# **PCBot: a Minimalist Robot Designed for Swarm Applications**

Jingxian Wang<sup>1</sup> and Michael Rubenstein<sup>1</sup>

*Abstract*— Complexity, cost, and power requirements for the actuation of individual robots can play a large factor in limiting the size of robotic swarms. Here we present PCBot, a minimalist robot that can precisely move on an orbital shake table using a bi-stable solenoid actuator built directly into its PCB. This allows the actuator to be built as part of the automated PCB manufacturing process, greatly reducing the impact it has on manual assembly. Thanks to this novel actuator design, PCBot has merely five major components and can be assembled in under 20 seconds, potentially enabling them to be easily mass-manufactured. Here we present the electro-magnetic and mechanical design of PCBot. Additionally, a prototype robot is used to demonstrate its ability to move in a straight line as well as follow given paths.

# I. INTRODUCTION

There are many envisioned applications of swarm robots, from distributed mapping and exploration [1, 2], modular self-reconfigurable robots [3–5], robotic warehouses [6], shape formation [7], etc. Often, these swarms' capabilities improve with increased numbers of robots. For example, a swarm with more robots can push heavier objects or move them faster in collective transport tasks [8, 9], or can approximate a desired shape in more detail in shape formation tasks [3].

The size of swarms in use is typically constrained by the robots' complexity, cost, and ease of manufacturing. To create larger swarms, designers are often forced to make them as simple, low cost, and easy to manually assemble as possible. Swarms of more complex robots such as SWARM-BOT [10] or e-puck [11] have been limited to under 100 robots. There were attempts to simplify swarm robots in the past, such as Kilobot [12], but even then, five minutes of manual assembly per robot limited the number of robots in a swarm to approximately 1000. On the contrary, biology frequently makes use of swarms and can have swarms that number in the 100,000 or millions of individuals such as those found in ant colonies, bird murmurations, or schooling fish.

One way to simplify the robots and ease their manufacturing is to rely on more automated manufacturing techniques for their assembly. One approach is to attach electrical components to a sheet that then self-folds to create the robot [13, 14], though robots created with this technique so far have been quite limited. Another approach is to take advantage of advances in printed circuit board (PCB) fabrication and component placement to build a robot almost entirely out of



Fig. 1. A photo of PCBot, a robot that uses a PCB-based actuator, and is designed for ease of manufacturing. PCBot's size is  $48 \times 42 \times 14$  mm and weighs 18 g.

a PCB, such as the HoverBots [15]. While Hoverbot partially inspired PCBot, it had limited battery life and can only work on a complex air-table with embedded magnets.

The number of actuators, as well as their complexity, also contributes a lot to a robot's overall complexity. Traditional robots use multiple motors and gearboxes in combination with wheels [6, 11] or propellers [16, 17] to move the robot. These driving mechanisms limit the robot's cost to hundreds of dollars, and part count to tens. In recent years, researchers have tried multiple ways to reduce the number and complexity of actuators, resulting in robots simpler in design and easier to manufacture. Kilobot [12], I-SWARM [18], and bristle bots [19, 20] replace traditional motors with vibration motors to eliminate the need of wheels and gearboxes. These robots usually have around 10 components and are a lot simpler than traditional robots, but the stochastic nature of vibration makes their movement unpredictable. On the other hand, Piccolissimo [21], Monospinner [22], and 1STAR [23] seek to reduce the number of actuators used to drive the robot. These robots feature only one actuator, usually directly 3D-printed body, and less than 10 components.

Robots with extremely simple actuators often rely on added complexity to the external environment. For example, HoverBots [15] are composed of only two major components: a battery and a PCB, and can move themselves on a special 'levitation-magnet table' using magnetic force between actuators integrated into the PCB and the table. However, the 'levitation-magnet table' is embedded with arrays of magnets and is complicated to manufacture and scale up. A simple robot described in [24] and Spinbot [25] utilize a planar table moving in an orbital manner to simplify the robot's locomotion. These two robots use electro-permanent magnets (EPMs) to control their attachment and detachment from the orbital shake table. EPMs used in these robots are

<sup>&</sup>lt;sup>1</sup>Jingxian Wang and Michael Rubenstein are with the Center for Robotics and Biosystems, Northwestern University, Evanston, IL 60601, USA jingxianwang2026@u.northwestern.edu, rubenstein@northwestern.edu

relatively time-consuming to produce [26], but give them the ability to move in any 2D direction on the table. While these robots are more complicated, the table they operate on is only a steel plate driven by several motors, and is simple to manufacture and scale up.

