Heterogeneous swarming for collaborative combat using Multi-agent Deep Reinforcement Learning.

Jacques Bois, Axel Puig, Loïc Rullière, Yonatan Teboul, Morgan Ossola, Alexandre Kotenkoff, Mathias Formoso

Heterogeneous Swarming for Collaborative Combat using Multi-Agent Deep Reinforcement Learning

Jacques Bois
MBDA
CentraleSupélec, Uni. Paris-Saclay
Paris, France
jacques.bois@mbda-systems.com

Axel Puig
MBDA
CentraleSupélec, Uni. Paris-Saclay
Paris, France
axel.puig@mbda-systems.com

Loïc Rullière
MBDA
Paris, France
loic.rulliere@mbda-systems.com

Yonatan Teboul
MBDA
Paris, France
yonatan.teboul@mbda-systems.com

Morgan Ossola
MBDA
Paris, France
morgan.ossola@mbda-systems.com

Alexandre Kotenkoff
MBDA
Paris, France
alexandre.kotenkoff@mbda-systems.com

Mathias Formoso
MBDA
Paris, France
mathias.formoso@mbda-systems.com

***Abstract* — Development of future weapon systems heavily relies on simulations where several agents interact in an environment. In such an environment, optimised decisions are crucial to leverage the capability of collaborative systems in a congested, cluttered, contested, connected, constrained [1] battlespace. In this work we present a simplified, yet complex, scenario derived from a notional air-to-surface mission featuring a set of heterogeneous effectors collaborating over a communication network in order to detect and engage a number of defended, relocatable ground targets. We propose a method based on a multi-agent, distributed version of SAC (Soft-Actor-Critic) and we highlight three benefits of our method: 1- Reinforcement Learning (RL) outperforms greedy and semi-random algorithms on a set of operational metrics. 2- Collaboration improves the global performance of the teamed agents. 3- Curriculum Learning is central to improve and speed-up the training of the agents.**

Keywords — Multi-Agent, Reinforcement Learning, Distributed Systems, Collaborative Decision Making. (*key words****)***

I. INTRODUCTION

Reinforcement Learning is a method for optimising the actions of an agent in an environment in order to maximise a reward function. The agent thus learns by trial and error by interacting with its environment. This method of machine learning gained momentum at the beginning of the 21st century, especially in automatics [2]. In the past few years, the use of deep neural networks made it possible to solve complex nonlinear problems such as Atari games [3] and more recently the game of Go [4].

Recent research in Reinforcement Learning focused on multi-agent problems where the reward no longer depends on the actions of a single agent but on its action combined with that of the other agents. This method made it possible to exhibit collaborative behaviours with a better performance than single agent RL [5] on a different set of problems, ranging from traffic regulation [6] to the coordination of drone fleets [7].

Armed forces are increasingly interested in collaborative combat, which requires the mastery of many new technological subjects at the crossroads of technology, tactics, doctrines and ethics. In particular, mastering fleet communications [8], formation flying of autonomous fleets [9], as well as tactical cooperation & coordination between air assets, whether manned or unmanned [10][11]. In addition to these topics, collaborative combat also increases the number of possible strategies for a given mission. Several works propose approaches to establish these strategies, with control laws derived from automatics [12], approaching the subject as an optimisation problem [13], or by getting inspired from animal strategies [14]. Finally, some propose more complex approaches combining different types of algorithms [15].

The present work proposes to approach the problem of defining collaborative combat tactics as a reinforcement learning problem. Several works highlighted the relevance of such algorithms for this type of problems [16]. RL is efficient to manage complex situations while bringing out innovative strategies, therefore its use in the context of collaborative combat seems to be increasingly relevant.

II. FRAMEWORK AND ENVIRONMENT

A. Markov Games framework

We consider the framework of Markov Games (Littman, 1994). It extends the framework of Markov Decision Process, to multi-agent problems.

We define:

- A set of states S,
- Action sets for each of N agents $A_1, \dots, A_N$,
- A state transition function which represents the probability distribution over possible next states, given the current state and actions $T: S \times A_1 \times \dots \times A_N \rightarrow P(S)$,

- A reward function for each agents that depends on the global state and actions of all agents $R_i: S \times A_1 \times \ldots \times A_N \rightarrow \mathbb{R}$.
- The observations of each agents, which contain partial information from the global state $s \in S$: $o_1, \ldots, o_N$
- The policy of each agent $\pi_i: O_i \rightarrow P(A_i)$ which returns a distribution over the set of actions given an observation,
- γ is a discount factor that determines how much immediate rewards are favoured over long-term gain.

