

Mission Possible: Dead Reckoning with Artificial Intelligence

Ted McRobie
Thales UK
Reading, Berkshire
ted.mcrobie@uk.thalesgroup.com

ABSTRACT

A stalwart of navigation and simulation, Dead Reckoning (DR) is a method which calculates an entity's current position by extrapolating from its previous state using known speed, heading, and elapsed time. However, traditional DR approaches rely on the assumption that incoming data streams are accurate and trustworthy, often resulting in unrealistic behaviour, lack of interactivity, and catastrophic vulnerability to data poisoning. These limitations compromise immersion, reduce training effectiveness, and expose systems to potential manipulation.

We propose a novel approach which integrates Artificial Intelligence (AI) and Machine Learning (ML), particularly Reinforcement Learning (RL), to enhance the realism, autonomy, and resilience of simulated entities. By leveraging historical AIS data and procedurally generated maritime environments, we aim to train RL agents capable of navigating complex scenarios with greater situational awareness and responsiveness. The proposed system architecture includes a simulation environment built using the Gymnasium framework for the integration of a convolution-based perception model, an anomaly detection layer to mitigate the effects of poisoned or unreliable data, and common interfaces to live AIS data, real-world terrain, and distributed simulation environment protocols for interoperability. We outline an experimental methodology using Deep Q-Networks (DQN), with the potential to scale to more advanced algorithms.

Evidence suggests that this combination of AI/ML with DR principles may yield improvements in realism, interactivity, and responsiveness, enhancing training and ensuring operational dominance in synthetic environments. Our proposed approach aims to produce digital shadows which maintain intelligent, context-aware behaviour even in the presence of data dropouts or latency, enabling entities to navigate plausibly and avoid unrealistic actions such as traversing terrain or disregarding battlespace constraints.

ABOUT THE AUTHORS

Ted McRobie graduated from the University of Portsmouth with a B.Sc.(Hons) in Physics, Astrophysics, and Cosmology and went on to graduate from the University of St Andrews with a M.Sc. in Astrophysics. During his time at University he completed projects on Theoretical Astrophysics, developing novel methodologies to model and simulate the first Stars. He is a Machine Learning Scientist at Thales Training and Simulation exploring Artificial Intelligence and its applications to Digital Twins.

Mission Possible: Dead Reckoning with Artificial Intelligence

Ted McRobie
Thales UK
Reading, Berkshire
ted.mcrobie@uk.thalesgroup.com

1 INTRODUCTION

Dead Reckoning (DR) is a term which emerged in the 1600s, utilized by navigators to gauge their course and distance when neither landmarks nor stars were visible. The term's origin relates to a technique whereby a log or floating object was released from the bow of the ship and the time it took for its passage of the stern was observed to estimate the ship's speed. Combining this with a compass to estimate heading and accounting for the effects of wind and ocean currents, navigators of the time could chart a route with a fair degree of precision for short distances. This technique is still used today in Modelling and Simulation (M&S) systems, where the basis of modern DR is that a platform's last known position and an approximation of the path it has travelled since can be used to estimate the platform's current location using Newtonian mechanics.

To be specific, DR is implemented in modern M&S systems to interpolate through and extrapolate from data streams from live entities sourced from third party application programming interfaces (APIs) to generate a live operating picture. This data is typically based on, or implicitly calculated from various sensors on the platforms themselves. The need for this has produced a market for businesses to detect and host broadcasted Automatic Identification System (AIS) or Automatic Dependent Surveillance-Broadcast (ADS-B) data from platforms, and today there are a range of data sources available online (AirNav, 2001; MarineTraffic, 2007; AviationEdge, 2010). This approach generates value by transforming simulated scenarios into dynamic live operating environments, offering operators an enriched training experience across not only maritime but also land and aerospace domains. Crucially, the mathematics behind modern DR is largely unchanged since the 17th century.

Model	Formula	Description	Error Scaling
DRM(FPW)	$P = P_0 + V_0\Delta t$	Fixed Orientation, World Frame	$\mathcal{O}(\Delta t)$
DRM(RPW)	1. $P = P_0 + V_0\Delta t$ 2. $[R]_{w \rightarrow b} = [DR][R_0]_{w \rightarrow b}$	Rotating Orientation, World Frame	$\mathcal{O}(\Delta t)$
DRM(RVW)	1. $P = P_0 + V_0\Delta t + \frac{1}{2}a\Delta t^2$ 2. $[R]_{w \rightarrow b} = [DR][R_0]_{w \rightarrow b}$	Rotating Velocity, World Frame	$\mathcal{O}(\Delta t^2)$
DRM(FVW)	$P = P_0 + V_0\Delta t + \frac{1}{2}a\Delta t^2$	Fixed Velocity, World Frame	$\mathcal{O}(\Delta t^2)$
DRM(FPB)	$P = P_0 + [R]_{w \rightarrow b}^{-1}([R_1]V_b)$	Fixed Orientation, Body Frame	$\mathcal{O}(\Delta t)$
DRM(RPB)	1. $P = P_0 + [R]_{w \rightarrow b}^{-1}([R_1]V_b)$ 2. $[R]_{w \rightarrow b} = [DR][R_0]_{w \rightarrow b}$	Rotating Orientation, Body Frame	$\mathcal{O}(\Delta t)$
DRM(RVB)	1. $P = P_0 + [R]_{w \rightarrow b}^{-1}([R_1]V_b + [R_2]A_b)$ 2. $[R]_{w \rightarrow b} = [DR][R_0]_{w \rightarrow b}$	Rotating Velocity, Body Frame	$\mathcal{O}(\Delta t^2)$
DRM(FVB)	$P = P_0 + [R]_{w \rightarrow b}^{-1}([R_1]V_b + [R_2]A_b)$	Fixed Velocity, Body Frame	$\mathcal{O}(\Delta t^2)$

Table 1: IEEE Standard DR formulæ (IEEE, 1998) and their corresponding error scaling with respect to time.

