Research Inventy: International Journal of Engineering And Science
Vol.11, Issue 9 (September 2021), PP 61-71
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Approaching an Inverse Problem in Mechanical Engineering using Genetic Algorithms

Rafael Saraiva Campos[1] and Carlos Eduardo Leme Nóbrega[2]
[1,2] CEFET-RJ – Centro Federal de Ensino Tecnológico Celso Suckow da Fonseca, Computer Engineering Department, Campus Petrópolis, RJ, Brazil
Corresponding Author: rafael.campos@cefet-rj.br

Abstract: *Artificial Intelligence (AI) based interdisciplinary applications are becoming increasingly pervasive, with associated current and foreseeable impacts on services, production, labour, and education. Consequently, the integration of AI training into Engineering undergraduate courses is of great importance. However, each area of Engineering has its peculiarities, and those should be taken into consideration when teaching AI-related topics. Accordingly, this paper employs a didactic example widely used in undergraduate Mechanical Engineering courses as an entry point to the applied study of AI. The problem in question is determining an oblique projectile trajectory from noisy simulated experimental data taken along its route. This is an inverse problem that is here solved using Genetic Algorithms, which is an implementation of the so-called Evolutionary Learning*

Keywords: *projectile trajectory, launch point location, inverse problem, optimization, artificial intelligence*

Date of Submission: 28-08-2021 Date of Acceptance: 12-09-2021

I. INTRODUCTION

Artificial Intelligence (AI) based interdisciplinary applications are becoming increasingly pervasive, and its actual and foreseeable impacts on services, production, and labour are subject of interest (and concern) to many [Frey and Osborne 2017]. Notably, for Mechanical Engineering companies, the use of AI-based solutions will be decisive to their competitiveness. In addition, those companies will face the liability of introducing new technologies, such as autonomous vehicles and automated machines in general, in which public safety must be assured [VDMA European Office 2018]. Therefore, the integration of AI training into Mechanical Engineering undergraduate courses is of paramount importance.

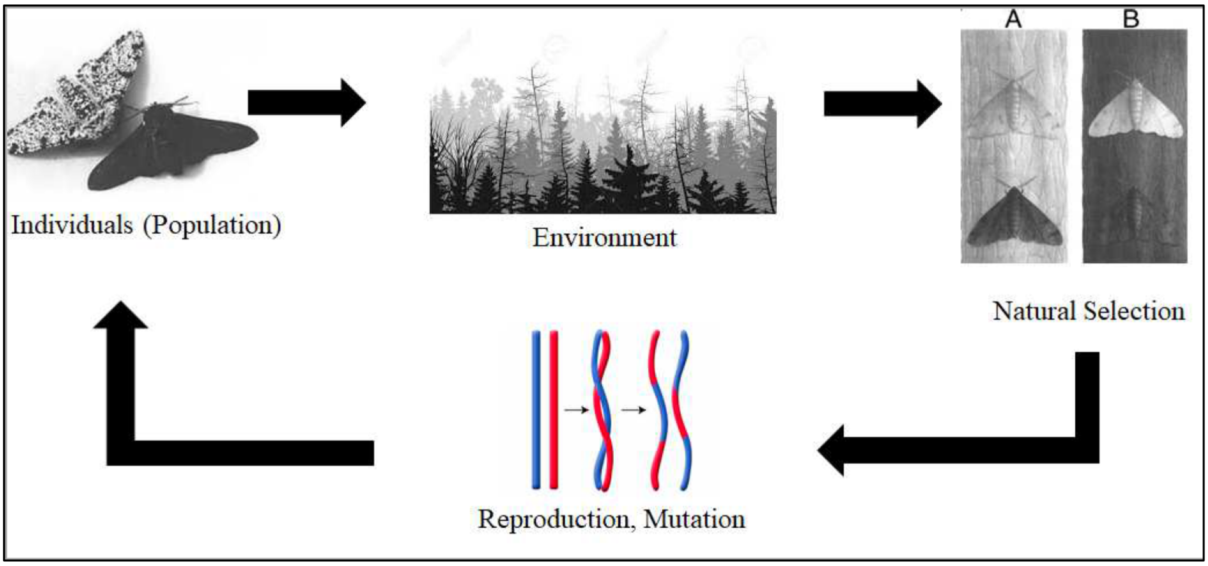
From this perspective, this paper proposes employing a didactic example from Dynamics, widely used in undergraduate Mechanical Engineering courses, as an entry point to the applied study of AI. The example in question is the projectile motion in an oblique launch, which here will be approached from an inverse problem perspective, i.e., the projectile trajectory reconstruction from experimental samples. Ordinarily, such problem is addressed in an inverse manner using iterative numerical methods, such as such as Gradient Descent and standard iterative least squares. Particularly, in [Nelson, Pachter and Musick 2005] a non-linear regression technique is applied to the ballistic trajectory determination from battlefield radar measurements, ultimately aiming at determining the location of an enemy artillery piece. Several armed forces have developed or acquired artillery hunting radar systems, such as the British Mobile Artillery Monitoring Battlefield Radar (MAMBA), capable of locating enemy artillery guns with a circular error probability (CEP) smaller than 30 meters [The British Army 2002].

