AFRL-AFOSR-JP-TR-2024-0018

Autonomous Coordination Policies in Ground-Air Unmanned System Interaction

Abbass, Hussein
University of New South Wales-NEW NCAGE Code
HIGH STREET
KENSINGTON, NSW, ,
AU

12/12/2023
Final Technical Report

Controlled By: USAF AFOSR
CUI Category:
Distribution Statement:
POC: JERMONT CHEN

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
20231212	Final	START DATE 20170925	END DATE 20190924

4. TITLE AND SUBTITLE
Autonomous Coordination Policies in Ground-Air Unmanned System Interaction

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
	FA2386-17-1-4054	61102F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Hussein Abbass

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of New South Wales-NEW NCAGE Code
HIGH STREET
KENSINGTON, NSW
AU

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD UNIT 45002 APO AP 96338-5002	AFRL/AFOSR IOA	AFRL-AFOSR-JP-TR-2024-0018

12. DISTRIBUTION/AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Humans use a variety of complex strategies when solving problems. These strategies are semantically ordered; that is, as the complexity of a problem increases, an explanation of a strategy used for a simpler problem would be related to that of a strategy used for a more complex problem. Machines, however, can learn complex strategies but they normally produce strategies that are different from those used by humans. This leads to loss of trust and ineffective human-machine teaming. We designed machine learning algorithms that can learn incrementally while preserving the meanings embedded in a simpler context as they learn a more complex one. One proposed apprenticeship bootstrapping algorithm has been tested in simulation and physical environments on ground-air interaction tasks, while the others were either demonstrated theoretically or tested on simulated synthetic tasks alone. The research has demonstrated that machine education and bootstrapping techniques should be integral parts of explainable and interpretable machine learning to assure a trustworthy learning machine.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	18

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
JERMONT CHEN	315-227-7003

Autonomous Coordination Policies in Ground-Air Unmanned Systems Interaction*

Hussein Abbass, Kathryn Kasmarik, Matthew Garratt, Michael Barlow, and Sreenatha Anavatti

School of Engineering and Information Technology
University of New South Wales Canberra
Northcott Drive, Campbell
ACT 2600, Australia

December 2019

Abstract

Humans use a variety of complex strategies when solving problems. These strategies are semantically ordered; that is, as the complexity of a problem increases, an explanation of a strategy used for a simpler problem would be related to that of a strategy used for a more complex problem. Machines, however, can learn complex strategies but they normally produce strategies that are different from those used by humans. This leads to loss of trust and ineffective human-machine teaming. We designed machine learning algorithms that can learn incrementally while preserving the meanings embedded in a simpler context as they learn a more complex one. One proposed apprenticeship bootstrapping algorithm has been tested in simulation and physical environments on ground-air interaction tasks, while the others were either demonstrated theoretically or tested on simulated synthetic tasks alone. The research has demonstrated that machine education and bootstrapping techniques should be integral parts of explainable and interpretable machine learning to assure a trustworthy learning machine.

*We wish to thank the staff employed on this project Tung Nguyen, Hung Nguyen, and Phi Vu Tran, and co-authors and collaborators on publications: Eleni Petraki, Sondoss Elsawah, and Robert Hunjet. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing anyone else.

Contents

1 Introduction

Humans are still dominating the sphere of intelligent behaviour in complex situations, and produce strategies that are significantly different from those produced by machines [19]. For instance, in the past, our research was able to evolve from scratch, and without any prior domain knowledge, walking gaits for different types of robots in a physics-based high-fidelity simulation environment [18]. Today, the behaviours we would like to learn are far more complex. Swarming is becoming a major research challenge [9]. In particular, how a swarm of Unmanned Aerial Vehicles (UAV) can autonomously coordinate its actions in a trusted [3] manner while supporting a swarm of Unmanned Ground Vehicle (UGV) [2]? This question is too complex, not because of the open-ended nature of words such as 'coordination' and 'trust', but primarily because we lack automated algorithms of sufficient level of complexity to achieve these goals.

