



Optimization of control strategy for a low fuel consumption vehicle engine

Jakub Sawulski, Maciej Ławryńczuk*

Warsaw University of Technology, Institute of Control and Computation Engineering, Faculty of Electronics and Information Technology,
ul. Nowowiejska 15/19, Warsaw 00-665, Poland

ARTICLE INFO

Article history:

Received 1 November 2018

Revised 2 March 2019

Accepted 21 April 2019

Available online 22 April 2019

Keywords:

Low fuel consumption vehicles

Optimization

Evolutionary algorithms

Optimal control

ABSTRACT

The objective of this work is to optimize control strategy of a vehicle which leads to low fuel consumption. A specialized car developed especially to take part in the Shell Eco-marathon competition is considered. At first a general first principle model of the vehicle is discussed, its parameters are tuned taking into account the data obtained experimentally and the model is validated using real measurements. Next, the optimization problem leading to low fuel consumption is discussed. The engine usage continuous-time control strategy is parametrized by means of 4 methods in order to reduce computational complexity. Since the resulting constrained optimization problem is nonlinear, as many as 5 evolutionary algorithms are examined in terms of quality of the final result and computational efficiency. Inefficiency of a classical gradient-based optimization method is demonstrated. Finally, a complete decision system which can act as an autonomous vehicle engine controller is discussed.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

Nowadays, one of the main development directions in the automotive industry is to improve energy efficiency of vehicles. There are two important reasons for that. Firstly, reduction of energy used by cars directly results in increasing economic efficiency of the transport sector. Secondly, reduction of energy leads to lowering CO₂ emission. It is necessary to stress the fact that the transport sector is responsible for 23 % of the global CO₂ emission resulted from fuel consumption [8]. Although electric cars become more and more popular due to their “zero-emission” propulsion system and relative low cost per driven distance, the majority of vehicles used are powered by classical internal combustion engines. Unfortunately, electric cars currently available on the market have important disadvantages: high production cost which mainly results from the battery manufacturing cost, limited range and underdeveloped charging points infrastructure. Furthermore, the use of electric cars does not generate any pollution, but production and recycling may be more dangerous for the environment than in the case of the classical vehicles based on combustion engines. As a result, there is still a great need to optimize energy consumption of petrol engines.

Research concerned with low fuel consumption vehicles is currently carried out by many commercial companies as well as scientific institutions. A great forum to demonstrate the latest achievement is the Shell Eco-marathon (SEM) organized by Royal Dutch Shell company. The most interesting competition, called the UrbanConcepts, takes place between energy

* Corresponding author.

E-mail addresses: J.Sawulski@ia.pw.edu.pl (J. Sawulski), M.Lawrynczuk@ia.pw.edu.pl (M. Ławryńczuk).

efficient city cars that are supposed to replace the ones used currently. The Shell Eco-marathon has been organized since 1985, the Europe 2017 edition engaged more than 200 teams from leading European technical universities. During the SEM series, several world records in fuel usage minimization have been established by especially designed vehicles [34].

Fuel consumption is determined by multiple factors. The control strategy of the vehicle engine is one of the most important. The problem is complex since it is possible to parametrize the control strategy in many ways, the issue is to define a good control policy and next optimize it in such a way energy consumption is minimized. In case of an internal combustion engine, obtaining optimal control strategy is not straightforward, especially due to its non-linear characteristic of power, torque and energy conversion efficiency. Hence, the resulting optimization problem is not trivial (i.e. nonlinear, non-convex) subject to some constraints imposed by technological criteria and requirements. Furthermore, energy consumption can be also limited by engaging “start-stop” systems or engine braking, especially used in urban areas when a lot of stops caused by traffic lights or jams can occur [38]. Such factors definitely make the problem of optimizing energy consumption difficult.

Traditionally, fuel consumption is mainly influenced by the driving technique used. Most teams competing in the SEM series utilize a lot of time for track analysis and driver training (also on the especially designed car simulators) to achieve the best results. This approach is significantly limited as a track change makes it mandatory to repeat the whole process. The leading teams work on driving automation which can improve their results. Advanced Vehicle Control Systems (abbreviated to ADAS) are also in the area of interest of scientists and automotive companies. Optimization based methods have been proposed to limit vehicle fuel consumption in which dynamic programming [2] is used off-line to generate the reference speed trajectory and Model Predictive Control (MPC) controller is used on-line to maintain this trajectory in car-following scenario [42]. Dynamic programming algorithm acting as off-line optimization engine with the MPC controller is also considered in [4,33] however this solution is mainly focused on electric cars whose performance characteristics allow to take advantage of high energy conversion efficiency in wide range of operating points. In [18] the Bellman's approach is used for gasoline vehicle efficiency optimization, but a real-time on-line solution is not implemented due to high computational complexity.

The objective of this work is to optimize control strategy of a vehicle which leads to low fuel consumption. Research is concerned with a specialized car developed especially to take part in the Shell Eco-marathon competition. At first a general first principle model of the vehicle is discussed, its parameters are tuned taking into account the data obtained experimentally and the model is validated using real measurements. Next, the optimization problem leading to low fuel consumption is discussed. The engine usage continuous-time control strategy is parametrized by means of 4 methods in order to reduce computational complexity (by reducing the number of decision variables of the optimization problem). Since the resulting constrained optimization problem is difficult (nonlinear, possibly non-convex) as many as 5 evolutionary algorithms are examined in terms of quality of the final result and computational efficiency: the Particle Swarm Optimization (PSO) algorithm [11], the Invasive Weed Optimization (IWO) scheme [20], the Firefly Algorithm (FA) [44], the Grey Wolf Optimizer (GWO) [21] and the Jaya Algorithm (JA) [30]. Finally, inefficiency of the classical gradient-based optimization method (the Sequential Quadratic Programming algorithm [27]) is demonstrated. Taking into account the obtained results, a complete decision system which can act as an autonomous vehicle engine controller is discussed. It may be used as a part of real solution intended to increase fuel consumption efficiency. Computational aspects of the proposed solution are also discussed.

This work is organized in the following way. At first, in Section 2, a first principle mathematical model of the vehicle is derived. Next, Section 3 deals with parametrization and optimization of control strategy for the vehicle engine, efficiency of the used optimization algorithms is compared. Section 4 discusses the proposed decision system and the possible choice of the hardware platform. Finally, Section 5 concludes the whole work.

2. Modelling of vehicle dynamics

2.1. PAKS Vehicle

The PAKS car shown in Fig. 1 is considered in this work. It was built in 2011 by Student Association for Vehicle Aerodynamics (SAVA), affiliated with the Faculty of Power and Aeronautical Engineering at the Warsaw University of Technology [35]. The vehicle was built with the aim of taking part in the UrbanConcept category during the Shell Eco-marathon competition series. The PAKS is a concept car of a one-person urban vehicle measuring about 350cm x 140cm x 120cm. Its body shape is based on a glider cockpit which leads to a low aerodynamic drag coefficient. The main element of its propulsion system is 72cm³ air-cooled, single cylinder Honda 139FMB engine, which can generate up to 4.5HP power. The genuine carburetor-based fuelling and ignition systems has been replaced by the indirect injection system based on ECUMASTER EMU standalone engine management system. It allows to use more adequate fuelling strategy based on various measurements instead of mechanical tuning of carburetor, which limits fuel consumption and pollution emission. The PAKS is also equipped with an expanded on-board data acquisition system consisting of the STM32 microcontroller-based devices connected with Controller Area Network (CAN) bus. The described structure, developed from scratch by the SAVA members, allows full customization of data-logging and real-time measurements of various parameters such as vehicle position, speed or acceleration [35].

During the SEM Europe 2013, which took place in Rotterdam, the PAKS won the first place in the UrbanConcept Gasoline category with the result 334km/L. The best result of fuel economy achieved by PAKS was 376 km/L during SEM Europe



Fig. 1. The PAKS vehicle during the SEM Challenger 2016 in Le Mans, France.

2014 competition (the second place in its category). From 2016, when competition was moved to London, results achieved by all teams were reduced by about 40%. The competition track lap was shortened for about 1/3 whereas the total length of trial distance remained almost unchanged, which caused greater number of required stops after every lap. Additionally, in contrast to the actually flat Rotterdam track, the part of London track is the railway overpass which introduces steep driveway and it also increases car fuel consumption. The best result achieved by the PAKS on the London track is about 190 km/L [34,35].

2.2. Model derivation

The mathematical model of vehicle dynamics has to contain two main parts which describe its motion and fuel consumption. The second part, as fully and only depending on the first one, can be solved separately after calculation of motion trajectories. Decomposition of the full complex task significantly simplifies obtaining the results of the fuel usage optimization problem defined next.

When it is possible, the mathematical process description is derived using fundamental physical laws. Otherwise, simple polynomial functions are used to numerically approximate unknown relations (datasets obtained from the real vehicle are used for this purpose). Such an approach is commonly used when the objective is to obtain a mathematical model of real processes [23].

The mathematical model which is next used in numerical simulations has been significantly simplified with the aim of obtaining computational efficiency of calculations. Inertia of rotational parts, such as wheels, has been neglected. Since mass is the main factor that influences rolling resistance, the vehicles that take part in the SEM competition in the UrbanConcept category rarely weight more than 150 kg. As a result, minor mass of rotational elements cannot accumulate high amount of energy.

The loss of kinetic energy caused by fast cornering also has been ignored. The average speed of cars is about 25 km/h. The maximum speed cannot exceed 50 km/h. The SEM track corners radius, according to official rules, it is never shorter than 9 m. Hence, forces coming from quickly driven corners may be neglected in comparison with rolling resistance or propulsion force.

2.2.1. Motion of vehicle

Equations which describe motion of the PAKS have been derived from the Newton's 2nd law. One of the conditions which allow to use this law is that a body can be idealized as a single point mass. Hence, size and shape of the body are irrelevant. In the general case it is a strong assumption, but in case of modelling of vehicle dynamics it should not disturb the obtained results.

Estimation of vehicle's acceleration, as a parameter of its motion, requires knowledge of mass and the set of forces acting on it. The first value is simply measurable. At the SEM2017 competition, the car weighted 121 kg, the driver with necessary equipment 71 kg, which in sum gives 192 kg. For modelling purposes, the vehicle is reduced to a point mass object. The set of forces acting on the vehicle and considered in modelling process is presented in Fig. 2 and consists of:

- a) The aerodynamic drag force evaluated as

$$F_{aero} = \frac{1}{2} \rho S c_x \dot{x}^2 \quad (1)$$

where $\rho = 1.225 \text{ kg/m}^3$ is air density at sea level according to International Standard Atmosphere (ISA), $S = 1.008 \text{ m}^2$ is cross-sectional area of vehicle, based on measurement of the Computer Aided Design (CAD) model, $c_x \approx 0.25$ is a drag coefficient of vehicle, based on Computational Fluid Dynamics (CFD) simulation and x is position of vehicle.

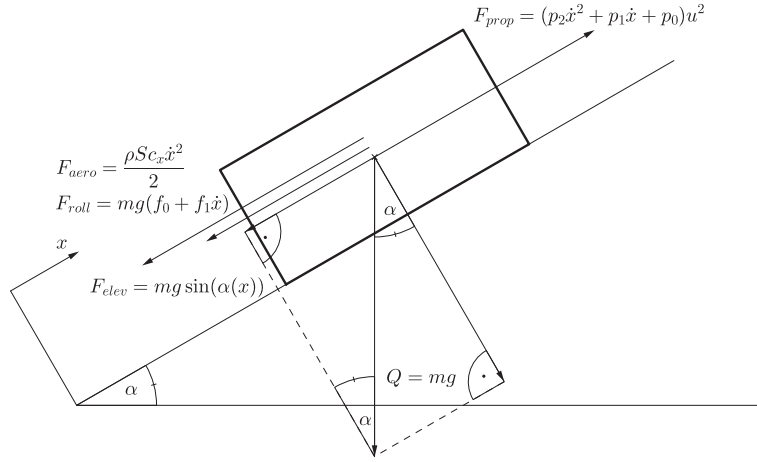


Fig. 2. Forces used in modelling of motion dynamics.

b) The rolling resistance force evaluated as

$$F_{roll} = mg(f_0 + f_1\dot{x}) \quad (2)$$

where $m = 192$ kg is mass of ready-to-run vehicle, $g = 9.81$ km/s² is standard gravity, f_0, f_1 are unknown rolling resistance coefficients. In most applications, vehicle speed-independent formula is used. In this work linear dependency is chosen [9]. Due to well reduced friction forces in the PAKS vehicle mechanisms, the expanded formula can give more accurate results as lower order of magnitude of resistance forces are expected in comparison to typical car.

c) The down-slope force evaluated as

$$F_{elev} = mg \sin(\alpha(x)) \quad (3)$$

where $\alpha(x)$ is the angle of the route slope as a function of the current vehicle position. It is assumed that the entire road profile is known. In case of the SEM competition track geometry is precisely measured by organizers and data is available for competing teams before the race.

d) The propulsion force evaluated as

$$F_{prop} = (p_2\dot{x}^2 + p_1\dot{x} + p_0)u^2 \quad (4)$$

where p_2, p_1, p_0 are unknown propulsion coefficients, u is a relative throttle opening variable from 0 to 1 (in case of the PAKS vehicle, equipped with butterfly throttle valve, it corresponds to the range from 0° to 90°).