Here we present a robot called PCBot that builds on Spinbot [25] and HoverBot [15] to further reduce the complexity of a robot system designed for swarm applications. This robot can precisely move on an orbital shake table using a novel PCB actuator inspired by Hoverbot.



Fig. 2. An exploded view of PCBot. PCBot is comprised of merely five major components: a 3D printed chassis, a PCB, a battery, a steel core which is just a screw, and a magnet with rubber on one side. All these components are off-the-shelf or mass-manufacturable and can be assembled in under 20 seconds.

#### **II. ROBOT DESIGN**

## A. Overall concept

PCBot, see Figs. 1, 2, sets out to further simplify the manufacturing of a swarm robot by introducing a novel bistable solenoid actuator built using coil embedded directly into the robot's PCB. PCBot operates on a horizontal orbital shake table, shown in Fig. 3. As shown in previous work [24, 25], individuals can move in an arbitrary 2-D direction relative to the table surface by simply controlling when it sticks (attaches) and slips (detaches) on the table, as shown in Fig. 5. While using this single orbital shake table increases the complexity of the setup, it decreases the overall swarm complexity because it allows the individual robot design to be much simpler.

In addition to sliding over the table surface, PCBot also passively spins around its center in synchronization with the movement of the table when operating. This feature enables PCBot to track the movement of the table using its polarized light sensor, saving the need for gyroscopes. In future generations, the rotation behavior will enable PCBots to sense distance and bearing to neighbors as well as communicate with them using mechanisms similar to those presented in [25].



Fig. 3. The setup of the test environment. A polarized light source hangs from the ceiling to provide robots with global orientation sensing. The orbital shake table surface is made of steel and is coated with a layer of HDPE sheet. PCBots operate on the surface of the orbital shake table.

For PCBot's only actuator, we designed a novel bi-stable solenoid to quickly control the friction coefficient between the robot and the table surface. The bi-stable solenoid, as shown in Fig. 4, consists of a coil created directly by traces in the robot's multi-layer PCB and a coil-driven magnet that can move up and down in the chassis and switch between sticking to the orbital shake table (called 'attached' state) or sticking to the spacer on 3D printed chassis close to the steel core (called 'detached' state). The concept of creating a coil directly in a PCB is inspired by [15]. This design allows us to build an actuator without the manufacturing complexity of wire-wound coils or electro-permanent magnets [25, 27] and significantly reduces the major components required to build a single robot to merely five: a 3D printed chassis, a PCB, a battery, a steel core (a simple screw), and a magnet with rubber on one side. This consequently reduced the time required for manual assembly from tens of minutes or hours seen with most other robots, down to under 20 seconds for PCBot.



Fig. 4. A simplified cross-section view of the electro-magnetic subsystem of PCBot. This figure shows the difference between the magnet's position in the 'attached' and 'detached' states. In 'attached' state, the magnet sticks to the surface, and in 'detached' state, the magnet sticks to the spacer on the chassis close to the steel core.



Fig. 5. A simplified diagram showing how the robot moves. In the 'attached' state, as shown in the left part of the figure, the magnet sticks to the table and PCBot revolves around the magnet. When PCBot detaches, the magnet no longer touches the table. Driven by inertia, PCBot maintains its momentum, thus follows a straight path and slides on the table, as shown in the bottom part of the figure. After switching back to 'attached' state, friction force quickly brings PCBot to the same speed as the table and keeps them relatively static, as shown in the right part of the figure. In each actuation cycle, PCBot moves relative to the table, as indicated by the orange arrow. This figure is not drawn to scale.

## B. Theory of operation

The overall environment for this robot consists of an arbitrary number of PCBots all moving on an orbital shake table. In addition, there is a polarized light source shining down on the robots, see Fig. 3. Each PCBot is equipped with an ATTINY84 microcontroller, a 70 mAh rechargeable lithium-ion battery, and a single actuation mechanism consisting of a bi-stable solenoid and its associated electronics. Furthermore, to sense its orientation in the environment as it spins around its magnet, PCBot uses an upward-facing light sensor covered with a polarizing filter. As a debugging aid, there is also an RGB LED to provide visual feedback to the operator.