Consider a simpler example, where an UAV supports an UGV without communication. Even in this case, that is simpler than a complete swarm-swarm interaction, there are many challenges that remain to be addressed. The UAV may drift because of winds causing loss of visual contact with the UGV; the UGV may use camouflage to hide while simultaneously causing identification problems for the UAV if the camouflage signature is unknown or becomes disturbed; or the data-link may be jammed. The implications include loss of UAV's situation awareness on where the UGV is, causing the UAV to invest significant resources (time and battery) to locate the UGV to continue to support it; not to mention the impact of this problem on mission success.

If the UAV is tele-operated by human(s), the humans will use a diverse set of strategies in these situations. They may use their knowledge of the mission to narrow the search space where the UGV might likely to be located; environmental clues, such as signatures of the UGV's wheels on the ground, or last known location and direction of the UGV to estimate its current and future position.

Recent advances in areas such as Reinforcement Learning (RL) and Inverse Reinforcement Learning (IRL) have generated new algorithms such as Apprentice Learning (AL) and the Actor Critic Model (ACM). These algorithms can learn from the human data to automatically generate policies that can guide the UAV autonomously. However, most experiments conducted in the literature are simple tasks in a simplified environment. Soon after the task becomes slightly more complex, the learning process becomes exponentially so. One of the main reasons for this problem is the lack of a systematic approach to incrementally add complexity to the task, which could help designing more sophisticated learning algorithms.

While the primary aim of the project was to design a robust reinforcement algorithm for ground-air interaction, the project went beyond robustness to include the challenge of designing an algorithm that could bootstrap from low-level skills, while being guided with semantic chunks. The project ended up designing multiple algorithms with to form a family of algorithms that could get connected to deliver transparent reinforcement learning models that are bootstrapped from models learnt from human data on the sub-skills required for a particular problem. A summary of the titles of the resultant publications in reverse chronological order is included below:

- Swarm Q-Leaming With Knowledge Sharing Within Environments for Formation Control [17].
- Apprenticeship Bootstrapping [13].
- Apprenticeship Bootstrapping: Inverse Reinforcement Learning in a Multi-Skill UAV-UGV Coordination Task [14].
- Apprenticeship Bootstrapping Via Deep Learning with a Safety Net for UAV-UGV Interaction [15].
- A Deep Hierarchical Reinforcement Learner for Aerial Shepherding of Ground Swarms [16].

- Machine Education: Designing Semantically Ordered and Ontologically Guided Transparent and Explainable Modular Neural Networks [1].

The remainder of this report will summarize the algorithms and corresponding experiments from the above published papers. In particular, Section 2 will present the swarm q-learning algorithm we initially designed to explore the problem space. This is then followed by the main contribution of the project which took the form of the Apprenticeship Bootstrapping algorithm in Section 3. We then presents the machine education methodology in Section 4, which completes the project by introducing a systematic methodology to structure a set of requirements into a series of semantic chunks.

2 Swarm Q-Learning with knowledge Sharing Within Environments

This section is based on paper [17]. Readers should refer to the paper for more details. Below, we will summarize the algorithm and experimental findings.

2.1 Swarm Q-Learning

The classic Swarm Q-Learning (SQL) algorithm was introduced in [10]. It adopts a Q-learning reinforcement learning algorithm for each agent with a modification of the reward function. At the conclusion of an episode, the algorithm assigns an extra reward to individuals based on their coordination abilities. When all members land on their target positions, they all get the extra reward. If some lands while others do not, they all get penalized with a negative reward. However, those who landed on their target positions are penalized much less than those who missed their target position. The algorithm is presented in Algorithm 1.

2.2 Knowledge Sharing

When evaluated on our test problem, the SQL algorithm did not converge on many occasions. Scalability is a challenge due to coordination neglect. The Swarm Q-Learning with knowledge Sharing Within Environments (SQL-SIE) algorithm decomposes the search space and allows agents to exchange knowledge for a better coordination.

