# On the Generality of the Evolved Driving Rules of an Agent Operating a Model of a Car

Hirotaka Yamazaki, Ivan Tanev, Tomoyuki Hiroyasu and Katsunori Shimohara

*Abstract*—We present an approach for automated evolutionary design of the functionary of driving agent, able to operate a software model of fast running car. The objective of our work is to automatically discover a set of driving rules (if existent) that are general enough to be able to adequately control the car in all sections of predefined circuits. In order to evolve an agent with such capabilities, we propose an indirect, generative representation of the driving rules as algebraic functions of the features of the current surroundings of the car. These functions, when evaluated for the current surrounding of the car yield concrete values of the main attributes of the driving style (e.g., straight line velocity, turning velocity, etc.), applied by the agent in the currently negotiated section of the circuit. Experimental results verify both the very existence of the general driving rules and the ability of the employed genetic programming framework to automatically discover them. The evolved driving rules offer a favorable generality, in that a single rule can be successfully applied (i) not only for all the section of a particular circuit, but also (ii) for the sections in several a priori defined circuits featuring different characteristics.

## I. INTRODUCTION

THE success of the computer playing sport games (e.g., chess [7], checkers, backgammon, tic-tac-toe [2], etc.) has long served as touchstone of the progress in the field of artificial intelligence (AI). The expanding scope of applicability of AI for the implementation of an agent with autonomous "learning" abilities includes soccer [9] and F1 racing [15] etc. [4], etc. Focusing in the domain of computerized car racing, in this work we consider the problem of the automated design of driving agent, able to operate a model of a fast running car. We are motivated by the opportunity to develop a driving agent, able to address some of the challenges that human drivers face. In order to drive a car fast around the circuit, the driver needs to define the best driving line and the way of approaching turns in the circuit. To realize the optimal line, the driver (agent) is also required to make precise sense and judgment about the state (i.e., position, orientation and velocity) of the car and its surrounding (distance to the apex of the turn, apex homing angle, etc.) and to react timely and precisely.

The *objective* of our work is to design the functionality of a driving agent, able to control a fast running software model of the remotely controlled scaled racing car in various circuits with diverse characteristics. The agent should be able to learn the most general rules about how to control the fast running car not only on a particular circuit, but also on several a priori defined circuits. The possibility to discover such general driving rules (if existent) implies that the agent, when trained on a set of circuits with diverse enough characteristics, would be able to optimally control the car in unknown circuits with the need of a little (if any) adaptation. The very feasibility of developing such a universal agent would open up opportunities to build a framework of adaptive racing games in which the human competes against the computerized racer in dynamic or a priori unknown (e.g., user-defined) tracks and dynamically changeable (moving cars, obstacles, etc.) environments. The proposed approach could also be applied for automated design of the control software for navigation of remotely operated vehicles capable to find an optimal solution to various tasks in uncertain and dynamic environment.

Achieving our objective implies that the following four tasks should be addressed: (i) formalizing the driving style by defining the set of its *key parameters*, (ii) defining the notion of *general driving rules* that, when applied in the context of the concrete features of the current surrounding of the car, would yield concrete values of the key parameters of the driving style, (iii) defining the set of relevant *features of the surrounding*, that would facilitate the creation of general driving rules, and (iv) developing an *algorithm paradigm* for automated determination of the fastest, yet general driving rules, which yield concrete, optimal values of the parameters of the driving style when applied for the currently negotiated section of the circuit.

The related work by Suzuki and Floreano [10] demonstrates the feasibility of applying an active vision for landmark navigation of a scale vehicle. Wloch and Bentley [15] applied genetic algorithms for automated optimization of the setup of the simulated racing car. However, neither the adaptation of the driving style to the setup of the car nor the generality of the evolved setup was considered. Togelius and Lucas [14] used scale models of cars in their research to demonstrate the ability of the artificial evolution to develop optimal neurocontrollers with various architectures. However, the effects of the inherent latencies in the video feedback of these controllers on either the precision or consistence were beyond the scope of their work. The work of Tanev et al [13] demonstrates that with addressing the problem of video feedback latency through an anticipatory modeling a human competitive driving agent could be automatically evolved via genetic algorithms. The driving agent, presented in their work is able to achieve lap times that are both faster and more consistent than those of a well-trained human operator. However, their work considers an agent obtained as a result of time-consuming simulated evolution on a single, a priori

Hirotaka Yamazaki and Tomoyuki Hiroyasu are with the Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto, Japan (e-mail: hyamazaki@mikilab.doshisha.ac.jp, tomo@is.doshisha.ac.jp)

Ivan Tanev and Katsunori Shimohara are with the Department of Information Systems Design, Doshisha University, Kyoto, Japan (e-mail: {itanev, kshimoha}@mail.doshisha.ac.jp)

known circuit. The question about how well such an agent would perform in unknown circuits is considered as beyond the scope of their work.

