ICAS 2021 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

DESIGN OF SIMULATION-BASED PILOT TRAINING SYSTEMS USING MACHINE LEARNING AGENTS

Johan Källström[1], Rego Granlund[2] & Fredrik Heintz[3]

[1]Linköping University, Linköping, Sweden, {johan.kallstrom, fredrik.heintz}@liu.se
[2]RISE SICS East, Linköping, Sweden, rego.granlund@ri.se

Abstract

The high operational cost of aircraft, limited availability of air space, and strict safety regulations make training of fighter pilots increasingly challenging. By integrating Live, Virtual, and Constructive simulation resources, efficiency and effectiveness can be improved. In particular, if constructive simulations, which provide synthetic agents operating synthetic vehicles, were used to a higher degree, complex training scenarios could be realized at low cost, the need for support personnel could be reduced, and training availability could be improved. In this work, inspired by the recent improvements of techniques for artificial intelligence, we take a user perspective and investigate how intelligent, learning agents could help build future training systems. Through a domain analysis, a user study, and practical experiments, we identify important agent capabilities and characteristics, and then discuss design approaches and solution concepts for training systems to utilize learning agents for improved training value.

Keywords: Air Combat Training; Flight Simulation; LVC Simulation; Machine Learning; Reinforcement Learning

1. Introduction

Providing efficient and effective training solutions for fighter pilots is becoming increasingly challenging. Due to the high operational cost of aircraft, limited availability of air space, and strict safety regulations, it is difficult to realize training scenarios with the desired contents and density in a live setting. Instead, virtual and constructive simulation resources must be used to a higher degree. Live, Virtual and Constructive (LVC) simulation aims to integrate real aircraft, ground-based systems and soldiers (Live), manned simulators (Virtual) and computer-controlled entities (Constructive) [1]. By using constructive simulation to augment the live and virtual aircraft operated by trainees, it is possible to improve training effectiveness by simulating scenarios with a large number of participating entities [2]. However, training value will depend on the quality of the agents used to control the constructive entities. Ideally, these agents should be able to act as synthetic instructors, and adapt their behavior to the training needs of the human trainees. This would allow us to minimize the number of human support personnel required for conducting training, which would lead to lower costs and improved training availability.

As illustrated in Figure 1, we can divide the users of training systems into two major categories: training audience and training providers. The training audience consists of those in training, e.g., pilots learning how to operate a new aircraft subsystem, while the training providers consist of those delivering the training, e.g., instructors, operators, and role-players. Instructors are responsible for the pedagogical contents of a training session, while role-players and scenario operators help deliver the training by participating as actors or controlling parts of the simulated scenario respectively. If synthetic agents were to become smarter, they could replace or augment human role-players, and reduce the amount of human input required for the training scenario to progress in the desired way. To further raise the level of autonomy of the system, agents could also assist instructors in evaluating the performance of the trainees, and in adapting the contents and characteristics of training

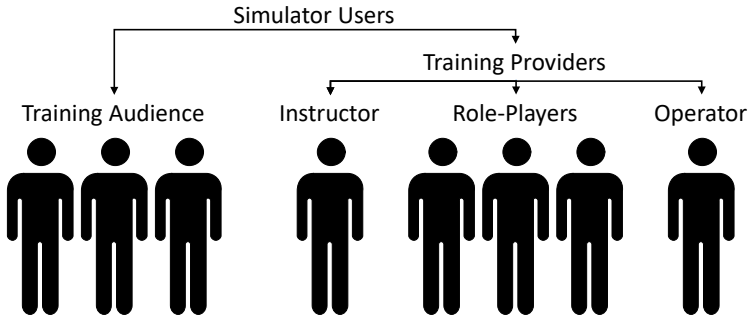

Figure 1 – Users of simulation-based pilot training systems (from [4]).

scenarios. However, creating behavior models for the agents is challenging, especially for end-users of training systems (e.g., instructors), who may not have the required expertise and experience [3]. In the past, this has constrained the use of agents in training. Now, with the recent advances in artificial intelligence (AI), there is hope that data driven methods will simplify the process of constructing intelligent agents, which could replace human support personnel in simulation-based training.

For learning sequential decision-making, reinforcement learning [5] has become the state of the art method. Guided by a human-designed reward signal, such agents can learn a policy purely by interacting with their environment. By leveraging deep learning [6], it has become possible to beat human champions in classic board games as well as multi-player computer games [7, 8, 9, 10]. The results have sparked interest in investigating applications of reinforcement learning in many domains, including air combat simulation. However, the focus has, to a large extent, been on maneuver optimization, rather than potential improvement of training value. To successfully design agents suitable for training, a good understanding of the domain and its actors is essential.

In this work, we proceed to study learning agents from a user perspective, with support from experienced fighter pilots. The goal is to learn more about how intelligent agents could be used to automate some of the tasks performed by human training providers. Our contributions and the structure of the paper can be summarized as follows.

- First, we perform an analysis of the domain of simulation-based training, using tools from Cognitive Work Analysis (CWA) [11] and the Joint Control Framework (JCF) [12]. The analysis is conducted from the perspectives of instructors and trainee pilots respectively. The purpose of the analysis is to identify constraints imposed on training providers when using different types of simulation resources, and to model the patterns of decision-making a synthetic agent must be capable of if it is to replace human role-players in air combat scenarios.
- Second, we conduct a user study, consisting of repeated interviews and a written survey, with the purpose of finding out what experienced pilots consider important agent capabilities and characteristics in different types of simulation-based training scenarios, and what challenges pilots are facing in their current training environment.
- Third, we conduct a study of human-agent interaction in an air combat scenario, where agents trained with a state of the art reinforcement learning algorithm cooperate with humans to solve an air policing task. The purpose of the experiment is to study how aspects of the agent design affects the agent's performance.
- Finally, we discuss design approaches and solution concepts within the context of a system architecture for a simulation-based training system that incorporates learning agents. The purpose is to provide a breakdown of the problem into smaller sub-problems, and provide framing for future research efforts.

The work forms a basis for future research on learning agents in simulation-based training.

2. Domain Analysis of Simulation-Based Training

In this section we conduct an analysis of the domain of simulation-based pilot training. The aim is to identify and illustrate how different types of simulation resources and tools affect the constraints imposed on actors that provide training, and what decision-making capabilities a learning agent would need to have to effectively participate in training scenarios, acting in a similar way as human role-players. In support of our study, we use two modeling tools: The Abstraction Hierarchy, and the Joint Control Framework Score (JCF-S) notation.

The abstraction hierarchy is a modeling tool used in Cognitive Work Analysis (CWA) [11]. CWA is a framework of methods and tools for analysis of the constraints imposed on actors, to support design of complex sociotechnical system. The abstraction hierarchy is used for work domain analysis, to identify constraints placed on actors by the system's purposes, values and priorities, functions, and physical resources [11].

The JCF-S notation was proposed as part of the Joint Control Framework (JCF) [12]. JCF-S is intended to support modeling of temporal aspects of human-machine interaction, at different levels of autonomy in cognitive control (LACC). The levels are summarized below.

1. The Physical level, which shows constraints related to physical actions.
2. The Implementation level, which shows constraints related to implementation properties.
3. The Generic level, which provides generic plans for common situations.
4. The Value level, which handles trade-offs among the system's objectives.
5. The Effect level, which deals with the system's purpose and goals.
6. The Framing level, which identifies the situation and context for control.

Levels 1 and 2 determine HOW control is realized, levels 3 and 4 deal with WHAT is done, and levels 5 and 6 are related to WHY the system exists. To model the joint control of man and machine, perception points (PP), decision points (DP), and action points (AP) are placed on six timelines, each of them representing one of the LACC levels. As a result, a pattern of the control loop of the joint system emerges. When agents are used in training scenarios, they should display a similar decision-making pattern as human pilots.

2.1 Abstraction Hierarchy

Figure 2 shows an abstraction hierarchy for simulation-based pilot training, which models the domain from the point of view of the organizations and instructors that provide training. The hierarchy identifies functions and objects that can be used to achieve the purpose of the system, as well as measures to evaluate its performance. The connections in the diagram illustrate the dependencies among the levels of the hierarchy. The functional purpose of the air combat training system is to ensure air force readiness by providing user-adapted training to fighter pilots, whilst considering constraints regarding physical resources, competencies, and time. Definitions of concepts used at the lower levels of the hierarchy are given below.

Value and Priority Measures

We have identified three important measures of system performance: *Training Effectiveness*, *Training Efficiency*, and *Training Availability*. *Training Effectiveness* measures the system's ability to deliver the type of training that allows trainees to develop towards the training goals. *Training Efficiency* measures the system's ability to deliver training while minimizing resource consumption. *Training Availability* measures the system's ability to deliver training when needed. We believe that learning agents could help improve performance in each of these measures, by affecting the contents of training scenarios as well as the way they are delivered.

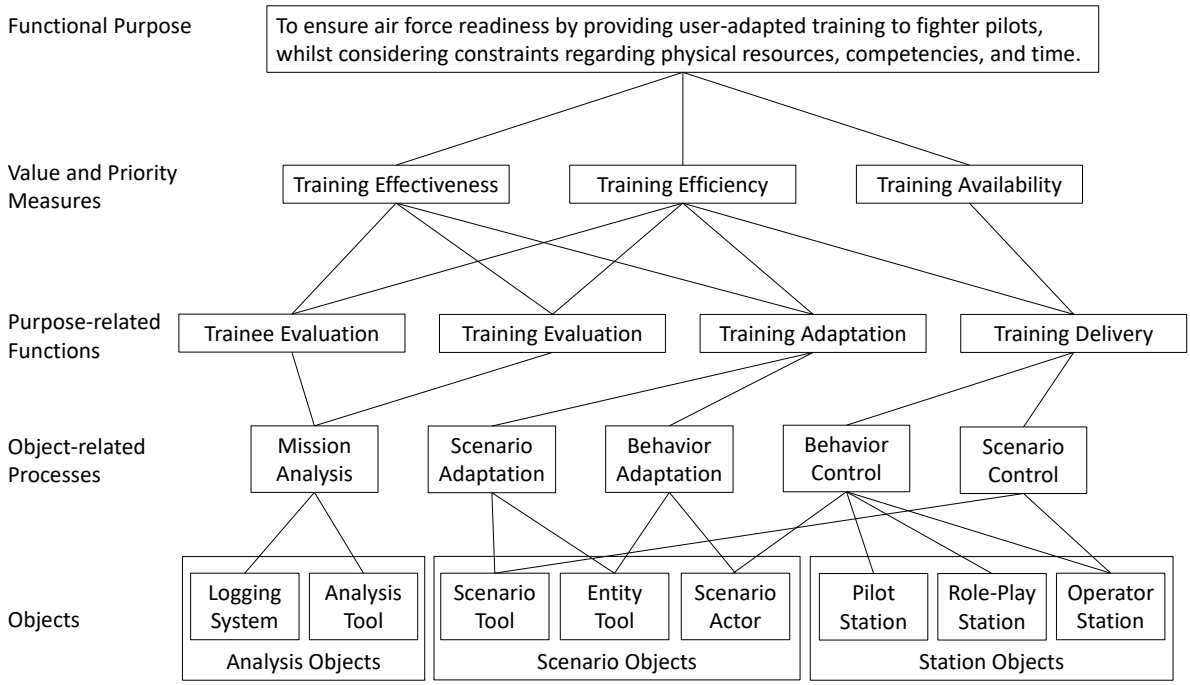

Figure 2 – Abstraction hierarchy diagram for a simulation-based pilot training system.