In an $M \times M$ square environment E, K agents need to land on K targets using L actions. The problem is decomposed into two sub-problems. First, agents need to navigate to reach the landing area (ROI: Region of interest), where they need to be positioned according to some formation. Second, agents move to their individual positions in the formation. Q-Learning is used for the first sub-problem. Once agents reach the boundary of the ROI, K Q-matrices are formed using the Q-Learning algorithm, with appropriate state-action values to guide each agent to its position on the formation. All agents share the K Q-matrices to find the closest target.

The first problem has the same state and action spaces and goals. Agents are able to share knowledge by exchanging Q-values. This knowledge sharing does not imply that all agents will converge to the same policy. To the contrary, agents form different experiences and policies due to the stochastic nature of the learning problem. SQL-SIE is presented in Algorithm 2.

Algorithm 1 Swarm Q-Learning (SQL)

Input : Maximum number of episodes (T), Maximum number of steps in one episode (ϕ), Action set allowed for an agent to perform (A), Number of agents (K)

Output: Q-table (Q)

Initialize the states and the Q-table for an agent

for $t = 1$ **to** T **do**
  Set $\varphi \leftarrow 1$
  while $\varphi < \phi$ *and all agents are not at target states* **do**
    for $k = 1$ **to** K **do**
      if *agent* k *is not at target states* **then**
        Obtain the current state of agent k
        Use agent k Q table and ϵ-greedy algorithm to select action $a_l \in A$
        Observe the next state and reward r
        Update Q-values for agent k
      end
    end
    $\varphi \leftarrow \varphi + 1$
  end
  Each agent receives a bonus reward r_{bonus} depending on status of the targets
  Update the Q-values for the final state-action pairs
end

Algorithm 2 Swarm Q-Learning with knowledge Sharing wIthin Environments (SQL-SIE)

Input : Maximum number of episodes (T), Maximum number of steps in one episode (ϕ), Action set allowed for an agent to perform (A), Number of agents (K)

Output: Q-table (Q)

Initialize the states and the Q-table for an agent

for $t = 1$ **to** T **do**

  Set $\varphi \leftarrow 1$

  while $\varphi < \phi$ *and all agents are not at target states* **do**

    for $k = 1$ **to** K **do**

      if *agent k is not at target states* **then**

        Obtain the current state of agent k

        Use agent k Q table and ϵ-greedy algorithm to select action $a_l \in A$

        Observe the next state and reward r

        Update Q-values for agent k

      end

    end

    $\varphi \leftarrow \varphi + 1$

  end

  Each agent receives a bonus reward r_{bonus} depending on status of the targets

  Update the Q-values for the final state-action pairs

  Calculate Value-shared Q-matrix $Q_t^*(s, a)$ using Equation **??**

  Copy $Q_t^*(s, a)$ to all Q-tables for all agents

end

2.3 Findings

Experiments were conducted to evaluate the performance of SQL-SIE by comparing it to the classic SQL, understanding the sensitivity of the algorithm to knowledge sharing, and its ability to scale. Finding included: knowledge sharing is important for stability and speed of learning; better performance was shown with more frequent knowledge sharing although this is expected to be a problem-specific phenomenon; and scalability and convergence have both improved significantly compared to classic SQL. The details of these results could be seen in [17].

3 Apprenticeship Bootstrapping

SQL-SIE demonstrated that a well-engineered decomposition of the learning problem improves the stability and rate of convergence of reinforcement learning (RL). However, more challenges remain for RL, an important one of which is the difficulties faced by a designer to define the right reward function.

3.1 Apprenticeship Learning

Inverse Reinforcement Learning (IRL) [12] offered a solution: use a human expert to demonstrate the task to the machine. IRL then learns the implicit reward function used by the human. The resultant reward function could be used explicitly or implicitly to design the RL algorithm.

Abbeel and Ng [5] borrowed concepts from imitation learning to define a new learning paradigm: apprenticeship learning (AL). While IRL recovers the reward function explicitly, AL approximates it using weighted linear features. The fusion of AL and IRL has been used successfully for different applications including helicopter controller design [4].

