Speed Control Optimization for Autonomous Vehicles with Metaheuristics

José Eugenio Naranjo, Francisco Serradilla, Fawzi Nashashibi

Article

Speed Control Optimization for Autonomous Vehicles with Metaheuristics

José Eugenio Naranjo [1,*] , Francisco Serradilla [1] and Fawzi Nashashibi [2]

[1]   INSIA, Artificial Intelligence Department, Universidad Politécnica de Madrid, 28031 Madrid, Spain; francisco.serradilla@upm.es

[2]   National Institute for Research in Computer Science and Automation (INRIA), 75012 Paris, France; fawzi.nashashibi@inria.fr

*   Correspondence: joseeugenio.naranjo@upm.es; Tel.: +34-910-7678-878

check for updates

Abstract: The development of speed controllers under execution in autonomous vehicles within their dynamic driving task (DDT) is a traditional research area from the point of view of control techniques. In this regard, Proportional – Integral – Derivative (PID) controllers are the most widely used in order to meet the requirements of cruise control. However, fine tuning of the parameters associated with this type of controller can be complex, especially if it is intended to optimize them and reduce their characteristic errors. The objective of the work described in this paper is to evaluate the capacity of several metaheuristics for the adjustment of the parameters K_p, $1/T_i$, and $1/T_d$ of a PID controller to regulate the speed of a vehicle. To do this, an adjustment error function has been established from a linear combination of classic estimators of the goodness of the controller, such as overshoot, settling time (t_s), steady-state error (e_{ss}), and the number of changes of sign of the signal (d). The error obtained when applying the controller has also been compared to a computational model of the vehicle after estimating the parameters K_p, K_i, and K_d, both for a setpoint sequence used in the adjustment of the system parameters and for a sequence not used during the adjustment, and therefore unknown by the system. The main novelty of the paper is to propose a new global error function, a function that enables the use of heuristic optimization methods for PID tuning. This optimization has been carried out by using three methods: genetic algorithms (GA), memetics algorithms (MA), and mesh adaptive direct search (MADS). The results of the application of the optimization methods using the proposed metric show that the accuracy of the PID controller is improved, compared with the classical optimization based on classical methods like the integral absolute error (IAE) or similar metrics, reducing oscillatory behaviours as well as minimizing the analysed performance indexes.

Keywords: Autonomous vehicles; genetic algorithms; intelligent control; cruise control

1. Introduction

The cruise control (CC) system is the internal module of the autonomous vehicles in charge of automatically managing the speed. The problem of cruise control has been addressed with classic PID controllers in [1] and [2], and specifically with genetic algorithms in [3]. Indeed, the evolution of CC is the adaptive cruise control (ACC), which extends the capabilities of the speed control to consider the safety headway from the preceding vehicle. Artificial intelligence-based controllers [4] of analytical controllers have been developed for this task [5,6]. However, the basis of those controllers is similar and are subject to further optimization of their parameters.

PID controller optimization has been discussed previously in various papers [7–9]. Then, [7] and [8] proposed the optimization of PIDs considering IAE and ISE as objective function respectively,

generating a good tuning but with oscillatory behaviours. In [9], a simplification of the PID algorithm to ease the tuning and that allows numerical optimization is presented, but only is extensible to simple plants. Specifically, in [10–12], genetic algorithms are used for such optimization, having demonstrated a good fit for a single setpoint and by using the ISE (integral squared error), ITSE (integral time squared error), IAE (integral absolute error), and ITAE (integral time absolute error) as a cost function [13] that represents a different norm of the time varying output error that converges to zero in the event that control is stable. In [14], a comparison of several classic and non-classic methods for the adjustment of PID controllers is made, but always using the integral errors as a cost function. However, those paper focus on the optimization of the PID gains, but always using the ISE /ITSE/IAE/ITAE performance indexes as a cost function, causing oscillatory behaviours in the control signal, as well as in the system output.

Moreover, genetic algorithms (GA) are a well-known optimization technique in contexts in which the calculation of the gradient of the error function is not possible. GA are also appropriate in spaces of optimization where there are multiple local minima, plateaus, ridges or other problematic situations for gradient-descent methods. Since Holland's founding work [15], they have been expanded and improved for multiple categories of problems, including the optimization of float valued parameters, combinatorial optimization, and grammatical evolution among others, and are still an active area of research today.

Memetic algorithms (MA) are an improvement of the GAs in which a local search phase is included within the GA stages. The local search can be done through several procedures, from First Improvement or Best Improvements techniques to the use of gradient descent itself. The GA part avoids the limitations derived from the use of gradient strategies and the MA part makes fine adjustment to the solutions found by GA.

Finally, mesh adaptive direct search (MADS) [16] performs a parameter search based on a mesh of random points from the variable space from which the most promising are selected to perform a local exploration in order to refine the solutions. We will use this technique, which has already demonstrated its optimization capacity, to compare it with the results obtained by the methods proposed in this article.

These three techniques, GA, MA, and MADS, have been applied to many optimization problems in various fields of knowledge. To name but a few, GAs have been applied in the field of medicine [17], in bank lending [18] and in the optimization of industrial processes [19]. MAs have been applied to energy demand estimation [20] and to task planning [21], while MADS was applied to Bayesian optimization [22] and to the design of new materials [23].

The proposals featured in the literature generally use a step response and the ISE, ITSE, IAE and ITAE metrics as cost function. In this paper, we propose a novel cost function that is a combination of four parameters that measure the performance of a controller based on the behaviour observed in its output: overshoot, settling time, steady-state error, and decay ratio instead of a single measure. This complex cost function guarantees a controller tuning without oscillations and with a high performance to obtain more realistic parameters that behave correctly in a wide range of situations and the estimation of the error is made for a sequence unknown to the optimization process, following a cross-validation strategy to ensure that we do not estimate the error optimistically. Using those error estimators as a cost function, the three optimization techniques (GA, MA, and MADS) have been used to optimize the parameters of the controllers, comparing them with the performance of a manually tuned controller.

2. Control Process to Be Optimized

2.1. Definition of a PID Controller

A PID (proportional integrative derivative) controller is a generic control mechanism for closed loop feedback, widely used in the industry for system control. The PID is a system that receives an error calculated from the desired output minus the output obtained and its output is used as input

into the system we want to control. The controller tries to minimize the error by adjusting the system input [24].

