Space Domain Warfare Simulation: Orbital Mechanics, ASAT Engagement, and Kessler Modeling

Research Paper — space-war-sim v0.1.0

April 2026

Abstract

As space becomes a contested operational domain, the need for high-fidelity simulation tools that integrate orbital mechanics, weapons effects, debris propagation, and command-and-control logic has become critical. This paper presents

space-war-sim

, a modular simulation engine written in Go that models space domain warfare across multiple orbital regimes. The engine couples Keplerian orbital mechanics with J2/J3/J4 perturbations, atmospheric drag via a simplified NRLMSISE-00 model, and SGP4/SDP4 propagation from Two-Line Element (TLE) sets. It implements Monte Carlo–based engagement modeling for four anti-satellite (ASAT) weapon classes—direct-ascent kinetic, co-orbital, directed-energy laser, and electromagnetic pulse (EMP)—with debris generation governed by the NASA Standard Breakup Model and Kessler syndrome cascade analysis. Additional subsystems model space situational awareness (SSA) tracking and conjunction analysis, electronic warfare (jamming, spoofing, link disruption), rules of engagement (ROE) authorization, ground station pass windows and link budgets, and satellite constellation degradation. Two reference scenarios—a South China Sea multi-domain scenario and a DA-ASAT engagement with debris modeling—demonstrate the engine's capability to simulate complex space warfare vignettes from detection through after-action review. The architecture supports cross-platform deployment (Linux, macOS, Windows, Docker) and produces structured JSON after-action reports for further analysis.

CONTENTS

Keywords: orbital mechanics , ASAT weapons , Kessler syndrome , space domain awareness , simulation , Monte Carlo , debris modeling , conjunction analysis , electronic warfare

Table of Contents

1. Introduction

1.1 Motivation

The space domain has transitioned from a permissive environment to a contested and congested one. The proliferation of anti-satellite (ASAT) capabilities—ranging from direct-ascent kinetic kill vehicles to directed-energy systems and co-orbital inspection satellites—has fundamentally altered the strategic landscape. Concurrently, the growing population of orbital debris, amplified by deliberate fragmentation events and the self-sustaining Kessler syndrome, threatens the long-term viability of critical orbital regimes.

Existing simulation tools often specialize in narrow aspects of the space domain: orbital mechanics propagation, debris environment modeling, or weapons effects analysis in isolation. Military planners and researchers require integrated tools that can simultaneously model orbital state evolution, sensor detection, weapons engagement, debris generation, electronic warfare effects, and command-and-control (C2) decision logic within a single, coherent simulation framework.

space-war-sim addresses this gap by providing a high-fidelity, modular simulation engine that couples these domains into a unified temporal simulation loop, driven by declarative YAML scenario configurations and producing structured after-action review (AAR) data for analysis.

1.2 Prior Work

Orbital propagation has a rich heritage, from the analytic SGP4/SDP4 algorithms derived from NORAD Spacetrack Report No. 3[1] to high-precision numerical integrators used by the 18th Space Defense Squadron. Debris environment models such as NASA's ORDEM and ESA's MASTER provide statistical characterizations of the debris population, while the NASA Standard Breakup Model[7] underpins fragmentation event simulation. Military engagement modeling has traditionally relied on classified tools; unclassified Monte Carlo–based kill probability models have been discussed in the open literature for DA-ASAT systems[9].

To the authors' knowledge, no open unclassified tool integrates all of the following in a single framework: Keplerian and SGP4 propagation, J2/J3/J4 perturbations with atmospheric drag, multi-class ASAT engagement with Monte Carlo confidence, NASA breakup debris modeling with Kessler cascading, conjunction analysis, electronic warfare, ROE-gated C2, ground station link budgets, and constellation degradation—all driven by scenario files and producing JSON AAR output.

1.3 Contributions

This paper makes the following contributions:

- Presents the architecture of a modular space domain warfare simulation engine, demonstrating how orbital mechanics, weapons effects, debris modeling, EW, C2, and SSA subsystems are composed via a central event-driven engine.
- Documents the mathematical models underpinning each subsystem: Keplerian propagation with J2 perturbation, RK4 numerical integration with drag, SGP4/SDP4 from TLE inputs, Monte Carlo ASAT engagement, NASA Standard Breakup Model debris generation, Kessler syndrome cascading, Alfano/Chan conjunction probability, JNR-based jamming effectiveness, and ROE-authorized engagement logic.
- Demonstrates the engine via two reference scenarios spanning multi-domain space warfare and focused ASAT engagement with debris modeling.
- Discusses cross-platform deployment, performance characteristics, and federation potential.

