Towards autonomous control of platform systems

W R van der Zwan* MSc; J van de Pol[1] PDEng; J Bergmans[1] PhD

* *RH Marine, NL*
* Corresponding Author. Email: walter.vanderzwan@rhmarine.com
[1] *TNO, NL*

Synopsis

Autonomous ship control requires a robust and resilient delivery of necessary services. The reduction of manning and increase of technology onboard results in an increased demand of automating onboard platform systems. Platform systems can be represented as a graph of interconnected components. The components are modelled as transfer functions between demand and supply. A method is proposed where this graph is traversed using straightforward algorithms to find all configurations able to deliver the required service. The different configurations are scored using a utility function to select the best one. We demonstrate this approach is feasible. Limited effort is needed to implement even if the number of components in the platform system is high where traditional automation methods struggle to deliver.

Keywords: graph, autonomous, control, command aim, platform systems, manning

1. Introduction: Towards autonomous platform control

Naval vessels have complex platform systems to support resilience to damage, support multiple warfare domains and various types of operation. The platform operator or engineering manager is faced with the task to select the best platform system configuration in support of the Command Aim. Implementing the selected configuration may require series of controls. To reach the ultimate goal of autonomous platform systems the available platform system configurations need to be determined, given a suitability score or ranked and ultimately implemented. To provide a path towards autonomous control various levels of automation and autonomy can be applied. This paper focusses on a method for generating platform system configurations and how to evaluate their suitability.

Author's Biography
Walter R. van der Zwan is a consultant and researcher at RH Marine in Schiedam, NL. He has a background in physics. He was involved in the development and testing of platform automation systems for navy ships in the first half of his career. In the second half his involvement was more on researching and developing new algorithms and cyber security.
Johan van de Pol is a systems engineer at TNO, NL. He has a background in electrical and software engineering. Throughout his career he has been involved in many different projects in the domain of defense safety and security.
Jeroen Bergmans is a researcher and system developer in the Intelligent Autonomous Systems department at TNO Defense, Security and Safety in The Hague, NL.

 https://doi.org/10.24868/10709

2. System configurations

In order to select the most suitable system configuration, we first have to generate the possible configurations given the current platform state and requirements. Subsequently, these options have to be ranked according to some scoring metric to select the most suitable configuration that can then be applied. In this section these steps are described.

2.1. Generating configurations

The platform systems can be in multiple states or configurations. Each configuration provides a performance level (e.g. minimum speed, manoeuvrability, or power). To achieve the requested output performance, components in the system require supply of resources from other components. We represent these interconnected components by a graph where each component is a node and along an edge both demand for resources and the available supplies are represented.

2.1.1. Graph logic

Each node in the graph represents a system component and edges between those nodes represent dependencies between those system components. We then can use a graph traversal to generate potential system configurations.

