eISSN 2783-6851

VILNIUS TECH
Vilnius Gediminas Technical University

NEW TRENDS in COMPUTER SCIENCES

2025
Volume 3
Issue 1
Pages 18–37
https://doi.org/10.3846/ntcs.2025.23627

DYNAMIC MULTI-SCALE SIMULATION FOR EVALUATING COMBAT EFFECTIVENESS AGAINST AERIAL THREATS

Irene NDINDABAHIZI[1]✉, Tom VANCAEYZEELE[1], Ben LAUWENS[2] Johan GALLANT[1]

[1]*Department of Weapon systems and Ballistics, Royal Military Academy, Brussels, Belgium*
[2]*Department of Mathematics, Royal Military Academy, Brussels, Belgium*

Article History:
- received 31 March 2025
- accepted 14 May 2025

Abstract. The development and assessment of modern weapon systems require efficient and flexible simulation tools. This paper introduces a multi-scale discrete-event simulation framework designed to evaluate the dynamic combat effectiveness of weapon systems. The framework combines high-resolution and low-resolution models to address the complexities of real-world engagements while maintaining computational efficiency. Physical processes are encapsulated as modular state transition functions, allowing seamless integration of a multi complexity level modeling approach. The framework's versatility is demonstrated through a case study analyzing the effectiveness of a tank weapon system against a fleet of drones. Non-deterministic methods such as Monte Carlo simulations for uncertainty quantification are used to evaluate probabilistic key metrics, such as projectile accuracy and lethality, providing insights into engagement dynamics and optimization of firing strategies. By leveraging a hybrid continuous/discrete approach and modular design, the framework enables comprehensive assessments of weapon effectiveness during an engagement, bridging gaps in traditional deterministic methodologies for both static and dynamic targets. Future enhancements will focus on optimizing sampling techniques for broader applicability of high-resolution stochastic simulations in modern combat scenarios.

Keywords: discrete event simulations, Julia, Monte Carlo, uncertainty quantification, error budget, multi-scale simulations.

✉Corresponding author. E-mail: ***irene.murebwayirendindabahizi@mil.be***

1. Introduction

Tracked armoured fighting vehicles have dominated land warfare for years due to their robust design and heavy firepower. However, modern conflicts have introduced new threats particularly from the air. The proliferation of low-cost armed Unmanned Aerial Vehicles (UAVs), being used against high value targets such as army vehicles, has exposed vulnerabilities in traditional tank designs. Typically, radio-controlled First Person View (FPV) drones – costing approximately \$500 – equipped with Improvised Explosive Devices (IEDs) fly over a tank – costing approximately \$3 million – and dive down to attack, exploiting the vulnerable top of the tank (Kunertova, 2024). Originally tanks were designed for surface-to-surface combat, so, the armour protection is essentially focusing in protecting against direct fire from other heavy weapons. The M-1 Abrams, for instance, has an advanced composite armour that only covers the front of the hull and the side skirts. Its thickness at the front is about 0.61 m and provides protection equivalent to a thousand millimetres of steel. In contrast, the upper protection is only 25 millimetres of steel (Green & Stewart, 2005).

FPVs are small, fast, agile and cheap. They are difficult to detect and can be used in large volumes to saturate defences (Molloy, 2024). Tanks are mostly ill-equipped to engage such low flying small aerial targets. Addressing this new reality with cost effective measures, requires innovative approaches to evaluate and enhance kinetic tank defenses against aerial threats. To address these challenges, this paper introduces a multi-scale discrete event simulation framework tailored for weapon effectiveness analysis. The simulation computes how many bursts of projectiles are required to incapacitate the threat before it reaches a threshold distance, where, even if destroyed, it can still pose a threat to the vehicle. The use of discrete-event simulation allows not only the simultaneous tracking of the movement of the tank, the targets and the projectiles entities, but also a multi-scale approach. Physical entities are agents defined by their characteristics (state variables) and are updated during the simulation. Physical laws model the state transitions and are implemented as black boxes within modules. They are used to update the state variables. Employing interchangeable physical models encapsulated in modular components, offers flexibility in combining high- and low-resolution simulations as required.

High-resolution methods, such as Monte Carlo (MC) simulations, are integrated to propagate uncertainty and analyze the stochastic behavior of key parameters. This approach enables comprehensive assessments of weapon systems, including projectile trajectory accuracy and lethality against aerial targets. MC is a sampling-based method. It is used to generate a distribution of points representing delivery accuracy. From this distribution, the likelihood of damage to the target can be computed. The probability of damage can be determined based on low-resolution closed form damage functions or a high-resolution damage matrix based on MC simulations. Chusilp et al. (2014) investigated the difference between a high-resolution damage matrix and a closed form damage function for area targets. These methods were developed for static surface targets. The discrete event modelling approach enables dynamic computation of the weapon's effectiveness by updating the parameters of the damage functions for each projectile/target encounter and is well fitted for moving targets. The modelling of the delivery accuracy of weapon systems has been presented in Driels (2004) for different levels of resolution. For a tank weapon system, a low-resolution model computation of delivery accuracy for surface targets is presented in Bunn (1993). For aerial targets, a high-resolution method based on uncertainty propagation using MC is presented in Ndindabahizi et al. (2022).

The proposed framework uses a hybrid continuous/discrete approach to simulation. Physical laws, which are described by differential equations are solved using continuous time solvers and are embedded in discrete-event processes. These processes are then queued and time managed by ConcurrentSim, a discrete-event package written in Julia (Bezanson et al., 2017). The remainder of this paper is organised as follows:

Sections 2 and 3 describe the methodology behind the proposed simulation framework, detailing its modular design and integration of physical models. Microscopic-level simulation is introduced to ballistics by simulating each projectile in a salvo as an entity and at macroscopic-level the salvo effect is computed using probabilistic functions. Uncertainty quantification simulations are based on these microscopic simulations where each parameter identified as a source of error is sampled. These sampled parameters are then used to generate new projectile entities/agents. Each of the sampled projectiles are then simulated and at the macroscopic level, probabilistic distributions are used to describe the behavior of the system.

Section 4 section describes the modelling of the discrete event processes and libraries used by the simulation. As an illustrative application, the framework evaluates the effectiveness of a tank weapon system engaging a fleet of UAVs using different projectiles. The study emphasizes the use of modular discrete-event processes to manage system interactions dynamically, offering a robust methodology to simulate real-world combat scenarios efficiently.

Section 5 introduces the case study parameters, including the modelling of weapon systems and UAVs.

Section 6 presents the results of the case study, discussing key metrics such as accuracy, kill probability, and engagement dynamics. The paper is then concluded by summarizing the findings and outlining future research directions.

This paper is an extension of work originally presented in the 38th annual European Simulation and Modelling Conference.

2. Measure of weapon accuracy

The weapon delivery accuracy is usually computed during the error budget evaluation process. This process consists of identifying and quantifying sources of weapon delivery error. The errors are assumed to be normally distributed and are expressed by their standard deviation (σ).

