Journal Title
XX(X):1–19
©The Author(s) 2021
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Formation control method based on artificial potential fields for aircraft flight simulation

André N. Costa[1], Felipe L. L. Medeiros[1], João P. A. Dantas[1], Diego Geraldo[1], and Nei Y. Soma[2]

Abstract

As simulation becomes more present in the military context for variate purposes, the need for accurate behaviors is of paramount importance. In the air domain, a noteworthy behavior relates to how a group of aircraft moves in a coordinated way. This can be defined as formation flying, which, combined with a move-to-goal behavior, is the focus of this work. The objective of the formation control problem considered is to ensure that simulated aircraft fly autonomously, seeking a formation, while moving towards a goal waypoint. For that, we propose the use of artificial potential fields, which reduce the complexities that implementing a complete cognition model could pose. These fields define forces that control the movement of the entities into formation and to the prescribed waypoint. Our formation control approach is parameterizable, allowing modifications that translate how the aircraft prioritize its sub-behaviors. Instead of defining this prioritization on an empirical basis, we elaborate metrics to evaluate the chosen parameters. From these metrics, we use an optimization methodology to find the best parameter values for a set of scenarios. Thus, our main contribution is bringing together artificial potential fields and simulation optimization to achieve more robust results for simulated military aircraft to fly in formation. We use a large set of scenarios for the optimization process, which evaluates its objective function through the simulations. The results show that the use of the proposed approach may generate gains of up to 27% if compared to arbitrarily selected parameters, with respect to one of the metrics adopted. In addition, we were able to observe that, for the scenarios considered, the presence of a formation leader was an obstacle to achieving the best results, demonstrating that our approach may lead to conclusions with direct operational impacts.

Keywords

Formation flight, simulation optimization, computer simulation, artificial potential fields

Introduction

Moving formations have been observed through centuries in the most variate forms and contexts. Scientists have investigated these structures through various methods, which aimed to identify some of the guiding parameters of such formations. Some of the most prominent methods consider density of individual aggregation, group polarity, and nearest neighbor distance and position.[1]

Inspired by several natural examples, humans have also invested efforts on cooperative ways of moving. For instance, human hunter-gatherers have employed foraging techniques that rely on specific movements, for finding food and retrieving it.[2] More recently, with the advent of airborne platforms, even more nature-inspired swarming movements have been employed by professionals of several fields, what has increased further with the use of drones and other autonomous entities in real or simulated operational scenarios[3].

In the modern military context, the use of formations may serve different purposes, such as: to maximize firepower, to saturate enemy forces, to minimize the opponent's maneuver options, and to enlarge sensor coverage area. In the air domain, more specifically, this may be summarized as mutual support, which one can define as "a contract within a flight of two or more aircraft that supports the flight's mission objectives. An effective mutual support contract will enable a flight to maintain the offensive while enhancing its survival in a hostile environment".[4]

Being such an important facet of air warfare, analysts have also to consider formation flying, when simulating scenarios. Due to increasingly tighter budgetary restrictions, armed forces from around the world have been investing on simulations, rather than live exercises.[5] However, the validity of a simulation is directly related to its model fidelity, which include both physical and behavioral aspects.

Therefore, to provide a way to enrich entity behavior fidelity, this research focuses on the ability of simulated entities, representing military aircraft, to fly in a formation. The formation should be parameterizable, allowing it to consider priorities, which should guide how the entities behave. Moreover, the formation is evaluated through simulation with respect to some dynamically changing

[1]Institute for Advanced Studies, Brazil [2]Aeronautics Institute of Technology, Brazil

Corresponding author:

André N. Costa, Institute for Advanced Studies, Trevo Coronel Aviador José Alberto Albano do Amarante, n° 1 - Putim, São José dos Campos - SP, CEP: 12.228-001, Brazil.

Email: negrao@ieav.cta.br

measurements, which are the basis for optimizing the formation control laws, for the given conditions. The underlying goal of the formation control problem considered in this work is to ensure that the simulated aircraft fly autonomously, seeking a formation, while moving towards a prespecified goal waypoint.

The approach we propose relies on artificial potential fields, which control the flight formation as they command headings and velocities for each entity. The reason for employing such method is to reduce the complexities that modeling a complete swarming mechanism could require and the amount of training tasks that machine learning methods would require. Furthermore, the use of a parametrized model allows for a faster evaluation of its results, which are simulated and optimized for the given conditions. The potential fields are a simple representation of multiple constraints and goals in a swarm system, controlling the overall attraction and repulsion movements between aircraft and to the goal waypoint. In summary, potential fields generate desired velocities and headings to define a trajectory for each entity, which leads them to keeping a formation and flying to the goal concomitantly.

Having defined the potential fields, we implement them within the cognitive models of the simulated aircraft, guiding their maneuvering decisions. With that, we execute a scenario, and evaluate the trajectory with regards to some metrics that represent the degree of mutual protection experienced during the flight. These ratings generate values that supervise an optimization method that aims to achieving a given objective, which could be the minimization of the distances between the aircraft, for instance. With variations of the initial conditions, such as the aircraft original locations, we can define more robust formation control laws, that can be used to guide the simulated aircraft in further scenarios.

We demonstrate the presented approach by a series of simulations with their results being subject to the predefined metrics. These metrics, which guide the optimization process by varying the control laws, allow for achieving increasingly better outcomes for the given inputs. Since, for different scenarios and initial conditions, one may need to employ specific behavior patterns, the proposed method allows for a flexible prioritization of some aspects of the formation flight, through its parametrization. During the tests, we consider two metrics to guide the optimization process calculating the simulation function value. The first is based on minimizing the sum of the overall distances between each of the entities two by two, as well as their distances to the goal waypoint. The second relates to the definition of an average trajectory line, minimizing the squares of the distances of each entity to this path. Besides, another variation of the tests comes by defining a formation leader, which performs a direct flight to the goal waypoint, regardless of the other aircraft within formation.

The main contribution of this work is bringing together artificial potential fields and simulation optimization to defining robust and flexible formation control for simulated military entities, encompassing the main behavioral aspects considered.

The remainder of this paper is structured as follows. Background section presents background information with regards to the addressed problem. Control Model section provides the description of the proposed model for heading and speed control, explaining how the formation is achieved and maintained throughout the simulation. Simulation section reports the details about the exploited simulation technologies and test scenarios, providing the necessary understanding of the models and the simulation environment we adopted. Optimization section addresses the optimization part of the work, with a discussion with respect to its metrics and how we use them to tune the parameters of the model. These sections congregate all the methods used by us to generate the results, which are then presented in Results and Analysis section, considering each of the performed tests, together with their analysis. Finally, Conclusions and future work section concludes the work, pointing out some of the proposed future developments.

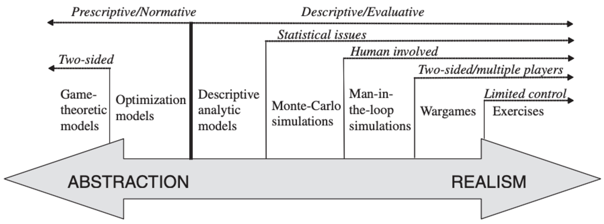

Figure 1. Combat modeling spectrum.[10]

Background

Simulation

A simulation is the attempt of reproducing the operation of a model over time.[6] Therefore, models are its most fundamental elements, representing the main characteristics of systems and processes. Rapid execution of simulation models is important in order to explore a wide variety of scenarios quickly.[7] These modeled aspects may be either behavioral or physical, defining functions and properties of the subject, which are expressed by assumptions concerning the operation of the systems. In summary, as stated by Abielmona et al.[8], the model represents the system itself, whereas the simulation represents its operation over time.

In the military context, combat modeling abstracts and simplifies combat units, including their behaviors, activities, and interrelations, in order to deal with defense-related problems and research questions.[9] The levels of simplification of these models define a combat modeling spectrum, as in Figure 1.

Mainly because analytic models are either not available or intractable, there is a need to formulate the four types to the right in Figure 1. These cases are where simulation takes place, since there are not many ways to achieve the required results other than imitating the system behaviors over time in controlled environments.

The human involvement within these types is what differentiates them, which can also be done by the terms live, virtual, and constructive (LVC) simulation.[11] In live simulations, real people operate real systems, while, in virtual simulations, real people operate simulated systems. Finally, in constructive simulations, simulated people operate simulated systems, as in Figure 2.

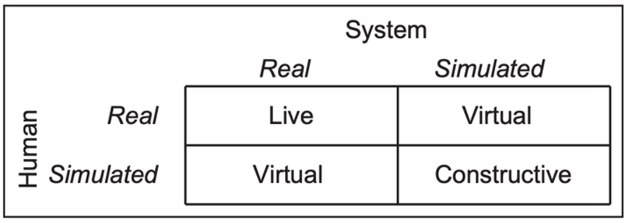

Figure 2. Combat modeling spectrum.[12]

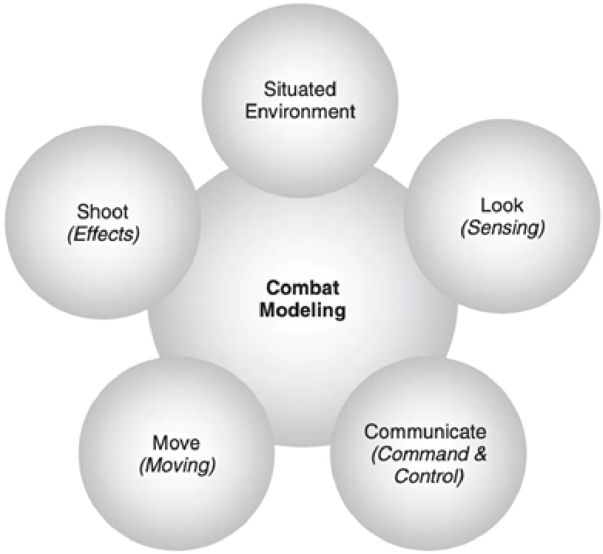

Figure 3. Combat modeling.[9]

In the context of this work, we consider only constructive simulations and that is the reason why the entity models within the scenario behave themselves autonomously, as there is no human intervention during simulation time. To develop entities of such nature, some of the core activities that every battlefield presents must be modelled by the analysts as presented in Figure 3.

Due to the complexity of dealing with all these core activities at once, this work focuses on modeling the moving behavior of the entity. Considering that the research fits in the context of air combat, the movement poses a great challenge due to the number of degrees of freedom available, which admit very flexible maneuvers. When taking into account multiple entities behaving concomitantly, their coordination can be even more challenging. This coordinated movement between a number of aircraft can be designated as formation flying. The United States Federal Aviation Administration (FAA) defines formation flight as: "more than one aircraft which, by prior arrangement between the pilots, operate as a single aircraft with regard to navigation and position reporting." In addition, it states that the separation between aircraft within the formation is the responsibility of the pilots within it, including transition periods when aircraft maneuver to attain separation from each other, as well as during join-up and breakaway.[13]

From this, we conclude that a main aspect of a formation is the separation between its aircraft. Besides, since the aircraft operate as one, they need to be cohesive, that is, they should not be too far from each other. Finally, the coordination

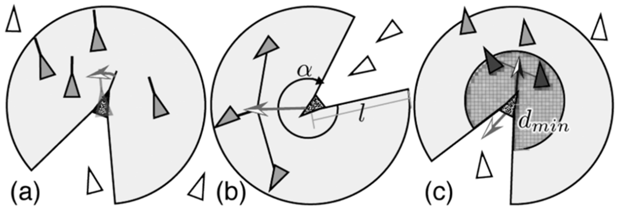

Figure 4. Alignment (a), cohesion (b), separation (c).[15]

of their movement provides a way that allows a coherent navigation, which can be defined as alignment. These are consistent aspects with respect to what was proposed in Reynolds' flocking model,[14] as Figure 4 depicts, being summarized as:

- Separation: steer to avoid crowding local flockmates.
- Alignment: steer towards the average heading of local flockmates.
- Cohesion: steer to move towards the average position of local flockmates.

