



ДИНАМИКА, БАЛЛИСТИКА, УПРАВЛЕНИЕ ДВИЖЕНИЕМ ЛЕТАТЕЛЬНЫХ АППАРАТОВ/DYNAMICS, BALLISTICS, AIRCRAFT MOVEMENT CONTROL

DOI: <https://doi.org/10.60797/IRJ.2026.172.56> EDN: QXTQAI**ARTIFICIAL INTELLIGENCE ALGORITHMS FOR HEAVY UAVS**

Research article

Ermilov A.S.^{1,*}¹ ORCID : 0009-0007-4549-172X;¹ RUDN University, Moscow, Russian Federation

* Corresponding author (eemilov-sasha[at]yandex.ru)

Suggested: 20.01.2026; Accepted: 06.04.2026; Published: 16.10.2026

Abstract

Unmanned Aerial Vehicles (UAVs) have become an integral part of modern life. However, one of the main limitations to their wider adoption remains the energy challenge — specifically, the limited capacity of onboard batteries and the efficiency of energy consumption. While significant resources are being invested in the search for alternative energy sources, lithium extraction, and the development of advanced battery technologies, far less attention has been given to the electronic systems responsible for managing that energy — namely, ESCs (Electronic Speed Controllers), switches, and control algorithms.

This situation can be compared to the automotive industry's transition from carburetors to electronic fuel injection systems, which significantly improved fuel efficiency. Similarly, UAV performance can be significantly enhanced not only through hardware upgrades but also by optimizing energy management via intelligent control electronics.

This article explores a new direction in UAV development: increasing flight range and speed without relying on additional power sources or fuel, but purely through the optimization of control systems and electronic regulation. Integrating Artificial Intelligence (AI) can further improve these systems' efficiency by extending UAVs flight durations. While the focus is on control algorithms designed for heavy-duty transport multicopters, many of the methods discussed can be applied across a broad range of UAV types to improve energy efficiency and maximize battery life. The key advantage lies in the fact that the most expensive components of the UAV — including the battery pack, frame, motors, and powertrain — remain unchanged. Enhancing only the electronic control systems provides a cost-effective solution to improve the performance of existing UAV fleets. Notably, even the use of smart electronic switches paired with standard ESCs can yield up to a 30% increase in flight autonomy.

Keywords: Unmanned Aerial Vehicle (UAV), drone, energy efficiency, Electronic Speed Controller (ESC), flight time, range extension, energy management, control algorithms, Artificial Intelligence (AI), multicopter, battery optimization, power electronics, heavy-lift drone, energy consumption.

АЛГОРИТМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ТЯЖЕЛЫХ БЕСПИЛОТНЫХ ВОЗДУШНЫХ СУДОВ

Научная статья

Ермилов А.С.^{1,*}¹ ORCID : 0009-0007-4549-172X;¹ Российский университет дружбы народов, Москва, Российская Федерация

* Корреспондирующий автор (eemilov-sasha[at]yandex.ru)

Предложена: 20.01.2026; Принята: 06.04.2026; Опубликовано: 16.10.2026

Аннотация

Современная жизнь уже немыслима без UAV's. Основным препятствием более широкого применения является энергетический фактор — запас энергии в аккумуляторах и ее эффективное расходование. Большие средства тратятся на поиск новых источников энергии, разработке новых месторождений лития и улучшение конструкций аккумуляторов. Но значительно меньше внимания уделяется электронным распределительным устройствам этой энергии (ESC регуляторы, коммутаторы, алгоритмы управления и работы).

Эту ситуацию можно сравнить с переходом автомобильной промышленности от карбюраторов к электронным системам впрыска топлива, что позволило значительно повысить топливную эффективность. С появлением в нашей жизни Искусственного Интеллекта (ИИ) стало возможным значительно повысить эффективность данных устройств и обеспечить значительный прирост автономной работы UAV's.

В данной статье приведены некоторые примеры алгоритмов для управления тяжелого транспортного мультикоптера, позволяющие значительно увеличить время его полета без использования дополнительных аккумуляторов и повышения полетного веса. Хотя рассмотрены алгоритмы управления тяжелым транспортным мультикоптером, многие описанные алгоритмы могут быть успешно использованы и в других типах UAV's для экономии энергии аккумуляторов и повышения времени автономной работы. Особо следует отметить, что даже использование электронных коммутаторов со стандартными ESC-регуляторами позволяет увеличить время автономной работы UAV's до 30%.

Ключевые слова: беспилотный летательный аппарат (БПЛА), дрон, энергоэффективность, регулятор скорости (ESC), время полета, дальность полета, система управления энергопотреблением, алгоритмы управления, искусственный интеллект (ИИ), мультикоптер, оптимизация батареи, силовая электроника, грузовой мультикоптер, энергопотребление.

Introduction

A standard BLDC (Brushless DC) system consists of a battery, an ESC (Electronic Speed Controller), and a BLDC motor [1], [2], [3], [4], as shown in Figure 1.

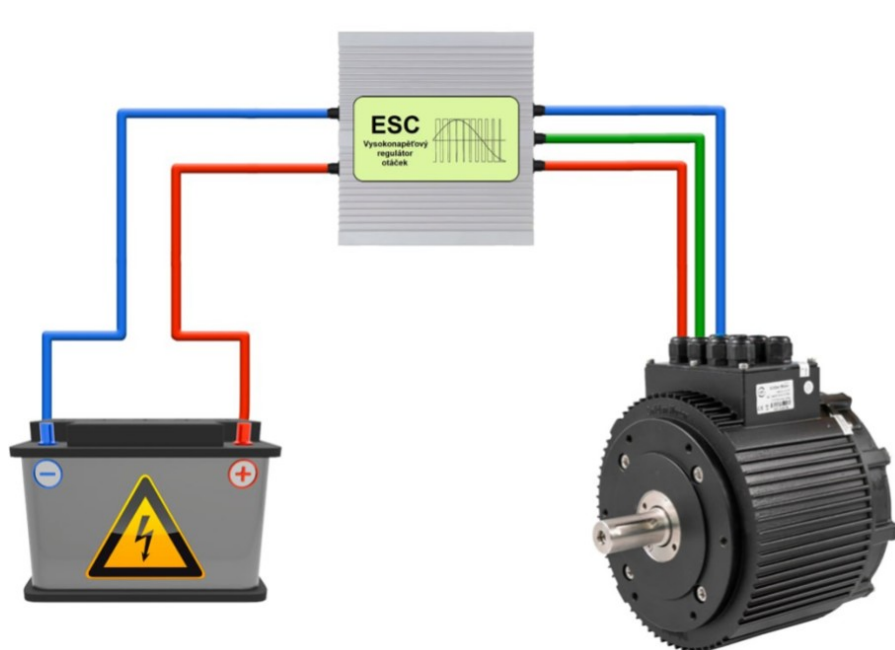


Figure 1 - Standard BLDC system
DOI: <https://doi.org/10.60797/IRJ.2026.172.56.1>

Batteries commonly used in UAVs include Lithium Polymer (Li-Po) [5], [6], [7], [8] and Lithium Iron Phosphate (LiFePO₄) cells [9], [10], [11], [12]. The latter are less widespread due to their higher cost and manufacturing complexity. While Li-Po batteries offer high energy density, they are sensitive to deep discharge and present a significant fire risk [11], [12], [13], [14]. In contrast, LiFePO₄ batteries are more thermally stable and fire-safe but have a lower energy density.

Recent advancements have introduced Lithium Titanate (LTO) [15], [16], [17] batteries have been introduced for use in heavy UAVs. Their key advantages include extremely fast charging—an important feature for aerial taxi applications—and reliable operation at low temperatures. However, all battery types share a common trait: a nonlinear discharge profile. This characteristic must be taken into account to ensure reliable UAV performance [18], [19].

The ESCs convert the energy stored in the battery into high-frequency pulses to drive BLDC motors [15], [20]. Most ESCs share a similar circuit architecture comprising of a microcontroller and high-power MOSFET transistors [21], [22], [23] with variations in synchronization algorithms, PWM frequency, and the power handling capacity of the output stage [24]. Silicon-based MOSFETs are used as power switches in nearly all ESCs [25], [26], [27]. However, there are already small-batch ESCs based on GaN (Gallium Nitride) transistors [20], [28], [29], [30], which outperform silicon-based designs and allow high-power, compact solutions. Their main drawback is high cost and the need for liquid cooling.

BLDC motors themselves follow a consistent design principle, differing mainly in size, number of poles, and magnet configurations [31]. Most motor windings are made from copper wire with enamel insulation rated for up to 130 °C [4], [32], while newer implementations employ nickel-plated copper wire with high-temperature insulation rated up to 300 °C [4], [33]. This enables operation of windings at temperatures up to 250°C, supporting current levels exceeding standard limits [34], preventing insulation breakdown during prolonged peak load conditions, and enabling short-term “boost” modes [33]. Additional thermal protection for rotor magnets is achieved through specialized heat-resistant coatings and barrier layers [35], [36], [37].

Furthermore, the use of ceramic ball bearings significantly extends motor lifespan, as bearings are the only components subject to mechanical wear in BLDC motors [20], [38].

In contrast to small drones, heavy multicopter-type UAVs also incorporate variable-pitch propeller systems. These allow the motor-propeller group to adapt to different loads and flight conditions [39], [40]. However, in most systems, all propellers are adjusted simultaneously: a higher pitch for light-load high-speed flight, and a lower pitch for heavy-load low-speed flight.

Proposed Modernized BLDC System

To enhance the capabilities of the standard BLDC system, it is upgraded with additional components, including ESC current sensors (I), motor tachometers (T), and an Artificial Intelligence (AI) module (Figure 2).

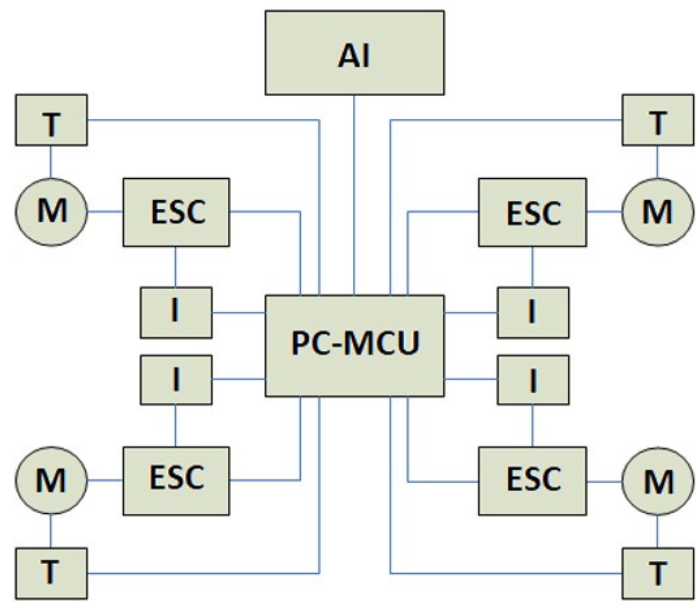


Figure 2 - Modernized BLDC System
DOI: <https://doi.org/10.60797/IRJ.2026.172.56.2>

The ESC current sensors (I) monitor the power consumption of the motors (M) under varying load conditions. The tachometers (T) track the actual rotational speed of the propellers, helping to detect and prevent slippage during high-load flight scenarios.

The flight controller PC-MCU handles real-time UAV navigation and stabilization tasks.

The Artificial Intelligence module (AI) stores data on all system components, including their operating limits and specifications. It also runs algorithms designed to optimize energy consumption during flight by intelligently managing power distribution and motor performance.

The following sections describe several practical AI-based algorithms that have been successfully implemented on a heavy-duty multicopter and have proven effective in real-world testing.

Proposed Algorithms, Experimental Verification, and Results

The proposed algorithms aim to optimize the energy consumption of onboard batteries in heavy-duty multicopter-type UAVs, depending on flight conditions (Figure 3).



Figure 3 - Heavy Transport Multicopter-Type UAV
DOI: <https://doi.org/10.60797/IRJ.2026.172.56.3>

3.1. Platform Balancing Algorithm for Heavy Multicopters

To transport irregular or oversized loads, heavy multicopters typically use an external suspension system, with the suspension point located at the UAV's center of gravity. However, this method is not always suitable—for example, during ground-effect flight at low altitudes (below 3 meters) or in strong wind conditions where there is a risk of load swinging. In such cases, proper load distribution on the platform and balancing of the UAV become crucial.

Often, due to the shape and size of the cargo, it is not possible to distribute the weight symmetrically. Examples of uniform and non-uniform load distribution on a heavy UAV platform are shown in Figure 4.

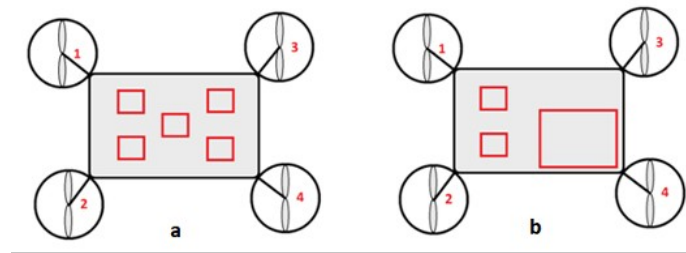


Figure 4 - Uniform (a) and non-uniform (b) load distribution on the UAV platform

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.4>

In the case shown in Figure 4b, the center of gravity is shifted. The flight controller attempts to maintain a level frame by varying the motor speeds. If the imbalance is minor, the multicopter can hover stably under calm conditions. However, motors will operate under different load conditions. The more heavily loaded motors will have a smaller dynamic reserve of RPM and power to respond to flight controller commands compensating for wind gusts or manual control inputs. As a result, the UAV may begin to drift toward the shifted center of gravity.

Furthermore, to maintain balance with uneven cargo placement, motor 4 must operate at significantly higher speeds than the other motors. This results in excessive heat generation, accelerated bearing wear, and a higher current load on its ESC. A component under sustained overload becomes a point of failure, with reduced reliability, shorter lifespan, and higher failure probability. In heavy multicopters, failure of even one of the four motors results in complete failure of the aircraft. Thus, cargo imbalance is a critical factor in UAV operation.

For a comparative analysis, let us calculate the load characteristics of motors in a conceptual UAV with a 2-meter arm length and a payload located 1 meter from the center of gravity. A schematic representation is shown in Figure 5.

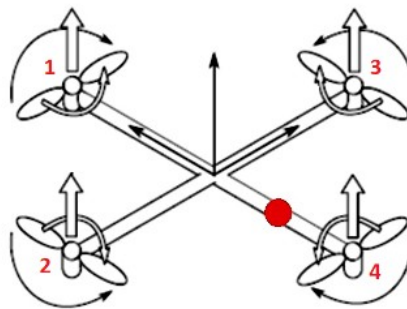


Figure 5 - Conceptual UAV

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.5>

The thrust of a fixed-pitch propeller is proportional to the square of its rotation speed and can be estimated using the simplified formula:

$$F = C_A \times \rho \times n^2 \times D^4, \quad (1)$$

where

C_A — thrust coefficient for an ideal propeller;

ρ — air density;

n — propeller rotational speed (RPM);

D — propeller diameter.

To maintain stable flight under asymmetric loading, the thrust produced by motor 4 must be three times that of motor 1. Similarly, motors 2 and 3 must each produce twice the thrust of motor 1.

This leads to the following required speed ratios between motors:

- Motor 1 : 1;
- Motors 2 and 3 : 1.41;
- Motor 4 : 1.73.

For example, if motor 1 is spinning at 5,000 RPM, motor 4 would need to spin at approximately 8,650 RPM while still retain sufficient headroom for maneuvering — which is often not feasible in real-world conditions.

Another critical factor for heavy multicopters is managing centrifugal forces generated by the spinning propellers, which can cause unwanted yaw (rotation about the vertical axis). As a propeller spins, centrifugal force acts outward along the blade, perpendicular to the axis of rotation.

The force on any given point of the blade is proportional to the square of its angular velocity, described by:

$$F_c = m \times \omega^2 \times r \quad (2)$$

where

m — mass of the blade segment;

ω — angular velocity;

r — radial distance from the axis.

If propellers are arranged in counter-rotating pairs (two clockwise, two counterclockwise) and spin at equal speeds, these centrifugal forces cancel out. However, when there are significant differences in RPM — as in this case, where some motors are spinning nearly twice as fast as others — a net rotational torque arises, which the flight controller must counteract. In some situations, the motors may lack the dynamic range to fully compensate, leading to instability.

This is why the platform balancing algorithm is critical not only for flight reliability but also for ensuring precise control.

In our heavy-lift multirotor systems equipped with variable-pitch propellers, this issue is addressed by dynamically adjusting the pitch of each propeller. Based on the load distribution across the platform, the balancing algorithm tunes individual blade pitch values to equalize the average rotational speed of each motor. This helps prevent overloading, extends service life, and ensures stable, energy-efficient operation.

3.2. Platform Balancing Algorithm Description

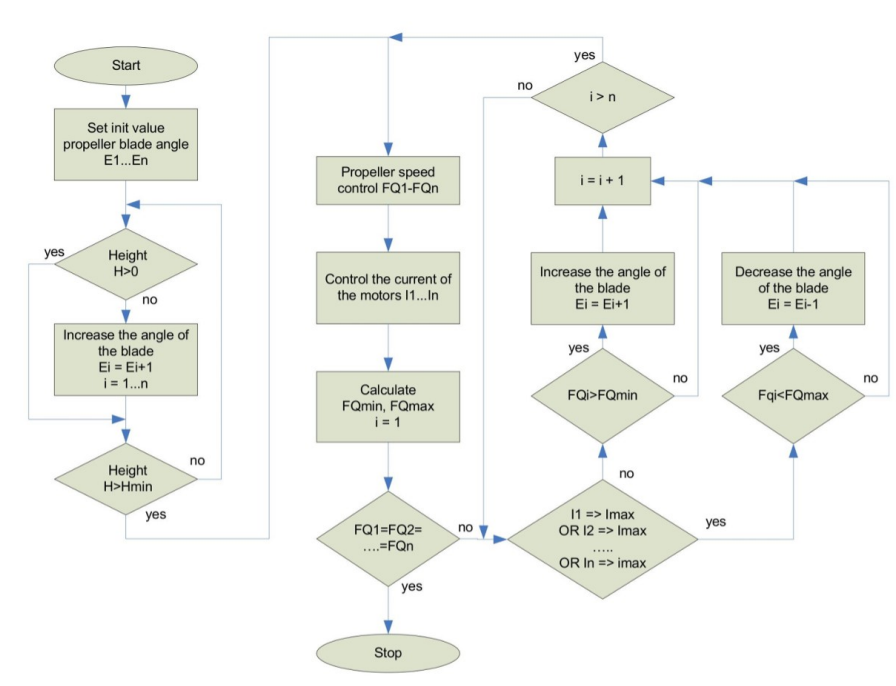


Figure 6 - Structural Diagram of the Platform Balancing Algorithm

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.6>

The PC-MCU flight controller initiates the balancing process by lifting the heavy multirotor UAV to a minimum safe altitude. During this phase, it measures the operating current of each ESC using current sensors (I) and records the motor RPM using tachometers (T).

The system then enters a continuous adjustment phase. It monitors ESC current in real time and makes dynamic pitch and RPM corrections: for arms where motors are spinning faster, the algorithm increases propeller pitch while proportionally reducing RPM. Conversely, for arms with lower RPMs, the pitch is reduced and motor speed increased. This adaptive process ensures all rotors operate at approximately the same average RPM, which helps evenly distribute mechanical wear—particularly on motor bearings.

In addition to the core logic, an AI-based module (not shown in the diagram) continuously monitors several critical parameters to ensure system reliability:

- ESC current on each rotor arm, to prevent exceeding safe peak thresholds;
- Motor temperature, to avoid stator overheating and subsequent rotor magnet demagnetization due to thermal stress;
- Propeller RPM and PWM signal duration to minimize heat buildup during rapid switching transitions in the ESC's MOSFET output stage.

Based on these inputs, the algorithm continuously adjusts system parameters to meet operational reliability requirements of all critical components. The algorithm was tested on a dedicated bench setup simulating a representative UAV. Measurement results were recorded at two stages:

- Initial stage: Lift to minimum altitude without adjusting motor speed or propeller pitch;
- Final stage: After applying RPM and pitch corrections.

The corresponding data is summarized in Table 1.

Table 1 - Experimental results before and after RPM-based propeller pitch adjustment

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.7>

	Initial Stage	Final Stage
Motor 1 RPM	1800	2850
Motor 2 RPM	2800	2800
Motor 3 RPM	2800	2800
Motor 4 RPM	3450	2750
Propeller Pitch 1	8	5
Propeller Pitch 2	8	8
Propeller Pitch 3	8	8
Propeller Pitch 4	8	10

The data clearly demonstrates the algorithm's effectiveness in equalizing rotor RPMs across the platform. The slight deviation in Motor 1's final RPM from the target value occurs because the thrust vector of Motors 2 and 3 does not align perfectly with the platform's center of gravity, introducing a minor imbalance in overall lift distribution.

From an aerodynamic standpoint, it would be advisable to expand the algorithm with constraints on permissible propeller pitch values to ensure stable and predictable behavior.

Below is a code snippet illustrating the implementation of the described algorithm:

```
#define NUMBER_ROTOR      4
#define BLADE_ANGLE_START 2
#define BLADE_ANGLE_MAX  20
#define HIGH_TAKEOFF      2000 //mm
#define MAX_CURRENT        10000 //mA
#define DELTA_CURRENT      100 //mA
#define DELTA_SPEED        100 //rpm
#define MAX_SPEED          10000 //rpm

int Blade_Angle[NUMBER_ROTOR];

// Takeoff and weighing
int Takeoff(void)
{
    int Index;
    int Max_Angle;
    for(Index = 0; Index < NUMBER_ROTOR; Index++) Blade_Angle[Index] = BLADE_ANGLE_START;
    Start_ESC();
    while(Get_High() < HIGH_TAKEOFF)
    {
        for(Index = 0; Index < NUMBER_ROTOR; Index++) Set_Blade_Angle(Blade_Angle[Index]);
        delay(100);
        Max_Angle = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Blade_Angle[Index]++;
            if(Blade_Angle[Index] > BLADE_ANGLE_MAX) Max_Angle = 1;
        }
        if(Max_Angle) break;
    }
}

// Speed equalization
int Speed_Equ(void)
{
    int Rotor_Speed[NUMBER_ROTOR];
    int Rotor_Current[NUMBER_ROTOR];
    int Max_Speed, Min_Speed, Sub_Speed, Max_Current, Index;
    int Max_Angle;

    while(TRUE)
    {
        Max_Speed = 0;
        Min_Speed = MAX_SPEED;
        Sub_Speed = 0;
        Max_Current = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Rotor_Speed[Index] = Read_Speed(Index);
            Rotor_Current[Index] = Read_Current(Index);
```

```

        if(Rotor_Speed(Index) > Max_Speed) Max_Speed = Rotor_Speed(Index);
        if(Rotor_Speed(Index) < Min_Speed) Min_Speed = Rotor_Speed(Index);
        if(Rotor_Current(Index) > Max_Current) Max_Current = Rotor_Current(Index);
    }
    for(Index = 0; Index < NUMBER_ROTOR; Index++) Sub_Speed = Sub_Speed +
(Max_Speed - Rotor_Speed(Index));
    Sub_Speed = Sub_Speed/NUMBER_ROTOR;
    if(Sub_Speed < DELTA_SPEED) break;

    Max_Angle = 0;
    for(Index = 0; Index < NUMBER_ROTOR; Index++)
    {
        if(Max_Current < MAX_CURRENT)
        {
            if(Rotor_Speed[Index] > Min_Speed) Blade_Angle[Index]++;
            if(Blade_Angle[Index] > BLADE_ANGLE_MAX) Max_Angle = 1;
        } else
        {
            if(Rotor_Speed[Index] < Max_Speed) Blade_Angle[Index]--;
            if(Blade_Angle[Index] < BLADE_ANGLE_START) Max_Angle = 1;
        }
    }
    if(Max_Angle) break;
}

// Current equalization
int Current_Equ(void)
{
    int Rotor_Current[NUMBER_ROTOR];
    int Max_Current, Index;
    int Max_Angle;

    while(Get_Stat_ESC()) // Rotors RUN
    {
        Max_Current = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Rotor_Current[Index] = Read_Current(Index);
            if(Rotor_Current[Index] > Max_Current) Max_Current = Rotor_Current[Index];
            if((Blade_Angle[Index] >= BLADE_ANGLE_MAX) || (Blade_Angle[Index] <=
BLADE_ANGLE_START) Max_Angle = 1;
            else Max_Angle = 0;
        }
        if(!Max_Angle)
        {
            for(Index = 0; Index < NUMBER_ROTOR; Index++)
            {
                if((Max_Current < (MAX_CURRENT - DELTA_CURRENT))
                Blade_Angle[Index]++;
                if((Max_Current > (MAX_CURRENT + DELTA_CURRENT))
                Blade_Angle[Index]--;
            }
        }
    }
}

```

3.3. Propeller Pitch Optimization Algorithm

This algorithm is used to determine the optimal propeller pitch settings based on the specific conditions of a predefined flight route. As with any navigation system, two modes are available: “Minimum Time to Destination” and “Shortest Distance”.

The “Minimum Time” mode prioritizes the fastest possible travel, aiming for the highest attainable flight speed. Conversely, the “Shortest Distance” mode in this context focuses on minimizing battery energy consumption, operating under an energy-efficient strategy for maximum endurance. The UAV's flight speed depends on propeller pitch, RPM, and is calculated using the simplified formula:

$$V = s \times H \times n \quad (3)$$

where

s — slip coefficient (typically 0.75);

H — propeller pitch;

n — propeller rotation frequency (RPM).

From this, two derived formulas are used within the algorithm for determining the optimal pitch. To determine the required pitch for a given speed and RPM:

$$H = V \times 1000 \times 60 / (0,75 \times n) \quad (4)$$

To determine the required RPM for a given speed and pitch:

$$n = V \times 1000 \times 60 / (0,75 \times H). \quad (5)$$

3.4. Algorithm Overview for Maximum Route Speed Configuration

In this mode, the algorithm sets the propellers to the maximum permissible pitch to achieve the highest possible speed. Throughout the flight, it continuously monitors motor RPM and ESC current to ensure operational safety and performance stability.

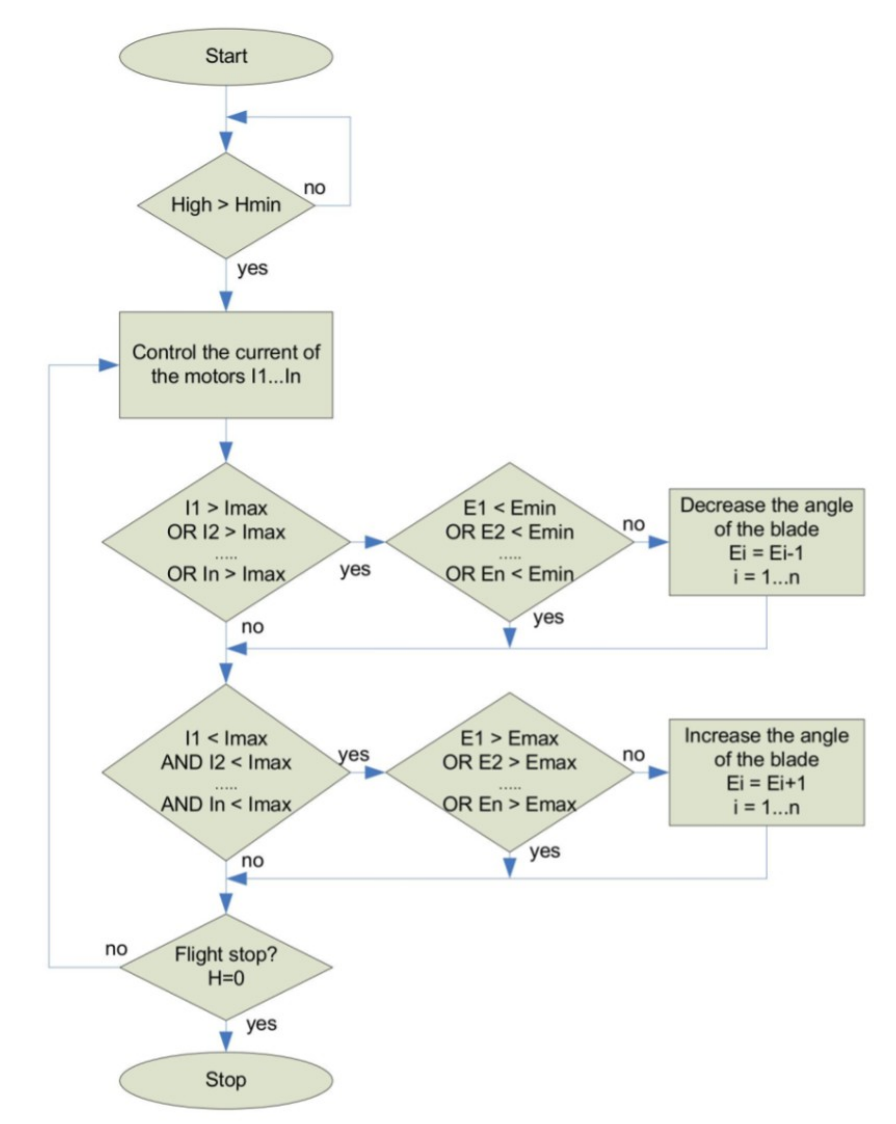


Figure 7 - Heavy Transport Multicopter-Type UAV

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.8>

The PC-MCU flight controller lifts the heavy multicopter to a minimal safe altitude and begins measuring the operating current drawn by each ESC using current sensor (I), as well as the propeller rotation speed via tachometers (T).

Subsequently, while continuously monitoring the ESC current in real time, the algorithm gradually increases the pitch of all propellers in sync until one of the following thresholds is reached: either the maximum allowable current on any ESC or the maximum pitch setting. The system uses RPM monitoring during this process to evaluate aerodynamic efficiency, since each propeller exhibits unique aerodynamic load characteristics under different conditions.

This algorithm was validated on a test bench simulating a typical UAV environment. Measurement results from the initial stage (liftoff to minimal altitude with default propeller pitch) and the final stage (after pitch optimization) are summarized in Table 2 below.

Table 2 - Measurement results Before and After Propeller Pitch Adjustment Based on Current Draw

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.9>

Motor RPM = constant 2800 RPM	Initial Stage	Final Stage
ESC Current 1, A	54	79
ESC Current 2, A	53	79
ESC Current 3, A	55	80
ESC Current 4, A	54	79
Propeller Pitch 1	8	12
Propeller Pitch 2	8	12
Propeller Pitch 3	8	12
Propeller Pitch 4	8	12

The results clearly demonstrate the effectiveness of the algorithm in increasing the multicopter's cruising speed by approximately 1.5 times, achieved through propeller pitch optimization and a controlled rise in motor current. This provides a clear performance advantage in short-distance operations, where battery life is not a critical constraint. The maximum permissible operating current for the ESCs was set at 100 A.

The following section contains the source code that implements this algorithm:

```
#define NUMBER_ROTOR      4
#define BLADE_ANGLE_START 2
#define BLADE_ANGLE_MAX  20
#define HIGH_TAKEOFF      1000 //mm
#define MAX_CURRENT       10000 //mA
#define DELTA_CURRENT     100 //mA

// Current equalization
int Current_Equ(void)
{
    int Rotor_Current[NUMBER_ROTOR];
    int Max_Current, Index;
    int Max_Angle;

    while(Get_Stat_ESC() && (Get_High() > HIGH_TAKEOFF)) // Rotors RUN and the required
height has been reached
    {
        Max_Current = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Rotor_Current[Index] = Read_Current(Index);
            if(Rotor_Current[Index] > Max_Current) Max_Current = Rotor_Current[Index];
            if((Blade_Angle[Index] >= BLADE_ANGLE_MAX) || (Blade_Angle[Index] <=
BLADE_ANGLE_START) Max_Angle = 1;
            else Max_Angle = 0;
        }
        if(!Max_Angle)
        {
            for(Index = 0; Index < NUMBER_ROTOR; Index++)
            {
                if((Max_Current < (MAX_CURRENT - DELTA_CURRENT))
                if((Max_Current > (MAX_CURRENT + DELTA_CURRENT))
                Blade_Angle[Index]++;
                Blade_Angle[Index]--;
            }
        }
    }
}
```

3.5. Battery Energy Optimization Algorithm

This algorithm adjusts propeller pitch based on the optimal motor RPM, load characteristics, and battery discharge profiles.

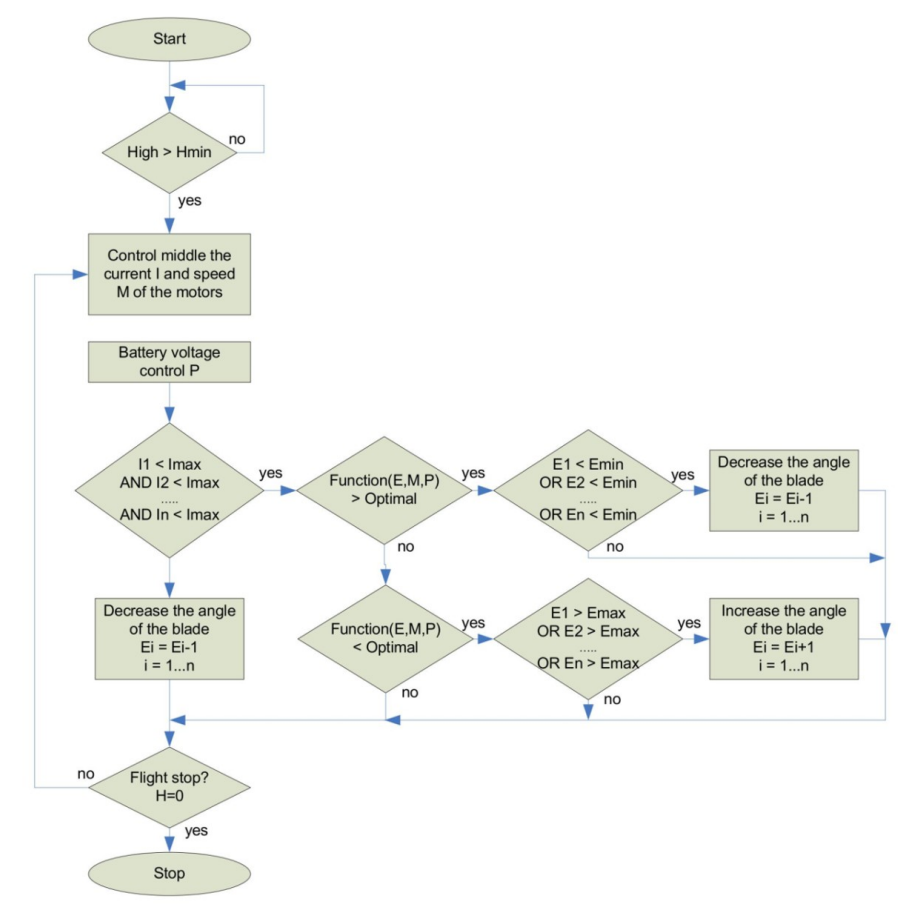


Figure 8 - Structural Diagram of the Algorithm for Optimal Battery Energy Consumption

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.10>

The flight controller PC-MCU lifts the heavy multicopter to a minimum altitude and begins measuring the operating current of the ESCs using current sensor I, along with motor RPM using tachometers T. While continuously monitoring ESC current draw, the algorithm synchronously increases the propeller pitch until the motor load and battery discharge parameters — stored in PC-MCU memory — reach optimal values. These parameters are adjusted in flight at predefined intervals based on remaining battery capacity. If the system determines that the energy reserve is insufficient or the discharge rate is excessive, it triggers an emergency alert. All flight and battery status data are logged to memory for future use.