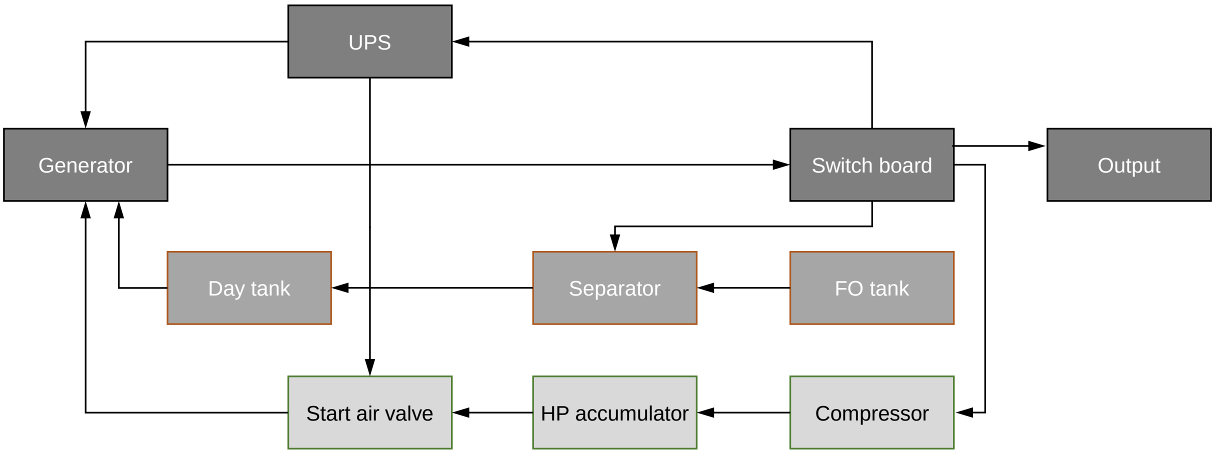

Figure 1 Typical graph of a power generator system

The figure above shows a simple system (a power generator) as a graph. Each component in the graph can have inputs and/or outputs and the configuration of each node defines the relation between the inputs and outputs. For example, the generator component in the image above has three inputs required to generate electrical power. For each of the inputs it has to be defined how (amount and/or for how long) that input influences the output of the node. E.g. how much fuel is needed to generate a certain amount of output power? The dependency between the inputs and the output is captured in a node model which models the component's functionality on a high abstraction level.

Capturing the system in such a graph evidently requires a configuration effort, but this configuration is local to each component and not dependent on other components. This makes for easier configuration than when one considers the system as a whole. Also, when making changes to the system (i.e. addition and/or removal of nodes), the graph allows for locally updating the configuration.

2.1.2. Algorithm

The developed algorithm will try to generate system configuration options based on an output request. Each component in the graph defines the relation between its inputs and outputs and based on those relations, each component should be able to answer the following basic questions:

- Based on a requested output demand, what should I demand from my inputs?
- Based on the provided input supply, what can I supply on my output?

 https://doi.org/10.24868/10709

Here, a supply answer comprises both the amount and the duration of the provided quantity. When each component in the graph is capable of answering the questions above, then we can apply a simple algorithm to generate system configurations. That simple algorithm is outlined below.

```
set demand on output outlet
until no more changes or output duration threshold:
  for each component:
    update input demands based on output demand
    update output supply based on input supplies (including durations)
```

Figure 2 Typical graph algorithm

Above algorithm can be performed more efficient if not all components are visited blindly, but instead components required to check their input or output are scheduled for a re-calculation. This would also suit an agent-based implementation where each component (or agent) executes when triggered by a changed input or output. However, the current proof-of-concept implementation follows the more straight forward algorithm as described above.

One thing to note is that components may provide multiple options which result in the same output. An example of such situation is propulsion power which may be provided by 2 different engines. In such cases, the evaluation may fork the calculation resulting in multiple solutions being generated. In a similar way, it can also occur that a component cannot deliver the requested supply (e.g. output demand is larger than its remaining capability) in which case no solution can be provided.

2.1.3. Simple Example calculation

This section will provide a very simple sample calculation explaining the algorithm. Consider the following (very simple) graph:

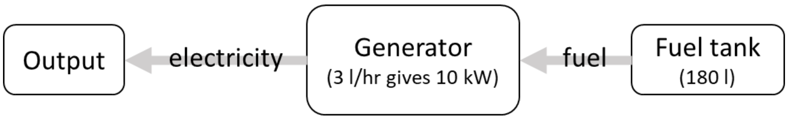

Figure 3 Example graph for a power generator

This graph defines a "Generator" which takes fuel from a buffer (the "Fuel tank") and transforms that into electricity. The table below shows the steps which are taken by the algorithm to come to an answer. Updates are highlighted in black and row six shows the final result.

Table 1 Logic steps for the example graph, for each step updated values are highlighted

Step	***Output***		***Generator***		***Fuel tank***
	supply	**demand**	**supply**	**demand**	**supply**
1. Operator request		10 kW			
2. Update node		10 kW		3 ltr/hr	
3. Update source		10 kW		3 ltr/hr	3 ltr/hr for 60 hr
4. Update node		10 kW	10 kW for 60 hr	3 ltr/hr	3 ltr/hr for 60 hr
5. Update output	10 kW for 60 hr	10 kW	10 kW for 60 hr	3 ltr/hr	3 ltr/hr for 60 hr
6. No more updates	10 kW for 60 hr				

This simple example shows clearly how a demand 'travels' through the graph and eventually comes back to provide a supply on the output.

2.1.4. Component model

The component model provides a model for system components and should define the relations between its in- and outputs: A demand for output has to be translated into demands for supply from input components. Conversely, supplies on its inputs have to be translated to determine the output supplied by the component itself.

 https://doi.org/10.24868/10709

One of the goals is to reduce the amount of configuration required. It is not desired that each and every component type has its own implementation of a model as this would require much additional development work. Therefore a very generic component is used which is used to model most components. A schematic representation of the generic component is shown below.