At the center of PCBot is its bi-stable solenoid actuator which allows it to attach and detach from the table surface. The actuation coil is built in the multi-layer PCB and generates a magnetic field that moves the Neodymium magnet up or down in the 3D printed chassis, as shown in Fig. 4. A boost regulator and an H-bridge are used to selectively drive current through this coil. When no current is applied on the coil, the magnet is bi-stable: it could stay in either 'attached' or 'detached' state. In the 'attached' state, there is a strong attraction force between the magnet and the steel beneath the table surface. Furthermore, there is a high friction coefficient between the rubber and the table surface. These two factors prevent the magnet from slipping, and the whole robot rotates around the magnet due to an offset center of mass. However, in the 'detached' state, the chassis which has a low friction coefficient is the only part of the robot that is contacting the table, and the friction is not enough to allow the robot



Fig. 6. Software and hardware architecture for the PCBot and its operating environment.

to continue moving with the table's movement. Thus, the robot slips and moves according to Newton's first law. An illustration of robot's movement mechanism is shown in Fig. 5. By controlling when PCBot attaches and detaches from the surface, we can move robots in the desired direction.

Aside from the movement mechanism, a key system for PCBot is the orientation tracking system made of an ambient light sensor covered by a polarizing filter. Because the light source in the environment is also polarized, the light intensity sensed by the ambient light sensor will change with the orientation of the PCBot. Thus, PCBots can track their orientation by feeding the ambient light strength sensed to a threshold-crossing-based software phase lock loop (sPLL).

### C. Magnetic field and geometric design

The core of PCBot design is the actuation mechanism, i.e. the electro-magnetic subsystem, shown in Fig. 4. This subsystem includes a magnet that can move up and down, a coil built in the multi-layer PCB, and a steel core to amplify the magnetic field. The chassis of the robot and the table surface serve to restrict the position of the magnet.

The key of the actuator is the bi-stable state of the magnet. A bi-stable magnet means the robot only needs to power the coil when changing the position of the magnet, but not keeping the magnet in a specific state. This dramatically decreases the power consumption of the mechanism compared to using an electromagnet to pull the magnet up in 'detached' state, because switching between states takes  $\approx 1 \text{ ms}$  but the electromagnet would need to be powered on during the entire 'detached' state for tens of milliseconds to seconds.

There are two necessary requirements for designing this system: first, that the magnet should be stable in either state when no current flows through the coil, second that the magnet should be able to switch between the states when current flows through the coil, as shown in Fig. 7.



Fig. 7. A simplified cross-section view of the electro-magnetic subsystem of PCBot. This figure shows how PCBot switches between the 'attached' and 'detached' states. When detaching, PCBot drives a current  $I_0$  through the coil and generates an upward magnetic force  $F(d, h_a, I_0) > 0$  on the magnet that pulls it upwards. When attaching, PCBot drives a current  $-I_0$  through the coil and generates a downward magnetic force  $F(d, h_d, -I_0) < 0$  that pushes the magnet downwards.

We first simplify the model and only consider the three most important and controllable parameters: the distance between the steel core and the steel table d, the distance between the magnet and the steel table h, and the current through the coil I. Other geometric parameters have a limited impact on the result and could be fine-tuned afterwards. The vertical force exerted on the magnet F(d, h, I) could be expressed as a function of d, h, I and computed using commercial finite element methods (FEM) software. Using this notation, our design requirements can be formally written as:

$$\begin{cases}
F(d, h_a, 0) < 0 \\
F(d, h_d, 0) > 0 \\
F(d, h, I_0) > 0 & (h \in [h_a, h_d]) \\
F(d, h, -I_0) < 0 & (h \in [h_a, h_d])
\end{cases}$$
(1)

where  $h_a, h_d$  are the position of the magnet in 'attached' and 'detached' state respectively, and  $I_0$  is the current through the coil when H-bridge is activated. When there is no current through the coil, and d is held constant, as h increases, the attraction force between the steel core and the magnet increases but the attraction force between the steel table and the magnet decreases, thus F(d, h, 0) is monotonically increasing with h. We can find the only equilibrium point  $h = h_{eq}$  that satisfies

$$F(d, h_{eq}, 0) = 0$$
 (2)

