**PROBLEMY MECHATRONIKI**
**UZBROJENIE, LOTNICTWO, INŻYNIERIA BEZPIECZEŃSTWA**

**ISSN 2081-5891**
**E-ISSN 2720-5266**

**13, 3 (49), 2022, 9-26**

**PROBLEMS OF MECHATRONICS**
**ARMAMENT, AVIATION, SAFETY ENGINEERING**

# Stochastic Movement Swarm Performing a Coverage Task with Physical Parameters

Krzysztof FALKOWSKI*, Michał DUDA

*Military University of Technology,*
*Faculty of Mechatronics, Armament and Aerospace,*
*2 Sylwestra Kaliskiego Str., 00-908 Warsaw, Poland*
**Corresponding author's e-mail address and ORCID*:
*krzysztof.falkowski@wat.edu.pl; https://orcid.org/0000-0002-0279-3791*

*Received: August 2, 2021. / Revised: August 30, 2021 / Accepted: July 22, 2022 / Published: September 30, 2022.*

DOI 10.5604/01.3001.0016.0047

**Abstract.** This paper describes an attempt of implementing physical parameters into a virtual swarm algorithm solution. It defines which physical parameters of the single object need to be known to properly transfer a virtual algorithm into a physical system. Considerations have been based on a stochastic movement swarm performing a coverage task. Time to finish the task and energy consumptions were measured for different numbers of drones in a swarm allowing to designate an optimal size of the swarm. Additional tests for changing variables allowed us to determine their impact on the swarm performance. The presented algorithm is a discrete-time solution, and every test is divided into steps. Positions of the drones are calculated only in time corresponding to these steps. Their position is unknown between these steps and the algorithm does not check if the paths of two drones cross between subsequent positions. The lower the time interval, the more precise results, but simulating the test requires more computing power. Further work should consider the smallest possible time intervals or additional feature to check if the paths of the drones do not cross.

**Keywords:** swarms, swarm algorithm, sweep coverage, coverage task, optimisation, stochastic movement

## 1. INTRODUCTION

Groups of drones can solve complex tasks even though they compose of simple objects cooperating without any central control, acting due to local interactions [1, 2]. The system composed of many simpler robotic objects tends to achieve better results than one highly advanced robot alone. This comes forward especially in the case of tasks that require spreading over a given area such as mapping or environmental monitoring [3].

Applications that require an entire research area to be covered or explored are called Coverage Tasks [4]. Swarm algorithms offer some advantages over a single object or human solutions. Swarms perform faster and more accurately. Also, they are tolerant to the failure of a single participant [5]. Furthermore, they are preferably used when the task area is dangerous or might not be accessed by humans [6].

A recent increase in interest in coverage tasks resulted in the rapid development of many different solutions. Generally, they might be divided into direct and stochastic methods. In direct methods, the movement of objects in the swarm is somehow defined when in stochastic methods they move randomly.

In direct solutions, one approach is fore planning the path of the swarm. One such solution was presented by Englot [7]. Knowing the task area in advance, it might be divided into samples. Then, it is possible to search for optimal configuration of the robots in the task giving the best results.

Another approach assumes that the task area is unknown and the search is carried out based on dynamic decisions made according to ongoing findings. As described by Soto [8] this might be achieved by programming more than one searching algorithm and by implementing a decisive algorithm that allows us to shift the strategy depending on a progressing situation. The other solutions might be based on decision coordination, where robots behave depending on their relative positions and integration of collected data [5, 9].

Direct methods achieve good results; however they require more advanced objects and a certain amount of computational complexity [6]. Solutions opposite to these ones are stochastic methods in which random movement of the drones is assumed and hence much simpler objects are sufficient.

One type of stochastic solutions is statistical optimisation methods. As presented by Kumar [10], stochastic movement might be described by probability distribution using a variety of probabilistic and geometric tools. This allows designating of statistical properties of multi-robot swarms in stochastic coverage tasks. Another statistical solution was described by Ayvali [11] and it is called Cross-Entropy Method. It allows optimising the parameters of the trajectory of stochastically moving objects in coverage tasks.

The other method to evaluate stochastic solutions are simulations which allow designating the outcome by simulating the same scenario repeatedly and drawing the average results. Such an approach was presented by Yanmaz [12] to confirm the statistical solution or by Zhang [13] to simulate a swarm of micro aerial robots that were pollinating a crop field.

Although all methods, described above, provide results in the behaviour of the swarms and optimisation of their performance, these papers consider only theoretical solutions and virtual swarms. None of them tries to raise the issue of real objects' system and how would the proposed algorithm work when implemented in a real solution. Physical parameters of the swarm, such as dimensions, mass, inertia, or energy consumption are neglected. In our work, we present a simulation of a swarm composed of physical objects and we try to designate an impact of varying physical parameters on the results.

This is the second stage of the project. In its first part [14] there was simulated a swarm of stochastically moving boids as virtual points. In this paper, the main assumptions are the same, but the algorithm was improved by adding physical parameters and motion dynamics involving mass, acceleration, kinetic energy, or energy required to keep flying objects aloft.

## 2. RESEARCH OBJECTIVES AND ASSUMPTIONS

This article describes a subsequent version of the authorial Swarm Algorithm. It simulates searching the task area with a swarm of randomly moving drones using the sweep coverage method [4].

The main assumptions are the same. The algorithm simulates a two-dimensional, circular research area with a given radius. Then, a group of a given number of boids is set up over it. They move randomly over the area providing collision avoidance. The movement of the drones is not determined, and it is independent for every drone separately. They constantly change the direction in which they move and the only limit to their movement is that they may not leave the task area. The time and energy required to finish the task are counted and provided as a result of the test.