The relationship between battery discharge time and discharge current is nonlinear, as shown in Figure 9. LiPo batteries typically exhibit a relatively linear discharge curve, whereas LiFePO₄ batteries show more pronounced nonlinearity, especially toward the end of the cycle — an important consideration for the algorithm.

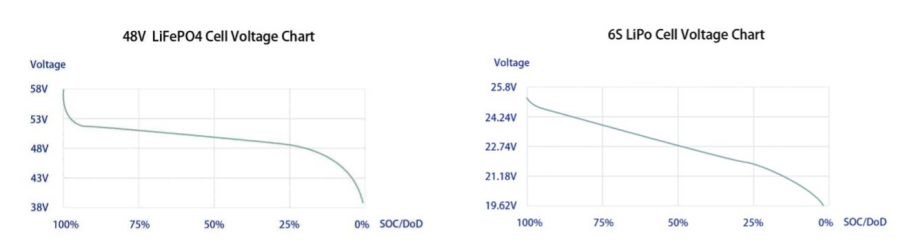


Figure 9 - Battery Discharge Time as a Function of Discharge Current

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.11>

Based on the data in Table 2, the recommended strategy is to maintain maximum speed during the first 75% of the flight (i.e., battery discharge period), then gradually reduce propeller pitch during the final 25% to ensure a smoother discharge. These two propeller pitch optimization algorithms can be run independently or together, depending on the processing power of the onboard computer. Ultimately, the AI system selects the flight mode based on route data, flight distance, and the actual energy reserve of the onboard batteries.

The following code demonstrates how this algorithm operates:

```

#define NUMBER_ROTOR          4
#define BLADE_ANGLE_START    2
#define BLADE_ANGLE_MAX      20
#define HIGH_TAKEOFF         1000          //mm
#define MAX_CURRENT          10000        //mA
#define OPTIMAL_REG          30000

int Blade_Angle[NUMBER_ROTOR];

// Optimization by speed and angle step
int Speed_Angle(void)
{
    int Rotor_Current[NUMBER_ROTOR];
    int Max_Current, Mid_Current, Mid_Speed, Index;
    int Max_Angle, Min_Angle, Power;

    while(Get_Stat_ESC() && (Get_High() > HIGH_TAKEOFF))    // Rotors RUN and the required
height has been reached
    {
        Max_Current = 0;
        Power = Get_Power();
        Mid_Current = 0;
        Mid_Speed = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Rotor_Current[Index] = Read_Current(Index);
            Mid_Current += Rotor_Current[Index];
            Mid_Speed += Read_Speed(Index);
            if(Rotor_Current[Index] > Max_Current) Max_Current = Rotor_Current[Index];
            if((Blade_Angle[Index] >= BLADE_ANGLE_MAX) Max_Angle = 1;
            else if(Blade_Angle[Index] <= BLADE_ANGLE_START) Min_Angle = 1;
            else {Max_Angle = 0; Min_Angle = 0;}
        }
        Mid_Current = Mid_Current/NUMBER_ROTOR;
        Mid_Speed = Mid_Speed/NUMBER_ROTOR;

        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            if((Max_Current < MAX_CURRENT) && !Max_Angle && (Function(Blade_Angle,
Mid_Speed, Power) < OPTIMAL_REG)) Blade_Angle[Index]++;
            if((Max_Current < MAX_CURRENT) && !Min_Angle && (Function(Blade_Angle,
Mid_Speed, Power) > OPTIMAL_REG)) Blade_Angle[Index]--;
            if((Max_Current >= MAX_CURRENT) && !Min_Angle) Blade_Angle[Index]--;
        }
        Set_Blade_Angle();
    }
}

```

Proposed Algorithms for the Extended BLDC System: Experimental Verification and Results

The use of electronic battery section switches and stator winding reconfiguration in BLDC systems has been widely studied as a means to extend flight time and improve battery energy efficiency. Integrating artificial intelligence allows dynamic control of these switches based on real-time flight data, further enhancing system performance.

To recap, this extended BLDC includes, in addition to the standard configuration, an electronic battery switch that reconfigures battery sections to deliver different voltages to the ESCs, and an electronic winding switch, which reconfigures stator windings to modify thrust and speed characteristics of the BLDC motors (Figure 10).

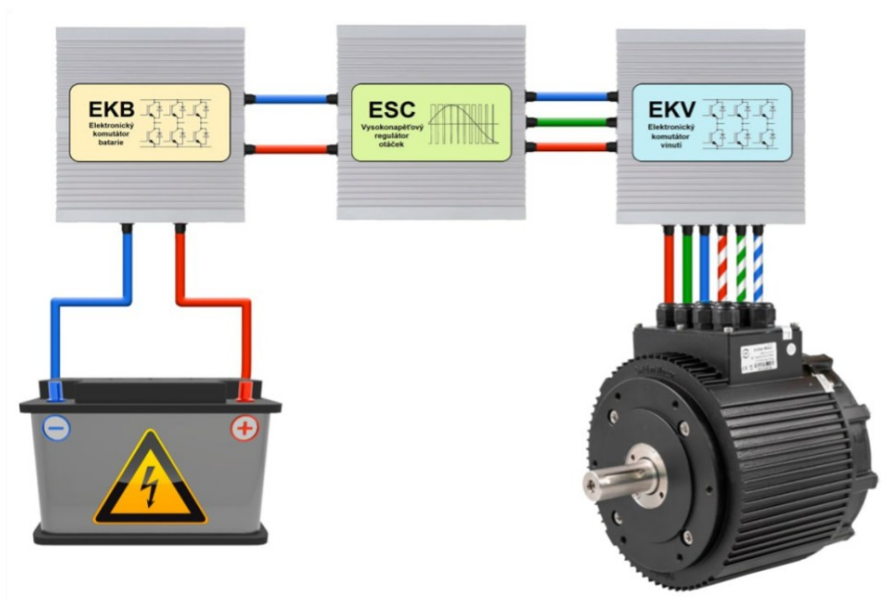


Figure 10 - Schematic representation of the Extended BLDC System
DOI: <https://doi.org/10.60797/IRJ.2026.172.56.12>

4.1. Battery Switch Control Algorithm

The battery switch algorithm functions similarly to an automatic transmission: higher "gears" are engaged as speed increases. A lookup table stored in the AI control unit maps the BLDC motor RPMs at which switching between battery configurations should occur. This keeps motor RPMs within optimal efficiency ranges. For UAVs using multi-stator BLDC motors, a separate switching table is required for each stator winding configuration.