Purpose-Related Functions

We have identified four purpose-related functions: *Trainee Evaluation*, *Training Evaluation*, *Training Adaptation*, and *Training Delivery*. *Trainee Evaluation* is concerned with evaluation of individual trainees, and identifying the proficiency gaps that must be filled to achieve training goals. *Training Evaluation* is concerned with evaluation of the complete training process, for the set of trainees in training. *Training Adaptation* is concerned with adapting training to individuals and groups to improve their progress towards training goals. *Training Delivery* is concerned with organizing and delivering training contents to trainees in the system of interest.

Object-Related Processes

We have identified five object-related processes that can help realize the higher-level purpose-related functions: *Mission Analysis*, *Scenario Adaptation*, *Behavior Adaptation*, *Behavior Control*, and *Scenario Control*. *Mission Analysis* is used to study how trainees perform in simulated missions over time. *Scenario Adaptation* is used to adapt training scenarios so that they suit trainees' current training needs. *Behavior Adaptation* is used to adapt the behavior of synthetic agents to fit current training needs and training scenarios. *Behavior Control* is used to control the behavior of scenario actors while running training scenarios, e.g., through partially manual control of synthetic agents in case they are not able to operate fully autonomously, or through instructions given to human role-players. The control can be through a representative interface, or though more abstract, generic interfaces. *Scenario Control* is used to control scenario properties other than agent behavior, e.g., activating additional entities in the simulation.

Objects

We have organized the objects that support object-related processes into three groups: *Analysis Objects*, *Scenario Objects*, and *Station Objects*. *Analysis Objects* enable logging of data from training sessions, as well as trainee performance analysis and tracking. *Scenario Objects* enable scenario construction, model construction (e.g., aircraft and weapon models) and modeling of agent behavior, and provides the actors that populate training scenarios to stimulate the trainees, i.e., synthetic agents or human-role-players. The *Station Objects* provide the interfaces for users, i.e., trainee pilots, role-players, operators and instructors. Pilots and role-players can participate through Virtual simulators of varying fidelity, or through Live aircraft.

2.2 Constraints when using Different Types of Simulation Resources

Figure 3 shows the relative importance of a set of constraints, for three different types of training simulations, Live, Virtual, and Constructive.

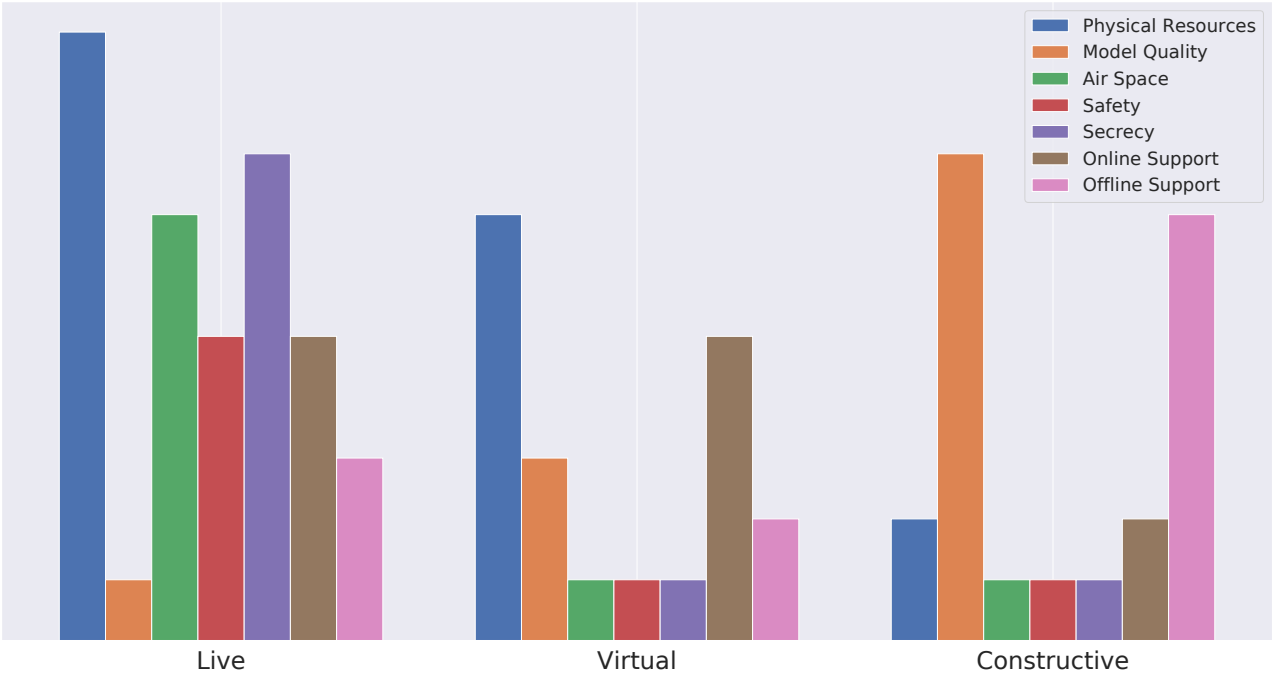

Figure 3 – Constraints affecting training for different types of simulation resources.

Live training simulation provides the highest possible fidelity in terms interaction with the aircraft and its subsystems. However, in the Live setting, training is highly affected by aspects of the physical world. For instance, the availability of vehicles and other types of systems may not be sufficient to realize complex scenarios. In particular, a military organization may not have access to systems that have the same performance and characteristics as those that are used by the enemy. Furthermore, operation of physical vehicles, e.g., aircraft, is highly expensive, which limits the amount of training that can be delivered in this setting.

Training in the Live setting is also constrained by the limited availability of air space, as well as safety regulations, which makes it difficult to realize scenarios with many entities, who are operating over a large geographical area. A large number of support personnel may also be required to plan and conduct such exercises. In addition, when acting in the open, there is a risk that systems' performance and tactics are revealed to opponents.

By using ground-based, Virtual simulators, the constraints imposed by the physical environment are lifted, and training delivery becomes easier. Still, there is a considerable cost related to populating complex scenarios with a large number of high-fidelity simulators. Constraints regarding model fidelity increase in this setting, in particular for within Visual Range (WVR) combat, where the effects of, e.g., g-forces is an important factor for pilot performance. To populate scenarios with only Virtual participants, some humans must act on the opponent's side. If they use high-fidelity pilot stations to play this role in the scenario, they can learn to understand the opponent's systems and tactics. However, if the simulators used for this type of role-play do not have sufficient fidelity, the training value will be low.

Constructive simulation makes it possible to realize large scenarios, populated by synthetic entities, which can replace human role-players. This reduces the need for physical resources, so that only the computation hardware for running the simulation software is required. Instead, the constraints are shifted to the fidelity of the simulation models, and the available offline support for building the

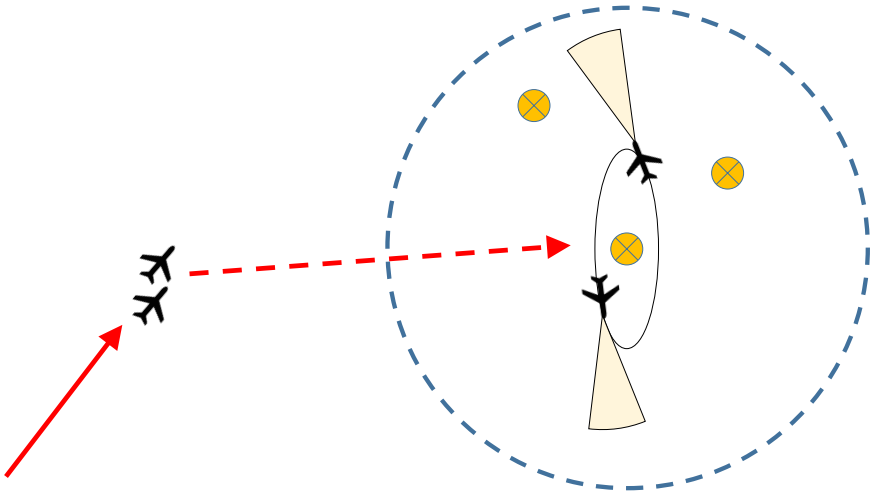

Figure 4 – Hostile entities approaching a Combat Air Patrol (CAP).

models, as well the simulation scenarios. In particular, it becomes challenging to construct behavior models for the synthetic entities, and adapting models to the training needs of individual trainees. Since the expertise required for such tasks may not be available locally, at each training facility, the turn-around time for updating training contents may be long. Instead, it may be necessary to have scenario operators manually control the flow of the tactical scenario to some extent.

Learning agents have the potential to reduce the constraints of constructive simulation, by simplifying the construction of high-quality behavior models. Data-driven methods can also provide objective evaluations of trainees, on a machine-readable format, which can support automated adaptation of simulation contents, so that training scenarios are always in pace with training needs.

2.3 Human-Machine Interaction for Decision-Making in Air Combat Scenarios

In this section, we study aspects of human-machine interaction in air combat scenarios. The aim is to illustrate to what extent perception, decisions, and actions are supported by the automation of the aircraft, and what parts of the control loop must be handled by the pilot alone. This information gives insight regarding requirements that must be fulfilled by synthetic agents that are to replace human pilots in training scenarios, and how to design the interface between the agent and the aircraft model, including its tactical systems. For synthetic agents, decisions made by human pilots must be sufficiently supported by AI, while the information available to support human decision-making should also be incorporated in decision-making algorithms to maximize performance.

