algorithms

MDPI

Article

Evolving the Controller of Automated Steering of a Car in Slippery Road Conditions

Natalia Alekseeva *, Ivan Tanev and Katsunori Shimohara

Department of Information Systems Design, Faculty of Engineering, Doshisha University 1-3, Tatara Miyakodani, Kyotanabe City, Kyoto 610-0394, Japan; itanev@sil.doshisha.ac.jp (I.T.); kshimoha@sil.doshisha.ac.jp (K.S.)

* Correspondence: alekseeva2017@sil.doshisha.ac.jp

Received: 8 June 2018; Accepted: 19 July 2018; Published: 21 July 2018

Abstract: The most important characteristics of autonomous vehicles are their safety and their ability to adapt to various traffic situations and road conditions. In our research, we focused on the development of controllers for automated steering of a realistically simulated car in slippery road conditions. We comparatively investigated three implementations of such controllers: a proportional-derivative (PD) controller built in accordance with the canonical servo-control model of steering, a PID controller as an extension of the servo-control, and a controller designed heuristically via the most versatile evolutionary computing paradigm: genetic programming (GP). The experimental results suggest that the controller evolved via GP offers the best quality of control of the car in all of the tested slippery (rainy, snowy, and icy) road conditions.

Keywords: autonomous vehicles; automated steering; slippery road conditions; PID controllers; genetic programming

1. Introduction

Since the 1930s, the authors of science fiction books have envisioned the advent of self-driving cars, and constructing such cars has been a challenge for the Artificial Intelligence (AI) community since the 1960s [1]. Nowadays, the advances of sensing (e.g., computer vision, laser and microwave ranging, etc.), mapping, networking, and machine learning have facilitated the rapid growth of research and prototyping of semi-autonomous and autonomous vehicles [2]. However, most of the implemented solutions so far seldom assume an unconditional and complete autonomy of the vehicle. Moreover, the currently available—up to level 3 (of the six levels, 0–5) of autonomous driving, as defined in 2014 by the Society of Automotive Engineers—industrial applications of semi-autonomous vehicles require that the drivers keep their hands on the steering wheel all the time during the operation of the "auto-pilot" [3]. Indeed, in a case of an unforeseen traffic situation or road condition, the control might have to be transferred to the presumably more dependable human driver [4]. We believe that the automation should abstain from unconditionally ceding the control of the car to the human driver, especially in heavy traffic situations or in challenging (such as, for example, slippery) road conditions due to the major drawback of this transfer: the reduced cognitive load of the (passive) human driver might result in a rough transition of control between the automation and the human [5]. In such situations, the behavior of the human might be either inadequate or too slow as they might be subjected to a sharp increase of their cognitive load and the psychological stress of dealing with the suddenly arising, challenging traffic situation (e.g., a suddenly appearing obstacle in front of the car in dense multi-lane traffic). In addition, the human driver might be underqualified to control the car in challenging road conditions (e.g., retaining control of an oversteering car on a slippery road). The recent traffic accidents—some of them fatal—involving autonomous cars drew attention to the problem of the cognitive conditions required of human drivers to allow them to serve as a dependable backup for the potentially imperfect automation [6–8]. Nevertheless, autonomous vehicles are expected to outperform human drivers in terms of improved overall safety of road traffic, and to result in a decrease of the total number of accidents caused by a slow or inadequate response of a human driver due to fatigue, inattention, and a lack of experience or qualification in dealing with extreme traffic situations and road conditions [3].

The task of automated driving can be decomposed into the following subtasks: (i) defining the desired trajectory (driving line) and the desired pattern of speed along this trajectory, (ii) keeping the actual trajectory of the car as close as possible to the desired one, and (iii) maintaining the actual speed of the car along this trajectory as close as possible to the desired one [9,10]. We considered the first and third subtasks to be beyond the scope of our current work; instead, we focused on the second one—in challenging, slippery road conditions—which could be solved by an appropriate steering of the car.

Currently, the canonical servo-control of steering [11]—which, as we will elaborate later, could be seen as an example of a proportional-derivative (PD) controller—is adopted as a formal model of the steering angle function (SAF): the function that continuously decides the current steering angle of the front (steering) wheels depending on both the lateral and angular deviation of the car from its intended trajectory. The model continuously attempts to minimize the values of these two deviations (i.e., the errors) by setting the steering angle to such a value that would result in both a prompt and stable (non-oscillatory) return of the car to its desired trajectory. This model mimics well the steering behavior of a human driver and, similarly to such a driver, provides a good quality of steering on dry, non-slippery roads. With an appropriate tuning of the relevant gain coefficients (depending on the specific features of the physical model of the particular car), on dry roads the servo-control could achieve a steering behavior that is very similar to that of a human driver in adequate cognitive condition [12].

However, to the best of our knowledge, there is no documented research on the applicability of the PD servo-control model for automated control of the car under more challenging—*slippery* (e.g., wet, snowy, or icy)—road conditions. We speculate that PD servo-control might not be adequate in such conditions because the vehicle dynamics model of a directionally unstable (e.g., understeering or oversteering) car on slippery roads is more complex and involves additional variables (beyond the lateral and angular deviations from the desired trajectory) pertinent to the state of the car than are involved in a car driven on normal, non-slippery roads [13]. Such complexity might not be expressed adequately by the relatively simple PD servo-control.

Moreover, in an eventual case of unacceptability of the PD servo-control as a steering model in such slippery conditions, which alternative model would be more appropriate, and how to develop it, would be open questions too.

These two concerns—the eventual inapplicability of the PD servo-control model and the lack of understanding of its alternative(s) for the steering of a car on slippery roads—motivated our research. Our *objectives* were (i) to examine the applicability of PD servo-control as an auto-steering model of a car on slippery roads, and (ii) to investigate the feasibility to develop heuristically the optimal (possibly nonlinear) steering model in such road conditions by means of simulated evolution via genetic programming (GP).

Our work is additionally motivated by the fact that despite the significant body of research on computational intelligence in The Open Racing Car Simulator (TORCS) [14,15], we are not aware of any implementation of SAF of the car on slippery roads in this simulation environment. Rather, the focus of such research is on the automated development (via either a simulated evolution or machine learning) of racing agents aimed at attending (and, ultimately, winning) simulated car races. The eventual incorporation of SAF on slippery roads would be seen as unneeded as the races are usually held on dry, grippy tracks where the cars seldom experience any significant directional instability.

On the other hand, the inspiration for the proposed heuristic development of SAF via GP is that, to the best of our knowledge, there are no documented attempts to automatically develop the SAF on slippery roads via evolutionary computing and GP in particular.

The related work includes the seminal work of Huang et al. [11], who have demonstrated that GP—as an unsupervised machine learning approach—could be successfully applied to heuristically develop from scratch a PD controller that stabilizes a car on slippery roads by eliciting precisely quantified asymmetric brake forces on its wheels. The control of the steering, however, has not been subjected to any optimization for the given slippery road conditions, but rather implemented as a generic, handcrafted PD servo-control that has performed well on dry, grippy roads. Similarily to GP, end-to-end reinforcement learning (e.g., of deep neural networks) does not require an explicit modularization of the controller [16,17]. Instead, the internal structure of the controller is automatically developed by the learning framework. The advantage of his approach is that it could discover and utilize the unusual features of the environment and develop sophisticated controllers that could solve the tasks for given (generic) environmental conditions. The drawbacks include the huge search space, a lack of understanding of how (and why) exactly the developed controllers work, and consequently a lack of confidence in their robustness and generality. Moreover, conversely to the proposed approach of employing GP, end-to-end reinforcement learning requires a human-annotated training set of data, i.e., the mapping of the states of the car and the environment (e.g., consecutive frames of a video feed) into appropriate steering commands obtained from a human (expert) driver. However, in our work we assume that such perception information is not always fully accessible (or, even if accessible, not necessarily with the same quality as the information used for training), for example, when driving in heavy-traffic, poor-visibility conditions (fog, snow, rain, night, etc.) or immediately behind a heavy vehicle (bus, truck, etc.). Therefore, in our work we propose the use of very simple perception information: only the current lateral and angular deviation of the car from the intended trajectory (i.e., the center of the lane). With such simple perceptions, for a human driver it would have been virtually impossible to achieve driving (let alone on slippery roads) that is good enough to serve as a representative trainer for the learner.