2. Architecture

2.1 Package Organization

The engine is implemented in Go, organized into thirteen internal packages that map directly to functional domains:

Package	Domain	Key Capabilities
`core`	Engine	Simulation loop, event bus, entity management, config loading
`orbital`	Orbital Mechanics	Keplerian, J2, Hohmann, conjunction, drag
`propagator`	Numerical Propagation	RK4 integrator, J2–J4 + drag, geodetic conversion
`sgp4`	SGP4/SDP4	

Package	Domain	Key Capabilities
		TLE parsing, analytic propagation, deep-space corrections
`atm`	Atmosphere	Simplified NRLMSISE-00, drag acceleration, orbit decay
`tle`	TLE Data	Well-known satellite TLEs, batch parsing
`maneuver`	Orbital Maneuver	Collision avoidance, delta-v budgeting, Tsiolkovsky equation
`ssa`	Space Situational Awareness	Tracking, sensor models, conjunction analysis
`asat`	ASAT Weapons	DA-ASAT, co-orbital, laser, EMP; Monte Carlo engagement
`debris`	Debris	NASA breakup, fragmentation events, Kessler cascade
`ew`	Electronic Warfare	Jamming, spoofing, link disruption
`c2`	Command & Control	ROE engine, space tasking orders, hostile act definitions
`groundstation`	Ground Stations	Pass windows, tracking, link budgets
`satellite`	Satellites	LEO/MEO/GEO platforms, constellation models, degradation
`scenario`	Scenario Loading	YAML configuration, entity/MSEL definitions
`aar`	After-Action Review	Event logging, statistics, JSON export

2.2 Engine Core and Event Bus

The `core` package implements the central `Engine` struct, which manages a collection of `SpaceEntity` objects, each characterized by a unique identifier, name, category (satellite, ground station, ASAT platform), threat-side affiliation (friendly, hostile, civilian, neutral), orbital state, health fraction, mass, and radar cross-section. The engine maintains a

`SpaceEnvironment` struct capturing solar flux F10.7, Kp geomagnetic index, solar wind speed, and toggles for J2 perturbation and atmospheric drag.

An internal event bus supports publish-subscribe communication between subsystems. Events are typed (e.g., detection, engagement, debris creation, ROE change) and carry structured data payloads. Subsystems subscribe to relevant event types, enabling loose coupling between the orbital propagator, SSA tracker, ASAT engagement logic, debris manager, and AAR recorder.

2.3 Simulation Loop

The engine executes a fixed-timestep simulation loop parameterized by `delta_t` (step size in seconds) and `time_scale` (temporal acceleration factor). At each step, the engine:

1. Advances the simulation clock by $\texttt{delta\_t} \times \texttt{time\_scale}$ seconds of simulated time.
2. Propagates all active entities' orbital states.
3. Evaluates sensor visibility and updates SSA tracks.
4. Processes MSEL (Master Scenario Events List) events scheduled at the current time.
5. Invokes registered step hooks for subsystem-specific logic (ASAT inventory checks, conjunction screening, EW effects).
6. Emits events to the bus for downstream consumers.

The loop terminates on a shutdown signal or when the configured scenario duration elapses.

2.4 Configuration and Scenario Loading

Scenarios are defined declaratively in YAML, specifying a name, description, duration, timestep, time scale, environment parameters, entity definitions (with orbital elements), and a MSEL event timeline. The `scenario` package validates all parameters at load time, ensuring positive durations, valid timestep values, and non-empty entity lists. The CLI provides a `--validate` flag for standalone scenario validation and `--list-scenarios` for discovery.

3. Orbital Mechanics

Accurate orbital state propagation is the foundation of any space simulation. The engine implements multiple propagation strategies, each suited to different fidelity requirements and computational budgets.

3.1 Keplerian Propagation

The `orbital` package provides classical two-body orbital mechanics. Orbital velocity at radius r is computed as:

$$v = \sqrt{\frac{\mu}{r}} \quad \text{where } \mu = 398{,}600.4418 \text{ km}^3/\text{s}^2$$

The orbital period follows Kepler’s third law:

$$T = 2\pi\sqrt{\frac{a^3}{\mu}}$$

For the ISS at 408 km altitude ($a \approx 6{,}779$ km), the computed period is approximately 5,560 s (92.7 min), consistent with observed values. Conversion between classical orbital elements ($a, e, i, \Omega, \omega, \nu$) and position-velocity state vectors is provided via standard algorithms from Bate, Mueller, and White[2].