In this framework, the objective of the ***i*th** agent is to find a policy that maximises their expected discounted rewards:

$$\pi_i = \underset{\pi}{\operatorname{argmax}} J_i(\pi)$$

$$J_i(\pi_i) = \mathbb{E}_{a_1 \sim \pi_1, \ldots, a_N \sim \pi_N, s \sim T}\left[\sum_{t=0}^{\infty} \gamma^t r_{it}(s_t, a_{1t}, \ldots, a_{Nt})\right]$$

B. *Environment and model*

Our environment contains two Surface Based Air-Defence (SBAD) systems that can move, or engage an asset and then fire. A Probability of Kill (Pk) is associated with the firing. The two systems have different ranges: short-range (SACP) and medium-range (SAMP). These SBAD systems follow a loop with the following steps:

- Deployment: preparation for engagement (10 min): no movement, no engagement;
- Engagement if possible
- Dismantling: preparation for motion (5 min): no movement, no engagement
- Relocation: random motion for 10 minutes at 5kph.

8 Multiple Rocket Launcher (MRL) systems can fire their surface-to-surface rounds and follow a ”shoot and scout” tactic:

- Fire (5 minutes) at a frequency of 4 rounds per minute;
- Movement (10 minutes): they move at 15kph in random direction and stay in pairs. They stay in the area protected by the air defence systems. The goal is to optimise the policy of the assets, while the behaviour of other elements are coded to be representative of their real operation, to minimise the number of rounds fired by the MRL systems (which therefore encourages their neutralisation).

The environment also contains a heterogeneous group of air assets, each one controlled by a reinforcement learning agent (blue team):

- 4 Observation assets (DRIL): they detect, recognise, identify and locate the elements in their field of view. This information is perfectly and instantaneously transmitted among all assets;
- 2 Jamming assets (EW): they can either activate the jamming mode, to reduce the Pk (probability of kill) of SBAD systems located in the jamming field of the asset, or the observation mode (ELS - Emitter Location System), to detect, identify and locate the elements in their field of view, with different performance parameters from DRIL assets;
- 8 Attacking assets (WPN): they can neutralise targets, either SBAD or MRL systems, by direct impact. They stay in a waiting area until they take the action “attack mode” (see below).

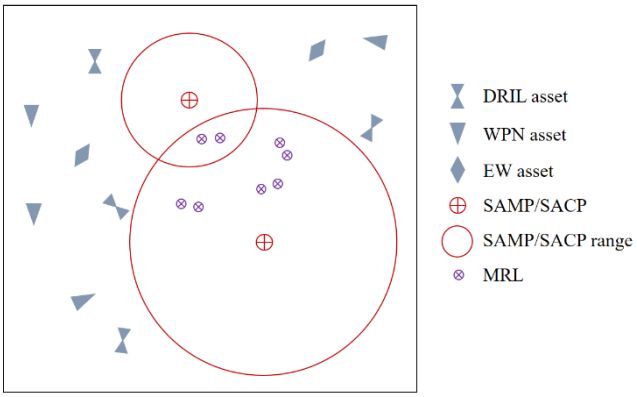

Fig. 1. Schematic representation of the simulated battlefield and its components

C. *State and action spaces*

The state space is the same for each asset. It contains the position and velocity of itself and of the other assets, the types and actions of the other assets, the position, heading and type of detected SBAD and MRL systems.

The action space differs depending on the type of assets. The action space of all the assets is composed of the orientation of the velocity vector. The action space of the WPN assets also contains the activation of an attack mode, and that of EW assets contains a choice of a mode which can be observation (ELS) or jamming.

D. *Multi- Agent Soft Actor-Critic (MASAC)*

Actor-Critic method [17] is a modified version of policy gradients methods [18][19] which reduces the high variance problem encountered in the original works.

