JDST., vol. 4, no. 2, p12-p36, 2021
https://doi.org/10.46713/jdst.004.02

www.jdst.eu, e-ISSN 2534-9813

Research Article

Data-driven behavioural modelling for military applications

Maarten Schadd[a], Nico de Reus[a], Sander Uilkema[a], Jeroen Voogd[a]

[a] *TNO, the Netherlands Organisation for applied scientific research*
Anna van Buerenplein 1, 2595 DA The Hague, The Netherlands
htto://www.tno.nl

ABSTRACT

This article investigates the possibilities for creating behavioural models of military decision making in a data-driven manner. As not much data from actual operations is available, and data cannot easily be created in the military context, most approaches use simulators to learn behaviour. A simulator is however not always available or is difficult to create. This study focusses on the creation of behavioural models from data that was collected during a field exercise. As data in general is limited, noisy and erroneous, this makes the creation of realistic models challenging. Besides using the traditional approach of hand-crafting a model based on data, we investigate the emerging research area of imitation learning. One of its techniques, reward engineering, is applied to learn the behaviour of soldiers in an urban warfare operation. Basic, but realistic, soldier behaviour is learned, which lays the groundwork for more elaborate models in the future.

ARTICLE INFO

RECEIVED: 09 Oct 2021
REVISED: 10 Nov 2021
ACCEPTED: 30 Nov 2021
ONLINE: 12 DEC 2021

KEYWORDS

Behaviour, Model, Imitation Learning, Simulation, Military Data.

✉ Corresponding Author: Tel.: + 31 61 091 04 38 E-mail: maarten.schadd@tno.nl

I. The Why and How of Behavioural Models

The increasing use of simulation in education, training, analysis, and decision support, leads to a higher demand for behaviour models of military decision making. In addition to the need for an accurate simulation of the physical behaviour, such as tank movements or bullet/missile trajectories, also realistic tactical behaviour of simulated entities or vehicles is required. The decision process of these virtual participants is captured in a behaviour model. Behavioural models were first introduced in [1], and we define them as operational, conceptual, psychological or tactical models of the behaviour of human-like, human-controlled, or autonomously operating real-world systems.

Examples of such real-world systems can be a tank directed by a commander; a ship commanded by a captain; a fighter jet flown by a pilot; an unmanned aerial vehicle (UAV) controlled by a ground-based operator; or the human actor itself, e.g., a foot soldier. Furthermore, we do not restrict the size of the systems. We, for example, also consider a battalion of tanks, a flotilla of ships or a UAV swarm as suitable subjects for behavioural models. In military simulations, when the machine decides the actions of a unit or force, these systems are known as Computer Generated Forces (CGFs).

The development and application of new behaviour models is a complex process. A lack of interoperability methods and standards leads to a splintered landscape of models that are mostly used in a single simulation system only. Earlier work [2] investigated in which phase of the development effective reuse of behavioural models can be achieved, as well as what supporting processes, technologies and standards are needed. One conclusion was that there is much interest in this field of research, with ongoing developments of tools and standards, and that AI (Artificial Intelligence) and its power to create well-performing models will play a large role in various military applications. Another conclusion was that currently there is insufficient value in reusing behavioural models in different environments for the Dutch Ministry of Defence. Rather than reusing models, more efficient and effective modelling is desired. One way to achieve this, is to use state-of-the art techniques in the research field of artificial intelligence [3].

In machine learning applications, correct and incorrect examples of behaviour or decisions are presented to a learning system in the hope that the system can generalize the examples. This is called supervised learning [4], and its success depends on many factors (e.g., algorithms, size and type of data, and implementation technique). A problem for the usage of actual data in a military context is that data may be classified or simply unavailable, as the number of military conflicts is fortunately low.

A second common approach is deploying a behavioural model in a simulator and using the generated data to improve the models' parameters; and the most common approach is reinforcement learning [5]. A difficulty with reinforcement

learning is that the reward function has to be carefully crafted and any error in the simulator can be exploited and lead to learning undesired behaviour [6]. Such errors may occur in unforeseen circumstances that humans never encounter, but algorithms *do* due to their exploration of the search space in millions of simulations. Furthermore, an accurate simulator has to be developed first, as mistakes in the simulations can be exploited or lead to learning unrealistic behaviour [7]. In a military setting, the reinforcement learning approach is difficult but promising [8].

For supervised learning large quantities of high-quality data are required, for reinforcement learning a high-quality reward function and simulator are required, while many use cases exist in which neither is available. When not having large volumes of high quality data, or a simulator capable of creating such volumes, many techniques from the field of Artificial Intelligence are not applicable. In such situations it is not clear which approach leads to the best results with the least amount of effort. Therefore, this study aims at creating behavioural models that display realistic behaviour in an efficient manner, while having little data and no simulators available. For this purpose, methods from the research area of imitation learning [9] are employed. The focus of imitation learning is to explicitly train a model with the behaviour of an expert in a teacher-pupil setting.[1] The model has learned the behaviour correctly if it can imitate the behaviour of the teacher. We apply these techniques in our research for creating behavioural models for soldiers and Boxer vehicles that operate in an exercise of an urban warfare operation. The collected data was very limited, and there were no means of creating more data or being able to test the model in a simulator.

With this research we aim at behavioural models that can contribute to (1) creating new training scenarios in which the behaviour of the computer generated forces is used for creating better scenarios [10]; (2) supporting after action reviews by comparing the data generated by trainees to the correct behaviour model that was learned beforehand with our approach; (3) comparing the model behaviour for basic combat techniques with the behaviour of soldiers in the field. If the soldier behaviour seems more successful, this can lead to ideas for adapting the basic combat technique; (4) generating realistic simulated entity behaviour for synthetic wrapping [11, 12]; and (5) the realization of simulation-based decision support to the commander by using the learned behaviour for advising decision makers.

Section II investigates the requirements on the data that is needed for creating the behaviour models. In Section III the use case for this study is introduced. The traditional approach of hand-crafting models is presented in Section IV. We present background information of the emerging field of imitation learning in Section V, and

[1] In imitation learning the teacher usually involves a human that makes decisions. It is however possible to create behavioral models of autonomous systems as well. This may be useful when there is no access to the source code or when modelling the system on a much higher abstraction level.

its application to the use case in Section VI. Finally, Section VII provides concluding remarks.

II. Getting the right data, and getting the data right

In order to create effective behaviour models, one has to understand under what circumstances, with what intelligence and with what orders, the behaviour was recorded. The behaviour data and models have to match the decision process. We therefore first have to understand the military decision process, before starting to retrieve data.