Similarly, we can define  $h_{eq}(I)$  as the equilibrium point closest to  $h_{eq}$  when current I flows through the coil, i.e.

$$F(d, h_{eq}(I), I) = 0 \tag{3}$$

For example, the plot for  $h_{eq}(I)$  and F(d, h, 0) with geometric configuration we eventually used is shown in Fig. 8. Using these definitions, we can simplify (1)~(3) to:

$$h_{eq}(I_0) < h_a < h_{eq}(0) < h_d < h_{eq}(-I_0)$$
(4)

When designing the PCBot, we had to choose a proper  $I_0, d, h_a$  and  $h_d$  that satisfies (4). In addition, we wanted



Fig. 8. Plot of equilibrium point's position  $h_{eq}(I)$  under various coil current I and robot geometry parameter d. Color in the background shows the force on magnet F(d, h, 0). The parameters we chose are  $I_0 = 5 \text{ A}$ , d = 7.5 mm,  $h_a = 1.85 \text{ mm}$ ,  $h_d = 2.90 \text{ mm}$ , as marked purple in the figure. The two crosses, which represent the chosen geometry, are between the two purple lines, with at least 0.1 mm margin. This means the coil can generate more than enough force to move the magnet up and down to switch between states.

to: minimize the power consumption, maximize the friction force between the magnet and the table in 'attached' state to prevent slipping, and also make sure that the robot design is robust to manufacturing variability. To minimize power consumption we should try to minimize  $I_0$ . With the same friction coefficient between the magnet and the table, the magnetic force on the magnet in 'attached' state determines the maximum friction force, so we should try to maximize  $-F(d, h_a, 0)$ . Furthermore, manufacturing error will make geometric parameters  $d, h_a, h_d$  vary, so it is also important to leave enough margin so (4) can still hold with variation in parameters.

The first thing we chose is the current we are willing to drive through the coil, which is limited by the electronics. In this case  $I_0 = 5$  A. Ideally, we would like to increase the force on the magnet to increase friction, which means choosing a point  $\{d, h\}$  that is as far left as possible on the equilibrium line  $h_{eq}(-I_0)$ . However, on the left side of the graph, manufacturing error can result in a large change in the required current to drive the magnet because equilibrium lines are tightly packed, so to compensate for that, we chose a point that is more to the right to match our manufacturing error. This result in the following parameters:  $I_0 = 5$  A, d = 7.5 mm,  $h_a = 1.85$  mm,  $h_d = 2.90$  mm.

## D. Mechanical design

In the mechanical design process, we were mainly concerned with the chassis's shape, the overall mass distribution, and the forces on the magnet. An abstract drawing of the mechanical system is shown in Fig. 9.



Fig. 9. Cross-section of robot's 3D printed chassis and the magnet.

We had three major goals for mechanical design of the system:

- 1) PCBot must slip along the surface in the 'detached' state.
- 2) PCBot must stick to the surface in the 'attached' state.
- 3) PCBot must stably rotate around the magnet in the 'attached' state.

Goals 1 and 2 mean we had to differentiate the friction force in 'detached' and 'attached' states as much as possible, so we chose Ultimaker Tough PLA material to 3D print the chassis, moisture-resistant HDPE as the orbital shake table's surface coating, and Neoprene rubber with criss-cross texture as the rubber sticker on the magnet. The static and kinetic friction coefficient between the chassis and the table is around 0.12 but the kinetic friction coefficient between the rubber and table is around 0.45 and the static friction coefficient is as high as 0.67. With this combination of material, the force from kinetic friction in the 'attached' state is around 73 mN, but merely 29 mN in the 'detached' state.

