Closed-loop separation control using machine learning

N. Gautier[†], J.-L. Aider[†], T. Duriez[⋆,§], B. R. Noack[⋆], M. Segond[‡], and M. Abel[‡]

[†] Laboratoire de Physique et Mécanique des Milieux Hétérogènes (PMMH), UMR7636 CNRS, École Supérieure de Physique et Chimie Industrielles de la ville de Paris
10 rue Vauquelin, 75005 Paris, France
[⋆] Institut PPRIME, CNRS - Université de Poitiers - ENSMA, UPR 3346
Département Fluides, Thermique, Combustion CEAT,
43, rue de l'Aérodrome F-86036 Poitiers cedex, France
§ Laboratorio de FluidoDinámica, CONICET / Universidad de Buenos Aires,
Facultad de Ingeneria, Paseo Colon 850, Ciudad Autonoma de Buenos Aires, Argentina
[‡] Ambrosys GmbH,
Albert-Einstein-Str. 1-5, D-14469 Potsdam, Germany

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

We present the first closed-loop separation control experiment using a novel, model-free strategy based on genetic programming,s called 'machine learning control' in the sequel. The goal is to reduce the recirculation zone of backward-facing step flow at $Re_h = 1350$ manipulated by a slotted jet and optically sensed by online PIV. The feedback control law is optimized with respect to a cost functional based on the recirculation area and a penalization of the actuation. This optimization is performed employing genetic programming. After 12 generations comprised of 500 individuals, the algorithm converges to a feedback law which reduces the recirculation zone by 80%. This machine learning control is benchmarked against the best periodic forcing which excites Kelvin-Helmholtz vortices. The machine learning control yields a new actuation mechanism resonating with the low-frequency flapping mode instability. This feedback control performs similarly to periodic forcing at the design condition but outperforms periodic forcing when the Reynolds number is varied by a factor two. The current study indicates that machine learning control can effectively explore and optimize new feedback actuation mechanisms in numerous experimental applications.

1. Introduction

Flow control is a rapidly evolving interdisciplinary field comprising many disciplines, like fluid mechanics, technological innovations for sensors and actuators, control theory, optimization and machine learning. Its potential engineering applications have an epic proportion, including aerodynamic of cars, trucks, trains, wind-turbines or gas-turbines, as well as medical equipments or chemical plants. Flow control is employed to reduce aerodynamic drag for cars (Beaudoin & Aider 2008; Gillieron & Kourta 2010; Joseph *et al.* 2013), to find alternative lift-off and take-off configurations for aircraft and or to improve mixing efficiency (M'Closkey *et al.* 2002).

There have been many successful implementations of passive and active open-loop flow control (Fourrié *et al.* 2010; Joseph *et al.* 2012; Gautier & Aider 2013*b*). Closed-loop control offers great potential for increased robustness and efficiency and is currently the subject of an increasing ongoing research efforts (Henning & King 2007; Tadmor *et al.* 2010; Beaudoin *et al.* 2006; Pastoor *et al.* 2008; Brandt *et al.* 2011; Semeraro *et al.* 2011;

Gautier & Aider 2013*a*). In experiments, most closed-loop controls are adaptive (Henning & King 2007; Beaudoin *et al.* 2006; Gautier & Aider 2013*a*) and based on slowly varying periodic forcing. In-time control remains very challenging because of the non-linear nature of fluid phenomena. Most model-based control designs are based on a (locally) linear reduced-order model and ignore frequency cross-talk. The low-dimensionality of the model is key for robustness and online capability in experiments (Noack *et al.* 2011; Bergmann & Cordier 2008; Hervé *et al.* 2012). Only a few control-oriented reduced-order models address frequency cross-talk (Luchtenburg *et al.* 2009; Luchtenburg 2010).

The challenges of model-based control design have led us to search for model-free control laws using machine learning methods such as evolutionary algorithms (Wahde 2008) or artificial neural networks (Lee *et al.* 1997). These have been successfully used in many disciplines such as bio-informatics, medicine and computer science (Harik 1997; Ferreira 2001; Shah-Hosseini 2009).

This study builds on and extends pioneering works in machine learning based control by Rechenberg (1994), Lee et al. (1997) and Milano et al. (2002) to name a few. The relevant techniques to achieve such a goal are genetic algorithms (GA), artificial neural networks (ANN) and genetic programming (GP). GAs where first suggested for flow control by (Rechenberg 1994). They have been used in fluid mechanics for shape optimization (Toivanen *et al.* 1999; Gardner & Selig 2003), to optimize feedback control schemes for wall Turbulence (Morimoto *et al.* 2002), to find optimal wing configurations in insect flight (Berman & Wang 2007), to minimize drag on a cylinder in numerical simulations (Milano & Koumoutsakos 2002*a*) or to fine tune a control law to minimize drag in a experimental turbulent channel flow (Milano & Koumoutsakos 2002*b*). It should be noted that in these examples featuring GA the goal is optimizing tuning parameters. The algorithm is used to search for the best parameters of a given, set control law. As GA can tunes parameters, it can only be employed to optimize given designs of control laws. Model-free control using machine learning methods has been pioneered by Lee *et al.* (1997) using ANN (Kim 2003). While ANN can approach solutions by a complex combination (described by the network structure) of sigmoids functions, the performance of such an algorithm depends on the learning scheme implemented. If the classical errors back-propagation learning scheme is implemented, then the search algorithm is gradient based and thus sensitive to local minima. Genetic algorithms can be used to avoid this problem.

The third technique (GP) is used to find a control law optimizing a cost function. As with GA, an exploration of the search space is achieved alongside cost functional minimization. The main difference and advantage compared to GA is that GP is optimizing arbitrary functions, allowing for use in a model-free manner and thus explore a larger search space. When comparing GP to ANN the advantage lies in the output of the GP, which is the expression of the function. This allows the study of the control law and the gain of knowledge on the flow physics.

These algorithms are commonly used in many logistic and pattern recognition tasks. However GP-based control laws are rare in experimental closed-loop flow control. One of the obstacles to application of GP to experimental flow control is that a large number of experiments is required to fulfill the criterion for statistical convergence. Recently, Duriez *et al.* (2014) used GP to find closed-loop control laws in flow control problems. This approach proved surprisingly effective when applied on complex dynamical systems to closed-loop turbulence control in an experiment (Parezanovic *et al.* 2014).