A low-resolution method based on unit effects (UFs) is commonly used (Strohm, 2013). UFs provide a measure of the miss distance caused by a unit error in a parameter (see Eq. (1)). They are range dependent and are often tabulated in shooting tables. UFs are computed using trajectory computations and are used to evaluate the accuracy of the shot.

$$UF = \frac{\Delta impact\ po\mathrm{int}}{\Delta x_i}, \tag{1}$$

where x_i is a parameter that is known with uncertainty (source of error). This approach assumes a linear relationship between error sources and the accuracy of the shot. The error is quantified parameter by parameter. Assuming that the errors are uncorrelated, they can be easily combined by assuming a normal distribution. For a given range, the error contribution of a parameter i is computed by Eq. (2):

$$\sigma_i\left(range\right) = \sigma_{source} * UF\left(range\right), \tag{2}$$

where σ_{source} is the error on the parameter (error source) and $UF(range)$ is the unit effect at the specified range. Figure 1 shows the processes involved in the computation of UFs. The total error is then computed by the root square of the sum of the variances of all error sources (Eq. (3)):

$$\sigma_{total}\left(range\right) = \sqrt{\sum_{i=1}^{i=N} \sigma_i\left(range\right)^2}\ . \tag{3}$$

A high-resolution method based on stochastic simulation can also be used (Zhang, 2020). Maintaining the assumption of uncorrelated error sources, Eq. (3) can be used in combination with Monte Carlo (MC) based simulations. Each error is propagated individually. Figure 2 shows the processes involved in this approach. The parameters described by a probability distribution are sampled. For each sample, a trajectory simulation is performed. The MC module then provides a distribution of detonation points which in turn can be fitted to a probabilistic distribution. Maintaining the assumption of non-correlation of error sources, these errors can also be combined if we assume a normal distribution. An additional stochastic approach to error propagation uses a simultaneous propagation of all errors through MC simulations. Figure 3 shows the processes involved in such simulation. All parameters are simultaneously sampled using their normal distributions. Trajectory computations are conducted for each set of the sampled data. A distribution of impact points is

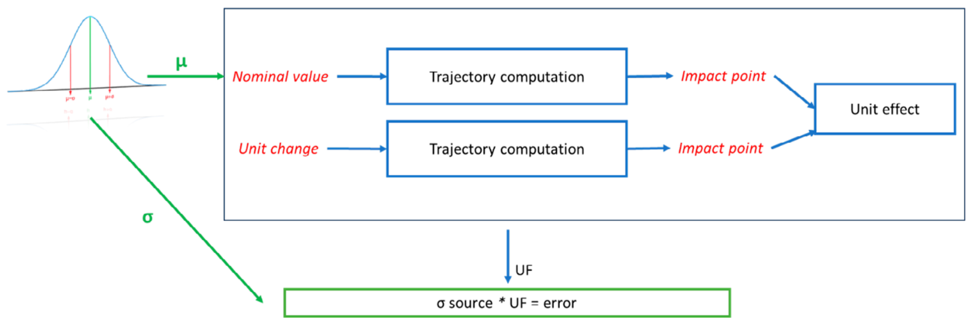

Figure 1. Computational method for weapon system delivery accuracy using Unit effects

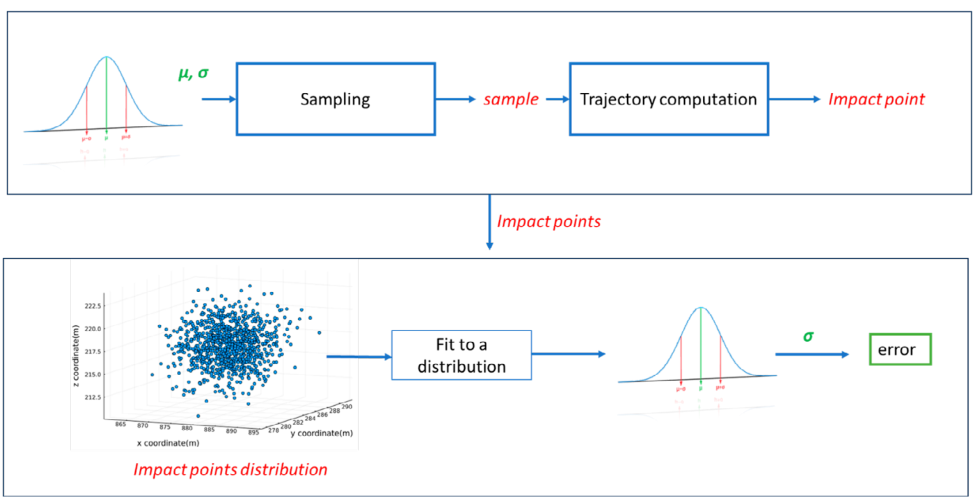

Figure 2. Computational method for weapon system delivery accuracy using MC simulations parameter per parameter

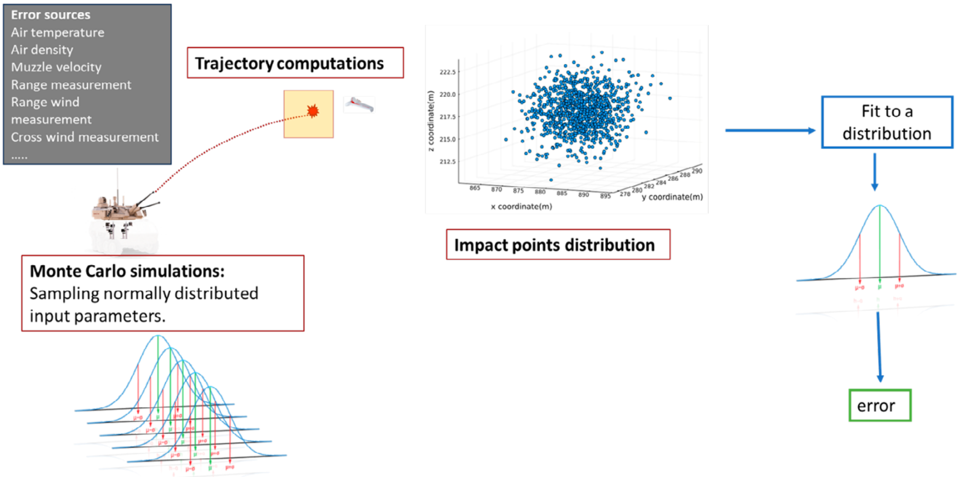

Figure 3. Computational method for weapon system delivery accuracy using MC simulations all parameters simultaneously

obtained which can be fitted to a normal distribution and provides directly the total error. Even with the computation cost associated with this method it can be attractive due to the fact that it accounts for parameters interactions. All these methods use trajectory simulations to propagate the errors to the target. McCoy (1999) describes a series of trajectory models used for spin-stabilised projectiles.