Rather than automatically (via GP), the steering controller of the car on slippery roads could be handcrafted by the developers applying various top-down approaches. Usually, handcrafted solutions are based on certain assumptions intended to simplify the complex dynamics of a skidding car. The challenge of these approaches is in deciding the adequate assumptions and abstractions for the domain-specific knowledge. In predictive control [18–20], for example, the controller decides the value of the control signal based on the predicted, rather than the current, values of parameters pertinent to the state of the car and its environment. The prediction could be seen as approximating the values of these parameters from their current values, their rate of change, and the prediction time that corresponds to the latencies in the control loop. Such prediction would be very accurate if the latencies are well-known, the laws that govern the changes of these latencies are also well-known, and the rate of change is (nearly) constant during the duration of the latency. In contrast to these approaches, the proposed method of automated, heuristic development of SAF via GP—as elaborated in detail in Section 2.5—assumes very little of such knowledge.

The remainder of the article is organized as follows. Section 2 elaborates on the adopted model of the car and its environment. It also explains the basics of the PD and PID servo-control models and the proposed approach of employing GP for evolution of the optimal SAF. Section 3 presents our experimental results, and Section 4 discusses the some of the limitations of the results obtained by GP. Finally, Section 5 draws a conclusion.

2. Materials and Methods

2.1. The Car in the Open Racing Car Simulator

We adopted The Open Racing Car Simulator (TORCS) [14,15] for simulation of both the car and its environment. The main parameters of the simulated car adopted in our research are shown in

Table 1, and its snapshot is depicted in Figure 1. The rationale of employing TORCS to model the car is motivated by the severity of the drawbacks of the existing alternative approaches. Indeed, evaluating the quality of the evolving SAF on a real car would be too unsafe, expensive, and slow. The hardware-in-the-loop (HIL) modeling [21] of the embedded electronic systems offers a possible solution to the former two issues. Indeed, in HIL, the real, hardware implementation of the driving aid (e.g., traction control, anti-locking brake system, electronic stability program, SAF, etc.) is connected to a computer that runs the driver-, vehicle-, and environment-modeling software. Such an approach, however, would be unable to solve completely the problems of (i) the cost (because of the need to build the real hardware and its computer interface) and (ii) significant runtime overhead (as the evaluation of the real hardware occurs in a real time). In our research, the proposed use of a car modeled in TORCS is intended to address all of the above-mentioned drawbacks of the alternative approaches of evaluating the developed SAF: TORCS is both free of charge and computationally efficient: one minute of simulated real time usually requires less than one second of runtime. Additional advantages of employing TORCS include (i) the realism of the simulation (by accounting faithfully for all the relevant forces that act upon a moving car) and (ii) the openness of its source code to the eventual modifications that would be needed for the intended implementation of SAF.

Table 1. Main Parameters of the Simulated Car.

Feature	Value
Model	Coupé Leicht Kompakt Deutsche Tourenwagen Meisterschaft (CLK DTM)
Length, m	4.76
Width, m	1.96
Height, m	1.17
Mass, kg	1050
Front/rear weight repartition	0.5/0.5
Height of center of gravity, m	0.25
Coefficient of friction of tires	1.0
Drivetrain	Front engine, rear wheel drive
Steering delay, s	0.1
Steering rate limit, °/s	30

Figure 1. A snapshot of the simulated car.

The computational efficiency of TORCS implies that a trial run of the car on a given track could be simulated in about one second of a computational runtime. This feature is very important for the computational efficiency of the evolutionary paradigm, GP, proposed to develop the optimal SAF. As a heuristic, trial-and-error approach, GP usually requires thousands of trials (fitness evaluations) before obtaining a feasible solution.

2.2. The Track

The flexibility and openness of TORCS allowed us to develop a dedicated track for testing the discussed steering approaches of the car. The layout of the track is a variant of the so-called "fish hook" requiring a "road edge recovery maneuver" [22]. It is often used for testing the dynamics (especially the tendency for a rollover) of real-world vehicles. It consists of two sectors: Sector 1, including a short straight followed by a left turn, and Sector 2, including a single, long 180° right turn. The track is illustrated in Figure 2, and its main characteristics are given in Table 2. The coefficient of friction μ between the tires of the car and the surface of the track could be set in TORCS to the values corresponding to dry (μ = 0.8, 1.0), rainy (μ = 0.5, 0.6), snowy (μ = 0.4), and icy (μ = 0.3) conditions, respectively.

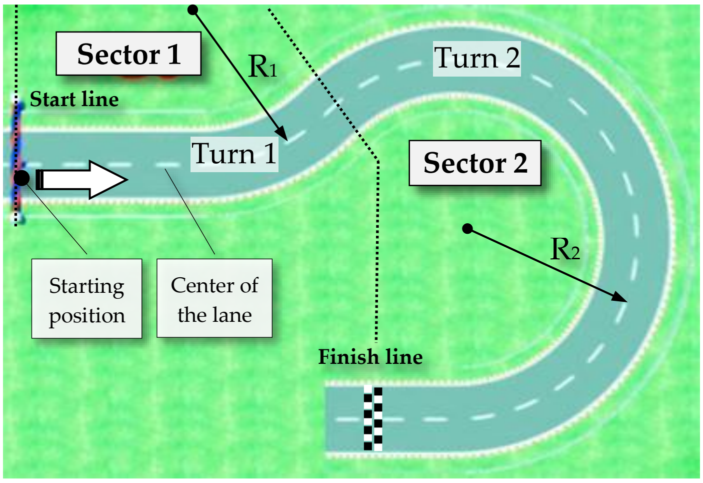


Figure 2. Hook-type test track.

Table 2. Main Features of the Test Track.

Feature	Value
Total length, m	300
Lane width, m	20
Length of sector 1, m	90
Radius of turn 1, R_1 m	50
Length of sector 2, m	210
Radius of turn 2, R_2, m	50

2.3. Servo Control as a PD Controller

The first of the three steering controllers that we investigated is based on the servo-control model of steering. It represents the canonical approach of defining the instant value of the steering angle in autonomous cars [23,24]. According to this model, the SAF that defines the steering angle δ is assumed to be the sum of the following two scaled parameters: the linear e and angular θ deviations of the car from its desired trajectory (e.g., the center of the lane) as illustrated in Figure 3. The SAF of the servo control model could be expressed as the following Equation (1):

$$\delta = k_1\, e + k_2\, \theta \tag{1}$$

where the scaling coefficients (gains) k_1 and k_2 are calculated as follows:

$$k_1 = \text{Steering\_Factor/Steering lock angle} \tag{2}$$

$$k_2 = 1/\text{Steering lock angle.} \tag{3}$$

The steering lock angle is the maximum angle of turning of the front wheels of the car, and its value, for the car used in our experiments, is 0.62 rad (35°). Alternation of the coefficients k_1 and k_2 affects the tradeoff between the main requirements of steering [25]: smoothness, fast response, and stability in the way the car returns to the center of the lane if it occasionally deviates from it. The servo control model is intended to steer the car in such a way that results in minimization of the mean value of the absolute deviation (the error) e of the geometrical center of the car from the center of the lane during a given trial.