3.2 Apprenticeship Bootstrapping

Apprenticeship bootstrapping generalizes Abbeel's AL to tasks where human experts are not available. Instead, it is possible to decompose the task into sub-tasks, with each sub-task requiring skills where a human expert is available. The overall task is then performed by a machine learning model that fuses the sub-models. When coupled with Deep Q-Network (DQN) [11], the resultant algorithm demonstrated robust performance. The algorithm modifies Abbeel's [5] original one with the extra steps shown in italics in Algorithm 3. The detailed description of the notations could be found in [14].

The reward function, $R(s)$, is a linear weighted, w, sum of the state feature vector, $\phi(s) = \{\phi(s_1), \ldots, \phi(s_i), \ldots, \phi(s_K)\}$, whereby:

$$R(s) = w^T \cdot \phi(s) = \sum_{k=1}^{K} w_k \phi_k(s), \forall s \in S \tag{1}$$

The basic assumption for the convergence of IRL to an appropriate reward function is that the human is an expert. In other words, IRL can't recover from human errors and will reveal the policy used by the human, which could be sub-optimal if the actions generated by the human are sub-optimal. The problem is to find the optimal weight vector w^* corresponding the optimal reward $R^*(s)$ as below:

$$R^*(s) = w^* \cdot \phi(s), \forall s \in S \tag{2}$$

Algorithm 3 Apprenticeship Bootstrapping (ABS) via Inverse Reinforcement Learning Algorithm; a modification of the algorithm presented in [5] with the additional steps shown in italics.

Input: : *A feature mapping ϕ, and sub-task demonstrations , $\{D_1, D_2, D_3, ..., D_H\}$.*
Output: : A number of policies $\{\pi^{(i)} : i = 0..n\}$.

Randomly choose policy $\pi^{(0)}$, estimate $\mu^{(0)} = \mu(\pi^{(0)})$ via Monte Carlo, and set $i = 1$.
Initializing the parameters of DQN.
for *each sub-task* **do**
    Calculating the expert feature expectations vector $\mu_E^{D_h}$ from h sub-task demonstrations, D_h, based on Equation 5
end for
Calculating the total expert feature expectations vector, $\mu_E^{Primitive}$ as described in Equation **??**.
Set $w^{(1)} = \mu_E^{Primitive} - \mu^{(0)}$ and $\bar{\mu}^{(0)} = \mu^{(0)}$.
Set $t^{(1)} = \|\mu_E^{Primitive} - \mu^{(0)}\|_2$.
if $t^{(i)} \leq \epsilon$ **then**
    terminate
end if
while $t^{(i)} > \epsilon$ **do**
    Using DQN to compute the optimal policy $\pi^{(i)}$ with $R = (w^{(i)})^T \phi$.
    Compute $\mu^{(i)} = \mu(\pi^{(i)})$ and set $i = i + 1$.
    Set $a = \mu^{(i-1)} - \bar{\mu}^{(i-2)}$.
    Set $b = \mu_E - \bar{\mu}^{(i-2)}$.
    Set $\mu^{(i-1)} = \bar{\mu}^{(i-2)} + \frac{x^T y}{x^T x} x$.
    Set $w^{(i)} = \mu_E - \bar{\mu}^{(i-1)}$.
    Set $t^{(i)} = \|\mu_E - \bar{\mu}^{(i-1)}\|_2$.
end while

IRL uses human demonstrations to approximate w^*. The value function for a fixed policy and a discount factor γ is:

$$V^{\pi}(s_0) = w \cdot E\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) \mid \pi\right] \tag{3}$$

The feature expectations vector, $\mu(\pi)$, is defined as:

$$\mu(\pi) = E\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) \mid \pi\right] \tag{4}$$

For an unknown dynamic model, Equation 5 is used.

$$\mu(\pi) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{\infty} \gamma^t \phi(s_t^{(i)}) \tag{5}$$

where m is the number of trajectories.