These models are described using differential equations: a 3 Degrees Of Freedom (3DOF) Point Mass Model (PMM) where the drag and the gravity are the only forces considered, a Modified Point Mass Model (MPMM) which adds an extra degree of freedom and a Six-Degree-Of-Freedom (6DOF) model which solves the 6 equations of motion. The choice of model is determined not only by the computational capacity, but also by the availability of the different aerodynamic coefficients used in these models.

From these simulations we can compute the direct hit probability. For simple geometry targets, there are closed form functions to compute the Single Shot Hit Probability (SSHP) (Przemieniecki, 2000). For instance, the SSHP on a rectangular target is given by:

$$P_{hit} = \iint_{TargetArea} p(x,y)\,dxdy\,, \tag{4}$$

where *p*(*x*,*y*) is a probability distribution. For independent horizontal *x* and vertical *y* errors, it is simply the product of the horizontal and vertical probabilities:

$$p(x,y) = p(x)\,p(y)\,. \tag{5}$$

For normal density distributions centered on the aiming point we have for a rectangular target:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_x}} \exp\left[-\frac{x^2}{2\sigma_x^2}\right]; \tag{6}$$

$$p(y) = \frac{1}{\sqrt{2\pi\sigma_y}} \exp\left[-\frac{y^2}{2\sigma_y^2}\right]. \tag{7}$$

The probability of hitting a rectangular target of dimensions 2a×2b can then be computed directly by integration:

$$P_{hit} = \int_{-a}^{a} p(x)\,dx \int_{-b}^{b} p(y)\,dy\,. \tag{8}$$

This approach can be extended to circular targets. For more complex target geometry, stochastic shotline-based techniques can be used to evaluate the hit probability.

3. Measure of weapon effectiveness

The weapon's effectiveness is a measure of merit for a given weapon system-target engagement scenario. The effectiveness measures are based on methodologies that use damage functions. These damage functions return the estimated damage to the target. They give the probability of damaging the target for either a direct hit or fragmentation of the munition $P_{k|h}$ (Ahner & McCarthy, 2018). The methodology presented in this paper is for fragmenting munitions. As presented in the previous section, the accuracy of the weapon is represented in this framework by a probability distribution. For a fragmenting munition, the detonation points are sampled from this distribution. The detonation points can then be used to compute the weapon's effectiveness in combination with the damage functions.

For high-resolution methods, the damage functions can be computed in matrix form. These methods are used for targets where extensive knowledge is available on the structure and its vulnerability (Deitz et al., 2009). In these methods each fragment trajectory from the detonation point to the target is computed taking into account the deceleration due to the drag force. The fragment path through the target is simulated using shotlines. For each individual fragment *j*, perforation models are used to evaluate the fragment lethality. Damage functions are then used to evaluate the probability of the fragment killing the target ($P_{k,j}$).

The overall kill probability for the projectile is then given by:

$$P_k = 1 - \prod_{j=1}^{n} \left(1 - P_{k,j}\right), \tag{9}$$

where *n* is the total number of fragments hitting the target.

Lower-resolution methods aggregate the fragment into a fragment zone (see Figure 4). The probability of a fragment hitting the target is computed by (Åkersson, 2022):

$$P_{hit} = \frac{A}{A_{zone,i}}, \tag{10}$$

where *A* is a projected area of the target prependicular to the fragment's path, $A_{zone,i}$ is the *i-th* fragment zone hitting the target. Within an $A_{zone,i}$ fragments are defined with the same characteristics (mass, density, velocity, etc). For this approach the number of fragments within the $A_{zone,i}$ that effectively perforate the target (η_{eff}) are evaluated. η_{eff} can be computed using perforation equations or simpler models based on the fragments kinetic energy. The fragments within the same $A_{zone,i}$ share the same kill probability ($P_{k,f}$). Therefore, Eq. (9) can be written as:

$$P_k = 1 - \left(1 - P_{k,f}\right)^{\eta_{eff}} . \tag{11}$$

In the context of small flying targets engaged with fragmenting projectiles, no geomety of the target is avaible. Therefore, given the size of FPV drones, a method that uses the Centroid Of Vulnerability (COV) instead of the detailed geometry is proposed. For this approach, a closed-form damage functions, such as the cookie-cutter damage function is used (Przemieniecki, 2000). The cookie-cutter is a step function defined by a lethal area outside which no damage is sustained, whereas inside the area the target is killed. Continuous functions which decrease with the distance from the detonation point can also be used. For each sampled detonation point, a cone of lethality is defined (lethal area). The dimensions of the cone are based on the fragment characteristics and the projectile state at the time of explosion (see Figure 5). If the COV is within the cone, then the damage to the target is assessed. If the target is within the lethal cone, the kill probability is equal to 1.

For a salvo of s rounds, a kill probability is computed for each round and combined to obtain the global kill probability (P_k) of the burst of s rounds through (Przemieniecki, 2000):

$$P_k = 1 - \prod_{j=1}^{s} \left(1 - P_{k,j}\right), \tag{12}$$

where $P_{k,j}$ is the kill probability of shot j. For a burst of projectiles, $P_{k,j}$ is modified to account for the projectiles in the burst.

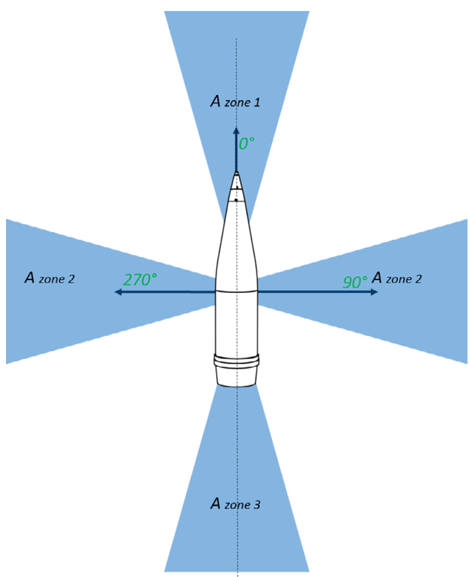

Figure 4. Fragment zones generated at projectile detonation

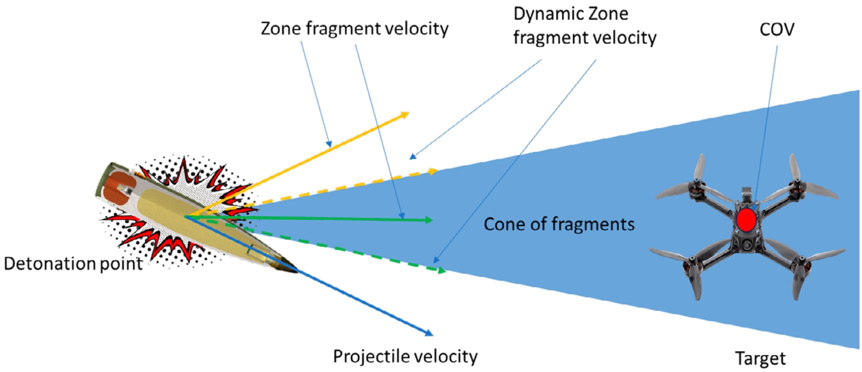

Figure 5. Cone of lethality

4. Discrete-event simulation implementation

4.1. Models libraries

In this paper, the simulation models the processes taking place in a tank Fire Control System (FCS) as presented in Figure 6. Using a discrete-event approach, it is possible to simulate the interaction of the different entities represented. When the commander identifies a target he issues a fire command to the gunner who tracks the target using his sights. The ballistic computer computes a firing solution which provides line of fire (LOF) angles. The target can then be engaged and damage assessed. For a salvo of s rounds fired at a moving target, the fire control operations are repeated for each round in order to compute the lethality as shown in Figure 7. Modelling the engagement using a discrete-event simulation approach allows jumping from event to event, speeding up the simulation time.

