```
#ifndef _PPFOLLOWER_
#define _PPFOLLOWER_

#include "main.h"
#include "ports.h"
#include "PurePursuitPathGen.h"
#include <vector>
#include <string>
#include <fstream>
#include <iostream>
#include <utility>
#include <array>


//structure for point storage

struct followPoint {
    double x, y, vel;
};


//experimental pure pursuit follower class
class PurePursuitFollower {
public:
    std::vector<followPoint> points;
    okapi::Timer timer = okapi::Timer();
    double lookahead;
    double last_fractional_index = 0;
    int last_closest_point = 0;
    double prev_vel = 0;
    double prev_left = 0;
    double prev_right = 0;
    double prev_time;
    std::pair<double, double> last_lookahead_point;
    std::pair<double, double> lookahead_point;
    followPoint closest_point;
    double curvature;
    double max_accel = 10.0;
    double prevtime;

    void calc_closest_point(double x, double y);
    void calc_lookahead(double x, double y);
    void read_from_file(std::string filename);
    void calc_curvature_at_point(double x, double y, double theta);
    std::array<double, 4> follow_sim(double x, double y, double theta);
    std::array<double, 4> follow(double x, double y, double theta);
    void read(PurePursuitPathGen obj);
    PurePursuitFollower(double lookahead);
};

#endif
```