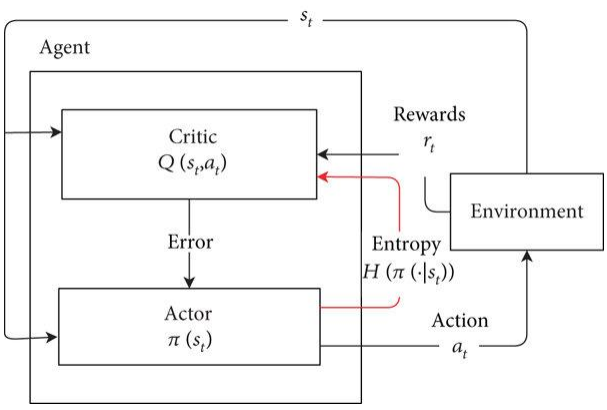

Fig. 2. Soft actor-critic algorithm

The algorithm has two parts. An actor samples actions in the environment, while a critic computes the gradient based on the temporal difference error. Decoupling actions sampling and policy updates allows to average experience and to reduce variance.

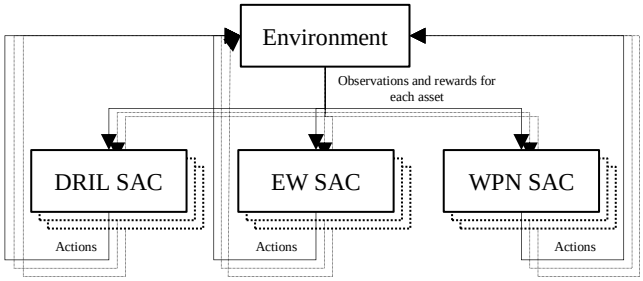

Fig. 3. MASAC: A SAC for each type of asset, replicated for each of the assets.

Three different actor-critic models are considered, one for each type of asset. They are then replicated according to the number of assets, since all the assets of a same type have the same role and can be exchanged. Every asset gets its own observations and rewards, so even if they follow the same policy, they take different actions.

This method, we call Multi-Agent Soft Actor Critic (MASAC), has several advantages. First, since all the assets of a same type learn the same policy, they can be swapped without consequences. Moreover, this solution is decentralised, therefore the loss of an asset should have a limited impact on the performance.

Every time a simulation is run, the observations, actions and rewards of all the assets of the same type are concatenated to train the actor and critic neural networks corresponding to this type of asset.

E. A distributed reward function

Table I below presents the different terms in the reward function.

In particular, some terms are distributed either individually to the involved asset, either partially to the asset group (DRIL, EW, WPN), or globally to the entire swarm. These distributed terms in the reward function may foster collaborative swarm actions.

Most of the terms are continuous, in order to tune the right balance between exploration time and providing arbitrary, subjective information to agents through their reward functions. However, some terms are discrete, in order to try and highlight a remarkable action led by the swarm of assets.

Reward terms are arbitrarily given an importance level according to their meaning. Table I presents these importance levels, which are scaled through fine-tuned coefficients.

Finally, mathematical functions are defined in order to quantify the terms presented in the table below. For instance, the anticollision reward function depends on the distance between the assets ; the detection reward function depends on the number of red assets detected by the blue team at each step ; the SBAD system neutralisation function depends on the distance between WPN assets and SBAD systems, it also has a discrete component depending on the effective neutralisation of SBAD systems ; the hazard reward function depends on the time spent by the blue assets in the interception domains of SBAD systems.

TABLE I. DECSRIPTION OF THE MASAC-DISTRIB REWARD FUCTION

Justification	**Nature**	**Distribution**	**Temporal nature**	**Importance**	**Order of appearance[a]**		
					DRIL	**EW**	**WPN**
Anticollision	Penalty	Individual	Continuous	Very high	1	1	1
Detection/Recognition/ Identification/Location	Reward	Partial	Continuous	Medium	2	2	-
Jamming	Reward	Partial	Continuous	Low	-	2	-
MRL system neutralisation	Reward	Global	Continuous, punctual	Very high	3	3	2
SBAD system neutralisation	Reward	Global	Continuous, punctual	High	3	3	2
Blue asset attrition	Penalty	Global	Punctual	High	4	4	4
Risk level (toward attrition)	Penalty	Individual	Continuous	Low	4	4	4

[a.] See subsection III-C.

III. EXPERIMENTAL METHOD FOR THE DEVELOPMENT OF COLLABORATIVE STRATEGIES

A. Evaluation metrics

Two **operational evaluation** metrics are defined. They reflect the operational objectives set by the mission commander of the blue team. These metrics are representative of a notional counter-battery scenario. They are, in order of priority:

1. **Minimise the number of rounds fired by the opposing weapons systems** (MRL systems). We measure the percentage reduction of the number of rounds fired compared to the maximum.
2. **Reduce the effectiveness of SBAD systems** to facilitate a follow-on attack. We measure the percentage reduction, averaged over 1 episode, of the number of operable surface-based air defence (SBAD) systems. An "operable SBAD" is neither jammed nor neutralised.
3. **Minimise attrition in the swarm of assets.** We measure the percentage of surviving assets at the end of the mission.