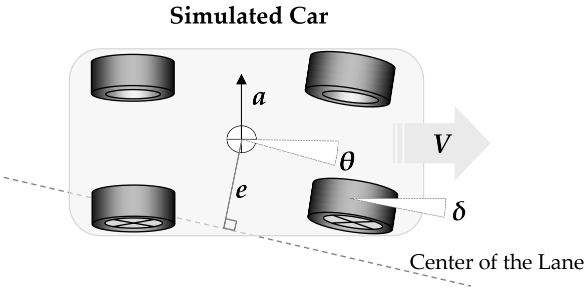


Figure 3. Most relevant variables pertinent to the state of the car.

Assuming that, for small angles θ and very short periods of time dt,

$$\theta \approx de/dx = de/(V\ dt) \tag{4}$$

where V is the speed of the car. For a constant speed V, Equation (4) could be rewritten as

$$\theta \approx k_v\ de/dt = k_v\ e' \tag{5}$$

where e' is the first derivative of the lateral deviation of the car from the center of the lane. Substituting the value of θ from Equation (5) in Equation (1), we obtain a different form of the servo-control model:

$$\delta = k_1\ e + k_2\ (k_v\ e') = k_1\ e + k^*{}_2\ e'. \tag{6}$$

Equation (6) expresses the servo-control model of steering as a PD controller. Its output—the *control variable*—the steering angle δ, is a sum of the proportional (P) and derivative (D) terms of the *error*: the deviation of the car from the center of the lane e. Because the *desired process value* (i.e., the desired deviation from the center of the lane) is 0, the absolute value of the error e is equal to the *measured process value*. The controller attempts to minimize the value of the *error* by constantly adjusting the steering angle δ, which, presumably, would yield a trajectory as close as possible to the center of the lane (Figure 4). We will explain the mechanism of tuning the parameters k_1 and $k^*{}_2$ in Section 3.

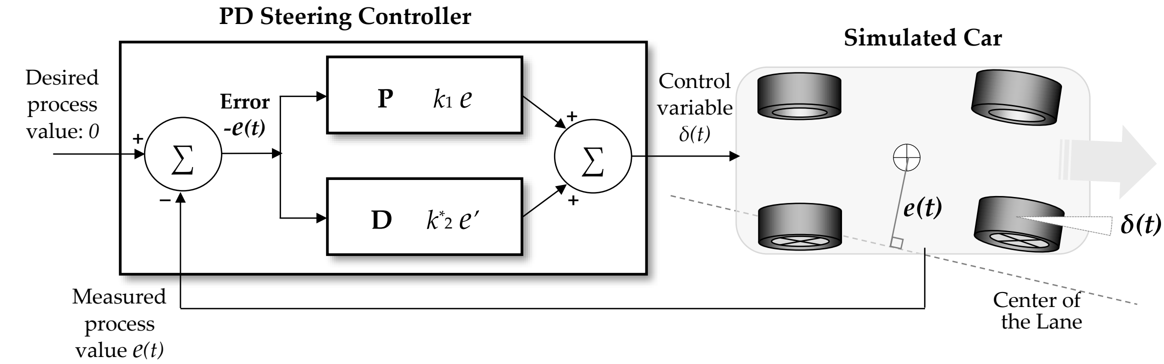


Figure 4. Servo-control model of steering as a proportional-derivative (PD) steering controller. The steering angle function (SAF), defining the steering angle δ, is implemented as the sum of the proportional (P) and derivative (D) terms of the *error:* the deviation *e* from the center of the lane.

2.4. Extending the Servo-Control Model: A PID Steering Controller

The second controller that we investigated is based on the above-discussed PD servo-control model, extended with an additional integral term. The rationale for incorporating the integral term is that it could provide an additional adjustment to the steering of the car if the car does not return to the center of the lane (e.g., due to skidding on cornering in slippery road conditions) for a certain period of time (in our experiments, 2 s):

$$\delta = k_1\, e + k^*_2\, e' + k_3 \int e\, dt. \tag{6}$$

In Section 3, we will explain the mechanism of tuning the parameters k_1, k^*_2, and k_3.

2.5. Evolving the Steering Controller via GP

The third controller we comparatively investigated was a controller utilizing (i) an extended set of parameters (EP) pertinent to the state of the car, and (ii) a relaxed—an arbitrary (rather than additive, as in the PD and PID controllers)—structure of its analytical model (RM) developed heuristically via GP. We denote it the GP-RMEP controller.

The rationale of adopting GP is based on our concerns that the above-mentioned PD and PID servo-controllers might not be adequate, because, compared to a car driven on normal, non-slippery roads, the vehicle dynamics model of the skidding, directionally unstable car on a slippery road is rather complex and might involve of additional parameters (beyond the lateral deviations from the desired trajectory) pertinent to the state of the car [13,26]. Such complexity might not be expressed adequately by the relatively simple, additive PD and PID servo-control models. In addition, due to this complexity, we could not devise an eventual analytical approach to the steering of the car that would effectively counter its directional instability. Moreover, the only parameter pertinent to the state of the car—the lateral deviation *e*—used in the SAF, based on the PD and PID servo-control models, might not represent adequately the state of the unstable car on slippery roads. Indeed, even featuring the same lateral deviation, the car may experience different yaw rates and different accelerations of these rates. Depending on how the actual yaw rate differs from the desired one (estimated from the steering angle and the wheelbase of the car according to the bicycle model of a car [3]), the directionally unstable car might suffer either from understeer or oversteer, requiring a different steering input from the controller. On the other hand, increasing the number of state parameters would further complicate the model of the steering controller, which could render the formal, analytical approaches to devise such a controller practically unfeasible.

Moreover, even if we assume that it is possible to handcraft the code (applying various top-down, theoretical approaches based on vehicle dynamics [25,27,28]) of a more complex, nonlinear PID SAF that controls the steering of an unstable, skidding car in such a way so it regains its stability in a more effective way than the simple PD and PID controllers based on the servo-control model, neither the degree of the optimality of such a code, nor the way to further improve it, would be apparent to the human developer. Therefore, an automated mechanism (i) to evaluate the quality of the coded SAF and (ii) to incrementally improve its intermediate version(s)—e.g., based on models of the natural evolution of species—might be needed.

The proposed approach of employing GP implies that the SAF is automatically designed—without an a priori defined structure nor complexity—by a computer system via simulated evolution through selection and survival of the fittest in a way similar to the evolution of species in nature [4,5,29]. Adopting GP offers the following advantages:

- Creative design: GP could evolve the controller's SAF of an arbitrary structure and complexity. Moreover, GP does not require an incorporation of any a priori, domain-specific expert knowledge. This is especially important when such knowledge is mostly empirical, rather than analytical.
- Emergent intelligence: Rather than being introduced by the human expert, the task-specific knowledge of how to steer the car on slippery roads might emerge solely from the interaction of the problem solver (the GP) and the steering evaluation (trial) runs of the car [30].
- Human competiveness: GP, as an automated process that does not follow the human top-down problem-solving logic, opens the possibility of creating a solution that would be better than one (i.e., based on the PD and PID servo control models) designed by a human [26].

The main features of the adopted GP, namely genetic representation, function set, terminal set, fitness evaluation, and genetic operations, are elaborated below.