Given the feature mapping function, ϕ, and demonstrations the Deep Q-learning (DQN) algorithm [11] recovers $\tilde{\pi}$ from the policies, π_E, of the human expert to find $\mu(\tilde{\pi})$ as close as possible to $\mu(\pi_E)$.

3.3 Apprenticeship Bootstrapping Performance

AB was tested on a simulated ground-air interaction task in the Gazebo environment. The simulations are conducted with three UGVs: UGV1, UGV2, and UGV3, and a single cooperating UAV. The aim is for the UAV to maintain all UGVs within Field of View (FoV) of the UAV's image sensor without losing any UGV from the FoV or creating a FoV that is much larger than what is needed to accommodate the smallest manifold containing the three UGVs. To collect human demonstrations, the system allows a human to teleoperate the UAV from a distance. A pictorial representation of the system is shown in Figure 1.

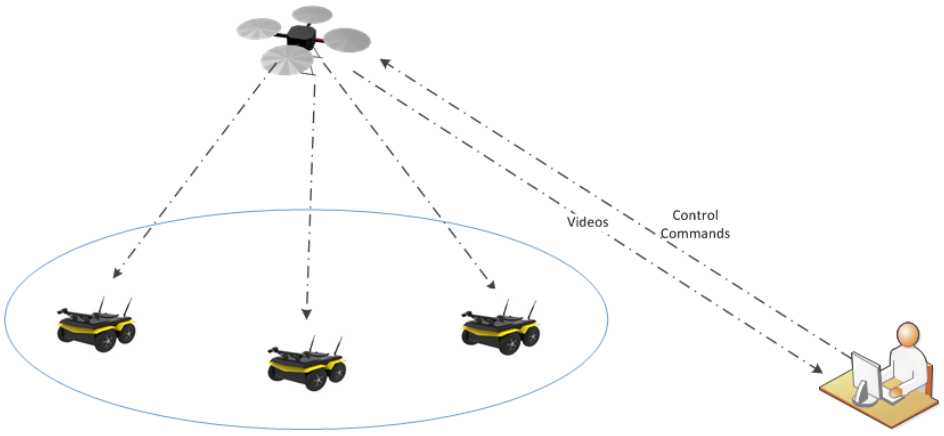

Figure 1: Ground-Air UAV-UGVs benchmark.

A human is used in three basic constrained scenarios requiring one maneuver each: Lateral-Movements-Fixed-Altitude maneuver, Climb-Only maneuver, and Descend-Only maneuver. We then used the human to perform all three maneuvers in a fourth scenario: Lateral-Movements-With-Climb-Descend maneuver. IRL was used in each case, with the fourth scenario providing a baseline for AB. AB we used to bootstrap a machine performance for the fourth scenario using the first three scenarios alone. The performance of IRL on the fourth scenario was compared to the performance of AB; with AB achieving similar performance to IRL. In other words, if a human was not available to conduct the fourth scenario requiring the skills to decide which maneuver to perform, AB is able to achieve similar to performance to the case if a human was available.

3.4 Safety Net in Physical Environments

The AB algorithm, similar to other RL models, could generate unexpected behaviors due to lack of algorithmic assurance and appropriate verification of machine learning models. This called for thinking of how the algorithms could be transferred to a physical environment where safety is paramount. The first solution was to design a behavioral safety net as shown in [15]. The second was to introduce the machine education approach presented in the following section.

The control environment is shown in Figure 2 in which the red dot is the centre of the top-down image from the UAV's camera, and the blue dot and blue circle are the centre of mass and the spread of the UGVs in the UAV's image, respectively. The cooperative control network architecture is shown in Figure 3.