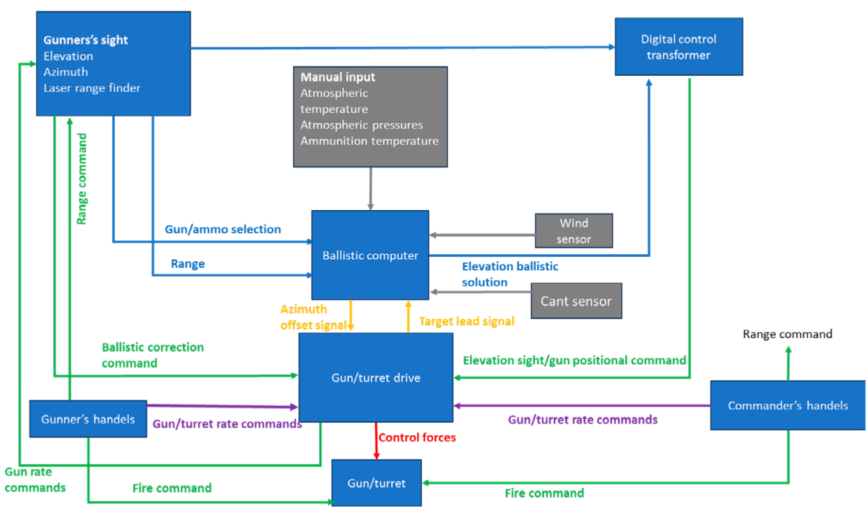

Figure 6. Tank fire control unit

To simulate the entire engagement process, there is a wide variety of physical phenomena to take into account. Physical models are implemented as modules for flexibility, this way, they can be easily accessed, extended and swaped. These models are stored in libraries implemented using Julia package structure.

The WeaponSystems package generates weapon systems as entities. State variables are defined under the generated weapon system as parameters. For example, in this paper a tank entity is defined. To create a tank, the following parameters need to be specified: hull, turret, canon and sight (see Figure 8). The position of the weapon system is uniquely defined by its altitude and latitude. For a weapon system in motion, its initial position and velocity can also be specified.

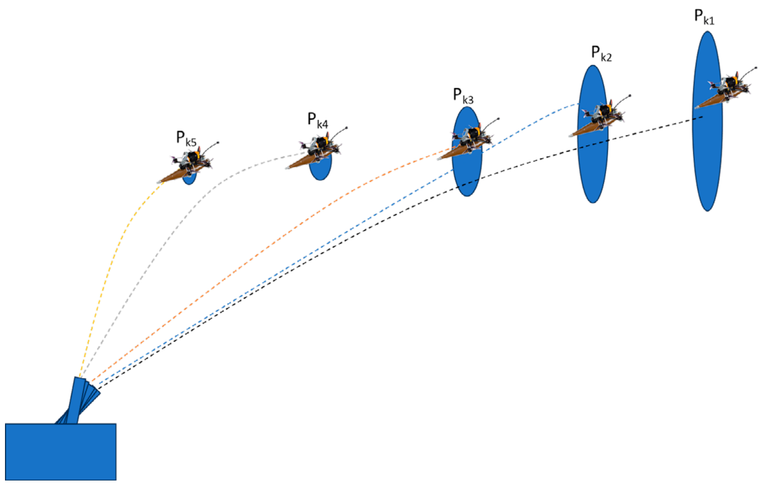

Figure 7. Tank weapon system effectiveness along UAV trajectory

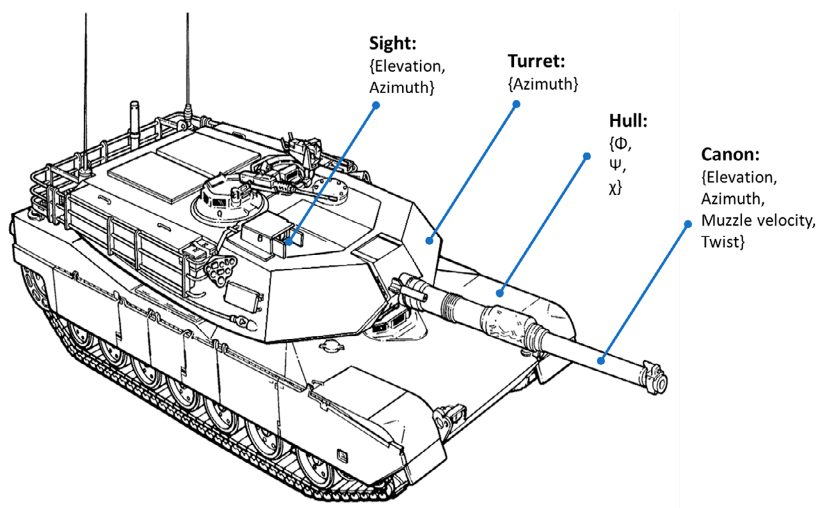

Figure 8. Tank weapon system

The ErrorBudget package propagates errors using MC simulations or UFs. In the context of stochastic simulations, the number of MC runs can be specified as input. The module then gives a probabilistic distribution of impact points. Errors on state variables expressed as distributions are used as input. For tabulated errors, delivery errors are computed by linear interpolation.

The ExternalAerodynamics package includes external ballistic models for the computation of projectile trajectories. The available models are PMM, MPMM and a 6 DOF model. The differential equations are solved using Julia's package DifferentialEquations.jl (Rackauckas & Nie, 2017).

The Cuas package is used to define the simulation framework. It generates engagement scenarios for countering UAVs by scheduling the various events involved in the "kill chain" (Dominicus, 2021). The multiple dispatch capability in Julia is used to automatically select the right scenario based on the inputs. Available scenarios are static/mobile, mobile/mobile, static/static for single and multiple threats.

The Cuas package is the main package that also contains functions that computes delivery accuracy. The implemented methods use the errors computed by the ErrorBudget package to evaluate hit probabilities. Low and high resolution damage assessment functions are also implemented in this package.

Within these libraries, various models can be combined to produce different simulations. Functions have been implemented using several methods. Based on the number or type of arguments, the simulation is adapted and the appropriate method is selected (Multiple dispatch). Figure 9 shows the structure of the implemented firingChain function, which has multiple methods. At a high level three methods for the function, corresponding to the different scenarios considered are implemented. Julia uses runtime dispatch, so, the scenario is selected at compilation time. Depending on the type of target, the appropriate method will be chosen for the weapon effectiveness evaluation. The code continuously adapts and generates the appropriate simulation.

