A comparative study of parameter estimation techniques applied to jettisoned external stores

G. Guglieri
giorgio.guglieri@polito.it
P. Marguerettaz
paolo.marguerettaz@polito.it
G. Simioni
giulia.simioni@libero.it
Politecnico di Torino
Dipartimento di Ingegneria Meccanica e Aerospaziale
Italy

ABSTRACT

The present work evaluates the performance of different optimisation techniques on a parameter identification problem of aeronautical interest. In particular, the focus is on the classical Least Square (LS) and Maximum Likelihood (ML) methods and on the CMAES (Covariance Matrix Adaptation Evolution Strategy), DE (Differential Evolution), GA (Genetic Algorithm) and PSO (Particle Swarm Optimisation) Meta-Heuristic methods. The test problem is the reconstruction from flight test data of the aerodynamic parameters of an external store jettisoned from a helicopter. Different initial conditions and the presence of measurement noise are considered. This case is representative of a class of problems of difficult solution because of nonlinearity, ill-conditioning, multidimensionality, non separability, and fitness function dispersion. Only reference algorithm implementations found in literature are used. The performance of each algorithm are defined in terms of fitness function value, sum of absolute errors of the estimated coefficients, computational time and number of function evaluations. The results show the efficiency of CMAES in finding the best estimates with the least computational cost. Moreover, tests reveal that traditional methods depend heavily on problem characteristics and loose accuracy at the increase of the number of unknowns.

Paper No. 4044. Manuscript received 26 July 2013, revised version received 13 December 2013, accepted 10 January 2014.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

NOMENCLATURE

C_D	drag coefficient
C_L	lift coefficient
C_M	pitching-moment coefficient
D	drag in wind frame
F_X, F_Y, F_Z	aerodynamic forces in body axis
h	height (m)
J	cost function
J_{XX}, J_{YY}, J_{ZZ}	moments of inertia about the X,Y,Z axis
l_{ref}	reference length
L	lift in wind frame
m	body mass (kg)
M	pitching moment in wind frame
M_x, M_y, M_z,	aerodynamic moments in the body axis
N	number of experimental measurements
p,q,r	angular rate about the X,Y,Z axis
$\dot{p},\dot{q},\dot{r}$	angular rate derivatives about the X,Y,Z axis
q_0, q_1, q_2, q_3	quaternion parameters
S_{ref}	reference surface
t	time (s)
T	temperature (K)
u,v,w	velocity along the X,Y,Z body axes
V_∞	free-stream velocity
x	candidate solution vector in estimation algorithms

Greek symbols

α	angle-of-attack
β	sideslip angle
γ	flight path angle
$\theta, \hat{\theta}$	parameter estimate before and at convergence
ρ	air density
φ_B, θ_B, ψ_B,	installation angles of the load

Vectors and matrix

$\mathbf{g}$	gravity acceleration vector (ms^{-1})
$\mathbf{Q}$	quaternion matrix
$\mathbf{V}$	velocity vector, (ms^{-1})
$\mathbf{x}$	system state vector
$\mathbf{y}$	observation vector
$\mathbf{Z}$	set of measurement z

Subscripts and Superscripts

aer	aerodynamic
B	store
e	estimated values from estimation algorithms
H	helicopter

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

1.0 INTRODUCTION

Determining aerodynamic forces and moments of a free flying body is a complex task. The three main used techniques are analytical and computational methods, wind-tunnel testing and parameter estimation methods. Computational Fluid Dynamics (CFD), analytical methods and wind-tunnel testing are used in the design phase to generate basic trends of the aerodynamic parameters. These methods are powerful but limitations are still present. CFD requires relevant computational effort unless strong assumptions are made. On the other hand, wind-tunnel testing suffers from scaling and model-tunnel interference problems and requires very precise tuning of the measurement rig. Aerodynamic parameters estimation from flight test data is a necessary complement to the other methods to better determine aerodynamic coefficients and their derivatives.

Parameter identification from flight test data is a field of great interest which has been widely investigated in the past[1-3]. The problem is a minimisation one. The correct parameters are those which minimise the error between flight test data and the results of a model generated with a tentative solution. Normally, in flight testing, appropriate manoeuvres are carried out in order to assess specific parameters one at the time[4]. In the particular case of external store jettison, it is impossible to prescribe specific manoeuvres because the store, once released, is a free flying object. Even trying to replicate exactly the same jettison condition in subsequent flight tests is extremely difficult. Aerodynamic data can be recovered only all at the same time from a unique, often limited, set of results.

Recovering aerodynamic data from the trajectory of a free flying object is an ill-conditioned problem[5,6]. In this context, the performance of the classical minimisation techniques, such as Least Square approximation and Maximum Likelihood, rapidly decreases due to the great number of unknown variables to be simultaneously identified. When the search space is complex, with multiple local optima or with discontinuities, Meta-Heuristics algorithms can still provide valid results[7,8].

The objective of this paper is the assessment of the performance of Meta-Heuristics methods[9,10] on the problem of the aerodynamic parameters estimation of a free flying object, more precisely an external store released from a helicopter. Two classical methods are also used as reference.

The first part of this work briefly explains the theoretical principles behind the minimisation techniques and their practical implementation. The next part describes the mathematical model used for the body motion simulation and the aerodynamic model. Finally, simulations with different bodies, different initial conditions and different minimisation methods are performed and results are discussed.

2.0 OPTIMISATION ALGORITHMS[11]

Given the cost function $f:\mathbb{R}^n \rightarrow \mathbb{R}$ and the set of candidate solution $x \in \mathbb{R}^n$, the aim of a generic optimisation algorithm is to find x so that:

$$\forall y \in \mathbb{R}^n : f(x) \leq f(y) \qquad \ldots (1)$$

The global optimum x is characterised by null gradient. Classical optimisation methods minimise the cost function by adopting gradient-based techniques. These methods are very effective in most cases. Nevertheless, there are some drawbacks. Gradient evaluation has a great computational cost, can't cope with discontinuous search spaces and tends to set the solution on local optimum. Many real world problems are characterised by nonlinearity, ill-conditioning, multidimensionality and non separability. This kind of problems shows the limits of gradient-base methods. Meta-Heuristic methods don't need any gradient information and use biology inspired techniques to minimise the

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

functional *f*. They handle local optima and discontinuous and rugged cost functions. The downside is greater computational burden.

The code is written in Matlab®. In all cases reference algorithm implementations available in literature were used. Default algorithm settings are used as their tuning is beyond the scope of this work.

2.1 LS – Least square method

Least square is the simplest optimisation method and has a very wide applicabilit[1]. The inputs are a group of reference points x_i and y_i, $i = 1, ,n$ and a function $f(x,)$ which depends on the given initial points x_i and on the unknown parameters θ. LS finds the values of θ that minimise the sum of the squares of the residual $v_i = y_i - f(x_i, \theta)$. The cost function J is defined as:

$$J = \sum r_i^2 = \sum (y_i - f(x_i, \theta))^2 \quad for \quad i = 1, ..., n \qquad \ldots (2)$$

2.2 ML – Maximum Likelihood method

ML method was developed by R. Fisher between 1912 and 1922 and it was extensively applied in aircraft identification at NASA Research Centre[1,12,13].

This method can be applied to all experiments where the outcome z depends on unknown parameters θ that has to be determined. Among all the feasible parameters, ML estimate $\hat{\theta}$ are those that maximise the likelihood function. Given θ and the experiments response **Z**, the likelihood function is the probability distribution function $p[\mathbf{Z}|\theta]$.

According to its definition, $p[\mathbf{Z}|\theta]$ assigns a probability value to each element of all the possible outcomes **Z** of the random experiment.

Mathematically the problem is:

$$\hat{\theta} = \max \; p[\mathbf{Z}|\theta] \qquad \ldots (3)$$

To estimate aerodynamic parameters from flight test data, the ML method requires the knowledge of the dynamic model of the aircraft and a set **Z** of measurements $z(i)$, $i = 1, ,N$.

The considered dynamic model is:

$$\begin{aligned} \dot{x} &= f[x(t), u(t), \theta] \\ y(t) &= g[x(t), u(t), \theta] \end{aligned} \qquad \ldots (4)$$

Flight tests data are considered affected by a measurement noise v which is in turn assumed to be white, stationary and have a Gaussian distribution with zero mean. The measurement equation is:

$$z(i) = y(i) + v(i) \quad \text{for } i = 1, ..., N \qquad \ldots (5)$$

Maximising the conditional probability function is equivalent to the minimisation of the following cost function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{N} [z(i) - y(i, \theta)]^T \mathbf{R}^{-1} [z(i) - y(i, \theta)] \qquad \ldots (6)$$

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

Where **R** is the covariance matrix of measurement noise.

The main advantages are that ML handles both linear and non-linear models and it converges to solution even with process and measurement noise.

