to weigh the importance of moving along the wall as opposed to maintaining a distance d^* from the wall. The control input u defines the step sequence, and k defines the total number of steps in a given sequence.

The cost function is minimized on a per-step basis. Before each step, we generate a target for the robot's movement. To generate this target, we first find the point d^* away from the wall along the shortest path connecting Buckybot's center to the wall. We then add a vector parallel to the wall of a prescribed length x_d, as illustrated in Fig. 4. By increasing this vector's magnitude, we promote fast traversal of the wall rather than maintaining an ideal distance of d^* from the wall.

To impose the counterclockwise traversal constraint, we define the positive direction parallel to the wall as the cross product of the +Z (out of page) direction with the shortest vector connecting the center of Buckybot to the wall. With the target determined, the robot considers its three viable movements and selects the one that minimizes the distance to the target. An alternative method could use the A* algorithm, applying heuristics to penalize walking too close or too far from the wall.

Anticipated Implementation Issues

As with many control algorithms, there are certain scenarios that may produce undesirable results. For this approach, we see two potential problems. First, corners, most notably 90° corners, are difficult to navigate. Second, narrow hallways and confined spaces can cause unnecessary confusion. To compensate, we propose the addition of a simple set of rules.

The first issue arises because Buckybot considers only the closest wall when planning its next move. Depending on which wall is determined to be closer, the target for the next step changes. In the right circumstances, this can cause the robot to get stuck in the corner, alternating between faces until perturbations in the system cause the eventual navigation of the corner. Of the solutions that compensate for this issue, the simplest involves an increase in the value of x_d, which results in improved navigation of corners. Additionally, a rule prohibiting backtracking movements can be applied; however, this has its own set of drawbacks, as one can easily imagine a scenario in which Buckybot might need to backtrack to successfully continue wall following. In future work,

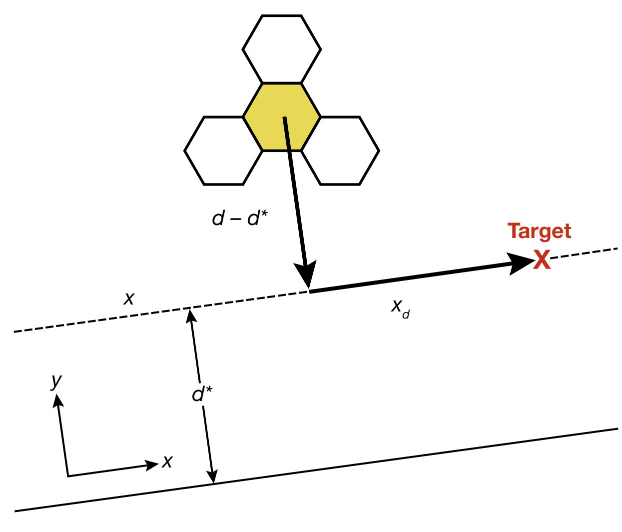

Figure 4. Buckybot wall-following planning. The center hexagon is the current resting face, and the three adjacent hexagons represent possible moves.

we will consider a more robust solution using the four remaining range finders to detect upcoming corners and allowing us to plan accordingly.

Narrow hallways and confined spaces can also cause confusion. This confusion arises from the imposed constraint that forces Buckybot to follow the closest wall in a counterclockwise direction. In a scenario in which Buckybot is placed nearly equidistant from a sufficiently parallel pair of walls, a situation can arise in which Buckybot's wall-following algorithm will alternate between walls. In this situation, Buckybot will get stuck alternating between faces until perturbations in the system cause it to select one wall over the other. This can be overcome by adding an exception to our imposed counterclockwise constraint on the wall-following direction. Specifically, if a wall is sensed on the opposite side of the robot, the robot will begin following in a clockwise direction. Note that this simple fix can cause further issues if sharp corners are present in the environment. As with the previous issue, we will address a more robust solution or set of solutions in future work.

EXPERIMENTAL ERROR DETERMINATION

Currently, Buckybot has yet to be equipped with range finders. To compensate, we independently evaluated both Buckybot's tipping performance and the range finders' performance to create a realistic simulation environment to validate our algorithms.

Test Setup

For range finder testing, we identified the LV-MaxSonar-EZ ultrasonic range finder (MaxBotix Inc.) as an accurate, easy to use, and cost-effective sensor. To determine the accuracy and noise of the LV-MaxSonar-EZ, we created a test rig to mount sensors 19.47° above horizontal (Fig. 5). For testing, we considered three individual sensors, each tested at distances ranging from 15.24 cm (6.0 in.) to 243.84 cm (96.0 in.) on an interval of 7.62 cm (3.0 in). At each distance, 25 measurements (consisting of five time-averaged readings each) were taken from each individual sensor. An Arduino Pro Mini using a 10-bit analog-to-digital converter was used to take readings from the range finders. Data were collected to the nearest 2.54 cm (1 in.) due to the automatic filtering of the LV-MaxSonar-EZ.

