

Maritime Warfare Simulation: Acoustic Propagation, ASW Engagement, and DIS Federation

Crab Meat Research

Simulation Systems Technical Series

April 2026

Abstract. This paper presents the architecture and implementation of a high-fidelity maritime warfare simulation engine designed for anti-submarine warfare (ASW) scenario development, analysis, and training. The simulation integrates physically-grounded models of underwater acoustic propagation — including Mackenzie sound speed profiles, convergence zone and bottom-bounce paths, and reverberation-limited detection — with radar horizon calculations, a multi-path sonar equation, aspect-dependent target strength and radar cross-section models, and a six-degree-of-freedom vessel dynamics framework based on the Maneuvering Modeling Group (MMG) standard. A command-level AI module evaluates contacts through threat scoring and CPA analysis, operating under a configurable Rules of Engagement engine with weapons-free, weapons-tight, and weapons-hold postures. Weapon systems include anti-ship missile flyout with CIWS and chaff countermeasures, torpedo runout-search-homing with proportional navigation, and compartment-level ship damage with progressive flooding. The simulation further incorporates Pierson-Moskowitz and JONSWAP sea spectra, an IEEE 1278.1 DIS federation bridge for distributed exercise participation, and a real-time Leaflet.js tactical display with Server-Sent Events. Cross-platform binaries (~4.6 MB) are produced for Linux, macOS, Windows, and Docker, with YAML-driven scenario configuration and JSON after-action review export.

Table of Contents

- [1. Introduction](#)
- [2. Architecture](#)
- [3. Surface & Submarine Platforms](#)

4. [Acoustic Modeling](#)
5. [Radar Modeling](#)
6. [Sensor Management & Track Keeping](#)
7. [AI & Command and Control](#)
8. [Weapons & Damage](#)
9. [Electronic Warfare](#)
10. [Hydrodynamics & Seakeeping](#)
11. [Scenario Engine & After-Action Review](#)
12. [DIS Federation Bridge](#)
13. [Performance & Cross-Platform Deployment](#)
14. [Conclusion](#)
15. [References](#)

1. Introduction

Naval simulation has evolved from simple war-gaming tables to sophisticated digital environments that must capture the interplay of ocean physics, sensor performance, weapon flyout, and human decision-making. Anti-submarine warfare (ASW) presents particular challenges: the underwater acoustic environment is governed by sound speed profiles that vary with temperature, salinity, and pressure; detection ranges change by orders of magnitude between direct-path, surface-duct, convergence-zone, and bottom-bounce propagation modes; and the kill chain — from first detection through classification, engagement authorization, and weapon delivery — involves multiple interacting subsystems under rules of engagement (ROE) that constrain when force may be applied.

We present a simulation engine, `maritime-sim`, designed to model these interactions at a level of fidelity suitable for scenario analysis, concept exploration, and operator training. The engine is written in Go and compiles to a single ~4.6 MB binary with no external runtime dependencies, enabling deployment across Linux, macOS, Windows, and containerized environments. It is driven by YAML scenario configurations and

produces structured JSON after-action review (AAR) output, making it suitable for both interactive use (via a real-time Leaflet.js tactical display) and batch-mode analysis.

The paper is organized as follows: Section 2 describes the overall architecture; Sections 3–10 detail the physics and engineering models; Section 11 covers the scenario and AAR engines; Section 12 describes the DIS federation bridge; Section 13 discusses performance; and Section 14 concludes.

2. Architecture

The engine follows a discrete time-step architecture centered on a thread-safe `Engine` struct that maintains an entity map, an event bus, and a collection of platform models. Each simulation step proceeds as follows:

1. Advance all platform models by $\Delta t \times \text{timeScale}$ seconds.
2. Publish entity state to any registered `FederationBridge` instances.
3. Execute step hooks (sensor checks, AI decisions, weapon updates, AAR snapshots).

