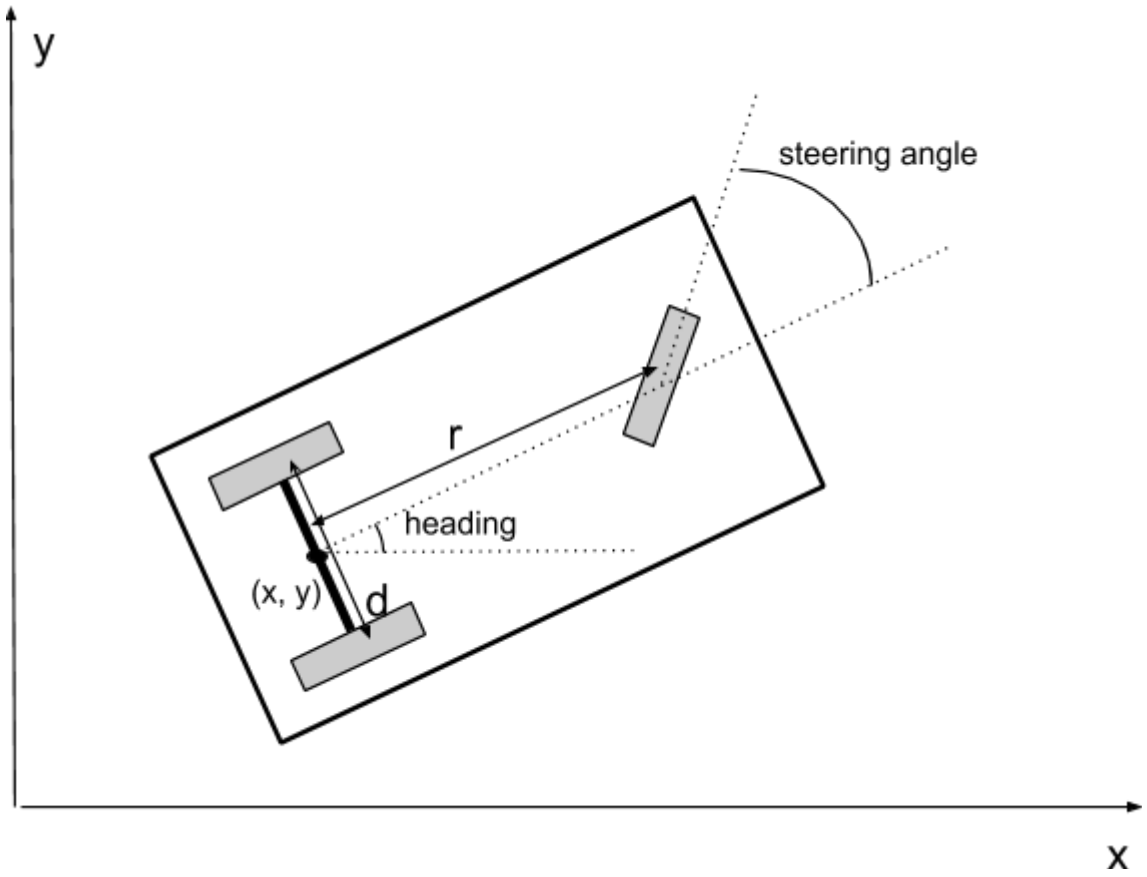
Pose estimator

You have been assigned to write a pose estimator designed to estimate the 2D position of the mobile platform (x, y, heading) based on odometry information using wheel encoder and imu data. The mobile platform is a wheeled tricycle with a steering mechanism that has been attached to the front wheel. **r** is a distance from the front wheel to back axis, **d** is a distance between the rear wheels.



We use a right handed coordinate system, where Z is up, X is forward and Y is to the left of the robot, with positive Yaw (around the z axis) being counter-clockwise and negative yaw being clockwise. The front wheel can rotate up to 90 degrees in both directions around the Z axis. The platform is moving on a 2D-plane (constant z=0). The mobile platform is expected to go straight when the steering angle is 0.

Parameters of the tricycle:

Front wheel radius = 0.2 m
Back wheels radius = 0.2 m
Distance from front wheel to back axis (**r**) = 1m

Distance between rear wheels (**d**) = 0.75m

A traction motor has been attached to the front wheel and is equipped with an encoder with a resolution of 512 ticks per revolution. The steering mechanism also includes an absolute encoder which provides an estimation of the steering angle (in radians).

The pose of the platform is a pose of the center of the rear axis. Initial pose is assumed to be (x, y, heading) = (0, 0, 0).

At each step, the new estimate of the position of the mobile platform is obtained through a call to an **estimate** method with the following API:

estimate (time, steering_angle, encoder_ticks, angular_velocity) **returns** estimated_pose (x, y, heading)

Input	**Description**	**Unit**
time	Time of reading of the input data	sec
steering_angle	Steering wheel angle	rad
encoder_ticks	Number of ticks from the traction motor encoder	Ticks (integer)
angular_velocity	Reading from a gyroscope measuring the rotation velocity of the platform around the Z axis	Rad / s

Output	**Description**	**Unit**
estimated_pose	New estimated pose. Tuple (x, y, heading) representing the estimated pose of the platform.	(m, m, rad)

You are expected to write documentation and test correctness for your code. Sample dataset for checking the code is available in the link below:
https://drive.google.com/open?id=1nkhp7D-udMCWT1J8LSaGG7aqDpJc1tIY

You can either use Python 2.7 and/or C++ 11. For Python, the Numpy library may be used. For C++, only the STL library may be used and your code must compile in a Linux environment (Ubuntu 16.04 preferred) with C++11 standards. Please do not write in C.