Inverse problem methodology has been widely used to retrieve parameters of governing equations from noisy experimental data [Groetsch 1999], [Aster, Borches and Thurber 2013]. In the simulations presented in [Nelson, Pachter and Musick 2005], as well as in this paper, given a set of samples of the projectile positions, the inverse problem approach aims reconstructing the projectile trajectory, ultimately to retrieve a geolocation estimate of the projectile launch point. However, instead of using classical methods, this paper approaches such problem using AI, more specifically, employing Genetic Algorithms (GAs), an Evolutionary Learning technique [Golberg 1989]. Given a set of noisy measurements, GA is used here as an optimization technique to minimize the ballistic trajectory reconstruction error. Another example of use of GA as an adaptive approach to an analytically solvable kinematics problem can be found in [Cordenonsi, Baroni and Thielo 1997], where GA was employed to obtain the best variation of a missile velocity vector in order to optimize the time required by the missile to reach its target.

Artificial Intelligence (AI) encompasses a broad scope of techniques inspired by natural processes: artificial neural networks (ANN), specialist systems, fuzzy inference systems, and genetic algorithms (GAs), to name just a few. ANNs concept is based on neurons and their connections. Specialist systems aim at imitating human inference ability. Fuzzy inference systems are inspired by some of the distinctive features of human linguistic processing. GAs notion is based on natural selection and genetic evolution [Golberg 1989], [Marsland 2015]. GAs are an implementation of evolutionary learning, being founded on the Darwinian concept of species evolution, summarized in Fig. 1. A species evolution can be understood as a type of collective learning, where individuals are configured by their DNA, which is composed of chromosomes containing sequences of genes. The higher an individual's fitness, the greater its chance of surviving and reproducing. Sexual breeding and mutation produce gene variability. Sexual breeding creates individuals for the following generation from the current generation batch. Mutations are stochastic errors in DNA replication which usually result in individuals incapable of surviving, but occasionally yield an adaptive advantage.

GAs have been widely applied to optimization problems. Each candidate solution is called an individual, coded by a sequence of numbers (the chromosome and its genes, respectively). At each iteration (generation) of the algorithm, the population comprises the set of all individuals. At each generation, a mating pool is formed, which is a subset of the current population comprising the individuals selected to create new individuals by genetic recombination (crossover). Crossover is the mixing of chromosomes segments from two individuals (parents), producing new individuals (offspring) for the subsequent generation. Each individual might be selected more than once, appearing multiple times in the mating pool, or not be selected at all. The selection strategy tries to mimic Nature, where fitter individuals are more likely to find a mate. A problem-specific fitness function assesses the individuals' fitness. The higher the candidate solution fitness, the higher its probability of being selected to reproduce by exchanging genes, yielding new individuals [Lawrence 1991], [Holland 1992]. Finally, mutation introduces aleatory changes to chromosome's genes. After crossover, mutation might occur to any individual in the new population with a given probability. For example, if the genes are binary represented, then mutation can be enforced just by flipping bits. For real-valued representations, different mutation implementations exist [Houck, Joines and Kay 1995].

Fig. 1. The Darwinian concept of evolution of species: i) individuals face selection from the environment; ii) the fittest individuals are more likely to survive long enough to procreate, thereby passing on their features to the next generation; an individual's fitness is based on the environmental conditions: note that, in this picture's example, when the air pollution darkens the trees trunks, the dark moths become less noticeable and therefore less susceptible to predators attacks, which results in an increase in the number of black moths compared to white moths along the generations; iii) sexual breeding ensures geneticgenetic diversity, as a result of genetic material swapping during crossover; mutation – an aleatory error in DNA replication – increases genetic diversity; iv) This cycle continues indefinitely



The GA cycle proceeds until a halt criterion is reached, such as a limit number of generations, maximum value of the aptitude function, or computation time limit. The last generation's fittest individual is a sub-optimal solution to the problem [Pardalos et al. 1995]. The GA evolutionary cycle is summarized as follows:

i. the first generation population is stochastically generated within pre-determined limits;
ii. a fitness function assesses each individual's aptitude;

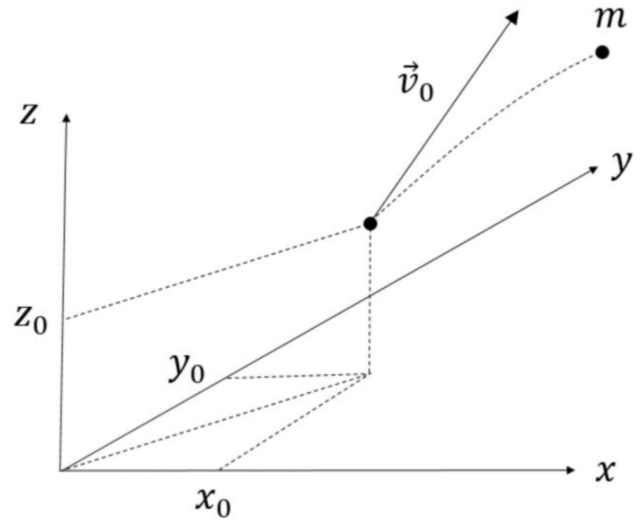
iii. stop, if the stop criterion is reached;
iv. a selection strategy creates the mating pool;
v. crossover and mutation) create new individuals for the following generation;
vi. stop if halt criterion has been meet, otherwise go back to step (ii).