To test Buckybot, we evaluated every possible tipping combination a total of 10 times. We used a Vicon tracking system (Vicon MX Giganet controller with four Vicon MX-T10 1.0-megapixel cameras). For this system, the position accuracy determined during calibration is consistently better than ±0.7 mm. For each tipping combination, two rigid bodies containing reflective markers were attached to Buckybot. One rigid body was attached opposite the resting face of Buckybot. The second rigid body was attached opposite the face Buckybot will tip to. An image of Buckybot with the two rigid bodies attached is shown in Fig. 6. Once the rigid bodies were attached, Buckybot was tipped using its relevant actuator a total of 10 times for each of the

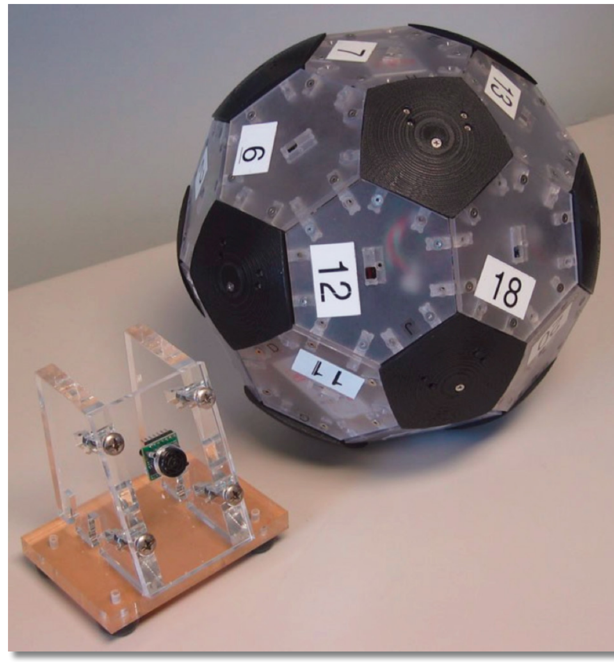

Figure 5. Image of the test rig (left) designed to mimic the 19.47° angle of the six faces used for distance measurements. Faces 2, 12, and 18 on Buckybot (right) represent three of these six faces.

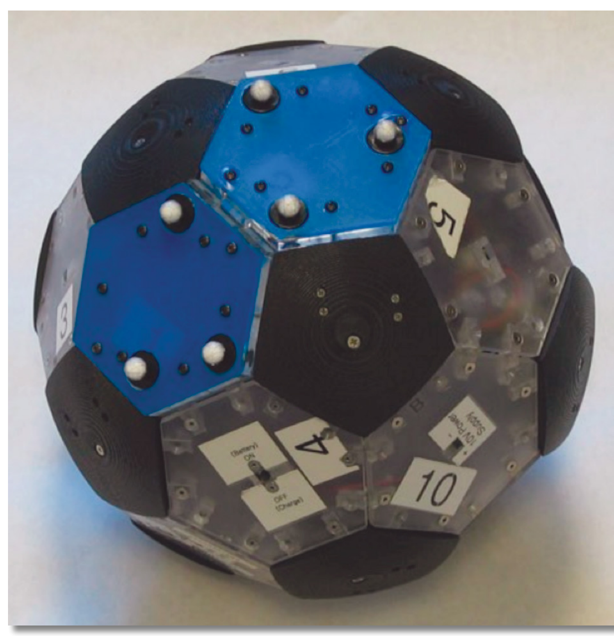

Figure 6. Image of Buckybot with Vicon tracking markers attached to faces opposite the current resting face and the face that Buckybot will tip to.

60 tip combinations, tracking its movement (position and orientation) through the entire sequence. In this work, we considered only the first and last frame of this sequence corresponding to the two resting states of Buckybot (pre- and post-tip) to determine the repeatability of tipping.

Test Results

We found that the range finders had low noise for ranges rate, from 30.48 cm (12.0 in.) to 182.88 cm (72.0 in.). At ranges closer than 30.48 cm (12.0 in.), the sensors did not act reliably and fluctuated greatly. At ranges farther than 182.88 cm (72.0 in.), two of the sensors notably lost accuracy. These issues will be addressed in *Simulation and Results*. Using this information, we used least-squares methods to develop a best-fit noise model. This yields the following:

$$d_{measured} = d_{actual} + (0.013d_{actual} + 0.448)\mathbf{w}, \quad (6)$$

where $d_{measured}$ is a simulated noisy measured distance (cm), d_{actual} is the true distance (cm), and $\mathbf{w}$ is normally distributed noise with mean zero and unit variance.