Optimization

This work proposes the use of a simulation of higher fidelity, which has control models for the simulated aircraft as a means to achieve more realistic results. The simulation is used to optimize an objective function subject to some constraints, what is the very definition of a very prolific research topic called simulation optimization (SO);[16]

Unlike model-based mathematical programming, SO does not assume that an algebraic description of the simulation is at hand. The simulation is available as a black box, allowing only evaluations of the objective function a particular set of input parameters, that is, the approaches only depend on input and output data from the simulation in their search for optimal input parameters.[17]

In other words, as stated by Carson and Maria,[18] SO can be defined as the process of finding the best values for the input variables from among all possibilities without explicitly evaluating each of them. The evaluations are, therefore, made through simulation, which is then coupled with optimization methods.

There are many different metaheuristic algorithms that analysts may use for simulation optimization. Since the performance of the metaheuristic methods has a great dependency on their parametrizations, one needs to carefully select the parameters used on any of these optimization techniques to achieve good results. To that end, Bartz-Beielstein et al.[19] propose the use of a methodology called Sequential Parameter Optimization (SPO) to efficiently design metaheuristics, being also applicable as an optimization method in itself.

This approach has its origins in Design of Experiments (DoE)[20] and in Design and Analysis of Computational Experiments (DACE),[21] which are very well-established terms within the literature. Kleijnen[22] proposes the term Design and Analysis of Simulation Experiments (DASE) for those with either deterministic or random simulation.

With respect to outputs, we realize that design and analysis of experiments are intertwined, inasmuch as the analysis uses an approximation of the I/O function of the experiment. To do so, the analysis usually relies on a metamodel, which can be of several types, such as polynomials,[23] kriging,[24] and the generalized linear model.[25] Researchers often use polynomial methods for local optimization problems, whereas, for global approximation, spatial correlation methods, such as kriging, are more recommended.[23] This is the reason why we adopt kriging in the context of our work.

Scientists use metamodels as fast surrogates for the objective function, facilitating the optimization of simulation models.[26] With the function metamodel provided by kriging, one can have a view on how the I/O function of the experiment behaves. This metamodel can be either directly optimized – in order to find maximum or minimum values – or further improved by performing more experiments. To define what are good factor combinations to properly intensify the search, we advocate for the use of a sequential method, based on Expected Improvement (EI), under computing budget constraints.

Jones, Schonlau and Welch[27] advocated the use of EI as a criterion to select points during a sequential optimization process in a methodology called Efficient Global Optimization (EGO). This is a popular search heuristic that tries to balance the exploration and the exploitation aspects of the optimization, in order to leave from possible local optima. More specifically, EGO selects input combinations based on maximizing the EI, which is computed through a kriging metamodel that approximates the simulation's I/O function.

In a higher level, one can interpret EGO as a form of SPO, applied to problem parameters, as previously defined. SPO is a framework based on active experimentation that aims to test some hypotheses, according to the following procedure:[28]

1. Select a model F (e.g., through kriging) to describe a functional relationship;
2. Select an experimental design, e.g., LHD;
3. Generate data, i.e., perform simulation experiments; and
4. Refine the model until the hypothesis can be accepted/rejected.

In other words, SPO uses the available budget sequentially, i.e., it utilizes information from the exploration of the search space to guide the search by building a metamodel (kriging). After that, it chooses new design points based on predictions (expected improvement) from the metamodel. Thus, it refines the metamodel stepwise to improve knowledge about the search space.[29] Its main goal is to ease the burden of objective function evaluations, when a single evaluation requires a significant amount of resources.[28]

In summary, our approach consists of a single objective optimization, with multimodal response surface, that is, a global optimization problem. With respect to the techniques employed in previous research, it is quite difficult to compare them, since the performance may vary according to the nature of the simulation, as well as the nature of the control variables, which are the input parameters. Besides, much of the literature does not present many performance evaluations and comparisons, affirming that many of the simulation optimization methods are quite different from those available commercially in this regard.

Formation control

The approach we proposed in this work relies on the use of artificial potential fields, which, in turn, define artificial potential forces that control the movement of the entities into formation and to the prescribed goal waypoint. According to some of the criteria available in the literature, the approach falls on solving an amorphous formation tracking problem, with decentralized control (relative and local), by a distance-based behavioral strategy.

Potential functions have been used in formation tracking problems, where a controller is designed based on the gradient of the corresponding potential function, taking into consideration a common interest of the whole group.[30] This is done by carefully choosing the potential function, guaranteeing the desired group behavior.

Even though this method relies on artificial potential fields, all the adaptiveness comes from purely mathematical relations, with no form of optimization per se. Vadakkepat, Tan, and Wang[31] proposed a new approach called Evolutionary Artificial Potential Fields (EAPF), which is truly relevant for our work, since it uses genetic algorithms to optimize the potential functions. Furthermore, their results are verified by simulation, what also relates to the proposed approach. The difference is that this is done with the goal of optimizing the path of the entity with respect to a goal-factor, an obstacle-avoidance-factor, and the minimum-path-length-factor, which are not directly applicable to formation control problems.

A similar approach is present by Zhang, Chen, and Chen,[32] since they also schemed a method to optimize a global obstacle avoidance path originated by artificial potential fields through genetic algorithms. The basic difference resides on the fact that they make a comparison between the current position of the entity and its position on the previous step of the simulation. If there is no difference between these positions, it is considered that the entity has reached a local minimum, what generates a change on an obstacle potential scale factor.

It is important to note that all simulations performed in these works possess a relatively low level of complexity, with very simplified models. Besides, since the problem to be solved is related to path planning, they often focus only on a single entity, thus, there is no concern with regards to formation movement.

Scharf[33] affirms that many behavioral-based methods and potential fields are frequently combined in formation control, whereas Oh, Park, and Ahn[34] asserts that amorphous formation control is generally related to these kinds of methods. The latter also relate artificial potential functions to distance-based approaches, when speaking of amorphous formations. Distance-based approaches are commonly associated with decentralized control, or relative and local, since there is not a global coordinated system, inasmuch as the distances are calculated with regards to each pair of entities.

In distance-based approaches, control laws are nonlinear even if entity models are linear with respect to the input variables, which makes it more complicated to utilize non-behavioral approaches, with very centralized controllers.[34] The artificial potential functions are responsible to control the inter-agent distances with the main advantage of agents needing less global information compared to position- and displacement-based control.

As a variation of the mentioned approach, we conducted some tests to assess the influence of the presence of a leader in the formation, which flies directly to the goal waypoint regardless of the other entities. This indirectly configures a type of leader-follower approach, even though no other alterations were performed within the original methodology.

In summary, although some global optimization was proposed on the field of path planning with artificial potential fields, it was not extended to formation control problems, whereas the formation control approaches that utilize artificial potential fields do not employ optimization methods. Moreover, the simulations performed for validating the methods were rather simplified, not resembling the level of fidelity of the one used by us. Therefore, our work contributes by integrating some of the methods presented in this section, to solve the formation control problem of simulated military aircraft.

Lastly, we highlight that approaches based on artificial potential fields have been paid more attention to because of advantages such as convenient calculation, simple implementation, and outstanding real-time performance[35]. This is also true when compared with machine learning-based methods, primarily because of the heavy training/retraining tasks they may need[36], which is even more relevant when speaking of military problems that have scarcely available data. Since many machine learning algorithms create a model from sample inputs to operate[37], this scarcity is essential when choosing which approach to employ and may be worked around by using synthetic data, for instance. However, in this case, the machine learning model results are highly dependent on the data generation method, which may only transfer the need for a more deterministic and explicit approach (such as artificial potential fields) from the formation control itself to the synthetic data creation process.

Control Model

The proposed model aims to defining commanded headings and speeds for the aircraft. These commands are then provided to each entity within the simulation, which possesses its own dynamic model. The models receive the commands as inputs and compute the next state for the given entity, considering its dynamic response, that is, the approach is not concerned with the control aspects of each aircraft, since this is taken in consideration within the aircraft model itself, complying with all restrictions imposed by its systems.

The main issue addressed by this approach is to maximize mutual support by keeping a formation, while flying to a predefined waypoint. Therefore, the entities must cope with two basic goals, which may be conflicting between each other: fly to waypoint and fly to formation. A typical scenario is depicted in Figure 5.

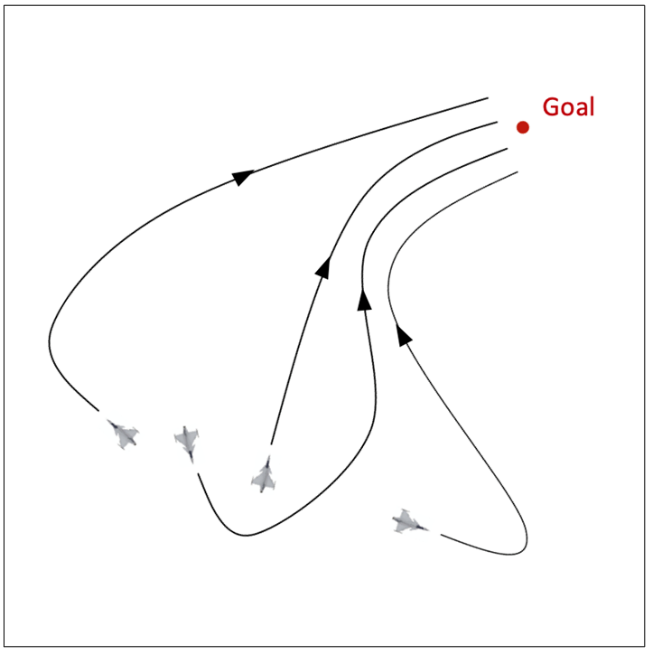

Figure 5. Formation problem scenario example.

Heading control

The main concerns of the proposed approach are to attract entities into a formation and to allow them to stay in that formation as they move to a predefined waypoint (move-to-goal). For producing the formation, attraction forces guarantee the cohesion and repulsion forces act in the separation, following the model proposed by Reynolds[14].

Join (cohesion) The first aspect of the model applies to how each aircraft is attracted to the other members of its formation, as presented in Equation 1.

$$\vec{F}_{cohesion,\ i,\ j} = \frac{1}{\vec{d}\left(p_i,\ p_j\right)^2} \tag{1}$$

The value $d\left(p_i,\ p_j\right)$ is the distance between the aircraft. Thus, for each aircraft there is a cohesion resultant force defined by Equation 2, that is the summation of all the artificial forces generated by the other members of the formation.

$$\vec{F}_{cohesion,\ i} = \sum_j \vec{F}_{cohesion,\ i,\ j} \tag{2}$$

Move-to-goal The same formulation would apply to how the aircraft must move towards the goal position, which is considered to be the most important factor when defining commanded headings and speeds. In a similar way, this would be done through the definition of an artificial attraction force between the *i*-th aircraft and the goal as expressed in Equation 3.

$$\vec{F}_{goal,\ i} = \frac{1}{\vec{d}\left(p_i,\ p_{goal}\right)^2} \tag{3}$$

When considering this move-to-goal force in conjunction with the join force presented, there is a rather undesirable behavior that may emerge. This is due to the fact that all forces may cancel each other, producing a zero resultant. To avoid that, we suggest the use of another type of force to account for the move-to-goal aspect of the formation control. Since the cause for the resultant force to be zero is $\vec{F}_{cohesion,\ i}$ to have the same magnitude as $\vec{F}_{goal,\ i}$ with opposite directions, the proposed move-to-goal force has its intensity given by Equation 4, pointing to the waypoint from the position of the *i*-th aircraft.

$$F_{mtg,i} = F_{cohesion,i} \times f \tag{4}$$

In Equation 4, f is a value that defines what is the priority between flying in formation and reaching the waypoint. It is called grouping factor and is defined in Equation 5.

$$f = \frac{10}{1 + \frac{99c}{100}} \tag{5}$$

Equation 5 sets a limit of 10 times that the move-to-goal behavior may be prioritized over formation flying or vice-versa, since value c may vary from 0 to 100. The lower bound for f is set on 0.1, which would mean that $F_{cohesion,i} = 10 \times F_{mtg,i}$, whereas the upper bound is 10, meaning the opposite, i.e. $F_{mtg,i} = 10 \times F_{cohesion,i}$. The reason to use Equation 5 instead of setting f directly as a value from 0.1 to 10 was for the users of the simulation platform to have a simpler slider in which they could select a value from 0 to 100.

Besides the case where $f = 1$, which should be avoided by the optimization process itself, with the formulation introduced in Equation 4, there is another case where the resultant force in the aircraft is set to zero, which is when the formation forces nullify each other. To avoid this to happen, we define that, if $F_{cohesion,i} = 0$, then $F_{mtg,i} = F_{goal,i}$.

Safety (separation) The second element on Reynolds[14] formulation is separation. Since the aircraft are attracted to each other, eventually they would end up colliding. To avoid this to happen, it is defined a repulsion area, which also has a transition zone, with no potential field acting. In summary, when the distance between an aircraft to another is smaller than the minimum safety distance, the attraction gravitational force is converted to repulsion, possessing a similar formulation with the only difference being a negative sign, as stated in Equation 6. This minimum distance may also vary according to the application, for instance, a reconnaissance mission could impose a larger distance between the aircraft flying in formation.

$$\vec{F}_{separation,\ i,\ j} = -\frac{1}{\vec{d}\,(p_i,\ p_j)^2} \tag{6}$$

Similarly, there is a summation of the repulsion forces of each aircraft, forming the resultant repulsion force stated in Equation 7.

$$\vec{F}_{separation,\ i} = -\sum_j \vec{F}_{separation,\ i,\ j} \tag{7}$$

If the resultant cohesion force is added to the resultant repulsion force, a resultant formation force is defined, as showed in Equation 8.

$$\vec{F}_{formation,\ \ i} = \vec{F}_{cohesion,i} + \vec{F}_{repulsion,i} \tag{8}$$

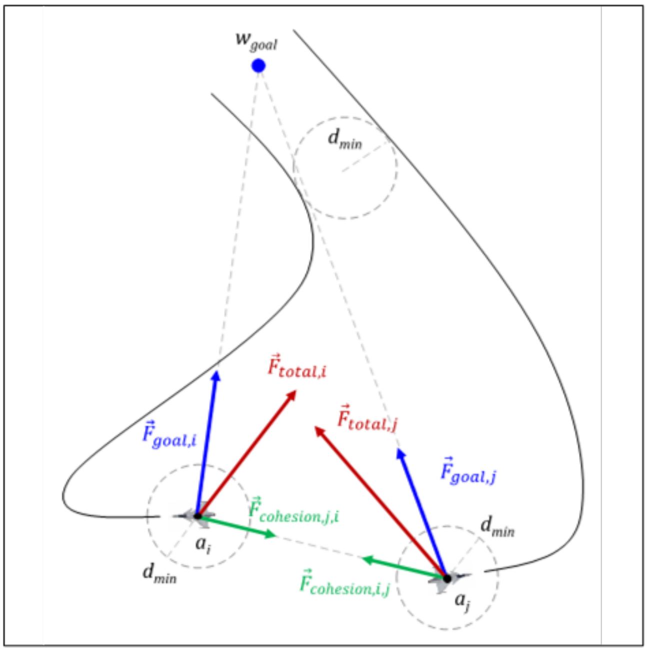

Figure 6. Gravitational grouping forces.

Finally, this resultant formation force can be added to the goal force, resulting in the total force that acts in a given entity, what is stated in Equation 9 and showed in Figure 6.

$$\vec{F}_{total,\ \ i} = \vec{F}_{goal,i} + \vec{F}_{formation,i} \tag{9}$$

Figure 6 presents a scenario in which there is no repulsion force, since the aircraft (a_i and a_j) are not within the minimum distance (d_{min}) for that to happen. We also point out that, even though we account for the separation in our formulation, collisions are not possible in the scenarios considered by us. This is because the aircraft remain within prespecified flight levels (altitude intervals), which differ from each other. For scenarios with a greater demand for collision avoidance, we suggest that the separation force to be stronger, maybe considering a higher order power in the denominator. Moreover, all calculations are made in two dimensions, even though the vectorial treatment would be the same for considering a third dimension, which could increase computational time for processing.

Speed Control (alignment)

Finally, the third element on Reynolds[14] formulation is alignment, which is performed by the speed control. We also controlled this by the distances between aircraft in a similar way. We define the commanded speed through some calculations that utilize the absolute value of the resulting force acting on a given aircraft. We convert this value on a percentage, which we apply to the maximum aircraft speed in order to define the resulting command speed as Equation 10 shows.

$$v_{commanded,i} = \frac{v_{max,i} + v_{min,i}}{2} + p\left(v_{max,i} - \frac{v_{max,i} + v_{min,i}}{2}\right) \tag{10}$$

The aircraft model defines the velocities v_{max}, i and $v_{min,i}$, which are kept constant in our case. The percentage value p, defines how fast the aircraft should fly and is defined by Equation 11.

$$p = \frac{\left|\vec{F}_{total,i}\right|}{\left|\vec{F}_{formation,i}\right| + \left|\vec{F}_{mtg,i}\right|} \quad (11)$$

From Equation 11, we conclude that the p value has its upper bound in 1. Therefore, the maximum value assumed by $v_{commanded,i}$ in Equation 10 is v_{max}, i, whereas the minimum value is $\frac{v_{max,i}+v_{min,i}}{2}$. The decision to use the average between maximum and minimum speeds as the lower bound was to accelerate the process of either formation flying or moving to the goal waypoint, what we justify by the fact that rarely pilots would employ the aircraft's minimum speeds during such a flight.

Simulation

One of the greatest strengths of the proposed approach is to utilize a reliable set of models to achieve simulation results close to what is expected in the reality. Since the main goal is to define CGF that will resemble what real pilots would do in the field, the behavior of the entities must be coupled with well-founded models, which present a sufficient level of fidelity. We do this through the use of the ASA Framework (Ambiente de Simulação Aerospacial in Portuguese),[38] which is a project developed in the Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Division from the Brazilian Air Force's Institute for Advanced Studies (Instituto de Estudos Avançados – IEAv, in Portuguese). Its main goal is to provide a computational solution that enables the simulation of operational scenarios of interest of the Air Force.

This is done through constructive simulation, where participants stablish scenarios, parameters, and command decisions, which are simulated to support the development of tactics, techniques, and procedures.[39] Therefore, the concept of simulation to which this work refers is the one where scenario elements are represented through autonomous agents that can take their own decisions based on artificial intelligence models or preestablished rules.

ASA's simulation engine is based on the Mixed Reality Simulation Platform (MIXR), previously known as OpenEaagles.[40] It is based upon EAAGLES (Extensible Architecture for the Analysis and Generation of Linked Simulations), a simulation framework that the United States Air Force develops and maintains to support several simulation activities. MIXR is an open-source set of code libraries in C++ for the creation of various applications for virtual and constructive simulation. This package includes models of different aeronautical systems and the environment in which they are inserted. Besides, there are diverse provisions for the inclusion of distinct behaviors for the entities modeled.

Models

Dynamics Each entity within MIXR can have many components and systems attached to it. When speaking of an aircraft, it is fundamental to include in its architecture a dynamic model. Taking advantage of its level of maturity, the MIXR development team has taken a decision to utilize JSBSim[41] as its primary dynamic model. This was also because, being open-source, MIXR could not include dynamic models from EAAGLES, which are classified. Since JSBSim is also coded in C++, extending the native MIXR dynamics class to utilize JSBSim was a rather straightforward effort. Thanks to C++ and the object-oriented nature of MIXR and JSBSim, multiple instantiations of JSBSim can be created and utilized within the same simulation.

The main purpose of a flight dynamics model is to propagate and track the path of an aircraft over the surface of the Earth, considering the forces and moments that act on the vehicle.[42] Therefore, there is a need to properly introduce characteristics of the aircraft, as well as the planet's (e.g. gravity and rotation rate).

For calculations with regards to translational and angular accelerations, as well as velocities, JSBSim bases itself on the formulation presented by Stevens and Lewis.[43] Although it is beyond the scope of our work to address this issue in depth, we acknowledge that JSBSim provides the necessary basis to support aircraft motion over a rotating, spherical earth, including effects of rotating mechanisms, such as engines and propellers, besides steady winds and turbulence.

Finally, JSBSim supplies a flight control system model, providing a set of components that can be linked to represent control laws for an aircraft, what is of paramount importance for our work.[42] These components include multi-purpose filters, switches, and gains, which together configure the arbitrary system modeling. Even though the model provides default values, the control models that we utilize in the context of this work are a result of efforts within IEAv's C4ISR Division towards properly tuning control parameters in order to achieve higher fidelity outcomes, which are validated by Brazilian Air Force pilots.

Behavior Beyond the models that represent how the aircraft interact physically with the environment and with each other, there is a need to model how the entities respond to the stimuli that come from these interactions. In the context of military simulation, usually the goal of autonomous entities is to represent lifelike intelligence, therefore requiring agents to be able to handle ever increasing complex tasks and situations, resulting in both their design and overall time in development to also grow considerably.[44]

Scientists have proposed many artificial intelligence methods to cope with these increasing complexities that the simulated scenarios pose in the behavioral standpoint. However, most of them struggle with blending action outputs of behaviors that are acting upon the same entity. This is because, although having the ability to present simultaneous execution of behaviors, usually these behaviors are coupled with the controller, meaning the behavior itself performs the action execution of the entity. Thus, there is no way to combine the action execution of different behaviors, what is much desired as it allows for new behaviors to emerge from existing ones, without the user having to implement a new predefined behavior, therefore reducing code complexity.[45]

We have identified the Unified Behavior Framework (UBF)[46] as an effective means to do this, as the behaviors in UBF do not execute the action for the agent, instead they only

Table 1. General parameter values for aircraft model.