Moreover, employing genetic algorithms (GA) in our previous work [12], we attempted to determine the optimal, but fixed and concrete values of the attributes of driving style (straight line velocity, turning velocity, apex approach angle, etc.), pertaining to each section of the circuit. This approach seems to be unfeasible for generalization across the various sections of the same circuit. Therefore, even a slight modification to the circuit would require additional evolutionary runs to define the new set of optimal values of the driving style parameters. Moreover, the generality of such an approach across different circuits could not be considered at all, at least because the newly introduced circuits might feature sections with completely different characteristics. In such case the once evolved optimal values of the set of driving style parameters pertaining to each of the sections in the previously considered circuit cannot be reused and the evolutionary optimization of the values of driving style parameters on newly introduced circuit should, most probably, start from scratch. The scalability of the approach of optimizing the concrete values of driving style parameters pertaining to each turn of the circuit is the last (but not the least important) concern of applying GA, as the size of the search space of evolutionary optimization increases exponentially with the linear increase of the number of corners in the circuit.

In our current work we attempt to address the problem of generality of the optimized driving agent by proposing an evolutionary framework, able to develop optimal, yet general driving "rules", rather than concrete optimal values of the driving style parameters. We shall verify the generality of applying the evolved single set of rules both for all the sections of a single circuit, and for all the sections of several a priori known circuits with different characteristics.

The remaining of this document is organized as follows. Section 2 explains the modeled configuration of the physical system. Section 3 elaborates on the formalizing the driving style and introduces the notion of driving rules. Section 4 explains the main attributes of the algorithm paradigm employed for evolution of the fastest, yet general driving rules. Section 5 presents the experimental results, and Section 6 draws a conclusion.

## II. MODELED CONFIGURATION OF THE PHYSICAL SYSTEM

### A. System Configuration

In our work we employ a software model of the scaled (1/24) racing car. The main mechanical characteristics of the car are shown in Table 1. Characteristics #6-#9 are experimentally obtained from the physical car running on the considered track surface - a polyvinyl chloride carpet with coefficients of static and kinematic friction between the rubber tires and the surface $\mu_S$=0.8 and $\mu_K$=0.7, respectively.

The perceptions of the agent are modeling the real physical system on which an image feed is obtained from a video camera mounted overhead (Figure 1). The camera features a

CCD sensor and lenses with wide field of view (66 degrees), which allows to cover a sufficiently wide area of about 2800mm x 2100mm from an altitude of about 2200mm. The camera is modeled to operate at 640x480 pixels mode, scaling down the scene to about 4mm per pixel. The video sampling interval is 30ms, and the video feed latency is 90ms.
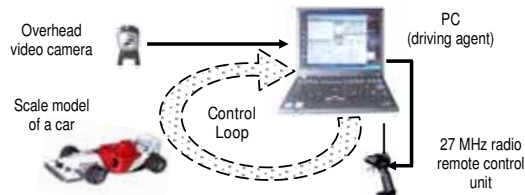


Fig. 1. Modeled configuration of the physical system

The car is operated by the agent via modeled ratio remote control with functionality including "forward", "reverse" and "neutral" throttle control commands and "left", "right" and "straight" steering controls.

TABLE 1
MECHANICAL CHARACTERISTICS OF THE CAR

| Parameter | Value |
|---|---|
| 1) Car: model and scale | Auldey F1, 1:24 |
| 2) Dimensions (l x w x h), mm | 200 x 86 x 54 |
| 3) Wheelbase, mm | 130 |
| 4) Steering angle, degrees | 30 |
| 5) Mass, g | 310 |
| 6) Max straight line velocity, mm/s   (scaled, km/h) | 2000  (172) |
| 7) Acceleration on full throttle, mm/s² | 800 |
| 8) Deceleration on reverse, mm/s² | -2000 |
| 9) Deceleration due to the mechanical drag on throttle lift-off, mm/s² | -600 |

### B. Software Simulator

The developed software simulator of the car and the environment allows to (i) "compress" the runtime of the fitness evaluation in the eventual implementation of agent's evolution and (ii) to verify the very feasibility of certain circuit configurations without the need to be concerned about the risks of possible damage to the environment or the car (or both) [12]. Furthermore, the internal model of the car and the environment comprises the kernel of the developed simulator. This model is continuously applied by the driving agent in order to anticipate the intrinsic, yet unperceivable state of the car from the currently available (outdated) perceptions. The model, calibrated with the concrete values of car's parameters (as summarized in Table 1) takes into consideration (i) the

Newtonian physics of the car including the handling attitude of car on cornering (neutral steering, oversteer and understeer), (ii) a random uniform noise of +/-1 pixel (equal to the experimentally obtained value) incorporated into the "tracking" of the car and (iii) the feedback latency of 90ms.