3.2 J2 Perturbation

Earth’s oblateness ($J2 = 1.082616 \times 10^{-3}$) causes secular drift in the right ascension of the ascending node (RAAN) and argument of perigee. The `orbital` package computes first-order secular rates[3]:

$$\frac{d\Omega}{dt} = -\frac{3}{2} \cdot n \cdot J_2 \cdot \left(\frac{R_E}{a} \right)^2 \cdot \frac{\cos(i)}{(1 - e^2)^2}$$

$$\frac{d\omega}{dt} = \frac{3}{4} \cdot n \cdot J_2 \cdot \left(\frac{R_E}{a}\right)^2 \cdot \frac{5\cos^2(i) - 1}{(1 - e^2)^2}$$

These secular rates are essential for realistic scenario evolution: sun-synchronous orbits ($i \approx 97.4°$) exploit the RAAN drift to maintain constant local solar time, while the argument of perigee drift at critical inclination ($i = 63.4°$) freezes perigee location—a property exploited by Molniya-type orbits and, notably, by the hostile ASAT platform in the South China Sea scenario.

3.3 Hohmann Transfers

The engine computes two-impulse Hohmann transfers for orbit raising and lowering maneuvers:

$$\Delta v = \sqrt{\frac{\mu}{r_1}} \left(\sqrt{\frac{2r_2}{r_1 + r_2}} - 1\right) + \sqrt{\frac{\mu}{r_2}} \left(1 - \sqrt{\frac{2r_1}{r_1 + r_2}}\right)$$

$$t_{\text{transfer}} = \pi \cdot \sqrt{\frac{a_t^3}{\mu}} \quad \text{where } a_t = \frac{r_1 + r_2}{2}$$

For a LEO-to-GEO transfer (400 km to 35,786 km), the computed Δv is approximately 3.9 km/s with a transfer time of ~19,000 s (~5.3 hours), consistent with standard references. The `maneuver` package additionally implements combined altitude-inclination changes, plane-change maneuvers, phasing maneuvers, and station-keeping, with fuel mass computed via the Tsiolkovsky rocket equation:

$$m_{\text{fuel}} = m_{\text{dry}} \cdot \left(e^{\Delta v / v_e} - 1\right)$$

3.4 Conjunction Analysis

The conjunction analysis module screens pairs of orbital elements over a specified time window to identify the time of closest approach (TCA) and minimum miss distance. Both objects are propagated at discrete time steps, and the minimum inter-object distance is identified. When combined with the SSA package's covariance-based methods (Alfano and Chan), collision probability P_c is estimated. This dual approach supports both fast screening (analytic conjunction) and high-fidelity assessment (covariance-based P_c).

The collision avoidance maneuver (CAM) module evaluates whether a conjunction warrants evasive action using the "4$\times$ rule" (miss distance should exceed four times the combined covariance radius) and computes the minimum-Δv maneuver to achieve a safe miss distance, choosing between along-track phasing and altitude-change strategies based on fuel cost.

3.5 RK4 Numerical Propagation

For scenarios requiring higher fidelity than analytic Keplerian propagation, the `propagator` package implements a fourth-order Runge-Kutta (RK4) integrator. The equations of motion include:

- **Two-body gravity:** $\mathbf{a} = -\frac{\mu \mathbf{r}}{|\mathbf{r}|^3}$
- **J2 perturbation:** Equatorial bulge correction terms in the radial, in-track, and cross-track directions.
- **J3 and J4 zonal harmonics:** Higher-order corrections for improved fidelity at medium altitudes.
- **Atmospheric drag:** $a_{\text{drag}} = -\frac{1}{2} \rho v^2 / B^*$, where ρ is computed by the atmospheric model and B^* is the ballistic coefficient.

The RK4 integrator supports configurable step sizes (default 10 s) and is used in the integration test suite to propagate the ISS for multiple orbits, verifying that altitude remains within expected bounds (300–600 km for a 408 km initial orbit) after propagation with J2 and drag enabled.

3.6 SGP4/SDP4 Propagation

The `sgp4` package implements the NORAD SGP4/SDP4 analytic propagation algorithm based on Spacetrack Report No. 3[1] with the Vallado et al. revisions[4]. Key features include:

- TLE parsing with satellite number, epoch, B* drag coefficient, and all six classical elements.
- Automatic deep-space detection (orbital period ≥ 225 min) with SDP4 deep-space corrections for resonance and long-period effects.
- Kepler's equation solved via Newton-Raphson iteration (10 iterations, tolerance 10^{-10}).
- ECI position and velocity output with geodetic (lat, lon, alt) conversion.
- Batch propagation over arbitrary time spans.

