

Part 1

My waypoint generation algorithm is the following lines of code:

```
function [W] = makePath(centroid)
    theta1 = deg2rad(30);
    theta2 = deg2rad(60);
    theta3 = deg2rad(90);
    theta4 = deg2rad(120);
    theta5 = deg2rad(150);
    theta6 = deg2rad(180);
    theta7 = deg2rad(210);
    theta8 = deg2rad(240);
    theta9 = deg2rad(270);
    theta10 = deg2rad(300);
    theta11 = deg2rad(330);
    theta12 = deg2rad(360);
    theta13 = deg2rad(390);
    theta14 = deg2rad(420);
    theta15 = deg2rad(450);
    theta16 = deg2rad(480);
    theta17 = deg2rad(510);
    theta18 = deg2rad(540);
    theta19 = deg2rad(570);
    theta20 = deg2rad(600);
    theta21 = deg2rad(630);
    theta22 = deg2rad(660);
    theta23 = deg2rad(690);
    theta24 = deg2rad(710);

    waypoint1 = centroid + [100;0];
    waypoint2 = centroid + rotate(theta1)*[100;0];
    waypoint3 = centroid + rotate(theta2)*[100;0];
    waypoint4 = centroid + rotate(theta3)*[110;0];
    waypoint5 = centroid + rotate(theta4)*[110;0];
    waypoint6 = centroid + rotate(theta5)*[150;0];
    waypoint7 = centroid + rotate(theta6)*[150;0];
    waypoint8 = centroid + rotate(theta7)*[200;0];
    waypoint9 = centroid + rotate(theta8)*[200;0];
    waypoint10 = centroid + rotate(theta9)*[220;0];
    waypoint11 = centroid + rotate(theta10)*[220;0];
    waypoint12 = centroid + rotate(theta11)*[230;0];
    waypoint13 = centroid + rotate(theta12)*[230;0];
    waypoint14 = centroid + rotate(theta13)*[240;0];
    waypoint15 = centroid + rotate(theta14)*[240;0];
    waypoint16 = centroid + rotate(theta15)*[250;0];
    waypoint17 = centroid + rotate(theta16)*[250;0];
    waypoint18 = centroid + rotate(theta17)*[260;0];
    waypoint19 = centroid + rotate(theta18)*[260;0];
    waypoint20 = centroid + rotate(theta19)*[270;0];
    waypoint21 = centroid + rotate(theta20)*[270;0];
    waypoint22 = centroid + rotate(theta21)*[280;0];
    waypoint23 = centroid + rotate(theta22)*[280;0];
    waypoint24 = centroid + rotate(theta23)*[290;0];
    waypoint25 = centroid + rotate(theta24)*[290;0];
    waypoint26 = waypoint1;
```

```

W = [waypoint1, waypoint2, waypoint3, waypoint4, waypoint5,
waypoint6, waypoint7, waypoint8, waypoint9, waypoint10, waypoint11,
waypoint12, waypoint13, waypoint14, waypoint15, waypoint16, waypoint17,
waypoint18, waypoint19, waypoint20, waypoint21, waypoint22, waypoint23,
waypoint24, waypoint25, waypoint26]';
end

```

In this code, I am calculating the position of the drone with respect to the center. The spiral is like a circle with a changing radius. The centroid is the center of the spiral and the “+ rotate” portion is changing the drone’s position with respect to the center. If I increase the angle inside of the rotate function I created (reference below), I can develop a new location that shows where the drone will end up.

```

function V = rotate(theta)
    V = [cos(theta) -sin(theta); sin(theta) cos(theta)];
end

```

The two functions I created work because new angles create a new position. If I integrate new angles in my waypoint vector that will be sent back for plotting that increase linearly (in my case by 30 degrees clockwise – position direction), I can see a uniform spiral pattern. This proves that the algorithm works.

Part 2

For part 2, I can represent the spiral in a simulator. The final result (after integration of the utility functions rotate and makePath) yields:

