

PROBABILITY MODELLING OF AIR DEFENCE PROCEDURES

Stary, V.

Department of Air Defence; University of Defence, Kounicova 65, 662 10 Brno, Czech Republic

E-Mail: vadim.stary@unob.cz

Abstract

Air defence operations are characterised by uncertainty at every stage of the engagement chain, from detection to destruction of aerial threats. Conventional deterministic analyses often overlook the stochastic nature of detection errors, classification uncertainty, system availability, and weapon effectiveness. This study introduces a probabilistic framework for modelling the air defence engagement process, structured according to the radar–command–fire unit cycle. Each decision point and transition is represented by an appropriate probability distribution, with parameters estimated from assumed or reported operational performance values. Using Monte Carlo simulation with several types of probability distributions in Python, the model evaluates the overall probability of successful target destruction under varying system configurations. Results highlight the sensitivity of the engagement chain to detection reliability, identification accuracy, and missile kill probability, offering valuable insights for system planners and operational analysts.

(Received in February 2026, accepted in May 2026. This paper was with the author 2 weeks for 2 revisions.)

Key Words: Air Defence, Probabilistic Modelling, Monte Carlo Simulation, Engagement Chain, Stochastic Analysis

1. INTRODUCTION

Air defence systems provide a layered protection mechanism against hostile aerial incursions. The success of these systems depends upon a sequence of coordinated actions: detection by radar, classification by command centres, assignment of targets to fire units, and the final engagement through missile launch. Each of these steps involves uncertainty, stemming from limitations in technology, environmental conditions, and human decision-making.

While traditional evaluations often employ deterministic success rates, a probabilistic perspective better reflects the operational realities of air defence. The approach presented in this paper seeks to capture the stochastic nature of the engagement process by associating probability distributions with each procedural step. This enables the estimation of an overall probability of target neutralisation, rather than relying upon simplistic point estimates.

The main contribution lies in the structured transformation of an air-defence engagement procedure into an executable stochastic dependency model that is transparent, modifiable, and suitable for scenario-based experimentation. In contrast to deterministic kill-chain multiplication, the proposed framework explicitly represents procedural dependencies between detection, identification, weapon assignment, readiness, engagement, guidance, and terminal effect. The model also distinguishes between serial and parallel dependency logic, enabling the analyst to identify where probability degradation accumulates and where redundancy has the greatest operational value.

The presented paper builds on and extends a previous study [1], which addressed Ordinary Differential Equations (ODE) and Discrete Event Simulation (DES) using deterministic time models of air defence. It is partially inspired by the paper called “Combat simulation framework including continuous detection system” [2], specifically the part which deals with detection probability.

Specifically, the paper provides:

1. a formal event-dependency representation of selected air-defence procedures (Chapter 2);

2. an executable Monte Carlo implementation with configurable probability distributions (Chapter 3);
3. a verification procedure based on analytical cross-checks and deterministic baseline comparison (Chapter 4);
4. a scenario-based case study showing how probability degradation propagates through the engagement chain under nominal, saturated, and degraded electronic-warfare conditions (Chapter 4 and Chapter 5).

1.1 Study objectives

The objective of this study is threefold:

- To design and describe a probabilistic model of the air defence engagement process as described in the standard workflow (radar → command post → fire units).
- To implement and simulate this model in Python, using Monte Carlo methods to obtain realistic estimates of success probabilities under different scenarios.
- To verify the implementation by comparison with analytical closed-form reliability calculations and a deterministic multiplication baseline.

Beyond these main objectives, the study also seeks to demonstrate the applicability of probabilistic modelling techniques to complex defence scenarios, where uncertainty and variability play a decisive role, e.g. for airbase air defence [3] or defence structure optimization [4].

2. OVERALL MODEL LAYOUT

The overall air defence engagement process is very complex, and it is composed of significant number of steps. It is also dependant on various conditions, specifically on the type of the system, the environment conditions, target type, and the engagement procedures itself. Generally, these steps (sub-procedures) can be described according to NATO Architecture Framework (NAF), see [5], from the following perspective (viewpoints):

- Operational Viewpoint (NOV) – Describes the tasks, activities, operational nodes, and information flows required to accomplish military missions. In the context of air defence, this includes the procedures for detection, target identification and classification, designation to a fire unit, and engagement execution.
- Technical Viewpoint (NTV) – Specifies the standards, rules, and conventions that systems and services must comply with. For this model, it captures the technical constraints and performance criteria, such as the reliability requirements of each subsystem.
- System Viewpoint (NSV) – Describes the systems, their composition, interconnections, and functions that support operational capabilities. In air defence, it represents the system architecture, including sensors, the command-and-control (C2) system, and effectors.

2.1 Air defence procedures

The Air Defence operations for the purpose of this paper are based on the reference [6] and partially use decision-making processes described in source [7]. The process begins with continuous air surveillance conducted by sensors or radar systems. Once an aerial object (AO) is detected, the system immediately assigns it a unique identification number. This is followed by a critical identification process to classify the object as either friendly or hostile.

