https://orcid.org/0000-0002-4168-7190

UDC 623.764

P.S. SAPATY*

SPATIAL MANAGEMENT OF AIR AND MISSILE DEFENCE OPERATIONS

*Institute of Mathematical Machines and Systems Problems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

Анотація. *Стаття стосується презентації, представленої на міжнародній конференції «Технології протиповітряної та протиракетної оборони», що проходила 16–17 листопада 2022 р. у Лондоні, Великобританія (день другий), і відображає зміст слайдів презентації. Робота описує застосування запатентованої і перевіреної в різних країнах світу Технології просторового захоплення (ТПЗ) та її Мови просторового захоплення (МПЗ) у галузі інтегрованої протиповітряної та протиракетної оборони (ІППО). З огляду на цілісну космічну навігацію та обробку за допомогою рекурсивного мобільного коду, що саморозгортається у розподілених світах, управління ТПЗ суттєво відрізняється від традиційного керування великими системами, адже вона складається з частин, які обмінюються повідомленнями. Динамічна мережа інтерпретаторів МПЗ може бути досить великою та охоплювати земне і небесне середовище як потужні просторові механізми. У роботі наведено приклад відстеження та знищення великої кількості крилатих ракет за допомогою просторової розвідки, що самостійно розвивається, за допомогою МПЗ і з використанням мереж радіолокаційних станцій. Також коротко описується група супутників, що розростається, на низьких навколоземних орбітах і шляхи їхнього потенційного застосування для організації ІППО. Починаючи з огляду Стратегічної оборонної ініціативи (SDI) і продовжуючи стислим описом останнього проєкта Агентства космічного розвитку, стаття демонструє рішення МПЗ для виявлення, відстеження і знищення балістичних ракет і гіперзвукових планерів із використанням груп супутників, розміщених на низьких навколоземних орбітах. Робота також показує, як організувати більш високий рівень спостереження за групами мобільних переслідувачів, які відстежують велику кількість цілей (потенційно як ракети, так і дрони), забезпечуючи за допомогою МПЗ глобальну обізнаність і навіть свідомість, що може суттєво покращити їхню продуктивність. Остання версія ТПЗ може бути реалізована на будь-яких платформах і введена в експлуатацію в найкоротші терміни, як і було зроблено з її попередніми версіями в різних країнах світу.*
Ключові слова: *протиповітряна та протиракетна оборона, радіолокаційні мережі, супутникові угруповання, Стратегічна оборонна ініціатива, Агентство космічного розвитку, гіперзвукові планери, Технологія просторового захоплення, Мова просторового захоплення, глобальна обізнаність.*

Abstract. *This paper relates to the accepted presentation presented at the international Air and Missile Defence Technology Conference, held on November 16–17, 2022, in London, the UK, (day two), reflecting the contents of the presentation slides. It describes applications of the patented and internationally tested Spatial Grasp Technology (SGT) and its Spatial Grasp Language (SGL) for integrated air and missile defense (IAMD). Based on holistic space navigation and processing by recursive mobile code self-spreading in distributed words, SGT differs radically from the traditional management of large systems since it consists of parts exchanging messages. The dynamic network of SGL interpreters can be arbitrarily large and cover terrestrial and celestial environments as powerful spatial engines. The paper contains an example of tracking and destruction of multiple cruise missiles by self-evolving spatial intelligence in SGL using networks of radar stations. It also briefs the growing multiple satellite constellation in low Earth orbits (LEO) for potential IAMD applications. Starting from the Strategic Defense Initiative (SDI) of the past and then briefing the latest project of the Space Development Agency, the paper shows SGL solutions for discovery, tracking, and destroying ballistic missiles and hypersonic gliders with the use of collectively behaving constellations of LEO satellites. It also shows how to organize higher levels of supervision of groups of mobile chasers fighting multiple targets (potentially both missiles and drones), by providing global awareness and even consciousness in SGL which can drastically improve their performance. The latest version of SGT can be implemented on any platform and put into operation in a short time, similar to its previous versions in different countries.*

Keywords: *air and missile defense, radar networks, satellite constellations, Strategic Defense Initiative, Space Development Agency, hypersonic gliders, Spatial Grasp Technology, Spatial Grasp Language, global awareness.*

DOI: 10.34121/1028-9763-2023-1-30-49

1. Introduction

This paper tells about the international conference held in London in November 2022 and the presentation accepted there [1], see also related Fig. 1 and Fig. 2. It is devoted to air and missile defense systems [1–4], and especially on how to organize them for effective protection of the population against the growing threats which can come from both air and space. Missile defense is considered a system, weapon, or technology involved in the detection, tracking, interception, and destruction of attacking missiles. Conceived as a defense against nuclear-armed intercontinental ballistic missiles (ICBMs), its application has broadened to include shorter-range non-nuclear tactical and theater missiles, cruise missiles, manned and unmanned aircraft, and other weapon systems. This adds urgency to the need for better and more integrated air and missile defense (IAMD). The transparency of the battlefield is also increasing due to the improved space-based sensors, sensors based on unmanned systems, and the technologies providing more precision, speed, and integration of air and missile weapons systems which are becoming of high value. The joint IAMD efforts should be well served by new operational concepts, where networked integration is of particular importance since the distributed defense envisions a more flexible air and missile defense.

The aim of the article is to investigate the suitability of the developed Spatial Grasp Technology [5–18] for the management of large distributed air and missile defense systems that may cover any national and international areas and even the whole world.

Based on completely different ideology and mechanisms in comparison with the conventional models and technologies, Spatial Grasp Technology allows us to obtain highly integral and extremely compact solutions in both terrestrial and celestial environments. Its applications already include such areas as intelligent network management, industry, social systems, collective robotics, military command and control, crisis management, national and international security, defense, distributed simulation, physical-virtual symbiosis, space-based systems, and even biology, psychology, and art.

The rest of the paper is organized as follows. Section 2 describes the representation of traditional systems and solutions in them as structures consisting of parts exchanging messages, with growing integrity and manageability problems when they become large and distributed.

Section 3 explains the main principles of the developed Spatial Grasp Model and Technology (SGT) which provides holistic world coverage with a parallel active code, gives some details of the basic Spatial Grasp Language (SGL) and its distributed interpretation.

Section 4 tells about different types of missiles which need advanced distributed air and missile defense systems and discusses their flight trajectories.

Section 5 describes an example of tracking cruise missiles with complex routes by distributed networks of radar stations under SGT, the latter providing mobile intelligence effectively following the missiles wherever they go.

Section 6 mentions rapidly growing satellite constellations that can be used for IAMD purposes.

Section 7 briefs the Strategic Defense Initiative (SDI) of the 80s with the key component called brilliant pebbles designed to destroy ballistic missiles, while expressing the pebbles functionality in SGL.

Figure 1 – The AMD Technology conference brochure

15.20 Distributed Control Technology for Air and Missile Defence Operations (Virtual)

- Global awareness and consciousness under Spatial Grasp paradigm
- Tracing cruise missiles by distributed networks of cooperating radar stations
- Collective supervision and destruction of hypersonic gliders by satellite networks
- Symbiosis of distributed interactive simulation with live control of large campaigns
- Technology implementation issues

Dr Peter Sapaty, Chief Research Scientist, **Ukrainian National Academy of Sciences**

Figure 2 – Conference presentation bullets

Section 8 briefs the Next-Generation Space Architecture launched by Space Development Agency (SDA) and provides mobile spatial solutions in SGL for the discovery, tracking, and destruction of hypersonic gliders with the use of cooperating constellations of low-orbit satellites.

Section 9 shows how to organize in SGL simultaneous chasing and destroying of multiple targets (that may be missiles or drones) with providing global awareness for a distributed team of chasers fighting many targets, thus drastically improving their performance.