The `tle` package provides representative TLEs for well-known satellites (ISS, Hubble, GPS IIF-1, GEO communications, and a hypothetical hostile ISR satellite in sun-synchronous orbit), enabling immediate scenario instantiation from realistic orbital data.

3.7 Atmospheric Drag

The `atm` package implements a simplified NRLMSISE-00 atmospheric density model[8] using the exponential atmosphere approximation with altitude-dependent scale heights and solar/geomagnetic corrections. The atmospheric profile is divided into eight layers:

Layer	Altitude (km)	Scale Height (km)	Density Range (kg/m³)
Troposphere/Stratosphere	0–25	6.3–8.5	$1.225 – 3.9\times10^{-2}$
Upper Stratosphere	25–50	6.3	$3.9\times10^{-2} – 1.0\times10^{-3}$
Mesosphere	50–80	6.7	$1.0\times10^{-3} – 1.8\times10^{-5}$
Lower Thermosphere	80–120	12+	$1.8\times10^{-5} – 2.2\times10^{-7}$
Mid Thermosphere	120–200	20+	$2.2\times10^{-7} – 2.5\times10^{-10}$
Upper Thermosphere	200–400	30+	$2.5\times10^{-10} – 7.0\times10^{-12}$
Exosphere Transition	400–600	40+	$7.0\times10^{-12} – 5.0\times10^{-13}$
Low Exosphere	600–1000	50+	$5.0\times10^{-13} – 1.5\times10^{-14}$

Solar flux (F10.7) corrections are weighted by altitude, with negligible effect below the mesopause (80 km) and full effect above 400 km. Geomagnetic activity (Kp index) further modulates thermospheric density through an expansion factor. The integration test suite verifies that high solar flux (F10.7 = 250, Kp = 8) produces higher density and drag than low solar flux (F10.7 = 70, Kp = 1) at 400 km altitude, as expected from physical principles.

4. Space Situational Awareness and Tracking

4.1 Sensor Models

The SSA subsystem models three sensor classes:

- **Ground radar:** Mechanical and phased-array radar with configurable range maximum, sensitivity (minimum detectable RCS in dBsm), and coverage azimuth. Range resolution and track update rate are parameterized.

- **Optical telescopes:** Passive electro-optical sensors with limiting magnitude, field of view, and sun-avoidance constraints. Detection range scales with target RCS and illumination geometry.
- **Space-based sensors:** SBIRS-type infrared sensors in GEO for launch detection and early warning, with staring and scanning modes.

Each sensor is geolocated with latitude, longitude, and elevation, and supports enable/disable toggling via MSEL events.

4.2 Track Management

The SSA manager maintains a track database with sequential track numbering (TN). Tracks progress through classification states: uncorrelated target (UCT), correlated, confirmed, and hostile. Classification confidence is a continuous [0, 1] value that increases with sustained sensor observation and decreases during coverage gaps. Track data includes the object's name, side affiliation, geodetic position, altitude, velocity, and RCS.

4.3 Conjunction Assessment

The conjunction analysis module implements both the Alfano and Chan methods for computing collision probability P_c from covariance ellipsoids and miss distance[5,6]. Given two objects' position covariance matrices and the miss vector at TCA, the probability of collision is computed by integrating the combined probability density over the collision cross-section. This supports go/no-go decisions for collision avoidance maneuvers.

5. ASAT Weapons and Engagement

5.1 Weapon Taxonomy

The `asat` package models five ASAT weapon classes:

Type	Altitude Range	Base P_k	CEP (km)	Inventory	Mechanism
Direct-Ascent (DA-ASAT)	200–1,000 km	0.90	0.05	12	Kinetic kill vehicle, exo-atmospheric intercept

Type	Altitude Range	Base P_k	CEP (km)	Inventory	Mechanism
Co-Orbital	400–36,000 km	0.75	0.10	4	Rendezvous and proximate detonation
Ground Laser	200–1,500 km	0.65	—	2	Directed energy, dazzle/degrade sensor
EMP	200–2,000 km	0.40	—	3	High-altitude nuclear EMP, electronics kill
Cyber	Any	0.30	—	—	Network intrusion, command uplink compromise

Each weapon type encapsulates its operational envelope, engagement geometry constraints, and expected effects. DA-ASAT weapons are modeled with specific altitude min/max bounds reflecting the kinematic reach of exo-atmospheric kill vehicles launched from surface platforms.