The handling attitude of the physical car on cornering are defined by the relation of the sliding angles (if any) of the tires of the front and rear axles of the car. The tires of the physical car on turning, operated at, or beyond the limits of the grip (friction, adhesion) forces, slide to some degree across the intended direction of traveling. The dynamic weight redistribution causes the grip levels at the front and rear wheels to vary as the turning car accelerates on "forward" or decelerates on either "neutral" or "reverse" throttle commands [5]. This, in turn, yields different sliding angles for the front and rear wheels, causing the car that tries to turns too fast to feature either a neutral steering (the slide angles of both axles assume the same values, and the car turns with a nominal or slightly smaller turning radius, as shown in Figure 2a), an understeer (slide angle of the front wheels are greater than those of the rears - the car turns with a wider turning radius, as shown in Figure 2b) or oversteer (slide angle of the front wheels are narrower than that of the rear ones- the car turns with a narrower turning radius, as depicted in Figure 2c).
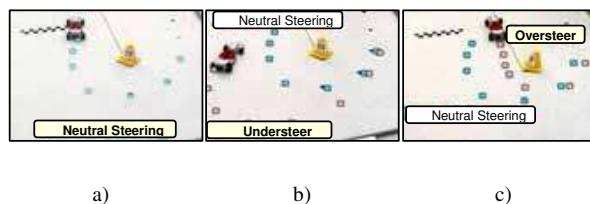


a)　　　　　　b)　　　　　　c)

Fig. 2. The three possible handling attitudes of the sliding car on cornering: neutral steering – the sliding angles of the wheels of both the front and rear axle are nearly the same (a), understeer – the sliding angles of the front wheels are greater than those of the rear wheels (b) and oversteer - the sliding angles of the front wheels are narrower than those of the rear (c)

In addition to the degradation of the maneuverability of the car, the sliding of the wheels results in a significant braking momentum which, in turn, reduces the velocity of the car. Moreover, the increased actual turning radius due to sliding of the understeering car means that the car might enter the run-off areas of the track or even hit the guardrails on tight corners of the track, which, in turn, might result either in a damage of the car, lost of momentum, or both. Therefore, the sliding of the understeering car limits the average velocity on cornering car (due to the lower than intended speeds along longer than intended arcs), which may have a detrimental effect on the overall lap times. The complexity of the effects of handling attitudes of the car on the lap time renders the task of optimizing the driving style of the agent quite challenging, which, in turn, additionally motivated us to consider an automated evolutionary approach to address it.

The realism of the software model of the car is proved by the very limited need of adaptation to the physical system of the driving agent, initially evolved in the software model [12]. Also, the realism is indirectly indicated by the consistence of both the obtained driving lines and the lap times of the physical car around predefined circuits [13], which is attributed to the negligibly small difference between the intrinsic, but unperceivable state of the car (i.e., position, orientation, and velocity) and the state, anticipated from the outdated perceptions. The precision of such anticipation can be seen as an evidence of the adequacy of the applied software model of the car.

### III. REPRESENTATION OF DRIVING RULES

Achieving our objective of designing a general driving agent, which is able to control the model of car in various circuits with diverse characteristics, implies that we shall address the following four tasks:
- How to formalize the driving style by defining the set of its key parameters,
- How to define the notion of general driving rules. The rules, when applied in the context of the current surroundings of the car, would yield an optimal, concrete values of the key parameters of the driving style,
- How to define the set of relevant features of the current surrounding of the car, that would facilitate the creation of general driving rules, and
- What algorithm paradigm could be used for automated evolution of fastest, yet general driving rules, which yield concrete, optimal values of the driving style parameters when applied for the currently negotiated section of the circuit?

In this Section we elaborate on the former three tasks. The following Section 4 we discusses the latter task.

#### A. Parameterization of the Driving Style

We consider the driving style as the driving line, which the car follows before and around the corners in the circuits, combined with the breakdown of the speed, at which the car travels along this line. Our choice of parameters of the driving style is based on the view, shared among the high-performance drivers from various teams in different racing formulas, that (i) the track can be seen as a set of consequent turns they need to optimize divided by simple straights, and that (ii) the turns with the preceding straights should be treated as a single whole [1][3]. Based on these standpoints, we introduce the following set of four key parameters of the driving style, pertaining to each of the turns of the circuit:
- Approach (homing) angle – the constant bearing of the apex of the turn. Higher values of the latter parameter yield wider driving lines featuring higher turning radiuses,
- Straight-line velocity - the velocity at which the car approaches the turn,
- Turning velocity – the velocity inside the turn, and

- Throttle lift-off zone – the distance from the apex at which the car begins slowing down from the straight-line velocity to the turning velocity.

