24th COBEM - 2017

24th ABCM International Congress of Mechanical Engineering
December 3-8, 2017, Curitiba, PR, Brazil

COBEM-2017-0932
INVERSE STOCHASTIC IDENTIFICATION OF UNMANNED PARACHUTE LAUNCHING POINT

Daniel Henrique Braz de Sousa
Aldélio Bueno Caldeira
Diogo Lopes Fernandes
Jorge Audrin Morgado de Gois
Department of Mechanical Engineering, IME, 22290-270, Rio de Janeiro, RJ, Brazil
braz2012@gmail.com
aldelio@ime.eb.br
diogo.lopes.fernandes@gmail.com
audrin@ime.eb.br

Abstract. *In the present work, an inverse problem approach is applied to identify the best launching point of unmanned parachute-cargo system in order to reach a specified target on the ground. The direct problem is based on 3 degrees of freedom model of the parachute flight dynamics, taking into account the wind profile and the airplane velocity. The inverse problem is solved by using the Particle Swarm Optimization algorithm, identifying the launching point coordinates and the airplane flight direction. The results show the feasibility of the proposed procedure to be implemented in a mission planner software, which reduces the embedded parachute control system requirements, improving the precision.*

Keywords: *inverse problem, particle swarm, dynamics, parachute.*

1. INTRODUCTION

Since its invention, the parachute was used for both recreational and military activities, being present in the two major wars, acting both in the transportation of people and material. This broad application is mainly due to the fact that the parachute is an inexpensive and simple way to reach difficult access places. However, this simplicity in the manufacture and use is not verified in flight dynamic model. The difficulty in parachute flight dynamics mathematical modeling is due to the fact that the parachute canopy not only interferes with the air around it but also deforms due to the air flow passing through it and its surroundings. It can be said, then, that the movement complexity of a parachute is basically due to three factors that determine the non-linear nature of the phenomenon: atmospheric and flight conditions, parachute shape and load geometry (**?**).

Nevertheless, even with all those difficulties, there are several parachute flight dynamics model, with 3, 6 or more degrees of freedom. These models are very important to predict the parachute landing point, mainly, in unmanned parachute systems.

In guided and unguided parachute systems, mission planner software, which is based on parachute flight dynamics models, can provide the optimum parachute launching point to improve the landing point precision (**??**).

In this work, an inverse problem approach is employed to identify the parachute launching point which minimizes the landing point error, taking into account the parachute characteristics, the payload, the atmospheric and airplane conditions.

2. MATHEMATICAL MODELING

In this modeling, described in **?**, the parachute-load system is considered a point, with mass equal to the sum of the mass of both. This simplification makes the only forces acting on the system are the drag and the weight.

For a model with 3 degrees of freedom the apparent mass matrix is a 3x3 diagonal matrix which the diagonal terms are given by (**??**):

$$\alpha_{33} = \frac{4}{3}\left(\frac{D_p}{3}\right)^3 \tag{1}$$

$$\alpha_{11} = \alpha_{22} = \frac{1}{2}\alpha_{33} = \frac{2}{3}\left(\frac{D_p}{3}\right)^3 \tag{2}$$

The apparent mass is a parameters introduced in the model to simulate the interaction effects between the air flow and the canopy (**?**). The parachute velocity can be written as:

$$\mathbf{V_u} = \mathbf{V_a} + \mathbf{W_u} \tag{3}$$

Where $\mathbf{V_u}$ is the velocity of the parachute system represented in the ground frame, $\mathbf{V_a}$ is the flow velocity represented in the body reference frame, and $\mathbf{W_u}$ is the wind velocity. Newton's 2nd Law establishes:

$$\mathbf{M_m}\dot{\mathbf{V}}_\mathbf{u} = \mathbf{F^{aero}} + \mathbf{F^{gravity}} \tag{4}$$

