UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

BAE SYSTEMS

Future Underwater Weapon Tactical/Homing Algorithm Optimisation Using Machine Learning

Ron Larham
BAE Systems, Maritime Services

#UDT2019

UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

BAE SYSTEMS

Overview

In this presentation I wish to talk about the development of what we describe as tactics, in particular torpedo tactics.

In our context "tactics" denotes the executive function of the weapon, how it responds to the environment to overcome obstacles to achieving its objective.

#UDT2019

UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

BAE SYSTEMS

Automobile Military Underwater Systems

- Torpedoes
- Anti-Torpedo Torpedoes
- Mine disposal vehicles
- Sea-Mines
- Autonomous underwater vehicles
- …

#UDT2019

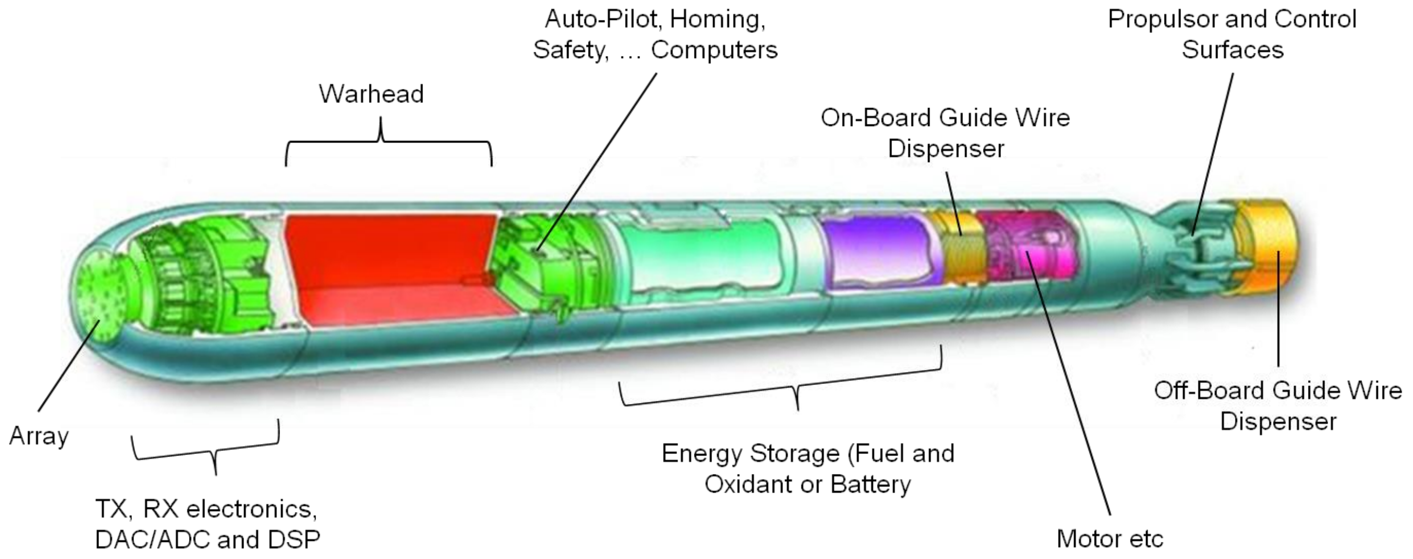
For Example

A Torpedo

A Mine Disposal Vehicle

BAE SYSTEMS

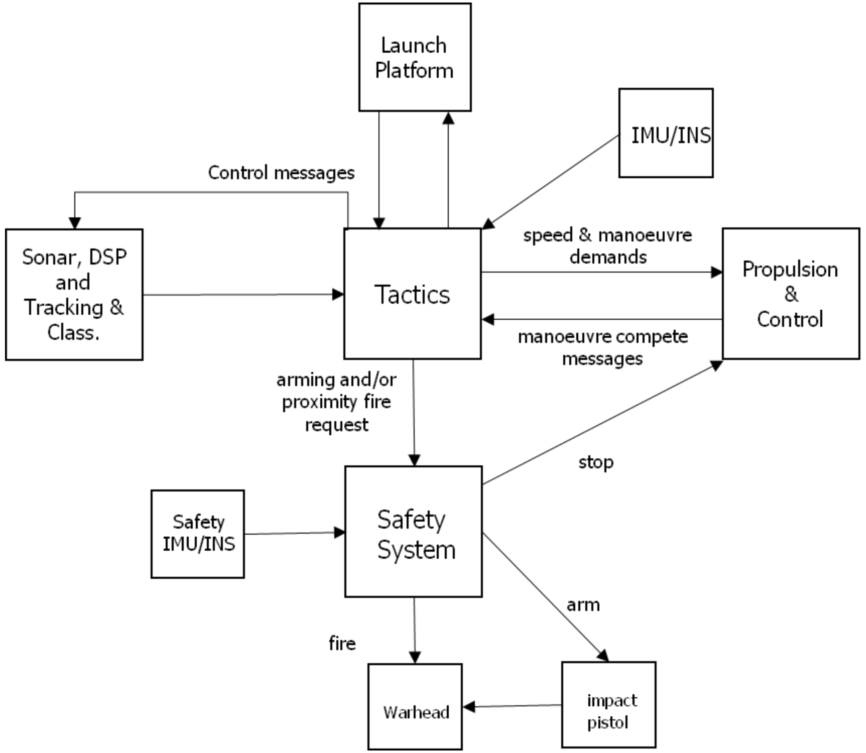
Contemporary Heavy Weight Torpedo


Auto-Pilot, Homing, Safety, … Computers
Propulsor and Control Surfaces
Warhead
On-Board Guide Wire Dispenser
Off-Board Guide Wire Dispenser
Array
Energy Storage (Fuel and Oxidant or Battery
TX, RX electronics, DAC/ADC and DSP
Motor etc

UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

BAE SYSTEMS

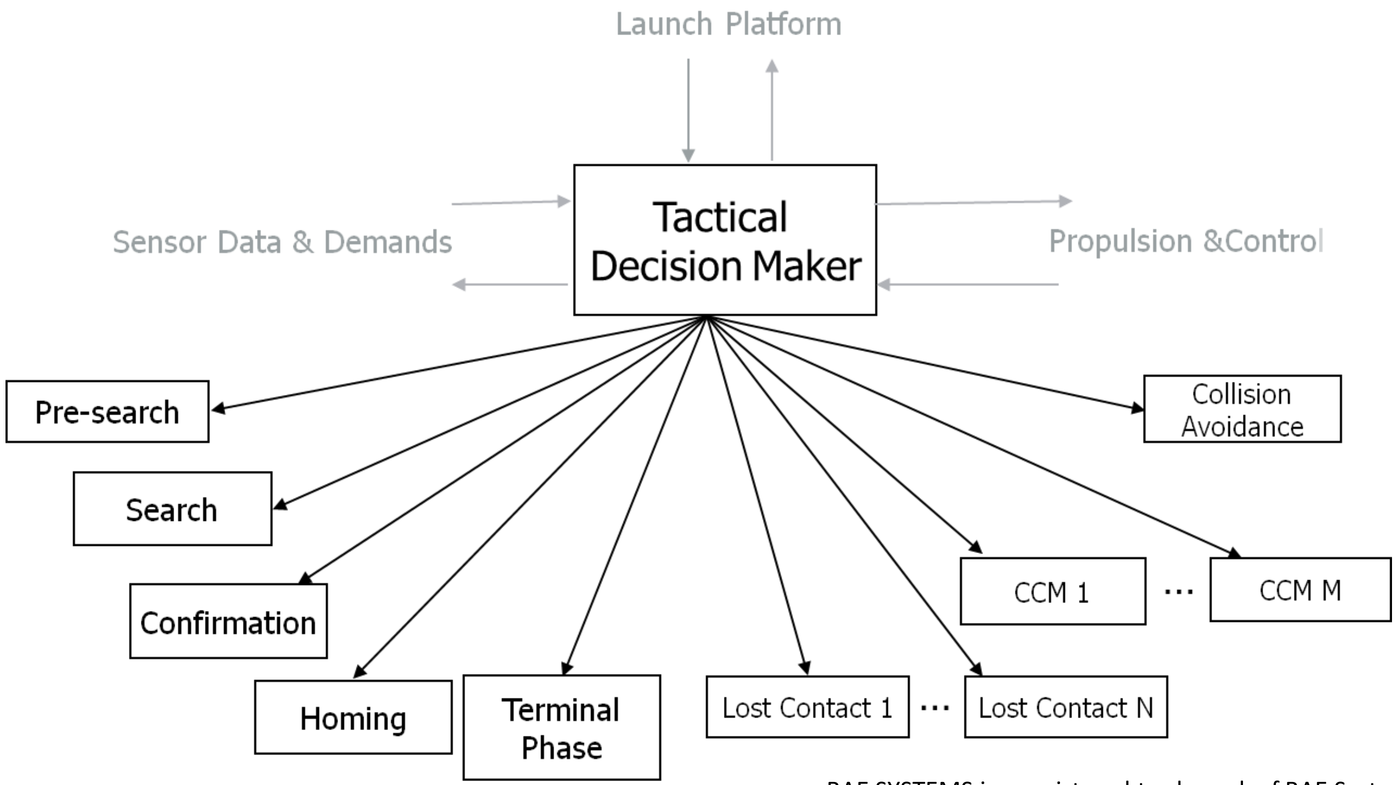
Simplified Torpedo Block Diagram (much omitted)