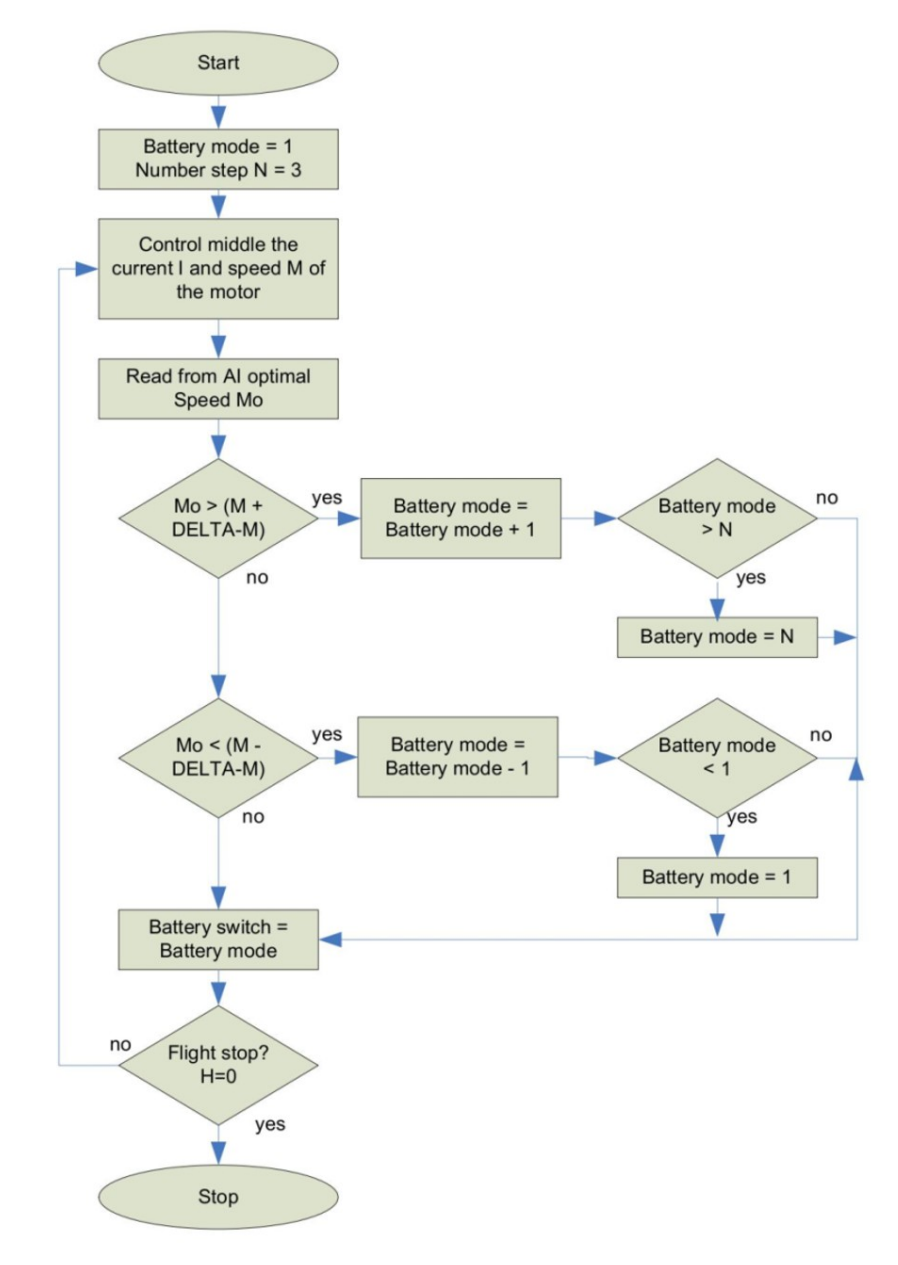


Figure 11 - Structural Diagram of the Battery Switch Control Algorithm

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.13>

The PC-MCU flight controller dynamically measures the propeller's rotational speed M using tachometers T and switches battery sections according to a predefined lookup table stored in memory. Simultaneously, the current consumed by the batteries is measured to ensure optimal discharge characteristics, which are also stored in the PC-MCU's memory.

This algorithm was tested on a test bench that simulated a hypothetical UAV. During testing, motor speed and propeller pitch were held constant, while the battery switch configurations were changed among 10S, 20S, and 40S voltages. During the tests, the motor's operating current and PWM pulse durations were measured.

The measurement results are summarized in Table 3.

Table 3 - Impact of Battery Voltage on Operating Current and PWM Pulse Duration

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.14>

Motor Speed = const. 2,800 RPM	Motor Current (A)	PWM Pulse Duration (μ s)
10S	24	42
20S	45	22
40S	80	12

The results indicate that at low motor speeds combined with large propeller pitch — typical of heavy multicopter operation — it is advantageous to use lower voltages. Longer PWM pulses reduce losses in the transition stages of MOSFET output stages and losses in the high-inductance stator windings. Consequently, this leads to lower heat losses.

To achieve higher rotational speeds of the propeller, it is necessary to increase the operating voltage, as low operating voltages often fail to reach the required rotational speeds. The achievable rotational speed of a BLDC motor can be calculated using the following formula:

$$RPM = (Q/6)/(N/2) \times 60 \quad (6)$$

where

RPM — propeller rotation frequency 1/min;

Q — PWM frequency, Hz;

N — Number of magnets in the rotor.

As with a robotic car gearbox, a certain level of hysteresis should be built into the algorithm to avoid undefined states at switching boundaries.

Below is the program code that explains how this algorithm works:

```
#define NUMBER_ROTOR      4
#define NUMBER_SWITCH     3
#define DELTA_SPEED      100

// Battery switch
int Battery_switch(void)
{
    int Mid_Current, Mid_Speed, Opt_Speed;
    int Battery mode = 1;
    Set_Battery_switch(Battery mode);

    while(Get_Stat_ESC() && (Get_High() > HIGH_TAKEOFF)) // Rotors RUN and the required
height has been reached
    {
        Mid_Current = 0;
        Mid_Speed = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Mid_Current = Mid_Current + Read_Current(Index);
            Mid_Speed = Mid_Speed + Read_Speed(Index);
        }
        Mid_Current = Mid_Current / NUMBER_ROTOR;
        Mid_Speed = Mid_Speed / NUMBER_ROTOR;

        Power = Read_Power();

        Opt_Speed = Read_AI_Speed(Mid_Current, Mid_Speed, Power);

        if(Opt_Speed > (Mid_Speed + DELTA_SPEED)) Battery mode++;
        if(Opt_Speed < (Mid_Speed - DELTA_SPEED)) Battery mode--;

        if(Battery mode > NUMBER_SWITCH) Battery mode = NUMBER_SWITCH;
        if(Battery mode < 1) Battery mode = 1;

        Set_Battery_switch(Battery mode);
    }
}
```

4.2. Stator Winding Switch Algorithm

This algorithm operates similarly to an automatic transmission in a car — except instead of shifting gears, it changes the stator winding configuration to optimize either torque or rotational speed. The optimal RPM ranges and switching thresholds are preprogrammed into the PC-MCU's internal memory and continuously monitored via tachometers (T).

A detailed description of multi-stator BLDC motor winding configurations and the electronic switching system design can be found in [4],[15].

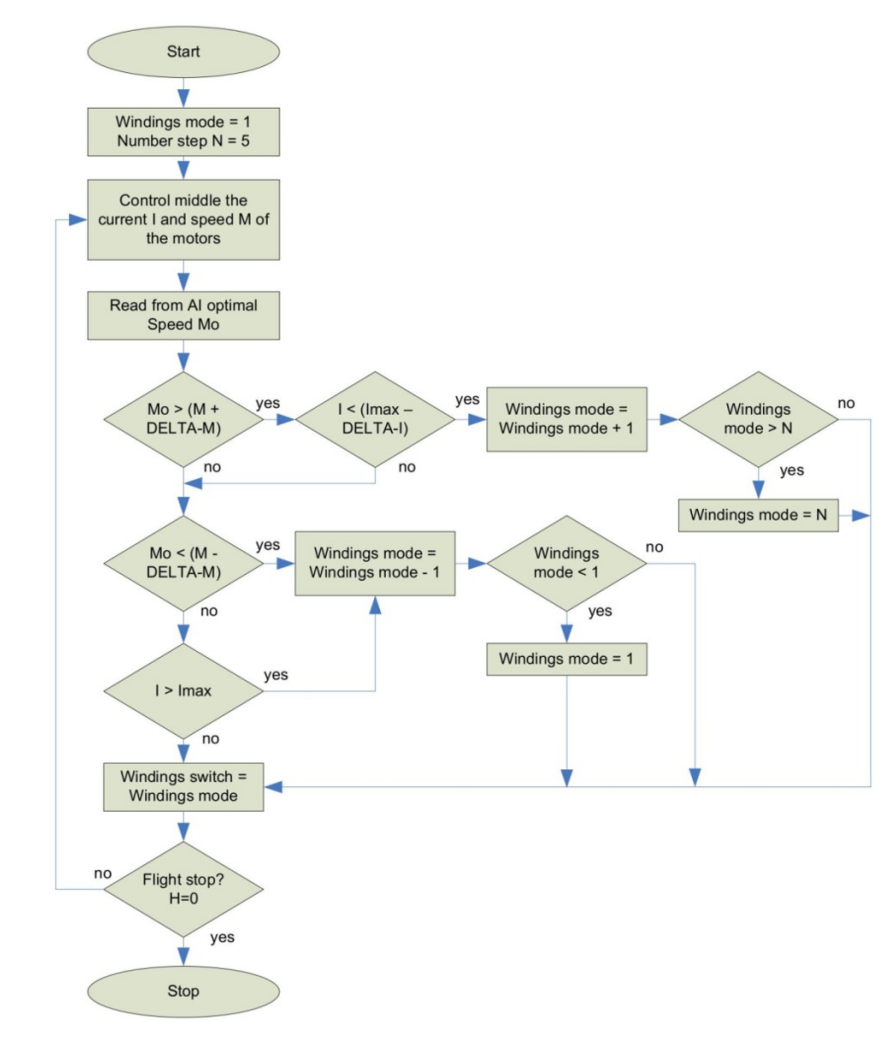


Figure 12 - Block Diagram of the Stator Winding Switching Algorithm

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.15>

In systems with conventional BLDC motors, the controller alternates between two standard winding configurations: “Star” and “Delta”. This switching enhances both motor thrust and energy efficiency. The electronic switch is lightweight and compact, requiring just nine switching **elements**.