Parameter	Value
Cruise speed	530 kn
Available fuel	3000 lbs
Wing area	300 ft^2
Wingspan	30 ft
Empty weight	17400 lbs
Bank angle	60º

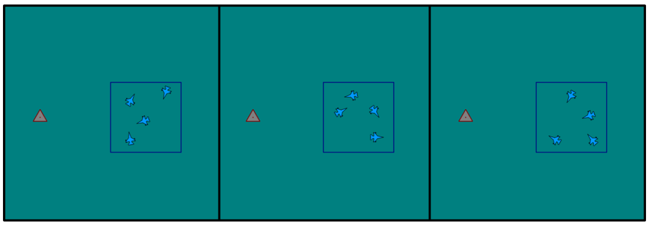

Figure 7. Scenario examples in the ASA Framework.

produce an action object comprised of parameters, which user-defined arbiters are able to manipulate and fuse together.

In the context of UBF, the tasks interpret the perceived state, that comes from sensors that feed the controller, recommending certain actions to be taken, what every behavior does by each traversal of the UBF tree.[47] By this method, the UBF tree remains decoupled from the specifics of the entity, enhancing the flexibility of the framework for use in different applications. These actions might represent small adjustments to the aircraft's actuators, but are typically more abstract representations, such as vectors indicating a desired direction and magnitude for the entity to fly. As such, behaviors can tailor the actions to the desired effect on the agent, but the details of the actuation of controls is left to the controller and is therefore not dealt with inside the UBF tree.[46]

Therefore, within UBF, we implemented a behavior according to the proposed formation control approach, defining the presented equations and furnishing them with data from the environment. Besides the calculations themselves, the behavior generates heading and speed commands, which are then passed to the flight control system in order to lead the entities to move in a prescribed way. MIXR framework is responsible to call all these steps within every simulation frame.

Test Scenarios

Having presented the mathematical formulation of the proposed approach, as well as some of the characteristics of the computational environment used for its implementation, it is appropriate for us to introduce the scenarios that we utilize to conduct the optimization of the formation control method. It is important to note that the goal is to conceive scenarios with characteristics that are compatible with the reality of the military operations, not necessarily to explore all existing possibilities.

Scenarios consist of four ASA Fighter aircraft, which have several subsystems, such as radar, datalink, and radar warning receiver (RWR). We based the definition of this number of aircraft on the doctrine of several Air Forces, such as the characterizations the Joint Chiefs of Staff (2019) stated, which define that the basic tactical unit in the Air Force consists of four or more aircraft divided in two or more elements. We kept constant the general parameters to configure each aircraft model throughout the scenarios. Table 1 shows some of these parameters.

These four aircraft can be initiated anywhere inside a specified region of the map, which we considered to be their FAOR (Fighter Area of Responsibility). There is an assumption that, after a combat phase, which places the aircraft in random positions, they must seek to maximize their mutual support through formation flying. Besides, the aircraft have the goal of flying to a predefined waypoint, which is located outside the limits of the FAOR. We consider this waypoint to be, for instance, an approach fix for an operating base and treat it as a fixed point throughout the scenarios.

For scenario generation, there are three variable parameters for each aircraft, namely: latitude, longitude and heading. All aircraft keep a specific flight level (altitude), to guarantee level separation. Therefore, trajectory intersection (collision) is not a concern. Figure 7 shows some examples of the utilized scenarios.

Since each of the four aircraft has three parameters, we defined each scenario by twelve values. The heading parameter goes from 0º to 360º, whereas latitude and longitude are confined to the area shown in Figure 7, varying from 1º N to 1º S and from 1º E to 1º W respectively. This represents a region that is centered in the Earth's equator, what guarantees that one degree in latitude and in longitude are almost equivalent values in kilometers, forming a square region.

To define values within the prescribed intervals, we utilize the Latin Hypercube Sampling (LHS), a space-filling technique. LHS is an attempt to ensure that all portions of the sample space are considered and that each of the input variables X_k has all portions of its distribution represented by input values, by dividing the range of each X_k into n strata of marginal probability $1/n$, and sample once from each stratum.[48] Considering the case where each X_i component is uniformly distributed over [0, 1] and a sample of size n is to be selected, first we divide the [0,1] domain of each $X_i, 1 \leq i \leq k$, into n intervals. The set of all possible Cartesian products of these intervals constitutes a partitioning of the k-dimensional sample space into n^k cells. Then we select n cells from the n^k population of cells in such a way that the projections of the centers of each of the cells onto each of the k axes yield n distinct points. Finally, we choose a point at random in each selected cell, defining the design.[49]

Certain types of Latin Hypercube Design (LHD) may be space-filling, while some may not. In order to increase the multidimensional uniformity of the method, scientists have proposed some variations based on, for example, the maximin distance criterion, which is the one we utilized in our work. This design seeks to maximize the minimum statistical distance between model inputs, which are post-processed by the LHS algorithm.[50]

Bartz-Beielstein, Stork and Zaeffer[28] implemented the minimax LHS algorithm in R[51] within the package called SPOT. Beginning with a random starting point in the design space and a matrix of randomly generated locations in the Latin hypercube, we apply the design methodology by choosing the next point from the matrix of available locations with the maximum minimum inter-point distance from the points already included in the design matrix. The algorithm proceeds by adding one point at a time until the design matrix has been completely generated, resulting in an LHS matrix with increased multidimensional uniformity.

Through the use of this R package, we generated 120 scenarios, which we specify by a set of 12 variables each, forming a 120x12 matrix of input values. This matrix originally contains numbers in the [0,1] interval, but, after applying the boundary values for each of the input variables, they become coherent with the definitions aforementioned.

Finally, after placing the aircraft within the delimited area, we set the goal waypoint to a location with latitude 0º N and longitude 3º E, being fixed throughout the scenarios, as Figure 7 portrays.

Optimization

General concept

Through an R script, the user can set parameters for the optimization method, as well as for the scenario generation, as done through LHS. As we already explained, we create scenarios stochastically, defining some input parameters for the simulations. On the other hand, the optimization method generates inputs with regards to the artificial potential fields that are used as control laws for the simulated entities.

In a conventional simulation optimization fashion, as Figure 8 depicts, following this setup phase, the simulation starts, making the CGF to perform according to the predefined parameters. After the simulation ends, the system records state data and shares it with the controlling script, which calculates the value of the objective function with regards to the specific run. With this, the optimization method generates a new set of input parameters, aiming to achieve better values for the objective function.

We already addressed the scenario generation and the simulation phases in previous sections, therefore, our focus now is to delineate how the R script will configure these processes, as well as to perform the optimization itself, that is, the optimization method and the function evaluation phases. The configuration is a rather straightforward phase, with the sole goal of setting parameters for the other methods.

Optimization method

Although the ASA Framework allows the inclusion of stochasticity within its models, in the context of this work, we did not consider any randomness, i.e., the models provide the same output whenever the combination of input values is the same. This is because the random simulation models that ASA's team have implemented so far in the framework are related to weapon effects, which we do not contemplate in our work. Note that either deterministic or stochastic designs treat the experiment as a black box, that is, only input/output (I/O) data are observed. We call the input values as factors, which can be either a parameter of the particular problem (the black box) or of the optimization algorithm itself, what leads to the following definition:[19]

- Algorithm parameters are related to the algorithm, which should be improved.
- Problem parameters describe the problem to be solved by the algorithm.

Whereas in real-life experimentation it is hard to vary a factor over many values, in computer experiments this restriction does not apply. As a result, one can observe a myriad of scenarios – combinations of factor values. One can use many methods to carefully select these combinations to cope with resource restrictions. A very consolidated technique is the LHS, which we already discussed.

This initial space-filling design, which, in our case, is the LHD, accounts for box constraints for the inputs. The simulations performed in ASA provide the information from the exploration, being configured as a set of scenarios that are also defined by LHS. The kriging metamodel, combined with EI, allows for sequentially increasing the initial design, i.e., after it analyzes the observations – so one can better understand the data generating process – it selects the next input combination. Whereas it selects some combinations to improve the kriging metamodel (global search), it adds some other combinations because they seem to be close to the local optimum (local search).[19] It gathers the final knowledge from the optimization of the latest metamodel, which is done through L-BFGS-B,[52] a quasi-Newton method that approximates the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm using a limited (L) amount of computer memory. Its B version is an extension to handle simple box constraints on variables. Figure 9 depicts this adopted optimization strategy.

In order to perform the optimization depicted in Figure 9, we utilize an R package called Sequential Parameter Optimization Toolbox (SPOT).[28] This library is a set of tools for model-based optimization and tuning of algorithms, including surrogate models, optimizers and DoE approaches, such as the LHD.

Initial configuration

We employ the empirical rule of $n = 10k$, for setting up the initial configuration. This means that, for each input variable, we generate ten variations, combining them with the others. In our case, since we control one variable related to the formation control approach, namely c, there would be only 10 samples to be run in the initial design. However, to better fit the space, we considered an initial scenario of 30 points, ensuring that we sampled all portions of the input space.

Function evaluation

From the simulation results, there is a need to define how we evaluate the objective functions, as Figure 8 states (the simulation node encompasses this process in Figure 9). In other words, we must define the metrics that will drive the optimization process in order to determine which are the best values for the input parameters. We propose two different approaches: one based in the distances between the entities and the other based on a goal trajectory geometrically predefined.

Distance-based To define a robust set of parameters that would represent the desired formation movement, a metric

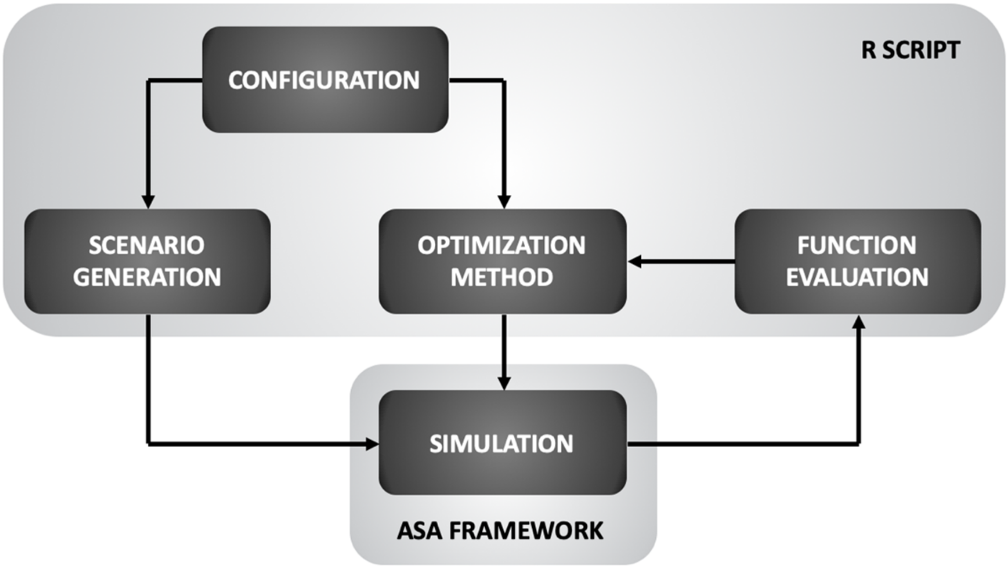

Figure 8. Conventional simulation optimization approach.

