Towards a Deep Reinforcement Learning based approach for real time decision making and resource allocation for Prognostics and Health Management applications

Ricardo Ludeke
Centre for Asset Integrity Management
Department of Mechanical and Aeronautical Engineering
University of Pretoria
Pretoria, South Africa
rpjludeke@gmail.co

P. Stephan Heyns
Centre for Asset Integrity Management
Department of Mechanical and Aeronautical Engineering
University of Pretoria
Pretoria, South Africa
stephan.heyns@up.ac.za

2023 IEEE International Conference on Prognostics and Health Management (ICPHM) | 979-8-3503-4625-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICPHM57936.2023.10194168

***Abstract* — Industrial operational environments are stochastic and can have complex system dynamics which introduce multiple levels of uncertainty. This uncertainty can lead to sub-optimal decision making and resource allocation. Digitalization and automation of production equipment and the maintenance environment enable predictive maintenance, which means that equipment can be stopped for maintenance at the optimal time instant. Resource constraints in maintenance capacity could however result in further undesired downtime if maintenance cannot be performed when scheduled.**

In this paper the use of a multi-agent deep reinforcement learning based approach for decision making is investigated to determine the optimal maintenance scheduling policy for a fleet of assets where there are maintenance resource constraints. By considering the underlying system dynamics of maintenance capacity, as well as the health state of individual assets, a near-optimal decision making policy is found that increases equipment availability while also maximizing maintenance capacity.

The proposed solution is compared to a run-to-failure corrective maintenance strategy, a constant interval preventive maintenance strategy and a condition based predictive maintenance strategy. The proposed approach outperformed traditional maintenance strategies across several asset and operational maintenance performance metrics. It is concluded that deep reinforcement learning based decision making for asset health management and resource allocation is more effective than human based decision making.

Keywords — deep reinforcement learning, multi-agent reinforcement learning, maintenance policy

I. INTRODUCTION

The increasing complexity and dynamics of production systems make it difficult for humans to determine the best possible decision making and resource allocation policies. Thus, the use of predictive maintenance alone is not enough and any upstream or downstream process inefficiencies can still lead to undesired downtime or lost production. From this perspective, it is important to consider all system dynamics when making decisions or allocating resources.

Reinforcement learning algorithms are well suited for determining optimal decision making policies in dynamic environments. Industrial operational environments are stochastic and can have complex system dynamics which introduce multiple levels of uncertainty. This uncertainty leads to sub-optimal decision making and resource allocation. Digitalization and automation of production equipment and the maintenance environment enable predictive maintenance, meaning that equipment can be stopped for maintenance at the optimal time. Resource constraints in maintenance capacity could however result in further undesired downtime if maintenance cannot be performed when scheduled.

This work investigates the applicability of using deep reinforcement learning to determine the optimal maintenance scheduling policy in a fleet of assets where there are maintenance resource constraints. By considering the underlying system dynamics of maintenance capacity, a near-optimal decision making policy can be found to increase equipment availability which ultimately results in increased revenue. We explore the use of deep reinforcement learning (DRL) for real-time decision making and resource allocation in a prognostics and health management (PHM) setting. The PHM setting investigated comprises a fleet of equipment that needs to be maintained with a constrained maintenance resource capacity. The fleet of equipment represents a multi-component system, with each respective piece of equipment representing a sub-component.

There is a finite maintenance resource capacity for the entire system. Each individual piece of equipment is controlled by an individual DRL agent. Each agent considers the current health state of its respective piece of equipment, as well as the available maintenance capacity, and decides whether to stop the equipment for planned maintenance, or to continue running the equipment. If an agent fails to stop the equipment before failure, corrective maintenance is performed. Agents cooperate in a multi-agent environment to learn a joint decision making policy that not only considers the health of individual pieces of equipment, but also considers the available maintenance capacity of the system. The decisions of individual agents can impact the maintenance capacity of the entire system, which can effectively also impact the decisions of other agents. The joint policy learnt by agents should thus account for the system interactions and enable optimal decision making across the fleet.