The following **technical metric** is averaged on a batch of episodes and allows to monitor the learning process evaluation.

- Variations in the reward function: average reward and standard deviations achieved on a mission, all smoothed over 10 missions.

B. Baseline methods

To benchmark our method, we used the following baselines:

- **ALEA.** Semi-random algorithm: decisions are made by generating random actions uniformly and independently, and applying a correction on the three components of the asset velocity to smooth the trajectory and to keep the mobiles inside the playground. For discrete actions, a threshold is applied. This algorithm ensure that the problem is not trivial to solve.
- **GREEDY.** Decisions are made by maximising the next step reward. This myopic strategy should give poor results if the reward function does not reduce the problem to an easy optimisation problem.
- **MASAC.** Algorithm defined in Section II, but without the collaborative incentives: all terms of the reward function are distributed individually. The goal is to measure the contribution of the partial or global distribution of the terms of the reward function.

In contrast to MASAC, **MASAC-Distrib,** our algorithm, uses a distributed reward function as described in Table I.

C. Curriculum learning

The definition of a training strategy reduces the complexity of the problem and improves the convergence and the results of the RL agent, while controlling the effectiveness of the learning [20]. It consists in a set of elementary functions, increasing by complexity, compounding a curriculum [21]. In our case, the task of each training phases are encompassing previous ones [22]. The 5 training phases are summarised below:

1. **ANTICOLLISION.** In the reward function, only the anticollision term is present (cf Table I). The goal of this phase is to teach the assets to fly and avoid collisions with each other and with the ground.
2. **SPECIALISATION.** The "specialisation" terms of the reward function (observation, jamming, attack) are introduced in Table I. In this phase only, WPN assets are omniscient about the positions of enemy assets. Assets learn how to neutralise targets without depending on the positions provided by other assets: the goal is to reduce the exploration needed to neutralise targets. The overall objective of this phase is to teach the assets their "expertise", assuming the others can do theirs.
3. **COLLABORATION.** In this phase, attacking assets depends on the information provided by the other groups of assets. In the case of the MASAC-Distrib algorithm, the reward function is modified to distribute rewards globally or partially according to Table I. The goal of this phase is to teach the assets to collaborate with each other. WPN assets are not omniscient anymore and rely on information provided by DRIL and EW assets.
4. **SURVIVE.** In this phase, risk and attrition penalties are added to the reward function. The goal is to encourage assets to consider the risk of surface-based air defence (SBAD) threats.
5. **ADAPTATION.** Finally the adaptation phase consists in training the agents to adapt to variations of ground mobiles initial positions. In order to progressively increase the complexity of the scenario, agents were trained on close to fixed initial positions. Prior to this phase, the initial positions of the mobiles on the ground varied by a maximum of 15 km between two missions. During this training phase, these initial positions varied by a maximum of 80 km between two missions, i.e. over the entire playground. This prevents the agent to memorise the zones in which enemy mobiles are to be found.

At each phase of curriculum learning, the weights of the neural networks are initialised using the weights trained during the previous phase. Hyperparameters are reinitialised and then monitored thanks to the evolution of the reward function and Q-values.

IV. RESULTS AND ANALYSIS

A. Experimental results

The plots of the rewards and Q-values obtained by MASAC and MASAC-Distrib algorithms showed the scientists the average rewards and cumulated Q-values obtained for each agent over the adapted number of missions. The training was systematically stopped when the reward function and the Q-values converged.

In the remaining parts of this section, we present the radar chart of the 3 operational metrics defined in subsection III A.

1) *Curriculum learning results*

a) ANTICOLLISION

Behaviour. Both algorithms MASAC and MASAC-Distrib are not differentiated since their reward function is the same at this stage. However, anti-collision measures were observed.

Metrics. The training seems to be satisfactory, considering the rewards obtained here, and the decrease in the number of collisions between RCs or with the ground, as shown in Table II.

TABLE II. RESULTS, ANTICOLLISION TRAINING PHASE

Batch	**Number of collisions *per hour* between RCs or with the ground**	
	AVG	**STD**
First 30 missions of the training phase	3.15	1.68
Last 30 missions of the training phase	0	0

b) SPECIALISATION

At this stage, the two algorithms MASAC and MASAC-Distrib are not differentiated, since their reward function is the same.