A. How does a military expert take a decision?

Before any decision in a military setting can be made, the situation has to be understood. By analysing all available information, situational awareness (SA) is created [13, 14]. In military terms, SA is the commander's understanding of the battlefield [15]. SA can be categorized in 3 levels [16]. The first is a perception of the elements of the current situation, e.g., knowing the positions and status of own and enemy forces. The second is a comprehension of the situation. By understanding the dynamics of the physical elements and people in the situation, one can interpret the situation. For example, whether an enemy aircraft is on an attack flight path or on a reconnaissance mission. The third level is the projection of future status of the situation, e.g., that the enemy aircraft will deliver a dangerous payload to the aircraft carrier if not intercepted. Only when high levels of SA are achieved, effective decisions can be made [17]. With enough experience, decisions can be made in a split second as situations are recognized instantly. These are called recognition-primed decisions [18] and are in essence data-driven (experience) mental behavioural models. Technology to enhance the SA of a commander, directly contributes to better decisions being made [19].

When the situation becomes complex, systematic methods are followed in order to not overlook important information. An example of such a process is the Military Decision Making Process (MDMP) [20]. This is a lengthy process and is not suitable for decisions made in minutes or seconds on the battlefield. Another example is the NATO Comprehensive Operations Directive (NATO COPD) [21]. In all these cases the information of the environment is studied in detail to gain SA, and only after sufficient SA is gained a decision is made. We distinguish four levels of behaviour. On (1) strategic level, decisions are made based on (multi)national goals. On (2) operational level, decisions are made for conducting large operations or campaigns. Decisions in anticipation of, or during, combat are taken on (3) tactical level. The decisions for following procedures for the operation of mechanical platforms are part of the (4) technical level.

Depending on the current level of behaviour, the amount of information taken into account varies, as well as the speed of decision making. Although behavioural

models can be used for any level of behaviour from split-second decisions to large planning sessions, we restrict the scope to *tactical* behaviour in this article.

B. How are decisions made using behavioural models?

In its bare essence, a behavioural model follows the same steps as humans do when making a decision. A prominent framework is the OODA loop [22]. The four phases of this loop are Observe, Orient, Decide, and Act. The observe and orient phases serve the single purpose of gaining SA. This military model has been successfully used in various autonomous agents [23], and is applied in a large variety of situations [24, 25, 26].

A second framework is called BDI: Beliefs, Desires and Intentions [27, 28]. The basic BDI paradigm is widely used to achieve human-like intelligence in an agent-based approach, but often falls short of truly 'smart agents', since the agents lack ideal characteristics such as 'Coordination and Learning' [25]. BDI has been extended in [29] and is now widely used in practice. BDI can be used in the OODA loop steps, and is commonly applied to the Orient and Decide steps [30].

In both cases the creator of the model decides what relevant factors of the world are included in the so-called world-model and how these factors are allowed to interact with each other. If the interaction is strictly defined, then a more classical approach such as a rule engine or decision tree is used. If no sufficiently well-defined model can be created, then the machine may receive the task to learn the relevancy of factors based on data (e.g., with a neural network). For all cases it holds that if a factor was omitted, either by not including its definition by the designer of the model, or excluding the relevant data, it is impossible for the model to take it into account. Therefore, the performance of the model stands or falls with the creator's insights in the problem at hand.

After carefully designing, tuning or learning a model, the use of the model is straightforward. The designed factors are input to the model and are transformed by predesigned or learned steps to produce a desired output. As the number of methods and combinations for designing, tuning and learning is immense, various research disciplines have emerged to focus on research areas of efficiently creating models. Many of these research areas require data for the creation of the behavioural model.

C. Retrieving useful data

When talking about military data, the first thing that comes to mind is the classification and the limitations of sharing the data [31]. As the classification level of information often is restricted., e.g. national or NATO level, the amount of data that can be obtained is limited. This means that any research depends on having the appropriate clearances and having appropriate contacts within the data supplying community, which is usually the MoD. Before obtaining permission of receiving data, one has to know and define what kind of data is desired.

The best data for creating behavioural models originates from actual combat operations. However, not much data was recorded from actual combat operations and the data that was recorded often is not usable for the creation of behavioural models. It is not feasible to generate data for the purpose of research, as it would require engaging an opposing force. Using historical data can also be problematic, as military technology and doctrine change quickly and data for a desired context does not exist.

A logical way forward is the use of data that is gathered during training and exercises. Such a training can either be (1) executed in a simulated environment, using a constructive simulation such as in VR Forces, which can simulate troops of many sizes [32], or (2) be executed live in the field with actual soldiers. The promise of using actual data is that behaviour models can be created without the need of creating (complex) simulators to facilitate the training. In this research we hope to achieve this promise despite all the problems that using raw data brings, such as noise and missing context. As use-case, an exercise with the Mobile Combat Training Centre has been selected, as described in Section III.

III. Use-case: Urban warfare with the Mobile Combat Training Centre

The mobile combat training centre (MCTC) [33] was introduced in 2003 by the Dutch ministry of Defence and enables soldiers to practice combat in the actual environment in a realistic setting, but without using ammunition. Lasers and sensors are utilized to simulate firing weapons. The system keeps track of the location of soldiers and vehicles, used ammunition, and health status. A variety of weapons (e.g., rifles, heavy machine guns, indirect fire), vehicles (e.g., Fennek, Boxer) and terrain (e.g., cross-country, urban) can be included in the exercise. All data that the systems generate are logged so that it is available for the after-action review. Figure 1 shows a soldier training with the MCTC. Note the laser sensors on the helmet that register when a soldier is hit, and the laser on the gun that is used to shoot at opposing forces.

An exercise was selected that took place in the Dutch training village Marnehuizen, which was entirely built to train Military Operations on Urban Terrain [34]. Figure 2 shows an overview of the village. In the selected exercise, the Blue forces entered the village at the bridge in the north-east and were tasked to clear the village of enemy forces. A house-to-house battle was fought, which lasted two days, until the last houses on the west side of the village were declared free of enemies.

Figure 1: A soldier training in MCTC [35]

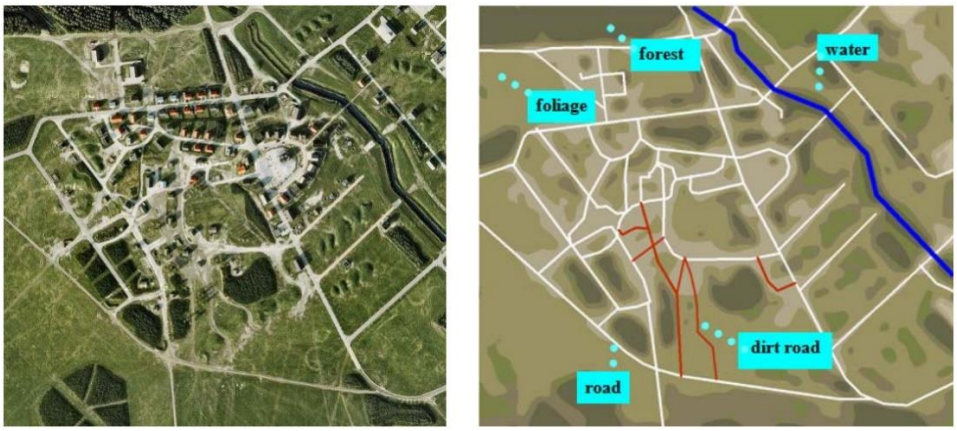

Figure 2: Terrain image of the training field for military operation in urban environments in Marnehuizen, The Netherlands. (Right) Parsed terrain map, semi-automatically derived from the left image [34].

