```cpp
#include "odometry.h"

/*
Odometry::Odometry(ADIEncoder left, ADIEncoder right, ADIEncoder back) {
  this->left = left;
  this->right = right;
  this->back = back;
}

OdomState Odometry::step() {
  int left = left.get();
  int right = right.get();
  int back = back.get();

  int leftchange = left-prevLeft;
  int rightchange = right-prevRight;
  int backchange = back-prevBack;

  double leftDistance = (leftchange / 360) * PI * ODOMWHEELDIM;
  double rightDistance = (rightchange / 360) * PI * ODOMWHEELDIM;
  double backDistance = (backchange / 360) * PI * ODOMWHEELDIM;

  prevLeft = left;
  prevRight = right;
  prevBack = back;

  double currHeading = 90-imu.get_heading();

  double anglediff = currHeading - prevHeading;
  prevHeading = currHeading;

  double localXOffset;
  double localYOffset;

  if (anglediff == 0_deg){
    localXOffset = backDistance;
    localYOffset = rightDistance;
  }
  else {
    localXOffset = 2.0 * sin(angleDiff/2.0) * ( ( backDistance/angleDiff ) + backDistance );
    localYOffset = 2.0 * sin(anglediff/2.0) * ( ( rightDistance/angleDiff ) + rightDistance );
  }
  double averageOrientation = currHeading + (deltaHeading/2.0);
  double r = sqrt( (localXOffset * localXOffset) + (localYOffset * localYOffset) );
  double theta = atan2(localYOffset , localXOffset);
  theta *= (180 / PI);

  theta -= averageOrientation;

  OdomState currState = {prevState.x + (localXOffset*1_in), prevState.y + (localYOffset*1_in),
}

}
*/
```