Notice that the first parameter solely defines the driving line, and the latter three parameters define the breakdown of the velocity along this line.

Viewing the desired values of these four parameters as values that the agent tries to maintain, the functionality of the agent can be algorithmically formalized in a way as shown in Figure 3. The usage of the values of the key driving style attributes are underlined in the figure and indicated as "desired". As Figure 3 illustrates, both the orientation (lines 7-10) and the speed (lines 12-15) of the car are continuously adjusted in order to match the desired values of the corresponding attributes. The open-loop adjustment of the car's velocity (lines 13 and 14) is implemented by macro-commands `ShiftGear(Gear)`, implemented via pulse-width modulation (PWM) of the sequence of "forward" (the motor is turned on) and "neutral" (motor is off) throttle commands with duty cycle of 120ms (4 sampling intervals). The possible values of the input parameter Gear (and, consequently, the possible values of the driving style parameters "Straight-line velocity" and "Turning velocity") are 1, 2, 3 or 4, which correspond to the duty ratios (i.e., the proportion of time during which the motor is turned on as a result of "forward" throttle command) of PWM of 0.25, 0.5, 0.75, and 1 respectively.

---

1. At each time step do begin
2. *//-- Perceptions:*
3. Obtain the agent's perceptions of car's state: position (P), orientation and speed (V);
4. Obtain the agent's perceptions of the environment: approach angle ($A_A$), and distance ($A_D$) to the current apex
5. *//-- Reaction of the agent to the current perceptions*
6. *//-- A) Steering control:*
7. if ($A_A$> <u>Desired $A_A$</u>) and (abs($A_A$ - <u>Desired $A_A$</u>)> <u>Desired Threshold $A_A$</u>)
8.   then SetSteering(Left)
9.   else if ($A_A$< <u>Desired $A_A$</u>) and (abs($A_A$ - <u>Desired $A_A$</u>)> <u>Desired Threshold $A_A$</u>)
10.       then SetSteering(Right)   else SetSteering(Straight);
11. *//-- B) Throttle control:*
12. if $A_D$ > <u>Desired Throttle Lift-off Zone</u>
13.   then ShiftGear (<u>Desired Straight Line Gear</u>)
14.   else ShiftGear (<u>Desired Turning Gear</u>);
15. end

Fig. 3. Functionality of driving agent

---

Compared to the typical approaches of polynomial (spline) interpolation of curves, the use of the single parameter "Approach angle" to solely define the driving line significantly simplifies the representation of the driving style which, in turn reduces the search space of the simulated evolution which is intend to employed for automated optimization of driving rules. Fixing the homing angle of the car implies that the driving line is interpolated as an equiangular spiral. The proposed approach is inspired by the smooth decrease of the turning radius (and the corresponding optimal cornering velocity) along the spiral, which gives the agent an opportunity to smoothly, and progressively apply both the steering and the brakes as the car negotiates the turn. Moreover, "the smooth is fast" is among the most important rules in high-performing driving as the smoothness of both the radial (due to steering) and lateral (due to acceleration or braking) forces allows for a better control of the car near the traction limits of the tires [1]. The layouts of the roads in some highway junctions which can be approximated by equiangular spirals (as illustrated in Figure 4) could be viewed as a verification of the technical plausibility of the proposed an approach [6].

From another perspective, the proposed use of the approach angle to solely define the trajectory of the car is biologically plausible in that the moving patterns of some birds (pigeons, penguins, waterfowl, etc.) and insects (bees, ants, etc.) are derived by moving at a constant angle relative to the landmarks (sun, moon, stars, etc.). Animals which leave their nest moving at a fixed angle with respect to a landmark needs to only reverse that angle for their return (light compass orientation).
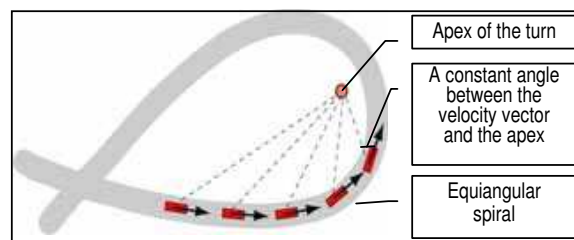


Fig. 4. A sample layout of a road, approximated by an equiangular spiral [6]