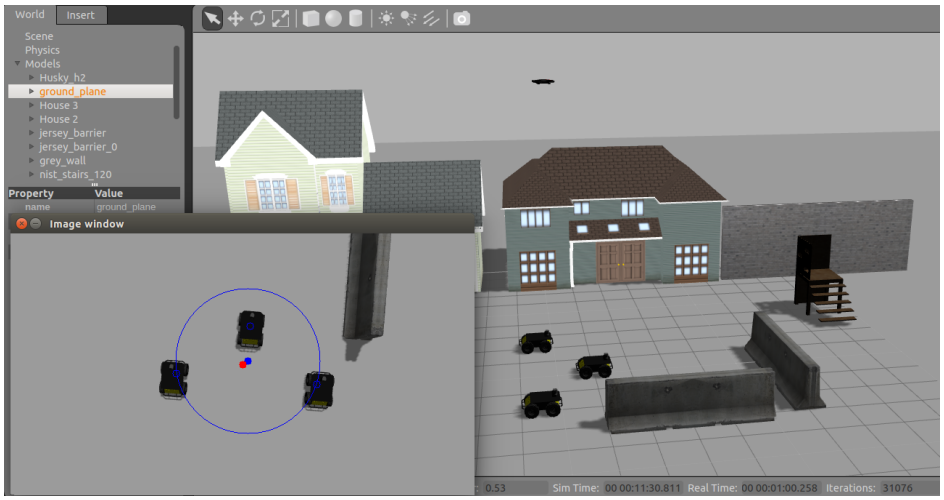

Figure 2: Control Environment for UAV and UGVs.

A safety net is a constraint system representing the polyhedron (in case of linear constraints) within which UAV actions are accepted. Linear constraints are used for simplicity due to ease of implementation, ability to explain them to end users, and existence of efficient constraint checkers to evaluate them. The choice of these constraints is made in a spiral approach, where the designer starts with an initial set of constraints that continue to be refined during the evolution of the system. A major design choice is the design of the action set of the UAV when one or more of these constraints get violated. for a quadcopter,

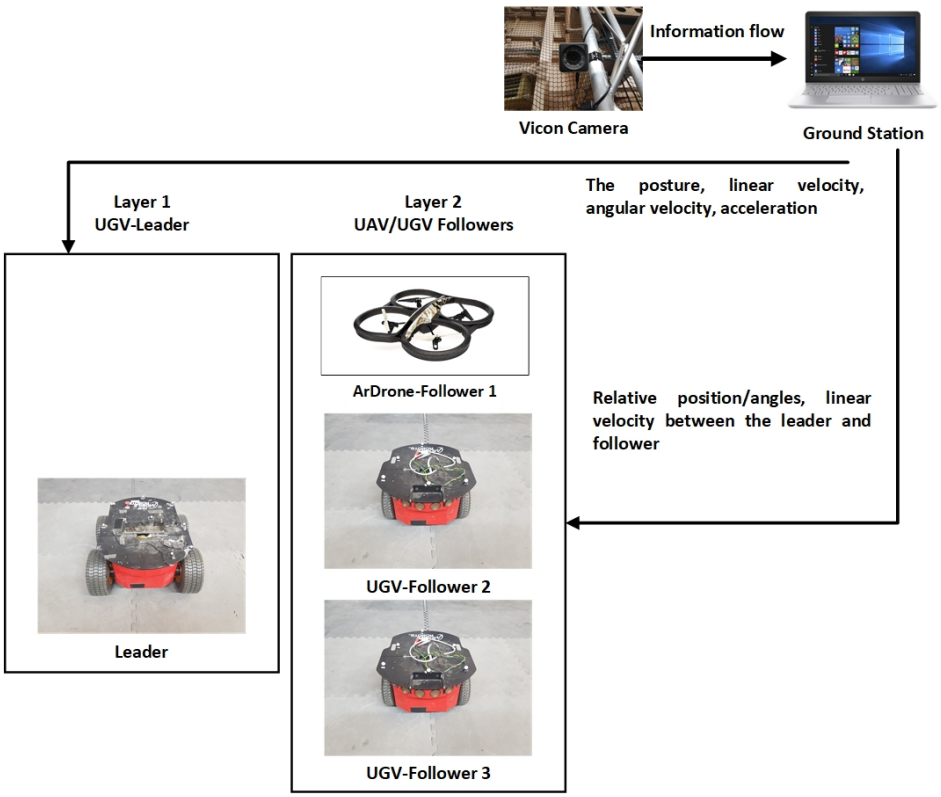

Figure 3: Overall Architecture Diagram.

a possible action when reaching a geofence could be to hover. This action is not appropriate for a fix-wings UAV with no hovering abilities.