The algorithm was tested on a simulation bench replicating a typical UAV environment. Throughout the experiment, voltage, motor speed, and propeller pitch were held constant. Only the winding configuration was changed — switching between “Star” and “Delta”. During the tests, current consumption and PWM pulse duration were measured.

The measurement results are summarized in Table 4.

Table 4 - Impact of Winding Configuration on Motor Current and PWM Pulse Duration

DOI: <https://doi.org/10.60797/IRJ.2026.172.56.16>

Motor Speed = const. 2,800 RPM	Motor Current (A)	PWM Pulse Duration (us)
Star	36	28
Delta	42	22

The results indicate that for low-speed operation with high propeller pitch — typical for heavy-lift multicopters — high-inductance winding configurations are more efficient. At higher RPMs, switching to a low-inductance configuration becomes preferable, as it supports higher currents and delivers greater power output.

However, if flight conditions (cruise speed, propeller load, RPM) permit, the system should favor the configuration with the lowest current draw to maximize battery efficiency.

The following code demonstrates the logic behind this switching algorithm:

```
#define NUMBER_ROTOR
#define NUMBER_SWITCH
```

4

5

```

#define DELTA_SPEED          100
#define DELTA_CURRENT        100
#define MAX_CURRENT          10000

// Windings switch
int Windings_switch(void)
{
    int Mid_Current, Mid_Speed, Opt_Speed, Max_Current;
    int Windings mode = 1;
    Set_Windings_switch(Battery mode);

    while(Get_Stat_ESC() && (Get_High() > HIGH_TAKEOFF)) // Rotors RUN and the required
height has been reached
    {
        Mid_Current = 0;
        Mid_Speed = 0;
        for(Index = 0; Index < NUMBER_ROTOR; Index++)
        {
            Mid_Current = Mid_Current + Read_Current(Index);
            if(Read_Current(Index) > (MAX_CURRENT + DELTA_CURRENT)) Max_Current =
1;
            if(Read_Current(Index) < (MAX_CURRENT - DELTA_CURRENT)) Max_Current =
0;
            Mid_Speed = Mid_Speed + Read_Speed(Index);
        }
        Mid_Current = Mid_Current / NUMBER_ROTOR;
        Mid_Speed = Mid_Speed / NUMBER_ROTOR;

        Power = Read_Power();

        Opt_Speed = Read_AI_Speed(Mid_Current, Mid_Speed, Power);

        if(Opt_Speed > (Mid_Speed + DELTA_SPEED) && !Max_Current) Windings mode++;
        if(Opt_Speed < (Mid_Speed - DELTA_SPEED) || Max_Current) Windings mode--;

        if(Windings mode > NUMBER_SWITCH) Windings mode = NUMBER_SWITCH;
        if(Windings mode < 1) Windings mode = 1;

        Set_Windings_switch(Windings mode);
    }
}

```

By combining the battery switching algorithm with the stator winding switching logic, a unified control algorithm can be developed — a comprehensive strategy that considers not only battery discharge characteristics and optimal BLDC motor speeds, but also the aerodynamic behavior of propellers at various RPMs and the transient response during topology switching.

If the switching thresholds for battery sections and stator winding configurations are properly decoupled and staggered, transient phenomena — particularly peak current spikes during transitions — can be significantly reduced.

Conclusion

In UAV systems, the term “power-to-weight ratio” typically refers to the relationship between available energy and total flight weight. The higher the power-to-weight ratio, the longer the UAV can remain airborne. It is generally expressed as:

$$Q/W, \quad (7)$$

where

Q — battery capacity;

W — UAV takeoff weight.

However, the integration of advanced BLDC systems into the UAV introduces an additional efficiency factor into this equation:

$$K \cdot Q/W, \quad (8)$$

where

K — efficiency coefficient of the BLDC system.

The more sophisticated the BLDC system — incorporating battery commutation, stator winding switching, and tightly integrated electronics — the higher its efficiency coefficient. As a result, the effective power-to-weight ratio increases, enabling longer flight durations even with the same battery capacity and takeoff weight, due to improved energy utilization. The slight weight gain from additional switching components is negligible, thanks to the compact and lightweight nature of modern electronics.

Naturally, the algorithms described in this paper are not the only ways to enhance UAV efficiency. Depending on specific UAV designs, additional software algorithms can be implemented, including:

- Point-to-Point Flight Optimization Algorithm.

This algorithm receives coordinates for the departure point (A) and destination (B), analyzes onboard battery reserves and UAV power characteristics, and dynamically adjusts flight parameters to either minimize energy consumption or maximize



travel speed. If available energy is insufficient to complete the planned route, the system alerts the operator and proposes alternative solutions.

- Stator Inductance Estimation Algorithm.

High-voltage BLDC windings often exhibit significant inductance. If PWM pulses are too short, energy is not effectively transferred and is lost as heat. This algorithm estimates the inductance of the stator windings and adjusts the minimum allowable PWM pulse width to minimize energy loss — a critical function in multi-stator BLDC motors, where inductance can vary depending on the selected winding configuration.

- PWM Pulse Width Adjustment Based on MOSFET Temperature.

The heating of power-stage MOSFETs is primarily driven by switching frequency. At high propeller RPMs, rapid switching can cause excessive heating and risk component failure. This algorithm monitors MOSFET temperature and, if a defined threshold is exceeded, evaluates current flight parameters and reduces propeller speed if necessary to avoid overheating — while notifying the operator of the intervention. This method can also be combined with the inductance estimation algorithm for further optimization.

Together, these software-driven algorithms and circuit-level optimizations allow UAV performance to be significantly improved without any hardware upgrades. In many cases, deploying newer versions of a UAV may only require a remote firmware update.

This level of adaptability lays the foundation for fully autonomous UAV logistics networks — enabling smart, reliable, and scalable aerial cargo delivery.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы на английском языке / References in English

1. Telli K. A comprehensive review of recent research trends on UAVs / K. Telli, O. Kraa, Y. Himeur [et al.] // arXiv. — 2023. — preprint arXiv: 2307. — Art. 13691. — DOI: 10.48550/arXiv.2307.13691.
2. Kim S.-C. Design and Analysis of BLDC Motor Driver for Hybrid Electric Vehicles / S.-C. Kim, N. Sangam, S. Pagidipala [et al.] // Part of the Lecture Notes in Electrical Engineering Book Series (LNEE). — Springer: Singapore, 2022. — Vol. 824. — DOI: 10.1007/978-981-16-7794-6_12.
3. Yaz M. Brushless Direct Current Motor Design and Analysis / M. Yaz, E. Ceti // COJ Electronics & Communications. — 2021. — № 2 (2). — DOI: 10.31031/COJEC.2021.02.000534.
4. Carev V. The Electronic Switch of Windings of a Standard BLDC Motor / V. Carev, J. Roháč, S. Tkachenko [et al.] // Applied Sciences. — 2022. — № 12 (21). — Art. 11096. — DOI: 10.3390/app122111096.
5. Suti A. Development and Experimental Validation of Novel Thevenin—Based Hysteretic Models for Li-Po Battery Packs Employed in Fixed-Wing UAVs / A. Suti, G. Di Rito, G. Mattei // Energies. — 2022. — № 15 (23). — Art. 9249. — DOI: 10.3390/en15239249.
6. Carev V. Electronic battery switch for UAV BLDC motor / V. Carev, K. Alloyarov, A. Kaychenov [et al.] // International Review of Aerospace Engineering IREASE. — 2024. — № 17 (1). — P. 12. — DOI: 10.15866/irease.v17i1.24555.
7. Kansa S. Battery Health Based Remaining Mission Time Prediction of UAV in Closed Loop / S. Kansa, M. Shekhar, S. JHA [et al.] // Conference: 2023 International Conference on Unmanned Aircraft Systems (ICUAS). — 2023. — DOI: 10.1109/ICUAS57906.2023.10155875.
8. Ermilov A.S. Application of Machine Learning for Adaptive Trajectory Control of UAVs Under Uncertainty / A.S. Ermilov, O.A. Saltykova // RUDN Journal of Engineering Research. — 2025. — Vol. 26. — № 1. — P. 7–16. — DOI: 10.22363/2312-8143-2025-26-1-7-16.
9. Mikołajczyk T. Energy Sources of Mobile Robot Power Systems: A Systematic Review and Comparison of Efficiency / T. Mikołajczyk, D. Mikołajewski, A. Kłodowski [et al.] // Appl. Sci. — 2023. — № 13. — P. 7547. — DOI: 10.3390/app13137547.
10. Üçgün H. The Comparison of Energy Sources Used in Unmanned Air Vehicles / H. Üçgün, U. Yüzgeç, M. Kesler [et al.] // People. — 2019. — № 5. — P. 30–38. — DOI: 10.7176/JSTR/5-6-05.
11. Zhanga C. A comprehensive review of electrochemical hybrid power supply systems and intelligent energy managements for unmanned aerial vehicles in public services / C. Zhanga, Y. Qiua, J. Chenb [et al.] // Energy AI. — 2022. — № 9. — Art. 100175. — DOI: 10.1016/j.egyai.2022.100175.
12. Townsend A. A comprehensive review of energy sources for unmanned aerial vehicles, their shortfalls and opportunities for improvements / A. Townsend, I.N. Jiya, C. Martinson [et al.] // Heliyon. — 2020. — № 6. — Art. e05285. — DOI: 10.1016/j.heliyon.2020.e05285.
13. Lei T. State of art on energy management strategy for hybrid-powered unmanned aerial vehicle / T. Lei, Z. Yang, Z. Lin // Chin. J. Aeronaut. — 2019. — № 32. — P. 1488–1503. — DOI: 10.1016/j.cja.2019.03.013.