The event bus provides publish-subscribe semantics with typed events (entity added, removed, damaged, destroyed, detection, weapon launch/impact). Channels buffer up to 100 events; overflow is silently dropped to prevent back-pressure from blocking the main loop.

Configuration is loaded from YAML files that specify the scenario name, duration, time step, environment conditions, entity initial conditions, and a Master Scenario Events List (MSEL). The engine supports configurable time acceleration, pause/resume, and dynamic time-scale changes during execution.

2.1 Package Organization

Package	Function
core	Engine loop, event bus, config, kinematics

platform/surface	DDG, FFG, CVN surface combatant models
platform/submarine	SSN, SSK submarine models with depth/battery
sensor/acoustic	Mackenzie SSP, sonar equation, TL, reverberation
sensor/radar	Radar equation, horizon, Swerling models
sensor/sensormgr	Sensor scheduling, search patterns, sonobuoy fields
sensor/trackmgr	Alpha-beta filtering, TMA, classification
sensor/signature	Aspect-dependent RCS and TS, radiated noise
ai	Commander AI, threat assessment, decision tree
comms/c2	ROE engine, formations, sensor allocation
scenario/mسل	Master Scenario Events List timeline
scenario/aar	AAR recorder, replay, scoring, JSON export
weapon/ashm	AShM flyout (Harpoon, YJ-83, Exocet)
weapon/torpedo	Torpedo runout/search/homing (Mk 48 ADCAP)
weapon/damage	Compartment flooding, fire, stability
weapon/engagement	Engagement manager, ROE check, weapon launch
ew/ecm	Chaff, noise jamming, torpedo decoys
ew/esm	ESM intercept, emitter library, bearing-only TMA
hydro	Pierson-Moskowitz, JONSWAP, 6-DOF, MMG maneuvering
env/ocean	Bathymetry, sound speed profiles, tides, currents
fed/dis	IEEE 1278.1 DIS federation bridge
webui	Leaflet.js tactical display with SSE

3. Surface & Submarine Platforms

3.1 Surface Combatants

The surface package provides parametric models for destroyer (DDG), frigate (FFG), and carrier (CVN) hulls. Each `SurfaceCombatant` carries displacement, dimensions, speeds, and sensor/weapon suite assignments. Presets match the Arleigh Burke-class DDG (9,200 tonnes, 155 m length, 30 kt max speed), the Oliver Hazard Perry-class FFG (4,200 tonnes, 29 kt), and the Nimitz-class CVN (100,000 tonnes, 30+ kt). Position propagation uses a flat-Earth WGS84 approximation sufficient for the tactical scales involved:

$$\Delta \text{lat} = \frac{v \cos \psi \cdot \Delta t}{111320}, \quad \Delta \text{lon} = \frac{v \sin \psi \cdot \Delta t}{111320 \cos(\text{lat})}$$

where v is speed over ground and ψ is heading. A separate current-propagation function adds ocean current vectors to the vessel's water-track velocity to obtain course and speed over ground.

3.2 Submarine Models

Submarines extend the platform model with depth control, cavitation speed, battery management, and sprint-and-drift tactics. The `Submarine` struct tracks current depth, ordered depth, and maximum depth rate. For diesel-electric boats (the Kilo SSK preset), battery capacity depletes proportional to speed cubed:

$$P(v) = P_{\max} \cdot \left(\frac{v}{v_{\max}}\right)^3$$

A snorkel function recharges batteries at periscope depth (≤ 20 m) at 500 kW. Cavitation speed increases with depth according to the square-root pressure model:

$$v_{\text{cav}}(d) = v_{\text{cav,surface}} \cdot \sqrt{1 + \frac{d}{10.3}}$$

Radiated noise follows a regime model: below cavitation speed, noise increases as $20 \log_{10}(v/v_{\text{quiet}})$; above cavitation, an additional 20 dB penalty is applied. The `SprintAndDriftState` controller alternates between sprint speed (fast

transit, cavitating) and drift speed (quiet listening), managing the time balance autonomously.