The original version of the algorithm did not take into account any physical parameters and this solution was supplemented with the following changes:

- certain physical form of the boid was assumed, which means in this solution it had a mass and some dimensions necessary for this consideration. Involving mass also means taking inertia into account;
- this algorithm is intended for quadcopter drones and hence boids were assumed to be flying. This means that energy consumed by a single drone is divided into energy required for hover the drone in the air and kinetic energy required for its horizontal movement.

Although considering a physical form of the drones, momentums of inertia were neglected. All moving objects were assumed to have only two degrees of freedom, which were linear translations along two perpendicular axes on a test plain. Also, for the movement, each object was treated as a point of mass.

The goal of this work was to simulate a stochastic, physical swarm performing a coverage task and to investigate a process of transferring a virtual solution to a real system. Furthermore, parameters that can indicate optimal configurations of the swarm were designated and the performed comparative tests allowed us to determine an impact of varying input parameters on the results.

## 3. SWARM ALGORITHM DESCRIPTION

The presented solution is a discrete-time algorithm, and every test is divided into successive iterations. Simulating physical behaviour required setting up a time scale so, it was assumed that every iteration in simulation corresponds to a one-second interval.

Furthermore, to simulate the dynamics of movement of drones, the following physical parameters have been taken into consideration:

- Acceleration ($a$) – drones do not accelerate to the maximal velocity instantly, but it requires some time. The rate of changing velocity over subsequent iterations is described by acceleration.
- Maximal velocity ($V_{max}$) – after reaching this value drones do not accelerate any further, but they move at constant velocity.
- Mass ($m$) – every boid has its assigned mass. This implies the weight of the drone and the energy required to move the mass horizontally.
- Rotor blade radius ($R_B$) – describes dimensions of a single fan of a quadrotor. This value is necessary to evaluate the power required to fly at constant altitude.
- Altitude ($H$) and implied by its density of air ($\rho$) taken from International Standard Atmosphere.
- Standard gravity ($g$) – which is required to calculate the weight of a flying drone.

At this stage of the research, there was no real-life object involved so, not all dimensions were known and those required were assumed. What follows for purposes of simulation, every drone was treated as a point of mass.

In every iteration, each drone tried to accelerate in a random direction unless it was already moving with a maximal velocity. The safe distance ($d_{safe}$) was set up. If a drone would step closer to another object, then this distance it stopped. The implemented algorithm was modelled on the one presented in an article [14] written by the authors of this paper.

In the case of stopping a drone, a simplification has been made, that the time for stopping completely is always one iteration regardless of the velocity that the drone was moving with when it started to deaccelerate.

The movement of boids depends on their current velocity and on the given acceleration. Assuming that the step is safe, and a drone is not stopping due to safe distance violation in iteration "i" it moves a distance equal to:

$$\Delta s_i = V_{i-1} \cdot \Delta t + \frac{a \cdot \Delta t^2}{2} \tag{1}$$

unless it is moving with $V_{max}$. Then, the distance equals:

$$\Delta s_i = V_{max} \cdot \Delta t \tag{2}$$

where $\Delta t$ is 1-second iteration time.

The energy consumption of a single drone is calculated by dividing it between the energy required to hover at a given height and to kinetic energy required to accelerate, move, or deaccelerate.

As proven by Lopez [15] and Rotaru [16], the energy required to hover a helicopter or quadrotor is dependent on air density ($\rho$), the area swept by rotors ($A$), and weight ($M$) of the machine itself. Maintaining a constant height can be achieved by delivering the constant power $P$ to the rotors which equals:

$$P = \sqrt{\frac{M^3}{2\rho A}} = \sqrt{\frac{(mg)^3}{2\rho A}} \tag{3}$$

where $A$ for quadrotor is the area swept by all four fans:

$$A = 4\pi R_B^2 \tag{4}$$

Air density was drawn from International Standard Atmosphere as it is dependent on Height of flight using linear interpolation on values from a table [17].

In equation (3), the power was given in Watts. Energy consumed in the test for hovering a single drone equals the power ($P$) multiplied by the time of the test ($T_I$). The total energy $HE$ used in an attempt to maintain the flight of the swarm was the energy required to hover a single drone multiplied by the number of drones ($D$):

$$HE = P \cdot T_I \cdot D \tag{5}$$

As shown by Reid [18], the power required for horizontal movement of the quadcopter is complicated to estimate as it depends on many different variables. Because at this stage no physical dimensions of the drone were known, another simplification was made.

The energy required to move the drone horizontally was calculated by standard kinetic energy equations. During accelerating in every iteration "i" drone required the energy equal to:

$$\Delta KE_i = \frac{m \cdot \left(V_i^2 - V_{i-1}^2\right)}{2} \tag{6}$$

during movement with $V_{max}$, this energy equals:

$$\Delta KE_i = \frac{m \cdot V_{max}^2}{2} \tag{7}$$

Stopping the quadcopter is also a complex process as it requires the machine to counteract the velocity. According to the assumption that momentums of inertia were neglected, this process was simplified, and it was assumed that energy spent to stop the movement depends on the velocity that the drone was moving with, before starting the deacceleration. It equals:

$$\Delta KE_i = \frac{m \cdot V_{i-1}^2}{2} \tag{8}$$

In every case of horizontal movement, any resistance of motion was neglected. Movement energy was also calculated in Joules per second. The total kinetic energy *KE* was calculated by summing up the partial components Δ*KE* consumed through the whole test by all the drones together.

The goal of every test was to cover 90% of a research area of the given radius $R_{area}$. In every iteration, every drone scanned the area of the certain radius $R_{scan}$ around it. Due to the assumption that drones are flying, this value was dependent on a height with a linear proportion as shown in Fig. 3.1.