Different formulæ are used dependent on reference frame and use case (see Table 1), but the logic behind them remains the same. While effective in controlled environments, the assumptions behind this logic can become problematic in dynamic or data-degraded scenarios. This can be due to sensor inaccuracies or environmental effects (although they are significant), but predominantly because the formulæ error scales with the temporal differential between which states are assessed. The risk associated with this iterative error is typically mitigated through frequent updates from trustworthy sources, though such trust is rarely absolute in practice. This solution is unfit as the fact remains that

there are large cumulative time differentials between "ground truth" data points - which are often so large that the reconciliation of differences between the simulated entity and its real counterpart is not possible without impacting the realism of the simulation.

"Trust" in data sources implies accurate, reliable, and verifiable information. However, the unclear intentions behind end-users' data usage can result in significant risks, regardless of system robustness. This underscores the need for a zero-trust mindset. Additionally, API-based systems may be vulnerable to data poisoning and may require low-latency wireless networks that are challenging for military operations. Thus, a different or adapted solution is necessary.

1.1 Machine Learning & Artificial Intelligence

Machine Learning (ML) is a subset of Artificial Intelligence (AI) that empowers computers to make inferences from data without being explicitly programmed. At its core, ML focuses on pattern recognition within datasets, enabling systems to make predictions and decisions based on new, unseen information. For example, email providers use algorithms to analyze past data (emails marked as spam by users) to identify patterns and characteristics that signify unwanted messages and classify those messages as spam (Dada et al., 2019; Kumaran, 2019).

One particularly advanced area is Deep Learning, which mimics the way brains operate by utilizing neural networks composed of multiple layers of perceptrons (Rosenblatt, 1958; Rumelhart, Hinton, and Williams, 1986). This complex architecture allows these systems to perform abstract problem solving and reasoning on a much higher scale than traditional ML algorithms. For instance, Deep Learning has made remarkable strides in image and voice recognition (Krizhevsky, Sutskever, and Hinton, 2017; Hannun et al., 2014), facilitating advancements in technologies such as self-driving cars and virtual assistants (Bojarski et al., 2016; Baskaran et al., 2021). These systems can recognize objects, speech patterns, and even emotions, exhibiting human-like behavior that's increasingly integral to our daily lives.

Another important subset of ML is Reinforcement Learning (RL), which further explores how machines can learn from their own actions and experiences. In RL, models are trained to make sequences of decisions by receiving feedback in terms of rewards or penalties, much like humans learn through trial and error. This method not only enables machines to solve complex problems in dynamic environments but also encourages them to develop strategies akin to human reasoning. For instance, RL has been used in developing game-playing algorithms that can outperform the best human players by learning and adapting their strategies in real-time (DeepMind, 2019). Together, these advancements are paving the way for increasingly intelligent systems that resonate with our understanding of cognition and behavior.

We propose that a combination of Deep Learning and RL is used to autonomously control the dynamics and behaviour of digital shadows of real platforms which are integrated into live operating synthetic environments. This adds value by resolving the following issues currently facing operators training in synthetic environments:

1. The interactivity of entities within simulations are compromised or entirely non-existent.
2. Specific scenarios and training exercises are susceptible to disruption by live entities.
3. Simulations become vulnerable to the risk of data poisoning.
4. Infrequent data updates and substantial latency issues break immersion.
5. Entities may operate outside of the constraints of simulation-defined parameters, such as terrain.

1.2 Objectives

The objective of this article is to propose a novel methodology for enhancing DR in synthetic environments by integrating AI/ML techniques, particularly RL. Specifically, we aim to:

- Identify and synthesize the limitations of traditional DR methods in dynamic and data-sparse environments.
- Propose an architecture which allows an RL agent to train on historical AIS data in procedurally generated environments to gain a generalized understanding of context-aware navigation.

- Outline a system framework which supports a robust, adaptive agent while mitigating the impact of unreliable or poisoned data sources.
- Establish a foundation for future integration of this system into distributed simulation frameworks such as DIS and HLA.

A system which meets these objectives should allow for more realistic behaviour of simulated entities - specifically where data is delayed, missing, or compromised - enabling them to react intelligently to their environment and operators. This has direct benefits for end users in training and test environments, offering more immersive and interactive scenarios, reducing reliance on implicitly trusted data streams, and supporting robust performance under degraded conditions. The system is intended to enhance training value, improve operator engagement, and lay the groundwork for future integration into interoperable simulations and digital twin frameworks.

2 MOTIVATION

2.1 Problem Synthesis

The application of traditional DR in simulated navigation systems presents several challenges that can hinder effective operational performance. Consider a scenario where three trusted data points provide a reliable foundation for determining a vessel's position (see Figure 1). These data points are considered completely trustworthy and accurate. This is a critical flaw; there is likely a large degree of error in the extrapolation of future dynamics; or the data used to calculate future dynamics may have been poisoned by an adversary. This induced bias results in formulæ producing flawed predictions of the vessel's trajectory.

Traditional DR calculates a deterministic curve of a platform's future movements. Consequently, in the likely event new data does not align with the current motion—often due to external influences or disruption—teleportation to a new position and new behaviour must be reconciled. This abrupt adjustment is problematic for operators, as not only do entities vanish from one location and reappear in another, these platforms do not react to simulation events and are non-interactive. This breaks the immersion and eliminates realism.