The objective of the present study is to use GP to control a separated flow for the first time. The control objective is chosen to be the reduction of recirculation bubble area downstream a backward-facing step (BFS). In Sect. 2, the experiment is described. The

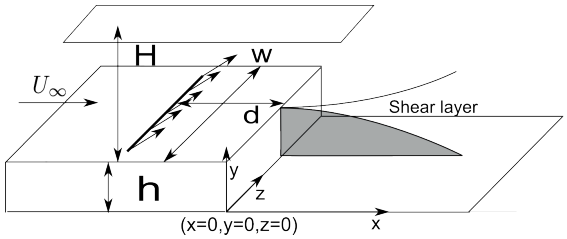


Figure 1: Sketch of the BFS geometry, position of the slotted jet and definition of the main parameters.

genetic programming control is presented in Sect. 3. The closed-loop control results are discussed and benchmarked against periodic forcing in Sect. 4. Sect. 5 summarizes the main finding and provides future directions.

2. Experimental Setup

2.1. *Water tunnel*

Experiments were carried out in a hydrodynamic channel in which the flow is driven by gravity. The flow is stabilized by divergent and convergent sections separated by honeycombs. The quality of the main stream can be quantified in terms of flow uniformity and turbulence intensity. The standard deviation σ is computed for the highest free stream velocity featured in our experimental set-up. We obtain $\sigma = 5.9 \times 10^{-4}\,\mathrm{m.s^{-1}}$ which corresponds to turbulence levels of $\sigma/U_\infty = 2.3 \times 10^{-3}$. For the present experiment the flow velocity is $U_\infty = 7.3 \times 10^{-2}\,\mathrm{m.s^{-1}}$ giving a Reynolds number based on step height $Re_h = U_\infty h/\nu = 1350$. This Reynolds number was chosen because of the limitations of the injection system.

2.2. *Backward-Facing Step geometry*

The BFS is considered a benchmark geometry for the study of separated flows: separation is imposed by a sharp edge creating a strong shear layer susceptible to Kelvin-Helmholtz instability. Upstream perturbations are amplified in the shear layer leading to significant downstream disturbances. This flow has been extensively studied both numerically and experimentally (Armaly *et al.* 1983; Hung *et al.* 1997; Beaudoin *et al.* 2004). The BFS geometry and the main geometric parameters are shown in figure 1. BFS height is $h = 1.5 \times 10^{-2}\,\mathrm{m}$. Channel height is $H = 7 \times 10^{-2}\,\mathrm{m}$ for a channel width $w = 15 \times 10^{-2}\,\mathrm{m}$. The vertical expansion ratio is $A_y = \frac{H}{h+H} = 0.82$ and the spanwise aspect ratio is $A_z = \frac{w}{h+H} = 1.76$. The injection slot is located $d/h = 2$ upstream of the step edge. The boundary layer thickness at the step edge is $\delta = 1.3h$.

2.3. *Sensor: 2D real-time velocity fields computations*

The sensor is built on a real-time computation of the vector fields. The velocity fields are computed based on snapshots from the seeded flow. The seeding particles are 20 μm neutrally buoyant polyamid particles. They are illuminated by a laser sheet created by a 2W continuous laser beam operating at $\lambda = 532$ nm. Images of the vertical symmetry plane are recorded using a Basler acA 2000-340km 8bit CMOS camera. Velocity fields are computed in real-time on a Gforce GTX 580 graphics card. The algorithm used to compute the velocity fields is based on a Lukas-Kanade optical flow algorithm called FOLKI developed by (Le Besnerais & Champagnat 2005). Its offline and online accuracy has been demonstrated and detailed by (Champagnat *et al.* 2011; Gautier & Aider 2014*b*).

Furthermore this acquisition method was successfully used in (Gautier & Aider 2013*a*; Davoust *et al.* 2012; Gautier & Aider 2014*a*). Velocity fields are computed over an area of $(17.2 \times 4.6) \times 10^{-4}\,\mathrm{m}^2$ which translates into a $9 \times 3\,h^2$ area. The time between two snapshots yielding one velocity field is $\delta t = 10 \times 10^{-3}\,\mathrm{s}$. 42 image pairs are processed per second. Figure 2 a) shows a typical example of the instantaneous velocity magnitude field downstream the step for the uncontrolled flow.

Recirculation plays a major role in the BFS flow and is overwhelmingly used for flow assessment as well as an objective for the control of flow separation (Henning & King 2007; Chun & Sung 1996). It has also been shown that the recirculation bubble can be linked to drag (although the relationship is far from trivial) (Dahan *et al.* 2012). We choose to evaluate the state of the flow through the instantaneous recirculation area, computed from instantaneous velocity fields, as our input. It is a 2D extension of the more common recirculation length evaluated using wall measurements. The recirculation area and recirculation length have been shown to behave the same way by Gautier & Aider (Gautier & Aider 2013*a*). The normalized instantaneous recirculation area $s(t)$ is computed using equation (2.1),

$$\begin{aligned} s(t) &= \frac{\int H(-u(x,y,t))\ \mathrm{d}x\mathrm{d}y}{A_0} \\ A_0 &= 1/T \int_0^T A_{uncont}(t)\mathrm{d}t, \end{aligned} \tag{2.1}$$

where H is the Heaviside function , $u(x, y, t)$ is streamwise velocity, A_0 is the time-averaged recirculation area for the uncontrolled flow and A_{uncont} is the instantaneous recirculation area for the uncontrolled flow. See figure 1 for x, y, z directions. The figure 2 b) shows the instantaneous recirculation area corresponding to the instantaneous velocity field shown on figure 2 b) and computed using equation (2.1).

2.4. *Actuator*

Actuation is provided using upstream injection through a spanwise slotted jet as shown in figure 1. The angle between the jet axis and the wall is 45^o. The jet flow is induced using a pressurized water tank. It enters a plenum and goes through a volume of glass beads designed to homogenize the incoming flow. The jet amplitude U_j is controlled by changing tank pressure. Because channel pressure is higher than atmospheric pressure this allows us to provide both blowing and suction. Maximum actuation frequency f_a is about 2Hz. To achieve closed-loop control, the control value $b = U_j/U_{max}$ (U_{max} being the maximum jet velocity) is computed as a function of the sensor value s inside a Labview project. The specific control laws are derived through genetic programming.

3. Genetic programming control

We propose a generic, model free, approach to closed-loop control of non-linear systems following (Duriez *et al.* 2014) we refer to this approach as machine learning control (MLC). Control laws are optimized with regard to a problem specific objective function using genetic programming (Koza *et al.* 1999). A first generation of control laws candidates $b_i^1(s)$, called individuals ($b_i^1(s)$ is the i^{th} individual of the 1^{st} generation), is randomly generated by combining user defined functions, constants and the sensor value s (see appendix A). Each individual is evaluated yielding a value for the cost function J. A new population b_i^2 is then generated by evolving the first generation. The procedure is iterated until either a known global minimum of J is reached or the evolution is stalled. This process is resumed in figure 3.