The PID controller is determined by three components: the proportional, the integral and the derivative. Each of these components has a greater influence on some characteristic of the output (setling time, overshoot, etc.) but also influences the others. This means that, despite the exhaustive tuning of those parameters, it is impossible to find a PID configuration able to reduce the setling time to 0, the over-oscillation to 0, the error to 0, etc. However, the objective is to adjust the PID parameters following a compromise among them, making adjustments depending on the requirements of the application [25,26]. Finally, there is an anti-windup element to guarantee the maintenance of the control signal within the actuation limits. Specifically, the anti-windup via constrained regulation with the observer's method that discharges the PID controller's internal integrator when the controller hits specified saturation limits has been used [27].

The proportional response is the basis of the three PID control components and, if the other two –integral control and derivative control– are present, these are added to the proportional response. "Proportional" means that the change present in the controller output is a multiple of the percentage of the change in measurement.

This multiple is called the "gain" of the controller. For some controllers, the proportional action is adjusted by means of such gain adjustment, while for others a "proportional band" is used. Both have the same purposes and effects.

$$P_{t+1} = K_p \cdot e(t) + u_b \tag{1}$$

where P_{t+1} is the control proportional action in *t+1*, K_p being the tunable proportional gain and *e(t)* the difference between the setpoint and the plant output in a time *t*. Finally, u_b is a bias or a reset, with a value of zero for simplification.

2.1.1. Integral Action

The integral action gives a response that is proportional to the integral of the error. This action eliminates the steady state error caused by the proportional mode. However, a longer establishment time is obtained, response is slower, and the period of oscillation is longer than in the case of proportional action.

$$I_{t+1} = I_t + \frac{K_p \cdot h}{T_i} \cdot e(t) \tag{2}$$

where I_{t+1} is the value of the integral component in *t+1*, I_t is the value if the integral component in *t*, T_i is the integral gain (generally expressed as the time it must take the control action to reach (equal or repeat) the proportional action), and *h* is the sampling period.

2.1.2. Derivative Action.

The derivative action gives a response proportional to the derivative from the error (error rate of change). Adding this control action to the previous ones reduces the excess of over-oscillations.

$$D_{t+1} = \frac{T_d}{T_d + N \cdot h} \cdot D_t - \frac{K_p \cdot T_d \cdot N}{T_d + N \cdot h} (y_t - y_{t-1}) \tag{3}$$

where D_{t+1} is the value of the derivative component in *t+1*, D_t is the value of the derivative component in *t*, T_d is the derivative gain, *N* is a noise filter, and *h* is the sampling period.

There are several adjustment methods for PID controllers but none of them guarantees finding a PID that makes the system optimum from the point of view of controller performance. Therefore, the most widely used is the Ziegler–Nichols [28] method, which tests PID parameters and depends on the output obtained by varying these parameters. Specifically, the parameters to be adjusted are the Kp, Ti and Td. This tuning method stem from assuming one or more criterion to specify a desired closed loop response. and are designed to optimize the controller for the desired performance criteria.

2.2. Model to be Controlled

For the design and optimization of vehicle speed control, only the throttle pedal action, normalized between 0 and 1, as defined in [29] is observed. Since the objective of this paper is the generation of controller tuning and optimization of methodologies, the philosophy followed is to simplify the motor vehicle speed model as much as possible, and only the action of two forces is considered.

On the one hand, the force that causes the vehicle to advance is the tractive force (K_{pt}), defined as:

$$K_{pt} = a_{max} \cdot W \tag{4}$$

Being a_{max} the maximum acceleration of the vehicle. On the other hand, this force is opposed by Rolling resistance (R_r), defined as:

$$R_r = F_r \cdot W \tag{5}$$

where a friction rate (F_r) of 0.015 N and a vehicle weight (W) of 1400 Kg is assumed. These values are considered valid for family car tyres with inflation pressure close to 179 Kpa, rolling on concrete [30].

Therefore, the acceleration of the model vehicle at time *t* will be equal to:

$$a_t = \frac{K_{pt} - R_r}{W} \tag{6}$$

Therefore, the speed generated at the next instant *t+1*, considering that the accelerator generates a normalized setpoint between 0 and 1 (u_t), measured in an increase in time Δt will be equal to:

$$v_{t+1} = u_t \cdot a_t \cdot \Delta t + v_t \tag{7}$$

Figure 1 shows the block diagram and signals of the control loop designed to support the system to be controlled.
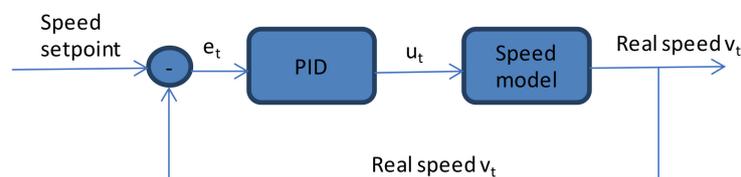

Figure 1. Block diagram of the system to be controlled.

To avoid problems in using a discrete controller with a continuous system a system sampling rate with much higher frequency than Shannon's theorem recommends has been used. The theoretical system bandwidth is below 10^{-3} Hz and the sampling frequency used is 10 Hz, which is well above twice the frequency of the theoretical system bandwidth, which prevents problems of aliasing. This means that the discrete system works practically as the equivalent continuous. In this case, the empirical bandwidth considered to guarantee the response of the vehicle must be much higher than the theoretical, usually between 10 Hz to 100 Hz.

2.3. Calculation of the Error of a PID Controller

To measure the error made by a PID controller, it is necessary to define a function that combines four parameters to be taken into account to measure the performance of a controller based on the behaviour observed in its output: overshoot (o), settling time (ts), the steady-state error (ess) and the decay ratio (d). Depending on the control application, requirements on setpoint follow the consideration of those performance indexes, which are defined in different ways, and there are also different standards for their definition. Next, each of them is described, as well as the measurement function used [31].

2.3.1. Overshoot

The overshoot (o) is the ratio between the difference of the controller output with respect to the setpoint steady-state value, once the reference value has been reached. In some industrial control applications, it is common to specify overshoots of 8–10%. However, in general, the goal of good control tuning is to have an overdamped response with no overshoot or maintaining it as small as possible. In order to measure the degree of overshoot the Equation (8) is used, being signal(t) is the system output and target is the desired output value.

$$o = \begin{cases} \max(signal(t)) - target, \text{when } e(t_0) > 0 \\ target - \min(signal(t)), \text{when } e(t_0) < 0 \end{cases} \tag{8}$$

2.3.2. Settling Time

Settling time is the time taken to reach a stable signal value, remaining within p% of its steady state value. This steady state stable value is understood as the one in which the signal varies less than p% from the previous value. In the case of this paper, p = 0.02% and t_s is defined in Equation (9) and is measured as the ratio between the iteration in which this occurs and the total number of iterations.

$$ts = \frac{t}{T_{max}} \ when \left| (signal(t) - signal(t-1) \right| < 0.0002 \tag{9}$$

where t is the time in which the condition is true, T_{max} the max sampling time, and signal is the speed value.