Another scenario is when the AI produces an action outside the performance envelope of the UAV. In this particular case, the corrective action could be generated from a fall-back classic control system to stabilize the UAV until the AI recovers.

4 Machine Education

The work in [15] presented some first steps in the design of a safety net when moving an AI model from a simulated environment to a physical environment where safety is paramount. However, it demonstrates that the safety and assurance problem should not be left to that late stage in the development cycle of the AI. To the contrary, it could be easier and more efficient to integrate this problem from the start.

The machine education approach presented in [1] is a novel idea emphasizing that a systemic approach towards the design of AI systems could solve challenging problems such as transparency and assurance.

While machine-teaching is still an emerging literature despite its foundations in Elman's work in 1993 [7], the emphasis of machine teaching is on the decomposition of the learning experience into chunks that are simpler to learn. The literature on the topic so far is in the hands of few machine learning experts with intimate knowledge of algorithms. The complexity of the mathematics limited the uses to simple tasks.

Machine education, however, is a very recent idea that was demonstrated only in a limited number of studies. In particular, the work of [6,8] were the first two studies relying on proper education methodologies to guide the design of a machine learning algorithm for shepherding. Nevertheless, the methodologies assumed that the task is well-defined.

The machine education approach presented in [1] offered an end-to-end methodology that sits between an AI illiterate user and an assured and transparent AI model. The overall framework is presented in Figure 4. The work demonstrated a systematic way to structure the semantics of the machine learning models and how to ensure semantic ordering. The mathematical definition of the concepts are available in [1].

5 Conclusion

While the project was concluded at this stage, we expect that the opportunity offered by linking machine education to the apprenticeship bootstrapping algorithm will present an integrated methodology for designing robust, trustworthy and explainable supervised and reinforcement learners. The key limitation of the work is the cost associated with running methodologies like machine education and apprenticeship bootstrapping. These methodologies require continuous systemic thinking, documentation and refining. These costs are popular in large organizations following appropriate system engineering practice, but could be a burden for scientists aiming to produce a machine learning algorithm for a particular problem without the skills and resources to follow the system engineering approach.

References

[1] Hussein Abbass, Sondoss Elsawah, Eleni Petraki, and Robert Hunjet. Machine education: Designing semantically ordered and ontologically guided transparent and explainable modular neural networks. In *2019 IEEE Symposium Series on Computational Intelligence (IEEE-SSCI)*, page In Print. IEEE, 2019.