Section 10 concludes the paper by acknowledging the suitability of the developed model and technology for the IAMD applications.

2. Representations of traditional systems and solutions in them

A system is traditionally considered a constellation of interacting elements or agents that have to act somehow to form a sort of unified whole [19–23], see Fig. 3.

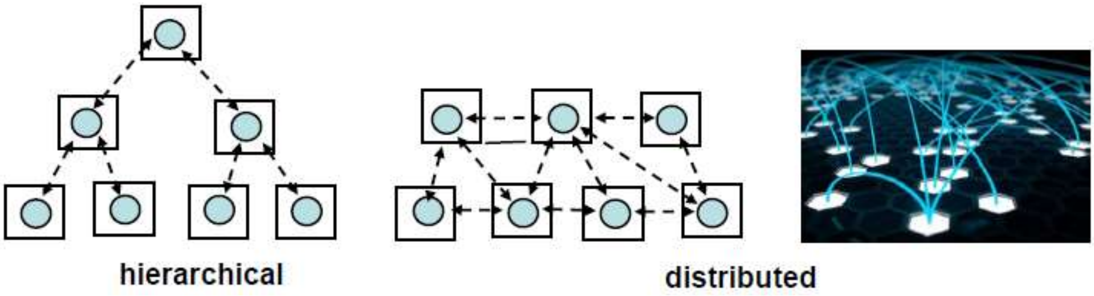


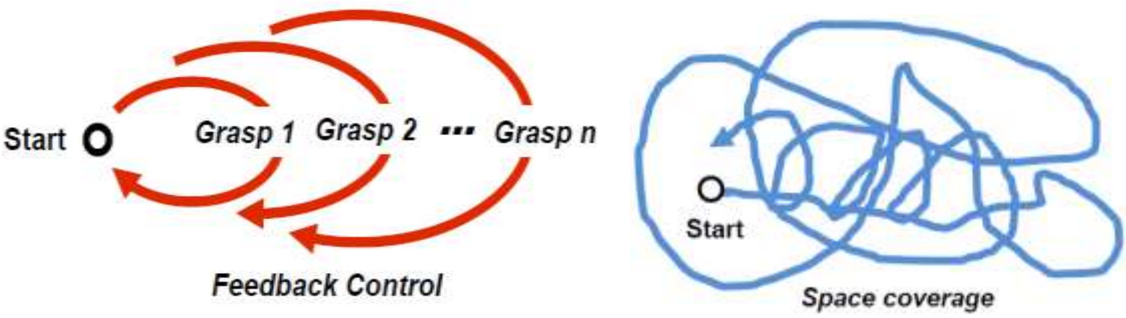
Figure 3 – Traditional systems representations that may be hierarchical, distributed, or combined

A distributed system is usually understood as one whose components communicate and coordinate their actions by passing messages. A distributed control system (DCS) may have many control loops where autonomous controllers are distributed throughout the system and there is no central supervisory control. The main problems associated with traditional system representations and organizations, especially when they are large, distributed, and dynamic, are how to provide their integrity, security, goal orientation, and overall control. This is especially complicated in situations with rapidly changing local and global goals, spatial distribution, varying number of components, and their communication topology.

3. Spatial Grasp Technology (SGT)

3.1. General description

Within Spatial Grasp Model and Technology [6–14], a high-level operational scenario is represented as an active self-evolving pattern rather than a traditional program. This pattern is expressed in recursive Spatial Grasp Language (SGL), and starting in any world point (or points) (see Fig. 4), propagates, replicates, modifies, covers, and matches the distributed environment in a parallel wave-like mode. This also combines feedback echoing the reached control states and obtained data, which may be remote, for making higher levels decisions (see Fig. 4*a*), with a virus-like space coverage of any extent, complexity, and virtuosity (see Fig. 4*b*), altogether providing holistic super-summative solutions unachievable by traditional agents-based systems.



a *b*

Figure 4 – Parallel wave-like world coverage and conquest under SGT

SGT allows for direct operation with different world representations. These are continuous and infinite Physical World (PW) accessible by physical coordinates, discrete Virtual World (VW) consisting of nodes and semantic links between them, and Executive World (EW) of active "doers" which can communicate. Different combinations of these worlds are also available, like VPW, VEW, EPW, and the most general VPEW.

3.2. Spatial Grasp Language (SGL)

The Spatial Grasp Language, its details can be found in [15–18], allows for expressing direct space presence and operations, unlimited parallelism, full distribution, dynamic and emergent hierarchies, unlimited mobility, as well as code clarity and compactness. The SGL universal recursive organization with operational scenarios called *grasp* can be expressed just by a single string in a formula-like mode:

grasp → *constant* | *variable* | *rule* ({ *grasp*, })

The SGL *rule* expresses certain action, control, description, or context accompanied by operands which can themselves be again any *grasp*. Top SGL details may be summarized as follows.

constant → *information* | *matter* | *custom* | *special*
variable → *global* | *heritable* | *frontal* | *nodal* | *environmental*
rule → *type* | *usage* | *movement* | *creation* | *echoing* |
verification | *assignment* | *advancement* | *branching* |
transference | *exchange* | *timing* | *qualifying*

Below is a brief explanation of some language features, with more constructs and details of their usage available in existing publications about the Spatial Grasp paradigm [6–18].

Variables

SGL has different variables supporting and serving its distributed and parallel spatial navigation and pattern matching, some of which can also move and replicate with the wave-like spreading code. There can be global variables, heritable variables, frontal variables, nodal variables, and environmental variables.

Description of object types and their usage

number(5) – defines an object as a number.
coordinate(X_Y) – defines an object as a coordinate in a two-dimensional space.
string('letter') – defines an object as a string with proper content.
matter("brick") – defines an object as a physical matter with proper content.
nodal(Name) – defines an object as a nodal variable with a proper name.
node('Peter') – defines an object as a virtual node with a proper name.
link('friend') – defines an object as a virtual link between nodes with proper names.

Elementary operations

add(5, 8) – adds two given numbers.
equal(23, 15) – compares two given numbers.
assign(Nodal, 5 – assigns a given number to a named nodal variable.
move(X_Y) – organizes spatial movement to a point with given coordinates.
create(Peter) – creates a new isolated virtual node with a proper name.
hop(Peter) – directly hops to a virtual node with a proper name.
linkup(friend, John) – connects the current node to another node with a named link.

Advancement in space with `gi` *as arbitrary complex SGL scenarios*

advance(g1, g2, g3) – the next scenario is developed from all nodes reached by the previous scenario, the results are defined by the last one.
slide(g1, g2, g3) – if any scenario fails, the next one is engaged from the same nodes, and so on.
repeat(g1) – the same scenario repeats from the nodes reached by the previous invocation till it is possible.

Branching in space with `gi` *as arbitrary complex SGL scenarios*

branch(g1, g2, g3) – scenarios develop in any order from the same starting position, and the results are united.
parallel(g1, g2, g3) – scenarios develop in parallel from the same starting position, and the results are united.
sequence(g1, g2, g3) – scenarios develop in a sequence from the same starting position, and the results are united.
or(g1, g2, g3) – any successfully developed scenario from the same starting position defines the result.
or_parallel(g1, g2, g3) – the first successful parallel scenario defines the result.
or_sequence(g1, g2, g3)) – the first sequentially activated successful scenario defines the result.
and(g1, g2, g3) – all scenarios must reply with success, and their results are united.
and_parallel(g1, g2, g3) – all parallel scenarios must reply with success, and their results are united.
and_sequence(g1, g2, g3) – all sequentially activated scenarios must reply with success, and their results are united.
if(g1, g2, g3 – if the first scenario is successful, there starts the second one, otherwise the third one from the starting position.
cycle(g1) – scenario repeats from the starting node till it is possible, all the reached results are united.
loop(g1) – scenario repeats from the starting node till it is possible, the last invocation gives results.
sling(g1) – scenario repeats from the starting node till it is possible, with the result obtained in the starting node.
whirl(g1) – scenario endlessly repeats from the starting node regardless of its success.