2.3.3. Steady-State Error

The steady-state error (ess) is the value of the controller error in a steady-state, in other words, the difference between the value of the signal after a stabilization time with respect to the setpoint within N iterations of the controller. In order to define an absolute criterion to compare the different controller tuning performance, this paper defines the closed loops of the controller as N = 350 as shown in Equation (10).

$$ess = \left| signal(N) - target \right| \tag{10}$$

where signal(N) is the system output value at the last control loop and target is the desired output value.

2.3.4. Decay Ratio

The decay ratio (d) is the ratio between two consecutive maxima of the error for a step change in setpoint or load. A typical case of a poorly adjusted PID controller is one in which the setpoint is reached and maintained for a while, but the control signal undergoes high frequency oscillations around that setpoint that cannot be minimized by the integral component. To avoid this case, these high frequency oscillations must be introduced as a parameter to be optimized to obtain a tuning of the controller that minimizes them. The decay, expressed as the ratio by which the oscillation is reduced during one complete cycle, or the ratio of successive peak heights, is represented in Equation (11).

$$d = \frac{c}{a} \tag{11}$$

where a is the coefficient, the amplitude of the oscillations in t−1 and the coefficient c, the amplitude of oscillation in t.

2.3.5. Global Error Function

In order to carry out an optimization of the definition of the gains that make up a PID controller, a function that considers the four performance indexes that have been defined previously must be developed. This novel function is defined in Equation (12) and assigns each of these performance

indexes a weight obtaining the error for this step. The global error is obtained by averaging the error of all the steps of the sequence for each instance, being *N* the number of speed steps in the sequence.

$$G_{error} = \sum_{i=1}^{N} \alpha \cdot o_i + \beta \cdot ts_i + \gamma \cdot ess_i + \delta \cdot d_i \tag{12}$$

The main novelty of this function is that it considers two aspects not sufficiently addressed when using traditional control performance criteria like IAE or similar: the consideration of the overshoot and decay ratio as a built-on parameter to be optimized. These two parameters are not minimized sufficiently when using traditional control performance metrics.

After analysing experimentally, the results by applying the controller to various examples, we have adjusted the parameters α, β, γ and δ with the following values: $\alpha = 3.0$, $\beta = 15.0$, $\gamma = 5.0$ and $\delta = 0.04$. These parameters have been chosen to ensure good control behaviour resulting from the optimization and to ensure a similar influence of the four measured metrics, since the range of variation of each one is different. In addition, they have been obtained experimentally by generating responses from the controllers to the parameters adjusted with different values, and have reached this solution, which is considered the nearest to the optimum.

These parameters regulate the relative importance given to each desirable aspect in a good performance control schema, since some of them may lead to opposite actions and it is necessary to decide whether we prefer, for example, a shorter stabilization time over a lower overshoot. In Section 4 Discussion, the effect of the modification of any of these parameters with a special impact on the result of the adjustment is studied.

It is important to highlight that for optimization and for the estimation of the error it is not enough to use a single step response. That would result in a controller that has been excessively adjusted for a specific situation, when the objective is a controller that behaves correctly in a wide variety of situations. To achieve this end, we have provided the optimizer with a sequence of 30 step-response setpoints for the adjustment of the parameters and a different sequence for evaluation of the error.

The error of a sequence of step-responses will then be calculated as a result of applying the controller to be optimized within the simulation of the model to obtain a vector of parameters that includes the vehicle speed over time, as well as obtaining the values of the four performance indexes, o, ts, ess, and d. This process is carried out for each step-response from the final speed of each step (Figure 2) and the following setpoint.
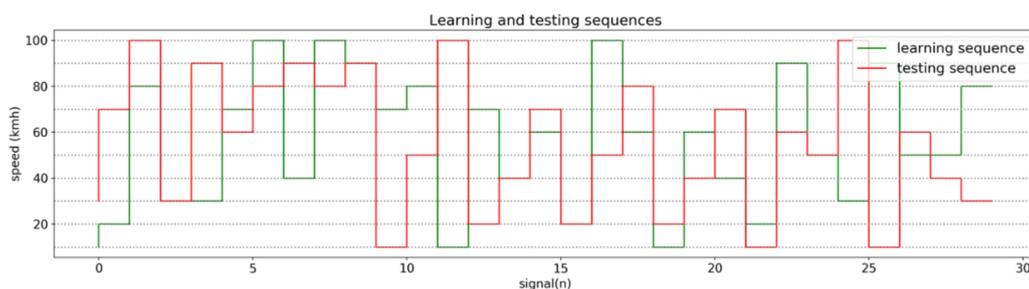

Figure 2. Learning and testing sequences.

3. Optimization of the Controller

Optimization encompasses a set of techniques that deals with finding the maximum or minimum of a function with one or several variables subject to a series of constraints on their values. For this purpose, there are many methods, both when the model is known (simplex, gradient descent and others) and when it is unknown (black-box methods). Genetics algorithms (GA), memetics algorithms (MA), and mesh adaptive direct search (MADS) are black-box methods. Thus, they do not require a detailed description of the models and are more robust against non-continuous or non-derivable

systems, or even systems that are not representable through equations, such as complex industrial algorithms or processes whose modelling would be unapproachable. The three black-box methods have been selected to carry out the optimization of the autonomous vehicle speed PID controller. For the first two, we will use our own implementation, while for the third, we will use the NOMAD implementation [32].

In our case, the variable to be minimized is the error obtained from the combination of ts, d, o and ess from the value of the three parameters Kp, 1/Ti and 1/Td, which are subject to the restrictions of being normalized within the range [0,3]. As the methods studied are black-box, no knowledge of the underlying model is required, so the same approach is applicable to any PID controller able to control any process with the sole requirement of having of an algorithm that simulates the process. In the case of a road vehicle, this simulation is made using the speed model of a vehicle proposed in Section 2.2.