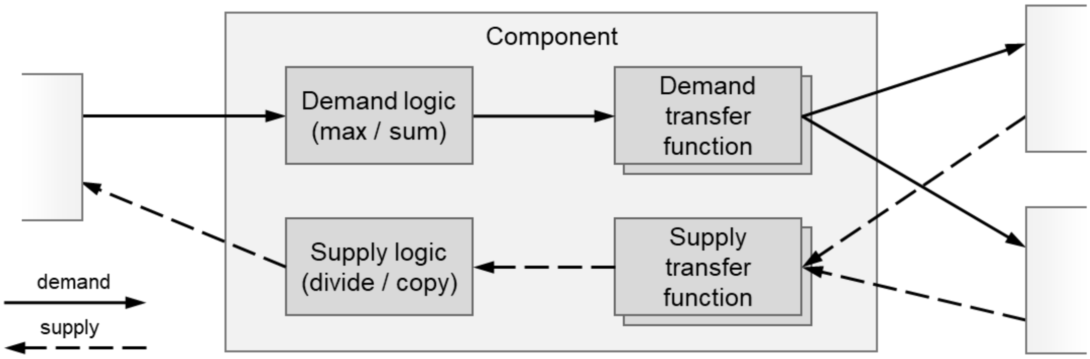

Figure 4 Generic component model

The *demand logic* defines how demands posed to the component are to be combined in the total demand. The *demand transfer functions* then translate this demand into outgoing demands (for the various types of needed supplies like fuel, electrical power, etc.) to the upstream components in the functional chain. When a supply is updated, the *supply transfer functions* translate these to the resulting available supply of the component and finally the *supply logic* determines how the supply is distributed over the demanding downstream components.

This generic component model is suitable for most system components, only some components need some specialization, most notably buffers (fuel tanks, batteries) because of their internal storage capacity.

For some components there are multiple solutions to fulfil the demand. A component could for example choose from several different suppliers (e.g. different engines in the propulsion chain) or a component might have different operational modes (e.g. brake applied or released). In this case the algorithm generates and evaluates each solution separately. This results in multiple overall system configurations. In the next section the selection of the most suitable configuration given current operational criteria is discussed.

2.2. Scoring configurations

Selecting the best platform system configuration requires some metric to rank the configurations. For naval ships this metric must be related to the Command Aim. The ranking itself is performed using a utility function which is configured based on the active Command Aim. The Command Aim and the use of utility functions are explained in the sections below.

2.2.1. Command Aim

The Command Aim is a “precise and clear statement of the current tactical priorities including the tactical aim, the priority threat and manoeuvrability. The command aim can change in the fluid scenario of naval warfare”. The managers and operators apply the command aim to make decisions to best contribute to the command aim considering aspects like readiness, manoeuvrability, warfare priorities, special duties, etc. The Command Aim provides guidance to set desired overall system performance and directives or parameters that can be applied in a utility function.

2.2.2. Demand parameters

The main input of the algorithm is a demand (desired amount) for each of the output nodes of the system. In the example propulsion-steering system this comprises of: 1. The minimum achievable propulsion speed, 2. The minimum rate-of-turn and 3. The level of stabilization. The graph algorithm can generate all possible component configurations that satisfy these output demands.

2.2.3. Utility function

The utility function is a weighted sum of various aspects of a single configuration state. Examples of these aspects are efficiency, current health, acoustic and heat signatures, etc. In the future the information extracted from

 https://doi.org/10.24868/10709

command aim is expected to be automated. The information extraction provides the "directives" or "input parameters" for the required services and utility function. At present this is manual input.

In the current implementation we have included the aspects below in the utility metric:

- *Efficiency*: Efficiency is measured by assessing the total fuel use for each of the generated configurations.
- *Duration*: The maximum duration determines how long the system can sustain the supplied output given the configuration. Because all supply calculations in our method explicitly yield both amount and possible duration, the duration is a direct result of the algorithm.
- *Health*: From the platform a qualitative health value (OK, caution, threat, and not OK) is provided for all components. By aggregating the health values of the components used by the configuration an overall health score is determined.
- *Robustness*: An aggregated robustness score can be derived by assigning a qualitative robustness value to every component. Just like the health score this results in a qualitative aspect score represented by a number.
- *Signatures*: For signatures (*Acoustic, Infrared, Pressure,* etc.) it is in principle possible to quantitatively assess a generated configuration. Signature management includes models to estimate own signature, signature propagation, and a threat model to determine whether the signature level for a platform configuration is acceptable or not. In the future these calculations have to be delegated to separate signature management systems. For the current demonstrator implementation we again use a qualitative lookup table to get a score that can be used to rank configurations on signature.