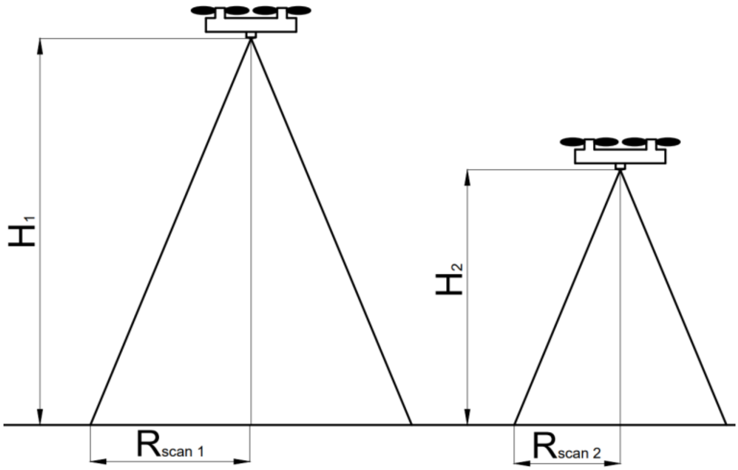


Fig.3.1. Dependance of the radius of scaned area ($R_{scan}$) of the height of flight (*H*)

# 4. TESTS

To evaluate an impact of different changing variables, a series of tests were carried out. Due to the stochastic movement of drones, for every result at least 10 tests were taken and then, an average outcome was drawn. The following values were constant during all of the tests:

Table 4.1. Parameters with constant values

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Standard gravity | $g$ | 9.81 | $ms^{-2}$ |
| Mass of the drone | $m$ | 1 | kg |
| Safe distance | $d_{\mathrm{safe}}$ | 10 | m |
| Maximal velocity | $V_{\mathrm{max}}$ | 10 | $ms^{-1}$ |
| Rotor blade radius | $R_{\mathrm{B}}$ | 0.2 | m |

For every set of parameters, series of tests were carried out for groups of $D$ = [1, 2, 3, 4, 5, 7, 10, 15, 25, 50, 100, and 150] drones. For every swarm size ($D$), the following values were counted and provided as results:

- the number of iterations ($T_{\mathrm{I}}$) required to finish the task. This allowed determining the duration of the attempt in seconds;
- the number of emergency stops ($S_{\mathrm{E}}$) during the attempt. This value shows how many steps were wasted due to the too close fly situation;
- the swarm effectiveness ($\varphi_{\mathrm{E}}$) allows showing what part of steps was used to move the drones and what part was consumed by stopping. It was calculated with the following formula:

$$\varphi_E = 1 - \frac{S_E}{T_I \cdot D} \tag{9}$$

- the total energy ($TE$) required to perform the task. It is the sum of energy ($HE$) required to hover the drones and the kinetic energy ($KE$) consumed to move the drones horizontally;
- the performance indicator ($\mu_{\mathrm{TE}}$) is a product of multiplying the total time ($T_{\mathrm{I}}$) and the total energy ($TE$). This allowed designating which drone count provides the best compromise between the shortest time to finish the task and the least energy spent to do so. Therefore, optimal solutions could be found.

The main attempt was performed fifty times for each group to obtain stable reference results which could be compared to the results of other tests. It was carried out with the following set up of variables shown in Table 4.2. Then, further tests were carried to designate an impact of changing those variables on the result. While one of the variables was tested, all the others were constant.

The only exception is the height of flight and the scanned area radius as they are depended on each other.

Table 4.2. Variables for the reference attempt

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Acceleration | $a$ | 2.5 | $m \cdot s^{-2}$ |
| Height of flight | $H$ | 30 | m |
| Air Density | $\rho$ | 1.198 | $kg \cdot m^{-3}$ |
| Power required to hover | $P$ | 27.9 | W |
| Scanned Area radius | $R_{scan}$ | 5 | m |
| Research Area radius | $R_{area}$ | 150 | m |

## 5. RESULTS ANALYSIS

The results of the first test were shown in Table 5.1 and Figures 5.1-5.3. Later, in the article, these results will serve as a point of reference allowing to designate an impact of changing variables on the results.

Table 5.1. Reference results

| Number of drones [-] | 1 | 2 | 3 | 4 | 5 | 7 | 10 | 15 | 25 | 50 | 100 | 150 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time [s] | 2191 | 1138 | 779 | 605 | 504 | 367 | 279 | 209 | 143 | 96 | 70 | 61 |
| Collision avoidance stops $E_S$ [-] | 0 | 42 | 95 | 123 | 175 | 218 | 329 | 535 | 821 | 1613 | 3111 | 4724 |
| Total energy TE [kJ] | 170.7 | 172.3 | 172.3 | 174.1 | 176.3 | 174.5 | 177.8 | 182.4 | 182.5 | 199.6 | 235.8 | 282.4 |
| Percentage of total energy used to hover [%] | 36% | 37% | 38% | 39% | 40% | 41% | 44% | 48% | 55% | 68% | 83% | 90% |
| Swarm effectiveness $\varphi_E$ [%] | 100% | 98% | 96% | 95% | 93% | 92% | 88% | 83% | 77% | 66% | 55% | 48% |
| Performance indicator $\mu_{TE}$ [MJ·s] | 374.1 | 196.1 | 134.2 | 105.3 | 88.9 | 64.1 | 49.7 | 38.1 | 26.1 | 19.2 | 16.5 | 17.1 |

As shown in Fig. 5.1, the time required to finish the task decreases exponentially with an increase in the number of drones in the swarm. On the other hand, the more drones in the area, the more collision avoidance stops, and they are rising linearly.