Behaviour. At the end of the training phase, the observed behaviours changed significantly: DRIL, EW and WPN assets fly on average much more in areas where SBAD defences and MRL systems are found. The assets learn to avoid triggering air-defence firings.

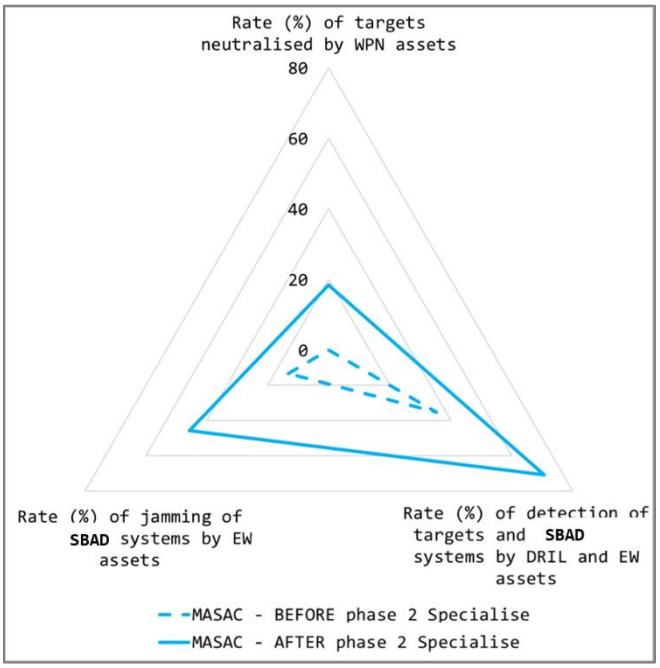

Fig. 4. Functional assessment, training phase N°2 Specialisation

Metrics. Fig. 4 allows to compare MASAC performance before and after the training phase: the training seems to be successful.

c) COLLABORATION

Behaviour.

MASAC-Distrib. At the end of the training phase, the observed behaviours changed significantly: DRIL, EW and WPN assets fly in formation in areas where surface-based air defence systems and targets are found. WPN assets wait until DRIL and EW assets have already spotted enemy mobiles before attempting to engage them. Tactics to avoid surface-based air defence systems are also in place.

MASAC. The training features much lower collaborative behaviour.

Metrics. According to Fig. 5-A, the training phase is successful for both algorithms, and for all metrics, in particular for MASAC-Distrib.

d) SURVIVE

Behaviour. At the end of the training, and for both algorithms, there is no significant change in the behaviour of the assets, except that they spend less time in the interception domains of ground-air systems.

Metrics. According to Fig. 5-B, the training phase is successful for both algorithms, regarding the surviving rate metric, in particular for MASAC-Distrib.

e) ADAPTATION

Behaviour. The observed behaviours are very similar for the MASAC-Distrib and MASAC algorithms. At the beginning of the training phase, the behaviour of the assets overfits the previous configuration of the mission area. The assets evolve in the area without taking into account the new location of the ground mobiles. Targets are no longer hit and assets run straight into the surface-based air defence (SBAD) systems' interception zones. At the end of the training, assets are able to adapt to the different positions adopted by the ground mobiles.

Metrics. According to Fig. 5-C, the significant variations in initial positions slightly reduces the global performance of both algorithms. In return, both algorithms improve their generalisation capability.

f) *Training from scratch*

In this experimentation, the agent is trained on the entire reward function *from scratch*: there is no curriculum learning strategy.

Behaviour. At the end of the training phase, the observed behaviours changed significantly, compared with the non-trained algorithm: the assets fly in the areas where the surface-based air defence systems and targets are found. No particularly collaborative behaviour are observed.

Metrics. According to Fig. 5-D, the curriculum approach significantly improves the global algorithm performance.