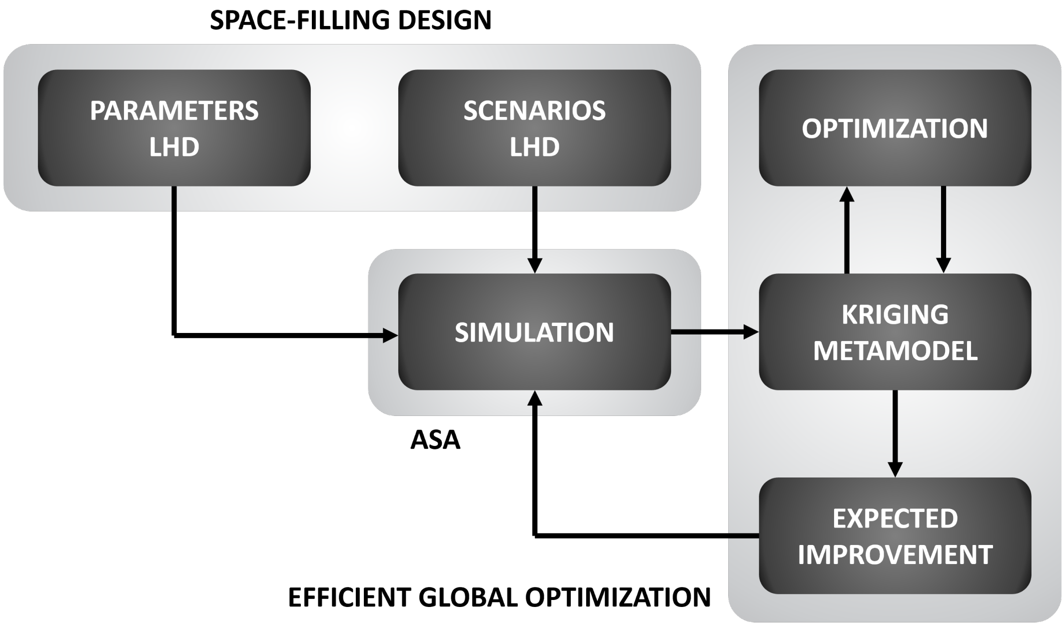

Figure 9. Adopted optimization strategy.

we propose is to find the simulation run in which the sum of the artificial potential forces in every frame is maximized, as Equation 12 states.

$$\max \sum_{t=0}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} G \frac{m_i m_j}{d(p_{i,t}, p_{j,t})^2} \tag{12}$$

This equation is valid for $i \neq j$ and has the potential of anomalies being generated due to the division by zero ($d = 0$). This may happen when the entities' trajectories cross each other, what is possible due to the level (altitude) separation between them. To avoid the risk of dividing by zero, we rewrite Equation 12 as follows:

$$\min \sum_{t=0}^{T} \sum_{i=1}^{n} \sum_{j=1}^{n} d(p_{i,t}, p_{j,t}) \tag{13}$$

Therefore, for each simulation run, we calculate the sum of the distances between each aircraft at any given instant, with the goal of the optimization process being to find the set of input parameters that minimizes it. What Equation 13 tries to find is the simulation run in which the aircraft fly longer in the closest formation and reach the goal waypoint in the fastest way.

Trajectory-based The alternative method we use for comparison consists of the definition of an analytical trajectory, which is used for the optimization of the artificial potential field function. We do that through the method of

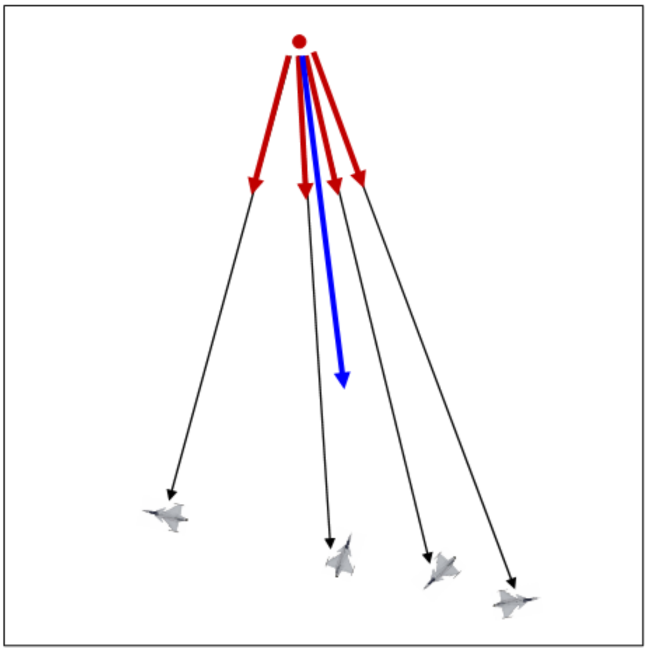

Figure 10. Average between angles, forming a simplified trajectory.

least squares, which compares the actual trajectory generated by the artificial potential fields with an analytical trajectory, defining an error value. With this value at hand, we optimize the parameters of the artificial potential function with the goal of minimizing the sum of the squares of the errors at each simulation frame, as Equation 14 shows.

$$\min \sum_{t=0}^{T} \sum_{i=1}^{n} \left(p_{i,t} - traj\left(i\right)\right)^2 \tag{14}$$

The factor $traj\left(i\right)$ indicates the trajectory function for the i-th aircraft. Therefore, this method is very dependent on the analytical trajectory defined at first. Since this definition is not the main focus of this work, we geometrically prescribe a very simple path. Basically, it draws a direct line between each aircraft's position and the goal waypoint, defining a vector. Then it normalizes these vectors into versors (unit vectors), being used to calculate an average between these vectors, which is simply their sum, as depicted in Figure 10.

In summary, the average between the vectors define a simplified path, which we use for comparison in relation to the entities' trajectories at every time step, according to Equation 14.

Results and analysis

Description of experiments

We performed the experiments based on scenarios defined by an LHD, which applies the 10k rule. Since a scenario is described by 3 input variables (latitude, longitude, and heading) for each of the 4 aircraft, as we previously stated, 120 scenarios were generated. These scenarios represent a mission of 15 minutes, running approximately 33 times faster than reality, yielding an average of 25 seconds for each run.

We evaluated each of these scenarios considering another initial LHD of 30 points, i.e., selecting 30 grouping values from 1 to 100, with the SPO methodology defining 90 more points (grouping values), totaling 120 simulation runs for each of the also 120 scenarios. Therefore, for each of the four experiments (distance-based leaderless, distancce-based with leader, trajectory-based leaderless, and trajectory-based with leader), we executed 14400 simulations (120 runs of 120 scenarios), which took approximately 100 hours to complete (more than 4 days). Thus, the total execution time for the presented results was 17 days, which refers to 57600 executions.

Figure 11. Example of initial design of an experiment.

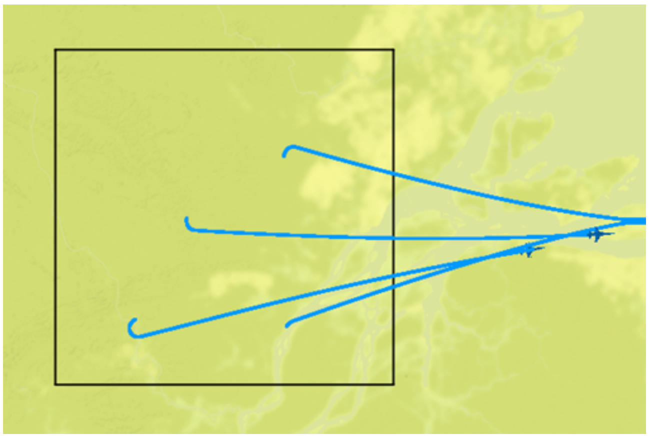

Figure 12. Aircraft trajectory for grouping value of 10.

The simulation ran on a laptop with standard configurations (CPU: Intel Core i7-6820HQ 2.70GHz; RAM: 16.0GB), which was done due to the operational facet of our work. Since the main idea is to be able to optimize behaviors for simulated entities within military scenarios, this method provided a means to use regular computers, considering a simulation budget that stays within the available time constraints. If needed, more powerful machines may be utilized, which would make the process much faster.

To illustrate some of these scenarios, we present in Figure 11 an initial design utilized for one the simulation runs.

From this initial design, we show in Figure 12 and Figure 13 what two different grouping values generate within our experimental framework.

As we can visually confirm from Figure 12 and Figure 13, a larger grouping value caused the aircraft to fly closer to a formation, even on challenging initial configurations, as the one Figure 11 depicts. This means that they could exercise a better mutual support for each other, which is our fundamental goal.

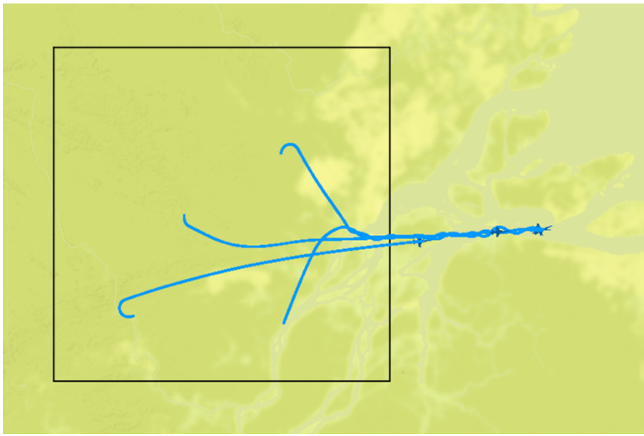

Figure 13. Aircraft trajectory for grouping value of 90.

Another important aspect of these illustrations is that the trajectories the formation control approach generated were smooth and verisimilar, being regarded as credible by the Air Force pilots who followed the simulation execution.

Since we simulated a considerably large number of scenarios, it would be too cumbersome to present all runs in a pictorial way. Instead, in the next section we present them in a more concise fashion, averaging the results of various scenarios.

Results

For each of the four experiments, we present two charts, which basically state the inputs and outputs of the simulations on each iteration. Each iteration is a set of 120 executions, which account for each one of the scenarios in the design. Therefore, each output value represented on the charts is the average of the metric values for all the 120 scenarios with a given input value (grouping value). Moreover, we identify the best value of each iteration as being the suggested grouping value for each metric. Besides the input and output charts, plotted against the run number, i.e. sequentially, we provide a graph that relates I/O data to each other, defining an approximate function to represent the relation between them. The input is the grouping value, whereas the output is the metric value.

Both distance-based and trajectory-based metrics can be either leaderless or with leader. The distance-based metric aims to minimize the overall pair-wise distance, considering both between aircraft and between aircraft and the goal. On the other hand, the trajectory-based metric considers a mean vector between all segments that connect an aircraft to the goal, computing the sum of the squares of the distance between each aircraft to this mean vector. We consider the presence of a leader through setting its grouping value to 0, which means that one of the aircraft does not consider the others, but only flies to the goal, expecting the other to adapt to its own trajectory.

Distance-based leaderless This test case considers the metric of minimization of the sum of the pairwise distances at each simulation tick, and regards all aircraft as equals, that is, there is no hierarchy imposed.

As all of the other charts, the first points of Figure 14 show the pilot design, accounting for an initial investigation of the search space. With a promising region identified, the optimization method intensifies the search around a point with the goal of finding the best value.

In this case, the best value is deemed to be 32278.0323 with a grouping value of 0.6335. The proximity of this value to 0 means that the aircraft highly prioritize moving to goal, instead of flying in formation. Moreover, from the output side of Figure 14 (green), one can observe that the initial 30 points from the grouping value LHD are able to quickly lead the search to values that are close to the one deemed as the best. However, although the input value seems to converge, there is a small oscillation of the metric value (output).

Figure 15 shows that, for this case, increasing the grouping value (input value), the function value also increases, what is undesirable. This conclusion reinforces that this metric prioritizes the move-to-goal behavior, instead of the formation flying.