If the object is confirmed as hostile, a track is generated, and a weapon system is assigned based on available options e.g. Fighter Aircraft (FA) or a Surface-to-Air Missile (SAM) Fire Unit (FU). The assigned unit then verifies its readiness before executing the mission. This involves an FA flying toward the target or a SAM system acquiring and tracking it.

The final stages include the actual engagement, missile launch, missile guidance and ultimately, the destruction of the target.

For study purposes, the air defence procedures are illustrated in the simplified diagrams shown in Figs. 1 and 2.

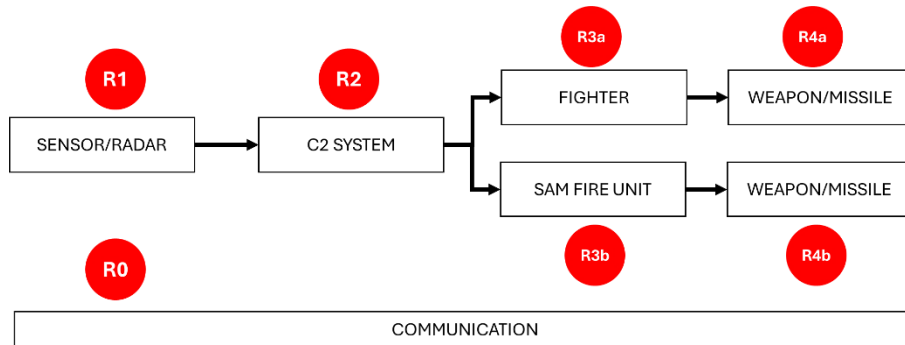


Figure 1: Simplified entity structure of Air Defence (NSV).

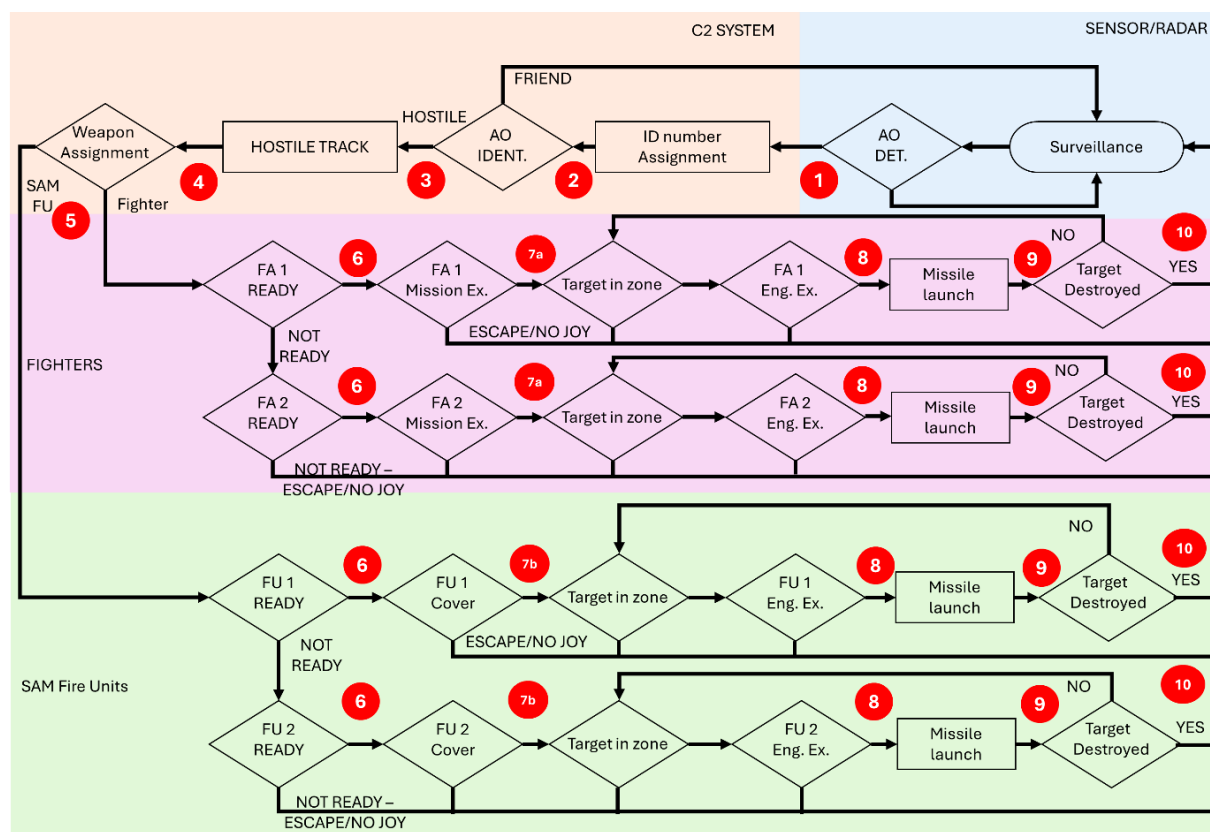


Figure 2: Operational diagram of Air Defence Procedures (NOV, NSV) [6].

