http://journal_aeronautical.af-acad.bg/.bg

ISSN 2815-293X
ISSN 2815-2948

Aeronautical Research and Development

4

"*Georgi Benkovski*"

Bulgarian Air Force Academy

Там, където поникват криле...

August , 2025

Aeronautical Research and Development Volume 4, 2025

http://journal_aeronautical.af-acad.bg

ISSN 2815-293X
ISSN 2815-2948

Aeronautical Research and Development

4

"*Georgi Benkovski*"

Bulgarian Air Force Academy

August, 2025

AERONAUTICAL RESEARCH AND DEVELOPMENT
Volume 4 ▪ Dolna Mitropolia ▪ 2025

"Georgi Benkovski" Bulgarian Air Force Academy

EDITORIAL BOARD

Prof. Dr. Asen Angelov Marinov – Editor in Chief

Assoc. Prof. Dr. Martin Milenov Kambushev, Assoc. Prof. Dr. Georgi Valentinov Stanchev, Assoc. Prof. Dr. Milen Atanasov Atanasov, Assoc. Prof. Dr. Lubomir Vasilev Mitov, Assoc. Prof. Dr. Stoyko Ognyanov Stoykov, Prof. Dr. Manush Petkov Hristov, Assoc. Prof. Dr. Toshko Marinov Marinov, Assoc. Prof. Dr. Vladimir Svetoslavov Savov, Assoc. Prof. Dr. Kolyo Penchev Kolev, Assoc. Prof. Dr. Danyo Marinov Lalov, Assoc. Prof. Dr. Kitka Todorova Toncheva, Assoc. Prof. Dr.Svetlana Yordanova Nedelcheva, Prof. Dr. Marin Simeonov Marinov, Assoc. Prof. Dr. Penyo Gospodinov Penev, Assoc. Prof. Dr. Konstantin Kirilov Metodiev, Dr. Nikolay Georgiev Kanchev.

Address

AERONAUTICAL RESEARCH AND DEVELOPMENT

Dolna Mitropolia, 5855

"St. St. Cyril and Methodius" №1, St.

asen_aerodynamics@abv.bg

Language Editor

Yavor Varbanov

"Georgi Benkovski"

Bulgarian Air Force Academy, 2025

Aeronautical Research and Development

4

Dolna Mitropolia, 2025

Contents

Programming language Energia-2.1 – implementing an extension of the capabilities for scientific and engineering calculations

Yoto Georgiev Georgiev

"Georgi Benkovski" Air Force Academy, Faculty of Aviation, Dolna Mitropolia, Bulgaria, y_g_g@abv.bg

Abstract: This paper presents a new revision of programing language Energia-2.1, in which are introduced new features – arrays.

Keywords: *programming language Energia, arrays, scientific research, computational physics*

1. Introduction

A programming language is a way for computer engineers to communicate with computers. Programming languages consist of a set of rules that allow string values to be converted into various ways, to generate machine code for the corresponding microprocessor. The potential for performing a given task is realized with the help of programming languages, which instruct how to perform a given task. Computer physical simulations are one of the foundations of modern science and engineering. In computer science and programming languages arrays are one of the most fundamental data structures. An array in computer science is an ordered sequence of elements of the same type. An individual element of an array is specified by the name of the entire array, followed by the sequential number (index) of the element. Arrays are constructed in different ways in different programming languages. The goal of creating the computer language *Energia* and its interpreter is to quickly and easily create program source code for engineering and scientific calculations. The designers' goals were to create a language that is simpler, yet more flexible, than Fortran and useful and processed with reasonable efficiency as well as definable with rigorous formality. The idea was to create a programing language counterpart to the French „*Gibiane*".

Normally, computer programs are written in a high-level language that must be translated into machine language in order to be understood by the computer and executed by the processor. This conversion is done by specialized programs and the process is called translation. Depending on how they work, they can be compilers and interpreters. The difference between an interpreter and a compiler can be summarized as follows: an interpreter retains control of the machine and performs actions prescribed by its input data, which is either program text or intermediate code. In general, an interpreter is a program that executes other programs. On the other hand, a compiler produces code that results in a stand-alone computer program.

The interpreter for programming language *Energia* was initially written and compiled in the Assembly language (GNU Binutils), as this is one of the languages widely used to develop interpreters and compilers. In the first version of the Energia-1.0 interpreter, the main components of the computer language were defined. In the next version, *Energia-1.1*, subroutines were added as the main building blocks of programs [1]. Subroutines are the building blocks of programs in the *Energia* programing language. They are blocks of instructions that are executed when called from another point in the program. In Energia-2.0, the interpreter was "rewritten" from the assembly language to the high-level programming language C++. This new version provides the same functionality, realized through a different approach [2]. The reason for that was the limited portability of the interpreter *Energia-1.1* programmed in assembly language, and the goal was to expand its applicability. The implementation of the interpreter in both Assembler and C++ uses the method of recursive-descent parsing for syntactic analysis of the program code. This method is based on a hierarchy of recursive functions. There are several notations used to describe the syntax of programming languages: Backus-Naur Form, Wirth syntax notation, Regex, Extended Backus-Naur Form. The syntax of programming language *Energia* is described with a Backus-Naur Form. Backus-Naur Form (BNF) consist of three components: a set of non-terminal symbols, a set of terminal symbols, and rules for replacing non-terminal symbols with a sequence of symbols. In Backus-Naur Form developed by John Backus and Peter Naur, grammar rules are expressed in the form of production rules that define how symbols are constructed. Each production

consists of a left-hand side (non-terminal) and a right-hand side (a combination of terminals and non-terminals). The interpreter for *Energia-2.0* is compiled with statically linked libraries in the C++ language. In the new version *Energia-2.1* of the programming language are introduced new features – arrays. *Energia-2.1* employs a general data type concept. In *Energia-2.1* variables, arrays and procedures are not quantities which are declared by type but only named by identifiers.

Scientific computing has evolved from being the only kind of computing that existed, to being a small part of how and why computers are used today. C++ and Fortran are the main computer language for use in scientific computing. Fortran is entirely intended as a computer language for programming numerical methods and is the undisputed leader in this category. Other programming languages used for computer simulations are Simula (C++ is a combination of the C programming language and the Simula 67), Modelica, Dymola(Dynamic Modeling Language), GASP-V, Omola, ObjectMath, SIDOPS+, VHDL-AMS, ASCEND, DYNAMO, ASTAP.

2. Arrays in computer science

Arrays are one of the most foundational data structures in computational science and programming languages. They provide an efficient way to store and access data in adjacent memory locations, making them essential in a wide range of programming applications. At a high level, an array is a collection of elements of same type and identified by an index used as a key. The key characteristics of arrays include indexing and contiguous memory as the elements are stored in contiguous memory locations, ensuring fast access via pointers or direct memory addresses [3]. The development of computer modeling of various complex and sophisticated dynamic systems since the mid-twentieth century has been very rapid and has been accompanied by an increase in the calculation and storage capabilities of computing technology. Modeling of complex physical processes is one of the promising areas in computer informatics, in which there is scope for further development and research in the coming decades. Computers are used to simulate systems ranging from nuclear reactors and airplanes to cardiovascular processes in medicine. The advantages of simulation modeling over experiments are the relatively low cost, high speed, completeness of information and the possibility of mathematical modeling of real conditions. The use of computer models for structural optimization of aircraft structures allows the construction of lighter aircraft [4, 5]. Computer models are used both for simulations of separated units [6] and the control of given objects [7], as well as for modeling processes such as turbulence during aircraft flights [8, 9].

The concept of an array can be traced back to the 1960s. Early programming languages such as Fortran did not support array structures. The first high-level programming language Fortran, was designed for scientific and engineering calculation. Although Fortran did not support array structures, it allowed the use of indexed variables, which simulated the behavior of arrays. In next versions of Fortran, arrays were added for supporting multi-dimensional arrays and array manipulation operations.

Programing language „*Algorithmic Language 1968“*, simply known as *Algol-68*, introduced a reference model of array and matrix slicing and concurrency. It was the first language to define the structure of algorithms and introduced concepts like block-structured programming. ALGOL's syntax influenced the design of many later languages, including C and Pascal. *Algol-68* featured arrays in a way that allowed programmers to define multi-dimensional arrays and access elements with indices.

In programming language ***C,*** created in 1972 by Dennis MacAlistair Ritchie, arrays became a core feature of the language and were integral to low-level memory management. ***C*** introduced the concept of "array pointers," where arrays were viewed as contiguous blocks of memory. In the 1980s, the ***C*** programming language played a key role in making arrays a fundamental part of modern programming. In computer language C++ was introduced the **std::vector**, a dynamic array (a resizable array) implementation that can grow or shrink in size during execution, offering a more flexible alternative to fixed-size arrays. Arrays are declared in different ways depending on the programming language. In the list below is shown declaration of array named „*Masiv*” that contains four integers in several programming languages:

Fortran	`INTEGER, DIMENSION(4) :: Masiv = [1, 2, 3, 4]`
Python	`Masiv = [1, 2, 3, 4]`
JavaScript	`let Masiv = [1, 2, 3, 4]`
Java	`int[]Masiv = {1, 2, 3, 4};`
C	`int Masiv[4] = {1, 2, 3, 4};`

C++	`int Masiv[4] = {1, 2, 3, 4};`
C#	`int[]Masiv = {1, 2, 3, 4};`
Ruby	`Masiv = [1, 2, 3, 4]`
Swift	`var Masiv: [Int] = [1, 2, 3, 4]`
Go	`var Masiv = [4]int{1, 2, 3, 4}`
PHP	`$Masiv = [1, 2, 3, 4];`
Kotlin	`val Masiv = arrayOf(1, 2, 3, 4)`
Rust	`let Masiv = [1, 2, 3, 4];`
R	`Masiv <- c(1, 2, 3, 4)`
Perl	`@Masiv = (1, 2, 3, 4);`
Lua	`Masiv = {1, 2, 3, 4}`
Haskell	`Masiv = array (1, 4) [(1, 1), (2, 2), (3, 3), (4, 4)]`
Scala	`val Masiv = Array(1, 2, 3, 4)`
Objective-C	`NSArray *Masiv = @[@1, @2, @3, @4];`
MATLAB	`Masiv = [1, 2, 3, 4];`
Elixir	`Masiv = [1, 2, 3, 4]`
Julia	`Masiv = [1, 2, 3, 4]`
TypeScript	`let Masiv: number[] = [1, 2, 3, 4];`
Swift	`var Masiv: [Int] = [1, 2, 3, 4]`
TypeScript	`let Masiv: number[] = [1, 2, 3, 4];`
Bash	`Masiv =(1 2 3 4)`
Racket	`(define Masiv (vector 1 2 3 4))`
Smalltalk	`Masiv := #(1 2 3 4).`
ActionScript	`var arr:Array = [1, 2, 3, 4];`
VHDL	`signal Masiv : int_array := (1, 2, 3, 4);`
Zig	`const Masiv: [4]i32 = [4]i32{1, 2, 3, 4};`

Each programming language has its own syntax and conventions, especially regarding whether array structures are static or dynamic in size.

3. Arrays in Energia-2.1

Vectors and matrices are widely used in scientific and engineering simulations, and their implementation in computer languages is through array-type data structures – one-dimensional arrays for vectors and two-dimensional arrays for matrices. Vectors and matrices are fundamental objects that play a crucial role in various fields such as physics, engineering, computer science, and more. The addition of the array structure in the Energy computer language is therefore of interest. In programming language *Energia-2.1* an array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier. Like a variable, an array must first be declared before it is used in the computer program. A typical declaration for an array in *Energia-2.1* is the size in square brackets, followed by the name of the array: [elements/array size] *name of the array*. Unlike other computer languages, in *Energia-2.1* it is not necessary to specify the type of elements in arrays because all arrays are of type double-precision floating-point numbers only. By default arrays are left uninitialized. This means that only name is determined and none of its elements is set to any particular value and their contents are undetermined at the point the array is declared. Indexing of an array starts from 0. It means that the first element is stored at the 0th index, the second at 1st, and so on. The reason why index starts from zero is that index in programming language *Energia-2.1* is used as an offset. Elements of an array can be accessed using their indices. From a mathematical perspective, 0-based indexing is often more natural and consistent in many algorithms and theories, especially those involving sequences, matrices, or graphs, where 0-based numbering can simplify the logic. As arrays in *Energia-2.1* are essentially contiguous blocks of memory, when you access an element in an array, the address of the first element is a base address. To access the **i**-th element, the interpreter calculates its memory address as *base_address + i * size_of_element*. If the array starts at index 0, this simplifies the address calculation.

The values of any of the elements in an array can be accessed just like the value of a regular variable of type double-precision floating-point number. The syntax is: name[index].

Example for accessing third element in array called "*ARR1"* and making variable "*b1*" holding the value of it:

```
b1 = ARR1[2]
```

In the next example to store the value 12.5 in the second element of "*ARR1"*, we could write:

```
ARR1[1] = 12.5
```

Elements of an array can be accessed by specifying the name of the array, then the index of the element enclosed in the array subscript operator "**[]**".

The below program illustrates how we can declare an array called „*Masiv"* with 4 elements in *Energia-2.1* and initialization of its values.

```
! Example of arrays
[10]Masiv_A1;
Masiv_A1 [0] = 1.2
Masiv_A1 [1] = 5.4
Masiv_A1 [2] = 3.3
Masiv_A1 [3] = 7.2
```

4. Validation of arrays in Energia-2.1 – discussion of the results

The main purpose of validation and verification is to prove the correctness of the program or, if errors are found, to correct them [10]. Validating computer code means evaluating the accuracy of the calculated values from the code against other results. Below are shown results from validation of array structures in *Energia-2.1*. The source code and the results from calculations are represented:

```
! Example of arrays in Energia-2.1

<< "Test arrays"
<<_ "Test arrays"
<<_ n
<< n  ! new line

: x1 = 10.5
: b1 = 12
: c1 = 1

: d2 = 1

: radius = 20
: plost = 0
: pi = 3.14


[8]Masiv_C;
[12]Masiv_A1;

[8]Masiv_B;

Masiv_A1 [0] = 3.2

Masiv_A1 [1] = ((( x1 + b1 + 3 ) - c1 ) / 2.0 )

c1 = ( c1 + b1 )

Masiv_A1 [2] = ( c1 + 1.0 )

<< "Masiv_A1[2] = "
<< Masiv_A1[2];
```

```
<< n

<< "c1 = "
<< c1;

<< "x1+c1 = "
<< ( x1 + c1 )

<< n

<< "Enter radius: "
>> radius;

plost = ( pi * ( radius ^ 2.0 ) )

Masiv_A1 [3] = ( plost + 5.0 )

<< "plost = "

<< plost;

d2 = Masiv_A1[2]

<< n

<< "d2="
<< d2;

<< n

<< "Masiv_A1[0] = "
<< Masiv_A1[0];

<< "Masiv_A1[1] = "
<< Masiv_A1[1];

<< "Masiv_A1[2] = "
<< Masiv_A1[2];

<< "Masiv_A1[3] = "
<< Masiv_A1[3];

Masiv_A1 [12] = ( pi * 10 )

<< "Masiv_A1[12] = "
<< Masiv_A1[12];

<< n
<< n

Masiv_B [0] = ( Masiv_A1[0] + Masiv_A1[1] )

<< "Masiv_B[0] = "
<< Masiv_B[0];

Masiv_C [0] = ( Masiv_A1[0] + 3 )

<< "Masiv_C[0] = "
<< Masiv_C[0];
```

The results from calculation are shown on Figure 1:

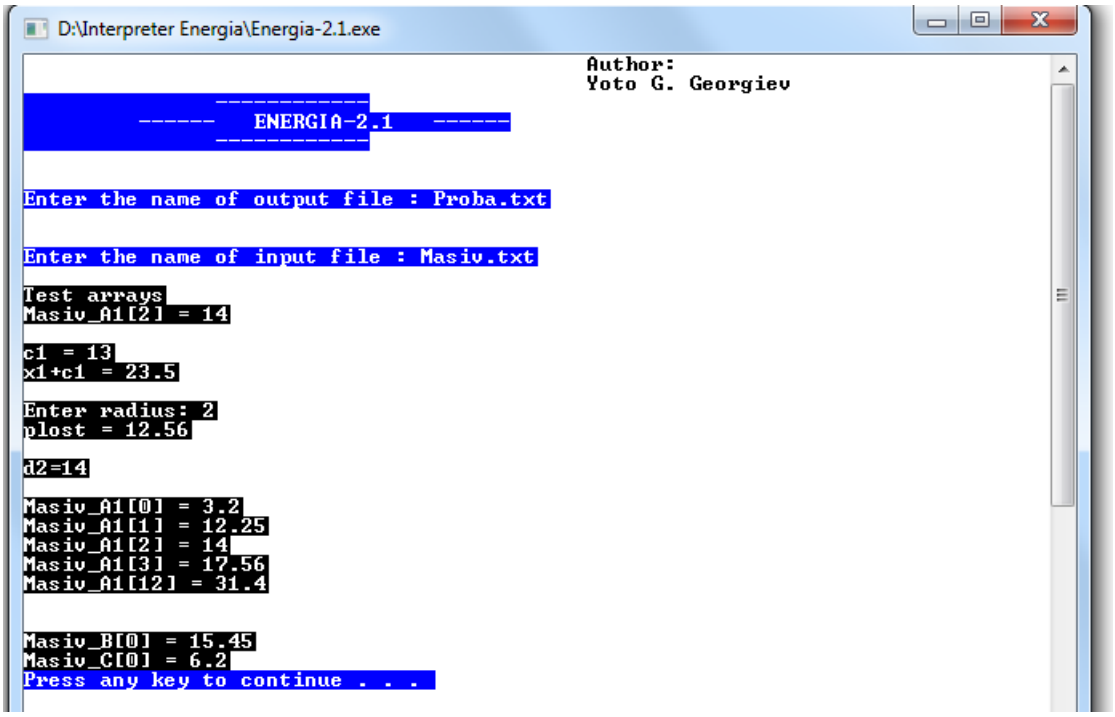


Figure 1. Results from calculation

The example above is used for validation of interpreter with added array data structures, as it can be seen that results are correct.

6. Conclusions

In this paper is presented a new revision of programing language *Energia-2.1*, in which are introduced new features – arrays. In programming languages arrays are one of the most foundational data structures. They are used for implementation of vectors and matrices in computer calculation of physical systems. Validation was made using source code with several different operations using implemented arrays in a modified version of the programming language *Energia*. The addition of array-type data structures allows the interpreter to be used for building artificial neural networks. In the past decade, artificial intelligence has attracted great attention and has been increasingly applied in the fields of science and technology. It is a new "technology" in simulation modeling. Popular machine learning algorithms include neural networks, decision trees, genetic algorithms, etc. A machine learning model is usually developed by training the model using a training data set, that is iteratively optimizing the model parameters to minimize editing errors, and evaluating the model using a test data set. The biggest attention is paid to Deep Learning - neural networks with more than three layers, which can automatically realize information extraction in a data set through their multiple processing layers. Machine learning seems to be an effective approach to developing models that can be surrogates for analytical models.

References

1. Георгиев Й., „Компютърен език Енергия-1.1“, Годишна международна научна конференция на Висше Военновъздушно Училище „Георги Бенковски“, гр. Долна Митрополия, 2024

2. Георгиев Й., „Сравнителен анализ на компютърния език Енергия“, Годишна международна научна конференция на Висше Военновъздушно Училище „Георги Бенковски“, гр. Долна Митрополия, 2024

3. Knuth, D. E., „The Art of Computer Programming, Volume 1: Fundamental Algorithms“, Addison-Wesley, 1997.

4. Kanchev N., „Numerical method for optimal design of the skin stiffeners configuration in a fixed wing structural layout“, Aeronautical Research and Development, Volume 2, 2023.
5. Kanchev N., „Fixed wing topology optimization: a numerical procedure for computational design model generation“, Aeronautical Research and Development, Volume 2, 2023.
6. Kambushev K., M. Kambushev, „Modeling of the Magnetic System of an Aircraft DC Generator“, International Conference on Electronics, Engineering Physics and Earth Science, 2023. DOI 10.1088/1742-6596/2339/1/011001
7. Calovska R., Stefan Biliderov, „Synthesis and research of a controller for microclimate stabilization in an automobile based on finite state machines“, Aeronautical Research and Development, Dolna Mitropolia, "Georgi Benkovski" Bulgarian Air Force Academy, Vol. 1, Issue: 1, 2022.
8. Илиева Д., К. Камбушев, „Моделиране на турбулентност при полет на безпилотен летателен апарат“, XXXII МНТК „АДП-2023”, Созопол, 2023, Издателство: Издателски комплекс на ТУ-София, 2023, ISSN 2682-9584, с. 96-100
9. Илиева Д., К. Камбушев, М. Камбушев, „Модел на атмосферата в урбанизиран район“, XXXII МНТК „АДП-2023”, Созопол, 2023, Издателство: Издателски комплекс на ТУ-София, 2023.
10. Georgiev Y., „Validation and verification in scientific computer simulation“, Aeronautical Research and Development, Dolna Mitropolia, "Georgi Benkovski" Bulgarian Air Force Academy, Volume 3, 2024

Компютърен език Енергия-2.1 –разширяване на възможностите за научни и инженерни изчисления

Йото Георгиев Георгиев

“Georgi Benkovski” Air Force Academy, Faculty of Aviation, Dolna Mitropolia, Bulgaria, y_g_g@abv.bg

Резюме: В статията е представена нова версия на програмния език Energia-2.1, в която са въведени нови елементи - масиви.

Programming language Energia-3.0 – A domain-specific programming language for efficient array processing and scientific computation

Yoto Georgiev Georgiev

"Georgi Benkovski" Air Force Academy, Faculty of Aviation, Dolna Mitropolia, Bulgaria, y_g_g@abv.bg

Abstract: This paper presents a compiler version of programing language *Energia*. *Energia*'s design rationale, with its emphasis on minimal syntax, reflects a strategic alignment with scientific computing needs. Built with a minimalist syntax, *Energia* supports array declarations, functional transformations, and conditional control flow. By reducing syntactic overhead, it boosts productivity and readability, while its array focus and GNU Compiler Collection integration ensure efficiency. Though limited in scope, *Energia*'s design offers a model for domain-specific language in performance-critical domains, in which are introduced new features – arrays and nodes.

Keywords: *programming language Energia, arrays, nodes, scientific research, computational physics*

1. Introduction

Modern scientific computing demands languages that balance expressiveness, performance, and ease of use. General-purpose languages like *C++* offer flexibility but often require verbose code for array operations and parallel tasks, leading to increased development time and potential errors. Domain-specific languages like *Fortran* address this by providing tailored abstractions for specific problem domains, but are too hard for learning and using. *Energia* emerges as a domain-specific language targeting efficient array processing and parallel computation, inspired by the need for concise, readable code in scientific applications such as numerical simulations and data analysis. *Energia* design emphasizes on the following goals: a succinct syntax for array operations, native support for functional transformations, and seamless integration with multi-threaded execution environments. This paper details *Energia-3.0* syntax, its compiler implementation, and an evaluation of its performance, positioning it as a practical tool for computational scientists and engineers. *Energia-3.0*, introduced for array-centric scientific processing tasks, aims to bridge the gap between high-performance computing and developer productivity. Its "**nod**e" construct democratizes parallel programming, while its limitations highlight a roadmap for growth.

The compiler for *Energia-3.0* is written in *C++* as it is the best choice for system programing. *Energia* performance parity with *C++* stems from its simple and clean constructs, while its concise syntax reduces development effort. The "**node**" construct simplifies parallel programming, making it accessible to non-experts. Its limitations such as the lack of advanced type systems and exception handling represent areas for future improvement.

2. From Energia-2.1 to Energia-3.0: Building an efficient domain-specific language compiler for scientific applications

In the initial implementation of the programming language, *Energia-1.0*, program execution was handled by an interpreter. In the next version, *Energia-1.1*, subroutines were added as the main building blocks of programs [1]. In *Energia-2.0*, the interpreter was "rewritten" from assembly language to the high-level programming language *C++*. This new version provides the same functionality, realized through a different approach [2]. In Energia-2.1 version data structure **array** was implemented as one of the most foundational data structures in computational science and programming languages. The next development of the computer language was the creation of a compiler. This was implemented with version 3.0 for faster calculations. A standout feature of *Energia-3.0* is its „**node**" construct, which defines reusable transformation pipelines for arrays (e.g., `node MyNode[5] { doubleFunc squareFunc }`). This abstraction simplifies parallel programming by encapsulating sequential function applications. In benchmarks, „node"-based transformations on large arrays leveraged multiple cores effectively, with

performance approaching hand-tuned *C++* parallel loops. This design makes parallel programming accessible to non-experts, as developers need not manage threads or synchronization explicitly – the compiler handles these details. Yet, the current implementation lacks fine-grained control over parallelism (e.g., thread count or scheduling policies) which could hinder optimization in highly specialized scenarios.

Scientific computer simulation is an imitation of the behavior of a physical system, expressed as a mathematical model of an electronic computing machine with limited mathematical capabilities. The six steps of a scientific computer simulation are conceptual modeling, mathematical modeling, discretization and algorithm selection, computer programming, numerical solution and, finally, solution representation calculating the quantities of interest from the numerical results. The structural hierarchy of components that make up scientific computer simulations is the construct that arranges the various components of a scientific computer simulation into a vertical structure of six levels: terms, assignment statements, coded physical functions, coded groups (subroutines/modules), computer code, and code sets. This hierarchy organizes scientific computer simulations. The structure of *Energia-3.0* is built on that principle for such organization of source code for simulated physical systems.

In most cases, the simulated system is defined by interconnected elements in a network (nodalization scheme). Discrete idealized models, also called lumped parameter models, have found wide application in practice. In models simulating fluid mechanics, the spatial discretization of the specific elements of the modeled system is most often done based on the finite volume approach. The commonality between the different models simulating physical phenomena is that the simulated system is divided into volumes of certain sizes - a nodalization scheme is created. This is where the idea of creating such structures as **nodes** comes from.

Energia-3.0 is integrated in *GNU Compiler Collection*. However, this tight coupling with GCC limits *Energia* portability to non-GCC environments, a trade-off that aligns with its goal of targeting scientific applications where *Fortran* and *C++* are prevalent.

3. Designing Energia program: a minimalist syntax for array-centric computing – less code, more science – power in simplicity: the Energia-3.0 domain-specific language

Energia is designed with a minimalist yet expressive syntax, drawing inspiration from functional programming paradigms and C-like constructs. Core features are array declarations, subprograms, control flow, node constructs, input/output operations and variable declarations. By examining these elements, the aim is to illustrate how *Energia* balances simplicity and functionality, making it an accessible yet effective tool for computational scientists and engineers.

Arrays are the cornerstone of *Energia-3.0*, reflecting its focus on numerical and data-intensive tasks. The syntax for declaring an array is straightforward:

Format: [size] name;

Example: `[10] numbers`;

This declares an array named numbers with 10 elements, implicitly typed as double-precision floating-point format. Unlike *C++* **double** numbers[10]; or std::**vector**<**double**> numbers(10);, *Energia*-3.0 omits the type keyword, reducing clutter. Elements are assigned using direct indexing, such as numbers [0] = 3.5. This simplicity aligns with scientific computing's frequent use of fixed-size arrays, though it lacks dynamic resizing unless explicitly annotated as a vector (with a comment like *! Originally a vector*).

Subprograms in *Energia* serve as reusable procedures, defined with a concise, functional-style syntax:

Format:

:: name() { body of subroutine}

Subprograms can also include blocks with multiple statements, offering flexibility within the minimalist framework. The „**::**" prefix distinguishes subprograms from other constructs, enhancing readability. The „**&"** operator invokes subprograms, a nod to C's function pointer syntax, enhancing familiarity.

Energia-3.0 supports basic control flow with a distinctive hashtag notation:

Conditional: #if (condition) { body }

Example: `#if (numbers[2] > 10.0) { << numbers[2]; }`

This mirrors *C++* **if**, executing the body if the condition holds.

For iterations there are two options **#do** and **#for** control flow loop structures: #for (var = start; condition; increment) { body of the loop}

Example: `#for (i = 0; i < 5; i + 1) { << i; }`

The **#do** structure is #do (condition) { body }. The **#** prefix sets control statements apart from regular commands, reducing keyword overload while maintaining clarity. Missing are advanced constructs like "switch-case", reflecting *Energia*'s focus on simplicity over exhaustive control options.

A unique feature of *Energia* - the **node** construct - enables parallel array transformations:

Format: node name[size] { subroutine1 subroutine2 ... } followed by name to invoke it.

Example:

```
:: doubleFunc(x) { x = x * 2 }
:: squareFunc(x) { x = x * x }

node MyNode[5] {
    doubleFunc
    squareFunc
}
```

This defines a pipeline where "doubleFunc" and "squareFunc" are applied sequentially to each element of "MyNode", a 5-element array. Invoking "MyNode" triggers the transformations. The syntax abstracts iteration and parallelism, making it intuitive for users unfamiliar with low-level threading, though it requires subprograms to be unary (single-parameter) functions.

Energia streamlines input/output with operator-based syntax: console output: `<< "text"` or `<< variable;` and input: `>> variable;`. The "<<" and "<<_" operators reduce the syntactic weight of input/output compared to *Fortran* and *C++* stream classes, aligning with *Energia* goal of brevity. However, they lack advanced formatting options, prioritizing ease over customization.

Variables are declared with a colon prefix, emphasizing their distinct role:

Format: : name or : name = value

Example: `: result` or `: result = 5.0`

This declares result as a double-precision floating-point format (size 64-bits), optionally initialized. Variables are global unless scoped within a subprogram, and the "**:**" prefix visually separates declarations from assignments (e.g., result = 3.5). This convention simplifies variable management but limits type diversity, as all variables are only 64-bit by design. By exclusively adopting the double-precision floating-point format data type for all variables, arrays, and subprogram operations, *Energia-3.0* prioritizes simplicity and efficiency in numerical computations over the flexibility of a multi-type system. Type systems in programming languages define the rules for data representation and manipulation, profoundly influencing a language's usability and performance. General-purpose languages like *C/C+/Python/C#,* offer rich type systems supporting integers, floats, doubles, and custom types, but this flexibility can complicate scientific computing tasks dominated by floating-point arithmetic. *Energia* which is designed for array manipulation and parallel processing, adopts a radical approach: a type system centered solely on the double-precision floating-point format (size 64-bits) data type. This double-centric design underpins its syntax and semantics, aiming to streamline numerical workflows in domains such as physics simulations, data analysis, and signal processing. We argue that while this choice aligns with scientific computing's needs, it introduces trade-offs that warrant careful consideration and future refinement. Scientific computing overwhelmingly relies on floating-point numbers, particularly the double-precision floating-point format type, for its balance of precision (64-bit IEEE 754) and range – sufficient for most numerical models, from climate simulations to quantum mechanics. *Energia* designers recognized that supporting multiple types (e.g., int, float, char) would add complexity – syntactic overhead, type conversion rules, and potential errors – without significant benefits in its target domain. By focusing on double-precision floating-point format, programming language *Energia* eliminates these burdens, aligning its type system with the practical realities of its intended use cases. Goals are: 1. Simplicity – remove type declarations from syntax, reducing cognitive load and code length; 2. Efficiency – optimize for floating-point operations, leveraging hardware support for double; 3. Consistency – ensure uniform behavior across arrays, variables, and subprograms, minimizing surprises. This rationale mirrors minimalist programming languages like MATLAB [3, 4, 5] which prioritize numeric types, but *Energia* takes it further by excluding all alternatives. We have syntactic clarity – omitting type declarations reduces code by 10-20% compared to *Fortran* and *C++*

equivalents. And it also enhances performance – uniform use of double aligns with CPU floating-point units, avoiding type conversion overhead. Eliminating type mismatches (e.g., *int* vs. *double* in *C++*) prevents a common scientific programming pitfall, as all operations assume double-precision floating-point format.

Energia-3.0 syntax is a deliberate exercise in minimalism, tailored to scientific computing's needs. Arrays are declared with ease, subprograms are concise yet functional, control flow is streamlined, nodes enable parallelism, and input/output is simplified – all unified by a 64-bit–centric type system. While this focus limits generality (e.g., no integers or exceptions), it empowers users to write efficient, clear code for array-based tasks. As domain-specific language, *Energia-3*.0 syntax offers a glimpse into how domain-specific design can enhance productivity without sacrificing performance, making it a valuable tool for its intended audience.

4. Evaluation of Energia-3.0 – benchmarking against C++

Energia-3.0 designed for scientific computing, targets array-centric tasks and parallel computation with a minimalist syntax and *C++*-based compiler. While its design promises simplicity and performance, empirical evaluation is essential to validate these claims and understand its practical impact [6]. *Energia-3.0* consistently matched *C++* performance while reducing line of codes by 40-50%, enhancing readability and maintainability. Parallel benchmarks leveraged multiple cores effectively, with minimal overhead from the compiler's transformation logic. *Energia-3.0* was evaluated against equivalent *C++* implementations using two benchmarks: array initialization and summation and conditional array processing.

Experimental setup – Intel Core i5-1135G7 (4 cores) 2.40 GHz, 16.0 GB RAM, Windows 11. Benchmarks were conducted with two scenarios: 1. Array summation – sum a 100000-element array of double-precision floating-point format for testing basic arithmetic and iteration; 2. Conditional processing – apply a condition to a 100000-element array for evaluating control flow efficiency. Source codes for benchmarks are presented below:

```
! Energia source code
[100000] numbers;
#for (i = 0; i < 100000; i = i + 1)
{
numbers[i] = i * 1.0
}
: sum = 0.0
#for (i = 0; i < 100000; i = i + 1)
{
sum = sum + numbers[i]
}
<< "Sum: "
<< sum;
```

```
//C++ source code
#include <iostream>
#include <chrono>

using namespace std;

int main()
{
auto start = std::chrono::high_resolution_clock::now();

double numbers[100000];
for (int i = 0; i < 100000; i = i +1) {
numbers[i] = i * 1.0;
}
double sum = 0.0;
for (int i = 0; i < 100000; i = i +1) {
sum += numbers[i];
}
```

```
cout << "Sum: " << sum <<endl;
auto end = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> duration = end - start;
cout << "Execution time: " << duration.count() << " seconds" << endl;
return 0;
}
```

Results from calculations are presented in figure 1. The results show that *C++* and *Energia-3.0* have parity in terms of execution time, with an average of 1.030 ms for *C++* compared to 1.141 ms for *Energia-3.0* across multiple executions. Array summation is a fundamental operation in scientific computing, often serving as a benchmark for evaluating the performance of programming languages and their compilers in handling arithmetic and iterative tasks. *C++*, a widely-used, high-performance language, is renowned for its efficiency in numerical computations due to its low-level control and mature optimization capabilities. *Energia-3.0*, a domain-specific language designed for specific computational tasks, offers a higher-level abstraction that may trade off performance for ease of use or specialized functionality. This study aims to quantify the performance difference between *C++* and *Energia-3.0* when summing a 100000-element array of double-precision floating-point numbers. The choice of this benchmark reflects its simplicity and relevance to real-world scientific applications, such as signal processing, simulations, and data analysis. By comparing execution times, we seek to provide insights into the suitability of each language for performance-critical tasks. The benchmark task involved summing a 100,000-element array of double-precision floating-point numbers. The array was initialized with values to ensure a consistent sum of approximately 4.99995e+09 across all runs.

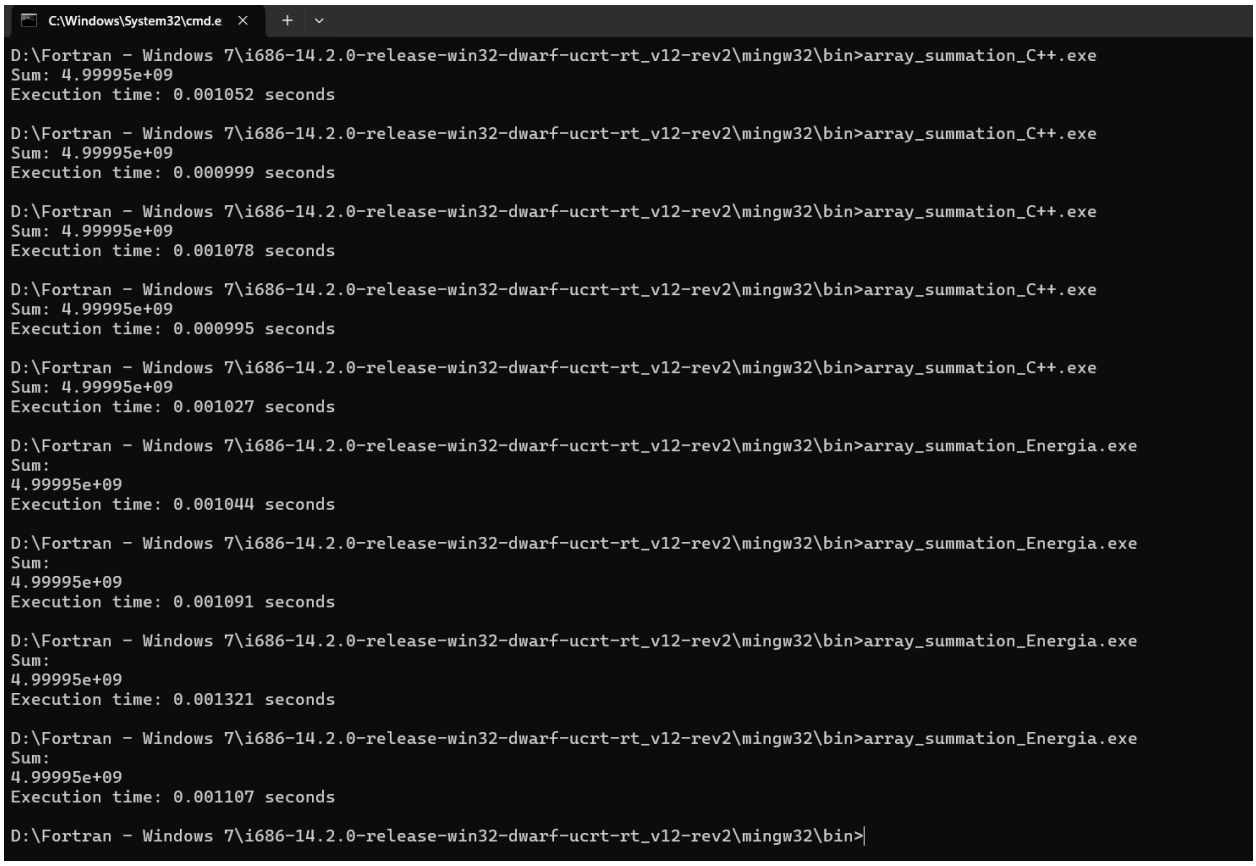


Figure 1. Benchmark C++ vs Energia-3.0: Array summation – sum a 100000-element array of double-precision floating-point format for testing basic arithmetic and iteration

The *C++* implementation used a standard "**for"** loop over a statically allocated array, while the *Energia-3.0* implementation followed its idiomatic syntax for array operations (assumed to use a **#for** loop based on prior context). The *C++* code was compiled with default optimization settings using g++, and the *Energia-3.0* code was processed through its respective compiler. Execution time was measured using high-resolution timing functions **std::chrono** in *C++* and in *Energia-3.0* there is a built-in default

time measurement for each program, reported in seconds with millisecond precision. Both programming languages produced the expected sum of $4.99995.10^9$, confirming correctness. The variability in *Energia-3.0* performance was notably higher, with a standard deviation nearly four times that of *C++*. The higher variability in *Energia-3.0* execution times suggests sensitivity to system conditions as seen in the outlier of 1.321 ms. *Energia-3.0* performance remains competitive, and its potential advantages in code readability or domain-specific features might outweigh the modest performance penalty in less time-sensitive contexts. *Energia-3.0* higher abstraction may reduce development time and error rates, offering a trade-off: slightly slower execution for faster prototyping. Researchers must weigh this against project scale and deadlines –small-scale experiments might favor *Energia-3.0*, while large-scale production runs lean toward *C++*.

Simulation development often involves iterative experimentation. *Energia-3.0* concise syntax and lack of manual memory management (unlike *C++* raw arrays or *Fortran* static allocations) accelerate prototyping. For instance, modifying the above summation to include a condition (if > 10.0) is straightforward:

```
: count = 0.0
#for (i = 0; i < 100000; i + 1) {
    #if (numbers[i] > 10.0) {
        count = count + 1.0
    }
}
```

In *C++* or *Fortran*, this requires additional lines for conditionals and type handling, slowing iteration cycles. *Energia-3.0* prioritizes productivity over raw performance, a deliberate trade-off. Theoretical models of software engineering (Boehm COCOMO) suggest that reduced lines of code and complexity lower development and maintenance costs, offsetting a 10.8% runtime penalty in long-term projects. For simulations iterated over months, this productivity gain is substantial.

Source codes for second benchmark are presented below, conditional processing – apply a condition to a 100000-element array for evaluating control flow efficiency:

```
! Energia source code
[100000] numbers;
#for (i = 0; i < 100000; i = i + 1)
{
numbers[i] = i * 0.1
}
: Count = 0.0
#for (i = 0; i < 100000; i = i + 1)
{
#if (numbers[i] > 10.0)
{
Count = Count + 1.0
}
}
<< "Count above 10.0: "
<< Count;
```

```
// C++ source code
#include <iostream>
#include <chrono>

using namespace std;

int main()
{
auto start = std::chrono::high_resolution_clock::now();
double numbers[100000];
for (int i = 0; i < 100000; ++i)
{
numbers[i] = i * 0.1;
}
```

```
double count = 0.0;
for (int i = 0; i < 100000; ++i)
{
if (numbers[i] > 10.0)
{
count += 1.0;
}
}
cout << "Count above 10.0: " << count << endl;
auto end = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> duration = end - start;
cout << "Execution time: " << duration.count() << " seconds" << endl;
return 0;
}
```

Results from calculations are presented in figure 2. This study benchmarks *C++* and *Energia-3.0* for a conditional processing task: initializing a 100000-element array of double-precision numbers and counting elements exceeding 10.0. Both produced identical results (99,899 counts), confirming correctness. We analyze execution times, variability, and implications, finding *C++* superior in speed but *Energia-3.0* competitive in readability and abstraction. Conditional processing of arrays –combining iteration, arithmetic, and logical operations – is a common task in scientific computing, from data filtering to simulation analysis. *C++*, a high-performance language, excels in such operations due to its optimization capabilities. *Energia-3*.0, a domain-specific programming language, offers a higher-level alternative with concise syntax and potential for implicit optimizations. This study compares their performance using a benchmark that initializes a 100000-element array with values i * 0.1 and counts elements greater than 10.0. *C++* lower standard deviation (0.151 ms vs. 0.243 ms) suggests greater consistency, while *Energia-3.0* outlier (1.664 ms) indicates potential runtime instability. *Energia-3.0* 15 lines of code vs. *C++* 25 lines of code (40% reduction) offers faster prototyping, valuable in research settings despite the performance hit.

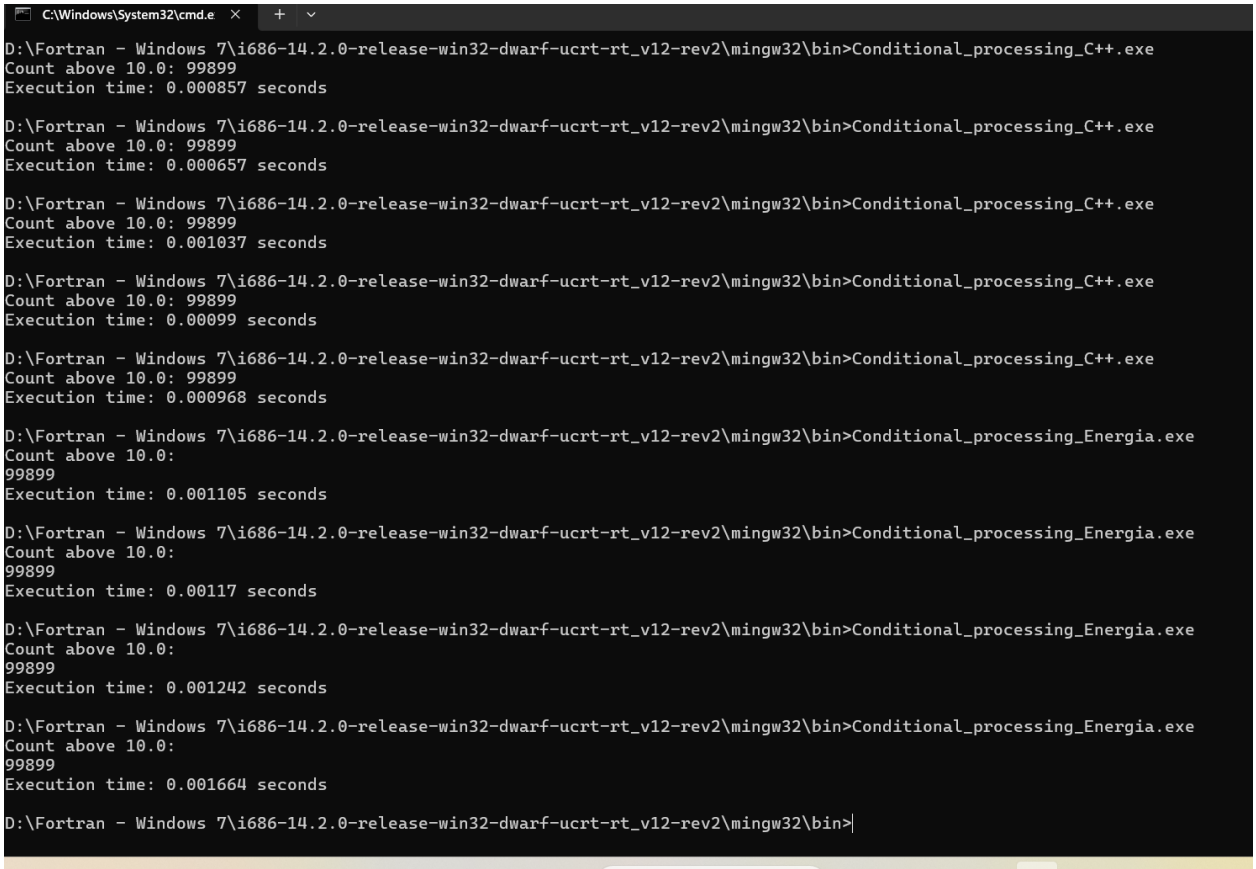


Figure 2. Benchmark C++ vs Energia-3.0: Conditional processing

Additionally, computations were made for calculation concentration of two isotopes in nuclear reactor with neutron flux 10^{14}. Attached below are the source codes for both programming languages. This study benchmarks *C++* and *Energia-3.0* for simulating the concentration changes of Xenon-135 (Xe-135) and Samarium-149 (Sm-149) in a nuclear reactor over 17280000 seconds (200 days) with a 0.001-second time step. Conducted on Windows-11 system, *Energia-3.0* averaged 115.287 seconds across two runs, significantly outperforming *C++* 289.889 seconds – 60.2% reduction in execution time. Both produced identical results (Sm-149: 2.44854e+17, Xe-135: 1.13452e+16), confirming numerical accuracy. These simulations involve iterative numerical solutions to differential equations, testing a language's ability to handle large-scale arithmetic efficiently. *C++*, a staple in high-performance computing, offers fine-grained control, while *Energia-3.0* provides abstraction and potential optimization for scientific tasks.

