Expert Systems with Applications 40 (2013) 6447–6458

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Advanced overtaking behaviors for blocking opponents in racing games using a fuzzy architecture

Luigi Cardamone [a], Pier Luca Lanzi [a,*], Daniele Loiacono [a], Enrique Onieva [b]

[a] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy
[b] Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain

ARTICLE INFO

Keywords:
Fuzzy systems
Simulated car racing
Overtake
Video games
Games
Artificial intelligence
Computational intelligence

ABSTRACT

In car racing, *blocking* refers to maneuvers that can prevent, disturb or completely block an overtaking action by an incoming car. In this paper, we present an advanced overtaking behavior that is able to deal with opponents implementing blocking strategies of various difficulty level. The behavior we developed has been integrated in an existing fuzzy-based architecture for driving simulated cars and tested using The Open Car Racing Simulator (TORCS). We compared a driver implementing our overtaking strategy against four of the bots available in the TORCS distribution and *Simplix*, a state-of-the-art driver which won several competitions. The comparison was carried out against opponents implementing three blocking strategies of increasing difficulty and two different scenarios: (i) a basic scenario with one opponent on a straight stretch to overtake as quickly as possible; (ii) an advanced scenario involving a race on a non-trivial track against several opponents. The results from the basic scenario show that our strategy can *always* overtake the opponent car; in particular, our strategy is slightly more risky than the other ones and may result in a little damage, however, all the other controllers show a more careful and safe policy that often prevents them to complete an overtaking maneuver. When racing against several opponents on complex tracks, our strategy results in the best trade-off between the time spent being blocked by an opponent ahead and the number of overtaking maneuvers completed.

1. Introduction

Overtaking plays a key role in the artificial intelligence (AI) of racing games both in terms of performance, believability and players' satisfaction (Cardamone, 2012; Cardamone, Loiacono, & Lanzi, 2009c). Alas, overtaking behaviors are very difficult to code (even in high-end games Lecchi, 2009) as they have to manage a large number of variables (e.g., the current trajectory, position, speed and acceleration of the driver's car and the nearby opponents, incoming bends, etc.,) and a variety of unexpected, challenging, and dangerous situations (e.g., change in opponents' trajectory, position, and blocking maneuvers, etc.).

Blocking maneuvers are strategies that a driver can use to prevent, disturb or possibly block an overtaking action by an incoming car and they are frequently employed in racing games both by human players and non-playing characters. They are typically performed by adapting the vehicle trajectory so as to block the racing line chosen by an incoming vehicle. Blocking maneuvers are typically simple to implement and can block even the most advanced overtaking strategies. A preliminary analysis we performed (Onieva, Cardamone, Loiacono, & Lanzi, 2010) on seven of the most competitive drivers available for TORCS shows that while most of these competitive drivers implement reliable overtaking strategies, their performance is dramatically reduced when facing opponents implementing even simple blocking strategies. Even the most advanced human-coded drivers (e.g., the winner of the 2009 TORCS Endurance World Championship) fails in dealing with basic blocking strategies.

In this paper, we present an advanced overtaking behavior, implemented using a fuzzy system, that can overtake challenging opponents using blocking strategies of increasing difficulty. We evaluated our overtaking behavior using two different scenarios: (i) a basic scenario with one opponent on a straight stretch to overtake as quickly as possible; (ii) an advanced scenario involving a race on a non-trivial track against several opponents. We considered opponents implementing three blocking strategies: (i) *limited blocking*, which adapts to the opponent's trajectory but avoids going too near to the track borders (so that the incoming car has still some chance of completing the overtaking maneuver); (ii) *slowly reactive blocking*, which adapts to the opponent's trajectory with a delay of one second but has no limitation and can completely block the overtaking maneuver; (iii) *fully reactive blocking*, which adapts to the opponent's trajectory with no delay and no limitation. This work extends our preliminary study (Onieva

* Corresponding author. Tel.: +39 0223993472; fax: +39 0223993411.
E-mail addresses: cardamoneluigi@gmail.com (L. Cardamone), pierluca.lanzi@polimi.it (P.L. Lanzi), daniele.loiacono@polimi.it (D. Loiacono), enrique.onieva@deusto.es (E. Onieva).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.04.030

et al., 2010) where we presented a much simpler driver that could only drive and overtake one opponent on a straight stretch. We compared our driver using our advanced overtaking behavior against five of the best drivers available for The Open Car Racing Simulator TORCS. Our results show that in the simpler scenarios our strategy can *always* overtake the opponent car; in particular, our strategy appears to be slightly more risky and may cause a little damage to the car while achieving 100% success; in contrast, more careful handcoded controllers demonstrate a safer policy that cause no damage to the car but too often do not complete any maneuver. Interestingly, our driver demonstrates an interesting emerging behavior in which it deceives the blocking opponent by initially moving to one side of the track (and thus making the opponent move on the same side) before overtaking on the opposite side. Our results in the more complex racing scenarios show that our strategy provides the best trade-off between the time spent being blocked by an opponent ahead and the number of overtaking maneuvers completed.