The main disadvantage of the ML method is that the minimisation relies on gradient-based algorithms. Its performance drops in presence of high number of degrees of freedom and several local optima which may lead to erroneous convergence.

2.3 GA – Genetic Algorithm

Genetic Algorithms were initially developed by John Holland in the 1970s[14]. Holland was inspired by the natural selection and the principle of the survival of the fittest. In nature, in a population of different individuals, those with the best characteristics tend to survive and generate offspring, while the weakest die. Over several generations the survival of the fittest brings to natural adaptation as the offspring are better suited to the environment than their parents.

This process is simulated in GA optimisation methodology to find the minimum of a fitness function. An initial random population of candidate solution is generated and spread on the search space. This population is encoded by a genetic representation in strings called chromosomes. The fitness function is evaluated to measure the performance of each individual. Individuals are ranked and a selection process determines which individuals will produce offspring.

The new generation is created by applying crossover and mutation operators to the parents. Crossover is a recombination technique which guarantees exchange of parts of strings, the genes, between two or more parent strings. Mutation changes parts of individuals according to probabilistic laws. This operator guarantees that all the regions of the search space could be explored and reduces the probability of convergence to local minima. The entire procedure is repeated iteratively until a termination criteria is satisfied.

2.4 DE – Differential Evolution

Differential Evolution (DE) appeared for the first time in 1995 thanks to the work of Storn and Price[15,16]. DE is a stochastic direct optimisation method similar to Genetic Algorithms. The basic DE method applies the following approach: a population of individuals, called agents, is randomly initialised and positioned within the search space. The cost function to be minimised is calculated for each individual. Arithmetic operations and genetic operators, such as crossover, mutation and selection, are applied to the population to improve its fitness until a convergence criterion is fulfilled.

DE is similar to GA with two main differences: it floating point real value encodings over bit-strings encodings and DE performs genetic operations by the use of mathematics formulae rather than GA logical operations.

The main advantage of DE is that it is a self-organising algorithm: the user must decide only few inputs and then the self-organising scheme expands or contracts autonomously the candidate solution distribution in the search space. This makes DE a robust and user-friendly optimiser.

2.5 CMAES – Covariance Matrix Adaptation Evolution Strategy

The family of the ES (Evolution Strategy) was developed in the sixties[17]. As the previous methods, ES imitates genetic reproduction and evolution laws. They are characterised by adopting real value encodings and employing only selection and mutation techniques.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

In particular, after initialisation and fitness function evaluation of λ particles, the selection process chooses the fittest $\mu \leq \lambda$ elements and mutation recombines them to create a second generation of λ children. In ES, mutation is obtained by the use of Gaussian Convolution and the mutation rate is the value of the distribution variance σ^2 which controls the Gaussian Convolution.

μ, λ and σ^2 are the main ES parameters and their values have a great influence on algorithm performances: μ controls exploitation of the search space while λ and σ^2 control its exploration. In the ES family, CMAES is the most robust search method which performs successfully in a wide variety of problems. It is particularly useful whenever the traditional methods fail: ill-conditioned functions, non-separable problems, rugged search domains with many local optima, presence of noise and in case of discontinuous search space.

Understanding the reasons of CMAES effectiveness and efficiency requires a brief theoretical discussion[18]. The ES minimisation approach is the following: the initial population is distributed in the search space, mutation adds normally distributed small values to each particle, and selection chooses the fittest particles to breed the new generation. Algorithm performance is very dependent on the mutation distribution parameters. Most ES algorithms keep them constant during the search process. CMAES, instead, applies self-adaptation of strategy parameters. The algorithm tunes automatically mutation distribution variables and updates them all along the search process. CMAES generates an initial candidate solution vector applying a multivariate normal distribution of points in the search space. The distribution is completely defined by three strategy parameters: mean, variance and covariance matrix. Until any termination criteria is satisfied, the three parameters are updated and exploited to define a new candidate solution. CMAES updates its strategy parameters implementing the following principles:

- Maximum Likelihood: mutations take place so that the likelihood of the fittest populations and successful search steps is increased. In other words, the aim is to maximise the selection probability of the best candidate solutions.
- Cumulative Step Adaptation (CSA): an evolution path is a sequence of consecutive steps done by the algorithm over consecutive generations of the search process. The knowledge of the complete past evolution path speeds up the convergence to the optimum. In the algorithm two evolution paths are implemented: one path helps in the step size control, whereas the second increases the covariance matrix adaptation towards successful directions.

2.6 PSO – Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) was developed in the mid-1990s by Kennedy and Eberhart[19]. As other meta-heuristic methods, PSO uses population of potential solutions to explore the search domain. Its unique feature is that, instead of destroying and creating new individuals at each algorithm iteration, PSO population is moved across the the search space. PSO is not modelled after natural evolution but after swarming behaviours in animals. In particular, PSO imitates fish schools or bird flocks. Individuals interact with one another and, at the same time, learn from the external environment, gradually moving into better regions of the problem space[19].

Given a search space, PSO looks for the global optimum point which minimises the fitness function exploiting a population of individuals, called particles. The information related to a particle is its position in the search domain, the fitness value in that position, its velocity, the previous best position found by that particle, the fitness value of the previous best position. The group of particles is called swarm.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

PSO operates similarly to other Meta-Heuristic methods: first there is the initialisation of the swarm which builds an initial random population and then a cycle of iterations begins in order to find the optimum point. The PSO peculiarity is that it doesn't perform any kind of selection. PSO operates on a population whose particles move on the search space. In the subsequent iterations the swarm position changes according to exchanges of information between particles. The swarm topology is the network which links each particle to the nearest ones (the neighbourhood) and allows information exchange. Thanks to the topology, each particle adjusts its trajectory towards its own previous best position and the best position of its neighbours.

3.0 SIMULATION MODEL

The mathematical model of the external load is represented by the system of equations of motion referred to the body axes. The body reference frame has the origin set at the body's centre of gravity and the *XZ* plane lies in the body plane of symmetry with *X* parallel to the fuselage reference line. *Y* axis completes the right handed reference frame. An additional inertial frame accounts for the body trajectory in space. Its origin is in the body centre of mass at $t = 0$ and its *X*-axis is oriented North, its *Y*-axis East and its *Z*-axis Down (NED). The atmospheric data is computed according to the International Standard Atmospheric (ISA) model as function of the altitude h.

3.1 Helicopter model

The helicopter model is a simple kinematic model. Body velocities and angular rates are prescribed and fixed during the entire drop manoeuvre. As in helicopter operations external stores are dropped only in emergency and usually at medium-high advancing speed or in autorotation, the main rotor wake effect is neglected. In this work, the helicopter effect is limited to the definition of the store initial drop conditions.

3.2 External body model

The external store is modelled as a 6 degrees of freedom rigid body. No additional hypothesis is made. In body axis and in vector form, the equations are:

$$\begin{aligned} \dot{v} &= \frac{\mathbf{F}}{\mathbf{m}} + \boldsymbol{\omega} \times \mathbf{v} + \mathbf{g_b} \\ \dot{\boldsymbol{\omega}} &= \mathbf{I}^{-1}(\mathbf{M} - \boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega})) \end{aligned} \quad \ldots (7)$$

where $\dot{\mathrm{V}}$ is the linear acceleration, **F** is the external force vector, $\boldsymbol{\omega}$ is the angular rate vector, $\boldsymbol{v}$ the linear speed vector, **gb** is the gravity acceleration vector in body axis, $\dot{\boldsymbol{\omega}}$ is the angular acceleration vector, **I** the inertia matrix and **M** is the external torque vector.

The **gb** vector is defined as:

$$\mathbf{g_b} = g \cdot \begin{pmatrix} \mathbf{Q}[1,3] \\ \mathbf{Q}[2,3] \\ \mathbf{Q}[3,3] \end{pmatrix} \quad \ldots (8)$$

where g is the gravity acceleration and $\mathbf{Q}[i,j]$ is the element from the i^{th} row and the j^{th} column of the **Q** rotation matrix from inertial to body frame in quaternion form:

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

$$\mathbf{Q} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_0q_3 + q_1q_2 & 2q_1q_3 - q_0q_2 \\ 2q_1q_2 - q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_0q_1 + q_2q_3 \\ 2q_1q_3 + q_0q_2 & 2q_2q_3 - q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad \ldots (9)$$

The angular variation is accounted by quaternions in order to avoid the singularities associated with Euler angles:

$$\begin{Bmatrix} \dot{q}_{s0} \\ \dot{q}_{s1} \\ \dot{q}_{s2} \\ \dot{q}_{s3} \end{Bmatrix} = \begin{bmatrix} 0 & -\omega_{s1} & -\omega_{s2} & -\omega_{s3} \\ \omega_{s1} & 0 & \omega_{s3} & -\omega_{s2} \\ \omega_{s2} & -\omega_{s3} & 0 & \omega_{s1} \\ \omega_{s3} & \omega_{s2} & -\omega_{s1} & 0 \end{bmatrix} \begin{Bmatrix} q_{s0} \\ q_{s1} \\ q_{s2} \\ q_{s3} \end{Bmatrix} \quad \ldots (10)$$

where $\mathbf{q_s}$ is the quaternion vector, $\dot{\mathbf{q}}_\mathbf{s}$ the quaternion rates vector and $\boldsymbol{\omega}_\mathbf{s}$ the store angular velocities vector. Three additional equations account for the position of the body with respect to the inertial frame.

$$X_s = \mathbf{R_s v_s} \quad \ldots (11)$$

where X_s is the position of the centre of mass of the external store, $\boldsymbol{\upsilon}_\mathbf{s}$ is the body velocity vector and $\mathbf{R_s}$ is the rotation matrix between the NED and the external store body frame.