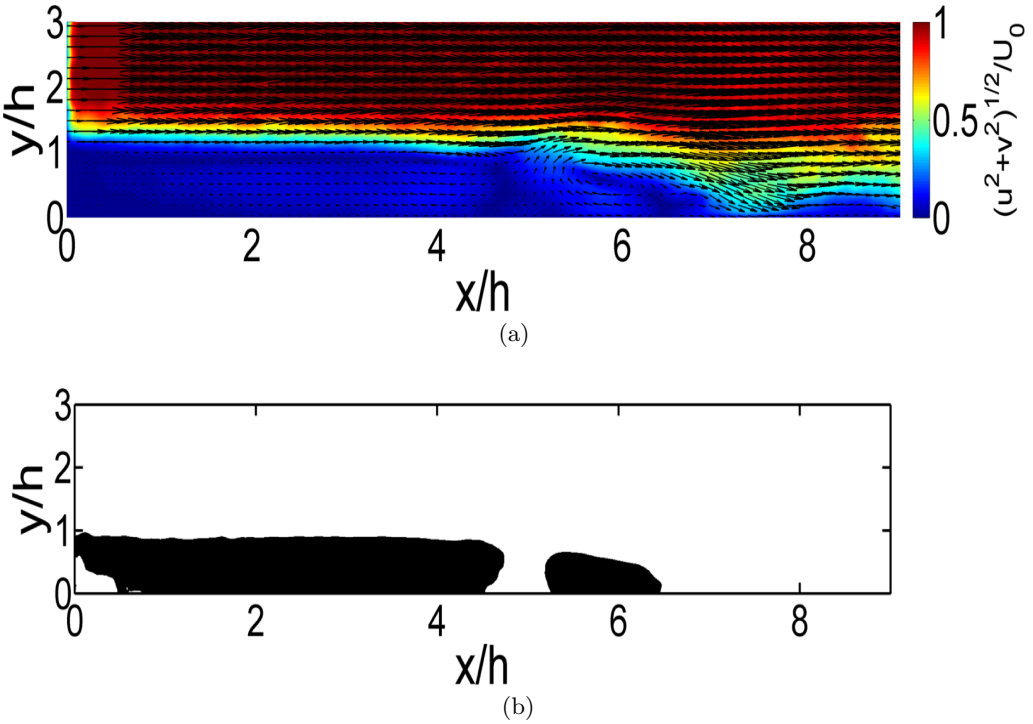


Figure 2: (a) Instantaneous velocity snapshot. Arrows show the instantaneous velocity field while the background color indicates the magnitude of the instantaneous velocity. The edge of the backward facing step is at $x = 0$ and $y = 1$. (b) Corresponding instantaneous mapping of the recirculation area. A black dot indicates a velocity vector with negative streamwise velocity.

3.1. *Population generation*

The number of individuals inside a generation has a strong influence on the process. While a large number of individuals will certainly lengthen the total time of the experiment, it will also ensure a higher diversity which is known to be a key parameter of all evolutive algorithms. In the present study, each generation is made of 500 individuals. This number of individuals is a good compromise between performance and testing time. It has proven to be enough to converge on most single input/single output problem, and is still manageable in terms of total experimental time.

The first individuals $b_i^1, 1 \leq i \leq 500$ are generated as expression trees made of user-defined nodes (see Appendix A). These nodes are functions (sin, cos, exp, log, tanh), basic operations $(+, -, \times, /)$, constants and the sensor input $s(t)$. The root of the expression tree, i.e. the value returned by the function it defines, is the control value. To build the expression tree, a recursive algorithm is used: a first node is chosen, then for each argument this node can accept, new nodes are added randomly until all terminal nodes do not accept any arguments (constants or sensor). The algorithm is made so that the first generation contains expression trees of different depth and density to ensure diversity in the population. Furthermore all individuals are different.

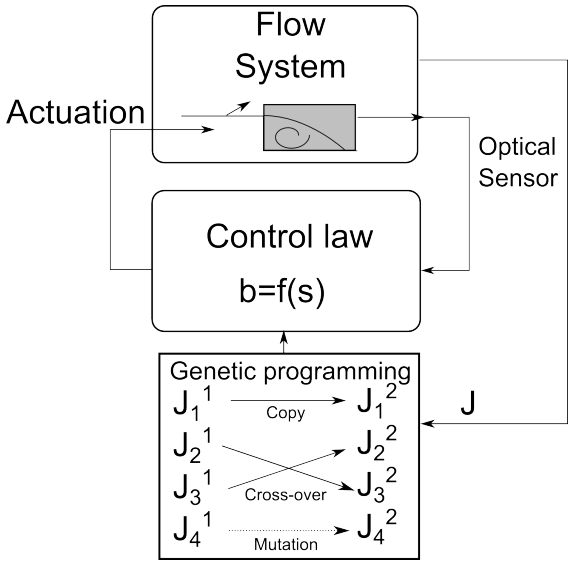


Figure 3: Control loop featuring genetic programming. Control laws $b(s)$ are evaluated by the flow system. This is done over several generations of individuals. New generations are generated by replication, cross-over and mutation J_i^n refers to the i^{th} individual of generation n.

3.2. *Evaluation*

Expression trees can be easily written as LISP expressions. The evaluation is done by translating the LISP expression (for example $(+(\sin s)(/\ 2.343\ s))$) into a control law inside the software responsible for the real-time closed-loop control ($b = \sin(s)+(2.343/s)$). The numerical value used to grade each individual is the cost function J linked to the control problem. In our case the goal is to reduce the recirculation area over the evaluation time with a penalization by the energy used for the actuation:

$$J = \langle s \rangle_T + w \langle |b| \rangle_T^2 > 0, \tag{3.1}$$

where T is the evaluation time ($T = 60\,\mathrm{s}$). The first component, quantifies the state of the flow and averages the sensor $s(t)$ during the evaluation time. $s(t)$ is normalized by the time-averaged uncontrolled recirculation area $A_0 =< s >_T$. Normalization is important as it allows corrections taking into account variations in the flow conditions (i.e. temperature variations, flow rate drifts). A_0 is recomputed every 250 individuals to compensate for any drifts. The second component, weighted by w, is tied to actuation energy and is normalized by the maximum jet velocity U_{max}. In the following, we choose $w = 3/2$, to strongly penalize high actuation costs. Although the choice for w is arbitrary, the value represents a trade-off between the gain on area reduction and actuation cost. Setting a low (respectively high) value of w means that the performance of the system is much more (respectively less) important than the cost of the control. A balanced value can be derived by evaluating how much one is ready to spend in energy to achieve a given performance. The ratio between performance gain and actuation cost of the most effective open-loop control (see section 4.3) suggests a value close to $w = 3/2$. This value was found to strike a good balance between recirculation reduction and actuation cost. It should be noted J could be appended to further constrain the control laws, for example it could be modified to penalize non zero-net mass flux actuation (by penalizing the