Finally all aspect scores are normalized and summed by weight. Using the weights the relative importance of the aspects can be adjusted. The result is a list of configurations. The presentation order is:

1. Available configurations complying with the required service levels sorted by utility score;
2. Available configurations not complying with required service levels sorted by utility score;
3. Unavailable configurations sorted by utility score (or estimated time back on line).

3. Proof of concept

A limited Proof of Concept (PoC) was developed to demonstrate the feasibility of generating and ranking configurations. The manoeuvrability service of a frigate was selected as a case study. It includes multiple platform systems and interactions. The Zr. Ms. Zeven Provinciën air defence and command frigate (1) was selected for the PoC. Publicly available information was applied.

A graph of the mobility system is defined (see Figure 5). The propulsion system is a Combined Diesel Or Gas turbine (CODOG) system for each of the two propellers. The fuel supply system was included to estimate the endurance of a configuration. A propeller can be driven by the diesel engine (cruise) or the gas turbine (boost mode). Also, the propellers can be trailing if driven by the water flow. If the brake is engaged, the shaft is blocked (no propeller revolutions). The steering system consists of two hydraulic power units per rudder. One unit is sufficient to control the rudder, but both units are needed for rudder roll stabilization. The turning circle of the ship is related to the effectiveness of the rudders. If the water flow of the rudders is disturbed (i.e. the shaft has the brake engaged) the rudder has low effectiveness. The level of rudder roll stabilization depends on the available rudder rate and rudder effectiveness.

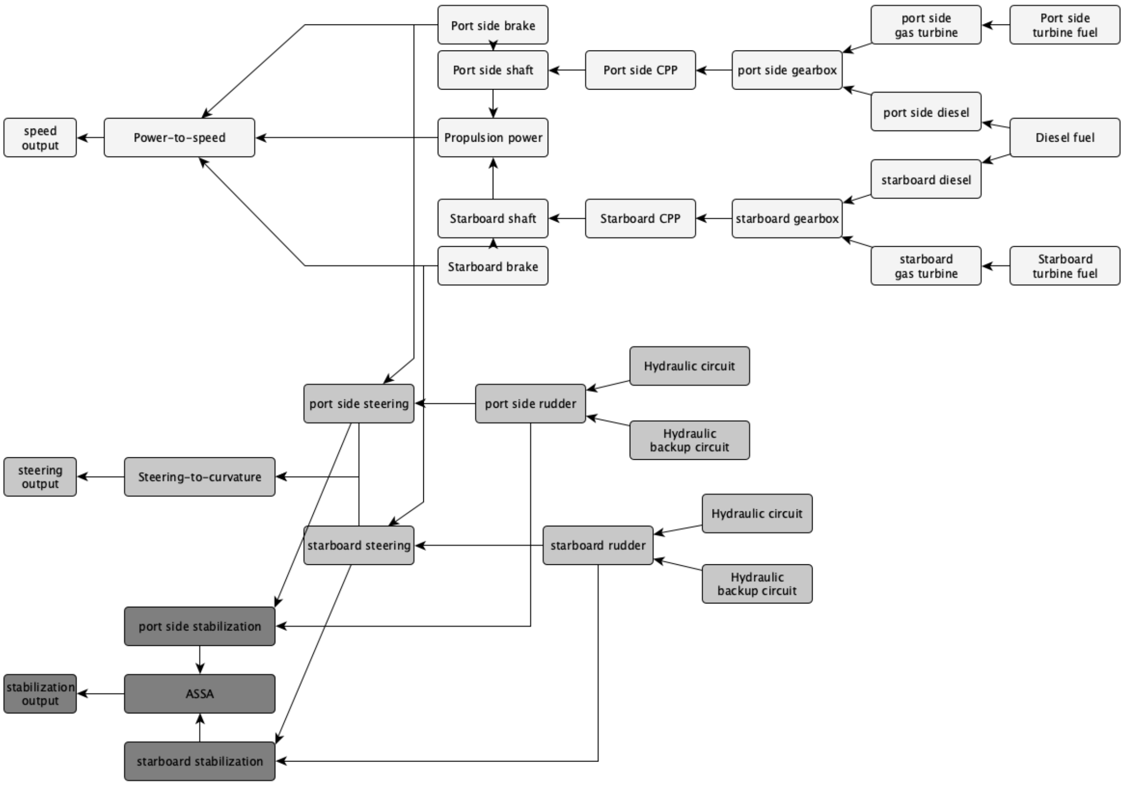

Figure 5 Graph of the mobility system.