3.3 Static axis-symmetric aerodynamic model

The aerodynamic model used in this work is an axis-symmetric static approximation. Given the external store velocities in body axes and the airflow conditions, it is possible to evaluate the Mach number M_n, the total angle-of-attack α and the geometric angle γ:

$$\begin{aligned} \alpha &= \arctan\left(\frac{\sqrt{v_y^2 + v_z^2}}{v_x}\right) \\ \gamma &= \arctan\left(\frac{v_x}{v_{yz}}\right) \end{aligned} \quad \ldots (12)$$

where $\boldsymbol{\upsilon}$ is the body velocities vector.

The aerodynamic forces and moments in wind axis are:

$$\begin{aligned} L &= \frac{1}{2} C_L(\alpha, \mathrm{M}_n) \rho V^2 S_{ref} \\ D &= \frac{1}{2} C_D(\alpha, \mathrm{M}_n) \rho V^2 S_{ref} \\ M &= \frac{1}{2} C_M(\alpha, \mathrm{M}_n) \rho V^2 S_{ref} l_{ref} \end{aligned} \quad \ldots (13)$$

where L, D, M are lift, drag and pitching moment, C_L, C_D, C_M are the aerodynamic coefficients in function of the angle-of-attack α and the Mach number M_n, ρ is the air density, V is the airspeed

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

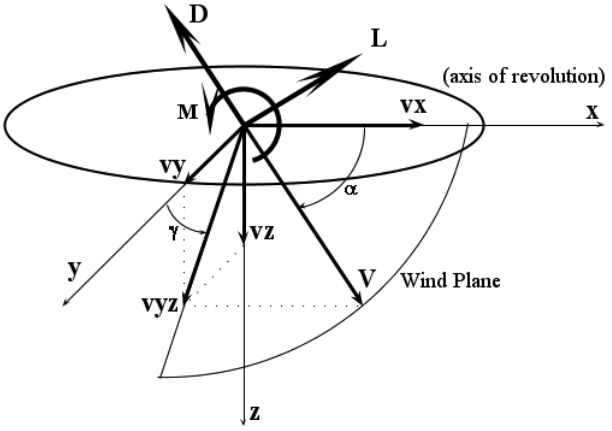


Figure 1. Body of Revolution, co-ordinate system[20].

and S_{ref} and l_{ref} are the reference area and the reference length.

The aerodynamic force and moment components in body axis are:

$$\begin{aligned} F_{xaer} &= L\mathrm{Sin}\alpha - D\mathrm{Cos}\alpha \\ F_{yaer} &= -\left(L\mathrm{Cos}\alpha - D\mathrm{Sin}\alpha\right)\mathrm{Cos}\gamma \\ F_{zaer} &= -\left(L\mathrm{Cos}\alpha - D\mathrm{Sin}\alpha\right)\mathrm{Sin}\gamma \\ M_{xaer} &= 0 \\ M_{yaer} &= M\mathrm{Sin}\gamma \\ M_{zaer} &= M\mathrm{Cos}\gamma \end{aligned} \quad \ldots (14)$$

3.4 Fitness function definition

The definition of the fitness function is, possibly, the most important part of the use of Meta-Heuristic methods[21]. Sometimes, even little differences can lead to dramatic changes in algorithm performance. As a matter of fact the fitness function is the way the minimisation algorithm 'sees' the search space. In present work, the fitness function is the sum of the errors between the reference trajectory and attitude and a simulation computed with a tentative vector of aerodynamic parameters. Much attention was paid to the definition of the fitness function. Many combinations of error definition and scaling between the error elements were tried with mixed success. In present work this fitness function is used:

$$\begin{aligned} J = \sum((x_r - x_e)^2 + (y_r - y_e)^2 + (z_r - z_e)^2 + \\ (q_{0r} - q_{0e})^2 + (q_{1r} - q_{1e})^2 + \\ (q_{2r} - q_{2e})^2 + (q_{3r} - q_{3e})^2 \end{aligned} \quad \ldots (15)$$

where J is the fitness of the tentative solution, the subscript [$_r$] refers to the reference trajectory and subscript [$_e$] refers to the estimated trajectory. Theoretically, in this application, when the

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

aerodynamic parameters vector tends to the exact solution, fitness J tends to zero. In real cases, due to the measurement noise and model uncertainties, this will not happen and an appropriate tolerance has to be defined.

4.0 COMPUTATION APPROACH

The procedure followed in this work analysis is here explained. A simulation with known aerodynamic data is performed and used as a reference. The reference trajectory is the input on which the different methods try to estimate the aerodynamic parameters. These parameters are then compared with the reference ones and errors are computed. Computational cost is also considered.

The performance of the different methods is evaluated on some realistic test cases. Simulations last 2s with a time step of 0·01s. Two general shapes with similarities to helicopter externally carried loads (fuel tanks, etc.) were identified: a cylinder and a finned streamlined body. Simulations are performed for two external bodies load conditions: heavy and light. High load is compatible with a external fuel tank completely filled, while the light condition is related to an empty tank. Due to the fact that helicopter normally releases external loads in emergency situations only, a stabilised autorotation condition is considered. A forward flight condition is added for comparison.

It has been assumed that the load axis of symmetry is parallel to the helicopter body frame X-axis. Before the drop, the external body only has a lateral offset $y = 1{\cdot}5$m with respect to helicopter body frame. X and Y axis offsets are zero. The installation angles are also zero. Both considered flight conditions are steady. Thus, the helicopter initial angular velocities p, q, r are zero. Attitude angle θ_H and sideslip angle β_H are also assumed null. The chosen dimensional and inertial characteristics of the simulated stores are listed in Tab. 1 while the initial conditions are reported in Tab. 2.

Table 1
External stores characteristics

		Circular cylinder	Finned streamlined body
Length	[m]	2·0	2·0
Diameter	[m]	0·5	0·5
Empty weight	[kg]	50	50
Full weight	[kg]	250	250
Empty J_{XX}	[kg·m^2]	1·563	1·250
Empty $J_{YY} = J_{ZZ}$	[kg·m^2]	17·448	10·625
Full J_{XX}	[kg·m^2]	7·813	6·260
Full $J_{YY} = J_{ZZ}$	[kg·m^2]	87·240	53·125
Offset	[m]	$X = 0$; Y 1·5; $Z = 0$	
Installation angles	[°]	$\varphi = 0$; $\theta = 0$; $\psi = 0$	

Table 2
Initial body velocities

	Level flight [ms^{-1}]	Autorotation [ms^{-1}]
V_{IAS}	25·72	25·72
V_X	25·72	23·70
V_Y	0	0
V_Z	0	10

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

The cylinder is an axis symmetric body, while the finned streamlined store is actually a rotational ellipsoid with four small fins. From an aerodynamical point of view, they can be both modelled as axis-symmetric bodies. The piecewise linear aerodynamic model here used needs the static lift, drag and moment coefficients in function of the total angle-of-attack $C_L(\alpha)$, $C_D(\alpha)$ and $C_M(\alpha)$. Jorgensen's method was used to create the aerodynamic database[22,23]. Aerodynamic data where further verified with other available references[24,25]. Coefficients were evaluated from zero to 180° with an interval of 18° and linearly interpolated between this values. The reference area and the reference length used to normalise the aerodynamic coefficients are the body cross section area and the body length. This model is compliant with commonly used industrial standards.

The calculated reference coefficients are reported in Tab 3 and graphically displayed in Fig. 2. The two bodies show similar drag and lift coefficient trends while moment coefficient tendency is quite different. The finned body displays un asymmetric behaviour with respect to $\alpha = 90°$ due to the presence of the fins.