Figure 5.2 shows that total energy consumption rises linearly along with a rise of a number of drones in the swarm. Furthermore, the more drones in the group, the greater fraction of energy is consumed to keep the drones hovering. For a group of 25 objects, it is more than 50% and for the largest group, almost 100% of the total consumed energy was used for hovering. This parameter is combined with collision avoidance stops. When the drone does not move due to safe distance violation, it still consumes the energy required to hover.

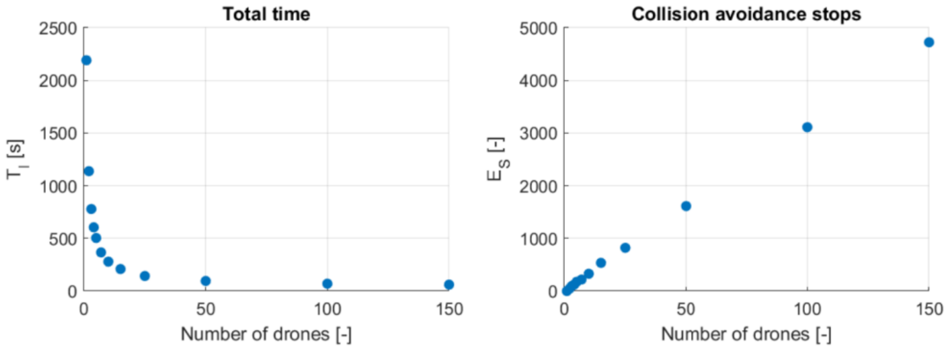


Fig 5.1. Time and number of collsion avoidance stops in reference tests

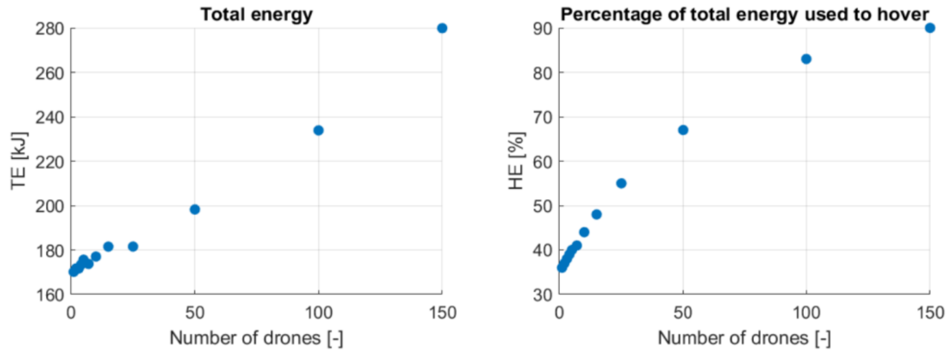


Fig. 5.2. Total energy and percentage of total energy consmed to hover in reference tests

As shown in Fig. 5.3 and Table 5.1, a performance indicator has its minimum for a group of 100 drones. This means that below that number even though the total energy consumption was lesser, the time required to perform the task was of greater value. For more numerous groups, the rise in the total energy was unequally greater than a decrease in the time. This means that the optimal solution for a given set of parameters is a group of 100 drones.

Swarm effectiveness shows that the more numerous groups, the less efficient it is. As long as a group of 100 drones provides the best balance between time and energy, it can be seen that almost 50% of energy was wasted. Therefore, for a greater effectiveness and still considerably short time of test, the groups of 7 or 10 drones provide better results, as their effectiveness is about 90% while the total time is significantly lesser than for less numerous groups.

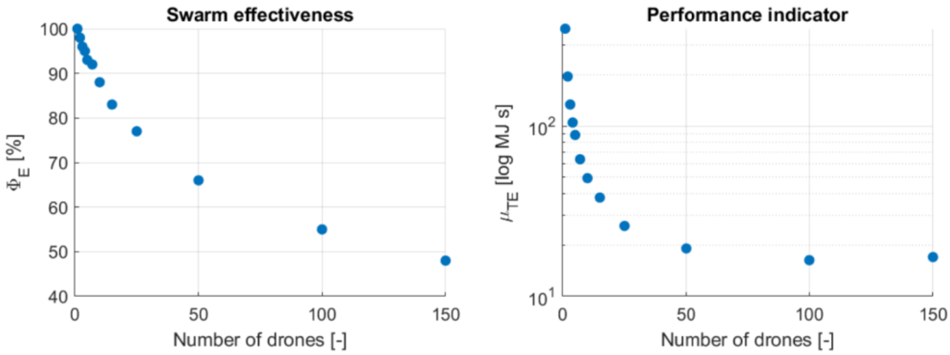


Fig. 5.3. Swarm effectiveness and performance indicator in reference tests

## 5.1. Tests for different values of research area radius

First comparative tests were carried out for varying research area radius. The tests were performed for $R_{area}$=[100, 150, 200] m. Figure 5.4 shows that the bigger the area, the more time is required to finish the task. What follows, more energy is required but the number of collision avoidance stops is nearly the same in all instances. More space allows the drones to move more freely and hence, swarm effectiveness rises. More energy is consumed for horizontal movement and due to the longer trials, energy consumed for hovering the drones also increases. Though, the increase in kinetic energy is greater as percentage of energy consumed to hover decreases. As it can be observed in Table 5.2, minimum of the performance indicator $\mu_{TE}$ moves towards the larger groups along with rising research area radius. This minimum means that for larger groups even though the time is lesser, the total consumed energy is incomparably higher than for smaller groups and the swarm is energy inefficient. The minimums were highlighted in Table 5.2.