Fig. 1 shows a simplified system perspective for defining the reliability of each entity, specifically:

- 0. Communication system.
- 1. Sensor/Radar.
- 2. C2 System.
- 3. According to the WS:
 - a. FA,
 - b. SAM FU.
- 4. According to the WS:
 - a. FA Missile,

b. SA Missile.

Fig. 2 illustrates the procedural steps of an Air Defence operation within the system's general architecture. The procedure begins with air surveillance provided by a sensor/radar. The following steps are:

1. AO detection.
2. ID assignment for each AO.
3. Proper identification (Friend/Hostile).
4. Hostile track production.
5. Weapon assignment according to available weapon system (WS), FA or SAM FU.
6. FA or SAM FU is in required readiness state.
7. According to the WS:
 - a. FA mission execution (e.g. flight to the target),
 - b. FU cover the target (e.g. SAM system is able to track the target).
8. Engagement execution.
9. Missile guidance execution.
10. Target fatal damage or destruction.

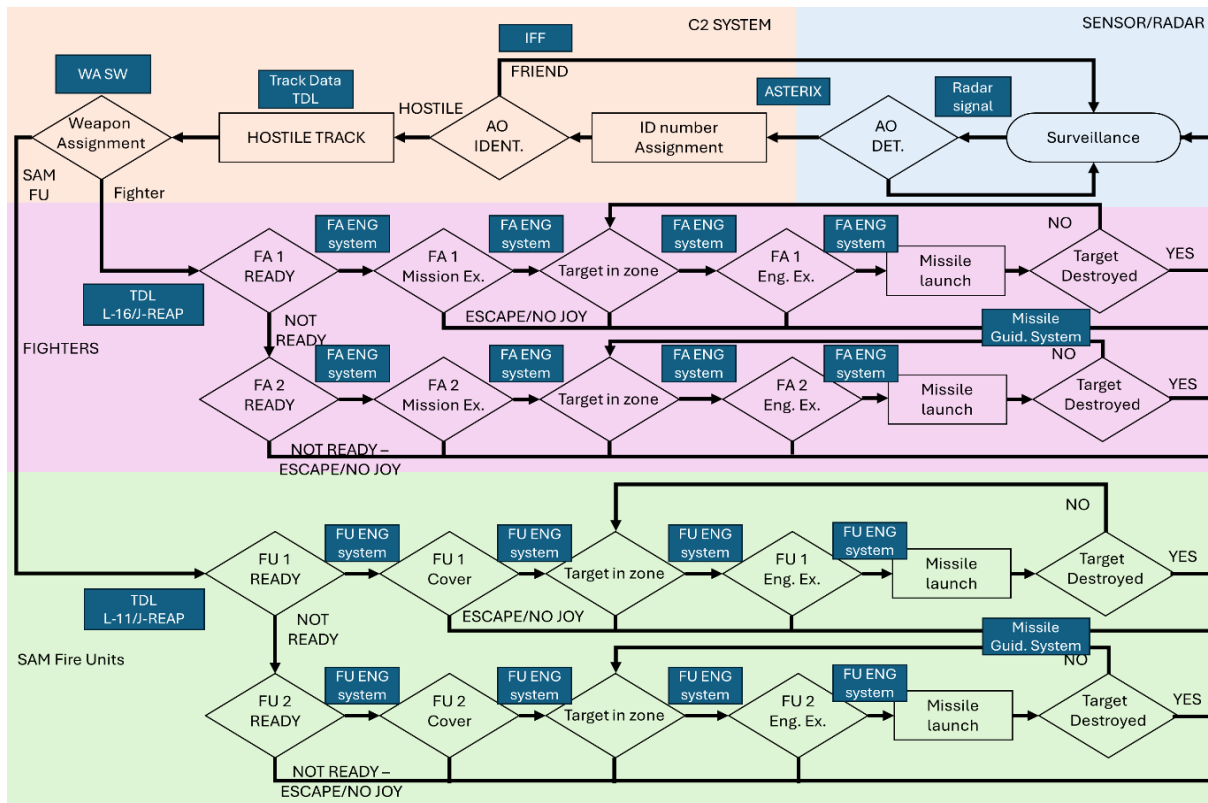


Figure 3: Operational diagram of Air Defence Procedures with technical standards (NOV, NTV) [6].

The Fig. 3 extends the procedural steps of an air defence operation by incorporating a generalized technical realization architecture (blue boxes). The main technical standards are associated with sensors, communications, and engagement systems, as outlined below:

- Radar signal – e.g. frequency (wavelength), modulation, etc.,
- ASTERIX – standard for information exchange between radar systems,
- IFF (Identification Friend or Foe) – standard for radar identification of friendly forces,
- Track data / TDL (Tactical Digital Link) – communication standards for the exchange of aerial track data,
- WA SW (Weapon Assignment Software) – specialized weapon control software used to assign weapon systems,

- TDL – Link 16 / J-REAP – communication standards providing C2 (Command and Control) capabilities (Link 16 or J-REAP-C),
- TDL – Link 11 / J-REAP – communication standards providing C2 capabilities (obsolete Link 11B or J-REAP-C),
- FA/FU Engagement System – Fighter Aircraft or Fire Unit engagement system, usually proprietary and specific to the type of fighter or SAM fire unit,
- Missile Guidance System – e.g. passive IR, active radar homing, command guidance, or laser beam-riding guidance.