All things considered, longitudinal vehicle motion can be described by

$$m\ddot{x} = F_{prop} - F_{aero} - F_{roll} - F_{elev} \quad (5)$$

Using Eqs. (1)–(5), one obtains

$$m\ddot{x} = \left(p_2u^2 - \frac{\rho S c_x}{2}\right)\dot{x}^2 + (p_1u^2 - mgf_1)\dot{x} + p_0u^2 - mgf_0 - mg \sin(\alpha(x)) \quad (6)$$

2.2.2. Fuel consumption

The injection time necessary for proper engine operation, which determines fuel consumption measured per time period, is calculated by a specialized low-level engine controller called Engine Control Unit (ECU). The injection time is evaluated as a function of many parameters, e.g.: static engine characteristics such as volumetric efficiency (VE) or real-time measured values such as throttle position and concentration of oxygen in exhaust-gas via lambda probe. The algorithm involving all parameters can be very complicated and in most cases is not accessible for the end-user as part of ECU software. Due to this limitation, a simplified model of fuel consumption must be used. Instantaneous fuel consumption is influenced by many factors, but for a simplified model it can be assumed that only two of them are important and must be taken into account: throttle angle and rotational speed of engine.

Quantity of fuel injected in every work cycle necessary for stable engine operation, directly depends on the volume of intaken fresh air (it determines volume of oxygen in chamber which can be used in combustion). The ability of load exchange is volumetric efficiency of the engine. This static characteristics of engine describes how much of fresh air is placed in the combustion chamber relative to its volume at specific conditions. For naturally aspirated engines it can reach up to 90–95%.

Since many factors influence fuel consumption, the task of injection dose calculation is delegated to an especially designed ECU unit while a higher-level optimizer should only calculate approximate efficiency of the desired driving strategy.

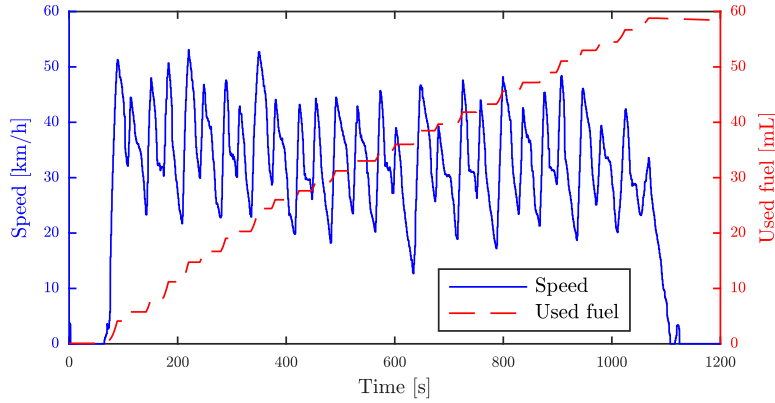


Fig. 3. Dataset for identification of model parameters.

Furthermore, instantaneous fuel consumption is almost fully independent of consumption in other time periods. In contrast to the differential Eq. (6) which describes vehicle motion, there is no phenomenon of error accumulation due to an inaccurate model. Moreover, fuel efficiency is “only” a quality indicator of the proposed control strategy. Knowledge of its exact value is not necessary: for a numerical comparison of control strategies a precise value of fuel consumed in driving cycle is not important, but relative performance difference between the consecutive control strategies.

For the aforementioned reasons a significantly simplified model of fuel consumption is used. Fuel efficiency of the engine is mainly determined by throttling intake air flow. Minimum consumption (measured per generated mechanical energy) is accomplished for around wide-open throttle. As a result of this consideration, almost all teams competing in the SEM series use the so called Pulse and Glide (PnG) control strategy [15]. According to this policy vehicle accelerates with maximum available power (pulse phase) followed by period of coasting with engine switched off (glide phase).

Prototypes of the fuel efficient cars usually have highly reduced aerodynamic drag and rolling resistance coefficients. Total resistive force generated during vehicle motion can have much lower order of magnitude than engine propulsion force. In this case the PnG strategy allows to use vehicle's engine for several seconds and coast for a relative long distance without significant loss of speed. Continuous use of engine in this case can be remarkably less efficient. In order to produce small amount of energy narrow, inefficient, throttle opening is unavoidable [36].

It is assumed that for wide-open throttle fuel consumption can be evaluated as a quadratic function of rotational speed of engine (vehicle). Moreover, to make fuel efficiency dependent on throttle opening, a linear change of fuel consumption based on throttle position is postulated. Hence, fuel consumption is estimated from the following simplified equation

$$\dot{q} = (r_2 \dot{x}^2 + r_1 \dot{x} + r_0)u \quad (7)$$

where $\dot{q}$ is injector instantaneous fuel flow, r_2 , r_1 , r_0 are unknown fuel efficiency coefficients and u is a relative throttle opening variable from 0 to 1.

To simulate fuel consumption using the engine switching on and off strategies, fuel quantity dissipated in engine start sequence has to be estimated. As the vehicle is equipped with odometry and fuel consumption measurement systems, it can be obtained by direct measurement instead of estimation.

2.3. Tuning of model parameters

The following parameters must be found during model identification:

- c_x : Aerodynamic drag coefficient,
- f_1, f_0 : Rolling resistance coefficients,
- p_2, p_1, p_0 : Propulsion force coefficients,
- r_2, r_1, r_0 : Fuel consumption coefficients,
- r_{start} : Fuel used for engine start.

Identification is based on real data collected during an attempt at the Le Mans Karting International track. Longitudinal profile of that track is notably flat, which allows to neglect F_{elev} force while other parameters are estimated. Because of this feature, the Ordinary Differential Equation (ODE) (6) is fully independent of absolute position of the car: position measurements are not involved for identification which is remarkably simplified. Datasets used for parameter identification (speed and used fuel quantity time series saved by datalogger) are presented in Fig. 3. The current speed is determined by a GPS receiver and fuel consumption is measured by a dedicated gasoline flow meter.

As the PnG policy has been applied by the driver, large variation of speed during a race can be observed. This strategy also gives possibility to consider phases with and without engine propulsion separately. Parameters involved in motion

Table 1

Model errors for roll characteristics (the test error of the chosen model is emphasized).

Polynomial order	Training dataset SSE	Test dataset SSE
1	50.5×10^3	8.91×10^3
2	42.0×10^3	7.52×10^3
3	41.5×10^3	7.54×10^3
4	41.5×10^3	7.48×10^3

Table 2

Model errors for fuel consumption (the test error of the chosen model is emphasized).

Polynomial order	Training dataset SSE	Test dataset SSE
1	5.6129	0.9760
2	5.5121	0.9523
3	5.5112	0.9526
4	5.4208	0.9480

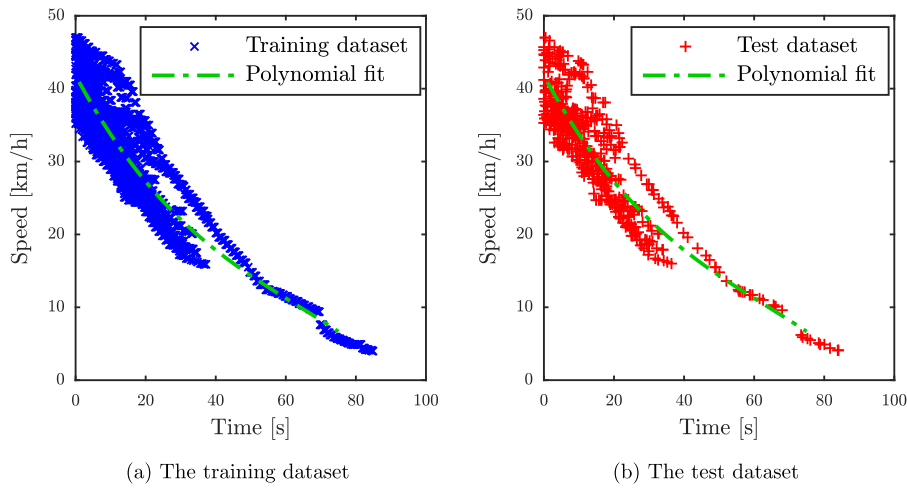


Fig. 4. Comparison between training and test data sets vs. the second-order polynomial model for roll characteristics.

characteristics can be split into two groups according to relation to propulsion or resistive forces. As long as a vehicle is coasting with engine switched off, only three parameters: c_x , f_1 and f_0 have the influence on vehicle behaviour. This approach leads to six variable estimation task which is decomposed into two tasks with three searched values. Throttle position has not been measured during test ride. It has been assumed that the driver used the technique with wide-open throttle (maximal or minimal flow through intake manifold) and engine operation time can be deduced from fuel usage plot. The time period with a constant value of consumed fuel leads to conclusion that it is a time of engine shut-down.

Data presented in Fig. 3 has been split according to engine operation. For all free-wheeling periods, additionally, data has been trimmed to ensure the same initial speed in every cycle. For both roll characteristics and fuel consumption, four polynomial models are considered: the first-order, the second-order, the third-order and the fourth-order models. Numerical values of the Sum of Squared Errors (SSE) for the training and test datasets are detailed in Tables 1 and 2. Considering the found models, taking into account the model errors for the test sets, it is evident that the second-order model structures must be selected. They are characterized by lower errors than the first-order ones whereas increasing models' order does not lead to any significant improvement of accuracy. Comparison between two data sets and the second-order polynomial models are presented in Figs. 4 and 5. The found rolling parameters are used to propulsion force curve fitting. For fuel efficiency estimation total consumption has been numerically differentiated to approximate the value of instantaneous usage. Simultaneous measurement of consumption and speed allows to remap fuel consumption from term of time to the current speed.

The estimated values of model parameters are:

$$c_x = 0.32 \quad (8)$$

$$f_1 = 1.33 \times 10^{-3}, \quad f_0 = 168 \times 10^{-6} \quad (9)$$

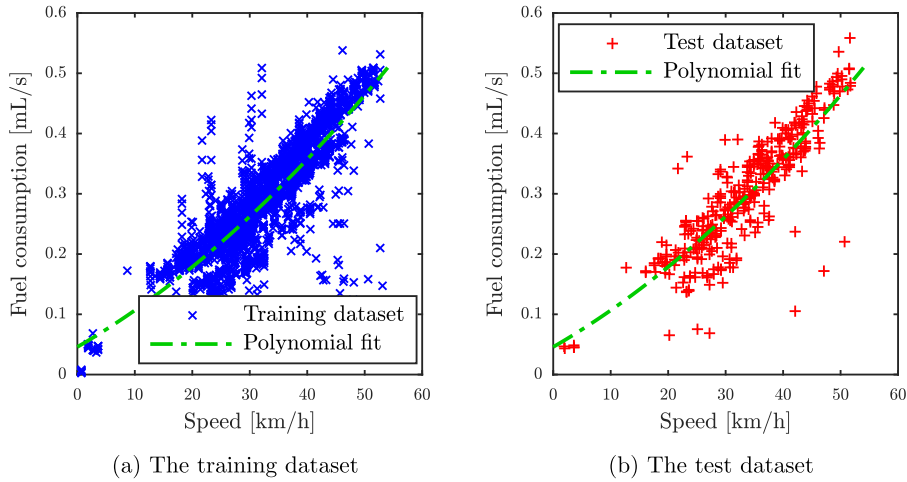


Fig. 5. Comparison between training and test data sets vs. the second-order polynomial model for fuel consumption.

$$p_2 = 1.039 \text{ N s}^2/\text{m}^2, \quad p_1 = -18.15 \text{ N s/m}, \quad p_0 = 233.1 \text{ N} \quad (10)$$

$$r_2 = 740 \times 10^{-6} \text{ mL s/m}^2, \quad r_1 = 19.8 \times 10^{-3} \text{ mL/m}, \quad r_0 = 45.8 \times 10^{-3} \text{ mL/s} \quad (11)$$

The obtained values may be interpreted in the following way:

- The value of the aerodynamic drag coefficient ($c_x = 0.32$) is tens of percent higher than the value estimated in CFD simulations (0.25). It may result from many reasons. CFD experiments assume that an accurate model is available. Despite the PAKS manufacturing process has been based on a detailed CAD design, during years of operation some elements changed their attributes. Mainly, degradation process affect body component dimensions due to minor crashes.
- The values of rolling resistance coefficients ($f_1 = 1.33 \times 10^{-3}$, $f_0 = 168 \times 10^{-6}$) are placed in expected range as reported in [33]. Static friction force for the record vehicle Pac-Car II built by ETH Zürich amounts to about 3 N. For comparison, the friction value generated by the PAKS tyres and mechanisms at speed 1 km/h is 2.9 N.
- The values of propulsion force coefficients ($p_2 = 1.039 \text{ N s}^2/\text{m}^2$, $p_1 = -18.15 \text{ N s/m}$, $p_0 = 233.1 \text{ N}$) are not comparable to their typical ranges. Usually, in automotive benchmarks torque and power of engine are measured and furthermore are made for unit isolated from drive-train. Additionally, it is hard to compare the PAKS engine parameters with any other reliable measured data. Available engine characteristics are based on mass-produced car units with, typically, three or more cylinders. Thoroughly modified, single-piston, 72 cm³ PAKS engine is incomparable with them in any straightforward way.