To illustrate how the capabilities of the pilot's tactical control loop (Observe-Orient-Decide-Act) are mapped to different levels of cognitive control, we study the engagement of two pilots in offensive and defensive counterair operations, i.e., the quest for a favourable air situation, air superiority, or air supremacy. For this study, we use the scenario illustrated in Figure 4 to set the context. In this scenario, two aircraft are flying a Combat Air Patrol (CAP) directed towards the south, to protect their assigned Fighter Area Of Responsibility (FAOR), which is illustrated by the blue circle in the figure. The FAOR contains three high-value assets, which are illustrated in yellow in the figure. Approaching from the west are two hostile aircraft, which intend to perform an opportunistic attack on the high-value assets, and must therefore first deal with the defending aircraft of the CAP. We assume that the blue fighters are trainees, while the red fighters are human role-players, who try to support the training of the trainees.

The engagement is modeled using the score notation, and the result is illustrated in Figure 5. To simplify the notation we only present the engagement of two of the aircraft in the scenario.

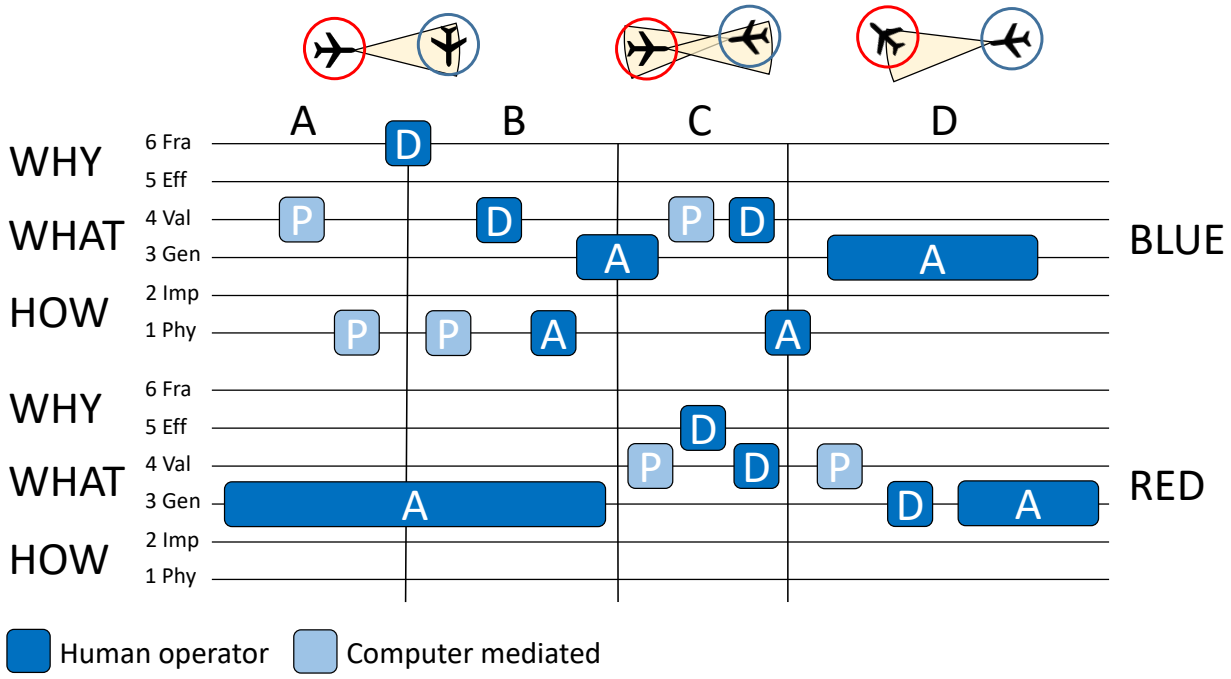

Figure 5 – Score for blue and red forces in a counterair operations scenario.

The timeline of the scenario is divided into four sections (A-D), where significant events occur. The behavior of the defending agent is presented in the top score (labeled *BLUE*), and the behavior of the attacking agent is presented in the bottom score (labeled *RED*). At the top of the figure, the geometry between the two aircraft in different sections of the scenario is illustrated.

In section A of the scenario, a hostile RED aircraft is approaching the FAOR of the opposing BLUE aircraft. The approach is carried out according to a pre-planned procedure (AP on level 3 Gen). The pilot of the BLUE aircraft is informed by the decision support system that it is in the radar field-of-view of the RED aircraft, and updates himself regarding the scenario geometry using the head-down displays (HDDs) (PP on levels 4 Val and 1 Phy respectively). He then makes a decision regarding the current threat level (DP on level 6 Fra).

In section B of the scenario, the pilot of the BLUE aircraft once again refers to the HDDs, to assess how to best deal with the threat (PP on level 1 Phy). The decision is then made that the most valuable course of action is to engage the target (DP on level 4 Val). After the decision has been communicated to the tactical air unit (AP on level 1 Phy), the pilot proceeds with target engagement according to doctrine (AP on level 3 Gen).

In section C of the scenario, the pilot of the RED aircraft is informed by the decision support system that it is in the radar field-of-view of the BLUE aircraft, which it is tracking (PP on level 4 Val). The pilot considers desirable effects related to tactical mission goals as well as trainees' training goals (DP on level 5 Eff), and decides to proceed into the BLUE aircraft's FAOR, with the hope of attacking a high-value asset (DP on level 4 Val). In the meantime, the pilot of the BLUE aircraft observes that the RED aircraft is now within range (PP on level 4 Val), and decides to fire a missile (DP on level 4 Val followed by AP on level 1 Phy).

In section D, after firing the missile, the pilot of the BLUE aircraft guides it towards the target according to doctrine, until handover (AP on level 3 Gen). The pilot of the RED aircraft is informed by the decision support system that there is an incoming missile (PP on level 4 Val), and performs an evasive maneuver to avoid the threat (DP followed by AP on level 3 Gen).

We can see that the pilot is supported by refined, abstract information, provided by the decision support system, to form his situational awareness. We can also see that several actions are pre-defined to handle a certain situation, and have a temporal extension, e.g., target approach procedures, missile guidance procedures, and evasive maneuvers. Finally, decisions on how to handle the situations that occur are often taken at the higher levels of cognitive control, were full automation may not currently be available. Therefore, the pilot still plays a vital part in the outcome of missions. He must have the capability to comprehend the situation, to identify and rank potential threats and targets. Then, when acting upon his situational awareness, the pilot must carefully choose how to use the tactical systems of the aircraft. These aspects need to be considered by learning algorithms

3. Learning Sequential Decision-Making for Air Combat Scenarios

In recent years, reinforcement learning has come to be the state of the art method for learning sequential decision-making. By leveraging deep learning [6], it has become possible to beat human champions in classic board games [7, 8], solve challenging robotics tasks [13, 14, 15], and learn how to play single and multi-player video games directly from pixel input [16, 9, 10]. The results have sparked interest in investigating applications of reinforcement learning in many domains, including air combat simulation.

Reinforcement learning allows an agent to learn a function for decision-making (policy π) by interacting with its environment in a form of trial-and-error learning [5]. A reinforcement learning problem is typically modeled as a Markov Decision Process (MDP), or derivations thereof. A Markov Decision Process is defined by the tuple (S, A, T, R, γ), specifying:

- S: The set of states of the process
- A: The set of actions of the process
- T: The transition dynamics of the process
- R: The reward function of the process
- γ: The discount factor indicating the importance of immediate and future rewards respectively

The agent interacts with its environment by selecting actions according to its policy ($a_t = \pi(s_t)$), and observes the resulting environment state (s_{t+1}) and the received reward (r_{t+1}). When the agent executes an action that results in high reward, that action is reinforced, so that it will be taken more often in the future. During learning, the agent must balance between exploration and exploitation, which is one of the greatest challenges of reinforcement learning. Exploration means that the agent selects exploratory actions to learn more about the environment, while exploitation means that the agent uses the knowledge gained so far to gather reward. The process is illustrated in Figure 6.

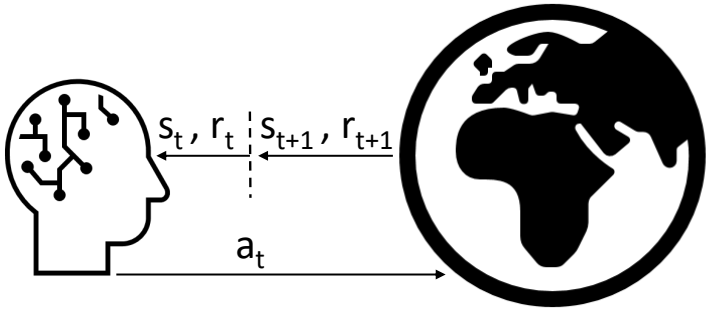

Figure 6 – Markov Decision Process.

The goal of the agent is to maximize its future expected return R_t when starting in state s_0 and then following policy π, which is captured in the state value function $V_\pi(s)$:

$$V_\pi(s) = E[R_t|s_0 = s] = E[\sum_{t=0}^{\infty} \gamma^t r_t|s_0 = s] \tag{1}$$

We can also define a state-action value function Q, which specifies the value of taking action a in state s and then following policy π:

$$Q_\pi(s,a) = E[\sum_{t=0}^{\infty} \gamma^t r_t|s_0 = s, a_0 = a] \tag{2}$$

The Q function can be used as a policy, by greedily selecting the action with highest estimated value. The Q function can be learned through Q-learning [17], by representing the Q function as a table of Q values, and applying the following update rule in each step of the episode:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha(r_t + \gamma max_a Q(s_{s_t+1}, a_{t+1}) - Q(s_t,a_t)) \tag{3}$$

As can be seen in the equation, the Q function is updated based on the difference in return estimates at different times, the so called temporal difference (TD) error, scaled by the learning rate α.

The tabular approach to reinforcement learning does not scale well to complex state and action spaces, which limits it applicability to many real-world problems. A breakthrough in reinforcement learning was the development of the Deep Q Networks (DQN) algorithm, which uses a neural network to represent the policy, making it possible for agents to learn how to play video games from pixels [16]. It uses a separate target network (updated less frequently than the policy network) to estimate the TD error, and trains the policy network with data sampled from a buffer of past experiences to stabilize learning. As one of the early scalable deep reinforcement learning algorithms, it has been evaluated in many application domains, including air combat simulation [18, 19, 20, 21, 4].

The DQN algorithm can only learn policies for discrete actions, which may limit the applicability to, e.g., robotics problems. The Deep Deterministic Policy Gradient (DDPG) algorithm extended deep reinforcement learning to domains with continuous actions [13]. It is an actor-critic architecture, that uses a deep Q network (the critic) to estimate the values of actions, to guide updates of the agent's (the actor's) policy. In air combat simulation, continuous actions can be valuable for platform maneuvering, and there have been a number of studies, primarily for within visual range (WVR) combat scenarios [22, 23, 24, 25, 26].