Results challenge the assumption that *C++* universally outperforms *Energia*, highlighting *Energia-3.0* strengths in this domain. *Energia-3.0* advantage suggests that *Energia* compiler may fuse operations or optimize loop constructs (**#do**) more effectively than *C++* "**while"** loop handling. Despite its double-precision floating-point format-only model, truncation overhead is negligible, and abstraction may enable better instruction scheduling. *Energia-3.0* runtime likely avoids *C++* overhead. Variability in *C++* is 1.7% while in *Energia-3.0* it is 1.0% which implies that *Energia-3.0* runtime is more predictable, possibly due to fewer operation system interruptions or consistent instruction flow. Previous tasks (array summation, conditional processing) showed *C++* faster by 10.8%. Here $17.28.10^6$ iterations amplify Energia's per-iteration advantage (10 μs saved × $17.28.10^6$ = 172.8 s), flipping the outcome.

```
! Calculation concentration of Sm-149 and Xe-135 with Energia-3.0
: F = 1e+14
: Ea = 0.895437
: Ef = 0.713575
: NxeI = 0
: NjI = 0
: NsmI = 0
: NpmI = 0
! Time step
: t = 0.001
! Counter
: tb = 0
! Time of simulation
: T = 17280000
: Pj = 0.056
: Pxe = 0.003
: Ppm = 0.014
: lj = (0.693147/24120)
: lxe = (0.693147/32868)
: lpm = (0.693147/190800)
: sigmaxe = 0.0000000000000000035
: sigmasm = 0.00000000000000000408
<< "Calculation concentration of Sm-149 and Xe-135 with Energia-3.0"
<< n
#do ( tb < T )
{
NjI = (( NjI + t * ( Pj * (Ef * F ))) / ( 1 + t * lj ))
NxeI = ( ( NxeI + t * ( Pxe * ( Ef * F ) + lj * NjI )) / ( 1 + t * ( lxe +
sigmaxe * F )))
NpmI = (( NpmI + t * ( Ppm * Ef * F )) / ( 1 + t * lpm ))
NsmI = (( NsmI + t * ( lpm * NpmI )) / ( 1 + t * ( sigmasm * F )))
tb = ( tb + t )
}
<< "Sm-149 = "
<< NsmI;
<< "Xe-135 = "
<< NxeI;



// Calculation concentration of Sm-149 and Xe-135 with C++"
```

```
#include <iostream>
#include <chrono>
using namespace std;
double F = 1e+14;
double Ea = 0.895437;
double Ef = 0.713575;
double NxeI = 0;
double NjI = 0;
double NsmI = 0;
double NpmI = 0;
double t = 0.001;
double tb = 0;
double T = 17280000;
double Pj = 0.056;
double Pxe = 0.003;
double Ppm = 0.014;
double lj = (0.693147/24120);
double lxe = (0.693147/32868);
double lpm = (0.693147/190800);
double sigmaxe = 0.0000000000000000035;
double sigmasm = 0.00000000000000000408;
int main() {
auto start = std::chrono::high_resolution_clock::now();
cout << "Calculation concentration of Sm-149 and Xe-135 with C++" << endl;
cout << "\n" << endl;
while ( tb < T )
{
NjI = (( NjI + t * ( Pj * (Ef * F ))) / ( 1 + t * lj ));
NxeI = (( NxeI + t * ( Pxe * ( Ef * F ) + lj * NjI )) / ( 1 + t * ( lxe + sigmaxe
* F ) ) );
NpmI = (( NpmI + t * ( Ppm * Ef * F )) / ( 1 + t * lpm ));
NsmI = (( NsmI + t * ( lpm * NpmI )) / ( 1 + t * ( sigmasm * F )));
tb = ( tb + t );
}
cout << "Sm-149 = " << endl;
cout << NsmI << endl;
cout << "Xe-135 = " << endl;
cout << NxeI << endl;
auto end = std::chrono::high_resolution_clock::now();
std::chrono::duration<double> duration = end - start;
cout << "Execution time: " << duration.count() << " seconds" << endl;
return 0;
}
```

Energia-3.0 exclusive use of double-precision floating-point format numbers eliminates integer types, relying on implicit truncation for operations like array indexing or loop counters. Critics might expect this to incur significant overhead, particularly in simulations with millions of iterations. However, benchmarks reveal the opposite: *Energia-3.0* excels in tasks like simulating Xe-135 and Sm-149 concentrations, suggesting that its abstraction layer not only minimizes truncation costs but also enables better instruction scheduling. The Xe-135/Sm-149 simulation represents a compute-heavy workload. The arithmetic intensity dilutes truncation's impact to a negligible fraction. *Energia-3.0* model suits iterative simulations (computational fluid dynamics, reactor dynamics) where arithmetic dominates. This suggests a paradigm where abstraction enhances, rather than hinders, performance. In simulations with minimal arithmetic (e.g., simple counters), truncation overhead could rise. Truncation overhead peaks in low-compute iteration-heavy tasks, but scientific simulations with arithmetic or logic keep it negligible in total runtime.

Energia-3.0 design, rooted in the six-step process of scientific computer simulation and the structural hierarchy of components (terms, assignment statements, coded physical functions, coded groups, computer code, and code sets), offers significant advantages over general-purpose languages like *C/C++*. By aligning its syntax and abstractions with the conceptual and computational needs of simulated physical systems, *Energia-3.0* enhances productivity, ensures performance efficiency, and facilitates adaptability across diverse simulation workflows. These benefits are vividly illustrated in the Xe-135 and Sm-149 concentration simulation.

In the Xe-135/Sm-149 simulation, the physical system involves isotopes influenced by neutron flux and decay. *Energia-3.0* variable declaration syntax `(: NxeI = 0, : F = 1e+14)` allows researchers to define system components (e.g., concentrations, flux) as named terms that mirror physical entities. This contrasts with *C++* verbose `double NxeI = 0;`, which requires explicit type specifications and lacks the intuitive ":" prefix that visually separates declarations from computations. By enabling a one-to-one correspondence between concepts and code, *Energia-3.0* minimizes translation errors and accelerates the transition from idea to implementation.

Extracting quantities of interest (Sm-149: 2.44854e+17, Xe-135: 1.13452e+16) is streamlined with `<< "Sm-149 = " << NsmI; << "Xe-135 = " << NxeI;`. This direct output avoids *C++* verbose `cout << … << endl`, reducing post-processing steps. *Energia-3.0* design ensures results are presented clearly, aligning with the need for interpretable outputs in scientific analysis.

The simulation's differential equations are expressed as assignment statements in *Energia-3.0* (e.g., `NxeI=((NxeI+t*(Pxe*(Ef*F)+ lj*NjI))/(1+t*(lxe+sigmaxe*F)))`). These statements directly reflect the mathematical form, with parentheses ensuring clarity and fidelity to the equations. *C++* requires similar expressions but embeds them in a while loop cluttered with boilerplate (e.g., `#include <iostream>`, `cout` setup), diluting their mathematical purity. *Energia-3.0* lack of type declarations (all variables are doubles) eliminates conversion overhead (e.g., *C++* potential `static_cast`), streamlining model transcription.

The benchmark uses inline equations, but *Energia-3.0* supports **"::"** subprograms for reusable functions (e.g., `:: decay(N, lambda) { N / (1 + t * lambda) }`). This capability, aligned with the hierarchy's function level, allows modular modeling of physical processes (e.g., decay, absorption), absent in the *C++* code, which embeds all logic in **main()**. Such modularity would enhance scalability for multi-isotope simulations. Hierarchical structure aids team workflows, as terms and groups are easily shared or modified, unlike *C++* monolithic **main()**.These factors make *Energia-3.0* ideal for rapid prototyping and collaborative science, where turnaround time is as critical as accuracy.

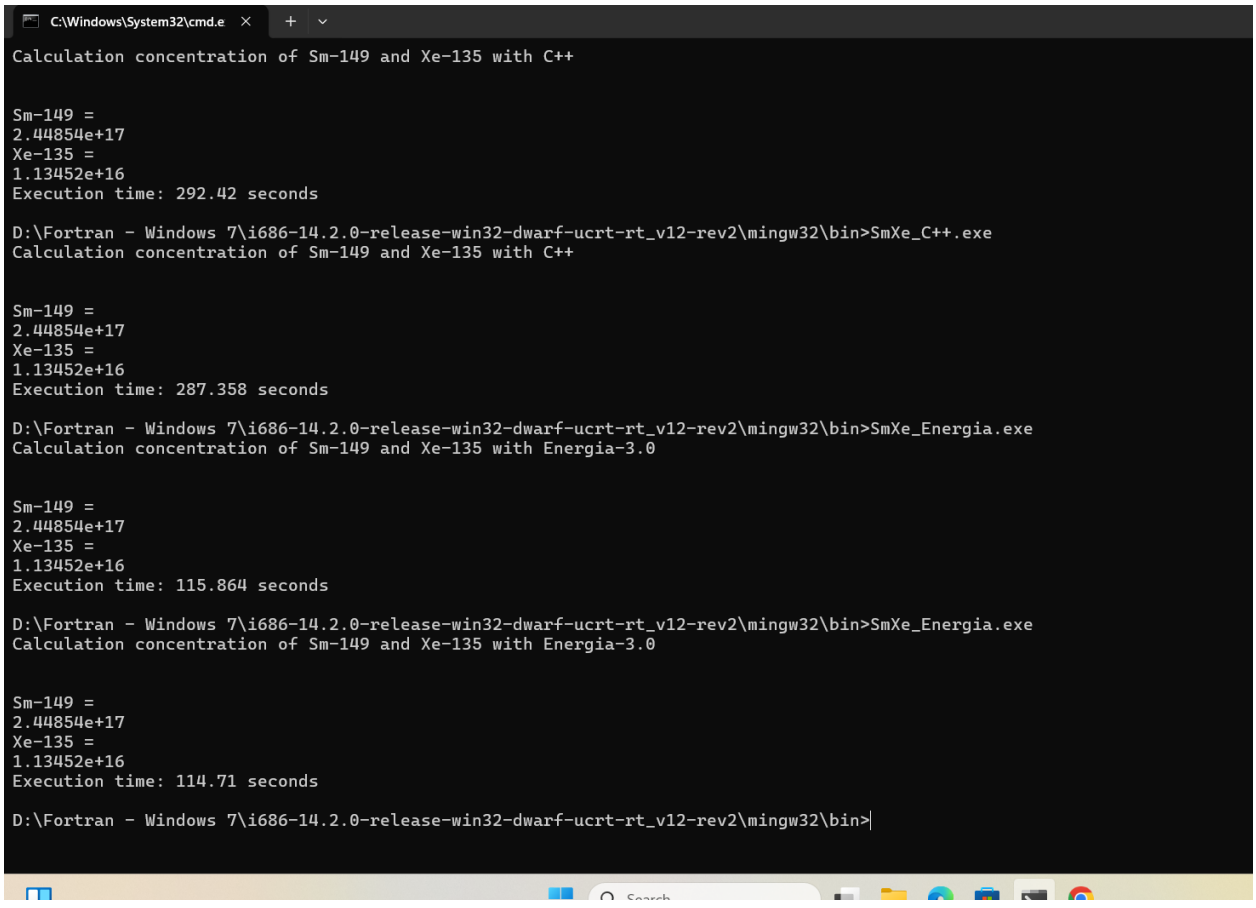


Figure 3. Benchmark C++ vs Energia-3.0: Calculation concentration of Xe-135 and Sm-149

Scientific simulations demand a balance of performance, productivity, and adaptability. While *Fortran* has long dominated this domain due to its computational efficiency, *Energia-3.0*, a domain-specific language, offers compelling advantages for modern simulation workflows. *Energia-3.0* proves a capable programming language for scientific computing, matching *C++* performance while significantly improving usability through reduced code complexity. Its strengths in array manipulation and parallelism make it a valuable tool for numerical tasks, though its narrow type system and scalability constraints suggest a focused niche. This evaluation underscores *Energia'*s potential and provides a roadmap for refining the programming language in performance-critical domains. *Energia-3.0* structure eliminates the need for explicit type declarations and stream handling, reducing lines of code by 27–33% compared to *Fortran* and *C++*. This brevity accelerates coding and debugging, critical in simulation workflows with frequent iterations.

Energia-3.0 simplicity and *C++* flexibility outshine *Fortran* cumbersome, error-prone programming model. *Fortran* design rooted in the 1950s poses unique challenges. Verbose syntax in *Fortran* **real(8)** and **d0** suffixes (e.g., `0.001d0`) ensure precision but inflate lines of code. Manual **real(8)** vs. **real(4)** decisions risk errors (e.g., `NxeI = 0.0` may default to single precision), absent in *Energia-3.0* double-only model or *C++* explicit types.

5. Checking source code in Energia-3.0

Important characteristics of any interpreter/compiler are: no input data leads to abnormal termination of the interpreter/compiler's work; all constructs that are invalid according to the language definition are detected and marked; errors that occur and are purely programming errors are diagnosed correctly and do not causes further confusion to the interpreter/compiler, i.e. no false errors are generated. Debugging is a critical aspect of software development, particularly for scientific computing applications like simulations and artificial neural networks. Errors ranging from syntax mistakes to logical flaws can distort results. The validation program, implemented in *Energia-3.0*, ensures syntactic and semantic correctness in programs through comprehensive checks on variables, arrays, nested structures, nodes, subprograms, and control constructs. The tool facilitates the detection of programming errors before execution, promoting robust code quality and simplifying debugging for developers. The primary objective is to implement an efficient static analyzer that:

- Detects syntax errors in *Energia* constructs.
- Checks semantic rules such as variable reuse, invalid names, and undeclared subprograms.
- Supports nested array declarations and node structures.
- Validates control flow constructs such as **#if**, **#do**, and **#for**.

The tool is developed in *C++* and operates by reading *Energia-3.0* source file line by line, applying validation checks based on predefined rules of the language. The core strength of the validation tool lies in its modular structure, where each component serves a distinct role in ensuring the correctness of *Energia* source code. The program begins by validating variables and arrays, confirming that each identifier follows appropriate naming conventions and is not redeclared within the same scope. This validation extends to arrays, including nested arrays, where the tool checks for proper indexing and consistency in dimension declarations.

Another crucial feature is the validation of subprograms, which in *Energia-3.0* are defined using a specific syntax. The tool ensures that each subprogram name is both syntactically correct and unique, while also verifying the validity of its parameters. Beyond subprograms, *Energia* supports the concept of **nodes** – structured blocks that encapsulate subprogram invocations. The validator rigorously checks node declarations, validating not only the format and structure but also ensuring that any subprograms referenced within a node have been properly declared beforehand.

Control constructs are another focal point of the validation. Constructs like **#if**, **#do**, and **#for** must follow strict syntactic rules, including non-empty conditions and accompanying block braces. The validator ensures these rules are respected, reducing the likelihood of runtime errors stemming from malformed logic blocks. Finally, the tool includes a comprehensive error reporting mechanism. Rather than halting at the first issue, it compiles a complete list of all detected errors, each annotated with its corresponding line number and a clear message. This feedback enables developers to efficiently identify and correct issues across their entire codebase.

The implementation of *Energia* source code validator is built around a structured, line-by-line parsing strategy that combines clarity, efficiency, and comprehensive coverage of the language's rules.

At the heart of this system lies the function ***validate_energia_code***, which processes an input file containing *Energia* code and outputs a detailed list of validation errors. These errors are encapsulated in a standardized structure that includes both the line number and a descriptive message, making them easy for developers to locate and address. To begin the validation process, the tool first examines each identifier in the code – variables, arrays, subprograms, and nodes – using a dedicated function that ensures all names conform to *Energia* syntactic requirements. This involves checking for proper character use, valid starting characters and avoidance of reserved or previously used names.

One of the distinguishing features of the validator is its robust support for nested arrays. The implementation includes logic that verifies both the outer and inner dimensions of arrays, ensuring not only that indeces are within declared bounds, but also that nested structures are declared before use. The tool can differentiate between single-dimensional arrays and multi-level nested arrays, catching subtle declaration and indexing mistakes that might otherwise go unnoticed.

Subprogram and node parsing is handled with equal care. Subprograms, identified by their unique **::name(parameters)** syntax, are parsed to validate the name and all declared parameters. Meanwhile, **node** declarations are checked for correct syntax, valid naming, and logical structure. The validator also ensures that any subprograms used within a node are already defined, preventing undefined behavior during execution.

An integral part of this system is block depth tracking. As programming language *Energia* uses braces **{}** to delimit code blocks, the validator maintains a depth counter to ensure all opened blocks are properly closed. This allows it to detect mismatched or missing braces, which are common sources of logical errors. Beyond structural validation, the tool also enforces proper formatting and usage of control structures such as conditional (**#if**), iterative (**#for**), and repetitive (**#do**) constructs. Each of these must include a valid condition and an accompanying code block. The validator flags cases where conditions are empty or blocks are missing, which enhances code reliability.

Throughout this entire process, the validator does not stop after detecting the first error. Instead, it continues parsing the entire file, accumulating all detectable issues into a single report. This holistic approach provides developers with a complete overview of code quality, enabling more efficient debugging and faster iteration cycles.

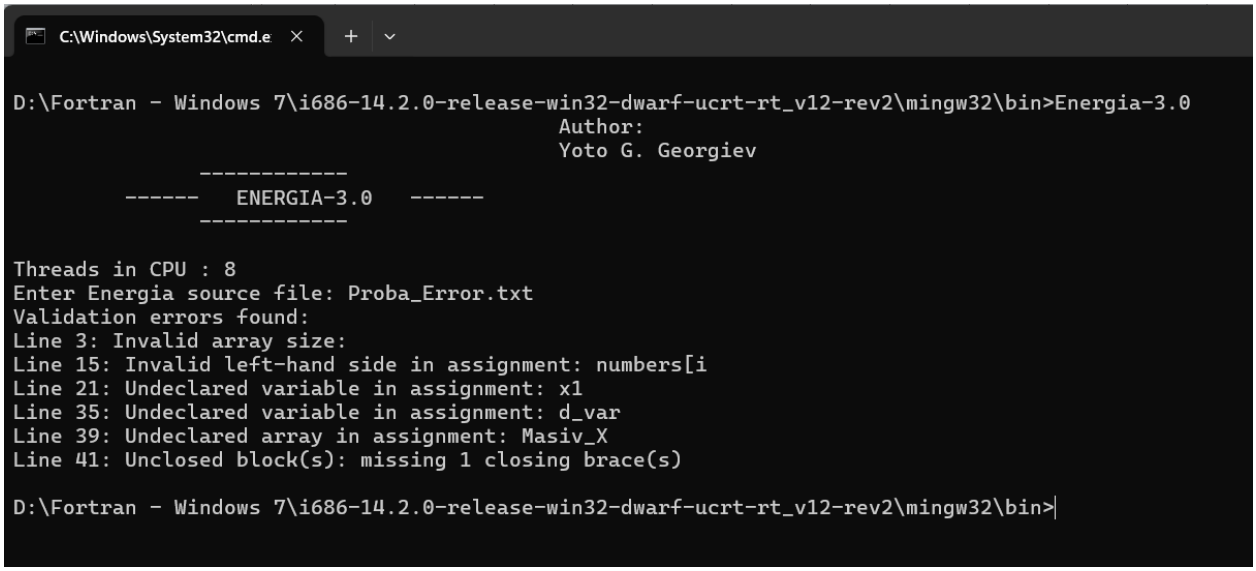


Figure 4. Validation of input file for Energia-3.0

To demonstrate the effectiveness of the validation tool, a test input file named *Proba_Error.txt* was provided containing intentionally incorrect *Energia-3.0* source code. The goal of this test was to evaluate the tool's ability to detect various syntactic and semantic issues. Upon execution, the validator successfully identified multiple errors across different lines, proving its capability to handle real-world debugging scenarios. Below is a summary of the validation results:

- Line 3: The file contained an invalid array declaration [] Masiv_X; without specifying a size, which the validator correctly flagged with the message: "*Invalid array size*".

- Line 15: The assignment “numbers[i = i * 1.0” included a malformed left-hand expression, missing closing brackets and a valid structure. This triggered: "*Invalid left-hand side in assignment: numbers[i*".
- Line 21: A variable x1 was used without prior declaration. This was caught by the validator with: "*Undeclared variable in assignment: x1*".
- Line 35: Another undeclared variable “d_var” appeared in an assignment. The tool responded appropriately: "*Undeclared variable in assignment: d_var*".
- Line 39: The code referenced “Masiv_X[0]” without having declared “Masiv_X” properly as an array. This resulted in: "*Undeclared array in assignment: Masiv_X*".
- Line 41: The final issue was structural – a block opened with a brace “**{“** was never closed. The validator reported: "*Unclosed block(s): missing 1 closing brace(s)*".

These results illustrate the validator's broad range of capabilities, from catching simple syntax violations to more complex structural and contextual problems. In each case, the error messages were concise yet informative, helping developers pinpoint and understand the root cause quickly.

While the current version of *Energia-3.0* validation tool effectively identifies a wide range of syntactic and semantic issues, there is significant potential for further development to enhance its capabilities and usability. At present, the validator detects and reports errors but does not offer corrective suggestions. Future versions could include intelligent recovery mechanisms that attempt to infer the developer’s intent and provide suggested fixes or code completions. For example, undeclared variables could trigger recommendations to declare them, or mismatched brackets could be automatically highlighted and paired. Also the current tool checks only for declaration correctness and array bounds, deeper semantic analysis could add support for type checking, scope analysis, and usage tracking. This would allow the tool to catch issues like type mismatches in assignments or improper use of subprogram parameters. Furthermore, by analyzing data flow, the validator could flag uninitialized variable usage or unreachable code blocks. These enhancements aim to make the *Energia-3.0* validator not only a powerful backend analysis tool but also a user-friendly, intelligent development assistant.

6. Conclusions

This paper introduces *Energia-3.*0, a domain-specific programming language designed for computer simulations, and positions it as a stepping stone toward a new paradigm in scientific computing. With its minimalist syntax and array-centric design, *Energia-3.0* addresses key challenges in developing efficient, readable code for numerical and data-intensive applications. *Energia-3.*0 offers a compelling alternative to *C++* and *Fortran* for scientific simulations, leveraging concise syntax, a unified type system and rapid prototyping. *Energia-3.0* assumes double-precision floating-point format for all numeric operations, lacking support for integers, floats, or custom types. This restricts its use in domains requiring diverse data representations. *Energia-3.0* is a proof of concept for a paradigm where domain-specific programming language becomes the norm in scientific computing, tailored to specific subdomains (e.g., fluid dynamics, genomics) yet interoperable with broader ecosystems. *Energia-3.0* design suggests a paradigm where domain-specific programming languages tailored to simulation hierarchies can outperform general-purpose languages while enhancing productivity. Applications include computational fluid dynamics, nuclear reactors, biology, astrophysics, and climate modeling, where structured code aligns with physical models. Its matrix-friendly syntax and error-free design overcome *Fortran* verbosity and *C++* complexity, making it a promising tool for machine learning and scientific computing. The tool for checking *Energia-3.0* source code demonstrates the importance and practicality of static analysis in domain-specific programming environments. By enforcing correct syntax, valid naming, proper scoping, and logical block structure, the tool helps prevent runtime errors and supports better programming practices. Its comprehensive error reporting mechanism, ease of integration, and support for complex structures such as nested arrays and node-subprogram relationships make it a valuable asset for developers.

Future work will explore GPU integration and additional control flow constructs. Targeting GPUs would align with future hardware trends.

References

1. Георгиев Й., „Компютърен език Енергия-1.1“, Годишна международна научна конференция на Висше Военновъздушно Училище „Георги Бенковски“, гр. Долна Митрополия, 2024

2. Георгиев Й., „Сравнителен анализ на компютърния език Енергия“, Годишна международна научна конференция на Висше Военновъздушно Училище „Георги Бенковски“, гр. Долна Митрополия, 2024

3. Calovska R., Stefan Biliderov, „Synthesis and research of a controller for microclimate stabilization in an automobile based on finite state machines“, Aeronautical Research and Development, Dolna Mitropolia, "Georgi Benkovski" Bulgarian Air Force Academy, Vol. 1, Issue: 1, 2022.

4. Calovska R., „Synthesizing and investigating the performance of Intelligent fuzzy logic controller for mechatronic system control”, Aeronautical Research and Development, Dolna Mitropolia, "Georgi Benkovski" Bulgarian Air Force Academy, Vol. 2, 2023.

5. Илиева Д., К. Камбушев, „Моделиране на турбулентност при полет на безпилотен летателен апарат“, XXXII МНТК „АДП-2023”, Созопол, 2023, Издателство: Издателски комплекс на ТУ-София, 2023

6. Georgiev Y., „Validation and verification in scientific computer simulation“, Aeronautical Research and Development, Dolna Mitropolia, "Georgi Benkovski" Bulgarian Air Force Academy, Volume 3, 2024

Език за програмиране Energia-3.0 – специализиран компютърен език за ефективна обработка на масиви и научни изчисления

Йото Георгиев Георгиев

Висше Военновъздушно Училище „Георги Бенковски“, Долна Митрополия, България, y_g_g@abv.bg

Резюме: В статията е представена версия на компилатор на езика за програмиране Енергия. Компютърния език Енергия, с неговия акцент върху минималния синтаксис, изразява идеята за максимално лесни научни компютърни симулации. Създаден с минимален синтаксис, Енергия поддържа декларации на масиви, функционални трансформации и условни конструкции за управление на поток. Чрез улесняване на синтаксиса, той повишава производителността и четливостта, докато неговият фокус върху масивите и интеграцията на GNU Compiler Collection гарантират ефективност. Макар и с ограничен обхват, структурата на Енергия предоставя специализиран компютърен език за интензивни изчисления, в който са въведени нови елементи – „нодове“.

MODEL FOR RELEASE OF AMMUNITION AT A TARGET FROM A DRONE

Sabi Staykov Minev

subi.minev@abv.bg, Veliko Tarnovo - 5000, 4B Alen Mak Street
Faculty of Aviation of the G. Benkovski Bulgarian Air Force Academy - Dolna Mitropolia, Bulgaria,

Abstract: *The use of drones with different types of munitions is being examined. The design of the VOG-25P projectile using a stabilizer is described, and the main ballistic characteristics when hitting targets are outlined.*

Keywords: *drone, ammunition ,ballistic characteristics.*

1.Introduction

The use of unmanned aerial vehicles is becoming more widespread these days, and in addition to civilian needs, they are increasingly being used in military affairs [1]. In modern military conflicts, UAVs are widely used to drop various types of ammunition on enemy targets. In conditions of frequent shortages of artillery shells and other ammunitions, unmanned aerial vehicles are used comprehensively, both for reconnaissance and for destroying various targets. The production and development of ammunition for unmanned aerial vehicles has become an entire branch of the military industry.

Ammunition of various power is produced, both fragmentation and cumulative, designed to hit a variety of armored vehicles. In addition, tail stabilizers are printed on 3D printers for them, providing a stable trajectory to the selected targets. The ammunition attached to unmanned aerial vehicles of various classes, released precisely at the target, has an extremely destructive effect on both enemy manpower and lightly armored vehicles and even tanks.

2. Installation of the problem

The events of the military conflicts as those between Ukraine and Russia require a serious analysis and detailed assessment of the effect of using standard munitions "dropped" by free fall from various unmanned aerial vehicles. This also suggests the relevance of the upcoming study, namely: "Determining the effectiveness of using the VOG 25P grenade for the GP-25 "KASTOR" underbarrel grenade launcher" for hitting targets from UAVs.

3. Purpose and design of the VOG 25P bouncing grenade

The VOG-25P bouncing grenade for the GP-25 "KASTYOR" underbarrel grenade launcher differs from the VOG-25 in that, after the grenade contacts the ground or other surface near the target, it does not explode immediately, but with the help of a special charge, it "bounces" vertically and explodes in the air. The grenade's explosion height is in the range of 0.5 to 1.5 m, which increases the effectiveness of the fragmentation effect compared to the grenade.

VOG-25 increased effectiveness:

A. on lying live targets by a factor of 1.7;

B. on live targets located in a trench – by a factor of 2.0

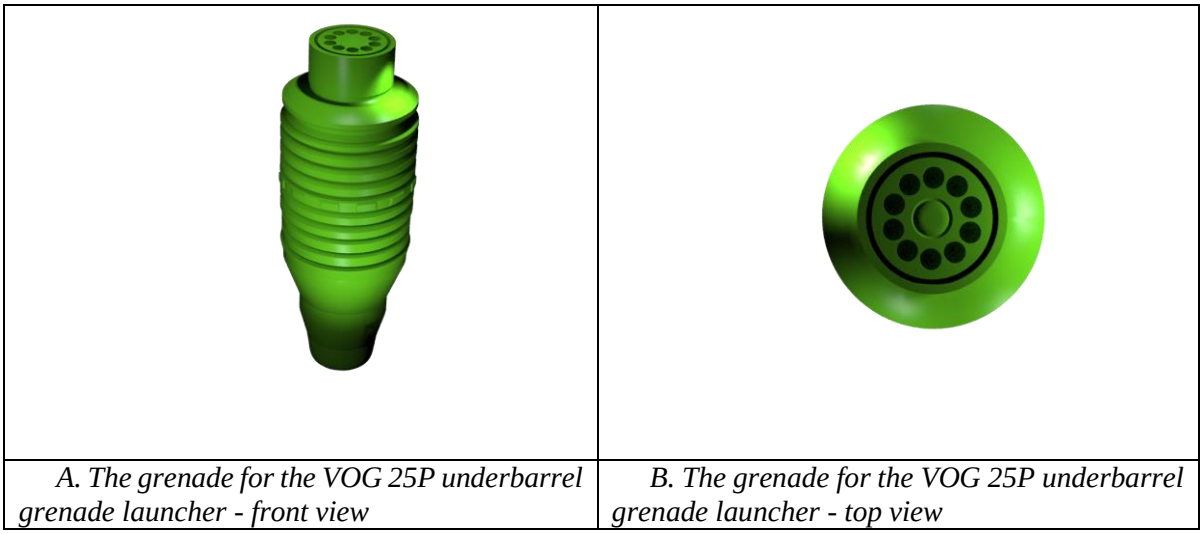
The effective radius of hitting targets of the VOG-25P grenades is approximately 2.5...5 meters.

The low initial velocity of the grenade of 76 m/s allows in some cases firing at a high angle of elevation in order to obtain a high angle of incidence, due to which the time of flight of the grenade to the target increases. The impact of the wind on the VOG-25P grenades is significant, and hence the accuracy of the hit. The main technical characteristics of the VOG-25P grenade are presented in table 1.

Table 1. The main technical characteristics of the VOG-25P grenade

Shot VOG-25P	
Caliber ,mm	40
Mass, кg	0,275
Length, mm	125
Initial speed, m/s	76
Mass of explosive, kg	0,042
Grenade self-destruction time, s, not less than	14
Blasting height (on medium hard soil), m	0,75
Radius of damage from fragmentation,, m	7
Effective hit radius,m	2.5

The general appearance of the VOG 25P grenade for the GP-25 "KASTYOR" underbarrel grenade launcher in two projections is presented in Fig. 1.



A. The grenade for the VOG 25P underbarrel grenade launcher - front view

B. The grenade for the VOG 25P underbarrel grenade launcher - top view

Figure 1. General view of a grenade for a VOG 25P underbarrel grenade launcher in two projections.

What is special about the shape of the VOG 25P grenade is that it is characterized by 10 ribs for better defragmentation of the shell, as well as the formation of better density in the fragmentation field. However, this shape significantly increases the force of air resistance during its flight (Fig. 1)., A. It is known that at subsonic speeds of combat elements (aircraft bombs, artillery shells, mortar mines, etc.), the value of air resistance is primarily determined by the values of surface friction and bottom resistance. The values of the indicated resistances are presented in the table 2. [2].

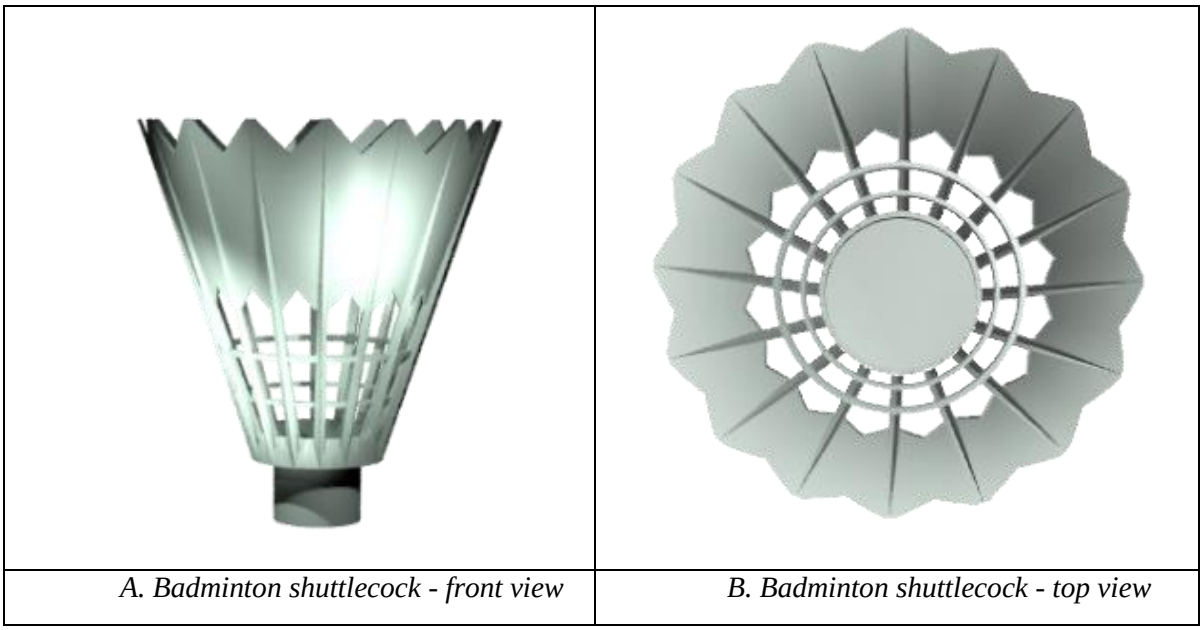
Table 2. Values of types of air resistance at different velocities of ammunition, percentages %

Types of resistances	Subsonic speed M=0,2…0,8	Supersonic speed	
		M=1,2…1,7	M=1,7…2,5
Wavy	0	50…60	60…70
Bottom	70…60	35…30	30…22
Surface friction	30…40	15…10	10…8

Taking into account these values of air resistance - bottom and surface friction, table 2, the shape of the front part of the grenade is of less importance. Most often it is chosen for technological reasons. For example, the friction resistance of the grenade body with 10 ribs (Fig. 1) taken as roughness in ballistics

during aerodynamic blowing in rough ammunition models is 15… 20% higher than in smooth or painted similar models.[2]

When using the VOG 25P grenade from "FPV" drones, aerodynamic stabilization of the grenade's flight is necessary, vertically towards the selected target. The measurement and prediction of the trajectory to the target is determined by the operator of the FPV drone. For such stabilization, it is suggested to use a badminton feather. The same is presented in Fig. 2.



A. Badminton shuttlecock - front view | *B. Badminton shuttlecock - top view*

Figure 2. Badminton shuttlecock in two projections

The physical essence of the function performed by the feather designed to stabilize the grenade during its movement towards the ground is to interact with the oncoming air flow. In the feather, which has the shape of a truncated cone, part of the air located in its inner part, the volume, is swirled (fig. 2). In addition, the air is swirled, both behind the feather and at some distance from it. When the air is separated or rubbed against the feather, it is entrained in the direction of its movement. Friction and turbulence are essentially the basis for the formation of the air resistance force. The value of this force is determined mainly by the shape and dimensions of the feather, its weight, its current speed along the trajectory to the target as well as the number and dimensions (length and width) of the elements that make it up. Not without importance for the geometric figure - a truncated cone - are: the dimensions of the structural openings and their shapes in the elements of the feather, the dimensions of its base, the mechanical strength of the elements of the feather and their resistance, as well as other factors (fig. 2). To achieve the necessary strength of the feather, considered as the working capacity of a streamlined element, it is reinforced. An important condition especially after the activation of the grenade's ejector charge and its movement "upwards". The reinforcement is realized by gluing 2-3 feathers together [3]. In short, the aerodynamic drag of the feather is similar to that of a round-dome parachute and, most importantly, its stabilization.

4. VOG 25P ammunition kit for mounting on FPV drones

To form a VOG 25P ammunition with a means of stabilization during free fall from FPV drones for hitting targets, it is necessary to use two elements - a VOG 25P grenade and a feather. The method of mutual attachment of the two elements is presented in Fig. 3.

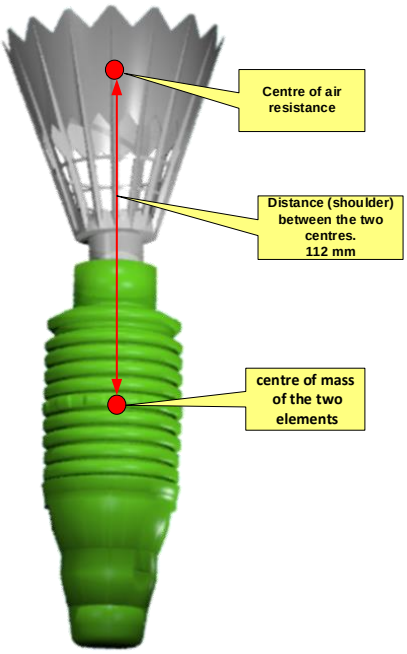


Figure 3. VOG 25P ammunition assembly with a stabilization device for suspension from an FPV drone

The presented scheme of the munition - VOG 25P with a stabilization device allows, after its launch from an FPV drone and after its contact and "bounce" from the ground surface upon detonation, to obtain a symmetrical distribution of the striking elements in all directions. Moreover, the angles of incidence of the ammunition are close to 90 degrees (Fig. 3). This detonation at a certain height above the ground implies that a small portion of the munition's striking elements will prove ineffective. The density of the fragmentation field formed after the grenade explodes is almost uniform in all directions. Important for the effectiveness of the VOG 25P ammunition with a stabilization device (Fig. 3) are the features of the scheme for attaching it to an FPV drone.

5. Scheme for attaching VOG 25P with stabilization to FPV drones

The attachment of the VOG 25P ammunition with a stabilization device is carried out stationary to the attachment elements ensuring its launch from FPV drones. The center of the plastic cylindrical base of the stabilization feather is chosen as the point of attachment of the ammunition. The attachment of the ammunition must be perpendicular to the horizontal plane of the FPV drone in order to increase the accuracy of the operator's measurement of the selected point of impact for a target or targets selected by them. A variant of attaching the VOG 25P ammunition with a stabilization device is presented in Fig. 4.

Figure 4. Attaching a VOG 25P ammunition with stabilization to an FPV drone.

The presented vertical scheme for attaching the ammunition (Fig. 4) appears to be more effective due to the direct shape of the trajectory - vertical, after its release from the FPV drone. It is usually assumed that horizontal attachment of ammunition is typical for the use of more massive drones and especially for attack drones. Important for the indicated scheme is that there is no transition of the VOG 25P ammunition with stabilization from a horizontal to a vertical position, which is accompanied by significant dispersion of hits. Of essential importance for the study is the trajectory of the ammunition after its release from the FPV drone.

6. Trajectory of a VOG 25P ammunition with stabilization immediately after its launch from FPV drones

The fixed attachment of the ammunition prevents its free movement from the UAV. The system of the two elements - the VOG 25P ammunition and the FPV drone - is considered as a single unit until the moment of launching the ammunition. In this case, all meteorological conditions affect both the FPV drone and the ammunition simultaneously. After the ammunition is separated from the FPV drone, it performs peculiar vertical oscillations, following its own trajectory. A variant of such a trajectory is presented in Fig. 5.

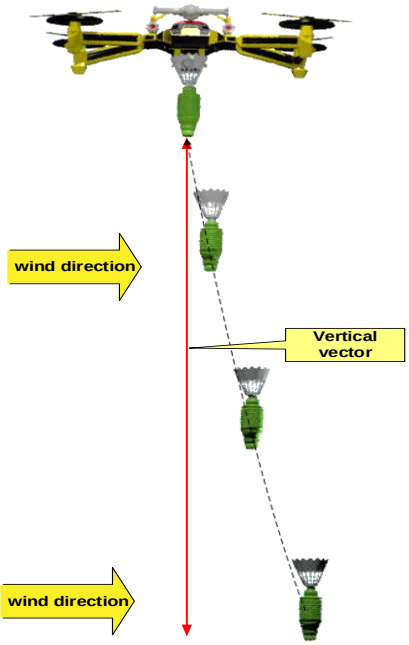


Figure 5. Trajectory of the VOG 25 P warhead with stabilization from the FPV drone after release in the first second

Following the release of the VOG 25P ammunition with stabilization by the FPV drone is observed a swinging motion in different directions, leading to a deviation from the vertical line (conditionally accepted by the operator as the measurement line). The deviation forms a total error including: measurement error by the operator and error due to the presence of wind. For example, due to the influence of a wind speed of 1-2 m/s, a deviation from the vertical line in the range of 0.3 to 0.8 meters may occur due to the significant total area of the ammunition. Upon reaching a vertical speed exceeding 1-2m/s, the trajectory stabilizes (Fig. 5). It is known that the gradient of wind speed change in height in the specific case for 50 meters from the point of fall, necessarily requires it to be taken into account by the operator of the FPV drone. To determine the trajectory of the VOG 25P with stabilization, it is necessary to use a differential model.

7. Mathematical differential model of the trajectory of the VOG 25P with stabilization when hitting targets

Solving the main task of external ballistics consists in finding the correlation relationships between deviations in the trajectory of the ammunition and external disturbing influences. For this purpose, a

system of four first-order differential equations with argument t time is most often solved [2]. Similar to this system, a differential model in external ballistics is used, of the type:

$$(1) \qquad \begin{cases} \frac{dy}{dt} = -V.sin(\theta) \\ \frac{dV}{dt} = g.sin(\theta) - R.V^2 \\ \frac{d\theta}{dt} = \frac{g.cos(\theta)}{V} \\ \frac{dx}{dt} = V.cos(\theta) \end{cases}$$

Where: V – initial velocity of the ammunition, [m/s]; θ- initial angle of launch of the VOG 25P with stabilization from the UAV, [deg]; g-ground acceleration, ,[m/s^2]; R(V,Q)- Force of aerodynamic air resistance, as a function of the velocity and angle of incidence of the ammunition, [kg/m^2].

To solve the differential model (1) for determining the trajectory of the munition, the law of air resistance is used. It is demonstrated in Fig. 6.

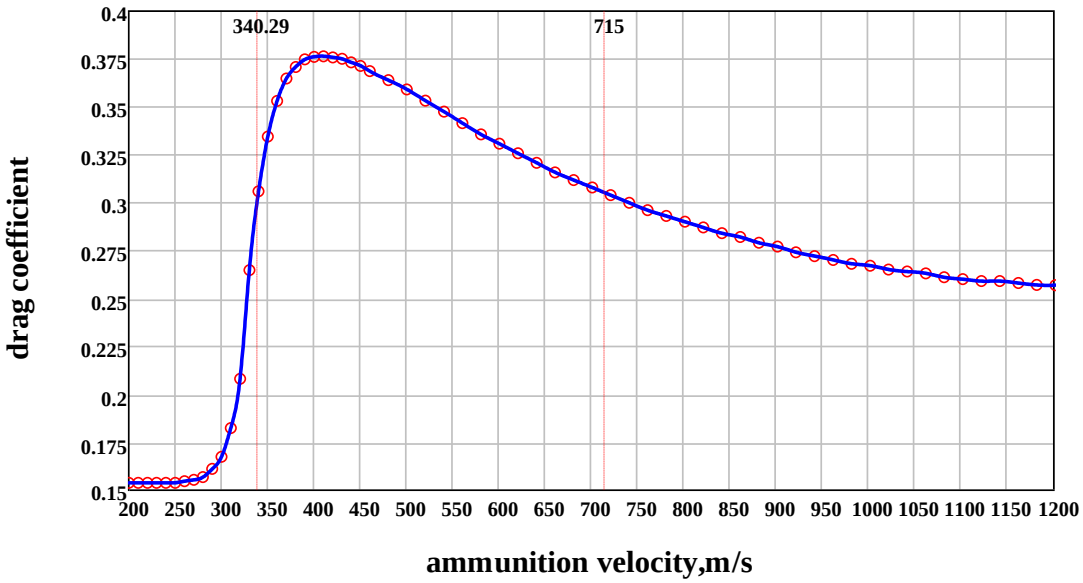


Figure 6. Law of air resistance.