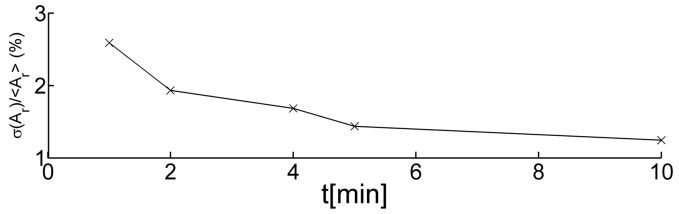


Figure 4: Standard deviation of mean recirculation area in percentage over ten evaluations for different evaluation periods.

difference between blowing and suction fluxes) or strong variations in actuation intensity (by penalizing the derivative of the control signal).

The evaluation time is also a key parameter as it will determine how long the whole experiment will last. Figure 4 shows that a one minute evaluation is enough to get significant statistics for an evaluation of J good enough to discriminate individuals by performance. As time is spent refilling the jet supply tank, the time between two evaluations varies and can reach two minutes. Approximately 1000 individuals, i.e. two generations, are evaluated over 24 hours.

Because control laws can be constants (e.g. $b = 0.2 + 0.17$) or be outside the actuator's operating range (e.g. $b = \exp(9.87) + (s \times s)$), each control law is pre-evaluated before it is applied to the flow. If it is found to saturate the actuator, it is assigned a very high cost. This step takes a few milliseconds and is done to ensure faster convergence by discarding uninteresting functions. Because of the random nature of the first generation most individuals saturate the actuator.

3.3. *Breeding of subsequent generations and stop criteria*

Once every individual of the current generation has been evaluated, they are sorted by their cost function value J. The five best individuals are evaluated again, the cost values are averaged and the population is sorted again. This re-evaluation procedure is repeated five times to ensure that the value of the best individuals is reliable. The individuals of the next generation are then produced through 3 different processes. Mechanisms are based on a tournament process: 7 individuals are randomly chosen, the individual elected to enter a breeding process is the one with the lowest cost function value. This ensures that the best individuals inside a generation will be used a lot, while less performing individuals still have a chance to be part of the next generation. Individuals selected this way will then be either replicated, mutated or crossed to generate the individuals of the next generation. The probabilities of replication, mutation and crossover are respectively 10%, 20% and 70%. This new generation is then evaluated and the whole process is iterated. The process can stop for two reasons. The first one is when J reaches 0 which in general does not occur. Most of the time, the process is stopped when the best values of J over the population stop improving over several generations.

4. Results

The approach described in § 3 has been applied to the backward-facing step plant presented in §2. Generation convergence is analyzed in § 4.1. The best control law of the final generation is presented in § 4.2. This control law is compared with open-loop forcing (§ 4.3) and tested for robustness with respect to the Reynolds number in §4.4.

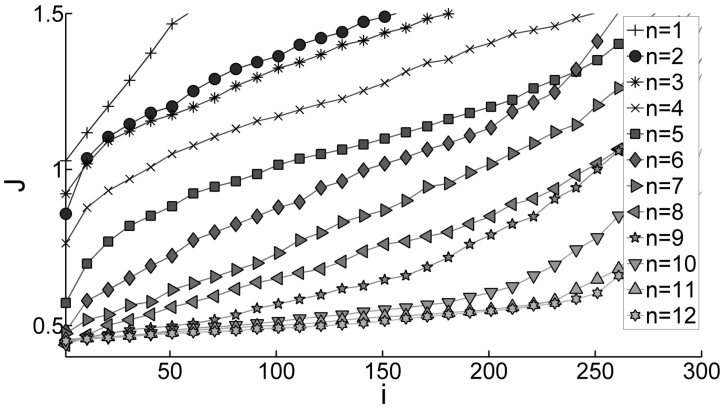


Figure 5: Cost functions of the first 300 individuals in all twelve generations.

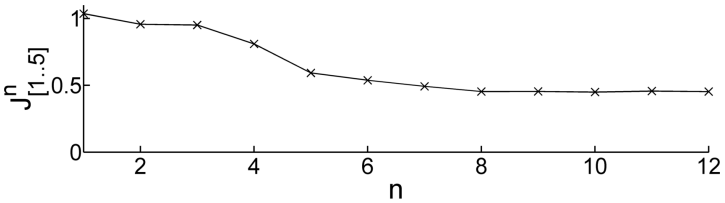


Figure 6: Cost functions averaged over the five best individuals for each generation

4.1. *Convergence of machine learning control*

The evolution of the cost function with increasing number of individuals is shown in Figure 5 for all twelve generations. All random control laws of the first generation are seen to be ineffective, i.e. produce only cost functions which are worse than the uncontrolled flow ($J_i^1 > 1$ for all i). Considering the large search space created by the many ways one can arrange the node functions, sensor and constants, it is not surprising that a Monte-Carlo process with 500 individuals is ineffective. A few effective control laws can be seen as soon as the second generation. The slope of the cost function J_i as a function of the index i is improved. This clearly shows that the search algorithm is effective in exploring and exploiting the search-space defined by the cost function. All subsequent generations perform better than the previous one, i.e. $J_i^{n+1} < J_i^n$. After the 9^{th} generation, the performance of the best individuals appear to converge.

The average of the best five control laws $J^n_{[1..5]} := \left(J_1^n + J_2^n + J_3^n + J_4^n + J_5^n\right)/5$ is shown in figure 6 for each generation. Convergence is reached after the 8^{th} generation. A good termination criterion appears to be to stop the iteration once the average of the cost function for the first 5 individuals no longer improves. In our case the experiment was stopped after no substancial enhancement was obtained in 5 generations from the 8^{th} to 12^{th} generation. As the number of generations increases the first 5 control laws become very similar and averaging over them is a more robust measure than taking just the best one.