For success in air combat, agents need to learn how to cooperate with teammates in complex, competitive environments. This is a challenging task for reinforcement learning, since the environment becomes non-stationary when multiple agents are learning concurrently. Lowe et al. proposed the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm, an extension of DDPG to multi-agent environments [27], to address this challenge. The algorithm proposes to learn policies in a centralized fashion, allowing the critic of each agent access to the observations and actions of all other agents in the system. This simplifies determining what effect the behavior of an individual agent has on the dynamics of the complete system. Though such approaches are valuable for air combat simulation, there has been surprisingly few research efforts in that direction, although interest seems to be increasing [28, 4, 29, 30, 31].

While existing work on reinforcement learning for air combat simulation has covered some ground in investigating the applicability of different types of learning algorithms, the focus has been on optimization of air combat maneuvers, rather than potential added value for the users of simulation-based training systems. For this reason, the next section provides an analysis of desirable agent capabilities and characteristics from a user perspective, to identify how learning agents could support instructors and trainees.

4. User Needs in Simulation-Based Training using Learning Agents

In this section, we present the results of a user study, which aimed at identifying how intelligent agents could help make simulation-based pilot training more efficient and effective. We discuss which capabilities and characteristics agents are expected to have, from the perspectives of trainee pilots as well as instructors.

4.1 Organization of the Study

The study consisted of repeated user interviews, and a follow-up written survey. The participants of the interviews and the survey were experienced fighter pilots from the Swedish air force, and experienced test pilots from Saab Aeronautics. Three pilots participated in the interviews, while twenty-five pilots participated in the survey. The age of the participants of the survey ranged from 30 to 57 years, and their years of experience as pilots ranged from 4 to 22 years. Density estimates for the age and experience of the participants are shown in Figure 7.

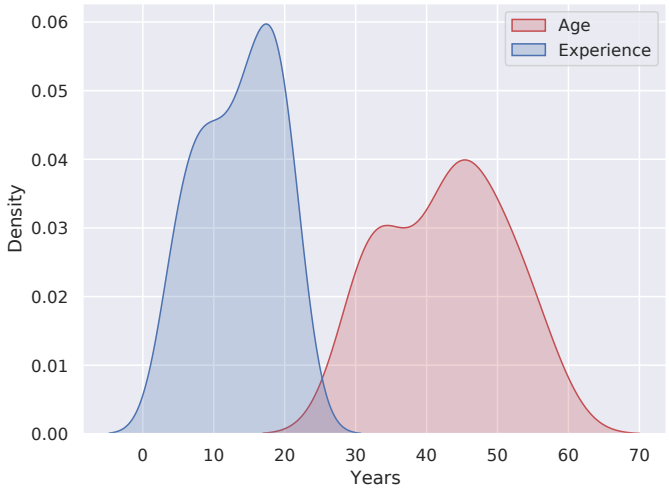

Figure 7 – Age and experience of survey participants.

The goal of the interviews was to allow pilots to describe current challenges in pilot training, and possible areas of improvement. In particular, the focus was on ways to automate training delivery to a higher degree using intelligent, learning agents, to reduce the dependency on support personnel such as role-players and scenario operators, and to improve the availability of high-quality training while reducing cost. Participants were initially asked to share their thoughts on training goals, training approaches, and training media, to give an unbiased overview of how training is currently conducted. Thereafter, the interviewers took a more active part, to identify the achievable training value when using agents in place of human role-players, to learn about challenges related to constructing training scenarios when using agents, and to discuss what role learning agents could play in simulation-based training systems in the future.

The interviews with pilots revealed a set of important factors that would need to be considered in the design of synthetic agents. For the written survey, based on the information gathered through the conducted interviews, a number of statements regarding desirable agent capabilities and characteristics were presented to the participants. They were asked to rate to what degree they agreed with the statements, for three different types of training: Basic Training, Tactical Procedure Training, and Mission Training. In the Basic Training phase, a pilot with previous experience in flying a different type of aircraft, or a different edition of an aircraft, is trained in basic flight maneuvers and system operation. In the Tactical Procedure Training phase, a pilot is trained in using, e.g., tactical sensors, data links, and weapon systems in typical combat scenarios. In the Mission Training phase, pilots are trained to cooperate in teams to carry out typical operational missions. The intention was to identify how user needs differed for these three types of training.

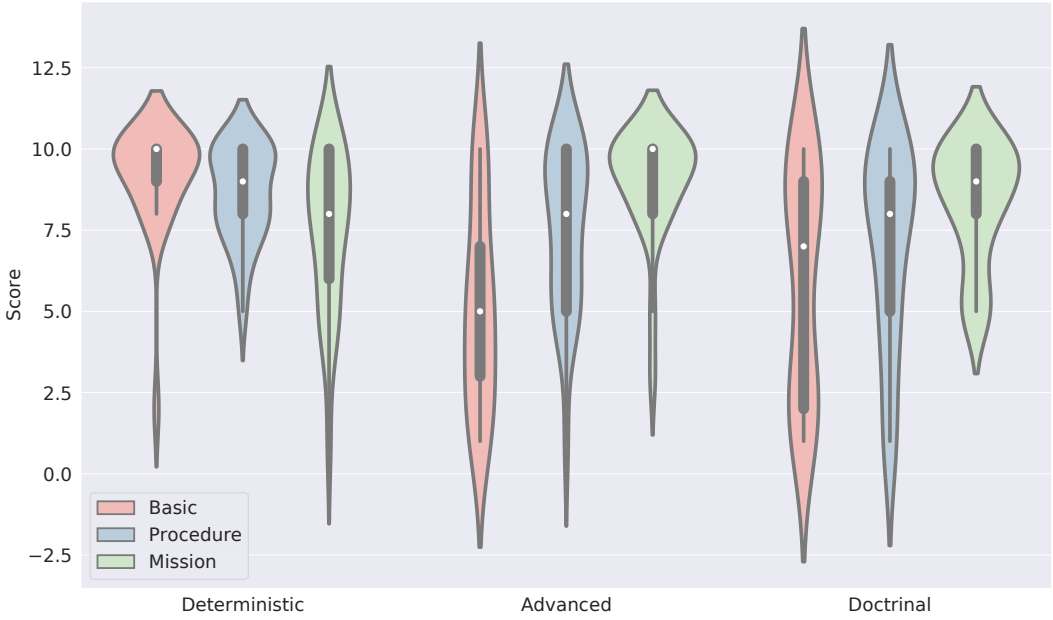

Figure 8 – Importance of different types of agent behavior.

The statements presented to respondees were divided into three categories: *Types of Agent Behavior*, *Human-Agent Interaction*, and *Agent Behavior in Training Scenarios*. Respondees were asked to give a score in the range one (low importance) to ten (high importance) for each statement, and they also had the possibility to add additional comments in free text. The results of the survey are presented as violin plots[1] for each category of training. In these plots, the outline of the "violin" is a density estimate for the answers, the thick bar in the center and the thin line in the center represent the interquartile range and the lower/upper adjacent values respectively, and the white dot on the bars represents the median score given by respondees.

4.2 Desirable Agent Capabilities and Characteristics

For the category of *Types of Agent Behavior*, the following statements were presented to respondees for rating:

- **Deterministic**: It is important that synthetic tactical entities can be given a deterministic behavior.
- **Advanced**: It is important that synthetic tactical entities can display advanced tactical behavior.
- **Doctrinal**: It is important that synthetic tactical entities can act according to doctrine.

The aim of his category of statements was to investigate the importance of different types of agent behavior in different types of training scenarios. The scores given by respondees, which indicate to what degree they agree with the statements, are presented in Figure 8.

Regarding the ability to assign agents a Deterministic behavior, we can see that this is important in all phases of training, although the scores vary more for mission training. The scores for Advanced behavior increases as we move from Basic Training, where the importance is modest, to Mission Training, where the importance is very high. Here, there is some variance in the scores for Basic Training and Tactical Procedure Training, while the pilots are in high agreement regarding the importance for Mission Training with advanced synthetic opponents. Finally, the importance of Doctrinal

[1] See, e.g., https://seaborn.pydata.org/generated/seaborn.violinplot.html

behavior shows a similar pattern as Advanced behavior, becoming more important as we move towards Mission Training, although it is fairly important already for Basic Training. As for Advanced behavior, the variance in the scores is higher for Basic Training and Tactical Procedure Training than for Mission Training.

In interviews, pilots expressed that in the initial phases of training, the requirements on synthetic opponents are rather modest. In this phase of training, it is important that the behavior of synthetic agents is predictable. The most important thing is to be able to create well defined, deterministic scenarios. For instance, when learning the functions and controls of a new sensor system, it may be distracting if opponents behave in an unpredictable manner. Instead, entities may move along predefined trajectories, or the positions of vehicles, including the trainee's own aircraft, may be frozen. In many scenarios, there are synthetic entities that are primarily used as background noise, and it is then desirable that they can perform simple tasks such as start and landing. For entities that play a tactical role in the scenario, there are also well established, standard maneuvers that they are expected to be able to perform, such as straight flight, gimbal turn, and pincer maneuver.

As the training progresses, more advanced agents, who can take defensive as well as offensive roles, are required to realize scenarios that allow trainees to develop their tactical proficiency. One pilot reasoned that a good base requirements for entity behavior is the ability to respond, in a believable way, to all orders available on the aircraft tactical data link. Furthermore, as explained by the participants in the study, providing agents that display advanced tactical behavior is a necessary, but not sufficient condition. It is also required that synthetic agents can follow a certain doctrine when acting in training scenarios, to prepare trainees for a variety of potential adversaries. Such a capability is a natural component of Mission Training, which is supposed to support preparations for specific missions. This means that when agent behavior models are developed using machine learning techniques, there must be a way to infuse domain knowledge in the learning process, so that the resulting behavior fulfills rules encoded in a specific doctrine.

For the category of *Human-Agent Interaction*, the following statements were presented to respondees for rating:

- **Challenging Opponent**: It is important that synthetic tactical entities can act as challenging opponents (e.g., by discovering and exploiting flaws in the human trainee's tactics and execution).
- **Wingman**: It is important that synthetic tactical entities can act as wingmen of human trainees, with intelligent behavior.
- **Voice Communication**: It is important that a synthetic tactical entity that act as wingman can communicate with human trainees through radio voice communication.

