void adjust() {

//Figure out whether Robot is drifting and in which direction

//Check if it has drifted to right and, if so, how severely

//First check sensor6 to see if it has drifted

if (avg[5] > wMax[5] + DRIFTC) {

//check to see if robot was last seen drifting to right

if (driftState == 1) {

adjC++;

}

//drifted to right, need to run rightmost motor slightly faster and leftmost slightly slower to correct

//analogWrite(enA, BASESPEED + adjC);

//analogWrite(enB, BASESPEED - adjC);

Serial.println();

Serial.print("wMax[5]: ");

Serial.println(wMax[5]);

driftState = 1; //store that we last drifted to the right

}

//Now check to see if it has drifted to left

if (avg[2] > wMax[2] + DRIFTC) {

//check to see whether robot was last seen drifting to left

if (driftState == -1) {

adjC++;

}

//drifted to left, need to run rightmost motor slightly slower and leftmost slightly faster to correct

//analogWrite(enA, BASESPEED - adjC);

//analogWrite(enB, BASESPEED + adjC);

Serial.println();

Serial.print("wMax[2]: ");

Serial.println(wMax[2]);

driftState = -1;

}

else {

driftState = 0;

//this might need to be moved or just straight up removed

adjC--;

}

}

void driftingLeft(){

if (average5to8 > average1to4 + DRIFTC) {

//check to see if robot was last seen drifting to right

if (driftState == 1) {

adjC = adjC++;

}

//drifted to right, need to run rightmost motor slightly faster and leftmost slightly slower to correct

int AMotorval = BASESPEED - adjC; //left motor

int BMotorval = BASESPEED + adjC; //right motor

if (AMotorval < 25) {

AMotorval = 35;

}

if (AMotorval > 100) {

AMotorval = 90;

}

if (BMotorval < 25) {

BMotorval = 35;

}

if (BMotorval > 100) {

BMotorval = 90;

}

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

analogWrite(enA, AMotorval);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enB, BMotorval);

Serial.println("sensors 5 to 8 are bigger so....");

Serial.println("Speed for left motor A: ");

Serial.println(BASESPEED - adjC);

Serial.println("Speed for right motor B: ");

Serial.println(BASESPEED + adjC);

driftState = 1; //store that we last drifted to the right

}

}

void driftingRight(){

if (average1to4 > average5to8 + DRIFTC) {

//check to see if robot was last seen drifting to right

if (driftState == -1) {

adjC = adjC++;

}

//drifted to right, need to run rightmost motor slightly faster and leftmost slightly slower to correct

int AMotorval = BASESPEED + adjC;

int BMotorval = BASESPEED - adjC;

if (AMotorval < 25) {

AMotorval = 35;

}

if (AMotorval > 100) {

AMotorval = 90;

}

if (BMotorval < 25) {

BMotorval = 35;

}

if (BMotorval > 100) {

BMotorval = 90;

}

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

analogWrite(enA, AMotorval);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enB, BMotorval);

Serial.println("sensors 1 to 4 are bigger so....");

Serial.println("Speed for left motor A: ");

Serial.println(AMotorval);

Serial.println("Speed for right motor B: ");

Serial.println(BMotorval);

driftState = -1; //store that we last drifted to the right

} else {

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

analogWrite(enA, BASESPEED);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enB, BASESPEED); //right motor

}

}

void PIDfollow()

{

int position = qtrrc.readLine(sensors);

for (int i = 0; i < 8; i++) {

Serial.print("Value relative to the line for sensor ");

Serial.print(i+1);

Serial.print(": ");

Serial.println(sensors[i]);

}

int error = position - 2500;

int motorSpeed = Kp \* error + Kd \* (error - lastError);

lastError = error;

int rightMotorSpeed = rightBaseSpeed + motorSpeed;

int leftMotorSpeed = leftBaseSpeed - motorSpeed;

if (rightMotorSpeed > rightMaxSpeed ) rightMotorSpeed = rightMaxSpeed; // prevent the motor from going beyond max speed

if (leftMotorSpeed > leftMaxSpeed ) leftMotorSpeed = leftMaxSpeed; // prevent the motor from going beyond max speed

if (rightMotorSpeed < 0) rightMotorSpeed = 0; // keep the motor speed positive

if (leftMotorSpeed < 0) leftMotorSpeed = 0; // keep the motor speed positive

Serial.print("Left motor speed: ");

Serial.println(leftMotorSpeed);

Serial.print("Right motor speed: ");

Serial.println(rightMotorSpeed);

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

analogWrite(enA, leftMotorSpeed);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

analogWrite(enB, rightMotorSpeed);

}