4. Acoustic Modeling

The acoustic package is the most physically detailed component of the simulation, implementing the sonar equation, multi-path propagation, Mackenzie sound speed profiles, reverberation, and ambient noise models.

4.1 Sound Speed: Mackenzie Equation

Sound speed is computed using the Mackenzie (1981) nine-term equation, which relates sound speed c (m/s) to temperature T (°C), salinity S (PSU), and depth D (m):

$$c = 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T^3 + 1.340(S - 35) + 1.630 \times 10^{-2}D + 1.675 \times 10^{-7}D^2 - 1.025 \times 10^{-2}T(S - 35) - 7.139 \times 10^{-13}TD^3$$

The ocean package provides `SoundSpeedProfile()`, which builds a depth-indexed profile from surface conditions through the mixed layer, thermocline, SOFAR channel axis, and deep isothermal layer. Mixed layer depth is parameterized as a function of wind speed ($10 + 2v_w + 3\sqrt{\text{SS}}$), and the SOFAR axis depth varies with latitude from ~1,200 m in the tropics to ~200 m in polar regions.

4.2 Transmission Loss Models

Three spreading models are implemented:

- **Spherical spreading:** $\text{TL} = 20 \log_{10}(r)$, valid in the near field and deep water.
- **Cylindrical spreading:** $\text{TL} = 10 \log_{10}(r)$, valid in ducted propagation (surface duct, SOFAR channel).
- **Hybrid spreading:** transitions from spherical to cylindrical at a range equal to $\pi \cdot \text{depth}$, capturing the transition from near-field spherical spreading to far-field trapped propagation.

Frequency-dependent absorption uses a simplified Francois-Garrison formulation with separate boric acid, MgSO_4 , and pure water terms:

$$\alpha(f, T, S, D, \text{pH}) = A_1(f, T, S) + A_2(f, T, S) + A_3(f)$$

where A_1 captures boric acid relaxation, A_2 captures magnesium sulfate relaxation, and A_3 captures pure water viscosity.

4.3 Propagation Paths

The `PropagationModes()` function evaluates five propagation paths for a given frequency, range, mixed layer depth, bottom depth, and sea state:

1. **Direct path** — always available; spherical spreading + absorption; limited by mixed layer depth.
2. **Surface duct** — available when mixed layer depth > 20 m; cylindrical spreading with a ~5 dB duct gain; range scales as 100× mixed layer depth.
3. **Convergence zone** — deep water only (bottom > 2,000 m); CZ spacing ~55 km; TL reduced by 10 dB in the annulus; width ~7 km; up to 4 CZs evaluated.
4. **Bottom bounce** — shallow water or deep water beyond first CZ; spherical spreading + 3 dB gain from bottom reflection.
5. **Deep sound channel (SOFAR)** — bottom > 4,000 m; cylindrical spreading at depth; ranges out to 100 km.

The `BestPropagationPath()` function selects the path with the lowest transmission loss at the given range, and `DetectionRange()` scans ranges from 1 km to 100 km to find the maximum range where the signal excess remains non-negative.

4.4 Sonar Equations

Active and passive sonar equations are implemented:

$$\begin{aligned} \textbf{Active:} \quad \text{SE} &= \text{SL} - \text{TL}_{\text{out}} + \text{TS} - \text{TL}_{\text{back}} - (\text{NL} - \text{DI}) - \text{DT} \\ \textbf{Passive:} \quad \text{SE} &= \text{SL} - \text{TL} - (\text{NL} - \text{DI}) - \text{DT} \end{aligned}$$

Ambient noise follows the Wenz curves, with wind/sea state, shipping, and biological components summed energetically. The main loop compares both noise-limited and reverberation-limited signal excess, selecting the worse (more conservative) value for detection decisions.

4.5 Reverberation

Surface, bottom, and volume reverberation are computed separately and summed energetically:

- **Surface reverberation** uses the Chapman-Harris scattering model with wind-speed-dependent surface scattering strength.
- **Bottom reverberation** uses Lambert's law with bottom-type-dependent scattering strength (mud: -35 dB, sand: -20 to -27 dB, rock: -15 dB).
- **Volume reverberation** models deep scattering layer contributions with frequency and depth dependence.