II. EXPERIMENTAL PROCEDURE

2.1. Parametric Equations of Projectile Motion.

The determination of the trajectory of a projectile launched with an oblique angle (as Fig. 2 depicts) is one of the most well-known Dynamics didactic problems in undergraduate Mechanical Engineering courses [Lange and Pierrus 2010]. It can be approached with different levels of Physics complexity. In one of its simplest forms, it is assumed that a projectile of mass m is launched at instant *t=0* from location (x_0, y_0, z_0) meters with initial speed $(\dot{x}_0, \dot{y}_0, \dot{z}_0)$ meters per second, subject to Earth's gravity acceleration g (m/s^2) and constant air resistance, expressed through a drag coefficient *C*.

Fig.2. Three-dimensional oblique trajectory of a projectile of mass m launched from position $(\boldsymbol{x_0}, \boldsymbol{y_0}, \boldsymbol{z_0})$ with initial velocity $\vec{\boldsymbol{v}}_{\boldsymbol{0}}$.



Using Newton's Second Law of Dynamics, the projectile motion in space can be modelled by

$$\vec{f}(t) = m\vec{a}(t) = -C\vec{v}(t) - mg\vec{k} \quad (1)$$

where $\vec{k}$ is the versor of the z-axis (vertical upward direction), $\vec{f}$ is resulting force in Newtons, $\vec{v}$ and $\vec{a}$ are the projectile instantaneous velocity and acceleration, respectively, given by

$$\vec{a}(t) = \ddot{x}(t)\vec{\imath} + \ddot{y}(t)\vec{\jmath} + \ddot{z}(t)\vec{k} \quad (2)$$

$$\vec{v}(t) = \dot{x}(t)\vec{\imath} + \dot{y}(t)\vec{\jmath} + \dot{z}(t)\vec{k} \quad (3)$$

From the preceding equations, one can write the following set of ordinary differential equations (ODEs)

$$m\ddot{x}(t) + C\dot{x}(t) = 0 \qquad (4)$$

$$m\ddot{y}(t) + C\dot{y}(t) = 0 \qquad (5)$$

$$m\ddot{z}(t) + \frac{C}{m}\dot{z}(t) = -g \quad (6)$$

whose solution w.r.t. parameter t yields the set of parametric equations of projectile motion

$$x(t) = x_0 + \frac{m}{C}\dot{x}_0\left(1 - e^{-Ct/m}\right) \quad (7)$$

$$y(t) = y_0 + \frac{m}{C}\dot{y}_0\left(1 - e^{-Ct/m}\right) \quad (8)$$

$$z(t) = z_0 - \frac{mg}{C}t + \frac{m}{C}\left(\dot{z}_0 + \frac{mg}{C}\right)\left(1 - e^{-Ct/m}\right) \quad (9)$$

from which one obtains

$$\dot{x}(t) = \dot{x}_0 e^{-Ct/m} \quad (10)$$

$$\dot{y}(t) = \dot{y}_0 e^{-Ct/m} \quad (11)$$

$$\dot{z}(t) = -\frac{mg}{C} + \frac{m}{C}\left(\dot{z}_0 + \frac{mg}{C}\right)e^{-Ct/m} \quad (12)$$

2.2. The Inverse Problem

To simulate the acquisition of corrupted experimental data, a total of I samples of the projectile position along the route shall be taken, inserting an additive Gaussian noise with standard deviation σ meters around the true position as follows

$$\hat{x}_j = \mathcal{N}\left(x(t_j), \sigma\right) = x(t_j) + \mathcal{N}(0, \sigma) \quad (13)$$

$$\hat{y}_j = \mathcal{N}\left(y(t_j), \sigma\right) = y(t_j) + \mathcal{N}(0, \sigma) \quad (14)$$

$$\hat{z}_j = \mathcal{N}\left(z(t_j), \sigma\right) = z(t_j) + \mathcal{N}(0, \sigma) \quad (15)$$

where t_j is the time when the jth sample was taken, j = 1,…, I, and $\mathcal{N}(\mu, \sigma)$ indicates a random value drawn from a Normal distribution with average μ and standard deviation σ.

In the proposed problem, C, g, and m are known constants. The initial velocity vector given by $(\dot{x}_0, \dot{y}_0, \dot{z}_0)$ is also assumed to be known. The inverse problem then aims at solving

$$\min_{(x_0, y_0, z_0)} \sum_{j=1}^{I} \left\|\vec{\delta}_j\right\|^2 \quad (16)$$

where

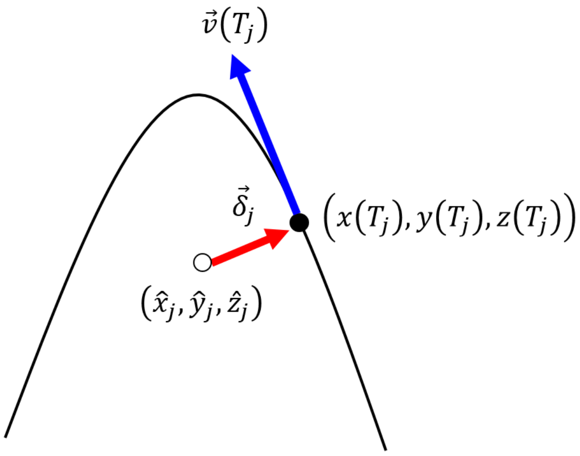
$$\vec{\delta}_j = \left(x(T_j) - \hat{x}_j\right)\vec{\imath} + \left(y(T_j) - \hat{y}_j\right)\vec{\jmath} + \left(z(T_j) - z_j\right)\vec{k} \quad (17)$$

is the vector whose norm yields the distance between the jth sample and the estimated projectile trajectory, as Fig. 3 indicates. Note that T_j is not the time when the jth sample was taken (given by t_j), but rather the value of the parameter t that satisfies

$$\vec{v}(T_j) \cdot \vec{\delta}_j = 0 \quad (18)$$

where $\vec{v}(T_j)$ – defined by Equation (3) – is tangent to the projectile trajectory at point $\left(x(T_j), y(T_j), z(T_j)\right)$.