In control theory literature, the problem of optimization of PIDs using standard IAE (integral absolute error) or similar indexes is considered a convex optimization problem. However, in the case of this paper, a combined optimization using four parameters and a sequence of steps, instead of a single step response, is considered. In consequence, in this case, is not assured that the PID optimization can be considered as a convex optimization problem [24–26]. This fact supports the use of meta-heuristics because they can scale possible local minima in the search space.

3.1. Genetic Algorithms (GA)

A genetic algorithm is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection [33]. GA can be considered as a multiagent system with distributed strategy where each agent competes to find the best solution to the problem. Then, the GA does not require to communicate information between them. The main algorithm evaluates the cost function for each agent and selects it to participate in the evolution proportionally.

This optimization method considers, initially, a seed with a set of random solutions that are improved through a sequential operation of selection, crossover and mutation processes by obtaining individuals (candidate solutions, or chromosomes) that will be refined in an iterative process, guaranteeing an individual has a greater probability of being selected when their fitness function is greater. The selection stage identifies the best individuals (highest value of the fitness function) to be evolved. Crossover combines information from two or more of the selected individuals to obtain new solutions to the problem. Finally, the mutation alters some of the crossed values of the new individuals obtained in order to favour the diversity of the solutions.

In our case, the solutions, represented by individuals, are formed by the coefficients Kp, 1/Ti and 1/Td. To maintain the values of the three coefficients within the same range, the fitness function, or degree of goodness of a solution, is calculated in Equation (13) using the error from Equation (12), taking into account that *target* is each speed step of the sequence and $error_{target}$ is the error of the controller for this speed step.

$$fitness = \sum_{\text{target}} \frac{1}{1 + error_{\text{target}}} \tag{13}$$

There is a set of meta-parameters that must be defined in the setup of the genetic algorithm in order to tune the optimization process. Those parameters are related with the performance of the optimization process but are independent of the system to be optimized. The meta-parameters are the mutation probability (Pmut), which indicates the ratio of random change of the individual genes in the mutation stage; the crossover probability (Pcross), which represents whether the selected chromosomes interchange their genes; the size of the population (Tpop), which is the number of chromosomes to be evolved in the genetic algorithm process; the number of individuals selected in the tournament selection stage (T); and the number of generations to be evolved to complete the optimization (N).

In particular, the meta-parameters set up to solve the problem of optimizing the PID configuration of a speed controller of an autonomous vehicle are: Pmut = 0.3; Pcross = 0.7; Tpop = 100; T = 4; N = 300 generations. If the population model used is generational, for the crossover the algorithm BLX-α [34] is used and, for the mutation process, a random value obtained following a Gaussian distribution of mean 0 and decreasing variance over the generations is added to the current value of the gene.

3.2. Memetic Algorithms (MA)

A memetic algorithm is basically a genetic algorithm in which, after selection, crossover and mutation phases, a local optimization phase is added [35]. In our case we use a local optimization by gradient descent with an adaptation of the RPROP algorithm [36].

The advantage of introducing a local search phase in genetic algorithms is that they facilitate convergence to an optimal solution in the vicinity of the solutions provided by the crossover and mutation processes. This is especially important in numerical optimization problems, as is our case.

In this way, the traditional GA phase is responsible for exploring in parallel a set of candidate solutions and the local search phase improves these solutions by making a fine adjustment. For this reason, in general, the MA need fewer iterations to converge to the optimal solution and usually find somewhat better solutions than traditional GA.

3.3. Mesh Adaptive Direct Search (MADS)

As a third optimization method, we have used the MADS algorithm [16] implemented in the NOMAD package [35]. This algorithm is based on a mesh search in which the input space is discretized for the rapid detection of areas of interest for optimization (SEARCH phase). Subsequently, a POLL phase is carried out in which a local search is applied to the most promising points to refine the solution.

The interest of introducing this algorithm in comparisons is that it is an algorithm widely used in optimization processes. It is available in the majority of standard software packages (for example Matlab) and has demonstrated a better optimization capacity compared to traditional methods, especially when using black-box models. In addition, the NOMAD package is easily linkable with algorithms pre-programmed in C++ or Python.

4. Optimization of the Speed PID Controller

In order to carry out the optimization of the autonomous vehicles speed PID controller, two random step-response setpoints have been generated. The first is used to conduct the optimization process and the second is used to analyze the results of the optimized controllers in a cross-validation strategy. For each optimization method to be compared, we have performed the optimization with the first setpoint step-response sequence and subsequently evaluated its error with the second setpoint sequence. Table 1 shows the results obtained after 300 iterations of the GA, MA and MADS optimization algorithms, and compared with the result of a PID controller optimized using IAE as optimization criteria. We have used the step-response sequence represented in Figure 2, showing in green the sequence used for the training of the optimization algorithms and in red the sequence used for the testing and comparison of the results. Those results are graphically represented in Figures 3 and 4.

$$IAE = \int_{0}^{\infty} |e(t)| dt \qquad (14)$$

As ground truth to compare our results, an optimization based on the IAE (Equation (14)), a controller performance index widely used in literature [30], has been performed. Figure 3 shows the graph resulting from applying the parameters Kp, 1/Ti and 1/Td obtained by minimizing the IAE and in the Table 1 are presented the values of the global error function for these parameters. The results show that the use of IAE performance index as metric for the optimization gives a rather high error in

our metric (global error), caused by an oscillatory effect around the setpoint. Traditionally this problem is solved by applying a low pass filter to the controller output, but the optimization strategy using our metric eliminates this need. However, in order to compare the results of the different optimizations in a realistic way, a simple moving average (SMA) low-pass filter with three periods of memory has been added at the output of the system optimized with IAE and those results have been compared with the previous optimizations.

For the graphical representation of the parameters in Table 1, the result of the optimization carried out by the IAE is shown in the Figure 3. In this figure is represented the output of the system in terms of speed and the value of the control signal, normalized in the [0,100] interval, considering the case of the controller parameters optimized using IAE with raw output (a) and the controller optimized with the IAE with a low pass filter at the output of the signal (b). This figure shows that the results are clearly different, both resulting with oscillatory control actions, extreme in the case of the unfiltered controller and softer in the case of the filtered one, less efficient than the controller optimized through metaheuristic techniques, and with some delays in the response. In this case, using the IAE as optimization index is impossible to assure the stability in the tuning optimization procedure, instead of the optimization with the new global error function that avoids instabilities due the selection of its parameters.