2.5.1. Genetic Representation, Function Set, and Terminal Set of the GP

The SAFs, evolved in GP, are represented genetically as parse trees. Because the SAF is an algebraic function of the parameters pertinent to the state of the car, we included the basic algebraic operations in the function set of the GP:

$$\text{Function Set} = \{\text{addition, subtraction, multiplication, (protected) division}\}$$

The division operation is protected in that an eventual division to zero does not result in an arithmetic exception, but rather in a constant value of 1. The terminal set consists of the arguments of the evolved SAF; i.e., the parameters pertinent to the state of the car. We extended this set from the three parameters e, e', and $\int e$ that we used in the PID controller based on the extended servo-control model by adding the speed of the car V, the steering angle δ, the lateral acceleration a, the lateral jerk a', the lateral angular deviation θ, and the yaw rate θ'. Also, the terminal set includes a random constant C within the range [0, 10]:

$$\text{Terminal Set} = \{e, e', \int e, V, \delta, a, a', \theta, \theta', C\}$$

The constant C allows the GP to evolve appropriate scaling when used as a multiplicative term and biasing (as an additive term) coefficients in the SAF. A sample parse tree is shown in Figure 5. The value of the SAF is evaluated from the parse tree by a canonical depth-first tree traversal algorithm.

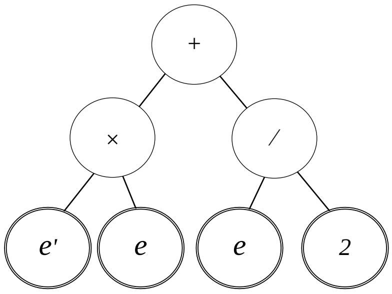


Figure 5. A parse tree, representing the sample SAF $\delta = e'\,e + e/2$.

2.5.2. Fitness Evaluation

The fitness value is intended to estimate the quality of the steering produced by the evolved SAF. We defined the criteria of such a quality from the desired characteristics of the trajectory of the car during the trial. It is implemented on a given test track (as shown in Figure 2) featuring a given friction coefficient, μ, as follows: first, the simulated car is initially positioned at the starting position of the track. Similarly to [3], in order to include in the evaluation of the evolved SAF its response to a step disturbance, the starting position is set to be parallel, 5 m off the center of the lane. Then, the car accelerates slowly to a given constant target speed. Then, in order to abstract the dynamics of the car from the effect of eventual excessive traction or braking forces, the car is maintained at a constant speed by simulated cruise control. To render the task of controlling the car challenging, but solvable, the target speed is set to 0.85 of the critical speed V_{CR}. The critical speed is the speed at which the car theoretically could not negotiate the corners of the test track with the given coefficient of friction without losing control, running off the track, and eventually crashing. This speed is approximated as the speed at which the centrifugal forces during steady-state cornering $F_C = mV_{CR}{}^2/R$ (where m is the mass of the car, and R is the radius of the turn) become theoretically equal to the friction force $F_F = \mu\ m\ g$ (where μ is the overall coefficient of friction and g is the gravitational acceleration). At the traveling speed of 0.85 of V_{CR}, the car inherently suffers from intermittent instability due to the yaw inertia both in the entry and exit of corners and due to the dynamics of lateral weight transfer in corners [3,14] that we intend to counter by the evolved SAF. The car, traveling at V_{CR} (and above), is theoretically uncontrollable. Consequently, there would be no existing SAF that results in a steerable car. Similarly, a car traveling considerably slower than V_{CR} does not suffer from any instability and its steering could be accomplished adequately by the canonical servo-control models. The speeds of the car during the trials on the track featuring different friction coefficients are shown in Table 3.

Table 3. Speed of the car during the fitness trial on the test track with different friction coefficients.

#Road Condition	Friction of Tires, μ_t	Friction of Road Surface, μ_s	Overall Friction, $\mu = \mu_t \times \mu_s$	Critical Speed, V_{CR}, m/s	Speed of the Car (0.85 V_{CR}), m/s
1	1.0	1 (dry)	1	22.13	18.82
2	1.0	0.8 (dry)	0.8	19.79	16.8
3	1.0	0.6 (rainy)	0.6	17.15	14.5
4	1.0	0.5 (rainy)	0.5	15.65	13.3
5	1.0	0.4 (snowy)	0.4	14	11.9
6	1.0	0.3 (icy)	0.3	12.12	10.3

The speed of the car is kept constant during the trial by a simple, handcrafted cruise control mechanism that maps the difference between the desired speed (e.g., as shown in Table 3, 13.3 m/s for the trial on a track with friction coefficient $\mu = 0.5$) and the actual one into an increment (or decrement) of the position of the accelerator pedal. As the car reaches the desired speed, the steering of the car is assumed by the evolved SAF. Then, the SAF starts to continuously (with a sampling frequency of 40 Hz) produce the desired steering angle δ calculated for the currently perceived values of the parameters pertinent to the state of the car. The intended, optimal, trajectory of the car steered by the evolved SAF should feature both a quick and smooth return to the center of the lane followed by a precise drive along it (Figure 6, Case 3). We view the trajectories shown in Figure 6 as Case 1 and Case 2 as suboptimal, as they represent either a too-slow (Case 1), or a too-quick, oscillating return (Case 3), respectively. Indeed, a too-slow return would be unable to either follow adequately the center of the lane on cornering or to return to it in a case of deviating from it due to skidding on slippery roads. On the other hand, a too-quick return would imply an inherently oscillating trajectory of the recovery.

In order to formally express the defined criteria of optimality of the SAF-induced steering, we defined the fitness function F as a weighed sum of the following two components: (i) the area A_T under the trajectory of the car around the center of the lane (as an integral of the absolute value of the lateral deviation e) and (ii) the average of the lateral velocity V_{L_AVR} (as an integral of the lateral acceleration a) of the car:

$$F = A_T + C_V\ V_{L\_AVR}. \tag{7}$$

The lower fitness values correspond to a better quality of steering, and, ultimately, the desired trajectory (Figure 6, Case 3) would feature an optimal trade-off between the values of the two components that result in a minimal fitness value. Indeed, the suboptimal trajectories would be subjected to a detrimental selection pressure by the GP either due to the too-wide area A_T under the trajectory (Figure 2, Case 1) or a too-high lateral velocity V_{L_AVR} (Figure 2, Case 2). We verified experimentally that the value 0.5 of the weight coefficient C_V in Equation (7) results in an optimal tradeoff between the values of the two additive components of the fitness function.

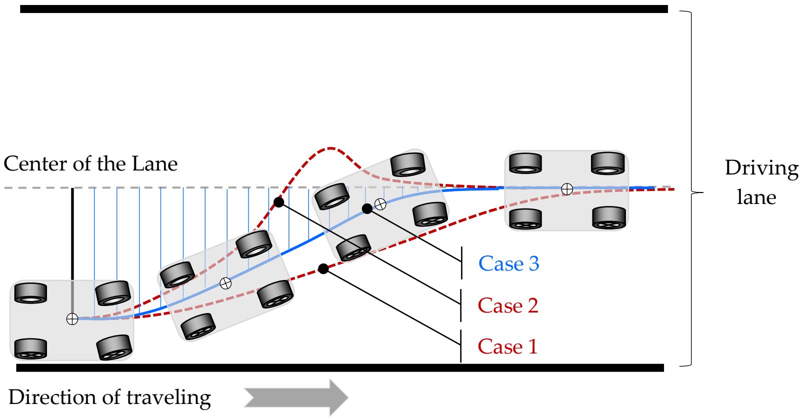


Figure 6. The three main cases of the trajectory of the car during the fitness trial: the preferred trajectory (Case 3) of the car would feature both a quick (i.e., with a narrow area under the trajectory) and oscillation-free (with a low average lateral velocity) return to the center of the lane. Case 1 and Case 2 illustrate a too-slow and a too-quick (oscillating) return to the center of the line, respectively.

