```
/*----------------------------------------------------------------*/
/*    5249Z-Ignite                                                */
/*    Version: 1.2.0                                              */
/*    File: NavMethods.cpp                                        */
/*    Description: Defines methods in NavMethods.h                */
/*----------------------------------------------------------------*/
#include "RobotConfig.h"
#include "RobotMethods.h"
#include "NavMethods.h"
#include "PID.h"

double maxSpeed = 80;
double yawAngle = 0;
double longitude = 0;
const double DIAMETER_WHEEL = 4;
PID longitudePID = PID(10.0/7.0, 0, 2.0/21.0, 0.01);//PID objects created
PID yawPID = PID(15.0/7.0, 0, 6.0/21.0, 0.01);
void resetPosition(){
    yawAngle = 0;
    longitude = 0;
    navInert.setRotation(0, degrees);
    //gyroDrive.setRotation(0, degrees);
}
double getRotation(double distanceHoriz){
    return (360*distanceHoriz)/(M_PI * DIAMETER_WHEEL);
}
void turnToAngle(double angle){
    mtrLeft.resetRotation();
    mtrRight.resetRotation();
    mtrLeftFront.resetRotation();
    mtrRightFront.resetRotation();
    longitude = 0;
    yawAngle = angle;
}
void driveToPos(double distance){
    mtrLeft.resetRotation();
    mtrRight.resetRotation();
    mtrLeftFront.resetRotation();
    mtrRightFront.resetRotation();
    longitude = getRotation(distance);
}
double longitudeError(){
    return (M_PI * DIAMETER_WHEEL)/360.0*(longitude - (mtrLeft.rotation(degrees) +
    mtrRight.rotation(degrees))/2.0);
}
double yawError(){
    return yawAngle - navInert.rotation(degrees);
    //return yawAngle - gyroDrive.angle();
}
int drivePID(){//Maintains set robot position
    while (true){
        longitudePID.setPoint = longitude;
        while (yawError() > 180){
            yawAngle -= 360;
        }
        while (yawError() < -180){
            yawAngle += 360;
        }
        yawPID.setPoint = yawAngle;
        double longitudeCurrent = (mtrLeft.rotation(degrees) +
        mtrRight.rotation(degrees))/2.0;
        double dLongitude = longitudePID.calculatePID(longitudeCurrent);
        double dYaw = yawPID.calculatePID(navInert.rotation(degrees));
        Brain.Screen.printAt(1, 30, true, "Long: %f", longitudeError());
        //Brain.Screen.printAt(1, 60, true, "Yaw: %f", gyroDrive.angle());
        Brain.Screen.printAt(1, 60, true, "Yaw: %f", navInert.rotation(degrees));
        double speedLeft = dLongitude + dYaw;
        double speedRight = dLongitude - dYaw;
        if (speedLeft > maxSpeed){
            speedLeft = maxSpeed;
        }
        if (speedRight > maxSpeed){
            speedRight = maxSpeed;
        }
        if (speedLeft < -maxSpeed){
            speedLeft = -maxSpeed;
        }
        if (speedRight < -maxSpeed){
            speedRight = -maxSpeed;
        }
        chassisLeft(speedLeft);
        chassisRight(speedRight);
        task::sleep(10);
    }
    return 0;
}
```