Optimization of active drag reduction for a slanted Ahmed body in a high-dimensional parameter space

Yiqing Li[1,2], Wenshi Cui[1,2], Qing Jia[1,2], Qiliang Li[1,2], Zhigang Yang[1,2,3] and Bernd R. Noack[4,5]†

[1] Shanghai Automotive Wind Tunnel Center, Tongji University, 4800 CaoAn Road, Jiading District, Shanghai 201804

[2] Shanghai Key Lab of Vehicle Aerodynamics and Vehicle Thermal Management Systems, Shanghai 201804, China

[3] Beijing Aeronautical Science & Technology Research Institute, Beijing, 102211, China

[4] LIMSI, CNRS, Université Paris-Saclay, Bât 507, rue du Belvédère, Campus Universitaire, F-91403 Orsay, France

[5]Hermann-Föttinger-Institut, Technische Universität Berlin, Müller-Breslau-Straße 8, D-10623 Berlin, Germany

(Received ?; revised ?; accepted ?. - To be entered by editorial office)

We numerically optimize drag of the 35° slanted Ahmed body with active flow control using Reynolds-Averaged Navier-Stokes simulations (RANS). The Reynolds number is $Re_H = 1.9 \times 10^5$ based on the height of the Ahmed body. The wake is controlled with seven local jet slot actuators at all trailing edges. Symmetric operation corresponds to five independent actuator groups at top, midle, bottom, top sides and bottom sides. Each slot actuator produces a uniform jet with the velocity and angle as free parameters. The drag is reduced by 17% optimizing all 10 actuation parameters as free input. The optimal actuation emulates boat tailing by inward-directed blowing with velocities which are comparable to the oncoming velocity. The results are well aligned with an experimental drag reduction of a square Ahmed body by boat tailing with using Coanda actuation (Barros *et al.* 2016). A key enabler for the optimization task is an algorithm which alternates between downhill simplex iteration for the exploitation of the best discovered local minima and Latin hypercube sampling for the exploration of possibly better minima. The combined algorithm is shown to be more efficient than the purely explorative and exploitive components. The optimization iteration is analyzed as control landscape with a proximity map. This map gives a two-dimensional impression of the drag topology in the 10-dimensional parameter space. We expect that the combination of RANS simulations, the proposed optimization algorithm and the control landscape analysis may guide many future active flow control plants.

† Email address for correspondence: zhigangyang@tongji.edu.cn, bernd.noack@limsi.fr

1. Introduction

In this study, we focus on active drag reduction behind a generic car model using Reynolds-averaged Navier-Stokes (RANS) simulations. Aerodynamic drag is a major contribution of traffic-related costs, from airborne to ground and marine traffic. A small drag reduction would have a dramatic economic effect considering that transportation accounts for approximately 20% of global energy consumption, Gad-el Hak (2006); Kim (2011). While the drag of airplanes and ships is largely caused by skin-friction, the resistance of cars and trucks is mainly caused by pressure or bluff-body drag. (Hucho 2002) defines bodies with a pressure drag exceeding the skin-friction contribution as bluff and as streamlined otherwise.

The pressure drag of cars and trucks originates from the excess pressure at the front scaling with the dynamic pressure and a low-pressure region at the rear side of lower but negative magnitude. The reduction of the pressure contribution from the front side often requires significant changes of the aerodynamic design. Few active control solutions for the front drag reduction have been suggested Minelli *et al.* (2016). In contrast, the contribution at the rearward side can significantly be changed with passive or active means. Drag reductions of 10% to 20% are common, (Pfeiffer & King 2014) have even achieved 25% drag reduction with active blowing. For a car at a speed of 120km/h, this would reduce consumption by about 1.8 liter per 100 km. The economic impact of drag reduction is significant for trucking fleets with a profit margin of only 2-3%. Two thirds of the operating costs are from fuel consumption. Hence, a 5% reduction of fuel costs from aerodynamic drag corresponds to over 100% increase of the profit margin.

The car and truck design is largely determined by practical and aesthetic considerations. In this study, we focus on drag reduction by passive or active means at the rearward side. Intriguingly most drag reductions of bluff body fall in the categories of Kirchhoff solution and aerodynamic boat tailing. The first strategy may be idealized by the Kirchhoff solution, i.e. potential flow around the car with infinitely thin shear-layers from the rearward separation lines, separating the oncoming flow and a dead-water region. The low-pressure region due to curved shear-layers is replaced by an elongated, ideally infinitely long wake with small, ideally vanishing curvature of the shear-layer. This wake elongation is achieved by reducing entrainment through the shear-layer, e.g. by phasor-control control mitigating vortex shedding (Pastoor *et al.* 2008) or by energetization of the shear-layer with high-frequency elongation (Barros *et al.* 2016). Wake disrupters also decrease drag, yet by energetizing the shear layer (Park *et al.* 2006) or delaying separation (Aider *et al.* 2010). Arguably, the drag of the Kirchhoff solution can be considered as achievable limit with small actuation energy.

The second strategy targets drag reduction by aerodynamic boat tailing. Geropp (1995); Geropp & Odenthal (2000) have pioneered this approach by Coanda blowing. Here, the shear-layer originating at the bluff body is vectored inward and gives thus rise to a more streamlined wake shape. Barros *et al.* (2016) has achieved 20% drag reduction of a square-back Ahmed body with high-frequency Coanda blowing in a high-Reynolds-number experiment. A similar drag reduction was achieved with steady blowing but at higher C_μ values.

This study focuses on drag reduction of the low-drag Ahmed body with rear slant angle of 35 degrees. This Ahmed body idealizes the shape of many cars. (Bideaux *et al.* 2011; Gilliéron & Kourta 2013) have achieved 20% drag reduction for this configuration in an experiment. High-frequency blowing was applied orthogonal to the upper corner of the slanted rear surface. Intriguingly, the maximum drag reduction was achieved in a narrow range of frequencies and actuation velocities and its effect rapidly deteriorated

for slightly changed parameters. In addition, the actuation is neither Coanda blowing nor an ideal candidate for shear-layer energization.

The literature on active drag reduction of the Ahmed body indicates that small changes of actuation can significantly change its effectiveness. Actuators have been applied with beneficial effects at all rearward edges (Barros *et al.* 2016), thus further complicating the optimization task. A systematic optimization of the actuation at all edges, including amplitudes and angles of blowing, is beyond reach of current experiments. In this study, a systematic RANS optimization is performed in a rich parametric space comprising the angles and amplitudes of steady blowing of five actuator groups: one the top, middle and bottom edge and two symmetric actuators at the corners of the slanted and vertical surface. High-frequency forcing is not considered, as the RANS tends to be overly dissipative to the actuation response.

The choice of optimization algorithm is critical for an acceptable computational load. A simple gradient method, like the downhill simplex method, may provide a drag minimum in many dozens of simulations, but there is no guarantee for a global minimum. On the other hand, exploratory strategies, like Monte Carlo methods or Latin hypercube sampling, will eventually come close to the global minimum, but tend to be prohibitively expensive. In this study, we combine exploitation and exploration in a single novel explorative gradient method strategy.

The manuscript is organized as follows. The employed optimization algorithms are introduced in § 2 and studied by performance comaprison in § 3. § 4 provides an example of fluidic pinball net drag power control optimization, which features 2-dimensional flow, in a three-dimensional actuation space based on DNS simulations. A simulation-based optimization of actuation for the low-drag Ahmed body characterized by 3-dimensional flow is given in § 5. § 5.1 and § 5.2 describes the configuration and RANS simulation. § 5.3, 5.4 and 5.5 describes the optimization for the one-, five- and ten-dimensional actuation space, respectively. The first one-dimensional space contains the streamwise velocity of the top jet actuator. The five-dimensional space comprises the streamwise velocities of the 5 symmetric actuations. And the ten-dimensional space includes in addition to these velocities also the orientation angle. Our results are summarized in § 6.

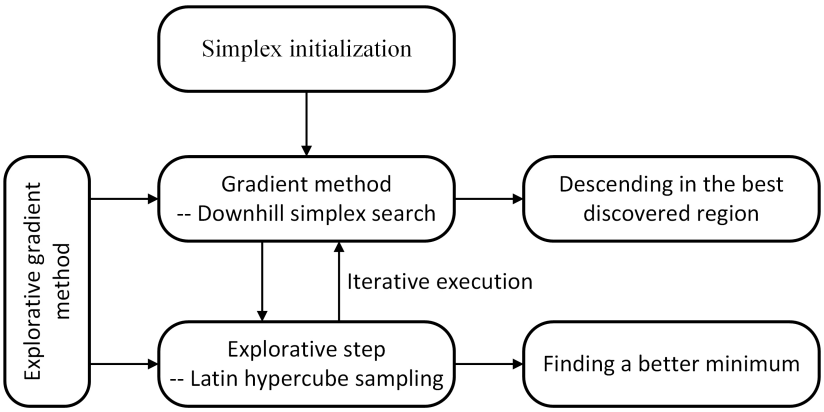


Figure 1: Sketch of the explorative gradient method. For details, see text.

2. Optimization algorithms

In this section, the employed optimization algorithms for the actuation parameters are described. Let $J(\boldsymbol{b})$ be the cost function—here the drag coefficient—depending on N actuation parameters $\boldsymbol{b} = (b_1, \ldots, b_N)$ in the domain Ω,

$$bmb = (b_1, \ldots, b_N) \in \Omega \subset \mathcal{R}^N.$$

Permissible values of each parameter define an interval, $b_i \in [b_{i,\min}, b_{i,\max}]$, $i = 1, \ldots, N$. In other words, optimization is performed in rectangular search space,

$$\Omega = [b_{1,\min}, b_{1,\max}] \times \ldots \times [b_{N,\min}, b_{N,\max}]\,. \tag{2.1}$$

The optimization goal is to find the global minimum of J in Ω,

$$\boldsymbol{b}^\star = \arg\min_{b\in\Omega} J(\boldsymbol{b}). \tag{2.2}$$

Several common optimization methods are investigated. Benchmark is the *downhill simplex method* (see, e.g. Press *et al.* 2007) as robust representative for gradient method (§ 2.4). This algorithm exploits gradient information from neighboring points to descent to a local minimum. Depending on the initial condition, this search may yield any local minimum. In the *random restart simplex (RRS) method*, the chance for finding a global minimum is increased by multiple runs with random initial conditions. The geometric coverage of the search space is the focus of *Latin Hypercube Sampling (LHS)* (see, again Press *et al.* 2007), which optimally explores the whole domain Ω independently of the cost values, i.e., ignores any gradient information. Evidently, LHS has the larger chance of getting close to the global minimum while the simplex algorithm is more efficient descending down a minimum, potentially a suboptimal one. *Monte Carlo (MC) sampling* (see, again Press *et al.* 2007), is a simpler and more common exploration strategy by taking random values for each argument, again, ignoring any cost value information. *Genetic algorithms* start with a Monte Carlo sampling in the first generation but then employ genetic operations to combine explorative and exploitive features in the following generations (see, e.g. Wahde 2008).

Sections 2.1–2.3 outline the non-gradient based explorative methods from the most explorative Latin hypercube sampling, to Monte Carlo sampling and the partially exploitive genetic algorithm. Sections 2.4 & 2.5 recapitulates the downhill simplex method and its random restart variant.

In section 2.6, we combine the advantages of the simplex method in exploiting a local

minimum and of the LHS in exploring the global one in a new *explorative gradient method* by an alternative execution (see figure 1). § 2.7 discusses auxiliary accelerators which are specific to the performed computational fluid dynamics optimization.

2.1. *Latin hypercube sampling–Deterministic exploration*

While our benchmark, the downhill simplex method exploits neighbourhood information to slide down to a local minimum, Latin hypercube sampling (LHS) (McKay *et al.* 1979) aims to explore the parameter space irrespective of the cost values. We employ a space-filling variant which effectively covers the whole permissible domain of parameters. This explorative strategy ('`maximin`' criterion in Mathematica) minimizes the maximum minimal distance between the points:

$$\{\boldsymbol{b}_m\}_{m=1}^{M} = \arg\max_{\boldsymbol{b}_m \in \Omega} \min_{\substack{i=1,\dots,M-1 \\ j=i+1,\dots,M}} \|\boldsymbol{b}_i - \boldsymbol{b}_j\| \,.$$

In other words, there is no other sampling of M parameters with a larger minimum distance. M can be any positive integral number.

For better comparison with the simplex algorithm, we employ an iterative variant. Note that once M sample points are created and they cannot be augmented anymore, for instance when learning of LHS was not satisfactory. We create a large number of LHS parameters $\boldsymbol{b}_j^\star$, $j = 1, \dots, M^\star$ for a dense coverage of the parameter space Ω at the beginning, typically $M^\star = 10^6$. As first sample $\boldsymbol{b}_1$, the center of the initial simplex is taken. The second parameter is taken from $\boldsymbol{b}_j^\star$, $j = 1, \dots, M^\star$ maximizing the distance to $\boldsymbol{b}_1$,

$$\boldsymbol{b}_2 = \mathrm{argmax}_{\boldsymbol{b}_j^\star, j=1,\dots,M^\star} \|\boldsymbol{b}_j^\star - \boldsymbol{b}_1\|.$$

The third parameter $\boldsymbol{b}_3$ is taken from the same set so that the minimal distance to $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ is maximized and so on. This procedure allows to recursively refine sample points and to start with an initial set of parameters.

2.2. *Monte Carlo Sampling—Stochastic exploration*

The employed space-filling variant of LHS requires the solution of an optimization problem guaranteeing a uniform geometric coverage of the domain. In high-dimensional domains, this coverage many not be achievable. A much easier and far more commonly used exploration strategy is Monte Carlo Sampling (MCS). Here, the mth sample $\boldsymbol{b}_m = [b_{1,m}, \dots, b_{N,m}]$ is given by

$$b_{i,m} = b_{i,\min} + \zeta_{i,m} \left(b_{i,\max} - b_{i,\min}\right), \tag{2.3}$$

$\zeta_{i,m} \in [0, 1]$ are random numbers with uniform probability distribution in the unit domain. The relative performance between LHS and MCS is a debated topic. We will wait for the results for an analytical problem in section 3.

2.3. *Genetic algorithm—Biologically inspired exploration and exploitation*

The genetic algorithm mimics natural selection process. We refer to Wahde (2008) as excellent reference. In the following, the method is briefly outlined to highlight the specific version and the chosen parameters.

In the current genetic algorithm, a chromosome corresponds to a vector of real parameters, a gene corresponds to a real number, and an allele corresponds to a real value (Wright 1991). Each population member is represented by a chromosome which is the parameter vector $\boldsymbol{x} = (x_1, x_2, \dots, x_m)[R$, and genes are the real parameters.

