Trajectory Optimization for Car Races using Genetic Algorithms

Extended Abstract

Dana Vrajitoru
Department of Computer and Information Sciences
Indiana University South Bend
1700 Mishawaka Ave, South Bend
IN 46635, USA
dvrajito@iusb.edu

ABSTRACT

In this paper, we present an application of genetic algorithms to a problem of optimizing a car trajectory in a closed loop car race setting. The goal is to minimize the amount of turning that the car needs to do such that it can drive faster, or the total distance that it needs to travel to finish the race. We compare the results with a procedurally computed trajectory and with an optimized version of it using smoothing methods.[1]

CCS CONCEPTS

• **Computing methodologies → Artificial intelligence → Control methods → Motion path planning**

KEYWORDS

trajectory optimization, autonomous cars, car race

1 INTRODUCTION

In this paper, we present an application of genetic algorithms (GAs) to a problem of optimizing the trajectory of an autonomous car in a race setting. Autonomous vehicles have become a critical research topic in the last years, as they are actually available industrially and their use can become commonplace on our roads in the near future. The application in this paper was developed in the TORCS system that simulates car races with multiple tracks available and customizable car controller [1]. We continue the work presented in [2] where a procedurally computed trajectory is used to train a neural network that can take in local road curvature data and produce a target trajectory value. A car controller can direct the car along the trajectory computed this way.

Algorithms for computing optimal paths within road constraints can be found in several papers. In this work, we are using the curve shapes computed in [3], which present similarities with the trajectory presented in [4]. Our work on trajectory calculation is similar to [4] in the problem settings, as the trajectory for the car is computed for a track situation in both cases, with the goal of optimizing the required time. In their approach, however, they pre-compute safe zones for the car on the road and use them to speed up the computation of the trajectory in real time.

Several approaches are present in the literature for track prediction, such as the track segmentation approach, where the track is divided into fragments classified as pre-defined types of polygons [5]. Another controller based on track segmentation is proposed by Onieva et al. in [6]. Their driving controller called AUTOPIA is one of the most successful competitors in the simulated racing car competition. These algorithms are also related to map-matching algorithms such as can be found in [7]. Genetic algorithms have been used in several studies for trajectory optimization, such as in [8, 9], although for a different kind of trajectory.

2 PROCEDURAL TRAJECTORY AND OPTIMIZATION

The TORCS system provides car and road state information that can be used by the car controller to drive the vehicle. Such information includes current speed, current lateral position with respect to the road border, current angle with the road centerline, and current free distance ahead in 19 directions starting from the car's axis direction and scanning left and right in 10^{o}s increments. For the research in this paper, we used the track E-Track5.

For the procedural trajectory, the road is first mapped by having a vehicle drive at constant speed along the track and computing the curvature of the road in each point using the free distance information. Then the road is segmented based on the contiguity of the sign of the curvature, as seen in Figure 1 right. Thus, each segment, marked in purple, represents either an almost flat region, or one where the road curves only to the right, or only to the left. For each segment, an optimal profile, shown in Figure 1 left, is used to calculate a locally optimal trajectory.

A smoothing algorithm can be used to optimize such a trajectory further. This algorithm comprises of a combination of two steps: a curvature descent, moving trajectory points in a direction to reduce the curvature, followed by an anti-aliasing transformation. Figure 2 shows the profiles of the curve obtained by these two algorithms.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07.
https://doi.org/10.1145/3319619.3326792

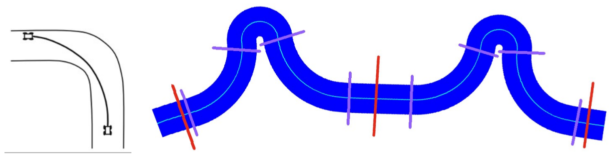

Figure 1: Curvature profile (left) and track segmentation for procedural (purple) and GA (red) optimization.

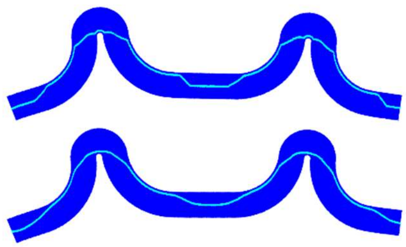

Figure 2: Procedural and optimized trajectories.