Table 1. Comparison of errors of the selected optimization methods.

|	Kp	1/Ti	1/Td	Optimization Process Error	Cross-Validation Error	ts (s)	d (Decay Ratio)	o (kmh)	ess (kmh)	IAE
GA	0.081	0.0111	1.981	1.873	3.321	0.195	4.8	0.267	1.874	5.453
MA	0.083	0.0104	1.928	1.865	3.265	0.180	6.0	0.248	1.873	5.442
MADS	0.058	0.0114	2.870	1.919	3.304	0.166	3.267	1.289	1.947	5.531
IAE	0.511	0.001	2.609	15.106	15.493	0.547	115.159	1.331	0.741	4.192
Filt. IAE	0.511	0.001	2.609	3.7412	5.1618	0.357	5.574	0.075	2.567	11.038
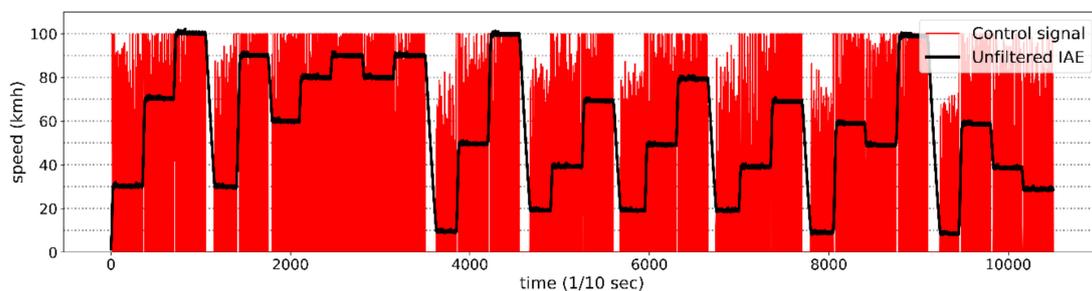
(a)
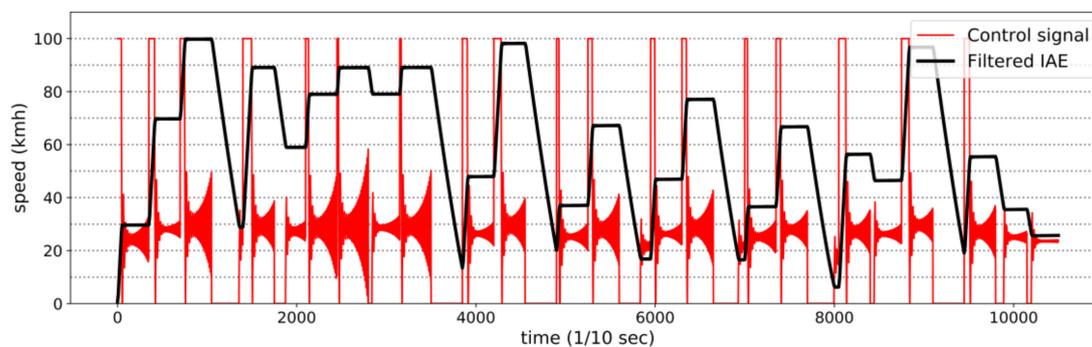
(b)

Figure 3. Representation of the performance of IAE optimized controller, with the system output unfiltered (**a**) and filtered (**b**).

(a)

(b)

(c)
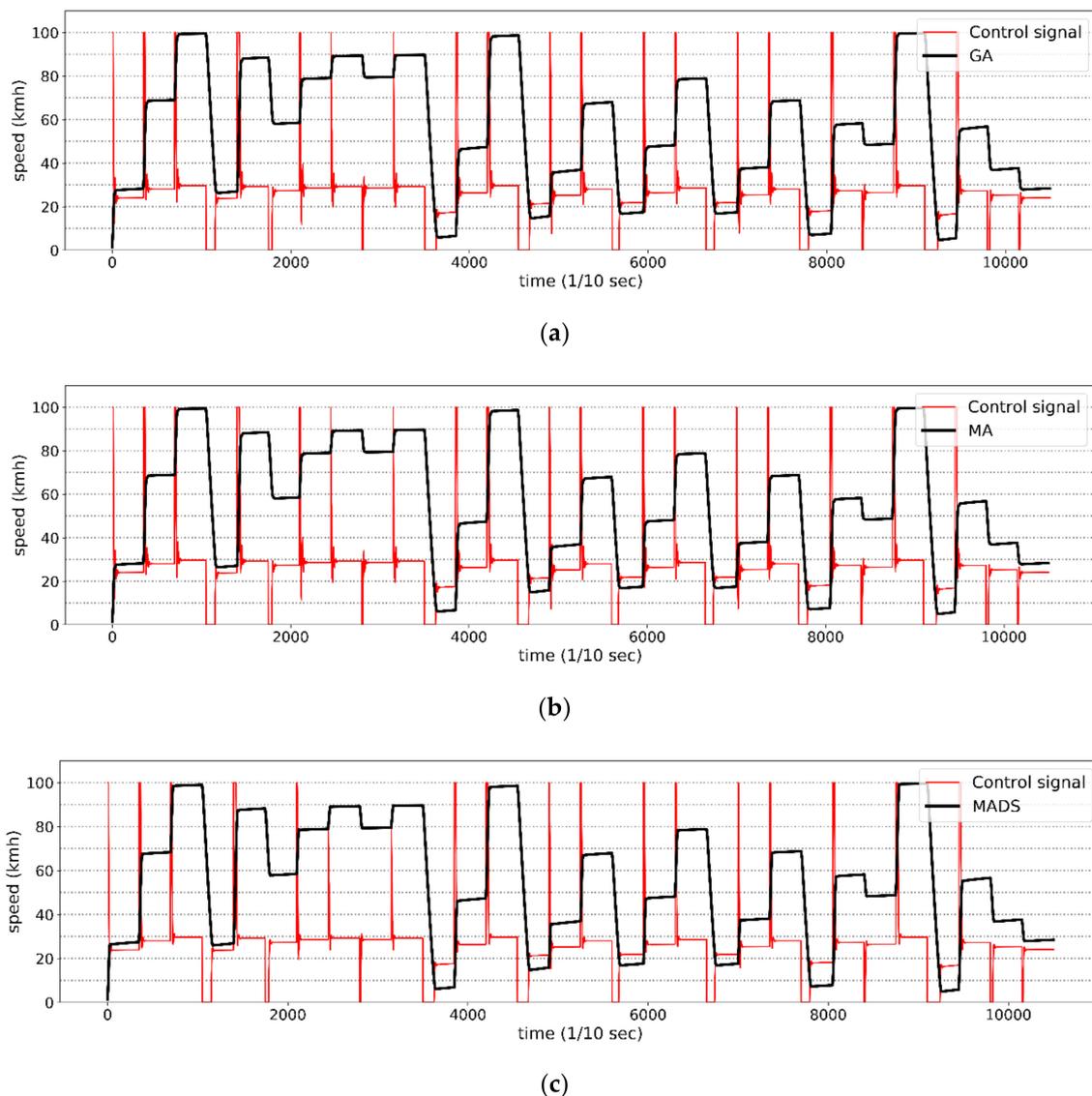
Figure 4. Representation of the performance of GA (**a**), MA (**b**) and MADS (**c**) optimized controllers.