Genetic algorithm evolves one generation of I parameters, also called *individuals*, into a

new generation with the same number of parameters using biological inspired genetic operations. The first generation is based on Monte Carlo sampling, i.e., completely random genoms. The individuals J_i^1, $i = 1, \ldots, I$ are evaluated and sorted by their cost

$$J_1^1 \leqslant J_2^1 \leqslant \ldots \leqslant J_I^1.$$

The next generation is computed with three genetic operations: *replication*, copying individuals verbatim in the new generations; *mutation*, randomly changing parts of the genom; and *crossover*, randomly exchanging parts of the genom of two individuals. Replication has a memory preserving effect, mutation serves explorative purposes and crossover has the tendency to breed better inviduals. In an outer loop, the genetic oprations are randomly chosen with probabilities P_r, P_m and P_c for replication, mutation and crossover, respectively. In the inner loop, i.e., after the genetic operation is determined, individuals from the current generation are chosen. Higher performing individuals have higher probability to be chosen. Following the genetic algorithm matlab routine, this proability is proportional to the inverse square-root of its relative rank $p \propto 1/\sqrt{i}$. In addition, a deterministic operation, called *elitism*, shall guarantee that the top N_e performers of a generation are included in the next generation.

The genetic algorithm terminates according to a predetermined stop criterion, here a maximum number of generations L or corresponding number of evaluations $M = IL$. For reasons of comparison, we renumber the inviduals in the order of their evaluation, i.e. $m \in \{1, \ldots, I\}$ belongs to the first generation, $m \in \{I + 1, \ldots, 2I\}$ to the second generation, etc.

The chosen parameters are the default values of matlab, e.g. $P_r = 0.05$ $N_e = 3$), $P_c = 0.76$, $P_m = 0.19$. COMPLETE

GA description rewriting:

1 The algorithm begins by creating a random initial population. In this example, the initial population contains $I = 50$ individuals. the Initial range in the Population options is $[-3, 3]$.

2 The algorithm uses the individuals in the current generation to create the next population:

a) Scores each member of the current population by computing its fitness value.

b) Scales the raw fitness scores to convert them into a more usable range of values. These scaled values are called expectation values. An individual with rank r (smaller fitness, smaller rank) has scaled score proportional to $1/\sqrt{r}$.

c) Selects members, called parents, based on their expectation.

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function. The scaled fitness values are called the expectation values. An individual can be selected more than once as a parent, in which case it contributes its genes to more than one child. The default selection option, Stochastic uniform, lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled value. The algorithm moves along the line in steps of equal size. At each step, the algorithm allocates a parent from the section it lands on.

d) Some of the individuals in the current population that have lower fitness are chosen as elite ($P_r = 0.05$). These elite individuals are passed to the next population.

e) Produces children from the parents. Children are produced either by making random changes to a single parent—mutation ($P_m = 0.19$) —or by combining the vector entries of a pair of parents—crossover ($P_c = 0.76$).

Crossover children by selecting vector entries, or genes, from a pair of individuals in the current generation and combines them to form a child;

Mutation children by applying random changes to a single individual in the current

generation to create a child. The default mutation function randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. The mutation chooses a direction and step length that satisfies bounds and linear constraints.

f) Replaces the current population with the children to form the next generation.

3 The algorithm stops when one of the stopping criteria is met. Here, the stopping criteria is the maximum generation $L = 20$.

2.4. *Downhill simplex search–Robust gradient method*

The downhill simplex method by Nelder & Mead (1965) is a very simple, robust and widely used gradient method algorithm. This method does not require any gradient information and is well suited for expensive function evaluations, like the considered RANS simulation for the drag coefficients, and for experimental optimizations with inevitable noise. A price is a slow convergence for the minimization of smooth functions as compared to algorithms which can exploit gradient and curvature information.

We briefly outline the employed downhill simplex algorithm, as there are many variants. First, $N+1$ vertices $\boldsymbol{b}_m$, $m = 1, \ldots, N+1$ in Ω are initialized as detailed in the respective sections. Commonly, $\boldsymbol{b}_1$ is placed somewhere in the middle of the domain and the other vertices explore steps in all directions, $\boldsymbol{b}_m = \boldsymbol{b}_1 + h\boldsymbol{e}_{m-1}$, $m = 2, \ldots, N+1$. Here, $\boldsymbol{e}_i = (\delta_{i1}, \ldots, \delta_{iN})$ is a unit vector in i-th direction and h is a step size which is small compared to the domain. Evidently, all vertices must remain in the domain $\boldsymbol{b}_m \in \Omega$.

The goal of the simplex transformation iteration is to replace the worst argument $\boldsymbol{b}_h$ of the considered simplex by a new better one $\boldsymbol{b}_{N+2}$. This is archived in following steps:

1) Ordering: Without loss of generality, we assume that the vertices are sorted in terms of the cost values $J_m = J(\boldsymbol{b}_m)$: $J_1 \leqslant J_2 \leqslant \ldots \leqslant J_{N+1}$.

2) Centroid: In the second step, the centroid of the best side opposite to the worst vertex $\boldsymbol{b}_{N+1}$ is computed:

$$\boldsymbol{c} = \frac{1}{N} \sum_{m=1}^{N} \boldsymbol{b}_m.$$

3) Reflection: Reflect the worst simplex $\boldsymbol{b}_{N+1}$ at the best side,

$$\boldsymbol{b}_r = \boldsymbol{c} + (\boldsymbol{c} - \boldsymbol{b}_{N+1})$$

and compute the new cost $J_r = J(\boldsymbol{b}_r)$. Take $\boldsymbol{b}_r$ as new vertex, if $J_1 \leqslant J_r \leqslant J_N$. $\boldsymbol{b}_m$, $m = 1 \ldots, N$ and $\boldsymbol{b}_r$ define the new simplex for the next iteration. Renumber the indices to the $1 \ldots N+1$ range. Now, the cost is better than the second worst value J_N, but not as good as the best one J_1. Start a new iteration with step 1.

4) Expansion: If $J_r < J_1$, expand in this direction further by a factor 2,

$$\boldsymbol{b}_e = \boldsymbol{c} + 2\,(\boldsymbol{b}_{N+1} - \boldsymbol{c})\,.$$

Take the best vertex of $\boldsymbol{b}_r$ and $\boldsymbol{b}_e$ as $\boldsymbol{b}_{N+1}$ replacement and start a new iteration.

5) Single contraction: At this stage, $J_r \geqslant J_N$. Contract the worst vertex half-way towards centroid,

$$\boldsymbol{b}_c = \boldsymbol{c} + (1/2)\,(\boldsymbol{c} - \boldsymbol{b}_{N+1})\,.$$

Take $\boldsymbol{b}_c$ as new vertex ($\boldsymbol{b}_{N+1}$ replacement), if it is better than the worst one, i.e. $J_c \leqslant J_{N+1}$. In this case, start the next iteration.

6) Shrink / multiple contraction: At this stage, none of the above operations was successful. Shrink the whole simplex by a factor 1/2 towards the best vertex, i.e. replace all vertices by

$$\boldsymbol{b_m} \mapsto \boldsymbol{b}_1 + \frac{1}{2}\,(\boldsymbol{b}_m - \boldsymbol{b}_1)\,, m = 2, \ldots, N+1.$$

This shrinked simplex represents the one for the next iteration. It should be noted that this shrinking operation is the last resort as it is very expensive with N function evaluations. The rational behind this shrinking is that a smaller simplex may better follow local gradients.

2.5. *Random restart simplex method–Preparing for multiple minima*

The downhill simplex method of the previous section may be equipped with a random restart initialization. As random initial condition, we chose a Monte Carlo sample as main vertex of the simplex and explore all coordinate directions by a positive shift of 10% of the domain size. It is secured that all vertices are inside the domain Ω. These initial simplexes attributes the same probability to the whole search space. The chosen small edge length makes a locally smooth behaviour probable—in absence of any other information. The downhill search is stopped after a fixed number of evaluations. We chose 50 evaluations as safe upper bound for convergence. It should be noted that the number of simplex iterations is noticably smaller, as one iteration implies one to $N+2$ evaluations.

Evidently, the random restart algorithm may be improved by appreciated the many recommendations of literature, e.g., avoiding closeness to explored parameters. We trade these improvements in all optimization strategies for simplicity of the algorithms.

2.6. *Explorative gradient method–Combining exploration and gradient method*

In this section, we combine the advantages of the exploitive downhill simplex method and the explorative LHS in a single algorithm.

Step 0—Initialize. First, $\boldsymbol{b}_m$, $m = 1, \ldots, M+1$ are initialized for the downhill simplex algorithm.

Step 1—Downhill Simplex. Perform one simplex iteration (§2.4) with the best $M+1$ parameters discovered so far.

Step 2⋆—Deal with Degenerated Simplex. Compute the distance D between each vertex and their geometric center point. If D_{min} is smaller than 50% of D_{max}, the simplex is degenerated. Draw a circle with D_{max} about the geometric center Point. Obtain 1000 random samples in the circle. Find a point in the circle which can construct a triangle with the other N points with a largest area to repalce the point with highest cost J_{max}.

Step 3—LHS. Compute the cost J of a new LHS parameter $\boldsymbol{b}$. As described above, we take a parameter from a precomputed list which is furthest away from all hitherto employed parameters.

Step 3—Loop. Continue with Step 1 until a convergence criterion is met.

The algorithm is intuitively appealing.

If the LHS discovers a parameter with a cost J in the top $M+1$ values, this parameter is included in the new simplex and corresponding iteration may slide down in another better minimum. It should be noted that LHS exploration does not come with the toll of having to evaluate the cost at $N+1$ vertices and subsequent iterations. The downside of a single evaluation is that we miss potentially important gradient information pointing to an unexplored much better minimum. Relative to random-restart gradients searches requiring a many evaluations for a converging iteration, LHS exploration becomes increasingly better in rougher landscapes, i.e. more complex multi-modal behaviour.

2.7. *Computational accelerators*

The RANS based optimization may be acclerated by enablers which are spectific to the chosen flow control problem. The computation time for each RANS simulation is based on the choice of the initial condition, as it affects the convergence time for the steady

solution. The first simulation of an optimization starts with the unforced flow as initial condition. The next iterations exploit the existence of a deterministic mapping from actuation parameters $\boldsymbol{b}$ to the corresponding averaged velocity field $\boldsymbol{u}(\boldsymbol{x})$. The initial condition of the mth simulation is obtained with the 1-nearest-neighbour approach: The velocity field associated with the closest hitherto computed actuation vector is taken as initial condition for the RANS simulation. This simple choice of initial condition saves about 60% CPU time in reduced convergence time.

Another 30% reduction of the CPU time is achieved by avoiding RANS computations with very similar actuations. This is achieved by a quantization of the $\boldsymbol{b}$ vector: The actuation velocities are quantized with respect to integral m/s values. This corresponds to increments of $U_\infty/30$ with $U_\infty = 30$m/s. All actuation vectors are rounded with respect to this quantization. If the optimization algorithm yields a rounded actuation vector which has already been investigated, the drag is taken from the corresponding simulation and no new RANS simulation is performed. Similarly, the angles are discretized into integral degrees.

3. Comparative study

3.1. *Analytical function*

The analytical function, characterized by three local minimum seperately at [1,-1], [-1,1], [-1,-1] and a global minimum at [1,1], is employed for the study of the six algorithms.

$$\begin{aligned} J(b_1, b_2) = 1 &- e^{-2(b_1-1)^2-2(b_2-1)^2} \\ &- \frac{1}{2}e^{-2(b_1+1)^2-2(b_2-1)^2} \\ &- \frac{1}{3}e^{-2(b_1-1)^2-2(b_2+1)^2} \\ &- \frac{1}{4}e^{-2(b_1+1)^2-2(b_2+1)^2} \end{aligned} \tag{3.1}$$

3.2. *Parameters of the optimization methods*

1. LHS: number of samples $M^\star = 10^3$.

2. GA: polulation size $m = 50$; generation $N = 20$; elite individuals $P_e = 5\%$; crossover rate $P_c = 0.76$; mutuation rate $P_m = 0.19$.

3. Simplex: expansion rate 2; single contraction rate 1/2; shrink rate 1/2.

4. RRS: simple deviation $\gamma = +0.3$; evaluation limit for each simplex 50; restart limit 20;

5. EGM: expansion rate 2; single contraction rate 1/2; shrink rate 1/2; sample 1000 points in the concentric circles with radius D_{Max} to solve degeration, where the D_{Max} is the distance between the furthest vertex and centroid.

3.3. *Visualization of the tested individuals*

1. Proximity map:
Figure 2 illustrates the evolution of evaluations in the parameter space.

2. Exploration and exploit:
LHS shows a uniform coverage of the geometry space, compared with the overlaps in MC and the blank in GA.
Simplex converges step by step from the boundary of the domain to the global minimum according to the gradient of the cost function.

3. RRS, GA and EGM:
RRS repeats gradient search from a randomly simplex. It cost most evaluations to find all the minimums but make limited exploration.
GA makes random exploration in the space until finding the individuals near the global minimum. This leads to a intense distribution near the gobal minimum and lack of exploration of the other space.
Impressively, EGM makes a fully exploration of the space by LHS, and further exploits the different minimum. Actually, the continous exploration provides the possibility for jumping out of the two local minimums and make the game safe. Furthermore, in face of optimization problem without known expression, normal in engineering ptoject, the EGM can work on the optimization and at the same time provide the learning data for better knowledge of the whole parameter space.

3.4. *The learning curve*

1. The learning curve (figure 3)
explains the optimization performance of the former algorithms.

2. Compare the algorithms of exploit and that of exploration:
Without using the gradient information, the algorithms (LHS, MC, GA) in the left

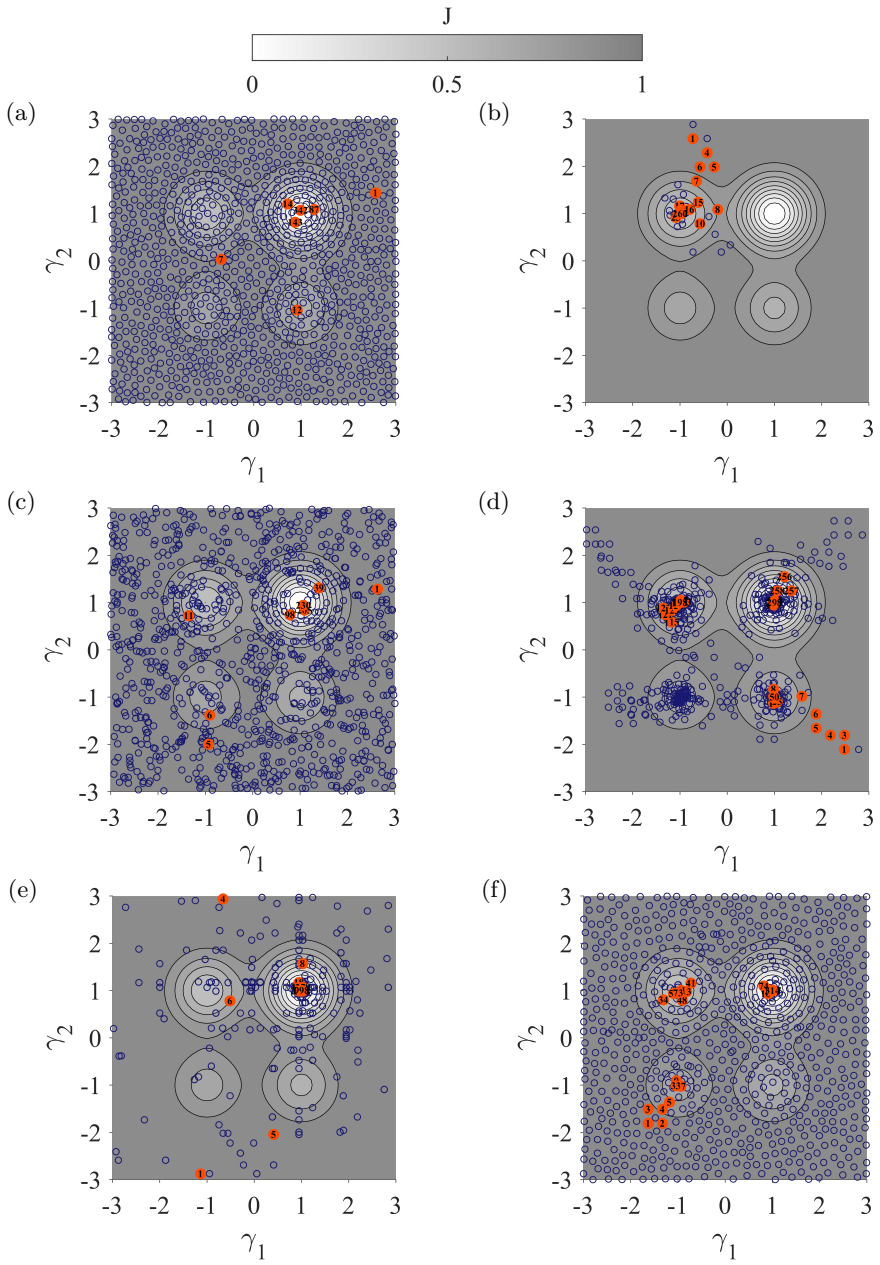


Figure 2: Tested individuals of *a*) LHS, *b*) Simplex, *c*) Monte Carlo, *d*) RRS, *e*) GA and *f*) EGM.

column show a smooth learning curve. With the decrease of the exploration (from LHS to MC and to GA), many horizonal stagnations appear and the decrease of the cost function slow down. This gives the reason for employing LHS as exploration tool. The graphs in the right column are characterized by steep slopes, which is the result of gradient search of the simplex steps. Simplex is chosen as the method to do exploit for its high efficiency in local optimization, which is obvious in the upper right graph.

