Sand Pattern with Puffer Fish: A Study of Male Pufferfish's Drawing Behavior during Reproduction Seasons

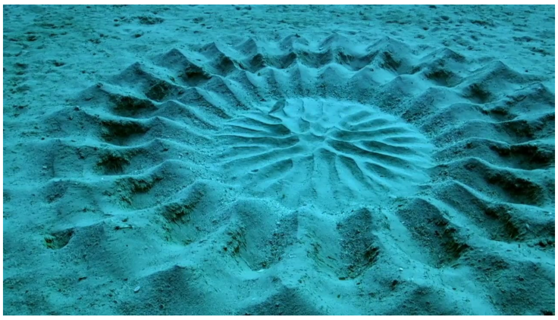
Yaming Xu, Weiyi He
Department of Interactive Media Arts, New York University Shanghai, Shanghai, China
(Tel: +8613162166160; E-mail: yx999@nyu.edu)

Abstract: This project studies the sand pattern created by male pufferfish during the reproduction seasons and re-created the similar pattern by using magnets, robot and sand table. The mechanism of the robot is also discussed in the paper.

Keywords: pufferfish, sand drawing, bio-inspired robotics, magnets

1. INTRODUCTION

20 years ago, around Amami-Oshima Island in Japan, 2-meter-wide sand patterns were discovered in the ocean. These sand patterns were nests constructed by male pufferfish during their reproduction seasons. These sand patterns are circular and symmetrical, with peaks and valleys on the edges.



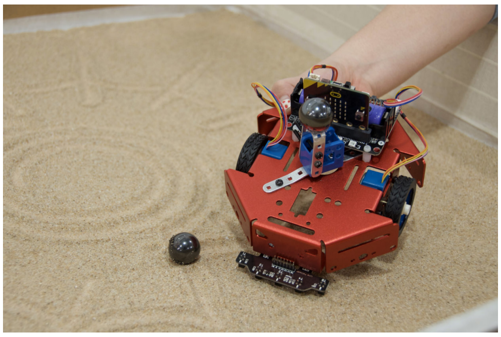
Male pufferfishes use this pattern to attract female pufferfish's attention during the reproduction seasons. They use their bodies to draw the pattern by swimming back and forth in circular and straight motions. It also collects shells to decorate the outer edge of the patterns. When the female pufferfish comes, the male pufferfish stirs up the sand to attract her attention. If the female pufferfish is satisfied with the nest, she will lay her eggs in the center of the nest and leave. The male puffer fish is responsible for egg hatching. It's still unknown about how female pufferfish chooses the nest, but there's one thing for sure: This structure slows down the speed of water flow, which creates a safe place for egg to hatch.

2. FIRST VERSION OF THE ROBOT

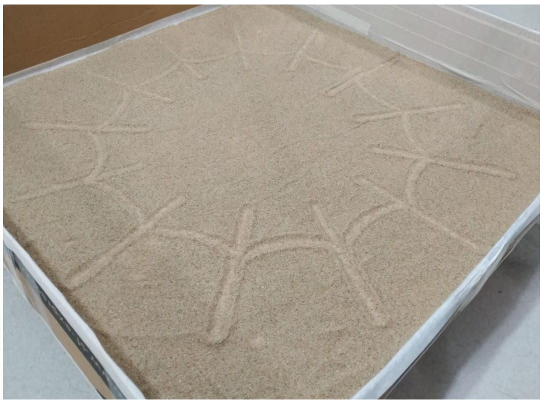
Materials:

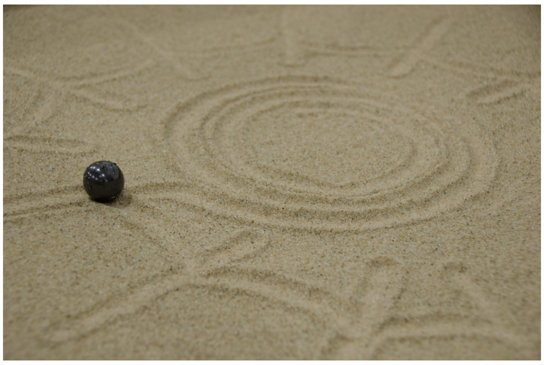
- KittenBot RobotBit v2.0
- Sand 5kg
- 91 * 91 cm sand table with 12mm thick bottom
- Magnet Ball * 2
- Stepper motor * 2

We made our first version for the midterm project. We started with the KittenBot RobotBit v2.0, and we realized that the original motors on the robot couldn't precisely move. Then we switched to stepper motors, which can move at a more accurate speed.



We designed a similar pattern which includes a spiral and an outline. The robot can perfectly draw the two pattern separately, but once we combined the two patterns, the bot can't go back to the starting point.