3. SIMULATION ARCHITECTURE

This chapter presents an interactive framework for an event-based stochastic simulator with explicit dependency modelling. The proposed system integrates discrete probabilistic event modelling, directed dependency graphs, and iterative simulation execution. Multiple classical probability distributions are supported through a unified abstraction layer, enabling flexible modelling of heterogeneous stochastic processes. The framework is further enhanced by a graphical user interface that facilitates model construction, parameterization, execution, and visualization. The approach is suitable for various simulation studies, and it is applied and verified on the air defence procedures described in Chapter 2 and in reference [6].

The simulation framework is implemented using the Python programming language, chosen for its extensive scientific computing ecosystem and suitability for rapid prototyping of simulation models. Core numerical operations rely on *NumPy*, while probabilistic sampling and probability density evaluation are provided by *SciPy*'s statistical modules. Event dependencies are represented and analysed using the *NetworkX* library, enabling graph-based modelling and cycle detection. Visualization is performed with *Matplotlib*, and interactive model configuration is supported through *Jupyter* widgets, allowing human-in-the-loop simulation design and execution.

The source code repository can be accessed via the reference [8]. The provided code can be implemented into comprehensive ODE DES simulation tool described by Stefek et al. [1], it can be also extended by the missile models described by Farlík et al. [9] or missile neural network models described by Wade [10].

3.1 General combined architecture

Simulation modelling plays a central role in the analysis of complex systems characterized by uncertainty, interdependencies, and non-deterministic behaviour. Event-based simulation allows systems to be represented as a sequence of stochastic events whose outcomes influence subsequent system states.

Traditional discrete-event simulation frameworks often require substantial programming expertise and offer limited interactivity. The system analysed in this chapter addresses this limitation by combining:

- modular stochastic event definitions,
- explicit dependency representation via directed graphs,
- iterative execution logic, and
- interactive model construction using a graphical interface.

The result is a flexible simulation environment that bridges probabilistic theory and practical experimentation.

Traditional Monte Carlo simulation is widely used for generating and evaluating uncertain scenarios in simulation studies, including recent work on virtual test-scenario generation in virtual reality environments [11].

3.2 Event-based stochastic representation

The Event class E_i was implemented to represent discrete actions within the model.

$$E_i = (X_i, P_i, D_i) \quad (1)$$

Each event is characterised by:

- X_i is a variable representing the event outcome with its name serving as an identifier,
- P_i is the probability distribution governing X_i ,
- $D_i \subseteq \{E_1, E_2, \dots, E_{i-1}\}$ define a list of dependencies to model (serial/parallel event structures).

When executed, an event samples an outcome from its distribution. Execution is restricted until all dependencies are satisfied.

3.3 Implemented probability distributions

The framework supports a class of common discrete and continuous distributions, which are defined by following equations [12]:

Binomial:
$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \quad (2)$$

Poisson:
$$P(X = k) = \frac{\mu^k e^{-\mu}}{k!} \quad (3)$$

Normal:
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)} \quad (4)$$

Exponential:
$$f(x) = \lambda e^{-\lambda x}, x \geq 0 \quad (5)$$

Geometric:
$$P(X = k) = (1 - p)^{k-1} p, \quad (6)$$

Uniform:
$$f(x) = \frac{1}{b - a}, x \in [a, b] \quad (7)$$

These are well-known distributions define both a sampling operator and a probability evaluation operator, enabling outcome likelihood estimation and can be easily adapted for the specific application.

The simulator calculates the "Total Combined Probability" of a specific simulation run. Assuming conditional independence of events given their parameters (which are static in this implementation), the joint probability P_{total} for a set of realized outcomes $\{x_1, x_2, \dots, x_n\}$ in the final iteration is calculated as the product of their individual likelihoods:

$$P_{serial} = \prod_{i=1}^n P(X_i = x_i) \quad (8)$$

$$P_{parallel} = 1 - \prod_{i=1}^n (1 - P(X_i = x_i)) \quad (9)$$

For binary events, the likelihood term corresponds directly to the probability of success. For continuous distributions, the sampled value represents a performance variable, such as execution time or tracking quality, and the probability contribution is evaluated through the corresponding density or through a threshold-based success criterion. In the application example, the final cumulative success probability is calculated from the realised success states of the dependency graph rather than from a direct multiplication of unrelated density values. This avoids interpreting probability density values as event success probabilities.

3.4 Iterative simulation engine

The execution of events was managed by the *IterativeSimulator* class, similar to reference [1]. A directed acyclic graph (DAG), see Fig. 4, of event dependencies was constructed using *NetworkX*, ensuring correct execution order. The simulator was designed to:

- Identify events with no dependencies and execute them initially.
- Resolve dependent events as their prerequisites are satisfied.
- Record outcomes.
- Compute joint probabilities of the observed outcomes.