A binary search finds the `ReverberationLimitedRange` where reverberation level equals detection threshold, providing a range cutoff independent of ambient noise conditions.

4.6 Target Strength and Radiated Noise

Target strength varies with aspect angle. The `TSPProfile` defines values at bow (typically lowest for submarines, ~5 dB), beam (highest, ~25 dB), and stern (~10-20 dB), with bilinear interpolation in between. For submarines, TS also varies with depth (reducing at -1 dB per 100 m) and speed (increasing at +2 dB per m/s above quiet speed due to cavitation onset). Radiated noise profiles define both broadband levels and narrowband tonal lines, with speed-dependent and depth-dependent penalties. Presets include the Kilo SSK (110 dB broadband at quiet speed, +8 dB per m/s above 2.6 m/s) and a DDG (130 dB broadband, +10 dB per m/s).

5. Radar Modeling

The radar package implements the monostatic radar equation in dBm:

$$P_r = P_t + G_t + G_r + 10 \log_{10} \left(\frac{\lambda^2 \sigma}{(4\pi)^3 r^4} \right)$$

Radar horizon is computed using the 4/3 Earth radius model:

$$d_{\text{horizon}} = \sqrt{2 R_{\text{eff}} h_a} + \sqrt{2 R_{\text{eff}} h_t}$$

where $R_{\text{eff}} = \frac{4}{3} \cdot 6,371$ km. Sea clutter uses a simplified GIT model with sea-state-dependent σ_0 , and Swerling fluctuation models (I through IV) provide detection probability adjustments. Detection probability is estimated via a simplified Marcum Q-function approximation using Albersheim's equation.

6. Sensor Management & Track Keeping

6.1 Sensor Manager

The `SensorManager` maintains a collection of sensors (active sonar, passive sonar, towed array, sonobuoy, radar, ESM, IR, MAD) and handles scheduling. Active sonar pings are emitted on a configurable interval (default 30 s). Sonobuoy fields can be deployed in circular patterns with configurable type (DIFAR, LOFAR, DICASS) and 4-hour lifetime. Towed arrays can be deployed and recovered dynamically. Search pattern generators produce waypoints for creeping search, expanding box, barrier, and dart patterns.

Platform-specific sensor suites are provided as factory functions: `DDGSensorSuite()` returns an SQS-53C active sonar (3.5 kHz, 30 km range), SQR-19 towed array (100 Hz, 50 km range), SPS-67 and SPS-49 radars, and SLQ-32 ESM. `SubSensorSuite("SSK")` returns an MGK-400 passive sonar and MG-519 active sonar with appropriate ranges.

6.2 Track Manager and Alpha-Beta Filter

The `TrackManager` maintains tracks with classification confidence levels (Possible, Probable, Confirmed), side identification (Friendly, Hostile, Neutral, Civilian), and state (Tentative, Firm, Precise, Lost). Position updates use an alpha-beta filter:

$$\hat{x}_k = \hat{x}_{k-1} + \alpha(z_k - \hat{x}_{k-1})$$

where α starts at 0.5 and decreases to 0.2 after 5 updates for smoother tracking. Tracks are promoted from Possible to Confirmed based on update count (10 updates = Confirmed). Lost tracks are declared after a configurable timeout (default 300 s).

6.3 Bearing-Only TMA

The `TMAEstimate()` function performs bearing-rate analysis from a history of passive sonar bearings. When the bearing rate is nearly zero, the target is assumed to be far (~50 km); otherwise, range is estimated as $v_{\text{assumed}} / \text{bearing rate}$. ESM bearing-only TMA uses a simplified linear regression of bearing versus time, with full EKF implementation noted as a future enhancement.