Fig.3. The Euclidean distance between the jth sample $(\hat{x}_j, \hat{y}_j, \hat{z}_j)$ and the GA reconstructed trajectory is given by the norm $\|\vec{\delta}_j\|$. Note that $\vec{v}(T_j) \perp \vec{\delta}_j$ for any $j = 1, \cdots, I$.

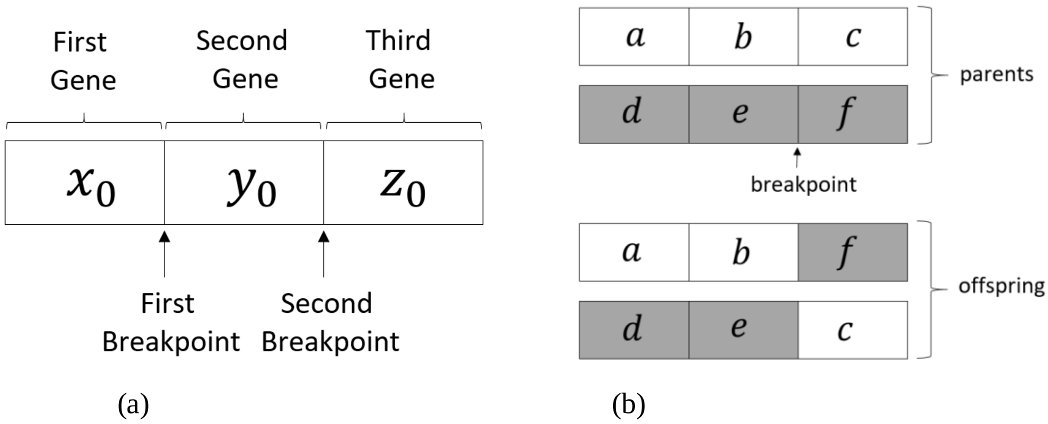


2.3. Reconstructing the Trajectory and Estimating the Launch Point using a GA

In this example, each individual of the population represents a location in the three-dimensional space (i.e., the origin of the projectile trajectory). Therefore, each individual's chromosome is a three-number sequence,

as Fig. 4a indicates. This simulation uses a population size of $N = 100$ individuals per generation. The initial population is randomly created, spreading individuals within a pre-determined region of space - a cube of side 1000 meters and centred at the origin.

Fig. 4. (a) Chromosome with three genes and two possible breakpoints; (b) Simple crossover with one breakpoint.



The fitness of each individual is used by the selection function to compose the mating pool. The fittest individuals have higher probability of being selected for reproduction (crossover), and thereby passing on their characteristics for the new generation. The aptitude is calculated by

$$S(x_0, y_0, z_0) = \frac{k}{\sum_{j=1}^{I} \|\vec{\delta}_j\|^2} \quad (19)$$

where $k = 1000$ is an empirically defined parameter. Note that the fitness function is fed with the values stored in the chromosome. Note also that it uses the inverse of the sum of the squared distances between the sample points and the reconstructed trajectory (defined as a function of x_0, y_0 and z_0), i.e., the lower the sum, the higher the individual fitness. The values of $\|\vec{\delta}_j\|$, where $j = 1, \ldots, I$, for a given individual are required to estimate its fitness. To obtain each $\|\vec{\delta}_j\|$ as indicated by Equation (17), the corresponding T_j must be calculated first, as defined by Equation (18), which is solved numerically using an algorithm that combines bisection, secant, and inverse quadratic interpolation methods, as defined in [Brent 1973], [Forsythe, Malcolm and Moler 1976]. This procedure is readily available for numerical computation through the **fzero** function in MATLAB [Kaisare 2017].

The selection strategy adopted in this example is the so-called normalized geometric selection [Houck, Joines and Kay 1995], where the population is ranked in descending order of aptitude and the probability of selecting the *i*th individual is given by

$$P(i) = \frac{q(1-q)^{i-1}}{1-(1-q)^N} \quad (20)$$

with $i = 1, \ldots, N$, where N is the population size, and q is a user-defined parameter indicating the probability of selecting the best individual ($i = 1$). In this example, $q = 0.08$.

During crossover, two individuals in the mating pool exchange genetic material. To that end, a breakpoint is randomly selected and the resulting segments are exchanged between the pair, yielding two new individuals (offspring) for the new generation, as Fig. 4b shows.

Also during crossover, random errors might occur, resulting in a mutation. Typically, in Nature mutations occur in a tiny percentage of a species population, and when they do happen, they result in non-viable individuals. Nonetheless, eventually, they may produce new features that improve the mutant aptitude to survive. In this example, a high rate of mutation is used (25%) to allow for increased genetic variability, i.e., to enhance the exploration of the candidate solutions space, augmenting the probability of finding the optimal solution. The rate of mutation indicates the percentage of crossover offspring that shall suffer mutation. For each of those individuals, mutation is implemented by randomly selecting one of its genes, and then replacing the gene's value with a uniform random number between pre-defined boundaries (in this case, between -500 and +500 meters).