Goal 3 means we had to design the chassis so that the torque of the driving force (inertial force in the table's comoving frame) should be maximized while the torque of friction force acting on the chassis when spinning around the magnet should be minimized. So we designed a special chassis with the cross intersection as shown in Fig. 9. With this chassis, the total mass of the robot is  $18.0 \,\mathrm{g}$  and the center of mass is 5.2 mm away from the robot's axis of rotation. The mass distribution is asymmetric enough so the driving force from the table's movement has enough torque to combat friction in 'attached' state. At the same time, the mass of the robot is still small enough so that it won't slip as it spins in the 'attached' state. Concerning the torque of the friction force, the chassis has two extrusions on the bottom that will contact the surface: a ring close to the rotation center that contacts the surface at  $P_1$ , and a line-shaped extrusion further away that contacts the surface at  $P_2$ .  $P_1$ and  $P_2$  situate on either side of the center of mass and will contact the surface at all times. This design minimized the torque of friction force in both the 'attached' and 'detached' states. In comparison, if the chassis is a cylinder shell, in the 'attached' state the friction force would act on the boundary of the chassis, making the arm of friction force the radius of the chassis. This is approximately four times longer than that of the current design, in which the arm of friction force is approximately halfway between  $P_1$  and  $P_2$ .

# E. Coil design

One of the most significant limitations for small robots is battery life. For PCBot, the power consumption when the actuation mechanism is not activated is 18 mW, which is mainly consumed by the microcontroller. However, if the robot moves once every second, the average power consumption of the actuation mechanism could go above 500 mW, making it the limiting factor for battery life.

The main reason why the actuation mechanism is so power consuming is because we used coils built using a multi-layer PCB to generate magnetic field, so the number of rounds of the coil is a magnitude lower and the resistance is a magnitude larger compared to conventional coils built using regular wires. A viable way to reduce the power consumption of the actuation mechanism without changing the robot's electro-magnetic and mechanical design is to optimize the PCB coil and make it more energy-efficient at moving the magnet. In the following discussion, we will present an abstracted model that ignores manufacturing constraints like vias and component placements to provide an analytical analysis. The PCB coil is built on a multi-layer board, but there is little difference between layers, so we will just analyze a single-layer PCB coil instead.

A single-layer PCB coil takes the form of a spiral, as shown in Fig. 10. The inner and outer radius  $r_0, r_1$  of the coil is determined by the size of the steel core and the size of the robot, and the minimum gap between two consecutive rounds of coil w is determined by the PCB manufacturer (usually 0.2 mm for PCBs with 2 oz copper weight).



Fig. 10. The coil's geometric structure for the optimized coil design (left) and simple coil (right).

We can describe the PCB coil using function  $r(\theta)$  which is the equation of the coil's center line in polar coordinates and  $W(\theta)$  which describes the coil's width. We assume that both the radius and the width of the coil change slowly, i.e.

$$r''(\theta) \ll r'(\theta) \ll r \tag{5}$$

$$W'(\theta) \ll W \tag{6}$$

Consider the distance D between the center line of two consecutive rounds of coil, we get the relationship between coil's radius  $r(\theta)$  and coil's width  $W(\theta)$  under assumption (6).

$$r(\theta + 2\pi) - r(\theta) = D = \frac{W(\theta) + W(\theta + 2\pi)}{2} + w$$
 (7)

With assumption (6), we have  $r(\theta+2\pi)-r(\theta) = 2\pi r'(\theta+\pi)$ and  $W(\theta) + W(\theta+2\pi) = 2W(\theta+\pi)$ . By applying this to (7), we can express  $W(\theta)$  using  $r(\theta)$  as

$$W(\theta) = \frac{r'(\theta)}{2\pi} - w \tag{8}$$

Now we can describe a coil using only  $r(\theta)$ . In the next step, we proceed to formulate the magnetic force on magnet  $F(r(\theta), I)$  and the power consumption of the coil  $P(r(\theta), I)$  with arbitrary coil configuration  $r(\theta)$  and coil current I that satisfy (6).

For the magnetic force on the magnet, because the force generated by the coil is much weaker than the magnetic field of the magnet itself, and all the materials in the system have constant differential permissively under our operating condition,  $F(r(\theta), I)$  is almost proportional to the current I, and can be written as:

$$F(r(\theta), I) = F(r(\theta))I \tag{9}$$

where  $F(r(\theta))$  can be determined using FEM simulations.