Synchronous-asynchronous mode of space navigation

synch_advance(g1, g2, g3) – the next scenario is applied only when the previous one is fully completed in space.
synch_repeat(g1) – the next scenario iteration is applied only when the previous one is fully completed.
asynch_advance(g1, g2, g3) – the next scenario is applied without waiting for the full completion of the previous one.
asynch_repeat(g1) – the next scenario iteration is applied without waiting for the completion of the previous one

Timing

start(g1) – starting at this point at the time defined by the scenario.
sleep(g1) – waiting at this point by the time defined by the scenario
allowed(g1, g2) – operation of the second scenario should not exceed the time defined by the first one/

Transference of control

run(g1) – transfer of control to the internal SGL procedure defined by the scenario.
call(g1) – transfer of control to the external system defined by the scenario.

The mode of navigation, evolution, and propagation

free(g1) – allows the scenario to develop as an independent branch not subordinate to global control.
stay(g1) – returns control to the starting node after any space development of the scenario.
seize(g1) – blocks access to the current node until the full completion of the current scenario.

Forward-feedback-forward propagation

maxdestination(g1) – propagation in space to multiple points with the return to the one with the maximum value.
mindestination(g1) – similar, but with the return to the one with the minimum value.

3.3. Programming example: finding the Shortest Path Tree and the Shortest Path in a network

Let us consider the creation of the Shortest Path Tree (SPT) in an arbitrarily large distributed network with the weights on its links and subsequent registering of the shortest paths (SP) based on this tree as in Fig. 5.

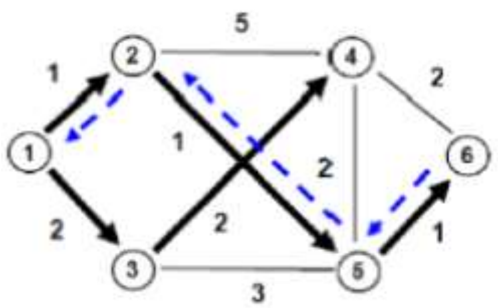


Figure 5 – Finding SPT and SP in an arbitrary network

The following examples expressed in SGL are extremely simple and compact in comparison with the usual solutions [24]. Despite being fully distributed and parallel, they are not based on traditional communicating parts or agents but rather on holistic recursive virus-like spatial navigation (as symbolically shown in Fig. 4). More data on finding SPT and SP using SGL can be found in [14].

Parallel finding and distribution of SPT from node 1:

```
frontal(Far); hop_node(1); Distance = 0;
repeat(hop_links(all); Far += LINK;
       or(Distance == nil, Distance > Far);
       Distance = Far; Before = PREVIOUS)
```

SP registering from node 1 to node 6 using the found SPT:

```
frontal(Path); hop_node(6);
repeat(append(Path, Name); hop(Before));
output(Path)
```

Result: 1, 2, 5, 6.

3.4. Distributed interpretation of SGL

The SGL interpreter (more details can be found in [6–14]) consists of specialized functional processors (rectangles) working with specific data structures (ovals) as shown in Fig. 6.

Each interpreter copy can handle and process multiple SGL scenario codes that currently happened to be its responsibility. Potentially integrated with any distributed systems, the SGL interpretation network can form a spatial computer with unlimited power for the simulation and management of distributed systems and worlds. Such collective engines can simultaneously execute many cooperative and competitive tasks without any central resources or control, as shown in Fig. 7.

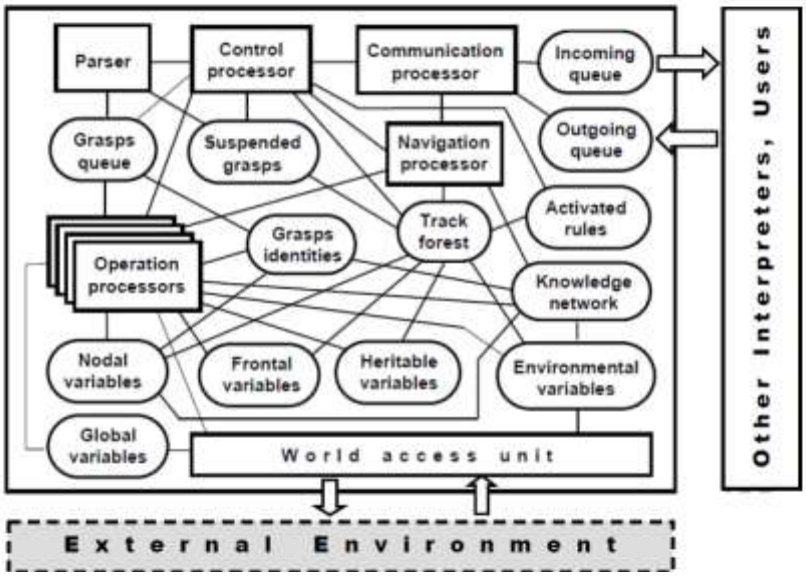


Figure 6 – Architecture of the networked SGL interpreter

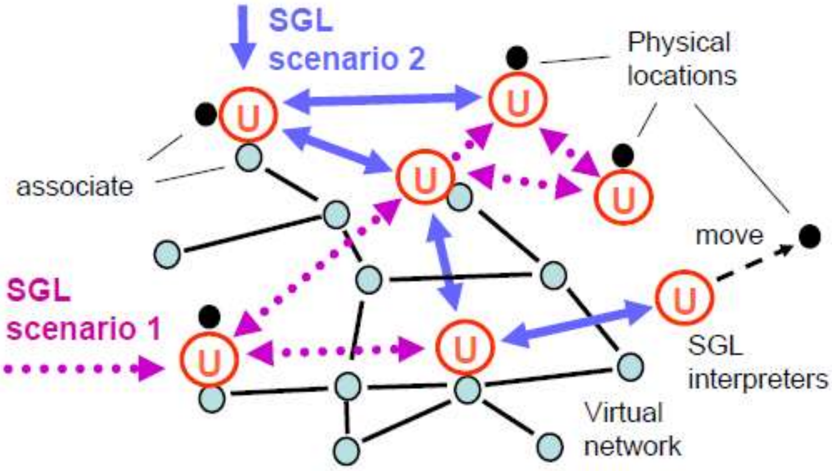


Figure 7 – Physical-virtual world processing by the SGL interpretation network

Communicating SGL interpreters can be potentially integrated into millions and even billions of copies, with their dynamic networks effectively forming powerful spatial engines capable of analyzing, ruling, and defending large distributed systems in any terrestrial and celestial environments, as shown in Fig. 8.