With the Buckybot test results, we found some differences in the accuracy of tipping onto certain faces. However, differences were not significant enough to justify accounting for them individually in this initial effort. As such, we define the mean and standard deviation in position and orientation over all possible moves combined. We further simplify the system by assuming that there is no correlation between position and orientation measurements. In doing so, we find the mean and standard deviations for our positions as $\mu_x = -0.745$ cm, $\sigma_x = 0.149$ cm, $\mu_y = 9.266$ cm, and $\sigma_y = 0.105$ cm, where the $+y$ direction is defined as the tipping direction, and the $+x$ direction is defined such that the $+z$ direction points out of the floor. The mean and standard deviation associated with our heading measurements are $\mu_\theta = 0.31°$ and $\sigma_\theta = 0.44°$, respectively. Note that θ is defined about the $+z$ axis relative to the initial $+y$ direction. With these values, equations matching the form of Eq. 6 can easily be assembled.

SIMULATION AND RESULTS

A simulation environment was created in MATLAB in which a virtual Buckybot (Fig. 7) can detect obstacles and walls. By using this simulation, we can evaluate the algorithms detailed in *Wall Following*. At each step, noise is added into the range finder measurements according to Eq. 6 and then reported to the nearest 2.54 cm (1 in.) to account for the automatic filtering of the LV-MaxSonar-EZ. As mentioned before, the sensors do not work reliably at distances less than 30.48 cm (12.0 in) or greater than 182.88 cm (72.0 in). At distances too close,

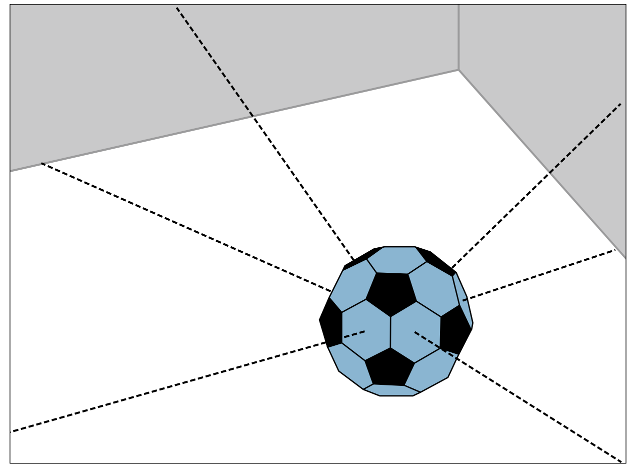

Figure 7. Image from the simulation environment.

the sensors do not settle to a final value. In our setup, we assume a static environment and that Buckybot is static between tips. Thus, if sensors do not settle in a fixed time interval, it can be reasoned that Buckybot is very close to an obstacle. At distances too far, the sensors jump to large values unexpectedly. Thus, if a distance greater than 182.88 cm (72.0 in.) is detected, we will ignore the sensor reading entirely.

The position of Buckybot is also altered at each step according to the statistics reported in *Test Results*. These noise parameters are added into the Buckybot position estimates with each step using equations similar to Eq. 6.

When running the simulation, range finder measurements are recorded in a matrix in which each column represents a sensor number and each row represents a step taken by the robot. The sequence of actuators fired is also recorded. After the simulation runs, we repeat the ideal trajectory of the robot using the sequence of actuators fired, only this time omitting noise. We can then replot the points where Buckybot sensed an obstacle. Figure 8 shows a sample of a single trial. After running multiple simulations of different environments, we found that the additive position error was more substantial than originally thought. By running Monte Carlo simulations, we found, for example, that the standard deviation in uncertainty in position after 100 steps (roughly traveling 929 cm) was 90.42 cm (35.6 in.) and the standard deviation in orientation was 11.5°. However, several methods to reduce this noise are discussed in the following section.