For the power consumption of the coil, because we usually need to actuate the coil for around 1 ms to move the magnet, and the characteristic time for the LR circuit composed by the power source and the coil  $t_{LR}$  has  $t_{LR} \approx 10^1$  us  $\ll$ 1 ms, Joule's law could be used to calculate the power consumption:

$$P(r(\theta), I) = I^2(R_{ext} + R_{coil}) \tag{10}$$

where  $R_{ext}$  is the internal resistance of the actuation circuit, including the internal resistance of the power source (ESR of the capacitor in our case) and the on state resistance of the H-bridge, and

$$R_{coil} = \int_{r_0}^{r_1} \rho_s \frac{r(\theta) \, \mathrm{d}\theta}{W(\theta)} = \int_{r_0}^{r_1} \rho_s \frac{r(\theta)}{r'(\theta)(\frac{r'(\theta)}{2\pi} - w)} \, \mathrm{d}r$$
(11)

where  $\rho_s$  is the surface resistivity. We can observe that  $P(r(\theta), I)/F(r(\theta), I)^2$  is a constant regardless of current I and is only related to the coil design  $r(\theta)$ . Thus the optimal coil configuration can be obtained by numerically minimizing

$$\frac{P(r(\theta), I)}{F(r(\theta), I)^2} = \frac{R_{ext} + \rho_s \int_{r_0}^{r_1} \frac{r(\theta)}{r'(\theta)(\frac{r'(\theta)}{2\pi} - w)} \,\mathrm{d}r}{F(r(\theta))^2} \tag{12}$$

In realistic conditions with  $r_0 = 2 \text{ mm}$ ,  $r_1 = 14 \text{ mm}$ ,  $R_{ext} = 0.25 \Omega$ ,  $\rho_s = 0.25 \text{ m}\Omega$ , the optimized coil shown in Fig. 10 can save 40% energy comparing to a simple coil with fixed 0.7mm width. Furthermore, geometric constraints created by vias and nearby components severely reduced the performance of the simple coil, while the optimized coil was not influenced as much. By using the optimized coil in PCBot, we saved a staggering 70% of energy compared to using a simple coil.

### **III. EXPERIMENTS**

We conducted several experiments to demonstrate the capability of the PCBot and test its performance. When not specified, the orbital shake table moved at  $115 \pm 2$  rpm at an orbital radius of 10.7 mm, and the robot would lift the magnet up for 100 ms in each actuation cycle, resulting in an average displacement of 5 mm. Robots are autonomous, and there is no position feedback.

We recorded the movement of the table and the robot using an overhead calibrated camera, and then track the fiducial trackers on the table and the robot to obtain their translation and rotation.

# A. Straight line movement

We first conducted 36 experiments where a PCBot is programmed to walk in a straight line following the X-axis for 10 actuation cycles ( $\approx 47 \text{ mm}$ ) to test the reliability of movement mechanism. Experiments are carried out in different places on the table's surface to test its sensitivity to the surface quality. The results are shown in Fig. 11.



Fig. 11. Experimental results for straight line movement. Subfigure (a) is a composition of PCBot's configuration in the initial state, and after moving 5 and 10 actuation cycles in a typical experiment. Red crosses and blue dots show the magnet's starting position in each actuation cycle, and light blue lines connecting these dots show the path robot takes. Subfigure (b) is a more detailed view of magnet's position in the experiment corresponding to Subfigure (a). Subfigure (c) shows the magnet's path in all 30 experiments. The red cross in this figure marks the ending point in each trial.

For comparison, we also did 30 similar experiments but with slightly different parameters and half the magnet up time (50 ms in each actuation cycle). The results are shown in Fig. 12.



Fig. 12. Experimental results for straight line movement when magnet are lifted for 50 ms in each actuation cycle. Notations are the same as Fig. 11.(c)

We observed that the PCBot would not slip in 'attached' state, but would rapidly move in the desired direction in 'detached' state. When the magnet is lifted for 100 ms in each actuation cycle, the average displacement per 10 actuation cycles is 47 mm and the standard deviation for the displacement's direction and distance are  $8^{\circ}$  and 7 mm respectively. When the magnet is lifted for 50 ms in each actuation cycle, the average displacement per 10 actuation cycle, the average displacement per 10 actuation cycles is 21 mm and the standard deviation for the displacement's direction and distance are  $4^{\circ}$  and 5 mm respectively. The accuracy of direction increases while the accuracy of distance decreases when the magnet is lifted for shorter period of time. Accuracy in both cases is sufficient as long as there is external feedback to correct the robot's course.