Where $\mathbf{M_m}$ is the mass matrix of the system considering the apparent masses, $\mathbf{F^{aero}}$ is the aerodynamic force, $\mathbf{F^{gravity}}$ is the gravitational force and $\dot{\mathbf{V}}_\mathbf{u}$ is the parachute system acceleration represented in the ground frame. The terms $\mathbf{M_m}$, $\mathbf{F^{aero}}$ and $\mathbf{F^{gravity}}$ are shown below:

$$\mathbf{M_m} = \begin{bmatrix} \alpha_{11} & 0 & 0 \\ 0 & \alpha_{22} & 0 \\ 0 & 0 & \alpha_{33} \end{bmatrix} \tag{5}$$

$$\mathbf{F^{aero}} = -\frac{QC_d(\alpha_{sp})S_0}{\|\mathbf{V_a}\|}\mathbf{V_a} \tag{6}$$

$$\mathbf{F^{gravity}} = \begin{bmatrix} 0 \\ 0 \\ (m+\alpha_{33})g \end{bmatrix} \tag{7}$$

Where the canopy circular area S_0 and the aerodynamic pressure Q are given by:

$$S_0 = \frac{\pi D_p^2}{2} \tag{8}$$

$$Q = \frac{\rho\|\mathbf{V_a}\|^2}{4} \tag{9}$$

So, we have:

$$\dot{\mathbf{V}}_\mathbf{u} = \begin{bmatrix} m+\alpha_{11} & 0 & 0 \\ 0 & m+\alpha_{22} & 0 \\ 0 & 0 & m+\alpha_{33} \end{bmatrix}^{-1} \left(-\frac{QC_d(\alpha_{sp})S_0}{\|\mathbf{V_a}\|}\mathbf{V_a} + \begin{bmatrix} 0 \\ 0 \\ (m+\alpha_{33})g \end{bmatrix} \right) \tag{10}$$

Thus, by integrating Eq.(**??**) it is possible to obtain the variation of the speed and position of the parachute with time. The functional dependence of the drag coefficient $C_d(\alpha_{sp})$ and the spatial angle of attack (α_{sp}), according to **?**, is represented in Fig. **??**.

The spatial angle of attack, which is shown in FIG. **??**, is calculated from the flow velocity vector $\mathbf{V_a}$ in relation to the body reference frame, which is parallel to the inertial reference frame.

The flow velocity vector $\mathbf{V_a}(u_a, v_a, w_a)$ is calculated as follows:

$$\mathbf{V_a} = \mathbf{V_u} - \mathbf{W_u} \tag{11}$$

Where $\mathbf{W_u}$ is the wind velocity vector in the inertial frame.

Thus, the spatial angle of attack is calculated as follows:

$$\alpha_{sp} = arc\,cos\left(\frac{w_a}{\sqrt{u_a^2 + v_a^2 + w_a^2}}\right) \tag{12}$$