### B. Representation of the Driving Rules

The driving rule in the proposed approach defines how to compute the four key parameters of the driving style (as elaborated in the previous Section 3.1) for the currently perceived surrounding of the car. We implement the driving rule as a set of four evolvable algebraic functions. Each one of the four parameters of the driving style is associated with exactly one algebraic function and the value of the corresponding parameter is set to the value of the function, evaluated for the concrete features of the currently negotiated section of the circuit. Therefore, the same driving rule might potentially yield different values of the four driving style parameters when evaluated in different sections of the circuit. The relevant features of the currently negotiated section of the circuit are obtained from the features of the waypoint (or turn) at which the car is currently homing (Figure 5): (i) the length of the vector from the previous to the current waypoints dist_p_c, (ii) the length of the vector from the current to the next waypoints dist_c_n, and (iii) the angle between these two vectors angle_p_c_n.

### IV. EVOLUTION OF DRIVING RULES

We assume that the four key parameters of the optimal driving style around different turns of a circuit might feature

different values, and that these values should be evaluated from a single driving rule comprising four algebraic functions (expressions). Therefore, the objective of automatic design of optimal driving rules can be rephrased as an automatic discovery of such a set of four expressions that yield optimal values of the four driving style parameters for each of the turns in the circuit. In this section we elaborate on the main attributes of the genetic programming (GP) framework, employed for automated discovery of the general driving rules [8] [11].
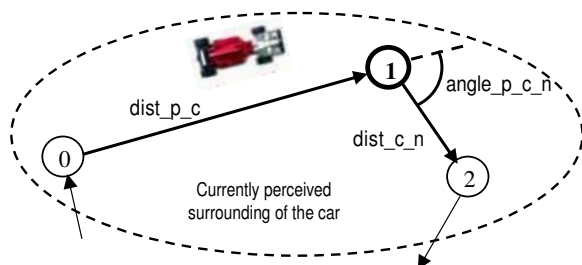


Fig. 5. Relevant features of the currently negotiated section of the circuit comprising three waypoints #0, #1 and #2: the length of the vector from the previous to the current waypoints (dist_p_c), the length of the vector from the current to the next waypoints (dist_c_n), and the angle between these two vectors (angle_p_c_n). The car is currently homing at waypoint #1.

The genotype in the proposed GP encodes for the evolving set of four algebraic expressions corresponding to the four parameters of driving style. The function set of GP comprises an addition, subtraction, multiplication and protected division. The terminal set consist of the three variables which correspond the three relevant features of the currently negotiated section of the circuit (as illustrated in Figure 5) and a random constant within the range between 0 and 10. The main attributes of GP are summarized in Table 2.

TABLE 2
MAIN ATTRIBUTES OF GP

| Category | Value |
|---|---|
| Function set | { +, -, *, / } |
| Terminal set | {dist_p_c, dist_c_n, angle_p_c_n, random constant [0..10] } |
| Population size | 100 individuals |
| Selection | Binary tournament, selection ratio 0.1, reproduction ratio 0.9 |
| Elitism | Best 4 individuals |
| Mutation | Random sub-tree mutation, ratio 0.01 |
| Trial interval | Two laps around a predefined circuit |
| Fitness | Average lap time of two consecutive laps (ms) penalized with 0.6 ms for "hitting" the "guardrails" |
| Termination criteria | Number of generations = 40 |

Considering the evolving rule as a genotype and the evaluated concrete values of the four parameters of driving style as a phenotype, the fitness evaluation of GP is accomplished in the following two phases:
● Developing the genotype into phenotype by setting the

values of the *four* parameters of the driving style pertaining to each of the turns in the circuit to the concrete values obtained from the evaluation of the *four* expressions comprising the driving rule (Figure .6), and
● A time trial of the car operated by the driving agent governed by the evaluated values of the four driving style parameters.

## V. EXPERIMENTAL RESULTS

We conducted an experiment on evolution of the driving rules in order to verify the ability of the proposed evolutionary approach to automatically discover (if existent) a single driving rule that is general enough to be able to adequately control the car around all the turns in a predefined circuit. We considered the following four predefined circuits featuring different characteristics: an O-shaped (two right, single-apex turns), 8-shaped (a right and a left, double-apex turns), S-shaped (a series of right and left turns) and 2S-shaped (concatenated two S-shaped) circuits. The results of fitness convergence aggregated over 20 independent runs of GP and the driving line of the car controlled by a sample best-of-run driving rule are illustrated in Figure 7. As Figure 7 depicts, the average of the best lap time over all runs improved from 3850ms to 3600ms for O-shaped (Figure 7a), from 5410ms to 5000ms for 8-shaped (Figure 7b), from 4800ms to 4120ms for S-shaped (Figure 7c), and from 7810ms to 7060ms for 2S-shaped (Figure 7d). The result of the evolution of driving rules on all four circuits, intended to verify the generality of the evolved driving rules across several circuits with diverse characteristics is shown in Figure 7e). As Figure7e) depicts, the aggregated lap time improves in average from 25430ms to 21520ms within 40 generations.