#UDT2019

BAE SYSTEMS

Tactics Diagram



Traditional Tactics Development Process

- Create structure and populate using design adapted from previous project, or if new project create dummy structure and stubs for individual tactics
- Create experimental TDM to call particular tactic and simulation variant to exercise the tactic. Hand fettle the tactic until judged to be satisfactory
- Repeat for other tactics
- When tactical system sufficiently populated hand fettle the TDM and tactics globally until performance adequate in the desired scenarios.
- Continue until all tactics and TDM calling rules optimised.
- Test on trials and alternative simulations, then retune if problems found

Notes: Need simulation/s which may be in development at same time as tactics design work.
"hand fettling" covers an awful lot of trial and error human intensive work.

Future Approaches to Tactics Design I

- The traditional development process is dependent to the availability of numerous trained developers, knowledgeable in the underwater warfare domain. In future that availability of the required number of developers and costs may make this impossible.
- In an ideal world we would like to specify the performance required of the system and use an automatic process to generate weapon/tactical software to meet the requirement
- The specification of performance requirements should already be available in the traditional process, as it is part of the acceptance criteria for a project.

BAE SYSTEMS

UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

Future Approaches to Tactics Design II

"IBM’s famous computer architect Frederick Brooks said that you cannot tackle increasingly large software development projects simply by throwing more people at them. It would be an immense help to the software development industry if, instead of having to manually develop every piece of a system, developers could specify the requirements of its key parts and let an evolutionary process deliver the solutions." (from Graham Kendall's piece in *The Conversation* on Evolutionary Computation, presumably quoting in turn from *The Mythical Man-Month*, 1975)

#UDT2019

Future Approaches to Tactics Design II

- Automated design can be done in principle as long as we can measure how far from the target performance a design is. In principle, if fantastically impractically, this could be done using stochastic search, possibly more practically using other global optimization heuristics like simulated annealing, deep neural networks and/or evolutionary/genetic algorithms.
- In the near term we should retain the general structure of the existing designs confining optimization to the parameters within the atomic tactics, and the tactic call rules in the TDM.
- It may be worth while to include the target behavior within the optimization loop. So the final tactical solution will be near optimal against the best tactics the target can employ.

Conventional vs ML Performance

- A field in which both highly hand optimised designs and ML designs exist and have been tried against one another is Computer Chess. The current star of ML chess systems is Alpha0 (A0) which uses a deep neural network trained in self play mode running on special purpose HW. The current champion among conventional (hand fettled) chess engines is Stockfish (Sf).

