Pursuit Curves for Autonomous Vehicle Path Following

Daniel J. B. Clarke

December 8, 2014

1 Introduction

There are many approaches to autonomous vehicle path planning in uncertain environments. One approach is to split the problem into two sub problems; one of planning position and one of planning velocity. [5, 6] This paper deals only with local situations where a path planning algorithm is already known. We use path following to smooth out the vehicle motion turning the complex maximization problem into one of path following.

The planned path should be given as a leading equation of position, velocity pairs of the form:

$$L(t) = \left[\bar{P}(t), V(t)\right] = \left[\bar{P}_0, V_0\right], \left[\bar{P}_1, V_1\right], \ldots, \left[\bar{P}_t, V_t\right] \tag{1}$$

We can look at how the path following algorithm has to change at any moment in time; this results in the following ODE:

$$F'(t) = \left[R_P \times \bar{\lambda}(\bar{L}_P(t), \bar{F}_P(t)), R_V \times \lambda(L_V(t), F_V(t))\right] \tag{2}$$

$$F(0) = F'(0) = [[0, 0], 0]$$

Where R_P and R_V are the constant rates at which we change position and velocity respectively. The initial conditions are [[0, 0], 0] if we consider position and velocity as relative. We assume a value of 0 if ever $L(t) = F(t)$.

This differential equations can be solved for $\bar{F}_P(t)$ and $\bar{F}_V(t)$ in special cases but not general solution exists. They will be analyzed in specific cases during common driving to see what properties are necessary for the leading functions $\bar{L}_P$ and $\bar{L}_V$ and if this is a feasible application for path following.

1.1 Autonomous Vehicle Motion

The derived functions $\bar{F}_P(t)$ and $\bar{F}_V(t)$ are in vector form but could be converted to vehicle properties quite simply. For $\bar{F}_P(t)$:[4]

$$\theta_{wheel}(t) = \arctan\left(\frac{\bar{F}_{P_y}}{\bar{F}_{P_x}}\right) \tag{3}$$

$\bar{F}_V(t)$ may only need to be adjusted by a factor for converting acceleration or break pressure from actual motion adjustment.

Because of the trivial conversion we will neglect converting resulting unit-vectors into $\theta_{wheel}(t)$ or actual acceleration/break pressure throughout this paper.

2 Pursuit Curves

2.1 Elementary Pursuit Curves

A pursuit curve is the curve that's formed when moving with uniform velocity towards another point in uniform velocity. Though the pursuit is a relatively simple process easily modeled by a differential equation, the pursuit curve itself is more complicated and only solvable in certain special cases. It still remains valuable to solve these special cases to speed up processing of the special cases. Though the goal of most pursuit problems is to catch up to the target, our goal is rather to follow the target indefinitely. For this reason we must choose our constant $c = 1$, making the pursuit problem indefinite.[2]

Whether classic or not, a pursuer's curve can be expressed like so:[2]

$$X(t) = x(t) + \lambda x'(t), \quad Y(t) = y(t) + \lambda y'(t) \tag{4}$$

Rewritten we get

$$x'(t) = \frac{X(t) - x(t)}{\lambda}, \quad y'(t) = \frac{Y(t) - y(t)}{\lambda} \tag{5}$$

If we combine x and y to $\bar{F}$ and X and Y to $\bar{L}$ we get...

$$\bar{F}'(t) = \frac{\bar{L}(t) - \bar{F}(t)}{\lambda} \tag{6}$$

It is now clear how we got Equation (2). We need to limit motion by a rate constant R which combines with constant λ, then we need the unit-vector of $\bar{L}(t)$ (defined as $\lambda(\bar{x}) = \frac{\bar{x}}{||\bar{x}||}$) to only attain the direction towards the intended position. With the 1-d vector L_V the unit-vector will naturally come out to either plus or minus 1.

2.2 Autonomous Vehicle Path Following

The ultimate goal of this paper is to apply to an autonomous vehicle. The pursuit strategy seems to be a feasible one when applied to the problem of adjusting vehicle rotation to follow a planned path.[4, 5, 6] Perhaps it is also feasible to apply to the velocity if we think of velocity as a position we can apply fundamentally the same pursuit strategy. One reason this seems like it might be feasible is because position and velocity often go hand-in-hand, and the same type of path smoothing we hope to gain from following position should smooth just as well for velocity. Path planning becomes a problem of selecting position-velocity pairs relative to the vehicle (A job that can be done by the visual apparatus[4]).

Though in this paper we define path following in a way which forms a complete pursuit curve, it may make more sense to continue using the pursuit strategy but to always consider velocities and positions as relative to our current velocity and position.[1, 4, 5]

3 Experimental Results

A program was created which applies our pursuit strategy eliminating the necessity of actually solving non-linear differential equations. We will analyze different common situations in driving to assess the validity of this method and try to solve them mathematically. Mathematical solutions, assuming they are less computationally complex than programatically applying the pursuit strategy, can be substituted for the specific cases for which they are defined.