A graphical visualization of the dependency network was also provided, enhancing the interpretability of the model structure and helping to clarify the topology of connections and events (procedures).

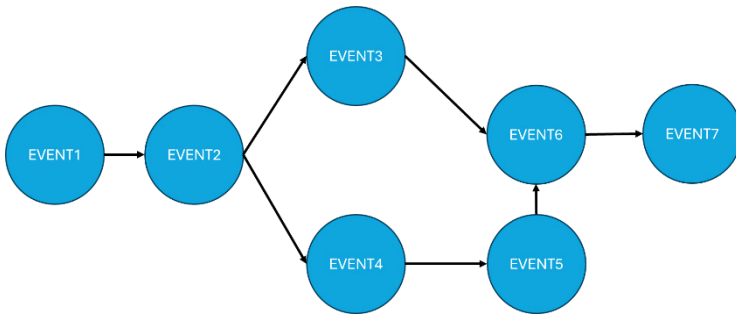


Figure 4: Example of directed acyclic graph (DAG).

3.5 Graphical User Interface

To ensure usability for non-programmers, a lightweight GUI was developed, see Fig. 5, using *ipywidgets*. Through this interface, users are enabled to:

- define events by specifying their name, distribution type, and parameters,
- establish dependencies between events,
- execute simulations while immediately viewing results and dependency graphs.

The GUI was intended to facilitate educational use and experimental exploration, where rapid model prototyping is required; therefore, a simple textbox and button layout was chosen.

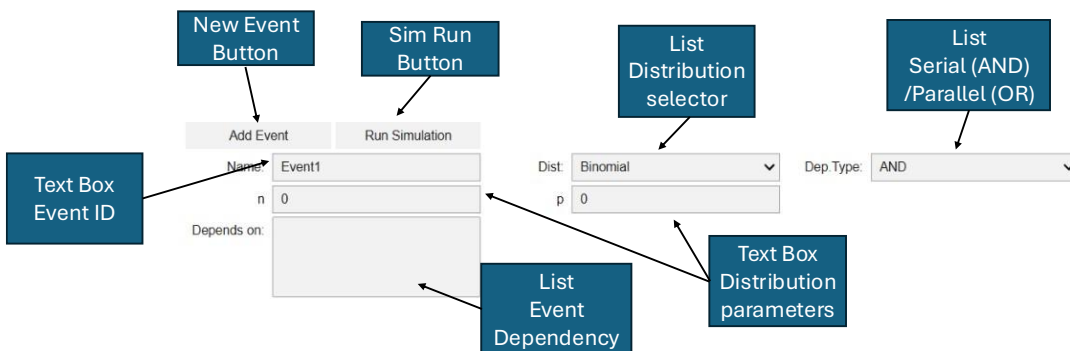


Figure 5: GUI of Event Probability Sim.

4. APPLICATION EXAMPLE

The objective of this chapter is to apply the probability simulation tool introduced in Chapter 3 to the complex air defence procedures detailed in Chapter 2. These procedures are modelled on various scenarios governed by estimated probability distributions. Due to the

sensitive nature of military tactical and technical data, the probability values presented herein are illustrative, serving primarily for experimental and functional evaluation. The input data architecture allows for easy modification to align with actual operational entities.

For example, to enhance the fidelity of the simulation, based on reference [3] methodology which replaces static destruction probabilities with a physics-based model derived from the weapon's Circular Error Probable (*CEP*). In this approach, the probability of a successful hit (P_{hit}) within a specific target radius (r) is calculated using the formula:

$$P_{hit} = 1 - e^{-(r/k \cdot CEP)^2 \cdot \ln(2)} \quad (10)$$

This formula models impact error as a two-dimensional normal distribution centred on the target, where k is a conversion factor. By using this equation, the simulation can dynamically account for variable missile accuracy and target size.

4.1 Scenario description and settings

The simulation environment is calibrated to three distinct operational realities to evaluate the robustness of the dependency model. All simulation parameters for each scenario are given in Table I.

Table I: Simulation settings.

Event ID	Event Name	Depend. (Parent)	Dist. Type	Scenario 1	Scenario 2	Scenario 3
E1	AO Detection	None	Binomial	$n=1, p=0.95$	$n=1, p=0.95$	$n=1, p=0.80$
E2	ID Assignment	E1	Poisson	$\mu=8.0$	$\mu=3.0$	$\mu=1.5$
E3	Proper ID	E2	Binomial	$n=1, p=0.98$	$n=1, p=0.95$	$n=1, p=0.90$
E4	Hostile Track Prod.	E3	Binomial	$n=1, p=0.99$	$n=1, p=0.95$	$n=1, p=0.90$
E5	Weapon Assignment	E4	Binomial	$n=1, p=0.99$	$n=1, p=0.95$	$n=1, p=0.90$
E6	Unit Readiness	E5	Binomial	$n=1, p=0.98$	$n=1, p=0.95$	$n=1, p=0.80$
E7a	FA Mission Exec.	E6	Normal	$\mu=10, \sigma=1$	$\mu=10, \sigma=1$	$\mu=10, \sigma=1$
E7b	SAM Tracking	E6	Normal	$\mu=10, \sigma=1$	$\mu=10, \sigma=1$	$\mu=10, \sigma=1$
E8	Engagement Exec.	E7a, E7b	Binomial	$n=1, p=0.99$	$n=1, p=0.95$	$n=1, p=0.90$
E9	Guidance Exec.	E8	Geometric	$p=0.96$	$p=0.90$	$p=0.60$
E10	Target Destruction	E9	Binomial	$n=1, p=0.99$	$n=1, p=0.98$	$n=1, p=0.90$

Scenario 1: Standard Interception (Low intensity)

Estimated probability above 0.8.