We would like to note that, for different steering tasks, we might need to keep the values of both components of the fitness of the evolved SAF separately (and to implement a two-objective optimization [31]) instead of fusing both these components into a single scalar value. This would allow us to obtain a set of Pareto-optimal SAFs that feature different combinations of the values of (i) the area under the trajectory of the car and (ii) the average of its lateral velocity. An SAF featuring a wide area under the trajectory might be needed in a slow and comfortable lane change on a low-traffic highway. On the other hand, an SAF that results in somehow oscillating trajectories with higher lateral accelerations might be needed to safely circumnavigate suddenly appearing obstacles. One of the major drawbacks of such a multi-objective approach is that the real-time selection of the appropriate SAF from the set of a priori obtained (valid) SAFs could be rather challenging. For the given task, however, we assumed that the proposed single-valued evaluation of the fitness of the evolved SAF is sufficient.

2.5.3. Genetic Operations

Algorithmically, we implemented the main genetic operations—selection, mutation, and crossover— as follows. We adopted a binary tournament selection—a robust, commonly used selection mechanism, which has proved to be efficient and simple to code—to create the mating pool of the SAF of the next generation. The single-point crossover operation swaps random nodes (and the corresponding subtrees) of the parse trees of two parents (randomly selected from the mating pool). The single point mutation replaces a random node of some of the newly created offspring with a randomly created subtree [29].

The main parameters of the adopted GP [8,32] are elaborated in Table 4, and its seven algorithmic steps are shown in Algorithm 1.

Algorithm 1. The main algorithmic steps of the GP algorithm.

```
Step 1: Creating the initial population of random SAFs;
Step 2: Evaluating the population;
Step 3: While Not (Termination criteria) Do Steps 4, 5, 6, and 7;
Step 4: Selecting the mating pool of SAFs of the next generation;
Step 5: Crossing over random pairs of SAFs of the mating pool;
Step 6: Mutating randomly the newly created offspring SAFs;
Step 7: Evaluating the population;
```

Table 4. Parameters of genetic programming (GP).

Parameter	Value
Evolved individuals	SAF δ
Genetic representation	Parse tree
Set of non-terminals (functions)	{+, -, *, /}
Set of terminals	Variables pertinent to the state of the car, and their derivatives: lateral deviation (e, e', $\int e$), speed (V), steering angle (δ), lateral acceleration (a, a') angular deviation (θ, θ'), and a random constant (C)
Population size	200 individuals
Selection	Binary tournament, ratio 0.1
Elitism	Best 4 individuals
Crossover	Single point, random, ratio 0.9
Mutation	Single point, random, ratio 0.05
Fitness value	Sum of (i) the area under the trajectory of the car around the center of the lane and (ii) the average of its lateral velocity.
Termination criteria	(#Generations >200) *or* (no improvement of fitness during 16 consecutive generations)

3. Experimental Results

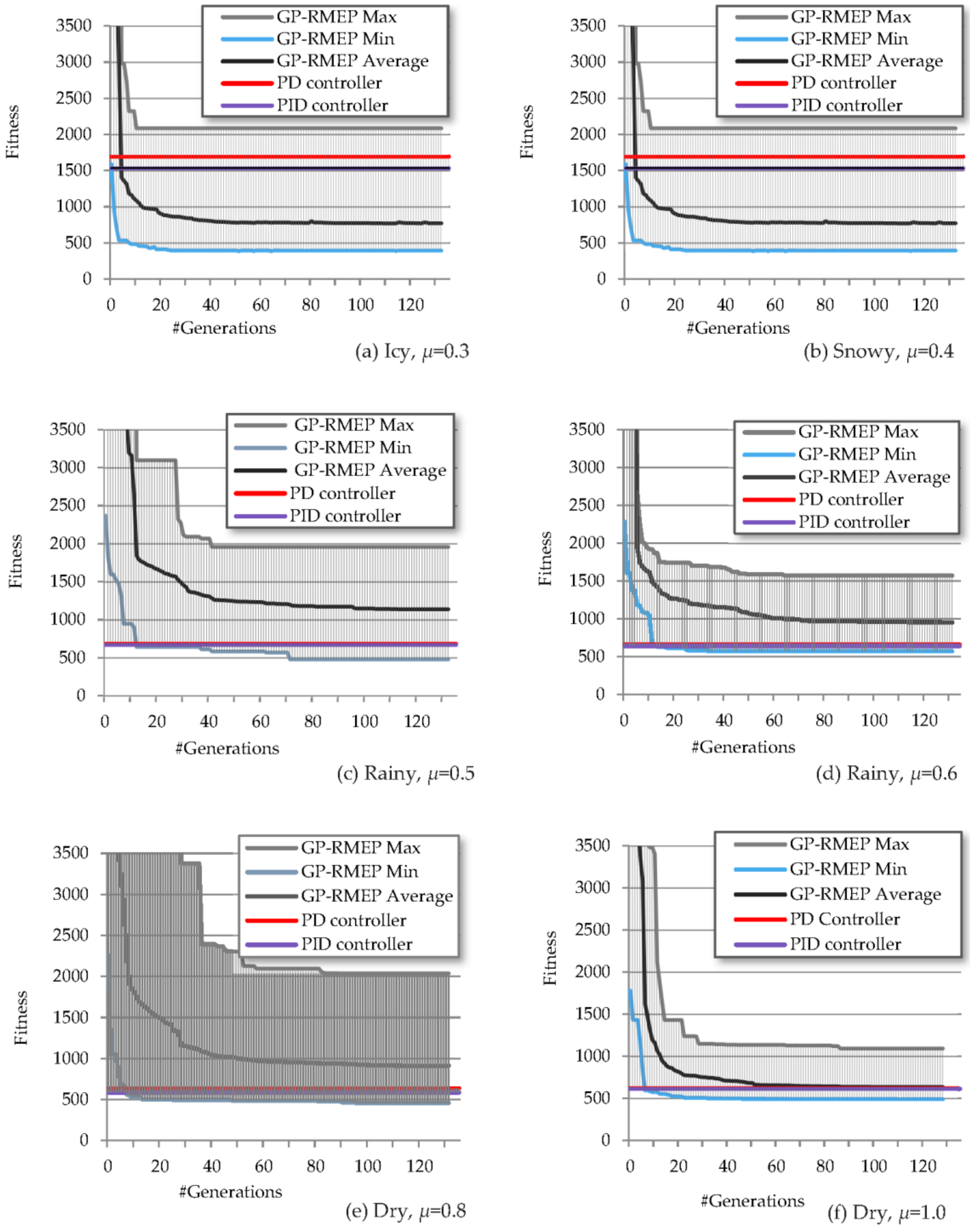
For each of the six road conditions (as described in Table 3) we employed GP and performed 20 independent evolutionary runs to obtain the best SAF. The fitness convergence characteristics of these independent runs are shown in Figure 7.

As Figure 7 illustrates, in all road conditions the fitness of the best-evolved SAF converges to values that are better (i.e., lower) than those of the best PD and PID controllers. We obtained the optimal values of parameters of the best PD and PID controllers by their complete enumeration (i.e., a "brute-force" search). For the PD controller, we evaluated 25 discrete values of each of the two parameters k_1 and $k^*{}_2$ (resulting in a size of the search space equal to $25^2 = 625$). For the PID, we borrowed the optimal values of the two parameters of the PD controller, and used only 10 discrete values in the vicinity of these values and 25 values for the third parameter k_3 (i.e., the size of the search space is 2500). The best fitness of the three proposed controllers—PD, PID, and GP-RMEP—and the optimal values of the coefficients of the PD and PID controllers are shown in Table 5.