The aim of this category of statements was to investigate the importance of having agents act in different types of roles in different types of training scenarios, as well as the importance of voice interaction with agents. The scores given by respondees, which indicate to what degree they agree with the statements, are presented in Figure 9.

Regarding the ability of agents to act as Challenging Opponents, we can see a quite wide range of scores from Basic Training to Mission Training. The median score for Basic Training is low, while the median scores for Tactical Procedure Training and Mission Training are high. There is quite large variance in the responses for the two simpler categories of training, while respondees are more in agreement for the category of Mission Training. Having synthetic, intelligent Wingmen, is considered important for Tactical Procedure Training and Mission Training, but slightly less important for Basic Training, where simpler scenarios, populated with fewer entities, are often used for training. The importance of Voice Communication received scores in the mid of the range, and with high variance.

Figure 9 – Importance of different types of agent roles and voice interaction.

In Basic Training, having too Challenging Opponents may make it difficult to focus on learning how to, e.g., operate sensor and weapon systems. Instead, as noted previously, opponents may be configured to move along predefined routes, while acting according to predefined, predictable rules. For Tactical Procedure Training and Mission Training, having Challenging Opponents is essential, to evaluate the performance of trainees, as well as to validate the effectiveness of developed tactics. Pilots reasoned that if agents had a learning capability, they could identify flaws in human-developed tactics, and learn to exploit those flaws.

To be challenging opponents, agents need to posses similar capabilities as human pilots. Among other things, key to winning the fight is to coordinate with your teammates, achieve high time on station. and to detect others while not being detected yourself. Together with teammates, agent need to select good formations, maintain a favorable scenario geometry in relation to enemies (e.g., position, altitude, and movement), and keep enemies outside stand-off distance while agents themselves move into stand-off distance. It is important to maintain pressure on the enemy and cover a lot of surface (depth and width). Agents should also be able to learn to identify weak opponents, and target them for attack in coordination with teammates. Agents must carefully consider when to engage an enemy based on its value and threat level, so as to not take unnecessary risk, or waste fuel and missiles, In a similar way, when using sensor or electronic warfare systems, emission management must be considered to balance the chance of detecting opponents while avoiding being detected by enemies.

Since pilots do not operate on their own in real-world missions, support for team training is of utmost importance, as indicated by the scores from the survey. In interviews, pilots expressed that having agents that are intelligent enough to act as Wingmen of trainees is valuable, since it makes it possible to train as you fight even when there are not enough human pilots available to populate complex scenarios. At a minimum, self-paced training with 2-vs-2 fighters and a strike force should be supported. This requires that synthetic agents can learn to understand the intentions of trainees, as well as their own role in the mission.

For success in air combat, it is important that the members of a unit coordinate their actions well. Therefore, some level of communication capability among human and synthetic agents may be re-

quired. The need for voice communication within mixed teams of human and synthetic agents was included in the survey since it is a rich form of communication, which in general may be challenging to realize in a believable way for synthetic agents. However, pilots reasoned that in air combat the information exchange over radio channels is often of a simple form, following a predefined protocol. Using domain knowledge makes it possible to predict what types of interaction will occur, which helps when building models for the speech understanding and speech synthesis of agents. Pilots also argued, that in many situations they know how to respond to teammates actions without communication, since the team is trained in executing coordinated maneuvers. However, realizing such a capability in a synthetic agent may be challenging.

Populating training scenarios with mixed teams of human and synthetic agents can make training more efficient and effective. When using machine learning to build agents, learning behavior that supports interaction with humans is important, but also challenging, since during learning agents typically act in a simulation where no humans are present. The reason for this is that a large number of iterations, i.e., many thousands of simulated missions, are required to learn advanced behavior. Therefore, learning methods that allow agents to learn behavior that generalizes to diverse environments and scenarios are important.

For the category of *Agent Behavior in Training Scenarios*, the following statements were presented to respondees for rating:

- **Agent Performance**: It is important that synthetic tactical entities have realistic performance (e.g., do not always execute weapon delivery and evasive maneuvers perfectly).
- **Element of Surprise**: It is important that there is an element of surprise in the tactical scenario (i.e., the scenario does not play out in the exact same way in each run).
- **Behavior Explainability**: It is important that it is possible to explain the behavior of synthetic tactical entities in debriefing sessions (e.g., why a missile was fired in a certain situation).

The aim of his category of statements was to investigate the importance of different agent characteristics in the context of a training scenario. The scores given by respondees, which indicate to what degree they agree with the statements, are presented in Figure 10.

For the category of Agent Performance, we can see that there is high variance in the scores, but the median is at the lower half of the range for Basic Training. For Tactical Procedure Training and Mission Training, on the other hand, the importance of having realistic performance is high. The importance of having an Element of Surprise and variation in training scenarios increases as we move from Basic Training to Mission Training. Behavior Explainability is of high importance in all types of training.

In interviews, pilots argued that it is important that it is possible to adjust the agents' performance to suit specific trainees and training scenarios. As one pilot said, the learning agents should ideally be able to act as the perfect pedagogical instructor, adapting their behavior to the current training needs of trainees. Synthetic agents should not be perfect (e.g., performing perfect evasive maneuvers or missile delivery); they should make similar mistakes as human opponents would, so that trainees can learn to exploit such mistakes. When using human role-players for training, these will try to adapt to the proficiency level of the trainees, and then sometimes make small, intentional mistakes, which the trainees are expected to exploit. This imposes requirements on the algorithms used for learning the behavior models of agents, requiring them to learn models that can be adjusted in a similar way as human role-players can be instructed how to act in a training scenario.

Training scenarios with variation are important for advanced tactical training, so that trainees do not simply learn how the scenario plays out each time, and base their decisions on that information. Variation also makes it more difficult for trainees to exploit possible deficiencies in the behavior models

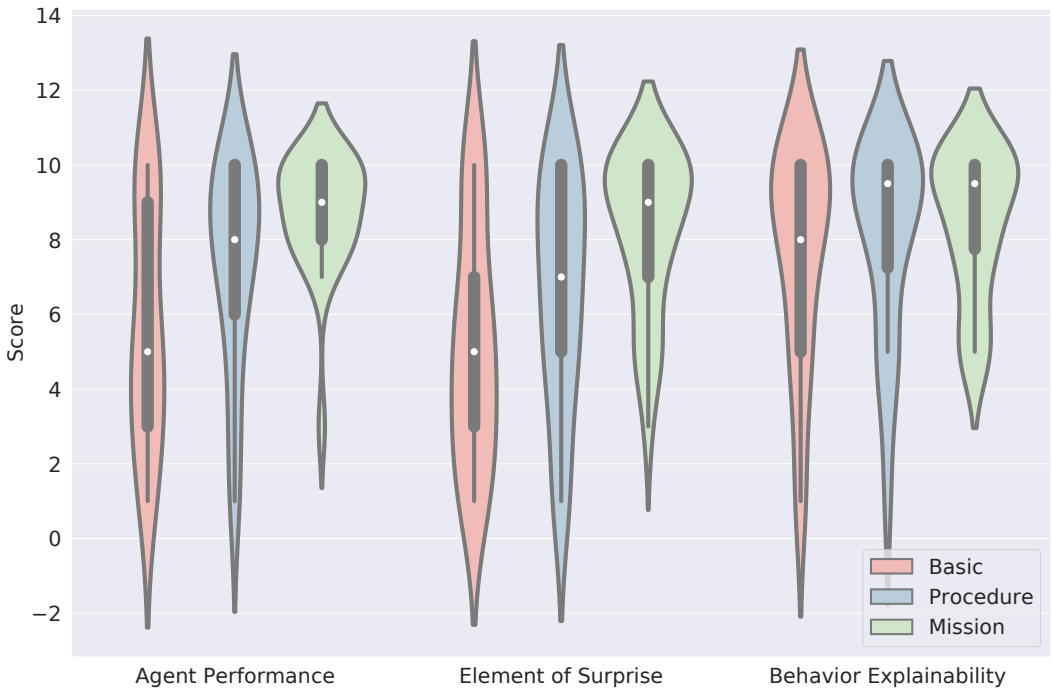

Figure 10 – Importance of different types of scenario characteristics.

used to control synthetic agents. The variation can be realized in different ways, e.g., by varying the goals of agents, by adapting the way in which agents try to achieve those goals, for instance by specifying different rules of engagement, and by varying the characteristics of the agents that populate the training scenarios, such as their proficiency, aggressiveness, and level of risk-taking.

Training sessions are typically concluded in a debriefing, where the outcome of the training scenario is discussed to determine what went well, what went less well, and areas for future improvement. In these sessions it is valuable if the decision-making process of the agent is transparent, so that the decisions made by agents at key points in the scenario can be understood by the human participants. When using traditional techniques for constructing agent behavior models, e.g., scripts, state machines, and behavior trees, tools for analysis of agents can be constructed by extracting suitable information from those models. For learning agents that use neural networks to represent the agent's policy, this process becomes more challenging, since neural networks are black-box models trained with data driven methods.

4.3 Limitations of Current Agent Technologies

In our interviews with experienced pilots, we discussed to what extent agents could currently be used to provide high quality training, and what challenges they were facing. Instructors are provided with behavior models from simulator engineers, but they may not fit all relevant training cases, especially as time passes, and aspects of the environment change. Therefore, it would be good if instructors could adapt training contents on their own, without the support of simulator engineers. However, instructors feel that this is difficult when using the tools that are currently available for behavior modeling. This is not surprising, since the construction of behavior models for multi-agent systems is a highly challenging task, and a very active area of research. When simulator engineers must be involved, the turn-around time increases, and It is also challenging to translate human domain expertise to model parameters that engineers can base their implementations upon.