3. Compare the algorithms including both exploit and exploration:

In RRS, 50% of the runs reach the minimum before 300 evaluations and experience long stagnation before convergence, which is more obvious in the other 50% runs with worse performance. This explains that Monte Carlo as exploration can hardly make fast contribution when the algorithm falls into local minimum. GA shows a learning process similar to MC. Convergence can be reached but the speed is random among all the runs. Only 10% of the runs reach the minimum before 300 evaluations. It is indicated that GA is not always efficient for the parameter optimization.

In EGM, no significant stagnations is observed in all the runs that successfully find the global minimum before 300 evaluations. The short stagnations are broken down continously. Compared with RRS, the exploration by LHS helps EGM to avoid the stagnation efficiently and thus cut the meaningless computation cost.

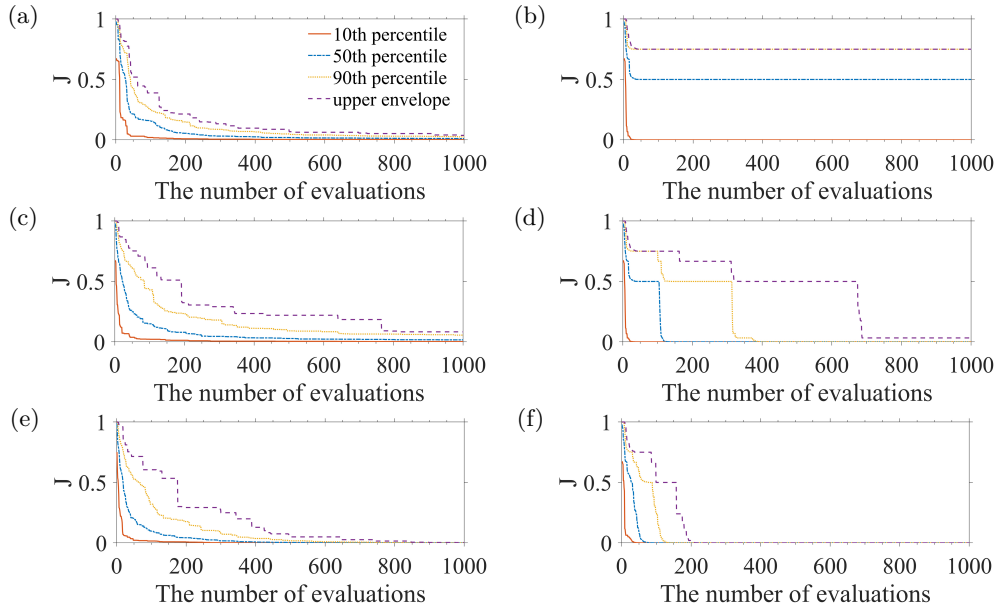


Figure 3: Learning curve of (*a*) LHS, (*b*) Simplex, (*c*) Monte Carlo, (*d*) RRS, (*e*) GA and (*f*) EGM in 100 runs. $10th$, $50th$, and $90th$ percentile indicates the J value below which 10, 40 and 90 percentage of runs at cunrrent evaluation falls.

3.5. *Discussion*

0. Table 1 concludes the average performance of the compared algoritms during the optimization and the fail rate (the percentage of the runs whose converged result is higher than 1% of the scaled cost funciton range $[0, 1]$ in 100 runs).

Describe from pure algorithms (LHS, MC, Simplex) and combined ones (GA, RRS, EGM): All the algorithms have relatively fair starting points at 20^{th} evaluation. Until 100^{th} evaluation, the purely explorative algorithms(LHS and MC) perform better than those employing exploitation. Both purely explorative and purely exploitative algorithms then slow down and have a fail rate higher than 50%, but for different reasons. The former is resulted by inefficeient exploration (especially MC) while the latter indicates falling into local minimum. Three algorithms employing both explotaition and exploitation show a significant advantage and almostly reach minimum for each run. EGM, which makes

Method	Evaluation				Failure rate
	20^{th}	100^{th}	500^{th}	1000^{th}	
LHS	0.5163	0.1456	0.0218	0.0129	0.55
MC	0.4810	0.1863	0.0441	0.0221	0.61
GA	0.4269	0.1441	0.0065	0.0002	0
S	0.4893	0.4675	0.4673	0.4673	0.73
RRS	0.4893	0.3208	0.0211	0.0003	0.01
EGM	0.5121	0.0621	0.0000	0.0000	0

Table 1: Average cost of different algorithms during eveluations in 100 runs.

a balance of exploration and exploitation, performs best from 100^{th} evaluation and converges fast to minimum before 500^{th} evaluation. RRS converges slowest, because of the inefficient MC step, and MC has a middle performance.

1. LHS achieves more than 70% and 95% reduction at 100^{th} and 500^{th} evaluation in average. The following 500 evaluations do not bring big difference. With reasonbale evaluations, the pure exploration can reach an area close to the global minimum but cannot converge well and lead to a failure rate of 55%.

2. MC performs a similar trend but has a slower speed of decrease in average cost function and a higer failure rate for the reduction of exploration.

3. By comparison, a better convergence appears in GA from the half of the evaluations (500^{th}) with the help of crossover and replication. And the failure rate is 0.

4. Simplex act to stuck in the local minimum after 100^{th} evaluation. However, a fast convergence is achieved within nearly former 10% evaluations. Given the three local minimums and one global minimum in the current example, the failure rate 73% indicates that simplex has a equal rate to fall into both global and local minimums.

5. In RRS, the random restarts added gradually drag the simplex out of current area every time the simplex progresses to some extent (50 evaluations). Only two independent 50-evaluation-long simplex optimizations can achieve better results than one 1000-evaluation-long simplex on average. And finally 4 restarts contibute to the global optimum at the rate of 99%. However, RRS still holds a slower convergence than GA because the random restarts only provide a possibiity instead of a direction for RRS to find the global minimum. Within limited evaluations, whether the global minimum area is detected depends on the number of minimums and evaluations; with enough evaluations, how depth the found global minimum depends on how close the random restart is to the true global minimum.

6. EGM implements the LHS and simplex step by turns. Once LHS step detects a better area, simplex starts the fast descent to the new valley. This leads to 88% reduction at only 10% cost of evaluations, and early convergence to global minumum at the half of evaluations. EGM adds more exploration into simplex to increase the possibiity of better area detection, and direct the optimization always to thoses areas by keeping the best individuals. The effective detection and instant utilization are crucial to the excellent performance.

7. To conclude

Obviously, combined algorithms (EGM, GA and RRS) perform better than single algorithms (LHS, MC and Simplex) in this race of optimization. In the single algorithms, LHS has the fastest decrease of cost function while simplex has the fastest convergence. EGM turns out to be the best combined algorithm by making a balance of exploration and exploit from LHS and simplex respectively.

4. Two-dimensional flow control—Net drag power optimization of fluidic pinball in a 3-dimensional actuation space

In this section, the explorative gradient method is applied to the fluidic pinball.

4.1. *Configuration*

Flow:
viscous incompressible uniform with speed U_∞; constant density ρ and kinematic viscosity ν; two-dimensional;

Object:
three equal circular cylinders with radius R; the centers of the cylinders form an equilateral triangle with sidelength $3R$.

Position:
symmetrical with respect to the flow; the leftmost triangle vertex points upstream; rightmost side is orthogonal to the oncoming flow; the transverse extend of the three cylinder configuration L = 5R.

Cartesian coordinate system:
x-axis — the direction of the flow; z-axis — the cylinder axes; y-axis — orthogonal to both; origin O — the mid-point of the rightmost bottom and top cylinder.

Denotation:
location $\boldsymbol{x} = (x, y, z) = x\boldsymbol{e}_x + y\boldsymbol{e}_y + z\boldsymbol{e}_z$; velocity $\boldsymbol{u} = (u, v, w) = u\boldsymbol{e}_x + v\boldsymbol{e}_y + w\boldsymbol{e}_z$; pressure p and the time t; non-dimensionalize quantities with cylinder diameter $D = 2R$, the velocity U_∞ and the fluid density ρ; the corresponding Reynolds number $Re_D = U_\infty D/\nu = 100$; with this non-dimensionalization, the cylinder axes are located at

$$\begin{array}{ll} x_F = -3/2\cos 30^\circ & y_F = 0 \\ x_B = 0 & y_B = 3/4 \\ x_T = 0 & y_T = +3/4 \end{array} \tag{4.1}$$

Here, the subscripts 'F ', 'B' and 'T 'refer to the front, bottom and top cylinder.

Actuation:
rotation of the cylinders with circumferential velocities $b_1 = U_F$,$b_2 = U_B$,$b_3 = U_T$; positive denotes the anti-clockwise direction, and vice versa.

Cost function:
Following Maceda *et al.* (2019), the minimization of the averaged parasitic drag power $\bar{J}_a$ with the penalization of the averaged actuation power $\bar{J}_b$,

$$\bar{J} = \bar{J}_a + \bar{J}_b \tag{4.2}$$

Direct numerical solution:
computation domain

$$\Omega = \{(x, y) : -6 \leqslant x \leqslant 20 \wedge |y| \leqslant 6 \wedge (x - x_i)^2 + (y - y_i)^2 \geqslant 1/4, i = 1, 2, 3)\} \tag{4.3}$$

unstructured grid—4225 triangles and 8633 vertices; intergate Navier-Stokes equation—an implicit Finite-Element Method; accuracy—third-order in space and time

4.2. *Results*

Parameter:
b_1—U_F; b_2—U_B; b_3—U_T; $b_i \in [-5, 5]$;

The exploraitve gragient method starts around:
1 the uncontrolled individual ($[0, 0, 0]$) in the middle of the space;
2 the individual diverts positively in the first parameter with a quarter of the length of

m	b_1	b_2	b_3	J
1	0	0	0	1.4611
2	2.5	0	0	3.0993
3	2.5	-2.5	0	4.9617
4	2.5	-2.5	2.5	5.1866

Table 2: Initial simplex ($m = 1, 2, 3$) for the five-dimensional downhill simplex optimization. b_i are the normalized actuation velocities and J corresponds to the drag coefficient.

the domain—$[2.5, 0, 0]$;
3 the individual diverts negatively in the second parameter with a quarter of the length of the domain—$[2.5, -2.5, 0]$ based on individual 2;
4 the individual add a symemtry actuation of the top actuation to the bottom—$[2.5, -2.5, 2.5]$ based on individual 3.
Table 2 shows the values of the individuals and corresponding cost. All vertices have a larger net drag power than the former benchmark:
1-2 The positive actuation of front actuation does harm to the optimization;
2-3 The negative actuation of bottom actuation does harm to the optimization;
3-4 The positive actuation of top actuation does harm to the optimization;
infer—$[\leqslant 0, \geqslant 0, \leqslant 0]$ does good to optimization.

Optimization process:
A global and single minimum is observed from 4b. Explorative gradient method starts from area close to the global minimum and then slips into the valley step by step, accompanied by uniform exploration of the whole space.

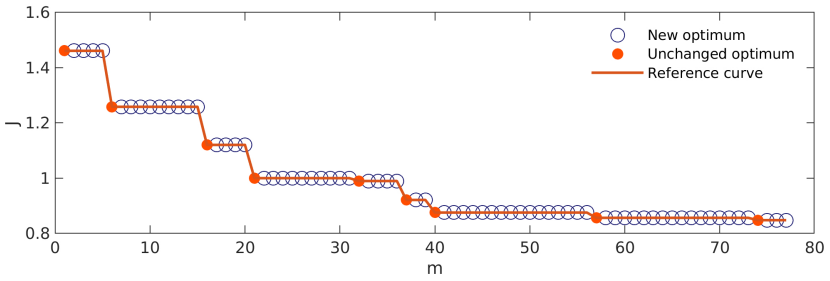
Best control:
$m = 74$, $b_1 = -0.0583$, $b_2 = 1.6839$, $b_3 = -1.7154$ and $J = 0.8481$

Individual denotation:
principle—choose the individual close to the vertical and horizon line the line $\gamma_1 = 2$ and the line $\gamma_2 = 0$.
A—uncontrolled; B—optimal; C—Coanda effect; D—the last step make all the simplex (A-6-D) into the first legend; E—the 2rd individual; F—at the middle between A and J; G,H,I,J—at the boundary;

γ:
H-E-D-A-F-J γ_2 is positive related to lift—negatively to the front cylinder b_F;
the individuals with $\gamma_2 = 0$ have a symemtry actuation of top and bottom cylinders.
G-C-B-A-I γ_1 is positive related to drag—the top and bottom cylinders b_3 and b_2.
The global minimum is near $\gamma_1 = 0$ and $\gamma_2 = 0$.

Net drag reduction and drag reduction:
This control law achieves a 42% net drag power reduction by simulating boat-tailing and leads to a smoother wake (figure 5b). This is comparable to the optimization result achieved by genetic programming (Maceda *et al.* 2019). Figure 6 shows the drag of top and bottom cylinders are significantly reduced, corresponding to 92% drag reduction (from 1.4597 to 0.1148).