Table 5.2. Performance indicator for varing $R_{area}$.

| | Number of drones [-] | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $R_{area}$ [m] | **1** | **2** | **3** | **4** | **5** | **7** | **10** | **15** | **25** | **50** | **100** | **150** |
| **100** | 76.9 | 40.3 | 30.0 | 24.4 | 19.1 | 17.1 | 13.4 | 9.7 | 8.4 | **7.5** | 8.4 | 11.9 |
| **150** | 374.1 | 196.1 | 134.2 | 105.3 | 88.9 | 64.1 | 49.7 | 38.1 | 26.1 | 19.2 | **16.5** | 17.1 |
| **200** | 1165.5 | 602.8 | 402.2 | 338.9 | 253.8 | 193.3 | 145.7 | 96.3 | 72.3 | 45.2 | 32.9 | **30.0** |

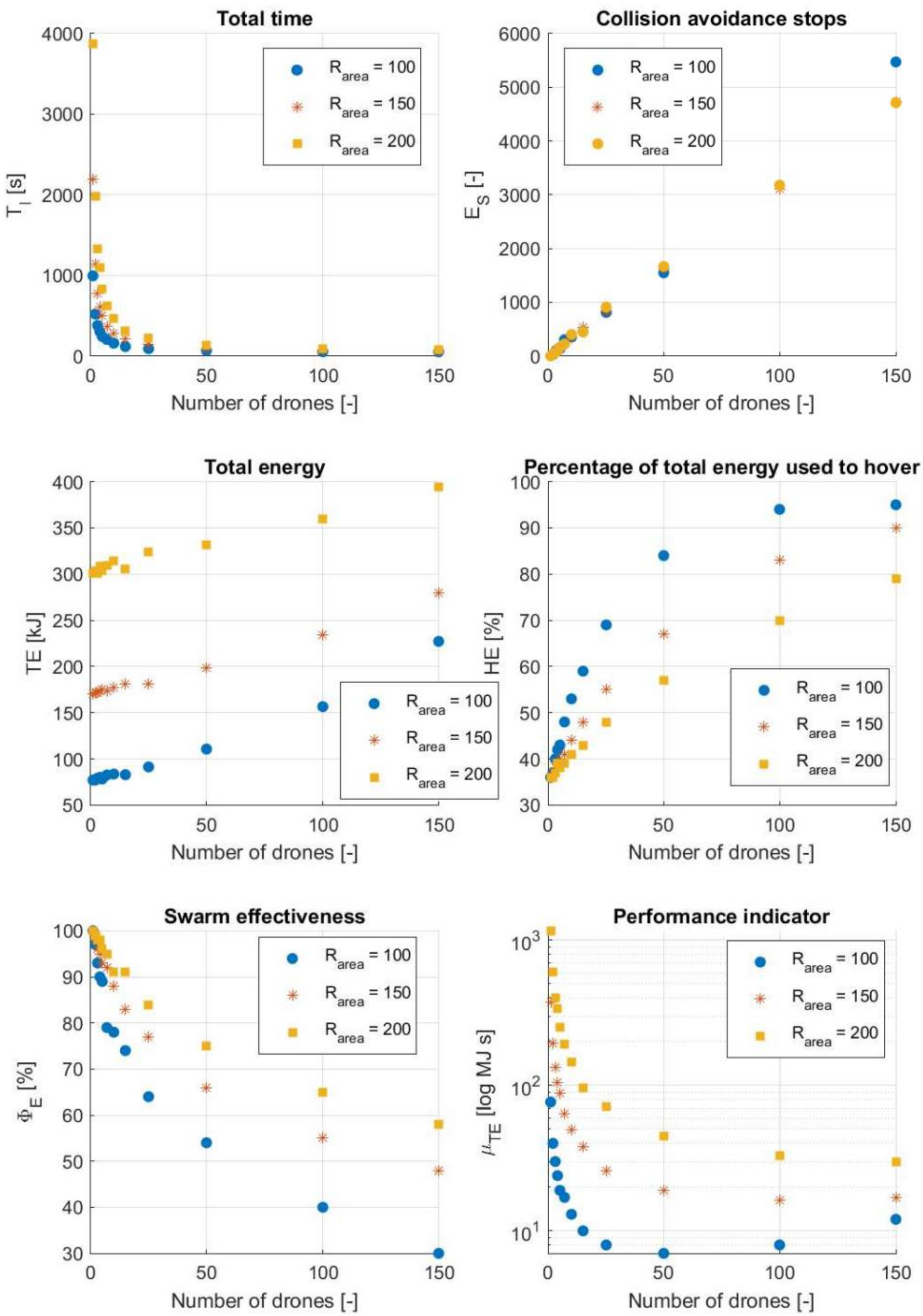


Fig. 5.4. Results of comparative tests for varying research area radius

## 5.2. Tests for different values of acceleration

Secondly, an impact of varying acceleration was verified. For $R_{area}$ = 150 m, and other parameters set to the same constant value, a series of tests were carried out for $a$ = [2; 2.5; 3.33] ms$^{-2}$. The results are presented in Figs. 5.5 and 5.6.

As it can be seen, varying acceleration has no major impact on the results. Mostly, all the results are the same and if they differ, the difference is within the limits of a statistical error.

The only trend might be observed on the swarm effectiveness plot. For high numerous swarms, the bigger the acceleration, the lower the effectiveness of the swarm. In more numerous groups, the higher acceleration means the greater chance to step within a collision avoidance zone of the other drone, therefore the effectiveness decreases.