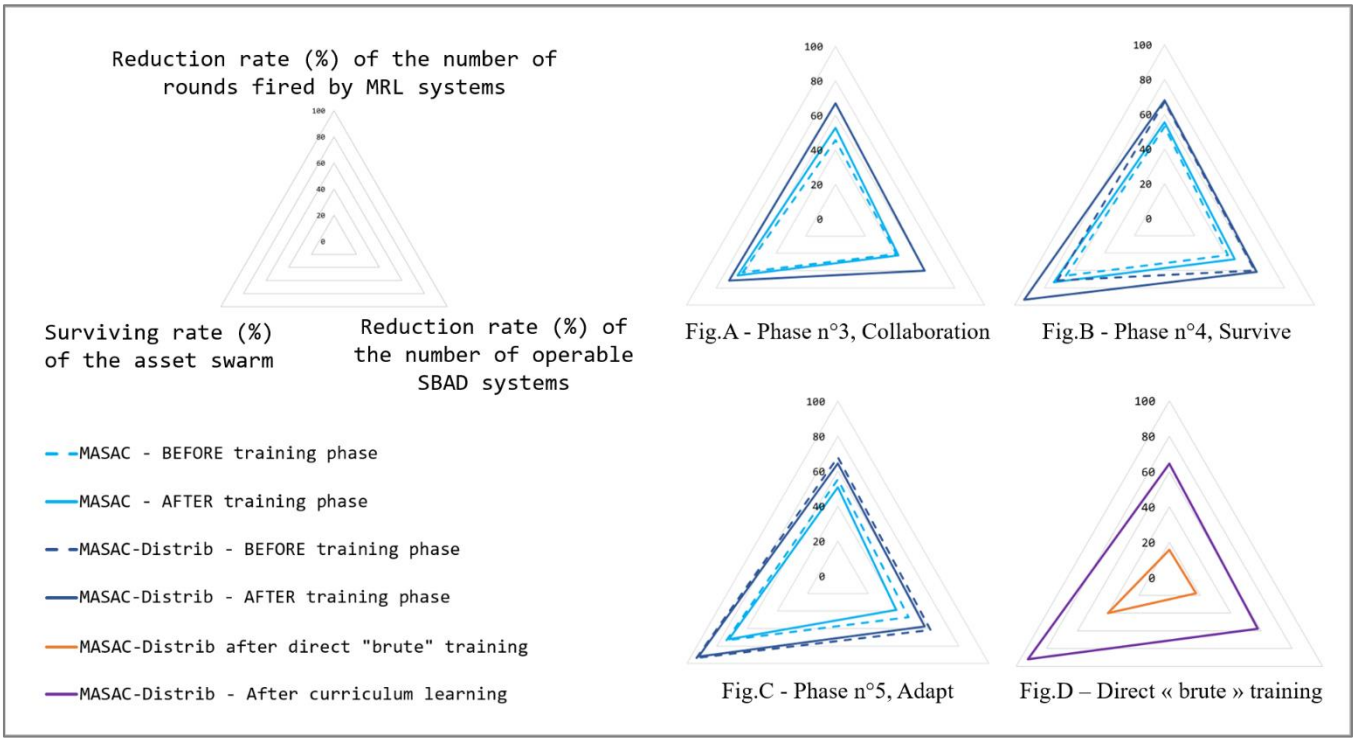

Fig. 5. Operational assessment, curriculum learning, phases n°3, 4, 5 and training *from scratch*.

2) Operational evaluation and benchmark

Behaviour.

ALEA. The behaviour of the agents controlled by the ALEA algorithm practically corresponds to a random walk, except that we impose that the agents remain in the mission zone. The trajectories being artificially smoothed, this last constraint explains the observation of sometimes less erratic behaviour.

GREEDY. The GREEDY algorithm allows the agents to fulfil some of their operational objectives. No collaborative behaviour is observed.

MASAC. The behaviour of the agents controlled by MASAC seems more intelligent and adaptative than the behaviour of the ALEA and GREEDY algorithms. In particular, we sometimes observe collaborative behaviours, and especially an excellent adaptation to the variations of the scenario.

MASAC-Distrib. The MASAC-Distrib algorithm takes up the clear improvements observed between MASAC and GREEDY, to which are added **numerous collaborative behaviours between agents**, in particular regarding target designation and coordination of jamming and attacks.

Finally, it should be remembered that the simulation, and therefore the observed behaviours, are **simplified** compared to a real environment**.** While the adaptability of the agents seems satisfactory within training phase N°5 ADAPT, it is not certain that the agents adapt correctly to greater variations in the scenario, nor real wartime conditions.

Metrics. In the following section, MASAC-Distrib is benchmarked against the algorithms presented in section III-C. The results are plotted as a bar chart featuring a confidence interval shown in black for each algorithm. Results were averaged over 100 missions.