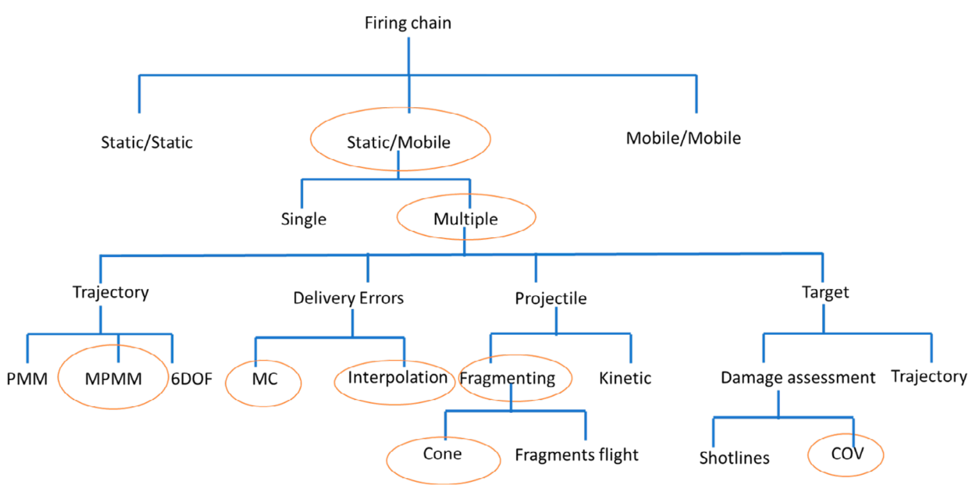

Figure 9. Multiple method function structure

4.2. Kill chain

The kill chain events implemented in the Cuas package are Find, Fix, Track, Engage, and Assess. Commonly known as the F2T2EA (Penney, 2023). Figure 10 shows a conceptual view of the simulation framework. These steps have been implemented as follow:

Find: Target detection occurs when a potential threat is detected in the surveillance area. The search operation depends on the weapon system. For a tank, it is primarily carried out by optics or thermal imaging. The optics are then in wide field of view (FOV) mode. The commander and gunner can have independently controlled sights, allowing the ability to simultaneously perform surveillance operations by the commander and target engagement operations by the gunner. This means that a new target can be acquired during simulation while engaging others. The code translates this into functions that generate targets. The time delay for this operation is the detection time (Δt_{det}).

Fix: Includes the recognition and identification operations. This is where the detected object is identified as a truck, tank, UAV or helicopter. The identification operation is performed in Δt_{reco} time. For this operation, the optics are switched to a higher resolution narrow FOV. The target is then identified as friendly, enemy or neutral with an associated identification time Δt_{id} . The commander then decides to engage the target (Δt_{dec}). The target is handed over to the gunner, who performs the acquisition for an engagement in $\Delta t_{firecommand}$.

Track: If the threat is confirmed, it is tracked continuously. After handover, the gunner's sight is slewed and aligned with the commander's sight along the line of sight (LOS) in Δt_{aim} + Δt_{slew}. This operation provides the target position and predicts the target position at one time of flight (TOF) in the future. The azimuth target velocity is estimated from the tracking commands and a kinematic lead component is computed to compensate for target motion during projectile flight.

Engage: The LOF must be determined. The computation of the weapon aiming point requires ballistic computations of the projectile trajectory and the target displacement during the projectile flight. Range operations are then performed to determine the range of the target ($\Delta t_{ranging}$). Projectile characteristics are used as input to the ExternalBallistics package to generate the projectile entity. A ballistic computer module integrated into the Cuas package provides the TOF. This TOF is then used for the prediction of the lead angle for a new intercept point ($\Delta t_{firecontrol}$). The gun mount is then positioned along the computed angles with respect to the current target position (see Figure 10). The round is then fired ($\Delta t_{firedemand}$ + $\Delta t_{firedelay}$ + $\Delta t_{shotexit}$ + TOF). A detonation point is determined for a fragmenting projectile. Error sources are passed to the ErrorBudget package to be propagated to the intercept point (see Figure 11). The ErrorBudget outputs a distribution of the detonation points.

Assess: After firing at a target, this step determines whether sufficient damage has been inflicted to allow terminating the engagement. Weapon delivery accuracy and damage assessment computations are performed. The damage assessment process involves MC simulations based on the sampling of detonation points. For each MC run, a vulnerability cone is generated based on the fragmentation characteristics of the projectile. If the centroid of the target is inside the cone, it is considered a successful hit. The MC simulations return a hit probability. The target damage function is then used to assess the amount of damage encountered ($\Delta t_{damageassessment}$ + $\Delta t_{reengagementdelay}$).

The shooter-to-target time window T can be defined as:

$$T = \Delta t_{\det} + \Delta t_{reco} + \Delta t_{id} + \Delta t_{dec} + \Delta t_{firecommand} + \Delta t_{aim} +$$
$$\Delta t_{slew} + \Delta t_{ranging} + \Delta t_{firecontrol} + \Delta t_{firedemand} + \Delta t_{firedelay} + \Delta t_{shotexit}. \quad (13)$$

During an engagement multiple rounds are fired and the engagment time necessary to defeat the target is used as a measure of effectiveness. The engagment time depends on T, the number of rounds fired, the TOF, the $\Delta t_{damageassessment}$ and $\Delta t_{reengagementdelay}$.