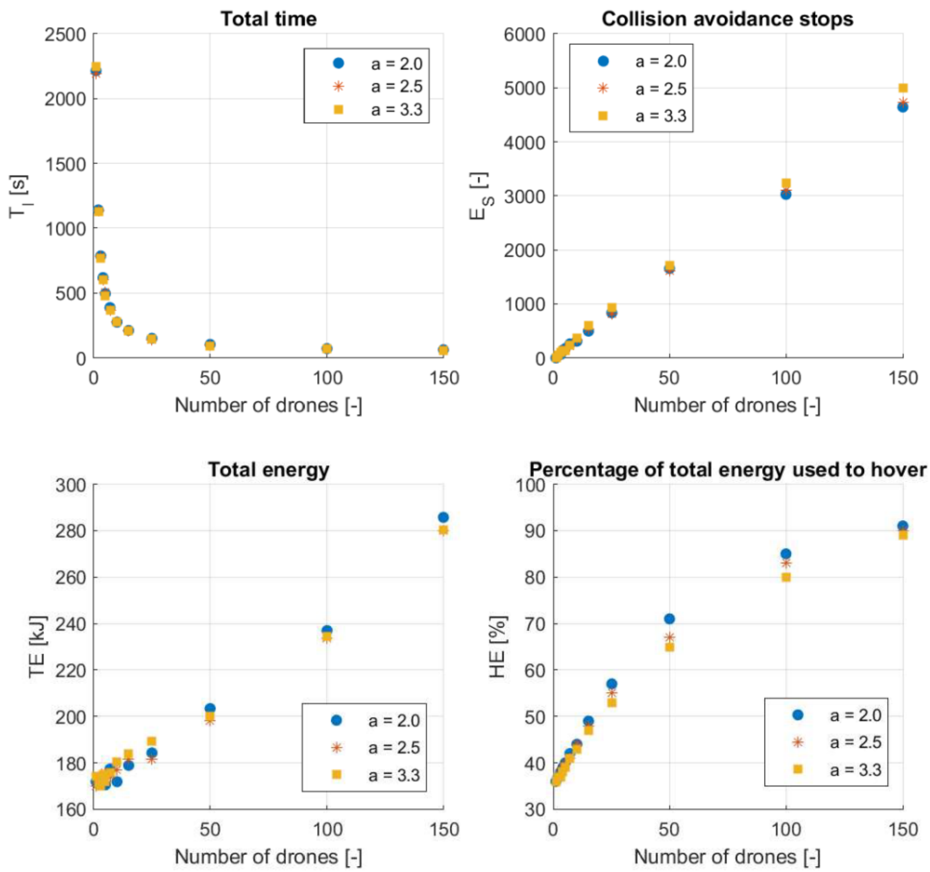


Fig. 5.5. Results of comparative tests for varying acceleration

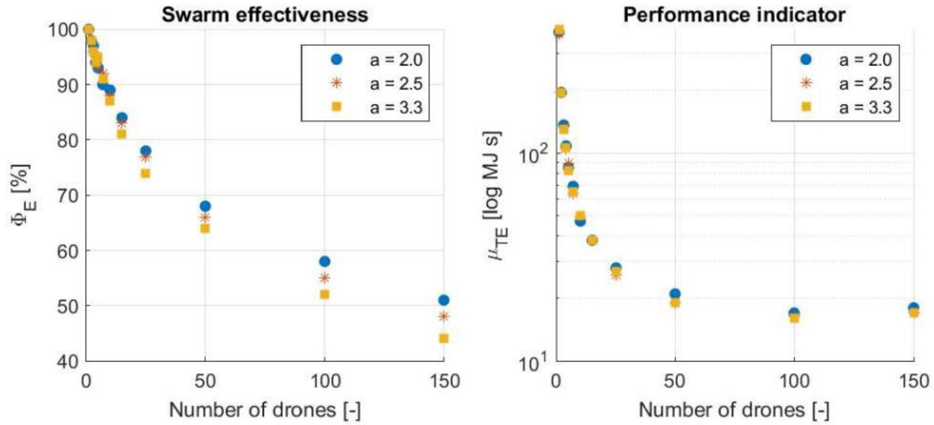


Fig. 5.6. Results of comparative tests for varying acceleration

## 5.3. Tests for different values of altitude

As it can be seen in all presented so far results, a great fraction of total energy is consumed by the energy required to hover the drone at a given altitude. This energy is altitude dependent so, it has been theorized that reducing the height of flight might reduce the total energy gradually. As mentioned earlier, the lower altitude means the higher air density, but it also comes with the smaller radius of the area scanned by the drone. The comparative tests were carried out for $H$ = 20 m. Table 5.3. below compares the parameters of the tests:

Table 5.3. Parameters for varying height of flight

| Tests | $H$ [m] | $\rho$ [kgm$^{-3}$] | $P$ [W] | $R_{scan}$ [m] |
|---|---|---|---|---|
| Comparative | 20 | 1.207 | 27.89 | 3.33 |
| Main | 30 | 1.198 | 27.99 | 5.00 |

As shown in Table 5.3, the decrease of $R_{scan}$ is significant when an increase in $\rho$ with a reducing altitude is almost negligible. Therefore, $P$ is the same for both altitudes. The results of the tests were shown in Fig. 5.7.

As it can be seen, total time and total energy increased significantly when the percentage of energy used to hover and the percentage of swarm effectiveness are almost the same. The decrease in power required to hover is too small compared to much smaller area scanned by a drone. The lower $R_{scan}$ causes the greater time required to finish the task when the energy consumed every second is nearly the same.