As described earlier, the value of r_{start} is directly measured. Test series has been performed with real-time measurement of speed and fuel flow. After the engine starts, the driver immediately begins accelerating. It is assumed that fuel quantity used before the moment when speed measurement starts rising is used for the engine start sequence. Due to a significant delay in GPS operation, a responsive odometry system has been engaged. Distance resolution is about 5 cm (1/36 of wheel circumference), distance measurement is refreshed every 0.1 s which gives absolute speed resolution of 1.8 km/h (lower ratio gives better speed resolution but introduces a measurement delay). This value has been set as limit between motion and quiescence of vehicle. Averaged datasets from 15 experiments are presented in Fig. 6. Measured value of start fuel is $r_{\text{start}} = 0.11 \text{ mL}$ which corresponds to approximately two seconds of engine work in the idle state.

2.4. Model verification

Vehicle model verification are based on an independent dataset acquired during the SEM2016 competition (London track). Characteristics of this route is quite different from the track where previous data set has been collected. In particular, it applies to its longitudinal profile since overpass has been placed on the route. There has been approximately 9.5 m elevation difference between the highest and the lowest points of the track with total length of 1659 m. For this reason, influence of slopes at vehicle motion cannot be easily neglected. Previously assumed, physical model of down-slope force ($F_{\text{elev}} = mg \sin(\alpha(x))$) is used, as track profile ($\alpha(x)$) is known and position (x) has been recorded via a GPS datalogger. Collected time series of speed and used fuel are presented in Fig. 7.

Model verification is performed according to the following simple work flow. Firstly, the desired time period from test data set is chosen. Engine operation time is obtained based on fuel usage changes. As an initial conditions for speed, position and fuel usage, measured values from simulation start time point are taken. After that, simulation is performed and the

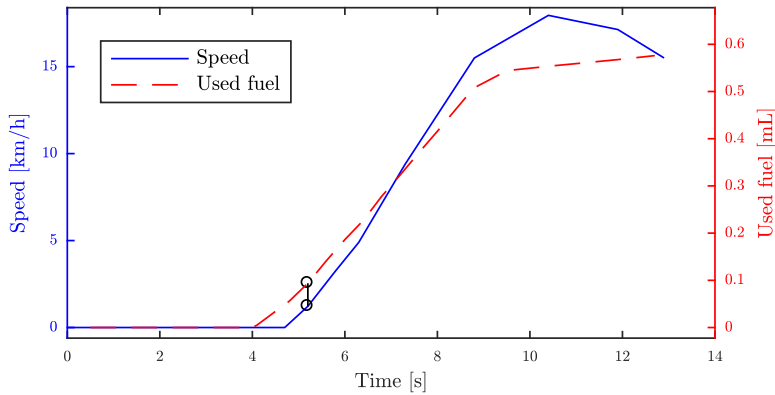


Fig. 6. Estimation of start fuel value.

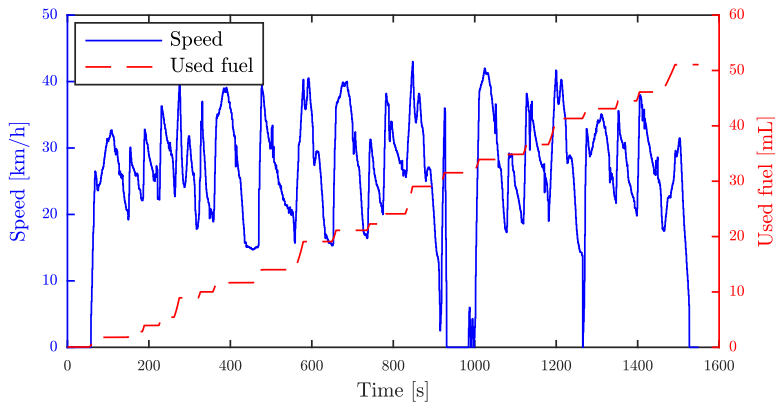


Fig. 7. Dataset for model verification.

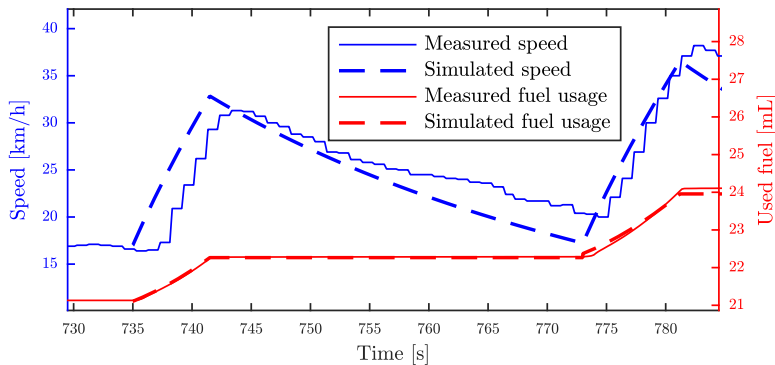


Fig. 8. Example model test.

result is compared to measured data. Example model simulated response vs. measured data is presented in Fig. 8. Model-based simulation gives reasonably accurate response. Some conclusions based on several simulations can be drawn:

- Simulated speed signal is ready before its measured counterpart. Typically, the positioning system based on a GPS is characterized by a time delay between the moments the signal is received and recognized. This effect is magnified by a limited refresh ratio of position reading. Speed, calculated by differentiation of position, is also delayed compared to its real value. There is no significant delay between simulated and measured used fuel quantity. In contrast to speed measurement, it is computed by counting pulses from an encoder-based fast-response flow meter.
- Model response better fits with measurements for acceleration than coasting periods. The engine generated propulsion force has higher order than the total anti-motion force, even if it works with low rotational speed. Assuming that unessential forces (neglected in modelling) have approximately invariant value, their relevancy is lower in relation to leading forces. In consequence, when propulsion is used, their influence is more unnoticeable.

Generally, model accuracy is acceptable to be used as an internal model of the embedded decision system. Simulated signal trajectories are close to measured ones. Moreover, training and test data sets for model parameter estimation have been collected during real competition trials. There is no information about aspects such as influence of other vehicles on the track, usage of brakes or other unexpected circumstances. Data comparison presented in Fig. 8 takes into account the time period equal to approx. 50 s which corresponds to distance approx. 1/3 of the SEM2016 lap length. As model tests share only initial condition with measured data, a simulation drift from measurement is unavoidable even if error of parameter estimation is infinitesimal. The vehicle model can be easily used for real-time simulation of the PAKS dynamics. To take into account more specific phenomena, sophisticated identification experiments need to be performed which may require additional resources such as advanced measurement system or detailed geodetic data for test track.

3. Optimization of control strategy for vehicle engine

3.1. Formulation of optimization problem

The model of vehicle motion and fuel consumption is defined by Eqs. (6) and (7). The resulting system of first-order ODEs is

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \left(\frac{p_2 u^2}{m} - \frac{\rho S C_x}{2m} \right) x_2^2 + \left(\frac{p_1 u^2}{m} - g f_1 \right) x_2 + \frac{p_0 u^2}{m} - g f_0 - g \sin(\alpha(x_1)) \\ \dot{x}_3 &= (r_2 x_2^2 + r_1 x_2 + r_0) u\end{aligned}\quad (12)$$

where u is the throttle opening and the state variables are: x_1 – vehicle position, x_2 – vehicle speed, x_3 – integrated fuel consumption.

In the general case the problem of obtaining control strategy which leads to minimal fuel usage is defined as finding the control trajectory $u(t)$ which minimizes a functional specified as integral of instantaneous fuel usage. If the control trajectory can be defined as a function of several parameters, this task is reduced to non-linear optimization task with constraints. To obtain this form, some necessary transformations are necessary.

The currently used strategy of using an internal combustion engine is to run it for certain time periods with its maximum available power. In terms of the SEM competition where one lap of a track is about 1700 m long, it corresponds to three or four engine short operation spans. Each of them, as a constant, maximum throttle is only used, is characterized by two parameters: Time or vehicle position at the moment of start-up and duration of engine operation. Hence, to obtain the complete control signal trajectory from six to eight parameters should be calculated.

A simple control policy may be applied by the driver. According to experience, he/she can estimate when and how long the engine should work to fulfil the maximal allowed race time constraint and consume the minimal amount of fuel. Unfortunately, this simple rule does not guarantee maximum possible fuel efficiency. Even though efficiency of the proposed policy is satisfactory, it cannot be assumed that the driver is able to realize this strategy with required accuracy. Moreover, unpredictable significant disturbances may occur. In such cases a human may not make accurate predictions, e.g. when speed is sufficient to reach the finish line after brake has been used contrary to the strategy. These disturbances are typical at the SEM competitions but also in urban traffic.

As long as the model of vehicle dynamics is available, it is reasonable to try to set up decision system as numerical method based on objective clues instead of intuitive human perception. This allows also to a numerical comparison of more complicated control policies than several single engine usages with maximum throttle opening.

Taking into account the way a human finds an optimal control policy, the nonlinear programming task leading to maximization of fuel efficiency (km/L) (minimization of fuel consumption) is

$$\max_u s \left(\int_0^{t_{\max}} \dot{x}_3 dt + |\{t : u(t^-) = 0 \wedge u(t^+) \neq 0\}| \cdot r_{\text{start}} \right)^{-1} \quad (13)$$

subject to

$$\dot{x}_1 = x_2 \quad (13a)$$

$$\dot{x}_2 = \left(\frac{p_2 u^2}{m} - \frac{\rho S C_x}{2m} \right) x_2^2 + \left(\frac{p_1 u^2}{m} - g f_1 \right) x_2 + \frac{p_0 u^2}{m} - g f_0 - g \sin(\alpha(x_1)) \quad (13b)$$

$$\dot{x}_3 = (r_2 x_2^2 + r_1 x_2 + r_0) u \quad (13c)$$

$$x_1(0) = 0 \quad (13d)$$

$$x_2(0) = 0 \quad (13e)$$

$$x_3(0) = 0 \quad (13f)$$

$$u(0) = 0 \quad (13g)$$

$$t_{\text{end}} \leq t_{\text{max}} \quad (13h)$$

$$x_1(t_{\text{end}}) = s \quad (13i)$$

$$x_2(0 < t \leq t_{\text{end}}) \geq 0 \quad (13j)$$

$$0 \leq u(0 < t \leq t_{\text{end}}) \leq 1 \quad (13k)$$

where s denotes the given length of one lap track. In case of the SEM2017 competition it equals to 1659 m. t_{max} denotes the maximum time to cover the distance. According to official rules, it equals to 234 s. This set of rules implies that mean speed of vehicle should be higher than approx. 25.5 km/h. The vector of decision variables $\mathbf{u}$ of the optimization problem (13) depends on the chosen problem parametrization method. They are thoroughly discussed in Section 3.4. The value of the maximized cost function is determined by an inverse of a sum of two parts: the integral of instantaneous fuel consumption and the costs of every single engine start-ups. The following constraints are used:

- The model of vehicle (the constraints (13a)–(13c)): This subset of constraints is presented as ODEs. For proper handling of those constraints, ODE integration (performed by the Cash–Karp method [28]) is involved which allows to obtain x_1 , x_2 and x_3 state trajectories for the complete considered period of time from $t = 0$ to $t = t_{\text{end}}$.
- The zero initial condition: at the beginning of the race the vehicle is stopped, i.e. its initial speed is zero, the control variable is also zero, no fuel is used (the constraints (13d)–(13g)): Those constraints are handled directly by proper setting of the ODE solver initial condition.
- At the time t_{end} the distance s has to be reached and $t_{\text{end}} \leq t_{\text{max}}$ which is the maximum time stated by competition rules (the constraints (13h), (13i)): The intermediate and the final value constraints can be handled and checked only after evaluation of embedded ODE integration. Satisfaction of those constraints is enforced by a proper selection of the cost-function, i.e. by using a penalty method (the external penalty is used). In this approach a penalty term proportional to the degree of violation of the consecutive constraints is simply subtracted from the maximized cost-function.
- The vehicle has to move permanently (the constraint (13j)): As in the previous case, handling of this constraint can be done at least after ODEs integration and taken into account by using a penalty term. There is no practical method to enforce generation of solution candidates which are known *a priori* to make a car not to stop or not to stepping back from the hill. This effect can be observed if engine is used to short or at an improper track part.
- The control variable constraint (13k): Handling of this constraint depends on the chosen parametrization of control strategy. If parametrization assumes periods of engine usage with constant amplitude, none of additional activities need to be performed. If the parametrization method assumes variable throttle amplitude, cubic constraints on the proposed throttle values have to be imposed and those constraints are resolved at the stage of candidate generation in each optimization algorithm iteration. Details of parametrization-dependent constraints handling are discussed in Section 3.4.

The set of constraints may be split into two groups. First of them (connected with the initial values and direct cubic constraints) is fulfilled by an appropriately accomplished candidate generation stage and the second one is satisfied by usage of the penalty method. In this specific case, for example, it is not possible to verify if the condition of minimum distance passed in maximum time is satisfied without ODE-constraints integration, the so called death-penalty function is used which subtracts from the final cost-function value a large constant. The death-penalty function is chosen as it can be easily implemented and fast evaluated, in this particular case, giving results no worse than by using a classical linear or square penalty function.

To sum up, the final form of the cost function calculation needs the following steps:

- Satisfaction of parametrization-specific constraints is checked:
 - In case of parametrizations which assume variable throttle opening during race, if maximal or minimal throttle amplitudes reach values out of the unit interval, they are truncated appropriately (constraint (13k)).
 - In case of parametrizations which use time points as parameters that have to form non-decreasing vector (to ensure a proper order of subsequent engine startups and shutdowns), if any consecutive values placed in the vector of parameters disturbs monotonicity, it is increased to satisfy this condition.
- The initial conditions for ODEs integration are applied (constraints (13d)–(13g)),
- ODEs integration is performed (constraints (13a)–(13c))
- As full information about $\dot{x}_3(t)$ and $u(t)$ is available, the temporary value of the cost function (13) is calculated.