Distance-based with leader In this test case, we consider the same metric as the previous, fixing the grouping value of one aircraft as 0, so that a leader behavior could have been mimicked.

With the presence of a leader, the best value obtained was 32252.9881 – being slightly better than the leaderless – with a grouping value of 0.7213. Figure 16 shows a very similar dispersion, however with a visible higher dispersion and a lower mean value. Again, there is an oscillation with respect to the metric value (output).

The presence of a leader seems to highly disturb the function presented in Figure 15, as Figure 17 shows, creating local minima, which could pose a challenge to the optimization method. Since the sampling method employed vary according to SPOT parameters, the differences may not be as evident as the charts showed.

Again, in this case, the optimization process led to a prioritization of the move-to-goal behavior, which indicates that the metric we adopted did not seem compatible with the desired mutual support.

Trajectory-based leaderless We conducted these experiments with a different metric concept, which considered the distances between each aircraft to a prespecified trajectory. This trajectory was a result of the vectorial sum of versors in the direction that connects each aircraft to the goal waypoint. Again, there was no hierarchy considered, that is, all aircraft's grouping values were subject to the optimization.

Changing the metric, the dispersions were very different (Figure 18), with a much lower variance. The input grouping value that lead to the best value was 32.1681, which is way higher than the previous results. This means that, for this metric, the entities tend to fly more in formation, however still prioritizing the move-to-goal behavior. The best metric value (output) generated by the best grouping value was 0.4761. Differently from the previous two cases, this one did not have a pronounced oscillation with regards to the output value.

Figure 19 presents a quite different function if compared with the previous ones, referent to the distance-based metric. This is because the trajectory-based metric prioritizes formation flying, when forcing the aircraft to fly closer to the mean vector to the waypoint.

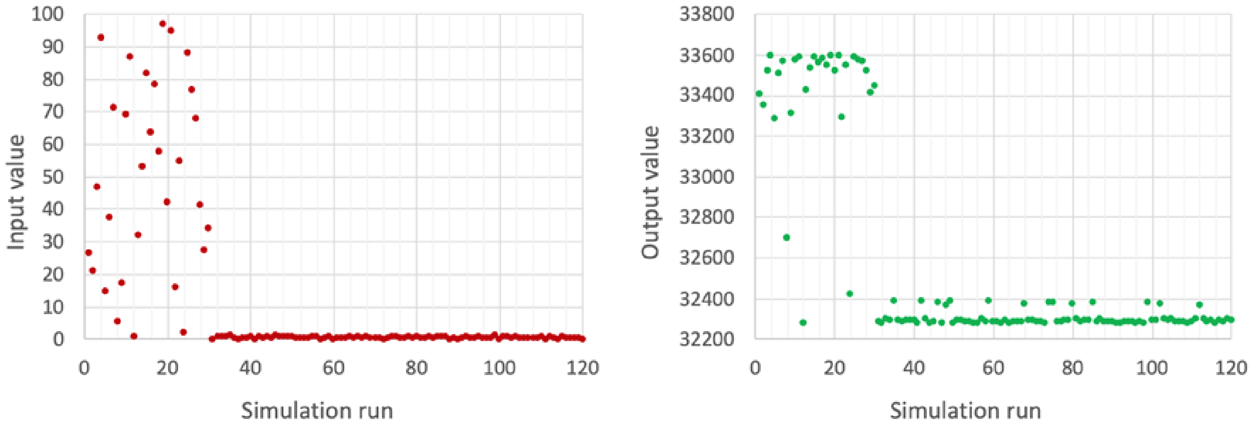

Figure 14. Distance-based leaderless inputs (grouping value) and outputs (metric value) per simulation run.

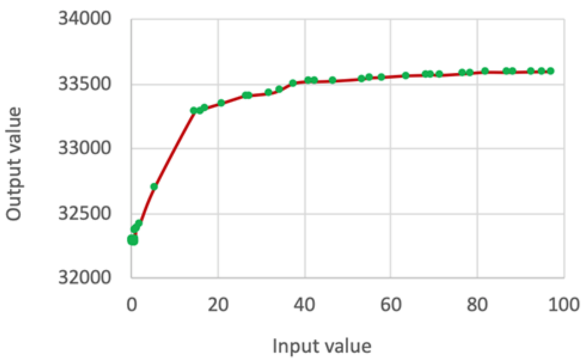

Figure 15. Distance-based leaderless I/O data.

Trajectory-based with leader Lastly, this set of runs was based on the same trajectory-based metric, however with the presence of a leader, which was again represented by setting the grouping value of one aircraft as 0, forcing it to ignore the others.

Figure 20 presents a similar dispersion, however with higher values for both inputs and outputs, leading to an also higher best value of 0.5394 at 36.6385. This means that, with the presence of a leader, the aircraft flew farther from each other, although having a larger grouping value. Again, the output value did not oscillate and was even steadier than the leaderless case.

Similarly to Figure 19, Figure 21 corresponds to a metric that prioritize formation flying, if compared to the distance-based metric. Interestingly, the presence of a leader smooths the function, which is an opposite effect of what happened with the other charts. Again, this could be due to the sampling performed through SPOT iterations.

In summary, this metric led to better results with respect to formation flying, although still giving a lot of weight to the move-to-goal behavior. This is because, when flying to the goal, the aircraft already tend to fly close to each other, since it is the same waypoint for all of them.

Analysis

To facilitate the analysis, Table 2 presents a summary of the results, adding the information of the iteration in which the best results were found. This information is gathered from the data presented in Figures 14, 16, 18, and 20, with regards to the minimum output value of each. The first conclusion that we can draw is that the presence of a leader does not affect much the cases where the distance-based metric was utilized. However, when considering the trajectory-based metric, there is a 13.6% reduction of the output value.

The reason why this happens is that, in leaderless behavior, all entities tend to fly closer, since none of them are moving directly to the goal, which is the case when the leader is present. This leads to a lower optimal grouping value (input), since the other aircraft do not have to follow a leader, that is, to compensate their trajectories in a more significant way. This is not verifiable in the distance-based case because the grouping values are already too low, being much less influenced by a null value of one of its aircraft.

This conclusion indicates that the presence of a leader may be worse for the formation with respect to the mutual support, since the leader entity has the liberty to maneuver as it wishes. At best, the presence of a leader performed in a similar way to the leaderless case, which reinforces this conclusion.

With respect to iterations, the distance-based metric achieved its best value in a faster way, which we explain by the characteristics of the curves in Figure 15 and Figure 17, which show an evident global minimum, if compared to the local minima that occurred. This is not the case in Figure 19 and Figure 21, since the minima are in a flatter area of the curves. Therefore, in the performance standpoint, the first metric is more efficient.

Another conclusion that can be drawn from the summarized data is that the distance-based metric led to a very dispersed movement of the aircraft. The reason for that was the large influence that the distances between aircraft and goal waypoint had on the overall metric value. Since these distances are much higher than the inter-agent distances (between aircraft), as a result of the way the initial scenarios were constructed, the most efficient way to minimize the metric value was to almost fly directly to the goal.

On the other hand, the trajectory-based metric forced the entities to move closer to their mean path to goal, which naturally led to a closer formation flight. This leads to the

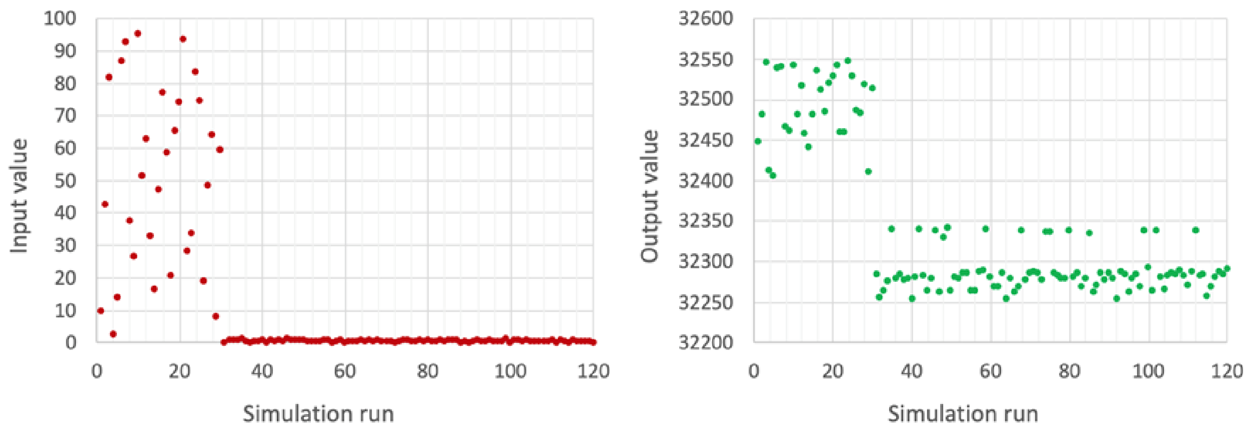

Figure 16. Distance-based with leader inputs (grouping value) and outputs (metric value) per simulation run.

Table 2. Best I/O data for each experiment.

Experiment	Best input	Best output	Iteration
Distance-based leaderless	0.6335	32278.0323	47
Distance-based with leader	0.7213	32252.9881	40
Trajectory-based leaderless	32.1681	0.4761	77
Trajectory-based with leader	36.6385	0.5394	106

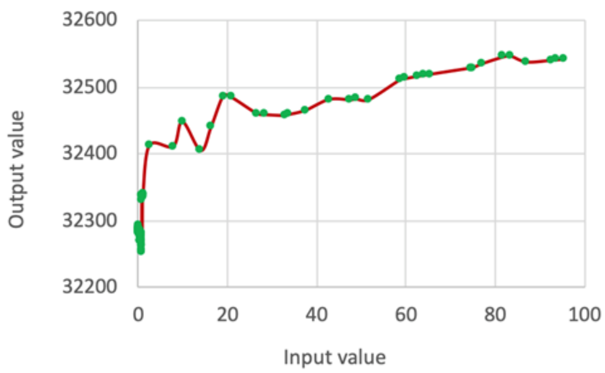

Figure 17. Distance-based with leader I/O data.

Table 3. Worst I/O data for each experiment.

Experiment	Worst input	Worst output
Distance-based leaderless	97.1481	33597.0400
Distance-based with leader	83.3377	32547.0492
Trajectory-based leaderless	2.3188	0.6525
Trajectory-based with leader	0.6572	0.7383

conclusion that the metric may be very influential on the final results and has to be carefully defined in order to encompass the main objectives of the mission. There is no way for the method to correctly prioritize some aspects of the mission other than relying on precise metrics.

Additionally, from the iteration column of Table 2 it can be seen that the SPOT methodology found the best value in a very fast way for the distance-based metric, whereas it took much longer to identify it on the trajectory-based case. This may be due to the highly unpredictable method of calculation of this metric, which changes its path at each frame, likely generating multimodality and even discontinuities in the objective function.

From the comparison between Table 2 and Table 3, we conclude that, in the case of the trajectory-based metric, the optimization method was capable of identifying an optimal value that is approximately 27% lower (better) than an arbitrarily defined grouping value could be (worst case). When considering the distance-based metric, the reduction was not so significative, which may be another indication that this metric does not encompasses some of the problem complexities, such as aforementioned.

This reduction of the metric value (objective function) indicates that the use of an optimization method is fundamental to determine robust grouping values. Therefore, if the commander's intent is to prioritize something other than mutual support, another metric should be established and further optimized, most likely generating completely different input suggestions.