5.2 Kill Probability Model

Kill probability P_k is computed as a function of altitude, aspect angle, and weapon characteristics:

$$P_k = P_{k,\text{base}} \cdot f_{\text{alt}}(h) \cdot f_{\text{aspect}}(\theta)$$

The altitude factor f_{alt} models reduced effectiveness at the extremes of the weapon's operational envelope, reaching unity at the optimal altitude and falling off toward the min/max bounds. The aspect angle factor f_{aspect} favors head-on geometries ($\theta = 0^\circ$) over tail-chase scenarios ($\theta = 180^\circ$), reflecting closing velocity effects on intercept kinematics.

5.3 Monte Carlo Engagement

Engagement outcomes are determined via Monte Carlo simulation with configurable trial counts and deterministic seeding for reproducibility. For each trial:

1. A random aspect angle offset is drawn from a uniform distribution.

2. Kill probability is computed from the model above.
3. A Bernoulli trial determines kill/no-kill.
4. Debris generation is computed from the target mass and altitude.

The aggregate result provides:

- `Hit` : boolean majority outcome across all trials
- `KillProb` : fraction of trials resulting in kill
- `DamageFraction` : mean damage inflicted
- `DebrisGenerated` : mean fragment count
- `Confidence` : 95% confidence interval half-width on P_k

Integration tests confirm that DA-ASAT achieves high kill probability at 400–500 km (within its optimal envelope), zero kill probability above 1,000 km (out of range), and that results are exactly reproducible given the same random seed.

5.4 Aspect Angle Effects

The model captures the kinematic advantage of head-on engagements. Integration testing verifies that head-on ($\theta = 0°$) kill probability exceeds tail-chase ($\theta = 180°$) probability for the same weapon-target geometry, consistent with higher closing velocities reducing intercept time-of-flight and seeker acquisition requirements.

5.5 Weapon Inventory

The `ASATInventory` manages per-entity weapon allocations, supporting queries of the form “can entity X engage a target at altitude Y?” The inventory selects the highest-P_k weapon within range, enabling automated engagement decisions subject to ROE constraints.

6. Debris and Kessler Syndrome

6.1 NASA Standard Breakup Model

The debris module implements the NASA Standard Breakup Model[7] for fragmentation event simulation. Given a target mass M (kg) at altitude h (km) and impact velocity, the model computes the fragment population across three size thresholds:

$$N(\gt L_{\text{cm}}) = 0.1 \cdot M^{0.83} \cdot L_{\text{cm}}^{-1.6} \quad \text{(for catastrophic fragmentation)}$$

- **Trackable fragments (>10 cm):** Catalogued objects trackable by the Space Surveillance Network.
- **>1 cm fragments:** Lethal but untrackable objects capable of catastrophic satellite damage.
- **>1 mm fragments:** Sub-lethal but degrading objects contributing to long-term environment contamination.

A fragmentation event is classified as catastrophic when the impactor kinetic energy exceeds 40 J/g of target mass; otherwise it is a non-catastrophic (low-intensity) event producing fewer fragments.

6.2 Fragmentation Events

Each fragmentation event records altitude, inclination, target mass, impactor mass and velocity, catastrophic flag, event type (ASAT, explosion, collision), timestamp, and random seed for reproducibility. The debris field manager accumulates events and maintains the current debris population, tracking both the number of objects and their total mass.

6.3 Kessler Cascade

The Kessler syndrome model[10] evaluates whether the debris population at a given altitude has reached a self-sustaining cascade. The cascade function computes the number of new debris objects generated over a specified time horizon by cascading collisions among existing fragments:

- **Collision rate:** Computed from debris density, orbital velocity, and cross-sectional area at the specified altitude shell.
- **Cascade trigger:** When the collision rate exceeds a threshold, each collision produces new fragments via the breakup model, which in turn increase the collision rate.
- **Damping:** Atmospheric drag at lower altitudes removes fragments over time, providing a natural damping mechanism that counteracts cascade growth.

The integration test suite demonstrates cascade analysis after a simulated explosion at 500 km, computing collision rates per year and new debris generation over a 10-year cascade horizon.

6.4 Orbital Lifetime

Orbital lifetime estimates follow a simplified empirical model:

Altitude (km)	Approximate Lifetime
<200	Days
200–300	Months
300–400	~2 years
400–500	~10 years
500–600	~25 years
600–800	~100 years
800–1,000	~500 years
>1,000	>1,000 years (effectively permanent)

This model underscores the strategic significance of altitude: a DA-ASAT engagement at 800 km produces debris with lifetimes measured in centuries, whereas the same event at 300 km would naturally clear within years.