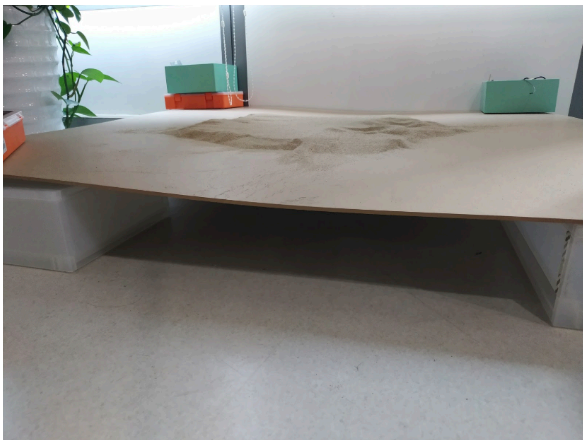
3.SECOND VERSION AND FAILURE

Materials:

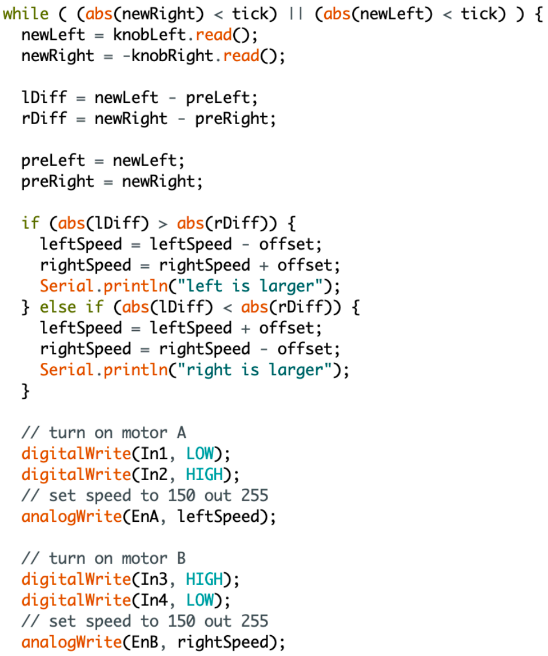
- Arduino Uno
- Motors with encoders * 2
- Motor driver
- A self-build robot with two wheels
- Sand 7.5kg
- 122 * 122 cm sand table with 5mm thick bottom
- Different shapes of super magnets

We have three goals for the second version: 1) Improve the accuracy; 2) Increase the scale; 3) Add depth to the drawing. In order to achieve these goals, we replaced stepper motors to motors with encoders, got a larger and thinner sand table, got some stronger magnets, and added more sand.

However, these changes also lead to some problems. First, heavy sand and thinner bottom made the board bend, and the curvature of the board prevents the robot to move smoothly.



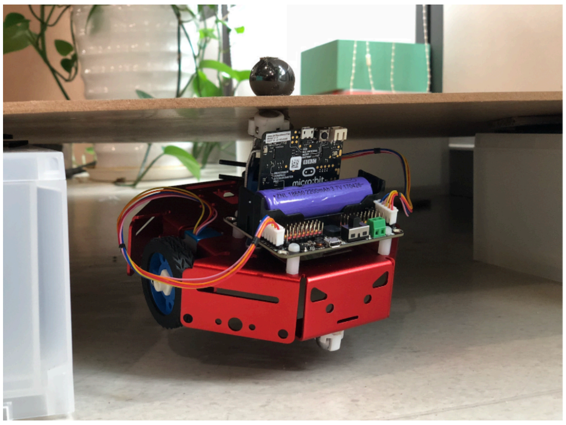
Second, motors with encoders can't move in a consistent and accurate speed. There are one input and one output for motors with encoders. The input is the ideal speed, and the output is the actual number of turns read by encoders. In order to make the two motors move in consistent and accurate speed, we programmed Arduino to compare the two output numbers and change input speed accordingly in every frame. However, because the motors keep



```
while ( (abs(newRight) < tick) || (abs(newLeft) < tick) ) {
  newLeft = knobLeft.read();
  newRight = -knobRight.read();

  lDiff = newLeft - preLeft;
  rDiff = newRight - preRight;

  preLeft = newLeft;
  preRight = newRight;

  if (abs(lDiff) > abs(rDiff)) {
    leftSpeed = leftSpeed - offset;
    rightSpeed = rightSpeed + offset;
    Serial.println("left is larger");
  } else if (abs(lDiff) < abs(rDiff)) {
    leftSpeed = leftSpeed + offset;
    rightSpeed = rightSpeed - offset;
    Serial.println("right is larger");
  }

  // turn on motor A
  digitalWrite(In1, LOW);
  digitalWrite(In2, HIGH);
  // set speed to 150 out 255
  analogWrite(EnA, leftSpeed);

  // turn on motor B
  digitalWrite(In3, HIGH);
  digitalWrite(In4, LOW);
  // set speed to 150 out 255
  analogWrite(EnB, rightSpeed);
```

changing their speed in every frame, they can't move in straight lines.

As I mentioned above, in order to create the depth of the drawing, we increased the magnet force and the weight of the sand. The last problem is, the magnet force becomes too strong and it lifts up the robots to the board.