(a)



(b)

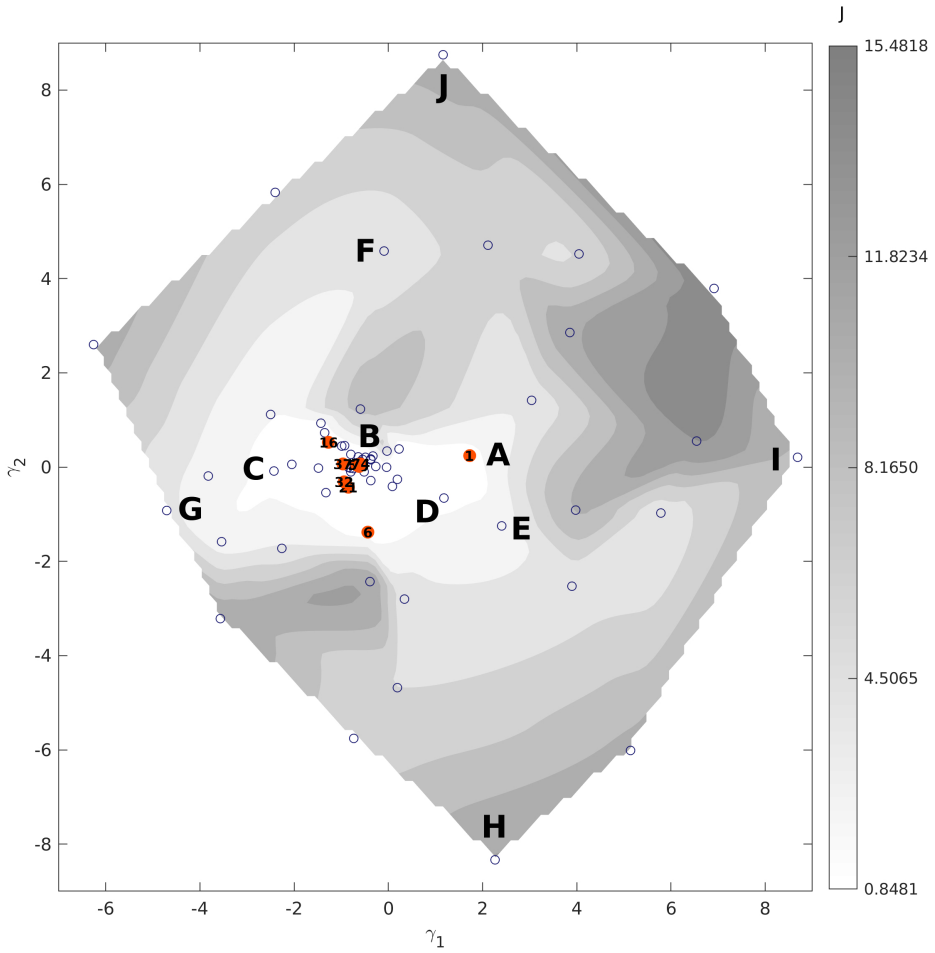


Figure 4: Optimization of the cylinder actuation during explorative gradient method. The actuation parameters and cost is visualized like in figure 3 and 2. m counts the DNS sumulation calls for net drag power computation.

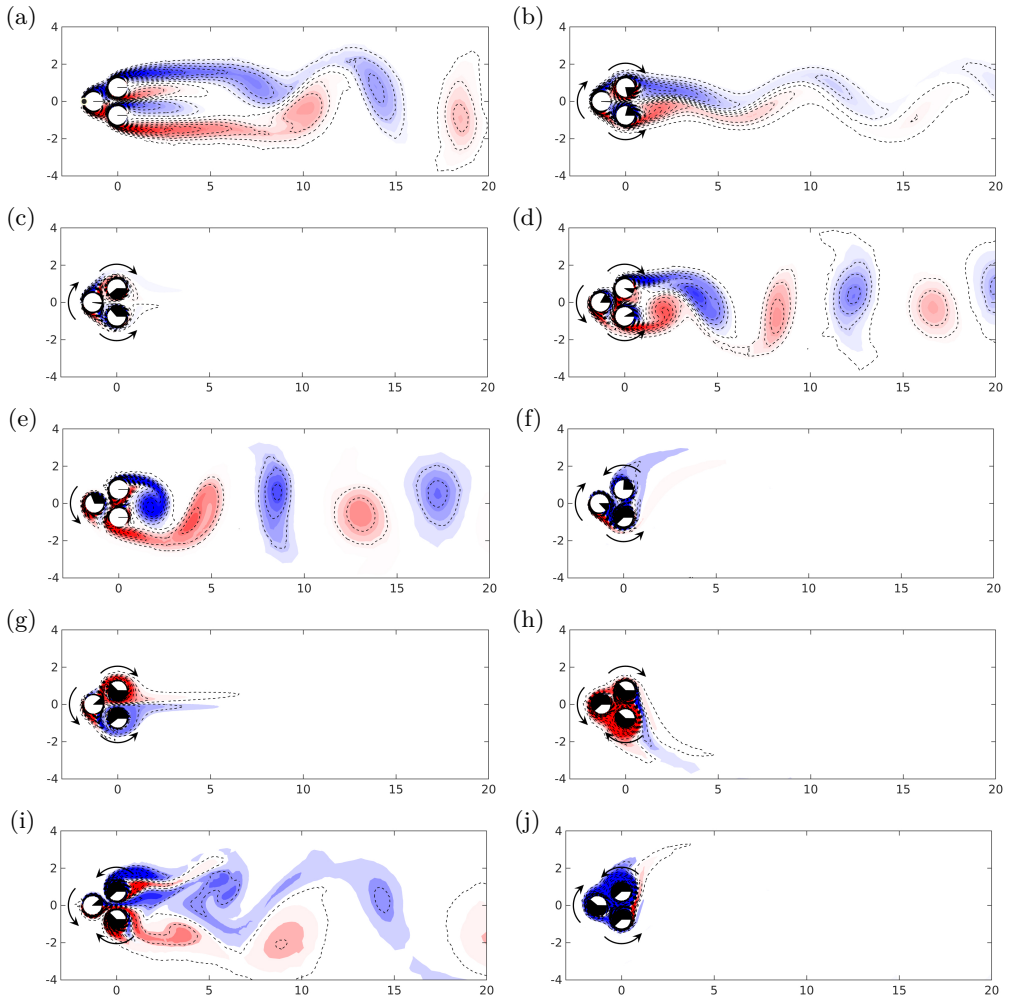


Figure 5: Flow of different control schemes in figure 4b.

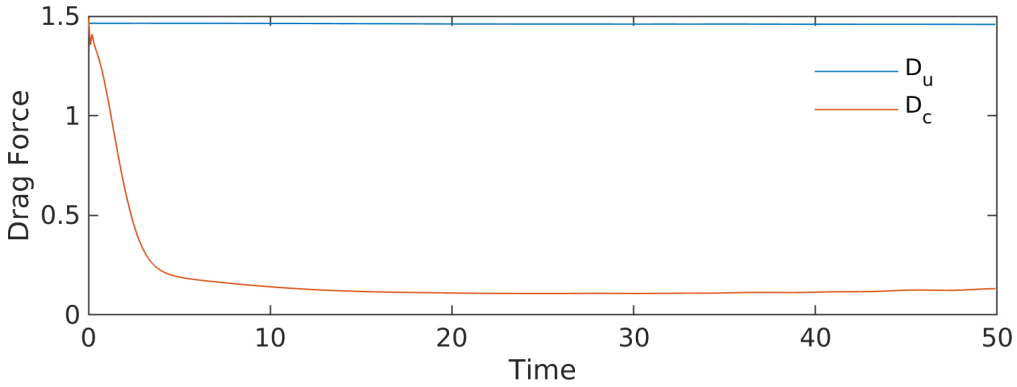


Figure 6: Evolution of drag on each cylinder over time. D_u denotes drag of unforced flow and D_c best control

5. Three-dimensional flow control—Drag optimization of an Ahmed body in a 10-dimensional actuation space

Starting point of the computational fluid dynamics plant is an experimental study of a low-drag 35° Ahmed body (Li *et al.* 2018). The investigated Ahmed body configuration (§ 5.1) has the same physical dimensions. The effect of actuation is assessed with a Reynolds-Averaged Navier-Stokes (RANS) simulation (§ 5.2).

5.1. *Configuration*

Point of departure is an experimentally investigated 1:3-scaled Ahmed body characterized by a slanted edge angle of $\alpha = 35°$ with length L, width W and height H of 348 mm, 130 *mm* and 96 mm, respectively. The front edges are rounded with a radius of $0.344\,H$. The model is placed on four cylindrical supports with a diameter equal to 10 mm and the ground clearance is $0.177\,H$. The origin of the Cartesian coordinate system (x, y, z), is located in the symmetry plane on the lower edge of the model's vertical base (see figure 7). Here, x, y and z denote the streamwise, spanwise and wall-normal coordinate, respectively. The velocity components in the x, y and z directions are denoted by u, v and w, respectively. The free-stream velocity is chosen to be $U_\infty = 30\,\mathrm{m/s}$.

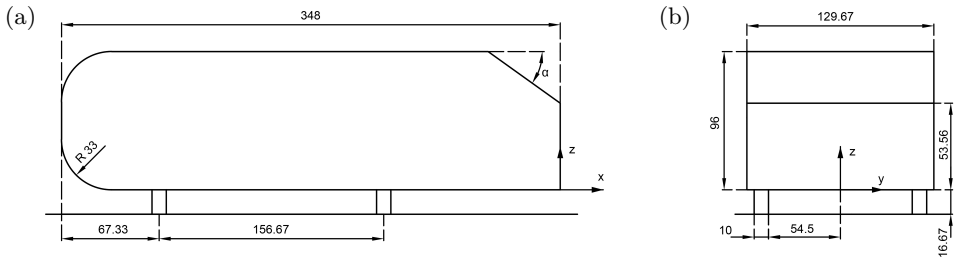


Figure 7: Dimensions of the investigated 1:3-scaled Ahmed body. *a*) Side view. *b*) Back view. The length unit is mm and the angle is specified in degrees.

Five groups of steadily blowing slot actuators (figure 8) are deployed on all edges of the rear window and the vertical base. All slot widths are 2 mm. The horizontal actuators at the top, middle and bottom side have lengths of 109 mm. The upper and lower sidewise actuators on the upper and vertical rear window have a length of 71 *mm* and 48 mm, respectively. The actuation velocities $U_1, \ldots, U_5$ are independent parameters. U_1 refers to the upper edge of the rear window, U_3 to the middle edge and U_5 to the lower edge of the vertical base. U_2 and U_4 correspond to the velocities at the right and left sides of the upper and lower window, respectively.

Following the experiment by Zhang *et al.* (2018), all blowing angles can be varied as indicated in figure 8. This study aims at minimizing drag as represented by the drag coefficient, $J = c_D$, by varying the actuation control parameters. The actuation velocity amplitudes U_i, $i = 1, \ldots, 5$ are capped by twice of the single optimum value as discussed in § 5.3. The actuation angles θ_i, $i = 1, \ldots, 5$ are fixed to 0°, i.e. streamwise direction, in a 5-dimensional optimization. The actuation angles are later added into the input parameters in 10-dimensional optimization, with variable angles $\theta_1 \in [-35°, 90°]$, $\theta_2, \theta_3, \theta_4, \theta_5 \in [-90°, 90°]$.

5.2. *RANS simulation*

A numerical wind tunnel (figure 9) is constructed using the commercial grid generation software Ansys ICEM CFD. The rectangular computational domain is bounded

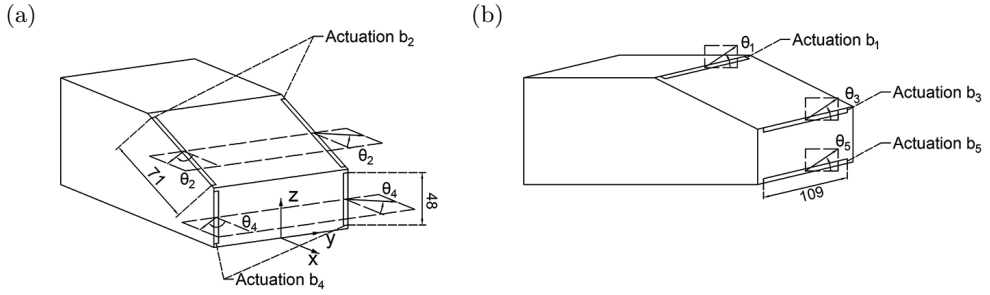


Figure 8: Deployment and blowing direction of actuators on the rear window and the vertical base. The angles θ_1, θ_2, θ_3, θ_4 and θ_5 are all defined to be positive when pointing outward or upward.

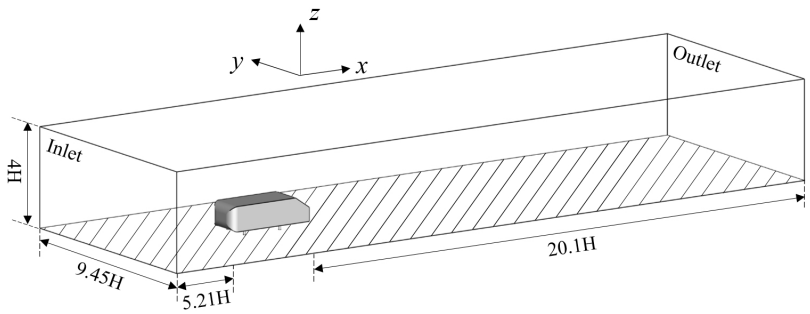


Figure 9: Computational domain of the RANS simulation.

Mesh grid points	2.5M	5M	10M
Drag Coefficient	0.294	0.313	0.318

Table 3: Drag coefficient based on different mesh resolutions.

by $X_1 \leqslant x \leqslant X_2$, $0 \leqslant z \leqslant H_T$, $|y| \leqslant W_T/2$. Here, $X_1 = -5.21\ H, X_2 = 20.17\ H$, $H_T = 4H$, and $W_T = 9.45H$. A coarse, medium and fine mesh using unstructured hexahedral computational grid are employed in order to evaluate the performance of RANS method for the current problem with different mesh resolutions. The statistics in Table 3 show that using a finer mesh can be expected to have negligible improvement on the accuracy of the drag coefficient. Hence, the more economical medium mesh 10 is used. This mesh consists of 5 million elements and features dimensionless wall values $\Delta x^+ = 20, \Delta y^+ = 3, \Delta z^+ = 30$. In addition to resolving the boundary layer, the shear layers and the near-wake region, the mesh near the actuation slots is also refined.