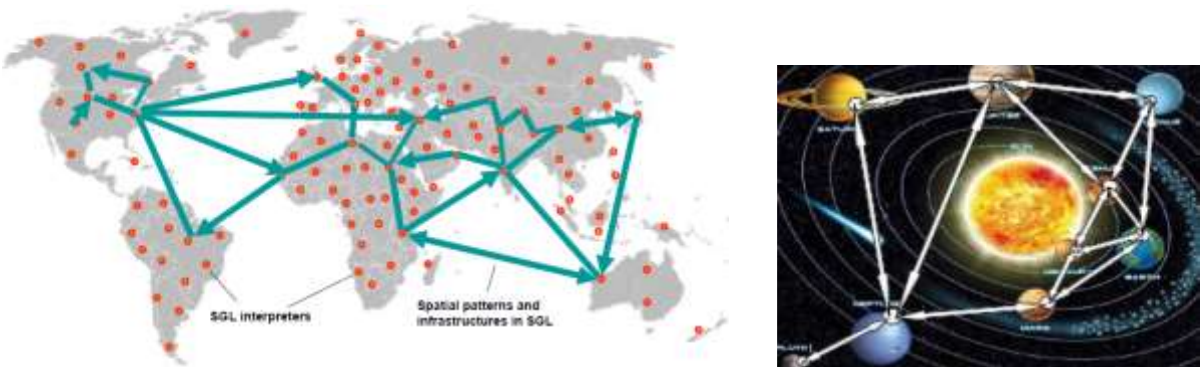


Figure 8 – Managing distributed terrestrial and celestial systems with the SGL interpretation networks

4. Different types of missiles and their trajectories

Below are briefed the main existing types of missiles that need advanced distributed air and missile defense systems for effective protection against their destructive influence (see also [25–28] and Fig. 9).

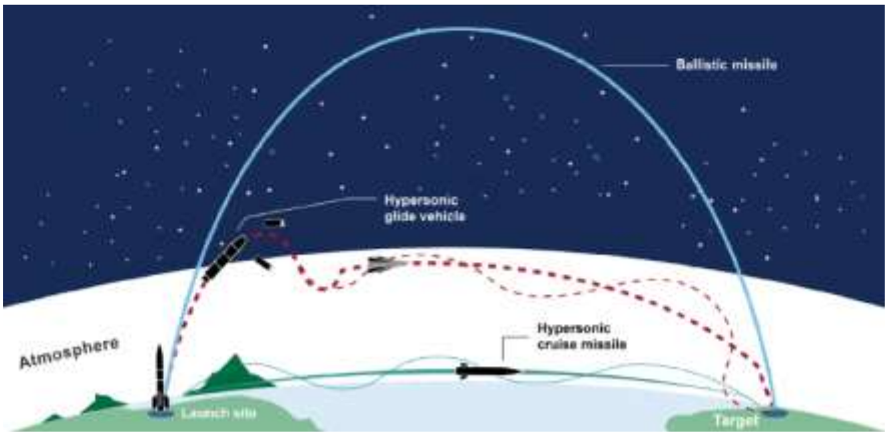


Figure 9 – Types of missiles and their flight trajectories

A ballistic missile uses projectile motion to deliver warheads to a target. These weapons are powered only during relatively short periods as most of the flight is unpowered. Short-range ballistic missiles stay within the Earth's atmosphere, while intercontinental ballistic missiles (ICBMs) are launched for a sub-orbital flight. A cruise missile is a guided missile used against terrestrial or naval targets. It remains in the atmosphere and flies the major portion of its flight path at an approximately constant speed. Cruise missiles are designed to deliver a large warhead over long distances with high precision. Modern cruise missiles are capable of traveling at high subsonic, supersonic, or hypersonic speeds, they are self-navigating and able to fly along a non-ballistic, extremely low-altitude trajectory. Hypersonic weapons are travelling at a hypersonic speed, i.e. between 5 and 25 times the speed of sound (1,6 to 8,0 km/s). Hypersonic glide vehicles maneuver and glide through the atmosphere at high speeds after an initial ballistic launch phase.

5. Tracking cruise missiles with the networks of radar stations under SGT

Distributed communicating radar stations operating under SGT can catch and follow moving objects (like cruise missiles [29–32], see also Fig. 10) throughout the whole region despite the limitations of individual sensors at radar stations.

Figure 10 – Cruise missiles and their trajectories

The radar discovering a new object starts distributed tracing operation, with the object’s visibility shifting to other radars after being lost by the current one. The propagation of the object route and the history of its behavior are collected by the SGT mobile intelligence virtually following its physical movement via the radar network. Depending on the collected history and current availability of the distributed shooting equipment, such an object may be destroyed or allowed to propagate further. Any number of mobile objects simultaneously propagating through the same area can be served by SGT (see Fig. 11).

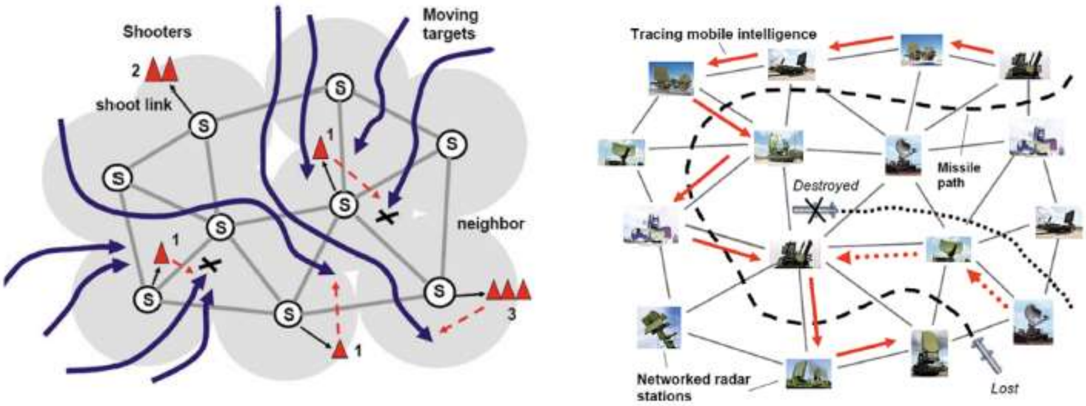


Figure 11 – Tracking and elimination of multiple cruise missiles
by the self-spreading mobile intelligence

Highly parallel and fully distributed SGL scenario for discovering, tracing, and destroying multiple cruise missiles with complex and unpredictable routes can be much simpler and more compact than with any other models and languages as follows.

```
hop(all_nodes);
frontal(Object, History, Threshold = …);
whirl(
   Object = search(aerial, new); visibility(Object) >= Threshold;
   free_repeat(
      loop(
         visibility(Object) >= Threshold); update(History, Object);
         if(dangerous(History), blind_destroy(Object)));
      max_destination(hop(all_neighbors); visibility(Object))))
```

6. Growing satellite constellations

The rapidly growing satellite constellations ([33–36], see also Fig. 12), especially in low Earth orbits (LEO), can be effectively used for advanced AMD operations aimed at the discovery, tracing, and elimination of high-speed supersonic and hypersonic objects with very complex, tricky, and unpredictable routes.

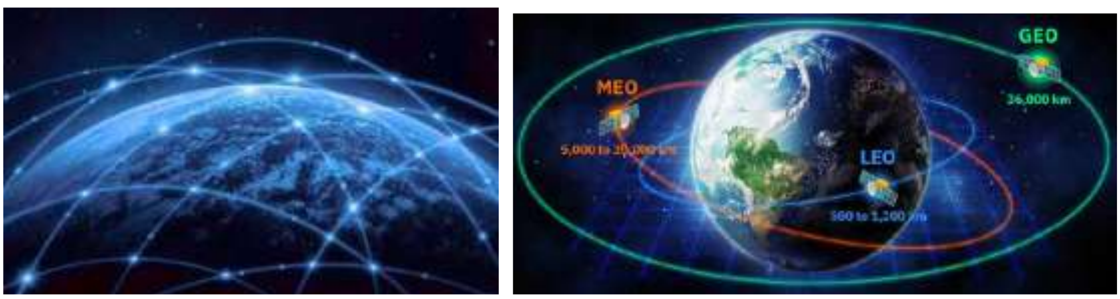


Figure 12 – Satellite constellations

Near-Earth space is also becoming increasingly privatized, with the number of LEO satellites predicted to grow dramatically from about 2,000 to over 100,000 due to the launch of planned satellite constellations. They are often even called mega-constellations by their expected size. The use of multiple satellites for AMD purposes is considered in the following sections, starting from the Strategic Defense Initiative of the 80s and then describing their use according to the latest project of the Space Development Agency.