Crossover and mutation yield new individuals, producing a new generation. Nonetheless, there is no assurance that the new individuals will have a higher aptitude than their ancestors. To prevent losing the best candidate solutions along the GA cycle, one can use *entire population substitution with elitism*, i.e., the descendants are crossover offspring, and a clone of the current generation's fittest individual is inserted into the following generation. This practice is referred to as *elitism* [Golberg 1989].

The GA cycle halts if the aptitude of the best (i.e., the fittest) individual does not improve by at least $\in = 10^{-4}$ during ϕ= 15 consecutive generations, or if the maximum number of generations $g_{max} = 1000$ is reached, whichever occurs first. The first criterion accounts for the fact that if the best individual's aptitude comes to a steady-state, it might indicate that the GA algorithm is stuck at a local maximum and consequently creating new generations would be pointless. Both criteria limit the total simulation duration, therefore the values of $\in$, ϕ and g_{max} must be empirically defined, according to the available resources, such as processing and storage capacities.

III. RESULTS AND DISCUSSIONS

The simulations were carried out for different numbers of samples (I = 10, 25, and 50) and levels of noise (σ = 9, 45.3, and 90.6 meters) using the GA optimisation toolbox for MATLAB [Houck, Joines and Kay 1995], assuming that a projectile with mass m = 15 kg is launched with muzzle velocity (i.e., initial speed) $\dot{x}_0 = \dot{y}_0 = \dot{z}_0 = 150$ m/s. Reasonable assumptions on the muzzle velocity and projectile mass can be made, based on the characteristics of typical artillery guns. The launch location is the origin of the coordinate system, i.e., $x_0 = y_0 = z_0 = 0$. Finally, C = 0.4, and g = 9.81 m/s2. Under these conditions, the three aforementioned values of the noise standard deviation correspond to 1%, 5% and 10% of the highest altitude reached by the projectile, respectively.

The objective of using GA is to minimize the projectile trajectory reconstruction error. However, if one considers a real world application of such technique - e.g., defensive measures, such as counter artillery in a combat situation - the final objective of the reconstruction is to enable locating the projectiles launching coordinates, in order to disable the attacker. In such a scenario, most probably only the final sections of the projectile flight will be accessible to the defenders. Accordingly, in this simulation the samples were randomly taken along the final two thirds of the projectile trajectory.

3.1. Convergence of the GA

Fig. 5 depicts the results obtained with I = 10 and σ = 9 meters. Fig. 5a shows the aptitude of the best individual (i.e., the candidate solution with the highest fitness value) per generation, as well as the average fitness of each generation population. The fitness function in a GA approach is analogous to the objective function in a classical optimization approach. As elitism is being used, the best individual fitness is a monotonically increasing function. Fig. 5b provides more insight into the convergence of the GA solution, depicting the estimated launch position indicated by the best individual on each generation. One can see that the estimated solutions steadily approach the real launch location (shown in blue). Fig. 5c shows the Euclidean distance between the real and estimated launch points along the generations. Note that after the 96th generation, the algorithm reaches a steady-state (leading to its termination after $\phi = 15$ generations), and that the estimated location of the launcher is within 10 meters of the real position.

3.2. Effect of Noise on Reconstruction Error

Fig. 6 shows the GA reconstructed trajectories using $I = 10$ samples corrupted by different levels of Gaussian noise. The errors in the launch point estimates are 9.6, 39.6 and 50.5 meters, respectively. Even with such a small number of samples (if compared to the 90 samples required to provide an estimate with similar error in [Nelson, Pachter and Musick 2005]), the algorithm can recognizably reconstruct the projectile trajectory. However, to state that the achieved error in the launch position estimates is acceptable, one would have to consider the devised application. For example, in a defensive measure to disable an enemy artillery site, an acceptable error in the estimation of the projectiles launching coordinates would have to be smaller than the maximum blast radius of the counter-battery missiles being used.

3.3. Effect of Number of Samples on Reconstruction Error

In order to better evaluate the impact of noise in the reconstruction error as a function of the number of samples, the algorithm is run 100 times for $I = 10$, 25 and 50 samples for three different values of σ. Fig. 7 summarizes the results, presenting the boxplots of the error in the estimation of launch point location. Fig. 7b shows that, for the lower level of noise ($\sigma = 9$ m), the median error is roughly the same for any number of samples: 21.6, 19 and 17.4 meters, for $I = 10$, 25 and 50 samples, respectively). On the other hand, Fig. 7f indicates that for the higher level of noise used in the simulation ($\sigma = 90.6$ m), the median error increases dramatically as the number of samples decreases from 50 to 10 samples. More samples result in more stable results, with lower dispersion and reduced maximum error. This is made evident by the results for $I = 50$: the boxplots for this number of samples are very similar for any of the values of noise considered in the simulation. For example, note that, while for $I = 10$ the median error rises from 21.6 to 130 meters (a 500% increase) when σ increases from 9 to 90.6 meters, for $I = 50$ the same variation of σ results only in a 100% augmentation in the median error (from 17.4 to 35.4 meters).

Fig. 5. Convergence of the GA-based solution: (a) population average (blue) and best individual (black) aptitude per generation; (b) Three dimensionalview of the estimated launch locations through the generations, with the initial (red) and final (green) estimates, as well as the real launch location (blue); (c) Euclidean distance in three-dimensional space between the best solution of each generation and the real launch point.