7. Electronic Warfare in Space

7.1 Jamming-to-Noise Ratio

The EW module computes the jamming-to-noise ratio (JNR) for both uplink and downlink jamming scenarios. For an uplink jammer at range R_j from the satellite and a legitimate ground station at range R_s:

$$\text{JNR} = \frac{P_j \cdot G_j \cdot G_r}{P_s \cdot G_s \cdot G_r} \cdot \left(\frac{R_s}{R_j}\right)^2 \cdot \frac{BW_s}{BW_j}$$

where P denotes power, G denotes antenna gain, BW denotes bandwidth, and subscripts j and s refer to jammer and signal respectively. The model accounts for off-axis antenna gain reduction for jammer positions outside the main lobe.

7.2 Spoofing

GPS spoofing effectiveness is assessed based on the spoofing signal power relative to the authentic signal at the target receiver. The model computes the signal-to-spoofing ratio and determines whether the target can be captured into a false lock, considering receiver autonomous integrity monitoring (RAIM) capabilities.

7.3 Link Disruption

Link disruption is modeled as a binary effective/ineffective outcome based on JNR exceeding a threshold (typically 0 dB for partial disruption, 10 dB for complete denial). The module reports the disruption status and the effective data rate reduction under jamming conditions.

8. Command and Control: ROE and Space Tasking Orders

The C2 module implements a rules of engagement (ROE) engine governing weapons authorization. The ROE state machine defines three states:

- **WEAPONS HOLD:** No engagement authorized. Defensive and observation-only posture.
- **WEAPONS TIGHT:** Engagement authorized only against confirmed hostile targets that have committed a defined hostile act. The engine maintains a list of hostile act definitions (e.g., close approach within threat distance, EW attack, ASAT launch detection).
- **WEAPONS FREE:** Engagement authorized against all confirmed hostile targets without requiring a specific hostile act.

ROE transitions are driven by MSEL events (e.g., “ROE changed to WEAPONS TIGHT” at T+1800s in the South China Sea scenario). The `SpaceTaskingOrder` (STO) structure encodes specific engagement instructions: target ID, weapon assignment, engagement window, and authorization chain. The C2 engine evaluates each STO against the current ROE state before authorizing ASAT engagement, preventing unauthorized weapons release even if the ASAT inventory and tracking conditions are met.

9. Ground Stations and Link Budgets

9.1 Pass Window Computation

Ground station pass windows are computed from the satellite's orbital elements and the station's geodetic coordinates. A pass occurs when the satellite's elevation angle above the station's horizon exceeds a minimum threshold (typically 5–10°). The module computes rise time, culmination (maximum elevation), and set time for each pass, enabling scheduling of tracking, command uplink, and data downlink operations.

9.2 Link Budget Analysis

The link budget module computes the carrier-to-noise density ratio (C/N_0) for satellite communication links:

$$C/N_0 = \text{EIRP} - L_{\text{path}} + G/T - k \quad \text{(dB-Hz)}$$

where EIRP is the effective isotropic radiated power, L_{path} is the free-space path loss (accounting for range, atmospheric attenuation, and rain margin), G/T is the receive system figure of merit, and k is Boltzmann's constant (-228.6 dBW/K/Hz). The module reports margin against the required C/N_0 threshold for the modulation and coding scheme, and flags links that fall below minimum quality thresholds.

A default set of well-known ground stations (e.g., USSF tracking stations, Diego Garcia, Thule, Clear) is provided with type classifications (radar, optical, S-band, etc.).

10. Satellite and Constellation Modeling

10.1 Orbital Regime Profiles

The `satellite` package provides pre-configured platform models for three orbital regimes:

Constellation	Regime	Altitude (km)	Inclination	Satellites	Purpose
Starlink-class	LEO	550	53°	~1,600	Communications
GPS-class	MEO	20,200	55°	~31	Navigation/PNT

Constellation	Regime	Altitude (km)	Inclination	Satellites	Purpose
Iridium-class	LEO	780	86.4°	~66	Communications

Each platform model specifies mass, RCS, and default health. Custom constellations can be defined with arbitrary numbers of satellites, orbital parameters, and operational characteristics.

10.2 Constellation Degradation

The constellation degradation model evaluates the impact of ASAT attacks or debris-induced collisions on constellation performance. Key metrics include:

- **Active satellite count:** Remaining operational satellites vs. initial constellation size.
- **Coverage fraction:** Percentage of the Earth's surface with minimum n-satellite visibility.
- **Capacity factor:** Ratio of current to design throughput (for communications constellations).
- **Constellation health:** Aggregate health score (0–1) reported in AAR output.