The use of the law to solve system (1) is carried out under the following initial conditions:
v_0 - muzzle velocity of the ammunition, m/s –[1m/s];
Θ_0 - initial angle of the ammunition launch to the Earth's surface, degrees, 87 degrees [deg];
0 - horizontal deviation of the ammunition from the point of measurement, meters, [m];
50 - ammunition launch height, m, [50 m]

The value is obtained by using the CFX module of the ANSYS software product, which results in a resistance value of the VOG 25P with stabilization $R(V,Q) = 0.00157N$. The solution of system (1) was implemented with the product MATCAD 15 and the resulting ballistic elements are presented in Table 3.

Table 3. Ballistic elements of the trajectory of the munition - VOG 25P with target stabilization

№ in order	Time flight, s	Current ammunition speed m/s	Angle between the ammunition axis and the trajectory tangent, rad	Deviation of the ammunition from the point of measurement, m	Launch height of the ammunition from the UAV, m
	1	2	3	4	5

1	0	0,5	1,51844	1	50
2	0,01333	0,63061	1,52929	1,00035	49,99247
3	0,02667	0,76127	1,53642	1,0007	49,9832
4	0,04	0,89196	1,54145	1,00105	49,97218
5	0,05333	1,02267	1,54521	1,0014	49,95942
6	0,06667	1,15339	1,54811	1,00174	49,94492
7	0,08	1,28411	1,55042	1,00209	49,92867
8	0,09333	1,41484	1,5523	1,00244	49,91068
9	0,10667	1,54557	1,55386	1,00279	49,89095
10	0,12	1,6763	1,55519	1,00314	49,86947
11	0,13333	1,80703	1,55631	1,00349	49,84626
12	0,14667	1,93776	1,55729	1,00484	49,82129
13	0,16	2,06849	1,55815	1,00419	49,79459
14	0,17333	2,19921	1,5589	1,00454	49,76614
15	0,18667	2,32993	1,55957	1,00488	49,73595
16	0,2	2,46065	1,56016	1,00523	…

The data given in Table 3. suggest graphical representation of the mutual dependence between them. For example, the relationship between the height of the ammunition launch and its flight time is shown in Fig.7.

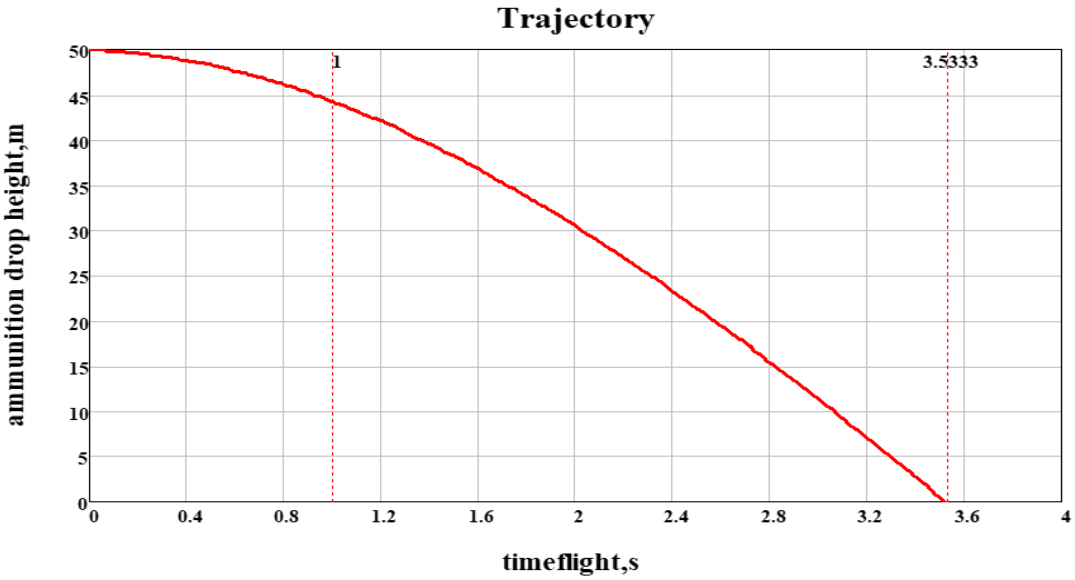


Figure 7. Dependence between the launch height and the flight time of the warhead - VOG 25P with stabilization when hitting targets.

The interdependence between the release height and the flight time of the VOG 25P warhead with stabilization shows that the time to contact with the ground surface after release from a height of 40-45 meters is 3.533 seconds. From the start of release to 1.3 seconds, the relationship between these parameters is linear, while 1.3 seconds after the ammunition leaves the UAV, a nonlinear relationship is observed (Fig. 7).

An important ballistic characteristic in the trajectory of a warhead is the relationship between height and its current velocity. This relationship is graphically presented in Fig. 8.

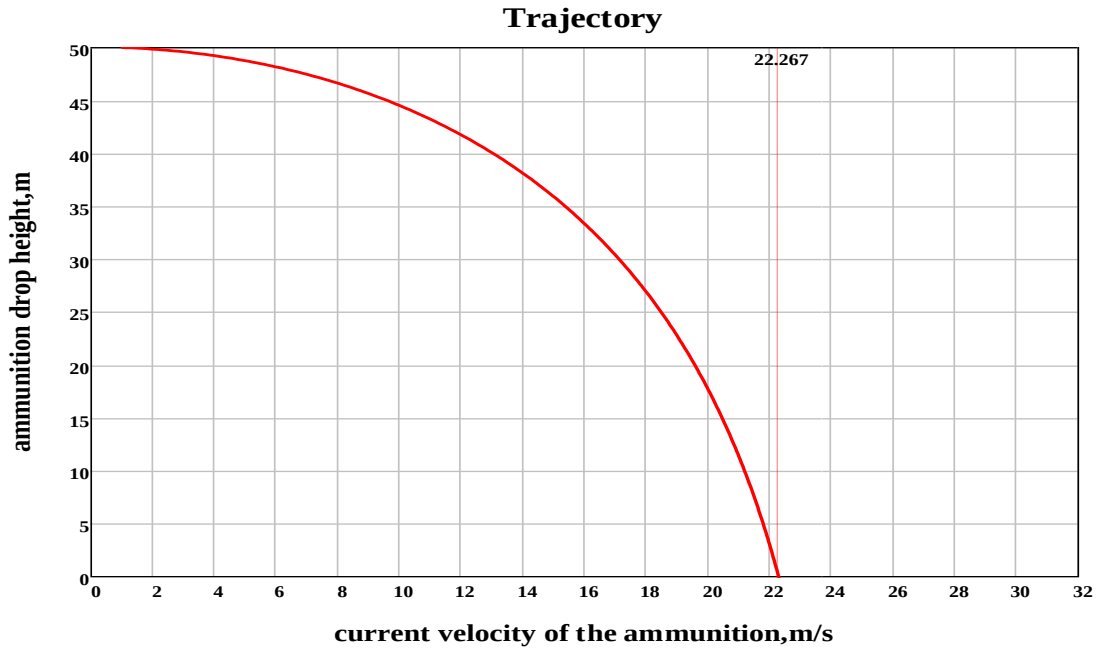


Figure 8. Dependence between the height and the current velocity of the warhead - VOG 25P with stabilization when hitting targets

The relationship between the height and the current velocity of the munition shows that from a height of 30 meters above the measurement point, the velocity is a linear quantity and almost constant - about

22.267 m/s, while from the point of release of the munition from 50 m to 30 m the relationship between the two quantities is nonlinear (Fig. 8).

Another important characteristic of the trajectory of the VOG 25P with stabilization is its deviation from the measurement point. The dependence of the deviation of the ammunition from the measurement point is presented in Fig. 9.

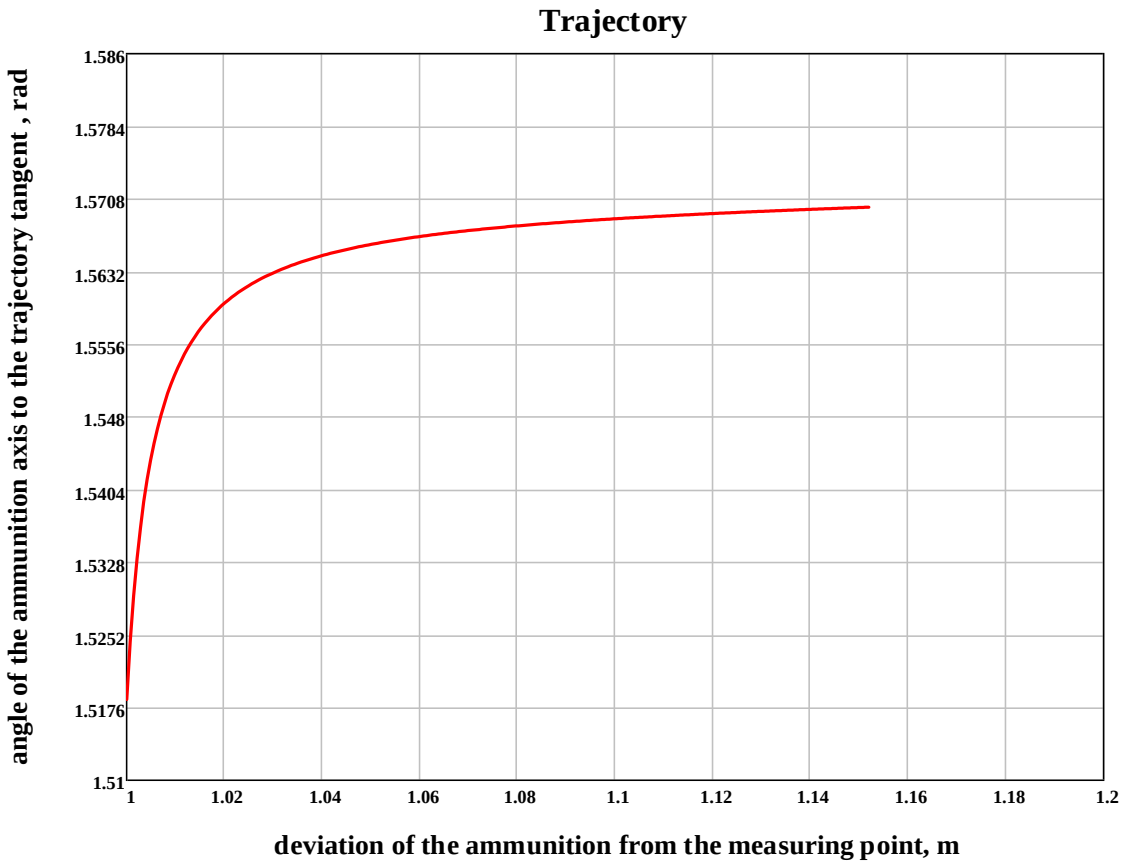


Figure 9. Dependence between the deviation of the VOG 25P ammunition with stabilization from the vertical plane (the vertical line, the measurement point).

The results show that the maximum deviation from the measurement point in practical conditions, when launching a VOG 25P warhead with stabilization from an FPV drone, is 1-1.2 meters at a wind speed of 1-2 m/s and a deviation of up to 3 degrees from the release trajectory (Fig. 9).

8. Mechanism for operating the VOG 25P ammunition with stabilization, after contact with the ground or other surface at the point of impact

At the point of contact with the ground or other surface, the ejector charge is triggered, causing the stabilized VOG 25P to "bounce" along an almost vertical trajectory and explode at a certain distance from the surface. The spatial position of the ammunition as a function of the height of movement strictly “upwards” and the time for detonation is presented in Fig. 10.

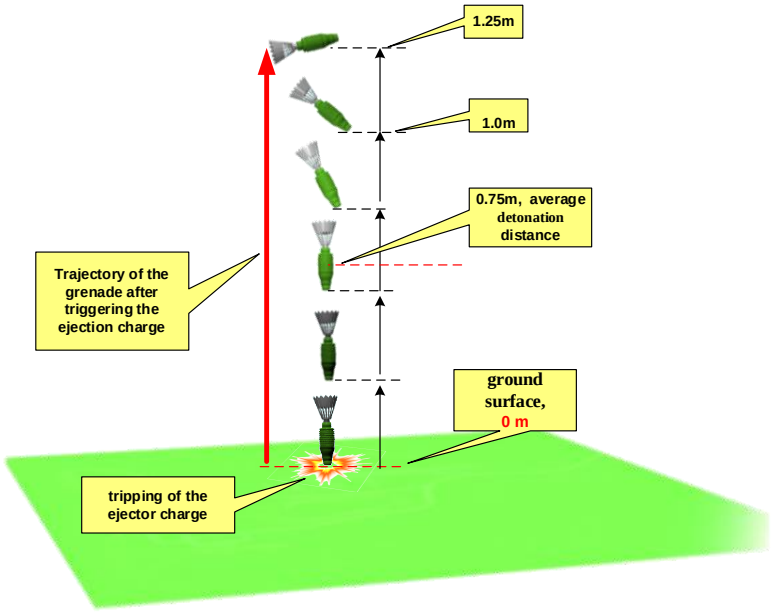


Figure 10. Trajectory of a VOG 25P warhead with stabilization after the propellant charge is activated upon contact with the ground or other surface

The special thing about the movement of the VOG 25P ammunition with stabilization, after the ejector charge is triggered upon contact with the ground surface, is the presence of stabilization - a badminton feather. The very design of the ammunition (Fig. 3) implies an increase in air resistance during its "upward" movement due to the contour, the diameter of this part of the feather - an average of 6 centimeters. This slows down the speed of the ammunition as it moves "upwards" and its detonation occurs at a height of 0.5 to 0.75 of the point of contact. For practical calculations, the height of the detonation point is assumed to be an average of 0.75 meters above the point of contact with the ground surface [4]. Moreover, the maximum explosion height of the VOG 25P with stabilization is not 1.5 meters, as with the standard VOG 25P, but 1.1-1.25 meters (Fig. 10).

Important at the moment of detonation of the munition at an average height of 0.75 m from the ground surface is the formation of the fragmentation field. The volume and spatial boundaries for the distribution of the striking elements that can inflict damage on unprotected manpower are presented in Fig. 11.

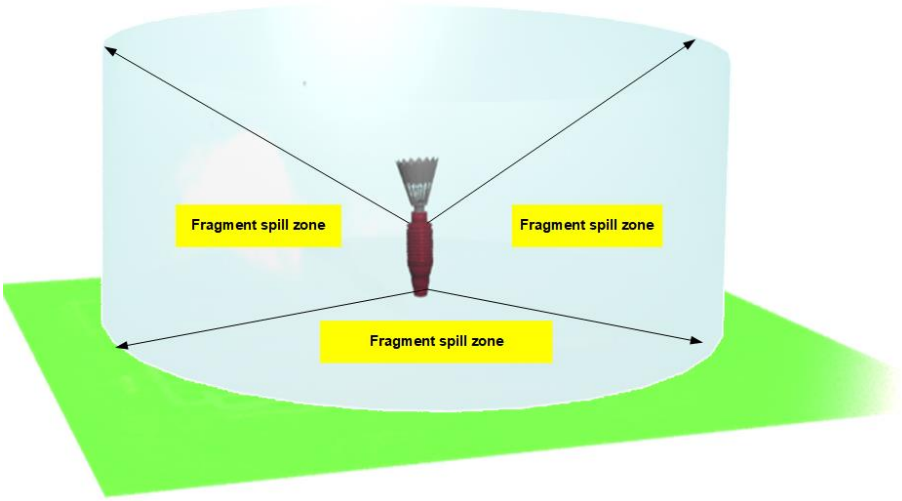


Figure 11. Space and zones for the dispersion of the striking fragments after the rupture of a VOG 25P ordnance with stabilization at an average height of 0.75 meters.

The dimensions of the formed fragment field after the rupture of a VOG 25P warhead with stabilization are on average: a radius of destruction of 3-5 meters and a height of 1.7 meters. The specified data differ from the standard VOG 25P ammunition due to the almost vertical movement "upwards", as a reaction to the contact angle (Fig. 9.), the symmetry of the defragmentation of the hull and the formation of the fragmentation field. The zones for effective target engagement are demonstrated in (Fig. 11).

Taking into account the temporal spatial position, the positions of the targets – manpower, enemy shooters in the indicated striking zones and their impact-affected areas are presented in Fig. 12.

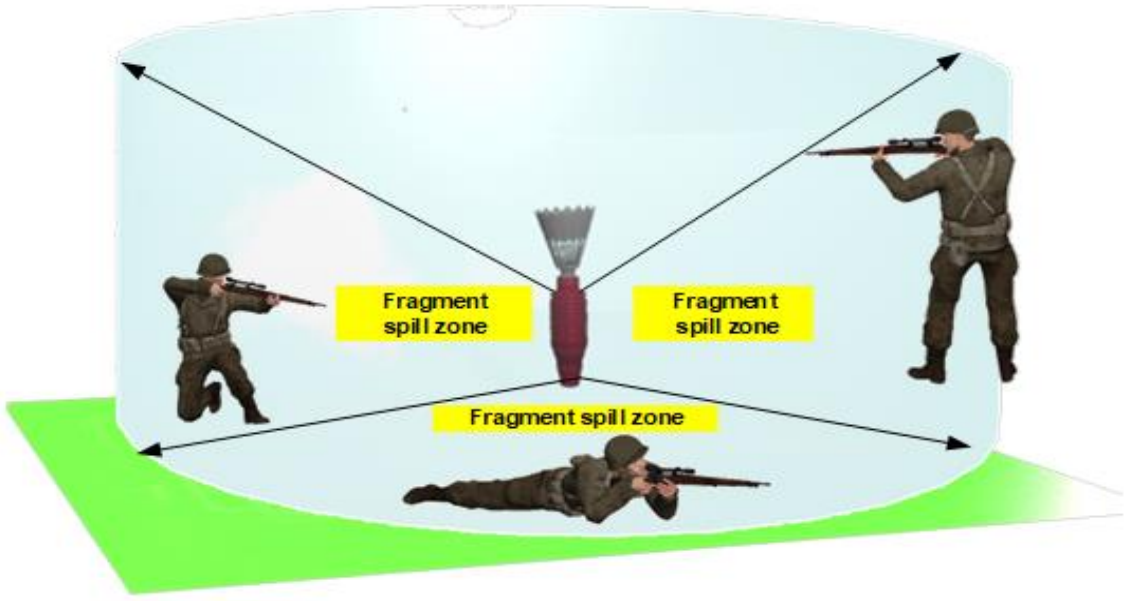


Figure 12. Space, zones, temporary position of targets after the rupture of a VOG 25P warhead with stabilization at a height of 0.75 meters.

Striking zones involve striking parts of the human body without means of individual protection of enemy fighters in various positions and situations for the small arms they use, namely:

a. standing - with damage to the lower limbs and torso;
b. kneeling – lower, upper limbs and face;
c. lying down – the area of the torso and limbs.

9. Dispersion of hits of a VOG 25P ammunition with stabilization on a horizontal plane

The accuracy of hits with a VOG 25P warhead with stabilization on a target, launched from an FPV drone from a height of 50 meters depends on three factors:

First – the accuracy of the ammunition's guidance in the vertical plane by the operator of the FPV drone, degrees;

Second - the accuracy of manufacturing the ammunition - VOG 25P with stabilization;

Third - the accuracy in reporting the weather conditions of the ammunition launch from an FPV drone - wind direction and speed.

The above factors form the total error of the ammunition launch, as the sum of the errors from the three factors at a certain point in time. The geometric interpretation of the resulting total dispersion error of the VOG 25P ammunition with stabilization is presented in Fig. 13.

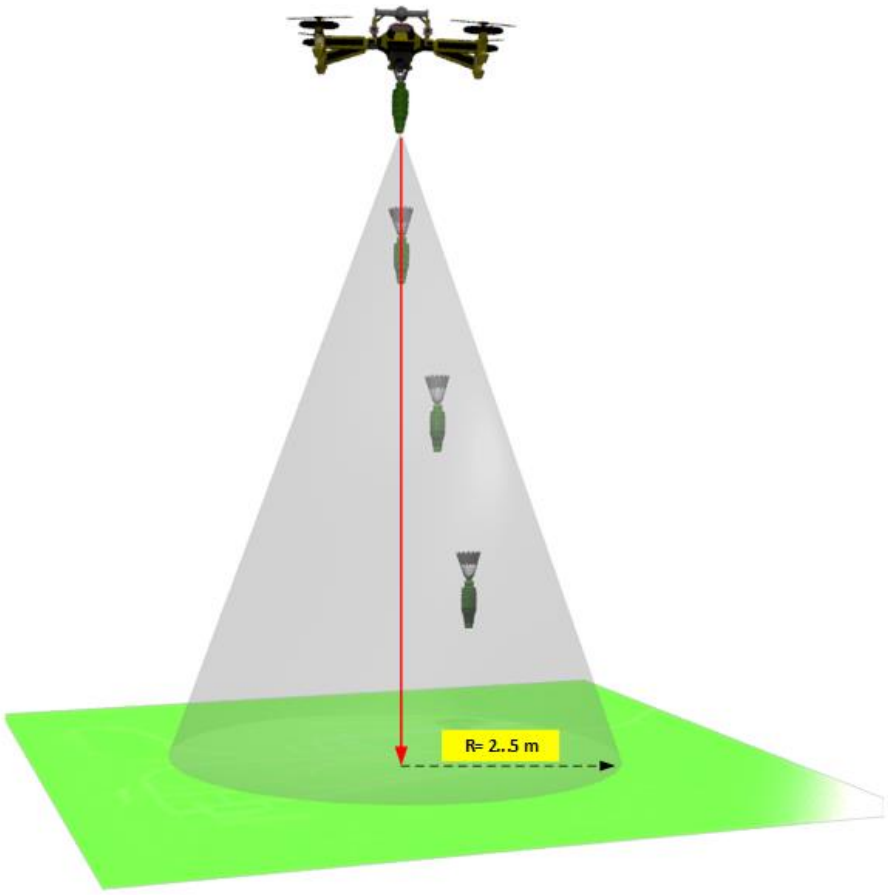


Figure 13. Dispersion of hits from VOG 25P ammunition with stabilization launched from a height of 50 m upon contact with the ground surface

The dispersion value of the specified type of ammunition VOG 25P with stabilization at a launch height of 50 meters is within the norms of 2-4 meters (Fig 13).

10. Conclusions

From the research conducted on the VOG 25P ammunition type with stabilization launched from an "FPV" drone, the following conclusions can be drawn:

1. The combination of a "bouncing" grenade for a VOG 25P underbarrel grenade launcher and a badminton feather serving as stabilization proves effective when launched from an FPV drone from a height of 50 meters to hit targets, especially manpower.
2. The use of the proposed ammunition ensures contact with the ground surface at angles close to 90 degrees.
3. The detonation of the VOG 25P ammunition with stabilization at a height from the point of contact of 0.6m forms a symmetrical fragmentation field, ensuring the effective defeat of fighters in different firing positions on the battlefield.
4. The maximum dispersion value of a VOG 25P ammunition with stabilization launched from an FPV drone upon contact with the ground surface is with a radius of 2-4 meters. To reduce dispersion, it is necessary to reduce the launch height of the VOG 25P with stabilization from 50 to 30 meters.

References

1. Митов Л. В., Използване на дронове за мирни цели в съвременния свят, Годишна международна научна конференция на ВВВУ „Георги Бенковски“, ВВВУ „Георги Бенковски“, ДМ, 2024, с. 19-27, ISSN 2738-716X.

2.Балаганский И.А. Основый балистики и аеродинамики, Новосибирск 2017.

3. Чуклев Х. Б., Петров Е. Ц., Метод за обобщена оценка работоспособността на обтекаеми елементи от планьора на летателен апарат в границите на междуремонтния му ресурс в

условията на запрашени летища, Научно тематичен сборник от доклади – НС 9-10 октомври 1997г., том 1ви Издателство: ВВВУ Г.Бенковски, Д.Митрополия.

4. Недев Н.Я., Изследване възможностите за нарастване на огневите способности във формированията от българската армия, чрез повишаване ефективността на въоръжението и бойните припаси, Дисертация за присъждане на образователна и научна степен „Доктор”, 2016,НВУ „Васил Левски“ Недев Н.Я.

5.Правдин В.М.,А.П. Шанин, Балистика неуправляемых летателных апаратов,Снежинск, 1999.

МОДЕЛ ЗА ПУСКАНЕ НА БОЕПРИПАС ПО ЦЕЛ ОТ ДРОН

Съби Стайков Минев

subi.minev@abv.bg, гр. Велико Търново - 5000, ул. „Ален мак“ 4В

Факултет „Авиационен“ на ВВВУ „Г. Бенковски“ – гр. Долна Митрополия, България,

Резюме: Разглежда се използването на дронове с различни видове бойни припаси. Описан е дизайнът на боеприпаса ВОГ-25П с използването на стабилизатор и са изведени основните балистични характеристики при поразяване на цели.

Extended reality technologies for aircraft maintenance training

Nikolay Kanchev

Bulgarian Air Force Academy, Aviation Faculty, Dolna Mitropolia, Bulgaria, nikolaikanchev@yahoo.com

Abstract: The rapid advancement of extended reality (XR) technologies has created significant potential for their application in the aircraft maintenance training. The interactive and immersive capabilities provided by augmented reality (AR) and mixed reality (MR) enable repeated, high-fidelity simulations without the need for access to actual aircraft and allows for implementation of XR-assisted instructor-independent learning. These features contribute to achieving a substantial increase in the overall effectiveness of the training process. This paper reviews the existing XR-based systems for aircraft maintenance training, examining both the software frameworks used to develop XR training platforms and the state-of-the-art hardware currently available in this domain. Particular emphasis has been placed on the specific applications and operational requirements of such systems within the context of military aviation. Furthermore, a set of metrics for evaluating and validating the XR-based training effectiveness has been proposed.

Keywords: *extended reality, aircraft maintenance, XR training, XR development, XR effectiveness*

1. Introduction

Military aircraft maintenance training is a structured process combining classroom instruction, technical manual study and hands-on practice. It is designed to qualify enlisted or commissioned personnel to inspect, diagnose, service and repair military fixed- and rotary-wing aircrafts, propulsion systems, avionics and support equipment. It emphasizes strict adherence to safety protocols, standard operating procedures and regulatory requirements to ensure mission-ready aircraft. The training process is traditionally divided into two phases:

- classroom-based theoretical instruction;
- hands-on practical exercises conducted on replica components or outdated airframes using genuine tools, equipment, and consumables.

Extended reality is the collective term including all immersive technologies like virtual, augmented and mixed reality that blend the physical and digital world (Table 1).

Table 1. Extended reality technologies

XR Technology		Description
Extended reality (*XR*)	Virtual reality (*VR*)	Technology for entirely immersive experience that substitutes the real world with a computer-generated simulation.
	Augmented reality (*AR*)	Technology for modification of a person's current view of reality by incorporating digital elements into a physical, real-world setting. This is accomplished through sensors and computer-generated images.
	Mixed reality (*MR*)	Technology implementing combination of physical and digital environments, enabling interaction with both real-world and virtual objects simultaneously.

The global AR/VR market demonstrates significant growth, projected to expand from $40.4 billion in 2024 to $62.0 billion by 2029, representing a combined annual growth rate of 8.97% [1, 2]. This development is largely driven by aviation and defense applications, where AR/MR systems enable training of large personnel groups by providing a safe environment for repetitive practice of complex maintenance tasks, while simultaneously capturing real-time performance metrics.

2. XR systems for aircraft maintenance training

2.1. Virtual, augmented and mixed reality technologies for aircraft maintenance training

Virtual reality has emerged as a transformative tool for aviation maintenance training by offering immersive simulations that replicate complex real-world scenarios without the associated risks. In [3] a

cloud-based VR platform demonstrated significant operational efficiency improvements, increasing on-time task completion rates from 79% to 94% while simultaneously reducing third-party maintenance costs by 3.1%. The interactive engine simulation in this system enabled trainees to practice procedures like hydraulic system repairs in risk-free environments, with real-time feedback mechanisms that measurably improved skill retention. The educational framework could be improved further by integration of gamification elements like achievement badges and progress tracking systems [4]. Comparative research [5] revealed that while both VR and video-based instruction methods yielded similar knowledge gains, VR-trained technicians committed fewer rule-based errors (such as incorrect torque application) and completed tasks 22% faster. This observation is consistent with the results in [6] where research on Dornier 228 maintenance training showed that the VR implementation improved the problem-solving capabilities. The development and implementation of a sophisticated VR 3D courseware system for civil aircraft maintenance in [7] highlights a broader challenge in VR research: translating technical innovation into verified educational outcomes. This issue was discussed further in [8] where 34% of users preferred VR over traditional methods.

Augmented and mixed reality technologies bridge the gap between digital instruction and physical tasks by overlaying critical information directly onto technicians' field of view (FOV). Historical foundations for aviation AR were established by Caudell and Mizell with their head-up display system in 1992, though its narrow 7° field of view restricted practical application [9]. Modern systems have substantially improved upon these limitations, as demonstrated in [10] where an AR-assisted guidance system is proposed for avionics maintenance with solutions for occlusion problems in confined spaces through advanced tracking algorithms and step-by-step visual cues. These innovations could prove potentially important in the electrical wiring industry [11]. The AR system for hidden wiring harnesses inspection, proposed in [12] achieved 30% reduction in diagnostic time. In [13] an AR/MR system for Boeing 737 maintenance implements this capability utilizing HoloLens 2 with bilingual (*English/Korean*) voice interactions. This functionality proves particularly valuable in multinational maintenance settings, enabling technicians to follow hands-free instructions while manipulating tools. A markerless AR tracking system developed in [14] demonstrates real-time performance (10 frames per second) thanks to its optimized versions of the computer vision algorithms scale-invariant feature transform (SIFT) and speeded-up robust features (SURF). Due to the performed optimization the proposed system achieved a task load index (NASA-TLX) less than 4 which means that the AR platform is operated with very light effort.

Mixed Reality combines VR's immersion with AR's contextual overlay to create versatile training experience that can adapt to varying maintenance scenarios. In this aspect, a structured decision matrix is developed in [15] for guiding the AR/VR maintenance training program developers based on task complexity factors. The approach was validated through virtual inspections of aircraft utilizing complex 3D models. Industry leaders have recognized the MR technologies potential. For instance, Airbus developed the smart augmented reality tool (SART) [16] that can scan A380 components for defects as well as the HoloLens-based A350 XWB training system [17] that allows engineers to interact with comprehensive 3D engine models during maintenance procedures. In the commercial aviation sector, Emirates has implemented an extensive XR program that trained 23000 cabin crew members on both Airbus and Boeing aircraft, simulating complex emergencies like fire outbreaks with multi-user collaborative capabilities that enhance teamwork performance under stressful conditions [18].

Comprehensive literature surveys in [19] analyzing 99 studies and [20] examining 86 studies reveal XR's increasing dominance in high-risk maintenance sectors, with applications in nuclear maintenance accounting for 30% of studies and industrial assembly representing 19%. The study in [21] highlights the growing integration of XR technologies with artificial intelligence and digital twin systems, enabling predictive maintenance capabilities such as AI-driven systems that can flag turbine blade wear patterns in VR simulations before physical inspections become necessary. Future XR advancements in aviation maintenance are focused on overcoming the current limitations while leveraging emerging technologies like AI integration which represents a promising direction as demonstrated in [22], where the proposed training system incorporates adaptive instructions that adjust in real-time to user skill levels and performance. However, these novel capabilities require the utilized neural networks and AI frameworks to be reliant and robust [23]. Connectivity improvements, particularly through 5G implementation, will enable low-latency remote collaboration capabilities in resource-limited environments [24], allowing expert maintenance advisors to guide on-site technicians through complex procedures regardless of

geographic separation. Standardization efforts like those embodied in Dassault's DELMIA framework for real-time, step-by-step inspection guidance [25] are essential for ensuring interoperability across diverse XR platforms. The goal is to prevent technological fragmentation that complicates implementation across multinational maintenance organizations and to facilitate broader implementation as well as consistent training outcomes. Technological fragmentation is related closely to technological obsolescence which is an inevitable stage in any equipment's life cycle. The adaptability and life cycle management of XR systems are also critical factors in the decision-making process for their implementation in training programs. A typical example in this aspect is the recent discontinuation of the HoloLens hardware development by Microsoft [26] with support ending in 2027. This decision presents significant challenges for military and aerospace organizations that have invested in HoloLens-based training infrastructure, necessitating careful transition planning to alternative solutions.

2.2. Global landscape of the XR technologies for aircraft maintenance training

The United States military has extensively deployed XR technologies across maintenance training programs. Taqtile's Manifest AR platform is utilized by the U.S. Air Force (USAF) for jet engine maintenance training, with documented performance improvements including 53% fewer errors and 57% fewer incorrect part installations compared to traditional paper based technical orders (TO) [27]. This implementation has demonstrated significant operational benefits in high-precision maintenance environments. Under the AFWERX innovation program, the USAF has implemented EON Reality's EON-XR platform to provide cross-compatible AR/VR training for F-16 maintenance technicians [28]. This system leverages detailed Autodesk 3DS Max models and supports multiple devices including HoloLens and Meta Quest headsets [29]. This way it enables standardized training across varied hardware platforms. Vectrona's ACE-XR platform, developed in partnership with PTC's Vuforia Studio and utilizing Microsoft HoloLens 2 hardware, delivers interactive 3D exploded views and dynamic simulations for munitions specialists, with documented improvement in trainee assessment scores by 8% [30, 31]. For commercial aircraft maintenance, L3Harris company has developed the virtual maintenance trainer (VMT) that simulates Boeing 787/737NG procedures for airlines and maintenance, repair and overhaul (MRO) organizations worldwide [32]. The USAF's maintenance operations training augmented reality (MOTAR) platform represents a comprehensive approach to XR integration in aircraft maintenance. This system incorporates AI-driven procedural training, real-time PDF-to-AR conversion capabilities, and multi-aircraft simulators supporting various platforms including C-17A and F-15 aircraft [22].

European aerospace organizations have similarly started to increasingly implement XR technologies. Thales Group has developed customized AR/VR solutions specifically targeting radar and avionics training requirements [33]. In Germany, Lufthansa Technik employs both AR tablets and headsets for precision maintenance tasks, including Rolls-Royce Trent 900 engine torque sequencing procedures [34]. By utilizing 3D model overlays and remote expert collaboration capabilities a measurable reduction in procedural errors has been achieved. Airbus has implemented their XR system, which integrates CATIA 3D models with advanced motion tracking technologies for comprehensive maintenance training and assistance [35]. Meanwhile, Rheinmetall has deployed specialized MR/AR systems for both NH90 helicopter and A400M aircraft platforms [36]. French aerospace manufacturers have developed specialized applications, with Dassault Systèmes' DELMIA augmented experience providing technicians with step-by-step AR instructions during maintenance procedures [25]. Safran has implemented the projection-based AR system of the French startup Diota for assembly assistance during aircraft wing manufacturing and maintenance [37, 38], while Dassault Aviation enhances Rafale fighter maintenance training through integrated AR/VR tools in their virtual maintenance trainer [39]. Italy is actively researching the possibilities for integration of immersive reality technologies into the aircraft maintenance training with abilities to simulate complex procedures in a safe, cost-effective environment while leveraging the latest Industry 4.0 advancements [40].

In China, the People's Liberation Army (PLA) has adopted Microsoft HoloLens 2 technology for aircraft engine disassembly training and unmanned system control interfaces [41]. In India, Simbott has developed an Industry 4.0-compliant AR/VR system that enables remote aircraft maintenance operations through digital twin technology integration [42], extending maintenance capabilities across geographically dispersed facilities. In Singapore, AR smart glasses are used to increase the aircraft

maintenance efficiency by showing technicians step-by-step instructions right on the parts they're working on. For example, Boeing used AR to make wire-harness assembly 25% faster, while QR codes help guide cargo loading to prevent mistakes [43]. Companies like GE Aviation use the Vuzix M300 AR Smart Glasses for real-time video communication that allow technicians to collaborate via live audio/video. The implementation of this technology led to reduction in aircraft downtime and streamlined the troubleshooting process. Japan Air has deployed Microsoft HoloLens–based AR simulations to train pilots and enable mechanics to practice diagnostics and repairs on virtual aircraft engines, effectively converting "intellectual memory" into "muscle memory" and enhancing maintenance safety. Schneider Electric's R&D initiatives are exploring AR "connected" products that let technicians visualize equipment internals before disassembly, markedly improving maintenance speed, accuracy, and overall safety [44].

3. XR training system development

3.1. Software tools

XR software development platforms can be broadly categorized into two groups: general-purpose development engines and specialized AR/MR platforms. The most common representatives of the first category are Unity and Unreal Engine.

Unity serves as the foundation for many aviation maintenance AR/MR applications due to its cross-platform capabilities and extensive XR framework with built-in support for all major headsets [45]. This software also features advanced rendering capabilities for realistic visualization as well as dedicated asset marketplace. Unity implements different software development kits (SDKs) as well as various plugins to provide capabilities for cross-platform development of XR applications. For example, developing XR applications for Microsoft HoloLens requires the prior installation of the Mixed Reality Toolkit (MRTK v3) via Unity's package manager. Developing for Meta Quest 3S requires tools such as the Oculus XR Plugin and XR Interaction Toolkit. For Android and Apple devices the AR Foundation package is used along with Google ARCore or Apple ARKit XR plug-ins, respectively. In all these cases the source code is written in C# highlighting the software's potential to reduce XR fragmentation where developers otherwise need to write separate code for each device. Unity's cross-platform capabilities are best illustrated by the architectural diagram shown in Fig. 1.

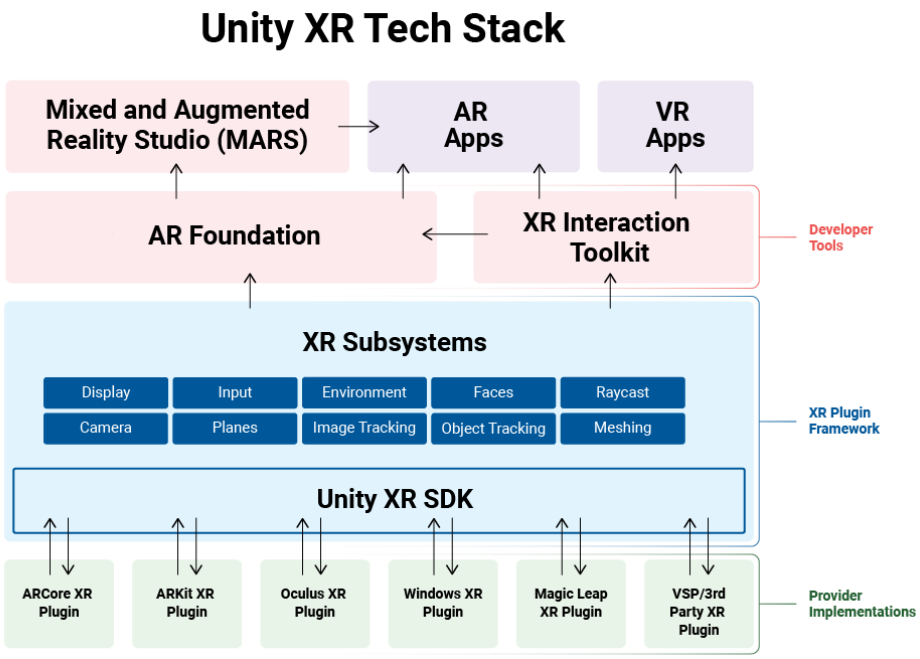


Fig. 1. Unity XR development framework architecture diagram [46]

The Unity's XR architecture represents a multi-layered toolkit that sits between AR/VR hardware (*phones, headsets*) and the immersive experience being developed. The bottom layer represents the platform specific plugins which provide access to the particular device features. The subsystems layer is a unified framework wrapping up all the device-specific plugins with common features like Display, Input, Camera, Plane detection, Image Tracking, Raycasting, Mesh generation, Face tracking, etc. The

code written for these subsystems works on any device plugin from the bottom layer. The AR foundation builds upon the subsystems allowing, for example, to place virtual objects on detected planes, tracking images, etc. The XR Interaction Toolkit provides a set of premade components for grabbing, pointing, etc. It also supports hand-tracking, touch inputs and other mechanisms for interaction with the immersive environment. At the top level, Unity's mixed and augmented reality studio (MARS) allows developers to create XR applications by simply adding device-specific plugins, writing application logic with the XR Interaction Toolkit and deploying the immersive experience to any supported XR device.

Unreal Engine is another well-established game engine that delivers photorealistic visualizations essential for high-fidelity aircraft maintenance training simulations [47]. It features a sophisticated rendering system and compatibility with leading XR headsets. Cross-platform development is facilitated through the OpenXR standard, supported device plugins, and built-in XR development templates.

In the second category two platforms stand out in the context of XR aircraft maintenance training: PTC Vuforia and HTX Labs EMPACT. PTC Vuforia provides a comprehensive XR development environment specifically optimized for industrial and maintenance training applications [48]. Key features of this platform are the Vuforia Engine for XR recognition and tracking, Vuforia Studio for efficient XwR experience authoring and Vuforia Expert Capture for workflow recording. The system provides options for internet-of-things (IoT) data connectivity as well as options for cloud deployment, enterprise-grade security and scalability. It could be easily integrated with CAD systems by using the software platform PTC Creo [49]. The EMPACT software of the HTX Labs is a cloud-based immersive training platform specifically designed for military and aviation applications [50]. Some of its distinguished features are the support for AI-enabled virtual training environment, broad spectrum of devices to deploy on, built-in performance tracking and analytics, comprehensive user management system.

3.2. Geometry model recommendations and XR design guidelines

Development of XR immersive experiences relates on well-established rules and guidelines designed to provide standardization and minimize XR fragmentation by applying both open [51] and specially developed standards [52, 53, 54, 55]. When designing XR immersive experience all assets must be modeled in real scale by avoiding unnecessary occlusion as well as jittering or incorrect physics for the non-static objects. The surface defining triangles should remain below 1000 for simple assets, 2000-5000 for mid-polygon details and up to 10000-50000 for very complex objects. It is a common practice the surfaces that are never seen by the camera to be removed. It is recommended to provide at least two levels of detail (LODs) for optimization of the overall performance. The designed models should be centered in the CAD workspace coordinate system. It is recommended that the texture size in general should be smaller than 2048 px (2K) or 1024 px (1K) if the latency of the immersive experience is to be reduced. A common practice is to keep the 3D model file size less than 25 MB.

The final models are commonly exported in two basic formats: FBX or GLTF/GLB. The binary FilmBox 3D data exchange file format (FBX) contains all the geometric definition and mesh data for the model. It is appropriate for detailed, high-fidelity projects requiring advanced animation or rigging. This file format doesn't contain any textures. Instead, all texture images must be placed into a separate Texture folder at the same hierarchy level as the FBX file. The text-based graphics library transmission format (GLTF) contains not only the geometry description but also all of its textures and mesh data into a single file. It is suitable for optimized workflows, especially when working with physically-based rendering (PBR) materials and lightweight assets. This type of file can be transferred in binary format between CAD and XR platforms using the GLB format.

The interactable targets have a minimum size (MS) and a recommended size (RS) based upon the intended interaction distance (ID):

$$MS = ID * 0.868 * 48$$
$$RS = ID * 0.868 * 56$$

where 0.868 is a factor, used to convert the density-independent pixels (dp) that provide uniform dimensions of the objects on any screen into a distance-independent millimeters (dmm), which represent a unit that remain constant regardless of the distance between a user and a virtual object. By definition 1 dmm equals 1 millimeter viewed from 1 meter away. The default value for the interaction distance is 1.75 m.

Standards suggest that the XR experience field of view should be 50°–90° with effective scene view from 30° to 50° in diagonal. All critical information (step indicators, warning) should remain in the central 30°×20° zone to ensure reliable perception and reduce head scanning. Since all current optical waveguide headsets impose a single focal plane laying at approximately 1.8-2.0 meters it is recommended to place the interactive holograms and overlays within 1.25 - 5 meters to minimize eye strain. Also jumps greater than 0.5 meters across focal depths should be avoided. The immersive experience tasks should be designed so that primary objects lie from 0.5 to 3 meters from the user's eyes. The most frequently manipulated targets must be placed at a distance greater or equal to 1 meter in order to prevent neck flexion beyond 20° and reduce posture fatigue. For accurate and quick selection of objects by gazing, hand gesture or controller the selection targets need to be about 1.5° to 2° of the field of view or 5-10 cm at 2 meters away. This size ensures over 90% hit accuracy even in busy or cluttered environments. Labels and warnings need to be at least 0.33° of the available field of view or about 12 mm tall characters at 2 meters so that they can be quickly and easily read, especially in bright factory lighting. The recommended angular resolution is at least 20 pixels per degree (ppd) in order to keep the images clear. Most modern devices achieve 40–60 ppd which preserves edge detail in line drawings, schematics and small numeric readouts. The immersive overlays should have at least 1000 nits of brightness in order to stay visible in different lighting conditions. They also need a contrast of at least 1000:1 against backgrounds, whether dark or shiny, to ensure easy reading. The delay between the trainee movements and the corresponding update on the display (motion-to-photon latency) should be less than 20 milliseconds with frame rates remaining above 60 fps for smooth performance. If these limits are exceeded it can lead to motion sickness and make virtual images appear unstable. Virtual assets should properly appear in front of or behind real objects with smooth changes in detail to facilitate the depth perception. Sudden appearance or detail changes closer than 1 meter from the user should be avoided to prevent distractions. Audio feedback like voice prompts should have very low delay (under 10 ms) and clear sound in the 500 Hz to 4 kHz range. Haptic feedback should use light vibrations between 100 and 250 Hz lasting less than 200 ms.