Table 3
Aerodynamic coefficients – Lookup tables

	Cylinder			Finned Body		
α(°)	C_L	C_D	C_M	C_L	C_D	C_D
0	0·000	0·209	0·000	0·000	0·039	0·000
18	1·051	0·540	1·161	0·348	0·151	0·621
36	2·385	1·901	1·809	1·094	0·830	0·014
54	2·837	4·028	1·695	1·522	2·124	–1·599
72	1·876	5·837	0·951	1·113	3·447	–3·515
90	0·000	6·260	0·000	0·000	4·009	–4·869
108	–1·876	5·837	–0·951	–1·126	3·443	–5·283
126	–2·837	4·028	–1·695	–1·550	2·104	–4·748
144	–2·385	1·901	–1·809	–1·127	0·784	–3·348
162	–1·051	0·540	–1·161	–0·370	0·082	–1·537
180	0·000	0·209	0·000	0·000	0·039	0·000

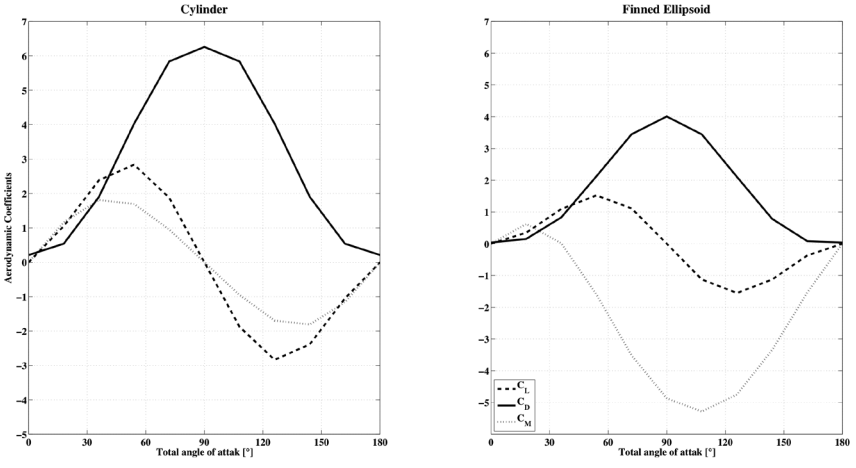


Figure 2. Cylinder and Finned Body aerodynamic coefficients.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

In real flight test measurement noise is always present. Therefore, in order to evaluate the methods performances in presence of noisy measurements, a second group of test cases was created by adding noise to the simulated reference data. This noise has a Gaussian distribution with zero mean and a variance compatible with real flight test data:
Noise is defined as:

$$noise = N(\mu; \sigma^2) \quad \ldots (16)$$

where μ and σ are:

$$Trajectory \quad \begin{cases} \mu = 0 \\ \sigma = 0 \cdot 01\ m \end{cases}$$

$$Attitude \quad \begin{cases} \mu = 0 \\ \sigma = 1^{\circ} \end{cases}$$

The estimation algorithms tested are LS, ML, CMAES, GA, DE, PSO. The tuning of the strategy parameters of the algorithms is beyond the objective of this work and default values are adopted.

The choice of the initial guess solution is very important and can greatly help the convergence of the algorithms. Unfortunately, in most practical cases, unless wind tunnel or CFD data are available, references can provide only a rough aerodynamic estimate. To account for this situation, the solution vector is initialised to zero. This is, in any case, a physical condition, equivalent to a ballistic flight in absence of atmosphere.

The availability of a rough estimate is, however, useful to define the solution search space limits. In present work the coefficients boundaries are defined as:

$$\begin{cases} -3 \leq C_L \leq 3 \\ 0 \leq C_D \leq 7 \\ 6 \leq C_M \leq 2 \end{cases}$$

All the algorithms stop when one of the following termination criteria is satisfied:

Table 4
Termination criteria

Maximum number of function evaluation	10^5
Maximum number of iterations	10^5
Tolerance on fitness function value	10^{-10}
Tolerance on estimates	10^{-10}

The outputs used to compare the algorithms are:

- Minimum fitness function value J reached
- Sum of absolute errors AE: this parameter quantifies the absolute difference between estimated coefficients and true values. AE is defined as:

$$AE = \sum_{i=1}^{N} |\mathbf{y_i} - \mathbf{z_i}| \quad \ldots (17)$$

  where $\mathbf{y}$ is the vector of n estimates and $\mathbf{z}$ is the vector of reference coefficients
- Number of fitness function evaluations FE necessary to reach convergence
- Computational time $t[s]$

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

5.0 RESULTS

The range of total angles of attack actually covered in a 2s simulation strongly depends on the initial conditions, on the aerodynamics and on the store inertia. The methodology here developed only performs the identification of the aerodynamic parameters relative to range of the angles of attack actually reached in the reference trajectory. No further measure of the identifiability level of the parameters is performed. Table 5 displays the number of unknown aerodynamic parameters in each test case.

Table 5
Search space dimensions

Shape	Test conditions	Unknowns
Cylinder	Empty load – Level flight	33
	Empty load – Autorotation	33
	Full load – Level flight	30
	Full load – Autorotation	27
Finned Body	Empty load – Level flight	12
	Empty load – Autorotation	9
	Full load – Level flight	12
	Full load – Autorotation	9

The finned body shows a much lower number of unknowns. This is due to the fact that the moment coefficient displays a stable point in the proximity of $\alpha = 40°$. The store will likely oscillate in the neighbourhood of that position. In the autorotation case, the initial angle-of-attack is even closer to that point and a further unknowns reduction is observed.

In the cylinder case the store is unstable and it starts to tumble as soon as released. It then covers the entire range of angle-of-attack ($\alpha = 0° \div 180°$). Predictably, in the heavy case the number of unknowns is somewhat smaller. The heavier store experiences the same amount of aerodynamic force as the light one, but, due to the higher inertia, it will accelerate less promptly.

In order to compare the performances of the estimation algorithms, all the methods used the same initial settings and the same set of termination criteria. The tested conditions are 16, considering the possible combination of load type, flight conditions and the eventual presence of noise.

Two main characteristics are important for a good optimiser: invariance and adaptation. An invariant algorithm is independent of the characteristics of the fitness function and of the initial conditions. Invariance guarantees the comparable performances for an entire problem class. Adaptation is complementary to invariance because the good optimiser must adapt to all kind of problems.

From the numerical point of view, the best performing algorithm is the one which converges to the minimum fitness function value and with the lowest absolute error. The fitness function measures the difference between estimated trajectories and measured data. Minimum fitness function is not enough to guarantee good performances. The absolute error AE quantifies the difference between estimated and reference aerodynamic coefficients. Therefore, the optimiser must provide both fitness function minimum and AE minimum.

Moreover, the minimum has to be found with the lowest possible number of function evaluations and the minimum possible computational time.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

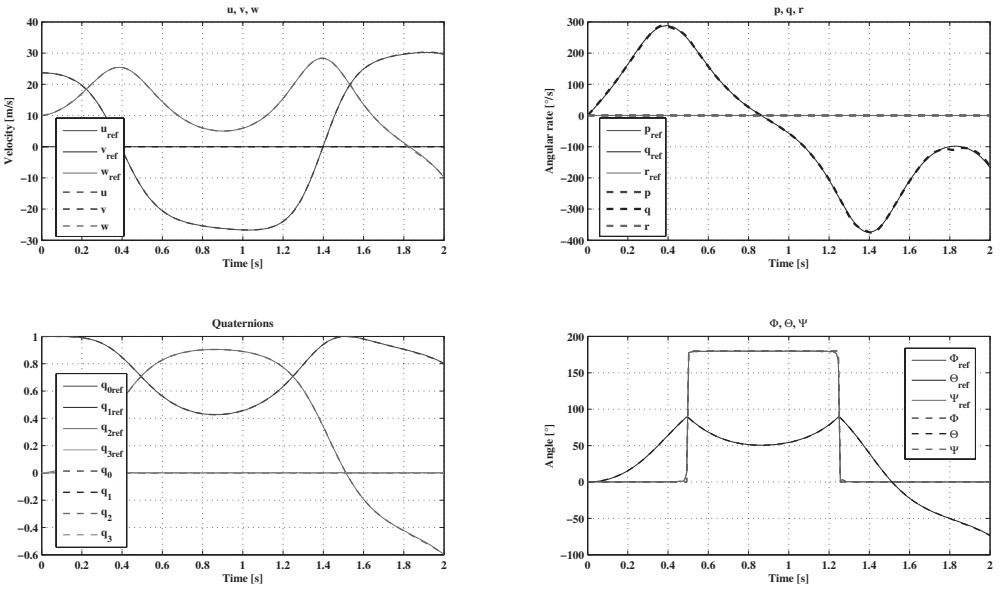


Figure 3. CMAES – Cylinder – No measurement noise – Velocities and attitude.