It is hypothesized that DRL based decision making for asset health management and resource allocation is more effective than human based decision making. The novel contribution of this work is a DRL framework for joint decision making in a multi-component environment with resource constraints, as well as using a single model that is trained end-to-end using equipment telemetry for both prognostic decision making and resource allocation for maintenance scheduling. The work presented here is based on research conducted at the University of Pretoria [1].

II. Reinforcement Learning

In recent years DRL has been used to determine optimal maintenance decision making policies for health management in several industries.

Reinforcement Learning (RL) is about solving sequential decision making problems, with the goal of maximizing some reward. When considering a real-world problem, such as playing a game, one can easily frame the problem in one's mind. During such gameplay, a sequence of actions needs to be taken to progress towards a winning state. The same thinking can be applied to the optimization of a maintenance strategy, in which asset availability needs to be increased whilst reducing cost. Humans solve these type of problems by taking advantage of available information and choosing an action that can be deduced as being the most beneficial. RL has great potential in solving such problems, as it mimics this same human decision making process.

The main components of RL, as shown in Fig. 1 on the left, are the agent and the environment. The environment is a representation of the observable world which the agent sees and interacts with. As the environment changes at each interaction step, the agent observes the changes in state s_t and selects an action a_t to take in the environment, which can in turn cause the environment to change. The agent receives a reward r_t signal from the environment, that reinforces how good or bad an action is given the current state of the environment. The goal for the agent is to learn a decision making policy that maximizes its cumulative future reward. RL methods allow agents to learn these optimal decision making policies in order to achieve this goal [2].

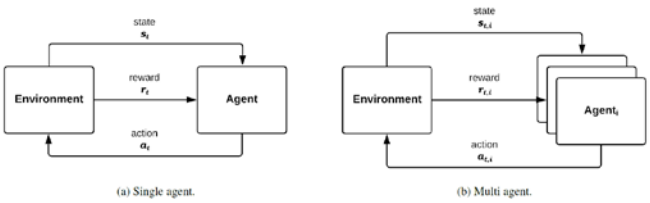

Fig. 1 The agent-environment interaction loop for a single agent on the left (a) and multiple agents on the right (b)

Multi-agent reinforcement learning (MARL) is commonly used for sequential decision making problems where multiple agents operate in a

common environment, as shown in Fig. 1 on the right. Each agent aims to optimize its own reward by interacting with the environment and the other agents [3]. MARL algorithms can be fully cooperative, fully competitive or a mix of the two. In this work, agents are cooperating in order to find the optimal maintenance strategy.

III. METHODOLOGY

To develop the methodology proposed in this work, we use the C-MAPSS turbofan engine dataset [4]. The characteristics of the C-MAPSS dataset has made it very popular for research on developing prognostic algorithms, because compared to several other publicly available data sets that are used for prognostic and health management research, it was generated using a high-fidelity simulation over the entire life-cycle of several engines, rather than measured from small scale experiments that are focused only on a small portion of equipment life. The range of parameters simulated is significant enough to represent the operation of a turbofan engine over its entire life-cycle. This dataset is also well documented and comprehensive. In a typical reinforcement learning environment, a direct simulation or model of the environment is available for agents to interact with. In this work a simulation environment is developed that samples from the static C-MAPSS dataset in order to simulate the degradation of engines rather than using the underlying high-fidelity simulation tool developed by NASA. The following sections describe the methodology in more detail.

A. Problem and data description

We consider a simulated maintenance strategy optimization problem that is designed to resemble a real-world operational system. An air freight company with a fleet of identical aircraft using identical turbofan engines delivers cargo 24/7/365. For every hour that an engine is down for maintenance, it is assumed that an aircraft is grounded, resulting in lost income. There is a maintenance team with limited resource capacity. If too many engines are taken down for maintenance, or if too many engines fail at the same time, the maintenance team will start backing up work resulting in further lost income. It is assumed that if an engine is stopped for maintenance before failure the time spent on maintenance is much less than if a failure needs to be repaired. The company has instrumented all engines with sensors and can monitor the condition of an engine at any point in time. Using the sensory information from engines in the fleet as well as the planned maintenance schedule, the company needs to decide when to schedule and stop engines for maintenance. The company needs to determine the optimal maintenance strategy in order to ensure maximum availability of engines across the fleet.