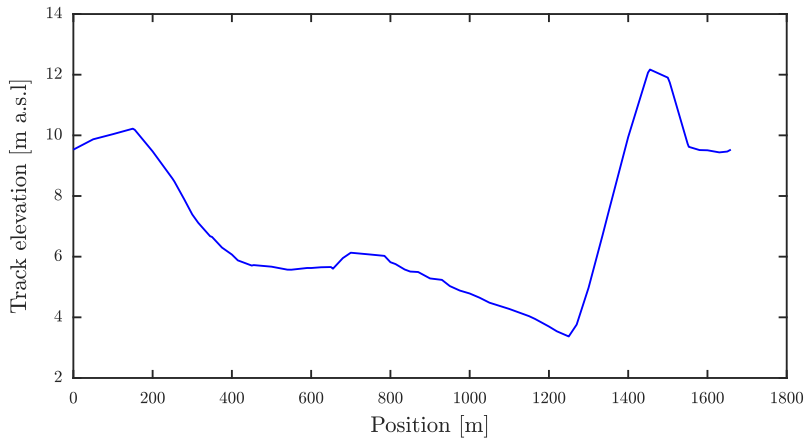


Fig. 9. Track longitudinal profile $h(x_1) : \frac{dh}{dx_1} = \tan(\alpha(x_1))$.

- e) Satisfaction of all transient and final value constraints (13h)–(13j) is checked. If they are fulfilled, the temporary value of the cost function is used as its final value. If not, the temporary value is reduced by an adequate penalty factor, calculation of which is based on measure of constraints not satisfaction.

Track longitudinal profile (London track) is depicted in Fig. 9 [40].

Two issues related to the optimization problem (13) must be emphasized:

- a) **Parametrization.** Since there is no possibility to obtain the optimal control strategy as an analytical function, some parametrizations methods have to be introduced to convert the task to a form in which several parameters are searched. For example, a continuous function can be approximated as a table of its values in some characteristic time points. However, it should be considered that convergence ratio or, in general, possibility to use a specific optimization algorithm can depend on the size of the decision vector. Typically, increasing dimensionality of searched space causes difficulties in finding solution. One may expect that no more than up to several parameters are sufficient to describe distinctive control policies.

An alternative way of solving the problem numerically is to use dynamic programming method according to Bellman equation. Due to continuous time problem formulation, an infinite number of possible states should be analyzed. Conversion to discrete time leads to a finite set, but it may be huge (it depends on the discretization time step). In view of future application using a hardware platform, it is likely to lead to memory problems.

- b) **Nonlinearity.** In case of a nonlinear model, the problem of maximizing fuel efficiency leads to a nonlinear optimization problem which does not have an analytical solution.

Since the optimization problem is in general nonlinear, likely non-convex, multi-modal and the constraints are difficult to satisfy, it is not recommended to use classical gradient-based optimization methods [1,24]. Conversely, some global optimization algorithm are considered. Evolutionary Algorithms (EAs) are relatively simple methods which use some heuristic strategies to explore the whole domain of interest. Typically, only cost-function values are used, not gradients. EAs are significantly more successful than simple searching methods (e.g. grid search).

3.2. Optimization algorithms used

The work flow of the global optimization methods used is the same. At the beginning, a fixed number of random vectors are generated. For each of them (called “particles”, “individuals”, etc.) the value of the maximized cost-function is evaluated. Next, the coordinates of vectors are updated according to some algorithm-specific policy. The consecutive iterations of the algorithm (called epochs or generations) are repeated until a termination criterion is reached. In this work 5 optimization methods are used.

3.2.1. Particle swarm optimization

The PSO algorithm is used to solve numerous complex optimization problems [11], also in a parallel way [22]. Although a great number of modification exist, e.g. [6,16,37], in this work the original formulation of PSO is used. The change of position (originally named “velocity”, $\mathbf{v}_i$) for each particle in i -th generation is obtained as weighted sum of “velocity” from the previous generation ($\mathbf{v}_{i-1}$) as well as disturbed (uniformly distributed random vectors $\mathbf{r}$ with elements from [0; 1]) differences between the current position relative to the best known position of particle ($\mathbf{p}$) and the best known position achieved by whole swarm ($\mathbf{b}$)

$$\mathbf{v}_i = c_0 \mathbf{v}_{i-1} + c_1 \mathbf{r}_1 (\mathbf{p}_i - \mathbf{x}_{i-1}) + c_2 \mathbf{r}_2 (\mathbf{b} - \mathbf{x}_{i-1}) \quad (14)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{v}_i \quad (15)$$

To make the algorithm converge to a stable equilibrium point, the weights must satisfy [25]

$$0 < (c_1 + c_2) < 4, \quad \frac{c_1 + c_2}{2} - 1 < c_0 < 1 \quad (16)$$

3.2.2. Invasive weed optimization

The IWO algorithm is reported to be successful in many tasks, e.g. classification [7], model fitting [39], scheduling [32]. It uses rules that govern weed colony behaviour [20]. Position and number of particles in subsequent generation are functions of “weeds” objective function values. Each of particles in every iterations generate variable number of new individuals according to

$$S = \left\lfloor S_{\min} + (S_{\max} - S_{\min}) \frac{f(\mathbf{x}) - f(\mathbf{w})}{f(\mathbf{b}) - f(\mathbf{w})} \right\rfloor \quad (17)$$

where $S_{\min}$ and $S_{\max}$ are minimal and maximal numbers of new individuals (called “seeds”), the current, best and worst particle in the whole swarm are $\mathbf{x}$, $\mathbf{b}$ and $\mathbf{w}$, respectively. The position of every “seed” is evaluated as the position of “parent weed” with added Gaussian noise component. Standard deviation of disturbance is non-linearly reduced during optimization from its maximum to minimum value. This policy causes no possibility to directly control the total number of individuals and the resulting number of evaluations of the objective function. The only possible estimation of calculation time is based on cases in which maximum or minimum “seeds” number is used. It is an impractical feature in perspective of using the IWO algorithm in a real-time decision system.

3.2.3. Firefly algorithm

The FA strategy is inspired by behaviour of fireflies [44]. It is also used to solve complex problems, e.g. identification [13], estimation [41] and feature selection [46]. The change of position of every particle is based on function called “brightness” (which can be obtained for example as the objective function) and the distance between individuals (d_{xy}). It is also randomly noised (by the vector $\mathbf{r}$ from unit interval of adequate dimension). In every generation the positions are updated according to

$$\mathbf{x}_i = \mathbf{x}_{i-1} + c_0 \exp(-c_1 d_{xy}^2) (\mathbf{x}_{i-1} - \mathbf{y}_{i-1}) + c_2 \mathbf{r} \quad (18)$$

for each pair of fireflies, denoted as $\mathbf{x}$ and $\mathbf{y}$, only if brightness function value for $\mathbf{x}$ is better than for $\mathbf{y}$. It is recommended to set $c_0 = 1$, $c_1 \in [0.1; 10]$ and $c_2 \in [0; 1]$ to get satisfactory convergence rate in typical applications [44].

3.2.4. Grey wolf optimizer

The GWO algorithm is based on behaviour of grey wolves’ pack [21]. It is used in many tasks, e.g. for feature selection [3], clustering [5], classification [48]. The position of each individual depends on three other individuals, marked as α , β and δ . They are chosen from the whole population as these which provide three best values of the cost-function. This choice is updated in every iteration of the algorithm. The position of each particle in the following generations is determined as a result of a complex formula involving positions of three best individuals, the position between the current “wolf” in relation to them and random factors. For the initial iterations particles explore the whole searched space and finally converge to a suboptimal point which approximate the position of “wolves” α , β , δ .

3.2.5. Jaya algorithm

Jaya is a relatively simple algorithm which follows the rule that single particles in each generation should be moved toward the best solution and evade from the worst [30], but it is reported to successfully solve complex engineering problems, e.g. parameter identification [17,45], economic optimization [29], antenna synthesis [31]. The best and the worst individuals are $\mathbf{b}$ and $\mathbf{w}$, respectively. The formula for updating positions of particles is

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{r}_1 (\mathbf{b}_i - \mathbf{x}_i) - \mathbf{r}_2 (\mathbf{w}_i - \mathbf{x}_i) \quad (19)$$

where $\mathbf{r}$ are random vectors with coordinates from [0; 1]. A characteristic feature of Jaya, unlike previously considered algorithms, is that it does not require any parameters.

3.3. Implementation issues

All considered optimization algorithms are implemented in the C language. As C is relatively low-level language, it allows to control code execution, especially in case of parallel computing. By using the OpenMP library pragmas, individually independent cost-function calls are performed in parallel mode on 4 threads using Intel i5 560M CPU.

For numerical integration of ODEs the Cash–Karp method is used [28]. Due to discontinuities occurring in control policies, the ODE solver wrapper splits integrated time span into periods with constant control value and solves the problem sequentially by restarting integration with the current initial value and a new control value. This workaround allows not

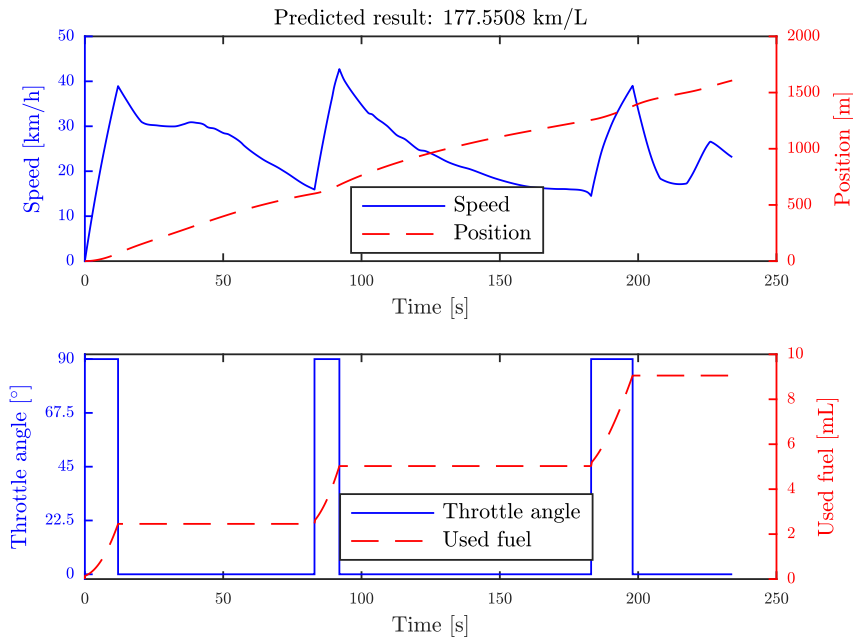


Fig. 10. Parametrization method 1: A solution obtained for 3 single starts policy.

to use very short integration steps at the ends of constant-control periods, which extends calculation time and decreases solution accuracy.

At first the results obtained for the PSO algorithm are presented and next efficiency of all algorithms is compared. Due to heuristic nature of the considered optimization algorithms, for all algorithm configurations the experiments are repeated 5 times and the best results are presented. Two stopping criteria are used: the maximum number of iterations (epochs) is set to 500 and optimization is stopped if there is no improvement of cost-function value in the last 50 epochs.

3.4. Parametrization methods

Optimization results for 4 different parametrization methods of the control policy are presented and discussed in Section 3.4.1–3.4.4, Section 3.5 compares the obtained results.

3.4.1. Parametrization method 1: Multiple start policy with maximum throttle opening

Parametrization which describes the control trajectory as single engine starts with constant, maximal throttle opening uses the smallest number of parameters. Since the strategy is used in global optimization problem and the number of parameters is equal to dimensionality of the searched space, this is an advantageous property which allows to solve it relatively fast. It is assumed that searched parameters are moments of time in which engine is started and switched off, respectively. The number of constraints for values of the decision variable vector is equal to the number of parameters. As a result, the searched vector can be written as

$$\mathbf{u} = [e_1 \ s_2 \ e_2 \ \dots \ s_N \ e_N]^T \quad (20)$$

where e_n is the time of the n -th engine operation span end, s_n is the time of the n -th engine operation span start and N is the number of engine operation spans.

It is required that the sequence of parameters is strictly monotonically increasing: engine must be in operation before it can be turned off. An additional constraint is imposed since the last parameter (the moment of last engine shutdown) has to be lower than the maximum time of trial according to the competition rules. A solution obtained by the PSO algorithm for the strategy with three engine uses is presented in Fig. 10. Convergence of the optimization procedure is shown in Fig. 11.