The utility function to score a configuration included the acoustic and infrared signatures. The acoustic signature is influenced by the type and number of engines online and the ship's speed. Gas turbines have lower vibration levels than diesel engines. The propellers start to suffer from cavitation around 18 to 20 knots. A blocked propeller generates noise due to the wake turbulence. A trailing propeller produces a lot less noise. The infrared signature of the exhaust gasses is related to the type and number of engines on line and the actual power of the engines. High power relates to higher exhaust gas temperatures and more exhaust gasses. This increases the infrared signature.

The utility function is strongly affected by the state of readiness. We applied readiness states 1 to 5 as defined in Fundamentals of Maritime Operations (2) page 186:

- Degree of readiness 5 is in force when a ship is at anchor or in a safe harbour. Only a small part of the crew will be on duty for watch, security and initial emergency response.
- Degree of readiness 4 applies to a ship at sea that is in principle doing nothing more than safe navigation in open water, for example during a transit. A small section of the crew will be on duty for safe navigation, regular safety and initial emergency response.
- Degree of readiness 3 applies in the event of increased activity or heightened risk. Personnel and equipment needed to perform the required activity or to avert an immediate threat or danger are directly available. This normally means that a third of the crew is on duty (three-section watch system)
- Degree of readiness 2 provides the highest possible degree of readiness that can be sustained over a prolonged period (two or three weeks). Generally speaking, this means that half the crew is on duty (two-section watch) and that as many systems as possible are available immediately or at extremely short notice.
- Degree of readiness 1 ('battle stations') means maximum readiness. The entire crew is on post and immediate employment of all systems and functionalities is possible. This degree of readiness can only be sustained for a limited period.

We assumed the following aspects are relevant:

- Robustness or resilience (ability to sustain damage without losing functionality);
- Efficiency (cost to operate related to consumables and maintenance);
- Signature (level of own ship signature).

The degree of readiness influences the weight of these aspects. At low readiness states the efficiency is more important than robustness or signature. For the proof of concept we applied the following weight factors:

Table 2 Weight factors for readiness levels

Readiness	Robustness	Efficiency	Signature
1	100	10	100
2	80	20	100
3	50	50	50
4	20	100	10
5	10	100	10

The signature weight factor needs to be distributed over the various signatures. Depending on the warfare priorities the relevance of each signature aspect differs. The warfare domains are Anti Air Warfare (AAW), Anti-Surface Warfare (ASuW), Anti-Submarine Warfare (ASW), Mine Warfare (MW) and Electronic Warfare (EW). We applied the following distribution:

Table 3 weight of signatures to warfare domains

Signatures	AAW	ASuW	ASW	MW	EW
Radar Cross Section	0.7	0.7			0.3
Electro Magnetic	0.1	0.1			0.7
Infrared	0.2	0.2			
Acoustic			1.0	0.3	
Magnetic				0.4	
Electric				0.1	
Pressure				0.2	

The infrared and acoustic signatures were implemented in the Proof of Concept. The acoustic signature came into play for ASW and MW, the infrared signature for AAW and ASuW. The command aim provides the warfare priorities. With the weight factors per warfare domain the relevance of each signature is provided. The signature weight factor is distributed over the signatures in accordance with the values of Table 3.

The Proof of Concept was an application to generate all possible configurations for the mobility service. A Human Machine Interface was created to enter the relevant derivatives of the Command Aim. The health status of the components in the system graph can be manipulated. The list of possible configurations was sorted by compliance with the mobility requirements (speed, turning circle and stability) and overall utility score. The selected configuration from the list of possible configurations is presented on top of the actual configuration. The operator can activate the newly selected configuration.