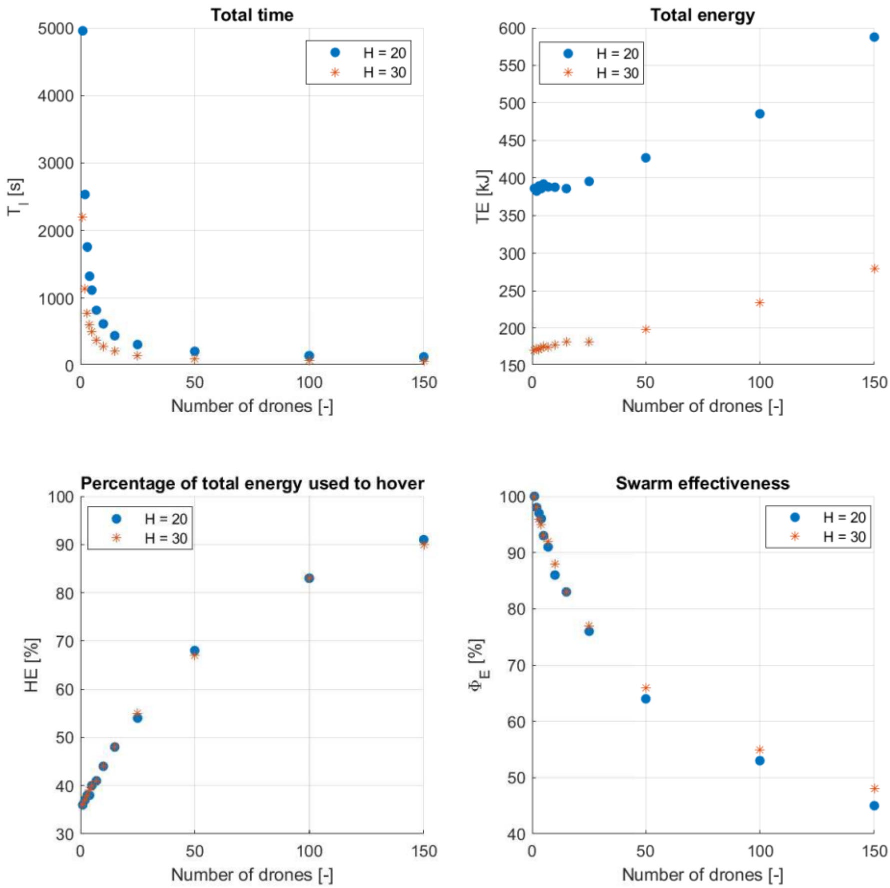


Figure 5.7. Results of comparative tests for varying altitude

## 6. CONCLUSIONS

An approach to a stochastic movement swarm with physical parameters was presented. The algorithm allows designating an optimal number of objects in the swarm according to varying physical parameters.

The presented work shows how complicated it is to transform an algorithm prepared on virtual simulations to work for a real, physical system. Many variables must be taken into consideration and a great part of them is very problematic to designate. Therefore, the results provided by a real swarm might have a great differentiation from those obtained in virtual simulations.

Although the presented algorithm provides certain results, it is still highly undeveloped as many crucial parameters were neglected. This can be seen by many simplifications assumed in this work.

Object dimensions might only be implemented to the algorithm after examining the resulting resistance of motions in an aerodynamic tunnel. Furthermore, the mechanics of motion for quadcopter is a complex process, especially while decelerating or rotating and hence, separate studies should be carried out on this subject.

Results of tests for a single set up of parameters show, that the rising number of drones in a group comes with positive and negative effects. The more numerous groups, the time to finish the task lowers exponentially. But the consequence of the larger group is that the safe distance is violated more frequently which means the bigger part of the drones does not move horizontally while they still have to hover. This means that for bigger groups the fraction of energy required to hover is rising and swarm effectiveness is lowering. That is why the performance indicator ($\mu_{\mathrm{TE}}$) was calculated, which is a product of multiplying the total time ($T_{\mathrm{I}}$) and the total energy ($TE$) that allows designating of optimal solutions between the shortest possible time and the least consumed energy. The results show that for every group's count, there can be found a minimum of that parameter which indicates an optimal solution for given conditions of the test.

Comparative tests showed that the acceleration of a single object has a minor impact on the performance of the whole swarm. In the case of height, it was proven, that as long as the area scanned by a single drone is bigger with a rising altitude, the swarm should fly the highest possible as the power required to hover the drones rising insignificantly with a rising height of flight. Though individual research should be conducted for scanners to find a real dependence of height and their performance.

It was possible to develop those results because of the assumed simplifications. The future work should adopt a physical model of the drone and carry out additional tests on it. Parameters such as drag, and mechanics of motions should be acquired. Only then the work on this algorithm might be taken to the next stage, where tests might be carried out with less or no simplifications at all. Furthermore, having a physical model would allow performing hybridized tests in which real objects might be combined with the virtual swarm.

Additionally, in further tests, different shapes of research areas should be taken into consideration. For this algorithm, a circular area is simple as it does not have any corners. Areas of polygonal shapes might prove inconvenient for stochastically moving drones.

The presented algorithm is a discrete-time solution and every test is divided into steps. Positions of the drones are calculated only in time corresponding to these steps. Their position is unknown between these steps and the algorithm does not check if the paths of two drones cross between subsequent positions. The lower the time interval, the more precise results but simulating the test requires more computing power.

Further work should consider the smallest possible time intervals or additional feature to check if the paths of the drones do not cross.

## FUNDING

The authors received no financial support for the research, authorship, and/or publication of this article.

## REFERENCES

[1] Brambilla, Manuele, Eliseo Ferrante, Mauro Birattari and Marco Dorigo. 2013. "Swarm robotics: a review from the swarm engineering perspective". *Swarm Intell* 7 (1) : 1–41. DOI: 10.1007/s11721-012-0075-2.

[2] Schranz, Melanie, Martina Umlauft, Micha Sende and and Wilfried Elmenreich. 2020. "Swarm Robotic Behaviors and Current Applications". *Front. Robot. AI* 7 : 36. DOI: 10.3389/frobt.2020.00036.

[3] Pavlic, P. Theodore, Sean Wilson, Ganesh P. Kumar and Spring Berman. 2013. An Enzyme-Inspired Approach to Stochastic Allocation of Robotic Swarms Around Boundaries. In Inaba, M., Corke, P. (eds) *Robotics Research*. Springer Tracts in Advanced Robotics, vol 114. Springer, Cham. DOI: 10.1007/978-3-319-28872-7_36.