The lack of situational awareness exacerbates this issue. Entities in the simulation operate purely based on live feeds, devoid of any contextual understanding of their environment. This absence of awareness limits operators' ability to anticipate actions or respond effectively to the ongoing dynamics of the simulation, ultimately rendering the entire experience less immersive and narrow in scope.

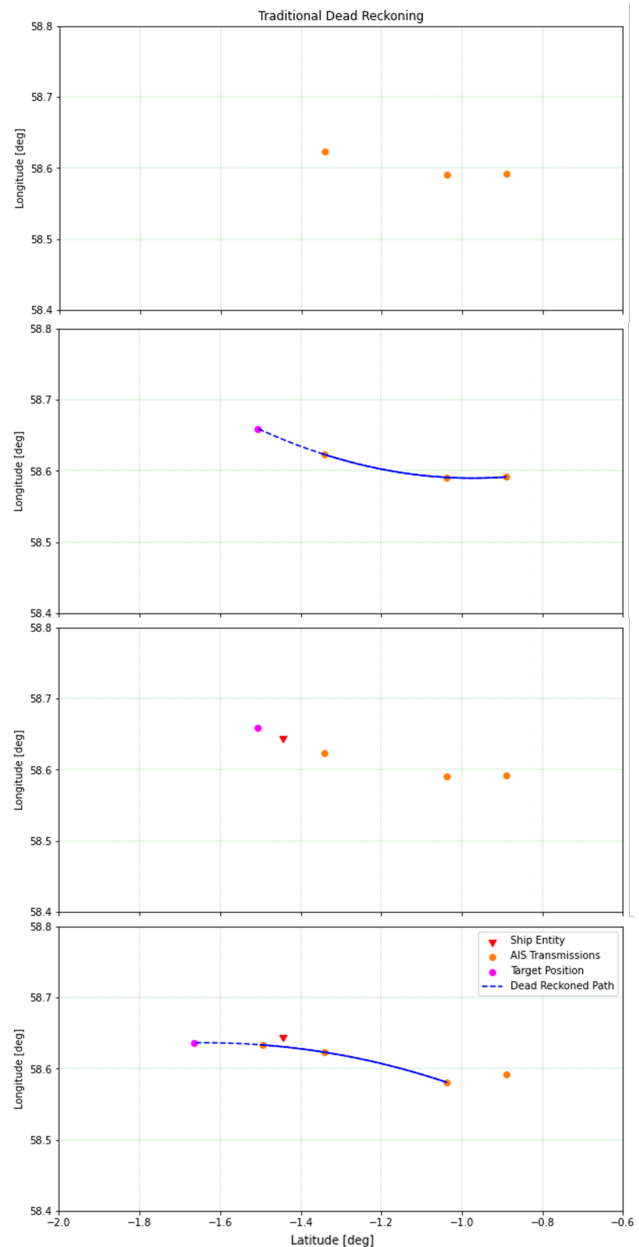


Figure 1: Plots illustrating the error of traditional DR. A notionally trusted set of real AIS data (*top*) is used to calculate trajectory of a simulated entity (*upper middle*). When new information becomes available which deviates from the original dataset (*lower middle*), this incurs error, which then becomes impossible to resolve while preserving realism (*bottom*).

Traditional remedies to this problem often involve adding layers of complexity to the prediction calculations in an attempt to achieve a more accurate approximation of the entity's position—a pursuit for closer alignment with a supposed ground truth. While this approach may yield marginal improvements in accuracy, it fails to eliminate the inherent errors embedded within the equations used to approximate the movement. Since these errors are built into calculations, they persist as an unsolvable element of the process.

The most effective solution to these challenges lies in granting entities autonomy. By embedding autonomous decision-making capabilities within the simulated entities, the system can better account for the complexities of navigation in real time. However, this autonomy introduces its own set of challenges, creating a trade-off between realism and representativeness in the simulation. On one hand, enhanced autonomy may lead to more lifelike behaviours that closely mirror how vessels operate in the real world. On the other, it could distance the simulation from where the physical entity is in the real world and how it is behaving.

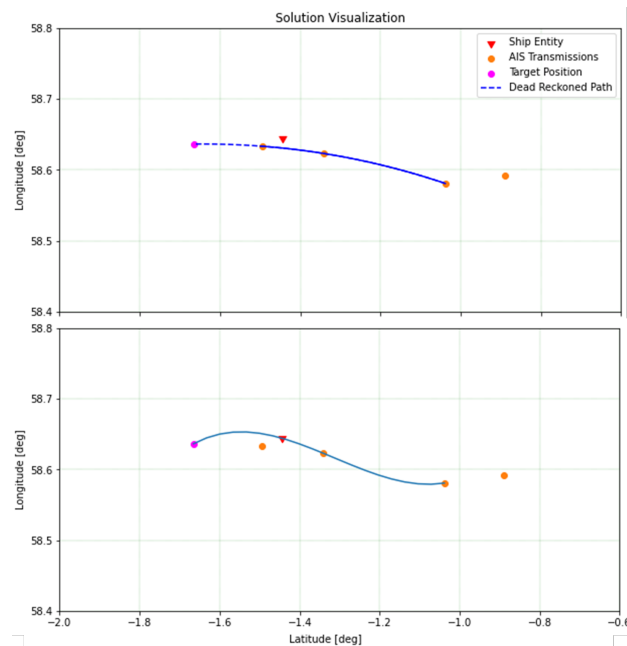


Figure 2: Illustration of enhanced realism in adopting autonomous methods as opposed to traditional DR methods.

representative behaviors which is illustrated in Figure 2.