The emergent driving line with the breakdown of the velocity of the car governed by sample evolved best of run driving rule on 2S circuit is shown in Figure 7.

In order to estimate the degree of optimality of the evolved general driving rules, we conducted a comparative analysis of the lap times of the agent governed by these general rules with the lap times of the agent controlled by the values of driving style parameters, directly evolved for each of the turns of the considered circuits. In order to directly evolve these values, we employed genetic algorithms (GA) with the main attributes which are identical to those of the GP used for evolution of the driving rules. The results are summarized in Table 4. As the data depicted in the Table 4 illustrate, the driving rules evolved via GP on particular circuits performs equally well (and, occasionally even better) than the fixed, well-tailored values of the driving style parameters. The maximal degradation of about 7% of the lap time associated with the driving rules in S-shaped circuit indicates the favorable generality of the evolved driving rules.

The slight superiority of the evolved driving rules over the optimized values of driving style parameters in O-shaped and 8-shaped circuits (3.1% and 0.2% faster, respectively) can be explained by the reduction size of the evolutionary search of GP compared to that of GA. Indeed, both the O-shaped and 8-shaped circuits feature identical turns. Therefore, the way the car negotiates them can be naturally expressed by a single

driving rule. Conversely, GA needs to evolve the values of the driving style parameters for each of the turns separately, and the equality of the values of these parameters should be explicitly discovered. The presumed implications of the size of search space in both GP and GA for O- and 8-shaped circuits are illustrated by the corresponding fitness convergence characteristics. The fitness of GP converges faster (Figure 8a and 8b) than that of GA (Figure 9a and 9b),

which suggests that a delaying the termination of the evolution (e.g., by setting the termination criteria of GA to more than 40 generations) would facilitate the additional improvement of lap times. Consequently, the superiority of GP over GA on O- and 8-shaped circuits should be acknowledged only in the condition of the considered budget of the evolutionary runs of no more than 40 generations.
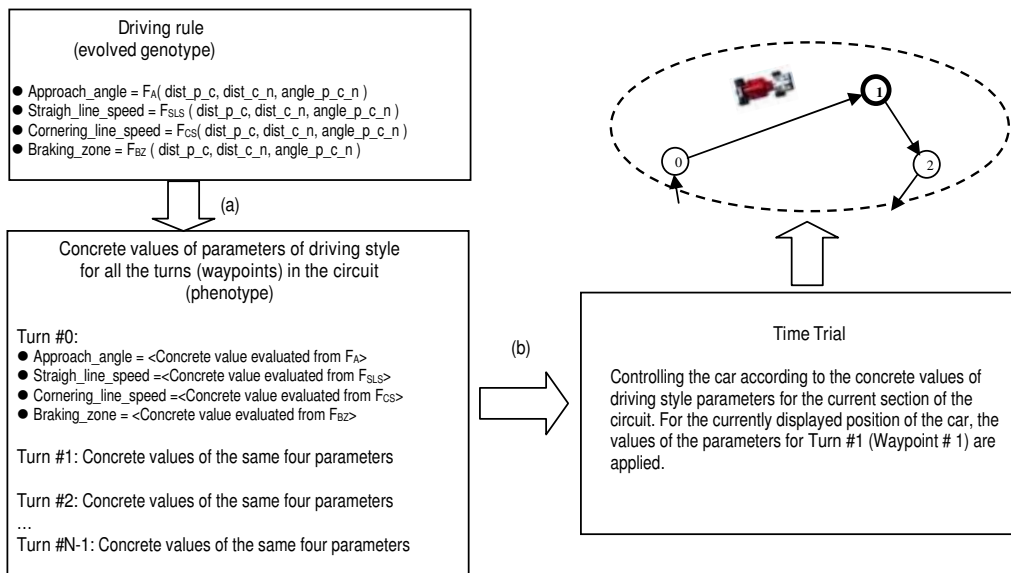