Comparing the moving average trendline (period 2) for each metric, we see that there are similarities in the general profile for both of them pairwise (Figure 22 and Figure 23), being altered due to the sampling performed throughout SPOT iterations. This was expected by us, since the metrics guide the process in the same way, with the entities behaving in a slightly different way to adapt to the leader when it is present.

Although the leaderless case indicated that the resemblance with the reality is not necessarily better for the entities' performance in simulated environments, we deemed important to confront the results obtained with operational knowledge within the Air Force. We did that by asking Brazilian Air Force pilots to check whether the generated trajectories were compatible with what they see in practice.

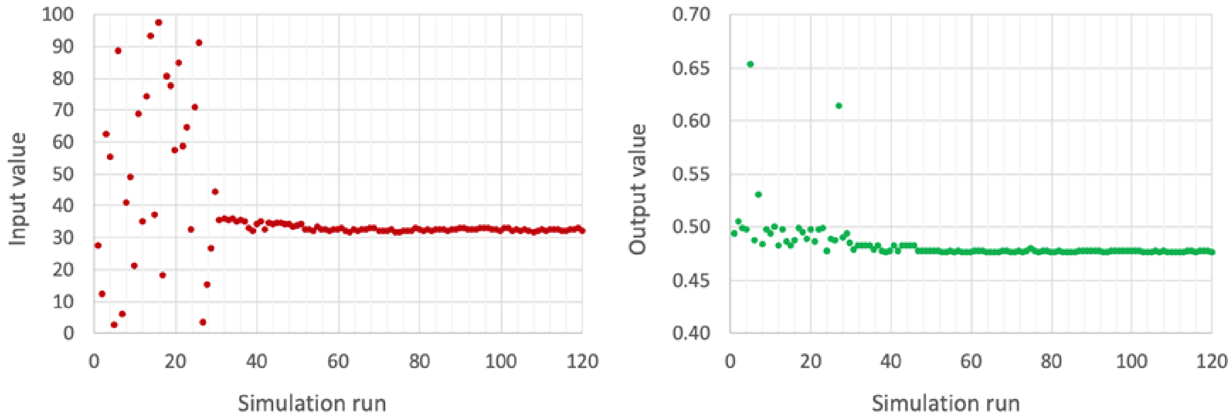

Figure 18. Trajectory-based leaderless inputs (grouping value) and outputs (metric value) per simulation run.

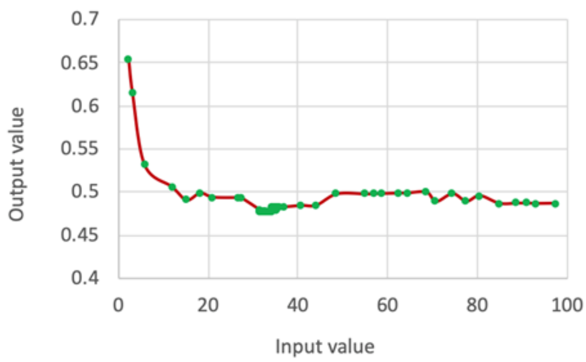

Figure 19. Aircraft trajectory for grouping value of 90.

From the pilots' perspective, the use of artificial potential fields was able to guide the entities in trajectories that were very similar to what they would achieve with real aircraft. The movements were smooth and did not present instabilities that a more direct control method could generate. This smoothness was sometimes even better than in reality, since oftentimes the pilots have to utilize high-performance maneuvers to adapt to their leader's movement, with higher accelerations and turn rates.

Since the system performed calculations for each entity at every simulation frame, the transitions were more subtle than a human being could do. This is because the machine is able to perform much more calculations per frame, and that the pilots have much more concerns to think about during the flight, including their own safety.

Another aspect pointed out by them is that the existence of a mutual support metric could guide them to fly better. During operations, pilots generally must follow their leader's commands, which are based on the leader's training and experience. It is not possible for the leader neither to precisely determine the best turn rates and accelerations for each member of its formation, nor to evaluate their trajectories with respect to mutual support. If such a metric were present in their displays, for instance, they could also try to adapt to them in a more straightforward way.

We stress that the metrics would have to be carefully defined and evaluated in order to lead to the correct conclusion, even more when speaking of real operations. Simulations could be used for that purpose, as we propose in this work. However, the conclusions we achieved are conditioned to the models within the simulation framework, which are not completely equivalent to the reality and may generate results that are not exactly what the pilots would verify if following the very same procedures while flying.

For instance, the number of degrees of freedom of the models we used is 4, but if they were to perfectly resemble reality it should be 6. That causes the simulated aircraft to not being able to drift, among many other implications that can alter their behavior if compared to real flight, Another example limitation of our approach resides on the fact that the aircraft might present an oscillation along its trajectory, as 13 shows. This is a limitation resulting from the proposed formation control method, whereas the previous example is related to the simulation itself.

Conclusions and future work

The first conclusion that we can draw from our work is that the use of artificial potential fields proved itself as an effective approach to formation flying of fixed-wing military aircraft within a simulated scenario. This was evaluated by Brazilian Air Force pilots, who observed the aircraft behavior resembling what they have seen in real military operations that involved formation flying.

Artificial potential fields were a simple way to encompass different behaviors, being of relatively easy implementation if compared to more detailed cognitive models, as well as less dependent on input data as machine learning techniques would be. In addition, their parametrization allowed for efficiently optimizing the represented behavior through simulation optimization.

Although presenting a computational challenge due to its high costs (time), the use of simulation as an objective function allowed for a more applicable set of results, since they are based on high-fidelity models of military systems. Even so, using a computer of regular configuration, the method did not take too long to achieve good results.

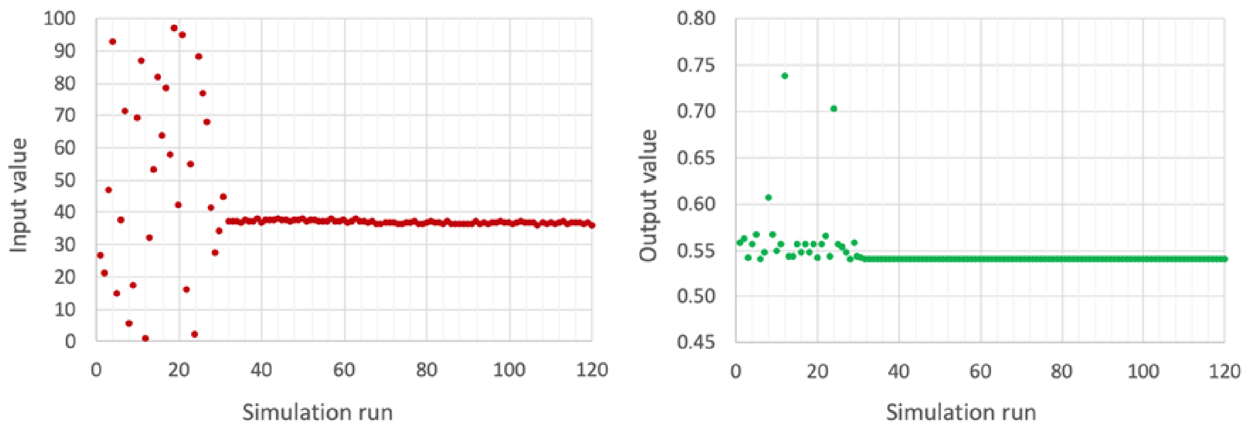

Figure 20. Trajectory-based with leader inputs (grouping value) and outputs (metric value) per simulation run.

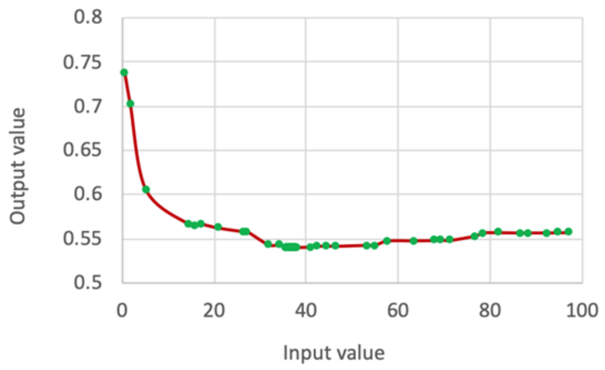

Figure 21. Trajectory-based with leader I/O data.

With the optimization, we were able to attain 27% gains relative to one of the metrics utilized (trajectory-based) if compared to arbitrarily selected parameters (worst case). However, when considering the other metric, the gains were much less expressive. That raises the issue of carefully selecting the metrics, which guide the optimization process.

The distance-based metric, which initially seemed to be a rather straightforward way to evaluate the degree of mission success – with respect to both formation flying and moving to goal – proved itself as a less effective form of measurement. Since the distances to the goal were higher than the distance between aircraft, they tended to ignore each other, flying directly to the objective waypoint. On the other hand, the trajectory-based metric forced the aircraft to fly closer to each other, yielding better results. In summary, we see a need to carefully define the metrics, so that the results satisfactorily represent the reality.

Nevertheless, sometimes the reality is not necessarily the goal of artificial intelligence. For instance, as the results of this work showed, the presence of a leader – what is the usual within real military operations – caused the formation to perform in a worse way if compared to the leaderless approach. This is a strength of this kind of methodologies, which allow analysts to test non-conventional behaviors and compare them with what is most often done in reality.

Another strength of the presented approach is that the utilized computational power can suit the available budget due to its sequential nature. This grants the flexibility that decision-makers may need when dealing with urgent matters, that should be simulated as quick as possible.

Finally, since the initial application of artificial potential fields was mainly made based on empirical parameters, an optimization method brought more robustness to the model. Through the optimization we were able to define good parameters that would work for more generic scenarios.

Future work includes applying the optimization methodology to other military missions in order to better understand its own performance, which could also be optimized by the same method (SPO). This would require new metrics, which could be developed for the same problem addressed as well. For instance, a better way to evaluate the mutual support would be highly valuable.

Parallelization techniques could also be studied, along with more complex budget allocation techniques. This is interesting considering that other scenarios could present higher levels of complexity with, for instance, weapon's effects. Events as such could also lead to more complex maneuvers, which should be encompassed by the artificial potential fields or by other movement techniques.

Acronyms

ASA *Ambiente de Simulação Aeroespacial*.

BFGS Broyden-Fletcher-Goldfarb-Shanno.

C4ISR Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance.

DACE Design and Analysis of Computational Experiments.

DASE Design and Analysis of Simulation Experiments.

DoE Design of Experiments.

EAAGLES Extensible Architecture for the Analysis and Generation of Linked Simulations.