CONCLUSIONS AND FUTURE WORK

In this article, we presented the preliminary implementation of wall-following algorithms and the mapping of unknown indoor environments relying on rudimentary dead-reckoning and ultrasonic range finders. The system of interest was the Buckybot mobile platform whose

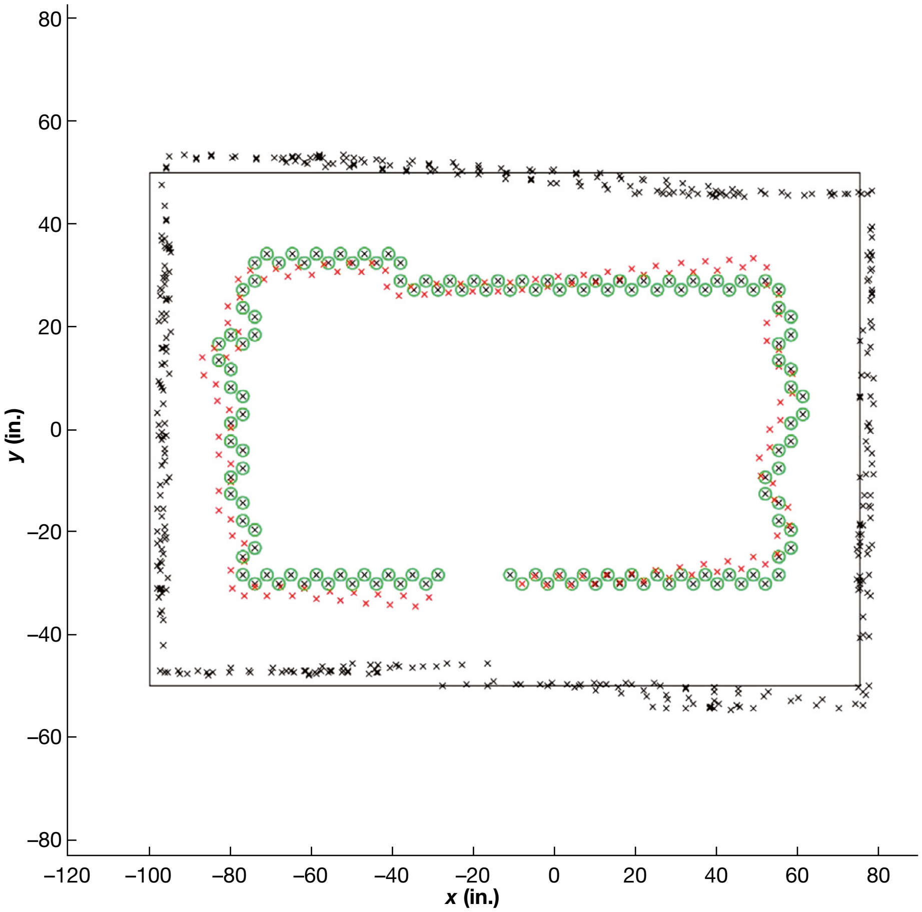

Figure 8. Map created by Buckybot simulation. Circles represent theoretical positions, red x's represent actual positions, and black x's represent locations where Buckybot simulation detected an obstacle assuming no uncertainty in position.

isotropic geometry allows for deployment in unconventional ways and is ideal for low-gravity or low-friction environments. The system currently locomotes by tipping using linear actuators; however, work is currently underway to replace these actuators with high-impulse spring actuators.[7] This would allow for smoother, faster tipping as well as the possibility of other modes of locomotion such as continuous rolling or jumping.

For the purposes of rapid capabilities assessment, the Arduino Pro Mini was used for simple data acquisition from range finding hardware. This choice of hardware was solely for the purposes of this testing, and this data acquisition capability can easily be incorporated into an extension of Buckybot's onboard control system without requiring the explicit addition of an Arduino Pro Mini.

Although intuition suggested that Buckybot's locomotion would result in a relatively low position uncertainty due to its discrete dynamics, we found through experimentation that the uncertainty was more significant than previously thought. We believe that this is an engineering problem and that solutions can be proposed to reduce the uncertainty in position and orientation significantly. For example, to reduce impact forces, we can simultaneously extend counteracting actuators. This will allow Buckybot to gradually come to rest on a tipping face, reducing the overall tipping speed and thereby reducing or eliminating the impact after a tip. Additionally, the rigid faces of Buckybot could be replaced with a compliant material to absorb some of the energy of impact.

Our implementation of a wall-following algorithm on the Buckybot platform was repeatedly demonstrated in simulation with positive results, even with realistic noise parameters applied. Mapping was also investigated, with results that were greatly affected by the position uncertainty associated with Buckybot's locomotion. In future work, we plan to incorporate our wall-following algorithms into a unified control scheme that is more robust and requires less special considerations. This control scheme will likely build on applications of linear programming or the A* algorithm. We also will investigate use of the Kalman filter and simultaneous localization and mapping algorithms to achieve better mapping capabilities. Lastly, we will consider other sensing systems such as networks of cameras to increase sensing capabilities.