3.3. XR hardware

The technical characteristics and features of the extended reality platform hardware must align with both the training system's objectives and the desired cost-effectiveness. The available headsets traditionally employ inside-out simultaneous localization and mapping (SLAM) using cameras and sensors to create precise, real-time environmental maps. They can utilize cameras and artificial intelligence (AI) algorithms to track hand and eye movements for detailed interaction. Depth sensors combined with AI enable understanding of objects and their behaviors within the environment. Untethered headsets require dedicated onboard processors such as a specialized holographic processing unit (HPU) to operate smoothly without cables. Examples include Snapdragon XR platforms and the HoloLens 2's combination of custom-built HPU and Snapdragon 850. High-speed wireless connections are essential for linking headsets to external computers where technologies like 5 GHz Wi-Fi could provide multi-gigabit data rates for smooth performance. An alternative approach is to reduce the device's processing load by delegating resource-intensive tasks like spatial mapping and AI to cloud-based services. The headsets are expected to support 2 to 3 hours of continuous use under heavy workload with fast charging capabilities achieving a full recharge in under 70 minutes and standby times of up to two weeks. For comfort during extended wear, headsets should weigh less than 600 grams and feature balanced designs. Adjustable elements like interpupillary distance (IPD) calibration and compatibility with glasses are necessary, along with effective thermal management through passive cooling and heat sinks to prevent overheating and sensor drift. Table 2 summarizes key features of several leading XR headsets in the industry.

Table 2. Leading enterprise-focused AR and MR devices relevant for aircraft maintenance

AR/MR device		Field of view	Resolution, [pixels per eye]	Weight, [g]	Sensors	Release year, price
Meta Quest 3S 	MR	96° x 90°	1832 x 1920	514	2 4MP RGB cameras, 4 VGA sensors (*inside-out tracking*), 2 flood LEDs, software depth sensor	2024 400 $ (*256GB*)
Apple Vision Pro 	MR	100° x 73°	3660 x 3200	600-650 (*excluding battery*)	Two high-resolution main cameras, six world-facing tracking cameras, four eye-tracking cameras, TrueDepth camera, LiDAR scanner, four IMUs, flicker sensor, ambient light sensor	2024 3499 $
Pico 4 Ultra Enterprise 	MR	105°	2160 x 2160	591	Dual 32MP RGB cameras, iToF depth-sensing camera, 4 environment tracking cameras, 5 internal cameras. Accelerometer, gyroscope, magnetometer.	2024 799 $
Varjo XR-4 Series 	MR	120° x 105°	3840 x 3744	665 (*without headband*)	Dual 20-megapixel cameras (*33 PPD*), extended range LiDAR (*7m range*), 200 Hz eye tracking, ambient light reconstruction cameras, inside-out tracking	2023 3990$
Microsoft HoloLens 2 	MR	43° x 29° 52° diagonal	1440 x 936 / 2048 x 1080	566	Eye tracking, spatial tracking, hand tracking, 4 integrated cameras *NOTE: Production discontinued in October 2024 with software support and critical security updates provided until December 31, 2027.*	2019 3500 $
MRTech 5G Explosion-Proof AR Glasses 	MR	N/A (*virtual screen*)	N/A (*virtual screen*)	350	AI-powered hazard detection & environmental monitoring sensors, multi-gas detection, thermal imaging capabilities, 3-degree of freedom head tracking, 3-axis gyro, accelerometer & integrated compass	2025 N/A
DigiLens Argo 	AR	30° diagonal	640 x 480	185	48MP RGB camera (*autofocus, lossless digital zoom*), multi-microphone system, inside-out tracking (*3/6 DoF, SLAM*), AI-powered hazard detection & environmental monitoring sensors	2024 N/A
Vuzix M4000 	AR	28°	854 x 480	222 (*without headstrap*)	12.8MP 4K 30fps camera (*auto-focus, LED flash*), 3-DoF head tracking (*3-axis gyro, accelerometer, integrated compass*), triple noise-cancelling microphones	2020 2500 $
Iristick G2 	AR	117° diagonal	426 x2 40	78	16MP camera (*6x optical zoom, liquid lens for fast autofocus*), 9-axis sensor (*accelerometer, gyroscope, compass*), dual microphones with beamforming	2021 2099 $
Holochip H50 AR Goggles 	AR	35° x 35° 50° diagonal	1440 x 1440	220	AI-enabled gesture recognition, spatial awareness, rendering, sensor input for data collection.	2025 N/A

The XR maintenance training system for military aviation must feature rugged MIL-SPEC hardware that can withstand bumps, dust, extreme temperatures and electromagnetic interference. The XR hardware for the Air Force maintenance training must be designed to meet the rigorous environmental and durability requirements of MIL-STD-810G/H, ensuring reliable performance under extreme conditions. Additionally, it should comply with ANSI Z87.1:2020 for impact-resistant safety eyewear and hold an IP65 rating for complete dust protection and low-pressure water ingress resistance. The system must deliver ultra-low-latency and centimeter-accurate tracking so that virtual overlays never drift off target. AR/MR systems like the U.S. Army's integrated visual augmentation system (IVAS) improve situational awareness but should be reliable to avoid mission failure [56]. These systems have to include a modular content pipeline that lets the instructors to rapidly build and update new training scenarios. The platform must integrate both the established learning management system (LMS) and logistics tools via well-established standards like the sharable content object reference model (SCORM) or Experience API (xAPI) in order to eliminate double-entry and data gaps. When training on aviation systems documented in a foreign language it is essential for the maintenance terminology to be accurate and consistent. [57]. Headsets for the maintenance training have to be lightweight, balanced and ergonomically designed with anti-fog optics and quick-release fittings to keep technicians comfortable all day. The user interface should remain minimal and context-sensitive showing only the current step without clutter or deep-level menus. Controls must be hands-free and reliable, combining accurate voice recognition in noisy hangars with gesture support for gloved hands. The system needs full offline capability with automatic syncing once back on the network plus peer-to-peer data sharing. It must be provided with real-time analytics dashboards with automated alerts on key metrics like time-to-proficiency, common failure points and help-request frequency. Data security must meet enterprise-grade standards, encryption at rest and at usage, multi-factor authentication and role-based access controls. There must be well-defined metrics to evaluate the probability of cybersecurity threats [58]. To support future expansion and development of the system it must include open application programming interfaces (APIs) as well as clear upgrade paths so that new features can be added without replacing the existing infrastructure.

4. XR for training effectiveness validation

4.1. General metrics for effectiveness validation

While traditional real-hardware drills have long been considered the benchmark for developing technical proficiency recent studies indicate that better cognitive and motor learning outcomes can be achieved through a blended approach which integrates live hardware exercises with high-fidelity simulators and collaborative debriefing sessions. However, the effectiveness of this approach depends not only on the equipment precision but also on the quality of the instructions, the accurate assessment of the results and the instructor’s expertise. On the other hand, these systems introduce their own challenges, like the high cost of the hardware and its maintenance, the complex process of adaptation to an existing and already established training program as well as the risk of ergonomic and user-acceptance problems like heavy or uncomfortable headsets, motion-sickness, etc. Another issue of these systems related to their application in the military is the security of the used computer networks, especially when the training materials, usage metrics and final assessments are stored on a remote server or accessed via cloud.

A general question related to the XR training systems for aviation maintenance is the recognition and approval of the XR-based training hours from the regulators. According to FAA [59] the XR device represents a tool for interactive computer-based/multimedia training (ICBT) which can cover theory, component location, fault diagnosis and procedural familiarization, but the full curriculum of hands-on classes on real aircraft is mandatory to develop critical manual skills. XR can’t substitute for physically routing cables or using a wrench with real torque readings. Same approach is adopted in EASA [60], where XR falls into the category of synthetic training devices. According to EASA regulations the standard training method involves instructors delivering lectures using a blackboard and training manuals. Blending this approach with multimedia-based training and virtual reality is recommended. However, the XR training can only support theoretical knowledge, fault-diagnosis and procedural drills

on the complex aircraft systems. It cannot satisfy the basic manual-skill requirements (*riveting, drilling, wiring*) where true hands-on practice is mandatory.

In this regard, integrating XR into aircraft maintenance training requires a comprehensive validation process based on clear effectiveness metrics. First, objectives should be precisely defined in order to extract relevant performance data from the system's detailed logs. Then, the evaluated outcomes should be compared between an XR-trained group and a traditional hands-on control group to assess the true impact of the XR integration. In this aspect several key metrics are proposed in Table 2 for evaluating the effectiveness of the XR training system.

Table 2. Metrics for evaluation of XR training system effectiveness

Category	Metric	Data to log from XR	Description
Task performance	**Task completion time (TCT)** $TCT = t_{end} - t_{start}$ where: • t_{start} is timestamp when the trainee begins the task; • t_{end} is timestamp when the task is completed.	Time period between the moments when a trainee starts and finishes each procedure or sub-step.	Identifies UI problems or confusion points.
	Error rate (ER) $ER = E/S$ where: • E is the number of committed errors (*critical, minor, etc.*); • S is the total number of procedural steps. Can be further classified as: $ER_{critical} = E_{critical}/S$ $ER_{minor} = E_{minor}/S$	Number and category of mistakes (*wrong value, skipped part, etc.*).	Shows which steps are most error-prone in XR versus traditional training.
	Rework ratio (RR) $RR = R/S$ where: • R is the number of steps repeated or redone; • S is the total number of steps in procedure.	How many times a trainee backtracks or repeats a step.	Indicates misunderstanding.
	Help dependency index (HDI) $HDI = H/S$ where: • H is the number of system-invoked help requests (*tooltips, overlays, etc.*); • S is the total completed steps.	Number of replays, voice prompts or hints invocations.	Reveals over-reliance on XR system assistance.
Learning and retention	**Learning gain (LG)** $LG = \frac{P_{post} - P_{pre}}{P_{max} - P_{pre}}$ where:	Comparison of the same metrics (*time, errors*) across repeated practice sessions.	A true learning gain shows a downward trend in errors and time,

	• P_{pre} is the score before training; • P_{post} is the score after training; • P_{max} is the maximum possible score.		plateauing at an expert level.
Category	**Metric**	**Data to log from XR**	**Description**
Learning and retention	**Retention rate (RR)** $RR = P_{delayed}/P_{post}$ where: • $P_{delayed}$ is the score from delayed post-test; • P_{post} is the score from immediate post-test.	Trainees assessment after XR training using physical mock-ups without AR assistance: immediately and again after a delay (*for example, 1 month*).	Validation of skill transfer and medium- to long-term retention.
Process adherence	**Sequence compliance rate (SCR)** $SCR = S_{correct}/S$ where: • $S_{correct}$ is the number of steps performed in the correct order; • S is the total number of steps.	Strict following of the standard operating procedure (*SOP*) step-by-step instructions.	Deviations from the correct sequence are high-risk in maintenance.
Cost-effectiveness	**Cost per competent trainee (CCT)** $CCT = C_{total}/N_{qualified}$ where: • C_{total} : total training cost (*hardware, software, instructor time, etc.*); • $N_{qualified}$: number of trainees meeting competency threshold.	Relative cost for proficient technician training.	Measures bottom-line training efficiency: the AR should drive down the cost-per-qualified-tech
Additional metrics	**Dwell time** (*ergonomic metric*)	How long the trainee looks at a particular component or instruction panel.	Highlights confusing UI or labeling
	Control-group benchmark (*comparative metric*)	Same metrics for XR training vs. traditional group hands-on training with added instructor hours required and material consumption.	Provides rigorous proof of XR's added value.

To illustrate how these metrics are applied, let's examine a scenario in which a trainee completes a 20-step XR training session, resulting in the following summarized data:

- start/end time: 0 / 1800 s;
- number of total errors E=3, from which 1 error is classified as critical;
- number of reworked steps: R = 4;
- number of help requests: H = 6;
- pre-training score: 60/100, post-training: 85/100, delayed post-test: 80/100;

- number of correctly sequenced steps: 18;
- total XR system cost: 500 000 $;
- total number of trainees certified:120.

The evaluated metrics for the XR-training system effectiveness estimation are presented in Table 3. The statistical analysis of each of these metrics for an individual trainee or a group provides valuable insights for improvement of the XR system or its utilization. Descriptive statistics (*mean, variance*) show the performance of the typical trainee and how much it varies. For example, if task completion times (TCT) vary by ±50% there is a probability that the XR workflow isn't consistent, for example, some overlays are confusing or take too long for loading. However, any XR system modifications based on trends observed in descriptive statistics should be validated using 95% confidence intervals to ensure the effects are statistically reliable. Without such validation, conclusions drawn from small or isolated samples risk being misleading due to random variation. On the other hand, the XR system should be validated with a control group with traditional hands-on training to prove its effectiveness.

Table 3. Evaluation of XR-training system effectiveness metrics

Metric	***Formula***	***Value***
Task completion time (TCT)	$t_{end} - t_{start}$	1800 - 0 = **1800 s**
Error rate (ER)	E/S	3 / 20 = **0.15**
Critical error Rate	E_{crit}/S	1 / 20 = **0.05**
Rework ratio (RR)	R/S	4 / 20 = **0.20**
Help dependency index (HDI)	H/S	6 / 20 = **0.30**
Learning gain (LG)	$\frac{P_{post} - P_{pre}}{100 - P_{pre}}$	(85-60)/(100-60) = **0.625**
Retention rate	$P_{delayed}/P_{post}$	80/85 ≈ **0.941**
Sequence compliance rate (SCR)	$S_{correct}/S$	18/20 = **0.90**
Cost per competent trainee (CCT)	$C_{total}/N_{qualified}$	500000/120 ≈ **$4167 $**

4.2. Military-based metrics for effectiveness validation

Several key metrics are proposed as relevant to XR maintenance training system from a military standpoint:

- **time-to-proficiency**: calendar days or training hours until a trainee passes a defined "ready for duty" assessment on real hardware;
- **training rate**: number of technicians fully trained per month (or year) with existing resources;
- **cost-per-qualified-technician**: total training cost (all hardware, software licenses, instructor labor, facility usage) divided by the number of trainees who meet certification;
- **post-training error rate**: frequency of maintenance mistakes or rework events on operational aircraft in the first 30–90 days after training;
- **mission capable rate**: percentage of the aircraft that's mission-ready at any given time;
- **instructor load**: instructor hours per trainee to reach proficiency.

Once the XR training metrics have been evaluated in a separate pilot training group and the approach proven effective it can be rolled out to other trainee groups and applied to different systems or aircraft. The XR training system development should aim to accelerate the certification time as well as to increase the number of trainees with minimal cost-per-qualified technician. The training should result in decrease of field-level errors and reworks and an increase in the number of mission capable aircrafts. The required instructor time should be reduced as well as the hours for practice on the real hardware. There should be no regression in performance or tool usage skills.

5. Conclusions

This paper demonstrates that XR technologies can improve military aircraft maintenance training when guided by clear objectives, adaptive and scalable development process as well as rigorous validation metrics. Studies reviewed confirm that XR can significantly improve precision, streamline processes and reduce training costs. A wide range of tools exists for designing and developing XR training systems - from specialized platforms to free, cross-platform development software. XR training systems should be designed to adapt to particular maintenance training program while adhering to established XR design guidelines and standards.

XR hardware production continues to expand across both consumer and industrial sectors. For aircraft maintenance training this hardware must be durable, ergonomically designed and supported by the official manufacturer. The integration of XR systems into aviation maintenance training follows a controlled, step-by-step process of evaluating metrics to validate effectiveness. This paper proposes both general and military-specific effectiveness metrics.

Despite all the advanced features and technological improvements of XR systems they cannot fully replace hands-on aviation maintenance training and are primarily suited for supporting theoretical modules. Relying on XR for complete practical training poses risks of developing improper habits and inaccurate spatial orientation. While XR technology enhances training efficiency hands-on skills and teamwork remain essential.

References

1. Statista, “Mobile augmented reality (AR) market revenue worldwide from 2023 to 2028,” Statista, Mar. 2024. [Online]. Available: https://www.statista.com/statistics/282453/mobile-augmented-reality-market-size. [Accessed: May 20, 2025]

2. Statista, “Mobile AR market revenue,” Statista, 2024. [Online]. Available: https://www.statista.com/statistics/282453/mobile-augmented-reality-market-size. [Accessed: May 20, 2025]

3. Li, S., “Design and development of aviation aircraft maintenance training platform based on VR technology,” Procedia Computer Science, Vol. 228, 2023, pp. 898–906. https://doi.org/10.1016/j.procs.2023.11.118

4. Gómez-Cambronero, Á., Miralles, I., Tonda, A., and Remolar, I., “Immersive virtual-reality system for aircraft maintenance education: A case study,” Applied Sciences, Vol. 13, No. 8, Apr. 2023, Art. no. 5043. https://doi.org/10.3390/app13085043

5. Anoop, G., “Efficacy of virtual reality-based simulations in training aviation maintenance technicians on maintenance procedures,” M.S. thesis, Ind. Eng. Dept., Clemson Univ., Clemson, SC, May 2024.

6. Wu, W.-C., and Vu, V.-H., “Application of virtual reality method in aircraft maintenance service—Taking Dornier 228 as an example,” Applied Sciences, Vol. 12, Jul. 2022, Art. no. 7283. https://doi.org/10.3390/app12147283

7. Tian, Y., Liu, S., Li, M., Yin, R., and Liu, H., “Research on 3D virtual training courseware development system of civil aircraft based on virtual reality,” in Proc. ACM Int. Conf. Artificial Intelligence and Virtual Reality (AIVR), Singapore, Jul. 2019, pp. 21–25. https://doi.org/10.1145/3348488.3348495

8. Starr, L. T., Shorts, K., and Vans, M., “Interactive aviation maintenance classroom,” in Engineering Reality of Virtual Reality, IS&T Int. Symp. Electron. Imaging, 2024, pp. 182–1182-7.

9. Caudell, T. P., and Mizell, D. W., “Augmented reality: An application of heads-up display technology to manual manufacturing processes,” in Proc. Hawaii Int. Conf. System Sciences, Jan. 1992, pp. 659–669. https://doi.org/10.1109/HICSS.1992.183317

10. Xue, Z., et al., “AR-assisted guidance for aviation,” Applied Sciences, Vol. 14, No. 3, 2024. https://doi.org/10.3390/app14031137

11. Georgiev, Y., “Automation of Production Process of Automotive Wire Harnesses”, Proceedings of International science and technical conference „Automation of Discrete Production Engineering”, Technical University - Sofia, No. 4, 2022, pp. 79–83. ISSN 2682-9584

12. McNeely, J. R., “X-RAY VISION: Application of augmented reality in aviation maintenance to simplify tasks inhibited by occlusion,” M.S. thesis, Dept. Comput. Sci., Naval Postgraduate School, Monterey, CA, USA, 2022. [Online]. Available: https://calhoun.nps.edu/handle/10945/70007.

13. Siyaev, A., and Jo, G.-S., “Towards aircraft maintenance metaverse using speech interactions with virtual objects in mixed reality,” Sensors, Vol. 21, Mar. 2021, Art. no. 2066. https://doi.org/ 10.3390/s21062066

14. De Crescenzio, F., Fantini, M., Persiani, F., Di Stefano, L., Azzari, P., and Salti, S., “Augmented reality for aircraft maintenance training and operations support,” IEEE Computer Graphics and Applications, Vol. 31, No. 1, Jan./Feb. 2011, pp. 96–101. https://doi.org/10.1109/MCG.2011.4

15. Eschen, H., Kotter, T., Rodeck, R., Harnisch, M., and Schüppstuhl, T., “Augmented and virtual reality for inspection and maintenance processes in the aviation industry,” Procedia Manufacturing, Vol. 19, 2018, pp. 156–163. https://doi.org/10.1016/j.promfg.2018.01.022

16. Airbus, “Airbus Group Unit Testia to supply augmented reality system to Spirit AeroSystems,” Airbus, Oct. 2021. [Online]. Available: https://www.airbus.com/en/newsroom/press-releases/2016-04-airbus-group-unit-testia-to-supply-augmented-reality-system-to. [Accessed: May 19, 2025]

17. Airbus, “Mixed reality trainer for A350 XWB,” Airbus, 2021. [Online]. Available: https://www.airbus.com/en/newsroom/press-releases/2017-11-airbus-develops-worlds-first-mixed-reality-trainer-for-a350-xwb. [Accessed: May 19, 2025]

18. XR Today, “23,000 Emirates cabin crew professionals to undergo Airbus and Boeing VR training,” XR Today, Oct. 2024. [Online]. Available: https://www.xrtoday.com/virtual-reality/23000-emirates-cabin-crew-professionals-to-undergo-airbus-and-boeing-vr-training/.[Accessed: May 19, 2025]

19. Kamińska, D., Sapiński, T., Wiak, S., Tikk, T., Haamer, R. E., Avots, E., Helmi, A. H., Ozcinar, C., and Anbarjafari, G., “Virtual reality and its applications in education: Survey,” Information, Vol. 10, No. 10, Oct. 2019, p. 318. https://doi.org/10.3390/info10100318

20. Guo, Z., Zhou, D., Zhou, Q., Zhang, X., Geng, J., Zeng, S., Lv, C., and Hao, A., “Applications of virtual reality in maintenance during the industrial product lifecycle: A systematic review,” Journal of Manufacturing Systems, 2020, in press. https://doi.org/10.1016/j.jmsy.2020.07.007

21. Rostami, M., et al., “A comprehensive review of extended reality and its application in aerospace engineering,” Progress in Aerospace Sciences, May 2025, p. 101118. https://doi.org/10.1016/j.paerosci.2025.101118

22. Air & Space Forces Association, “MOTAR BELLE Labs overview,” Air & Space Forces. [Online]. Available:https://www.airandspaceforces.com/app/uploads/2020/06/MOTAR_BELLE-Labs-Overview_4 - May-2020_V2.pdf. [Accessed: May 19, 2025]

23. Kambushev, K., “Izsledvane na nadezhdnostta na nevronni mrezhi izpolzvani za aproksimirane na funktsii”, Proceedings of XXIX International science and technical conference „Automation of Discrete Production Engineering”, Technical University - Sofia, 2020, pp. 107–109. ISSN 2682-9584

24. Wang, W., et al., “Augmented reality in maintenance training for military equipment,” Journal of Physics: Conference Series, Vol. 1626, 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1626/1/012184

25. Dassault Systèmes, “Maintenance,” Dassault Systèmes. [Online]. Available: https://www.3ds.com/products/delmia/augmented-experience/maintenance. [Accessed: May 19, 2025]

26. Microsoft, “Microsoft stops HoloLens 2 production, support to end in 2027,” Microsoft. [Online]. Available:https://learn.microsoft.com/en-us/answers/questions/2151213/microsoft-stops-hololens-2-production-support-to-e. [Accessed: May 19, 2025]

27. Taqtile, “Jet engine maintenance training with augmented reality,” Taqtile. [Online]. Available: https://taqtile.com/case-studies/jet-engine-maintenance/.[Accessed: May 19, 2025]

28. EON Reality, “US Air Force maintenance training in AR and VR,” EON Reality. [Online]. Available: https://eonreality.com/us-air-force-maintenance-training-in-ar-and-vr/.[Accessed: May 19, 2025]

29. EON Reality, “EON-XR platform,” EON Reality. [Online]. Available: https://core.eon-xr.com/. [Accessed: May 19, 2025]

30. PTC, “Vectrona immersive augmented reality training with US Airforce,” PTC. [Online]. Available: https://www.ptc.com/en/case-studies/vectrona-immersive-augmented-reality-training-with-us-airforce. [Accessed: May 19, 2025]

31. Sheppard AFB, “Sheppard AFB hosts XR Working Group,” Sheppard AFB. [Online]. Available: https://www.sheppard.af.mil/News/Article-Display/Article/3649586/sheppard-afb-hosts-xr-working-group/. [Accessed: May 19, 2025]

32. L3Harris, “Virtual maintenance trainer,” L3Harris. [Online]. Available: https://www.l3harris.com/ all-capabilities/virtual-maintenance-trainer. [Accessed: May 19, 2025]

33. Thales Group, “Training & simulation,” Thales Group. [Online]. Available: https://www.thalesgroup.com/en/markets/specific-solutions/training-simulation. [Accessed: May 19, 2025]

34. Lufthansa Technik, “Augmented reality,” Lufthansa Technik. [Online]. Available: https://www.lufthansa-technik.com/en/augmented-reality. [Accessed: May 19, 2025]

35. Airbus, “Stepping into the virtual world to enhance aircraft maintenance,” Airbus, Feb. 2019. [Online]. Available: https://www.airbus.com/en/newsroom/stories/2019-02-stepping-into-the-virtual-world-to-enhance-aircraft-maintenance. [Accessed: May 19, 2025]

36. Rheinmetall, “Extended reality,” Rheinmetall. [Online]. Available: https://www.rheinmetall.com/en/products/extended-reality-digital-worlds/extended-reality. [Accessed: May 19, 2025]

37. Aviation Week Network, “Augmented reality, robotics tools drive efficiencies for Safran,” Aviation Week. [Online]. Available: https://aviationweek.com/mro/emerging-technologies/augmented-reality-robotics-tools-drive-efficiencies-safran. [Accessed: May 19, 2025]

38. Le Réseau des Carnot, “Diota becomes first French publisher of augmented reality software,” Le Réseau des Carnot. [Online]. Available: https://www.lereseaudescarnot.fr/en/carnot-partnership-success-stories/diota-becomes-first-french-publisher-augmented-reality-software. [Accessed: May 19, 2025]

39. Dassault Aviation, “La Fabrique Défense,” Dassault Aviation. [Online]. Available: https://www.dassault-aviation.com/en/group/news/la-fabrique-defense/. [Accessed: May 19, 2025]

40. Italian Defence Technologies, “Training 4.0: Augmented, virtual and immersive reality in aircraft maintenance training,” Italian Defence Technologies, Jun. 2021. [Online]. Available: https://www.italiandefencetechnologies.com/training-4-0-augmented-virtual-and-immersive-reality-in-aircraft-maintenance-training/. [Accessed: May 17, 2025]

41. Brar, A., “China’s state media shows military using Microsoft’s HoloLens 2 headsets,” Newsweek, Dec. 2023. [Online]. Available: https://www.newsweek.com/china-peoples-liberation-army-microsoft-hololens2-mixed-reality-headsets-1852381

42. Simbott, “VR aviation training,” Simbott. [Online]. Available: https://simbott.com/vr-aviation-training/. [Accessed: May 20, 2025]

43. Velichko, M., “How augmented reality is used in commercial aviation,” Jasoren. [Online]. Available: https://www.jasoren.com/augmented-reality-in-commercial-aviation/. [Accessed: May 27, 2025]

44. Heartwood, “3D interactive tech talk – Augmented reality training for Japan Airlines, Schneider Elec,” Heartwood Blog, 2024. [Online]. Available: https://hwd3d.com/blog/3d-interactive-tech-talk-ar-japan-air-schneider/. [Accessed: May 27, 2025]

45. Unity, “AR & VR solutions for government & aerospace industry,” Unity. [Online]. Available: https://unity.com/solutions/government-aerospace. [Accessed: May 26, 2025]

46. Unity Technologies, “Unity – Manual: XR architecture,” Unity Technologies. [Online]. Available: https://docs.unity3d.com/6000.1/Documentation/Manual/XRPluginArchitecture.html. [Accessed: May 26, 2025]

47. Unreal Engine, “Beyond the manual: VR training on aircraft maintenance, ” Unreal Engine. [Online]. Available: https://www.unrealengine.com/en-US/spotlights/beyond-the-manual-vr-training-on-aircraft-maintenance. [Accessed: May 26, 2025]

48. PTC, “The U.S. Air Force adopts immersive mixed reality training to fuel better engagement, learning retention, and outcomes,” PTC, Nov. 2024. [Online]. Available: https://www.ptc.com/en/case-studies/vectrona-immersive-augmented-reality-training-with-us-airforce. [Accessed: May 27, 2025]

49. PTC, “Creo: design the way it should be,” PTC, Apr. 2025. [Online]. Available: https://www.ptc.com/en/products/creo. [Accessed: May 27, 2025]

50. Johannavirtanen, “Mastering machinery with mixed reality: How HTX Labs and Varjo are transforming aircraft maintenance training,” Varjo.com, May 2024. [Online]. Available: https://varjo.com/case-studies/mastering-machinery-with-mixed-reality-how-htx-labs-and-varjo-are-transforming-maintenance-training/. [Accessed: May 27, 2025]

51. The Khronos Group, “OpenXR – High-performance access to AR and VR —collectively known as XR— platforms and devices,” The Khronos Group, Dec. 2016. [Online]. Available: https://www.khronos.org/openxr/. [Accessed: May 27, 2025]

52. ISO/IEC TR 23844:2023, ISO, 2023. [Online]. Available: https://www.iso.org/standard/ 77144.html

53. ISO/IEC TR 23843:2020, ISO, Oct. 2020. [Online]. Available: https://www.iso.org/standard/ 77143.html

54. IEEE 2048 VR/AR Working Group (VRARWG) – Home, IEEE, [Online]. Available: https://sagroups.ieee.org/2048wg/

55. IEEE Standards Association, “IEEE Standards Association,” IEEE Standards Association. [Online]. Available: https://standards.ieee.org/ieee/3322/11073/

56. Army University Press, “The tactical considerations of augmented and mixed reality implementation,” Army University Press. [Online]. Available: https://www.armyupress.army.mil/Journals/Military-Review/English-Edition-Archives/May-June-2022/Kallberg/. [Accessed: May 27, 2025]

57. Katsarska, V., “Convergin Pathways: The role of English in aviation maintenance training,” in Interwoven Threads: Exploring the Nexus of Languages, Security, and Technology, Georgi Benkovski Publishing Complex, 2024, pp. 146–153. ISBN 978-954-713-166-8 (print); ISBN 978-954-713-167-5

58. Hubenov, D., Zheleva, P., and Shehov, Ch., “Determining the probability of occurrence of a threat event in information systems in the Republic of Bulgaria and other countries,” in Proc. “20 years Bulgaria in NATO and NATO in Bulgaria” Sci. Conf. with Int. Participation, Rakovski Natl. Defence Coll., Sofia, 2024, pp. 197–209. ISBN 978-619-7711-52-3 (online)

59. Federal Aviation Administration, “Advisory Circular (AC) 147-3C: Certification and operation of aviation maintenance technician schools,” FAA, Mar. 2022. [Online]. Available: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentID/1041282

60. European Aviation Safety Agency, Regulation (EU) No 1321/2014, Annex IV – Part 147: Maintenance Training Organizations, EASA, 2014. [Online]. Available: https://www.easa.europa.eu/ en/document-library/regulations/regulation-eu-no-13212014

Технологии с разширена реалност за обучение по техническа експлоатация на въздухоплавателните средства

Николай Кънчев

Висше военновъздушно училище „Георги Бенковски“, факултет „Авиационен“,
гр. Долна Митрополия, България, nikolaikanchev@yahoo.com

Резюме: Интензивното развитие на технологиите за разширена реалност води до възникване на потенциал за тяхното приложение в процеса на обучение на инженерно-технически състав при усвояване на конструкцията и експлоатацията на въздухоплавателните средства. Възможностите за интерактивност и визуализация чрез добавена и смесена реалност за провеждане на голям брой повторения без необходимост от реалния летателен апарат, както и за асистирано обучение без инструктор позволяват значително повишаване на ефективността на обучението. В статията е направен обзор на съществуващите системи с разширена реалност за обучение по техническа експлоатация, на програмните средства за разработване на платформи за обучение с разширена реалност, както и на актуалния хардуер в това направление. Поставен е акцент върху приложенията и изискванията към такъв тип системи във военната авиация. Предложени са показатели за оценка и валидиране на ефективността на обучението, проведено чрез система с разширена реалност.

Two-step structural optimization procedure for quadcopter frames

Nikolay Kanchev

Bulgarian Air Force Academy, Aviation Faculty, Dolna Mitropolia, Bulgaria, nikolaikanchev@yahoo.com

Abstract: This paper introduces a two-step structural optimization procedure for lightweight quadcopter frames that integrates solid isotropic material with penalization (SIMP) and level-set methods. Initially, an arm submodel exposed to combined thrust and bidirectional torque underwent SIMP-based topology optimization. This process resulted in a coarse material distribution, achieving a 60% reduction in mass for each arm and a 46% decrease in the total frame weight while maintaining a substantial safety factor. Subsequently, the SIMP output was refined using a level-set algorithm that sharpens boundaries and optimizes stiffness, further reducing arm mass by up to 75% (58.2% overall frame weight reduction) with a safety factor of approximately 14 and minimal tip deflection. The optimized arm was post-processed to ensure manufacturability and validated through additional finite-element analyses. The proposed two-step structural optimization methodology facilitates the development of lightweight, robust quadcopter structures suitable for additive manufacturing that enhance UAV endurance and payload capacity.

Keywords: *multicopter, quadcopter, frame optimization, topology optimization*

1. Introduction

Multicopters are a type of rotorcraft that uses three or more lift-generating rotors to achieve flight. Unlike traditional helicopters, multicopters typically use fixed-pitch propellers to generate the required thrust. Motion control and stability were managed by independently varying the rotational speed of each rotor, thereby adjusting the individual thrust and torque. The diverse applications of multicopters across various sectors can be attributed to their simple mechanical design, advanced on-board electronics and payload capabilities. In the field of unmanned aerial vehicles multicopters have been noted for their maneuverability and precise hovering abilities. They are widely used in areas such as aerial photography and videography, terrain mapping, infrastructure inspection and monitoring, precision agriculture, various delivery services, search and rescue operations, surveillance and security, hobby flying and racing [1]. This has resulted in increased research and development of multicopters with more advanced control systems [2, 3] and complex on-board equipment. The increase in relative equipment mass requires the exploration of new methods for reducing the frame structural weight while maintaining and even improving its efficiency.

Multicopters are typically categorized according to the number of rotors. The quadcopter, featuring four rotors, is by far the most popular type due to its inherent stability, operational simplicity and relatively low cost. Hexacopters, equipped with six rotors, offer enhanced stability, increased payload capacity as well as rotor redundancy, allowing for a safe landing in the event of a single motor failure. Octocopters with eight rotors provide maximum stability, lift capacity and redundancy, enabling flight with up to two motor failures. However, the increased number of propellers results in higher power consumption, which consequently increases the gross weight. This interrelation among the parameters indicates that the multicopter design process involves trade-offs with conflicting constraints (Table 1). In result, the four-rotor configuration has been widely adopted in multicopter development and production. Quadcopters achieve an optimal balance among simplicity, overall cost, performance, and ease of use.

Quadcopters typically have their rotors arranged in a "+", "H" or "X" configuration (Fig. 1). The ‘+’ configuration features one arm pointing directly forward, one directly backward, one left, and one right. This design offers maneuverability and simplifies flight control, as each motor handles movement in a specific direction. However, the "+" frame may not be as durable during crashes because only one arm absorbs the impact. In the "H" configuration, the front and rear motors are connected by longer, parallel arms to a central body or plate, making it suitable for photography or video recording due to the absence of propellers in the field of view. The "H" frame offers good stability, especially in roll, because of the wider arm separation, but this configuration is typically heavier and may lack torsional stiffness if not well designed. The "X" frame is the most symmetrical configuration, with the arms extending outward from a central point at 90° to each other. This configuration is compact, capable of absorbing impacts

with two arms, and provides balanced flight characteristics for both pitch and roll. Components are often stacked vertically in the center with the battery commonly mounted underneath. The "X" frame is the most popular configuration used in quadcopter design.

Table 1. Basic quadcopter design trade-offs

Trade-off	Parameter A	Parameter B	Description
Energy & endurance	Battery capacity (mAh)	Aircraft weight (g)	↑ Capacity → ↑ energy stored → ↑ flight time, but each additional gram requires more lift power, reducing net endurance.
Propulsion	Motor KV (RPM/Volt)	Torque	↑ KV motors spin faster (*good for small props / speed*) but produce less torque; ↓ KV favor larger props and heavier payloads at lower top speeds
Propulsion	Propeller diameter & pitch	Throttle response & drag	Larger / higher-pitch props boost static thrust and lift efficiency but increase aerodynamic drag and moment of inertia, slowing transient response
Airframe	Frame weight	Structural durability & cost	Lightweight materials (*carbon-fiber, plastics*) cut weight but cost more and are more damage-prone; heavier aluminum/composites improve robustness at expense of flight time
Aerodynamics	Streamlined fairings / shrouds	Simplicity & maintainability	Streamlining or ducting reduces drag and improves hover efficiency, but adds parts, assembly time, and repair complexity
Payload & performance	Payload mass (g)	Flight time & agility	Every extra gram demands more thrust and power, for example, a 40 % payload increase can cut endurance by ≈25 %; it also adds inertia, reducing maneuverability

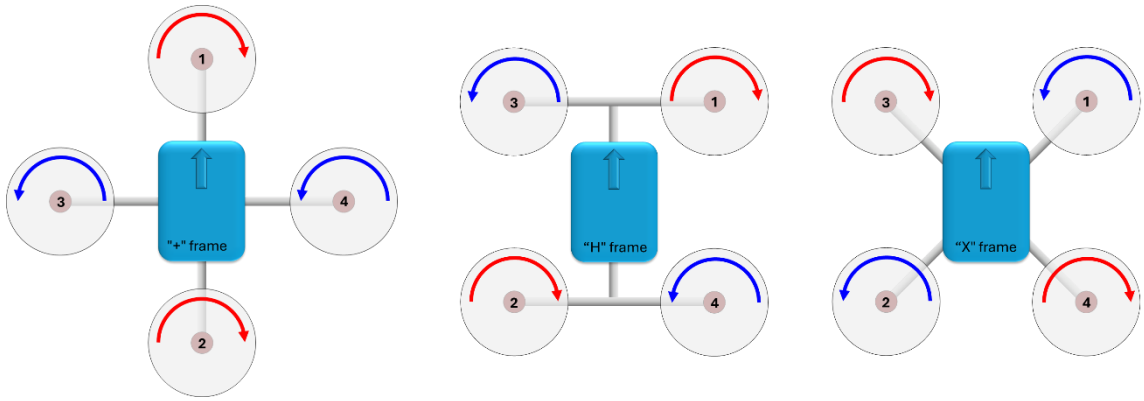


Fig. 1. Typical quadcopter frame configurations

Quadcopter frames are typically classified in two main categories: microframes (less than 150 mm) and miniframes (≥ 150 mm). During forward flight, quadcopter frames generate drag, and thus, larger frames require more power from the motors, leading to increased battery depletion rates. In [4] the drag produced by three different frame types was analyzed during forward flight at 10 m/s with a 30-degree tilt. The results from this study indicate that the X-frame generated the least amount of drag during flight.

The weight of the frame is critically important because it significantly affects nearly every aspect of the drone's performance and durability. Typically, the frame accounts for approximately 30% of a quadcopter's gross weight [5]. Structural optimization of the frame involves a delicate balance between competing factors such as weight, strength, stiffness and cost. Reducing the weight enhances flight time, increases payload capacity, improves performance due to a higher thrust-to-weight ratio and reduces inertia, which in turn improves the control precision and impact forces. The rigidity and stiffness of the

frame minimize deflections, thereby enhancing the flight stability and control. Additionally, repairability is an essential feature currently in demand and is defined as the capability to easily replace damaged components such as individual arms or motor mounts.

2. General approach for quadcopter frame structural optimization

Quadcopter structural optimization involves refining the frame to achieve an optimal balance of weight, strength, aerodynamics and vibration resistance by utilizing computational tools, such as finite element analysis, topology optimization and material selection. The primary methods for weight reduction include material selection and structural design optimization.

The materials most suitable for quadcopter frames are carbon fiber composites, lightweight alloys and various 3D printing and reinforced plastics. Carbon fiber is the most common and effective material for high-performance frames due to its extremely high stiffness and strength-to-weight ratio. Additively manufactured frames are typically printed from polylactic acid, acrylonitrile butadiene styrene, polyethylene terephthalate glycol, carbon fiber-reinforced plastics, and nylon. Polylactic acid (PLA) is the easiest to print and offers excellent rigidity and surface finish. However, it is quite brittle, with low humidity resistance and poor gluing ability, making it less suitable for load-bearing, long-term frames. Acrylonitrile Butadiene Styrene (ABS) is preferred over PLA when a higher toughness and heat resistance are required. It exhibits good abrasion resistance and can be post-processed (acetone smoothing); however, it emits fumes, is UV-sensitive, and tends to warp without an enclosed, heated build chamber. Polyethylene Terephthalate Glycol (PETG) combines good toughness, chemical and humidity resistance, and notable abrasion resistance. It is slightly heavier than PLA and ABS. PETG's lower warping and ease of printing makes it a popular mid-range choice for structural parts. Carbon fiber reinforced plastics (CFRP) are composite filaments in which short strands of chopped fiber are mixed into a base polymer (nylon, PETG, PLA). The fibers significantly increase the tensile strength and stiffness of the base material, making them the best choice for structural rigidity and impact resistance in fused deposition modeling for 3D printing. This material has an excellent strength-to-weight ratio. Carbon fibers stabilize the print and often help to reduce warping issues. Disadvantages include the increased wear rate of standard brass nozzles, higher cost of the filament, and printing difficulties. Nylons, such as PA6, PA12, and PA6/66, are highly durable and flexible plastics with excellent impact and temperature resistance. Structures printed from these materials are extremely tough and crash-resistant. However, nylon is hygroscopic, prone to warping during printing, and generally more challenging to print than PETG or PLA. The general mechanical properties of these materials are listed in Table 2.

Table 2. Mechanical properties of common 3D printing materials

Properties/material	***Nylon 6/6***	***ABS***	***PLA***
Yield strength (MPa)	82.75	44.1	49.5
Mass density (kg/m³)	1130	1080	1300
Ultimate tensile strength (MPa)	82.75	50	50
Poisson's ratio	0.35	0.422	0.39
Young's modulus (GPa)	2.93	2.9	3.5
Shear modulus (MPa)	1000	805	2400

Current research on optimal design methodologies has increasingly focused on topology optimization techniques and generative design algorithms. Topology optimization is a computational design process aimed at finding the best distribution of materials within a given domain to minimize an objective, such as compliance, under constraints. The resulting structures often have intricate and efficient designs, which are significantly lighter. Similarly, generative design employs AI and algorithms to explore numerous design solutions based on parameters such as load requirements, material properties, and manufacturing constraints, often producing optimized, lightweight geometries that are challenging to create manually. Topology optimization procedures are computationally intensive and rely on algorithms with multiple tuning parameters. Therefore, as with any finite-element

numerical model, it is advisable to conduct a preliminary sensitivity study of these parameters and to carry out proper validation and verification of the results [6, 7].

Two predominant approaches for topology optimization are the solid isotropic material with penalization (SIMP) method and level set method (LSM) (Fig. 2).

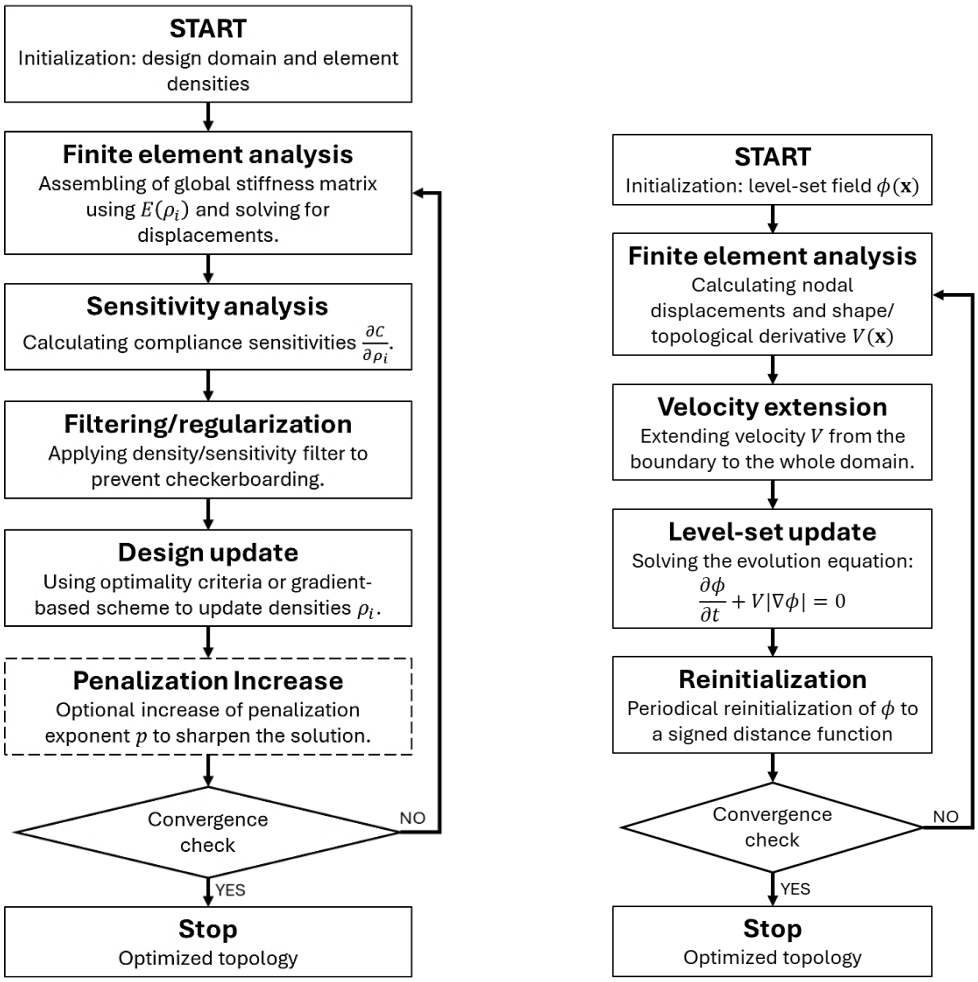


Fig. 2. Flowchart of most common topology optimization methods
a) Solid isotropic material with penalization (SIMP); b) Level set method (LSM)

The SIMP method formulates topology optimization as a material distribution problem, in which each finite element is assigned a pseudo-density variable $\rho \in [0,1]$ indicating void ($\rho = 0$) or solid ($\rho = 1$). The element stiffness is interpolated via a power-law relationship as follows:

$$E(\rho) = E_{min} + \rho^p (E_0 - E_{min})$$

where E_0 is the Young's modulus of the solid material, E_{min} is a small stiffness to avoid singularity, and $p > 1$ is the penalization exponent that drives the densities toward 0 or 1. Introduced by Bendsøe and Kikuchi [8, 9] and further developed by Rozvany and Zhou [10], SIMP is a widely implemented density-based topology optimization scheme.