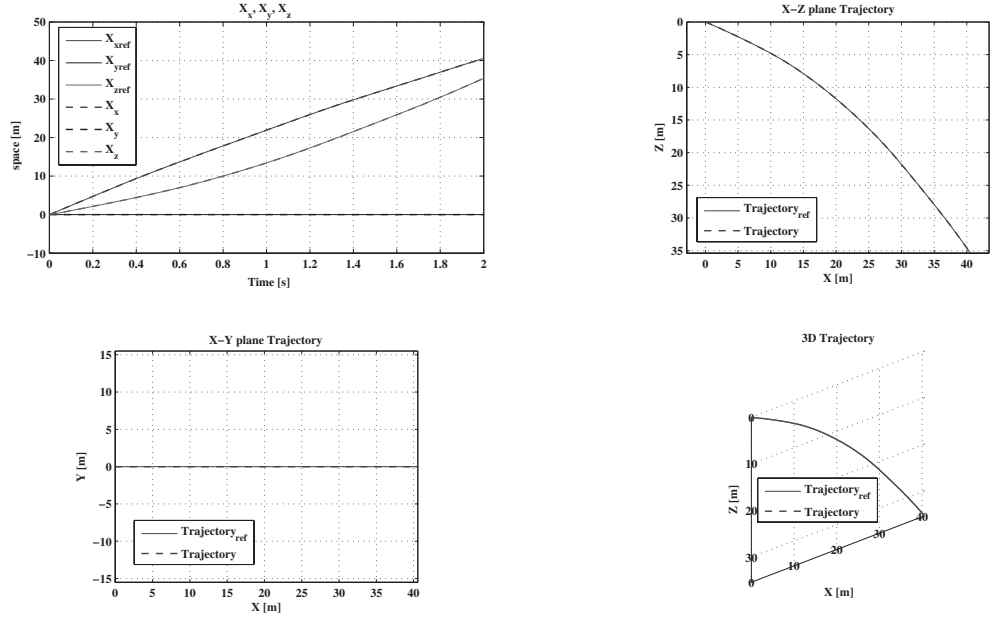


Figure 4. CMAES – Cylinder – No measurement noise – Trajectory.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

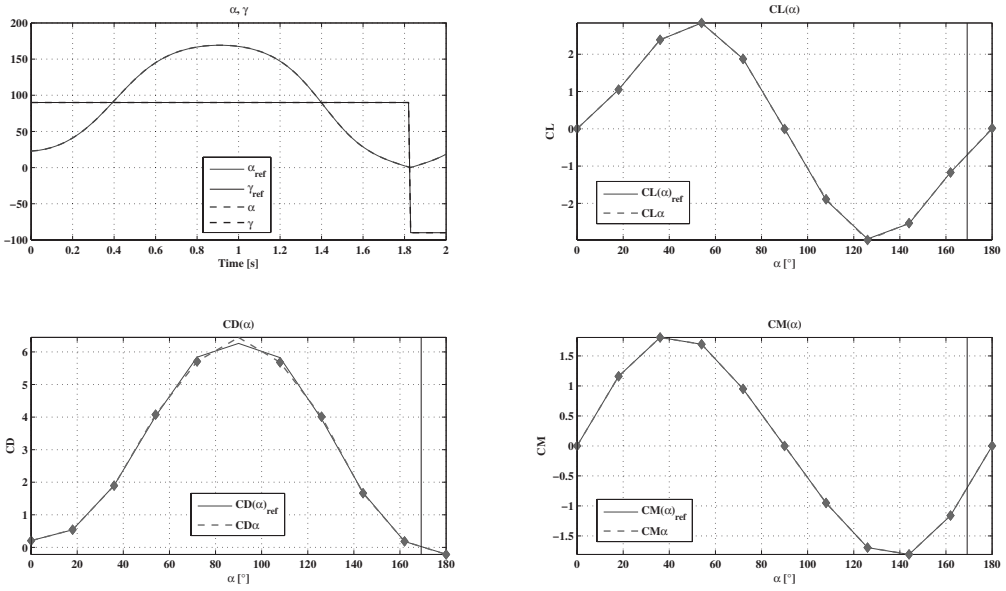


Figure 5. CMAES – Cylinder – No measurement noise – Aerodynamic angles and coefficients.

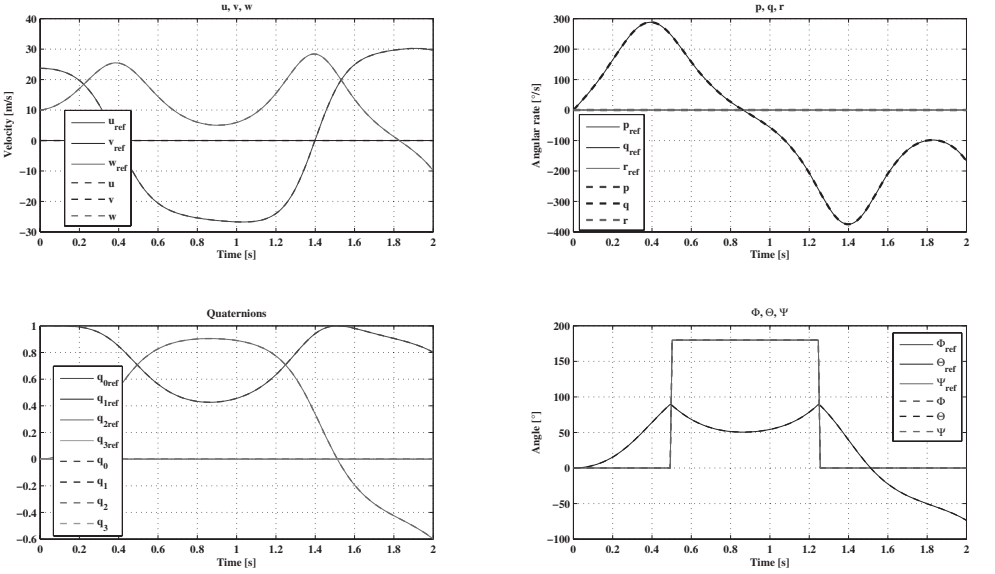


Figure 6. CMAES – Cylinder – With measurement noise – Velocities and attitude.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

5.1 Simulation results

Each simulation yields several plots showing released store linear and angular velocities, its trajectory and attitude, its aerodynamic angles and the aerodynamic parameters. Each result is plotted versus the data generated with the reference aerodynamic parameters.

For brevity only a single, but representative, case is here shown. Figures from 3 to 8 display the behaviour of the empty cylinder dropped from a helicopter in autorotation flight condition with and without the presence of measurement noise.

Figures from 3 to 5 clearly show that, in the ideal case without noise, CMAES is able to match the reference data with very good accuracy. In particular aerodynamic coefficients are almost perfectly identified.

Figures from 6 to 8 display the case with measurement noise. Identified aerodynamic coefficients are not as good as in the previous case, but they still match well the trend of the reference coefficients. Moreover, the trajectory generated with the reconstructed aerodynamics still matches very well the reference data.

5.2 Fitness function and absolute error

Simulation results are analysed by comparing the best fitness function and minimum error of the different methods for each test case at the reaching of a termination criteria. Only in few cases convergence is reached within the allowed maximum number of function evaluations.

Table 6
Finned Body – No measurement noise – Minimum fitness and absolute error

Algorithm	Empty Level flight		Empty Autorotation		Full Level flight		Full Autorotation	
	J	Error	J	Error	J	Error	J	Error
LS	4·16E+1	6·76	5·18E+2	1·55E+1	1·70E-3	2·59	8·37E-4	2·88
ML	5·07E-11	1·98E-4	1·06E+2	1·59E+1	4·42E-12	1·24E-4	2·84E-12	1·42E-4
CMAES	8·94E-14	5·14E-6	6·58E-13	1·88E-5	1·78E-13	1·27E-5	6·57E-13	4·07E-5
DE	9·03E-2	1·11E+1	4·25E+2	2·90E+1	4·55E-3	4·56	6·76E-11	4·26E-4
GA	6·71	6·09	3·51E-1	9·01	3·58E-2	3·96	9·04E-2	6·90
PSO	4·91E-2	6·20	1·53E-2	2·94	2·56E-4	1·48	1·37E-4	1·25

Table 7
Cylinder – No measurement noise – Minimum fitness and absolute error

Algorithm	Empty Level flight		Empty Autorotation		Full Level flight		Full Autorotation	
	J	Error	J	Error	J	Error	J	Error
LS	2·44	4·06E+1	5·04E+1	4·62E+1	1·79E-1	2·49E+1	1·17E-1	1·90E+1
ML	4·94E-3	1·93E+1	9·23E-2	3·16E+1	2·32E-7	6·44E-1	7·15E-7	9·72E-1
CMAES	1·38E-13	1·46E-4	3·30E-7	7·17E-1	5·66E-11	8·90E-3	5·25E-10	2·65E-2
DE	6·14E-1	5·00E+1	9·77E-1	4·24E+1	4·64E-2	3·68E+1	8·12E-2	3·21E+1
GA	1·90	4·50E+1	7·05E	3·00E+1	2·87E-02	1·7E+01	5·39E-2	1·91E+1
PSO	1·44	3·61E+1	5·63E-1	4·24E+1	1·53E-02	1·79E+01	4·87E-3	1·78E+1