At present time, the highest training value is achieved when using agents as opponents. This reduces the need for support personnel, who do not receive training, to participate in training scenarios. It also reduces the need for expensive equipment, e.g., aircraft or high-fidelity simulators. However,

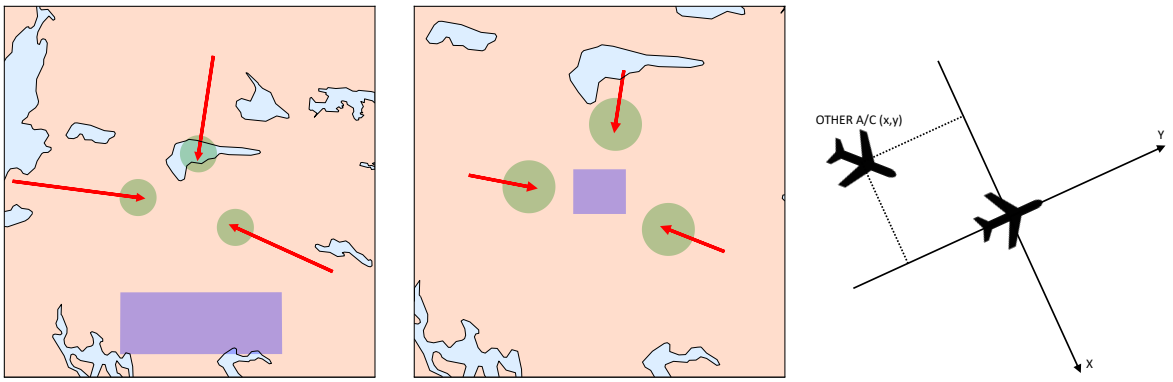

Figure 11 – Training scenario to the left, test scenario in the middle, and state space to the right.

handcrafted behavior models often result in behavior that comes across as scripted, static, and predictable. To get the variation required in a stimulating learning environment, a lot of manual work and time must be invested, and the cost of keeping in pace with training needs may be high. By using machine learning, it could become possible to construct behavior models that continually adapt to changes in the training environment, e.g., encounters with new trainees, introduction of new aircraft systems, and changes in trainees' tactics.

As it seems, users currently tend to be sceptical about replacing team members with agents. This is due to the richer interactions within a team of cooperating pilots. Pilots also expressed that even if agents had human-level intelligence, it may sometimes be desirable and preferable to train in units populated by the other members of your wing, since these are the people you would cooperate with in combat. However, having intelligent, synthetic wingmen opens up the opportunity to realize self-paced training, where individual pilots can train on their own at a time that suits them, without the support of an instructor, and without the need to populate training scenarios with other human pilots. For self-paced training, it would be valuable if synthetic pilots could be modeled to act in a similar way as specific human pilots, i.e., the team members of the trainee.

5. A Study of Human-Agent Interaction in an Air Policing Scenario

In this section, we present results from a practical experiment, intended to illustrate aspects of human-agent interaction when using machine learning agents in an air combat scenario. In this experiment, human operators were teamed with agents trained using reinforcement learning, to solve a task in an air policing scenario. The intention was to study how the mixed team of humans and agents performed on the task, and how the behavior of humans and agents differed. For these practical experiments, the participants were three experienced simulator engineers, and two of them also experienced pilots, although not fighter pilots.

5.1 Experiment Design

For the study of human-agent interaction, we used an air policing scenario developed in previous work [4] to train the agents. In this scenario, three agents should escort potential threats out of their air space, in order to protect three high-value assets. Incoming threats are controlled by handcrafted behavior models, implemented using behavior trees [32]. To escort a threat out of protected air space, an agent needs to fly within 5 km of this threat. The challenge for the agents is to learn to allocate threats among themselves, so that each agent can escort a threat out of protected air space. The scenario is illustrated to the left of Figure 11. Before each episode of training, defending agents spawn in random positions and with random heading in the blue rectangle. Threats approach along the arrows in red, towards the high-value assets in green.

To promote cooperation, the learning agents received a shared reward defined as:

$$r_t = -\sum_{i=1}^{3} \min(\|p_{a_i} - p_{d_1}\|, \|p_{a_i} - p_{d_2}\|, \|p_{a_i} - p_{d_3}\|) \quad (4)$$

where p_{a_i} refers to the position of attacker *i* and p_{d_k} refers to the position of defender *k*. The action space of an agent allowed it to fly forward, or turn left or right with a load factor of 2-4 g. The observation space of the agent was defined as the relative position of all other agents, in a body-fixed coordinate system, as illustrated to the right of Figure 11. To help the agent predict were the other agents in the scenario are going, it is given a stack of observations from the last 4 time steps in the episode as input to its policy. The agents' policies are represented by multilayer perceptrons (MLP), with 2 hidden layers, each with 64 neurons and the ReLU activation function. The policy is executed at a frequency of 1 Hz.

We trained the agents over 90k episodes, with each episode lasting for 600 steps, i.e., 10 min. We used the MADDPG algorithm [27], a learning rate of $\alpha = 10^{-2}$, a discount factor of $\gamma = 0.95$, and trained using the Adam optimizer [33].

After training, we study transfer of learning by evaluating agents in a slightly different scenario, illustrated in the middle of Figure 11. This scenario has a more compact geometry, which can make it more challenging to decide which agent should approach which threat. With this scenario, we wanted to analyze how the selected design of action space, observation space, reward system, and training approach affected the performance of the agent. Additionally, we replaced one of the synthetic agents with a human operator, to investigate how human pilots and learning agents could coordinated their actions to solve a cooperative task. We ran five iterations of the experiment, with starting positions and heading of defenders' aircraft generated by random for each iteration. Human pilots controlled their aircraft with a standard gaming joystick mounted on a desktop, observed the environment in an out-the-window (OTW) view presented on a monitor, and could also observe the positions of other entities in the scenario through a 2D map presented on a monitor to the right of the OTW view.

After human pilots had completed their five iterations, they were asked to answer the following questions based on their experience:

- **Q1**: How easy was it to determine an optimal target allocation?
- **Q2**: To what extent was the agents' behavior reasonable?
- **Q3**: How valuable do you think the following modes of interaction for coordination in these (and similar) scenarios would be?
  - **Q3.1**: Situational awareness map on the aircraft head-down display that shows, e.g., the position and combat value of other A/C.
  - **Q3.2**: Link text message with current high-level goals (e.g., priority target) of other A/C.
  - **Q3.3**: Speech message with current high-level goals (e.g., priority target) of other A/C.

5.2 Results

To the left of Figure 12 we present the mean and standard deviation for rewards received by teams with only agents compared to the rewards received by teams with a mix of agents and humans. We can see that the performance of the two categories of teams is similar, although teams with a human participant perform slightly better. From a qualitative point of view, when observing the outcome of each iteration, it could be seen that both teams were trying to split up and have one pilot approach each threat, which is the optimal tactic for the scenario. However, humans were better at quickly resolving conflicts, when two pilots started approaching the same target.

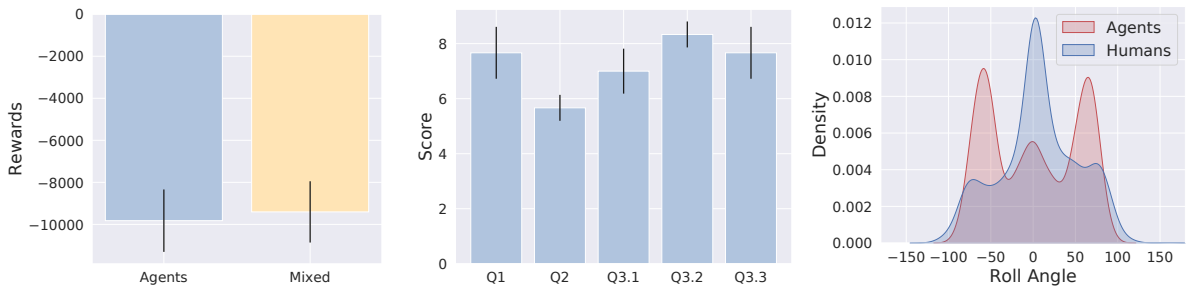

Figure 12 – Rewards to the left, survey results in the middle, roll angle distribution to the right.

In the middle of Figure 12 the results of the survey are presented as mean and standard deviation for the answers of the three pilots. Humans felt that in three of the five iterations it was clear how to allocate the threats among the pilots, while in two iterations it was not obvious what the optimal target allocation would be. The behavior of agents received fair scores by the human pilots. The major complaint was that when there was a conflict in target allocation, with a human pilot and an agent approaching the same target, agents might not immediately realize this and select a new target. In the experiments with pure agent teams, it was noted that when conflicts arise agents may also have difficulties determining which agent has the most favorable position to keep pursuing the target. Human pilots felt that the 2D map was valuable for coordinating with the other agents, but also reasoned that additional information presented in either text or speech messages, e.g., the targets selected for pursuit by agents, would further simplify the task. This is most likely true for the synthetic agents as well. To incorporate such functionality, the action and observation spaces of the agents could be modified. By letting the agent act by selecting which threat to engage, rather than acting by commanding the desired turn rate, information about the agent's selected target becomes available, and can be distributed over data link. This makes the agent's behavior more explainable and transparent, and coordination with human pilots could be improved. In a similar way, by modifying the agent's observation space, and including information about targets that have been selected for pursuit by other pilots in the scenario, the decision-making task of the agent could be simplified.

To the right of Figure 12 we can see the distribution of the aircraft roll angle for agents and human pilots over the iterations of the experiment. It can be seen that agents are turning frequently, while human pilots are more frequently flying straight. This is related to the design of the agents' action space as well as the design of their reward signals. The action space in this experiment is a low level action space, with continuous actions. This makes it challenging for the agent to explore, and to find the optimal action for each state. Furthermore, there is no component in the reward design that gives the agent an explicit incentive to avoid aggressive control of the aircraft. Either modifying the agent's action space, having it act using more abstract, high level actions, or modifying the agent's reward signal to penalize aggressive maneuvers, could help make the agent's behavior more human-like and believable.

The results of these experiments illustrate that when designing learning agents it is important to use architectures, abstractions, and training schemes that generalize over a wide range of environments, missions, and adversaries. We further discuss ways of constructing agents in the next section.

6. Introducing Learning Agents in Simulation-Based Pilot Training Systems

In this section, we first present an architecture for a simulation-based pilot training system, which automates parts of training adaptation and training delivery by incorporating learning agents. Based on this architecture, and the results of our cognitive work analysis, user study, and practical experiments, we then discuss design approaches and solution concepts that could produce agents with desirable capabilities and characteristics.

6.1 System Architecture for Simulation-Based Training using Learning Agents

A system architecture for a simulation-based pilot training system, which incorporates learning agents for improved efficiency and effectiveness, is illustrated in Figure 13. The architecture is an extended and modified version of an architecture proposed in previous work [4, 34].