7. Strategic Defense Initiative

The Strategic Defense Initiative (SDI) was a long-term technology research program developed to examine the feasibility of developing a defense against a ballistic missile attack [37–40], see Fig. 13.

Figure 13 – Strategic Defense Initiative project

The SDI program was officially launched in 1984. Its key component was known as brilliant pebbles and offered as a space-based weapon for the Global Protection Against Limited Strikes (GPALS). Brilliant pebbles were designed to destroy ballistic missiles when they were the most vulnerable, i.e. at the first stages of their flight while still keeping many warheads to be released later during the flight.

Discovery & destruction of targets by pebbles in SGL

An extended operation of pebbles for “boost”, “post-boost”, and possibly even “mid-course” stages of flight can be organized in SGL as shown below (see Fig. 14). Pebbles will make regular updates of the missile coordinates and try to reduce the distance to them, each time at least by *Limit1* value. If pebbles come closer to the missile on *Limit2*, they will try to attack and finally destroy the missile in a collision impact.

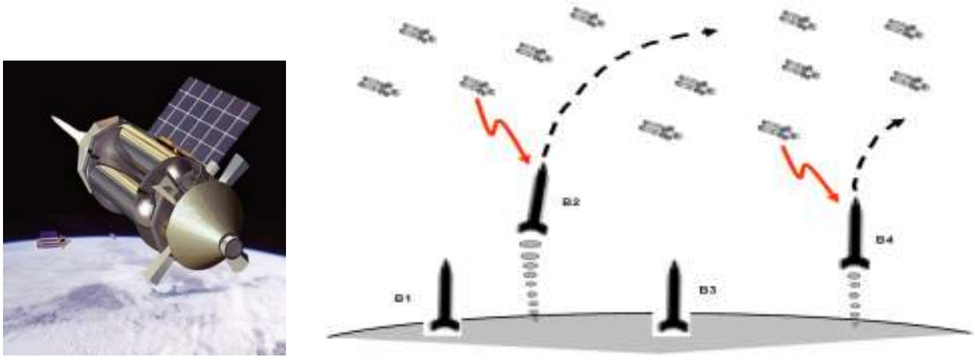


Figure 14 – Fighting multiple missiles with brilliant pebbles

```
hop(all_pebbles); Limit1 = …; Limit2 = …;
repeat(
   Target = search(missile_launch);
   if(nonempty(Target),
      repeat(update(Target); move_reduce(WHERE, Target, Limit1);
             if(close(WHERE, Target, Limit2), engage(Target))));
   sleep(delay))
```

8. Next-generation space architecture

This Notional Architecture has been recently launched by the Space Development Agency (SDA) [41–45], see Fig. 15.

Figure 15 – SDA Notional Architecture

This SDA Architecture plans to fight growing space-based threats, quickly provide hypersonic defense, track hypersonic threats from space, as well as arm satellites with lasers to shoot down missiles, etc. Unlike the SDI project, this architecture is focused on intensive cooperation and the collective behavior of many satellites. The implementation of some of its planned functionality for tracking and elimination of hypersonic gliders (see Fig. 16) with the use of LEO satellites operating under SGT will be considered.

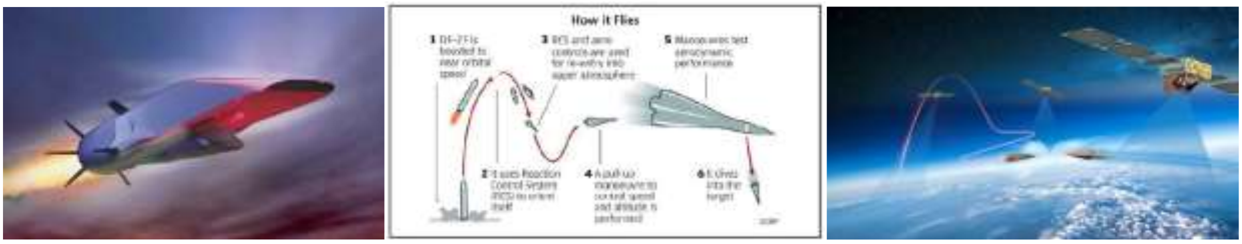


Figure 16 – Hypersonic gliders and their trajectories

8.1. Single-threaded tracking of hypersonic gliders

The moment when the satellite sensor sees a new object for the first time (within a given visibility threshold) is the start of a distributed tracking operation, after which the object will be continually monitored by this satellite until its visibility remains acceptable. Otherwise, the monitoring shifts to its neighbor having the best vision of the object after analyzing its visibility in all neighbors, see Fig. 17.

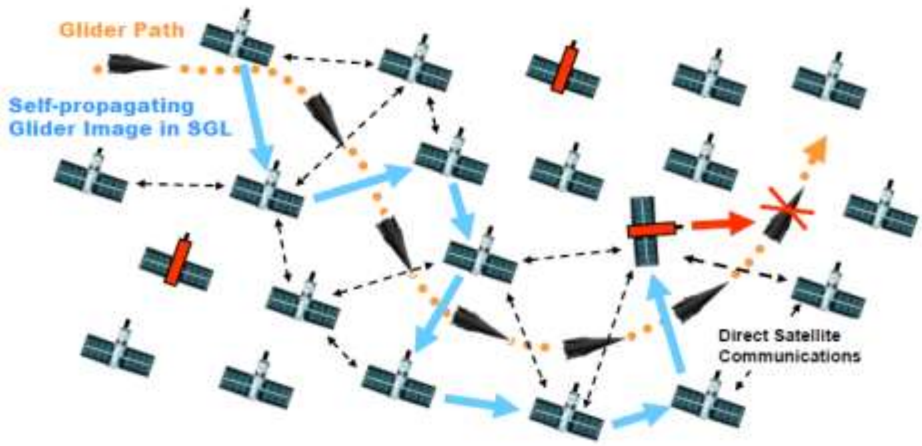


Figure 17 – Single-threaded tracking of a hypersonic glider

The history of the object's movement and behavior can be collected and updated at each passed satellite by the SGT mobile intelligence individually assigned to this object and accompanying its physical movement via the satellite network. The object may be eventually eliminated by satellites having shooting (laser-based) capability, as marked in red in Fig. 17. The related SGL scenario will be as follows.

```
frontal(Depth = …, Type = …, Object, History, Threshold1 = …,
        Threshold2 = …, All = {stay, (hopfirst(Depth, Type);
run(All))});
hopfirst(any, Type); run(All);
whirl(
  Object = search(glider, new); visibility(Object) >= Threshold1;
  repeat(
   loop(visibility(Object) >= Threshold1;  update(History, Object);
   if(and(TYPE = destroyer, danger(History),
          distance(Object, WHERE)< Threshold2),
      pursue_destroy(Object)));
   max_destination(hop(Depth, Type); visibility(Object))))
```

8.2. Multithreaded parallel tracking

In dynamic satellite network topologies, multithread tracing can also be allowed, where the next tracing stages may take place from more than a single neighbor with a sufficient vision of the object as in Fig. 18.