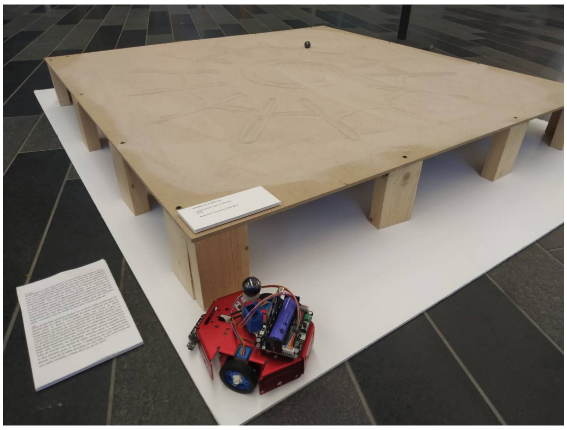
4. THE FINAL UPDATE AND SOLUTION

Materials:

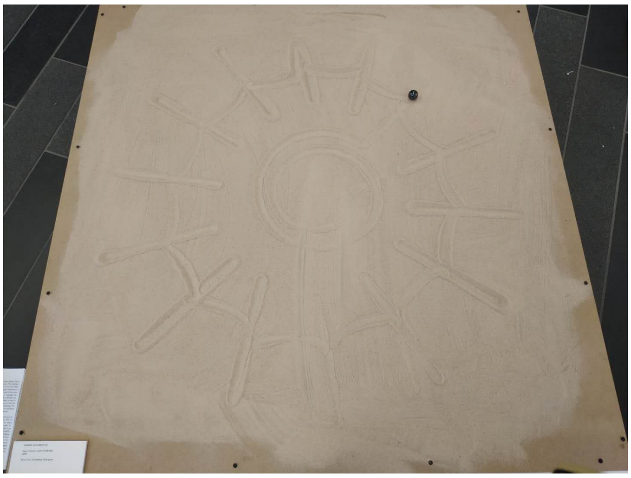
- KittenBot RobotBit v2.0
- Sand 4kg
- 122 * 122 cm sand table with 12mm thick bottom
- Magnet Ball * 2
- Stepper motor * 2
- Wood Support * 16

For the final version, we stepped back to the first version and focused on solving the problem of scale and accuracy.

In order to make the table stronger, we add wood legs to each edge of the board. With the support of the wood legs, the board won't bend and influence the robot's movement. Meanwhile, we reduced the amount of the sand and switched back to the original magnets balls. In order to make the distance between ground and board equal, we also put a board underneath the legs of the sand table.



As for the accuracy, we found that steppers are easier to control, so we switched back to our original stepper motors. We changed the code to improve the visual and also the accuracy.



5. REFLECTIONS

We experienced success and failure during the whole process. Summarized the iterations between the three versions, we realized that it's important to control variables in each iterations. Too many variable changes will make the iteration complex and hard to control. Besides, it's also important to pay attention to the relationships between different variables. The change in one variable sometimes causes another variable to change.

For our future improvement, we are planning to experiment with different ways to draw patterns with depth, so we can test the relationship between water speed and the pattern shapes.

6. FINAL CODE

```
let angle2 = 0
let angle = 0
basic.forever(function () {
    angle = 18
    angle2 = 60
    for (let i = 0; i < 25; i++) {
        robotbit.StepperDual(angle, angle2)
    }
    for (let i = 0; i < 3; i++) {
        robotbit.StepperDual(angle, angle2)
        angle += 1
        angle += 1
    }
    for (let i = 0; i < 30; i++) {
        robotbit.StepperDual(angle, angle2)
    }
    for (let i = 0; i < 2; i++) {
        robotbit.StepperDual(angle, angle2)
        angle += 1
        angle += 1
    }
    for (let i = 0; i < 35; i++) {
        robotbit.StepperDual(angle, angle2)
    }
    robotbit.StpCarTurn(-90, 48, 120)
    robotbit.StpCarMove(15, 48)
    for (let i = 0; i < 15; i++) {
        robotbit.StpCarMove(11, 48)
        robotbit.StpCarMove(-22, 48)
        robotbit.StpCarMove(11, 48)
        robotbit.StpCarTurn(130, 48, 120)
        for (let i = 0; i < 150; i++) {
            robotbit.StepperDual(4, 2)
        }
        robotbit.StpCarTurn(-50, 48, 120)
    }
    robotbit.StpCarTurn(180, 48, 120)
    robotbit.StpCarMove(15, 48)
    robotbit.StpCarTurn(90, 48, 120)
    angle2 = 23
    angle = 65
    for (let i = 0; i < 35; i++) {
        robotbit.StepperDual(angle, angle2)
    }
    for (let i = 0; i < 2; i++) {
        robotbit.StepperDual(angle, angle2)
        angle += -1
        angle += -1
    }
    for (let i = 0; i < 30; i++) {
        robotbit.StepperDual(angle, angle2)
    }
    for (let i = 0; i < 3; i++) {
        robotbit.StepperDual(angle, angle2)
        angle += -1
        angle += -1
    }
    for (let i = 0; i < 25; i++) {
        robotbit.StepperDual(angle, angle2)
    }
    robotbit.StpCarTurn(180, 48, 120)
})
```

7. REFERENCE

Bates, Mary. “What's This Mysterious Circle on the Seafloor?” *National Geographic Society Newsroom*, 15 Nov. 2017, blog.nationalgeographic.org/2013/08/15/whats-this-mysterious-circle-on-the-seafloor/.