The level-set method [11] defines structure with a scalar function $\phi(\mathbf{x})$. The interface between the solid and void is the zero level-set $\{\boldsymbol{x} \mid \phi(\boldsymbol{x}) = 0\}$. Regions where $\phi > 0$ represent material and $\phi < 0$ indicate voids. The evolution of ϕ follows the Hamilton–Jacobi equation, moving the interface based on a velocity field from the shape or topological sensitivities. This provides a clear boundary and naturally manages topological changes, such as merging or splitting.

Both solid isotropic material with penalization (SIMP) and the level set method (LSM) utilize gradient, or sensitivity, information. However, the practical differences in their final accuracy stem from their parameterizations, numerical treatment of boundaries/interfaces, and mesh dependency. SIMP employs element-wise density sensitivities: SIMP calculates $\partial C/\partial \rho_i$ (the sensitivity of compliance to density) and updates densities using optimality criteria or method of moving asymptotes (MMA), fully

within a gradient-based framework. LSM utilizes shape sensitivities and evolves the level-set function $\phi(\boldsymbol{x})$ via a Hamilton–Jacobi equation driven by the normal velocity $\mathcal{V} = -\partial J/\partial n$, which is derived from shape gradients [12].

Applying topology optimization and generative design methods allows lightweighting the frame by integrating and consolidating components, combining multiple functions into a single part, and reducing the need for fasteners and brackets. The goal is to optimize the entire quadcopter frame as a nondecomposable part. Typically, the design domain for such frames is modeled as a solid block or cylinder, with loads applied at the motor mounts and fixed bottom surface of the central body. In [5], a single-part quadcopter frame design utilizing topology optimization with the SIMP method for additive manufacturing using ABS via the fused filament fabrication (FFF) method was developed. Three options for the initial model were tested: square, cylindrical, and hexagonal. The objective is to achieve maximum compliance with minimal mass constraints. The design for additive manufacturing requires the application of a 45-degree overhang constraint for the final topology. In addition, cyclic and symmetrical constraints are applied. The results demonstrated a weight reduction of over 80% when employing a two-step procedure. In [13], the optimal design of a quadcopter frame in Fusion 360 was achieved using SIMP topology optimization by comparing the 3D printed structure with the DJI F450 frame. The results show that the optimal 3D printed ABS frame is twice as strong and stiff, with a lower mass than DJI F450. The generative design produced 74 feasible solutions from which eight variants were chosen. The final frame topology was determined by strength and vibrational analyses. Symmetric boundary conditions during modeling ensured an optimal center of gravity for better drone control. A DJI F450 frame was designed in [14] using topology optimization and generative procedures, with the objective of achieving a 10% mass reduction. The frame included four arms made of PA66-GF and two central body plates made of PCB. The solid square model was used as the initial domain. The optimal topology guides the final design and postprocessing of the frame. This article highlights that topological optimization is suitable for tasks with numerous specific requirements and constraints, whereas generative design is more appropriate when the task has few basic criteria, allowing the final solution to be selected from several generated options. The ultralight and robust quadcopter frame design in [15] focuses on improving the thrust-to-weight (T/W) and power-to-weight (P/W) ratios. For this purpose, SIMP topology optimization followed by generative design was used with constraints suitable for additive manufacturing. The final weight was reduced by 50%, and the ratios were increased: T/W by 6.75% and P/W by 6.08%. The safety factor increased by 11.8% along with a 70% reduction in the maximum equivalent stress. In [16], a procedure for the design and optimization of a hexacopter for precision agriculture applications was presented. Optimization was achieved through the selection of appropriate materials. The frame was constructed using various materials, including an aluminum base, carbon fiber arms, and polyamide motor mounts. The different frame variants were subjected to numerical testing using a finite element analysis. The composite material was modeled in advance using ANSYS ACP software, comprising 15 layers of unidirectional epoxy carbon, with an overall thickness of 1.5 mm. An alternative approach for quadcopter frame design and optimization was demonstrated in [17], in which a truss-optimization generative design framework was proposed. The optimization was performed on a predefined 3D truss resembling the drone arm, utilizing a genetic algorithm (Non-dominated Sorting Genetic Algorithm II). The optimal structure is 3D printed using multi jet fusion (MJF), a binder-jet technology for additive manufacturing that produces parts by fusing fine powder materials such as Nylon 12 with detailing agents. In [18], a multicopter frame underwent two-step SIMP topology optimization using solid cylindrical design domains. The optimization problem is modified with an additional constraint for inertia relief compensation, that is, force balance for the unconstrained object, by subtracting the body inertia from the applied force vector. The final geometry is identified through the structural components that can be produced by sheet metal stamping. The crash resistance of the frame was tested numerically using ANSYS LS DYNA, which is a specialized software for simulating complex real-world problems involving large deformations, nonlinear material behavior, and dynamic events. The weight reduction in the second step was 45%, and the mass distribution ensured crash resistance on impact from a 3-meter altitude with an airspeed of 4 m/s.

Modeling the entire frame as a single component necessitates the replacement of the entire structure following a crash or impact rather than substituting only one specific part. Given that the X-frame is the most widely adopted design for a general-purpose quadcopter, design optimization can focus solely on the frame's arms or its central body. In the two-step optimization process, the arms are optimized first,

followed by the central body. The loads from the optimized arm models can be readily transferred to the central body model using a sub-modeling approach for finite element analysis. This study is addressing only the arm-optimization problem.

3. Two-step procedure for quadcopter frame topology optimization

To study the application of topology optimization to a multicopter structural design, a quadcopter with a takeoff mass of 2 kg was considered. By applying the typical 2:1 thrust-to-weight ratio [19], the required lift is 4 kg, or approximately 40 N. This evaluates to approximately 10 N for each propeller at maximum thrust. The finite element analysis included at least two load steps to account for the torque from the BLDC motor in both possible directions. According to the T-Motor datasheets[1], the torque can be estimated to approximately 2-5% of the propeller thrust, which translates to approximately ±0.5 Nm. For this quadcopter weight, the frame diagonal is typically in the range of 500-650 mm so for the arm of the frame a length of 230 mm was selected. Statistically, the arm width can be determined using the expression $M_{OD} \approx 0.1F_{size} + 5$ obtained by linear regression (*Table 3*). This resulted in an arm width of 40 mm.

Table 3. Motor outer diameter (OD) based on frame size

Frame diagonal (mm)	Typical motor series (stator size)	Motor OD range (mm)	Example drones / motors
200–300	2204–2306 (*22–23 mm stator*)	~28–35	Small 5″ FPV quads (*5″ props*)
300–400	2308–2508 (*23–25 mm stator*)	~32–42	DJI Phantom 3/4 (*2312 motor, 23 mm*); 6–7″ long-range quads
400–500	3510 (*35 mm stator*) and up	~45–55	DJI Inspire 1 (*3510 motor, 35×10 mm*)
500–600	6010 (*60 mm stator*) and up	~60–70+	DJI E2000 (*6010 motor, 60×10 mm*); T-Motor U7 (*⌀60.7 mm*)

From the quadcopter model, an arm submodel is isolated with the boundary conditions applied (Fig. 3). The arm consisted of a load-bearing beam with a motor mount surface and two attachment bushings at the root. Both the arm and motor mounts are from ABS, and the bushings are from aluminum alloy. The initial mass of the arm is $m_0 = 0{,}3725\ kg$. The bushings are fixed and the motor mount is loaded with a vertical thrust force and two-way torque from the motor. The torque is applied by two separate load cases. The geometry is discretized with 120000 4-node tetrahedral finite elements. The finite element analysis results reveal the displacement and stress fields in the model (Fig. 4).

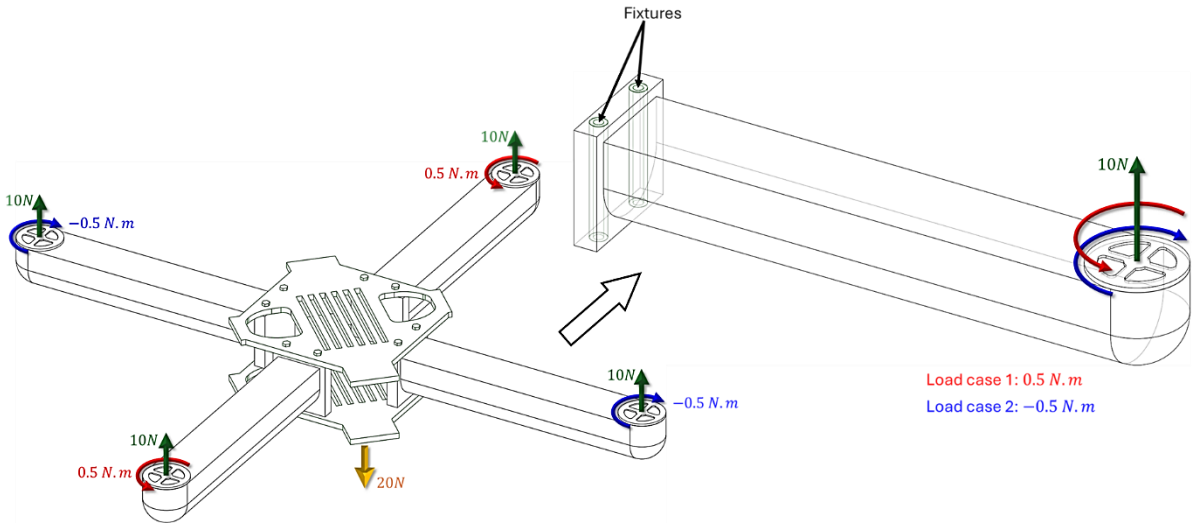


Fig. 3. Quadcopter model boundary conditions: quadcopter frame and arm submodel

[1] T-Motor Multirotor Motors, https://uav-en.tmotor.com/Multirotor/Motors/ (accessed May 15, 2025)

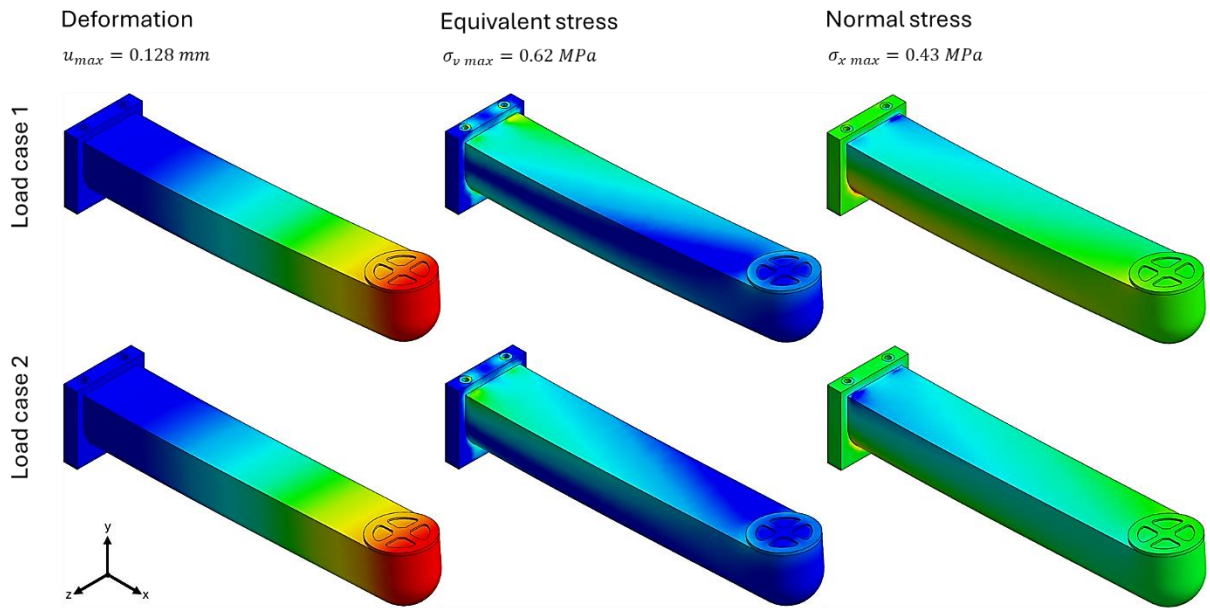


Fig. 4. Results from the finite element analysis of the arm submodel for the two separate load cases

Topology optimization using the SIMP method excludes the fixtures and the motor mount from the design region in order to preserve their geometry. The resultant topology is post-processed, cleaned, and numerically verified through additional finite element analysis. First, topology optimization with reduced penalization is performed to drive the solution into simpler topology with less holes (Fig. 5). In result, 60% reduction in the arm's weight is achieved which leads to lightening the frame with 46% (Fig. 6). Obtained results demonstrated a very large safety factor $f \approx 25$ for ABS material. This allows for further optimization of the structure with increased penalization to force the generation of more intricate and complex topology.

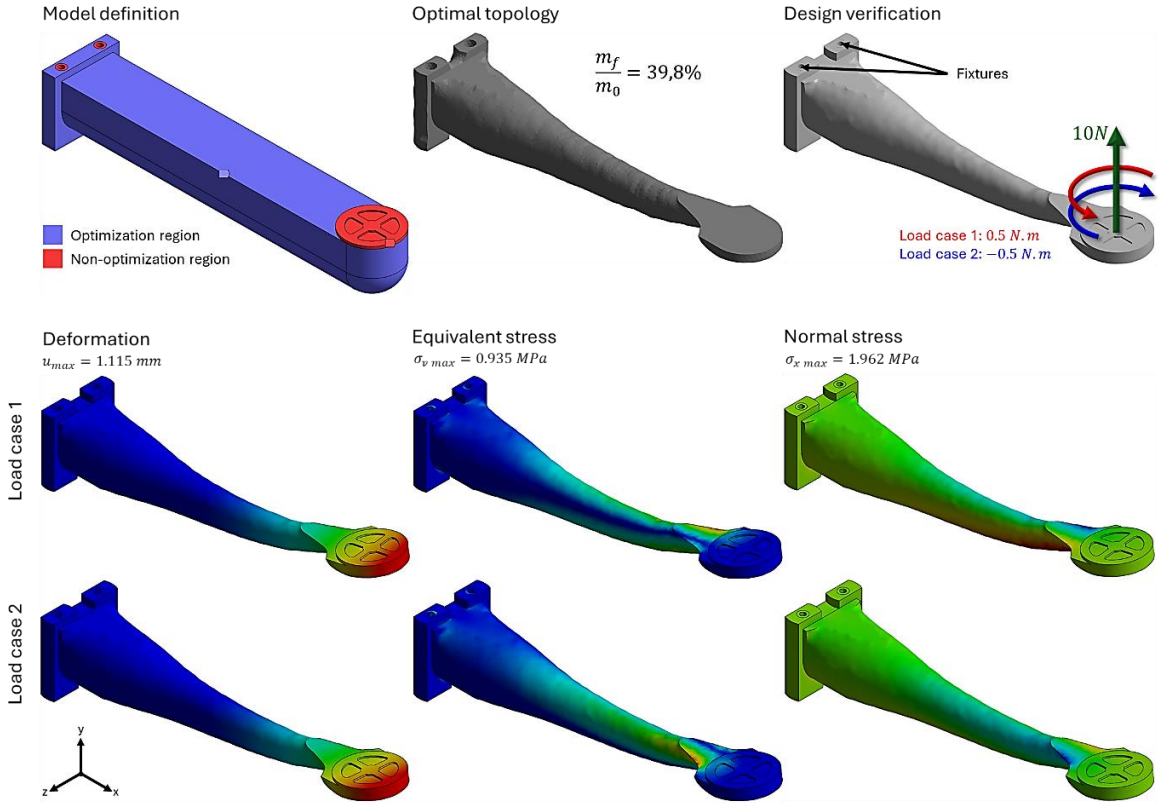


Fig. 5. Quadcopter arm topology optimization with numerical verification of the optimal topology

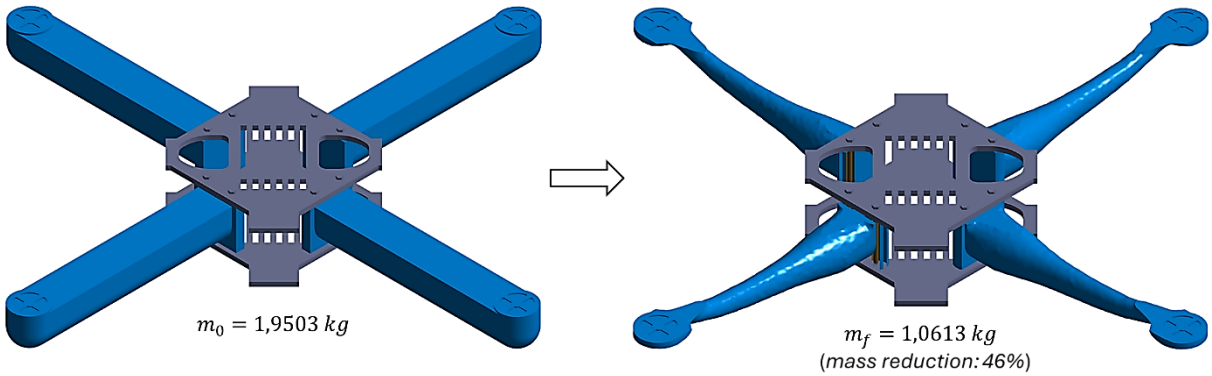


Fig. 6. Quadcopter frame equipped with the optimized arms

The result from the topology optimization with high penalization is post-processed and loaded again with the same boundary conditions to obtain the displacement field in its structure. The result from the finite element procedure is passed to consecutive topology optimization with low penalization to obtain the final optimized structure with maximum stiffness and minimum weight (Fig. 7).

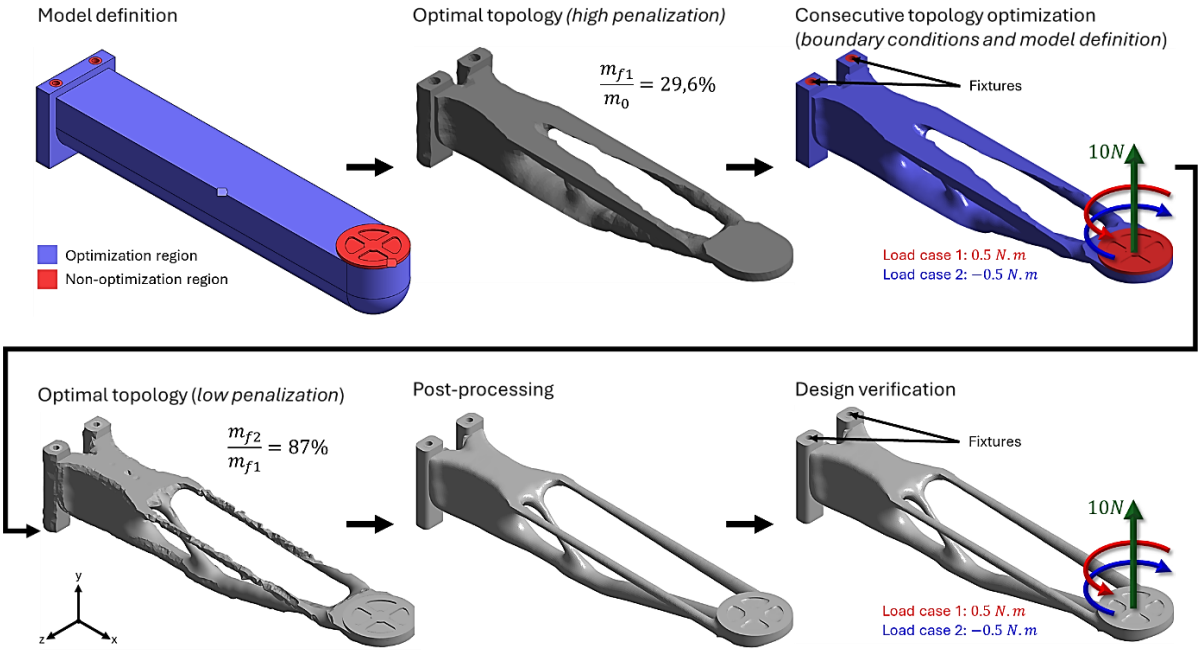


Fig. 7. Two-step topology optimization of the quadcopter arm

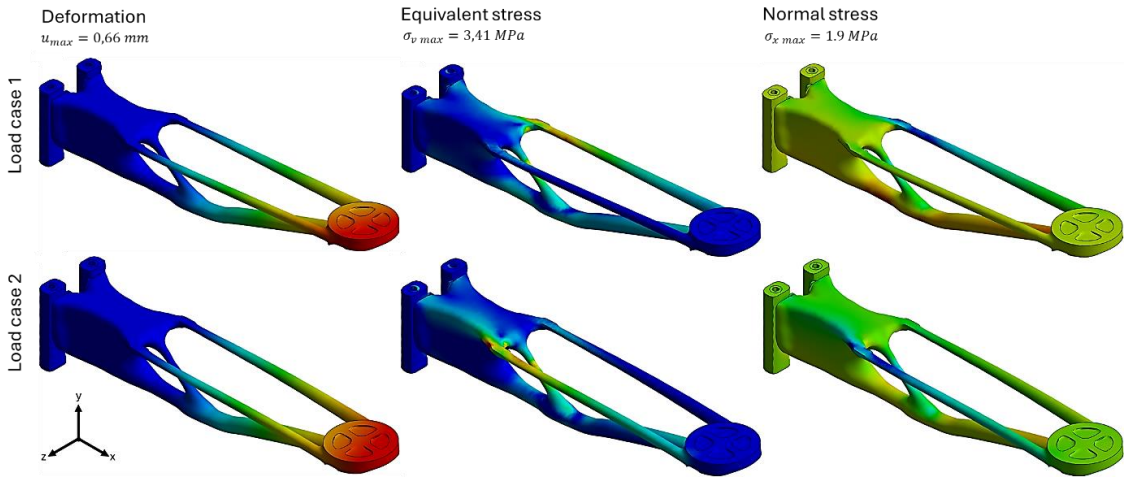


Fig. 8. Numerical verification of the optimal arm topology

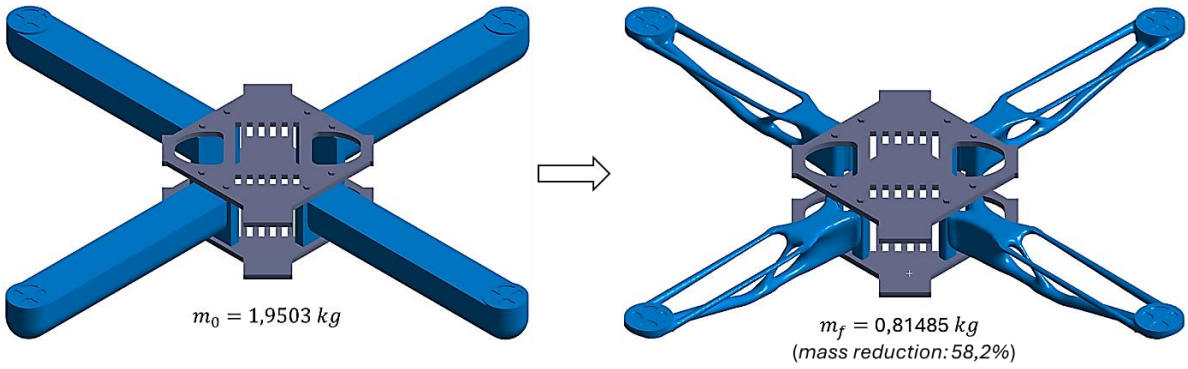


Fig. 9. Quadcopter frame with arms optimized for maximum stiffness and minimal mass

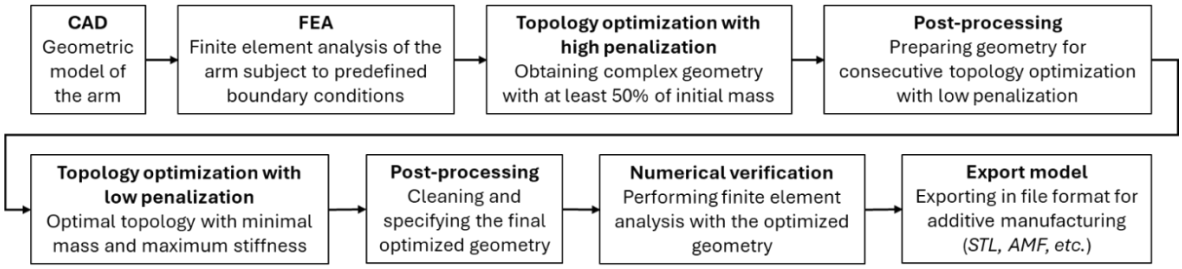


Fig. 10. Flowchart of two-step procedure for topology optimization of a quadcopter arm

4. Conclusion

The proposed two-step optimization process combines SIMP and level-set methods to significantly reduce quadcopter frame weight without losing structural integrity. The SIMP phase quickly identifies a load-bearing topology, cutting arm mass by 60% and frame weight by nearly 50%, with a high safety margin. The level-set refinement further reduces the arm mass to 75% and the overall frame weight to 58.2%, maintaining uniform stress distribution, low tip deflection, and an acceptable safety factor. The final design is then post-processed for additive manufacturing.

References

1. Mitov, L. “Uses of Drones for Peaceful Purposes in the Modern World”. Proceedings of the Bulgarian Air Force Academy's Annual International Scientific Conference. 2024, pp. 19–27. ISSN 2738-716X.

2. Biliderov, S.; Kamenov, K.; Calovska, R.; Georgiev, G., “Synthesis and testing of an algorithm for autonomous landing of a UAV under turbulence, wind disturbance and sensor noise,” Engineering Proceedings, Vol. 70, 2024, Art. 41, pp. 1–10. https://doi.org/10.3390/engproc2024070041

3. Kambushev, K., “Research of a PID controller built with neural networks, and possibilities of application in UAV control systems,” in Proceedings of the XXXIII International Scientific and Technical Conference ADP 2024, 2024, ISSN 2682-9584.

4. Al-Haddad, L. A.; Jaber, A. A.; Giernacki, W.; Khan, Z. H.; Ali, K. M.; Tawafik, M. A.; Humaidi, A. J., “Quadcopter unmanned aerial vehicle structural design using an integrated approach of topology optimization and additive manufacturing,” Designs, Vol. 8, No. 3, 2024, Art. 58, pp. 1–14. https://doi.org/10.3390/designs8030058

5. Nvss, S.; Esakki, B.; Yang, L.-J.; Udayagiri, C.; Vepa, K. S., “Design and development of unibody quadcopter structure using optimization and additive manufacturing techniques,” Designs, Vol. 6, No. 1, 2022, Art. 8, pp. 1–12. https://doi.org/10.3390/designs6010008

6. The American Society of Mechanical Engineers, An illustration of the concepts of verification and validation in computational solid mechanics, ASME V&V 10.1-2012, The American Society of Mechanical Engineers, New York, 2012. ISBN 9780791834152.

7. Georgiev, Y., “Validation and verification in scientific computer simulation,” Aeronautical Research and Development, Georgi Benkovski Bulgarian Air Force Academy, Dolna Mitropolia, 2024, ISSN 2815-2948.

8. Bendsøe, M. P., “Optimal shape design as a material distribution problem,” Structural Optimization, Vol. 1, 1989, pp. 193–202.

9. Bendsøe, M. P.; Kikuchi, N., “Generating optimal topologies in structural design using a homogenization method,” Computer Methods in Applied Mechanics and Engineering, Vol. 71, No. 2, 1988, pp. 197–224. https://doi.org/10.1016/0045-7825(88)90086-2

10. Rozvany, G. I. N.; Zhou, M.; Birker, T., “Generalized shape optimization without homogenization,” Structural Optimization, Vol. 4, Nos. 3–4, 1992, pp. 250–252. https://doi.org/ 10.1007/BF01742754

11. Wang, M. Y.; Wang, X.; Guo, D., “A level set method for structural topology optimization,” Computer Methods in Applied Mechanics and Engineering, Vol. 192, Nos. 1–2, 2002, pp. 227–246. https://doi.org/10.1016/S0045-7825(02)00559-5

12. Van Dijk, N. P.; Maute, K.; Langelaar, M.; Van Keulen, F., “Level-set methods for structural topology optimization: A review,” Structural and Multidisciplinary Optimization, Vol. 48, No. 3, 2013, pp. 437–472. https://doi.org/10.1007/s00158-013-0912-y

13. Bright, J.; Suryaprakash, R.; Akash, S.; Giridharan, A., “Optimization of quadcopter frame using generative design and comparison with DJI F450 drone frame,” IOP Conference Series: Materials Science and Engineering, Vol. 1012, No. 1, 2021, Art. 012019, pp. 1–10. https://doi.org/10.1088/ 1757-899X/1012/1/012019

14. Kowalik, M.; Śliwiński, M.; Papis, M., “Utilization of topology optimization and generative design for drone frame optimization,” in Research and Education in Aircraft Design Conference, Warsaw, Poland, 2024. Online]. Available: https://psaa.meil.pw.edu.pl/READ2024/Papers/ paperID75.pdf

15. Balayan, A.; Mallick, R.; Dwivedi, S.; Saxena, S.; Haorongbam, B.; Sharma, A., “Optimal design of quadcopter chassis using generative design and lightweight materials to advance precision agriculture,” Machines, Vol. 12, No. 3, 2024, Art. 187, pp. 1–12. https://doi.org/10.3390 /machines12030187

16. Gutierrez-Rivera, M. E.; Rumbo-Morales, J. Y.; Ortiz-Torres, G.; Gascon-Avalos, J. J.; Sorcia-Vázquez, F. D. J.; Torres-Cantero, C. A.; Buenabad-Arias, H. M.; Guillen-Escamilla, I.; López-Osorio, M. A.; Zurita-Gil, M. A.; Calixto-Rodriguez, M.; Rosales, A. M.; Juárez, M. A., “Design, construction and finite element analysis of a hexacopter for precision agriculture applications,” Modelling, Vol. 5, No. 3, 2024, pp. 1239–1267. https://doi.org/10.3390/modelling5030064

17. Olsen, S. T., “Optimizing a quadcopter frame prototype with a novel generative design framework,” M.S. thesis, University of Oslo, Institute for Informatics, Faculty of Mathematics and Natural Sciences, 2021.

18. Guo, H.; Li, M.; Sun, P.; Zhao, C.; Zuo, W.; Li, X., “Lightweight and maintainable rotary-wing UAV frame from configurable design to detailed design,” Advances in Mechanical Engineering, Vol. 13, No. 7, 2021. https://doi.org/10.1177/16878140211034999

19. Shivani, S.; Kumar, K., “Optimizing quadcopter structural design: A numerical analysis on strength and stability,” in Deepak, B. B. V. L.; Bahubalendruni, M. R.; Parhi, D.; Biswal, B. B., Eds., Recent Advancements in Product Design and Manufacturing Systems: IPDIMS 2023, Lecture Notes in Mechanical Engineering, Springer, Singapore, 2025. https://doi.org/10.1007/978-981-97-6732-8_1.

Двуетапна процедура за структурна оптимизация на рамката на квадрокоптер

Николай Кънчев

Висше военновъздушно училище „Георги Бенковски“, факултет „Авиационен“,
гр. Долна Митрополия, България, nikolaikanchev@yahoo.com

Резюме: В статията е представена двуетапна процедура за оптимално проектиране на леки рамки за квадрокоптер с използване на методите за топологична оптимизация SIMP и LSM. Първоначално, модел на отделно рамо от рамката се оптимизира топологично по метода SIMP под действие на комбинирано натоварване от теглителната сила на витлото и двупосочен въртящ момент. В резултат се генерира проста топология с 60% по-ниска маса за отделното рамо или 46% за цялата рамка при запазване на значителен по величина коефициент на сигурност. Последващата топологична оптимизация е реализирана по метод LSM за получаване на ясно изразени свързани структури с повишена коравина и допълнително понижение на масата със 75% (58.2% за цялата рамка) при достатъчно висок коефициент на сигурност и минимална деформация. Оптималната геометрия на рамото е обработена допълнително за удовлетворяване на производствените ограничения и е верифицирана числено посредством допълнителен крайно-елементен анализ. Предложената двуетапна процедура позволява проектиране на здрави и леки рамки за квадрокоптери, подходящи за адитивно производство, чрез които се постига повишаване както на продължителността на полета, така и на масата на полезния товар.

Implementing a Proportional-Derivative Controller for Quadcopter Attitude

Konstantin Metodiev

Department of Aerospace Control Systems, Space Research and Technology Institute, Bulgarian Academy of Sciences, Sofia, Bulgaria, komet@space.bas.bg

Abstract: This article describes a mathematical model of a quadcopter spatial motion. A proportional-derivative controller is used in the model for maintaining the quadcopter attitude. The resulting system of first-order ordinary differential equations (with respect to linear and angular velocities) with constant coefficients is solved by explicit Dormand-Prince method of order 4 with adaptive time step. The method is implemented in GNU Octave environment using `ode45` function designed for a system of non-stiff first-order ordinary differential equations. The mathematical model is validated with exact solutions. Obtained results for the controller performance are presented in graphical form and discussed.

Keywords: *Quadcopter, Controller, Attitude stabilization, GNU Octave*

1. Introduction

Quadcopter flight dynamics, including both translational and rotational motion, is described by Newton-Euler equations. These equations make clear how linear and angular momentum are being changed in terms of externally applied forces and moments like thrust, weight, torques, gyro effects. Quadcopter flight dynamics is strongly nonlinear and complex due to coupling of translational and rotational motions, aerodynamic and gyroscopic effects, and so on. This makes precise control and stability of the quadcopter a demanding task.

In practice, quadcopters use various control algorithms for stability and maneuverability. For instance, Proportional-Integral-Derivative controllers are commonly used for maintaining stability by adjusting motor speeds based on feedback from sensors. Combining sensor data (like gyroscopes, accelerometers, GPS) to estimate the quadcopter's state (position, velocity, orientation) for precise control is a prerequisite to Linear-Quadratic Regulators.

Few implementations of proportional-derivative (PD) controller to maintain quadcopter attitude do exist. Some of them follow a comprehensive algorithm published by Luukkonen in [1]. Unfortunately, most of implementations (widely available in internet) have been created in SimuLink environment resulting in bulky block diagrams that are difficult to read and decipher and somewhat inaccurate to say the least. Gibiansky in [2] provides another solution manual on how to implement a PD controller together with a source code in MatLab. Alderete in [3] and Tytler in [4] have published tutorials on quadcopter flight dynamics, with the latter also publishing a source code in Python. Benic at al. in [5] also provide with a thorough mathematical model of quadcopter spatial motion. Plenty of papers on the topic do exist with certain level of accuracy.

The current paper goal is to implement a proportional – derivative controller for maintaining the quadcopter attitude in a simple and self-explanatory GNU Octave script. No trajectory control is being considered whatsoever. Most of formulae and equations are not derived but solely written down and credits are given to references for further explanation. Links to both developed source code and CAD model are also given in References section.

2. Materials and methods

The body frame of reference is a coordinate system fixed to the quadcopter. As opposed to the inertial frame of reference, the body reference frame helps to understand how the body moves relative to itself. In current study case, the reference frame origin is typically placed at the mass center of the quadcopter. The axes are aligned with the quadcopter's structure depending upon flight configuration: either "×" or "+," fig. 1. In current study, flight configuration "+" is solely being considered.

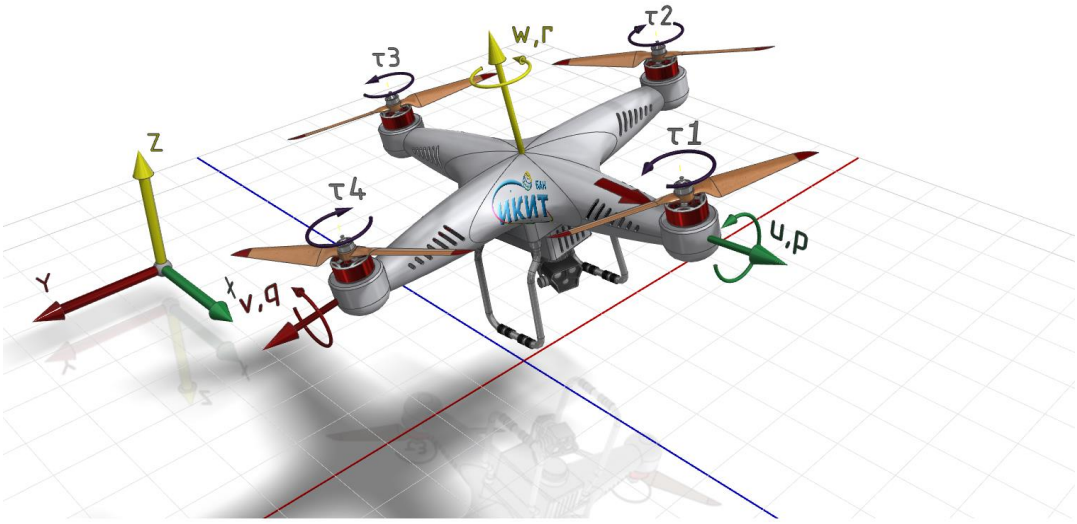


Figure 1. Inertial and body reference frames, flight configuration "+"

Quadcopter motion might be described in terms of translational and rotational movements both in inertial and body frames of reference. State vector in body reference frame, according to symbolic adopted in fig. 1, is following

$$\left\|\mathbf{v} \quad \boldsymbol{\omega}\right\|^T = \left\|u \quad v \quad w \quad p \quad q \quad r\right\|^T \tag{1}$$

where $\mathbf{v} = [u, v, w]^T$ are linear velocities, $\boldsymbol{\omega} = [p, q, r]^T$ are angular rates. System first order ordinary differential equations governing the quadcopter motion in body reference frame reveals linear and angular momentum (of a mechanical system) conservation

$$\begin{Vmatrix} m\mathbf{I}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I} \end{Vmatrix} \begin{Vmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{Vmatrix} + \begin{Vmatrix} \boldsymbol{\omega} \times (m\mathbf{v}) \\ \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \end{Vmatrix} = \begin{Vmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{Vmatrix} \tag{2}$$

where $\mathbf{I}_{3,3}$ is a 3 by 3 identity matrix, $\mathbf{0}_{3,3}$ is a 3 by 3 zero matrix, $m = const$ stands for the quadcopter mass, $\mathbf{f}$ and $\boldsymbol{\tau}$ denote externally applied forces and torques respectively. It is admissible to simplify the inertia tensor $\mathbf{I}$ further due to symmetry

$$\mathbf{I} = \begin{Vmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{Vmatrix} \tag{3}$$

The inverse of a diagonal matrix might be found by replacing the main diagonal elements with their reciprocals. As regards vector $[[\boldsymbol{\omega}\times(m\mathbf{v})]; [\boldsymbol{\omega}\times(\mathbf{I}\boldsymbol{\omega})]]^T$, it might be rather computed by multiplying Coriolis centripetal matrix and the state vector as follows:

$$\begin{Vmatrix} \boldsymbol{\omega} \times (m\mathbf{v}) \\ \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega}) \end{Vmatrix} = \begin{Vmatrix} [m\boldsymbol{\omega}]\times_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & [-\mathbf{I}\boldsymbol{\omega}]\times_{3,3} \end{Vmatrix} \begin{Vmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{Vmatrix} \tag{4}$$

where symbol $[]\times_{3,3}$ denotes a skew symmetric matrix, for example

$$[m\boldsymbol{\omega}]\times_{3,3} = m \cdot \begin{Vmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{Vmatrix}; \quad [-\mathbf{I}\boldsymbol{\omega}]\times_{3,3} = -1 \cdot \begin{Vmatrix} 0 & -I_{zz}r & I_{yy}q \\ I_{zz}r & 0 & -I_{xx}p \\ -I_{yy}q & I_{xx}p & 0 \end{Vmatrix} \tag{5}$$

Finally, the system first order ordinary differential equations with constant coefficients describing the quadcopter motion in body frame of reference takes the ultimate form

(6) $$\begin{Vmatrix}\dot{\mathbf{v}}\\ \dot{\boldsymbol{\omega}}\end{Vmatrix}=\begin{Vmatrix}m\mathbf{I}_{3,3} & \mathbf{0}_{3,3}\\ \mathbf{0}_{3,3} & \mathbf{I}\end{Vmatrix}^{-1}\left(\begin{Vmatrix}\mathbf{f}\\ \boldsymbol{\tau}\end{Vmatrix}-\begin{Vmatrix}[m\boldsymbol{\omega}]\times_{3,3} & \mathbf{0}_{3,3}\\ \mathbf{0}_{3,3} & [-\mathbf{I}\boldsymbol{\omega}]\times_{3,3}\end{Vmatrix}\begin{Vmatrix}\mathbf{v}\\ \boldsymbol{\omega}\end{Vmatrix}\right)$$

The vectorized representation above is appropriate for GNU Octave to do calculations fast.
Benić at al., [5] recommend using a hybrid coordinate system to work out a solution to the governing equations. Particularly, the linear momentum conservation is described in inertial frame of reference whilst the angular momentum conservation is defined in body frame of reference. In this case, the centripetal force $\boldsymbol{\omega}\times(m\mathbf{v})$ is zero, hence the upper left corner (3-by-3) of Coriolis matrix (4) is zero too. External forces and torques $[\mathbf{f}, \boldsymbol{\tau}]$ acting on the quadcopter frame are to be transformed accordingly. The following rotation matrix (from body to inertial frame of reference) might be used for this purpose (rotation sequence ZYX only)

(7) $$\mathbf{R}=\begin{Vmatrix}C_\psi C_\theta & C_\psi S_\theta S_\varphi - S_\psi C_\varphi & C_\psi S_\theta C_\varphi + S_\psi S_\varphi\\ S_\psi C_\theta & S_\psi S_\theta S_\varphi + C_\psi C_\varphi & S_\psi S_\theta C_\varphi - C_\psi S_\varphi\\ -S_\theta & C_\theta S_\varphi & C_\theta C_\varphi\end{Vmatrix}$$

where *Sx = sin(x)* and *Cx = cos(x)*. The transformation matrix $\mathbf{R}$ is orthogonal, hence $\mathbf{R}^{-1}=\mathbf{R}^{\mathrm{T}}$.
Alternatively, the rotation matrix might be derived using quaternions. Given normalized axis *a* and angle of rotation θ, the equivalent components of quaternion are:

(8) $$q_w=\cos(\theta/2)\quad q_x=a_x\sin(\theta/2)\quad q_y=a_y\sin(\theta/2)\quad q_z=a_z\sin(\theta/2)$$

The formula above is applied to Cartesian components of rotation axis $[a_x,0,0]$, $[0,a_y,0]$, $[0,0,a_z]$ and corresponding Euler angles $[\varphi,0,0]$, $[0,\theta,0]$, $[0,0,\psi]$ following initially assigned sequence, for instance XYZ. The obtained three quaternions *q1, q2, q3* are then multiplied.
Quaternions are frequently split into a scalar term *s* and a vector term $\mathbf{v}$, [6]:

(9) $$q=\begin{bmatrix}s & \mathbf{v}\end{bmatrix}\quad s\in\mathbb{R}\quad \mathbf{v}=a\mathbf{i}+b\mathbf{j}+c\mathbf{k}$$

Using the notation above, the quaternion product is computed according to formula

(10) $$q_1q_2=(s_1s_2-\mathbf{v}_1\mathbf{v}_2)+(s_1\mathbf{v}_2+s_2\mathbf{v}_1+\mathbf{v}_1\times\mathbf{v}_2)$$

Then, according to Kuipers, [7], the transformation matrix (from body to inertial frame of reference) for arbitrary rotation sequence is derived from the product *[qw qx qy qz] = (q1q2)q3* as follows

(11) $$\mathbf{R}=\begin{Vmatrix}2(q_w^2+q_x^2)-1 & 2(q_xq_y-q_wq_z) & 2(q_xq_z+q_wq_y)\\ 2(q_xq_y+q_wq_z) & 2(q_w^2+q_y^2)-1 & 2(q_yq_z-q_wq_x)\\ 2(q_xq_z-q_wq_y) & 2(q_yq_z+q_wq_x) & 2(q_w^2+q_z^2)-1\end{Vmatrix}$$

Externally applied force due to motor spinning written in body frame of reference is, [1]

(12) $$\mathbf{T}=\begin{Vmatrix}0\\ 0\\ k\sum_{i=1}^{4}\omega_i^2\end{Vmatrix}$$

Externally applied force due to quadcopter weight written in body frame of reference is, [1]

(13) $$\mathbf{g}=\mathbf{R}^{-1}\begin{Vmatrix}0\\ 0\\ mg\end{Vmatrix}$$

where $g = 9.81\ m/s^2$ is the Earth gravitational acceleration. The total external force vector applied to the quadcopter in body frame of reference is following sum

(14) $$\mathbf{f}=\mathbf{T}+\mathbf{g}$$

Externally applied torque due to motor spinning written in body frame of reference is, [1]

(15) $$\mathbf{E} = \begin{Vmatrix} lk\left(-\omega_2^2 + \omega_4^2\right) \\ lk\left(-\omega_1^2 + \omega_3^2\right) \\ b\sum_{i=1}^{4}\left(-1\right)^i \omega_i^2 \end{Vmatrix}$$

where *k* is lift constant, *b* is drag constant, *l* is distance between quadcopter mass center and motor axis. Externally applied torque due to gyroscopic effect written in body frame of reference is, [5]

(16) $$\mathbf{o} = -\left(\boldsymbol{\omega} \times \begin{Vmatrix} 0 \\ 0 \\ 1 \end{Vmatrix} \right) I_r \sum_{i=1}^{4}\left(-1\right)^i \omega_i$$

where *Ir* is the propeller moment of inertia. The total external torque vector applied to the quadcopter in body frame of reference is following sum

(17) $$\boldsymbol{\tau} = \mathbf{E} + \mathbf{o}$$

Euler angles time rate of change (inertial frame of reference) are related to body frame angular rates according to following formulae, [3]:

(18) $$\begin{Vmatrix} p \\ q \\ r \end{Vmatrix} = \begin{Vmatrix} 1 & 0 & -S_\varphi \\ 0 & C_\varphi & C_\theta S_\varphi \\ 0 & -S_\varphi & C_\theta C_\varphi \end{Vmatrix} \begin{Vmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{Vmatrix}$$

(19) $$\begin{Vmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{Vmatrix} = \begin{Vmatrix} 1 & S_\varphi T_\theta & C_\varphi T_\theta \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi / C_\theta & C_\varphi / C_\theta \end{Vmatrix} \begin{Vmatrix} p \\ q \\ r \end{Vmatrix}$$

where *Sx = sin(x), Cx = cos(x),* and *Tx = tan(x)*. Note singularity if $\theta = \pm\pi/2$ rad.