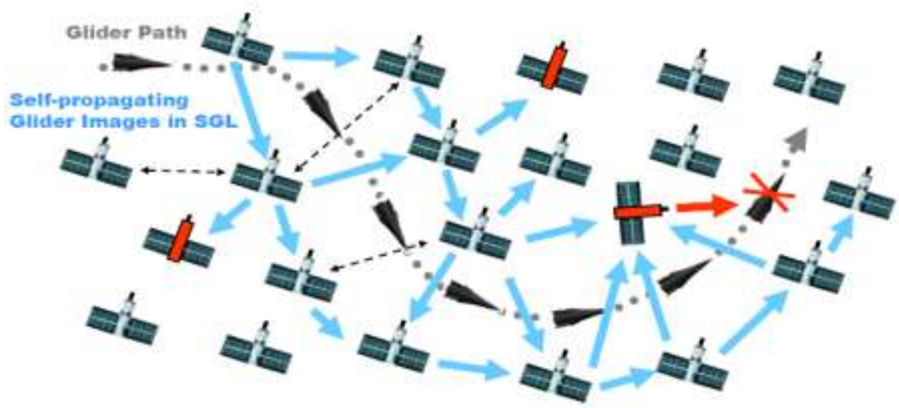


Figure 18 – Multithreaded tracking of a hypersonic glider

Even if the object is not currently visible from any neighbor after being lost by the current satellite, further flexibility can be achieved where the tracing activity is transferred to some or all its neighbors with any visibility in hope to eventually reach satellites seeing this object under the given threshold as follows in SGL.

```
frontal(Depth = …, Type = …, Object, History, Threshold1 = …,
        Threshold2 = …, All = {stay, (hopfirst(Depth, Type);
run(All))};
hopfirst(any, Type); run(All);
whirl(
  Object = search(glider, new); visibility(Object) >= Threshold1;
    repeat(
    loop(
      visibility(Object) >= Threshold1); update(History, Object);
      if(and(TYPE = destroyer, danger(History),
             distance(Object, WHERE) < Threshold2),
         pursue_destroy(Object)));
      or_sequence((hop(Depth, Type); visibility(Object) >= Threshold1),
                  hopforth(Depth, Type)))))
```

8.3. Managing custody observation by a mobile scenario in SGL

Watching Earth-based custody nodes (from which missiles can originate) from space will need frequent changing of their observing satellites, as each of them rapidly moves in space and can observe a certain point on Earth only within a few minutes as in Fig. 19.

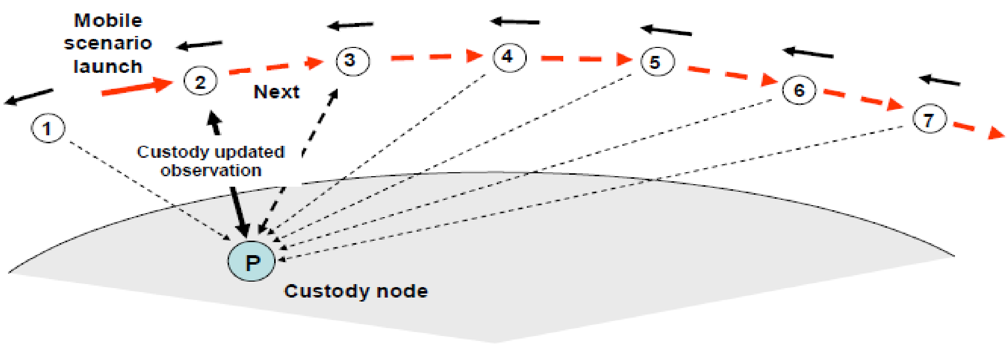


Figure 19 – Management of custody nodes by a satellite constellation under SGT

The predominantly stationary custody nodes may be symbolically considered as moving through the dynamic satellite network themselves. Below is an example of the SGL code.

```
frontal(Custody = X_Y, History, Depth = …, Type = …, Threshold = …);
hopfirst(any, Type);
repeat(distance(WHERE, Custody) > Threshold; hopfirst(Depth, Type));
repeat(
   if(distance(WHERE, Custody) <= Threshold,
      (update(History, observe(Custody)); output(History)),
      min_destination(hop(Depth, Type); distance(WHERE, Custody))))
```

8.4. Integrating custody and tracking functionality

After fixing a launch at the custody object, the glider-tracing mobile intelligence will be activated in SGL that will accompany this glider wherever it goes via the satellite network as in Fig. 20.

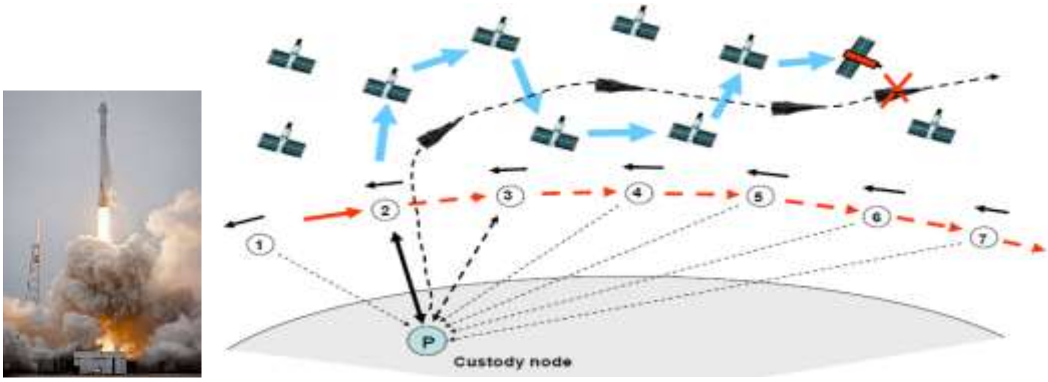


Figure 20 – Integrated custody and missile tracking management

This integrated SGL scenario will work repeatedly and endlessly, and if a new glider launch is detected in the observed custody, another tracking intelligence is associated with this object and will follow it via the satellite networks. At the same time, the continuing custody observation is also updated.

```
frontal(
  Custody = X_Y, History, Threshold1 = …, Threshold2 = …, Threshold3 =
…);
min_destination(hop(all); distance(WHERE, Custody));
repeat(
  if(distance(WHERE, Custody) <= Threshold1,
      (update(History, observe(Custody);
       if(belong(glider_launch, History),
           free_repeat(
             loop(
              if(visibility(glider) >= Threshold2,
                 if(and(TYPE = destroyer, distance(Object,
WHERE)<Threshold3),
                    pursue_destroy(glider))));
              max_destination(hop(neighbors); visibility(glider))))),
     min_destination(hop(neighbors); distance(WHERE, Custody))))
```

9. Chasing and destroying multiple targets

This chapter provides an example of a swarm of “chasers” that are constantly moving, discovering, and eliminating distributed targets, where both chasers and targets can potentially represent missiles, drones, or any other objects as in Fig. 21. The example is given in SGL.

Figure 21 – Possible types of multiple mobile objects

The chapter will also show how to enrich the swarm of chasers with a sort of global awareness (possibly, even consciousness [46–52]) over the whole operational area, which would allow individual chasers and the whole swarm to drastically improve their performance. This global awareness may be naturally and deeply embedded into the communicating chasers as a part of their regular functionality. It can also be organized in SGL as an additional superior level that can constantly migrate and oversee the swarm body and surrounding area, regularly supplying this global vision to individual units. In the end, there will be shown how this global awareness and consciousness can be organized outside the body of chasers and how they can function remotely from other systems.

9.1. Supplying a swarm of chasers with deeply embedded distributed awareness

By enriching the swarm with a sort of global awareness over the operational area, it is possible to substantially improve its performance, locally, and as a whole. This global awareness quality may be naturally embedded into the communicating chasers in a fully distributed way where targets seen by individual chasers are regularly exchanged with their neighbors, enriching their awareness, and these neighbors (of all they know and see) exchange with their neighbors too, and so on (see Fig. 22).