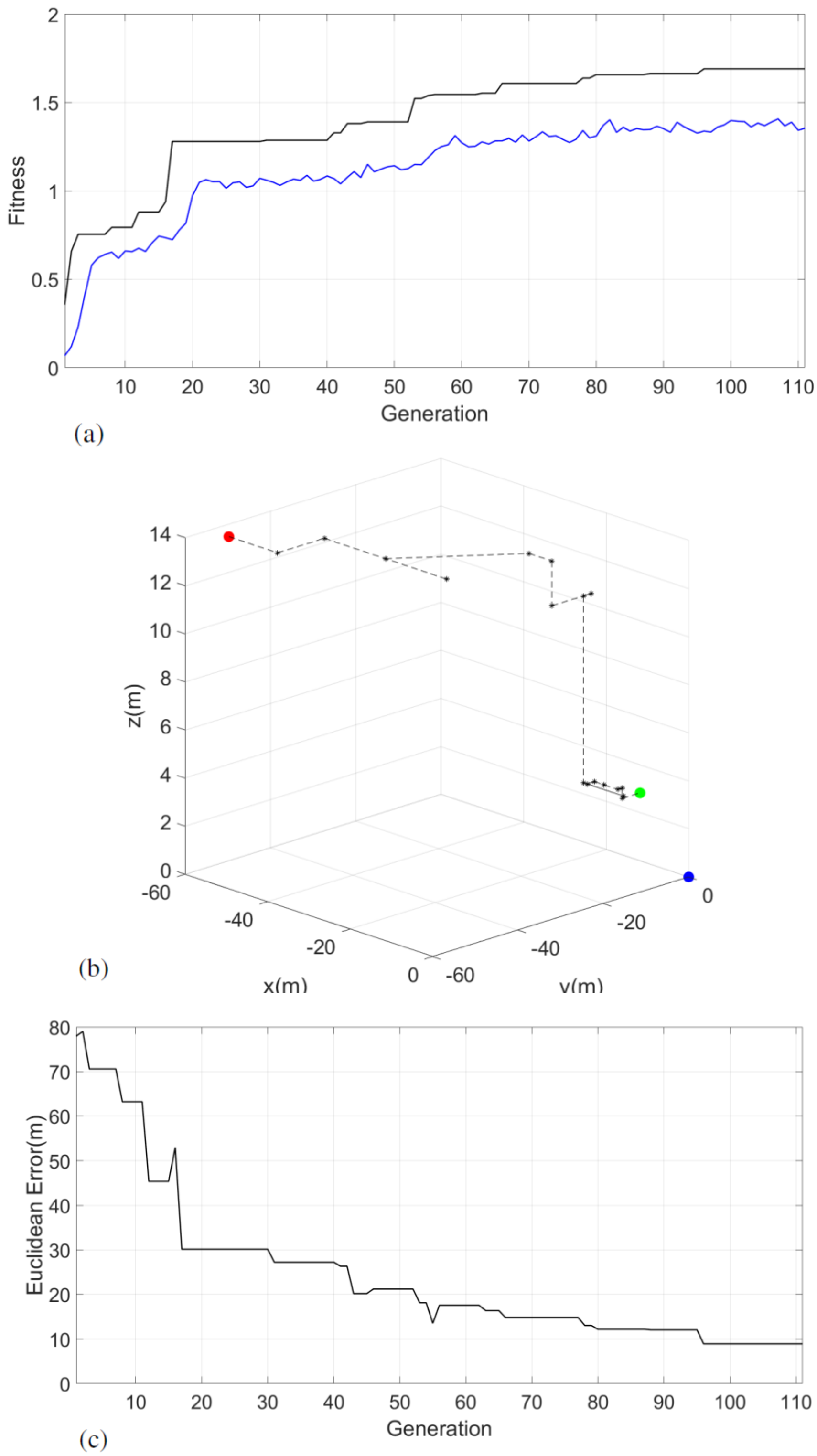


Fig. 6. Real (blue dots) and reconstructed (black dots) trajectories for σ equal to (a) 9 m, (b) 45.3 m, and (c) 90.6 m. The locations of the noisy samples are also depicted (red dots).