Using the simulation tool for C-MAPPS, Saxena [5] created five data sets under varying conditions, known as the PHM08 Challenge Dataset. Data sets consist of multiple multivariate time series. Each time series is from a different turbofan engine and the data can be considered to be from a fleet of engines of the same type. Throughout the simulations, wear parameters were varied to simulate continuous degradation trends. Each engine starts with different degrees of initial wear and manufacturing variation which is unknown. Telemetry was captured from various sensors on parts of the system, recording the effects of degradation on sensor measurements as time progressed until the engine failed. The data were contaminated with sensor noise. There are a total of 709 unique training trajectories in the dataset.

B. Simulation Environment

In order to simulate a real-world operational environment for training a reinforcement learning algorithm, the following is required:

- A simulated environment of the operational system that can be stepped through in time, such as the telemetry sensor readings from the C-MAPSS turbofan engines.
- A task or possibly a set of actions, that an agent needs to perform to reach some end state, such as letting the engine continue to run, or to stop for maintenance before failure.
- A measure of performance, or score, that can be used as a reward or penalty to tell the agent how well it is performing.

The objective is to create a simulated real-world operational environment where multiple agents can interact and coordinate their decision making and resource allocation actions in order to optimize the up- time of the controlled assets in the operational environment. The simulated environment was

developed as a custom multi-agent environment using the RLlib reinforcement learning library. RLlib is an open-source library for reinforcement learning that offers both high scalability and a unified API for a variety of applications [6]. The following sub-sections describe the developed environment's components in more detail.

C. State and observation state

Fig. 2 shows a flow chart of the agent-environment interaction loop.

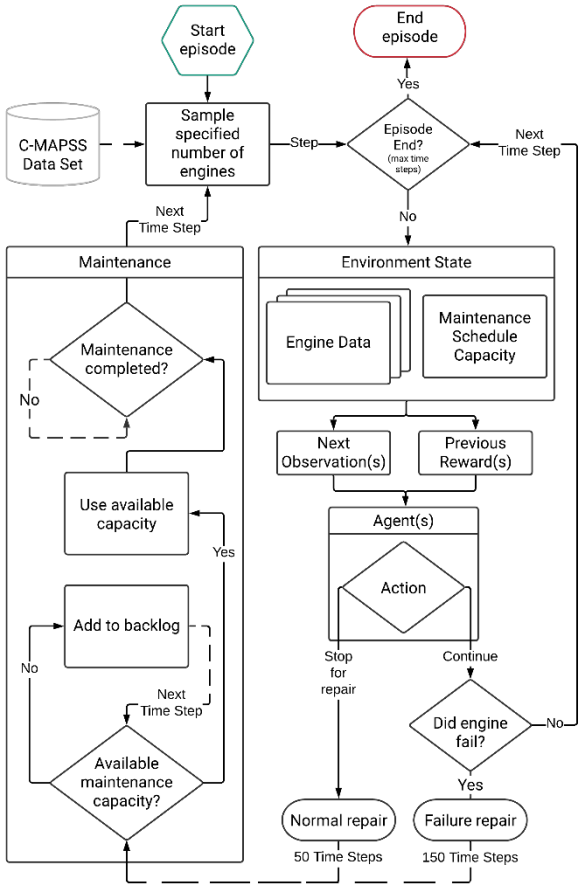

Fig. 2 Flowchart of the agent-environment interaction loop for the simulated maintenance problem.

$$s_t^i = \left(s_t^{i,e}, s_t^c\right) \tag{1}$$

At the start of each agent-environment interaction loop, the number of turbofan engines to be controlled is specified. The specified number of engine run-to-failure trajectories are randomly sampled from the C-MAPSS dataset. Each engine is controlled by a single agent. For example if a fleet of four engines are selected then four agents will also be created. At each time step the telemetry sensor readings $s_t^{i,e}$ from each engine are used as the observations for the respective controlling agent i. Each agent also observes the current available maintenance capacity at each time step for the entire system s_t^c as shown in equation 1.