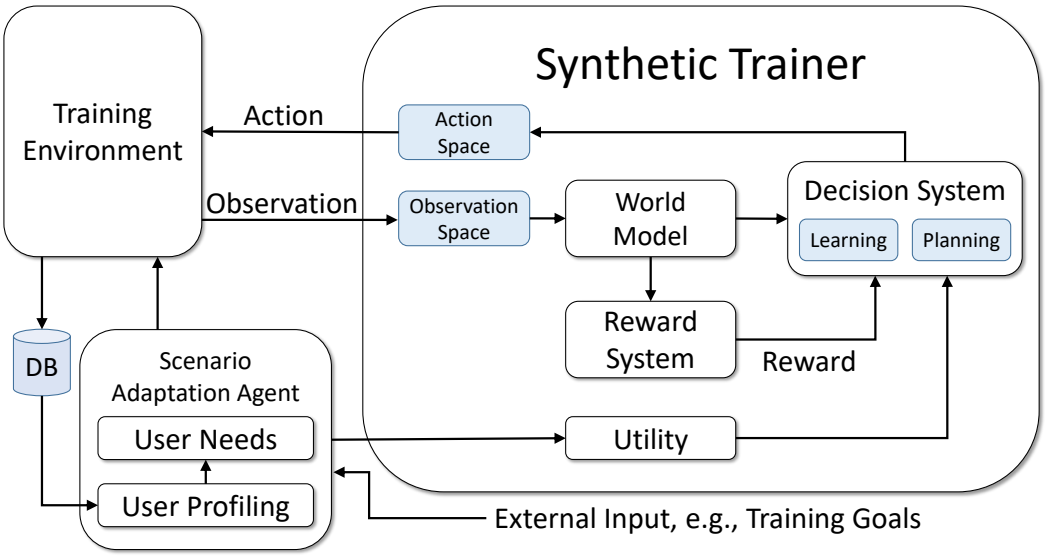

Figure 13 – System architecture for training system using learning agents.

The architecture integrates agents to support organisations and instructors in adapting training to trainees' training needs (Scenario Adaptation Agent), and delivering it in an efficient manner (Synthetic Trainer Agent). In training scenarios, agents participate as synthetic trainers, with the same purpose as human role-players. These agents act in one of the roles of the scenario, either as opponents or teammates of the trainee pilots. In the same way as role-players, their major goal is to provide a stimulating training environment to the trainees. Offline, between training sessions, agents analyze data generated in past sessions, for User Profiling, identifying trainees' weaknesses and strengths, and inferring current User Needs for future improvement of proficiency.

The goals of the trainer agent are modelled through its Reward System, which captures important features for successful decision-making in air combat scenarios. An individual agent's preferences among reward features are determined by the agent's Utility function.

The trainer agent's ability to perceive the state of its environment is essential for decision-making in the complex domain of air combat. The agent's perception is formed by two components: the agent's interface to the fused information of the aircraft's sensors (Observation Space) in combination with the agent's internal beliefs regarding the state of the world, based on its past observations (World Model).

The Decision System of the trainer agent realizes its capability to learn how to act (using the actions of its Action Space) based on passed experience, as well as its capability to evaluate and plan future actions based on its learned understanding of the dynamics of its environment, e.g., its ability to predict future behavior of other agents.

Agents enter the training environment through Computer Generated Forces (CGF) software, where their capabilities can also be further improved offline, in part based on interactions with human trainees.

We provide further discussions regarding these components, and ways of realizing them, in the sections below.

6.2 Reward System and Utility Function

For reinforcement learning agents, the goals that should be achieved are expressed through a reward signal. In the proposed architecture, the reward signal is generated by each agent's internal reward system, based on current and past states of the world, according to the agent's perception. The reward signal is a vector, whose components represent important features of the scenario, which should affect the agent's decision-making over time. For instance, rewards could be given for achieving advantageous geometry relative opponents, for detecting other entities while avoiding being detected, for missile hits, and for sensible resource management. This information supports control at level 4 of the LACC. The overall value of states and actions are determined by applying the agent's utility function U over the vector returns:

$$u = U(\boldsymbol{V}_\pi(s)) \tag{4}$$

This gives a scalar value u that supports ordering of policies.

The reward system design is one option for infusing domain knowledge in the behavior of the learning agent, to bias the learning process towards desirable characteristics, e.g., making an agent learn behavior which is in line with a certain doctrine. By using different combinations of reward components, a diverse set of agents can be created, which can make training more varied and stimulating. The components of the reward vector represent the objectives of the agent, and finding optimal policies results in a multi-objective optimisation problem.

One way to construct the reward system is to let a human pilot demonstrate how to solve a certain task, and then inferring a reward signal from this information, or simply rewarding the agent for behaving in a similar way as the human pilot [35, 36, 37]. Demonstrations are often used to support training of human pilots, so this approach leads to a man-machine interface that feels natural for the instructor, and reduces the need for explicit programming of agents.

Since dense reward systems, which give frequent feedback to the agent, as well as rewards based on demonstrations, introduce a bias in the agent's policy, they may prevent the agent from finding an optimal policy. To create very challenging opponents it may instead be desirable to use sparse reward signals, e.g., only rewarding the agents for winning a fight according to some metric. This allows agents to freely explore the world, and novel tactics and doctrines may emerge.

Agents that are to act as synthetic trainers for humans can not only consider components of the tactical scenario, they must also include information about the trainees' learning needs in their reward system, so that the decisions made during a training session are based on reasoning about training effect at level 5 in the LACC. This includes observing the trainees' proficiency in different aspects of air combat, and identifying ways of giving the trainees the right stimulation to improve their proficiency over time. This process is supported by analysis of data from past training sessions offline, through the Scenario Adaptation agent's profiling of trainees, and inference of training needs to meet the organisation's training goals. Agent behavior is then further adjusted during the progression of a training session, in a similar way as human role-players would adapt their behavior to the performance of trainees.

6.3 Observation Space and World Model

The design of the agent's observation space determines which features of the environment will be considered when making decisions. As illustrated in our analysis of decision-making in air combat scenarios, human pilots use low level features (LACC level 1) as well as more abstract value-based information (LACC level 4) to support their decision-making. For efficient learning of policies, agents should be supported by similar information. This includes, e.g., knowledge about the performance of own and opponents' vehicles, sensors, and weapon systems, which human pilots would have acquired in theoretical study.

To realize intelligent behavior, agents can not act only based on the immediate observation of the world, but must instead consider its whole history of observations. This functionality is realized by the agent's world model, which uses memory mechanisms to learn an abstract model of the state of the world, which can support decision-making. The model can be infused with domain knowledge, by explicitly modeling such features that human pilots believe are important for success in air combat, for instance, predictions regarding other agents' goals, beliefs, and future behavior. To support adaptation of behavior to trainees' current training needs, the world model should also provide abstract information related to training effect, e.g., estimates of trainees' proficiency. The functionality of the world models enables the agent to frame the current situation, and to reason about the effect of its actions (levels 5 and 6 of the LACC).

One challenge for learning agents is that they typically learn their policies in an environment populated exclusively by other synthetic agents, i.e., they do not interact with humans. The reason for this is the large number of iterations required for learning algorithms to converge. For agents to interact effectively with humans in training sessions, they need to have the capability of adapting their behavior to a wide range of teammates and opponents. One way to achieve this is to maintain a diverse population of agents while learning new policies, and to assemble teams of agents by random sampling from this population before each episode of learning [9]. Another approach is to use meta-learning, where agents learn to model characteristics and behavior of other agents based on few observations [38, 39]. Online, during training sessions, such an approach could be used as a basis for modeling a specific human trainee.

6.4 Action Space and Decision System

The design of the agent's action space has great impact on its ability to explore, and will affect its final learned behavior. For air combat simulation, it may be desirable to constrain the behavior of the agent, so that it resembles a certain opposing force. By using parametric action spaces, actions can be made available for selection only when certain conditions are fulfilled. Such approaches have been used to make sure that learning agents abide to the rules of games [8, 10]. In air combat simulation, for instance, rules of engagement can be encoded in the action space, to restrict when and how target engagement is allowed.

By including temporally extended actions in the design of the agent's action space, it becomes possible to learn a policy over actions at level 3 of the LACC. The options framework for hierarchical reinforcement learning provides a formalism for learning with temporal abstractions [40]. An option $\omega \in \Omega$ is defined as a tuple $(I_\omega, \pi_\omega, \beta_\omega)$, where:

- Ω is the set of available options
- I_ω is the initiation set, specifying in which states the option can be selected
- π_ω is the intra-option policy, i.e., the policy used once the option has been selected
- β_ω is the termination condition of the option, specifying the probability of the active option terminating in a state, to allow a new option to be selected

One benefit of temporal abstractions and hierarchical reinforcement learning is that agents' policies can become easier to understand [41]. Another benefit is that the performance of learning can be improved when a problem is broken down into a set of sub-problems, which are then dealt with in a decision-making hierarchy.

The options used for air combat simulation could be handcrafted, to replicate how temporally extended actions are executed according to a nation's air combat doctrine. This is a natural approach when the designer has a clear idea about how an extended action should be performed, e.g., actions

that have been optimized based on the laws of physics, such as missile guidance. Using handcrafted options as building block for learning tactics is also more likely to result in behavior that is believable to humans, than learning with low level actions alone, which can sometimes have undesirable effects, as illustrated in our practical experiments. For areas where there is greater uncertainty regarding how the agent should act to solve a task, there are also algorithms that make it possible to learn hierarchical policies from scratch, e.g., the option-critic architecture [42], the double actor-critic [43], and feudal reinforcement learning [44], which makes it possible to discover complex, novel actions.

As noted in our discussion on reward systems and utility functions, designing air combat policies is a multi-objective optimisation problem. Multi-objective reinforcement learning (MORL) provides systematic methods for learning sets of policies that are Pareto optimal, meaning that for at least one objective there is no policy that gives higher return [45, 46, 47]. We believe that this is a natural approach for reinforcement learning in the air combat domain, where tradeoffs between conflicting objectives are often required. The method supports decision-making at level 4 of the LACC. To adjust training to fit the needs of individual trainees, suitable agent policies can be selected from the set of Pareto optimal policies [48, 4, 29].

In MORL, there are two types of optimization criteria that are used when learning policies, scalarized expected returns (SER) and expected scalarized returns (ESR):

$$V_u^{\pi}(s) = U\Big(E\Big[\sum_{t=0}^{\infty}\gamma^t \boldsymbol{r}_t | s_0 = s\Big]\Big), SER \tag{5}$$

$$V_u^{\pi}(s) = E\Big[U\Big(\sum_{t=0}^{\infty}\gamma^t \boldsymbol{r}_t\Big) | s_0 = s\Big], ESR \tag{6}$$

The SER criterion aims to optimize the average outcome of several episodes, while the ESR criterion tries to optimize the average outcome of each episode. For air combat training, the ESR criterion may be the most suitable one, since from a safety perspective pilots want to optimize their chances of survival in each mission, rather than their expected survival rate over a complete campaign. We can see that for linear utility functions the criteria are the same, since the positions of expectation and utility functions can be interchanged. However, we argue that the utility function of a fighter pilot is not a linear function. For instance, safety may be considered infinitely more important than other objectives up until a certain probability of survival, resulting in a utility function with non-linear thresholds.