- Lela0 (L0) is an implementation of the ideas behind A0 on HW more usual HW for chess engines. This allows L0 (actually LelaChess0 Lc0) and Sf to compete on a more level playing field. In the 2018-19 Top Chess Engine Championship Sf came first and Lc0 second, and in the super-final playoff of 100 games Sf won by 1 point.

- This is probably the best comparison of the relative performance to be expected from highly optimised ML and hand-fettled systems for the same task on comparable HW. We may conclude that at present Hand-Fettled is still just competitive, but the effort required to improve it is much greater than that for ML.

BAE SYSTEMS

Comments on ML

- Leela(-Chess) Zero – is self taught DNN so there is no manual which explains the rationalisation of any particular move. GMs might explain what A0/L0 thinks as it plays but they are just rationalizing the observed behavior, there will still be completely inexplicable behaviors. We cannot even explain human behavior …

  Game 12 of the 2018 World Chess Championship Final is a nice example, All engines were showing White (Carlson) drawish except L0, which had White win in 50(?) moves. Initially all humans could not see the point of L0's moves. Eventually they claimed to understand/could explain the idea.

- Also we need to distinguish between machine learning and AI. If we have a trained system while it is possible that the system which trains itself then plays could be an AI, the part that plays **any particular** game is just an algorithm not the AI.

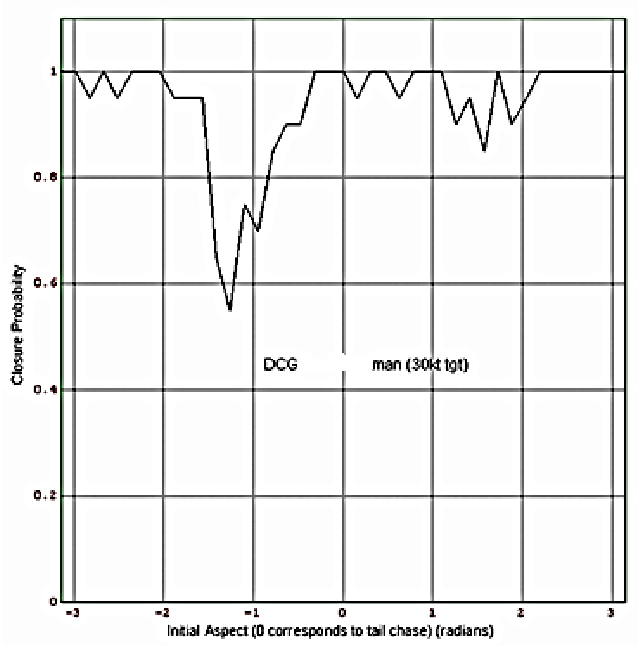
Caveat

The design of the tactics must be robust against assumptions of target signatures and tactics … our systems should do something sensible even when the target/environment etc are something or doing something not specified in the contract – we are not here just to "answer the exam question" (of Contract Acceptance Criteria)

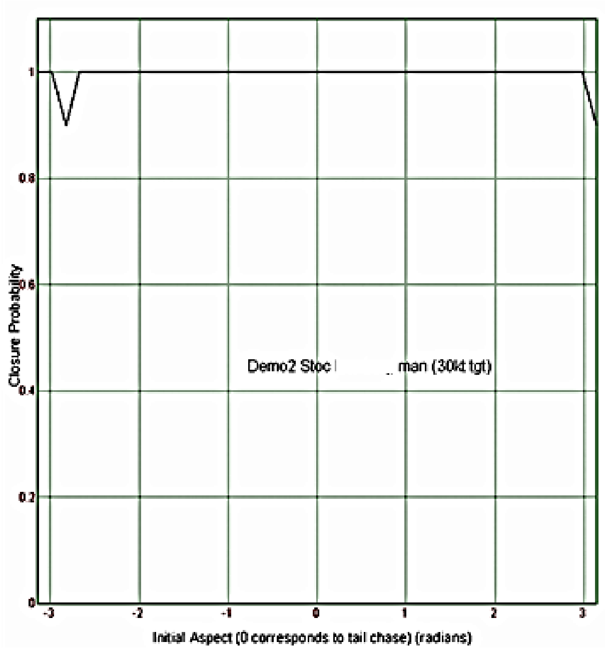
UDT Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

BAE SYSTEMS

Example 1 Terminal Homing



Hand Optimised; Closure Prb.