Experiments with four and five engine operation periods have been also performed. As discussed in Section 3.5, very similar control and state trajectories have been obtained with one main difference: the length of every single engine operation time is significantly reduced. It causes less variability of vehicle speed. Fig. 12 presents convergence of optimization procedure for obtaining strategy with five engine operation periods. Increasing number of parameters does not give noteworthy improvement of fuel efficiency. Unfortunately, approximately twice as many algorithm epochs have to be performed in order to stop by number of stall iterations condition. In contrast to the policy with three engine start-ups which involve 5 parameters, for strategy with five start-ups, 9 parameters are required. Greater dimension of parameters vector slows down algorithm convergence. In the first case a solution close to the final one is achieved after 40–50 epochs whereas as many

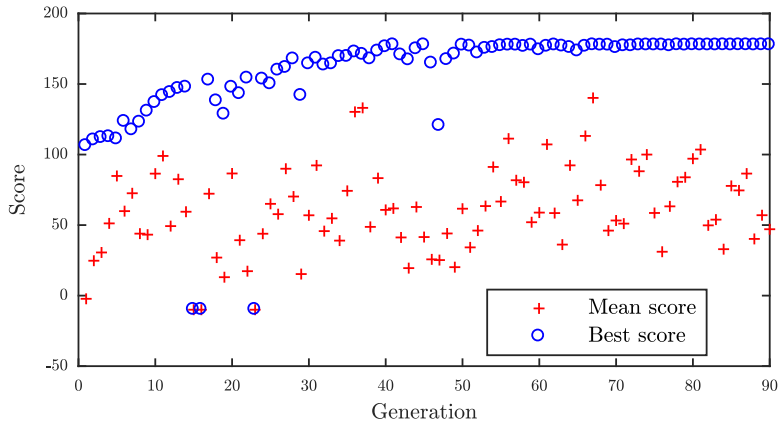


Fig. 11. Parametrization method 1: Convergence of the PSO algorithm to a solution obtained for 3 single starts policy.

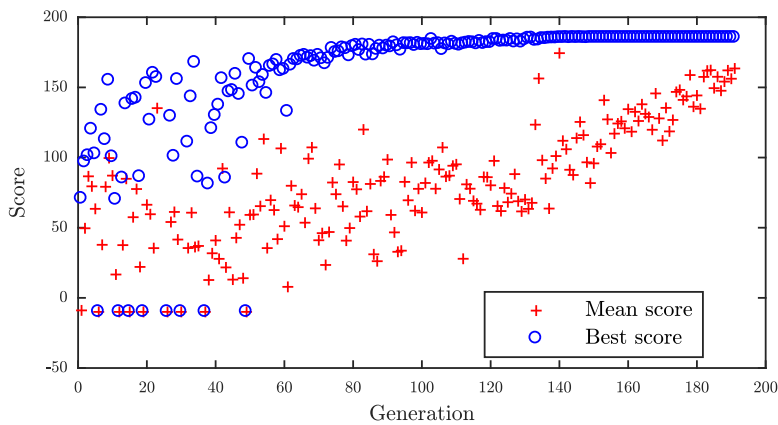


Fig. 12. Parametrization method 1: Convergence of the PSO algorithm to a solution obtained for 5 single starts policy.

as 120 epochs are necessary in the second one which means that numerical effort of the optimization process is increased approx. three times.

As a result of special vehicle design with well-reduced influence of the resistive forces and aerodynamic drag, engine power is many times greater than the power required to maintain a constant speed. If greater number of engine operation periods is required, it implies that every period has to be appropriately shortened in order to generate comparable mechanical energy amount.

Engine start sequence is characterized by approximately constant, fuel cost. Due to this fact, increasing the number of single engine start-ups does not give better efficiency, although the vehicle speed trajectory is steadier and does not reach high values.

It can be observed that in order to obtain favourable fuel consumption the value with this type of parametrization, the vehicle has to be accelerated to speed significantly higher than average speed and should coast to almost zero speed. Although such a driving strategy can be difficult to apply in typical driving scenarios, it is widely used during competitions of low fuel consumption vehicles.

3.4.2. Parametrization method 2: Multiple start policy with variable throttle opening

As an extension of the control trajectory discussed in Section 3.4.1, a trajectory with multiple periods of engine operation and variable level of throttle opening for each period is considered. This strategy slightly extends the vector of optimized parameters: there is one additional value (throttle opening) for each engine operation interval. As a result, the searched vector can be written as

$$\mathbf{u} = [a_1 \ e_1 \ s_2 \ a_2 \ e_2 \ \dots \ s_N \ a_N \ e_N]^T \quad (21)$$

where a_n is the amplitude of the n -th engine operation span, e_n is the time of the n -th engine operation span end, s_n is the time of the n -th engine operation span start and N is the number of engine operation spans.

In addition to the set of constraints used in the previous parametrization method, a constraint related to the throttle opening limits must be taken into account. As PAKS throttle uses butterfly valve, minimum value of opening is 0° and maximum value is 90° .

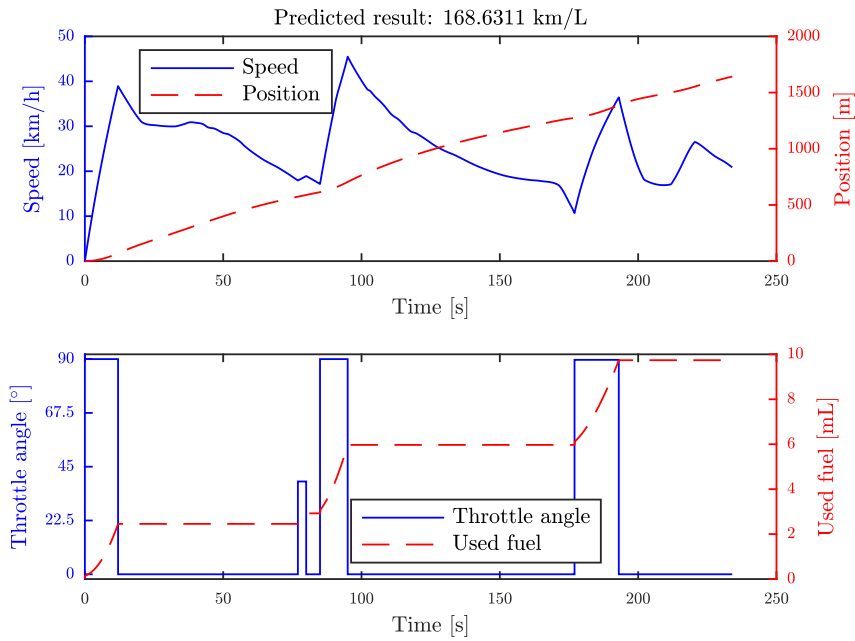


Fig. 13. Parametrization method 2: A solution obtained for 4 single starts with variable throttle opening policy.

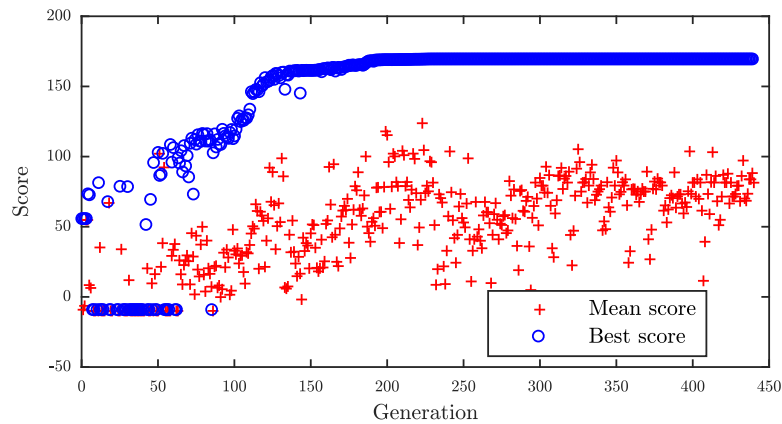


Fig. 14. Parametrization method 2: Convergence of the PSO algorithm to a solution obtained for 4 single starts with variable throttle opening policy.

Fig. 13 depicts state and control trajectories obtained by optimization of policy with four engine operation periods and variable throttle opening in each period. The proposed control policy is related to the one presented in Fig. 10 with constant full-throttle at engine operation periods. In the considered case, however, amplitude and duration of the second throttle operation is reduced which means that obtained strategies are quite similar. In the three other cases, throttle openings have maximal values allowed. It can be assumed that the maximum fuel efficiency is obtained in case of maximum throttle opening.

It can be observed that the absolute value of the cost function in the case of non constant throttle opening is several percent higher than in the case where throttle openings are elements of the searched vector. The difference between fuel consumption at the optimal and the maximal throttle point is not significant, but there is a significant change of optimization problem complexity. In this situation, a theoretically more efficient strategy with near-maximum throttle opening does not give better fuel-efficiency results as optimization ratio of convergence is lower (especially when maximal time of execution constraint is imposed) and there is no guarantee that it converges to the same local minimum in the extended search space.

Fig. 14 depicts a solution history over optimization process execution. It can be compared with Fig. 12 which shows the same data for obtaining trajectory with one more engine operation period and constant throttle openings. The extended vector of parameters in the first case causes that the PSO algorithm reaches immediate surroundings of solution approx. 50 epochs later than in the case of maximum throttle opening. Assuming that in this case the optimization process can

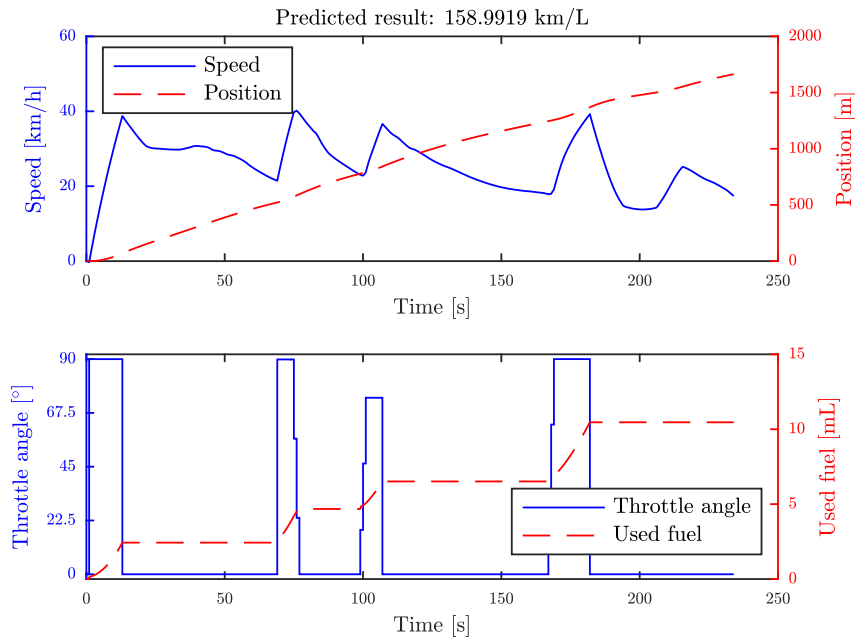


Fig. 15. Parametrization method 3: A solution obtained for 4 single starts with variable and continuous throttle opening policy.

be aborted after approx. 140 epochs (more epochs do not lead to any significant improvement of the solution) and in the current case this value is approx. 190 epochs, elongation of optimization by some 37% is observed, whereas length of the decision vector increases only by approx. 20%.

3.4.3. Parametrization method 3: Multiple starts with variable throttle opening and continuous opening trajectory

Two previously used parametrization methods assume that zero rising and falling time of a control signal can be reached. In many applications it is impossible to obtain such fast signal changes. It should be examined if it is possible to increase energy efficiency by control policy based on slowing control slopes. This type of control trajectory parametrization also complicates the optimization task. For each engine operation period, two additional parameters: rising and falling time have to be calculated. As a result, the searched vector can be written as

$$\mathbf{u} = [r_1 \ a_1 \ f_1 \ s_2 \ a_2 \ f_2 \ \dots \ s_N \ r_N \ a_N \ f_N]^T \quad (22)$$

where r_n is the rising time of the n -th engine operation span, a_n is the amplitude of the n -th engine operation span, f_n is the falling time of the n -th engine operation span end, s_n is the time of the n -th engine operation span start and N is the number of engine operation spans.

The set of constraints imposed on optimized parameters is also similar to the previously considered cases. They are implicated by the logical order of events. Firstly, engine has to be started, used for some period of time and stopped. After this sequence, the next one can be performed. There are also limits for minimum and maximum throttle opening angle.

The trajectories of speed, position, used fuel and throttle angle obtained as solution of optimization task in the case of four engine operation periods and non-constant throttle angle value are presented in Fig. 15. The control sequence, despite more optimized parameters, is analogous to the trajectory for the case of maximal and zero rising/falling times throttle angle operation periods. Slopes in the control trajectory are steep even though there is only one constraint which forces non-negative rising and falling time. In three of four engine operation periods, maximum throttle openings are proposed as optimal. This case shows that more parameters involved into optimization, which introduces more flexibility in obtaining optimal control sequence, does not give significantly better final results.

It should be pointed out that computation of the control strategy characterized by more parameters also causes bigger numerical effort in comparison with two previous parametrization methods. Convergence of optimization procedure to described solution is presented in Fig. 16. A trajectory specified by values of 15 parameters results in convergence to the final solution in approx. 320–330 epochs. In comparison with the previous case which uses 11 parameters it means that computational burden is increased by 100% while the number of searched parameters is increased only by approx. 37%.

Knowing that the zero rising/falling times are feasible in the engine control task, it can be assumed that non-zero times strategy does not give better solution. Values of the cost-function in both cases are comparable, but time necessary to find the solution significantly increases which is caused by an expanded vector of parameters. It can be stated that a much more appropriate choice is to use the control trajectory described by the shorter vector of decision variables which gives the solution faster.

