* The wheel that is controlling the overall movement (to a particular distance) is essentially going to drive to the distance indicated by the furthest x or y point (farthest distance from the origin, i.e. robots *current* position). The greater the distance in the Cartesian square input grid space, the greater the speed.
* The *turning wheel* is going to vary in speed and polarity in accordance to the *angle* deviation (of the point’s radius) from the x-axis.

|  |
| --- |
|  |
| **Figure x.** Plotting sample points, by splitting the 100x100 square Cartesian grid into fractions, we can deduce the expected output values of the wheels for these points in order to develop a clearer understanding of the procedures involved in deducing the left and right motor outputs. |

From observing the irregular sample points on the example Cartesian grid and using the rules laid out by the prior assumptions of differential wheeled driving, the following relationship becomes apparent:

|  |  |  |
| --- | --- | --- |
| Quadrant of input Coordinate | Left Wheel motor | Right Wheel motor |
| 1 | Controls Distance | Controls Turn |
| 2 | Controls Turn | Controls Distance |
| 3 | Controls Distance | Controls Turn |
| 4 | Controls Turn | Controls Distance |

|  |
| --- |
|  |
| **Table x.** The actual outputs deduced by the algorithm programmatically *proves* the above. A console application was written to test the algorithm, and print it’s inner workings. |

|  |
| --- |
|  |
|  |
| **Figure x, Figure x.** A colour surface plot (top view and angled view, respectively) depicting the value of **magnitude difference** in relation to the value of x and y. Note that gridSize = 100, so that would be the absolute value (min and max) of x, y and the magnitude difference. More importantly, note that the value of magnitude difference tends to 0 as the absolute values of x and y become equal (indicating |φ| = 45°). |

**The *Turn* Function**

Where:

denotes the motor power sent through to the *turn* wheel.

represents the turn-coefficient.

**The *45° Distance* Function**

The aim of this function is to deduce the motor power output for the wheel that determines the distance towards a 45° point associated input coordinate’s quadrant.

Essentially, this distance is determined by the largest absolute value in the input coordinate (either x or y).

***In simple terms:***

Hence ***turn = distance* when robot travelling N or S**, and ***turn = -distance* when robot travelling E or W**.

The **turn-coefficient** tells us the degree to which the input coordinate deviates from the 45° point, and to which direction. The turn-coefficient has a range from +1 to -1, with a turn-coefficient of 0 denoting the point to be at exactly 45° from the origin. A value from 0 to +1 means that the turn is less than a 45° turn, and therefore the turn wheel would have to provide supplementary (same polarity) movement towards the direction of the distance wheel in order to *cut* from the 45° turn that the direction wheel would be making if it were to move *on its own* (this type of turn could be called an “acute” turn, as the turn < 45°). On the other hand, if the value lies between 0 and -1, the turn is greater than 45°, and therefore the turn wheel would have to offset the angle *beyond* the 45° that the direction wheel would achieve on its own, this is accomplished by travelling in the reverse direction to the direction wheel (this type of turn could be termed an “obtuse” turn, as the turn > 45°).

The **magnitude difference** tells us how close or far away the input coordinate lies from the 45° point associated with the quadrant. Ultimately, this gives us the magnitude of the *turn* motor power in order to achieve the correct Cartesian distance.