Reynolds-Averaged Navier-Stokes (RANS) simulations using the realizable $k-\epsilon$ model with the constant parameters

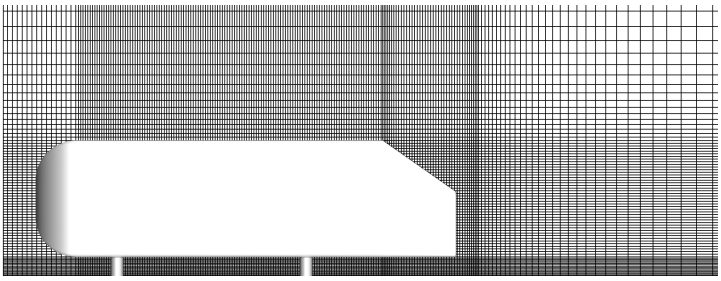


Figure 10: Side view of the computational grid used for RANS.

$$\left.\begin{aligned} \sigma_k = 1.0, \sigma_\varepsilon = 1.2, C_2 = 1.9 \\ C_1 = \max\left(0.43, \frac{\eta}{\eta+5}\right) \\ \eta = \left(2\sum_{i,j=1}^{3} E_{ij}E_{ji}\right)^{1/2} \frac{k}{\varepsilon} \\ E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \end{aligned}\right\}$$

$\sigma_k = 1.0$, $\sigma_\epsilon = 1.2$, $C_2 = 1.9$ are performed employing the commercial flow solver Ansys Fluent. The spatial discretization is based on a second-order upwind scheme in the form of SIMPLE scheme based on a pressure-velocity coupling method. RANS simulation have been frequently and successfully been used to assess actuation effects from steady blowing (Ben-Hamou *et al.* 2007; Dejoan *et al.* 2005; Muralidharan *et al.* 2013; Viken *et al.* 2003). We deem RANS simulations to provide reasonable qualitative and approximately quantitative indications for actuator optimization and plan an experimental validation in the future. Partially Averaged Navier-Stokes (PANS) simulations (Han *et al.* 2013) and Large Eddy Simulation (LES) (Krajnović 2009; Brunn & Nitsche 2006) are trusted higher-fidelity simulations for drag reduction with active flow control but are computationally orders of magnitudes more demanding.

5.3. *Formulation of optimization problem based on streamwise blowing at the top edge*

The formulation and constraints of the optimization problem is motivated by the drag reduction results from the top actuator blowing in streamwise direction. Figure 11 shows the drag coefficient in dependency of streamwise blowing velocity, all other actuators being off. The blowing velocity varies in increments of 5 m/s from 0 m/s to 60 m/s, i.e. reaches twice the oncoming velocity.

The drag coefficient is quickly reduced by modest blowing, has a shallow minimum near the actuation velocity $U_{b1} = 25$ m/s before quickly increasing with more intense blowing. This optimal value corresponds to 5/6 of the oncoming velocity. The best drag reduction is 5% with respect to the unforced flow $C_d = 0.3134$. Near $U_1 = 45$ m/s, the drag rapidly rises beyond the unforced value.

This behaviour motivates the choice of actuation parameters. The first five actuation parameters are normalized jet velocities $b_i = U_i/U_{b1}$, $i = 1, \ldots, 5$ introduced in § 5.1. Thus, $b_1 = 1$ corresponds to minimal drag with a single streamwise-oriented top actuator. All b_i are capped by 2: $b_i \in [0, 2]$, $i = 1, \ldots, 5$. At $b_1 = 1.8$, point 'C' in figure 11, actuation yields already drag increase. The first vertex of the amoeba of the downhill

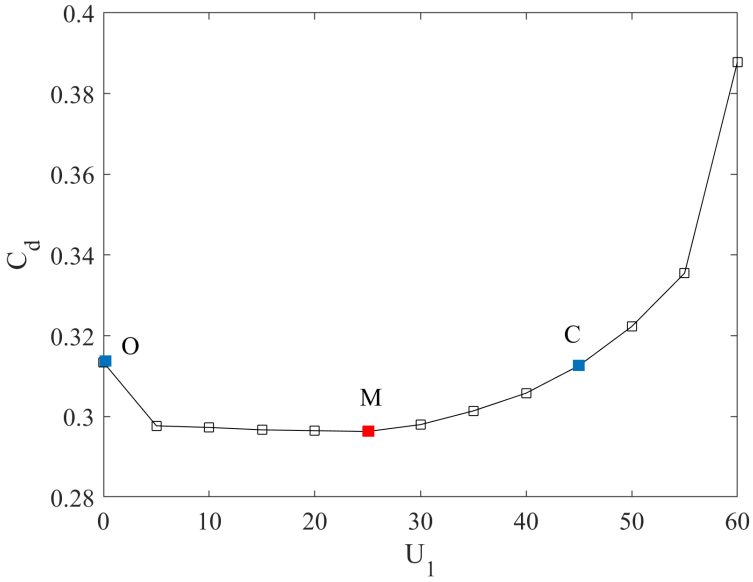


Figure 11: Drag coefficient as a function of the blowing velocity U_1 of the streamwise-oriented top actuator. Here, 'O' marks the drag without forcing, 'M' the best actuation, and 'C' the smallest actuation with worse drag than for unforced flow.

simplex search is put at $b_1 = b_2 = b_3 = b_4 = b_5 = 1.8$. From figure 11, we expect a drag minimum at lower values, hence the next five vertices test the value 1.6, e.g. $(b_1, b_2, \ldots, b_5) = (1.8 - 0.2\delta_{1,m-1}, 1.8 - 0.2\delta_{2,m-1}, \ldots, 1.8 - 0.2\delta_{5,m-1})$ for $m = 2, \ldots, 6$. If excluded out-of-domain parameters $b_i \geqslant 2$ would yield a maximum drag reduction, the downhill simplex algorithm can be expected to move to the outer border of the actuation domain, thus indicating too restrictive constraints. We refrain from starting already with a much larger actuation domain, as the exploration with LHS and the proposed explorative gradient method will consistently test too many large velocities. An increase of the upper velocity bound by a factor 2, for instance, implies that only 2^{-5} or around 3% of uniformly distributed sampling points are in the original domain and 97% of the samples are outside.

The next five parameters characterize the deflection of the actuator velocity with respect to the streamwise direction (see § 5.1), $b_{i+5} = \theta_i/(\pi/2)$, $i = 1 \ldots, 5$, and are normalized with 90°. Now all b_i, $i = 1, \ldots, 10$ span an interval of width 2, except for the more limited deflection b_6 of the top actuator. Summarizing, the domain for the most general actuation reads

$$\Omega := \left\{ \boldsymbol{b} \in \mathcal{R}^{10} : \begin{array}{ll} b_i \in [0, 2] & \text{for } i = 1, \ldots, 5 \\ b_i \in [-35/90, 1] & \text{for } i = 6 \\ b_i \in [-1, 1] & \text{for } i = 7, \ldots, 10 \end{array} \right\}. \tag{5.1}$$

The choice of $\boldsymbol{b}$ as symbol shall remind about the control B-matrix in control theory and is consistent with many earlier publications of the authors, e.g. the review article by Brunton & Noack (2015).

m	b_1	b_2	b_3	b_4	b_5	J
1	1.8	1.8	1.8	1.8	1.8	0.4153
2	1.6	1.8	1.8	1.8	1.8	0.4048
3	1.8	1.6	1.8	1.8	1.8	0.4109
4	1.8	1.8	1.6	1.8	1.8	0.3996
5	1.8	1.8	1.8	1.6	1.8	0.4075
6	1.8	1.8	1.8	1.8	1.6	0.4040

Table 4: Initial simplex ($m = 1, \dots, 6$) for the five-dimensional downhill simplex optimization. b_i are the normalized actuation velocities and J corresponds to the drag coefficient.

5.4. *Optimization of the streamwise trailing edge actuation*

The drag of the Ahmed body is optimized with streamwise blowing from the five slot actuators. We apply a simplex downhill search, Latin hypercube sampling and the explorative gradient method of § 5.4.1, § 5.4.2 and § 5.4.3, respectively.

5.4.1. *Downhill simplex algorithm*

Following § 5.3, the downhill simplex algorithm is centered around $b_i = 1.8$, $i = 1, \dots, 5$ as first vertex and explores a lower actuation $b_{m-1} = 1.6$ in all directions for vertices $m = 2, \dots, 6$. Table 4 shows the values of the individuals and corresponding cost. All vertices have a larger drag than for the unforced benchmark $C_d = 0.3134$. And all vertices with $b_i = 1.6$ are associated with a smaller drag indicating a downhill slide to small actuation values consistent with the expectations from § 5.3.

Figure 12 (top) shows the evolution of the downhill simplex algorithm with 200 RANS simulations. Like in § 3, solid red circles mark newly found optima while open blue circles record the best actuation so far. The drag quickly descends after staying shortly on a plateau at $m \approx 20$. From there on, the descend becomes gradual. The optimal drag $J = 0.2908$ is reached with the 148th RANS simulation and corresponds to 7% drag reduction. The optimal actuation reads $b_1 = 0.7264$, $b_2 = 0.5508$, $b_3 = 0.1533$, $b_4 = 0.6746$, $b_5 = 0.7716$. While the middle horizontal jet has small amplitude, the other actuation velocities on the four edges of the Ahmed body are 55% to 77% of the optimal value achieved with single actuator.

Figure 12 (bottom) illustrates the downhill search in a control landscape $J(\gamma_1, \gamma_2)$ described in § 2. Here (γ_1, γ_2) feature vectors defining a proximity map of the five-dimensional actuation parameters $(b_1, \dots, b_5)$. This landscape indicates a complex topology of the five-dimensional actuation space by many local maxima and minima in the feature plane. This complexity may explain why most simplex steps did not yield a better cost. The feature coordinate γ_1, γ_2 arise from a kinematic optimization process and have no inherent meaning. The simplex algorithm is seen to crawl from right $\boldsymbol{\gamma} \approx (2, 0)$ to the assumingly global minimum at $\boldsymbol{\gamma} \approx (-0.6, 0)$ through an elongated curved valley.

5.4.2. *Latin hypercube sampling*

Figure 13 (top) shows the slow learning process associated with Latin hypercube sampling (LHS) starting with the simplex reference point $b_1 = \ldots = b_5 = 1.8$. Apparently the optimization is ineffective. Only 3 new optima are successively obtained in 200 RANS simulations. The remaining simulations yield worse drags than the best discovered before. At the 70th RANS simulation, the best drag coefficient of $C_d = 0.2928$, with $b_1 = 0.0994$, $b_2 = 0.9587$, $b_3 = 0.1276$, $b_4 = 0.0289$ and $b_5 = 1.0393$. corresponds to 5% reduction like the one-dimensional top actuator $b_1 = 1$, $b_2 = b_3 = b_4 = b_5 = 0$. Intriguingly, only

the upper side and bottom actuator have b_i amplitudes near unity while remaining parameters are less than 13% of the one-dimensional optimum. These results show that near optimal drag reductions can be achieved with quite different actuations. Moreover, individual actuation effects are far from additive. Otherwise, the almost complimentary LHS optimum for actuators 2–5 and the one-dimensional optimum of § 5.3 should yield 10% reduction with $b_1 \approx 1$, $b_2 \approx 1$, $b_3 \approx 0.13$, $b_4 \approx 1$ and $b_5 \approx 1$.

Figure 13 (bottom) shows the LHS in the control landscape. In the first iteration, LHS jumps to the opposite site of domain and finds better drag. The next successive two improvements are in a good terrain but the optimum at $m = 70$ is still far from the assumingly global minimum at $\boldsymbol{\gamma} = (-0.6, 0)$ (see figure 12). The exploratory steps uniformly cover the whole range of feature vectors.

5.4.3. *Explorative gradient method*

From the figure 14, the explorative gradient method is seen to converge much faster than the downhill simplex algorithm. The best actuation is found at the 65th RANS simulation yielding the same drag coefficient $C_d = 0.2908$ of the downhill simplex algorithm with only slightly different actuation parameters $b_1 = 0.6647$, $b_2 = 0.4929$, $b_3 = 0.1794$, $b_4 = 0.7467$ and $b_5 = 0.7101$.

The fast convergence of the explorative gradient method is initially surprising since up to 50% of the steps are for explorative purposes, i.e. shall identify distant minima. However, the control landscape in figure 13 reveals how the explorative LHS steps help the algorithm to prevent the long and painful march through the long and curved valley. The proposed new algorithm operates like a visionary mountain climber, who performs not only local uphill steps but sends drones to the remotest location to find better mountains and terrains.

5.5. *Optimization of the directed trailing edge actuation*

In this section, the actuation space is enlarged by the jet directions of all slot actuators. The jets may now be directed inwards or outwards as discussed in § 5.1. The actuation optimization for drag reduction is performed with explorative gradient method (§ 5.5.1). The unforced and three actuated Ahmed body wakes are investigated in § 5.5.2.

5.5.1. *Explorative gradient method*

We employ the explorative gradient method as best performing method of § 5.4 for the 10-dimensional actuation optimization problem. The search is accelerated by starting with a simplex centered around the optimal actuation of the five-dimensional problem. The first vertex of table 5 contains this optimal solution. The cost is 4‰ lower than the previous section as the RANS integration for the first flow is not fully converged. The next five vertices represent isolated actuations at the optimal value but directed 45° outwards for the side edges and upwards for the middle horizontal actuator. The corresponding drag values are larger. The next five vertices deflect the jets in opposite direction by 45° or the maximum 35° of the top actuator, giving rise smaller drag than the previous deflection. The drag of middle horizontal actuator remains close to the unforced benchmark because the jet velocity is small. Figure 15 (top) illustrates the convergence of the explorative gradient method. After 289 RANS simulations, a drag coefficient of 0.2586 is achieved corresponding to a 17% drag reduction. The optimal actuation values read $b_1 = 0.8611$, $b_2 = 0.9856$, $b_3 = 0.0726$, $b_4 = 1.0089$, $b_5 = 0.8981$, $b_6 = -0.3000$ corresponding to $\theta_1 = -27°$, $b_7 = -0.4666$ ($\theta_2 = -42°$), $b_8 = 0.7444$ ($\theta_3 = 67°$), $b_9 = -0.4888$ ($\theta_3 = -44°$), and $b_{10} = 0.2444$ ($\theta_3 = -22°$). All outer actuators have velocity amplitudes near unity and are directed inwards, i.e. emulate Coanda blowing.

m	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_10	J
1	0.6647	0.4929	0.1794	0.7467	0.7101	0	0	0	0	0	0.2895
2	0.6647	0	0	0	0	1/2	0	0	0	0	0.3268
3	0	0.4929	0	0	0	0	1/2	0	0	0	0.3226
4	0	0	0.1794	0	0	0	0	1/2	0	0	0.3168
5	0	0	0	0.7467	0	0	0	0	1/2	0	0.3476
6	0	0	0	0	0.7101	0	0	0	0	1/2	0.3060
7	0.6647	0	0	0	0	-35/90	0	0	0	0	0.3091
8	0	0.4929	0	0	0	0	-1/2	0	0	0	0.3085
9	0	0	0.1794	0	0	0	0	-1/2	0	0	0.3187
10	0	0	0	0.7467	0	0	0	0	-1/2	0	0.3001
11	0	0	0	0	0.7101	0	0	0	0	-1/2	0.3354

Table 5: Initial individuals in the optimization of the directed trailing edge actuation. b_i, $i = 1, 2, 3, 4, 5$ represent the actuation amplitudes U_i of the ith actuator. b_i, $i = 6, 7, 8, 9, 10$ denotes the actuation angle θ_i of the $(i - 5)$th actuator. J is the drag coefficient.

The third middle actuator blows upward with low amplitude. The strong inward blowing seems to be related to the additional 10% drag reduction as compared to the 7% of streamwise actuation.

Figure 15 (bottom) shows the search process in a proximity map. It should be noted that this control landscape is based on data in a ten-dimensional actuation space and hence different from the 5-dimensional space in § 5.4. The algorithm quickly descends in the valley while many exploration steps probe suboptimal terrain. One reason for this quick landing in good terrain is the chosen initial simplex around the optimized actuation in the five-dimensional subspace.