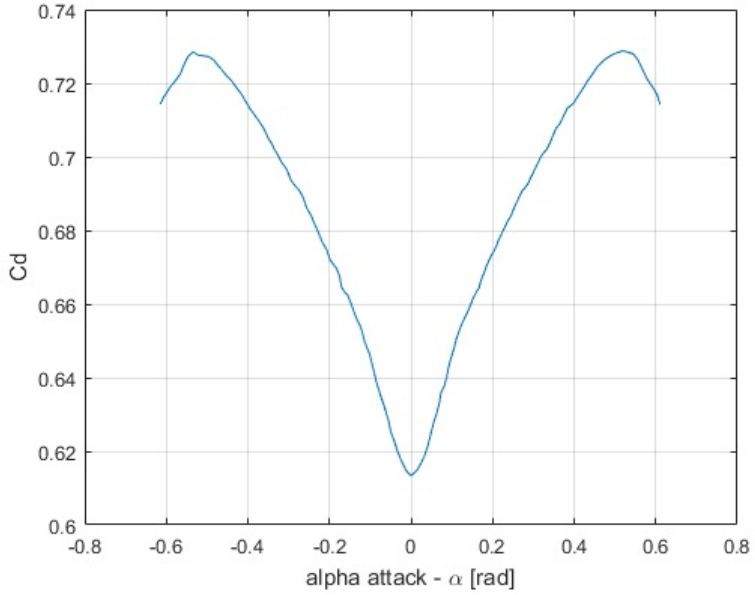

Figure 1. C_d as a function of angle of attack α_{sp}

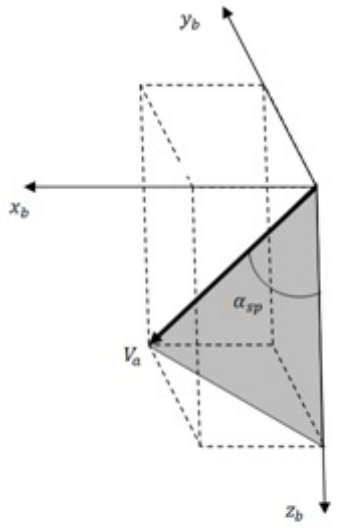

Figure 2. Flow velocity represented in the body reference frame

3. INVERSE PROBLEM

In direct problems, the objective is to determine the effects of a physical phenomenon based on its causes. Inverse problems can be defined as problems that have as objective to determine the causes (initial conditions, contour or properties of the material), based on experimental measures of the effects or computational simulations of the physical phenomenon under study.

Because it is difficult to guarantee the existence and uniqueness of the solution of an inverse problem, they are classified as ill-posed problems.

Then, in order to transpose the difficulties due to the problem be ill-posed, in general, an inverse problem is reformulated as a well-posed problem turning it into an optimization problem of the value of an objective function.

Thus, to solve an inverse problem through an optimization problem we must initially choose the objective function to be optimized (**??**).

In addition to the choice of function, it is also necessary to apply constraint relations on the variables of the objective

function, in the form of equations and inequalities.

3.1 Particle Swarm Algorithm

In 1995, James Kennedy and Russell Eberhart developed the Particle Swarm Optimization (PSO) algorithm to solve problems in the continuous domain (**?**). This algorithm is based on a theory of socio-cognitive learning. A summary of the principles that govern this learning process, according to **?** is:

- Evaluate: the ability of individuals to sense the environment to estimate their own behavior.
- Compare: take into consideration the experiences of neighbors in the decision-making process.
- Imitate: the act of imitation is the center of human social organization, being responsible for the acquisition and maintenance of mental abilities.

Initially, two variations for the algorithm were proposed by Kennedy and Eberhart: the best global version (*gbest*), where one particle is capable of influencing all others, and the local best version (*lbest*), where a particle only suffers Influence of particles that are in a particular neighborhood. The most common, simplest and most used algorithm is gbest, which was used in this work.

In order to take into account the individual and collective experience in particle evolution, it is necessary to store the best combination of parameters that the particle already has in its history (p_i) and, in the gbest version, the best combination of parameters already obtained by any particle At any time (p_b). Thus, for the calculation of the velocity at time $t+1$ the following iterative equation is used (**?**):

$$v_i(t+1) = v_i(t) + \varphi_1\beta_1[p_i - x_i(t)] + \varphi_2\beta_2[p_b - x_i(t)] \tag{13}$$

And the position of the particle is updated using the following equation:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{14}$$

Where φ_1 and φ_2 are the cognitive and social constants that are responsible for a weighting between the influence of these two parameters; β_1 and β_2 are random numbers uniformly distributed between 0 and 1; p_i is the best position already occupied by each particle and p_b is the best value obtained in all iterations across the population.

Figure **??** explains the basic logic of model operation.