Once an engine is stopped for maintenance, or because of failure, a new engine trajectory is sampled from the dataset – resembling an engine returning to full health after maintenance. Since each engine's trajectory is unique and fails at different time steps, or because agents can take independent actions to stop an engine for maintenance, the system is stochastic and resembles a real-world operational environment.

In this simplified maintenance strategy, the maintenance team only has a finite resource capacity. In the environment any maintenance that is scheduled, but exceeds the current capacity, is delayed until capacity frees up. This adds a team dynamic to the environment where each agent needs to consider the actions of the other agents, as well as the available current and future maintenance capacity. To simulate the resource capacity, if an engine is stopped for preventative maintenance before failure, a fixed number of steps are allocated. If an engine fails before being stopped, corrective maintenance usually takes longer to complete and a larger number of steps are allocated. The environment continues to run for a fixed time period. Agents continue to interact with the environment over this period, with the objective of maximizing the uptime of the fleet of engines under control, while also ensuring maintenance capacity is available.

D. Action space

The goal for each agent is to prevent engine failure, but also not to stop an engine for maintenance too early. The agent can take one of two discrete actions at each time step: stop the engine for maintenance a_m, or let the engine continue to run a_r.

$$a_t^i \in \{a_m, a_r\} \tag{2}$$

E. Reward

The first objective for an agent is to prevent engine failure. Using reward shaping, the reward grows exponentially as the engine reaches the end of its life. This encourages the agent to accumulate the maximum reward by stopping the engine close to its end of life. If the agent however fails to stop the engine before failure, a large negative reward is given to discourage agents from failing to prevent failure.

$$r_t^i = \begin{cases} 1 - \left((t_{failure} - t_i)/t_{failure}\right)^{0.35} & if\ t < t_{failure} \\ -t_{failure}/2 & if\ t = t_{failure} \end{cases} \quad (3)$$

The second objective of all agents is to maximize the fleet's collective uptime over an episode, by ensuring that there is always sufficient maintenance capacity. Agents need to take actions and allocate resources effectively in order to ensure that the collective uptime over an episode is optimal. This could mean taking actions that reduce an agent's individual reward, by scheduling individual maintenance early in order to free up maintenance schedule capacity for other engines. The group reward is given as a fraction of the current maintenance capacity over the capacity set-point per step.

F. Agent implementation

The agents learn the optimal maintenance strategy under constrained maintenance resource capacity. The agents observe the sensor readings from their individual engines, as well as the current maintenance capacity, and decides whether to let the engine continue running or to stop the engine for maintenance. If an agent decides to let an engine continue to run and the engine has failed, it is rewarded negatively to prevent this action from being taken in the given state. If the engine is stopped for maintenance shortly before failure the agent is rewarded positively to ensure this action is taken when in the same state. Agents are also rewarded for ensuring the maintenance capacity is not exceeded and thus learn to take actions that ensure that optimal scheduling is achieved.

The agent was implemented using an RLlib TensorFlow model and configured to interact with the custom multi-agent environment. The implemented model uses the clipped PPO policy gradient algorithm [7]. The model network weights are trained using stochastic gradient descent. The implemented model uses a fully connected neural network to learn feature representations of the environment's observations. Since the observations include the noisy sensor measurements of the engines, the problem is treated as a partially observed Markov decision process. The feature representations are thus further processed by a recurrent neural network, specifically using long short term memory (LSTM) cells, in order to capture any time dependencies over consecutive observations, such as a deviation in engine health over consecutive time steps. The output of the model is an action distribution that is evaluated in order to determine the next action. Fig. 3 shows the implemented multi-agent architecture.