Proportional – derivative controller is derived according to Luukkonen, [1]. The acceleration in direction of *z* axis is affected by thrust *T* whilst Euler angles are affected by torques $\tau\varphi$, $\tau\theta$, $\tau\psi$:

$$T = \left[g + K_{z,d}\left(\dot{z}_d - \dot{z}\right) + K_{z,p}\left(z_d - z\right) \right] \frac{m}{C_\varphi C_\theta}$$

(20)
$$\tau_\varphi = \left[K_{\varphi,d}\left(\dot{\varphi}_d - \dot{\varphi}\right) + K_{\varphi,p}\left(\varphi_d - \varphi\right) \right] I_{xx}$$
$$\tau_\theta = \left[K_{\theta,d}\left(\dot{\theta}_d - \dot{\theta}\right) + K_{\theta,p}\left(\theta_d - \theta\right) \right] I_{yy}$$
$$\tau_\psi = \left[K_{\psi,d}\left(\dot{\psi}_d - \dot{\psi}\right) + K_{\psi,p}\left(\psi_d - \psi\right) \right] I_{zz}$$

where subscript *d* denotes "desired" state, *Cx = cos(x)*. Motor angular velocities are computed from eq. (12) and eq. (15)

(21)
$$\omega_1^2 = \frac{T}{4k} - \frac{\tau_\theta}{2kl} - \frac{\tau_\psi}{4b} \quad \omega_2^2 = \frac{T}{4k} - \frac{\tau_\varphi}{2kl} + \frac{\tau_\psi}{4b}$$
$$\omega_3^2 = \frac{T}{4k} + \frac{\tau_\theta}{2kl} - \frac{\tau_\psi}{4b} \quad \omega_4^2 = \frac{T}{4k} + \frac{\tau_\varphi}{2kl} + \frac{\tau_\psi}{4b}$$

3. Validation

The developed source code was put to the test with an exact solution to following problem in rigid body dynamics. A projectile is being fired at angle α and initial velocity **V0**, fig. 2. The force of gravity **G** = $[0, -mg]^T$ is defined in inertial reference frame whilst the force of air drag **R** = $[-kmv, 0]^T$ is defined in body reference frame. Find a law describing the projectile motion! Adopted notations and units of measurement are: **V0** = 100*$[\cos(\alpha), \sin(\alpha)]^T$ m/s is initial velocity; g = 9.81 m/s^2 is Earth gravitational acceleration; k = 1 s^{-1} is proportional coefficient.

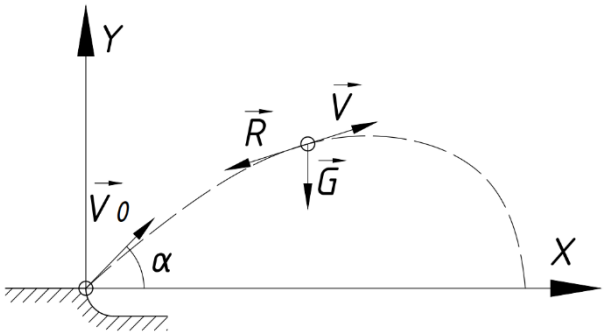


Figure 2. Validation case according to problem statement

Equations of projectile motion according to fig. 2 are:

$$m\ddot{x} = -km\dot{x}$$
$$m\ddot{y} = -km\dot{y} - mg \tag{22}$$

These equations are independent of each other, they can be solved separately, and they have identical characteristic equations with real distinct roots, i.e.

$$\rho^2 + k\rho = 0 \quad \rho_1 = 0 \quad \rho_2 = -k \tag{23}$$

Consequently, the general solution is:

$$x = C_1 e^{0t} + C_2 e^{-kt} \quad y = C_3 e^{0t} + C_4 e^{-kt} + u \tag{24}$$

where $u = At$ is a particular solution. After replacing the particular integral in the second equation of system (22), we get $A = -g/k$, i.e., $u = -gt/k$. Finally, the complete solution is

$$x = C_1 + C_2 e^{-kt} \quad y = C_3 + C_4 e^{-kt} - gt/k \quad \dot{x} = -C_3 k e^{-kt} \quad \dot{y} = -C_4 k e^{-kt} - g/k \tag{25}$$

In order to find constants $C_{1..4}$, it is sufficient to plug following initial conditions into system (22):

$$t = 0 \quad x = 0 \quad y = 0 \quad \dot{x} = v_0 \cos\alpha \quad \dot{y} = v_0 \sin\alpha \tag{26}$$

Ultimately, parametric equations of motion are:

$$x = \frac{v_0 \cos\alpha}{k}\left(1 - e^{-kt}\right) \quad y = \frac{1}{k}\left(v_0 \sin\alpha + \frac{g}{k}\right)\left(1 - e^{-kt}\right) - \frac{g}{k}t \tag{27}$$

Identical results might be obtained using Symbolic package available in GNU Octave.
Initial conditions (26) were plugged into model being developed, system equations (6) and test problem (22) to perform a simulation of projectile motion and compare obtained solution with exact one. The quadcopter rotors had been turned off. The two results match exactly, fig. 1.

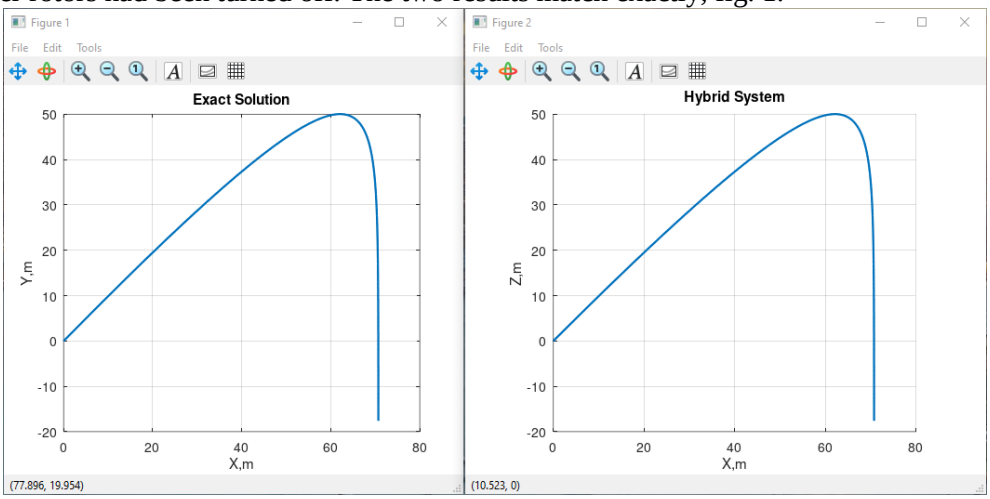


Figure 3. Validation case: exact (left) and numerical solution

In addition, the rotation matrix (11) has been put to the test by MatLab function `quat2rotm(q)` for validation purposes. Again, the obtained results match.

4. Results

Numerical test has been carried out with following initial conditions: mass m = 0.468 kg; moment of inertia Ixx = Iyy = 4.856E-03 kg.m^2, Izz = 8.801 kg.m^2, rotor moment of inertia Ir = 3.357E-05 kg.m^2, l = 0.225 m, b = 1.140E-07, k = 2.980E-06, g = 9.81 m/s^2, angle ψ0 = 180 deg, angle φ0 = 0 deg, height = 0 m. The PD controller purpose is to stabilize parameters ψ = 0 deg, φ = 10 deg, height = 10 m. Values of PD coefficients used in eq. (20) have been borrowed from [1]:

Table 1. Values of PD coefficients

Parameter	**Value**	**Parameter**	**Value**
Kzd	2.5	Kzp	1.5
Kφd	1.75	Kφp	6
Kθd	1.75	Kθp	6
Kψd	1.75	Kψp	6

Obtained results are shown in following fig. 4:

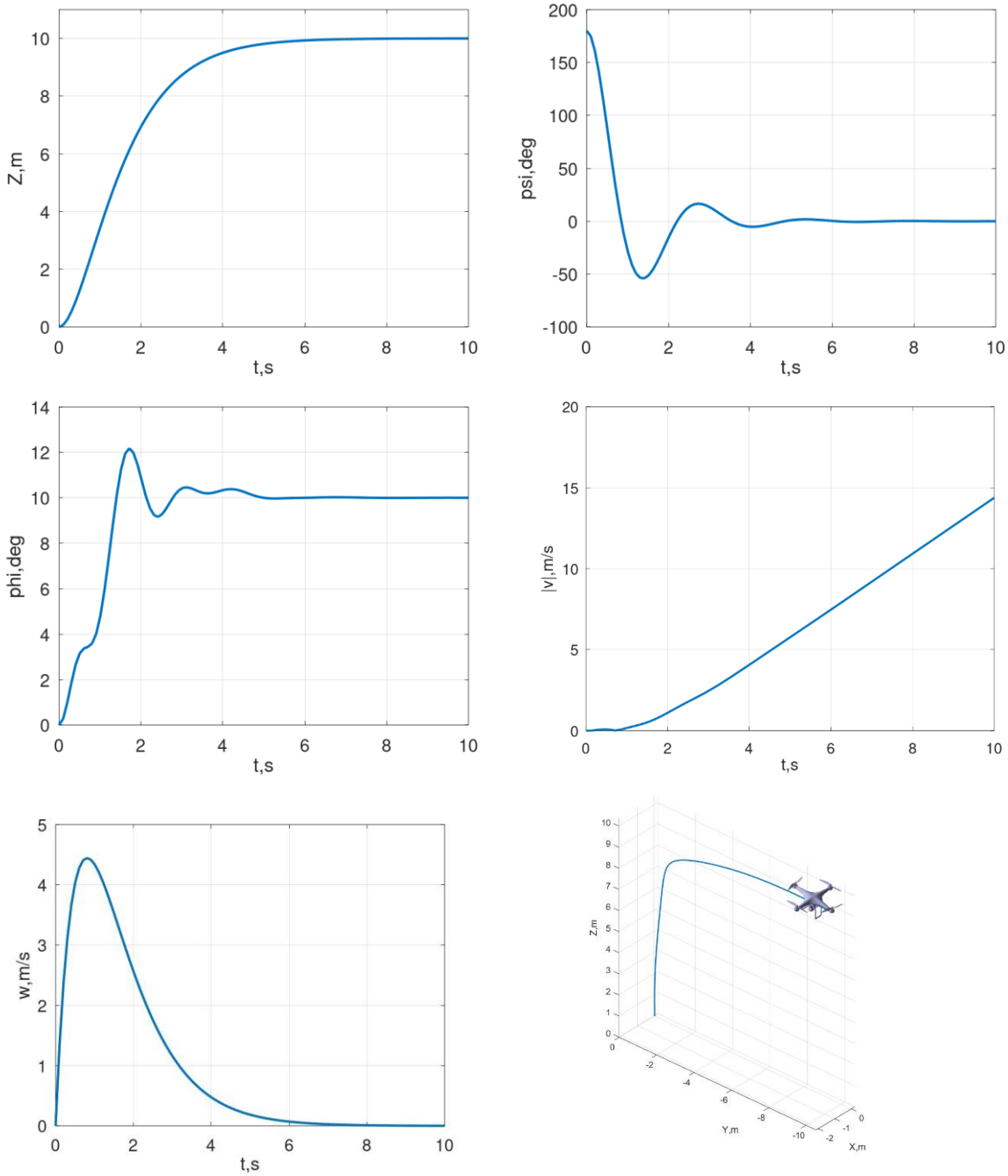


Figure 4. PD controller in use

During flight, the PD controller is adjusting motor revolutions as shown in fig. 5

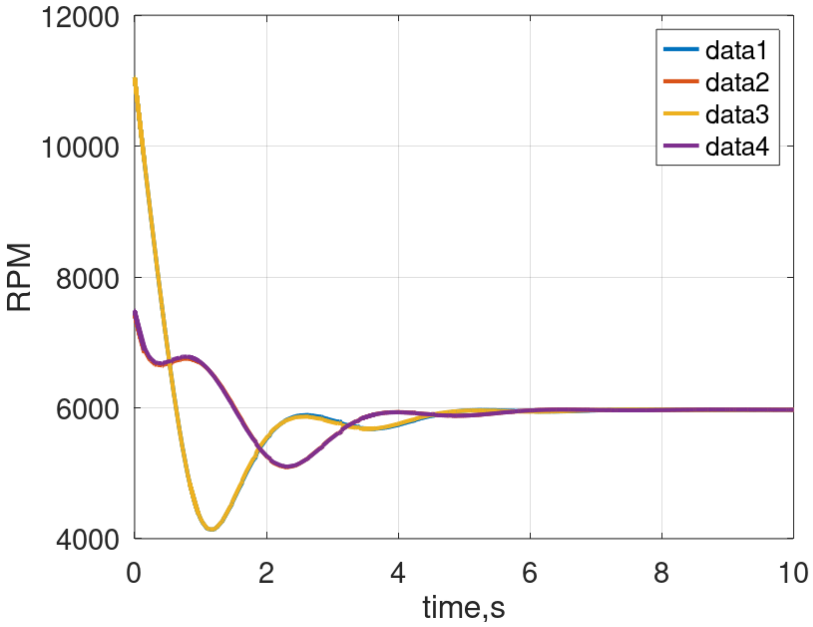


Figure 5. Motors' revolutions

5. Conclusions

The quadcopter is flying along a straight line at height 10 m, as it is shown in fig. 4, after tilting at an angle $\varphi = 10$ deg. The algorithm does not follow route assigned in advance. Hence, the quadcopter performs as expected. Regarding the transient response, a great deal of overshoot is observable in reaching target Euler angles. This presumably might be compensated by adjusting controller coefficients, which is necessary in order to design a controller that behaves predictably and effectively. GNU Octave, [8] with free license has been used to implement the project. Developed source code can be downloaded from link [9]. The CAD model presented in fig. 1 has been developed in Autodesk Inventor environment with one-month free license. It can be downloaded from [10]. Both the code and the CAD model are main contribution of the presented paper.

References

1. Luukkonen, T., Modelling and Control of a Quadcopter, research project, Aalto University, 2011
https://sal.aalto.fi/publications/pdf-files/eluu11_public.pdf
2. Gibiansky, A., Quadcopter Dynamics and Simulation, blog archive, 2012
https://andrew.gibiansky.com/blog/physics/quadcopter-dynamics/
3. Alderete, T., Simulator Aero Model Implementation, NASA Ames Research Center, Moffett Field, California
https://aethercosmology.com/uploads/short-url/rBEb7fKKTz0TCW5RpEDSOFPmQyl.pdf
4. Tytler, Ch., Anatomy of Motion, A Blog on Autonomy and Dynamics …, 2018
https://pdfcoffee.com/2-quad-pdf-free.html
https://github.com/charlestytler/QuadcopterSim
5. Benić, Z., P. Piljek, D. Kotarski, Mathematical Modelling of Unmanned Aerial Vehicles with Four Rotors, Interdisciplinary Description of Complex Systems 14(1), 88-100, 2016
https://hrcak.srce.hr/file/223725
6. Wyss-Gallifent, J., Lecture notes Math431, Mathematics and Geometry for Computer Graphics, chapter Quaternions, Department of Mathematics, University of Maryland, 2021
https://math.umd.edu/~immortal/MATH431/book/ch_quaternions.pdf
7. Kuipers, Jack, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality, Princeton University Press, 1999, ISBN 0-691-10298-8, p. 126
8. https://octave.org/
9. https://github.com/samolet4e/Quadcopter/
10. https://grabcad.com/library/quadcopter-35

Реализиране на пропорционално-диференциращ регулатор за стабилизация на квадрокоптер

Константин Методиев

Секция Аерокосмически системи за управление, Институт за космически изследвания и технологии, Българска академия на науките, София, България, komet@space.bas.bg

Резюме: В настоящата статия е описан математически модел на пространственото движение на квадрокоптер. Пропорционално-диференциращ регулатор е използван в модела за стабилизация на квадрокоптера в пространството. Получената система обикновени диференциални уравнения от първи ред (спрямо линейните и ъглови скорости) с постоянни коефициенти е решена по явен метод Dormand-Prince от ред 4 с адаптивна стъпка по време. Методът е реализиран в среда GNU Octave чрез функция `ode45`, предназначена за решаване на система от нетвърди системи обикновени диференциални уравнения от първи ред. Математическият модел е валидиран с точни решения. Получените резултати за работата на регулатора са представени в графичен вид и обсъдени.

ANTI-SATELLITE MILITARY SYSTEMS FOR INFORMATION SUPERIORITY IN MILITARY OPERATIONS

Asen Angelov Marinov

Bulgarian Air Force Academy, Faculty of Aviation, asen_aerodynamics@abv.bg

Abstract: Anti-satellite technologies are military systems developed to destroy, damage or jam enemy satellites. They play a key role in modern space military strategies, being used to neutralize enemy intelligence, communication and navigation assets. The article provides an overview of existing anti-satellite military systems for information superiority in conducting military operations.

Keywords: anti-satellite, satellites, drone.

1. Introduction

Anti-satellite technologies are military systems designed to destroy, damage, or jam enemy satellites. They play a key role in modern space warfare strategies, being used to neutralize enemy intelligence, communications, and navigation assets.

The main types of anti-satellite weapons are presented in Figure 1 [2,5].

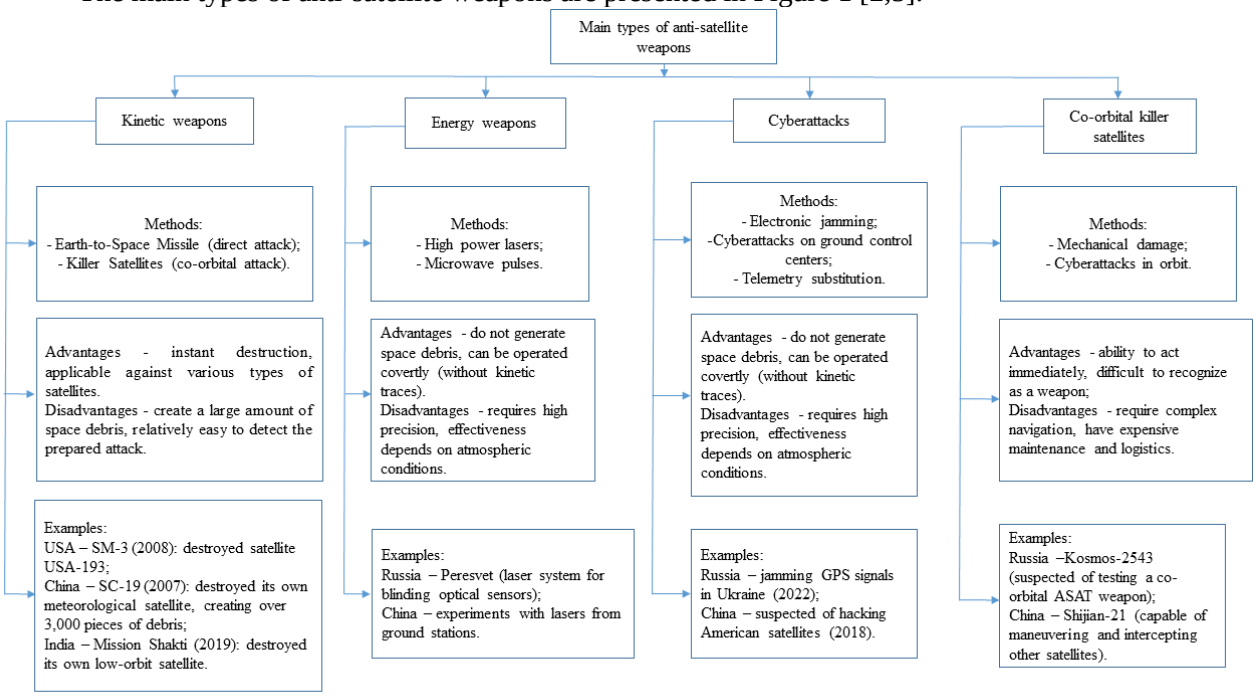


Figure 1. Main types of anti-satellite weapons

2. Basic methods of anti-satellite warfare

The basic methods of conducting anti-satellite warfare are presented in Figure 2 [6,5].

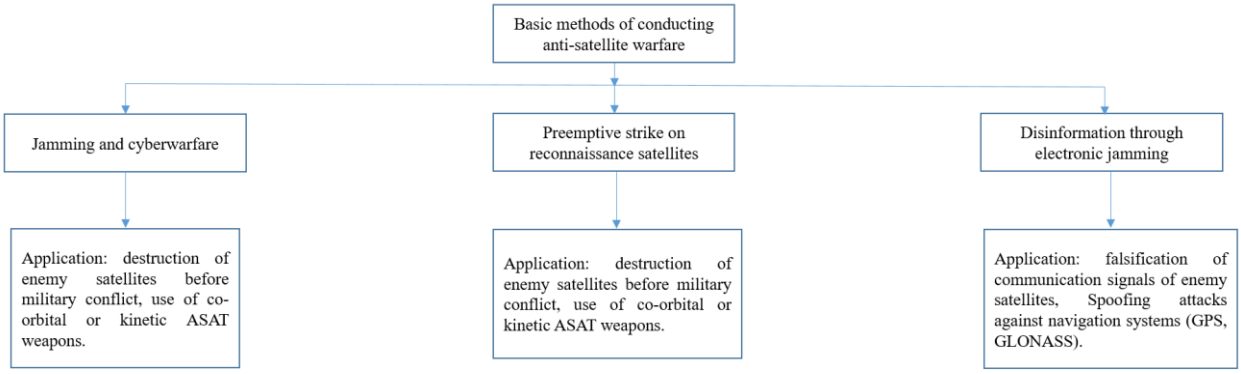


Figure 2. Basic methods of conducting anti-satellite warfare

The main countries possessing ASAT capabilities are presented in Figure 3 [9].

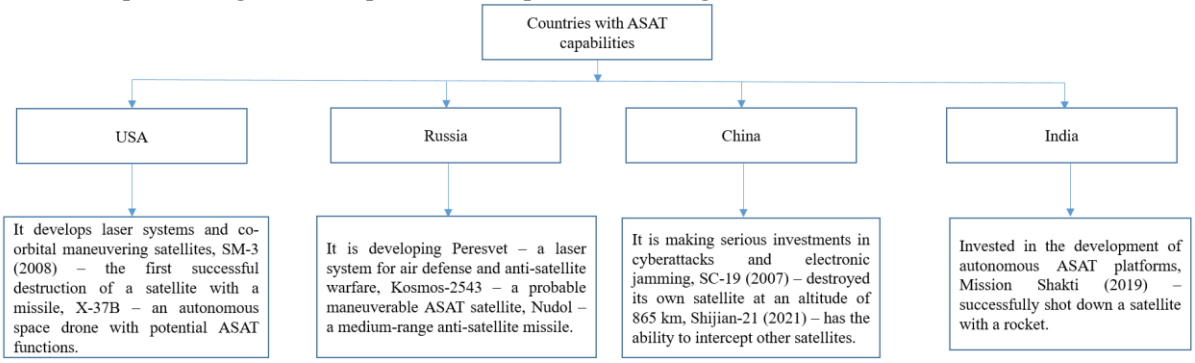


Figure 3. Major countries with ASAT capabilities

3. Development of anti-satellite weapons

The development of anti-satellite weapons is schematically presented in Figure 4 [3,4].

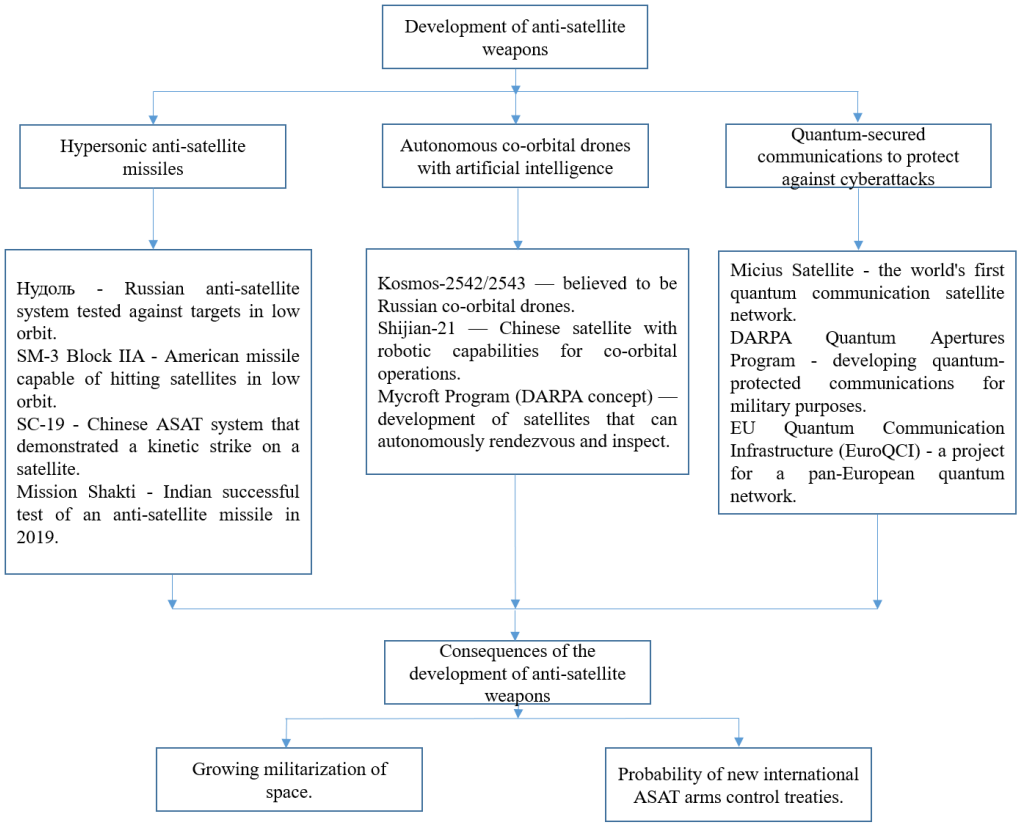


Figure 4. Development of anti-satellite weapons

3.1. Co-Orbital Autonomous Drones – the Future of Orbital Warfare

Co-orbital autonomous drones are killer satellites that can autonomously maneuver in orbit, identifying, tracking, and potentially neutralizing enemy satellites. They are part of a new generation of anti-satellite (ASAT) systems that use artificial intelligence (AI) and advanced maneuvering systems.

Figure 5 presents the main characteristics of autonomous co-orbital drones [5, 8].

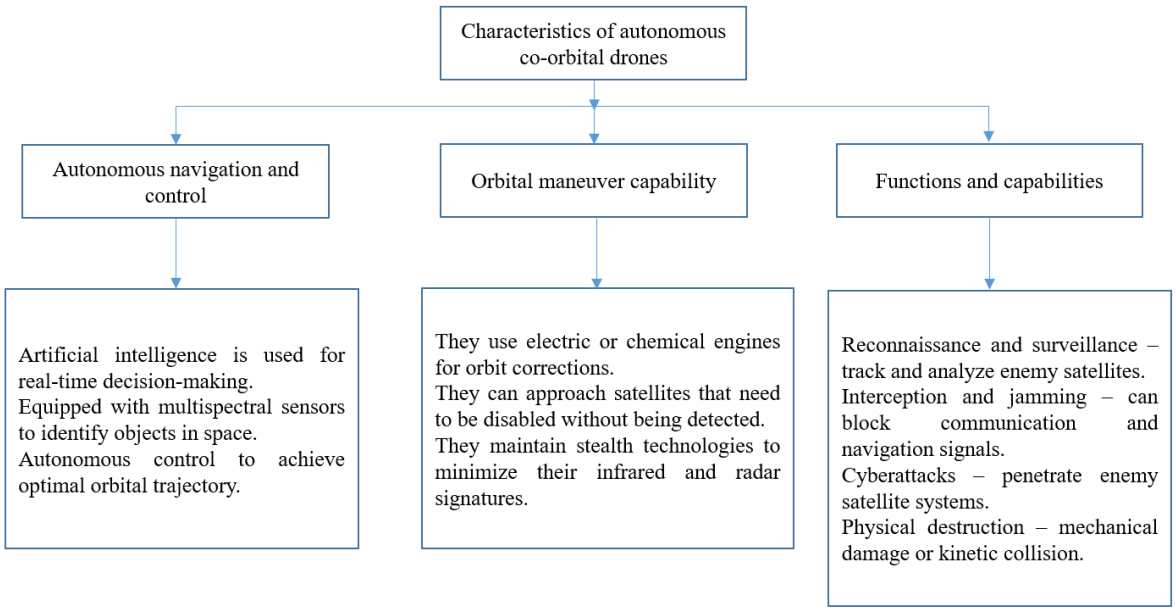


Figure 5. Characteristics of autonomous co-orbital drones

4. Basic tactics of co-orbital drones

The basic tactics of co-orbital drones are presented in Fig. 6 [1,6].

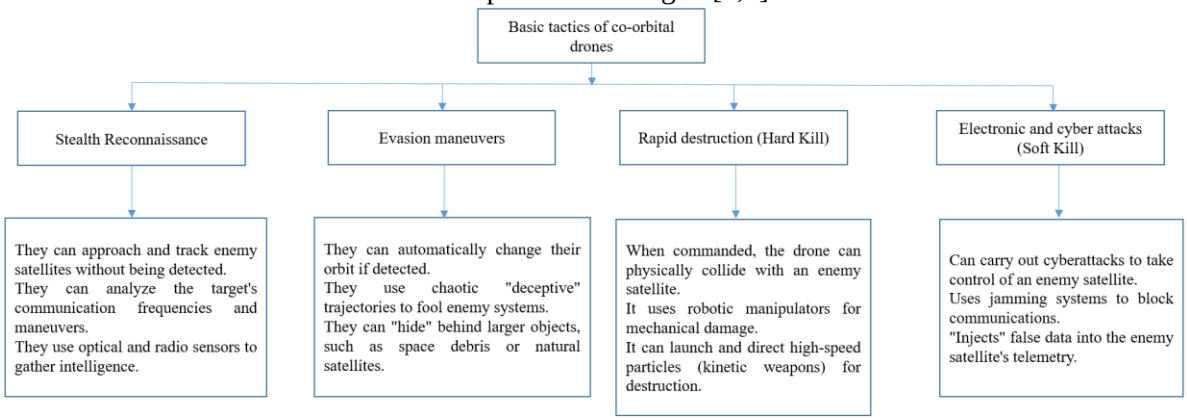


Figure 6. The basic tactics of co-orbital drones

4.1. Passive Shadowing

The goal of passive tracking (Passive Shadowing) is the undetected tracking of an enemy satellite, in which the drone occupies an almost identical orbit, but with a small phase difference (e.g. 10–100 km behind the target). This makes the drone difficult to detect and allows it to collect sensor information about the trajectory, maneuvering behavior, radio and thermal emissions, and potential weak spots for a long time.

To achieve the goal, the same orbit is chosen as that of the target satellite - altitude, inclination, and eccentricity. The drone is placed at a small phase difference behind or in front of the satellite. It is necessary to use minimal position corrections so as not to reveal its presence, and the sensors (optical, infrared, radio frequency) operate in passive mode - they only observe, without emitting signals.

Advantages of passive tracking are that it makes detection almost impossible, it is suitable for reconnaissance and characterization of objects, monitoring repairs, does not violate international treaties, as it does not cause direct damage or interference. It can also be used to monitor one's own satellites in case of failure or need for diagnostics.

Risks and limitations are associated with performing a sudden maneuver, in which the drone can be detected, in certain geometries it can be detected by optical telescopes on Earth and does not allow direct impact (only observation).

The technical characteristics of passive tracking are presented in Figure 7 [7, 8].

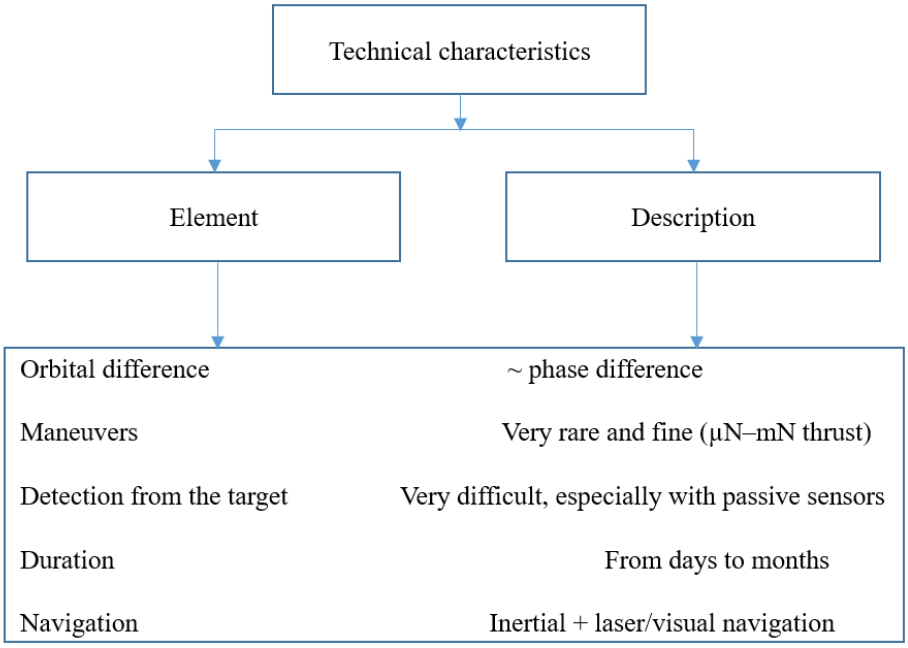


Figure 7. Technical characteristics

Examples of similar (suspected) missions are Kosmos-2542 and 2543 – Russian satellites observed approaching other objects and Shijian-17 – a Chinese object with co-orbital maneuvers and an unconfirmed function [7, 9].

Generally, 6 stages can be defined, presented in Table 1.

Table 1

Stage Action Goal	Stage Action Goal	Stage Action Goal
1.	Analysis and Planning	Target Identification
2.	Launch	Entering an appropriate orbit
3.	Setup	Establishing phase separation
4.	Surveillance	Gathering intelligence
5.	Stealth Tactics	Avoiding detection
6.	Completion	Data transmission or escalation

In the first stage, the analysis and planning are aimed at identifying the target satellite, determining orbital parameters and calculating the phase difference for covert tracking. This includes [5]:

• orbit selection - altitude, inclination and period;
• determination of phase distance;
• analysis of visibility and expected communications.

The second stage involves launch and orbit insertion to insert the drone into a similar orbit to the target satellite, with a slight phase difference, as the mission can be group or solo with the ability to perform maneuvers to fine-tune the orbit and minimal signal emission to avoid detection.

The third stage requires position adjustment to maintain a stable phase distance to the target, electric or ion engines can be used, maneuvers are performed using the Earth's shadow for cover and infrequent adjustments are made to save fuel.

During the fourth stage (passive surveillance), the goal is to collect intelligence data without emitting any signals, optical and infrared sensors are used, radio frequency signal monitoring is performed and the use of LiDAR (Light Detection and Ranging)/laser rangefinders is possible.

The fifth stage (Detection Avoidance) aims to maintain stealth and minimize the risk of detection, being characterized by very low temperature and electromagnetic radiation, without any active communications during the flight, and maneuvering is performed only when necessary.

The sixth stage (Mission Completion or Escalation) aims to transmit data or move to the next phase, download data to a control center, exit from orbit (deorbit), and if necessary, move to active actions (jamming, sabotage).

4.2. Proximity Flyby

A close-orbit flyby is a maneuver in which a drone or satellite approaches a target object in orbit at a short distance to collect intelligence, perform an inspection, or exercise a demonstration of capabilities. Its purpose is to observe the design, behavior, inspection, or calibration of the target satellite.

Tactical tasks for which a close-orbit flyby can be used include visual and infrared inspection of the external configuration, analysis of electromagnetic activity, checking for possible vulnerabilities, and demonstration of access and potential impact capability.

The stages of execution include [5]:

1. Phasing Maneuver - the drone performs maneuvers to synchronize its orbital period with that of the target to ensure the exact time and place of rendezvous.

2. Approach Phase:

• movement in a local coordinate frame (RTN – Radial, Tangential, Normal);

• the distance is reduced from hundreds of km to a few km.

3. Flyby Proper:

• passing a minimum safe distance;

• Data collection through passive and active sensors.

4. Retreat/Drift Away - after completing the observation, the drone moves away, restoring its phase distance to avoid detection or collision.

The typical duration for the stages is about 60 minutes.

Countermeasures (on the target side):

• evasive maneuvers;

• activation of a safe mode;

• detection through laser or radar sensors;

• emission of warning signals.

4.3. Co-Orbital Parking

Orbital dwell is a maneuver in which a co-orbital drone takes a stable position near a target satellite, without approaching it too closely. This position allows it to continue observation or be ready for rapid intervention, while avoiding direct detection [5].

The goal is a constant presence in a close orbit, with the drone positioned at the same height and inclination to be able to react quickly when needed and is used for tactical readiness, surveillance or electronic attack.

This maneuver allows for continuous observation of the target, minimal intervention - passive behavior, preparation for flyby, rapprochement or interception and maintaining a stable relative position through minimal corrections.

Its advantages include - difficult detection (especially if the stay is carried out in the shadow or behind the target sensors), fuel economy (relatively low need for maneuvers) and the ability to immediately transition to other maneuvers - reapprochement, orbit change, withdrawal.

Types of orbital stay [5]:

- trailing - the drone is in the same orbit, but behind the target;

- leading - slightly ahead of the target to stay in close projection;

- above/below the target - slightly higher/lower orbit to stay in close projection;

- synchronous stay - the same orbit, but with finely compensated position maneuvers.

Typical orbital stay parameters and trajectory are presented in Table 2.

Table 2

Parameter	Value
Distance from target	In kilometers
Relative velocity	Positionally stable
Duration	Hours, days, weeks
Sensors used	Passive, wide-spectrum

4.4. Phasing Maneuver

A catch-up maneuver is a temporary move of a spacecraft to a lower (or higher) orbit in order to change its orbital speed relative to a target object, so that after a certain number of orbits it becomes phase-locked.

Its purpose is to be used in satellite interception, by changing the orbital speed in order for the drone to catch up (or avoid) the target through natural orbital cycles.

This requires precise calculations and fuel.

When the target is ahead, the drone lowers its orbit → its orbital period shortens → it moves faster → overtaking occurs.

When the target is behind, the drone climbs to a higher orbit → its orbital period increases → it moves slower → the target "overtakes" it.

To perform a catch-up maneuver, the following steps must be performed [5]:

1) calculate the phase lag or overtaking (in degrees or km);

2) determine a suitable time orbit – with a shorter or longer orbital period;

3) perform a maneuver (ΔV_1) to enter a phasing orbit;

4) wait for the right time (phase alignment);

5) maneuver (ΔV_2) to return to the starting orbit upon reaching the required position.

The time to approach the target can be calculated as follows:

$$t = \frac{\Delta\theta}{2\pi} \cdot \frac{T_t . T_p}{|T_t - T_p|},$$

where:

T_t – target period;

T_p – period of the phasing orbit;

$\Delta\theta$ – phase difference.

The main characteristics of the phasing maneuver are presented in Table 3.

Table 3

Parameter	Value (example)
ΔV_1 and ΔV_2	Depending on the orbit
Duration	Minutes to hours
Energy efficiency	High
Applications	Meeting, replacement, observation

The Phasing Maneuver allows for "invisible approach" (without direct approach from the start), flexibility in precision collision or observation, and is a fundamental part of co-orbital behavior and servicing of satellites.

4.5. Stealth approach [2]

The goal is for the drone to approach a target satellite without being detected by onboard sensors, ground surveillance, or other security systems by:

• minimizing thermal/RF signature;

• avoiding direct trajectories;

• using orbital geometry for concealment.

Step-by-Step execution sequence:

1) initialization - the drone enters an orbit compatible with that of the target, but with a temporary offset (in phase, altitude, or inclination);

2) passive phasing - uses a temporarily lower orbit to approach in phase - without propulsion pulses near the target;

3) orbital dwell - stays in a close parallel orbit 10–50 km from the target in its dead zone;

4) final shadow approach - approaches during a passage into the Earth's shadow, when infrared sensors are less effective;

5) Position stabilization - maintains co-orbital position with minimal adjustments - thus remaining undetectable.

Disadvantages:

• requires extremely precise navigation equipment;
• potential collision in case of synchronization error;
• limited window of action (in shadow and minimal signals).

An example comparison between aggressive and stealth approaches is presented in Table 4.

Table 4

Parameter	Aggressive approach	Stealth approach
ΔV (fuel consumption)	High	Moderate – low
Detection	High	low
Duration	Short	Longer
Risk of detection	High	low

4.5.1. Passive surveillance after stealth approach [2, 5]

The goal of passive surveillance after stealth approach is for the drone to remain in a close but hidden orbital position relative to the target and collect information about communications, movement, telemetry, physical structure (via passive sensors).

Execution sequence:

1) positioning - stands behind or below the target satellite;
2) passive surveillance - uses radio/optical/IR sensors without transmission;
3) detection avoidance - maintains a constant angle outside the "line of sight";
4) data collection - records operating modes, orbital maneuvers and signals;
5) target tracking - uses automatic co-orbital tracking with minimal ΔV.

Disadvantages:

• possible detection when illuminated by the Sun;
• positioning errors can lead to collision;
• passive data has limited accuracy.

4.5.2. Orbital Maneuver for Interruption, Replacement or Deactivation

The purpose of the orbital maneuver for disruption, replacement or deactivation is an action against the target satellite after successful approach and tracking – through [2]:

- disruption;
- replacement;
- deactivation.

The purpose of disruption is temporary or partial neutralization without physical contact, by:

• emission of electromagnetic pulses;
• jamming by a directional jammer;
• laser or microwave attack on optics/sensors.

Its advantage is that it is difficult to detect and leaves no traces.

Replacement requires high maneuverability and synchronization, and deactivation aims at permanent destruction of the target, by a kinetic impact, which creates space debris and is easily detected.

5. Surveillance via multiple drones

The goal is to use a group of coordinated micro- or nanosatellites (drones) to detect maneuvers, interference, or attacks from multiple directions [5].

The implementation of surveillance using multiple drones is performed in the following stages:

• approach to the target - the drones are deployed in phase-shifted orbits, providing the possibility of a covert approach from different planes;
• swarm creation - the drones synchronize their positions at a certain distance from the target and form an "envelope" around the satellite;
• active surveillance - synchronous operation of multispectral sensors, RF receivers, and visual cameras and creating a 360-degree picture in real time;

• data analysis and interception - the drones capture side channels, emissions, or anomalies, and the data is transmitted to a ground station or main platform.

6. Soft-kill

The goal is to neutralize (temporarily or permanently) the functionality of a target satellite without physical contact by [2]:

- jamming - broadcasting a powerful signal to the satellite's communication frequencies and interrupting the connection with ground control;

- spoofing - sending false commands or data by imitating a legitimate control signal, which can lead to self-destruction, system shutdown or moving to the wrong orbit;

- GNSS spoofing/jamming to disrupt satellite navigation (e.g. GPS, GLONASS), which would be particularly effective against autonomous satellites.

7. Conclusion

The dependence of modern military operations on satellites is fundamental and affects almost all aspects of the strategic, operational and tactical levels of combat operations through:

• connectivity and communications - SATCOM (Satellite Communications) is the backbone of communication between command centers and combat units, especially in difficult-to-reach, mobile or maritime theaters of operations and provides a stable connection when ground networks are jammed;

• navigation and synchronization (GNSS) - systems such as GPS, Galileo, GLONASS, ... provide precise positioning of troops, platforms and sensors, time synchronization - critically important for cryptography, logistics and coordination of strikes and guidance of missiles, and drones also rely on satellite guidance;

• intelligence, surveillance and reconnaissance (ISR) - satellite intelligence (IMINT, SIGINT, ELINT) provides global overview without the need for a physical presence, constant monitoring of movements, bases, fleets, critical infrastructure and tracking of enemy satellites and anti-satellite activity;

• combat management - satellites support C4ISR systems (Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance), providing situational awareness (SA), maintaining mission autonomy of dispersed forces, supporting platform-neutral tactics (coordination between the Air Force, Navy, Army and Space);

• strategic deterrence and warning - satellite early warning (EW) systems monitor the launch of ballistic missiles and are part of nuclear deterrence through a guaranteed possibility of a retaliatory strike;

• High dependence creates risks of deactivation by anti-satellite weapons (ASAT), cyberattacks against ground stations, jamming and spoofing, and the loss of satellites can lead to operational paralysis.

Without satellites, modern armies would go “blind,” losing synchronization, precision and communications – turning their high-tech forces into disorganized units.

References

1. Kaplan E., Hegarty C, Understanding GPS/GNSS: Principles and Applications, Library of Congress Cataloging-in-Publication Data, 2006
2. Harrison T., Counterspace: The Next Hour of Spaceflight, CSIS Space Threat Assessment, 2022.
3. Lutes C., Toward a Theory of Spacepower, Spacepower: Doctrine and Strategy, National Defense University Press, 2011.
4. Wong W., Military Space Power: A Guide to the Issues, Praeger Security International, 2010.
5. Air University Press / Maxwell AFB, Military Space Power.
6. Center for Strategic and International Studies (CSIS), „Space Threat Assessment“ and „Counterspace Weapons“.
7. JP 3-14 – Space Operations,2020.
8. NATO Space Handbook, 2022.
9. Space Security Index.