The inclusion of situational awareness in autonomous platforms allows them to make intelligent decisions about how to react to simulation events, even those that may not mirror real-world occurrences. For instance, an autonomous entity can recognize an emergent threat in the simulation and alter its course or alter its mission priorities to address that threat effectively. This capacity for proactive engagement provides operators with substantial interactivity, enabling them to influence the simulation meaningfully.

This shift towards realistic interactions fosters improved training outcomes, as operators can engage with entities in ways that mimic tactical decision-making processes encountered in real operations. As a result, training scenarios prepare operators to respond adeptly to various contingencies. The heightened realism, combined with situational awareness and interactivity, ensures that operators are better equipped to achieve operational dominance.

There are existing examples of implementing AI/ML for DR in the physical domain in the context of navigation (Giordano, 2024) through leveraging a combination of platform sensors, last known GPS position, and compensation functions. However, this technology is designed for the purposes of navigation of a physical platform as opposed to the simulation of a larger, interoperable system; and is locked to the maritime domain for Unmanned Surface Vessels (USVs).

2.2 Solution Proposal

An autonomous system can enhance the realism within simulation environments (Kerbusch, Keijser, and Smit, 2018; Ahammed, 2024), fostering a more effective training and operational framework that directly contributes to achieving operational dominance. However while the integration of autonomy increases realism in the simulation, it is essential to acknowledge the potential discrepancies between simulated and real-world behaviors. This divergence is more pronounced in traditional DR systems where the entities are merely “following the math” and maintaining linear trajectories through terrain without taking external influences into account. In contrast, autonomous systems are equipped to navigate complex environments, adjusting their movements based on terrain, obstacles, and real-time challenges, leading to more

3 METHODOLOGY

3.1 Reinforcement Learning Agent

The complexity of an RL agent extends beyond mere decision-making; it encompasses various architectures and methodologies that influence its effectiveness. Broadly, RL techniques can be categorized into model-free and model-based approaches. Model-free methods, such as Q-learning and policy gradients, focus solely on learning from interactions and rewards without constructing an explicit model of the environment. In contrast, model-based methods attempt to create a representation of the environment, allowing the agent to simulate potential outcomes and plan its actions more strategically. This distinction underscores a critical divergence in the learning objectives of RL systems. Given the goal of optimizing an agent's ability to navigate a ship in an environment where the action and state spaces are relatively straightforward, we advocate for the use of model-free methods. This approach is particularly beneficial as we anticipate increasing the complexity of the environment through development, allowing the agent to effectively adapt and learn from direct interactions without the constraints of a modeled environment.

We propose experimenting with a simplified version of the problem. For this phase, we propose implementation of Deep Q-Network (DQN) due to its effective value-based policy optimization, reliably stable training process, and rapid experiment iteration (Mnih et al., 2015; Fan et al., 2020; Torres, 2020) making it a prime candidate for decision-making in structured simulation environments (Su et al., 2024). Following this prototyping stage, we propose incrementally increasing environment and action space complexity and analysing the capabilities and efficacy of various on- and off-policy Reinforcement Learning (RL) algorithms for improved stability and performance such as Twin Delayed Deep Deterministic Policy Gradient, Proximal Policy Optimization, and Soft-Actor Critic.

3.1.1 Environment

The environment is designed to offer a challenging yet adaptable simulation for training RL agents. The terrain is developed using 2D simplex noise, which facilitates the generation of varied and natural-looking land masses (Archer, 2011). This noise is manipulated through a square logarithmic gradient to produce diverse shapes and sizes of islands and coastlines. The application of a threshold allows these land masses to be centered within a defined array, encouraging the emergence of intricate edges and channels. This results in a complex navigational landscape that promotes dynamic pathfinding challenges for the simulated ship platform, as depicted in Figure 3.

The environment is built upon the widely adopted Gymnasium framework (Towers et al., 2024), ensuring compatibility with existing RL tools and libraries, including Stable Baselines3 (Raffin et al., 2021). This integration allows for streamlined development and testing, taking advantage of community support and resources. Furthermore, the implementation of the environment should foster ease of use, enabling straightforward customization and extension, which permits the addition of further complexity as needed during training and experimentation.

To maintain realism of ship dynamics and control, traditional Newtonian mechanics are employed at a higher update rate to model the vessel's dynamics. This approach provides a solid foundation for how the ship behaves with minimal cumulative error, thus enhancing the authenticity of the simulation. The learning agent is granted control over two critical parameters: acceleration and angular velocity. This dual control aspect empowers the agent to execute nuanced navigation strategies, enabling effective maneuvering of the ship through varying terrain and the avoidance of obstacles.

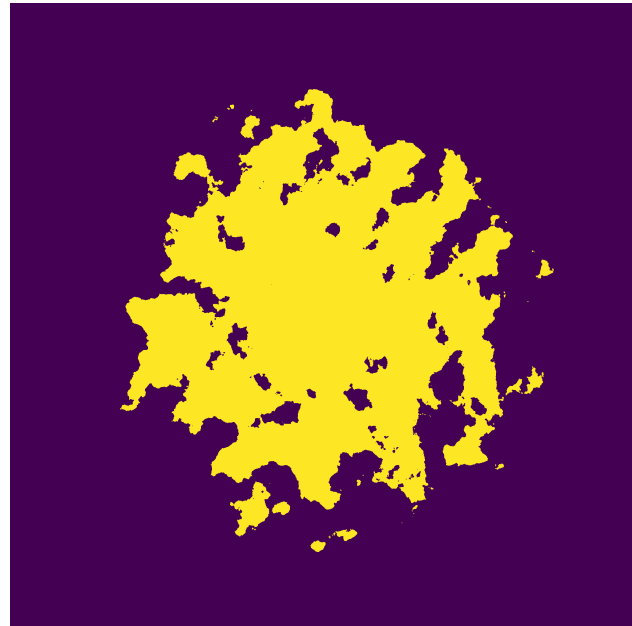


Figure 3: Example of procedurally generated terrain (yellow) generated using simplex noise method.