The topology of the control landscape of figure 15 is investiged with discrete steepst descent lines connecting neighboring data points in figure 16. For each investigated actuation vector, the nearest five neighbours are considered. If all neighbours have higher drag, the vector is considered as a local minimum and marked by a red point. Otherwise, a gray dashed arrow is plotted to the best of these neighbour. This steepest descent is continued until a local minimum is reached. The corresponding path is called (discrete) steepest descent line. Line segments shared by at least 10 of these pathes may be considered as important valleys towards the minimum and highlighted as black solid arrow. The visiting times of each individual are marked by colorcoding. The global minimum of all data points is visited most, 54 steepest descend lines end here. The resulting pathways of 'mountain trails' to 'expressways' may give an indication of the directions to be expected from local search algorithms. Moreover, crossing steepest descent lines indicate that the two-dimensional proximity maps oversimplifies a higher-dimesional landscape structure. Intriguingly, the steepest descent line become more aligned to each other in the valleys leading to global data minimum, i.e. the most relevant regions for optimization.

5.5.2. *Discussion of streamwise and directed jet actuators*

In the following, the physical structures associated with the optimized one-, five- and ten-dimensional actuation are discussed. Evidently, more degrees of freedom are associated with more opportunities for drag reduction. Expectedly, the drag reduces by 5% to 7% to 17% as the dimension of the actuation parameters increase from 1 to 5 to 10, respectively. Intriguingly, the increase of drag reduction from the optimized top actuator to the best 5 streamwise actuators is only 2%. For the square-back Ahmed body, Barros

Case	Drag reduction	Actuation parameters				
		Top	Upper side	Middle	Lower side	Bottom
A)	0%	—	—	—	—	—
B)	5%	$b_1 = 1$	$b_2 = 0$	$b_3 = 0$	$b_4 = 0$	$b_5 = 0$
C)	7%	$b_1 = 0.6647$	$b_2 = 0.4929$	$b_3 = 0.1794$	$b_4 = 0.7467$	$b_5 = 0.7101$
D)	17%	$b_1 = 0.8611$	$b_2 = 0.9856$	$b_3 = 0.0726$	$b_4 = 1.0089$	$b_5 = 0.8981$
		$\theta_1 = -27°$	$\theta_2 = -42°$	$\theta_3 = 67°$	$\theta_3 = -44°$	$\theta_3 = -22°$

Table 6: Investigated optimized actuations in comparison to the unforced benchmark. The table shows the achieved drag reduction and corresponding actuation parameters for A) the unforced benchmark, and for the optimized B) top streamwise actuator, C) all streamwise actuators, D) all deflected actuators.

(2015) experimentally observed that the individual drag reductions from the streamwise blowing actuators on the four trailing edges roughly add up to the total drag reduction of 10% with all actuators on. This additivity of actuation effects is not corroborated for the slanted low-drag Ahmed body. Intriguingly, the inward deflection of the jet-slot actuators substantially decreases drag by 10%. This additional drag reduction of 10% has also been observed for the square-back Ahmed body when the horizontal jets were deflected inward with Coanda surfaces on all four edges (Barros *et al.* 2016). Improved drag reduction with inward as opposed to tangential blowing was also observed for the 35° high-drag Ahmed body (Zhang *et al.* 2018) and the square back version Schmidt *et al.* (2015).

Table 6 summarizes the discussed flows, associated drag reduction and actuation parameters. For brevity, we refer to flows with no, one-dimensional, five-dimensional and ten-dimensional actuation spaces as case A, B, C and D, respectively. The actuation energy may be conservatively estimated by the energy flux through all jet actuators: $\sum_{i=1}^{5} \int dA_i \rho U_i^3/2$. Here, the actuation jet fluid is assumed to be accelerated from 0 to the actuation jet velocity U_i and then deflected after the outlet, e.g. via a Coanda surface. In this case, the actuation energy of cases B, C and D would correspond to 3.2%, 3.0% and 7.9% of the parasitic drag power, respectively. This expenditure is significantly less than the saved drag power. The ratio from saved drag power to actuation energy is comparable for a truck model where steady Coanda blowing with 7% energy expenditure yields a 25% drag reduction (Pfeiffer & King 2014). This estimate should not be taken too literally as actuation energy strongly depends on the realization of the actuator. It would be less, more precisely $\sum_{i=1}^{5} \int dA_i \, \rho \cos(\theta_i) \, U_i^3/2$. when the actuation jet fluid leaves the Ahmed body through a slot directed with the jet velocity and can be expected much less when this fluid is taken from the oncoming flow, e.g. from the front of the Ahmed body.

Figure 17 displays iso-surfaces for the same Okubo-Weiss parameter value Q for all four cases. The unforced case A (figure 17*a*) shows a pronounced C-pillar vortices extending far into the wake. Under streamwise top actuation (case B, figure 17*b*), the C-pillar vortices significantly shorten. The next change with all streamwise actuators optimized (case C) is modest consistent with the small additional drag decrease. The C-pillar vortices are slightly more shortened (see figure 17*c*). The inward deflection of the actuation (case D) is associated with aerodynamic boat tailing as displayed in figure 17*d*. The separation from the slanted window is significantly delayed and the sidewise separation is vectored inward.

This actuation effect on the C-pillar vortices is corroborated by the streamwise vorticity contours in a transverse plane on body height downstream ($x/H = 1$). Figure 18 shows

this averaged vorticity component for case A–D in subfigure *a*–*d*, respectively. The extend of the C-pillar vortices clearly shrink with increasing drag reduction.

Figure 19 shows the streamwise velocity component and streamlines of the transversal velocity in the same plane for the same cases. Cases B and C feature a larger region of upstream flow while case D has a narrowed regions of backflow. From these visualizations, one may speculate that the drag reduction from streamwise actuation (cases B and C) is due to a wake elongation towards the Kirchhoff solution while the inward directed actuation (case D) is associated with drag reduction from aerodynamic boat-tailing.

This hypothesis about different mechanisms of drag reduction is corroborated from the streamlines in the symmetry plane $y = 0$ in figure 20. The tangential blowing (see subfigures *b*, *c* leads to an elongated fuller wake as compared to the unforced benchmark (subfigure *a*). The top shear-layer is oriented more horizontal under streamwise actuation—consistent with the Kirchhoff wake solution. The inward-directed actuation (see subfigure *d*) also elongates the wake but gives rise to a more streamlined shape. The top and bottom shear-layers are vectored inward.

The drag reduction can more directly be inferred from the C_p distribution of the rearward windows in figure 21. The 5% drag reduction in subfigure *b*) for case B is associated with a pressure increase of the vertical surface. The additional 2% drag decrease for case C in subfigure *c* is accompanied by an increase over vertical and slanted surface. The aerodynamic boat-tailing of case D with 17% drag reduction alleviates significantly the pressures on both surfaces.

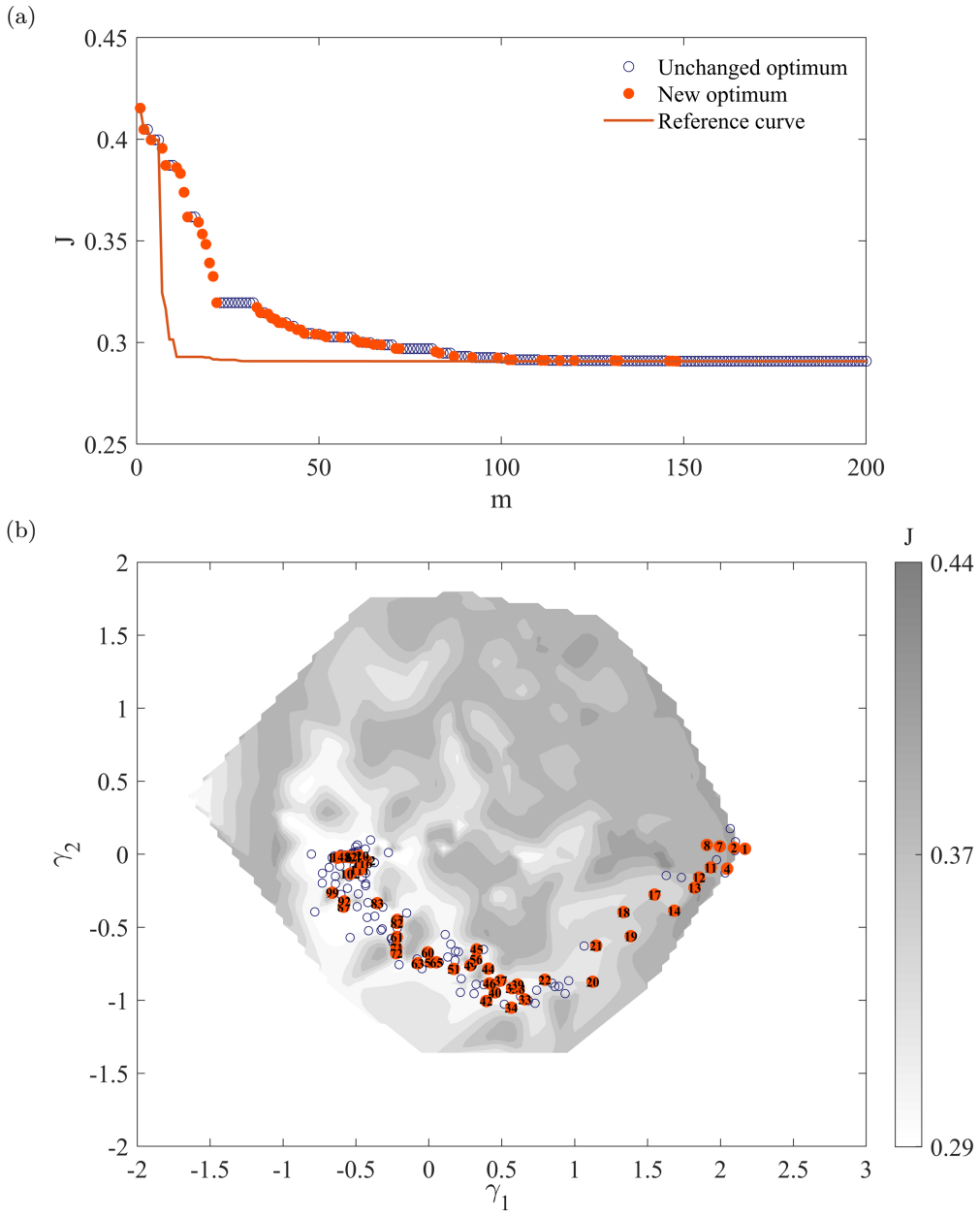


Figure 12: Optimization of five tangential jet actuator groups with the Simplex method. The top figure displays the best achieved drag reduction in terms of the number of evaluations (RANS simulations). The bottom figure shows the proximity map of all evaluated actuations. The contour plot corresponds to the interpolated cost function (drag coefficient) from all RANS simulations of this section. As in section 3, solid red circles mark newly find optima while open blue circles mark unsuccessful explorations of cost functions.

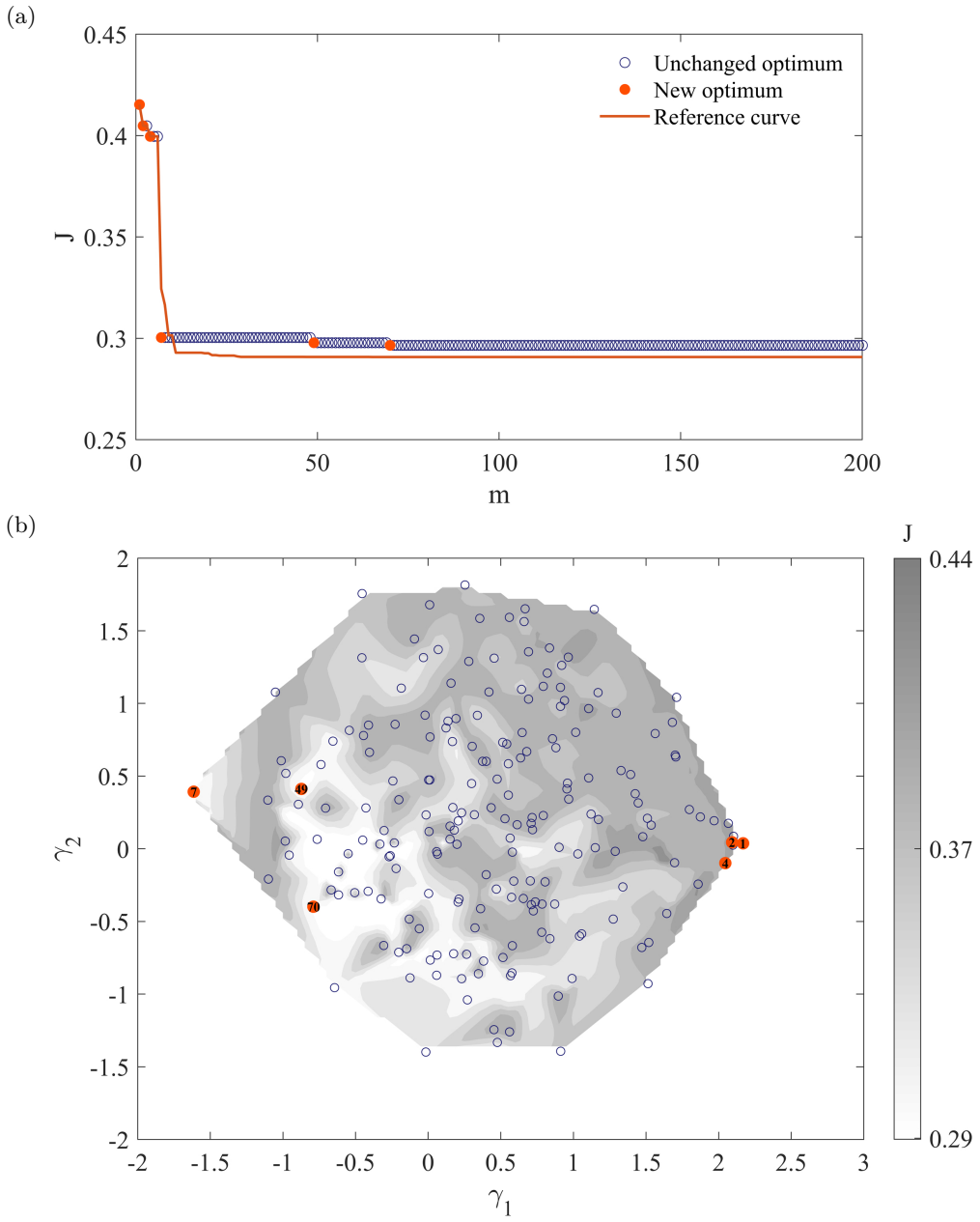


Figure 13: Optimization of five tangential jet actuator groups with the LHS method. The top figure displays the best achieved drag reduction in terms of the number of evaluations (RANS simulations). The bottom figure shows the proximity map of all evaluated actuations. The contour plot corresponds to the interpolated cost function (drag coefficient) from all RANS simulations of this section. As in section 3, solid red circles mark newly find optima while open blue circles mark unsuccessful explorations of cost functions.