ПРОТИВОСПЪТНИКОВИ ВОЕННИ СИСТЕМИ ЗА ИНФОРМАЦИОННО ПРЕВЪЗХОДСТВО ПРИ ВОЕННИ ОПЕРАЦИИ

Асен Ангелов Маринов

Резюме: Противоспътниковите технологии са военни системи, разработени за **унищожаване, повреда или заглушаване на вражески спътници**. Те играят ключова роля в съвременните космически военни стратегии, като се използват за неутрализиране на разузнавателни, комуникационни и навигационни активи на противника. В статията е направен обзор на съществуващите противоспътникови военни системи за информационно превъзходство при провеждане на военни операции.

Mathematical model for the development of a critical situation in the event of illegal interference with the work of the crew in flight, based on Markov random processes

Nikolay Zagorski

Space Research and Technology Institute, Bulgarian Academy of Sciences, Sofia, Bulgaria,
nzagorski@space.bas.bg

Abstract: This article presents an analytical method for studying the flow of events in cases of unlawful interference with the work of the crew during a flight based on the Markov random process method.

Key words: *terrorist act, event flow, stationary flow*

1. Introduction

Currently, international civil aviation is perceived as practically the safest mode of mass transport. According to data from the International Civil Aviation Organization (ICAO), in recent years there has been a continuous and steady growth in the volume of air transport of passengers and cargo. In 2012, aviation operators performed approximately 31.2 million scheduled commercial flights, which is an increase of 3.5% on a three-year basis, as presented in Fig. 1.

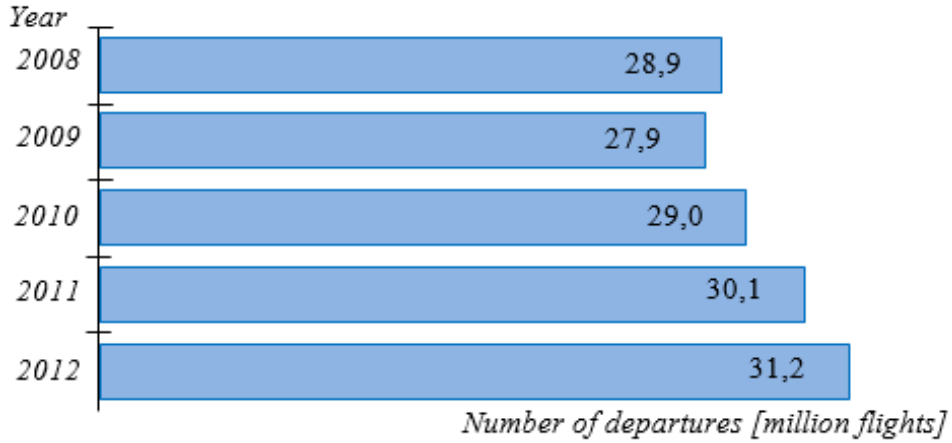


Fig. 1. Global civil aviation air traffic volume

In 2012, the global air transport system carried approximately 2.9 billion people, with scheduled air services showing a 5.5% increase in passenger-kilometres compared to previous years [1]. Furthermore, the wider aviation system now includes several interconnected systems that are geopolitically distinct, yet technologically complex and multi-functional. ICAO Member States agree that, given the complexities and expected increase in the sustainable growth of civil aviation, further efforts are needed to improve aviation safety.

2. Proactive monitoring and enhancing aviation safety

The ICAO Universal Programme for the Ongoing Inspection of the Aviation Safety Oversight Organization provides detailed information on the effective implementation of ICAO Standards and Recommended Practices (SARPs) [2].

Aviation accidents (crashes) in the activities of aviation operators have become such rare events that the Federal Aviation Administration (FAA) of the United States has introduced for use the indicator “fatalities per 100 million passengers carried”. Fig. 2 presents ICAO data on changes in the number of aviation accidents.

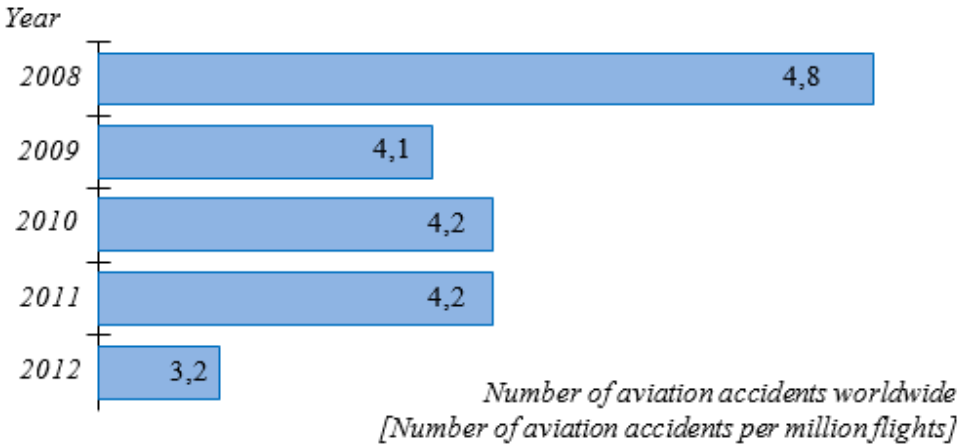


Fig. 2. Frequency of aviation accidents involving civil aviation aircraft worldwide

But despite the significant decrease in the accident rate worldwide, the threat of acts of unlawful interference with the work of the crew in the activity of civil aviation continues to grow in air transport. As a result of terrorist attacks on aircraft and airport terminals alone over the past 20 years in the late 19th and early 20th centuries, more than a thousand people died and huge material losses were caused [1]. The total number of such acts has decreased, but they have become more significant in scale and in the amount of victims and material damage caused. The number of hijackings and attempts to hijack aircraft, the taking of passengers and crew members' hostage has not decreased.

In our modern times, scientific research, in particular in the field of developing mathematical modeling methods, has also made a significant contribution to solving the problem of aviation safety.

When studying random processes occurring in complex systems with discrete states and continuous in time, the so-called "***event streams***" are often used. A stream of events is a sequence of homogeneous events that follow one after another at some random time intervals [3, 4].

The sequence of events, called "***acts of unlawful interference***", occurs at some points in time and constitutes a flow of events. The flow of acts of unlawful interference is a ***stationary flow*** if the probability depends only on the time interval, and not on where exactly on the time axis this section is located.

The flow of acts of unlawful interference is a ***flow without consequences*** if, for any two non-overlapping time intervals, the number of events falling in one of these intervals does not depend on the number of events falling in the other.

A flow of events, such as acts of unlawful interference, constitutes an ***ordinary flow*** if the probability of two or more events occurring in a short time interval is negligible compared to the probability of just one event.

Thus, the flow of acts of illegal interference is the ***simplest flow*** (or, ***stationary Poisson flow***). This fact allows the application of the apparatus of Markov random processes to the study of acts of illegal interference. The Markov random process describes probabilities that are a function of time, a system of differential equations, also called the Kolmogorov-Fokker-Planck equations (a time-continuous Markov chain). The main feature of Markov processes is that their conditional probability density functions $f(t, X, \tau, Y)$ satisfy partial differential equations of parabolic type [3, 4]. When compiling these equations, it is convenient to use a "***state graph***", the vertices of which are states, and the connecting arcs – the possible transitions from one state to another. In Fig. 3 presents a state graph that describes the possible transitions for the random refinement process upon the occurrence of an act of illegal interference.

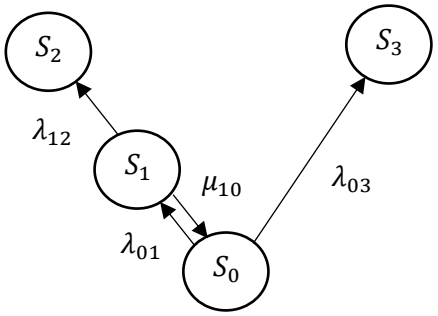


Fig. 3. State graph for possible transitions for the random process of perfecting an act of illegal interference

The conditional probability of a transition from one state to another is called the transition probability from state i to state j [3, 4]. Transition probabilities are elements of the transition matrix. The sum of the transition probabilities from each column of the matrix is equal to one:

(1) $\sum_{j=1}^{k} P_{ij} = 1 \ (i = 1, 2, \dots, k)$.

The derivative of the probability of the state k is equal to the sum of the terms, each of which is the product of the weight of the arc incident on the vertex k by the probability of the state to which it is directed. The weight of the arc is taken to be positive if the arc originates from the vertex k and negative if the arc is directed towards the vertex k.

When a random process occurs in a system, it is called a "Markov process" if, for each time instant t_0, the probable characteristics of this process in the future depend only on its state at the given time instant and do not depend on when and how the system reached this state [3, 4].

The process of counteraction by aviation security services at airports and in aviation operators against subjects of illegal activity can be considered a random Markov process, especially taking into account the fact that parameters from the "past" can be included in the "present". If considered in this way, the process can be attributed to the group of continuous processes in time with discrete states.

This means that such a process is described by a finite number of possible states that can be uniquely determined. For example, the act of illegal interference has been committed, or the act has been prevented, i.e. the state of the process can be discretized into separate states, which can be assigned numbers in the sequence of execution of the process. The considered continuity in time in such a process consists in the fact that acts of illegal interference occur at indefinite, random moments of time that are not known and fixed in advance.

The analysis of random processes with discrete states is carried out using a state graph. Such a state graph is described by a flow of events [5]. Each arrow in the state graph, which shows the flow of events, also shows the intensity of the flow of events that takes the system from one state to another along the depicted flow (along the given arrow). The intensity of the flow of events that lead the system from state S_i to state S_j is denoted by X_{ij}. The reverse flow from event S_j to event S_j is denoted by λ_{ij} or μ_{ij}. A state graph with arrows of transitions and the intensities of these transitions is called a marked (labeled) graph.

Let us compose and examine the following system states describing aviation security, as Fig. 3 presents a labeled graph of this system:

S_0 – normal or safe state of the system, no acts of unlawful interference in flight;

S_1 – an attempt to commit prohibited acts by passengers on a flight was discovered;

S_2 – an attempted act of unlawful interference with a flight was prevented;

S_3 – an act of unlawful interference with a flight has been committed.

The mathematical model is built on the basis of the marked graph (Fig. 3). If the system has the following possible states: S_1, S_2, …, S_n, then the probability of the existence of the i-th state is $P_i(t)$, i.e. the probability that at time t the system is in state S_i. Therefore, for each time point the following is valid:

(2) $\sum_{i=1}^{n} P_i(t) = 1$.

To determine all $P_i(t)$ for all $i = \overline{1, n}$, Kolmogorov equations are compiled and solved. In these equations, the unknowns represent the probabilities of the individual states of the system [6, 7]. On the left side of each of the equations is the derivative of the probability for some i-th state. On the right side of the equations are the sum of the products of the probabilities of all states from which arrows (transitions) to the given state emerge, minus the sum of the intensities of all flows leading the system out of the given state, multiplied by the probability of the given i-th state.

Using the rule thus indicated, a system of differential equations can be constructed that describe the marked state graph, which is presented in Fig. 3. In this case, it is assumed that all initial conditions are zero, except for the initial one, which is equal to the probability of a credible event.

$$(3)\quad \begin{cases} \frac{dP_0(t)}{dt} = \mu_{10}P_1 - (\lambda_{01}+\lambda_{13})P_0, \\ \frac{dP_1(t)}{dt} = \lambda_{10}P_0 - (\mu_{10}+\lambda_{12})P_0, \\ \frac{dP_2(t)}{dt} = \lambda_{12}P_1, \\ \frac{dP_3(t)}{dt} = \lambda_{13}P_0. \end{cases}$$

This is a system of four homogeneous differential equations with constant coefficients and it can have an analytical solution under the assumption that λ=const. Due to the limitations of the volume of the presented materials, the sequence for solving the differential equations is not presented. The solutions have the following form:

$$(4)\quad P_0 = c_1e^{-\alpha_1 t} + c_2e^{-\alpha_2 t} = e^{\frac{(-\alpha_2+\lambda_{01}+\lambda_{13})^{-\alpha_1 t-\lambda_1 t}}{a_2-a_1}} + e^{\frac{(-\alpha_1+\lambda_{01}+\lambda_{13})^{-\alpha_1 t-\lambda_1 t}}{a_2-a_1}};$$

$$(5)\quad P_1(t) = \frac{(-\alpha_1+\lambda_{01}+\lambda_{13})(-\alpha_2+\lambda_{01}+\lambda_{13})}{\mu_{01}(a_2-a_1)}\text{x}(-e^{-\alpha_1 t} + e^{-\alpha_2 t});$$

$$(6)\quad P_2(t) = \lambda_{12}\frac{(-\alpha_1+\lambda_{01}+\lambda_{13})(-\alpha_2+\lambda_{01}+\lambda_{13})}{\mu_{10}(a_2-a_1)}\text{x}\left(\frac{-e^{-\alpha_1 t}}{\alpha_1} - \frac{e^{-\alpha_2 t}}{\alpha_2}\right) - \frac{\lambda_{12}(-\alpha_1+\lambda_{01}+\lambda_{13})(-\alpha_2+\lambda_{01}+\lambda_{13})}{\mu_{01}a_1a_2};$$

$$(7)\quad P_3(t) = \frac{\lambda_{13}(-\alpha_1+\lambda_{01}+\lambda_{13})}{\alpha_1(a_2-a_1)}e^{-\alpha_1 t} - \frac{\lambda_{13}(-\alpha_2+\lambda_{01}+\lambda_{13})}{\alpha_2(a_2-a_1)}e^{-\alpha_2 t} + \frac{\lambda_{13}(-\alpha_2+\lambda_{01}+\lambda_{13})}{\alpha_2(a_2-a_1)} - \\ - \frac{\lambda_{13}(-\alpha_2+\lambda_{01}+\lambda_{13})}{\alpha_1(a_2-a_1)} = \frac{\lambda_{13}(-\alpha_2+\lambda_{01}+\lambda_{13})}{\alpha_1(a_2-a_1)}(1-e^{-\alpha_1 t}) + \frac{\lambda_{13}(-\alpha_2+\lambda_{01}+\lambda_{13})}{\alpha_2(a_2-a_1)}(1-e^{-\alpha_2 t}).$$

In this way, all probabilities for the states as a function of time have been obtained. Then we can proceed to the stage of verification of the obtained expressions based on relation (2). Verification is a check of the relation:

$$(8)\quad P_0(t) + P_1(t) + P_2(t) + P_3(t) = 1.$$

Next, one must check the coincidence of the final probabilities of the events at the initial moment of time and examine the functional dependences of the probabilities of the events as a function of time.

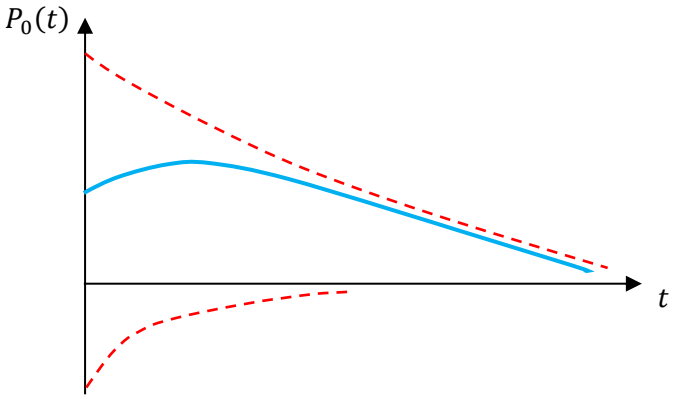


Fig. 4. Functional dependence of $P_0(t)$

Fig. 4 shows an example curve of the functional dependence $P_0(t)$.

Let's examine the functional dependence $P_1(t)$:

$$(9)\quad \lim_{t\to 0} P_1(t) = \frac{(-\alpha_1+\lambda_{01}+\lambda_{13})(-\alpha_2+\lambda_{01}+\lambda_{13})}{\mu_{10}(a_2-a_1)}\text{x}(-e^{-\alpha_1 t} + e^{-\alpha_2 t}) = 0.$$

$$\lim_{t\to 0} P_1(t) = \frac{(-\alpha_1+\lambda_{01}+\lambda_{13})(-\alpha_2+\lambda_{01}+\lambda_{13})}{\mu_{10}(a_2-a_1)}\text{x}(0+0) = 0.$$

In a similar way, the analytical dependences of $P_1(t)$, $P_2(t)$ and $P_3(t)$, were obtained, which are graphically presented, respectively, in Fig. 5, Fig. 6 and Fig. 7.

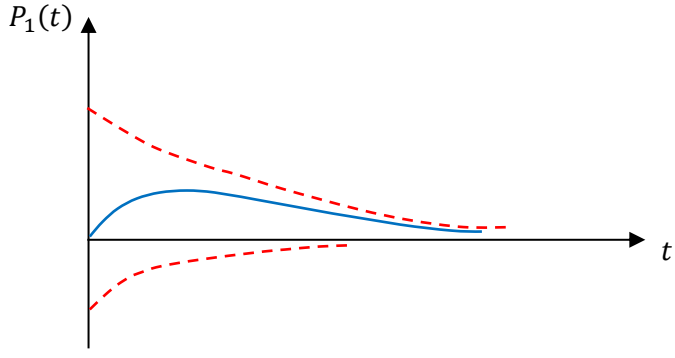


Fig. 5. Functional dependence of $P_1(t)$

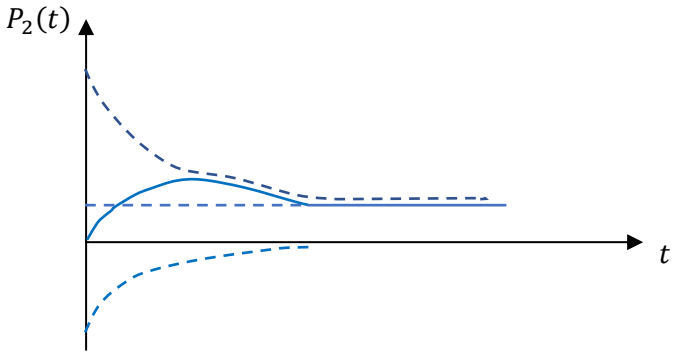


Fig. 6. Functional dependence of $P_2(t)$

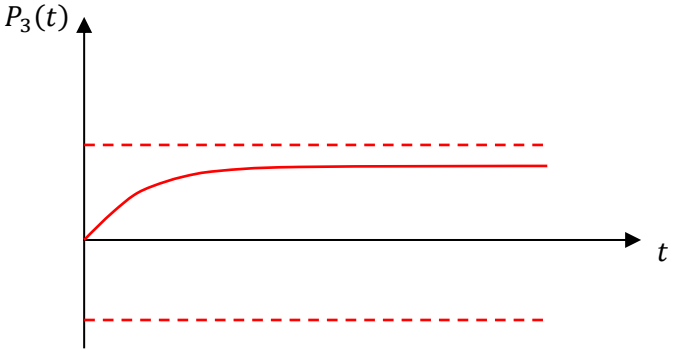


Fig. 7. Functional dependence of $P_3(t)$

3. Conclusion

The presented reasoning and the obtained results allow to determine the probabilities of the occurrence and development of a critical situation related to illegal interference in the work of the flight crew during a flight, to implement proactive surveillance and to increase the level of aviation safety.

References

1. Flight safety. State of Flight Safety in the World. ICAO. 2013 Edition. H3C5H7. 54 p. www.icao.int
2. ICAO Doc 10004. Global Aviation Plan. 2020-2022 Edition. 164 p. www.icao.int
3. *Ventzel E. S., Probability Theory: a textbook for universities/E. S. Ventzel. - 11th ed., reprinted. - Moscow: Knorus, 2016. - 658 p.: ill., diagrams – ISBN 978-5-406-00476-0 (in Russian language)*
4. *Gmurman V. E., Probability Theory and Mathematical Statistics: Textbook for Universities / V. E. Gmurman – 12th ed. – Moscow: Yurait Publishing House, 2022. 479 p. ISBN 978-5-534-00211-9 (in Russian language)*
5. Kolmogorov, A. N. Basic concepts of probability theory, Moscow: Librokom Publishing House, 2013, 113 p. ISBN 978-5-397-03703-7 *(in Russian language)*
6. Kolmogorov, A. N., A. G. Dragalin, Mathematical logic, M.: MSU, 1984, 119 p. *(in Russian language)*
7. *Gmurman V. E., Guide to solving problems in probability theory and mathematical statistics: a textbook for universities / V. E. Gmurman. - 11th ed., revised. and additional. - Moscow: Yurait Publishing House, 2020. - 406 p. - (Higher education).* – ISBN 978-5-534-08389-7 *(in Russian language)*

Математически модел за развитие на критична ситуация при незаконна намеса в работата на екипажа в полет, основана на случайни процеси на Марков

Николай Загорски

Резюме: В настоящия материал е представен аналитичен метод за изследване на потока от събития при актове за незаконна намеса в работата на екипажа по време на полет на основата на метода за случайни процеси на Марков.

Study of the equilibrium of an external fire extinguishing device of the "Bambi Bucket BB4453" type in flight of an AS532AL Cougar helicopter

Nikolay Zagorski

Space Research and Technology Institute, Bulgarian Academy of Sciences, Sofia, Bulgaria, nzagorski@space.bas.bg

Abstract: This article examines and presents the reasons for the influence of the parameters of the external fire extinguishing device of the "Bambi Bucket BB4453" type on its equilibrium during flight of the AS532AL Cougar firefighting helicopter.

Key words: *helicopter, external fire extinguishing device, balance, firefighting flight*

1. Introduction

An external fire extinguishing device of the "Bambi Bucket BB4453" type [1], attached to a central cable under the AS532AL Cougar, has a significant impact on the dynamics of changes in the helicopter parameters during a firefighting flight. This is known both from the practice of flight operations in these aviation activities and from a number of theoretical studies [2–4]. Moreover, there is no comprehensive study of the dynamics of the "Helicopter–External Fire Extinguishing Device" system described in the scientific literature, which would include a description of the issues of equilibrium, balancing, stability, controllability and trajectory movement of the specified system. In turn, the solution of this problem will allow the application of a scientifically based approach to increase the safety of flights when extinguishing fires with the AS532AL Cougar helicopter.

2. Research methodology

If the external fire extinguishing device is represented as two separate elements (central cable and "BB4453"), which are connected by means of a spherical joint at point O_3 (as presented in Fig. 1), then the equilibrium conditions of the central cable and the equilibrium conditions of "BB4453" must be considered separately.

We will assume that the external fire extinguishing device has an axisymmetric shape, with its center of mass located in the vertical plane of symmetry $O_2X_2Y_2$. The external fire extinguishing device is attached to the central cable O_1O_3 by means of 24 short ropes, called "spider webs", which will also be assumed to be absolutely rigid rods. The ends of the "spider web" meet at point O_3, where they are connected to the central cable by means of a spherical joint.

Let us consider the equilibrium of the external fire extinguishing device relative to point O_3. The equations that describe the equilibrium of "BB4453" are derived from the equation of motion of this device [5]:

$$\begin{aligned} &\vec{G}_{el} + \vec{R}_{Ael} - \vec{R}_{el} = 0 \\ &\vec{M}_{Gel} + \vec{M}_{Ael} = 0, \end{aligned} \quad (1)$$

where G_{el} is the gravity force of the external fire extinguishing device; R_{Ael} is the aerodynamic force acting on "BB4453"; R_{el} is the resultant force acting on "BB4453"; M_{Gel} is the moment of gravity of "BB4453"; M_{Ael} is the aerodynamic moment of "BB4453".

In a steady horizontal straight flight, the external fire extinguishing device deviates from the position that the device occupies in the hovering mode of the helicopter, mainly in the plane that coincides with the plane of symmetry of the helicopter $O_2X_2Y_2$. In this sense, we will consider only the longitudinal movement of the "BB4453". In this case, the angular position of the fire extinguishing device relative to the ground is described by the pitch angle ϑ_2 of the "BB4453".

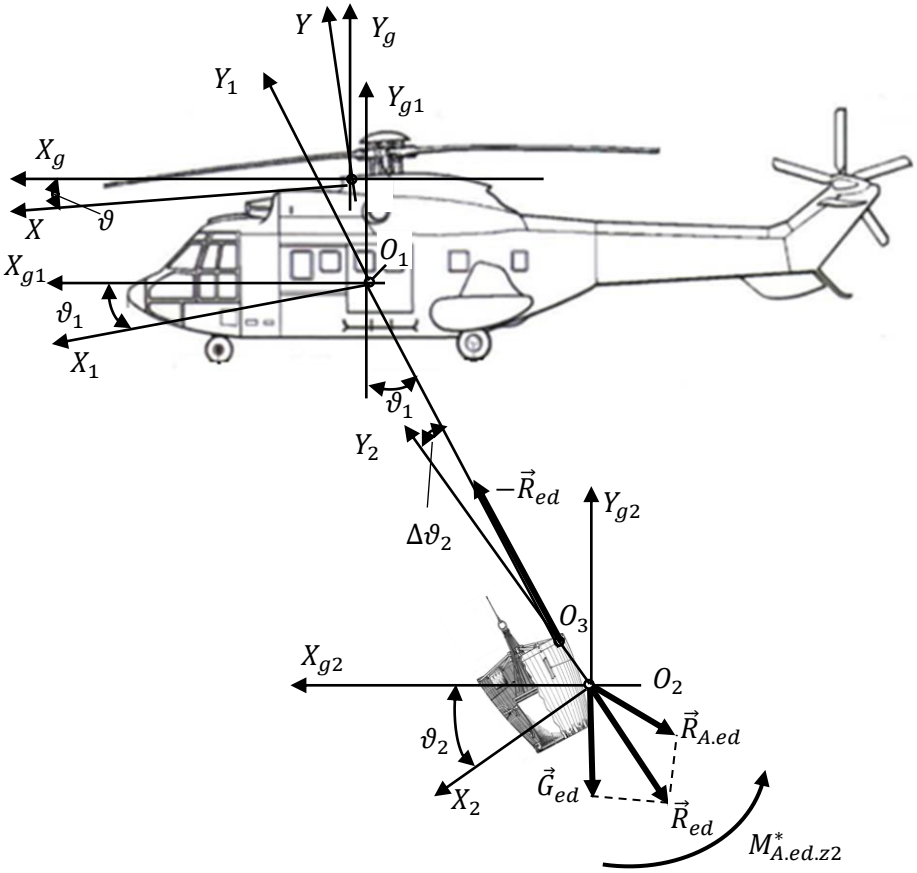


Fig. 1. Equilibrium diagram of the external fire extinguishing device

Let us determine what the magnitude of the equilibrium pitch angle of the external device depends on, as presented in Fig. 1. For this purpose, the equation for the equilibrium of the "BB4453" in pitch in scalar form can be used:

(2) $G_{el} r_{el} \sin\vartheta_2 + m_{zel} \frac{\rho V_{hf}^2}{2} S_{el} l_{el} - X_{el} r_{el} = 0,$

where r_{el} is the distance from point O_3 to the center of mass of the external device - point O_2, with $r_{el} < 0$; ϑ_2 is the pitch angle of "BB4453"; m_{zel} is the pitch moment coefficient of the external fire extinguishing device; V_{hf} is the speed of steady horizontal straight flight of the helicopter; S_{el} and l_{el} are, respectively, the characteristic area and characteristic linear size of the external device; X_{el} is the longitudinal aerodynamic force of "BB4453".

From here, the formula for determining the equilibrium pitch angle of the external fire extinguishing device can be obtained:

(3) $\vartheta_2 = \arcsin\left(\frac{X_{el} r_{el} - m_{zel} \frac{\rho V_{hf}^2}{2} S_{el} l_{el}}{G_{el} r_{el}}\right).$

In the further analysis, some assumptions can be made that will simplify the calculations, but at the same time will not significantly affect the accuracy of the result. We will assume that the pitching moment coefficient of the external fire extinguishing device is negligibly small, i.e. $m_{zel} = \approx 0$. Then we will get:

(4) $\vartheta_2 = \arcsin\left(\frac{c_{x\,el} S_{el}}{m_{el}} . \frac{\rho V_{hf}^2}{2g}\right).$

The analysis of formula (4) shows that under the adopted assumptions, the following parameters have the main contribution to the value of the equilibrium pitch angle of the external fire extinguishing device in a stable horizontal straight flight of the helicopter:

- the coefficient $c_{x\,el}$ of the longitudinal aerodynamic force of "BB4453";
- the characteristic area S_{el} of "BB4453";

- the mass m_{el} of the external fire extinguishing device.

If we additionally introduce the notation:

(5) $c = \frac{c_{x\,el} S_{el}}{m_{el}}$,

we will obtain the following expression for the equilibrium pitch angle:

(6) $\vartheta_2 = \arcsin\left(c \frac{\rho V_{hf}^2}{2g}\right)$.

In its structure and composition, the coefficient *c* corresponds to the so-called ballistic coefficient, which is used in calculating the trajectory of aircraft using a ballistic flight trajectory [6]. For this reason, hereinafter in the material, the coefficient *c* will be considered as the *ballistic coefficient of the external fire extinguishing device*.

It can be summarized that the pitch angle of "BB4453" depends on the ballistic coefficient of the external fire extinguishing device, as well as on the flight parameters of the helicopter - speed and altitude.

Let us consider the conditions for equilibrium of the central cable, as presented in Fig. 1. As was assumed above, if the moment $m_{zel} = \approx 0$, the axis of the central cable will pass through the center of mass of the fire extinguishing device. Then, the pitch angles of the external device and the central cable will approximately coincide, i.e. $\vartheta_1 \approx \vartheta_2$. This means that under the assumptions made, the central cable and the external fire extinguishing device with the "*spider web*" type rope system can be considered as a single body, and the equilibrium of this system is considered relative to the point of attachment of the central cable to the helicopter, i.e. point O_1 in Fig. 1.

In an established straight horizontal flight, the pitch angle of the central cable ϑ_1 coincides with the angle of deviation of the cable from the vertical position, which is one of the most important parameters that characterize the equilibrium position of the external fire extinguishing device in flight.

(7) $\vartheta_2 = -arc\sin\left(\frac{c_{x\,el} S_{el}}{m_{el}} . \frac{\rho V_{hf}^2}{2g}\right) = -arc\sin\left(\frac{c_{x\,el} S_{el}}{m_{el}} . \frac{\rho V_{hf}^2}{2g} \cos\vartheta_2\right)$,

or,

(8) $\vartheta_2 = -arc\tan\left(\frac{c_{xa\,el} S_{el}}{m_{el}} . \frac{\rho V_{hf}^2}{2g}\right)$.

If we put the expression (5) for the ballistic coefficient in formula (8), we will get:

(9) $\vartheta_2 = -arc\tan\left(c_a . \frac{\rho V_{hf}^2}{2g}\right)$,

where $c_a = \frac{c_{x\,el} S_{el}}{m_{el}}$ is the ballistic coefficient of the external fire extinguishing device in the velocity coordinate system.

3. Analysis of the results obtained

For external devices with a spherical shape, such as the shape of "BB4453", attached to a helicopter in flight and taking into account the fact that $\vartheta_1 = \vartheta_2$, the dependences of the pitch angle value of the central cable ($|\vartheta_1|$) in an established straight horizontal flight on the flight speed V_{hf} at different values of the ballistic coefficient (c_a) have been constructed, as presented in Fig. 2.

From the point of view of applicability in aviation practice, these dependencies are applicable only for first-approximation calculations for external devices that have a shape close to a sphere. At the same time, it should be noted that if the shape of the device differs significantly from a sphere, then when flowing with the oncoming air, a lifting force $Y_{a\,el}$ can be formed, which can have a positive or negative value, and which will affect the value of the equilibrium pitch angle of the external body.

In this case, to determine the equilibrium pitch angle of the external fire extinguishing device (or of the central cable, if $\vartheta_1 = \vartheta_2$) it is necessary to know the dependence of the lift coefficient of the "BB4453", or its aerodynamic quality $K_{el} = c_{ya\,el}/c_{xa\,el}$ on the angle of attack. Then the equilibrium pitch angle of the external fire extinguishing device can be determined by the formula:

(10) $\vartheta_2 = -arc\tan\left(\frac{1}{\frac{2g}{c_a \rho V_{hf}^2} - K_{el}}\right)$.

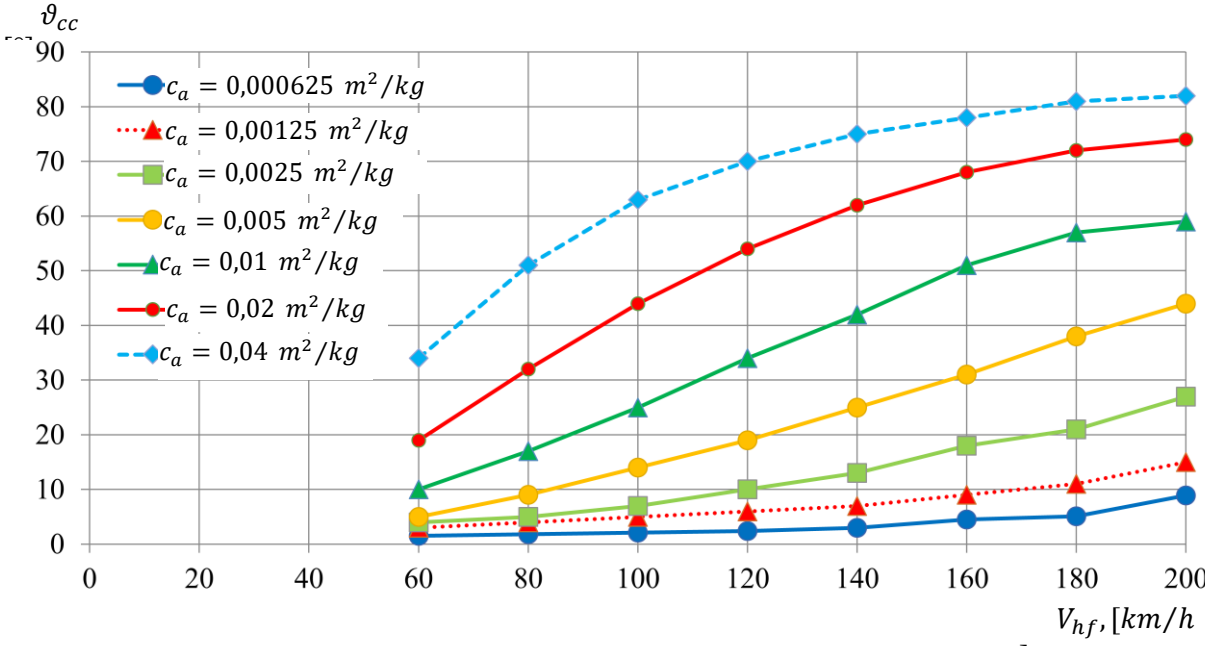


Fig. 2. Value of the pitch angle of the central cable of an external device with a spherical shape, depending on the speed of an established horizontal straight flight and on its ballistic coefficient

4. Conclusion

In conclusion, it can be concluded that the ballistic coefficient of the external fire extinguishing device *c* represents its complex characteristic and fully determines the equilibrium position of the "BB4453" and the other elements of the external device attachment system when performing an established horizontal straight flight at a given height and with a given helicopter flight speed. The greater the ballistic coefficient of the external device *c*, the greater in absolute value the pitch angle of the "BB4453" and the angle of deviation of the central cable of the device from its vertical position. When using the ballistic coefficient c_a, the aerodynamic quality of the external device must be taken into account, if $K_{el} \neq 0$.

References

1. SEI Industries LTD. | Engineered Solutions. www.sei-ind.com /product-families.
2. Efimov V.V. The dynamic stability of the cargo of the cable external suspension of the helicopter. All-Russian Scientific and Technical Journal "Flight". 2011. No. 3. p. 26–32 (in in Russian language).
3. Cicolani, L. S., G. Kanning. Equations of Slung-Load Systems, Including Multilift Systems. NASA Technical Paper 3280. National Aeronautics and Space Administration. Office of Management. Scientific and Technical Information Program. 1992. 132 p.
4. Stuckey, R. A. Mathematical Modeling of Helicopter Slung-Load Systems. Department of Defence. Defence Science and Technology Organisation DSTO-TR-1257. Fishermans Bend. Victoria. Australia 3207. 2001. 113 p.
5. Kozlovsky, V. B. Helicopter with cargo on external suspension/B. B. Kozlovsky, S. A. Parshentsev, V. C. Efimov; in order. C. B. Kozlovsky. - M.: Mechanical engineering. 2008. 304 p. (in Russian language).
6. Dmitrievsky, A. A. External ballistics: tutorial. for university students/A. A. Dmitrievsky, L. N.Lysenko - Ed. 4th, reworked. And add. - M.: Mashinostroenie, 2005. 608 p. (in Russian language).

Изследване на равновесието на външно пожарогасително устройство от типа "Bambi Bucket BB4453"в полет на хеликоптер AS532AL Cougar

Николай Загорски

Резюме: В настоящия материал са изследвани и представени причините за влияние на параметрите на външното пожарогасително устройство от типа „Bambi Bucket BB4453" на неговото равновесие при полет на хеликоптера AS532AL Cougar за гасене на пожари.

Modeling and Linearization of a Hybrid Energy System with Two-Stroke Internal Combustion Engine and Lithium-Polymer Battery

Georgi Georgiev

Bulgarian Air Force Academy, Faculty of Aviation, goro78@gmail.com

Abstract: The primary limiting factor for multirotor unmanned aerial vehicles (UAVs) powered solely by electrochemical batteries is the limited amount of available onboard energy. Typical flight duration ranges between 15-30 minutes, rarely exceeding 90 minutes even when using high-efficiency battery systems. In this context, hybrid energy systems combining two or more energy sources are considered a promising alternative for extending autonomous flight time without compromising the advantages of battery electric propulsion.
This study presents the development and evaluation of a hybrid power unit for multirotor UAVs, consisting of an internal combustion engine (ICE) driving an electrical generator (EG), connected to a common DC bus in parallel with a lithium-polymer (LiPo) battery. Modeling, simulation, and identification of the individual subsystems comprising the energy module have been performed.
The results demonstrate that the control system ensures efficient and synchronized utilization of both energy sources. The analysis shows that the hybrid configuration achieves up to a two-fold increase in energy density, significantly improving the flight autonomy of multirotor UAVs.

Keywords: *multirotor UAV; hybrid power unit; LiPo battery; internal combustion engine; electrical generator*

1. Introduction

Advances in electronics, materials science, and manufacturing technologies have enabled the development of compact and powerful electric motors, high-efficiency electrochemical batteries, integrated sensor arrays, and multifunctional flight control systems. This has facilitated the widespread adoption of multirotor unmanned aerial vehicles (UAVs) across various sectors. Despite these technological achievements, a key limitation of this class of aircraft remains their restricted flight autonomy and operational range, which are directly dependent on takeoff mass (including airframe and payload) as well as the capacity and energy density of onboard batteries. Battery-powered, fully electric multirotor UAVs currently dominate the market. They typically provide flight durations between 15 and 30 minutes, with rare exceptions reaching up to 90 minutes of autonomy when using high-performance batteries.

The present study is motivated by the need to overcome the limitations of systems relying exclusively on electrochemical batteries. The proposed solution involves a hybrid propulsion system comprising an internal combustion engine (ICE) driving an electrical generator (EG), operating in tandem with a lithium-polymer (LiPo) battery. This configuration forms a hybrid power unit suitable for integration into multirotor UAVs.

The specific energy density of gasoline (approximately 12 kWh/kg) is roughly two orders of magnitude higher than that of modern LiPo batteries (approximately 0.2–0.5 kWh/kg) [1]. Consequently, hybrid energy systems have the potential to substantially extend the flight duration and operational range of multirotor UAVs. Hybrid propulsion systems are well-established in automotive and rail transport [2,3]. Unlike ground vehicles, where motion is constrained to two dimensions and gravitational limitations are less pronounced, the integration of hybrid systems in aircraft presents significantly more complex engineering challenges. This provides additional motivation for investigating hybrid configurations in aerial platforms.

The autonomy of multirotor UAVs is the subject of intensive research aimed at identifying optimal energy system configurations that meet requirements for low weight, high energy density, and efficient power management. Various approaches have been documented in the literature, primarily focusing on hybrid systems for fixed-wing, vertical takeoff and landing (VTOL) aircraft [4,5], as well as alternative platforms such as airships [6,7]. Multidisciplinary optimization applied to different UAV classes demonstrates that hybrid-electric configurations can significantly improve flight performance [8].

Through simulation and bench testing, researchers can predict the behavior of various energy sources and evaluate the feasibility of hybrid power modules before conducting flight tests [9].

2.Hybrid Power System

2.1. Overview

The design of the power system for unmanned aerial vehicles (UAVs) requires a balance between two key aspects: ensuring a reliable energy source for extended autonomous operation and maintaining a high dynamic response of the system under rapidly changing loads. Purely electric systems, typically used in multirotor UAVs, offer excellent dynamic performance due to the use of electric motors with precise thrust control, but they suffer from limited flight duration due to the low energy density of batteries. In this context, the considered hybrid power system combines the advantages of an internal combustion engine (ICE) and electric propulsion.

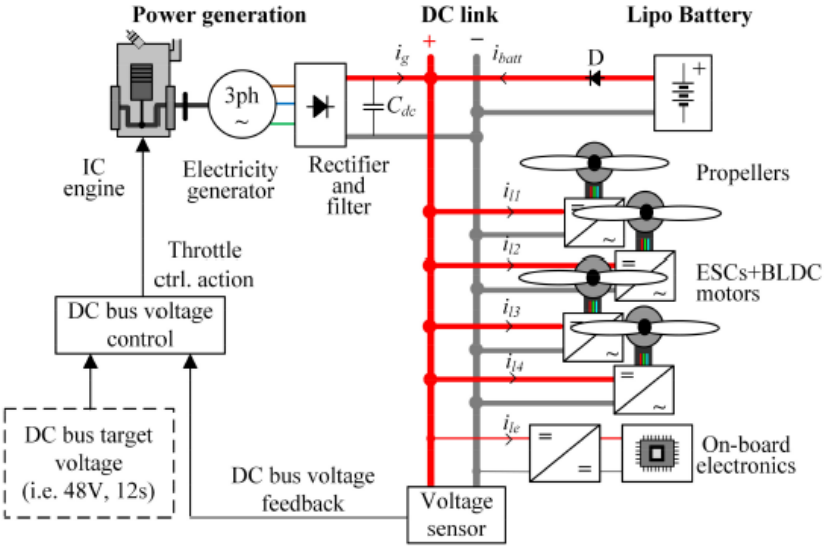


Figure 1. Considered hybrid power unit topology.

As illustrated in Figure 1, the internal combustion engine (ICE) is mechanically connected to a three-phase electric generator, whose energy is converted through a full-wave rectifier and filter and then injected into a common DC link. In this way, the ICE-EG unit provides the base power supply during quasi-stationary flight modes, such as hovering or smooth maneuvers.The torque control of the ICE, and thus the power output, is achieved through a servo-controlled throttle mechanism regulated by a DC link voltage controller. The controller uses feedback from the measured voltage and a predefined reference voltage (e.g., 48V) to maintain voltage stability on the DC link by adjusting the throttle valve. This ensures efficient energy transfer from the ICE to the UAV's electrical power system.

A LiPo battery is connected in parallel to the DC link via a blocking diode, which prevents uncontrolled back-charging. The battery acts as a high-dynamic buffer source, capable of compensating for sudden peak loads—such as during aggressive maneuvers or takeoff—by supplying current to the DC link within milliseconds. Typical discharge currents can reach up to 50 A, enabling immediate voltage stabilization despite the inertial delay of the ICE-EG unit.

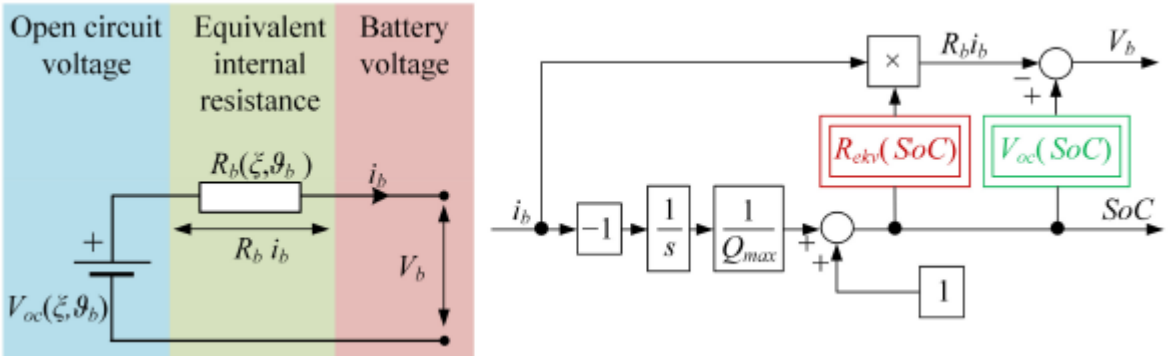
The energy from the DC link is distributed to the individual propeller drives, composed of electronic speed controllers (ESCs) and brushless DC motors with permanent magnets (BLDC), as well as to the onboard electronics. Voltage stability and fast load response are critical for effective thrust control and overall flight dynamics.

This hybrid-powered architecture, based on a centralized DC link, is fully applicable to multirotor UAVs with four or six propellers. By combining the high energy density of fuel with the instantaneous

responsiveness of the battery, the system enables long-duration missions with high maneuverability and resilience to transient loads.

2.2. Battery Model

The battery modeling approach used in this work is tailored to the need for sufficient accuracy during transient simulations while maintaining low computational complexity. A quasi-static Thevenin-type model [10,11] has been selected, as it provides a good approximation of the terminal voltage and internal losses during short-term but frequent load variations—typical for the battery's role as a power buffer in a hybrid energy system (see Figure 2a).



(a) (b)

Figure 2. LiPo Battery modeling: (a) Quasi-static Thevenin model; (b) battery model block diagram realization.

The model includes:

- An ideal voltage source V_{oc}, representing the open-circuit voltage, which depends on the state of charge (SoC), state of health (SoH), and battery temperature θ_b;
- An equivalent internal resistance R_b, which is also a function of the above variables;
- The terminal voltage V_b, which is measured at the battery output.