The evaluation of the 12 generations translates into a week of continuous automatized experiments. At the end of the week, the training phase is finished and an effective, closed-loop control law has been obtained. While a week of experiment is an investment, it has to be compared with the time needed by other methods to obtain a viable real-time closed-loop control, which usually is accounted in months. This places this approach in the

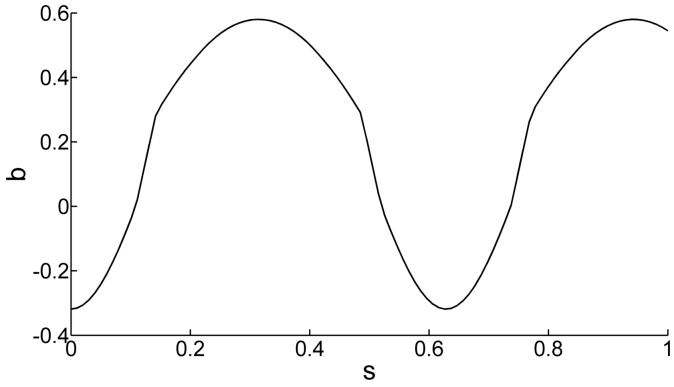


Figure 7: Graph of the best control law $b = K(s)$ ((4.1)) obtained after 12 generations.

category of control algorithm that can be used to obtain a controller in a comparatively small amount of time.

4.2. *Analysis of the best control law obtained through genetic programming*

The control law $b = K(s)$ has a complex mathematical expression, given in equation (4.1).

$$b = \exp -0.1138 \times \log(\cos(\log(s)) - \sin(\cos(\tanh(\sin(s))))). \tag{4.1}$$

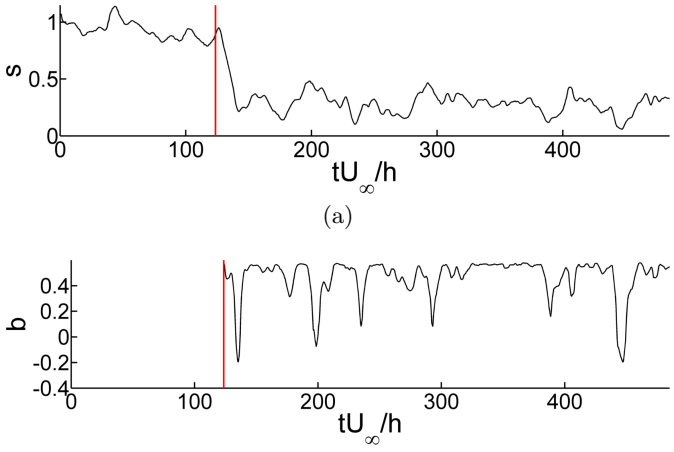
Yet, the graph of the best control law for the final generation has a simple structure as shown on figure 7 for $0 \leq s \leq 1$. Indeed, the controlled normalized recirculation area is by definition positive, $s \geq 0$. Experimentally, the instantaneous recirculation area associated with the best law is found to be always smaller than the averaged uncontrolled region resulting in $s \leq 1$. Note that the control law leads to a combination of blowing and suction as a function of s.

The actuation command b has an interesting non-proportional and non-monotonous dependency on s with two similar maxima $b \approx 0.6$ (injection) and two similar minima $b \approx 0.3$ (suction) which shows that it could not be obtained through a linear process. Near uncontrolled values for the recirculation zone ($s \sim 1$), large jet injection reduces the area. This injection decreases with s until suction sets in at intermediate values ($s \sim 0.73$). In the post-transient regime ($0.12 < s < 0.32$), injection increases with recirculation area. For $s < 0.12$, suction sets in. Most of the time, injection $b \approx 0.5$ occurs. During short periods with low recirculation zones, suction sets in or, at minimum, injection is significantly reduced.

It is interesting to look at the time-series of $s(t)$ (figure 8a) and the corresponding actuation amplitude $b(t)$ (figure 8b) for the best closed-loop control law. Once the control starts at time $t = 25$ s (vertical red line) the recirculation area is quickly and strongly decreased down to 20 %, corresponding to a 80% reduction on average. For $0 < s < 0.3$, actuation b is roughly an affine function of s (see figure 7). The figure 8b shows that the actuation is indeed a succession of short period of suction followed by a longer period of blowing ($b \approx 0.45$).

This feedback loop creates oscillations at 0.1 Hz, as observed in figures 8a, 8b and confirmed by the frequency analysis for the actuation signal shown in figure 9. We conjecture that this frequency is the flapping frequency of the recirculation bubble which is typically an order of magnitude lower than the shear layer shedding frequency (Spazzini *et al.* 2001) (shown in figure 10b), as illustrated by figure 9.

The 0.1 Hz feedback dynamics is probably triggered by the choice of our input, the



(a)

(b)

Figure 8: (a) System response s to the control law, the vertical line shows when control starts at $t = 120\,\mathrm{s}$

. (b) Corresponding actuation b.

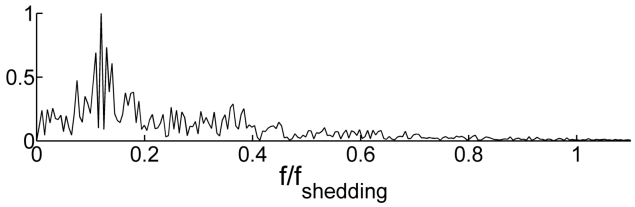


Figure 9: Normalized frequency spectrum obtained by Fourier transform of the actuation signal, frequency is normalized by vortex shedding frequency

instantaneous recirculation area, and the natural flapping frequency. The periodic events of reduced injection or low suction are awarded by the cost function which penalizes the actuation.

It is important to notice that the low frequency nature of this actuation enables "slow" actuators to positively affect high Reynolds number flows. It may remove strong constraints on the actuator in industrial settings which usually deal with high frequency vortex sheddings (typically a few hundreds Hz for full-scale automotive aerodynamics). In addition, it has been shown that recirculation area and recirculation length behave in the same way (Gautier & Aider 2013*a*). Wall pressure sensors could be used to evaluate recirculation length in real-time (Henning & King 2007) which could be used as an input to this new control law, making realistic applications viable .