3 GENETIC ALGORITHMS OPTIMIZATION

To apply the GAs to this problem, we started by segmenting the road based on almost flat stretches that are sufficiently long, such as the ones marked in red in Figure 1. The trajectory can be computed by the GA on each of the segments, which can also lead to an easily parallel implementation. This segmentation aims to optimize the trajectory even between the smaller segments used by the procedural algorithm and take advantage of possible shortcuts that can be taken.

For each segment, we represent the potential trajectory in the chromosome as a set of control points equally spaced on the road, taking values from -1 for the left border of the road to +1 for the right border, and representing a transversal proportional displacement with respect to the road centreline. Each of these points is represented by the same number of binary genes. Then the actual trajectory is computed by linear interpolation between the control points.

In terms of fitness, preliminary tests using the total curvature of the trajectory on the segment, the maximum curvature, and the total length of the trajectory, have shown the total distance to be the best measure for evolutionary purposes. The goal is to minimize the length, so lower numbers are better.

We performed a number of tests with a varying density of control points. The population size is of 100 and the number of generations is limited by 5000. We used the uniform crossover with a swap probability of 0.3 and an elitist monotonous reproduction. Table 1 presents the results in terms of total length of the trajectory and in terms of total curvature, which is the sine of the curvature angle. The density parameter represents the number of road points for every control point of the trajectory. We compare the GAs with the procedural and optimized trajectories, as well as with a constant trajectory fixed to the centre of the road.

Table 1: GA Trajectory Optimization Results for E-Track 4

Trajectory	Density	Length	Total Curvature
Constant		648.55	12.21
Procedural		595.09	338.32
Optimized		585.85	120.72
GA	5	3179.41	424.91
GA	10	1402.93	296.96
GA	20	857.23	154.83
GA	30	721.70	133.34
GA smoothed	30	643.82	56.31

The last line in the table represents a combination of the GA with the anti-aliasing algorithm used to optimize the procedural trajectory. From the table we can see that larger density numbers, meaning lower numbers of control points, lead the algorithm to a better performance, as it lowers both the trajectory length and the total curvature. The GA-optimized trajectory is better than the procedural one in terms of total curvature for the last three lines.

4 CONCLUSIONS

In this paper, we presented an application of genetic algorithms to a problem of trajectory optimization for autonomous cars. We used a road segmentation approach to represent the trajectory in genetic form and the trajectory length as the fitness function. Our experiments showed that the GA produces comparable and in some cases better results to the procedural approach when combined with an anti-aliasing algorithm. They also showed that using a smaller number of control points is sufficient and can improve the performance.

REFERENCES

[1] B. Wymann, C. Dimitrakakis, A. Sumner, E. Espié, C. Guionneau, and R. Coulom, (2013). *TORCS, The Open Racing Car Simulator*, v1.3. http://www.torcs.org.

[2] D. Vrajitoru (2018). Global to Local for Path Decision using Neural Networks. In *Proceeding of the Pattern Recognition and Artificial Intelligence Conference (PRAI'18)*, ACM International Conference Proceedings Series, August 15-17, Union, NJ, 117-123.

[3] E. Velenis and P. Tsiotras, (2005). Minimum time vs maximum exit velocity path optimization during cornering, In *Proceedings of 2005 IEEE International Symposium on Industrial Electronics*, Dubrovnik, Croatia, 355–360.

[4] A. Liniger and J. Lygeros (2015). A viability approach for fast recursive feasible finite horizon path planning of autonomous RC car. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, (HSCC '15)*, New York, NY, USA, 1–10.

[5] J. Quadflieg, and M. Preuss (2010). Learning the track and planning ahead in a racing car controller. In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG10)*, Copenhagen, Denmark, 395–402.

[6] E. Onieva and D. A. Pelta (2012). An evolutionary tuned driving system for virtual racing car games: The AUTOPIA driver. *International Journal of Intelligent Systems*, 27(3), 217–241.

[7] S. S. Rathour, A. Boyali, L. Zheming, S. Mita, and V. John (2017). A Map-based Lateral and Longitudinal DGPS/DR Bias Estimation Method for Autonomous Driving. International Journal of Machine Learning and Computing, Vol. 7, No. 4, August 2017, 67-71.

[8] M. Schlueter and M. Munetomo (2018). Massively parallelized co-evaluation for many-objective space trajectory optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'18),* Kyoto, Japan, 306-307.

[9] N. Padhye (2008). Interplanetary trajectory optimization with swing-bys using evolutionary multi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08),* Atlanta, GA, USA, 1835-1838.