7. AI & Command and Control

7.1 Commander AI

The `CommanderAI` evaluates all active tracks through a `ThreatAssessor` that computes a composite threat score (0–1) based on:

- Range factor: +0.4 if <5 km, +0.2 if <15 km, +0.1 if <30 km.
- CPA factor: +0.3 if CPA <1 km, +0.15 if <5 km.
- Closing rate: +0.2 if CPA time <5 minutes.
- Submarine bonus: +0.15 for contacts below surface.
- Speed factor: +0.1 for contacts above 10 m/s.
- Side classification: +0.2 for hostile, 0 for friendly, ×0.3 for neutral.

Decisions follow a threshold-based logic: threat score ≥ 0.7 and range < 15 km triggers attack (subject to ROE), score ≥ 0.4 triggers track/pursue, and below 0.4 continues patrol. A minimum 10-second interval prevents decision thrashing.

7.2 Rules of Engagement

The `ROEEngine` enforces three postures:

Posture	Condition	Result
Weapons Free	Target classified hostile	Engage authorized
Weapons Tight	Hostile act confirmed & in range	Engage authorized
Weapons Hold	Self-defense only	Engage denied unless fired upon

Self-defense always authorizes return fire regardless of posture. Range gates (min/max) further constrain engagement authorization.

7.3 Formation Management

The C2 package provides formation management for column, line abreast, wedge, circle, and ASW screen patterns. The ASW screen generator places ships evenly around a guide at a configurable radius. A `ShootLookShoot()` function computes optimal salvo size for a given per-shot kill probability, firing until cumulative $P_k \geq 0.9$ or ammunition is exhausted.

8. Weapons & Damage

8.1 Anti-Ship Missiles

The ASHM package models missile flyout in three phases: boost (accelerate to cruise), cruise (sea-skimming at constant altitude), and terminal (accelerated approach). Probability of kill depends on target RCS and seeker type. CIWS engagement is modeled in the terminal phase with a per-burst SP_k of 0.6, and chaff seduction diverts missiles with 50% probability when chaff is active and the missile is within 50%

of seeker range. Presets include the RGM-84 Harpoon (Mach 0.7, 124 km range, 221 kg warhead), YJ-83 (Mach 0.8/0.9 terminal, 200 km), and MM40 Exocet (Mach 0.9, 70 km).

8.2 Torpedoes

Torpedoes progress through runout (wire-guided straight run to enable range), search (snake, circle, or spiral pattern), and homing (proportional navigation with $N = 4$ and $20^\circ/s$ max turn rate). The Mk 48 ADCAP preset specifies 28 m/s max speed, 38 km range, 800 m depth rating, 300 kg blast warhead, and snake search pattern. Warhead damage is computed for both under-keel hits (structural damage 0.6–0.9, flooding rate proportional to warhead weight) and side hits (structural damage 0.3–0.5). Shaped-charge warheads receive a $2.5\times$ effectiveness multiplier.

8.3 Ship Damage Model

The `ShipDamageModel` divides the hull into watertight compartments, each with volume, water level, door state, and fire status. Damage applies blast overpressure (triggering fire above 50 kPa, magazine explosion above 30 kPa if a magazine is hit). Flooding propagates through open doors between compartments. List angle is computed from asymmetric flooding moments relative to metacentric height (GM). Capsize is declared at list $>45^\circ$ or trim $>30^\circ$. The DDG preset defines six compartments (forward main, forward machinery, aft machinery, helicopter hangar, Mk 41 VLS magazine, aft main) with a 1.5 m GM and 5,000 m³ reserve buoyancy.

9. Electronic Warfare

9.1 ECM: Chaff and Jamming

Chaff blooms reach peak RCS after a configurable time, then decay with a 3-minute half-life. Drift follows wind direction, and altitude decreases at the configurable fall rate. Noise jammers are modeled with jammer-to-signal ratio (JSR) computation:

$$\text{JSR} = 10 \log_{10} \left(\frac{\text{ERP}_j}{4\pi R_j^2 B_j} \right) - 10 \log_{10} \left(\frac{\text{ERP}_r}{\sigma \lambda^2 (4\pi)^3 R_r^4} \right)$$

Burnthrough range is found by binary search (50 iterations) for the range where JSR equals a required threshold. Torpedo decoys (Nemo, SCAD, Mobray types) model seduction probability based on range advantage over the ship and torpedo homing mode.