The logged MCTC data contains the location of soldiers and vehicles at regular intervals. Also, fire events, hit events, kill events, and vehicle associations (when a soldier enters or exits a vehicle) are present in the data. This data can give a rough overview of the current state of the battlefield to the trainer. The consistency of the data is somewhat lacking in several aspects. Soldier locations are only provided every 15 seconds, and are snapped to a cell on a grid (with cell size of roughly 1m x 1m). The orientation of the soldiers is not reported. Sometimes soldiers move several grid cells at once, for example when driving quickly in a vehicle. It is also not always clear whether a soldier is inside or outside a building, as the wall of the building can run through the centre of such a grid cell. Other limitations include that it is not always clear what soldiers are firing at, and (un-) boarding vehicles is noisy. These limitations are not a problem for gaining a rough overview of the state of the operation for which the data was intended, but do form an additional hurdle for training models.

IV. HAND-CRAFTED BEHAVIOUR MODELS

A straightforward way to improve the realism of a military behavioural model is to create the structure of the model manually, and tune its parameters based on collected data. In this manner the expert stays in control of what the model can learn and the parameter tuning should be easy to perform. The created model can be seen as a method of combining data with expert knowledge. The model most often reflects a tactic or behaviour that is well-defined in the current doctrine, such as bounding overwatch [36, 37]. In such an approach, however, the model will never be smarter than its creator, as there exists no room for creativity in the man-made structure. When more freedom is given to the algorithm, more creativity can be observed, that can even surpass human performance [38, 39]. The hand-crafted models do, however, have the advantage of being highly understandable and explainable to military experts, as the structure of the model closely resembles the decision making process of an expert. Such a model can for instance be used in after action reviews by comparing the model generated from the data with the model of the correct behaviour and thus help the training instructor who only has limited time for analysing all the data from the training to brief the trainees. Large differences in the model parameters are indicators of learning points.

In this section, we want to show how a hand-crafted behavioural model can be created and tuned with data from the military exercise in Marnehuizen. The identified use case is the behaviour of a Boxer vehicle that supplies fire support for soldiers that perform house-to-house combat. The vehicle is called to the building, provides suppressive fire, and retreats so that it is not susceptible to anti-armour ammunition for a long time. A schematic overview of this behaviour is shown in Figure 3.

The behaviour displayed in Figure 3 has to be abstracted into a model. In this study, we purely consider the timing aspect. Other aspects, such as the relative position between Boxer and infantry, or between Boxer and building, are left for future work. We distinguish between five steps:

1. The time needed for the Boxer to move into a firing position.
2. The time that the Boxer provides suppressive fire before the infantry starts moving.
3. The time needed for the infantry to move to the building.
4. The time interval between the infantry arriving at the building and the Boxer departing.
5. The time needed to clear the building and restarting at step 1.

Figure 3: A schematic overview of the fire support provided by a Boxer vehicle. (1) Top-left: The initial position with the Boxer shown in red, and an infantry group in blue. (2) Top-right: The Boxer vehicle approaches the building in the lower-right corner and provides suppressing fire. (3) Lower-left: The infantry approaches the building. (4) Lower-right: The Boxer vehicle retreats.

For this study, we focus on steps 2 and 3. In order to determine these parameters, it is essential to know when the Boxer and the infantry arrives at the building. The other parameters can be derived with similar approaches as described below.

In the exercise, several buildings are approached as shown in Figure 3, and each iteration of this procedure can be analysed. Annotating the locations of buildings and when such an iteration starts and ends, based on the position of the Boxer vehicle and the infantry group, is done manually, and is already a challenging task. As there are multiple vehicles, the first question is: which Boxer is currently providing suppressive fire? Is the Boxer actually firing at the selected building? Fire events are part of the data set, but when the shot does not connect with a hit event, it is not known in what direction the shot was fired. Specially with suppressive fire, most shots do not hit any sensor that could register the firing direction. This makes it guesswork whether the Boxer is providing suppressive fire on the building, or

firing at something else. Also, the movement of the infantry group is not trivial. The groups that move from building to building are not defined as groups in the order of battle (Orbat): they are selected on the spot from the available soldiers in the platoon (which *is* defined in the Orbat) and altered for each iteration. In order to be able to measure the effectiveness of any algorithm that has to learn the behaviour of (groups of) soldiers and supporting vehicles, the dataset was manually annotated by selecting Boxer vehicles that provide fire support, and the timesteps when the infantry cleared a building.

From the algorithmic point of view, we define the moment that the Boxer arrives at the scene to provide fire support as the timestep at which the vehicle is located closest to the building. An example of how the distance of a Boxer vehicle changes over time is shown in Figure 4. The large peak at the beginning of the exercise is because the Boxer is parked at a large distance while not actively participating.

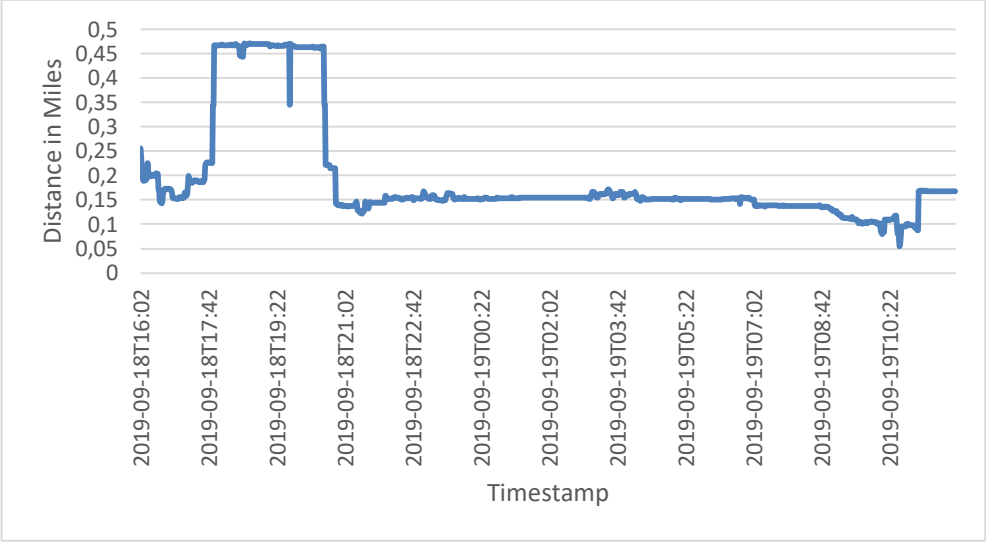

Figure 4: The distance of a Boxer (y-axis, measured in miles) to a target building over time (x-axis).