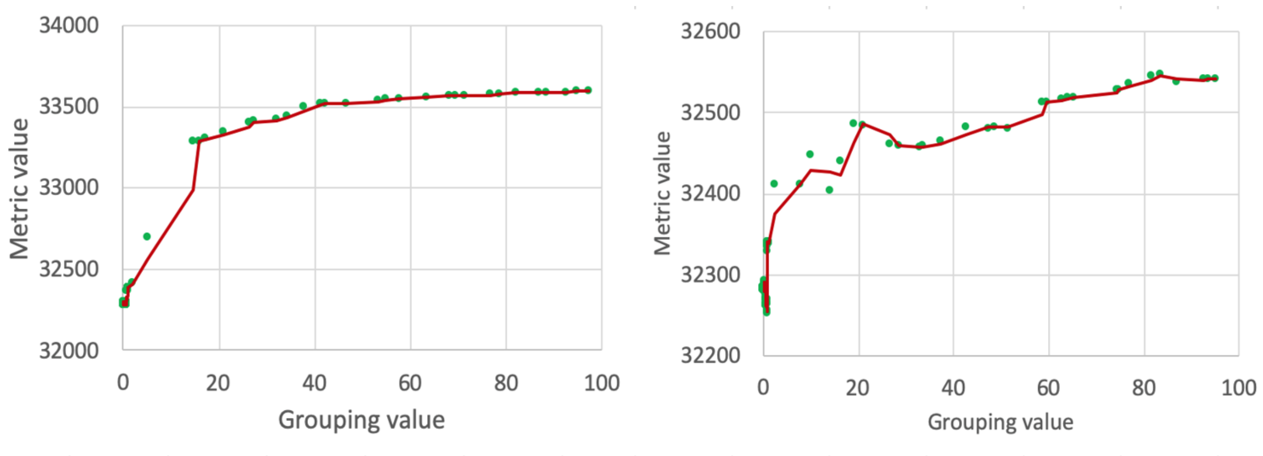

Figure 22. Moving average trendline for distance-based I/O data: leaderless (left) and with leader (right).

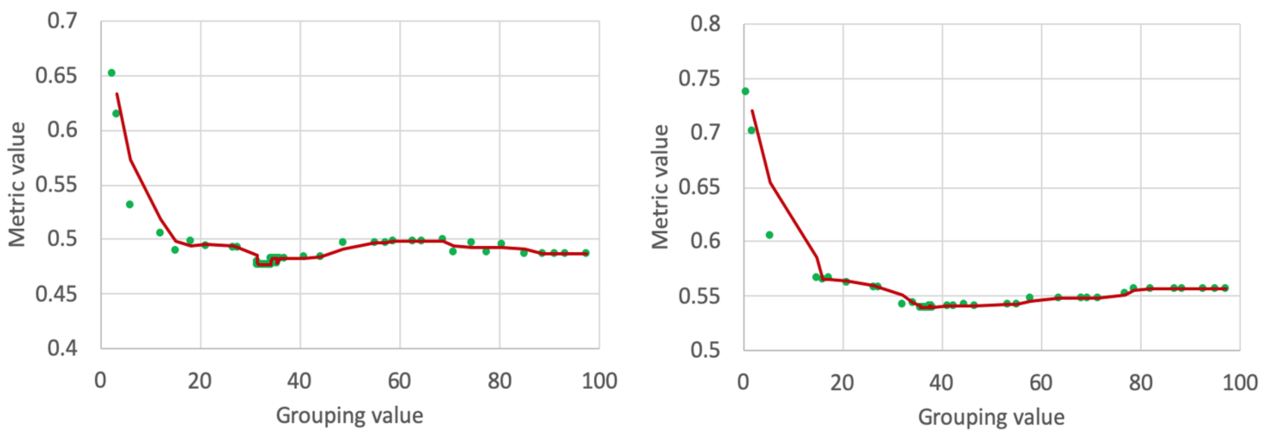

Figure 23. Moving average trendline for trajectory-based I/O data: leaderless (left) and with leader (right).

EAPF Evolutionary Artificial Potential Fields.

EGO Efficient Global Optimization.

EI Expected Improvement.

FAA Federal Aviation Administration.

FAOR Fighter Area of Responsibility.

I/O Input/Output.

IEAv *Instituto de Estudos Avançados*.

LHD Latin Hypercube Design.

LHS Latin Hypercube Sampling.

LVC Live, virtual, and constructive.

MIXR Mixed Reality Simulation Platform.

RWR Radar Warning Receiver.

SO Simulation Optimization.

SPO Sequential Parameter Optimization.

SPOT Sequential Parameter Optimization Toolbox.

UBF Unified Behavior Framework.

References

1. Murakami H, Niizato T and Gunji YP. Emergence of a coherent and cohesive swarm based on mutual anticipation. *Scientific reports* 2017; 7: 46447.
2. Chatters JC. Hunter-gatherer adaptations and assemblage structure. *Journal of anthropological archaeology* 1987; 6(4): 336–375.
3. Larson J, Isihara P, Flores G et al. A priori assessment of a smart-navigated unmanned aerial vehicle disaster cargo fleet. *SIMULATION* 2020; 96(8): 641–653.
4. Air Education and Training Command. *Employment Fundamentals T-38C/Introduction to Fighter Fundamentals (IFF)*, 2020. URL https://static.e-publishing.af.mil/production/1/aetc/publication/aetcttp11-1/aetcttp11-1.pdf.
5. Davis PK and Henninger AE. *Analysis, analysis practices, and implications for modeling and simulation*, volume 176. Rand Corporation, 2007.
6. Banks J. *Discrete event system simulation*. Pearson Education India, 2005.
7. Kim YJ, Mavris D and Fujimoto R. Time-and space-parallel simulation of air traffic networks. *Simulation* 2019; 95(12): 1213–1228.
8. Abielmona R, Falcon R, Zincir-Heywood N et al. Recent advances in computational intelligence in defense and security. In *Recent Advances in Computational Intelligence in Defense*

and Security. Springer, 2016. pp. 1–9.
9. Tolk A. *Engineering principles of combat modeling and distributed simulation*. John Wiley & Sons, 2012.
10. Washburn AR, Kress M et al. *Combat modeling*, volume 139. Springer, 2009.
11. Department of Defense, Alexandria, VA. *Modeling and simulation (M&S) glossary*, 2011.
12. Hodson DD and Hill RR. The art and science of live, virtual, and constructive simulation for test and analysis. *The Journal of Defense Modeling and Simulation* 2014; 11(2): 77–89.
13. Federal Aviation Administration, Washington, DC. *Pilot/controller glossary*, 2019.
14. Reynolds C. Boids background and update. *http://www red3d com/cwr/boids/* 2001; .
15. von Mammen S and Jacob C. The spatiality of swarms — quantitative analysis of dynamic interaction networks. In *Proceedings of Artificial Life XI*. Winchester, UK: MIT Press, pp. 662–669. URL http://vonmammen.org/publications/2008-Alife-ReengineerSwarms.pdf.
16. Fu MC. Optimization via simulation: A review. *Annals of operations research* 1994; 53(1): 199–247.
17. Amaran S, Sahinidis NV, Sharda B et al. Simulation optimization: a review of algorithms and applications. *Annals of Operations Research* 2016; 240(1): 351–380.
18. Carson Y and Maria A. Simulation optimization: methods and applications. In *Proceedings of the 29th conference on Winter simulation*. pp. 118–126.
19. Bartz-Beielstein T, Chiarandini M, Paquete L et al. *Experimental methods for the analysis of optimization algorithms*. Springer, 2010.
20. Antony J. *Design of experiments for engineers and scientists*. Elsevier, 2014.
21. Simpson T, Toropov V, Balabanov V et al. Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come-or not. In *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*. p. 5802.
22. Kleijnen JP. Design and analysis of simulation experiments. In *International Workshop on Simulation*. Springer, pp. 3–22.
23. Barton RR. Simulation optimization using metamodels. In *Proceedings of the 2009 winter simulation conference (WSC)*. IEEE, pp. 230–238.
24. Cressie N. The origins of kriging. *Mathematical geology* 1990; 22(3): 239–252.
25. Nelder JA and Wedderburn RW. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)* 1972; 135(3): 370–384.
26. Quan N, Yin J, Ng SH et al. Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints. *Iie Transactions* 2013; 45(7): 763–780.
27. Jones DR, Schonlau M and Welch WJ. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 1998; 13(4): 455–492.
28. Bartz-Beielstein T, Stork J, Zaefferer M et al. *Package 'SPOT'*, 2019. URL https://cran.r-project.org/web/packages/SPOT/index.html.
29. Bartz-Beielstein T, Lasarczyk CW and Preuß M. Sequential parameter optimization. In *2005 IEEE congress on evolutionary computation*, volume 1. IEEE, pp. 773–780.
30. Barnes LE. *A Potential Field Based Formation Control Methodology for Robot Swarms*. PhD Thesis, University of South Florida, USA, 2008. AAI3326047.
31. Vadakkepat P, Tan KC and Ming-Liang W. Evolutionary artificial potential fields and their application in real time robot path planning. In *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, volume 1. IEEE, pp. 256–263.
32. Zhang Q, Chen D and Chen T. An obstacle avoidance method of soccer robot based on evolutionary artificial potential field. *Energy Procedia* 2012; 16: 1792–1798.
33. Scharf DP, Hadaegh FY and Ploen SR. A survey of spacecraft formation flying guidance and control. part ii: control. In *Proceedings of the 2004 American control conference*, volume 4. IEEE, pp. 2976–2985.
34. Oh KK, Park MC and Ahn HS. A survey of multi-agent formation control. *Automatica* 2015; 53: 424–440.
35. Lin X, Wang ZQ and Chen XY. Path planning with improved artificial potential field method based on decision tree. In *2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. IEEE, pp. 1–5.
36. Huang Y, Ding H, Zhang Y et al. A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach. *IEEE Transactions on Industrial Electronics* 2019; 67(2): 1376–1386.
37. Sudhakara P, Ganapathy V and Sundaran K. Route planning of a wheeled mobile robot (wmr) using enhanced artificial potential field (e-apf) method. In *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE, pp. 12–15.
38. Costa AN and Geraldo D. Modelagem comportamental para simulacao construtiva de combate alem do alcance visual. In *Simpósio de Aplicações Operacionais em Áreas de Defesa*. São José dos Campos: ITA, pp. 165–170.
39. Fletcher J. Education and training technology in the military. *science* 2009; 323(5910): 72–75.
40. Worth DB, Woolley BG and Hodson DD. Swarmsim: a framework for modeling swarming unmanned aerial vehicles using hardware-in-the-loop. *The Journal of Defense Modeling and Simulation* 2021; 18(2): 105–124.
41. Berndt J. Jsbsim: An open source flight dynamics model in c++. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*. p. 4923.
42. Berndt J and De Marco A. Progress on and usage of the open source flight dynamics model software library, jsbsim. In *AIAA modeling and simulation technologies conference*. p. 5699.
43. Stevens BL, Lewis FL and Johnson EN. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.
44. Kamrud AJ, Hodson DD, Peterson GL et al. Unified behavior framework in discrete event simulation systems. *The Journal of Defense Modeling and Simulation* 2017; 14(4): 471–481.
45. Kamrud AJ. *Unified Behavior Framework for Discrete Event Simulation Systems*. Master's Thesis, Air Force Institute of Technology, 2015.
46. Roberson DM. The unified behavior framework for the simulation of autonomous agents. Technical report, Air Force Institute of Technology, Wright-Patterson AFB, OH, 2015.
47. Woolley BG. *Unified behavior framework for reactive robot control in real-time systems*. Master's Thesis, Air Force

Institute of Technology, 2007.

48. McKay MD, Beckman RJ and Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 2000; 42(1): 55–61.
49. Santner TJ, Williams BJ, Notz WI et al. *The design and analysis of computer experiments*, volume 1. Springer, 2003.
50. Deutsch JL and Deutsch CV. Latin hypercube sampling with multidimensional uniformity. *Journal of Statistical Planning and Inference* 2012; 142(3): 763–772.
51. The R Project for Statistical Computing. URL https://www.r-project.org/. Acessed: 2021-01-22.
52. Zhu C, Byrd RH, Lu P et al. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)* 1997; 23(4): 550–560.