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

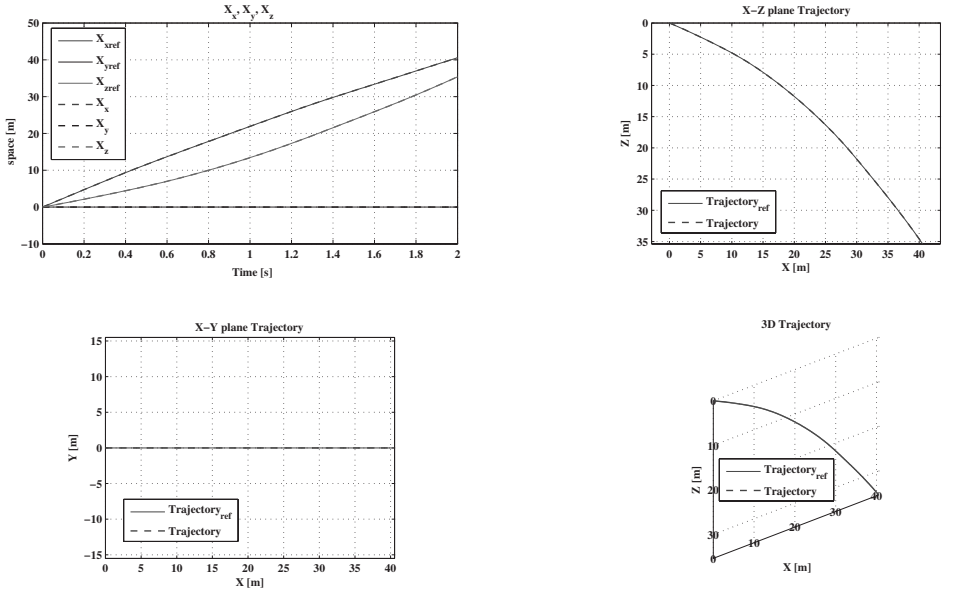


Figure 7. CMAES – Cylinder – With Measurement noise – Trajectory.

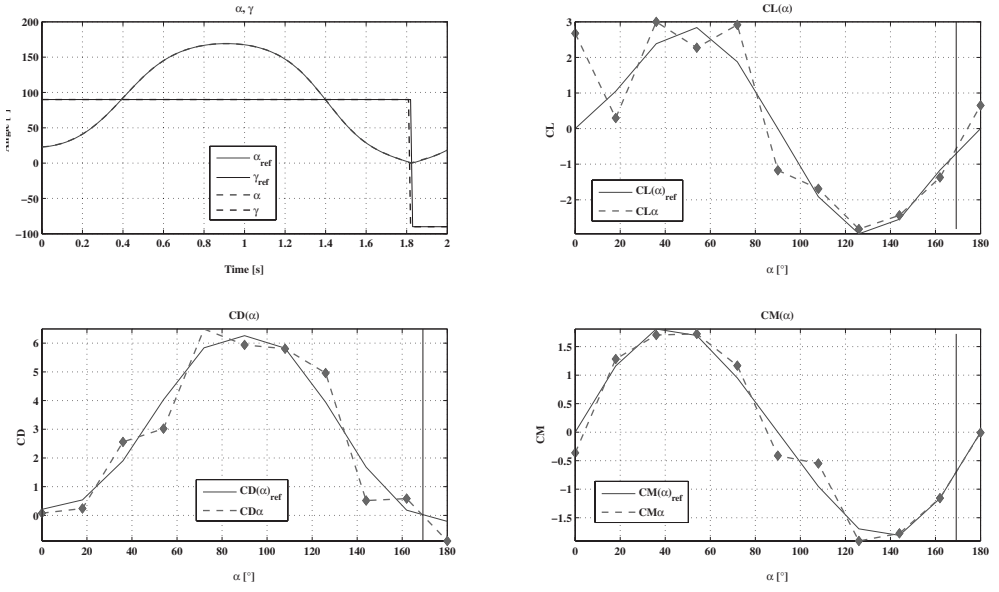


Figure 8. CMAES – Cylinder – With measurement noise – Aerodynamic angles and coefficients.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

Table 8
Finned body – With measurement noise – Minimum fitness and absolute error

	Empty Level flight		Empty Autorotation		Full Level flight		Full Autorotation	
	J	Error	J	Error	J	Error	J	Error
LS	1·68E+1	3·14	3·64E+2	9·24	1·54E-1	3·14	1·31E-1	4·15
ML	9·63E+2	8·57E+1	1·16E+3	2·70E+1	8·17E+2	1·90E+1	1·04E+3	1·45E+1
CMAES	1·50E-1	2·05	1·28E-1	2·24	1·30E-1	2·00	1·45E-1	3·96
DE	5·65E+1	3·10E+1	1·50E-1	1·82	7·65E+1	4·08E+1	1·62E-1	6·13
GA	9·89E-1	6·05	7·38E-1	7·59	1·00	6·53	2·09E-1	4·14
PSO	1·34E-1	3·36	3·07E-1	5·42	1·53E-1	·40	1·60E-1	3·36

Table 9
Cylinder – With measurement noise – Minimum fitness and absolute error

	Empty Level flight		Empty Autorotation		Full Level flight		Full Autorotation	
	J	Error	J	Error	J	Error	J	Error
LS	2·66	4·08E+1	5·29E+1	4·68E+1	2·88E-1	2·30E+1	2·14E-1	1·72E+1
ML	2·53E+3	7·57E+1	1·23E+3	9·69E+1	8·89E+2	7·78E+1	4·60E+2	4·37E+2
CMAES	1·35E-1	1·51E+1	1·45E-1	1·64E+1	1·31E-1	2·05E+1	1·40E-1	1·81E+1
DE	4·58E-1	3·39E+1	3·75E-1	4·99E+1	1·66E-1	2·94E+1	1·77E-1	2·19E+1
GA	4·03	3·95E+1	1·85E+1	5·27E+1	1·73E-1	2·24E+1	1·92E-1	2·00E+1
PSO	5·40E-1	2·88E+1	9·87E-1	3·84E+1	1·85E-1	2·34E+1	1·59E-1	1·67E+1

Tables from 6 to 9 collect the fitness function values and the sum of absolute errors of all simulations for all algorithms.

A consistent performance pattern is noticeable across all the tests: CMAES outperforms the other algorithms in most cases. In the simulations without noise, the orders of magnitude of fitness function and *AE* are respectively 10^{-14} and 10^{-12} and 10^{-6} and 10^{-2}. This excellent results worsen in presence of noise where the orders of magnitude of *J* and *AE* become respectively 10^{-1} and $10^{0} \div 10^{1}$. Anyway, the CMAES still performs better than the other algorithms.

Without measurement noise, both *J* and *AE* are very low as the algorithm is able to flawlessly identify the aerodynamic parameters along the entire angle-of-attack range reached by the store.

More interestingly, in the presence of noise, *J* and *AE* are still acceptably low. In particular *J* is one to two orders of magnitude inferior to *AE*. Because *J* is the sum of the errors on the trajectory, this fact means that, despite a sizeable error in the aerodynamic parameters identification, a simulation performed with the reconstructed aerodynamic model is still accurate.

ML seems to enter in competition with CMAES in some cases. Anyway, its good results are limited to cases with no noise. Moreover ML is very sensitive to the search space dimension and the estimate quality worsens when the number of unknowns rises. LS shares the same problems of ML but it is less sensitive to noise.

PSO performs consistently across all the test cases which makes it less sensitive to initial conditions and solution space shape. Anyway the quality of coefficients estimates is always lower than CMAES results. It is more difficult to generalise the behaviours of GA and DE. They seem very dependent on initial values and search space shape.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

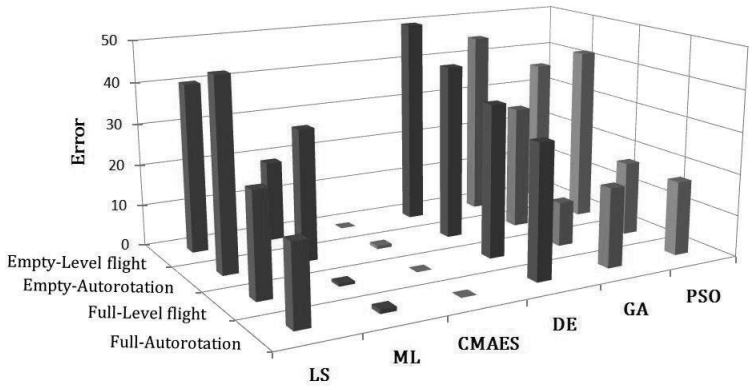


Figure 9. Absolute errors of tests on cylinder without measurement noise.