The agent requires suitable observations in order to draw inferences from the result of cumulative actions. Navigation can be a challenging problem to solve as terrain morphology is abstract and basic observations can often oversimplify the problem. A common approach to this problem calculating the closest point to a coastline, or the distance to coastline at discrete bearing increments. While these solutions can be suitable for individual frames to allow an algorithm to infer future collisions, this solution does not build an adequate picture of the morphology of its surroundings.

We propose that a small 2 dimensional image array of a platforms immediate surroundings is captured and run through a convolution layer, and the resulting feature map is flattened and integrated into the observation space (Figure 4). This preprocessing step allows for the extraction of essential features from the surrounding environment while reducing the dimensionality of the input data. This should improve the algorithms spatial perception capabilities (Mnih et al., 2013) as well as learning stability (Van Hasselt, Guez, and Silver, 2016).

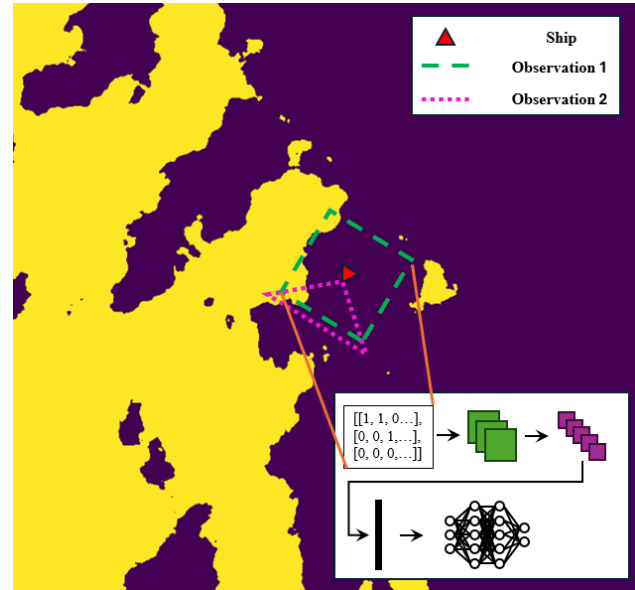


Figure 4: Illustration of observation space proposal. The immediate surroundings of the ship will be captured, convolved, flattened, and then fed into the observation space of the algorithm.

3.1.2 Data

The cornerstone of the training process will be historical AIS data which can be acquired from a range of online AIS data services (MarineTraffic, 2007; Astra Paging Ltd, 2011). This data provides detailed information about vessel positions, including latitude and longitude, speed, heading, and timestamps as visualized in Figure 5. Such data

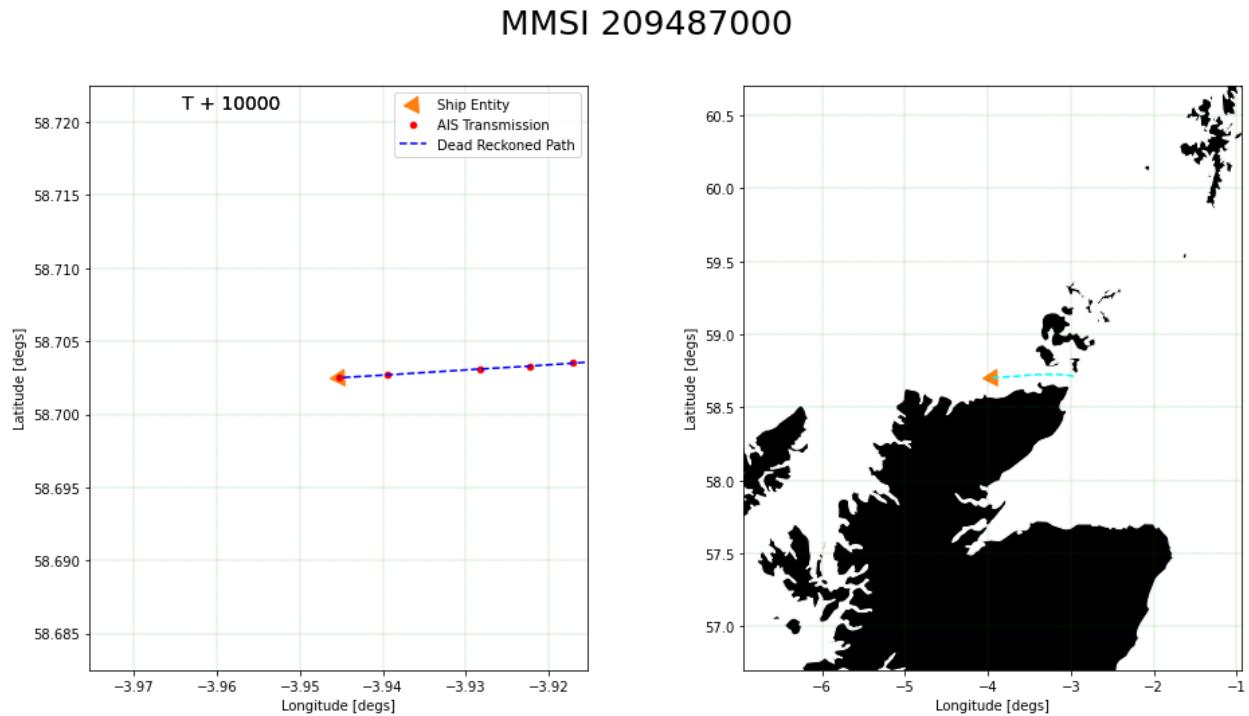


Figure 5: Example of a real platform navigating around Scotland detected through AIS broadcasts.