To further adapt agents' behavior to fit trainees' needs, planning algorithms can be used to adjust the agents' policies online, while a training session is in progress. These algorithms use the agent's world model to do simulated rollouts, to explore the effects future actions would have on the outcome of the mission. One family of planning algorithms that has had great success in, e.g., games of various forms is monte-carlo tree search (MCTS) [49]. MCTS can be combined with the learned value functions of the agent, to improve its performance [7, 8]. In training scenarios, planning could be used to adapt behavior to maximize the current utility of the agent, which is related to the training effect of trainees, and level 5 of the LACC.

6.5 Adapting Agent Behavior to Inferred Training Needs

To support adaptation of simulation contents and agent characteristics to current training needs, the Scenario Adaptation agent should learn a model of different aspects of trainees' proficiency, based on their performance in past training sessions. Performance measurements can include, e.g., measurements describing an agents' flight path, risk exposure, resource management, and success rate in engagements in missions. The model is used as input to the agents that participate in training, and affects their utility for different types of behavior, to achieve maximum training effect. This corresponds to having agents with a capability of perception and decision-making at LACC level 5, which is a highly challenging task, but recent machine learning techniques have shown promising results.

In addition to supporting adaptation of agent behavior, the Scenario Adaptation agent should also be able to adjust the contents of training scenarios, so that suitable components for improving trainees' performance are included. Here there is an overlap between human training and training of synthetic agent's behavior. For efficient learning, synthetic agents should be exposed to increasingly challenging problems, at a rate determined by their rate of improvement. In reinforcement learning, this is called curriculum learning [50, 51]. It is possible that curriculum learning techniques that have proven effective for training of synthetic agents could be adapted for training of humans as well, but further research on the topic is required.

6.6 Training Environments

The training environment is where agents and human trainees interact. Here, it is desirable to have high-fidelity models for the vehicles operated by the actors in the simulated scenario, as well as environment properties of various sorts, e.g., weather effects. However, when learning agent behavior using current state of the art reinforcement learning techniques, many iterations of missions are required, leading to long simulation times if a complex simulator is used. For this reason, it is valuable to have the possibility to adjust the fidelity of the simulation in several steps. The initial learning can then take place in lower fidelity environments, for many iterations, and the learned policies can then be successively transferred to environments with higher fidelity models for fine-tuning.

For our evaluations of learning agents, we use simulations of varying complexity and fidelity. Concepts are developed and analyzed in desktop simulations, with simple scenarios, and then further developed for integration in the target environment. The target environment is a high-fidelity tactical simulation, used for training of fighter pilots that operate the Saab Gripen aircraft. In this environment, evaluations with multiple manned stations can be performed. In operational training, the simulations are intended for future use in ground-based simulators as well as embedded training solutions in the aircraft, which itself is integrated in a distributed simulation network using a data link.

7. Conclusion

In this work, we studied introduction of intelligent, learning agents in simulation-based pilot training systems from the users' perspective. We analysed how agent technologies relate to constraints imposed on actors in training systems, and what decision-making patterns should be supported by agent designs. Through interviews, a survey, and practical experiments we learned about requirements on agent capabilities and characteristics, challenges and shortcomings of current agent technologies, and aspects of human-agent interaction with agents constructed using state of the art reinforcement learning techniques. Finally, we discussed design approaches and solution concepts for a training system architecture that integrates learning agents.

We conclude that the ongoing revolution in artificial intelligence is providing great opportunities for improvement of training efficiency and effectiveness. While our focus in this paper was on military training, many of the discussed concepts have broader applicability, e.g., for training simulations of other sorts, including training of pilots for civilian flight, where agents could help realize dense air traffic and patterns of life, automated setting of adversarial weather conditions and malfunctions, as well as automated evaluation and profiling of trainees.

For future research, we recommend further development of learning agents from the users' perspective, to steer progress in the most valuable direction. Real-world air combat scenarios provide many challenges to agents, e.g., cooperation and competition in scenarios with many agents, decision-making under partial observability and uncertainty, and the need to prioritize among multiple conflicting objectives, such as tactical mission goals, resource consumption, and safety. Therefore, the domain of air combat training is an excellent benchmark for reinforcement learning algorithms, and there are many exciting directions of research left to explore.

8. Contact Author Email Address

The corresponding author is Johan Källström, mailto: johan.kallstrom@liu.se.

9. Acknowledgements

This work was partially supported by the Swedish Governmental Agency for Innovation Systems (grant NFFP7/2017-04885), and the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

This work was supported by computation resources provided by the Swedish National Infrastructure for Computing (SNIC) at Tetralith/NSC partially funded by the Swedish Research Council through grant agreement no. 2020/5-230.

10. Copyright Statement

References

[1] Ernest H Page and Roger Smith. Introduction to military training simulation: a guide for discrete event simulationists. In *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*, volume 1, pages 53–60. IEEE, 1998.

[2] JJ Roessingh and GG Verhaaf. Training effectiveness of embedded training in a (multi-) fighter environment. Technical report, NATIONAL AEROSPACE LAB AMSTERDAM (NETHERLANDS), 2009.

[3] Nigel Gilbert. *Agent-based models*, volume 153. Sage Publications, Incorporated, 2019.

[4] Johan Källström and Fredrik Heintz. Multi-agent multi-objective deep reinforcement learning for efficient and effective pilot training. *FT2019*, 2019.

[5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[7] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[8] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[9] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

[10] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[11] Neelam Naikar. Cognitive work analysis: an influential legacy extending beyond human factors and engineering. *Applied Ergonomics*, 59:528–540, 2017.

[12] Jonas Lundberg and Björn JE Johansson. A framework for describing interaction between human operators and autonomous, automated, and manual control systems. *Cognition, Technology & Work*, pages 1–21, 2020.

[13] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[14] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[17] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[18] Roel Rijken and Armon Toubman. The future of autonomous air combat behavior. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3089–3094. IEEE, 2016.

[19] Pin Liu and Yaofei Ma. A deep reinforcement learning based intelligent decision method for ucav air combat. In *Asian Simulation Conference*, pages 274–286. Springer, 2017.

[20] Xiaoteng Ma, Li Xia, and Qianchuan Zhao. Air-combat strategy using deep q-learning. In *2018 Chinese Automation Congress (CAC)*, pages 3952–3957. IEEE, 2018.

[21] Xianbing Zhang, Guoqing Liu, Chaojie Yang, and Jiang Wu. Research on air confrontation maneuver decision-making method based on reinforcement learning. *Electronics*, 7(11):279, 2018.

[22] Qiming Yang, Yan Zhu, Jiandong Zhang, Shasha Qiao, and Jieling Liu. Uav air combat autonomous maneuver decision based on ddpg algorithm. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pages 37–42. IEEE, 2019.

[23] Weiren Kong, Deyun Zhou, Zhen Yang, Yiyang Zhao, and Kai Zhang. Uav autonomous aerial combat maneuver strategy generation with observation error based on state-adversarial deep deterministic policy gradient and inverse reinforcement learning. *Electronics*, 9(7):1121, 2020.

[24] Zijian Hu, Kaifang Wan, Xiaoguang Gao, Yiwei Zhai, and Qianglong Wang. Deep reinforcement learning approach with multiple experience pools for uav's autonomous motion planning in complex unknown environments. *Sensors*, 20(7):1890, 2020.

[25] Bo Li, Zhigang Gan, Daqing Chen, and Dyachenko Sergey Aleksandrovich. Uav maneuvering target tracking in uncertain environments based on deep reinforcement learning and meta-learning. *Remote Sensing*, 12(22):3789, 2020.

[26] Kaifang Wan, Xiaoguang Gao, Zijian Hu, and Gaofeng Wu. Robust motion control for uav in dynamic uncertain environments using deep reinforcement learning. *Remote sensing*, 12(4):640, 2020.

[27] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390, 2017.

[28] Guanyu Zhang, Yuan Li, Xinhai Xu, and Huadong Dai. Efficient training techniques for multi-agent reinforcement learning in combat tasks. *IEEE Access*, 7:109301–109310, 2019.

[29] Johan Källström and Fredrik Heintz. Agent coordination in air combat simulation using multi-agent deep reinforcement learning. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2157–2164. IEEE, 2020.

[30] Jian Xiao, Gang Wang, Ying Zhang, and Lei Cheng. A distributed multi-agent dynamic area coverage algorithm based on reinforcement learning. *IEEE Access*, 8:33511–33521, 2020.

[31] Weiren Kong, Deyun Zhou, Zhen Yang, Kai Zhang, and Lina Zeng. Maneuver strategy generation of ucav for within visual range air combat based on multi-agent reinforcement learning and target position prediction. *Applied Sciences*, 10(15):5198, 2020.

[32] Michele Colledanchise and Petter Ögren. *Behavior Trees in Robotics and AI: An Introduction*. CRC Press, 2018.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] Johan Källström. Adaptive agent-based simulation for individualized training. In *19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), May9–13, 2020, Auckland, New Zealand.*, pages 2193–2195. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[35] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[36] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[37] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4572–4580, 2016.

[38] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *International conference on machine learning*, pages 4257–4266. PMLR,

2018.

[39] Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pages 4218–4227. PMLR, 2018.

[40] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[41] Matthew Smith, Herke Hoof, and Joelle Pineau. An inference-based policy gradient method for learning options. In *International Conference on Machine Learning*, pages 4703–4712. PMLR, 2018.

[42] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1726–1734, 2017.

[43] Shangtong Zhang and Shimon Whiteson. Dac: The double actor-critic architecture for learning options. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[44] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549. PMLR, 2017.

[45] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[46] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. A practical guide to multi-objective reinforcement learning and planning. *arXiv preprint arXiv:2103.09568*, 2021.

[47] Roxana Rădulescu, Patrick Mannion, Diederik M Roijers, and Ann Nowé. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–52, 2020.

[48] Johan Källström and Fredrik Heintz. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Adaptive and Learning Agents Workshop (ALA-19) at AAMAS, Montreal, Canada, May 13-14, 2019*, pages 1–7, 2019.

[49] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[50] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.

[51] Wojciech Czarnecki, Siddhant Jayakumar, Max Jaderberg, Leonard Hasenclever, Yee Whye Teh, Nicolas Heess, Simon Osindero, and Razvan Pascanu. Mix & match agent curricula for reinforcement learning. In *International Conference on Machine Learning*, pages 1087–1095. PMLR, 2018.