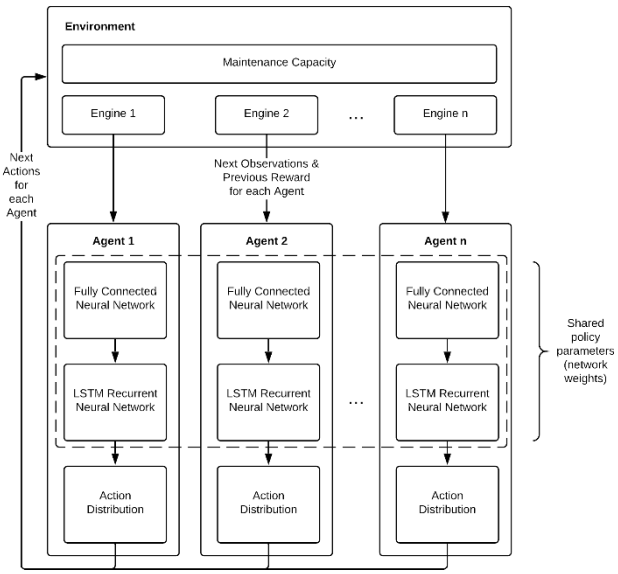

Fig. 3 Flow diagram of multi-agent deep learning architecture implementation.

In the multi-agent environment, agents were trained on individual observations, but the underlying network or policy weights and parameters were shared in order to learn a policy that can be used by any agent in the fleet.

G. Traditional maintenance strategies

In this work we consider three alternative reactive and proactive maintenance strategy implementations, namely run-to-failure corrective maintenance, constant interval scheduled maintenance and condition based predictive maintenance are discussed.

The corrective maintenance strategy implemented allows all engines to continue running until failure. This strategy does not consider the current health of any of the engines or the maintenance capacity and no repair interventions are made before failure.

Constant interval maintenance is a proactive preventive maintenance strategy with maintenance performed at constant intervals without considering the current health of any of the engines or the maintenance capacity. All historical failure trajectories are however used in order to perform a Weibull analysis, which is further used to determine the optimum preventive maintenance time. A 3-parameter Weibull distribution was fitted to the failure trajectories of the C-MAPSS engine data.

Condition based maintenance is a proactive predictive maintenance strategy in which maintenance decisions are made using equipment sensor telemetry to predict the real-time condition of the turbofan engines. An engine is stopped for maintenance once its condition deteriorates below a specified threshold. For this work we implemented a recurrent neural network approach. The details of these traditional maintenance strategies are not the focus of paper.

IV. Maintenance Strategy Performance Results

After model training converged, the different maintenance strategies were compared on the same randomly sampled trajectories over a single episode using the same environment parameters that were used during training. These experiments were repeated a thousand times to determine the mean and standard deviations for the reported metrics.

A. Availability and uptime

Availability is a maintenance performance metric that has a direct impact on the production capability of a system and thus significantly influences revenue. The implemented environment ensured that all engines were scheduled to operate over the entire episode, meaning that the availability and uptime metrics are equal. Fig. 4 compares the availability achieved using the different maintenance strategies over several episodes. As expected, it is seen that the run-to-failure corrective maintenance strategy achieved the worst availability of 38.16 ± 2.85%.

The scheduled preventive maintenance strategy achieved significantly better availability of 63.2±4.79%. The condition based predictive maintenance model achieved a mean availability of 68.04 ± 3.59%, however performed slightly worse than the DRL based approach. The implemented DRL based predictive maintenance strategy achieved the highest availability of 72.65 ± 1.86%, which means that only 27.35% of the time was spent on scheduled repairs. This represents a 55.77%, 25.68% and 14.42% improvement in uptime over the corrective, scheduled and condition based maintenance strategies respectively.

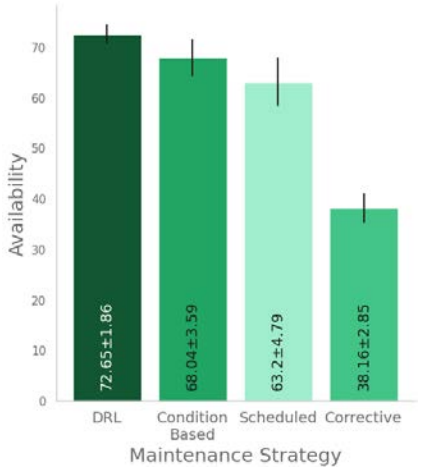

Fig. 4 Comparison of availability between different maintenance strategies over multiple episodes

Fig. 5 shows availability over a single episode. It is seen that significant periods of downtime appear in the scheduled maintenance strategy, where the availability remains horizontal for a longer period (around a normalized time of 60) compared to the condition based maintenance and DRL based

approaches. This could indicate that the maintenance capacity was reached and maintenance was moved into the maintenance backlog, resulting in periods with less productivity.