9.2 ESM

The ESM receiver model computes received power from free-space path loss, compares against receiver sensitivity, and attempts library classification using frequency, PRI, and pulse width matching against known emitter signatures. Bearing accuracy is a configurable parameter. A simplified bearing-only TMA using linear regression of bearing rate provides initial range estimates from passive sensors.

10. Hydrodynamics & Seakeeping

10.1 Sea Spectra

The hydro package implements the Pierson-Moskowitz spectrum for fully developed seas:

$$S(\omega) = \frac{\alpha g^2}{\omega^5} \exp\left(-1.25 \left(\frac{\omega}{\omega_0}\right)^4\right)$$

where $\omega_0 = g/U_{10}$ is the peak frequency and $\alpha = 0.0081$ is the Phillips constant. The JONSWAP spectrum adds a peak enhancement factor $\gamma = 3.3$ with fetch-dependent scaling. Both spectra can be decomposed into wave components for irregular sea surface generation via linear superposition with deterministic phases.

10.2 Six-Degree-of-Freedom Dynamics

The 6-DOF model solves coupled equations of motion in the body frame with added mass, linear and quadratic damping, Coriolis/centripetal, and hydrostatic restoring forces:

$$m + m_{a,i} \ddot{x}_i = F_{\text{damping},i} + F_{\text{coriolis},i} + F_{\text{restoring},i} + F_{\text{prop},i} + F_{\text{rudder},i}$$

for each of the six degrees of freedom (surge, sway, heave, roll, pitch, yaw). Euler integration advances the state by Δt per step. The MMG (Maneuvering Modeling Group) 3-DOF maneuvering model decomposes hull, propeller, and rudder forces using Abkowitz polynomial coefficients and propeller open-water characteristics ($K_T = K_0 + K_1 J + K_2 J^2$).

11. Scenario Engine & After-Action Review

11.1 MSEL Timeline

The Master Scenario Events List engine manages time-ordered events including scenario start/end, sensor activations, ROE changes, weather changes, injections, and weapon authorizations. Events are added to a timeline and triggered by the `Tick()` function when simulation time reaches the event time. Handler functions can be registered for each event type, enabling scripted scenario progression.

11.2 AAR Recorder

The `Recorder` captures entity state snapshots at configurable intervals and logs detection, weapon, and damage events. The `ComputeStats()` function summarizes total simulation time, event counts, and final entity states. The `ExportJSON()` function writes the complete AAR record — snapshots, events, and statistics — to a JSON file. A `Replay()` function produces time-indexed frames at a given time step for playback visualization.

A `ScoringEngine` allows scenario authors to define weighted objectives with custom score functions (e.g., “survived”, “contacts classified”, “hostiles neutralized”), computing an overall weighted score.

12. DIS Federation Bridge

The `fed/dis` package implements an IEEE 1278.1 (DIS 7) federation bridge that serializes Entity State PDUs (144 bytes each) and transmits them over UDP multicast. The `DISBridge` struct manages a UDP connection and publishes entity state updates with site, application, and entity IDs. The PDU format includes protocol version 7, exercise ID, force ID, entity type fields, and ECEF position/orientation/velocity. The engine publishes all entity states to registered federation bridges at each step, enabling integration with other DIS-compliant simulators in a distributed exercise.

13. Performance & Cross-Platform Deployment

The simulation compiles to a single statically-linked binary (~4.6 MB on Linux amd64). A 2-hour ASW scenario with 3 entities, sensor detection checks every 10 steps, AI decisions every 30 steps, and AAR snapshots every 100 steps runs in real time at $1\times$ time scale with negligible CPU usage on modern hardware. The engine's mutex-protected entity map supports concurrent reads from the web UI SSE endpoint while the main loop advances.