4.3. *Comparison to periodic forcing*

Pulsing jet injection at the natural shedding frequency is an effective way of reducing recirculation area (Chun & Sung 1996; Pastoor *et al.* 2008; Gautier & Aider 2013*b*) making it a natural benchmark for this approach. We choose a periodic forcing at the Kelvin-Helmholtz frequency with a duty cycle of 50%, which means the jets are turned on half the time. An effective way of computing the natural shedding frequency is to compute the swirling strength criterion $\lambda_{ci}(s^{-1})$. This criterion was first introduced and subsequently improved by (Chong *et al.* 1990; Zhou *et al.* 1999). It was also recently

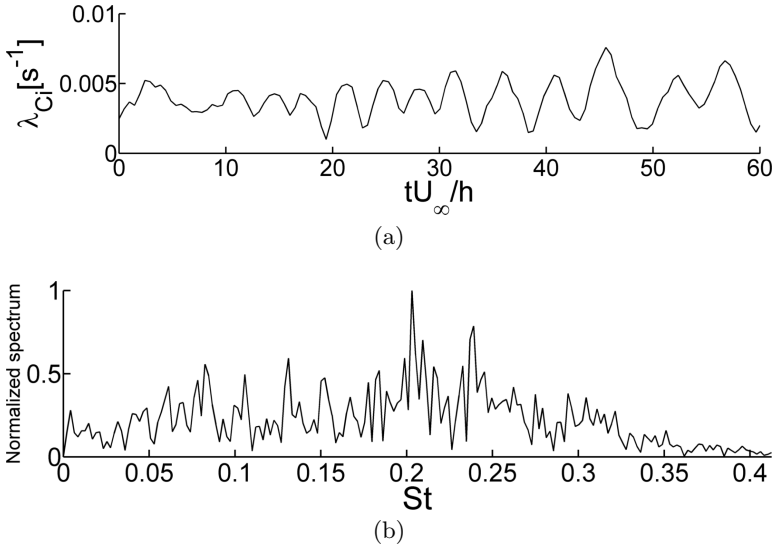


Figure 10: $\lambda_{Ci}(t)$ signal over 60 non-dimensionalized time units, uncontrolled flow, (b) Normalized frequency spectrum showing a peak at $S_t = 0.2$ corresponding to 0.97 Hz.

used as an input in closed-loop flow control experiments (Gautier & Aider 2013*a*). For 2D data λ_{Ci} can be computed quickly and efficiently following equation (4.2),

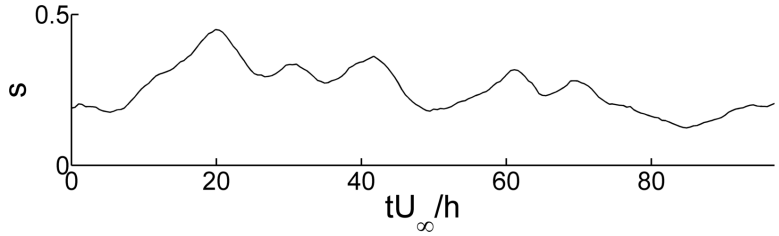
$$\lambda_{Ci} = \frac{1}{2}\sqrt{4\det(\nabla \mathbf{u}) - \mathrm{tr}(\nabla \mathbf{u})^{\mathbf{2}}}, \tag{4.2}$$

when such a quantity is real, else $\lambda_{Ci} = 0$. The shedding frequency is obtained by spatially averaging λ_{Ci} in the vertical direction at $x = 5h$. The sampling frequency is $f_s =$ 10Hz. This is equivalent to counting vortices as they pass through an imaginary vertical line at $x = 5h$. Figure 10a shows the corresponding scalar over 60 non-dimensionalized time units. Figure 10b shows the frequency spectrum obtained by Fourier transform. The natural shedding frequency is well defined and close to 1 Hz. This gives us the frequency for periodic forcing.

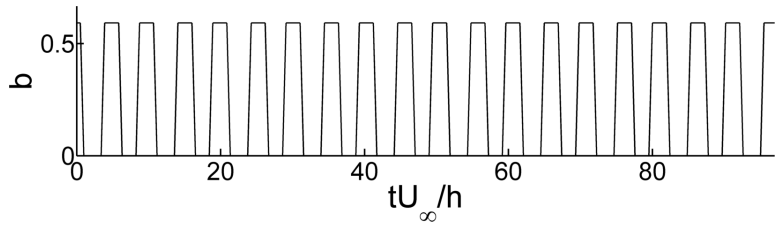
Figure 11a shows the reduction of the recirculation area using periodic forcing (figure 11b). Control is effective in reducing the recirculation. The cost function for this control law is $J_{periodic} = 0.423$ which is quite similar to the one found through genetic programming ($J_{genetic} = 0.419$). The MLC law still performs slightly better. Intriguingly, similar performances are achieved but with quite different dynamics and frequencies. The periodic forcing excites the Kelvin-Helmholtz frequencies at 1 Hz while the MLC law exploits the flapping frequency around 0.1 Hz.

4.4. *Robustness*

The control law was tested for various Reynolds numbers in order to test its robustness. Table 1 shows the resulting cost functions. The cost function does not increase by more than 20% while Reynolds number changes by a factor 2. The cost function for the open-loop forcing is also shown. This open-loop forcing is done at the optimal shedding frequency at $Re_h = 1350$ and is not changed with Reynolds number. Because it is tied to the recirculation area MLC control law adapts to changes in operating conditions, ensuring consistent, reliable performances. Because natural frequency changes with free-



(a)



(b)

Figure 11: (a) System response to periodic forcing over 100 convective time units. (b) Corresponding actuation.

Re_h	900	1350	1800
$J_{closed-loop}$	0.33	0.42	0.59
$J_{open-loop}$	0.75	0.42	0.76

Table 1: Cost function at different Reynolds numbers.

stream velocity, the open-loop control shows poor performance at different Reynolds numbers. Performance variations for the MLC law are due to changes in jet to cross-flow momentum ratio. To further improve robustness the control law could be amended in the following way: $b = f(Re_h)K(s)$ where f is a function to be determined.

5. Conclusion

Machine learning control has been used to determine a cost effective control law minimizing recirculation on a backward-facing step flow. During the twelve generations needed to converge towards the control law, the population has evolved toward solutions better fitting the problem, which is translated in lower cost function values. The process is stopped when no amelioration can be foreseen, judging by the statistical values returned by the algorithm. As no convergence can be proven, there is no guarantee this control is optimal. However the nature of the cost function allows to judge the performance of the solution and whether actuation can be rated as effective.

Without deriving a model for the input-output system, the genetic programming approach is able to converge on an efficient and robust control law linking a real-time measure of recirculation area to actuation value. Though the design of the experiment is kept at its simplest, an 80% reduction of the recirculation area has been achieved.

Moreover, the control law obtained highlights a mechanism which is not based on the exploitation of the Kelvin-Helmholtz instability as for the open-loop control, but rather on lower frequencies which are likely related with frequencies of the recirculation region.

Robustness can be increased by closed-loop laws. It is demonstrated that the control law designed by genetic programming for a given Reynolds number stays efficient in other operative conditions as compared to open-loop designed control with rapidly deterioring performance. Robustness would be reinforced by integrating a condition change during evaluation, though it would lengthen overall evaluation time.