B. Planned Maintenance Percentage

Planned maintenance percentage (PMP) is an operational maintenance performance metric that considers how much maintenance is planned or unplanned. From Fig. 6 the planned maintenance percentage for the different strategies can be interpreted. Run-to-failure maintenance achieved 0% PMP as can be expected, seeing as the intention for this strategy is to let engines run to failure with no planned maintenance.

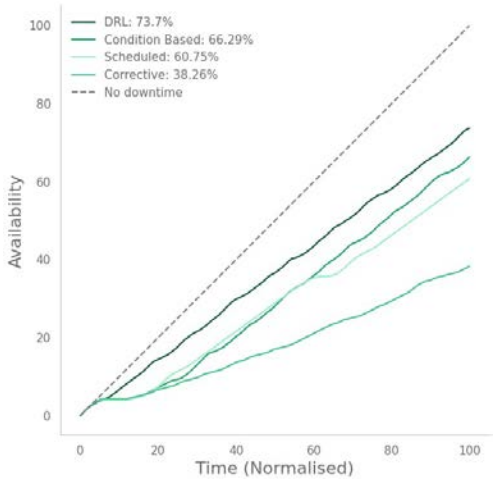

Fig. 5 Comparison of availability between different maintenance strategies over a single episode

Scheduled maintenance aims to achieve 100% PMP, however on average 1.98 engines fail before planned maintenance can be performed resulting in a PMP of 95.8%. The condition based maintenance strategy achieved a mean PMP of 92.97%. The DRL based strategy achieved a mean PMP of 97.3% with less than one engine failing on average before planned maintenance is scheduled. In comparison to the scheduled maintenance and condition based maintenance strategies, the DRL strategy resulted in a relatively small improvement in PMP of 1.5% and 4.33% respectively. The limitation to regular scheduled maintenance is that the actual health of assets are not considered to adapt the planned schedule, however the condition based maintenance and DRL strategies do consider the actual asset health. The benefit of considering the actual health only becomes apparent when also comparing asset utilization.

C. Utilization

Utilization is a measure of how effectively the asset was utilized over its lifetime. Utilization represents the total scheduled production hours over the total possible hours that the asset could have worked in a period, or over its lifetime. Fig. 7 shows utilization and utilization adjusted by PMP.

Run-to-failure maintenance achieves 100% utilization, which is a trade-off with several other factors such as increased cost and time lost due to increased maintenance time required to repair severe failures. The PMP adjusted utilization is 0% as this strategy ensures that all assets run to failure. The scheduled maintenance strategy achieves a mean utilization of 66.02 ± 2.63% and a PMP adjusted utilization of 63.11 ± 3.26%. This is a direct result of the optimal preventive maintenance interval calculated for optimal interval scheduled maintenance.

The condition based maintenance strategy achieves a mean utilization of 83.8 ± 1.44% and a PMP adjusted utilization of 78.07±2.98%. The improvement over scheduled maintenance is understandable, as the actual asset health is taken into account to determine when to stop engines for maintenance. The DRL strategy achieved a mean utilization and PMP adjusted utilization of 92.74 ± 1.55% and 90.18 ± 2.74% respectively. When compared to the scheduled maintenance and condition based maintenance strategies, this represents a 27.07% and 12.11% respective increase in utilization which ultimately contributes more to the asset life cycle cost.

By considering the asset health the DRL strategy is capable of extracting more life out of each engine without causing serious failure. The increased utilization of the DRL approach compared to the condition based maintenance strategy can be directly attributed to the predetermined stopping threshold selected to train the condition based model. The DRL agents were able to learn a decision making policy that was not constrained by

a predefined threshold, which suggests that this result is not significant, as the condition based predictive model can be fine-tuned to achieve the same, or better, utilization. The intention is not to achieve state of the art performance, but rather to evaluate whether the DRL approach is capable of performing both prognostic decision making that results in good utilization as well as perform maintenance scheduling which ensure optimal capacity and resource allocation ability.