The smallest distance of a Boxer vehicle to the building is chosen as the beginning of the fire support. This measure may be faulty, as driving past the building after it is cleared may reduce the distance even further, but it is a straightforward computation. Figure 5 shows the absolute difference between the calculated and

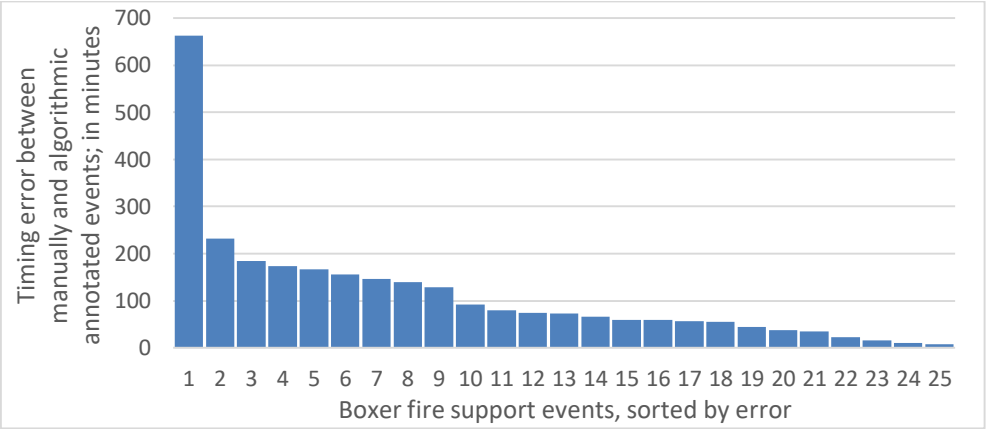

Figure 5: The difference to the manually annotated events, measured in minutes, made by selecting the closest distance of the Boxer to the building. Each building is a separate column, and the columns have been sorted by error (i.e., the building in column 1 has the highest error, and the building of column 25 is the building with the lowest error)

manually annotated events[2], measured in minutes. In the worst case, the algorithm is more than 600 minutes wrong. As the exercise takes two days, and there is no movement in the night, choosing a moment on the wrong day gives a large error. It can be concluded that this approach for detecting when a Boxer provides fire support is not very accurate.

For detecting when infantry is clearing a building, a slightly different approach can be taken. As groups of soldiers that clear a building are shuffled regularly during the exercise, we have to find in the data which (sub)groups of soldiers are actually clearing which building. For this, we define the moment of clearing as that moment when X soldiers are within Y meters of the building, and the parameters X and Y should be carefully chosen. Note that any X soldiers of the Blue forces, independent of their assignment in the order of battle, are sufficient for triggering this condition. For each building, a different set of soldiers can trigger the condition. The parameters X and Y can be chosen by using the provided data, as can be seen in Table 1. The best results are achieved by selecting the timestamp at which 5 soldiers are within a 15 meters radius of the building. Figure 6 shows the error obtained with this setting for each building.

[2] The moment of having a building cleared was chosen manually based on an overview of the battlefield. When enough soldiers entered the building or are located in its vicinity, with no enemy troops close by, the timestamp was recorded. It has to be said that finding a good timestamp for each building was not an easy task by using this data.

Table 1 The average difference in minutes of detecting when soldiers clear buildings to the manually annotated timestamps; for different number of soldiers and distance parameters. (x) indicates that x times the clearing of a building was not detected with that setting, as it did not occur that the needed number of soldiers was close enough to the building during the exercise. The number represents the average error made on the 26 buildings in the exercise.

Distance (right) Soldiers (down)	5m	10	15	20	25	30	35	40	45	50
1	57	77	149	185	197	226	276	310	382	384
2	9	33	69	76	84	115	139	150	157	214
3	17 (3)	5	58	67	78	85	99	130	141	170
4	24 (4)	6	5	58	72	82	89	99	109	144
5	33 (10)	10	3	31	62	78	85	95	105	113
6	27 (15)	14 (2)	5	6	59	75	81	91	101	108
7	33 (17)	22 (2)	7	6	56	73	78	87	97	107
8	23 (21)	24 (3)	8	7	57	65	77	84	94	104

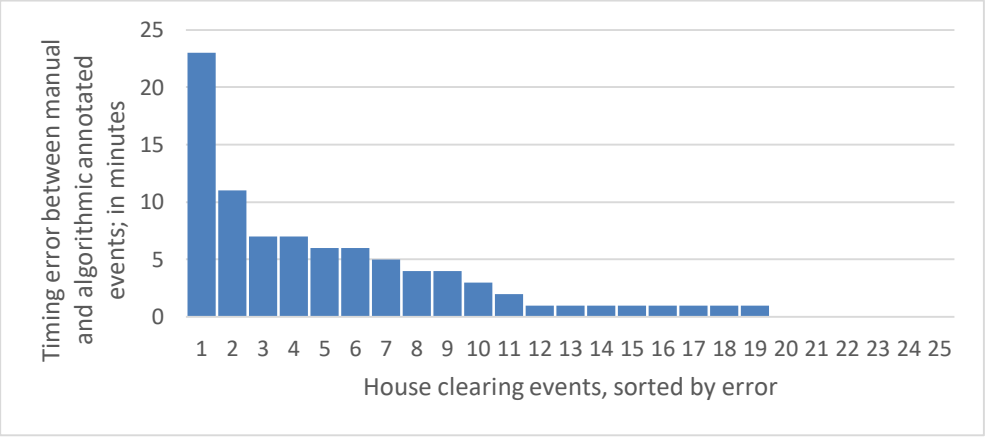

Figure 6 The difference to the manually annotated events, measured in minutes, made by selecting 5 soldiers with a 15-meter radius of the building. Each building is a separate column, and the columns have been sorted by error (i.e., the building in column 1 has the highest error, and the building of column 25 is the building with the lowest error)

This section showed that it is possible, but not easy to tune expert models with military data. The main challenge is that there is a mismatch between the level of behaviour for which data is logged and the level that we are trying to model (see Section II A). The data is logged on the technical level (e.g., shots being fired without knowing the firing direction) and the decisions we are trying to model are on the tactical level (e.g., clearing the building). If the data would have been created on

the tactical level (e.g., timestamps of clearing a building), as well as being more precise and consistent, expert models could much more easily be created. Automatically enriching technical-level data in the data acquisition step with tactical information is a challenging topic on itself. We now have created two models that contribute the boxer fire-support doctrine (see Figure 3). Several more models are needed in order to complete the boxers doctrine, but as it is hard to create expert models from military data, we decided to investigate an entirely different approach: imitation learning.

V. THE PROMISE OF IMITATION LEARNING

Imitation learning techniques attempt to mimic human behaviour for a given task [9, 40]. These techniques fall in the broader category of observational learning. In observational learning in general, the original behaviour does not have to be created by a willing or knowing participant [41]. Imitation learning can be viewed as a special case of observational learning where the purpose of learning is to reproduce exactly the same actions as the original in identical situations, and realistic behaviour for previously unseen situations. Imitation learning is closely related to learning from demonstration in which a human purposely demonstrates how to perform the task in order to make the agent perform the same task [42, 43]. The term learning from demonstration is often used in robotics [44, 45, 46, 47].