Note that the figures shown were generated straight from the implementation of $F'(t)$ by python. For simplicity $R_P = R_V = 1$, the paths are crafted by myself but should be generatable by a path planning algorithm.

3.1 Straight Driving

The case of straight driving is a simple case where the position of the vehicle is unchanged while the velocity is such as driving down a road from stop sign to stop sign.

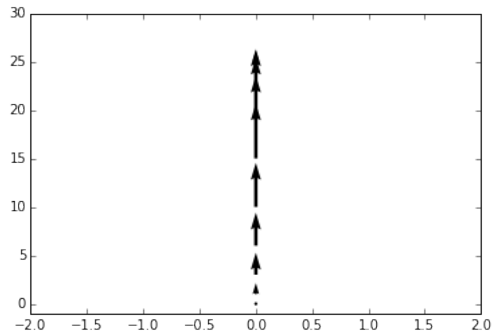

Figure 1: L(t)= [[0, 1], 0], [[0, 1], 5], [[0, 1], 0]

$$F'(t) = [L_P(0), R_V \lambda(L_V(t), F_V(t))]$$

The equation is then simplified to:

$$F'_V(t) = R_V \lambda(L_V(t), F_V(t)) = R_V \frac{L_V(t) - F_V(t)}{||L_V(t) - F_V(t)||}; F(0) = 0$$

$$F'_V(t) = \pm R_V$$

$$F_V(t) = \pm R_V \times t$$

That-is, the velocity can be solved for at any moment in time by multiplying $R_V \times t$. This, though simple means for straight driving we can simplify the algorithm to 1 calculation per step as opposed to $\frac{t}{R_V}$ calculations.

3.2 Lane Changing

The case of lane changing is also relatively simple. It involves maintaining a constant velocity but adjusting the angle of the vehicle such that the velocity brings us into the next lane over.

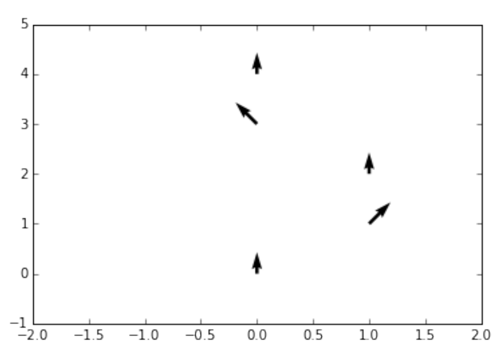

Figure 2: L(t)= [[0, 1], 1], [[1, 1], 1], [[0, 1], 1], [[0, 1], 1], [[-1, 1], 1], [[0, 1], 1]

$$F'(t) = [R_P\lambda(L_P(t), F_P(t)), L_V(0)]$$

The equation is then simplified to:

$$F'_P(t) = R_P\lambda(L_P(t), F_P(t)) = R_P\frac{L_P(t) - F_P(t)}{||L_P(t) - F_P(t)||}; F_y(t) = K, F_x(0) = 0$$

$$F_{P,x} = R_P L_P(t)\int\frac{1}{\sqrt{L_P(t)^2 + F_P(t)^2}} - R_P\int\frac{F_P(t)}{\sqrt{L_P(t)^2 + F_P(t)^2}}$$

$$F_{P,x} = R_P\left[L_P(t)\ln(\sqrt{L_P(t) + F_P(t)^2 + F_P}) - \sqrt{L_P(t)^2 + F_P(t)^2}\right]$$

Already, though it is still possible to attain an integral, the result outweighs the simple algorithmic process of simply stepping through the differential equation.

3.3 Street Intersection

Street intersections are the next big form of motion that are often encountered during an autonomous vehicle route. By simply the pursuit strategy we can complete a right or left turn as before, with a short $L(t)$.

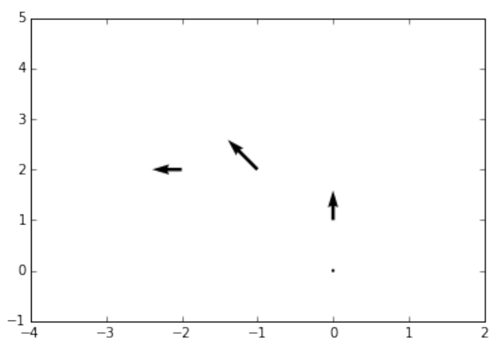

Figure 3:

$$L(t) = [[0, 1], 1], [[-1, 1], 1], [[-1, 0], 1]]$$

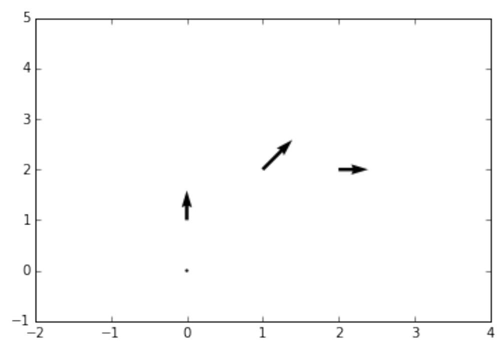

Figure 4:

$$L(t) = [[0, 1], 1], [[1, 1], 1], [[1, 0], 1]]$$

Again:

$$F'(t) = [R_P\lambda(L_P(t), F_P(t)), R_V\lambda(L_V(t), F_V(t))]$$

Assuming the case of a right turn with a constant velocity our results are much like with lane changing but now with both x and y components.

$$F'_P(t) = R_P\lambda(L_P(t), F_P(t)) = R_P\frac{L_P(t) - F_P(t)}{||L_P(t) - F_P(t)||}; F_y(0) = 0, F_x(0) = 0$$

$$F_{P,x} = R_P\left[L_{P,x}(t)\ln(\sqrt{L_{P,x}(t) + F_{P,x}(t)^2 + F_{P,x}}) - \sqrt{L_{P,x}(t)^2 + F_{P,x}(t)^2}\right]$$

$$F_{P,y} = R_P \left[L_{P,y}(t) \ln(\sqrt{L_{P,y}(t) + F_{P,y}(t)^2 + F_{P,y}}) - \sqrt{L_{P,y}(t)^2 + F_{P,y}(t)^2} \right]$$

And again these solutions are much less useful than simply applying a pursuit strategy iteratively.

3.4 Obstacle avoidance

Obstacle avoidance would be handled entirely by the path-planning algorithm. Some sources seem to enter different states when a collision is evident (due to an obstacle).[5, 6] Ours would behave the same as lane changing or any other path. One might expect reaction to be slightly slower because the path planner would have to detect the evident collision and update the path and let the algorithm follow it out of collision. But by increasing R_V the acceleration will increase dramatically getting the following algorithm to avoid swiftly.

This requires that R not be a constant as it is treated in this paper. With R as a constant, our algorithm is helpless in the necessity of a swift change of motion.

4 Conclusion

Though the results look quite decent with the examples I chose, further research should be made to solidify this method. Some things that might be changed include:

1. Adding R_P and R_V finding algorithms, maintaining values of 1 is unreasonable; 1 doesn't allow us to use floating point numbers.

2. Looking into the relation between L_P and L_V to simplify the path further.

3. Making L_P a position rather than an orientation. This would make the path planning algorithm simpler.

4. Developing a better way to check that we've reached a destination (in the program).

5. Computing all values relative to the current position and velocity ([[0, 0], 0]). This should simplify and make more realistic our approach.

Pursuit Curves for Autonomous vehicle path following is certainly a valid approach, particularly for adjusting position as done in other research[4, 5, 6] but also for velocity adjustments. The specific approach taken in this paper has its value but the above mentioned points need to be considered in order for an implementation to be feasible. This type of approach simplifies the path planning algorithm turning it into a problem of placing position-vector pairs in virtual space to be followed and by the nature of the following, the path is smoothed.

References

[1] Carl E Mungan (2005). A classic chase problem solved from a physics perspective. Eur. J. Phys. 26 985. doi:10.1088/0143-0807/26/6/005 Received 6 April 2005, in final form 22 May 2005. Published 8 August 2005. 2005 IOP Publishing Ltd

[2] J. C. Barton and C. J. Eliezer (2000). On pursuit curves. The Journal of the Australian Mathematical Society. Series B. Applied Mathematics, 41, pp 358-371. doi:10.1017/S0334270000011292.

[3] O. Ibidapo-Obe, O.S. Asaolu, A.B. Badiru, Generalized solutions of the pursuit problem in three-dimensional Euclidean space, Applied Mathematics and Computation, Volume 119, Issue 1, 25 March 2001, Pages 35-45, ISSN 0096-3003

[4] Xiang Yin, Noboru Noguchi, Jongmin Choi, Development of a target recognition and following system for a field robot, Computers and Electronics in Agriculture, Volume 98, October 2013, Pages 17-24, ISSN 0168-1699

[5] Xunyu Zhong, Xungao Zhong, Xiafu Peng, Velocity-Change-Space-based dynamic motion planning for mobile robots navigation, Neurocomputing, Volume 143, 2 November 2014, Pages 153-163, ISSN 0925-2312

[6] Xing-Jian Jing, Behavior dynamics based motion planning of mobile robots in uncertain dynamic environments, Robotics and Autonomous Systems, Volume 53, Issue 2, 30 November 2005, Pages 99-123, ISSN 0921-8890