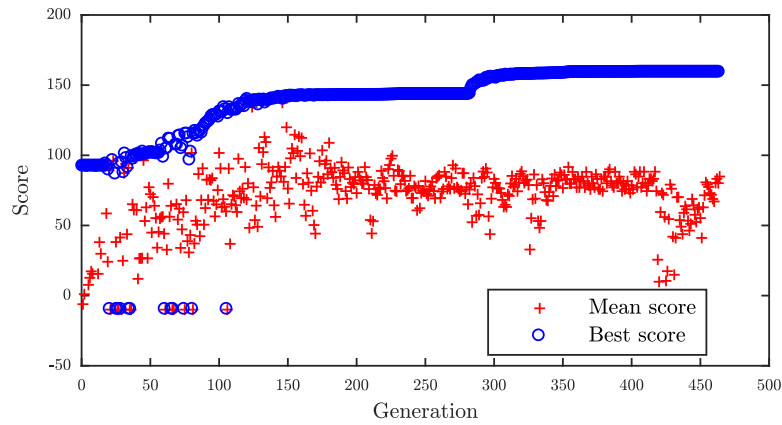


Fig. 16. Parametrization method 3: Convergence of the PSO algorithm to a solution obtained for 4 single starts with variable and continuous throttle opening policy.

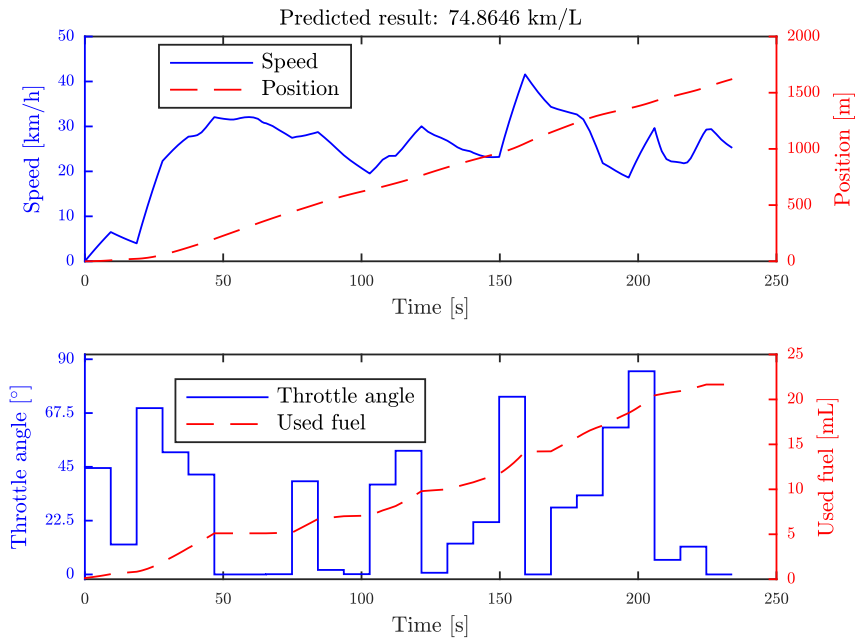


Fig. 17. Parametrization method 4: a solution obtained for 25 changes of throttle opening and continuous engine operation.

3.4.4. Parametrization method 4: Changing throttle opening with continuous engine operation

The fourth type of parametrization uses trajectories in which engine is started at the beginning of the route and stay running until its end. This control policy is highly different from the previously considered ones. It allows to save the amount of fuel which is consumed for several engine start-ups in strategies with periodical propulsions. On the other hand, mechanical energy generated by engine working in optimal operating point during the trip many times exceeds the amount necessary to its overcoming. The proposed parametrization assumes that decision variables are values of throttle angle at several, equal time periods, which in sum, cover the maximal allowed time of trip. This time is split into N sections, which is also the number of decision variables. As a result, the searched vector can be written as

$$\mathbf{u} = [a_1 \ a_2 \ \dots \ a_N]^T \quad (23)$$

where a_n is the amplitude of the n -th engine operation span and N is the number of engine operation spans. Cubic constraints are imposed on those values to force minimum (0°) and maximum (90°) throttle angle.

During experiments control strategies with: 5, 10, 15, 20, 25, 30, 35 and 45 constant control value periods are considered. They correspond to splitting trip for 48 s–6 s time periods. Example solution results obtained with this class of strategies is presented in Fig. 17. It can be observed that the optimization procedure leads to a solution in which for some control periods throttle angle is close to its minimal or maximal values. On this basis, it can be supposed that optimization converges to a solution similar to the strategy with periods of engine operation. Full time span divided into only several periods leads to its

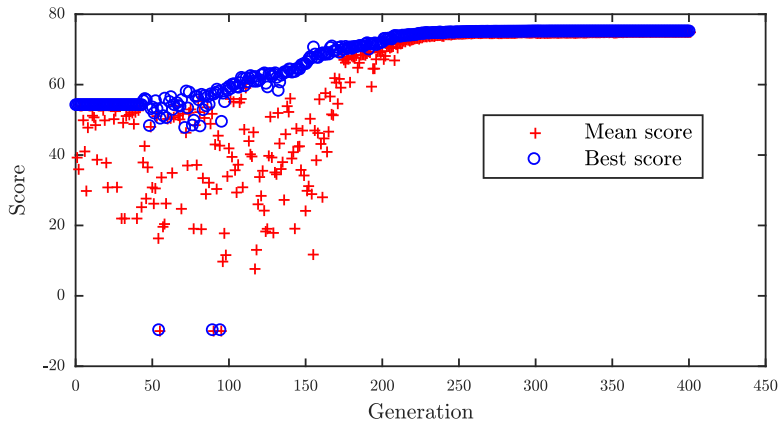


Fig. 18. Parametrization method 4: convergence of the PSO algorithm to a solution obtained for 25 changes of throttle opening and continuous engine operation.

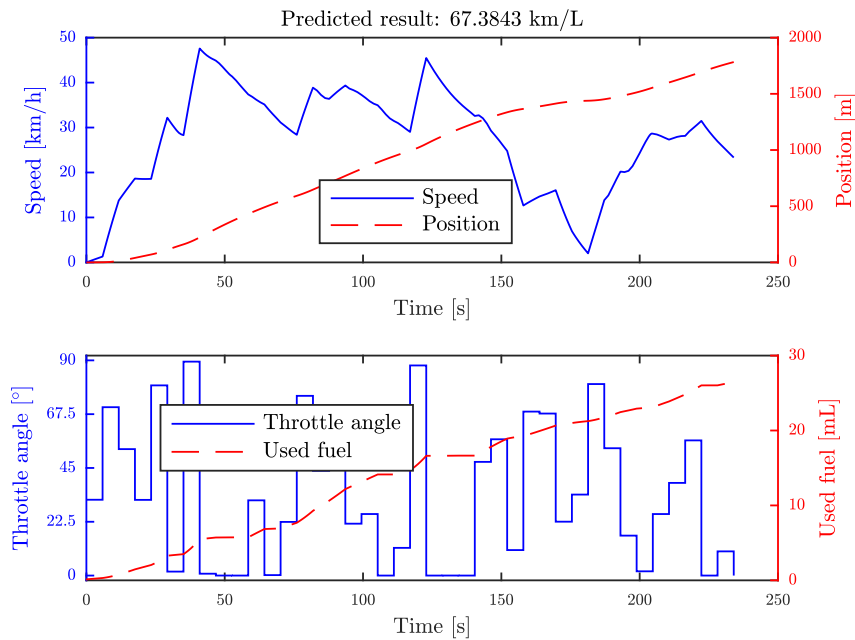


Fig. 19. Parametrization method 4: a solution obtained for 40 changes of throttle opening and continuous engine operation.

inadequate length. In contrast, discretization with short constant control periods allows to precise match control value, but it may be computationally expensive to obtain the solution since a long vector of decision variables results in complicated optimization process which takes significant larger resources.

Convergence characteristics in case of the path divided into 25 equal time periods is presented in Fig. 18. In the initial epochs a temporary best solution, which is randomly chosen before algorithm start, is not revised for about 50 iterations. In the next phase, solution slowly converges to the final one. It can be supposed that the whole swarm, at the beginning, is focused on a randomly chosen local minimum and after several iterations a better solution is found. Similar character of convergence history is obtained in cases with more parameters (shorter periods of constant control value). Increasing decision variables vector length leads to relatively chaotic solutions. An example solution for strategy with time split into 40 periods is presented in Fig. 19. Greater dimension of feasible solutions hyperplane induces more local maxima. Evolutionary algorithms allow to relatively fast find maxima in case of a small number of them. However, generally, there is no guarantee of finding global optimal point. Especially, if extended vector of decision variables is used, in order to obtain comparable solution, adequate enhanced swarm has to be used, which lengthens calculation time. It can be concluded that evolutionary algorithms are able to calculate values only for a limited number of parameters. In order to searching solutions from large spaces, EA population needs to be enlarged, but this action is not costless: more memory and computation time resources are necessary.

Table 3

Comparison of analyzed control policies.

Parametrization method	Test number	Description	Number of decision variables	Score (fuel efficiency) [km/L]	Computation time [s]
1	1	3 single starts with maximum throttle opening	5	177.6	21.3
1	2	4 single starts with maximum throttle opening	7	176.5	330
1	3	5 single starts with maximum throttle opening	9	185.4	180
2	4	3 single starts with variable throttle opening	8	177.8	288
2	5	4 single starts with variable throttle opening	11	168.6	164
3	6	3 single starts with maximum throttle opening and variable rising/falling time	8	175.9	276
3	7	4 single starts with maximum throttle opening and variable rising/falling time	11	172.4	253
3	8	5 single starts with maximum throttle opening and variable rising/falling time	14	174.0	565
3	9	3 single starts with variable throttle opening and variable rising/falling time	11	167.3	563
3	10	4 single starts with variable throttle opening and variable rising/falling time	15	159.0	492
4	11	5 changes of throttle opening and continuous engine operation	5	75.7	45.7
4	12	10 changes of throttle opening and continuous engine operation	10	103.8	41.7
4	13	15 changes of throttle opening and continuous engine operation	15	84.4	41.1
4	14	20 changes of throttle opening and continuous engine operation	20	77.8	2.15
4	15	25 changes of throttle opening and continuous engine operation	25	74.9	40.6
4	16	30 changes of throttle opening and continuous engine operation	30	84.4	43.0
4	17	35 changes of throttle opening and continuous engine operation	35	54.2	55.3
4	18	40 changes of throttle opening and continuous engine operation	40	67.4	2.01

3.5. Comparison of obtained control policies

As a result of performed experiments, 18 proposed control policies have been examined. The simplest ones need only several parameters whereas the most complex uses several dozen of them. The results are presented in Table 3 in terms of the number of optimized decision variables and the best solution obtained.

The best result obtained in simulations is 185.4 km/L. The best result achieved by the PAKS vehicle during the SEM2017 competition is 190 km/L by using the strategy with three periods of engine operation and near maximum throttle angle. Some 5% difference for result is acceptable taking into account that the vehicle model is significantly simplified. A precise value of final fuel efficiency obtained by evolutionary algorithm is not the most important parameter: it gives information about approximated result, but more essential is the possibility to compare proposed trajectories.

Control policies with periodically used engine, in general, are more effective than continuous engine operation with throttle angle control. This result confirms findings of some works (i.e. [15]) and specific characteristics of internal combustion engine efficiency.

The results obtained in the class of solutions with repeated sequence of accelerating and coasting periods are comparable. There is no possible to assess if differences are caused by real differences between policies or only by inaccurate modelling or optimization process random factors.

As the results obtained in all experiments are comparable to those achieved in real runs, as trajectories as simple as possible can be taken into account in future research. Optimization calculations based on lower dimensional spaces are performed significantly faster and does not require enlarged hardware resources. For these reasons, the strategy with three periods of engine operation and maximum throttle has been selected for further examination. For the problem of passing any other route, the number of this periods have to be considered individually. It is highly related to the road profile, thus the chosen number is appropriate only for the PAKS vehicle passing the SEM2017 route.

Taking into account optimization calculation time (the last column in Table 3), similar conclusions can be drawn. The solutions returned by parametrizations with greater number of decision variables give similar fuel efficiency caused by maximum or near to maximum engine load, but the time of computation is longer. Significantly low execution times are achieved when the parametrizations which split the whole simulated period into subperiods with constant control value are used. Those methods do not converge as a result of inadequate length of the periods in comparison with the process dynamics. Similarly to the case when infeasible points are found in the consecutive iterations, cost-function evaluations are aborted (what significantly reduces total numerical cost of the optimization) and a small constant value is applied.

3.6. Comparison of optimization algorithms

Five different evolutionary strategies have been examined during optimization of the control policy with three single engine operation periods and maximum throttle angle (5 parameters). Test for each algorithm have been repeated ten times.