D. Maintenance Capacity

Fig. 8 shows the mean maintenance capacity over an episode for all maintenance strategies. With a maintenance capacity set-point of 50 steps (using CP = 50 and CU = 150), the DRL strategy is the only strategy that maintained a significant positive maintenance capacity over all experiments, with a mean and minimum capacity of 14.14 (28.28%) and 10.8 (21.6%) respectively. The condition based maintenance strategy achieved a mean and minimum capacity of 0.73 (1.46%) and -8.34 (-16.68%) respectively. The scheduled maintenance strategy achieved a mean and minimum capacity of 2.09 (4.18%) and -8.65 (-17.3%) respectively. The corrective maintenance strategy was completely overwhelmed and achieved a mean and minimum capacity of -151.13 (-319.56%) and -165.64 (-331.3%) respectively.

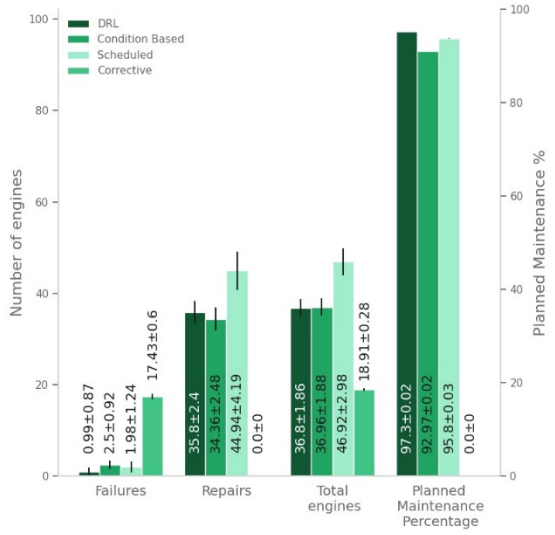

Fig. 6 Planned maintenance percentage achieved between different maintenance strategies

This result suggests that the DRL solution is able to learn how to schedule maintenance such that the maintenance capacity is considered and prioritized during decision making in order for capacity to remain positive. When considering both maintenance capacity and utilization it is shown that the DRL approach is in fact capable of performing optimal decision making and resource allocation in a multi-component system with maintenance resource constraints, compared to condition based maintenance, scheduled maintenance and corrective maintenance.

V. Conclusions

The main contribution of this work is the development of a multi-agent deep reinforcement learning (DRL) framework for joint decision making in a multi-component environment where a fleet of equipment is maintained with resource constraints. The proposed work uses an actor-critic policy optimization algorithm where each agent, or actor, has its own critic. Agents cooperate by learning a joint decision making policy. A fleet of movable equipment, from several turbofan engines in the C-MAPSS dataset, is considered, where telemetry sensor measurements from individual turbofan engines are used to make real-time prognostic health state estimations for the individual engines by using a multi-agent RL approach. The implemented environment also has maintenance resource constraints which agents needs to consider at each time step in order to schedule maintenance while trying not to exceed the available capacity.

One advantage of the implemented DRL framework presented here is that it enables decision making in multi-component environments where there are resource constraints. This can be extremely helpful in operational environments where uncertainty makes decision making and resource allocation difficult.

The second advantage is using a single model that is trained end-to-end using equipment telemetry and maintenance capacity for both prognostic decision making and resource allocation for maintenance scheduling. This approach reduces the complexity needed to implement a model into a real-world environment, where different data streams or models might need to be connected or coordinated, which represents its own challenges.

The implemented deep reinforcement learning based approach was capable of finding a near-optimal decision making policy that considered individual asset health as well as overall maintenance capacity in a multi-component environment with maintenance resource constraints.