This scenario represents a nominal air-defence operation against a conventional, non-stealth threat under clear weather conditions. Sensor performance and command-and-control functions operate near optimal levels, with built-in redundancy ensuring that early detection and identification steps are almost certain. As a result, downstream events retain high conditional success probabilities, effectively neutralizing logical bottlenecks and producing a high overall target-destruction probability. This scenario therefore serves as a baseline for system verification and validation.

Scenario 2: Saturated Attack (Swarm – multiple targets)

Estimated probability above 0.5.

This scenario models a realistic combat engagement characterized by standard operational friction and the absence of special redundancy measures. While hardware operates within nominal tolerances, execution is constrained to single-attempt protocols, making the system vulnerable to the cumulative effects of minor inefficiencies. The logical dependency at the engagement stage acts as a probabilistic filter: the combination of independent tracking processes naturally reduces the aggregate success probability to a moderate, near “coin-flip” outcome. This reflects the uncertainty typical of near-peer conflict environments.

Scenario 3: Stealth and Electronic Warfare (EW) Environment

Estimated probability about 0.15.

This scenario simulates a highly degraded operational environment dominated by heavy electronic countermeasures, stealth characteristics, and potential physical system suppression Anti-Access / Area Denial (A2/AD). Critical sensor and guidance nodes experience significantly reduced reliability and elevated signal noise, causing a rapid loss of probability mass early in the kill chain. The compounding effect of these degradations – amplified by the system’s logical dependencies – leads to a systemic collapse in performance, where the likelihood of a successful intercept becomes negligible despite nominal functionality of terminal hardware.

All scenarios have the same event structure, which is defined in column 3 (Dependency – Parent) and illustrated by DAG in Fig. 6.

4.2 Model verification and baseline comparison

Before evaluating the three operational scenarios, the simulator was verified using analytical cross-checks. For a purely serial chain of independent binary events, the closed-form reliability of the chain is based on Eq. (8) and for a parallel redundancy structure with two independent alternative branches, the equivalent success probability is based on Eq. (9).

The verification process consisted of three steps. First, single-event test cases were executed to confirm that binomial, Poisson, geometric, normal, exponential, and uniform sampling produced values consistent with their parameterisation. Second, simple serial and parallel dependency graphs were compared with the analytical reliability expressions above. Third, extreme input cases were tested: probabilities equal to 0, probabilities equal to 1, disconnected events, and invalid cyclic dependencies. The graph-based implementation rejected cyclic structures and executed only valid directed acyclic graphs, see Table II.

The deterministic multiplication baseline is intentionally simpler than the proposed simulation model. It assumes independent binary events and does not represent alternative paths, conditional execution, or heterogeneous event distributions. Nevertheless, it provides a useful sanity check. Agreement between the analytical baseline and the Monte Carlo result for simplified test graphs indicates that the event-dependency logic is implemented correctly. Deviations in the full air-defence scenarios are expected because the proposed model includes parallel branches, conditional execution, and non-binary performance variables.

Table II: Verification table.

Test case	Analytical result	Monte Carlo result	Interpretation
Single binary event, $p = 0.900$	0.900	approx. 0.900	Sampling and I/O verified
Serial chain (0.950, 0.900, 0.850)	0.727	approx. 0.727	Serial chain verified
Parallel chain (0.950, 0.900, 0.850)	0.99925	approx. 0.99925	Parallel chain verified
Binary event, $p = 0$	0.000	0.000	Boundary verified
Binary event, $p = 1$	1.000	1.000	Boundary verified

4.3 Scenarios implementation and results

According to the Table I, the data were inserted, executed and evaluated in the simulator, see Fig. 6 and Table III, which is described probability degradation in each step.

The analysis indicates that upstream redundancy is the primary determinant of overall system survivability. In addition to the simulation results, a deterministic multiplication baseline was calculated for the binary parts of the engagement chain. This baseline is useful because it shows the probability loss that would be expected if all stages were treated as independent serial events. The comparison confirms the expected tendency: the serial baseline

produces a monotonic decrease in reliability as additional kill-chain stages are added, while the proposed dependency model additionally captures the effect of alternative engagement paths and logical bottlenecks. The largest difference appears at the engagement execution stage, where the convergence of fighter-aircraft and SAM fire-unit branches changes the cumulative probability more strongly than a simple linear sequence would suggest.