By analyzing the correlation between the robot's movement and position, we discovered that the major cause for its non-uniform displacement is the difference in friction coefficient across the HDPE sheet used. The moving direction is related to the phase lag between the rotation of the PCBot and the rotation of the table, and the moving distance is related to the acceleration of PCBot when slipping on the table. Both the phase lag and the acceleration depends on the friction coefficient between the table and the chassis. This problem could be reduced by switching to a sheet with better uniformity across the surface or reducing step size by reducing the time in the 'detached' state in each actuation cycle. Other factors that might contribute to the non-uniformity are the slight variation in table rotation speed, imprecision in leveling of the table, and bumps on the table surface.

## B. Path following

We would also like to demonstrate the PCBot's path following capability, so we programmed a PCBot to move in a rectangular pattern where each edge is 15 actuation cycles long ( $\approx 70$  mm). The results are shown in Fig. 13.

As shown in Fig. 13, the PCBot could follow the path we gave. However, because PCBot does not have position feedback, the final position at step 60 is not aligned with the starting position. Their movement direction is relatively accurate, but their movement speed was not as consistent.



Fig. 13. Experimental results for path following experiment. This figure shows the composition of PCBot's position in the initial state and after moving every 15 actuation cycles. Red crosses and blue dots shows the magnet's starting position in each actuation cycle, light blue lines connecting these dots shows the path robot takes, and the yellow path shows the expected trajectory of PCBot.

Again, this is likely due to the variability of friction across HDPE sheet.

## C. Power consumption analysis

As suggested in section II-E, a general concern for small robots is battery life. In our case, the total distance PCBot can move is directly related to the number of actuations the onboard battery can support. Thus, it is important to know how much energy PCBot consumes in each actuation cycle and how many actuation cycles can the battery support. In our experiments, actuation time was set to 1000 us for detaching and attaching. On average, the power consumption when the movement mechanism was not activated was 18 mW, and each actuation cycle cost 68 mJ of energy. If we only consider the power consumption of the actuation mechanism, the 70 mAh on-board Li-ion battery allows the PCBot to move for about 14,000 actuation cycles. Under current operating conditions where the PCBot moves an average of 4.7 mm per actuation cycle, PCBots can move for about 66 m before the battery depletes. However, the distance PCBots can move in each actuation cycle can be further increased if we increase the time in 'detached' state. When PCBots stay in the 'detached' state for more than 200 ms in each actuation cycle, they can move more than 10 mm per step. This means that PCBots could travel for more than 140 m before the battery depletes. Considering that PCBots are less than 5 cm in size, this is a significant distance.

## IV. CONCLUSION

In this paper, we presented PCBot, a minimalist robot for swarm applications, explained its theory of operation, optimized its performance, and demonstrated its movement capability in real experiments. The design based on external actuation and novel bi-stable solenoid actuator built using PCB-based coil allows PCBots to possess basic control and precise movement capability with merely five major components.

PCBot only demonstrates its ability to move at present and lacks the inter-robot sensing and communication needed for a complete swarm robot. In future work, we will outfit them with IR transceivers to allow them to perform interrobot communication and bearing/distance sensing, similar to [25, 28]. It is worth noting that all these could be done by simply adding more circuitry to the PCB and should not involve adding more complexity to the assembly. With these future additions, PCBot would be a very simple, mass-manufacturable, yet powerful robot, making it a great candidate as testbed for a variety of swarm algorithms.