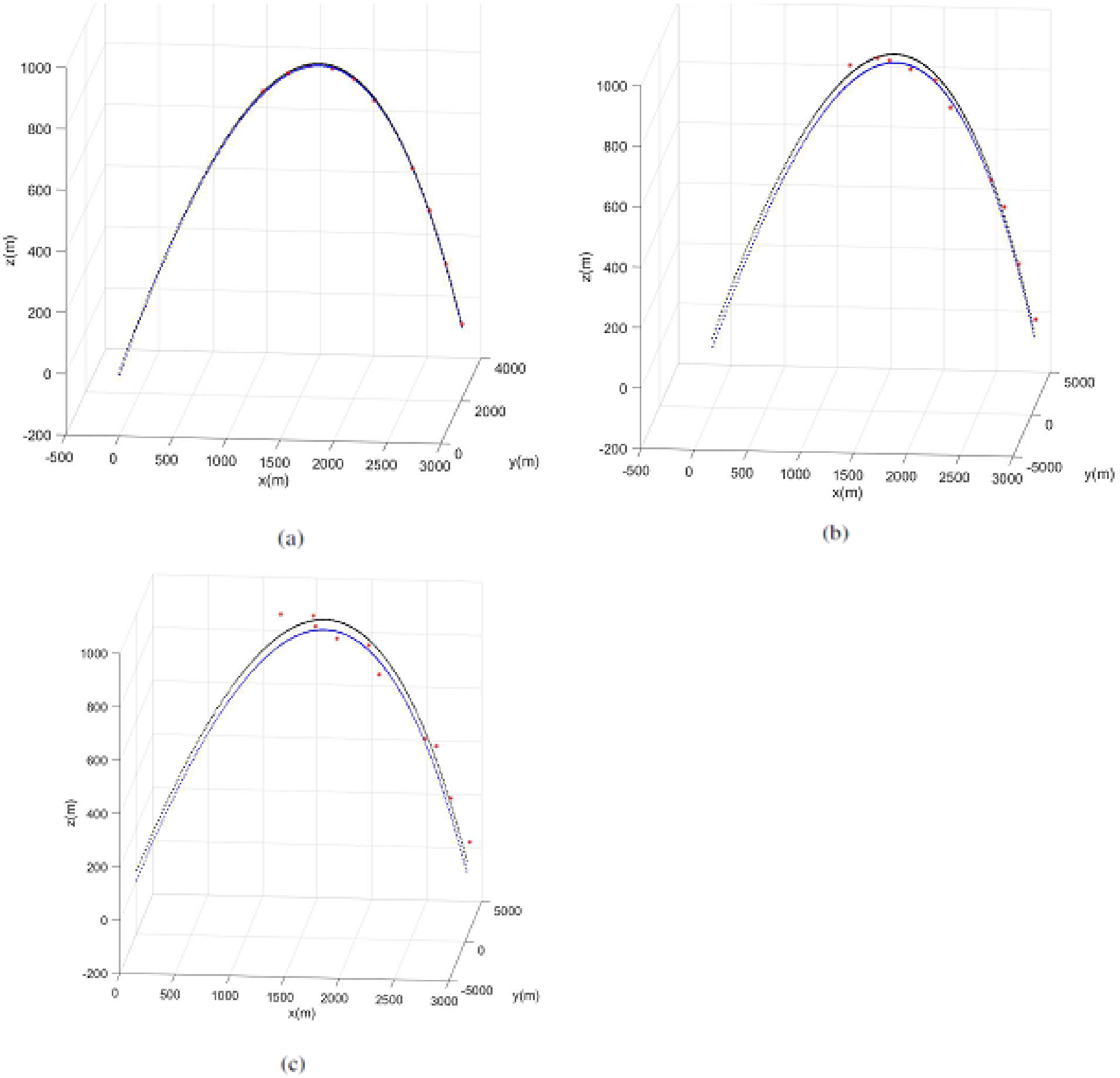
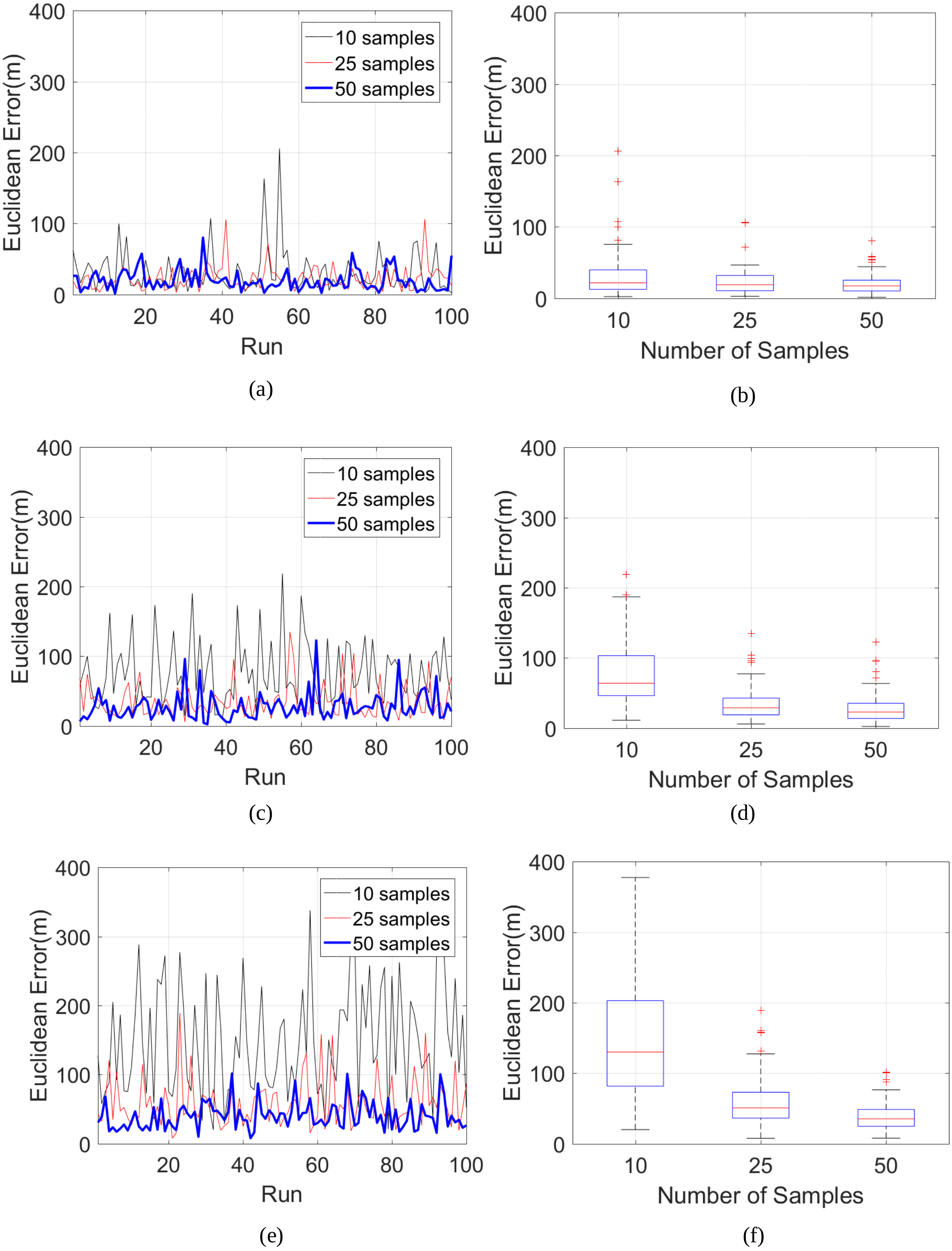