Table 5. Best performing steering controllers.

#Road Condition	**Overall Friction μ**	**Controller**				
		PD, $\delta = k_1\, e + k^*_2\, e'$		**PID, $\delta = k_1\, e + k^*_2\, e' + k_3 \int e\, dt$**		**GP-RMEP, an Arbitrary SAF $\delta\, (e, e', \int e, V, \delta, a, a', \theta, \theta', C)$**
		Values of k_1 and k^*_2	**Fitness Value**	**Values of k_1, k^*_2 and k_3**	**Fitness Value**	**Fitness Value**
1	1	0.1145, 3.378	622	0.3079, 1.677, 0.067	511	498
2	0.8	0.271, 1.677	636	0.2472, 2.244, 0.012	526	545
3	0.6	0.2472, 1.866	661	0.229, 2.055, 0.0321	546	430
4	0.5	0.1864, 2.244	687	0.113, 2.433, 0.048	584	373
5	0.4	0.1135, 3.378	765	0.150, 3.189, 0.0013	702	314
6	0.3	0.3322, 2.055	1693	0.1257, 4.512, 0.0415	1212	374

As the results shown in Table 5 demonstrate, the PID controller outperforms the PD controller in that its fitness is lower in all road conditions. However, the quality of steering of the best-evolved GP-RMEP controller is even better than that of PID, and the difference between them widens with the decrease of the friction coefficient, reaching a maximum of about 4 times (1532 versus 374) on icy roads (μ = 0.3). Conversely, on grippy, dry roads (μ = 0.8 and μ = 1.0), this difference is not very significant (587 versus 545 and 613 versus 498, respectively), implying that the PID model provides good enough steering of the car in these road conditions.



(a) Icy, μ=0.3

(b) Snowy, μ=0.4

(c) Rainy, μ=0.5

(d) Rainy, μ=0.6

(e) Dry, μ=0.8

(f) Dry, μ=1.0

Figure 7. Fitness convergence characteristics for 20 independent runs of GP evolving the SAF of the GP-RMEP controller for friction coefficients μ = 0.3 (**a**), μ = 0.4 (**b**), μ = 0.5 (**c**), μ = 0.6 (**d**), μ = 0.8 (**e**), and μ = 1.0 (**f**), respectively. The bold curves correspond to the mean, while the envelope shows the minimum and maximum values in each generation. The best fitness of the PD and PID controllers (obtained via a complete enumeration of the values of their parameters) are shown as horizontal lines.

The lack of generality is one of the well-documented drawbacks of solutions obtained via GP [4,5,26] that hinders the applicability of this algorithm to real-world problems. Indeed, we could not be sure about how well the SAF that was evolved in a single car driven at a fixed speed on a fixed track featuring a fixed coefficient of friction would perform in different situation(s). Ultimately, we should have considered an evolving SAF that performs (nearly) equally well on several fitness cases that correspond to these different conditions. Moreover, in order to bridge the inevitable reality gap, we should have implemented an evolutionary adaptation of a set of the best SAFs, evolved on the simulated car, to a real one driven on a real track. However, in our current, seminal work, we will report the results of testing the best SAF evolved via GP with a single fitness case (fixed track, fixed coefficient of friction, and driven at fixed speed $V = 0.85\ V_{CR}$) in different road conditions. As shown in Figure 8, all GP-RMEP controllers, evolved for a particular slippery condition (e.g., for μ equal to 0.3, 0.4, 0.5, or 0.6, respectively) feature a reasonably small degradation (if any) when tested in different (unforeseen during the evolution) road conditions.

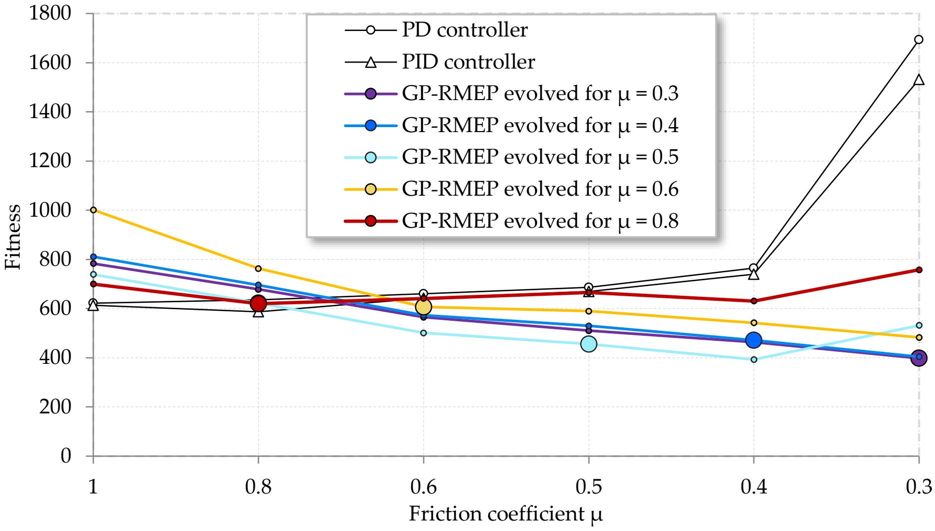


Figure 8. Generality of the best-evolved steering functions manifested by a reasonable change in their respective fitness value when tested on different (unforeseen during the evolution) friction coefficients. The fitness of the steering functions evolved for a particular friction coefficient is indicated by larger circular markers.

For example, the GP-RMEP controller, evolved on snowy roads with $\mu = 0.5$, offers a comparatively good quality of steering (i.e., lower fitness values) compared to the alternative PD and PID controllers when tested on roads with friction μ equal to 0.3 (icy), 0.4 (snowy), and 0.6 (rainy). The analytical expression of the SAF of this GP-RMEP controller is shown in Equation (8).

$$\delta = (V - e' + a + 8e + 14) + \frac{0.571(7 - V)}{\delta\left(-e' + \frac{42}{a'}\right)}[\theta'\left(-\frac{e}{108} + 8\right)(e' + \frac{\theta(\frac{30}{a'} - e)}{7 - V} + 5 - \theta - \frac{30}{\theta a'}) + \frac{3a(1 - 0.2e)}{7eV\left(3a + 1.125 + \frac{7}{8V}\right)}] \tag{8}$$

It is not uncommon that the solutions obtained via GP are way too complex to be easily comprehended by a human [5]. The presented best-evolved SAF is not an exception to this trend, and we cannot explain precisely either why or how the SAF, shown in Equation (8), works. We could only confirm that the SAF implements a proportional-derivative (PD) control of the steering in that both (i) the direct values of parameters and (ii) their derivatives are incorporated in its code. The only integral term included in the terminal set of the GP—the integral of the lateral deviation—is not incorporated by the GP in the best-evolved SAF, which is consonant with the findings that the integral term could be obsolete in some nonlinear PID controllers [27].

The dynamics of the steering angle and the deviation from the center of the lane of the car steered by the sample best-evolved SAF shown in Equation (8) in snowy ($\mu = 0.5$) road conditions are illustrated in Figures 9 and 10, respectively. The same figures also show the behavior of the car steered by the best PD and PID controllers with values of parameters optimized for this particular road condition (as shown in Table 5). As these two figures illustrate, the lateral deviation of the car steered by the sample best-evolved steering function is significantly lower than that of the best solution of the servo-control model, especially during steady-state cornering in the left Turn 1 and right Turn 2. Also, during the transition between these two turns the steering is smoother and more stable (non-oscillatory).