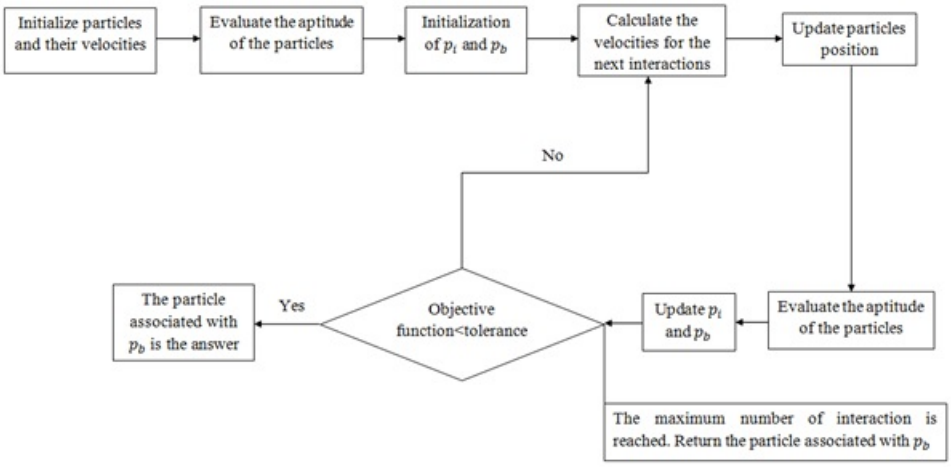

Figure 3. Flowchart of the Particle Swarm Algorithm

4. INVERSE PROBLEM SOLUTION

With the particle swarm optimization algorithm and the parachute model implemented, the inverse problem associated with the location of a launch point can be solved so that the load lands in to the desired point. Thus, the problem to be solved is to determine a launch point so that load lands in a desired point, with the launch height and the launch speed already set.

The problem has three free parameters, which are the launch point coordinates (x, y) and the launch direction (θ). The simulation was done using MATLAB because of the program's functionalities and the fact that the algorithms do not demand a great computational work.

4.1 Implementation of the parachute modeling

Dynamic model of the parachute, in the optimization algorithm, is responsible for relating the input parameters, which are being optimized, to their landing point.

The graphic representing the variation of the aerodynamic coefficients with the spatial angle of attack is shown in FIG **??**.

Drag force and apparent mass factors are dependent on the specific mass of the air. Thus, as specific mass varies with temperature and height, the "atmosisa" function present in MATLAB is used, which returns the pressure, temperature, air density and speed of sound as outputs from the height input. This function uses the "International Standard Atmosphere" to calculate the output data.

4.2 Implementation of the particle swarm algorithm

The algorithm implemented in MATLAB follows exactly the characteristics of the algorithm explained in the description of the model. Thus, the constants used were:

Table 1. Particle Swarm Algorithm Parameters

Parameters	Data
Cognitive constant (φ_1)	1.5
Social constant (φ_2)	1.5
Number of particles (n)	20
Number of parameters (p)	3
Tolerance	$1.0m$

The above tolerance defines that a given set of parameters that is related to a landing point that is less than 1 m from the desired landing point satisfies the solution of the problem and can be considered as an optimal solution.

For the search space definition, a square with a side length of $2000m$, centered at the desired point of impact and with the sides aligned with the coordinate axes is defined and within which the x and y coordinates of the launch point of each particle are initialized . Initialization of the throwing direction is done randomly, with values within the range of 0 to 2π relative to the x-axis.

5. RESULTS

5.1 Results of the parachute modeling

In order to evaluate the implementation of the three degree of freedom parachute model, we compared the results obtained with the three degrees of freedom modeling presented in **?**.

In **?**, a real launch was executed whose initial conditions are exposed in TAB **??** and were also used to generate a simulation in 3 degrees of freedom. The wind profile used is shown in Fig. **??**, these graphs were obtained through an approximation of the wind graphs present in **?**.

Table 2. Simulation input data performed in **?**

Parameters	Data
Launching Point	X=1209ft, Y=-606,7ft, Z=8213ft
Launching Velocity	V_x=21,73ft/s, V_y=-3,471ft/s, V_z=-32,85ft/s
Mass	1061,5kg
Apparent mass	α_{11}=236kg, α_{22}=236kg, α_{33}=472kg
Parachute Diameter	19,5m

Figure **??** shows a comparison between the trajectories described by the parachute. The curves in blue represent the simulation made in (**?**) and the ones in red represent the curve generated by the implemented model. The difference