#### REFERENCES

- [1] Dieter Fox et al. "Distributed multirobot exploration and mapping". In: *Proceedings of the IEEE* 94.7 (2006), pp. 1325–1339.
- [2] Harris Chi Ho Chiu et al. "Tentacles: Self-configuring robotic radio networks in unknown environments". In: *IROS*. IEEE. 2009, pp. 1383–1388.
- [3] Kyle Gilpin, Ara Knaian, and Daniela Rus. "Robot pebbles: One centimeter modules for programmable matter through self-disassembly". In: *ICRA 2010*. 2010, pp. 2485–2492.
- [4] Petras Swissler and Michael Rubenstein. "FireAnt: A modular robot with full-body continuous docks". In: *ICRA 2018*. 2018, pp. 6812–6817.
- [5] Mark Yim et al. "Modular self-reconfigurable robot systems [grand challenges of robotics]". In: *IEEE RAM* 14.1 (2007), pp. 43–52.
- [6] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz. "Coordinating hundreds of cooperative, autonomous vehicles in warehouses". In: *AI magazine* 29.1 (2008), pp. 9–9.
- [7] Melvin Gauci, Radhika Nagpal, and Michael Rubenstein. "Programmable self-disassembly for shape formation in large-scale robot collectives". In: *DARS*. Springer, 2018, pp. 573–586.
- [8] Roderich Groß and Marco Dorigo. "Towards group transport by swarms of robots". In: *International Journal of Bio-Inspired Computation* 1.1-2 (2009), pp. 1–13.
- [9] Michael Rubenstein et al. "Collective transport of complex objects by simple robots: theory and experiments". In: *AAMAS 2013*. 2013, pp. 47–54.
- [10] Francesco Mondada et al. "Swarm-bot: From concept to implementation". In: *IROS 2003*. Vol. 2. 2003, pp. 1626–1631.
- [11] Francesco Mondada et al. "The e-puck, a robot designed for education in engineering". In: *ICARSC* 2009. Vol. 1. CONF. IPCB: Instituto Politécnico de Castelo Branco. 2009, pp. 59–65.

- [12] Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. "Programmable self-assembly in a thousandrobot swarm". In: *Science* (2014), pp. 795–799.
- [13] Samuel Felton et al. "A method for building selffolding machines". In: Science (2014), pp. 644–646.
- [14] Martin EW Nisser et al. "Feedback-controlled selffolding of autonomous robot collectives". In: *IROS* 2016. 2016, pp. 1254–1261.
- [15] Markus P. Nemitz et al. "HoverBots: Precise Locomotion Using Robots That Are Designed for Manufacturability". In: *Frontiers in Robotics and AI* 4 (2017).
- [16] Calvin Gregory and Andrew Vardy. "microUSV: A low-cost platform for indoor marine swarm robotics research". In: *HardwareX* 7 (2020), e00105.
- [17] Hugh Durrant-Whyte, Nicholas Roy, and Pieter Abbeel. "Construction of Cubic Structures with Quadrotor Teams". In: *RSS 2012*. 2012, pp. 177–184.
- [18] Heinz Woern, Marc Szymanski, and Joerg Seyfried."The I-SWARM project". In: *RO-MAN 2006*. 2006, pp. 492–496.
- [19] William P. Weston-Dawkes et al. "Towards rapid mechanical customization of cm-scale self-folding agents". In: *IROS 2017*. 2017, pp. 4312–4318.
- [20] Juan Edmundo Pozo Fortunić. Design and implementation of a bristle bot swarm system. Pontificia Universidad Catolica del Peru-CENTRUM Catolica (Peru), 2017.
- [21] Matthew Piccoli and Mark Yim. "Piccolissimo: The smallest micro aerial vehicle". In: *ICRA 2017*. 2017, pp. 3328–3333.
- [22] Weixuan Zhang, Mark W Mueller, and Raffaello D'Andrea. "A controllable flying vehicle with a single moving part". In: *ICRA 2016*. 2016, pp. 3275–3281.
- [23] David Zarrouk and Ronald S. Fearing. "1STAR, A one-actuator steerable robot". In: *ICRA*. 2014.
- [24] Hanlin Wang and Michael Rubenstein. "Autonomous mobile robot with independent control and externally driven actuation". In: *IROS*. 2016.
- [25] Andrew SaLoutos and Michael Rubenstein. "Spin-Bot: An Autonomous, Externally Actuated Robot for Swarm Applications". In: *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 211–224.
- [26] Steven Ceron et al. "Towards a Scalable, Self-Reconfigurable Robot with Compliant Modules". In: *IEEE MRS 2019*, 2019, pp. 47–49.
- [27] Nialah Jenae Wilson et al. "Scalable and robust fabrication, operation, and control of compliant modular robots". In: *Frontiers in Robotics and AI* 7 (2020), p. 44.
- [28] Marc Gyongyosi et al. "Low cost sensing and communication system for rotor-craft". In: *IROS 2017*.