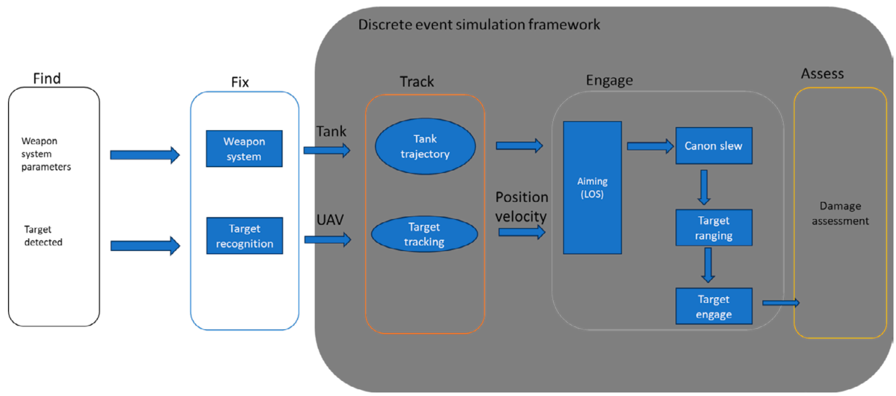

Figure 10. Conceptual view of the discrete-event framework

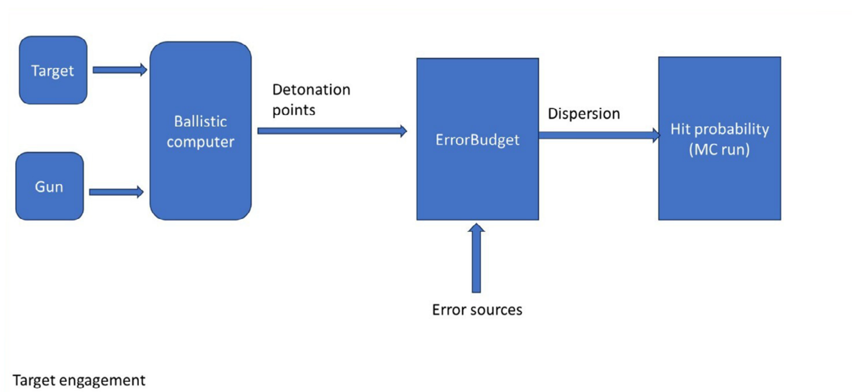

Figure 11. Modular breakdown of the target engagement process

In this paper we will present a shoot-look-shoot policy (Seo et al., 2012) but a shoot-shoot-look policy can be implemented by simply changing the order of the processes. At the end of the assess process, if the target is still alive, it is re-engaged.

To achieve modularity, the implemented framework uses a hierarchy of processes. Figure 11 shows an example of sub-processes involved in the the target engagement process (ballistic computer, error budget and the hit probability).

5. Case study

5.1. Description

For illustration purposes, a case study of a tank engaging three UAVs flying towards the weapon system at a constant speed is considered. A shooter-to-target window time of 6 seconds is assumed. The relative distance between the UAVs remains unchanged. Therefore, the three UAVs can be seen as a single target with three centres of vulnerability (see Figure 12). A static/mobile scenario is chosen. The weapon system is static and the target is moving. The case study combines high and low resolution modelling approaches.

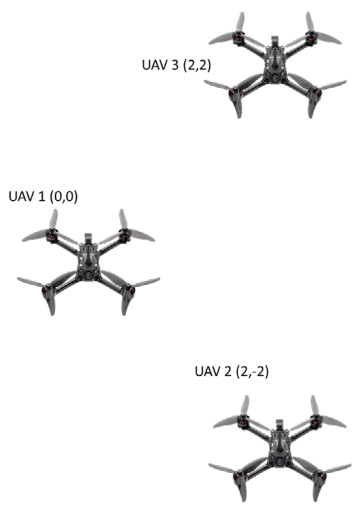

Figure 12. Targets relative positions

The model resolution level is chosen based on the information available. Where possible, the most accurate model is selected. The simulation runs until the targets are destroyed or have reached the weapon system. Then useful information such as the number of rounds required to destroy the target, the engagement time or kill distance is extracted. This information can in turn be used to make decisions about projectiles or weapon systems selections. The simulation starts when a target is identified (UAV) and a weapon is assigned (tank). First the threat is detected: three UAVs flying at the same altitude. The targets are flying in a fixed formation at a speed of 30 m/s. The relative position of the UAVs is also provided. The tank sight's elevation and azimuth angles provide the initial LOS angles of 10° and 15° respectively.

During the simulation we will compare the effectiveness of the weapon system against the targets detected at 1 km, 2 km, 3 km and 4 km. We will also vary the number of projectiles per burst.

5.2. Weapon system parameters

The tank weapon system considered is a 30 mm medium caliber weapon system integrated onto an infantry vehicle. The main armament of the tank is a NATO-standard 120 mm which is designed to engage other armored vehicles at long ranges. The secondary armament is the 30 mm canon that can fire high-explosive incendiary rounds and programmable airburst munitions. This medium caliber canon is a replacement of typically used 12.7 mm or 7.62 mm machine guns. The tank weapon system is generated using the WeaponSystems library, the user must provide the hull, turret, canon and sight information. The projectile is generated using the ExternalBallistics library by providing its mass and calibre. Optionaly, the user can provide the position, the velocity, the TOF, the momentum of inertia (I_x,I_y), the position of the center of gravity, the spin velocity, the angle of repose and the aerodynamic coefficients. The 30×173 mm PGU-13/B projectile data are used for trajectory computations. Firing accuracy computations are based on the MPMM trajectory model. Two types of fragmenting projectiles are used: a High Explosive Incendiary (HEI), which causes high collateral damage and is not recommended in urban areas, and a Kinetic Energy Time Fused (KETF) round with low collateral damage and therefore better suited for urban use. The projectile lethal area is defined using fragments zone data, and is generated using the ExternalBallistics library. The fragments zone data are defined by an upper and a lower angle defining the A_{zone}, the number of fragments, the mass of fragments, the velocity of the fragments and the density of the fragment's material. The projectile lethality area is defined using the dynamic zone angles of the A_{zone}. The HEI has two lethality cones defined using the $\mathrm{A_{zone}}$ angles: the first from 0° to 15° and the second from 30° to 45° and their symmetric counterpart. The KETF projectiles use only one fragmentation cone: from –15° to 15°. To increase the probability of hit, the rounds are fired in bursts. The number of rounds per bursts is varied from 1 to 5.

The delivery accuracy is computed by propagation of errors in parameters. The tank is equipped with sensors to measure temperature, pressure, density, projectile velocity, cant, range and wind speed. The measured quantities are used to compute the ballistic correction. These measurements introduce errors due to their accuracy (random component). Another source of random errors is inherent to the measured quantities (such as wind gust). Other error components which stay constant within the same occasion are assumed corrected by boresighting, zeroing and fire control (for ballistic correction). The correction operations introduce also an error component. These remaining errors are random and are the ones taken into account in the simulations. The parameters distributions are assumed to be fitted to a normal distribution. Their expectation is used as an input value and the standard deviation as a measure of the error. Table 1 shows the errors propagated using MC simulations. For

Table 1. Measurement errors

Name	Standard deviation
Target range	5 m
Muzzle velocity	6.309 m/s
Cant	0.5°
Cross wind	1.798 m/s
Range wind	3.353 m/s
Air temperature	4.444°

other parameters where error sources are difficult to measure (jump, fire control and ballistic dipersion), the contribution to the overall dispersion is tabulated. These errors are range dependent and a linear interpolation is used to compute the error (see Figure 13). A constant value expressed in mils is used for boresight (0.22 for the horizontal contribution and 0.15 for the vertical contribution).