Figure 4 shows the comparison between the PID controller with parameters optimized by GA, MA and MADS, representing the resulting speed and the value of the control signal. In this case, the results are very similar as indicated by the parameters used to calculate the global error of the controller, specified in Table 1. The lower value of the global error function shows the better performance of the MA (3.265) over the MADS (3.304) and GA (3.321). The three axes of the figure are quite similar, meaning that all the methods converge to the same optimum. Anyway, there is a clear correspondence between the overshoot of the control signal, being higher in the MADS, mainly in the speed reduction.

Figure 5 also represents a detail of the performance of MA in a single step response.

Finally, Figure 6 shows the comparison between PID controllers with optimized parameters using the MA and MADS algorithms. As shown in Table 1, the global errors derived from the optimization of the controller parameters using these methods are very similar, since the MADS is the second optimization method with a lower value in this parameter. There are practically no differences between the performance of both controllers, which means that, on the one hand, the graphical representations of Figure 6 overlap and, on the other, that the optimization process has found the best values for the adjustment of the PID parameters, which means that the optimization method can be considered as concluded.
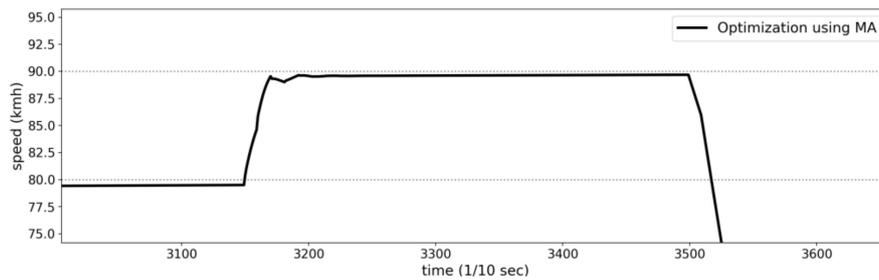
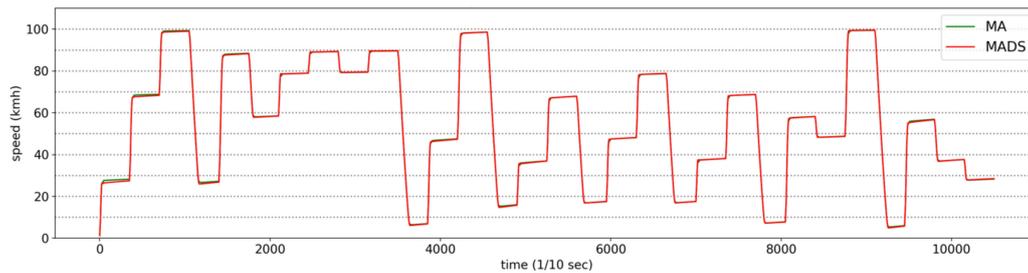**Figure 5.** Detail of the performance of MA in one step response.

Figure 6. Representation of the performance of MADS versus MA optimized controllers.

Additionally, signal disturbance tests have been carried out in order to analyze the proposed controllers to study the robustness as well as the changes in the parameters of the model to test stability of the controller in its complete range of actuation.

Table 2 represents the result of the robustness analysis for the case of a disturbance in the reference signal of a 0.1% with random noise. This percent has been chosen from the speed sensor maximum error specification of the selected model.

Table 2. Signal disturbance test results.

	Cross-Validation Error	ts (s)	d (Decay Ratio)	o (kmh)	ess (kmh)	IAE
GA	3.3944	0.2030	2.5906	0.2704	1.7271	5.4625
MA	3.3736	0.2084	2.6414	0.2499	1.732	5.4345
MADS	3.4230	0.1899	2.5420	0.2716	1.8053	5.5235
IAE	23.7196	0.6275	200.8048	0.9417	0.6916	4.2138
Filt. IAE	5.3917	0.2104	16.1354	0.098	2.5798	11.038

As shown in the table, the results in disturbance situations are very similar to the ideal case. This fact means that the designed controllers are robust under disturbance.

Regarding the analysis of the stability of the system, the model has been modified by selecting the maximum value of weight of the vehicle (2000 kg) and the results have been compared with the ideal case initially selected (1400 kg). The results of this analysis have been represented in Table 3.

Table 3. Sensitivity analysis of the model against changes in the weight of the vehicle.

	Cross-Validation Error	ts (s)	d (Decay Ratio)	o (kmh)	ess (kmh)	IAE
GA	3.1666	0.2218	0.1203	0.2275	1.722	5.5649
MA	3.1524	0.2196	0.1279	0.2082	1.7279	5.5582
MADS	3.2197	0.2111	0.0009	0.2242	1.8078	1.8077
IAE	24.6843	0.744	203.8431	0.8666	0.6880	4.3846
Filt. IAE	4.5715	0.4	0.43596	0.0370	2.40	11.644

Those results show that the system cross-validation error remains in the same range than results of the controllers previously trained, independently of the modification of the parameters of the model. The only value strongly affected by the increasing of the weight of the vehicle is the decay ratio,

drastically reduced in the GA, MA, and MADS cases. This reduction is caused by the subsequent reduction of the acceleration rate of the car, making the model less reactive to the control signals.

Discussion

In this paper, a methodology for the adjustment of PID controllers has been presented, based on (1) a new proposal of error metric as a linear combination of four classic estimators of the goodness of the PID controllers, (2) the tuning of the parameters based on sequences of step-responses versus evaluation with a single setpoint, which makes the tuning more robust to new situations, and (3) the measurement of the error with a new sequence, different from that used in the tuning, which makes the expected error more realistic in operation.