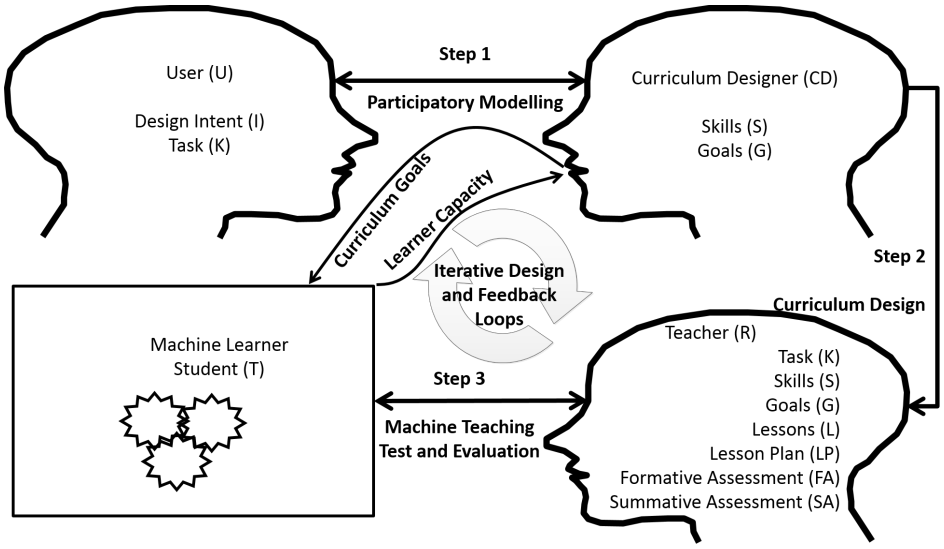

Figure 4: The methodological framework for machine education.

[2] Hussein A Abbass, George Leu, and Kathryn Merrick. A review of theoretical and practical challenges of trusted autonomy in big data. *IEEE Access*, 4:2808–2830, 2016.

[3] Hussein A Abbass, Eleni Petraki, Kathryn Merrick, John Harvey, and Michael Barlow. Trusted autonomy and cognitive cyber symbiosis: Open challenges. *Cognitive computation*, 8(3):385–408, 2016.

[4] Pieter Abbeel, Adam Coates, and Andrew Y Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[5] 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. ACM, 2004.

[6] Nicholas R Clayton and Hussein Abbass. Machine teaching in hierarchical genetic reinforcement learning: Curriculum design of reward functions for swarm shepherding. In *IEEE Congress on Evolutionary Computation*, Wellington, New Zealand, 2019. IEEE.

[7] Jeffrey L Elman. Learning and development in neural networks: the importance of starting small. *Cognition Voume 48*, 1993.

[8] Alexander Gee and Hussein Abbass. Transparent machine education of neural networks for swarm shepherding using curriculum design. In *International Joint Conference on Neural Networks*, Budapest, Hungary, 2019. IEEE.

[9] John Harvey, Kathryn Merrick, and Hussein A Abbass. Application of chaos measures to a simplified boids flocking model. *Swarm Intelligence*, 9(1):23–41, 2015.

[10] Hitoshi Iima and Yasuaki Kuroe. Swarm reinforcement learning method for a multi-robot formation problem. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 2298–2303. IEEE, 2013.

[11] 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.

[12] Andrew Y Ng and Stuart J Russell. Algorithms for inverse reinforcement learning. In *ICML*, pages 663–670, 2000.

[13] Hung Nguyen, Matthew Garratt, and Hussein Abbass. Apprenticeship bootstrapping. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[14] Hung Nguyen, Matthew Garratt, Lam Thu Bui, Hussein Abbass, et al. Apprenticeship bootstrapping: Inverse reinforcement learning in a multi-skill uav-ugv coordination task. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2204–2206. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[15] Hung Nguyen, Vu Tran, Tung Nguyen, Matthew Garratt, Kathryn Kasmarik, Michael Barlow, Sreenatha Anavatti, and Hussein Abbass. Apprenticeship bootstrapping via deep learning with a safety net for uav-ugv interaction. page arXiv preprint arXiv:1810.04344, 2018.

[16] Hung T Nguyen, Tung D Nguyen, Matthew Garratt, Kathryn Kasmarik, Sreenatha Anavatti, Michael Barlow, and Hussein A Abbass. A deep hierarchical reinforcement learner for aerial shepherding of ground swarms. In *International Conference on Neural Information Processing*, pages 658–669. Springer, 2019.

[17] Tung Nguyen, Hung Nguyen, Essam Debie, Kathryn Kasmarik, Matthew Garratt, and Hussein Abbass. Swarm q-leaming with knowledge sharing within environments for formation control. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[18] Jason Teo and Hussein A Abbass. Multiobjectivity and complexity in embodied cognition. *IEEE Transactions on Evolutionary Computation*, 9(4):337–360, 2005.

[19] Shir Li Wang, Kamran Shafi, Theam Foo Ng, Chris Lokan, and Hussein A Abbass. Contrasting human and computational intelligence based autonomous behaviors in a blue–red simulation environment. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(1):27–40, 2017.