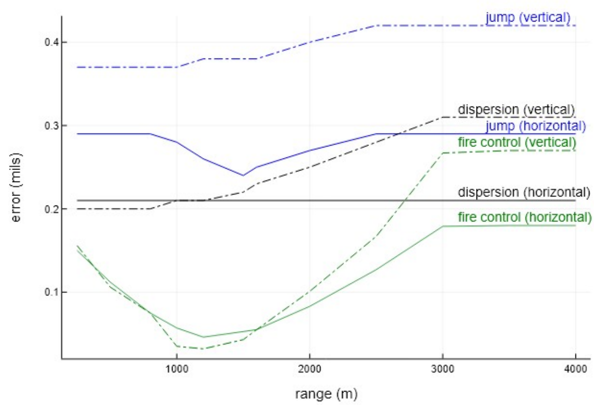

Figure 13. Jump, fire control and ballistic dispersion errors

6. Simulation results

Figure 14 shows the delivery accuracy computed for targets intercepted at different ranges. The centre of the plots is the aimpoint and the probability of detonating in the vicinity is the highest. These plots represent a bivariate probability distribution of detonation points in a plane perpendicular to the projectile trajectory. The probability of detonation near the aimpoint increases as the range decreases. These detonation points distributions are the results of stochastic error propagation simulations. Per parameter, 10k MC runs are used for error propagation. For a total of 6 parameters (Table 1), a total of 60k simulations are required to generate the detonation points distribution for each range.

Figure 15 shows the standard deviation in the vertical and horizontal directions computed from the MC simulation as a function of range. The computation of each point is calculated at the cost of 60k MC simulations. In order to speed up simulation time, a function fitted to these points can be used to predict the accuracy of the projectile for other ranges. The detonation points coordinates can be expressed as normal distributions $d_i(\mu_i,\sigma_i)$, where i = 1,2,3 and represent the 3 directions. From these distributions, detonation points are sampled to compute the damages to the target. 10k MC runs are used for the computation of the kill probability using the cone of vulnerability method. The projectile state (velocity and angle of fall) at the detonation moment is kept constant, only the position is sampled.

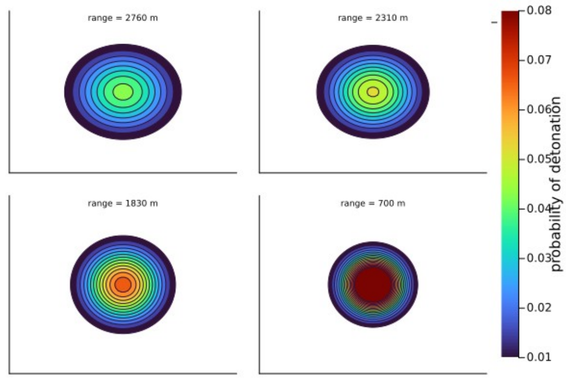

Figure 14. Probability distribution for detonation points for a target

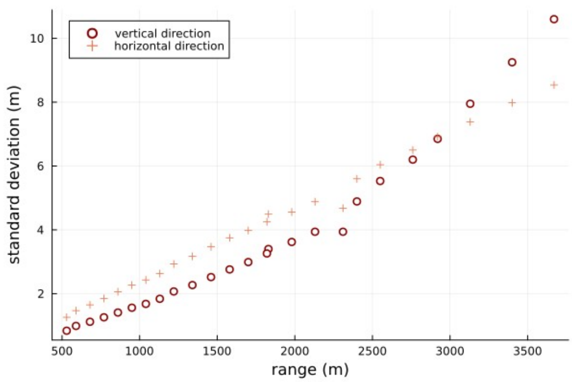

Figure 15. Error propagation in function of range

The results for targets detected at 1 km are shown in Table 2. The maximum number of projectiles considered per burst is 5. For the HEI projectile, only 3 bursts are required to defeat the targets. The first burst of projectiles will hit the targets at a range of 850 m, the second at 790 m and the third at 700 m. The KETF round requires a total of 6 bursts. The HEI round is significantly more effective than the KETF. This is to be expected as the lethal area is larger for the HEI projectile. If the number of rounds per burst is reduced, the number of bursts will logically increase. However, the total number of rounds consumed is reduced for the KETF round, while for the HEI, it remains more or less stable. The total engagement time – necessary to defeat the target – increases with the reduced number of rounds per burst. If the targets are engaged with a KETF round, there is no time to engage a second target that had been detected at the same time (the kill range is less than 500 m). If the KETF round has to be used, a new shooting strategy must be investigated or the lethal area has to be optimised. Both type of simulations can be easily done using the developed framework.

Targets detected at 2 km range, results are presented in Table 3. An extra 2 rounds of 5 HEI projectiles bursts is necessary to defeat the target. The kill probability decreases with range. Thus, the first rounds are less effective than when the targets are engaged at 1 km. The kill range is 1340 m, sufficient enough to engage other targets. As observed previously, the total number of KETF projectiles consumed is significantly decreased when we decrease the number of projectiles per burst. But, the engagement time is still increasing. Engaging the target with KETF requires almost the double the time necessary for HEI rounds. Even with 5 projectiles per burst there is still limited time to engage other targets.

Table 2. Simulation results for targets detected at 1 km range

nr of projectiles/ burst	nr of bursts	total nr of projectiles	engagement time (s)	kill range (m)	projectile
5	3	15	14	700	HEI
4	4	16	16	610	HEI
3	5	15	19	520	HEI
2	6	12	22	460	HEI
1	11	11	33	130	HEI
5	6	30	21	460	KETF
4	7	28	23	400	KETF
3	8	24	25	310	KETF
2	10	20	30	190	KETF
1	12	12	34	40	KETF

Table 3. Simulation results for targets detected at 2 km range

nr of projectiles/ burst	nr of bursts	total nr of projectiles	engagement time (s)	kill range (m)	projectile
5	5	25	26	1340	HEI
4	6	24	29	1220	HEI
3	7	21	32	1130	HEI
2	10	20	42	860	HEI
1	14	14	53	530	HEI
5	12	60	46	680	KETF
4	14	56	52	530	KETF
3	15	45	54	440	KETF
2	18	36	61	230	KETF
1	21	21	68	50	KETF

If the target is engaged at a range of 3 km, the results are shown in Table 4. The kill range is further increased, but projectile consumption is also quite high. There is an excess of 2 bursts of HEI (5 rounds per burst) necessary to defeat a target detected at 3 km compared to a target detected at 2 km. The engagement time is however almost the double. The KETF rounds at 3 km range are not very efficient in our chosen firing policy and for the projectile characteristics considered (lethal area).

Engaging targets detected at 4 km range becomes inefficient even with an HEI round. The results are presented in Table 5. A gain of approximately 250 m in kill range at the cost of 10 more rounds is observed (2 bursts of 5 rounds).

Table 4. Simulation results for targets detected at 3 km range

nr of projectiles/ burst	nr of bursts	total nr of projectiles	engagement time (s)	kill range (m)	projectile
5	7	35	44	1800	HEI
4	8	32	48	1680	HEI
3	10	30	56	1440	HEI
2	12	24	63	1200	HEI
1	20	20	87	510	HEI
5	18	90	80	660	KETF
4	19	76	83	600	KETF
3	20	60	86	510	KETF
2	22	44	90	360	KETF
1	26	26	99	90	KETF

Table 5. Simulation results for targets detected at 4 km range

nr of projectiles/ burst	nr of bursts	total nr of projectiles	engagement time (s)	kill range (m)	projectile
5	9	45	68	2050	HEI
4	11	44	77	1780	HEI
3	13	39	85	1540	HEI
2	16	32	96	1210	HEI
1	22	22	114	640	HEI

Figure 16 shows the evolution of the kill probability for each UAV for the 4 detection distances. The 3 targets are engaged with bursts of 5 HEI rounds. The kill probabilities are very similar for the 3 targets. At 1 km and 2 km, the increase in kill probability with each encounter is fast. At these distances the hit probability is high. At 4 km the increase is slow due to the low hit probability. For targets that are detected at 3 km, the probability of kill increases rapidly to match that of targets detected at 4 km. The projectile consumption and engagement time are not the same for the 3 km and 4 km cases. Therefore, if a target is detected at 4 km, it is of interest to engage it at 3 km. There is no significant differences in kill range.