In addition to its broad application in robotics, imitation learning is also being applied to simulators and games. The actions of the player can in this manner be recorded easily, and the simulator or game can be used for training purposes [48, 49, 50, 51]. Some applications focus on imitating the exact player behaviour in order to use the learned behaviour for other purposes. In [52] for example, the behaviour of players on a race track is learned so that new tracks can be tested using the models, rather than having human play testers. Other work focuses on using the examples by humans to create super-human performance [53, 54, 55].

Imitation learning can roughly be grouped into three categories. (1) In the most basic form, one has a labelled set of states. The labels are the action that the human chose in given state. Now the problem can be approached as a supervised learning task, similar to classification tasks. This approach is known as behaviour cloning [47]. Behaviour cloning does not require access to a simulator. (2) When one does have access to a simulator, and therefore the state-transition kernel, we speak of direct policy learning [50]. In this category it is known what the available actions of an actor are in each state, and a transition policy can be learned. The transition policy chooses out of all available actions the most desirable one. (3) When one is interested in learning the weights of the value of state properties that a human uses when evaluating future states, we speak of inverse reinforcement learning [56]. These methods typically use the transition kernel to look at possible future states

in order to create an explainable evaluation function of a state that resembles the preferences of the human demonstrator.

The main difference between handcrafted models and imitation learning is the degree of freedom that the algorithm has to correctly reproduce behaviour. In the hand-crafted models of the boxer that provides fire support, we chose that the distance is the most discriminating factor for the decision that fire support is currently being provided. The only parameter to tune is the distance threshold. In an imitation learning setting, the algorithm is being provided with all state information and is given the freedom to decide itself what the most relevant features are. Such approaches are particularly successful in domains where it is hard to create a well-fitting model manually [57].

VI. IMITATING THE MILITARY EXPERT

Imitation learning has also seen several applications in the military domain [58, 59]. For example in [60], imitation learning is applied to learning decision policies of computer generated forces. The learned behaviour can afterwards be used for the training of soldiers in the simulator [39].

What the beforementioned research has in common, is that a human-in-the-loop simulator is used for collecting human examples. It is exactly known what the current state is, what the possible actions are and what the next state will be after taking an action. This makes the creation of behavioural models possible. In the case of the MCTC data, however, only the state information is available, and there is no knowledge about the currently available actions, or what the information position of a soldier is. For example, only the position of a soldier is known, and not the direction that the soldiers is facing or what potential actions the soldier is considering. This problem is defined in the literature as *Imitation from observation (Ifo)* [61]. Ifo can be further subdivided into model-based and model-free. In model-based, either a translation has to be learned from state to actions, or from state-action pairs to the next state. The MCTC use-case falls in the model-free category. Within this category, we can further distinguish into (1) adversarial methods that use a simulator to collect data and compare the data to the expert demonstrations, and (2) reward engineering [62], which learns a reward function for states. Typical examples are learning a task by watching video images of a person performing the desired task [63, 64].

As no executable simulator is available for MCTC, only reward engineering is a viable option for the MCTC use case. We develop a system that when given the current state of the engagement, is able to predict the state a certain number of seconds in the future. This is closely related to [65], which use the difference between the predicted state and the actual state as reward function in a reinforcement learning setting. The main difference is that no reinforcement learning can be done with the MCTC data as no simulator is available.

We have to define what 'state' means in terms of MCTC. The collected data holds the complete data, of all soldiers and vehicles, of blue and red forces. If the entirety of the engagement is seen as state (i.e., the states of all players and of everything in the environment), then there are astronomically many next states possible, as for instance each soldier or vehicle can move in any direction. It is also not the case that a soldier decides on his/her own action with all global information, but rather with his or her own local information. We therefore simplify the state definition to the local surroundings of a soldier, and try to predict the next position of the soldier.[3] Although there is much more to the state of a soldier, such as shooting status, health status, current posture, we currently only focus on predicting the next position in order to evaluate the suitability of reward engineering and the suitability of the data provided by the MCTC.

Surrounding state features are abstracted into a grid, and each combination of grid cell and feature is an input for the decision. The soldier that makes the decision is located in the centre of the grid. It is possible that a real soldier takes information outside the grid into account (e.g., when visibility is good, or when receiving information over radio), but we only take information into account that falls inside the grid cells. It is also possible that too much information is currently taken into account, as information is included that is not in line of sight (e.g., when a building stands in the way). Various features can be added that soldiers may consider: location of rivers, time of day, current mission, munition left, current health, actions taken in the past, etc. The closer this resembles how soldiers actually reason, the more accurate the learning result is expected to be.

In our setup we use a grid of 8x8 with a real world size of 83 meter by 83 meter each, as shown in in Figure 7. We take the vicinity of friendly and enemy soldiers into account. In the state of Figure 7 there is 1 friendly soldier in the cell north-west of the soldier, while all others hold 0 friendly soldier, and there is an enemy soldier in the south-west. The soldier that is located outside the grid is not taken into account. We also take into account what action was taken in the past (i.e., the locations in the past three episodes). This input grid is duplicated and filled for each of the three historical episodes. We choose to take episode steps of 15 seconds, as that matches the rate at which data is collected at MCTC. Any shorter is not useful as then no new location is communicated between episodes.

As supervised learning target, a grid of 3 by 3 is used with cells being 2 meters wide and tall, as shown in Figure 8. The cell size matches the resolution at which data is logged. The grid has a 1 on the location that the unit moved to, and 0s

[3] One can argue that soldiers never operate alone, but are always is part of a fire team of 4 or fewer people. Therefore a behaviour model for a fire team could be more suitable. As in this exercise fire-team compositions change frequently, and it was not logged what the current fire teams are, learning individual soldiers behaviours is more promising.

elsewhere. In the case when the next known position of the unit is outside the grid, the closest grid position is chosen as target.

Figure 7: The input grid of local features for making the decision. Each cell is 83x83 meters and we count the number of friendly and enemy soldiers in each cell. The input grid of the last three episodes forms the input of the neural network.

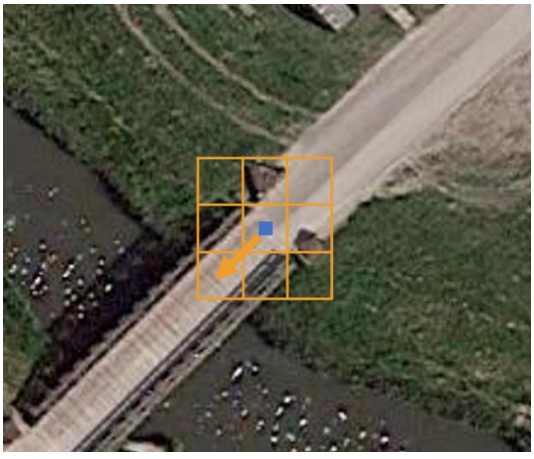

Figure 8: The output of the decision. A 3x3 grid of movement locations, each cell is 2x2 meters. The arrow indicates that according to the MCTC data the position of the

soldier after 15 seconds was in the bottom left grid cell, and this cell is used as supervised label of the situation.