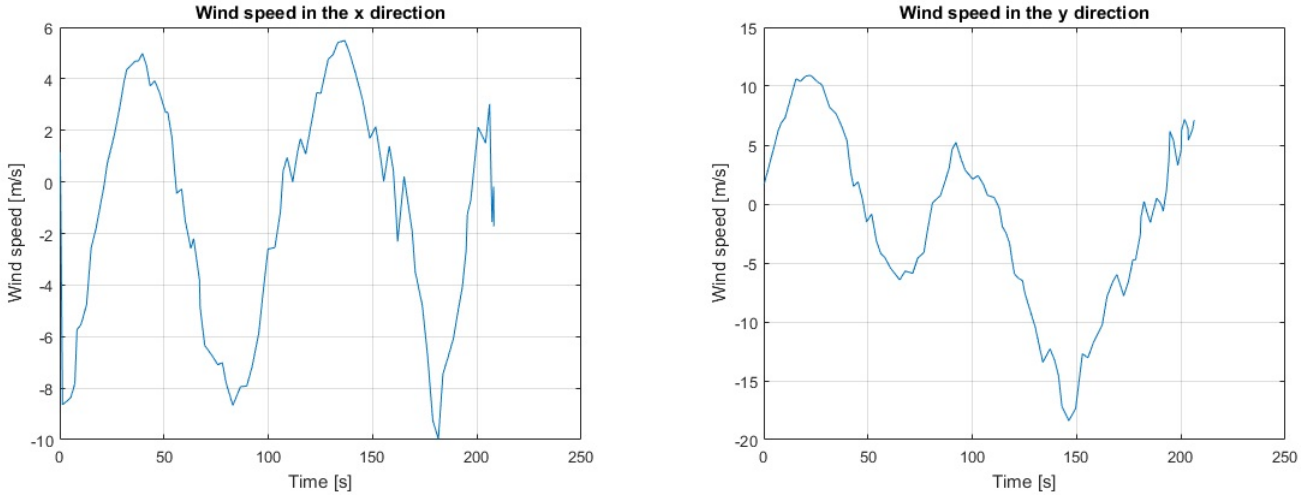

Figure 4. Wind profile used

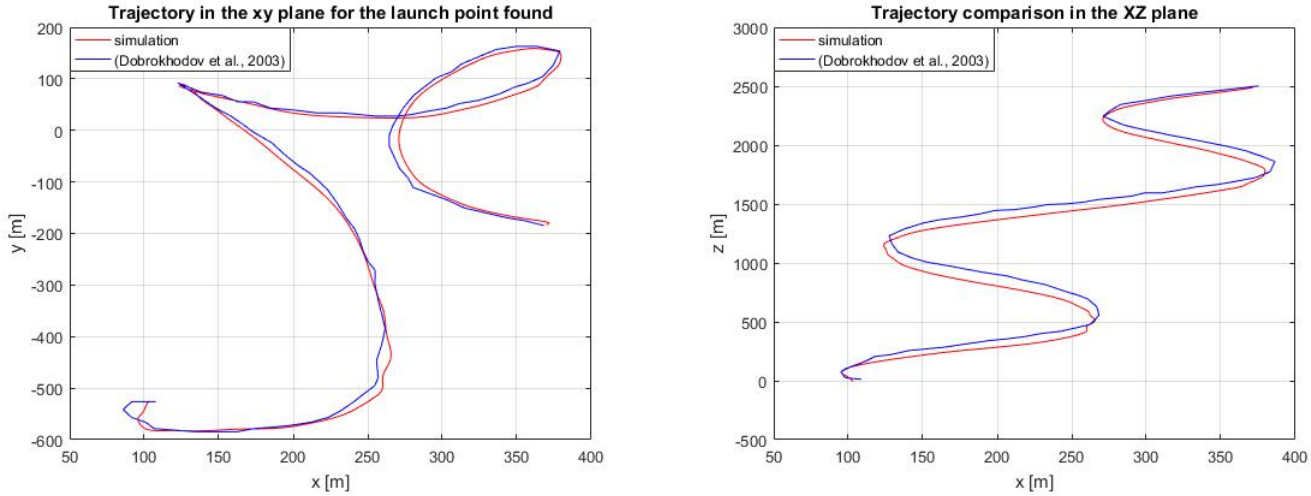

Figure 5. Comparison of the trajectories in the xy and xz planes

between the two curves in both graphs can be explained by the approximation used for the wind profile, since there was little variation between the two graphs.

5.2 Results of the particle swarm algorithm

For the simulation of the solution of the inverse problem, the wind profile used in item 5.1 was used in the parachute model. The data of launch height, apparent masses, parachute diameter and mass presented in table **??** were also used.

For the initial launch speed, the value of $36.5m/s$ was used, which is the velocity of launching of loads used in the Brazilian Army when using the C-130 aircraft for this purpose.

It should be noted that the data in Tab. **??** regarding the parachute are from a cargo parachute G-12 used for cargo launching by the Brazilian Army, whose specifications can be found in G-12 parachute datasheet.