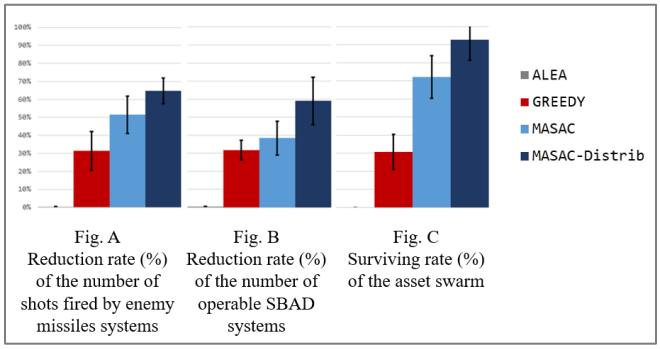

Fig. 6. Results, operational metrics in exploitation (test) mode

TABLE III. RESULTS SUMMARY

Algorithm	**Reduction rate (%) of the number of rounds fired by MRL systems**		**Reduction rate (%) of the number of operable SBAD systems**		**Surviving rate (%) of the asset swarm**	
	AVG	*STD*	**AVG**	*STD*	**AVG**	*STD*
ALEA	**0**	*0*	**0**	*0*	**0**	*0*
GREEDY	**31**	*11*	**32**	*6*	**31**	*10*
MASAC	**51**	*10*	**38**	*9*	**72**	*12*
MASAC-Distrib	**65**	*7*	**59**	*13*	**93**	*11*

Table III (above) summarises the results of Figure 6. These results were obtained by running each algorithm over 100 missions, in its evaluation mode.

B. Results analysis

1) Benchmark: a strong interest for DRL algorithms

It can be seen from Fig. 6 that the performance of the algorithm on the main operational metric is systematically ranked in the following order, from worst to best performance: ALEA, GREEDY, MASAC, MASAC-Distrib.

The semi-random algorithm ALEA completely fails to meet the operational objectives. The very low, but non-zero, observation and interference rates are due to the random exploration of the mission area by the DRIL and EW assets. The performance of the other algorithms is above that of ALEA, which confirms that problem is not trivial.

The functional performance of the MASAC algorithm is consistently above the performance of the GREEDY algorithm in all considered metrics (see Fig. 6). **DRL achieves results at least similar to a classical deterministic greedy approach in this scenario**.

Standard deviations are represented by black bars in Fig. 6. These **standard deviations are satisfactory** from a statistical point of view, as they allow a perfect distinction between all MASAC and MASAC-Distrib results. Nevertheless, they represent a significant proportion of the mean values with which they are associated. Reducing these standard deviations may improve the stability of the decisions, and thus the confidence that can be placed in these algorithms. It should be noticed, however, that **the GREEDY algorithm obtains standard deviations with the same order of magnitude as both DRL algorithms.**

Finally, according to section IV-A-2, where the final behaviours of MASAC and MASAC-Distrib are compared to other algorithms, **the interest of DRL seems to lie in a substantial improvement of the adaptability of the agents**.

2) Collaboration: interest of distributed reward functions

The secondary metrics may be divided into two classes:

- Performance measures of coercive actions, such as attrition rates of assets, surface-based air defence systems and targets,
- Performance measures of distant actions, such as MRL availability rates, jamming rates, and observation rates.

The MASAC and GREEDY algorithms obtain disparate results on the former, and relatively close on the latter. Paradoxically, remote actions such as jamming or observation are more "direct" than coercive actions: they do not imply a chain of actions and rely little on information provided by other assets, as they themselves are the source of information on positions for example. Coercive actions, on the other hand, require knowledge of the position of mobiles on the ground, or the implementation of collaborative strategies to reduce attrition while seeking to meet operational objectives, for example. The significant difference between the performance of the GREEDY and MASAC algorithms on coercive action measures could thus be explained by a **better collaboration in the MASAC case than within the GREEDY algorithm**. This collaboration encompasses the attacks of WPN assets relying on sensing and jamming performed by DRIL and EW assets.

Moreover, the MASAC-Distrib algorithm consistently performs better than the MASAC algorithm. This corroborates the results and behaviours identified in the training and **confirms the interest of the partial or global distribution of terms of the reward function**.