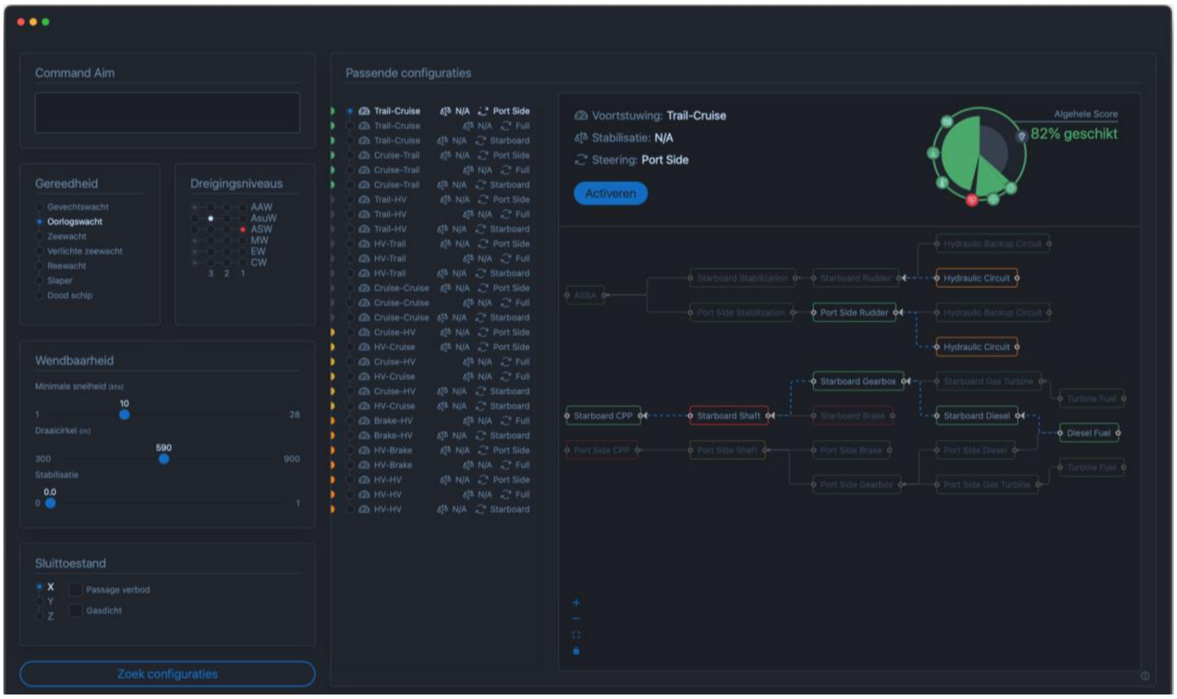

Figure 6 HMI of the Proof of Concept application

The readiness is set under the heading “Gereedheid” (“Oorlogswacht” is readiness state 2). The warfare domains are set under “Dreigingsniveaus”. It is possible to set up to three warfare domains. The minimum achievable speed and turn circle are set under “Minimale snelheid” and “Draaicirkel”. The stabilization is set under “Stabilisatie”.

Figure 7 Detail of HMI to set command aim derivatives

 https://doi.org/10.24868/10709

Changing the Command Aim derivatives results in a new sorted list of configurations. The operator can select any configuration from the list for evaluation. The best configuration should be on top of the list if the utility function has been configured correctly.

The Proof of Concept supported operators to select a system configuration. For the PoC all possible configurations were included, even if these configurations can't provide the required mobility services.

4. Conclusions

The study demonstrated that it is possible to represent platform systems as a graph of functional dependencies between components in such a way that all possible system configurations can be evaluated using basic graph algorithms. The study demonstrated that it is possible to define a metric to evaluate these configurations in a useful way. The proof of concept demonstrated how operators can be supported in selecting system configurations in complex situations. The effort needed to define the graph and relationships is limited. The graph can be modified without changing the application. This reduces the maintenance effort in case of modifications to the platform systems.

The next steps are to extend the functionality to other services and to include additional parts of the platform systems. Automatic or autonomous control of the platform systems requires controlled state transitions and closed loop control of systems.

Acknowledgements

The study "Platform Automation @ Functional Level" was funded by the Netherlands Defence Material Organisation. The authors would like to thank DMO for providing valuable feedback and comments during the study.

References

1. Luchtverdedigings- en commandofregat (LCF) | Materieel | Defensie.nl. *www.defensie.nl.* [Online] 17 06 2022. https://www.defensie.nl/onderwerpen/materieel/schepen/luchtverdedigings--en-commandofregatten-lcf.

2. Royal Netherlands Navy Directorate of Operations Maritime Warfare Centre. Fundamentals of Maritime Operations, Netherlands maritme military doctrine. *defensie.nl.* [Online] 13 February 2014. [Cited: 27 June 2022.] https://english.defensie.nl/downloads/publications/2014/02/13/netherlands-maritime-military-doctrine.