enables a thorough analysis of navigational behaviors and movement patterns over time, thereby allowing for realistic simulations of ship movements during training. In addition to AIS data, the incorporation of relevant environmental factors is paramount to ensure that the RL agent can effectively navigate dynamic maritime environments. This includes topographical maps that illustrate coastlines and other potential navigational hazards.

With respect to the training methodology, AIS transmissions will be broadcast in a sequence of time-series simulations which the agent will detect. Each simulation episode will commence by initializing the ship's position based on the first detected track of a ship Maritime Mobile Service Identity (MMSI) number. The agent will navigate within simulated environments that reflect historical and environmental factors, enabling it to learn from realistic scenarios. The agent's target waypoint will be the next AIS track to be broadcast for that vessel, and logging its decision-making activities to assess rewards and penalties related to successfully reaching target waypoints. Once the vessel has reached the target waypoint or a time threshold based on when the next AIS broadcast was detected has lapsed, the episode is reset and a new target waypoint is set.

In production, reliance on historical AIS data will no longer be possible. To address this limitation, a DR system will serve as a foundational method for waypoint prediction. This traditional navigation technique estimates a vessel's current position from a previously known location, accounting for speed, time, and directional changes.

3.1.3 Reward

We propose positive rewards be assigned for desirable actions, such as successfully reaching a waypoint. A substantial and discrete positive reward can be granted upon arrival at each waypoint, encouraging the agent to prioritize this objective. Additionally, smaller continuous rewards can be provided for each step taken towards the waypoint, promoting progress and sustained movement in the right direction. If the agent maintains a safe distance from land or obstacles while en route, additional continuous rewards to reinforce safe navigation practices and distance management could yield positive results.

Conversely, we propose penalties be implemented to discourage unwanted actions. A significant penalty should be incurred followed by immediate simulation termination if the agent collides with terrain, providing a deterrent for such actions. If the agent strays too close to land, a smaller continuous penalty can be applied to encourage it to adjust its course and maintain safe navigation. Furthermore, if the agent significantly deviates from the direction of the waypoint (for instance, if it moves away from the target instead of towards it), a modest penalty can be applied to discourage inefficient movement patterns.

Another layer of complexity can be introduced by incorporating temporal penalties, wherein the agent incurs a penalty for reaching waypoints too early or too late. This incentivizes a balance between cautious navigation and efficient route planning.

Ultimately, the design of the rewards and penalties should be iterative and may require tuning based on empirical results to establish a balance that encourages optimal agent behavior while effectively navigating the challenges of the environment. In essence, the reward structure should foster goal-directed actions while minimizing potentially harmful behaviors, enabling the agent to learn successful navigation strategies over time.

3.2 System Architecture

The proposed system architecture (see Figure 6) revolves around a central system management component, which serves as the core orchestrator of the entire maritime navigation framework. This system manager is designed to interface with a backend gymnasium environment which hosts the RL agent. By situating the system manager at the heart of the architecture, we establish a scalable, flexible and efficient control mechanism which can coordinate various subsystems, ensuring coherent operation and data flow throughout the platform.

Integral to this architecture is the terrain interface, which allows the system manager to accept various geospatial data formats, including geotiffs and Digital Terrain Elevation Data (DTED) files. The adoption of open standards such as those endorsed by the Open Geospatial Consortium (OGC) promotes interoperability among diverse systems and datasets. This is particularly advantageous for defense applications, where the DTED structure is widely utilized for

terrain representation. By adhering to these open standards, the architecture facilitates seamless integration with different data sources, ensuring that the system can be readily adapted to various operational contexts and requirements.

Furthermore, the system manager's connectivity extends to a live AIS provider, allowing the architecture to detect real-time AIS broadcasts from vessels operating in the vicinity. This capability enables the RL agent to simulate these live platforms, creating representative models of existing maritime operations. The incorporation of real-time data ensures the RL agent can engage with dynamic and realistic scenarios, thereby enhancing the training and operational effectiveness of the overall system. By simulating these live platforms, the architecture can also support interactive operations in live simulations, providing valuable insights and predictive capabilities based on current physical and digital maritime conditions.

To amplify the system's functionality and compatibility with existing military and civilian frameworks, we envision the use of commonly accepted distributed simulation protocols such as the Distributed Interactive Simulation (DIS) and High-Level Architecture (HLA). By broadcasting the simulated scenarios across this digital thread, the architecture enables many operators to participate in the simulation environment, fostering collaboration among various entities. This interoperability with established simulation frameworks allows for a more comprehensive understanding of maritime operations and enhances the training capabilities of personnel involved in defense scenarios.

Finally, we propose an anomaly detection module utilizing AI-driven algorithms be implemented. This module would analyze real-time data using machine learning techniques such as Isolation Forests (Liu, Ting, and Zhou, 2008) or Autoencoders (Nelay and Turgeon, 2024) to identify outliers and inconsistencies. This decision-making layer would evaluate the validity of data based on detected anomalies, triggering real-time alerts for operators while dynamically adjusting the behavior of autonomous systems based on data reliability.