Genetic programming has proven to be efficient at resolving multi-input/multi-output problems. Thus adding more freedom to the algorithm, by adding control outputs and sensors inputs, will increase the number of mechanisms the control laws can access in order to reduce the cost function. For instance adding time delays or derivative of the sensor would allow an embedding of the dynamical system and thus allow the control law to access more information on the state of the system.

This novel approach of experimental flow control competes with other approaches in terms of efficiency and robustness. Being model-free and capable of producing virtually any kind of control law (linear, non-linear, with history of sensors and actuators) it can be used in a systematic fashion, with a known time consumption on the plant. This enables the method to be used on flows with a specific geometry which has not been thoroughly investigated, such as a detailed vehicle model or a turbine geometry.

6. Acknowledgements

NG and JLA wish to thank the DGA for their support, TD and BN acknowledge funding by the French ANR (Chaire d'Excellence TUCOROM and SEPACODE), MS and MA acknowledge the support of the LINC project (no. 289447) funded by ECs Marie-Curie ITN program (FP7-PEOPLE-2011-ITN).

REFERENCES

Armaly, B.F., Durst, F., Pereira, J.C. & Schönung, B. 1983 Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.* **127**, 473–496.

Beaudoin, J.-F. & Aider, J.-L. 2008 Drag and lift reduction of a 3D bluff body using flaps. *Exp. Fluids.* **44** (4), 491–501.

Beaudoin, J-F., Cadot, O., Aider, J-L. & Wesfreid, J.E. 2004 Three-dimensional stationary flow over a backwards-facing step. *Eur. J. Mech. B-Fluid* **38**, 147–155.

Beaudoin, J-F., Cadot, O., Aider, J.-L. & Wesfreid, J.E. 2006 Drag reduction of a bluff body using adaptive control methods. *Phys. Fluids.* **18**, 085107.

Bergmann, M. & Cordier, L. 2008 Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *J. Comput. Phys.* **227**, 7813–7840.

Berman, G. & Wang, Z. 2007 Energy-minimizing kinematics in hovering insect flight. *J. Fluid Mech.* **582**, 153–168.

Brandt, L., Sipp, D., Pralits, J. & Marquet, O. 2011 Effect of base flow variation in noise amplifiers: the flat-plate boundary layer. *J. Fluid Mech.* **687**, 503–528.

Champagnat, F., Plyer, A., Le Besnerais, G., Leclaire, B., Davoust, S. & Le Sant, Y. 2011 Fast and accurate piv computation using highly parallel iterative correlation maximization. *Exp. Fluids.* **50**, 1169–1182.

Chong, M.S., Perry, A.E. & Cantwell, B.J. 1990 A general classification of 3-dimensional flow fields. *Phys. Fluids.* **2**, 765–777.

Chun, K. B. & Sung, H. J. 1996 Control of turbulent separated flow over a backward-facing step by local forcing. *Exp. Fluids.* **21**, 417–426.

Dahan, J.A., Morgans, A.S. & Lardeau, S. 2012 Feedback control for form-drag reduction on a bluff body with a blunt trailing edge. *J. Fluid Mech.* **704**, 360–387.

Davoust, S., Jacquin, L. & Leclaire, B. 2012 Dynamics of m = 0 and m = 1 modes and of streamwise vortices in a turbulent axisymmetric mixing layer. *J. Fluid Mech.* **709**, 408–444.

Duriez, T. Parezanović, V. Laurentie, J.-C. Fourment, C. Delville, J. Bonnet, J.-P. Cordier, L. Noack, B. R. Segond, M. Abel, M. W. Gautier, N. Aider, J.-L. Raibaudo, C. Cuvier, C. Stanislas, M. & Brunton, S. 2014 Closed-loop control of experimental shear layers using machine learning (invited). In *7th AIAA Flow Control Conference*, pp. 1–16. Atlanta, Georgia, USA.

Ferreira, C. 2001 Gene expression programming: A new adaptive algorithm for solving problems. *Complex Systems.* **13**, 87–129.

Fourrié, G., Keirsulck, L., Labraga, L. & Gillieron, P. 2010 Bluff-body drag reduction using a deflector. *Exp. Fluids.* **50**, 385–395.

Gardner, B. & Selig, M. 2003 Airfoil design using a genetic algorithm and an inverse method. *41st Aerospace Sciences Meeting and Exhibit*

Gautier, N. & Aider, J-L. 2013*a* Control of the separated flow downstream a backward-facing step using real-time visual feedback. *P. R. Soc. A* **469**.

Gautier, N. & Aider, J-L. 2013*b* Effects of pulsed actuation upstream a backward-facing step. *Proceedings GDR2502 Controle Des Decollements.*

Gautier, N. & Aider, J-L. 2014*a* Experimental feed-forward control of the backwards-facing step flow. *under consideration for publication at J. Fluid Mech. Preprint available at arXiv*

Gautier, N. & Aider, J-L. 2014*b* Real time, high frequency planar flow velocity measurements. *To be published in Journal of Visualization, avalaible on arXiv under "Real-time planar flow velocity measurements using an optical flow algorithm implemented on GPU"*

Gillieron, P. & Kourta, A. 2010 Aerodynamic drag reduction by vertical splitter plates. *Exp. Fluids.* **48**, 1–16.

Harik, G.R. 1997 Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms. PhD thesis, University of Michigan.

Henning, L. & King, R. 2007 Robust multivariable closed-loop control of a turbulent backward-facing step flow. *J. Aircraft* **44**.

Hervé, A., Sipp, D., Schmid, P. & Samuelides, M. 2012 A physics-based approach to flow control usingsystem identification. *J. Fluid Mech.* **702**, 26–58.

Hung, L., Parviz, M. & John, K. 1997 Direct numerical simulation of turbulent flow over a backward-facing step. *J. Fluid Mech.* **330**, 349–374.

Joseph, P., Amandolese, X. & Aider, J. L. 2012 Drag reduction on the 25 degrees slant angle ahmed reference body using pulsed jets. *Exp. Fluids.* **52** (5), 1169–1185.

Joseph, P., Amandolese, X., Edouard, C. & Aider, J.-L. 2013 Flow control using mems pulsed micro-jets on the ahmed body. *Exp. Fluids.* **54** (1), 1–12.

Kim, J. 2003 Control of turbulent boundary layers. *Phys. Fluids* **15** (5), 1093–1105.