Fig. 6. The fitness evaluation accomplished in two phases: (a) developing the genotype (driving rule) into the phenotype (values of the driving style parameters), and (b) time trial with the driving agent being governed by the evaluated values of the four driving style parameters.
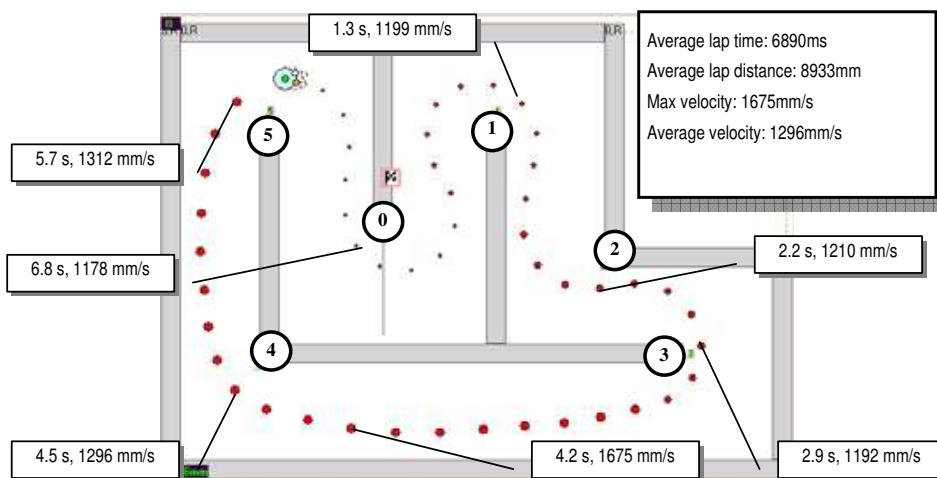


Fig. 7. The driving line and the breakdown of velocity of the car controlled by evolved sample set of best-of-run driving rules. The start-finish line of the illustrated 2S circuit is located next to the waypoint #0. The running direction is clockwise. The principal points of lap are marked with the timestamp information (the time into the lap) and the velocity of the car.
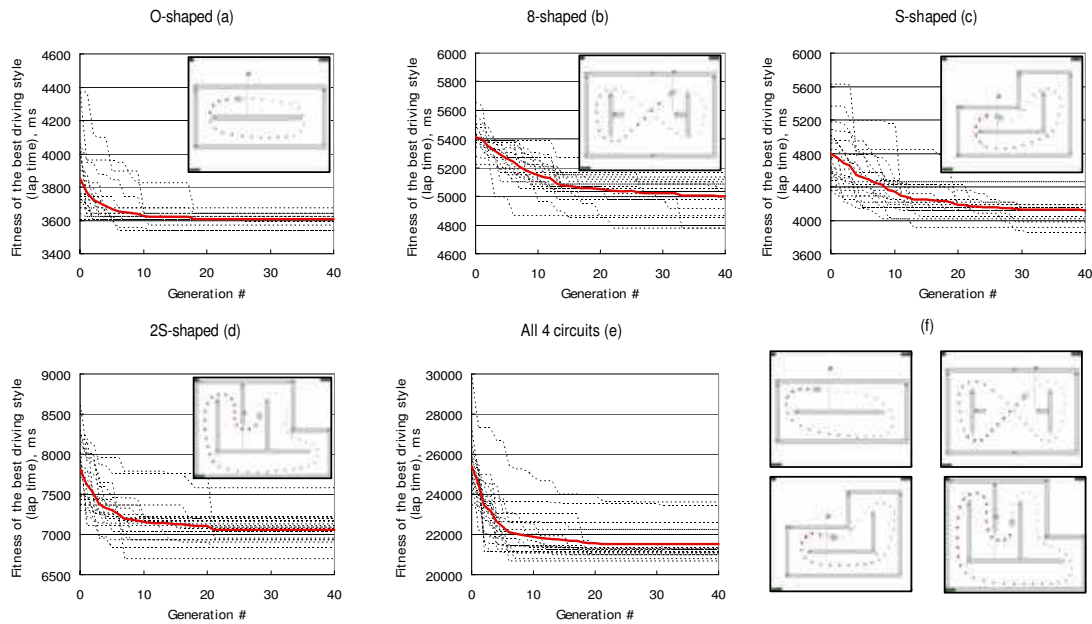
Fig. 8. The fitness convergence and the driving line of a sample best-of-run driving rules, evolved in 20 independent runs on O-shaped (a), 8-shaped (b), S-shaped (c) and 2S-shaped circuits (d), respectively. The results of evolution on all four circuits are shown in (e), and the sample best-of-run driving lines are depicted in (f).
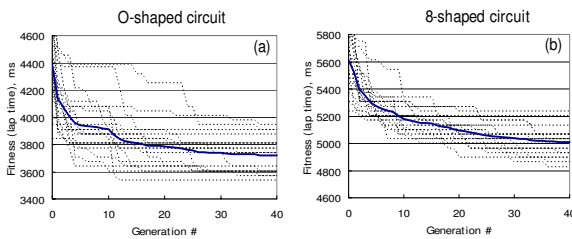


Fig. 9. Fitness convergence of 20 independent runs of GA evolving the values of the four driving style parameters in O-shaped (a) and 8-shaped (b) circuits respectively.