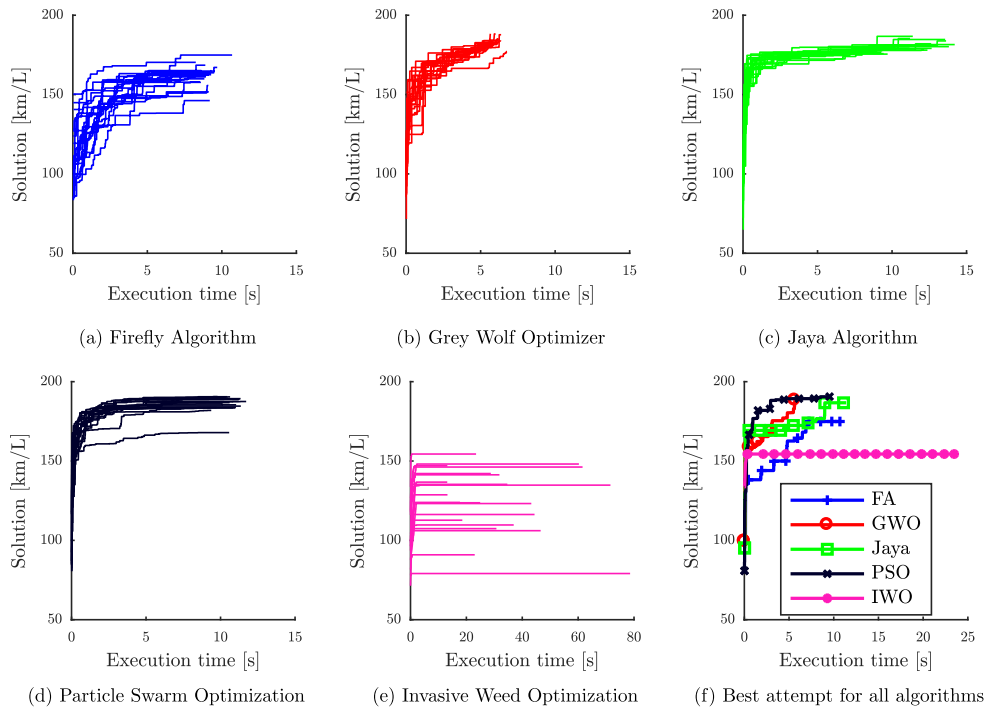


Fig. 20. Convergence of compared evolutionary algorithms.

For all algorithms, population size has been set to 200 individuals and 1000 epochs (iterations) have been performed. Results of all trials are presented in Fig. 20.

The IWO algorithm is characterized by the worst convergence and performance. Despite the tests with different set of parameters and increased number of particles, it is not possible to obtain a satisfactory rate of algorithm convergence. Not directly controllable number of particles causes also relative long time of execution.

The best scores are achieved by Jaya, PSO and GWO methods. Solutions obtained by the FA approach are unsatisfactory and differ in the consecutive attempts. Single attempts in cases of PSO and GWO also could lead to noticeably worse solution. Jaya execution lasts significantly longer, but the final solution is very similar for all performed tests. The fastest is the GWO algorithm.

Moreover, in all performed repetitions of Jaya, PSO, GWO and Firefly algorithms, time of execution is approximately equal. High repeatability is an essential factor in context of real-time decision system based on optimization procedures. For typical numerical problem, computational complexity can be determined, typically on the base of its inputs. Evolutionary algorithm which engages random factors, in the worst case cannot return solution in finite time. It can lead to difficulties in implementation. The considered set of test cases shows that the obtained solution time is almost constant for the specified set of stop conditions and algorithms used.

Extended test series has been performed for PSO and GWO algorithms for different numbers of individuals and epochs. The obtained data are presented in Fig. 21. For various numbers of epochs and individuals as many as 10 attempts of optimization process have been carried out and mean time execution and the value of the final result are considered. The solutions returned by both methods are similar in terms of the cost-function value. It should be pointed that generally the GWO algorithm returns solution twice faster than the PSO. This difference can be caused by a method of generating candidates for the following iterations. In case of a not feasible candidate, the cost-function is not evaluated and a substantial amount of time is saved. It can be also assumed that using more than 750 epochs and 75 particles does not lead to obtaining noteworthy better final solution.

In general, it can be also assumed that satisfactory order of solution accuracy can be obtained by the GWO approach very fast, within 5 s. As the time allowed by SEM rules for completing one lap route (above 1600 m) is about 230 s, it gives possibility for real-time optimization process execution with more than 40 strategy refreshments per single lap.

3.7. Comparison with gradient-based algorithm

Although heuristic optimization algorithms are able to efficiently solve the optimization task (13), it is necessary to investigate whether or not the classical gradient-based methods may be used as well. It is because numerous very efficient methods for solving nonlinear problems are available [1,24], which may turn out to be faster than heuristic optimization

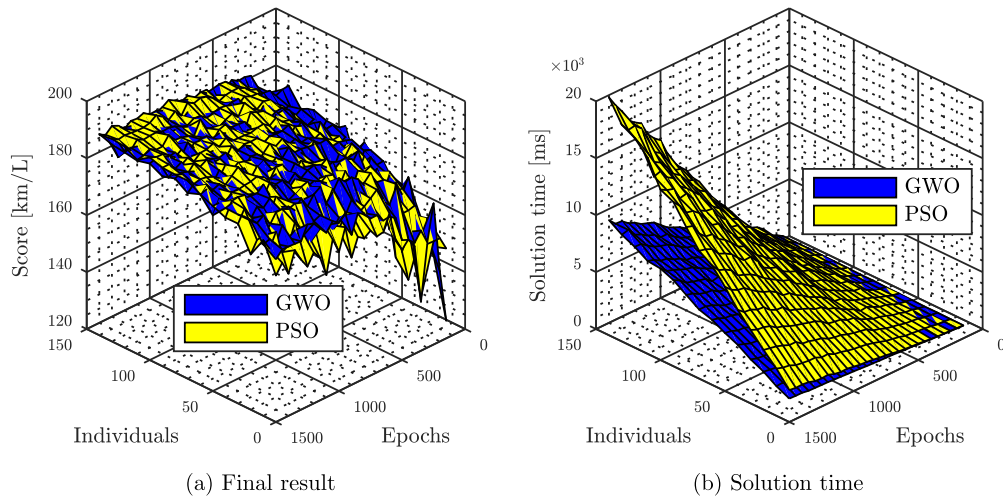


Fig. 21. Detailed comparison between the GWO and PSO algorithms.

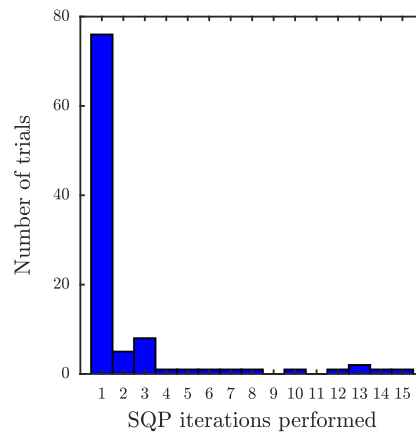


Fig. 22. The number of trials vs. the actual number of iterations performed in 100 executions of the SQP algorithm from 100 randomly generated feasible points.

methods. In particular, the Sequential Quadratic Programming (SQP) algorithm [26,27] is considered as a very good method for solving complicated constrained nonlinear optimization problems because of its efficiency (relative low computational complexity) and robustness. The SQP algorithm has numerous applications, e.g.: model predictive control [10], active noise reduction [12], controller tuning [14], model identification [19] and optimal design [43,47]. Unlike all heuristic optimization methods considered in this work, the SQP algorithm does not consider a set of candidate solution vectors (individuals), but it iteratively finds only one solution vector. It is generated using the previous one as well as the optimization direction determined by the first- and the second-order derivatives (these are evaluated numerically in this work).

Since the SQP algorithm uses gradients, the rudimentary formulation of the general optimization problem (13) must be modified. The classical death-penalty function leads to non-differentiability of the cost-function in neighbourhood of constraints, which is not a problem for the heuristic optimization because they do not use gradients. However, this problem must be solved when the SQP algorithm is used. The external penalty function used in the case of the heuristic algorithms is replaced by a linear one. It makes it possible to take into account the constraints (13h) and (13i).

To compare results obtained by the SQP method with those found by the evolutionary algorithms presented in Section 3.6, only parametrization with 3 single engine operation spans with the maximum throttle is considered. As many as 100 trials of the SQP algorithm have been performed. Each algorithm execution starts from a randomly generated point. Initial solution generation method ensures that all performed engine switching time-points occurs in the logical order (i.e. the first engine shutdown occurs after the first start, the second engine start after the first shutdown etc.). The results and observations concerned with the SQP algorithms are the following:

- a) For almost 80% of the initial points the SQP algorithm is stopped in the first iteration since it is not able to find any better solution than the initial point. Fig. 22 depicts the number of trials of the SQP algorithm vs. the actual number of iterations in 100 trials.

Table 4
The best results obtained by the SQP algorithm.

Score (fuel efficiency) [km/L]	Iterations
99.22	1
97.96	1
97.87	1
97.05	1
95.71	1
95.70	1
95.44	1
95.28	1
94.62	1
94.27	1

- b) Some 20% of the initial points make it possible to continue optimization, but after a few iterations calculations are stopped. The maximal number of iterations is 15.
- c) Some starting points do not lead to any solution improvement in the consecutive iterations which is caused by trying to finding new solutions in the infeasible region. The majority of solution trials which end in more than one iteration do not achieve solutions which satisfy the constraints (13h) and (13i).
- d) The penalty factor which linearly depends on the constraint (13i) is so huge that at the final phase of the optimization process any feasible solution (which leads to reaching finish line in the time limit) is not returned, even if a proportional factor of the penalty is significantly reduced.
- e) The best ten results (in terms of fuel efficiency) are presented in Table 4. Each execution which returns valuable solution is stopped after the first iteration of the SQP algorithm. It leads to conclusion that no solution improvement is performed by the SQP method and the best solutions are obtained randomly rather than from the real optimization process. It should be pointed that the solutions obtained this way are at least 50% worse than in case of evolutionary algorithms presented in Section 3.6 (Tab. 3).
- f) In some of the SQP optimization trials a negative value of fuel efficiency is obtained. Of course, such a result is physically impossible, it is caused by the value of the linear penalty function which is applied when the constraints (13h)–(13i) are not satisfied.

Those constraints are responsible for enforcement of achieving the finish line within the imposed time limit. Otherwise, the value proportional to the distance lack is subtracted from the achieved fuel efficiency. In those cases, the penalty component (the subtracted value) of the cost-function is huge and no satisfactory solution is found, even if the proportional coefficient of the penalty is significantly reduced.

It should be also pointed out that according to the Eco-marathon rules, the better choice is to reach the finish line faster which implies worsening the result (fuel efficiency) than not to reach it during the time limit which leads to cancellation of a trial with zero fuel efficiency.

All things considered, although the gradient-based SQP algorithm is reported in the literature to very efficiently solve numerous constrained optimization problems, it cannot be used to find the optimal control strategy for a low fuel consumption vehicle engine. The SQP algorithm is not able to solve the considered optimization task. The nature of the optimization problem (13) is different from that of the classical tasks, e.g. used in model identification [19]. Hence, application of heuristic optimization algorithms is fully justified.

4. Development of decision system

The proposed system architecture is presented in Fig. 23. Since the optimization process is based on the vehicle model, it is highly reasonable to introduce mechanisms of its correction due to model inaccuracies. Observable differences between predicted and measured vehicle responses for applied control strategy can be used as inputs for disturbance detection module. Because some model parameters subsets are strongly correlated (for example: the set of propulsion force coefficients), a neural network acting as an parameter approximator may be used. According to Fig. 23, the desired control strategy is obtained by the optimization algorithm based on the actual vehicle state (4) and model responses (2) for a candidate strategy (1). The established control policy is applied to the real vehicle and its model (3). Responses of them (4 and 5) are feedback signals which can be used for calculation of the model correction (based for example on neural network classifier) which should be applied (6).

All system components have to work as real-time applications. As long as developing model correction module may be highly sophisticated and can be done separately, consideration about proper optimization algorithm selection has been made. For application purposes, various types of calculation engine are considered.

First of them is a classical PC-class computer located outside the vehicle. Because the PAKS is equipped with a telemetry data link, it allows to use a powerful x64 processor. This platform allows for easy adaptation of designed procedures and algorithms as PC was used for experiments made so far. Therefore, only a wrapper for numerical procedures and communication interface should be developed. The main disadvantage of this approach is unreliability of the data link. Based on

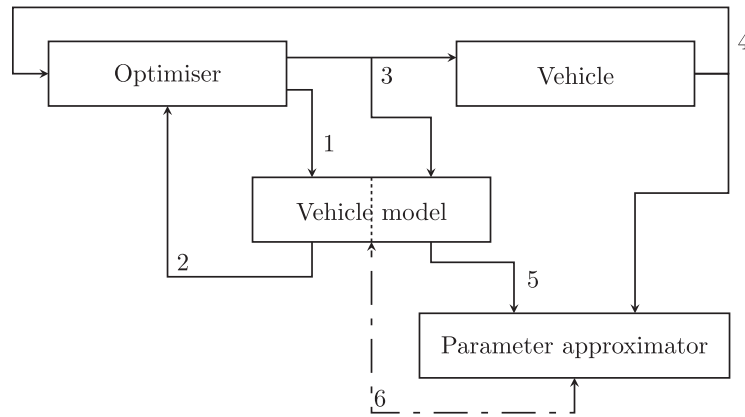


Fig. 23. The proposed structure of the decision system.

publicly available 433 MHz ISM band, transmission is exposed to interferences. Due to limited by law power of transmitter also real range of used radiomodems is significantly reduced in urban areas. Those limits are not dominant in context of usage as a system of experimental vehicle, but the system is not scalable for multiple vehicles in any straightforward way. Even so, this disadvantages may be reduced by growing wireless technologies in near future and it will give possibilities for usage of cloud computing architecture.

An alternative way to implement on-line optimization is to engage a dedicated on-board embedded system. The main advantage of this approach is independence from external infrastructure which can be achieved if it can be assumed that the track geometry is known. Due to significant computational effort connected with engine usage optimization, a few hardware solutions may be considered.