Koza, J.R., Bennett III, F.H. & Stiffelman, O. 1999 *Genetic Programming as a Darwinian Invention Machine*, *Lect. Notes Comput. Sc.*, vol. 1598. Springer.

Le Besnerais, G. & Champagnat, F. 2005 Dense optical flow by iterative local window registration. In *IEEE Image Proc.*, pp. 137–140.

Lee, C., Kim, J., Babcock, D. & Goodman, R. 1997 Application of neural networks to turbulence control for drag reduction. *Phys. Fluids* **9** (6), 1740–1747.

Luchtenburg, D.M. 2010 Low-dimensional modelling and control of separated shear flows. PhD thesis, Berlin Institute of Technology.

Luchtenburg, D.M., Dirk, G., Gunther, M., Noack, B., Rudibert, K. & Gilead, T. 2009 A generalized mean-field model of the natural and high-frequency actuated flow around a high-lift configuration. *J. Fluid Mech.* **623**, 283–316.

M'Closkey, R.T., King, J.M., Cortelezzi, L. & Karagozian, A.R. 2002 The actively controlled jet in cross-flow. *J. Fluid Mech.* **452**, 325–335.

Milano, M. & Koumoutsakos, P. 2002*a* A clustering genetic algorithm for cylinder drag optimization. *J. Comput. Phys.* **175**, 79–107.

Milano, M. & Koumoutsakos, P. 2002*b* A clustering genetic algorithm for cylinder drag optimization. *Proc. 3rd Symp. Smart Control of Turbulence, Tokyo* pp. 107–113.

Morimoto, Kenichi, Iwamoto, Kaoru, Suzuki, Yuji & Kasagi, Nobuhide 2002 Genetic algorithm-based optimization of feedback control scheme for wall turbulence. In *Proc. 3rd Symp. Smart Control of Turbulence*, pp. 107–113.

Noack, B. R. Morzyński, M. & Tadmor, G. 2011 *Reduced-Order Modelling for Flow Control. CISM Courses and Lectures.* 528. Vienna: Springer-Verlag.

Parezanovic, V., Laurentie, J.-C., Duriez, T., Fourment, C., Delville, J., Bonnet, J.-P. Cordier, L., Noack, B. R., Segond, M., Abel, M., Shaqarin, T. & Brunton, S. L. 2014 Mixing layer manipulation experiment – from periodic forcing to machine learning closed-loop control. *Flow Turbul. Combust.* **In press**.

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.

Rechenberg, I. 1994 Evolution strategy. *Frommann-Holzboog, Stuttgart*

Semeraro, O., Bagheri, S., Brandt, L. & Henningson, D.S. 2011 Transition delay in a boundary layer flow using active control. *J. Fluid Mech.* **677**, 63–102.

Shah-Hosseini, H. 2009 The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *Int. J. Bio-inspired Comp.* **1**.

Spazzini, P.G., Luso, G., Onorato, M., Zurlo, N. & Cicca, G.M. Di 2001 Unsteady behavior of a back-facing flow. *Exp. Fluids.* **30**, 551–561.

Tadmor, G. Lehmann, O. Noack, B. R. Cordier, L. Delville, J. Bonnet, J.-P. & Morzyński, M. 2010 Reduced order models for closed-loop wake control. *Philos. T. Roy. Soc. A* **369** (1940), 1513–1524.

Toivanen, Raino AE Mäkinen Jari, Périaux, Jacques & Cloud Cedex, France 1999 Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms. *Int. J. Numer. Meth. Fluids* **30**, 149–159.

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

Zhou, J., Adrian, R.J., Balachandar, S. & Kendall, T.M. 1999 Mechanisms for generating coherent packets of hairpin vortices. *J. Fluid Mech.* **387**, 535–396.

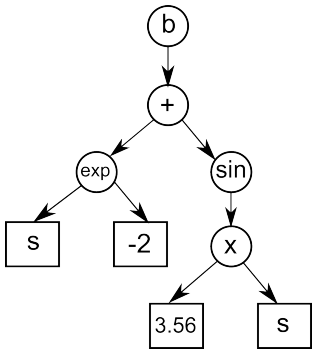


Figure 12: A typical expression tree.

Appendix A

In this appendix we provide further explanations about how control laws are translated from expression trees (§ A.1) and how the mutation and crossover operations are performed (§ A.2).

A.1. *Control laws and expression trees.*

An expression-tree can be viewed in a graphical way as a tree-like representation of the function under consideration as in figure 12 with nodes (round shapes) representing user-defined functions and leaves (square shapes) representing the constants and inputs of the function. The root of the tree (the top node) is the function output. This tree can also be described by a LISP expression. A LISP expression is easily generated and manipulated from a computational point of view. For instance the function $b(s) = \exp(s-2)+\sin(3.56*s))$ which is represented by the tree in figure 12 is represented by the LISP expression $+((\exp(+(-2s)(\sin(*3.56s))))$. The fact that the operator comes first allows to generate, evaluate and manipulate the individual with recursive functions.

A.2. *Genetic programming operations on expression trees.*

The figure 13 illustrates how the operations of mutation and crossover are performed.

The mutation operations (left) are performed by selecting a node, erasing the node and its subtree and growing a new subtree randomly. Part of the information contained in the individual is kept while new information is allowed to enter the population. The mutation operation increases the diversity and is responsible for exploring the search space with larger steps. The crossover operation (right) consists in selecting one node in each of the two individuals under consideration. The nodes and their subtrees are then exchanged. No new content is brought in but combinations of operations from good individuals (they both won a 7 contestants tournament) are tested together. The crossover is responsible for exploring the search space around individuals that are performing correctly. By adjusting the crossover and mutation probabilities, it is possible to adjust the genetic programming way of converging. A high rate of mutation will explore more of the search space while a high rate of crossover will converge faster around detected minima, whether global or local.

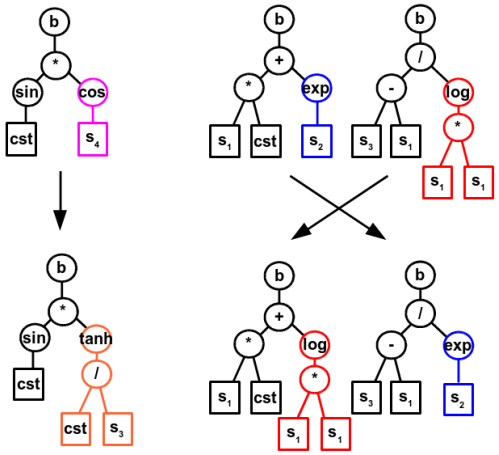


Figure 13: Left: a possible mutation of an individual. Right: a possible crossover between two individuals