As previously described, the landing point is the origin of the xy plane, i.e. the desired landing point is $(0, 0)$.

Figure **??** shows the initial distribution of the particles in the search space in one of the simulations. Fig. **??** and Fig. **??** show, respectively, the particle distribution after 10 interactions and the final distribution for the same simulation.

Analyzing Figures **??**, **??** and **??** it is possible to perceive the movement of the particles in the search space in order to converge to solve the problem.

Figure **??** shows the variation of the objective function with the number of iterations, i.e. the distance between the landing point of the best particle and the landing point of interest, in each interaction.

Observing Figure **??** we can see the convergence of the method to minimize the objective function which in this case is the distance to the desired landing point. Figure 10 shows the trajectory in the xy plane performed by the parachute

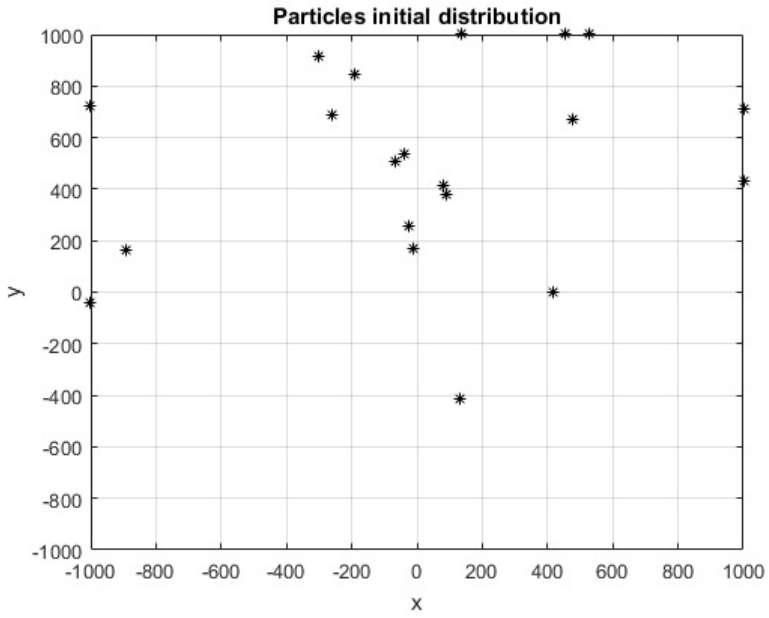

Figure 6. Initial distribution of the particles

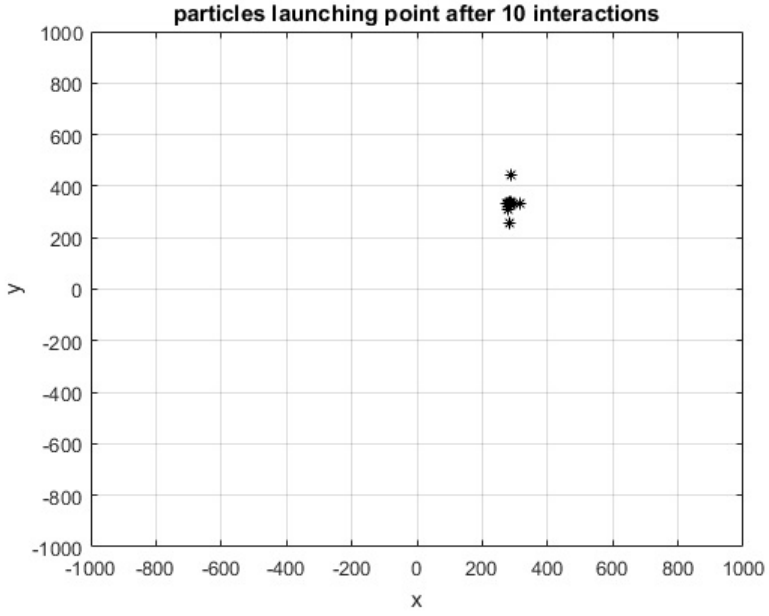

Figure 7. Particle distribution after 10 interactions

launched with the initial conditions of position and direction of launch found by the algorithm.