Under those premises, the parameter sets for a cruise control PID controller for an autonomous vehicle with GA, MA, and MADS techniques have been obtained. All of them improve the ground truth controller, designed using the IAE performance index as criteria, as well as with a version of this controller with the output filtered in order to reduce oscillatory behaviours.

In the experimental optimization, the memetic algorithm obtains the best results of the three methods studied, improving the error of the ground truth by 12%. The inferior performance of GA and MADS may be due to the fact that the search space for the best parameters for PID controllers seems to be dominated by a multitude of local minima, so local search processes lead to premature convergence and facilitate a decline in these minima. Specifically, the MADS algorithm converges in some executions to a different optimum, which seems to support the theory of multiple local minima. The memetic algorithm, however, is able to find the global or quasi-global minimum thanks to its ability to explore multiple solutions in parallel. In any case, the differences between all the controllers are small, and they make for a good selection of the best controller parameters.

During the heuristic adjustment of the weight of the four error measurement parameters, i.e., α, β, γ, and δ, the enormous importance of their selected values, depending on the statistical result obtained in the optimization, has been verified, since the optimization methods will try to find the solution with least possible error and this optimization depends on how this error is defined. If the error is not well defined, cases appear such as the one stated by the Goodhart law [37], in which optimization is achieved successfully but the result, although the error is low, is not the desired one. Table 1 shows that none of the algorithms exceeds all others in the four parameters measured on the evolution of the signal. Although the MA is the one that provides the best global optimization and the best optimizations for overshoot and ess, it obtains third place in t_s and d.

In particular, very good solutions can be found regardless of the minimization of the number of the decay ratio (which seems to be opposed to the other three error estimators), obtaining graphs like those in Figure 7 in which they can provide excellent *ess*, overshoot and t_s values at the expense of an oscillatory control signal. These results are similar to those obtained with the traditional metrics AAE, IAE, ITAE and ISE, which, by minimizing the quadratic error or its variants, despite small oscillations at the cost of a shorter stabilization time, weighs much more on these error metrics.
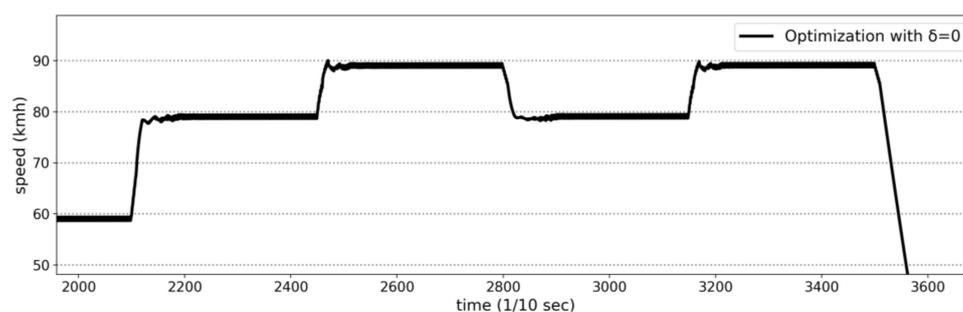
Figure 7. Detail of some step responses disabling the decay ratio *d* ($\delta = 0$), clearly showing the oscillations in the controller performance.

Finally, in reference to the computational cost of the optimization with the three proposed methods, Table 4 shows the average results, considering a computer with an Intel I9 CPU (6 double core) and 16 GB of RAM.

Table 4. Computational cost of the optimization.

	Average Time/Generation(s)	Average Generations	Total Time(s)
GA	1.18	20	23.6
MA	1.72	10	17.2
MADS	48.67	-	48.67

Clearly, the computational cost is very similar in all methods, mainly due the high-end hardware.

5. Conclusions

This paper presents an evaluation of the capacity of several metaheuristics for the adjustment of the parameters Kp, Ki, and Kd of a PID controller for the regulation of the speed of an autonomous vehicle. For this purpose, different adjustment parameters have been established based on a linear combination of classic estimators of the goodness of the controller, such as overshoot, settling time, steady state error and the decay ratio. These parameters have been compared with those obtained by applying the controller to a computer model of the vehicle speed after estimating the parameters Kp, Ki and Kd, both for a setpoint step-response sequence used in the tuning of the system parameters (training) and for a different sequence (testing) and therefore unknown by the system, all implemented in simulation using a simple dynamic model of the speed behaviour of a vehicle. As a result of these simulations, a measure of the improvement of each metaheuristic has been carried out with respect to the use of classical performance indexes like IAE. It has been shown that all the techniques improve the adjustments made by optimizing IAE, and that memetic algorithms are those that generate an optimal adjustment of the PID parameters.

The main novelty of the paper is not only the new weighted global function definition, but the use of this function to finely tuning the gains of a PID controller using meta-heuristics optimization techniques, comparing the performance of those methods with the classical optimization approaches, based on IAE or similar performance criteria. The only similar approach in literature is the optimization using the IAE or similar function as optimization criteria using generic algorithms [10–12], not memetics or MADS. The results of those optimization using IAE or similar generate controllers with more oscillations, as shown in the comparison used as ground truth in this paper. However, those oscillations can be avoided using low pass filters, but then the IAE or similar performance criteria are always worse than the alternative optimization through meta-heuristics.

As part of a future study, we are considering the possibility of adjusting the α, β, γ, and δ coefficients using metaheuristics, and analyzing whether it is possible to improve control through multiple PIDs that are automatically adjusted with the same methodology followed in this article.

Although the work related in this paper has been carried out in simulation, as future work, it is proposed to test the controllers using physical autonomous vehicles as well as to conduct an online fine tuning of the parameters using the same techniques.