14. Boukoberine M.N. A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects / M.N. Boukoberine, Z. Zhou, M. Benbouzid // *Appl. Energy*. — 2019. — № 255. — Art. 113823. — DOI: 10.1016/j.apenergy.2019.113823.
15. Carev V. Multilayer Brushless DC Motor for Heavy Lift Drones / V. Carev, J. Roháč, M. Šipoš [et al.] // *Energies*. — 2021. — № 14 (9). — P. 2504. — DOI: 10.3390/en14092504.
16. Dang G. Lithium Titanate Battery System Enables Hybrid Electric Heavy-Duty Vehicles / G. Dang, M. Zhang, F. Min [et al.] // *J. Energy Storage*. — 2023. — № 74. — Art. 109313. — DOI: 10.1016/j.est.2023.109313.
17. Nemeth T. Lithium Titanate Oxide Battery Cells for High-Power Automotive Applications – Electro-Thermal Properties, Aging Behavior and Cost Considerations / T. Nemeth, P. Schröer, M. Kuipers [et al.] // *J. Energy Storage*. — 2020. — № 31. — Art. 101656. — DOI: 10.1016/j.est.2020.101656.
18. Ermilov A.S. Methods of a Priori Statistical Analysis of Disturbed Motion of Aircraft in Turbulent Environments / A.S. Ermilov, O.A. Saltykova // *RUDN Journal of Engineering Research*. — 2024. — Vol. 25. — № 4. — P. 348–356. — DOI: 10.22363/2312-8143-2024-25-4-348-356.
19. Pourrahmani H. Unmanned Aerial Vehicles: The Possibility of Using Fuel Cell, Battery, and Turbocharger Technologies / H. Pourrahmani, N. Akbari, J.V. Herle // *ECS Meeting Abstracts MA2023*. — 2023 — № 01 (1). — P. 416–416. — DOI: 10.1149/MA2023-011416mtgabs.
20. Carev V. Power high frequency driver of ESC controller UAV / V. Carev, K. Alloyarov, A. Kaychenov // *International Review of Aerospace Engineering (IREASE)*. — 2024. — Vol. 2. — № 1. — DOI: 10.15866/irease.v17i3.24923.
21. Rødal G.L. Gate-Drive Circuits for Adaptive Operation of SiC mosfets / G.L. Rødal, D. Pefitsis // *IEEE Trans. Power Electron.* — 2024. — № 39. — P. 8162–8186. — DOI: 10.1109/TPEL.2024.3382335.
22. Alves L.F.S. Review on sic-mosfet devices and associated gate drivers / L.F.S. Alves, P. Lefranc, P. Jeannin [et al.] // *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 20–22 February 2018. — Lyon, 2018. — P. 824–829. — DOI: 10.1109/ICIT.2018.8352284.
23. Wu Y. Evolution, Challenges and Applications of Modern MOSFETs / Y. Wu // *Applied and Computational Engineering*. — 2023. — № 24 (1). — P. 294–301. — DOI: 10.54254/2755-2721/24/20230724.
24. Huh N. Hybrid PWM Control for Regulating the High-Speed Operation of BLDC Motors and Expanding the Current Sensing Range of DC-link Single-Shunt / N. Huh, H.-S. Park, M.H. Lee [et al.] // *Energies*. — 2019. — № 12. — P. 4347. — DOI: 10.3390/en12224347.
25. Saif E. Hybrid Power Systems in Multi-Rotor UAVs: A Scientific Research and Industrial Production Perspective / E. Saif, 'I. Emino'glu // *IEEE Access*. — 2022. — № 1. — P. 438–458. — DOI: 10.1109/ACCESS.2022.3232958.
26. Bottaro E.A. Circuit simulation-based comparison of power electronics devices in a five-level converter for UAV applications / E.A. Bottaro, C. Ventura, S.A. Rizzo // *AIMS Energy*. — 2024. — № 12 (4). — P. 905–923. — DOI: 10.3934/energy.2024042.
27. Palma M. GaN Advantage over MOSFET in Inverters for Drones. An Experimental Comparison / M. Palma, F. Unnia, M. De Rooij // *Conference: 2024 IEEE 10th International Power Electronics and Motion Control Conference (IPEMC2024–ECCE Asia)*. — 2024. — DOI: 10.1109/IPEMC-ECCEAsia60879.2024.10567278.
28. Martínez-Heredia J.M. Development of GaN TechnologyBased DC/DC Converter for Hybrid UAV / J.M. Martínez-Heredia, F. Colodro, J.L. Mora-Jiménez [et al.] // *IEEE Access*. — 2020. — № 8. — P. 88014–88025. — DOI: 10.1109/ACCESS.2020.2992913.
29. Shu J. Gate Driver Design for SiC Power MOSFETs With a Low—Voltage GaN HEMT for Switching. Loss Reduction and Gate Protection / J. Shu, J. Sun, Z. Zheng [et al.] // *IEEE Trans. Power Electron.* — 2024. — № 39. — P. 5558–5566. — DOI: 10.1109/TPEL.2024.3353460.
30. Bottaro E. Circuit Models of Power MOSFETs Leading the Way of GaN HEMT Modelling — A Review / E. Bottaro, S.A. Rizzo, N. Salerno // *Energies*. — 2022. — № 15. — P. 3415. — DOI: 10.3390/en15093415.
31. Mohanraj D. A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications / D. Mohanraj, R. Arulavid, R. Verma [et al.] // *IEEE Access*. — 2022. — № 10. — P. 54833–54869. — DOI: 10.1109/ACCESS.2022.3175011.
32. Selema A. Electrical Machines Winding Technology: Latest Advancements for Transportation Electrification / A. Selema, M.N. Ibrahim, P. Sergeant // *Machines*. — 2022. — № 10. — P. 563. — DOI: 10.3390/machines10070563.
33. Zouzou N. Modeling and Experimental Characterization of Nickel—Coated Copper Wires for the Design of Extremely High-Temperature Electrical Machines / N. Zouzou, T. Dang, S. Duchesne [et al.] // *IEEE Trans. Magn.* — 2020. — № 56. — P. 1–9. — DOI: 10.1109/TMAG.2019.2961063.
34. Carev V. Modernization of BLDC Motors for UAVs / V. Carev, J. Roháč, S. Tkachenko [et al.] // *International Review of Aerospace Engineering (IREASE)*. — 2023. — Vol 16. — № 1 (2023). — DOI: 10.15866/irease.v16i1.23087.
35. Choi J. Enhanced Efficiency of the Brushless Direct Current Motor by Introducing Air Flow for Cooling. Heat Mass Transf / J. Choi, J.H. Lee, Y.G. Jung [et al.] // *Und Stoffuebertragung*. — 2020. — № 56. — P. 1825–1831. — DOI: 10.1007/s00231-020-02827-8.
36. Mazur M. Heat Exchange Analysis of Brushless Direct Current Motors / M. Mazur, W. Skarka, M. Kobielski [et al.] // *Energies*. — 2024. — № 17. — P. 6469. — DOI: 10.3390/en17246469.
37. Safril S. Design of Cooling System on Brushless DC Motor to Improve Heat Transfers Efficiency / S. Safril, M. Mustofa, M. Zen [et al.] // *Evergreen*. — 2022. — № 9. — P. 584–593. — DOI: 10.5109/4794206.
38. Guo J. Research on the influence of thermal expansion of steel shaft on dynamic characteristics of full ceramic bearing-rotor system / J. Guo, Y. Wu, X. Zhang [et al.] // *Advances in Mechanical Engineering*. — 2022. — № 14 (7). — Art. 168781322211093. — DOI: 10.1177/16878132221109349.



39. Podsedkowski M. Variable Pitch Propeller for UAV-Experimental Tests / M. Podsedkowski, R. Konopinski, D. Obidowski [et al.] // *Energies*. — 2020. — № 13. — P. 5264. — DOI: 10.3390/en13205264.
40. Carev V. Automatic Adjustment of the Multicopter Propeller Pitch / V. Carev, K. Alloyarov, A. Kaychenov [et al.] // *International Review of Aerospace Engineering (IREASE)*. — 2024. — № 17 (2). — P. 60. — DOI: 10.15866/irease.v17i2.25045.