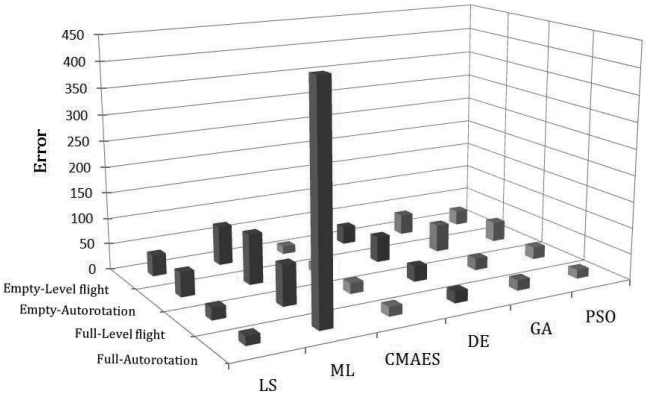


Figure 10. Absolute errors of tests on cylinder with measurement noise.

The following figures provide a visual representation of the aforementioned results. In particular, the following graphs compare the absolute errors. Figures 9 and 11 refer to simulations without noise, whereas Figs 10 and 12 refer to tests with noise measurements.

5.3 Computational time and number of function evaluations

Figures 13 and 15 show the computational time and the absolute errors of simulations on cylinder and finned body tested with and without measurement noise. Each graph is followed by the diagram of number of function evaluations versus absolute error (Figs 14 and 16). In each plot the best trade-off between absolute error and computational time is circled. For clarity classical methods are here omitted.

The following observations can be made:

- No measurement noise: CMAES always guarantees the minimum AE at the least computational time; PSO and DE are the most time consuming because they exploit all the possible function evaluations to minimise the function.
- Measurement noise: PSO competes with CMAES in providing good results in presence of noise. Figure 13 shows that PSO is, generally, more time consuming.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

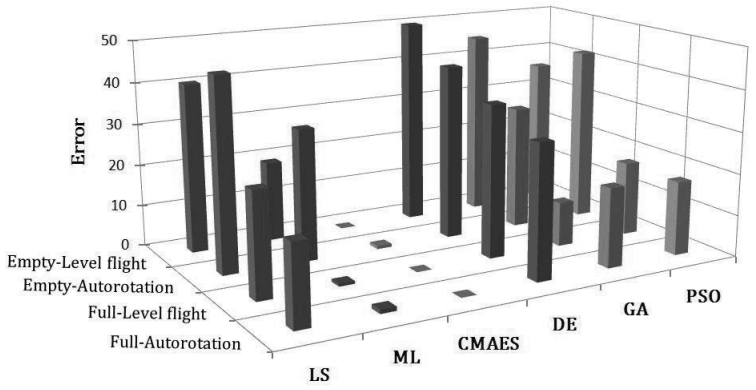


Figure 11. Absolute errors of tests on finned body without measurement noise.

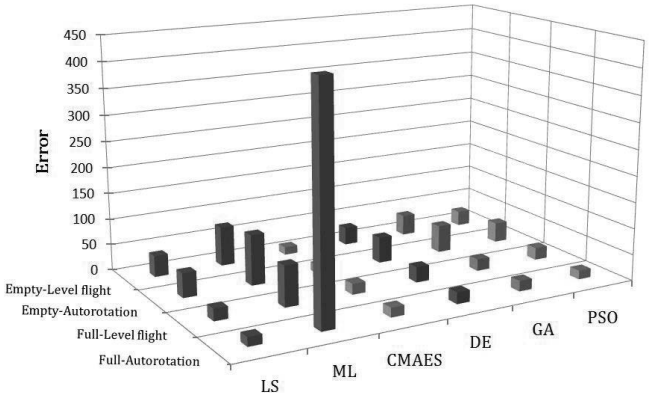


Figure 12. Absolute errors of tests on finned body with measurement noise.

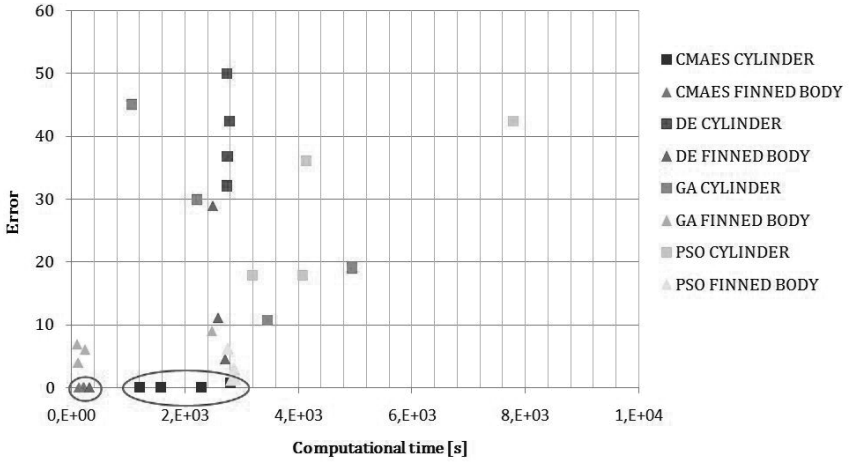


Figure 13. Computational time vs Absolute error, tests without measurement noise.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

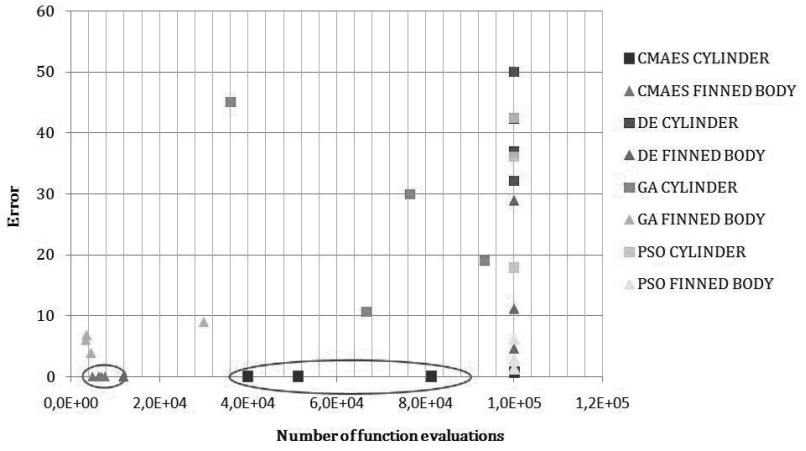


Figure 14. FE vs Absolute Error, tests without measurement noise.

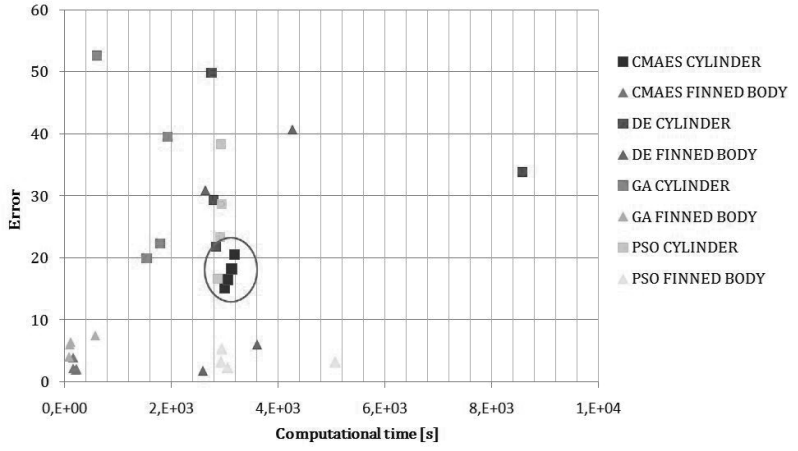


Figure 15. Computational time vs Absolute error, tests with measurement noise.