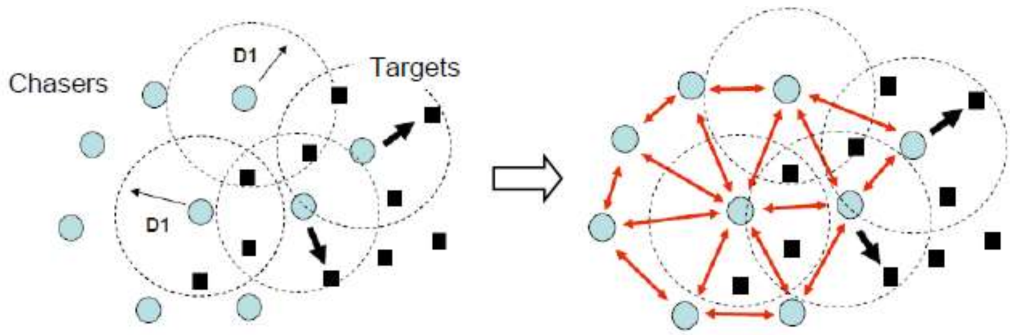


Figure 22 – Distributed global awareness

This makes all swarm members gradually become aware of all targets in the region, despite not all of them being visible individually. Their movement is always organized in a proper direction, firstly, to be closer to the targets, and secondly, to destroy them when it is possible. An example of such an SGL solution may be as follows.

```
hop_chasers(all);
nodal(D1 = distance, Targets); frontal(Exchange);
repeat(
    extend(Targets = search(D1);
    select_move_destroy_remove(Targets);
    stay(Exchange = Targets; hop(neighbors);
         merge(Targets, Exchange)))
```

9.2. Supplying a swarm of chasers with superior and migrating global awareness and consciousness

There can also be provided a higher-level awareness operating autonomously and independently over the basic swarm organization using the fully interpretive and mobile SGL nature. Initially applied in any swarm node, it will be able to contact all nodes in parallel, collect all of them they see, and then distribute this global vision in parallel to all the nodes. For security reasons, the focus of such a superior consciousness may constantly migrate between the chasers (see Fig. 23) and the SGL solution that follows.

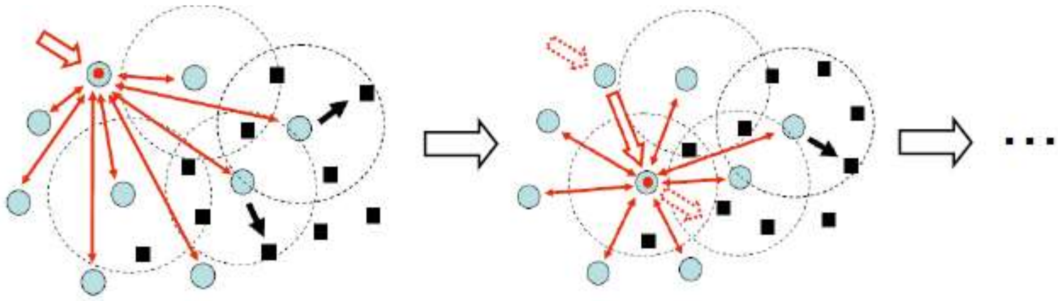


Figure 23 – Separated migrating superior awareness and consciousness

```
nodal(D1 = distance, Targets); frontal(Global);
parallel(
    (hop_chasers(all);
     repeat(
        extend(Targets) = search(D1);
        select_move_destroy_remove(Targets))),
    (hop_chaser(any);
     repeat(
        Global = merge(hop_chasers(all); Targets);
        stay(hop_nodes(all); merge(Targets, Global));
        move(any_neighbor))))
```

Also, for higher security and survivability, where each chaser unit can be potentially damaged or destroyed at any time on a battlefield, many such migrating consciousness centers can be introduced and operated in parallel. This will always guarantee the capability of this superior awareness and consciousness with any number of remaining chasers. To organize such a parallel consciousness capability, the last grasp of the above scenario can be modified as follows, with the `number` identifying the recommended number of direct neighbors to be reached:

```
move(any_neighbor) → move(neighbors, number)
```

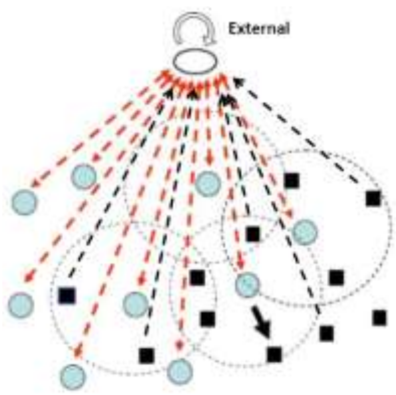


Figure 24 – Using external global awareness

9.3. Supplying a swarm of chasers with eternal global awareness and consciousness activated outside

This global awareness-consciousness mechanism can also reside outside the swarm body, i.e operate within other systems at any terrestrial or celestial distance from the swarm as follows (see Fig. 24). This outside consciousness can also be independent of the swarm organization, and collect what swarm units can see directly and immediately. It can also use more powerful external observation capabilities (like advanced radars) for overseeing the targets which may reside in the expected operational area or well beyond it.

```
nodal(D1 = distance, D2 = distance2, Targets);
frontal(Global);
parallel(
  (hop_chasers(all);
   repeat(
     extend(Targets) = search(D1);
```

```
    select_move_destroy_remove(Targets))),
 (hop(External);
  repeat_stay(
   Global = search_collect_targets(D2);
   hop_chasers(all); merge(Targets, Global))))
```

This can also be combined with embedded distributed awareness (see Subsection 9.1 and Fig. 22), effectively integrating internal and external global awareness sources for the swarm of chasers swarm as follows.

```
nodal(D1 = distance, D2 = distance2; Targets);
frontal(Global, Exchange);
parallel(
 (hop_chasers(all);
  repeat(
    extend(Targets = search(D1);
    select_move_destroy_remove(Targets);
    stay(Exchange = Targets; hop(neighbors);
         merge(Targets, Exchange))),
 (hop(External);
  repeat_stay(
   Global = search_collect_targets(D2);
   hop_chasers(all); merge(Targets, Global))))
```

10. Conclusions

This paper described in detail how to practically organize very effective, intelligent, and universal solutions for fighting different types of hostile objects, including cruise missiles, hypersonic gliders, and military drones under the developed parallel and distributed model and technology. It also explained how to provide in SGL a sort of global vision and awareness for mobile multi-component fighting systems in distributed environments, with each unit originally seeing only a part of the operational area. The technology has a simple and effective implementation on any platform, which was prototyped and tested in different countries for the previous versions of the technology.

REFERENCES

1. Air and Missile Defence Technology. London UK. 2022. November 16–17. URL: https://www.smgconferences.com/documentportal/ComplementaryContent/4209.pdf.
2. Karako T., Rumbaugh W. Distributed Defense: New Operational Concepts for Integrated Air and Missile Defense, Rowman & Littlefield. 2017. December 29. URL: https://books.google.com.ua/books/about/Distributed_Defense.html?id=cRVEDwAAQBAJ&redir_esc=y.
3. P. van Hooft, Boswinkel L. Surviving the Deadly Skies, Integrated Air and Missile Defence 2021–2035. *The Hague Centre for Strategic Studies*. 2021. November. URL: https://hcss.nl/wp-content /uploads/2021/12/Integrated-Air-and-Missile-Defense-HCSS-Dec-2021.pdf.
4. NATO Integrated Air and Missile Defence. Last updated. 2022. January. URL: https://www.nato.int/cps/en/natohq/topics_8206.htm.
5. Rozman J. Integrated Air and Missile Defense in Multi-Domain Operations. SPOTLIGHT 20-2. URL: https://www.ausa.org/sites/default/files/publications/SL-20-2-Integrated-Air-and-Missile-Defense-in-Multi-Domain-Operations.pdf.
6. Sapaty P. A distributed processing system: European Patent Office N 0389655. Publ. 10.11.93.
7. Sapaty P. Mobile Processing in Distributed and Open Environments. New York: John Wiley & Sons, 1999. 436 p.
8. Sapaty P. Ruling Distributed Dynamic Worlds. New York: John Wiley & Sons, 2005. 254 p.