TABLE 3

COMPARISON OF THE AVERAGE LAP TIMES OBTAINED THROUGH A DIRECT OPTIMIZATION OF THE DRIVING STYLE PARAMETERS EMPLOYING GA AND THE DRIVING RULES, EVOLVED VIA GP.

| Circuit / Evolutionary approach | O-shaped | | 8-shaped | | S-shaped | | 2S-shaped | |
|---|---|---|---|---|---|---|---|---|
| | Lap time ms | Degradation % | Lap time ms | Degradation % | Lap time ms | Degradation % | Lap time ms | Degradation % |
| GA for particular circuit | 3721 | --- | 5011 | --- | 3848 | --- | 6684 | --- |
| GP for particular circuit | 3604 | -3.1 | 5002 | -0.2 | 4120 | 7.0 | 7060 | 5.6 |
| GP for all four circuits | 4095 | 13.6 | 5430 | 8.4 | 4725 | 22.8 | 7219 | 8.0 |

## VI. CONCLUSION

We presented an approach of automated design of the functionality of driving agent, able to operate a software model of fast running car. We verified the ability of the employed evolutionary paradigm to automatically discover a single set of driving rules that are general enough to adequately control the car around all the turns in a predefined circuit. The optimality of the evolved general driving rules is verified by the favorable comparison against the corresponding lap times obtained by direct evolutionary optimization of the values of the main driving style attributes.

In our future work we are planning to investigate how well the rules, evolved in predefined circuit (or circuits) would perform in a priori unknown circuits. We anticipate that the degree of diversity of characteristics of the learning circuits might have a significant influence on the generality of such rules. Also, we contemplate a possible need of adaptation of the driving agent evolved in predefined circuits to the characteristics of unknown ones. The proposed generative representation of the driving rules would allow implementing such an adaptation as a direct tuning of the evolved values of driving style parameters in a way consonant with the biologically plausible model of phenotypic plasticity.

*2008 IEEE Congress on Evolutionary Computation (CEC 2008)*

REFERENCES

[1] R. Bentley, "Speed Secrets: Professional Race Driving Techniques, " Motorbooks International, 1998.

[2] D.B. Fogel, "Blondie24 : Playing at the Edge of AI," Morgan Kaufmann, 2001.

[3] P. Frere, "Sports Cars and Competition Driving," Bentley Publishing, 1992.

[4] J.D. Funge, "Artificial Intelligence for Computer Games," Peters Corp., 2004.

[5] T. Gillespie, "Fundamentals of Vehicle Dynamics," Society of Automotive Engineers International, 1992.

[6] Google Maps, Image of the junction near Matsubara city in Osaka Prefecture, Japan, URL: http://maps.google.com/maps? ll=34.59,135.575&spn=0.00354,0.0042

[7] IBM Corporation, "Deep Blue", URL: http://www.research.ibm.com/deepblue/, 1997.

[8] J.R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", Cambridge, MA, MIT Press, 1992.

[9] Robocup, URL: http://www.robocup.org/02.html, 2005.

[10] M. Suzuki and D. Floreano, "Active Vision for Neural Development and Landmark Navigation." In 50th Anniversary Summit of Artificial Intelligence, pp. 247-248, 2006.

[11] I. Tanev and K.Shimohara, "XGP: XML-based Genetic Programming Framework", Proceedings of the 34th Symposium of the Society of Instrument and Control Engineers (SICE) on Intelligent Systems, pp.183-188, 2007.

[12] I. Tanev, M. Joachimczak and K. Shimohara, "Evolution and Adaptation of an Agent Driving a Scale Model of a Car with Obstacle Avoidance Capabilities", Proceedings of the Ninth International Conference on the Simulation of Adaptive Behavior (SAB 2006), LNAI 4095, Springer-Verlag Berlin Heidelberg, pp.619-630, 2006.

[13] I. Tanev and K. Shimohara, "On Human Competitiveness of the Evolved Agent Operating a Scale Model of a Car", Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007), Sep.25-28, 2007, Singapore, pp.3646-3653, 2007

[14] J. Togelius and S. M. Lucas, "Evolving Controllers for Simulated Car Racing", Proceedings of IEEE Congress on Evolutionary Computations (CEC-2005), Edinburgh, UK, September 2-5, pp.1906-1913, 2005.

[15] K. Wloch and P. Bentley, "Optimizing the Performance of a Formula One Car Using a Genetic Algorithm", Proceedings of the 8th International Conference on Parallel Problem Solving from Nature, Birmingham, UK, September 18-22, pp.702-711, 2004