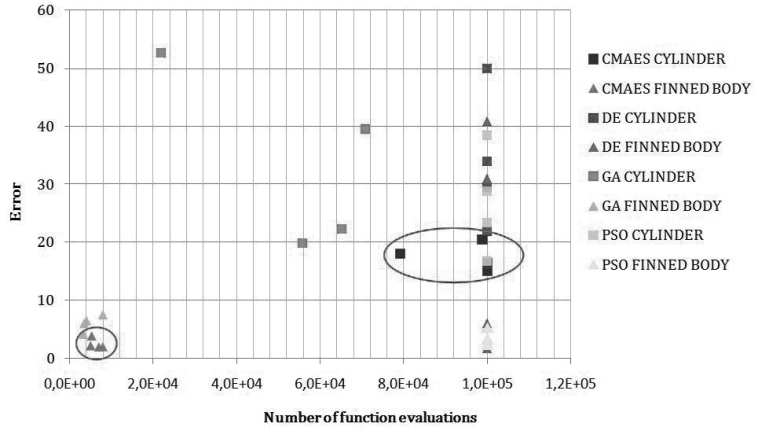


Figure 16. FE vs Absolute Error, tests without measurement noise.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

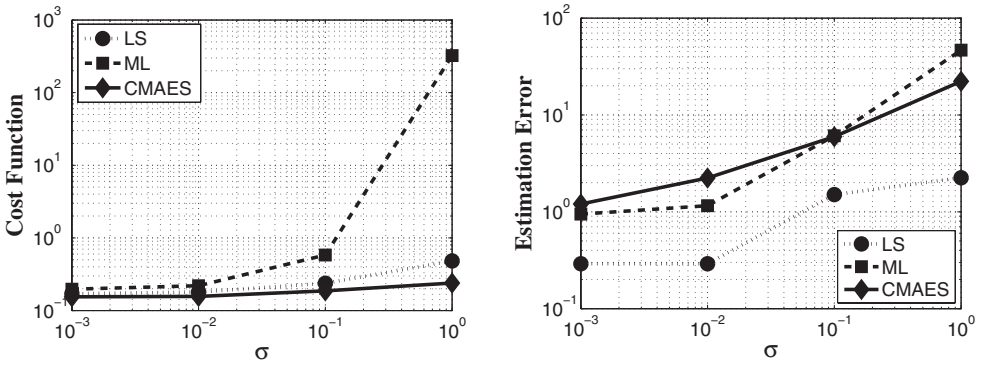


Figure 17. Comparison of the identification results with different first guess solution.

5.4 Initial guess solution effect

As explained in Section 4, all the analysis are performed starting from a null vector initial guess solution. This choice is rather extreme since the correct solution may be quite far from the initial guess. A first guess estimate closer to the exact parameters could greatly improve the algorithms convergence, in particular in the case of the classical ones. The effect of the initial guess solution is here evaluated by feeding the different algorithms (in this analysis only LS, ML and CMES) with an initial solution increasingly different from the real one. As for the reference trajectories some random noise is added to the reference aerodynamic:

$$noise = N(\mu;\sigma^2) \qquad \ldots (18)$$

where μ and σ are:

$$\begin{cases} \mu = 0 \\ \sigma = 10^{-3} \div 10^{0} \end{cases}$$

Only the light cylinder in autorotation with measurement noise case is here considered.

Table 10
Comparison of the identification results with different first guess solution

	LSE		ML		CMAES	
σ	J	Error	J	Error	J	Error
10^{-3}	0·1704	0·2911	0·1968	0·9453	0·1543	1·2030
10^{-2}	0·1794	0·2910	0·2194	1·1555	0·1564	2·2365
10^{-1}	0·2375	1·5032	0·5814	6·0863	0·1864	5·9627
10^{-0}	0·4817	2·2459	323·2926	46·6959	0·2391	22·2223

Table 10 and Figure 17 show the results of the analysis. It is clear that classical algorithms display a distinctive advantage if the guess solution is sufficiently close to the real one. Unfortunately this level of knowledge of the body aerodynamics is often unavailable. Interestingly the CMAES displays a somewhat greater error in the parameters reconstruction, but can always match the reference trajectory more accurately.

This analysis confirms the sensibility of the classical methods to local minima and verifies the robustness of CMAES.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

6.0 CONCLUSIONS

The system identification methodology is here applied to solve a parameter estimation problem. Classical estimation methods LS and ML were compared with Meta-Heuristics methods CMAES, GA, DE and PSO in order to assess their performance on a complex test case: the identification of the aerodynamic parameters of a free falling object, In the specific case an external store dropped from a helicopter.

The results of simulations support the following conclusions:

- Traditional methods ML and LS perform well only with a limited number of unknowns and with moderate measurement noise or if the initial guess solution is quite close to the actual one.
- CMAES and PSO perform consistently in each test case and show little dependency on the initial condition and search space shape. Anyway, CMAES always outperforms PSO.
- GA and DE performances vary with the function characteristics and are not competitive with CMAES in this test cases.
- The use of genetic algorithm on difficult parameter identification problems is a possible option.

REFERENCES

1. KLEIN, V. Estimation of aircraft aerodynamic parameters from flight data, *Progress in Aerospace Sciences*, **26**, pp 1–11, 1989.
2. MAINE, R.E. and ILIFF, K.W. Identification of Dynamic Systems – Applications to Aircraft Part 1: The Output Error Approach. AGARD Flight Test Techniques Series – Volume 3, AGARD – NATO Advisosy Group for Aerospace Research and Development, April 1986.
3. HAMEL, P.G. and JATEGAONKAR, R. Evolution of flight vehicle system identification, *J Aircr*, 1996, **33**, (1), pp 9–28.
4. MULDER, J.A., SRIDHAR, J.K. and BREEMAN, J.H. Identification of Dynamic Systems – Applications to Aircraft Part 2: Nonlinear Analysis and Manoeuvre Design. AGARD Flight Test Techniques Series – Volume 3, AGARD – NATO Advisosy Group for Aerospace Research and Development, April 1986.
5. BOBASHEV, S.V., MENDE, N.P., POPOV, P.A. SAKHAROV, V.A., BERDNIKOV, VIKTOROV, V.A., OSEEVA, S.I. and SADCHIKOV, G.D. Algorithm for determining the aerodynamic characteristics of a freely flying object from discrete data of ballistic experiment, *Part 1, Technical Physics*, 2009, **54**, (4), pp 504–510.
6. BOBASHEV, S.V., MENDE, N.P., POPOV, P. A., SAKHAROV, V.A., BERDNIKOV, V.A., VIKTOROV, S.I. OSEEVA, and SADCHIKOV, G.D. Algorithm for determining the aerodynamic characteristics of a freely flying object from discrete data of ballistic experiment, *Part 2, Technical Physics*, 2009, **54**, (4), pp 511–519.
7. SHI, Y., QIAN, W., WANG, Q. and HE, K. Aerodynamic parameter estimation using genetic algorithms. In IEEE Congress on Evolutionary Computation, 2006.
8. GOES, L.C.S. and VIANA, F.A.C. Life cycle and gradient based optimization applied to estimation of aircraft aerodynamic derivatives by the output error method. In 2nd International Conference on Engineering Optimization, 2010.
9. YANG, X.S. *Nature-Inspired Meta-heuristic Algorithms:* 2nd Ed, Luniver Press, Frome, UK, 2010.
10. GOLDBERG, D.E. Genetic algorithm in search, optimisation and machine learning, AddisonWesley, Reading, MA, 1989.
11. NELLES, O. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer, 2001.
12. ILIFF, K.W., MAINE, R.E. and MONTGOMERY, T.D. Important factors in the maximum likelihood analysis of flight test maneuenvers. Technical Paper 1459, NASA – National Aeronautics and Space Administration, April 1979.
13. MORELLI, E.A. and KLEIN, V. Accuracy of aerodynamic model parameters estimated from flight test data. *J Guidance, Control and Dynamics*, 1997.
14. HOLLAND, J.H. *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, MI, USA, 1975.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press

15. PRICE, K. and STORN, R.M. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, *J Global Optimization*, 1997.
16. PRICE, K. STORN, R.A. and. LAMPINEN, J.A. *Differential Evolution: A Practical Approach to Global Optimization*, Springer, 2005.
17. HANSEN, N. *The CMA evolution strategy: a comparing review*. In LOZANO, J.A. LAR- RANAGA, P. INZA, I. and BENGOETXEA, E. (Eds), *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pp 75–102. Springer, 2006.
18. HANSEN, N. *The CMA Evolution Strategy: A Tutorial*, June 2011.
19. KENNEDY, J., EBERHART, R.C. and SHI, Y. *Swarm intelligence*, Morgan Kaufmann Publishers, San Francisco, USA, 2001.
20. BARANOWSKI, L. Equations of motion of a spin-stabilized projectile for flight stability testing, *J Theoretical and Applied Mechanics*, 2013, **51**, (1), pp 235–246.
21. FANTINUTTO, G., GUGLIERI, G. and QUAGLIOTTI, F. Flight control system design and optimisation with a genetic algorithm, *Aerospace Science and Technology*, 2005, **9**, (1), pp 73–80.
22. JORGENSEN, L.H. Prediction of static and aerodynamic characteristics for slender bodies alone and with lifting surfaces to very high angles of attack. Technical Report TR R-474, NASA – National Aeronautics and Space Administration.
23. JORGENSEN, L.H. Prediction of static and aerodynamic characteristics for space-shuttle-like and other bodies at angles of attack from 0° to 180°. Technical Report TN D-6996, NASA- National Aeronautics and Space Administration.
24. HOERNER, S.F. *Fluid-Dynamics Drag*, Published by author, 1965.
25. HOERNER, S.F. *Fluid-Dynamics Lift,* Published by author, 1985.

https://doi.org/10.1017/S0001924000009398 Published online by Cambridge University Press