We train a fully connected neural network with 3 hidden layers, and 100 hidden neurons per layer. We use rectified linear unit activation functions and the mean squared error as loss function. An interesting discussion is how to evaluate the performance of the created neural network. Although a small difference in the predicted location does not seem bad, a series of small differences can cumulate to a large difference later on. At the same time, a decision can bring the soldier into a different context (e.g. passing the building on the left or the right side). The actions following this decision point can differ greatly (e.g. taking cover when going left, compared to successfully clearing the building when going right). We therefore cannot evaluate the realism of the behaviour of the soldier unless the exact position and state has occurred in the original data.

We therefore measure the realism of the learned behaviour in two ways. (1) Based on the original data, the precision and recall on the test set are used, which are quantitative measures. (2) We replay the military exercise in which one or several units are controlled by the learned model and judge its behaviour. All other units are placed and moved using the original data. This provides insight on the learned behaviour, which is a qualitative measure.

Table 2 shows standard measures for supervised learning methods: accuracy, precision, recall and f1-score. Keep in mind that there are 9 output cells, and the probability of randomly guessing correct is 0,11 and that in this setting all four measures are expected to have a value around 0,11 for random guessing. The training set has been balanced so that each output cell had an equal number of examples. Table 2 indicates that the accuracy is higher than random guessing, but still far away from predicting the next state consistently.

Table 2 Quantitative measures for predicting the next soldier state.

Measure	Value
Accuracy	0,2220
Precision	0,2558
Recall	0,2210
F1 Score	0,2169

For analysing the behaviour of the learned model, we place a single soldier that is controlled by the model in the exercise. **Error! Reference source not found.** shows the movement path of a soldier that is created by the model, compared to that of the original. Here we see that the neural network roughly moves in the same manner as the original soldier moved. Although not visible in **Error! Reference source not found.**, also the timing is roughly the same. This example also highlights

the difficulty of working with this data. The locations of the original soldier (green) sometimes make large jumps (e.g. the first data point in the east has no neighbour close by).

Figure 9: Comparing the movement of the model with the actual movement. The highlighted blue locations are the soldier that is steered by the neural network. The highlighted green locations are the actual locations of the original soldier. The starting point of both was at the eastern side of town, and both moved gradually to the west.

By analysing several of these traces, we can conclude that the model learned two behaviour traits that resemble actual soldier behaviour. (1) It is beneficial to stay close to friendly soldiers. Soldiers often move as a group, and the model usually chooses to move towards friendly units. (2) When the historical movements are heading in one direction, the probability that the next movement is also in that direction is high. As soldiers have a certain task, clearing a building, it makes sense that soldiers keep moving towards the objective until reaching it. Although these traits make sense, they also create unrealistic behaviour in certain situations. (1) When multiple soldiers are controlled by the model, they tend to stick to each other and stop moving. Artificial soldiers do not want distance themselves from each other. (2) When a model controlled soldier enters a territory in which there are no friends or foes, it tends to keep walking in the same direction until exiting the battlefield. As the prediction is dominated by recent historical movements, and all other inputs are 0, the model decides to keep walking in the same direction. One of the causes for this is that the current task is not part of the input features.

We argue that this result shows that a first step is made towards automatically creating a model of a soldier's decision making process based on the method of

reward engineering. Although only basic behaviour is currently learned, we foresee that more complex patterns can emerge when more types of inputs, such as terrain characteristics and orders, are included in the learning process.

VII. Conclusions

This article investigated the possibilities for creating behavioural models of units using military decision making in a data-driven manner. We showed that it is possible to tune parameters of models that are created by subject-matter experts with military data, but even when data is annotated manually it is not straightforward to do so. As the data is collected with other goals in mind, important behavioural context is not available, which hinders the efficient use of the data for our purposes. We investigated the emerging research area of imitation learning and applied it to the use case of learning to predict soldier movement during an urban building clearing exercise. Such techniques may not only recreate realistic soldier behaviour in identical situations, but also may generalise the behaviour to obtain realistic behaviour in previously unseen situations. While the research area knows many subareas, only reward engineering seems currently applicable when neither having a simulator available, nor the possibility to retrieve the set of possible actions in a state to learn an action policy. We demonstrated the method of reward engineering by trying to predict the next state of a soldier based on local state information. Two basic soldier behaviour traits have been learned by the neural network, which in some situations create realistic behaviour, while in other situations illogical behaviour is displayed. We argue that the illogical behaviour can still be improved upon with additional feature inputs.

Our overall conclusion is that imitation learning methods seem very promising for creating behaviour models of military decision making. If successful, the behavioural models that are created in this fashion can be beneficial to the military in several ways. Think for instance of contributing to creating new training scenarios in which the behaviour of the computer generated forces is improved, supporting after action reviews by comparing the trainees' behaviour to the learned correct behaviour, assisting in comparing and possibly adapting basic combat procedures to the behaviour displayed in the field, supporting synthetic wrapping where simulated entities can display accurate behaviour. Depending on the accuracy of the developed models, some applications may be easier to support than others. For example, may the demands on accuracy be higher in a decision support setting compared to a synthetic wrapping setting.

In the future, we want (1) to create automatic methods for pre-processing the MCTC data by creating additional context on the tactical level. Methods such as estimating the current point of view, or what (type of) order is currently executed come to mind. This additional context can then help to improve parameter tuning of models. (2) We want to improve the feature set of the reward engineering

approach in order to make the behaviour more realistic. (3) We would like to explore explainable learning methods in order to make the learned behaviour more explicit. The explanation can then be used for various purposes, such as after action reviews.

ACKNOWLEDGEMENTS

This research contributes to the research programme V/L1801 AIMS (AI for Military Simulation) by researching methods to efficiently and effectively creating military behavioural models that can be used to explain and simulate of (human and entity) behaviour.

REFERENCES

[1] I. Ajzen and M. Fishbein, "Attitude-Behavior Relations: A Theoretical Analysis and Review of Empirical Research," *Psychological Bulletin,* vol. 84, no. 5, pp. 888-918, 1977. http://dx.doi.org/10.1037/0033-2909.84.5.888

[2] N. de Reus, J. Voogd and J. v. Oijen, "Model Reusability - phase-1: survey (TNO 2020 R10939)," TNO, The Hague, 2020.

[3] J. Roessingh, A. v. O. J. Toubman, G. Poppinga, M. Hou and L. Luotsinen, "Machine Learning Techniques for Autonomous Agents in Military Simulation," in *IEEE Interactional Conference on Systems, Man, and Cybernetics (SMC)*, 2017.

[4] S. J. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[5] R. Sutton and A. Barto, Reinforcement Learning: An Introduction, Cambridge, MA: MIT Press, 1998.