In Figure **??** the landing point obtained by the algorithm is shown. The distance from this point to the origin, which is the desired landing point, is $0.6134m$, showing that it is within the tolerance used in the solution method of the inverse problem.

As stated earlier, there is no single solution to this problem. Tab. **??** shows the results for five different simulations. As can be seen, each simulation shows a different result, but all within the tolerance used. This is a feature of inverse problems.

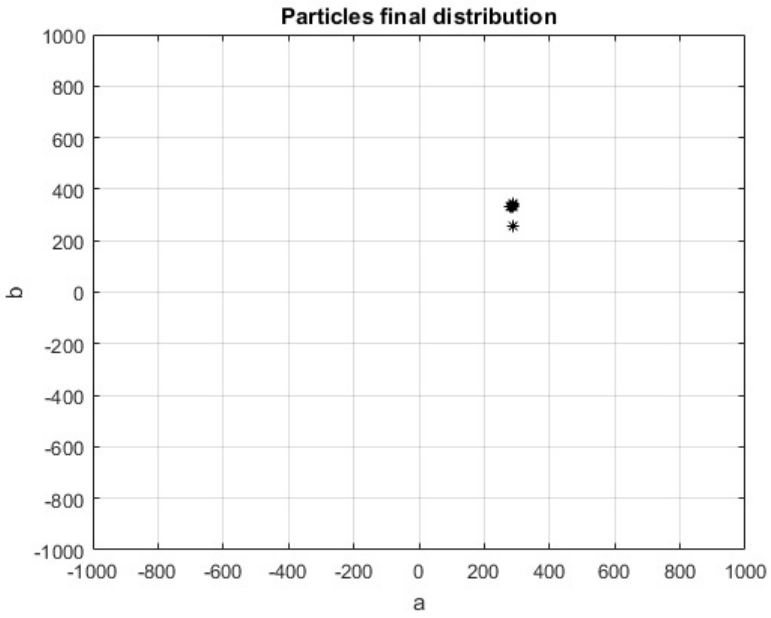

Figure 8. Final particle distribution

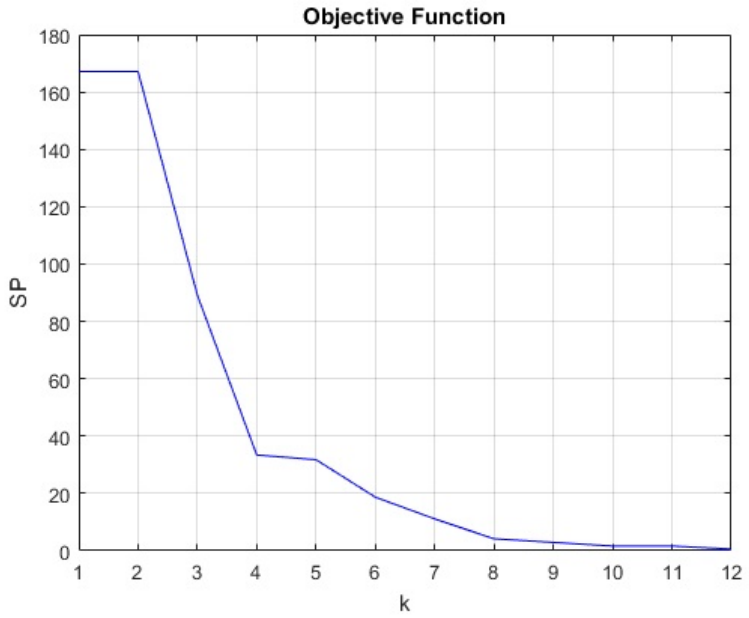

Figure 9. Variation of the objective function with the number of iterations

Table 3. Results of five different simulations

Simulation	Landing point $(x; y)$ in m	Distance to the desired point in m
1	(-0.6251; 0.1877)	0.6527
2	(0.6158; 0.1955)	0.6461
3	(0.2071; 0.2046)	0.2912
4	(-0.3768; 0.7095)	0.8034
5	(-0.5276; 0.2425)	0.5807

6. CONCLUSIONS

The simulation with 3 degrees of freedom is the least accurate, However, depending on the intended application, this model can be a good choice because it is easy to implement and does not require a great computational effort.