Table III: Cumulative probability degradation in the engagement chain.

Event ID	Event name	Scenario 1	Scenario 2	Scenario 3
E1	AO Detection	99.75 %	95.00 %	80.00 %
E2	ID Assignment	99.72 %	90.27 %	62.15 %
E3	Proper ID	99.68 %	85.76 %	55.93 %
E4	Hostile Track P.	98.68 %	81.47 %	50.34 %
E5	Weapon Assign.	97.69 %	77.40 %	45.31 %
E6	Unit Readiness	97.65 %	73.53 %	36.25 %
E7a	FA Mission Ex.	97.65 %	73.53 %	36.25 %
E7b	SAM Tracking	97.65 %	73.53 %	36.25 %
E8	Engagement Exec.	94.41 %	51.36 %	11.82 %
E9	Guidance Exec.	90.63 %	46.22 %	7.09 %
E10	Destruction	89.73 %	45.30 %	6.38 %

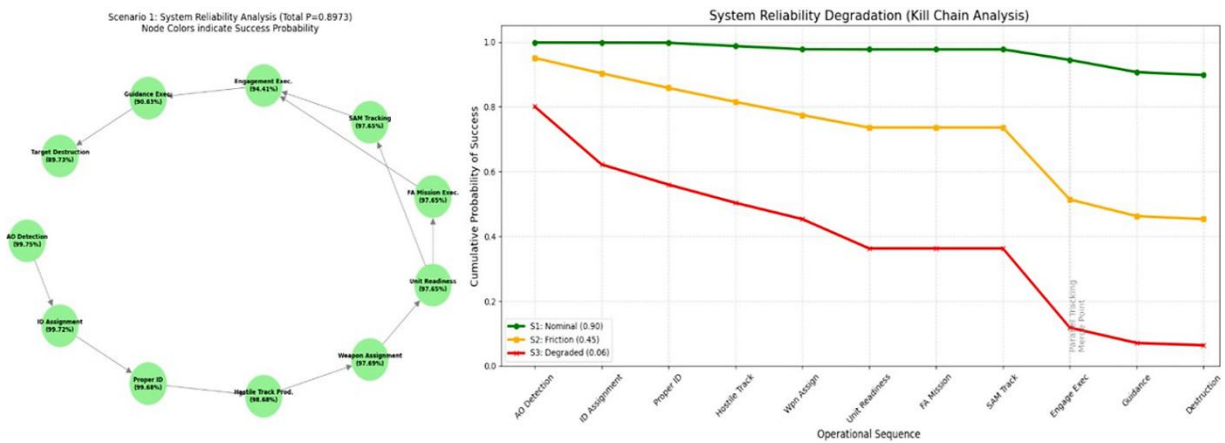


Figure 6: Generated DAG and probability degradation of kill chain.

This comparison does not constitute empirical validation, but it verifies that the simulation behaves consistently with elementary reliability theory. Empirical validation would require operational test data, classified engagement records, or controlled experimental data, which were outside the scope of this study.

Specifically:

In **Scenario 1**, redundant verification stages mitigate probabilistic degradation, enabling the system to withstand the logical penalty imposed at the Engagement Execution bottleneck and achieve an overall success probability of 89.73 %.

In contrast, **Scenario 2** lacks such mitigating mechanisms; cumulative operational inefficiencies progressively reduce system reliability, and the convergence at the engagement stage results in an approximately halved success probability of 45.30 %.

Scenario 3 is characterized by early and compounding failures within the kill chain, with system reliability falling below 50 % prior to weapon assignment, ultimately yielding a near-total mission failure probability of 6.38 %.

The gathered data verifies the functional behaviour of the simulator for the defined scenarios and confirms that the calculated probability trends are consistent with the assumed scenario severity.

4.4 Discussion and limitations

The presented case study has several limitations. First, the numerical probabilities are illustrative and are not derived from classified operational test data. Second, the current model assumes static event parameters within each scenario. Third, event correlations are not yet fully represented, although in real air-defence operations detection, identification, tracking, and guidance performance may be affected by common environmental or electronic-warfare factors. Fourth, continuous event variables require threshold-based interpretation before they can be incorporated into cumulative mission success probabilities. These limitations do not reduce the value of the model as a transparent framework for scenario comparison, sensitivity analysis, and methodological development.

5. CONCLUSION

This research builds upon and expands the methodology established in a previous study [1], which utilized Ordinary Differential Equations (ODE) and Discrete Event Simulation (DES) to create deterministic time models of air defence.

The main achievement of this paper is the transformation of a representative air-defence engagement procedure into a configurable stochastic dependency model. The proposed framework combines event-based Monte Carlo simulation, directed acyclic dependency graphs, and a simple graphical interface for model construction. This makes it possible to examine not only final target-destruction probability, but also the step-by-step degradation of success probability along the engagement chain.

The verification process showed that the simulator is consistent with closed-form reliability calculations for simplified serial and parallel dependency structures. The comparison with a deterministic multiplication baseline confirmed that the proposed model behaves as expected in elementary cases while offering greater flexibility for scenarios involving redundancy, conditional execution, and heterogeneous event distributions.