The equation describing the battery voltage under a given current i_b is:

$$V_b = V_{\text{oc}}(\xi, \theta_b) - R_b(\xi, \theta_b) \cdot i_b \tag{1}$$

Here, the current i_b is positive during discharge and negative during charge.

The battery's State of Charge (SoC, denoted as ξ) is determined by integrating the discharged charge ΔQ_b relative to the maximum capacity Q_{max}, as follows:

$$\xi = 1 - \frac{\Delta Q_b}{Q_{\max}}, \quad \Delta Q_b = -\int i_b\, dt \tag{2}$$

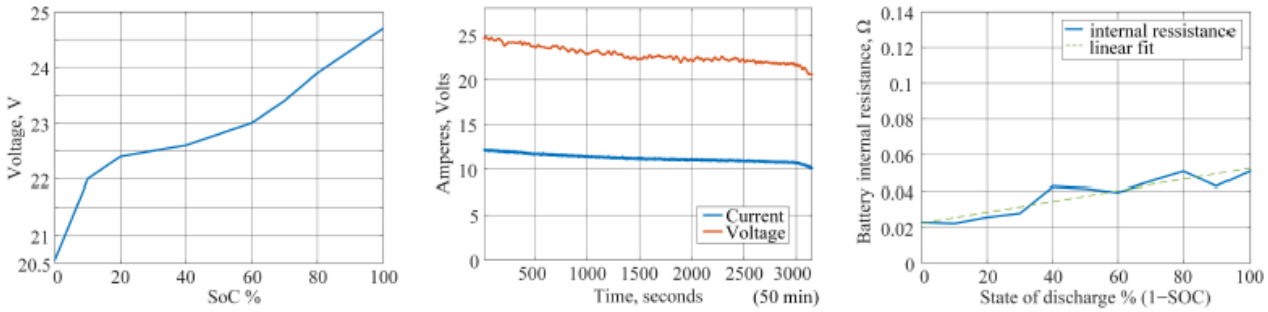
This definition is implemented in the block diagram representation of the model shown in Figure 2b, where:

- The current SoC is calculated through integration and normalization by Q_{max}
- The SoC serves as input to nonlinear functions for V_{oc} and R_b
- The terminal voltage V_b is computed at the output using equation (1)

Polarization Effects and Model Validity:

Although the Thevenin model is quasi-static and doesn't account for electrochemical polarization dynamics (characterized by primary voltage response delays), this simplification is justified by the battery's operational profile involving short-duration pulse discharges for peak load coverage. In such regimes, the voltage drop caused by internal resistance dominates over polarization effects, which exhibit time constants in the range of tens of seconds and become significant only during prolonged

discharges. Consequently, the adopted model remains suitable for simulating peak load conditions and maintains good compatibility with the overall dynamics of the hybrid system.



(a) (b) (c)

Figure 3. LiPo Battery identification: (a)OCVvs. SoC; (b)Continuous discharge curve example (2 ohmload); (c) Experimentally–recorded and extracted internal resistance vs. state of discharge (1−SOC), for discharging operation.

Experimental Identification

The experimental characterization procedure involves two main phases, with results presented in Figure 3:

1.OCV-SoC Curve Test

- The battery is fully charged (to 4.15V/cell) and left undisturbed for approximately 3 hours to stabilize
- Sequential discharge pulses (10-minute duration) are applied, each followed by a rest period
- The resulting open-circuit voltage versus SoC relationship (Figure 3a) shows strong nonlinearity, particularly at charge extremes - characteristic behavior of LiPo batteries

2.Internal Resistance (Rb) Test

- Conducted under continuous, smooth discharge conditions (Figure 3b) to minimize capacitive effects
- Instantaneous internal resistance $R_b=\Delta V/\Delta I$ is calculated from time-domain voltage/current measurements at various SoC levels
- Results (Figure 3c) demonstrate R_b's gradual increase during deeper discharge (decreasing SoC), approximated by a linear function for simulation purposes

Figure 3 shows the identification results obtained at constant ambient temperature (30°C). These results reveal:

- The strongly nonlinear OCV versus SoC relationship shown in Figure 3a
- The mild increasing trend of internal resistance versus discharge state (1-SoC) shown in Figure 3c
- Derived from combined voltage/current responses in Figure 3b and OCV versus SoC during the discharge test

Figure 3 shows the identification results obtained at constant ambient temperature (30°C). These results indicate a strongly nonlinear OCV versus SoC dependence as shown in Figure 3a, and a relatively mild increasing trend of the battery's internal resistance with battery discharge state (1-SoC) as shown in Figure 3c. The latter result is obtained by combining the battery voltage and current responses from Figure 3b with the battery OCV versus SoC during the time period when the discharge test in Figure 3b was performed.

2.3. Internal Combustion Engine (ICE) Model

The modeling of the two-stroke internal combustion engine is carried out using the so-called Mean Value Engine Model (MVEM) [12,13]. This approach captures the essential static and dynamic characteristics of the engine without delving into the high-frequency (fast) cyclic dynamics, which would require a much more complex and computationally intensive simulation. The model is intended for simulation studies and control system design.

In the considered case, the engine is a two-stroke type with a small intake manifold volume. This implies that the dynamics of air filling in the manifold are extremely fast and can therefore be neglected. This simplifying assumption allows the use of MVEM as a suitable modeling basis and also enables further reduction of the model to a first-order system, particularly when the goal is low-bandwidth control.

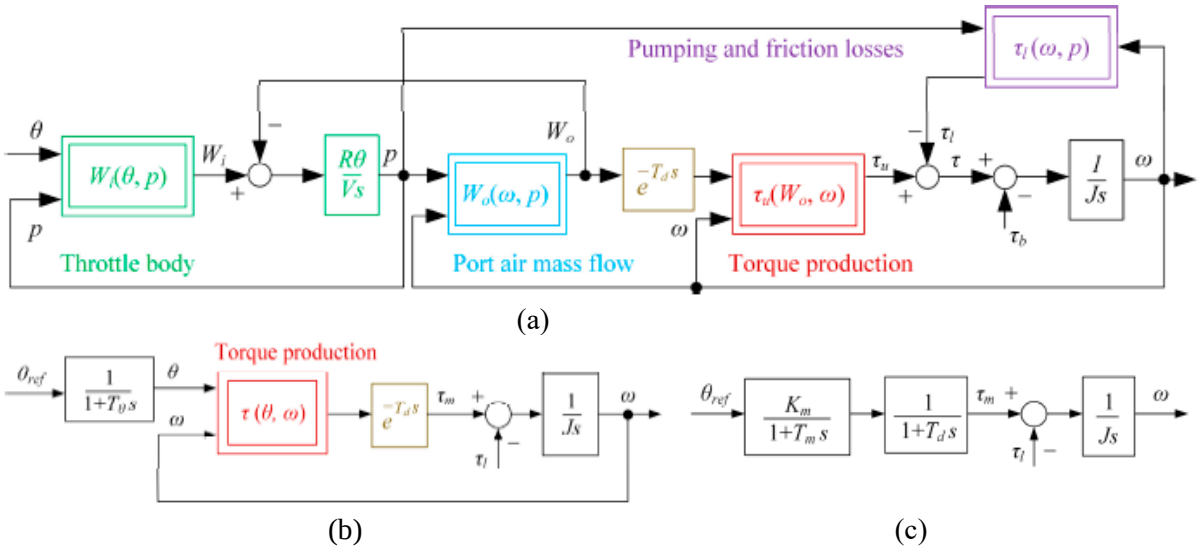


Figure 4. ICE modeling: (**a**) Non-linear mean value model of ICE; (**b**) simplified non-linear model of ICE; (**c**) model linearized in the vicinity of ICE operating point.

The main block diagram of the complete nonlinear MVEM model is shown in Figure 4a. It includes the individual engine components: throttle valve, air flow, torque generation, as well as friction and pumping losses. The model features two primary state variables: the intake manifold pressure p and the engine angular speed ω. All other dependencies—including those describing the torque—are derived from static (typically three-dimensional) maps based on experimental data.

The next step in the modeling process involves simplifying the MVEM by neglecting the intake manifold dynamics. This leads to a first-order model (Figure 4b), in which only one state variable remains—the angular speed ω. In this simplified model, throttle dynamics are retained and represented as a first-order linear process with a time constant, while the torque t_M is modeled as a static function of throttle position and speed: $t_M=f(\theta,\omega)$ where θ is the throttle opening angle.

Additionally, the model incorporates a so-called torque development dead time T_d. This delay reflects the physical limitation that prevents instantaneous changes in torque in response to variations in input parameters. The delay is approximated using the first two terms of the Taylor expansion of an exponential function, assuming that the closed-loop bandwidth of the control system is significantly lower than the internal response bandwidth of the engine. In the case of a two-stroke engine, where combustion occurs on every crankshaft revolution, the dead time can be roughly estimated using the following formula:

$$T_d \approx \frac{60}{n} \tag{3}$$

where n is the engine speed in revolutions per minute (RPM).

The resulting simplified model includes the key elements required for simulation and control: throttle dynamics, a static torque model, and dead time. For the purposes of control system design, this model is linearized around a specific operating point—typically the point of maximum torque for a given engine speed and throttle opening. In the case described in the document, this operating point corresponds to a throttle opening of approximately 70% and an engine speed of around 9500 RPM.

The result of the linearization is shown in Figure 4c. It represents a classic first-order linear model with delay, where the system dynamics are described using transfer functions of the form:

$$\frac{K_m}{(1+T_m s)(1+T_d s)} \tag{4}$$

where K_m is the gain coefficient, T_m is the engine time constant, and T_d is the torque development delay. This entire model is used for stability analysis and the design of appropriate controllers—such as PID or LQR

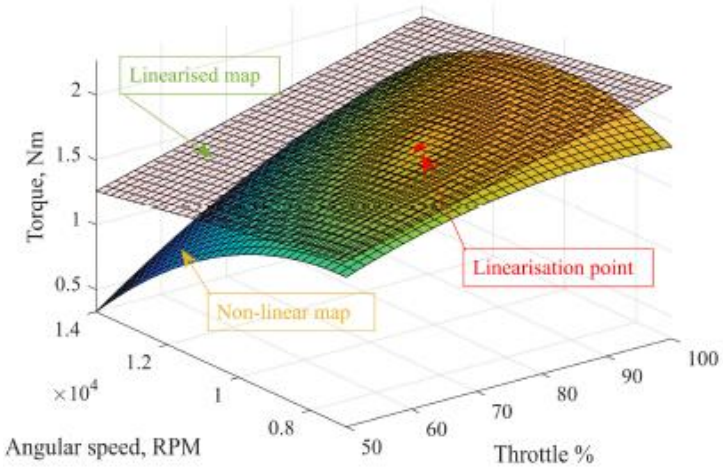


Figure 5. Torque map linearized near the ICE operating point

An additional important part of the analysis is the three-dimensional torque–throttle–speed map shown in Figure 5b. This map, obtained either experimentally or through simulation, illustrates how the engine torque varies depending on throttle opening and engine speed. The linearization is performed precisely at the peak point of this surface, which corresponds to the engine's optimal operating condition—aimed at maximizing fuel efficiency and minimizing losses. It is at this point that the generator connected to the engine produces maximum current, which is crucial in hybrid propulsion systems.

2.4. Model of the Electric Generator and Rectifier

The model of the electric generator and rectifier describes a brushless permanent magnet synchronous generator (BPMS), which operates based on the principle of inducing electromotive force (EMF) in the three stator phases as the rotor rotates[14,15].

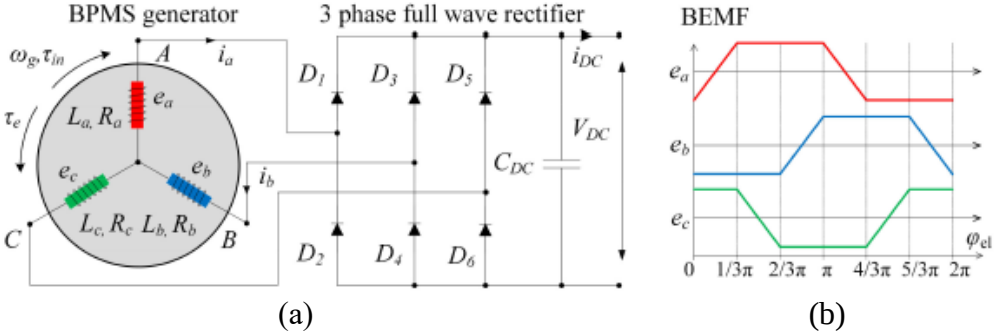


Figure 6. BMPS: (a) electrical circuit of the three-phase BPMS EG and its connection to the three-phase full-wave diode rectifier; (b) trapezoidal back-emf waveforms of individual EG phases.

The generator consists of a rotor with permanent magnets, a stator with windings, optionally integrated Hall sensors for rotor angle detection, and electrical phase outputs. Depending on the shape of the induced voltage (back electromotive force or BEMF), the machine may be of the trapezoidal BEMF type (commonly known as BLDC—Brushless DC Motor) or the sinusoidal BEMF type (commonly referred to as PMSM—Permanent Magnet Synchronous Motor). These two types differ in magnet geometry, air gap configuration, winding distribution, and magnetization profile.

Electrically, each phase of the BPMS generator has an internal resistance R_{ph} and inductance L_{ph}. The induced voltage in a given phase e_l depends on the rotor speed ω_g, the magnetic flux ψ_m, and the mechanical rotor position α_g, and is calculated using the following formula:

$$e_l = K_e \omega_g \phi_m \left[\theta_g - \frac{2\pi(l-1)}{3}\right] \tag{5}$$

where l=1,2,3 for the three phases, and p is the number of pole pairs. The electromag τg generated by each phase with current i_l is given by:

$$\tau_g = K_e \sum_{l=1}^{3} \phi_m \left[\theta_g - \frac{2\pi(l-1)}{3}\right] i_l \tag{6}$$

When the machine is connected to a three-phase diode rectifier (as shown in the figure), during each commutation, two of the three phases conduct current (if their voltage exceeds the bus voltage plus the diode drops), while the third remains non-conductive. In this case, it can be assumed that:

$$i_1 = -i_2 = i_{eq} \tag{7}$$

- i_1: Current in phase 1
- i_2: Current in phase 2
- i_{eq}: Equivalent current representing the combined behavior

The negative sign indicates opposite current directions in the two phases.

$$L_{eq} = 2L_{ph}, \quad R_{eq} = 2R_{ph} + 2r_d \tag{8}$$

- L_{eq}: Equivalent inductance [H]
- L_{ph}: Phase inductance [H]
- R_{eq}: Equivalent resistance [Ω]
- R_{ph}: Phase resistance [Ω]
- r_d: Dynamic resistance of the diode [Ω]

The equation describes the equivalent parameters of a three-phase generator (BPMS) connected to a full-wave diode rectifier, assuming that two phases conduct simultaneously. It is used for:

- Simplifying system modeling
- Power loss analysis
- Designing control algorithms

(9) $$e_{eq} = K_{eq} \cdot \omega_g = 2K_e \cdot \omega_g$$

- e_{eq}: Equivalent electromotive force (EMF) [V]
- $K_{eq}=2K_e$: Equivalent voltage constant [V·s/rad]
- ωg: Rotor angular speed [rad/s]

(10) $$\tau_g = K_{eq} \cdot i_{eq} = 2K_e \cdot i_{eq}$$

- τ_g: Electromagnetic torque [N·m]
- i_{eq}: Equivalent current [A]

Doubling the constant K_e reflects the fact that two generator phases are conducting at any given time. These formulas simplify system analysis by reducing the three-phase behavior to an equivalent DC model.

In the experimental setup, the motor used is the MTO6384–170–HA–C, which is a lightweight, external-rotor type with a back-EMF constant of 170 KV, meaning 170 revolutions per minute per 1 V of induced voltage. This implies that for every 1 V increase in the induced voltage, the speed increases by 170 rpm.

- Its main specifications are:
- **Maximum current:** 65 A, **Rated current:** 60 A
- **Power supply:** 2 to 12 LiPo cells
- **Phase-to-phase resistance:** 0.04068 Ω
- **Phase-to-phase inductance:** 40.43 μH
- **Maximum power output:** 3550 W
- **Weight:** 830 g

3. Conclusions

The present study demonstrates the successful modeling, simulation, and integration of a hybrid energy system for multirotor unmanned aerial vehicles (UAVs), combining a two-stroke internal combustion engine (ICE) and a LiPo (lithium-polymer) battery. The main scientific and practical contributions can be summarized as follows:

The hybrid configuration achieves up to a 2x increase in energy density compared to purely battery-powered systems, thanks to the high specific energy of gasoline (~12 kWh/kg). This significantly extends the autonomous flight time while preserving the benefits of electric propulsion such as precise control and fast dynamic response.

The study proves that hybrid systems combining ICE and LiPo batteries are a viable solution for extending the endurance of UAVs without compromising their dynamic performance.

References

1. Fischer, M.; Werber, M.; Schwartz, P.V. Batteries: Higher energy density than gasoline? *Energy Policy* 2009, *37*, 2639–2641.

2. Cipek, M.; Pavkovi´c, D.; Petri´c, J. A control-oriented simulation model of a power-split hybrid electric vehicle. *Appl. Energy* 2013,

101, 121–133.

3. Cipek, M.; Pavkovi´c, D.; Kljai´c, Z.; Mlinari´c, T. Assessment of battery-hybrid diesel-electric locomotive fuel savings and emission

reduction potentials based on a realistic mountainous rail route. *Energy* 2019, *173*, 1154–1171.

4. Abdul Sathar Eqbal, M.; Fernando, N.; Marino, M.;Wild, G. Hybrid propulsion systems for remotely piloted aircraft systems.

Aerospace 2018, *5*, 34.

5. Friedrich, C.; Robertson, P.A. Hybrid-electric propulsion for aircraft. *J. Aircr.* 2015, *52*, 176–189.

6. Recoskie, S.; Fahim, A.; Gueaieb, W.; Lanteigne, E. Hybrid power plant design for a long-range dirigible UAV. *IEEE Trans.*
Mechatron. 2013, *19*, 606–614.
7. Recoskie, S.; Fahim, A.; Gueaieb, W.; Lanteigne, E. Experimental testing of a hybrid power plant for a dirigible UAV. *J. Intell.*
Robot. Syst. 2013, *69*, 69–81.
8. Donateo, T.; De Pascalis, C.L.; Ficarella, A. Synergy effects in electric and hybrid electric aircraft. *Aerospace* 2019, *6*, 32.
9. Depcik, C.; Cassady, T.; Collicott, B.; Burugupally, S.P.; Li, X.; Alam, S.S.; Hobeck, J. Comparison of lithium ion Batteries, hydrogen
fueled combustion Engines, and a hydrogen fuel cell in powering a small Unmanned Aerial Vehicle. *Energy Convers. Manag.* 2020,*207*, 112514.
10. He, H.; Xiong, R.; Fan, J. Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an
experimental approach. *Energies* 2011, *4*, 582–598.
11. Pavkovi´c, D.; Krznar, M.; Komljenovi´c, A.; Hrgeti´c, M.; Zorc, D. Dual EKF-based state and parameter estimator for a LiFePo4
battery cell. *J. Power Electron.* 2017, *17*, 398–410.
12. Deur, J.; Ivanovi´c, V.; Pavkovi´c, D.; Jansz, M. Identification and speed control of SI engine for idle operating mode. *Sae Tech. Pap.*
2004.
13. Rajamani, R. *Mean Value Modeling of SI and Diesel Engines*; Springer: Boston, MA, USA, 2012.
14. Mevey, J.R. Sensorless Field-Rriented Control of Brushless Permanent Magnet Synchronous Motors. Master's Thesis, Kansas
State University, Manhattan, KS, USA, 2009.
15. Hanselman, D.C. *Brushless Permanent Magnet Motor Design*; TheWriters' Collective: Cranston, RI, USA, 2003.

Моделиране и линеаризация на хибридна енергийна система с двутактов двигател с вътрешно горене и литиево-полимерна батерия

Георги Георгиев

Резюме: Основният ограничаващ фактор за многороторните безпилотни летателни апарати (БЛА), захранвани единствено от електрохимични батерии, е ограниченото количество налична енергия на борда. Типичната продължителност на полета варира между 15-30 минути, като рядко надвишава 90 минути, дори при използване на високоефективни батерийни системи. В този контекст хибридните енергийни системи, комбиниращи два или повече източника на енергия, се считат за обещаваща алтернатива за удължаване на времето за автономно полета, без да се компрометират предимствата на електрическото задвижване с батерии. Това изследване представя разработването и оценката на хибриден силов агрегат за многороторни БЛА, състоящ се от двигател с вътрешно горене (ДВГ), задвижващ електрически генератор (ЕГ), свързан към обща постояннотокова шина паралелно с литиево-полимерна (LiPo) батерия. Извършено е моделиране, симулация и идентифициране на отделните подсистеми, съставляващи енергийния модул.

Резултатите показват, че системата за управление осигурява ефективно и синхронизирано използване на двата енергийни източника. Анализът показва, че хибридната конфигурация постига до двукратно увеличение на енергийната плътност, което значително подобрява автономността на полета на многороторните безпилотни летателни апарати.

TRANSITION TO CLIMATE-SUSTAINABLE AVIATION TRANSPORT

Ivan Benkov

Georgi Benkovski Air Force Academy, 1 Cyril and Methodius St, Dolna Mitropolia, Republic of Bulgaria
email: ibenkov@af-acad.bg , phone: + 359 87 881 5199

Abstract

The aviation industry, which is among the most dynamic industries, generating economic growth, employment and facilitating trade, tourism, connectivity between people from different geographical regions, exerts significant environmental pressure on the environment and the organismic world. The strongest environmental impact is caused by the residues of fuel combustion in aircraft engines - carbon dioxide, nitrogen oxides, sulfur dioxide, carbon monoxide, water vapor. Of these, the largest share has the greenhouse gas - CO_2.

The aviation sub-sector is among the first to embark on the path of sustainable development. The transition to climate-sustainable air transport is a very difficult, complex, science-intensive and capital-intensive process. The main commitments of the aviation industry to reduce its impact on the climate are aimed at technological innovations in aircraft, the introduction of sustainable aviation fuels, operational improvements to flights, increasing the efficiency of the infrastructure, and the implementation of market and regulatory mechanisms.

Keywords: *aviation transport, transition, sustainable development, greenhouse gas , climate change , sustainable aviation fuel*

I. INTRODUCTION

Sustainability is a leading concept in the economic, environmental, social and management theory and practice of modern society. It is aimed at stimulating the transformation of unsustainable economic models into alternative ones. Sustainable development requires rational and efficient use of resources to meet the needs of both present and future generations, without disrupting the stability of ecosystems.

The aviation industry, which is among the most dynamic industries today, is subject to numerous studies and analyses in terms of its contribution to economic growth, employment, trade, tourism and connectivity between people. In parallel with the positives, air transport has also been studied in terms of the significant environmental pressure it exerts on the environment and the organismic world. A comprehensive assessment of the negative contribution of aviation to global atmospheric problems is contained in the Special Report on Aviation and the Global Atmosphere, which was prepared at the request of International Civil Aviation Organization (ICAO) by the Intergovernmental Panel on Climate Change (IPCC) in cooperation with the Scientific Assessment Panel to the Montreal Protocol on Substances that Deplete the Ozone Layer and was published in 1999 [1].

The most significant and large-scale environmental impact of aviation over the past two decades is related to climate change. Emissions from the combustion of fuels in aircraft engines contribute to the anthropogenic greenhouse effect. Climate impacts include carbon dioxide (CO_2), nitrogen oxides (NOx), aerosols and their precursors (sulfates and soot), water vapor, persistent linear inversion trails, and the generated cirrus clouds. All of them are included in the list of air pollutants and greenhouse gases established by the Kyoto Protocol (1997). The greenhouse gas with the largest share of the radiative impact on the climate is CO_2. The term

"radiative forcing" (RF) refers to the disturbed energy balance between the Earth and the atmospheric air in the period after the Industrial Revolution (since 1750), caused by the changed content of gases and particles in the atmosphere, measured in units of W/m^2.

Depending on the effect they cause, these components can be divided into two groups. The majority of them have a positive radiative effect: CO_2, water vapor, persistent inversion trails, cirrus clouds, soot. Only the effect of the emission of sulfate ions has an opposite sign (-RF). The total final effect of NOx is positive, but this factor has a three-dimensional meaning. Under its influence, tropospheric ozone O_3 is synthesized (positive RF), there is a long-term reduction of methane CH_4 (negative RF) and an additional minimal reduction of ozone O_3 (negative RF) [2].

Due to its specificity, the aviation sector disperses the emitted emissions mainly in the high layers of the atmosphere (at an altitude of 8-12 km, and supersonic aviation up to 15-20 km), which induces chemical and physical processes with climatic consequences - formation of tropospheric ozone and condensation trails, formation of cumulus clouds, etc. Despite the fact that at present, the CO_2 emitted by the aviation industry has a relatively small contribution to climate change (just over 2%), in the coming decades it will increase, in parallel with the growth of the industry and by 2050 it may triple. With 13.9% of greenhouse gas emissions generated, aviation ranks second in the transport sector after road transport [3].

Achieving climate neutrality is a long-term strategic goal for aviation until 2050. Revolutionary changes are about to take place during this limited period.

This report explores the first steps on the path to transitioning to climate sustainability in the aviation industry.

II. SMOOTH TRANSITION TO CLIMATE SUSTAINABILITY OF AVIATION TRANSPORT

The aviation industry is among the first to take the path of sustainable development. The transition to climate-sustainable aviation is a complex, long-term, knowledge-intensive and capital-intensive process. In October 2021, a decisive step was taken towards achieving zero net carbon emissions by 2050 [5]. Accomplishing this goal is part of the implementation of the global goal of the Paris Agreement to limit global warming to $1.5\ ^0C$ compared to pre-industrial levels. These commitments were confirmed in October 2022 during the 41st Assembly of the ICAO.

The aviation industry's responsibilities to reduce its climate impact can be achieved through a package of comprehensive measures - technological innovations in aircraft; implementation of sustainable aviation fuels; operational improvements to flights; increasing the efficiency of infrastructure, regulatory and market mechanisms [4].

II.1. Technological innovations for greener aircraft

Continuous innovation and improved technology over the last 40-50 years have led to remarkable results in the aviation sector. With the introduction of jet engines, fuel consumption and emissions from aircraft have been reduced by up to 70% [6] and noise by 75%. Europe's ambitious targets to accelerate the trend of reducing CO_2 emissions per passenger/kilometer from aviation by 75% by 2050 compared to the 2000 reference year are too high [7]. This revolutionary change must be achieved in parallel with a 90% reduction in NOx and a 65% reduction in perceived noise [8].

Fleet renewal with greener aircraft is at the heart of the strategy to achieve sustainability goals in aviation. Replacing old generations of inefficient aircraft with technologically innovative models will help airlines move closer to their zero-carbon footprint goals. As already noted, for a large part of the modern aircraft models in operation, fuel costs, noise and emissions are significantly reduced compared to their predecessors. For example, the new Airbus

A320neo emits 50% less nitrogen oxides (NOx), achieves a 75% reduction in noise and a 16% reduction in fuel consumption compared to the previous generation A320 [9]. The new generation aircraft feature advanced aerodynamics, lightweight materials and more efficient engines. These improvements reduce fuel consumption and greenhouse gas emissions. Improved fuel efficiency is a key factor in reducing operating costs, increasing economic benefits (increasing flight range and payload) and reducing environmental impact.

It should be noted that, despite the economic, social and environmental benefits, innovations cannot be implemented everywhere and in a short time. The reasons for this are well known. The average operational life of most aircraft is between 20 and 30 years. Research and development in the industry lasts for years, requires serious financial resources (billions of dollars), and the implementation of rationalizations goes through stages of testing, evaluation and certification in order to reach their real application. The development of a conventional commercial aircraft lasts approximately six years and involves the design of up to 6 million parts [10].

The development of the "ecodesign" of aircraft is aimed at minimizing their impact on the environment. During the design phase, significant parameters are assessed: ecodesign of individual components and details, use of appropriate materials and production technologies, assembly method, maintainability, operational characteristics, recycling of waste and materials obtained after scrapping the aircraft (end-of-life). The design and selection of materials for each detail is carried out based on an assessment of their environmental impact throughout their life cycle, as well as their capacity of secondary use (as spare parts) and recyclability after the end of their useful life. The aspiration of aviation designers is to reduce the proportion of non-renewable materials used. The production of these complex and expensive aircraft is the result of the activities of numerous manufacturing companies and to achieve the desired environmental result, commitment to the cause and coordination of the efforts of all participants are necessary requirements.

The main goals of technological innovations are related to improving fuel efficiency by reducing the net weight of aircraft, the fuel consumption and increasing the size of the payload as well as the length of the flight. Reducing fuel consumption directly leads to a reduction in carbon emissions. Reducing the fuel used by one ton saves nature from the harmful impact of 3.16 tons of CO_2 emissions. The main requirements for the structural materials used in modern aircraft are a high degree of tensile and shear strength, hardness and ductility, durability, wear resistance, fire resistance and others. The high quality standards applied to metal alloys, composite materials, fuels, and special fluids are a guarantee of security, safety, economy and environmental friendliness.

A wide range of composite materials is used in the construction of modern aircraft (Airbus A350, Boeing 787 Dreamliner and others). They effectively replace the classic aluminum alloys, titanium and others. Synthetic polymers, reinforced with glass fibers (GFRP) and sandwich materials with phenolic resins as a matrix system find application in the interior due to their low weight-to-stiffness ratio [11]. With limited application at the moment, but very promising in the future, are composites of renewable materials based on epoxy (bio resins) cross-linked with natural fibers (flax, hemp and Ramie fibers). The use of composite materials also leads to a lightening of the engines by about 500 kg., if the fan blades and the entire engine housing are made of them. The American giant "General Electric Aerospace" (GE - Aerospace) successfully introduced the GE-90 jet engine for wide-body aircraft with composite fan blades that are twice as strong and one-third lighter than traditional titanium blades [12].

Future design intentions to modify conventional aircraft design, such as the blended wing and body (BWB) design, are radically different. The expectation is that this revolutionary design will increase aerodynamic efficiency by minimizing all non-lift surfaces on the aircraft, and have significant potential for improvement in fuel efficiency.

Technological innovations are part of a continuous process of reducing emissions from aircraft engines. Today's aircraft engines are the result of a century of evolution, having transformed them into more powerful, more efficient and more reliable versions of their predecessors. A measure of the fuel efficiency of aircraft is the fuel/ passenger kilometer indicator, which indicates the amount of fuel burned per every passenger-kilometer flown, consumed during the entire flight time period (taxiing, takeoff, cruise, approach and landing [13]. Efficiency is often defined as fuel consumption relative to the payload (in tons per 1 km of flight with the indicator fuel / ton -km. Since 2020, when certifying new aircraft, ICAO has been applying a strict standard regarding their fuel efficiency during a cruise flight - ICAO metric value (MV). The CO_2 metric system is based on three elements associated with aircraft technology and design: Cruise point fuel burn performance; Aircraft size; Aircraft weight.

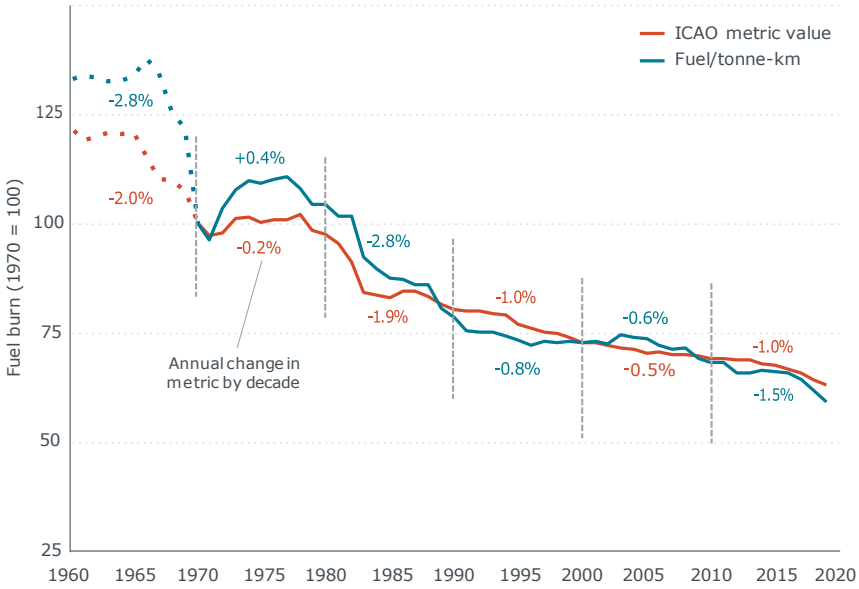


Figure 1. Average fuel burn of new commercial jet aircraft, 1960 to 2019 (1970=100) [14]

The average fuel consumption of new jet aircraft has decreased by about 40% in terms of block fuel intensity from 1970 to 2019, or an average annual decrease of about 1.0%. Over the same period, the average annual decreases in cruise fuel efficiency for newly introduced aircraft have been 1.3% and 1.1% (MV for the average aircraft) [14].

The evolution of aircraft engines will continue to be driven by the two basic demands for higher performance and sustainability. Although, historically, new types of engines have displaced their predecessors, nowadays the four main types of aviation engines, although technologically modified, continue to be used - piston, turboprop, turbojet, turbofan. The latter, which are the highest achievements of aeronautical engineering, are the most used today, mainly due to their high fuel efficiency and low noise levels. In the medium term, aviation engine manufacturers are betting on more radical engine designs, such as open rotor and open fan engines, which will further improve fuel efficiency. The design of an open rotor engine aims to reduce fuel consumption by 20-30 % compared to traditional turbofan engines and to reduce noise compared to conventional gas turbine engines [15].

Radical are the projects to change the conventional aircraft design, among which the main one is the design with a blended wing and body (BWB). This new design will increase aerodynamic efficiency by minimizing all surfaces of the aircraft that do not contribute to the generation of lift, and potentially offer major improvements in fuel efficiency. The future of

aviation depends largely on innovations in hardware and software, as well as in the implementation of artificial intelligence. The ultimate goal is the digital transformation of aviation.

II.2. Implementation of sustainable aviation fuels

The accelerated and large-scale deployment of sustainable aviation fuels (SAF) is essential for the decarbonization of aviation.

To qualify as sustainable, aviation fuel must:

- be obtained from a low-carbon raw material that is mined continuously and repeatedly;
- be an alternative to conventional aviation energy sources and produced using alternative technologies;
- meet the same technical specifications and have the same properties as conventional jet fuel and be able to be mixed with them without requiring changes to existing technologies and fuel systems;
- be safe in storage and operation;
- have a lower net carbon footprint throughout its life cycle.

SAF are fuels that are produced from renewable sources. The raw materials currently used for their production are diverse:

- vegetable and animal oils (soybean, rapeseed, corn, palm), waste fats (oil), waste from food production (fat, tallow, etc.);
- solid household waste (food and other organic waste);
- cellulose waste (wood, agricultural waste and forest residues);
- oilseed plants (Jatropha *curcas, Brassica carinata)* or plants generating abundant biomass for a short growing season, grown on marginal lands or in rotation with food crops (Miscanthus *giganteus, Panicum virgate*)
- some types of algae (Clorella, *Scenedesmus, Saccharina, Laminaria)*

The largest net reduction in CO_2 over the entire life cycle is SAF produced by plants and algae, as they are natural transformers of CO_2 into organic energy-rich matter. Studies show that jet fuel based on the vegetable raw material camelina (*Kamelina sativa)* reduces net CO_2 emissions by about 80% [16]. The produced "drop-in" alternative fuels have the same or identical quality characteristics as those of conventional jet fuels, and in some parameters even surpass them. This type of fuel has a slightly higher calorific value (reducing fuel consumption), a higher flash point (greater safety during ground handling when refueling), and a lower freezing point (increasing the effective operating range of the aircraft).

Currently, numerous technological processes have been developed and certified for the production of sustainable aviation fuels. The largest relative share have the fuels, the production of which is based on natural raw materials and undergoes three main technological processes:

- Through Fischer-Tropsch (FT) synthesis, synthetic fuel is produced from raw materials: natural gas (Gas This Liquid - GTL), coal (Coal This Liquid - CTL) and biomass (Biomass This Liquid - BTL), such as dried hay, agricultural waste, wood chips and forest waste.
- Hydrogenated biomass oils (HBO) are produced from animal fats as well as oils from different types of oil plants (Camelina /wild flax- *Camelina sativa,* jatropha - *Jatropha curcas* and glasswort or algae oils).
- Hydrotreated cellulose fiber fuels (HCF) are produced using cellulosic biomass derived from forestry and agricultural waste.

II.3. Flight operational improvements

Potential reserves for reducing harmful emissions from aviation are also found in air traffic management activities. Air traffic has almost doubled in the last two decades. This growth has intensified the work of air traffic control (ATC) authorities, which must improve the flight regulation system, reduce aircraft downtime on the ground, and thus their emissions. Airlines are committing to initiatives that support efforts to establish a single sky. One such program is AIRE, the Atlantic Interoperability initiative to Reduce Emissions, spearheading energy efficient ATM operations to lower engine emissions and aircraft noise [17]. One of the main goals is to shorten flight routes. Reducing flight time by one minute would result in a saving of 100 kg of aviation fuel.

Optimization of the flight path, more precise estimates of the take-off and landing times of aircraft allow fuel savings and reduce gas emissions. Additional reserves are also revealed when reaching optimal cruising flight altitudes with less air traffic and greater freedom of piloting. A common practice for saving fuel after landing is taxiing with one engine, external electrical power to the aircraft at the airport, saving up to 200 kg fuel per hour.

II.4. Regulatory and market mechanisms

The European Union has taken measures to reduce emissions from air transport by including airlines in the emissions trading scheme from 2022. The emissions trading scheme applies to all flights with a starting point and a destination in the European Economic Area (EU, Iceland, Liechtenstein and Norway) [18].

The politicians' intentions are to discontinue the issuance of free emission permits from 2026, which will accelerate the process of implementing "green" hydrogen as the most environmentally friendly energy source, and redirect revenues from the sale of permits to the development of new technologies through the Innovation Fund [19].

For flights outside the European Economic Area from the beginning of 2021, the EU and the International Civil Aviation Organization are working on the implementation of the voluntary market initiative - Carbon Offsetting and Reduction Scheme for International Aviation (CORSIA) [20], under which airlines can offset their emissions through market-based instruments and environmental projects. Participation in CORSIA is planned to become mandatory from 2027.

According to ReFuelEU Aviation Regulation [21], fuel suppliers should mix 2% SAF into conventional Jet A1, and this proportion will increase up to 34% by 2040 in order for SAF to reach 70 % of the jet fuel composition by 2050. At this time, no regulations are being discussed regarding Avgas 100 LL, probably due to the low relative share of its use.

Decarbonization is one of the main strategic goals for sustainable development of airlines. One of the leading European airlines, Wizz Air, plans to reduce carbon emissions by 25% by 2030 compared to 2019 [22]. The “Flight to Net Zero” plan, based on five main pillars, outlines the company’s roadmap to achieve net zero emissions by 2050:

-53% decarbonization through expanded use of SAF;

-21% decarbonization through technological advances in aircraft and engines;

-7% decarbonization through fleet change;

-4% decarbonization through air traffic reform;

-2% decarbonization through operational efficiency [23].

III. Conclusion

The aviation industry’s transition to achieving sustainable climate neutrality is not a new development, but it has significantly intensified over the past decade. A remarkable achievement is that modern aircraft flights emit an average of 54.3% less greenhouse gases than they did in 1990. This is a result of new technologies, increased operational and infrastructure

efficiency, fleet replacement, improved fuel efficiency and many other innovations. Public interest in the future of aviation will continue to grow, and the ambitious goals of achieving zero net carbon emissions in relatively short term will accelerate the pace of implementation of more environmentally friendly engines, new powertrains, renewable energy sources (including electricity, hydrogen), the use of "green" products and services related to the maintenance and repair of aviation equipment and airport infrastructure, as well as in the accompanying administrative, control and other activities.

REFERENCES:

1. Aircraft Engine Emissions, https://www.icao.int/environmental-protection/Pages/aircraft-engine-emissions.aspx , available 30.05.2025
2. David S. Lee, David W. Fahey , Piers M. Forster , Peter J. Newton , Ron CN Wit , Ling L. Lim , Bethan Owen , Robert Sausen , *Aviation and global climate change in the 21st century* , Atmospheric Environment , Volume 43, Issues 22–23, 2009, Pages 3520-3537, ISSN 1352-2310, https://doi.org/10.1016/j.atmosenv.2009.04.024.
3. Reducing emissions from aviation, https://climate.ec.europa.eu/eu-action/transport/reducing-emissions-aviation_en , available 30.05.2025
4. Aviation industry reducing its climate impact, https://aviationbenefits.org/environmental-efficiency/climate-action/, available 30.05.2025
5. Resolution on the climate, energy and environmental State aid guidelines (CEEAG) , https://oeil.secure.europarl.europa.eu/oeil/en/procedure-file?reference=2021/2923(RSP)#gateway , available 30.05.2025
6. Mrazova, M., *Future directions of fuel efficiency in aviation industry*, INCAS BULLETIN, Volume 5, Issue 4/ 2013 doi: 10.13111/2066-8201.2013.5.4.8]
7. European Commission: Directorate-General for Research and Innovation and Directorate-General for Mobility and Transport, Flightpath 2050 – Europe's vision for aviation – Maintaining global leadership and serving society's needs, Publications Office, 2011, https://data.europa.eu/doi/10.2777/50266
8. European Commission: Directorate-General for Research and Innovation and Directorate-General for Mobility and Transport, Flightpath 2050 – Europe's vision for aviation – Maintaining global leadership and serving society's needs, Publications Office, 2011, https://data.europa.eu/doi/10.2777/50266
9. Pratt & Whitney, GTF engine https://prattwhitney.com/products-and-services/products/commercial-engines/pratt-and-whitney-gtf
10. Sim, KH, Wang, GF, & Kim, TJ (2018). *Status of titanium alloy industry for aviation in the world and development strategy of Chinese enterprises*. DEStech Trans. Soc. Sci. Educ. Hum. Sci.
11. Bachmann, J., Yi, X., Gong, H. et al. Outlook on ecologically improved composites for aviation interior and secondary structures. CEAS Aeronaut J 9, 533–543 (2018). https://doi.org/10.1007/s13272-018-0298-z
12. GE Aerospace, GE90 Engine, https://www.geaerospace.com/commercial/aircraft-engines/ge90
13. Kharina, A. and Rutherford, D. (2015). *Fuel efficiency trends for new commercial jet aircraft: 1960 to 2014* , https://theicct.org/wp-content/uploads/2021/06/ICCT_Aircraft-FE-Trends_20150902.pdf
14. Xinyi Sola Zheng , Dan Rutherford , Ph.D. , Fuel burn of new commercial jet aircraft : 1960 to 2019,2020,Available at: https://theicct.org/wp-content/uploads/2021/06/Aircraft-fuel-burn-trends-sept2020.pdf

15. Open-Rotor Engines and Aviation Decarbonization https://www.iba.aero/resources/articles/will-open-rotor-engines-be-the-next-great-leap-in-decarbonisation/
16. Shonnard, D., Williams, L., Kalnes, T., Camelina-Derived Jet Fuel and Diesel: Sustainable Advanced Biofuels, 2010, https://aiche.onlinelibrary.wiley.com/doi/epdf/10.1002/ep.10461
17. Atlantic Interoperability Initiative this Reduce Emissions (AIRE), https://sesar.eu/sites/default/files/AIRE_2010-11_executive_summary.pdf
18. Reducing carbon emissions: EU goals and policies, https://www.europarl.europa.eu/topics/bg/article/20180305STO99003/namaliavane-na-vghlerodnite-emisii-tseli-i-politiki-na-es
19. Cutting emissions from planes and ships: EU actions explained, https://www.europarl.europa.eu/topics/bg/article/20220610STO32720/namaliavane-na-emisiite-ot-samoleti-i-korabi-merkite-na-es].
20. Carbon Offsetting and Reduction Scheme for International Aviation https://www.icao.int/environmental-protection/CORSIA/Pages/default.aspx
21. Regulation (EU) 2023/2405 of the European Parliament and of the Council of 18 October 2023 on ensuring a level playing field for sustainable air transport (ReFuelEU Aviation),https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32023R2405
22. https://www.wizzair.com/en-gb/information-and-services/about-us/sustainability#economy-part2
23. Wizz Air's roadmap to zero emissions by 2050 – a manifesto for urgent action in aviation, 2025, on-line https://pan.bg/view_article-1-591535-showimg1-pytnata-karta-na-Wizz-Air-kym-nulevi-emisii-do-2050-g-manifest-za-speshni.html, accessed on 27.05.2025

ПРЕХОД КЪМ КЛИМАТИЧНО УСТОЙЧИВ АВИАЦИОНЕН ТРАНСПОРТ

Иван Бенков

Резюме: Авиационната индустрия, която е сред най-динамичните индустрии, генериращи икономически растеж, заетост и улесняващи търговията, туризма, свързаността между хора от различни географски региони, оказва значителен екологичен натиск върху околната среда и организмовия свят. Най-силно въздействие върху околната среда се причинява от остатъците от изгарянето на гориво в двигателите на самолетите - въглероден диоксид, азотни оксиди, серен диоксид, въглероден оксид, водни пари. От тях най-голям дял има парниковият газ - CO_2.

Подсекторът на авиацията е сред първите, които поемат по пътя на устойчивото развитие. Преходът към климатично устойчив въздушен транспорт е много труден, сложен, наукоемък и капиталоемък процес. Основните ангажименти на авиационната индустрия за намаляване на въздействието ѝ върху климата са насочени към технологични иновации в самолетите, въвеждане на устойчиви авиационни горива, оперативни подобрения на полетите, повишаване на ефективността на инфраструктурата и прилагане на пазарни и регулаторни механизми.