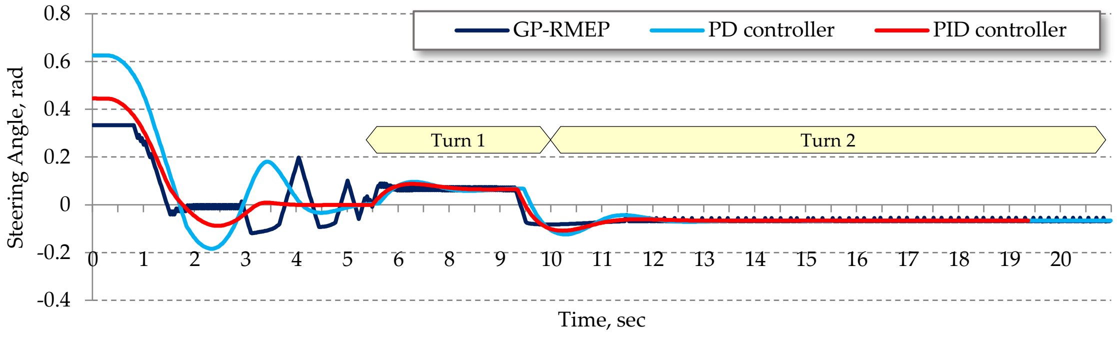


Figure 9. The dynamics of the steering angle of the car steered by the sample best-evolved SAF shown in Equation (8) in snowy (μ = 0.5) road conditions.

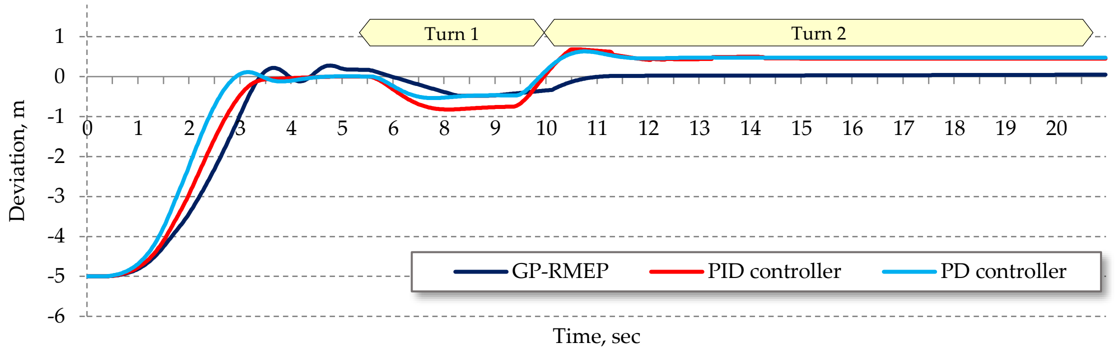


Figure 10. The dynamics of the deviation from the center of the lane of the car steered by the sample best-evolved SAF shown in Equation (8) in snowy (μ = 0.5) road conditions.

As Figure 9 illustrates, the maximum value of the steering angle of the car steered by GP-RMEP during the reaction to the initial step disturbance is about 2 times lower (0.3 rad versus 0.62 rad) than that of the PD and PID controllers. This in turn results in a smoother return of the car to the center of the lane. Moreover, the steering angle produced by GP-RMEP appears to be limited to a particular maximum value of 0.3 rad. A similar phenomenon could be observed during the transitions between Turn 1 and Turn 2, with the only difference that the limit is much lower (−0.06 rad), which facilitates the smooth, non-oscillatory transitions between the turns. We think that the PD and PID controllers could not yield limited values of the steering angle in these two situations because the linearity of these controllers implies that such a limitation would compromise their ability to satisfy the other (somehow contradicting) requirements of the steering controller; namely, to follow the center of the lane closely and to return to the lane swiftly when having deviated from it. We assume that we could borrow this know-how, discovered by GP, to design a PID steering controller with coefficients that adaptively (possibly nonlinearly) vary the gain of the three terms P, I, and D depending on the current driving situation. Moreover, these coefficients, rather than being constants, could be evolved (e.g., via GP) as functions of some parameters pertinent to the state of the car.

4. Discussion

Some of the best SAFs evolved on icy and snowy roads featured slight steering oscillations with frequencies of about 1~5 Hz. We cannot conclude whether these oscillations have a beneficial effect on the controllability of the car. Most likely, they appear as a result of a neutral genetic code in the evolved SAF: a code that has no (neither a beneficial nor a detrimental) effect on the behavior of the car. Indeed, these steering oscillations do not manifest themselves in an oscillating trajectory of the car, because, due to the slippery conditions, the front tires slip excessively and could not provide sharp directional control of the car. Consequently, the oscillations do not result in any measurable oscillations of the lateral acceleration of the car, which was a metric that we used in the calculation of the fitness of the evolved SAF. Therefore, the GP could not impose any selection pressure against the evolved SAF that results in these oscillations. Despite that, the oscillations apparently do not affect detrimentally the steering of the car. From the standpoint of the feasibility of the practical implementation of the evolved SAF on real cars, such oscillations might be highly undesirable due to the uncomfortable vibrations and accelerated wear of the tires and the components of the steering system of the car. We speculate that by modifying the fitness function—e.g., by introducing a third additive component (in addition to the area under the trajectory of the car around the center of the lane and the average of its lateral velocity) that explicitly reflects the severity of steering oscillations—we could discourage the GP from evolving oscillating steering functions.

The realization of steering controllers in the real world could be done in two steps in accordance with the concept of evolutionary robotics [33]. The first step, as discussed in our work, involves the evolution of a generic steering solution on the software model of the car in TORCS. The second stage, intended to bridge the inevitable reality gap, could be implemented as an evolutionary adaptation of the evolved generic solutions to real cars in real-world environments.

Also, in our discussion about the generality of the evolved SAF, we assumed the possibility to deploy a single, general SAF that could be good enough in any (slippery) road condition. The disadvantage of such an approach would be that such a general SAF might be somehow inferior to a dedicated SAF evolved for a particular road condition (as illustrated in Figure 8). On the other hand, the advantage is that the system is not required to determine (in a real time) the current (instant) road conditions, i.e., the current (instant) coefficient of friction between the tires and the road. Our assumption—supported by the fact that most of the existing driving aids activated in slippery road conditions, such as anti-locking brake systems, traction control, and electronic stability programs, rely on the detection of the slippage of the tires rather than on the actual coefficient of friction (one of the underlying reasons for such a slippage)—was that such a determination is rather challenging: it requires a significant computational (signal-processing) power; the obtained result is approximate, and it might be obtained in a limited number of driving situations. However, the recent advances in automotive control suggest that in the near future these challenges might be successfully addressed [34].

5. Conclusions

We investigated the development of controllers for automated steering of a realistically simulated car on slippery roads. We comparatively investigated three models of such a controller—a PD controller built in accordance with the servo-control model of steering, a PID controller as an extension of the servo-control, and a controller based on a relaxed model and an extended set of parameters and designed heuristically via genetic programming. The experimental results verified that the controller developed via GP offers the best quality of control of the car in all of the tested slippery—rainy, snowy, and icy—road conditions. In our future work, we are planning to coevolve—via genetic programming—(i) the automated steering of the car and (ii) the dynamics of the asymmetric brake forces on its wheels (i.e., its electronic stability program) in order to achieve an even better controllability of the car on slippery roads.

Author Contributions: Conceptualization, N.A. and I.T.; Methodology, N.A., I.T.; Software, N.A., I.T.; Validation, I.T. and K.S.; Formal Analysis, N.A., I.T.; Investigation, N.A.; Resources, N.A.; Data Curation, N.A.; Writing-Original Draft Preparation, N.A.; Writing-Review & Editing, N.A., I.T., K.S.; Visualization, N.A.; Supervision, I.T.; Project Administration, I.T.; Funding Acquisition, I.T.