[6] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew and M. Igor, *Emergent Tool Use From Multi-Agent Autocurricula,* arXiv:1909.07528, 2019.

[7] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew and I. Mordatch, "Emergent Tool Use From Multi-Agent Autocurricula," *arXiv preprint arXiv:1909.07528,* 2019.

[8] A. Toubman, G. Poppinga, J. J. Roessingh, M. Hou, L. Luotsinen, R. A. Løvlid, C. Meyer, R. Rijken and M. Turčaník, "Modeling CGF Behavior with Machine Learning Techniques," in *Proceedings of the 2015 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*, National Training and Simulation Association, 2015, pp 2637-2647.

[9] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in Cognitive Sciences,* vol. 3, no. 6, pp. 233-242, 1999. http://dx.doi.org/10.1016/S1364-6613(99)01327-3

[10] N. Abdellaoui and A. P. G. Taylor, "Comparative Analysis of Computer Generated Forces' Artificial Intelligence," in *RTO-MP-MSG-069 - Current uses of M&S Covering Support to Operations, Human Behaviour Representation, Irregular Warfare, Defence against Terrorism and Coalition Tactical Force Integration*, NATO, 2009.

[11] NATO MSG-098, "STO-TR-MSG-098: Urban Combat Advanced Training Technology Architecture," NATO, 2018.

[12] NATO MSG-099, "STO-TR-MSG-099: Urban Combat Advanced Training Technology Standards," NATO, 2018.

[13] M. Endsley, "Situation awareness and human error: designing to support human performance," in *Proceedings of the High Consequence Systems Surety Conference*, SA Technologies, Albuquerque, NM, 1999, pp 2-9.

[14] W. Howell, "Engineering psychology in a changing world," *Annual Review of Psychology,* vol. 44, pp. 231-263, 1993. http://dx.doi.org/10.1146/annurev.ps.44.020193.001311

[15] Department of the army, "Field Manual 101-5," Staff Organization and Operations., Washington, DC, 1997.

[16] M. Endsley, "Theoretical Underpinnings of Situation Awareness: A Critical review," in *Situation Awareness Analysis and measurement*, M. R. Endsley and D. J. Garland, Eds. Lawrence Erlbaum Associates, Mahwah, NJ, 2000, pp 3-32.

[17] A. Wellens, "Group situation awareness and distributed decision making: from military to civilian applications," in *Individual and Group Decision Making*, N. Castellan Jr., Ed. Lawrence Erlbaum Associates, Mahwah, NJ, 1993, pp. 267-291.

[18] G. A. Klein, "A recognition-primed decision (RPD) model of rapid decision making," in *Decision making in action: Models and methods*, G. A. Klein, J. Orasanu, R. Calderwood and C. E. Zsambok, Eds. Ablex, Norwood, NJ, 1993, pp. 138-147.

[19] J. P. Holmquist and S. L. Goldberg, "Dynamic Situations: The Soldier's Situation Awareness (NATO RTO-TR-HFM-121-Part-II)," NATO, 2007.

[20] Department of the Army, "Field manual 6.0 - mission command: command and control of army forces," Washington, DC, U.S., 2003.

[21] Supreme Headquarters Allied Powers Europe (SHAPE), "Allied Command Operations Comprehensive Operations Planning Directive COPD interim v2.0," NATO, 2013.

[22] J. Boyd, "A discourse on winning and losing: Air University Library Document No. M-U 43947," Maxwell Air Force Base, AL, 1987.

[23] C. Heinze, S. Goss, T. Josefsson, K. Bennett, S. Waugh, I. Lloyd, G. Murray and J. Oldfield, "Interchanging agents and humans in military simulation," *AI Magazine,* vol. 23, no. 2, p. 37–47, 2002.

[24] M. Plehn, “Control warfare: Inside the OODA loop (Master's Thesis),” Maxwell Airforce Base School of Advanced Airpower Studies, Maxwell AFB, AL, 2000.

[25] J. Tweedale, N. Ichalkaranje, C. Sioutis, B. Jarvis, A. Consoli and G. Phillips-Wren, “Innovations in multi-agent systems,” *Journal of Network and Computer Applications,* vol. 30, no. 3, p. 1089–1115, 2007. http://dx.doi.org/10.1016/j.jnca.2006.04.005

[26] B. Clough, “Metrics, schmetrics! How the heck do you determine a UAV’s autonomy anyway?,” in *Proceedings of the Performance Metrics for Intelligent Systems Workshop*, Gaithersburg, Maryland, 2002.

[27] M. Bratman, Intention, plans, and practical reason, Cambridge, Mass: Harvard University Press , 1987.

[28] M. Georgeff and F. Ingrand, “Decision-making in an embedded reasoning system,” in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, N. S. Sridharan, Ed. Morgan Kaufmann Publishers, Incorporated, San Francisco, CA, 1989, pp. 972-978.

[29] A. S. Rao and M. P. Georgeff, “BDI Agents: From Theory to Practice,” in *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS-95)*, V. Lesser, Ed. MIT Press, San Francisco, USA, 1995, pp. 312-319.

[30] H. S. Nwana, “Software Agents: An Overview,” *The Knowledge Engineering Review,* vol. 11, no. 3, pp. 205-244, 1996. http://dx.doi.org/10.1017/S026988890000789X

[31] NATO RTG-051, “NATO Guide to Data Collection and Management for Analysis Support to Operations,” NATO, 2020.

[32] VT MAK, “What is Aggregate-Level Simulation Anyway?,” *What’s Up MÄK,* vol. 17, no. 5, pp. 2-2, 2015.

[33] S. Groen, “De Genie versus het Simulatiecentrum Landoptreden,” *Promotor,* vol. 39, no. 4, pp. 33-39, 2015.

[34] J. de Jong, G. Burghouts, H. Hiemstra, A. te Marvelde, W. van Norden and K. Schutte, “Hold your fire!: Preventing fratricide in the dismounted soldier domain,” in *Proceedings of the 13th International Command and Control Research and Technology*, Bellevue, WA, USA, 2008.

[35] A. de Boer, “Naar een hoger level,” *Landmacht,* vol. 13, no. 3, 2015.

[36] F. Kamrani, L. Luotsinen and R. Løvlid, “Learning objective agent behavior using a datadriven modeling approach,” in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, Piscataway, NJ, 2016, pp. 2175-2181. https://doi.org/10.1109/SMC.2016.7844561

[37] J. J. Roessingh, A. Toubman, J. v. Oijen, G. Poppinga, R. A. Løvlid, M. Hou and L. J. Luotsinen, “Machine learning techniques for autonomous agents

in military simulations - Multum in parvo," in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, Piscataway, NJ, 2017, pp. 3445-3450. http://dx.doi.org/10.1109/SMC.2017.8123163

[38] L. J. Luotsinen, F. Kamrani, P. Hammar, M. Jandel and R. A. Løvlid, "Evolved creative intelligence for computer generated forces," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, Piscataway, NJ, 2016, pp. 3063-3070. http://dx.doi.org/10.1109/SMC.2016.7844707

[39] T.-H. Teng, A.-H. Tan and L.-N. Teow, "Adaptive computer-generated forces for simulator-based training," *Expert Systems with Applications,* vol. 40, no. 18, p. 7341–7353, 2013. http://dx.doi.org/10.1016/j.eswa.2013.07.004

[40] A. Hussein, M. M. Gaber, E. Elyan and C. Jayne, "Imitation Learning: A Survey of Learning Methods," *ACM Computing Surveys,* vol. 50, no. 2, pp. 1-35, 2017. http://dx.doi.org/10.1145/3054912

[41] S. Ontañón, J. L. Montana and G. A. J., "A dynamic-bayesian net-work framework for modeling and evaluating learning from observation," *Expert Systems with Applications,* vol. 41, no. 11, p. 5212–5226, 2014.