The application example demonstrated that early kill-chain stages have a decisive influence on final system effectiveness. Detection, identification, weapon assignment, and readiness errors propagate through the dependency graph and substantially reduce the final probability of target destruction. The results also show that redundancy in early and intermediate stages can mitigate probability degradation, whereas degraded sensor and guidance conditions may lead to rapid collapse of overall engagement effectiveness.

The numerical results should be interpreted as illustrative because the input parameters were not derived from empirical operational data. Nevertheless, the framework can support system design, capability assessment, training, and decision support when suitable data are available.

Future work will focus on empirical calibration, sensitivity analysis, correlated event modelling with neural network implementation based on reference [13], dynamic parameter adaptation, and integration with more detailed missile and command-and-control simulation models.

ACKNOWLEDGEMENT

The work presented in this article has been supported by the Czech Republic Ministry of Defence – University of Defence Brno development program “Conduct of airspace operations” DZRO AIROPS.

REFERENCES

- [1] Stefek, A.; Casar, J.; Stary, V.; Gacho, L. (2022). Coupling of ODE and DES models for simulation of air defence in war-gaming experiment, *International Journal of Simulation Modelling*, Vol. 21, No. 1, 41-52, doi:[10.2507/IJSIMM21-1-586](https://doi.org/10.2507/IJSIMM21-1-586)
- [2] Ham, W. K.; Kwon, Y.; Park, S. C. (2014). Combat simulation framework including continuous detection system, *International Journal of Simulation Modelling*, Vol. 13, No. 4, 395-408, doi:[10.2507/IJSIMM13\(4\)1.262](https://doi.org/10.2507/IJSIMM13(4)1.262)
- [3] De Nijs, P. Simulation-Based Assessment of Passive Airbase Defences in Peer Warfare (Finabel – The European Land Force Commanders Organisation), from <https://finabel.org/simulation-based-assessment-of-passive-airbase-defences-in-peer-warfare/>, accessed on 11-02-2026
- [4] Vu, D. H.; Baláž, T.; Racek, F. (2017). Optimization of defence ability of defence structures by using Monte Carlo simulation, *Advances in Military Technology*, Vol. 12, No. 2, 205-215, doi:[10.3849/aimt.01149](https://doi.org/10.3849/aimt.01149)
- [5] Architecture Capability Team, Digital Policy Committee. NATO Architecture Framework, Version 4.1, from <https://www.nato.int/content/dam/nato/webready/documents/publications-and-reports/NATO-Architecture-Framework-v4-1-en.pdf>, accessed on 01-05-2026
- [6] Stary, V.; Farlik, J.; (2020) Aspects of Air Defence Units C2 System Modelling and Simulation, Mazal, J.; Fagiolini, A.; Vasik, P. (Eds.), *Modelling and Simulation for Autonomous Systems (MESAS 2019)*, Lecture Notes in Computer Science, Vol. 11995, Springer, Cham, 351-360, doi:[10.1007/978-3-030-43890-6_28](https://doi.org/10.1007/978-3-030-43890-6_28)
- [7] Tobisova, A.; Kalavsky, P.; Senova, A.; Rozenberg, R. (2024). Application of simulation models for decision-making processes in aviation companies, *International Journal of Simulation Modelling*, Vol. 23, No. 2, 299-310, doi:[10.2507/IJSIMM23-2-686](https://doi.org/10.2507/IJSIMM23-2-686)
- [8] GitHub repository, Stary, V. AirDefenceProbabSim: Probability Modelling of Air Defence Procedures, from <https://github.com/staryv/AirDefenceProbabSim/>, accessed on 01-05-2026
- [9] Farlik, J.; Hamtil, I.; Krátký, M. (2019). Modelling missile flight characteristics by estimating mass and ballistic parameters, *Advances in Military Technology*, Vol. 14, No. 1, 47-57, doi:[10.3849/aimt.01247](https://doi.org/10.3849/aimt.01247)
- [10] Wade, B. M. (2021). Creating surrogate models for an air and missile defense simulation using design of experiments and neural networks, *The Journal of Defense Modeling and Simulation*, Vol. 18, No. 4, 273-284, doi:[10.1177/1548512919877987](https://doi.org/10.1177/1548512919877987)
- [11] Yao, M.; Li, Z. (2025). A Monte Carlo simulation-based method for generating virtual test scenarios in virtual reality environments, *Technical Gazette*, Vol. 32, No. 2, 389-396, doi:[10.17559/TV-20240306001380](https://doi.org/10.17559/TV-20240306001380)
- [12] Law, A. M. (2024). *Simulation Modeling and Analysis*, 6th edition, McGraw Hill, New York
- [13] Raska, P.; Ulrych, Z.; Baloun, J.; Malaga, M.; Lenc, L. (2024). Using adaptive neural networks for optimising discrete event simulation, *International Journal of Simulation Modelling*, Vol. 23, No. 2, 227-238, doi:[10.2507/IJSIMM23-2-678](https://doi.org/10.2507/IJSIMM23-2-678)