One of them is a single board computer, for example, Raspberry Pi or similar. Popular ARM Cortex A53 based systems can be relatively easily used as an optimization platform whose processor can achieve a clock frequency comparable to a classic PC. Furthermore, the workflow typical of all considered heuristic optimization algorithms assumes that solution space is searched according to straightforward rules. It can be expected that the cost-function is evaluated multiple times independently and it is the most computationally complex part of whole solution process. This feature allows for successful parallelization of the optimization procedure, which can be achieved using an OpenMP interface.

In this way, duration of single optimization process will be improved, but devices family, typically based on Linux family OS, does not provide any tools to force real-time execution of software. It should be pointed out that random character of evolutionary algorithms is not ensures returning solution in certain time. In combination with a general purpose operating system it can leads to unpredictable behaviour.

The most robust platform can be provided by an especially designed hardware embedded system. Fast microcontrollers equipped with floating point units allows to develop bare-metal application, which provides tools for full software flow control. Especially, hardware (timer) interrupts can be applied for limiting maximum refresh time for optimal control policy. An extension of this concept may be to use a dedicated device structure, e.g. Field Programmable Gate Array (FPGA).

Implementation of ODE integration Cash-Karp routine does not require complicated hardware resources. In connection with a reconfigurable FPGA device and possibility of algorithms' parallelization, it allows to use arrangement of increased number of relative simple, parallel ODE coprocessors for fast optimization without using high-performance, dedicated single processor.

5. Conclusions

The main goal of this work is to formulate and compare several possible control strategies leading to low fuel consumption of a vehicle. For this purpose a mathematical model of vehicle dynamics has been derived from basic physical principles and its parameters have been estimated using real data, recorded for the PAKS car.

4 different parametrization methods of control policy are formulated. Generally, the strategies assuming periodic engine operation with high power are more efficient than these using continuous operation with partial power usage which is caused by enormous high engine power relative to dissipation power developed by resistive forces. The results obtained by this approach are comparable and slightly better than those achieved during the Shell Eco-marathon competition with a human controlled vehicle. The best results obtained in simulations are approximately 180 km/L. They are obtained by the PSO algorithm with limited algorithm operation time, assuming the parametrization strategy with periods of engine operation and coasting. The strategies with continuous engine operation are typically significantly less efficient and the obtained results do not exceed 110 km/L. With relaxed restrictions for algorithm execution time and increased number of iterations PSO, GW and Jaya algorithms may return solutions equal to approx. 195 km/L. Those results are better than for

human-control approach and they are independent of driver experience and gives possibility of easy recalculation after significant disturbances occur.

Efficiency of as many as 5 evolutionary optimization methods is compared in terms of the best solution obtained and calculation time. The most promising results are obtained by the Jaya, and in particular, the PSO and GWO approaches which are significantly faster in comparison to other methods. Despite the expanded test series performed using the Firefly and IWO methods, they cannot give similar results. Additionally, inefficiency of the classical gradient-based optimization method is demonstrated.

Although in this work a specific, experimental vehicle PAKS is considered, the presented methodology is not limited to this particular case. The discussed methods may be easily adopted for any vehicle. In such a case a mathematical model of the vehicle model should be developed, especially with extensions related to specific application, e.g. automatic selection of the optimal gear for vehicles with gearbox.

All things considered, the results of simulations carried out show that it is possible to effectively solve the problem of fuel usage optimization in cars. During future research it is planned to develop a dedicated hardware embedded system (based on a powerful microcontroller) which could solve on-line the optimization task and perform extended tests for a prototype urban car. In contrast to simulations, in practice some mechanisms of model on-line adaptation may be necessary.

Declarations of interest

None.

References

- [1] M.S. Bazaraa, J. Sherali, K. Shetty, *Nonlinear Programming: Theory and Algorithms*, Prentice Hall, 1993.
- [2] R. Bellman, *Dynamic Programming*, Princeton University Press, 1957.
- [3] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing* 172 (2016) 371–381.
- [4] D.T.M. Espindola, *Real-time optimal control of a low consumption electric vehicle*, Université de Lorraine, 2014 Phd thesis.
- [5] M. Fahad, F. Aadil, Z. Rehman, S. Khan, P.A. Shah, K. Muhammad, J. Lloret, H. Wang, J.W. Lee, I. Mehmood, Grey wolf optimization based clustering algorithm for vehicular ad-hoc networks, *Computers and Electrical Engineering*, in press.
- [6] W. Fang, J. Sun, H. Chen, X. Wu, A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population, *Inf. Sci.* 330 (2011) 19–48.
- [7] Y.Z. Feng, W. Yu, W. Chen, K.K. Peng, G.F. Jia, Invasive weed optimization for optimizing one-agar-for-all classification of bacterial colonies based on hyperspectral imaging, *Sensors Actuators B* 269 (2018) 264–270.
- [8] IEA: CO2 emissions from fuel combustion highlights, 2016, Paris: International Energy Agency (IEA).
- [9] R.N. Jazar, *Tire dynamics*, Vehicle Dynamics, Springer, Cham, 2017.
- [10] A. Kelman, F. Borelli, Bilinear model predictive control of a HVAC system using sequential quadratic programming, in: *IFAC Proceedings Volumes 44.1*, the 18th World Congress of The International Federation of Automatic Control Milano (Italy), 2011, pp. 9869–9874.
- [11] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1948.
- [12] W.U. Khan, Z.F. Ye, N.I. Chaudhary, M.A.Z. Raja, Backtracking search integrated with sequential quadratic programming for nonlinear active noise control systems, *Appl. Soft Comput.* 73 (2018) 666–683.
- [13] X. Lei, F. Wang, F.X. Wu, A. Zhang, W. Pedrycz, Protein complex identification through Markov clustering with firefly algorithm on dynamic protein-protein interaction networks, *Inf. Sci.* 329 (2016) 303–316.
- [14] X.m. Liang, S.C. Li, A.B. Hassan, A novel PID controller tuning method based on optimization technique, *J. Central South Univ. Technol.* 17 (2010) 1036–1042.
- [15] S.E. Li, H. Peng, K. Li, J. Wang, Minimum fuel control strategy in automated car-following scenarios, *IEEE Trans. Vehicular Technol.* 61 (2012) 998–1007.
- [16] Y. Lu, D. Yan, J. Zhu, D. Levy, A variant with a time varying PID controller of particle swarm optimizers, *Inf. Sci.* 297 (2015) 21–49.
- [17] X. Luo, L. Cao, L. Wang, Z. Zhao, C. Huang, Parameter identification of the photovoltaic cell model with a hybrid jaya-NM algorithm, *Optik: Int. J. Light Electron Opt.* 171 (2018) 200–203.
- [18] H.T. Luu, L. Nouvelière, S. Mammar, Dynamic programming for fuel consumption optimization on light vehicle, in: *IFAC Proceedings Volumes 43.7*, 6th IFAC Symposium on Advances in Automotive Control, 2010, pp. 372–377.
- [19] M. Ławryńczuk, Identification of wiener models for dynamic and steady-state performance with application to solid oxide fuel cell, *Asian Journal of Control*, in press, <https://doi.org/10.1002/asjc.2038>.
- [20] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, *Ecol. Inf.* 1 (2006) 355–366.
- [21] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [22] L. Mussi, F. Daolio, S. Cagnoni, Evaluation of parallel particle swarm optimization algorithms within the CUDA™ architecture, *Inf. Sci.* 181 (2011) 4642–4657.
- [23] O. Nelles, *Nonlinear System Identification. From Classical Approaches to Neural Networks and Fuzzy Models*, Springer, Berlin, 2001.
- [24] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 2011.
- [25] R.E. Perez, K. Behdinan, Particle swarm approach for structural design optimization, *Comput. Struct.* 85 (2007) 1579–1588.
- [26] M.J.D. Powell, Variable metric methods for constrained optimization, in: A. Bachem, M.G.A. tschel, B. Korte (Eds.), *Mathematical programming: the state of the art*, Springer, New York, 1983, pp. 288–311.
- [27] M.J.D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, in: G.A. Watson (Ed.), *Numerical Analysis, Lecture Notes in Mathematics*, volume 630, Springer, Dundee, 1978, pp. 144–157.
- [28] W.H. Press, S.A. Teukolsky, A. Saul, W.T. Vetterling, T. William, B.P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Cambridge University Press, 2007.
- [29] R.V. Rao, A. Saroj, Economic optimization of shell-and-tube heat exchanger using jaya algorithm with maintenance consideration, *Appl. Thermal Eng.* 116 (2017) 473–487.
- [30] R.V. Rao, Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems, *Int. J. Ind. Eng. Comput.* 7 (2016) 19–34.
- [31] J.L. Ravipudi, M. Neebha, Synthesis of linear antenna arrays using jaya, self-adaptive jaya and chaotic jaya algorithms, *International Journal of Electronics and Communications* 92 (2018) 54–63.
- [32] H.Y. Sang, P.Y. Duan, J.Q. Li, An effective invasive weed optimization algorithm for scheduling semiconductor final testing problem, *Swarm Evol. Comput.* 38 (2018) 42–53.
- [33] J.J. Santin, C.H. Onder, J. Bernard, D. Isler, P. Kobler, F. Kolb, N. Weidmann, L. Guzzella, The world's most fuel efficient vehicle: Design and development of pac car II. vdf hochschulverlag AG an der ETH zürich, 2007.

- [34] S. Eco-marathon, Shell global, Web. 7 Oct 2018. <http://www.shell.com/energy-and-innovation/shell-ecomarathon.html>.
- [35] Student association for vehicle aerodynamics (in polish), wikipedia. wikimedia foundation, 2018, http://pl.wikipedia.org/w/index.php?title=Studenckie_Ko%C5%82o_Aerodynamiki_Pojazd%C3%B3w&oldid=54197069.
- [36] P. Sulikowski, R. Maroński, Quasi-periodic control technique for minimizing fuel consumption during record vehicle competition, *Arch. Mech. Eng.* 60 (2) (2013) 185–197.
- [37] C.L. Sun, J.C. Zheng, J.S. Pan, An improved vector particle swarm optimization for constrained optimization problems, *Inf. Sci.* 181 (2011) 1153–1163.
- [38] Transportation research board, tires and passenger vehicle fuel economy: Informing consumers, improving performance – special report 286, 2006, Washington DC.
- [39] K. Uyar, E. Ülker, B-Spline curve fitting with invasive weed optimization, *Appl. Math. Model.* 52 (2017) 320–340.
- [40] G. Vanier, London 2017 track information for students, shell eco-marathon, 2016. <http://www.shell.com/energy-and-innovation/shell-ecomarathon/europe/for-europe-participants/important-information.html>.
- [41] H. Wang, W. Wang, Z. Cui, X. Zhou, J. Zhao, Y. Li, A new dynamic firefly algorithm for demand estimation of water resources, *Inf. Sci.* 438 (2018) 95–106.
- [42] A. Weißmann, D. Görges, X. Lin, Energy-optimal adaptive cruise control combining model predictive control and dynamic programming, *Control Eng. Practice* 72 (2018) 125–137.
- [43] C.L. Xiao, H.X. Huang, Optimal design of heating system in rapid thermal cycling blow mold by a two-step method based on sequential quadratic programming, *Int. Commun. Heat Mass Transf.* 96 (2018) 114–121.
- [44] X.S. Yang, *Nature-inspired Metaheuristic Algorithms*, Luniver Press, Frome, 2008.
- [45] K. Yu, J.J. Liang, B.Y. Qu, X. Chen, H. Wang, Parameters identification of photovoltaic models using an improved JAYA optimization algorithm, *Energy Convers. Manag.* 150 (2017) 742–753.
- [46] Y. Zhang, X.F. Song, D.W. Gong, A return-cost-based binary firefly algorithm for feature selection, *Inf. Sci.* 418–419 (2017) 561–574.
- [47] J.T. Zhang, H. Zhu, S.Q. Zhou, C.S. Zhao, Optimal design of a rod shape ultrasonic motor using sequential quadratic programming and finite element method, *Finite Elements Anal. Des.* 59 (2012) 11–17.
- [48] Z. Zhou, R. Zhang, Y. Wang, Z. Zhu, J. Zhang, Color difference classification based on optimization support vector machine of improved grey wolf algorithm, *Optik: Int. J. Light Electron Opt.* 170 (2018) 17–29.



Jakub Sawulski was born in Ciechanów, Poland, in 1993. He obtained his B.Sc. in robotics in 2016 from Faculty of Aerospace and Power Engineering and M.Sc. in automatic control in 2018 from Faculty of Electronics and Information Technology, both at Warsaw University of Technology. In 2016 he won the scholarship of The Minister of Science and Higher Education in Poland for outstanding scientific achievements. Currently, he carries out research towards his Ph.D at the Institute of Control and Computation Engineering at the same University. He is a member of Students' Association for Vehicles Aerodynamics and has participated in Shell Eco Marathon for the last four years resulting in podium places in Urban Concept Gasoline category. His research interests include: microcontrollers, optimal control, modeling and simulation, soft computing methods, in particular evolutionary algorithms.



Maciej Ławryńczuk was born in Warsaw, Poland, in 1972. He obtained his M.Sc. in 1998, Ph.D. in 2003, D.Sc. in 2013, in automatic control, from Warsaw University of Technology, Faculty of Electronics and Information Technology. Currently he is employed at the same university at the Institute of Control and Computation Engineering as an associate professor. He is the author or a co-author of 6 books and more than 100 other publications, including more than 30 journal articles. His research interests include: advanced control algorithms, in particular Model Predictive Control (MPC) algorithms, set-point optimization algorithms, soft computing methods, in particular neural networks, modelling and simulation.