4 DISCUSSION

4.1 Implementation

Prototyping has begun for this system, with the future ambition to integrate this system with the UK Digital Twin Centre's wider digital twin ecosystem; the Digital Crucible, to provide a comprehensive framework for optimized maritime simulations. This facilitates the development of generic digital shadows, which serve as real-time counterparts to physical vessels, continuously adapting based on incoming data and past experiences. We propose development begins with experimentation of the proposed prototype, allowing for initial validation of the predictive capabilities of the integrated AI models in maritime navigation and interactivity. This phase focuses on collecting data and assessing effectiveness of waypoint tracking, trajectory forecasting, and reactivity to changing environments. Following this experimentation, we propose advancing to incorporate more sophisticated RL techniques proposed in 3.1.

There are also a number of approaches considered throughout the design process which, through experimentation, could yield promising results:

- **Temporal joint probability approximation:** Deep learning can be harnessed to forecast gridded locations a vessel is likely to reach and the expected time to arrive. This approach uses temporal joint probability to

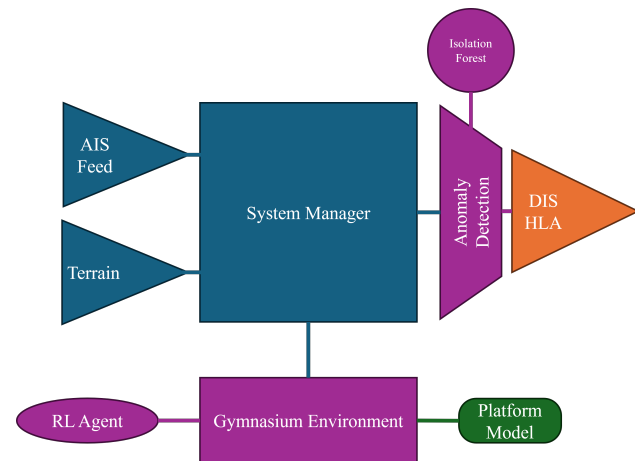


Figure 6: Design of the system architecture. A core system management component acts as the key interface between all subsystems. Live AIS data and terrain is fed into the system manager, and the Gymnasium environment controls the platform and its dynamics based on its initialised state dictated by that input. This is then analysed by an anomaly detection layer prior to being broadcasted to a wider interoperable system via a DIS or HLA framework.

capture the relationship between a ship's trajectory and travel time, allowing for more nuanced predictions. This predictive capability is additive to the situational awareness capabilities of the algorithm by estimating when the next message from the vessel will arrive, significantly improving operational responsiveness.

- **Physics-Informed Neural Network (PINN) Dynamics:** PINNs offer a sophisticated method for modeling platform behavior in maritime applications, achieving unprecedented fidelity in simulating platform dynamics (McRobie, 2024). By embedding physical principles, operator commands, and onboard sensor data into the neural network architecture, PINNs create a highly detailed representation of how various environmental factors influence a ship's movement and embeds trusted methods into the algorithm
- **Field of View Optimization:** An alternative approach to more efficiently feed the algorithm with relevant environmental features and reduce computational complexity is to reduce the field of view to where there is relevant information to be captured (Schaul et al., 2015). We propose a cone-shape slice as opposed to a 2 dimensional image array to convolve which closely aligns with the ships forward trajectory. By emphasizing the areas that impact immediate navigation decisions, this cone-shaped observation can improve the agent's perception of obstacles and navigational aids earlier in the training process. Such targeted observation could also reduce computational complexity, allowing the model to focus on crucial information rather than processing the entirety of the surrounding landscape.
- **Recurrent Neural Networks (RNNs):** the implementation of RNNs to process the sequence of observations over time enables the RL agent to capture temporal dependencies in its surroundings, effectively learning how the environment evolves and how past experiences influence current navigation decisions (Schäfer, 2008; Kap-turowski et al., 2019). By combining the spatial features learned from the convolution with environmental temporal dynamics, the agent could develop more robust navigation strategies.

4.2 Conclusion

In this article, we have proposed a novel approach to modernize DR in simulation environment by introducing AI/ML driven autonomy. The limitations of traditional DR methods were outlined, particularly its reliance on accurate, low-latency data, and presented a conceptual system architecture incorporating RL agents and real-time data ingestion.

The proposed methodology meets the state objectives by:

- Clearly identifying and detailing the core problems of legacy DR approaches.
- Presenting a feasible, modular system design integrating an RL agent trained on historical AIS data, and operating with live AIS data which resolve the itemized issues raised in 1.1.
- Defining an environment and reward structure capable of supporting intelligent agent training.
- Demonstrating how the system could interface with broader simulation standards to support interoperability and scalability.

The approach offers clear end-user benefits; enhanced realism in training scenarios, improved situational awareness of simulated entities, and greater resilience in the face of unreliable or delayed data. By enabling AI-driven entities to behave autonomously and intelligently under operational constraints, the system has the potential to reduce reliance on implicitly trusted data streams and support more immersive, effective training - ultimately contributing to improved preparedness and operational dominance in synthetic environments.

ACKNOWLEDGEMENTS

We wish to express gratitude to Ian Henderson for his insights and inspiration drawn from his original work, which has significantly impacted this research. We would also like to thank colleagues at Thales for their invaluable support and insightful feedback. Appreciation goes especially to Charlotte Grinyer and Kat Heseltine-Flynn, whose expertise and willingness to share their knowledge have greatly enriched this work. Their contributions have played a vital role in shaping this research.