References

1. Shaout, A.; Jarrah, M.A. Cruise control technology review. *Comput. Electr. Eng.* **1997**, *23*, 259–271. [CrossRef]
2. Osman, K.; Rahmat, M.F.; Ahmad, M.A. Modelling and controller design for a cruise control system. In Proceedings of the 5th International Colloquium on Signal Processing & Its Applications, Kuala Lumpur, Malaysia, 6–8 March 2009; pp. 254–258.
3. Rout, M.K.; Sain, D.; Swain, S.K.; Mishra, S.K. PID controller design for cruise control system using genetic algorithm. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 4170–4174.
4. Naranjo, J.E.; González, C.; García, R.; de Pedro, T. ACC+Stop&Go maneuvers with throttle and brake fuzzy control. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 213–225.
5. Bageshwar, V.L.; Garrard, W.L.; Rajamani, R. Model predictive control of transitional maneuvers for adaptive cruise control vehicles. *IEEE Trans. Veh. Technol.* **2004**, *53*, 1573–1585. [CrossRef]
6. Corona, D.; De Schutter, B. Adaptive cruise control for a SMART car: A comparison benchmark for MPC-PWA control methods. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 365–372. [CrossRef]
7. Sekara, T.B.; Matausek, M.R. Optimization of PID controller based on maximization of the proportional gain under constraints on robustness and sensitivity to measurement noise. *IEEE Trans. Autom. Control* **2009**, *54*, 184–189. [CrossRef]
8. Killingsworth, N.J.; Krstic, M. PID tuning using extremum seeking: Online, model-free performance optimization. *IEEE Control Syst. Mag.* **2006**, *26*, 70–79.
9. Toscano, R. A simple robust PI/PID controller design via numerical optimization approach. *J. Process Control* **2005**, *15*, 81–88. [CrossRef]
10. Wang, P.; Kwok, D.P. Optimal design of PID process controllers based on genetic algorithms. *Control Eng. Pract.* **1994**, *2*, 641–648. [CrossRef]
11. Chang, W.D. A multi-crossover genetic approach to multivariable PID controllers tuning. *Expert Syst. Appl.* **2007**, *33*, 620–626. [CrossRef]
12. Samakwong, T.; Assawinchaichote, W. PID controller design for electro-hydraulic servo valve system with genetic algorithm. *Procedia Comput. Sci.* **2016**, *86*, 91–94. [CrossRef]
13. Özdemir, M.T.; Özturk, D. Comparative performance analysis of optimal pid parameters tuning based on the optics inspired optimization methods for automatic generation control. *Energies* **2017**, *10*, 2134. [CrossRef]
14. Ribeiro, J.M.; Santos, M.F.; Carmo, M.J.; Silva, M.F. Comparison of PID controller tuning methods: Analytical/classical techniques versus optimization algorithms. In Proceedings of the 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 28–31 May 2017; pp. 533–538.
15. Holland, J. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1975.
16. Audet, C.; Dennis, J.E., Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.* **2006**, *17*, 188–217. [CrossRef]
17. Ghaheri, A.; Shoar, S.; Naderan, M.; Hoseini, S.S. The applications of genetic algorithms in medicine. *Oman Med. J.* **2015**, *30*, 406. [CrossRef] [PubMed]
18. Metawa, N.; Hassan, M.K.; Elhoseny, M. Genetic algorithm based model for optimizing bank lending decisions. *Expert Syst. Appl.* **2017**, *80*, 75–82. [CrossRef]
19. Mariajayaprakash, A.; Senthilvelan, T.; Gnanadass, R. Optimization of process parameters through fuzzy logic and genetic algorithm—A case study in a process industry. *Appl. Soft Comput.* **2015**, *30*, 94–103. [CrossRef]
20. Gong, X.; Liu, Y.; Lohse, N.; De Pessemier, T.; Martens, L.; Joseph, W. Energy-and labor-aware production scheduling for industrial demand response using adaptive multiobjective memetic algorithm. *IEEE Trans. Ind. Inform.* **2018**, *15*, 942–953. [CrossRef]
21. Wang, X.; Tang, L. A machine-learning based memetic algorithm for the multi-objective permutation flowshop scheduling problem. *Comput. Oper. Res.* **2017**, *79*, 60–77. [CrossRef]
22. Acerbi, L.; Ji, W. *Practical Bayesian optimization for model fitting with Bayesian adaptive direct search, In Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 1836–1846.

23. Gheribi, A.E.; Harvey, J.P.; Bélisle, E.; Robelin, C.; Chartrand, P.; Pelton, A.D.; Le Digabel, S. Use of a biobjective direct search algorithm in the process design of material science applications. *Optim. Eng.* **2016**, *17*, 27–45. [CrossRef]

24. Boyd, S.; Hast, M.; Åström, K.J. MIMO PID tuning via iterated LMI restriction. *Int. J. Robust Nonlinear Control* **2015**, *26*, 1718–1731. [CrossRef]

25. Hast, M.; Åström, K.J.; Bernhardsson, B.; Boyd, S. PID design by convex-concave optimization. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 4460–4465.

26. Mercader, P.; Åström, K.J.; Baños, A.; Hägglund, T. Robust PID design based on QFT and convex–concave optimization. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 441–452. [CrossRef]

27. Shamma, J.S. Anti-windup via constrained regulation with observers. *Syst. Control Lett.* **2000**, *40*, 261–268. [CrossRef]

28. Ziegler, J.G.; Nichols, N.B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, *64*, 759–768. [CrossRef]

29. Aparicio, F.; Jiménez, F.; Sánchez, J. Development and use of vehicle dynamics simulation software as support for Road Vehicles Theory teaching. *Comput. Appl. Eng. Educ.* **2009**, *17*, 467–478. [CrossRef]

30. Aparicio, F.; Vera, C.; Díaz, V. *Teoría de Vehículos Automóviles, Escuela Técnica Superior de Ingenieros Industriales*; UPM: Selangor, Malaysia, 2001.

31. Astrom, K.J.; Hagglund, T. *PID Controllers: Theory, Design, and Tuning, Instrument Society of America, Research Triangle Park*; International Society of Automation: Triangle Park, NC, USA, 1995.

32. Le Digabel, S. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Trans. Math. Softw. (TOMS)* **2011**, *37*, 44. [CrossRef]

33. Mitchell, M. *An Introduction to Genetic Algorithms*; MIT Press: Cambridge, MA, USA, 1998.

34. Haupt, S. *Practical Genetic Algorithms*; State College, Pennsylvania, John Wiley & Song, Inc.: Hoboken, NJ, USA, 2004; pp. 123–190.

35. Moscato, P. *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*; California Institute of Technology: Pasadena, CA, USA, 2000.

36. Riedmiller, M. Advanced supervised learning in multi-layer perceptrons—From backpropagation to adaptive learning algorithms. *Comput. Stand. Interfaces* **1994**, *16*, 265–278. [CrossRef]

37. Goodhart, C. *Goodhart's Law. The Encyclopedia of Central Banking*; Edward Elgar Publishing: Cheltenham, UK, 2015; p. 227.