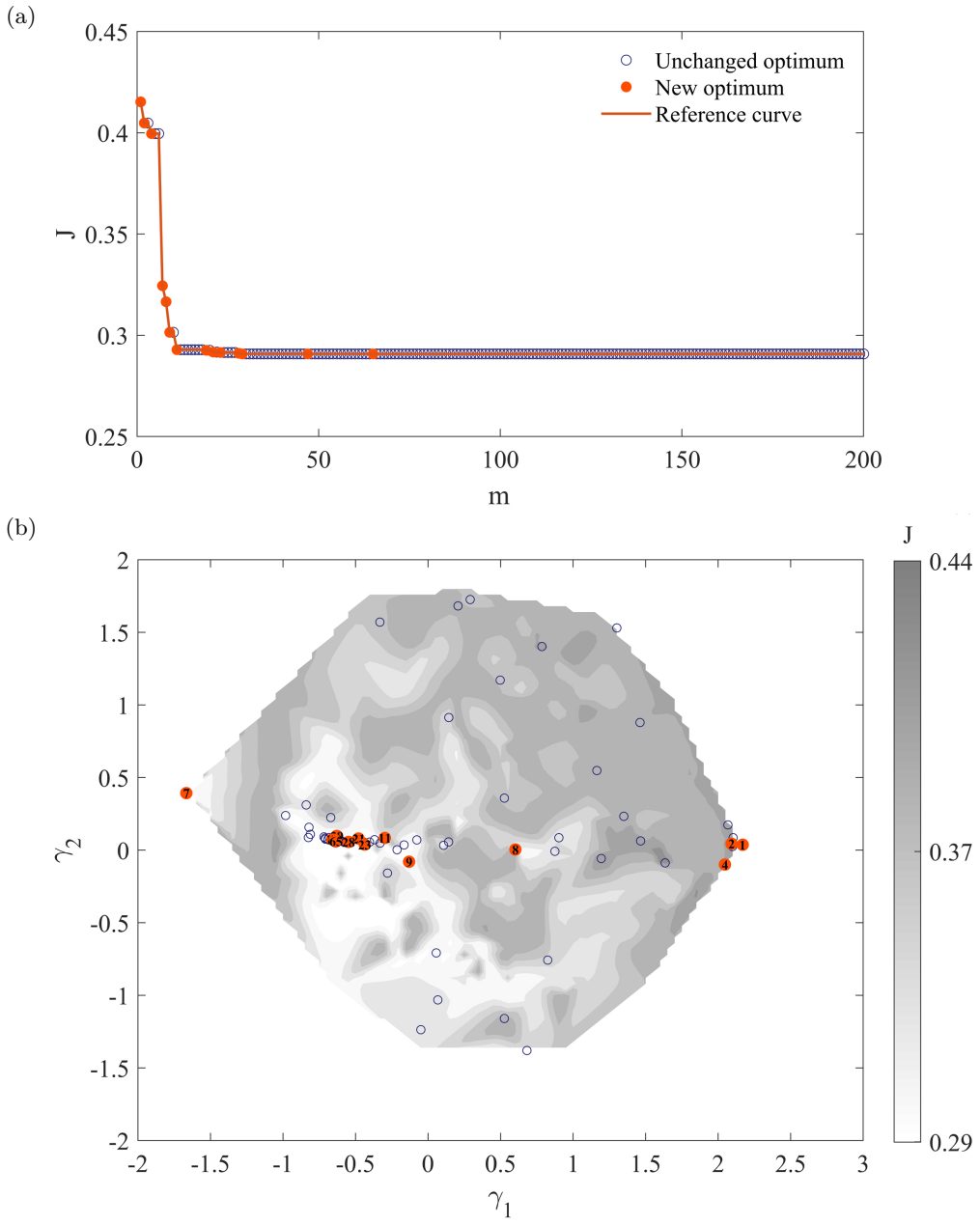


Figure 14: Optimization of five tangential jet actuator groups with the explorative gradient method. The top figure displays the best achieved drag reduction in terms of the number of evaluations (RANS simulations). The bottom figure shows the proximity map of all evaluated actuations. The contour plot corresponds to the interpolated cost function (drag coefficient) from all RANS simulations of this section. As in section 3, solid red circles mark newly find optima while open blue circles mark unsuccessful explorations of cost functions.

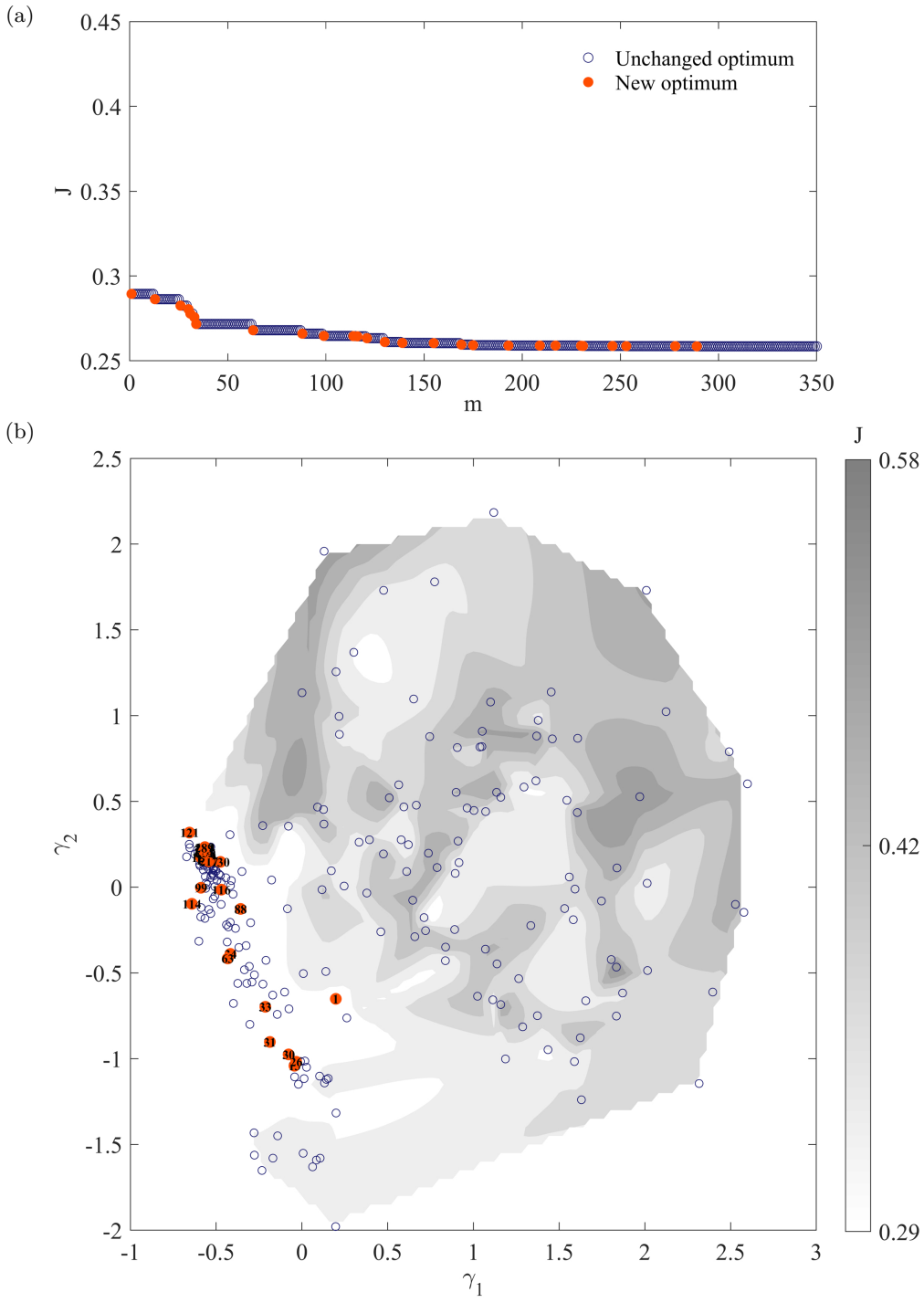


Figure 15: Like 14 but for the 10-dimensional optimization of the orientable trailing edge actuation.

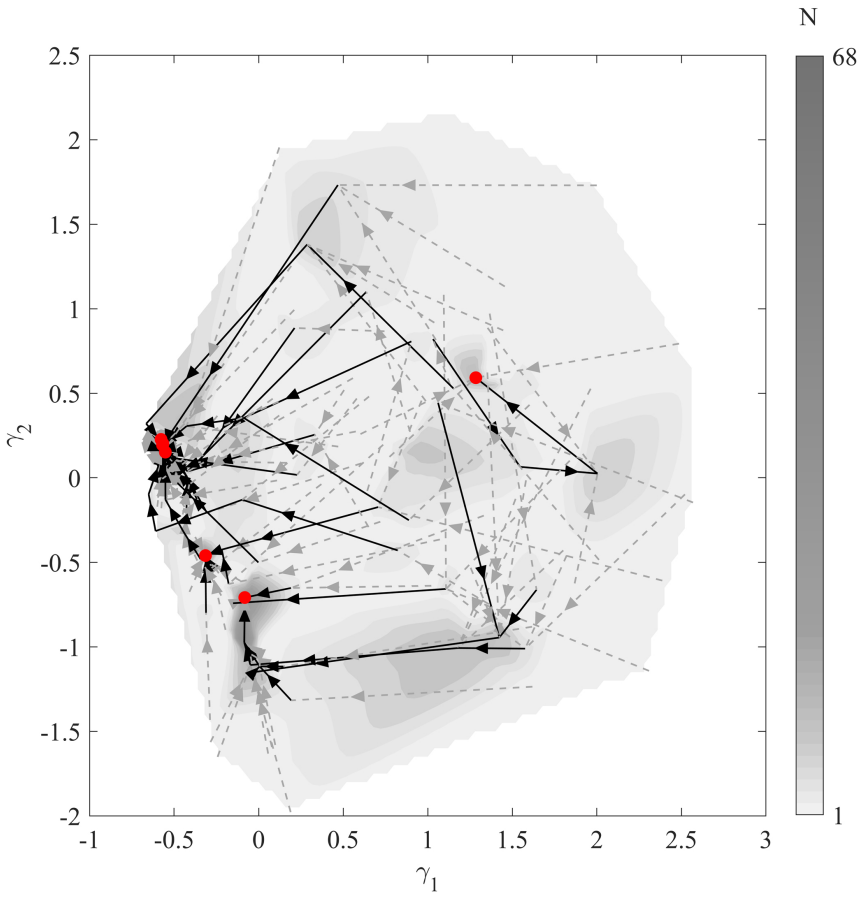


Figure 16: Steepest decent lines of the control landscape depicted in 15. For details see text.

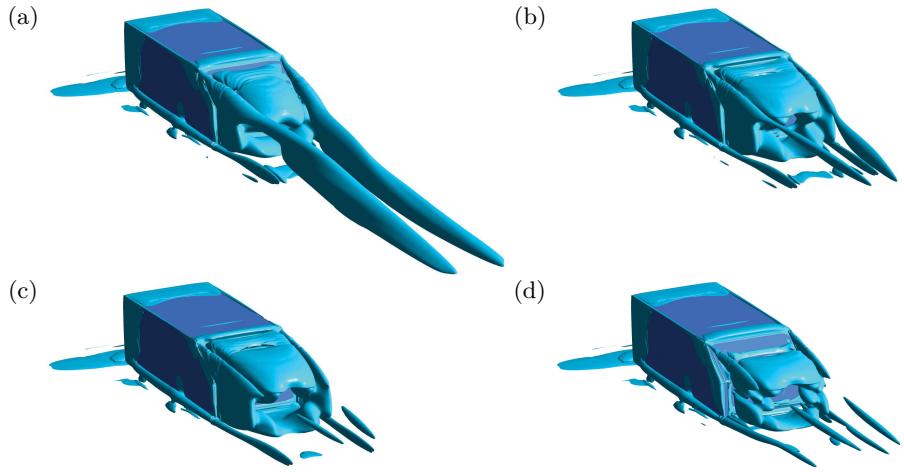


Figure 17: Okubo-Weiss parameter Q of flow. *a*) without control and under *b*) 1D, *c*) 5D and *d*) 10D control respectively, where $Q = 15000/s^2$.

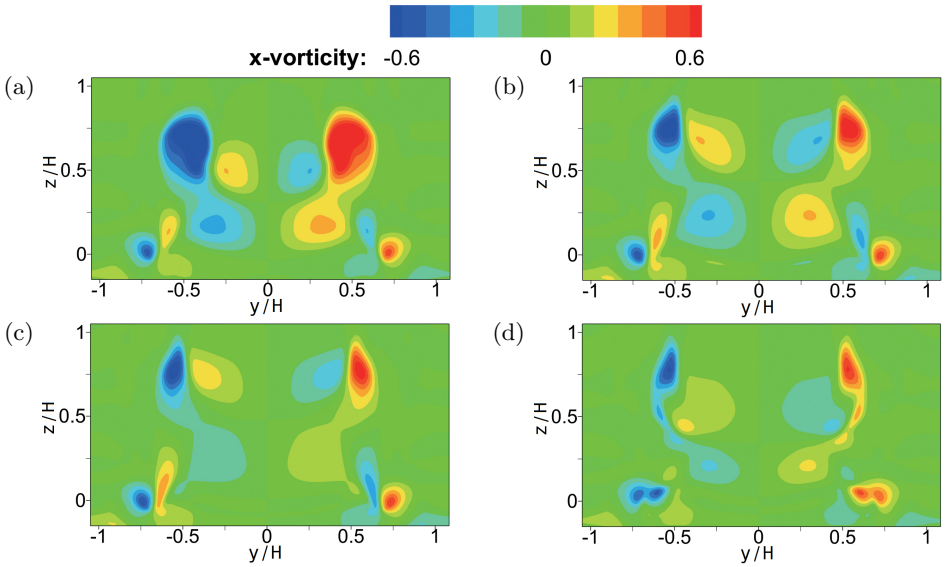


Figure 18: Streamwise vorticity component in near-wake plane $x/H = 1$. *a*) without forcing and under *b*) 1D, *c*) 5D and *d*) 10D control respectively.

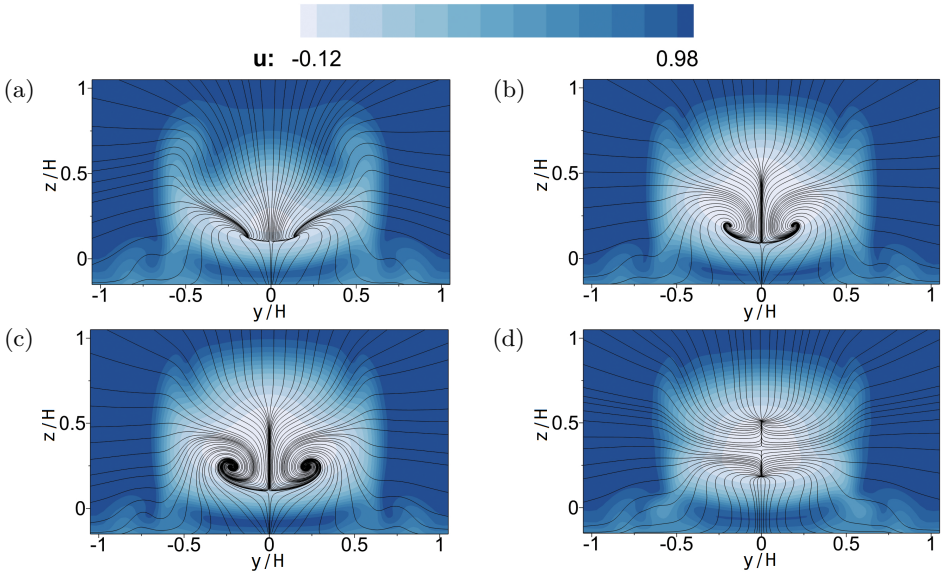


Figure 19: Streamwise velocity component in the near-wake tranversal plane $x/H = 1$ and streamlines from the in-plane velocity components. *a*) without forcing and under *b*) 1D, *c*) 5D and *d*) 10D control respectively.