Fig. 7. Errors (Euclidean distance in three-dimensional space) of the launch point estimates over 100 runs of the GA, and the corresponing boxplots, for σ equal to (a,b) 9 m, (c,d) 45.3 m, and (e,f) 90.6 m.



IV. CONCLUSION

Machine learning (ML) importance in a myriad of engineering applications is increasing at a speeding pace. Among the wide variety of ML paradigms, one comes across the so-called Genetic Algorithms (GAs), one of the most important implementations of Evolutionary Learning. The simulated experiment presented here uses GA in the inverse problem of reconstructing the three-dimensional trajectory of a projectile from noisy samples, ultimately aiming at estimating the launch point location. Applying GA in a well-known and easy-to-visualize problem is expected to allow for a better understanding of the selected ML methodology, providing an entry point
for Mechanical Engineering undergraduate (and graduate) students into this field of Artificial Intelligence (AI).

In order to better assess the proposed approach validity, and also to take into consideration its inherent stochastic nature, a statistical analysis of the launch point estimation error is carried out, by running the GA 100 times, each time with a randomly generated initial population. The achieved results indicate that the GA approach yields a precision comparable to that provided by classical numerical or analytical solutions.

As a future development, one might consider including the initial speed, projectile mass and drag coefficient as parameters of this optimization problem, coding these features as additional genes into the candidate solutions chromosomes. This will increase the computational load and increase the simulation time, but improve the proposed adaptive search applicability, in addition to making this didactic example more challenging for AI students.

Conflict of interest

There is no conflict to disclose.

REFERENCES

[1]. Aster, R. C.; Borches, B.; Thurber, C. H. Parameter Estimation and Inverse Problems. 2. ed. Oxford Academic Press, 2013.
[2]. Brent, R. Algorithms for Minimization Without Derivatives. Prentice-Hall, 1973.
[3]. Cordenonsi, A. Z.; Baroni, D. A.; Thielo, M. R. Trajectories Control of a Projectile Using Genetic Algorithms in one Simulated Environment. In: Proceedings of International Symposium on Nonlinear Theory and its Applications, p. 1289–1292, 1997.
[4]. Forsythe, G. G.; Malcolm, M.; Moler, C. Computer Methods for Mathematical Computations. Prentice-Hall, 1976.
[5]. Frey, C.; Osborne, M. The Future of Employment: How Susceptible Are Jobs to Computerisation? Technological Forecasting and Social Change, v. 114, p. 254–280, 2017.
[6]. Golberg, D. E. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989.
[7]. Groetsch, C. W. Inverse Problems: Activities for Undergraduates. MAA Press, 1999.
[8]. Holland, J. Adaptation in natural and artificial systems. MIT Press, 1992.
[9]. Houck, C.; Joines, J.; Kay, G. M. A genetic algorithm for function optimization: a Matlab implementation. NCSU-IE TR, vol. 95, n. 9, 1995.
[10]. Kaisare, N. S. Computational Techniques for Process Simulation and Analysis Using MATLAB. CRC Press, 2017.
[11]. Lange, O. L.; Pierrus, J. Solved Problems in Classical Mechanics: Analytical and Numerical Solutions with Comments. Oxford Academic Press, 2010.
[12]. Lawrence, D. Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991.
[13]. Marsland, S. Machine learning: an algorithmic perspective. CRC Press, 2015.
[14]. Nelson, E.; Pachter, M.; Musick, S. Projectile Launch Point Estimation from Radar Measurements. In: Proceedings of the 2005 American Control Conference, 2005.
[15]. Pardalos, M. P. et al. Parallel search for combinatorial optimization: Genetic algorithms, simulated annealing, tabu search and GRASP. In: International Workshop on Parallel Algorithms for Irregularly Structured Problems, p. 317–331, 1995.
[16]. The British Army. Artillery Locating Devices - MAMBA - Mobile artillery Moniroting Battlefiedl Radar (Ericsson ARTHUR). 2002. Available[Online]: www.armedforces.co.uk/army/listings/l0117.html.
[17]. VDMA European Office. Artificial Intelligence in Mechanical Engineering - Perspectives and Recommendations for Action. 2018. Tech. Report. 976536291-45.

Rafael Saraiva Campos. " Approaching an Inverse Problem in Mechanical Engineering using Genetic Algorithms." *International Journal of Engineering and Science*, vol. 11, no. 9, 2021, pp. 61-71.