2. Related work

Car racing games are a popular testbed for methods of computational intelligence (Butz & Lonneker, 2009; Cardamone, Loiacono, & Lanzi, 2009a, 2009b, 2009c; Ebner & Tiede, 2009; van Hoorn, Togelius, Wierstra, & Schmidhuber, 2009; Munoz, Gutierrez, & Sanchis, 2009; Onieva, Pelta, Alonso, Milanes, & Perez, 2009; Perez, Recio, & Saez, 2009; Fujii, Nakashima, & Ishibuchi, 2008; Ho & Garibaldi, 2008a; Loiacono et al., 2008). This is probably due both to the availability of several open-source frameworks with realistic physics and engaging graphics (TORCS; Venzon; Wolf-Dieter et al.) and also to the many (10–20) competitions on simulated car racing orgazined since 2007 at major conferences (e.g., IEEE CEC 2008–2010, ACM GECCO 2009–2012, IEEE CIG 2009–2012, EvoStar 2011–2012, Loiacono et al., 2008, 2010b). Most of the works published in this area focus either (i) on the development of complete drivers using a wide variety of methods (e.g., neural networks Cardamone et al., 2009a, 2009c, fuzzy logic Onieva et al., 2009; Perez et al., 2009; Fujii et al., 2008; Ho & Garibaldi, 2008a, 2008b, evolutionary algorithms Ebner & Tiede, 2009, supervised learning Cardamone, Loiacono, & Lanzi, 2009b; van Hoorn et al., 2009; Munoz et al., 2009; Quadflieg, Preuss, Kramer, & Rudolph, 2010); or (ii) on the parameter optimization of human-designed bots (Butz & Lonneker, 2009; Butz, Linhardt, & Lonneker, 2011; Quadflieg, Preuss, & Rudolph, 2011; Preuss, Quadflieg, & Rudolph, 2011) and car setup (Wloch & Bentley, 2004; Cardamone, Loiacono, & Lanzi, 2010; Kemmerling & Preuss, 2010). In this section, we provide a brief overview of the published works related to this study.

2.1. Fuzzy systems for car racing

Fuzzy systems have been seldom used in car racing games. Onieva et al. (2009) developed a modular architecture in which the general driving was implemented by a fuzzy system controlling the target speed; the overtaking behavior was implemented by a separate heuristics that modified the target speed when opponents were detected. In Ho and Garibaldi (2008b), introduce the concept of Context-Dependent fuzzy system, in which the membership functions of the fuzzy variables are not fixed but change according to the context. The proposed approach is applied to design a controller for the car racing competition held at FuzzIEEE 2007. In Ho and Garibaldi (2008a), the same authors, present an improved version of the controller for the 2007 CIG Simulated Car Racing Competition. The driver has a two-layers architecture that combines a high-level path planner with a low-level execution controller based on fuzzy logic. In Fujii et al. (2008), fuzzy rules are generated from a set of training patterns; the study compares two methods for generating such training patterns and two representations of the sensory information (third person vs. egocentric). In Perez et al. (2009), present a driver based on a fuzzy controller for the 2008 CIG Simulated Car Racing Competition. First, they designed the rules and the fuzzy sets of a base driver. Then, they applied a genetic algorithm to optimize the parameters of the fuzzy sets. In Onieva et al. (2010), we presented an initial study of blocking in car racing games based on our experience in the organization of the Simulated Car Racing Competition (Loiacono et al., 2008, 2010b); we showed that even the most competitive controller can fail to overtake even the most basic blocking strategies on very simple straight track sections; we also showed that a simple fuzzy controller could tackle blocking behaviors that more advance drivers failed to manage.

2.2. Overtaking behaviors in car racing games

Although overtaking strategies are known to play a key role in the development of competitive drivers (Cardamone et al., 2009c; Butz & Lonneker, 2009; Butz et al., 2011), few published works focus on this topic.

Cardamone et al. (2009c) applied neuroevolution (more precisely NEAT Stanley, 2004) to evolve a competitive driver capable of overtaking in complex situations. The architecture comprised one neural network for the driving alone and one neural network for overtaking. The two networks were evolved separately on different scenarios. The network for overtaking was activated on top of the main driving behavior (in a sort of behavior-based architecture) when an opponent close to the car was detected. Loiacono, Prete, Lanzi, and Cardamone (2010a) applied simple reinforcement learning to learn separate strategies for (i) overtaking an opponent on a straight stretch by exploiting the drag effect; and for (ii) overtaking an opponent close to a turn using braking delay. Butz et al. (2011) developed a driver for TORCS based on a sensory-to-motor policy, optimized using an evolutionary strategy with Covariance Matrix Adaptation (CMA-ES). Overtaking was implemented as a module that (i) monitored the opponents position and speed around the driver and (ii) projected the opponents position onto the driver's track sensors. Accordingly, the opponents were perceived as obstacles on the track and the overtaking actions were performed by the basic driving module.

3. TORCS

The Open Racing Car Simulator (TORCS) is a state-of-the-art open source car racing simulator which provides a sophisticated physics engine, full 3D visualization (see Fig. 1), several tracks and models of cars, and different game modes (e.g., practice, quick race, championship, etc.). The car dynamics is simulated by a powerful physics engine that takes into account many aspects of racing cars such as traction, aerodynamics, fuel consumption, etc. Each car is controlled by an automated driver or *bot*. At each control step, a bot can access the current game state, which includes several information about the car and the track as well the information about the other cars on the track, and can control the car using the gas/brake pedals, the gear stick, and steering wheel. The game distribution includes several programmed bots which can be easily customized or extended to build new bots.

4. Overtaking and blocking behavior

Driving a racing car on a track alone is a task easy to program and to learn using computational intelligence methods. In contrast, driving a racing car against other opponents is very complex be-