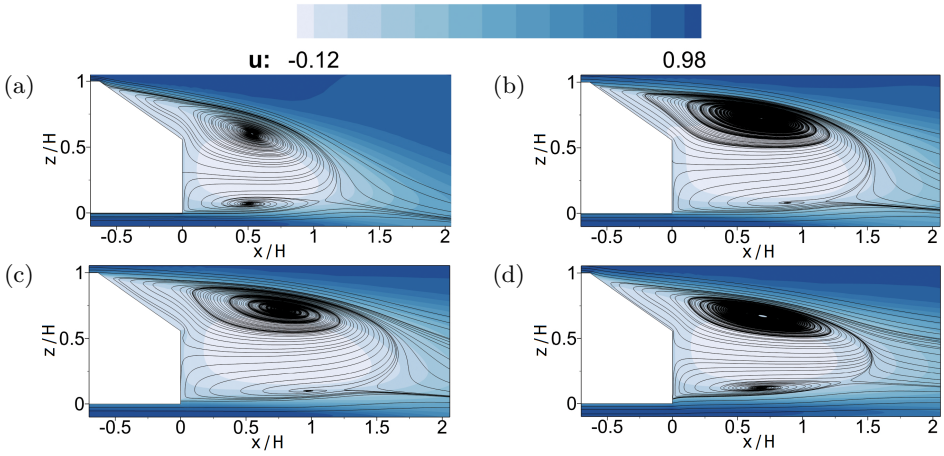


Figure 20: Streamwise velocity component in the symmetry plane $y = 0$ and streamlines from the in-plane velocity components. *a*) without forcing and under *b*) 1D, *c*) 5D and *d*) 10D control respectively.

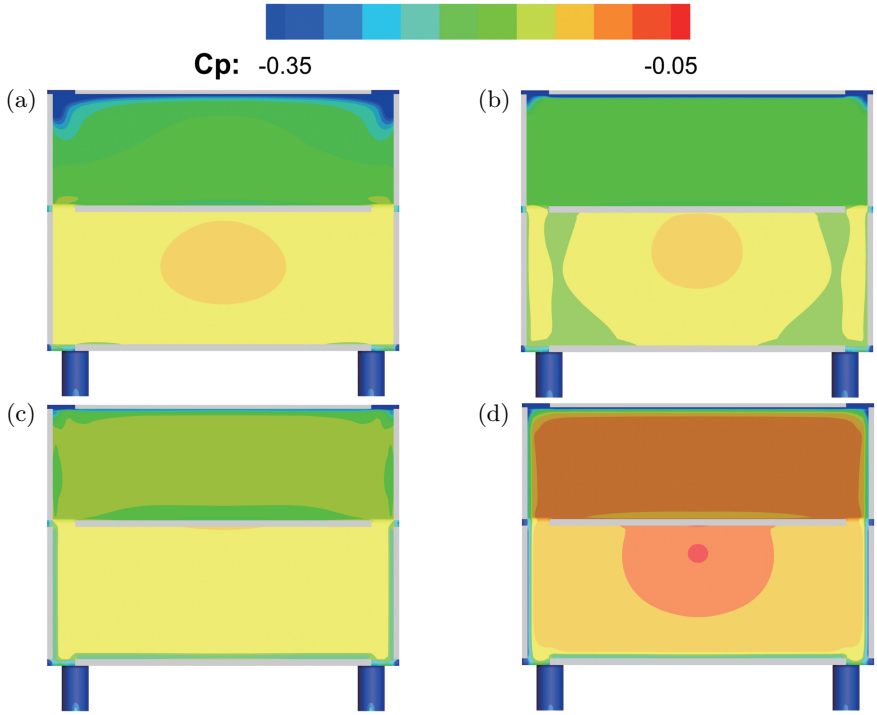


Figure 21: Pressure coefficient on the slant and vertical base of flow. *a*) without forcing and under *b*) 1D, *c*) 5D and *d*) 10D control respectively.

6. Conclusions

This numerical study proposes a novel optimization approach for active bluff-body control exploiting local gradients with a downhill simplex algorithm and exploring new better minima with Latin hypercube sampling. The computational load for the exploration is less than for the simplex iterations. This compares favourably with a shotgun downhill optimization typically requiring dozens of converged downhill simplex applications.

This approach named *explorative gradient method (EGS)* minimizes the drag of an 35° slanted Ahmed body at Reynolds number $Re_H = 1.9 \times 10^5$ with independent steady blowing at all trailing edges. The 10-dimensional actuation space includes 5 symmetric jet slot actuators or corresponding actuator groups with variable velocity and variable blowing angle. The resulting drag is computed with a Reynolds-Averaged Navier-Stokes (RANS) simulation.

The approach is augmented by auxiliary methods for initial conditions, for accelerated learning and for a control landscape visualization. The initial condition for a RANS simulation with a new actuation is computed by the 1-nearest neighbour method. In other words, the RANS simulation starts with the converged RANS flow of the closest hitherto examined actuation. This cuts the computational cost by 60% as it accelerates RANS convergence. The actuation velocities are quantized to prevent testing of too similar control laws. This optional element reduces the CPU time by roughly 30%. The learning process is illustrated in a control landscape. This landscape depicts the drag in a proximity map—a two-dimensional feature space from the high-dimensional actuation response. Thus, the complexity of the optimization problem can be assessed.

In a analytical example, the explorative gradient method is found to outperform the downhill simplex method converging to suboptimal minimum and a Latin hypercube sampling needing too many iterations. The slanted Ahmed body with 1, 5 and 10 actuation parameters constitutes a more realistic plant for an optimization algorithm. First, only the upper streamwise jet actuator is optimized. This yields drag reduction of 5% with pronounced global minimum for the jet velocity. Second, the drag can be further reduced to 7% with 5 independent streamwise symmetric actuation jets. Intriguingly, the actuation effects of the actuator are far from additive—contrary to the experimental observation for the square-back Ahmed body Barros (2015). The optimal parameters of a single actuator are not closely indicative for the optimal values of the combined actuator groups. The control landscape depicts a long curved valley with small gradient leading to a single global minimum. Interestingly, the explorative step is not only a security policy for the right minimum. It also helps to accelerate the optimization algorithm by jumping out of the valley to a point closer to the minimum.

A significant further drag reduction of 17% is achieved when, in addition to the jet velocities, also the jet angles are included in the optimization. Intriguingly, all trailing edge jets are deflected inward mimicking the effect of Coanda blowing and leading to fluidic boat tailing. The C-pillar vortices are increasingly weakened with one-, five- and ten-dimensional actuation. Compared with the pressure increase at C-pillar in one- and five-dimensional control, the ten-dimensional control brings a substantial pressure recovery over the entire base. The achieved 17% drag decrease with constant blowing is comparable with the experimental 20% reduction with high-frequency forcing by Bideaux *et al.* (2011); Gilliéron & Kourta (2013).

For the 25° high-drag Ahmed body, (Zhang *et al.* 2018) have achieved 29% drag reduction with steady blowing at all sides, thus significantly outperforming all hitherto existing active flow control studies cited therein. The actuation has only been investigated for few selected actuation values. Hence, even better drag reductions are perceivable. Yet, the

unforced high-drag Ahmed body has a significantly higher drag coefficient of 0.361 than the low-drag version and is hence not fully comparable. Their reduced drag coefficient of 0.256 is almost identical with the one of this study.

We expect that our RANS-based active control optimization is widely applicable for virtually all multi-input steady actuations or combinations of passive and active control Bruneau *et al.* (2010). The explorative gradient method mitigates the chances of sliding down a suboptimal minimum at a acceptable cost. The 1-nearest neighbour method for initial condition and the actuation quantization accelerate the simulations and learning processes. And the control landscape provides the topology of the actuation performance, e.g. the number of local minima, nature and shape of valleys, etc.

7. Acknowledgements

This work is supported by Shanghai Key Lab of Vehicle Aerodynamics and Vehicle Thermal Management Systems (Grant No.18DZ2273300), by public grants overseen by the French National Research Agency (ANR) ANR-11-IDEX-0003-02 (iCODE Institute project, IDEX Paris-Saclay), and ANR-17-ASTR-0022 (FlowCon), 'ACTIV_ROAD'.

We have profited from stimulating discussions with Steven Brunton, Siniša Krajnović, Francois Lusseyran, Navid Nayeri, Oliver Paschereit, Luc Pastur, Richard Semaan, Wolfgang Schröder, Bingfu Zhang and Yu Zhou.

REFERENCES

Aider, J.-L., Beaudoin, J.-F. & Wesfreid, J. E. 2010 Drag and lift reduction of a 3d bluff-body using active vortex generators. *Exp. Fluids* **48**, 771–789.

Barros, D. 2015 Wake and drag manipulation of a bluff body using fluidic forcing. PhD thesis, École Nationale Supérieure de Mécanique et d'Aérotechnique, Poitiers, France.

Barros, D., Borée, J., Noack, B. R., Spohn, A. & Ruiz, T. 2016 Bluff body drag manipulation using pulsed jets and Coanda effect. *J. Fluid Mech.* **805**, 442–459.

Ben-Hamou, E., Arad, E. & Seifert, A. 2007 Generic transport aft-body drag reduction using active flow control. *Flow Turbul. Combust.* **78** (3-4), 365.

Bideaux, E., Bobillier, P., Fournier, E., Gilliéron, P., El Hajem, M., Champagne, J.-Y., Gilotte, P. & Kourta, A. 2011 Drag reduction by pulsed jets on strongly unstructured wake: towards the square back control. *Int. J. Aerodyn.* **1** (3-4), 282–298.

Bruneau, C.-H., Creusé, E., Depeyras, D., Gilliéron, P. & Mortazavi, I. 2010 Coupling active and passive techniques to control the flow past the square back Ahmed body. *Comput. & Fluids* **39** (10), 1875–1892.

Brunn, A & Nitsche, W 2006 Drag reduction of an ahmed car model by means of active separation control at the rear vehicle slant. In *New Results in Numerical and Experimental Fluid Mechanics V*, pp. 249–256. Springer.

Brunton, S. L. & Noack, B. R. 2015 Closed-loop turbulence control: Progress and challenges. *Appl. Mech. Rev.* **67** (5), 050801:01–48.

Dejoan, A, Jang, Y. J. & Leschziner, M. A. 2005 Comparative LES and unsteady RANS computations for a periodically-perturbed separated flow over a backward-facing step. *J. Fluids Eng.* **127** (5), 872–878.

Geropp, D. 1995 Process and device for reducing the drag in the rear region of a vehicle, for example, a road or rail vehicle or the like. United States Patent **US 5407245 A**.

Geropp, D. & Odenthal, H.-J. 2000 Drag reduction of motor vehicles by active flow control using the Coanda effect. *Exp. Fluids* **28** (1), 74–85.

Gilliéron, P. & Kourta, A. 2013 Aerodynamic drag control by pulsed jets on simplified car geometry. *Exp. Fluids* **54** (2), 1457.

Gad-el Hak, M. 2006 *Flow Control: Passive, Active, and Reactive Flow Management*. Cambridge university press.

Han, X., Krajnović, S. & Basara, B. 2013 Study of active flow control for a simplified vehicle model using the PANS method. *Int. J. Heat Fluid Flow* **42**, 139–150.

Hucho, W.-H. 2002 *Aerodynamik der stumpfen Körper. Physikalische Grundlagen und Anwendungen in der Praxis*, 2nd edn. Wiesbaden: Vieweg Verlag.

Kim, J. 2011 Physics and control of wall turbulence for drag reduction. *Phil. Trans. Roy. Soc. A* **369** (1940), 1396–1411.

Krajnović, S. 2009 Large eddy simulation of flows around ground vehicles and other bluff bodies. *Phil. Trans. R. Soc. A* **367** (1899), 2917–2930.

Li, Y., Cui, W., Jia, Q. & Yang, Z. 2018 Wake control on a simplified vehicle using steady blowing. In *Proceedings of the 8th International Conference on Fluid Mechanics (ICFM8)*, pp. 1–6. Paper.

Maceda, Guy, Noack, Bernd, Lusseyran, F., Deng, Nan, Pastur, Luc & Morzynski, Marek 2019 Artificial intelligence control applied to drag reduction of the fluidic pinball. *PAMM* **19**.

McKay, M. D., Beckman, R. J. & Conover, W. J. 1979 Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** (2), 239–245.

Minelli, G., Krajnović, S., Noack, B. R. & Basara, B. 2016 Active flow control of a frontstep with a rounded leading edge. *Flow, Turbul., Combust.* **97** (1), 1235–1254.

Muralidharan, K, Muddada, S. & Patnaik, B. S. V. 2013 Numerical simulation of vortex induced vibrations and its control by suction and blowing. *Appl. Math. Model.* **37** (1-2), 284–307.

Nelder, J. A. & Mead, R. 1965 A simplex method for function minimization. *J. Comput.* **7**, 308–313.

Park, H. Lee, D. Jeon, W.-P. Hahn, S. Kim, J. Choi, J. & Choi, H. 2006 Drag reduction in flow over a two-dimensional bluff body with a blunt trailing edge using a new passive device. *J. Fluid Mech.* **563**, 389–414.

Pastoor, M., Henning, L., Noack, B. R., King, R. & Tadmor, G. 2008 Feedback shear layer control for bluff body drag reduction. *J. Fluid Mech.* **608**, 161–196.

Pfeiffer, J. & King, R. 2014 Linear parameter varying active flow control for a 3d bluff body exposed to cross-wind gusts. In *AIAA Paper, 2014-2406*.

Press, W.H. Flamery, B.P. Teukolsky, S.A. & Vetterling, W.T. 2007 *Numerical Recipes, The Art of Scientific Computing*, 3rd edn. Cambridge, UK, etc.: Cambridge University Press.

Schmidt, H.-J., Woszidlo, R., Nayeri, C. N. & Paschereit, C. O. 2015 The effect of flow control on the wake dynamics of a rectangular bluff body in ground proximity. *Exp. Fluids* **56**, 151.

Viken, S., Vatsa, V., Rumsey, C. & Carpenter, M. 2003 Flow control analysis on the hump model with RANS tools. In *41st Aerospace Sciences Meeting and Exhibit*, p. 218.

Wahde, M. 2008 *Biologically Inspired Optimization Methods: An Introduction.* WIT Press.

Wright, Alden H 1991 Genetic algorithms for real parameter optimization. In *Foundations of genetic algorithms*, , vol. 1, pp. 205–218. Elsevier.

Zhang, B. F., Liu, K., Zhou, Y., To, S. & Tu, J. Y. 2018 Active drag reduction of a high-drag ahmed body based on steady blowing. *J. Fluid Mech.* **856**, 351—396.