9. Sapaty P. Managing Distributed Dynamic Systems with Spatial Grasp Technology. Springer, 2017. 304 p.
10. Sapaty P. Holistic Analysis and Management of Distributed Social Systems. Springer, 2018. 252 p.
11. Sapaty P. Complexity in International Security: A Holistic Spatial Approach, Emerald Publishing, 2019. 184 p.
12. Sapaty P. Symbiosis of Real and Simulated Worlds under Spatial Grasp Technology. Springer, 2021. 244 p.
13. Sapaty P. Spatial Grasp as a Model for Space-based Control and Management Systems. CRC Press, 2022. 202 p.
14. Sapaty P. The Spatial Grasp Model: Applications and Investigations of Distributed Dynamic Worlds. Emerald Points, 2023. URL: https://www.ottobookstore.com/book/9781804555750.
15. Sapaty P. A Brief Introduction to the Spatial Grasp Language (SGL). *Journal of Computer Science & Systems Biology*. 2015. Vol. 09 (02). P. 76–92.
16. Sapaty P. Spatial Grasp Language (SGL). *Advances in Image and Video Processing*. 2016. Vol. 4, N 1. P. 7–36.
17. Sapaty P. A Language to Comprehend and Manage Distributed Worlds. *Mathematical machines and systems.* 2022. N 3. P. 9–27.
18. Sapaty P. A wave language for parallel processing of semantic networks. *Computers and Artificial Intelligence*. 1986. Vol. 5 (4). P. 289–314.
19. System. URL: https://en.wikipedia.org/wiki/System.
20. Distributed systems. URL: https://computersciencewiki.org/index.php/Distributed_systems.
21. Distributed control system. URL: https://en.wikipedia.org/wiki/Distributed_control_system.
22. Buyya R., Selvi T. Parallel and Distributed Computing. *Mastering Cloud Computing*. Elsevier Inc., 2013. URL: https://www.sciencedirect.com/topics/computer-science/distributed-programming.
23. List of concurrent and parallel programming languages. URL: https://en.wikipedia.org/wiki/List_of_concurrent_and_parallel_programming_languages.
24. Dijkstra's Algorithm in C++ | Shortest Path Algorithm. 2021. May 20. URL: https://favtutor.com/blogs/dijkstras-algorithm-cpp.
25. Missile Defense: Better Oversight and Coordination Needed for Counter-Hypersonic Development. GAO-22-105075. Jun 16, 2022. URL: https://www.gao.gov/products/gao-22-105075.
26. Ballistic missile. URL: https://en.wikipedia.org/wiki/Ballistic_missile.
27. Cruise missile. URL: https://en.wikipedia.org/wiki/Cruise_missile.
28. Hypersonic weapon. URL: https://en.wikipedia.org/wiki/Hypersonic_weapon.
29. Xian Y. Cruise missile route planning based on quantum immune clone algorithm. *J. Inf. Comput. Sci*. 2012. Vol. 9. P. 8.
30. Osborn K. F-35 intercepts cruise missile to defend ship during important test. *The National Interest*. 2016. September 14. URL: https://nationalinterest.org/blog/the-buzz/f-35-intercepts-cruise-missile-defend-ship-during-important-17707.
31. Zinger W.H., Krill J.A. Mountain top: beyond-the-horizon cruise missile defense / J. Hopkins APL technical digest. 1997. Vol. 18, N 4. URL: https://www.jhuapl.edu/Content/techdigest/pdf/V18-N04/18-04-Zinger.pdf.
32. JLENS: Co-ordinating Cruise Missile Defense – And More, UTC by Defense Industry Daily staff. 2017. Feb. 13. URL: https://www.defenseindustrydaily.com/jlens-coordinating-cruise-missile-defense-and-more-02921/.
33. Skibba R. How satellite mega-constellations will change the way we use space. *MIT Technology Review*. 2020. February 26. URL: https://www.technologyreview.com/2020/02/26/905733/satellite-mega-constellations-change-the-way-we-use-space-moon-mars/.
34. United Nations Register of Objects Launched into Outer Space, The United Nations Office for Outer Space Affairs. URL: http://www.unoosa.org/oosa/en/spaceobjectregister/index.html.
35. Dredge K., M. von Arx, Timmins I. LEO constellations and tracking challenges. September/October 2017. URL: www.satellite-evolution.com.
36. Curzi G., Modenini D., Tortora P. Review Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions. *Aerospace*. 2020. Vol. 7 (9), N 133. URL: https://www.mdpi.com/2226-4310/7/9/133/htm.

37. Strategic Defense Initiative. The White House. 1984. URL: https://fas.org/irp/offdocs/nsdd/nsdd-119.pdf.
38. The Strategic Defense Initiative: Program Facts. 1987. URL: https://www.everycrsreport.com/files/19870722_IB85170_64d13e614c37eecbed39c00741ddfb269f814fef.pdf.
39. Gattuso J. Brilliant Pebbles: The Revolutionary Idea for Strategic Defense. 1990. URL: https://www.heritage.org/defense/report/brilliant-pebbles-the-revolutionary-idea-strategic-defense.
40. Brilliant Pebbles. URL: https://en.wikipedia.org/wiki/Brilliant_Pebbles.
41. Space Development Agency Next-Generation Space Architecture. 2019. URL: https://www.airforcemag.com/PDF/DocumentFile/Documents/2019/SDA_Next_Generation_Space_Architecture_RFI%20(1).pdf.
42. Magnuson S. Web Exclusive: Details of the Pentagon's New Space Architecture Revealed. 2019. URL: https://www.nationaldefensemagazine.org/articles/2019/9/19/details-of-the-pentagon-new-space-architecture-revealed.
43. Messier D. Space Development Agency Seeks Next-Gen Architecture in First RFI. 2019. URL: http://www.parabolicarc.com/2019/07/07/space-development-agency-issues-rfi/.
44. TRACKING, SDA. URL: https://www.sda.mil/tracking/.
45. CUSTODY, SDA. URL: https://www.sda.mil/custody/.
46. Chella A., Manzotti R. Artificial consciousness. ResearchGate. 2011. December. URL: https://www.researchgate.net/publication/225838750_Artificial_Consciousness.
47. Galland D., Grønning M. Spatial Consciousness. ResearchGate. 2019. April. URL: https://www.researchgate.net/publication/330753755_Spatial_consciousness.
48. Massimini M. The Distribution of Consciousness: A Difficult Cartesian Chart, Research-Gate. April, 2016. URL: https://www.researchgate.net/publication/307808562_The_Distribution_of_Consciousness_A_Difficult_Cartesian_Chart.
49. Nelson R., Bancel P.A. Exploring Global Consciousness. ResearchGate. 2016. October. URL: https://www.researchgate.net/publication/268001767_Exploring_Global_Consciousness.
50. Moran M. Network theory sheds new light on origins of consciousness. Medical Xpress. 2015. March 11. URL: https://medicalxpress.com/news/2015-03-network-theory-consciousness.html.
51. Sapaty P. Symbiosis of real and simulated worlds under global awareness and consciousness. *The Science of Consciousness (TSC) conferences*. Tucson, Arizona, 2020. URL: https://eagle.sbs.arizona.edu/sc/report_poster_detail.php?abs=3696.
52. Sapaty P. Simulating Distributed and Global Consciousness Under SGT, book Chapter. 2021. March 6. URL: https://link.springer.com/chapter/10.1007/978-3-030-68341-2_6.

Стаття надійшла до редакції 23.10.2022