[42] Y. Duan, M. Andrychowicz, B. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel and W. Zaremba, "One-Shot Imitation Learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Eds. Curran Associates, Red Hook, NY, 2017, pp. 1087-1098.

[43] C. Finn, T. Yu, T. Zhang, P. Abbeel and S. Levine, "One-Shot Visual Imitation Learning via Meta-Learning," in *Proceedings of Machine Learning Research*, S. Levine, V. Vanhoucke and K. Goldberg, Eds. PMLR, 2017, pp. 357-368.

[44] C. G. Atkeson and S. Schaal, "Learning tasks from a single demonstration," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA97)*, IEEE, Piscataway, NJ, 1997, pp. 1706-1712. http://dx.doi.org/10.1109/ROBOT.1997.614389

[45] D. C. Bentivegna, "Learning from Observation Using Primitives," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, Georgia, 2004.

[46] B. D. Argall, S. Chernova and M. Veloso, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems,* vol. 57, no. 5, p. 469–483, 2009. http://dx.doi.org/10.1016/j.robot.2008.10.024

[47] D. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," (Technical Report CMUCS-89-107), Carnegie Mellon University, Pittsburgh, Pennsylvania, 1989.

[48] B. Gorman, C. Thurau, C. Bauckhage and M. Humphrys, “Bayesian imitation of human behavior in interactive computer games,” in *Proceedings of the International Conference on Pattern Recognition (ICPR’06)*, IEEE, Piscataway, NJ, 2006, pp. 1244-1247. http://dx.doi.org/10.1109/ ICPR.2006.317

[49] S. Priesterjahn, O. Kramer, A. Weimer and A. Goebels, “Evolution of reactive rules in multi player computer games based on imitation,” in *International Conference on Natural Computation (ICNC 05)*, Springer, New York, NY, 2005, pp. 744-755. http://dx.doi.org/10.1007/11539117_105

[50] J. Ho and S. Ermon, “Generative Adversarial Imitation Learning,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg I. Guyon and R. Garnett, Eds. Curran Associates, Red Hook, NY, 2016, pp. 4565-4573.

[51] K. Judah, A. Fern and T. G. Dietterich, “Active Imitation Learning via Reduction to I.I.D. Active Learning,” in *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, AUAI Press, Corvallis, OR, 2012, pp. 428-437.

[52] J. Togelius, R. De Nardi and S. M. Lucas, “Making racing fun through player modeling and track evolution,” in *Proceedings of the SAB'06Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, G. N. Yannakakis and J. Hallam, Eds. University of Southern Denmark, Odense, Denmark, 2006, pp. 61-70.

[53] S. Ross, G. Gordon and A. Bagnell, “A reduction of imitation learning and structured prediction to noregret online learning,” in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson and M. Dudík, Eds. PMLR, 2011, pp. 627-635.

[54] G. Li, M. Mueller, V. Casser, N. Smith, D. L. Michels and G. B., “Oil: Observational imitation learning,” in *Robotics, Science and Systems*, A. Bicchi, H. Kress-Gazit and S. Hutchinson, Eds. University of Freiburg, Freiburg im Breisgau, Germany, 2019. http://dx.doi.org/10.15607/ RSS.2019.XV.005

[55] G. Stein and A. J. Gonzalez, “Building high-performing human-like tactical agents through observation and experience,” in *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, IEEE, Piscataway, NJ, 2011, pp. 792-804. http://dx.doi.org/10.1109/TSMCB.2010.2091955

[56] P. Abbeel and A. Y. Ng, “Apprenticeship learning via inverse reinforcement learning,” in *Proceedings of the 21st International Conference on Machine Learning*, Association for Computing Machinery, New York, NY , 2004, pp. 1-8. http://dx.doi.org/10.1145/1015330.1015430

[57] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton and Y. Chen, "Mastering the game of go without human knowledge," Nature, vol. 550, no. 7676, pp. 354-359, 2017. http://dx.doi.org/10.1038/nature24270

[58] H. Nguyen, M. Garratt, L. Bui and H. Abbass, "Supervised deep actor network for imitation learning in a ground-air UAV-UGVs coordination task," in *IEEE Symposium Series on Computational Intelligence (IEEE SSCI)*, IEEE, Piscataway, NJ, 2017, pp. 1-8. http://dx.doi.org/10.1109/SSCI.2017.8285387

[59] B. Park and H. Oh, "Vision-Based Obstacle Avoidance for UAVs via Imitation Learning with Sequential Neural Networks," *International Journal of Aeronautical and Space Sciences,* vol. 21, no. 3, p. 768–779, 2020. http://dx.doi.org/10.1007/s42405-020-00254-x

[60] G. Berthling-Hansen, E. Morch, R. A. Løvlid and G. O. E., "Automating Behaviour Tree Generation for Simulating Troop Movements (Poster)," in *2018 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, IEEE, Piscataway, NJ, 2018, pp. 147-153. http://dx.doi.org/10.1109/COGSIMA.2018.8423978

[61] F. Torabi, G. Warnell and P. Stone, "Recent Advances in Imitation Learning from Observation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, International Joint Conferences on Artificial Intelligence Organization, San Jose, CA, 2019, pp. 6325-6331. http://dx.doi.org/10.24963/ijcai.2019/882

[62] A. Gupta, C. Devin, Y. Liu, P. Abbeel and S. Levine, "Learning Invariant Feature Spaces to Transfer Skills with Reinforcement Learning," in *5th International Conference on Learning Representations, ICLR 2017*, OpenReview.Net, 2017.

[63] Y. Liu, A. Gupta, P. Abbeel and S. Levine, "Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NY, 2018, pp. 1118-1125. http://dx.doi.org/10.1109/ICRA.2018.8462901

[64] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine and G. Brain, "Time-Contrastive Networks: Self-Supervised Learning from Video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Piscataway, NY, 2018, pp. 1134-1141. http://dx.doi.org/10.1109/ICRA.2018.8462891

[65] D. Kimura, S. Chaudhury, R. Tachibana and S. Dasgupta, "Internal Model from Observations for Reward Shaping," *ArXiv,* vol. abs/1806.01267, 2018.