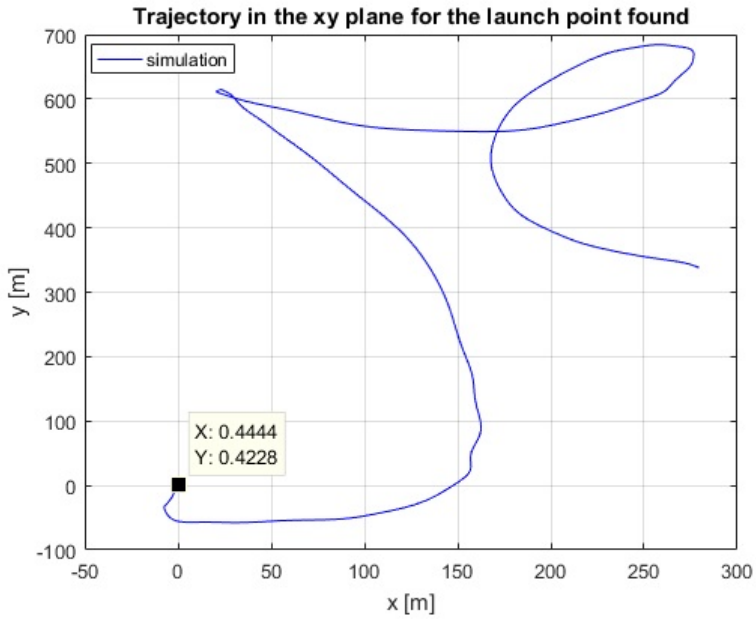

Figure 10. Parachute trajectory in the xy plane using the launch data obtained

The particle sworm algorithm for the solution of the inverse problem proved to be stable and with fast convergence, taking in average 25 interactions to converge. As it was seen, there are several solutions that satisfy the tolerance imposed for the solution, so, for each simulation the answer will be different, hardly repeating.

Since the particle swarm algorithm only uses the result of the parachute model simulation, it is possible to improve the results by using more complex and accurate parachute models. However, it is emphasized that the use of these models will increase the computational effort and, therefore, the cost-benefit of the use of more complex models must be evaluated.

Given the good results obtained with the particle swarm algorithm, it is possible to use it for the construction of a mission planning software that would have to predict the launch point given the conditions of wind profile, launch height, launching speed, type of parachute and desired landing point coordinates. This software can work in conjunction with a guiding system to be installed in the load-parachute system in order to minimize the effort of the guiding control system, by launching from an optimum point and, thereby, achieve better accuracy in relation to the landing point.

7. REFERENCES

Benney, R., MacGrath, J., McHugh, J., Meloni, A., Noetscher, G. and Tavan, S., 2007. "DoD JPADS programs overview & NATO activities". *Aerodynamic and Decelerator Systems Technology Conference and Seminar, AIAA*.

Brennen, C., 1982. "A review of added mass and fluid inertial forces". *Naval Civil Engineering Laboratoty*.

Colaço, M., Orlande, H. and Dulikravich, G., 2006. "Inverse and optimization problems in heat transfer". *Journal of the Brazilian Society of Mechanical Science and Engineering*, Vol. XXVIII, No. 1, pp. 1– 24.

Dellicker, S., Benney, R. and Brown, G., 2001. "Guidance and control for flat-circular parachutes". *Journal of Aircraft*, Vol. 38, No. 5, pp. 809–817.

Dobrokhodov, V., Yakimenko, O. and Junge, C., 2003. "Six degree of freedom model of a controlled circular parachute". *Journal of Aircraft*, Vol. 40, No. 3, pp. 482–493.

Eberhart, R. and Kennedy, J., 1995. "A new optimizer using particle swarm theory". *Sixth International Symposium on Micro Machine and Human Science*.

Kabanikhin, S., 2008. "Definitions and examples of inverse and ill-posed problems". *Journal of Inverse and Ill-Posed Problems*, Vol. 16, pp. 317–357.

Orlande, H., Colaço, M., Cotta, C., Guimarães, G. and Borges, V., 2011. "Notas em matemática aplicada". In *Problemas Inversos em Transferência de Calor*, Sociedade Brasileira de Matemática Aplicada e Computacional.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.