As mentioned previously, work has been conducted in collaboration with The Johns Hopkins University Department of Mechanical Engineering senior design program to develop actuators for a softball-sized (8.9 cm) Buckybot.[7] Although initial efforts were promising, we are still working to improve overall actuator performance. The current actuator design involves a three-stage mechanism consisting of a charging phase during which a small DC motor slowly charges the actuator spring, a fire phase during which the energy from the actuator spring is released in a high-impulse linear extension, and a fast retract phase during which the extension is quickly retracted and locked back into place for charging. The specific goal of this work is to develop actuators capable of producing extension impulses high enough to enable both jumping and fast rolling of the

Buckybot platform. Actuators are designed to be fired in sequence to enable predictable parabolic jumping for increased terrain handling, including stair climbing. To enable rolling, the fast retract phase was added to the actuator functionality to retract actuators before they hinder rolling.

In addition to traditional range finders for mapping, a variety of tools and sensors are being considered for the Buckybot platform. Specifically, the geometry lends itself to a variety of sensing modalities, including acoustic sensing, chem/bio, and visual sensing. In circumstances in which identical sensors can be distributed around the body of Buckybot, simple gradient-based path generation techniques can be used to guide the robot toward or away from a desired source (e.g., a chemical plume or an acoustic source). In a case in which multiple cameras are distributed around the surface of Buckybot, camera calibration and image meshing techniques can be used to give an operator a stable view from any direction as the robot moves. Additionally, these meshed video feeds can also enable an operator to effectively look around without requiring movement from the robot.

ACKNOWLEDGMENTS: The authors thank Mr. Wolfger Schneider and Mr. Rafal Szczepanowski for their extensive contributions to the development of the Buckybot platform and Dr. Gregory Chirikjian for providing range finders for this work. The authors also thank Mr. Aaron Goldblum and Ms. Anupama Challa for their assistance with the sizable data collection associated with this work. This work was supported by the Janney Publication Program and Independent Research and Development funds provided by APL.

REFERENCES

[1]rotundus, "Robot Design," http://www.rotundus.se/design.html (Aug 2011).

[2]Wait, K., Jackson, P., and Smoot, L., "Self Locomotion of a Spherical Rolling Robot Using a Novel Deformable Pneumatic Method," in *Proc. IEEE International Conf. on Robotics and Automation (ICRA2010)*, Anchorage, pp. 3757–3762 (2010).

[3]Hokamoto, S., and Manabu, O., "Dynamic Behavior of a Multi-Legged Planetary Rover of Isotropic Shape," in *Proc. 6th International Symp. on Artificial Intelligence and Robotics & Automation in Space (i-SAIRAS 2001)*, St.-Hubert, Quebec, Canada, pp. 1–6 (2001).

[4]Mason, R., and Burdick, J., "Trajectory Planning Using Reachable-State Density Functions," in *Proc. IEEE International Conf. on Robotics and Automation (ICRA '02)*, Washington, DC, pp. 273–280 (2002).

[5]Pivtoraiko, M., Knepper, R., and Kelly, A., "Differentially Constrained Mobile Robot Motion Planning in State Lattices," *J. Field Robotics* **26**(3), 308–333 (2009).

[6]Hart, P., Nilsson, N., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Systems Sci. Cybernetics* **4**(2), 100–107 (1968).

[7]Johnson, R., Ferguson, D., Kegelman, J., Lefkowitz, J., Rajpal, T., et al., "Senior Design Project: Miniature Actuator for Throwable Robot," *Johns Hopkins APL Tech. Dig.* **28**(3), 272–273 (2010).

The Authors

This investigation was conducted out of curiosity during the team's off-hours during the summer of 2011 while **Robert C. Grande** was interning as part of the operations team of STEREO in the Civilian Space Department. The test bed was completed by **Michael D. M. Kutzer** and **Christopher Y. Brown** and in 2007 using Independent Research and Development funds. The original Buckybot concept was initially developed by **Mehran Armand** and colleagues in 2005. Since this effort, Robert Grande has continued his education and is currently working toward a M.S. in aerospace engineering at MIT. Michael Kutzer is investigating methods associated with the modeling and control of highly dexterous robotic manipulation systems. Christopher Brown's work currently focuses on efficient, effective, and predictable small-scale locomotion. Mehran Armand also continues to work in robotics research, with a focus on computer-assisted surgical systems. For further information on the work reported here, contact Michael Kutzer. His e-mail address is michael.kutzer@jhuapl.edu.

The *Johns Hopkins APL Technical Digest* can be accessed electronically at **www.jhuapl.edu/techdigest**.