Cross-platform builds are produced via Go's native cross-compilation:

Platform	Format	Size
Linux amd64	.tar.gz / .deb	~4.6 MB
Linux arm64	.tar.gz	~4.3 MB
macOS Intel	.tar.gz	~4.6 MB
macOS ARM	.tar.gz	~4.4 MB
Windows amd64	.zip	~4.7 MB
Docker	stsgym/maritime-sim:latest	Alpine-based

The web UI uses an embedded filesystem (`embed.FS`) to serve a Leaflet.js tactical display that receives real-time position and track updates via Server-Sent Events. The

`/api/state` REST endpoint provides the current tactical picture as JSON for integration with external tools.

14. Conclusion

We have presented a comprehensive maritime warfare simulation engine that integrates physically-grounded acoustic propagation (Mackenzie SSP, multi-path TL, reverberation), radar detection, aspect-dependent signatures, 6-DOF hydrodynamics, weapon flyout and damage modeling, AI-driven command decisions under ROE constraints, and IEEE 1278.1 DIS interoperability — all in a single portable binary. The modular package structure allows individual models to be validated against analytical solutions and at-sea measurements, while the YAML scenario format and JSON AAR output facilitate both interactive exploration and batch-mode analysis.

Key areas for future development include: (1) a full extended Kalman filter for bearing-only TMA replacing the current linear regression; (2) Rayleigh reflection coefficients in the bottom-loss model for shallow-water propagation; (3) cooperative ASW with multi-platform sensor fusion; (4) environmental injection from real-time oceanographic data; and (5) integration with higher-fidelity propagation models such as GRAM or RAM for validation benchmarks.

References

1. Urick, R. J. Principles of Underwater Sound, 3rd ed. Peninsula Publishing, 1983.
2. Mackenzie, K. V. “Nine-term equation for sound speed in the oceans.” *Journal of the Acoustical Society of America*, 70(3):807–812, 1981.
3. Pierson, W. J. and Moskowitz, L. “A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii.” *Journal of Geophysical Research*, 69(24):5181–5190, 1964.
4. Hasselmann, K. et al. “Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP).” *Deutsche Hydrographische Zeitschrift, Reihe A*, 8(12), 1973.
5. IEEE Standard for Distributed Interactive Simulation — Application Protocols. IEEE Std 1278.1-2012.

6. Chapman, R. P. and Harris, J. H. "Surface backscattering strengths measured with explosive sound sources." *Journal of the Acoustical Society of America*, 34(10):1592–1597, 1962.
7. Francois, R. E. and Garrison, G. R. "Sound absorption based on ocean measurements: Part I: Pure water and magnesium sulfate contributions." *Journal of the Acoustical Society of America*, 72(3): 896–907, 1982.
8. Wenz, G. M. "Acoustic ambient noise in the ocean: Spectra and sources." *Journal of the Acoustical Society of America*, 34(12):1936–1956, 1962.
9. Munk, W. H. "Sound channel in an exponentially stratified ocean, with application to SOFAR." *Journal of the Acoustical Society of America*, 55(2):220–226, 1974.
10. Abkowitz, M. A. *Stability and Motion Control of Ocean Vehicles*. MIT Press, 1969.
11. Yasukawa, H. and Yoshimura, Y. "Introduction of MMG standard method for ship maneuvering predictions." *Journal of Marine Science and Technology*, 20:37–52, 2015.
12. Skolnik, M. I. *Radar Handbook*, 3rd ed. McGraw-Hill, 2008.
13. Albersheim, W. J. "A closed-form approximation to Robertson's detection probability." *IEEE Transactions on Aerospace and Electronic Systems*, AES-17(4):517–521, 1981.
14. Naval Oceanography Command. NAVOCEANO Oceanographic and Atmospheric Master Library. Stennis Space Center, MS.