To understand the effect of the number of rounds per burst on the kill probability, Figure 17 shows the kill probability for each UAV when they are detected at 2 km. For a number of rounds per burst over 3, the three UAVs have similar kill probabilities on the whole range. If the number of projectiles per burst is decreased below 3, a difference in kill probabilities is observed. This figure shows that there is a marginal loss in killing range if the number of projectiles per burst is reduced from 5 to 3.

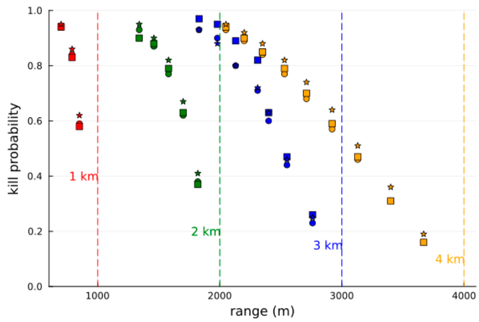

Figure 16. Kill probability for the 3 UAVs engaged with 5 projectiles per burst.
○ UAV 1, □ UAV 2, ⋆ UAV 3

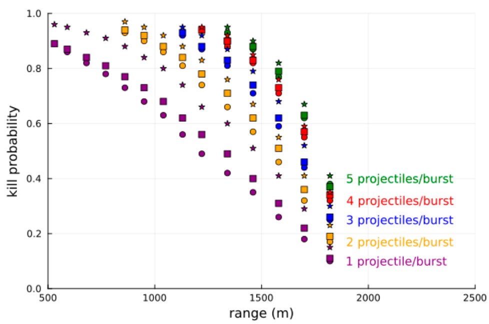

Figure 17. Kill probability for the 3 UAVs detected at 2 km.
○ UAV 1, □ UAV 2, ⋆ UAV 3

7. Conclusions

This study presents a versatile multi-scale discrete-event simulation framework for assessing the dynamic combat effectiveness of weapon systems, demonstrated through a case study involving a tank engaging aerial drone threats. The framework's modular design allows the seamless integration of high- and low-resolution models, facilitating flexibility in addressing various operational scenarios and levels of available information.

Simulation results highlight the effectiveness of different projectile types and firing strategies, providing actionable insights into weapon system performance against evolving threats. The inclusion of Monte Carlo-based uncertainty quantification ensures robust assessments of delivery accuracy and lethality, which are critical for modern combat engagements.

The findings underline the potential of discrete-event simulations to optimize engagement strategies and enhance the design of countermeasure systems. Future research will focus on refining computational efficiency for high-resolution modeling and exploring the framework's application to more complex scenarios, such as multitarget engagements and adaptive threat responses.

References

Ahner, D., & McCarthy, A. (2018). Response surface modeling of precision-guided fragmentation munitions. *The Journal of Defense Modeling and Simulation*, *17*(1), 83–97. https://doi.org/10.1177/1548512918811138

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bunn, F. L. (1993). *The tank accuracy model* (Technical repoprt). ARL Army Research Laboratory.

Chusilp, P., Charubhun, W., & Koanantachai, P. (2014). Monte Carlo simulations of weapon effectiveness using Pk Matrix and Carleton damage function. *International Journal of Applied Physics and Mathematics*, *4*(4), 280–285. https://doi.org/10.7763/IJAPM.2014.V4.299

Deitz, P. H., Reed, H. L., & Klopcic, T. J. (2009). *Fundamentals of ground combat system ballistic vulnerability/Lethality* (Vol. 230). American Institute of Aeronautics and Astronautics.

Dominicus, J. (2021). *New generation of counter UAS systems to defeat of Low Slow and Small* (*LSS*) *air threats* (Technical report). Royal Netherlands Aerospace Centre NLR.

Driels, M. (2004). *Weaponeering: Conventional weapon system effectiveness*. American Institute of Aeronautics and Astronautics.

Green, M. & Stewart, G. (2005). *M1 Abrams at War*. Zenith Press.

Kunertova, D. (2024). *Learning from the Ukrainian battlefield: Tomorrow's drone warfare, today's innovation challenge* (Technical repoprt). Center for Security Studies (CSS), ETH Zürich.

McCoy, R. (1999). *Modern exterior ballistics: The launch and flight dynamics of symmetric projectiles*. A Schiffer design book. Schiffer Publishing.

Molloy, O. (2024). *Drones in modern warfare: Lessons learnt from the War in Ukraine* (Technical report). Australian Army Research Centre. https://doi.org/10.61451/267513

Ndindabahizi, I., Vanhay, J., Lauwens, B., Lobo, V., & Gallant, J. (2022). A novel method for tank gun error budget analysis. In *Proceedings – 32nd International Symposium on Ballistics*, *BALLISTICS 2022* (pp. 950– 960). DEStech Publications. https://doi.org/10.12783/ballistics22/36125

Penney, H. R. (2023). *Scale*, *scope*, *speed* & *survivability: Winning the kill chain competition* (Policy Paper Series, vol. 40). Mitchell Institute for Aerospace Studies.

Przemieniecki, J. S. (2000). *Mathematical methods in defense analyses*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/4.862137

Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl – a performant and featurerich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.151

Seo, K. M., Choi, C., & Kim, T. G. (2012). Evaluating the effectiveness of shoot-look-shoot tactics using discrete event modeling and simulation. In J. H. Kim, K. Lee, S. Tanaka, & S. H. Park (Eds.), *Advanced methods*, *techniques*, *and applications in modeling and simulation* (pp. 352–360). Springer Japan. https://doi.org/10.1007/978-4-431-54216-2_39

Strohm, L. S. (2013). *An introduction to the sources of delivery error for direct-fire ballistic projectiles*. (Technical report ARL-TR-6494). Army Research Laboratory. https://doi.org/10.21236/ADA588846

Zhang, J. (2020). Modern Monte Carlo methods for efficient uncertainty quantification and propagation: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, *13*(5), Article e1539. https://doi.org/10.1002/wics.1539

Åkersson, B. (2022). Kranaattien ja maalialkioiden fysikaalisiin ominaisuuksiin perustuva vaikutusmalli. A Simplified Method for Computing the Lethality of Fragmenting Munitions Based on Physical Properties: (Suomenkielinen tiivistelmä aiheesta). *Tiede ja ase*, *2021*(79).