The DRL framework outperformed traditional maintenance strategies across several maintenance performance metrics. It is concluded that deep reinforcement learning based decision making for asset health management and resource allocation is more effective than human based decision making. The proposed approach could be further extended as follows:

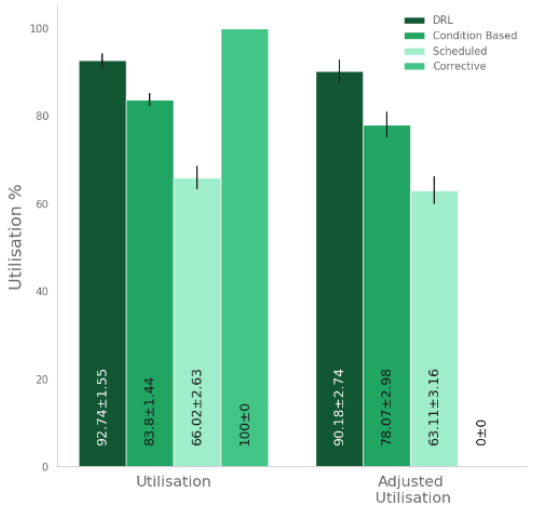

Fig. 7 Utilization and PMP adjusted utilization achieved between different maintenance strategies

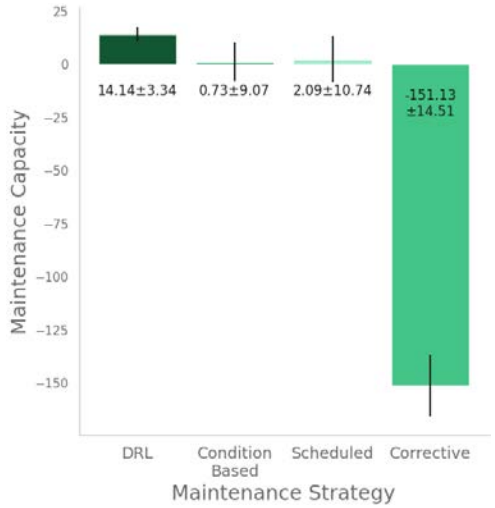

Fig. 8 Average maintenance capacity achieved between different maintenance strategies

- Consider other resource constraints that could have an impact on maintenance scheduling.
- Stochastic constraints should also be considered where constraints are not simply resource related. Lagrangian relaxation is successfully used by [8] to incorporate stochastic constraints into a DRL framework for inspection and maintenance planning under incomplete information.
- Consider equipment or data sets with several failure modes.
- Use model based reinforcement learning with planning techniques, such as Monte Carlo Tree Search, to enable agents to plan ahead.
- Combine unsupervised learning techniques with the implemented approach to learn a model of the environment. World Models [9] could be a potential approach to address this and the previous point.
- Implement further agent communication through the use of a centralized critic.
- Further evaluation is needed for the construction and shaping of reward functions to explicitly include constraints or other metrics in the reward formulation.

The proposed DRL framework is general enough without any specific limitations on the type of maintenance problem that can be addressed and provides the versatility to be extended to maintain other types of equipment or to consider other types of constraints.

REFERENCES

[1] L. Ludeke (2021). Towards a deep reinforcement learning based approach for real time decision making and resource allocation for prognostics and health management applications.

[2] L. Graesser & W.L. Keng (2019). Foundations of deep reinforcement learning: Theory and practice in python. Addison-Wesley Professional.

[3] L. Busoniu, R. Babuska, & B. De Schutter (2008). A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(2), 156–172.

[4] K.I. Parker & T.-H. Guo (2003). Development of a turbofan engine simulation in a graphical simulation environment. https://ntrs.nasa.gov/citations/20030093721.

[5] A. Saxena, K. Goebel, D. Simon & N. Eklund (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. 2008 international conference on prognostics and health management, 1–9.

[6] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan & I. Stoica (2018). Rllib: Abstractions for distributed reinforcement learning. International Conference on Machine Learning, 3053–3062.
[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford & O. Klimov (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
[8] C. Andriotis, & K. Papakonstantinou (2020). Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. arXiv preprint arXiv:2007.01380.
[9] D. Ha & J. Schmidhuber (2018). World models. arXiv preprint arXiv:1803.10122.