When debris events occur at altitudes intersecting a constellation's orbital shell, the collision probability for each surviving satellite is updated, and expected losses are computed. This enables assessment of second-order effects: a DA-ASAT engagement against a single target at 500 km may produce a debris field that degrades a Starlink-class constellation operating at 550 km over subsequent years.

11. Scenarios

11.1 South China Sea Space Domain

The primary reference scenario models a multi-domain space warfare vignette in the South China Sea region with a 7,200 s (2-hour) duration, 1.0 s timestep, and 10× time acceleration. The scenario includes:

Entities (7):

- Friendly: GPS-IIF-01, GPS-IIF-02 (MEO, 20,200 km, 55°), GEO-Comms-01 (GEO, 35,786 km), ISR-LEO-01 (LEO, 400 km, 51.6°)
- Hostile: Hostile-ISR (sun-synchronous LEO, 500 km, 97.4°), Hostile-Comms (GEO), Hostile-ASAT-Platform (LEO, 500 km, 63.4° critical inclination)

- Civilian: ISS (LEO, 408 km, 51.6°, 420,000 kg, RCS 500 m²)

Environment: F10.7 = 150, Kp = 3, J2 and drag enabled.

MSEL Timeline:

Time (s)	Event
0	Scenario start
600	Sensor activate — GPS constellation active tracking
1,200	Detection — Hostile ISR satellite detected
1,800	ROE change — WEAPONS TIGHT
3,600	Close approach — Hostile ISR proximate to friendly ISR-LEO
5,400	ROE change — WEAPONS FREE
7,200	Scenario end

This scenario exercises the full detection–tracking–classification–ROE–engagement chain, including the escalation from Weapons Hold through Weapons Tight (requiring hostile act confirmation) to Weapons Free.

11.2 DA-ASAT Engagement with Debris

The second reference scenario focuses on a single DA-ASAT engagement with detailed debris modeling over 3,600 s (1 hour) with 0.5 s timestep and 5× time acceleration:

Entities (4):

- Friendly: USS Hopper Space Track (ground station), SBIRS-GEO-1 (early warning), DA-ASAT Battery (12 SM-3-class interceptors)
- Hostile: Hostile-ISR-7 (sun-synchronous LEO, 500 km, 97.4°)

Environment: F10.7 = 180, Kp = 5 (elevated solar activity).

MSEL Timeline:

Time (s)	Event
0	Scenario start

Time (s)	Event
300	Detection — Hostile ISR-7 detected by SBIRS
900	Track promoted to confirmed
1,200	ROE: WEAPONS FREE for ASAT engagement
1,500	DA-ASAT launched against Hostile-ISR-7
2,100	Weapon impact — kinetic kill, target destroyed
2,400	Debris cloud from ASAT engagement
3,600	Scenario end

This scenario demonstrates the kill chain from SBIRS launch detection, through track promotion and ROE authorization, to kinetic engagement and post-engagement debris field characterization. The integration test suite validates the complete chain: SGP4 initialization from TLE → RK4 propagation with J2 and drag → ASAT Monte Carlo engagement → NASA breakup debris generation, confirming that the debris field contains trackable fragments consistent with a 500 kg target at 500 km.

12. After-Action Review and Federation

12.1 AAR Recording and Export

The `aar` package provides comprehensive after-action review capabilities. Events are logged with timestamp, type (detection, engagement, jamming, debris, ROE change), source entity ID, target entity ID, and a human-readable message. The recorder supports:

- **Structured event log:** Time-ordered sequence of all simulation events.
- **Statistics computation:** Aggregate metrics including total events, detection/engagement/jamming/debris event counts, track count, friendly and hostile losses, and constellation health fraction.
- **JSON export:** Complete AAR data serialized to JSON for programmatic analysis, visualization, or ingestion by downstream tools.
- **Console summary:** Human-readable text summary printed at simulation end.

- **JSON stdout mode:** Machine-readable result output (`--json` flag) for CI/CD integration and automated testing pipelines.

12.2 Federation Considerations

The engine's event-driven architecture supports future federation via the publish-subscribe event bus. Subsystems communicate through typed events, enabling:

- **Heterogeneous simulation:** External high-fidelity propagators or weapons effects models could subscribe to the event bus and provide updates without modifying core engine logic.
- **Distributed execution:** Entity management and propagation could be distributed across processes, synchronized via the event bus.
- **Live-virtual-constructive (LVC) integration:** The AAR event stream provides a natural interface for injecting live data or replaying historical events.