[4] Gage, W. Douglas. 1992. "Command control for many-robot systems". *Unmanned Syst.* 10 (4) : 28–34.

[5] Marjovi, Ali, Joao Goncalo Nunes, Lino Marques and Anibal de Almeida. 2009. Multi-robot exploration and fire searching. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1929-1934, doi: 10.1109/IROS.2009.5354598.

[6] Couceiro, S. Micael, Patricia A.Vargas, Rui P. Rocha and Nuno M.F. Ferreira. 2014. "Benchmark of swarm robotics distributed techniques in a search task". *Robotics and Autonomous Systems* 62 (2) : 200-213. http://dx.doi.org/10.1016/j.robot. 2013.10.004.

[7] Englot, Brendan and Franz S. Hover. 2014. Sampling-Based Coverage Path Planning for Inspection of Complex Structures. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling*. ICAPS 2012.

[8] Soto, Carlos Xavier and Robin R. Murphy. 2011. A survey of animal foraging for directed, persistent search by rescue robotics. In *Proceedings of the 9th IEEE International Symposium on Safety, Security, and Rescue Robotics*, (SSRR) pp. 314-320. DOI: 10.1109/ssrr.2011.6106744.

[9] Fox, Dieter, Jonathan Ko, Kurt Konolige, Benson Limketkai, Dirk Schulz and Benjamin Stewart. 2005. Distributed Multi-Robot Exploration and Mapping. In Proceedings of the IEEE. 94 (7). DOI: 10.1109/CRV.2005.36.

[10] Kumar, P. Ganesh, and Spring M. Berman. 2017. "Design of Stochastic Robotic Swarms for Target Performance Metrics in Boundary Coverage Tasks". arxiv-1702.02511v1 [cs.RO].
[11] Ayvali, Elif, Hadi Salman and Howie Choset. 2017. Ergodic coverage in constrained environments using stochastic trajectory optimization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 5204-5210, DOI: 10.1109/IROS.2017.8206410.
[12] Yanmaz, Evsen, Carmelo Costanzo, Christian Bettstetter and Wilfried Elmenreich. 2010. A discrete stochastic process for coverage analysis of autonomous UAV networks. In *Proceedings of the 2010 IEEE Globecom Workshops* pp. 1777-1782. DOI: 10.1109/GLOCOMW.2010.5700247.
[13] Zhang, Fangbo, Andrea L. Bertozzi, Karthik Elamvazhuthi and Spring Berman. 2018. "Performance Bounds on Spatial Coverage Tasks by Stochastic Robotic Swarms". *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 63 (6) : 1563-1578. DOI:10.1109/TAC.2017.2747769.
[14] Falkowski, Krzysztof and Michał Duda. 2021. "Swarm behavior optimization methods based on an original algorithm." *Problemy mechatroniki. Uzbrojenie, lotnictwo, inżynieria bezpieczeństwa* / *Problems of Mechatronics. Armament, Aviation, Safety Engineering* 12 (3) : 53-70. DOI:10.5604/01.3001.0015.2428.
[15] Lopez, J.C. 2021. *Quadrotor Simulation*. Import from: https://engineering.usu.edu/students/clubs/shpe/quacopter-simulation (10.05.2021).
[16] Rotaru, Constantin and Michael Todorov. 2018. Helicopter Flight Physics. In *Flight Physics Models, Techniques and Technologies* (Ed. Volkov, K.). London, UK: IntechOpen ISBN 978-953-51-3807-5.
[17] Cavcar, M. 2000. *The International Standard Atmosphere (ISA*). Import from: https://scholar.google.com/citations?user=WpEp_wcAAAAJ&hl=en (15.05.2021).
[18] Reid, S. John. 2020. *Drone flight – what does basic physics say?*. JSR 2018/2020, Version 3.

# Stochastyczny ruch roju wykonujący zadanie przeszukiwania z uwzględnieniem parametrów fizycznych

Krzysztof FALKOWSKI, Michał DUDA

*Wojskowa Akademia Techniczna,*
*Wydział Mechatroniki, Uzbrojenia i Lotnictwa,*
*ul. gen. S. Kaliskiego 2, 00-908 Warszawa*

**Streszczenie.** W artykule opisano próbę implementacji parametrów fizycznych do rozwiązania algorytmu wirtualnego roju. Określono, które parametry fizyczne pojedynczego obiektu muszą być znane, aby poprawnie przenieść wirtualny algorytm do systemu fizycznego. Rozważania oparto na stochastycznym roju ruchu wykonującym zadanie przeszukiwania. Zmierzono czas wykonania zadania i zużycie energii dla różnej liczby dronów w roju, co pozwoliło na wyznaczenie optymalnej wielkości roju. Dodatkowe testy zmieniających się zmiennych pozwoliły określić ich wpływ na wydajność roju. Przedstawiony algorytm jest rozwiązaniem dyskretnym i z każdym testem jest podzielony na kroki. Pozycje dronów są obliczane tylko w czasie odpowiadającym tym krokom. Ich pozycja między tymi krokami jest nieznana, a algorytm nie sprawdza, czy ścieżki dwóch dronów przecinają się między kolejnymi pozycjami. Im krótszy odstęp czasu, tym dokładniejsze wyniki, ale symulacja testu wymaga większej mocy obliczeniowej. Dalsze prace powinny uwzględniać możliwie najmniejsze odstępy czasu lub dodatkową funkcję do sprawdzenia jeśli ścieżki dronów się nie przecinają.

**Słowa kluczowe:** algorytm roju, zasięg przemiatania, zadanie pokrycia, optymalizacja, ruch stochastyczny