Funding: This work was funded in part by the MEXT-supported Program for Strategic Research Foundation at Private Universities in Japan (2014–2018).

Conflicts of Interest: The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Holland, J. *Adaption in Natural and Artificial Systems*; The University of Michigan Press: Ann Arbor, MI, USA, 1975.
2. Stone, P.; Brooks, R.; Brynjolfsson, E.; Calo, R.; Etzioni, O.; Hager, G.; Hirschberg, J.; Kalyanakrishnan, S.; Kamar, E.; Kraus, S.; et al. *Artificial Intelligence and Life in 2030*; One Hundred Year Study on Artificial Intelligence: Report of the 2015–2016 Study Panel; Stanford University: Stanford, CA, USA, 2016; p. 52.
3. Mousel, T.; Treis, A. Hands off Detection Requirements for UN R79 Regulated Lane Keeping Assis. IEE S.A., Luxembourg. Available online: https://www.unece.org/fileadmin/DAM/tra (accessed on 3 April 2018).
4. Ross, P. IBM Program would Help Robocars Decide Whether to Cede Control. IEEE Spectrum. 2017. Available online: https://spectrum.ieee.org/caronline (accessed on 25 January 2018).
5. Zeeb, K.; Buchner, A.; Schrauf, M. Is take-over Time All that Matters? The Impact of Visual-cognitive Load on Driver Take-over Quality after Conditionally Automated Driving. *Accid. Anal. Prev.* **2016**, *92*, 230–239. [CrossRef] [PubMed]
6. State of California. Department of Motor Vehicles. Report of Traffic Collision Involving an Autonomous Vehicle (OL 316). 2016. Available online: https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/autonomousveh_ol316+ (accessed on 27 January 2018).
7. Lin, P. Who's at Fault in Uber's Fatal Collision? *IEEE Spectrum*. 2018. Available online: https://spectrum.ieee.org/car (accessed on 28 January 2018).
8. Lin, P. Tesla Autopilot Crash: Why We Should Worry about a Single Death. IEEE Spectrum. Available online: https://spectrum.ieee.org/cars-that-think/transportation/self-driving/tesla-autopilot-crash-why-we-should-worry-about-a-single-death (accessed on 11 July 2016).
9. Paden, B.; Čáp, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55. [CrossRef]
10. Pendleton, S.D.; Andersen, H.; Du, X.; Shen, X.; Meghjani, M.; Eng, Y.H.; Rus, D.; Ang, M.H. Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines* **2017**, *5*, 6. [CrossRef]
11. Huang, J.; Tanev, I.; Shimohara, K. Evolving a General Electronic Stability Program for Car Simulated in TORCS. In Proceedings of the 2015 IEEE Conference on Computational Intelligence Games (CIG 2015), Tainan, Taiwan, 31 August–2 September 2015; pp. 446–453.
12. Koza, J.R. Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **1994**, *4*, 87–112. [CrossRef]
13. Gillespie, T.D. *Fundamentals of Vehicle Dynamics*; Society of Automotive Engineers: Warrendale, PA, USA, 1992.
14. TORCS—The Open Racing Car Simulator. Available online: http://sourceforge.net/projects/torcs/ (accessed on 5 May 2018).
15. Wymann, B.; Dimitrakakis, C.; Sumner, A.; Espié, E.; Guionneau, C. TORCS, the Open Racing Car Simulator, v1.3.5. 2013. Available online: http://www.torcs.org (accessed on 1 May 2018).
16. Sallab, A.E.; Abdou, M.; Perot, E.; Yogamani, S. Deep Reinforcement Learning Framework for Autonomous Driving. *Electron. Imaging* **2017**, *2017*, 70–76. [CrossRef]
17. Chi, L.; Mu, Y. Deep Steering: Learning End-to-End Driving Model from Spatial and Temporal Visual Cues. *arXiv* **2017**, arXiv:1708.03798
18. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 566–580. [CrossRef]
19. Carvalho, A.; Gao, Y.; Gray, A.; Tseng, H.E.; Borrelli, F. Predictive Control of an Autonomous Ground Vehicle Using an Iterative Linearization Approach. In Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems-(ITSC), The Hague, The Netherlands, 6–9 October 2013; pp. 2335–2340.
20. Gao, Y.; Lin, T.; Borrelli, F.; Tseng, E.; Hrovat, D. Predictive Control of Autonomous Ground Vehicles with Obstacle Avoidance on Slippery Roads. In Proceedings of the ASME 2010 Dynamic Systems and Control

Conference, Cambridge, MA, USA, 12–15 September 2010; American Society of Mechanical Engineers: New York, NY, USA, 2010; pp. 265–272.

21. Ledin, A. Hardware-in-the-Loop Simulation. *Embed. Syst. Program.* **1999**, *12*, 42–60.
22. Liebemann, E.K.; Meder, K.; Schuh, J.; Nenninger, G. Safety and performance enhancement: The Bosch electronic stability control (ESP). In Proceedings of the 2004 International Congress on Transportation Electronics, Detroit, MI, USA, 18–20 October 2004.
23. Nikulin, V.; Podusenko, A.; Tanev, I.; Shimohara, K. Regression-based Supervised Learning of Autosteering of a Road Car Featuring a Delayed Steering Response. *Int. J. Data Sci. Anal.* **2018**. [CrossRef]
24. Aly, M. Real Time Detection of Lane Markers in Urban Streets. *IEEE Intell. Veh. Symp.* **2008**, 7–12. [CrossRef]
25. Melder, N.; Tomlinson, S. Racing Vehicle Control Systems using PID Controllers. In *Game AI Pro*; pp. 491–500. Available online: http://www.gameaipro.com/GameAIPro/GameAIPro_Chapter40_Racing_Vehicle_Control_Systems_using_PID_Controllers.pdf (accessed on 21 July 2018).
26. Koza, J.R.; Keane, M.A.; Yu, J.; Bennett, F.H., III. Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming. *Genet. Program. Evol. Mach.* **2000**, *1*, 121–164. [CrossRef]
27. Wang, W. The New Design Strategy on PID Controllers. In *PID Controller Design Approaches-Theory, Tuning and Application to Frontier Areas*; InTech: London, UK, 2012.
28. Marinoa, R.; Scalzia, S.; Nettob, M. Nested PID steering control for lane keeping in autonomous vehicles. *Control Eng. Pract.* **2011**, *19*, 1459–1467. [CrossRef]
29. Koza, J.R. *Genetic Programming 2: Automatic Discovery of Reusable Programs*; MIT Press: Cambridge, MA, USA, 1994.
30. Angeline, P.J.; Kinnear, K.E., Jr. Genetic Programming and Emergent Intelligence. In *Advances in Genetic Programming*; MIT Press: Cambridge, MA, USA, 1994; pp. 75–98.
31. Coello, C.; Lamont, G.B.; van Veldhuizen, D. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: New York, NY, USA, 2007.
32. Kinnear, K. Generality and Difficulty in Genetic Programming: Evolving a Sort. In Proceeding of the Fifth International Conference on Genetic Algorithms, San Mateo, CA, USA, 17–21 July 1993; Morgan Kaufmann: Burlington, MA, USA, 1993.
33. Nolfi, S.; Floreano, D. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*; MIT Press: Cambridge, MA, USA, 2000.
34. Beal, C.E. Independent Wheel Effects in Real Time Estimation of Tire-Road Friction Coefficient from Steering Torque. In Proceedings of the 8th IFAC Symposium on Advances in Automotive Control (AAC), Norrköping, Sweden, 20–23 June 2016; pp. 319–326.