13. Performance and Deployment

The engine is compiled as a single static binary (~2.3 MB) with no runtime dependencies beyond the operating system. Cross-compilation targets five platform combinations:

- Linux amd64 (tar.gz and .deb package with systemd unit)
- Linux arm64
- macOS Intel (amd64)
- macOS Apple Silicon (arm64)
- Windows amd64

A Docker image based on Alpine 3.19 is provided for containerized deployment. Build metadata (version, commit hash, build date) is injected via `-ldflags` at compile time. The CLI supports shell completions for bash, zsh, and PowerShell.

Computational performance is dominated by orbit propagation. The RK4 integrator with J2 and drag at 10 s timestep processes ~556 steps per ISS orbit (~92 min simulated time). Integration tests confirm that 10-orbit propagation (5,560 steps) completes in milliseconds, and the full engagement chain (SGP4 → RK4 → Monte Carlo → breakup model) executes within interactive timeframes.

The Go runtime's concurrent garbage collection and native parallelism support position the engine well for scaling to larger scenario orders (hundreds to thousands of entities), though the current implementation processes entities sequentially within the simulation loop. Future work may parallelize propagation across CPU cores.

14. Conclusion

This paper has presented space-war-sim, a modular simulation engine for space domain warfare that integrates orbital mechanics, ASAT engagement modeling, debris propagation, electronic warfare, command-and-control, and space situational awareness within a single, coherent framework. The engine's architecture—organized as thirteen loosely-coupled packages communicating through a central event bus—enables independent development and testing of each subsystem while ensuring consistent temporal evolution across the simulation.

The mathematical models underlying each subsystem are grounded in established references: Keplerian mechanics and J2 perturbation from Vallado[3] and Bate, Mueller, and White[2]; SGP4/SDP4 from Spacetrack Report No. 3[1] and the Vallado et al. revision[4]; conjunction probability from Alfano[5] and Chan[6]; the NASA Standard Breakup Model[7]; and NRLMSISE-00[8] for atmospheric drag. The Monte Carlo engagement framework provides statistically meaningful kill probability estimates with configurable confidence, and the debris model captures both immediate fragmentation effects and long-term Kessler cascade dynamics.

Two reference scenarios demonstrate the engine's capability to model realistic space warfare vignettes, from multi-domain escalation with ROE constraints to focused ASAT engagement with debris characterization. The YAML-driven scenario format and JSON AAR output enable reproducible experimentation and integration with analysis pipelines.

Future work may address: parallel propagation for large constellation scenarios, higher-fidelity atmospheric and radiation models, additional weapon types (e.g., nuclear detonation effects), human-in-the-loop C2 interfaces, and High Level Architecture (HLA) federation for distributed simulation.

References

[1] Hoots, F. R. and Roehrich, R. L., "Models for Propagation of NORAD Element Sets," Spacetrack Report No. 3, 1980.

[2] Bate, R. R., Mueller, D. D., and White, J. E., Fundamentals of Astrodynamics, Dover Publications, 1971.

[3] Vallado, D. A., Fundamentals of Astrodynamics and Applications, 4th ed., Microcosm Press, 2013.

[4] Vallado, D. A., Crawford, P., Hujsak, R., and Kelso, T. S., "Revisiting Spacetrack Report #3," AIAA 2006-6753, 2006.

[5] Alfano, S., "A Numerical Implementation of Spherical Object Collision Probability," Journal of the Astronautical Sciences, Vol. 53, No. 1, 2005, pp. 103–109.

[6] Chan, F. K., Spacecraft Collision Probability, The Aerospace Press, 2008.

[7] Krisko, P. H., "The NASA Orbital Debris Engineering Model: A New Implementation," Advances in Space Research, Vol. 37, 2006, pp. 1198–1204.

[8] Picone, J. M., Hedin, A. E., Drob, D. P., and Aikin, A. C., "NRLMSISE-00 Empirical Model of the Atmosphere: Statistical Comparisons and Scientific Issues," Journal of Geophysical Research—Space Physics, Vol. 107, No. A12, 2002.

[9] Gregory, D. L., "Anti-Satellite Weapons: Counter-Orbit Capabilities and Strategic Implications," Strategic Studies Quarterly, Vol. 10, No. 3, 2016, pp. 42–60.

[10] Kessler, D. J. and Cour-Palais, B. G., "Collision Frequency of Artificial Satellites: The Creation of Debris," Journal of Geophysical Research, Vol. 83, No. A6, 1978, pp. 2637–2646.