References

- Ahmed, Gias (2024). “Sim-To-Real: Uniting Simulation and Reality For Unprecedented Results”. In: *Nothing But AI*.
- AirNav (2001). *AirNav Radar - Global Flight Tracking Intelligence — Live Flight Tracker and Airport Status — 54.96983,-3.39264,z5*. <https://www.airnavradar.com/@54.96983,-3.39264,z5>. [Accessed 23-04-2025].
- Archer, Travis (2011). “Procedurally generating terrain”. In: *44th annual midwest instruction and computing symposium, Duluth*. 2.5 Simplex Noise, pp. 378–393.
- Astra Paging Ltd (2011). *VesselFinder*. <https://www.vesselfinder.com/>. [Accessed 16-05-2025].
- AviationEdge (2010). *Satellite Tracker API - Aviation database and API — aviation-edge.com*. <https://aviation-edge.com/satellite-tracking-api/>. [Accessed 23-04-2025].
- Baskaran, G et al. (Apr. 2021). “To Build A Virtual Assistant By Using Artificial Intelligence”. In: *The Open Artificial Intelligence Journal 2*, p. 1134. DOI: 10.6084/m9.figshare.14446467.
- Bojarski, Mariusz et al. (2016). “End to end learning for self-driving cars”. In: *arXiv preprint arXiv:1604.07316*.
- Dada, Emmanuel Gbenga et al. (2019). “Machine learning for email spam filtering: review, approaches and open research problems”. In: *Heliyon 5.6*.
- DeepMind (2019). “AlphaStar: Mastering the real-time strategy game StarCraft II”. In: *Google Deepmind*.
- Fan, Jianqing et al. (2020). “A theoretical analysis of deep Q-learning”. In: *Learning for dynamics and control*. PMLR, pp. 486–489.
- Giordano, Brandon (2024). *AI Navigation for Unmanned Vessels in GPS-Denied Environments*. URL: <https://www.decadefense.com/post/ai-dead-reckoning-precise-navigation-for-gps-denied-navy-unmanned-surface-vessels> (visited on 12/10/2024).
- Hannun, Awni et al. (2014). “Deep speech: Scaling up end-to-end speech recognition”. In: *arXiv preprint arXiv:1412.5567*.
- IEEE (1998). *IEEE Standard for Distributed Interactive Simulation - Application Protocols*. 1278.1-2012. Annex E.6. SISO Standards Activity Committee. 3 Park Avenue, New York NY 10016-5997, USA.
- Kapturovski, Steven et al. (2019). “Recurrent Experience Replay in Distributed Reinforcement Learning”. In: *International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=r1lyTjAqYX>.
- Kerbusch, Philip, Bas Keijser, and Selmar Smit (2018). “Roles of AI and simulation for military decision making”. In: *STO Meeting Proceedings MP-IST-160*. NATO.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2017). “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM 60.6*, pp. 84–90.
- Kumaran, Neil (2019). *Spam does not bring us joy—ridding Gmail of 100 million more spam messages with TensorFlow — Google Workspace Blog — workspace.google.com*. <https://workspace.google.com/blog/product-announcements/ridding-gmail-of-100-million-more-spam-messages-with-tensorflow>. [Accessed 28-04-2025].
- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (2008). “Isolation forest”. In: *2008 eighth IEEE international conference on data mining*. IEEE, pp. 413–422.
- MarineTraffic (2007). *MarineTraffic: Global Ship Tracking Intelligence — AIS Marine Traffic — marinetraffic.com*. <https://www.marinetraffic.com/en/ais/home/>. [Accessed 23-04-2025].
- McRobie, Ted (2024). “PINNball Wizard, Conjuring Digital Twins with Physics-Informed Neural Networks”. In: *Interservice/Industry Training, Simulation and Education Conference*. National Training & Simulation Association.
- Mnih, Volodymyr et al. (2013). “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602*.
- Mnih, Volodymyr et al. (2015). “Human-level control through deep reinforcement learning”. In: *nature 518.7540*, pp. 529–533.
- Neloy, Asif Ahmed and Maxime Turgeon (2024). “A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs”. In: *Machine Learning with Applications*, p. 100572.
- Raffin, Antonin et al. (2021). “Stable-baselines3: Reliable reinforcement learning implementations”. In: *Journal of machine learning research 22.268*, pp. 1–8.
- Rosenblatt, Frank (1958). “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review 65.6*, p. 386.
- Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). “Learning representations by back-propagating errors”. In: *nature 323.6088*, pp. 533–536.
- Schäfer, Anton Maximilian (2008). “Reinforcement learning with recurrent neural networks”. PhD thesis. *Universität Osnabrück*.

- Schaul, Tom et al. (2015). "Prioritized experience replay". In: *arXiv preprint arXiv:1511.05952*.
- Su, Wenjia et al. (2024). "Adaptive decision-making with deep Q-network for heterogeneous unmanned aerial vehicle swarms in dynamic environments". In: *Computers and Electrical Engineering* 119, p. 109621.
- Torres, Jordi (2020). *Deep Q-Network (DQN)-II*. URL: <https://towardsdatascience.com/deep-q-network-dqn-ii-b6bf911b6b2c/>.
- Towers, Mark et al. (2024). "Gymnasium: A standard interface for reinforcement learning environments". In: *arXiv preprint arXiv:2407.17032*.
- Van Hasselt, Hado, Arthur Guez, and David Silver (2016). "Deep reinforcement learning with double q-learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1.