Machine Optimised; Closure Prb

#UDT2019

Example 2 Passive Sonar Area Search Operations

The search area was approximately 46x165 nautical miles. The searcher searches at 15kts for a total of 40 hours. The target evades at 5kts with 3 hours between course changes and reflects off of the search area boundaries.

From: Kierstead & DelBalzo, *A Genetic Algorithm Applied to Planning Search Paths in Complicated Environments*, MOR **8** #2 2003 pp45-59

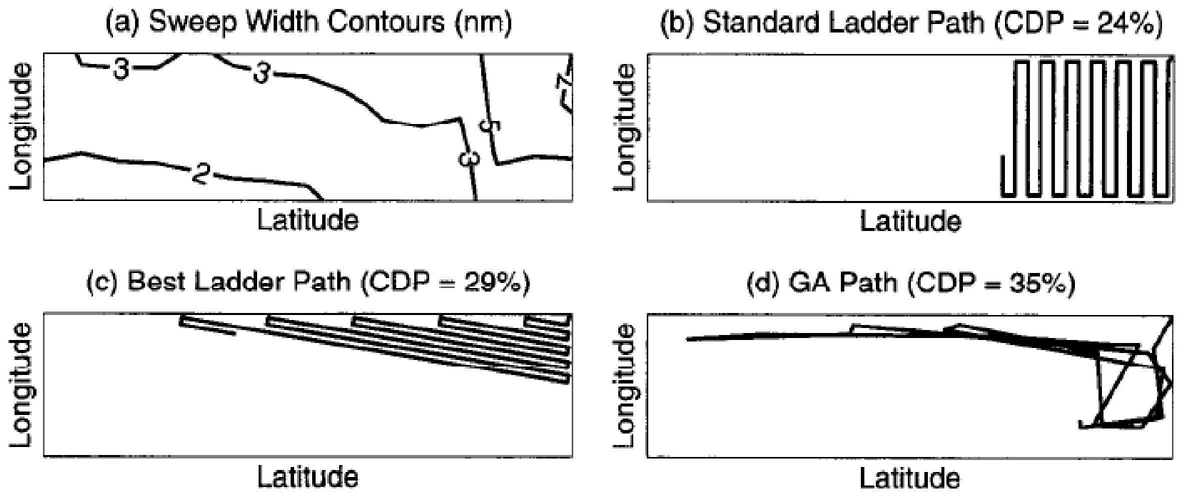


Figure 9. Search paths in a realistic environment.

Timing and Processor Requirements

An important consideration when considering automatic optimisation is the processing requirements.

For a complete set of runs against the requirement (the basic measure of effectiveness of a design) between 1000 and 2000 hours of run time on a single core of 2017 commodity level PC hardware was needed. This is effectively the processing time required for a single evaluation of the objective in the optimisation process.

While there is no theoretical limit on the number of processors/cores that could be deployed for optimisation a plausible limit is of the order of a few thousand (despite the 3×10^6 achieved in super computers by 2013). I will assume a few thousand cores for convenience. This will reduce the elapsed time for evaluation of the objective for a design to something like 0.5hrs (due to the high degree of parallelism we may assume we will realise this degree of speed-up).

The optimisation will require a very large number of runs and probably many thousands of times the time required for a full evaluation of the objective. This can be mitigated if necessary by using more selective samples in the optimisation loop.

UDT
Undersea Defence Technology
13-15 May 2019
Stockholmsmässan, Sweden

BAE SYSTEMS

Thank you

#UDT2019