3) Contribution of curriculum learning

The training *from scratch* phase allows for a substantial improvement in results, as shown in Fig.5-D, but these results are very far from what is obtained with the sequential curriculum training. **The interest of progressiveness is therefore validated.**

Moreover, the MASAC-Distrib behaviours observed at the end of the training *from scratch* phase present relevant features, such as assets systematically flying over the targets area, but no collaborative behaviour is observed, such as collaborative target designation or collaborative spatial deployment. On the contrary, the MASAC-Distrib algorithm, trained with a curriculum approach, present numerous collaborative behaviours between agents, such as: collaborative target designation, coordination of jamming and attacks, formation flight, and collaborative planning of actions. **Collaborative swarm behaviour may complexify the training, in our case, curriculum is a key enabler to keep agent training efficient.**

V. CONCLUSION

Our MASAC-Distrib algorithm improves the operational performance of a swarm of assets in an air-to-surface complex mission. While it confirms the interest of Multi-Agent RL to optimise complex and collaborative decisions, many improvements need to be implemented in order to integrate these algorithms into operational systems. As part of the challenge we still need to face, we can mention the representativeness of the simulated environment and the robustness of the agents to improve generalisation on real-world wartime environments; anticipation of the decisions in order to improve trust with the teamed operator. Moreover, we need to integrate DRL in a weapon system architecture in a way that is compatible with safety, reliability and supervision requirements.

ACKNOWLEDGMENT

The authors would like to thank the engineers who were involved in this project for the quality of their contributions.

REFERENCES

[1] UK Ministry of Defence (2015), Future Character of Conflict, *https://www.gov.uk/government/publications/future-character-of-conflict*, visited on 1st June 2022

[2] Kim, H., Jordan, M., Sastry, S., & Ng, A. (2003). Autonomous helicopter flight via reinforcement learning. *Advances in neural information processing systems*, *16*.

[3] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[4] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of go without human knowledge. *nature*, *550*(7676), 354-359.

[5] Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning* (pp. 330-337)

[6] Bazzan, A. L. (2009). Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Autonomous Agents and Multi-Agent Systems, 18(3), 342-375.

[7] Hüttenrauch, M., Adrian, S., & Neumann, G. (2019). Deep reinforcement learning for swarm systems. Journal of Machine Learning Research, 20(54), 1-31.

[8] Fan, J. R., Li, D. G., Li, R. P., & Wang, Y. (2020). Analysis on MAV/UAV cooperative combat based on complex network. *Defence Technology*, *16*(1), 150-157.

[9] Wang, X., Yadav, V., & Balakrishnan, S. N. (2007). Cooperative UAV formation flying with obstacle/collision avoidance. *IEEE Transactions on control systems technology*, *15*(4), 672-679.

[10] Chandler, P. R., Pachter, M., & Rasmussen, S. (2001, June). UAV cooperative control. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)* (Vol. 1, pp. 50-55). IEEE.

[11] Liang, X. U., Xuanhong, P. A. N., & Ming, W. U. (2018). Analysis on manned/unmanned aerial vehicle cooperative operation in antisubmarine warfare. *中国舰船研究*, *13*(6), 154-159.

[12] Hou, Y., Liang, X., He, L., & Zhang, J. (2019). Time-coordinated control for unmanned aerial vehicle swarm cooperative attack on ground-moving target. *IEEE Access*, *7*, 106931-106940.

[13] Zhen, Z., Xing, D., & Gao, C. (2018). Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm. *Aerospace Science and Technology*, *76*, 402-411.

[14] Duan, H., Zhao, J., Deng, Y., Shi, Y., & Ding, X. (2020). Dynamic discrete pigeon-inspired optimization for multi-UAV cooperative search-attack mission planning. *IEEE Transactions on Aerospace and Electronic Systems*, *57*(1), 706-720.

[15] Ziyang, Z. H. E. N., Ping, Z. H. U., Yixuan, X. U. E., & Yuxuan, J. I. (2019). Distributed intelligent self-organized mission planning of multi-UAV for dynamic targets cooperative search-attack. *Chinese Journal of Aeronautics*, *32*(12), 2706-2716.

[16] Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., ... & Vicente, R. (2017). Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, *12*(4), e0172395

[17] Konda, V. R., & Tsitsiklis, J. N. (2003). Onactor-critic algorithms. *SIAM journal on Control and Optimization*, *42*(4), 1143-1166.

[18] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3), 229-256.

[19] Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, *12*.

[20] Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., & Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*

[21] Wang, X., Chen, Y., & Zhu, W. (2021). A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[22] Shao, K., Zhu, Y., & Zhao, D. (2018). Starcraft micromanagement with reinforcement learning and curriculum transfer learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *3*(1), 73-84.