Learning Weather Navigation Skills from Human Pilots Demonstrations using Airborne Radar Data

Moslem Kazemi
Merlin Labs Inc.
Boston, MA, USA
Email: moslem.kazemi@merlinlabs.com

Alexander Naiman
Merlin Labs Inc.
Boston, MA, USA
Email: alex.naiman@merlinlabs.com

***Abstract*—Weather navigation is one of the essential capabilities for building an autonomous flight navigation system. It is a challenging task for human pilots that requires a great deal of training, background knowledge, and experience. Manually designing such knowledge into an autonomy system, if at all possible, would require large engineering effort and painstaking manual tuning to achieve human-level performance. In this work, we frame weather navigation as an imitation learning problem where human pilot demonstrations are used to learn how to navigate a weather region. We adopt an inverse optimal control formulation which seeks a cost function, i.e., a mapping from weather features to a scalar value cost, under which human pilot demonstrations are optimal. For a given weather navigation problem, the learned cost function is used to generate a costmap that in turn is used by a deterministic planner for navigating the weather region. As a proof of concept we apply this framework to a simplified problem of in-flight weather navigation using airborne radar data. Through a number of training and validation tests we demonstrate its effectiveness in learning from human pilot demonstrations and producing results that are comparable, and in some cases on par, with that of human pilots in generating routes to navigate a variety of weather scenarios. Although the simplified proof of concept presented here focuses on using airborne radar data only, the underlying approach is flexible, and we discuss the remaining challenges and ways of extending it to include other sources of weather data and achieve improved performance.**[1]

I. Introduction

Weather navigation is one of the challenging tasks in aviation. It requires a substantial amount of background knowledge and it involves a number of critical decisions and actions performed by the pilots for efficient and safe flights in the presence of adverse weather conditions. It consists of a wide spectrum of skills from data gathering and fusion, to inference, reasoning, and avoidance strategies. Skilled pilots are highly capable of performing these skills effectively and in a timely manner where deemed relevant in different phases of a flight.

Weather navigation is an essential capability in autonomous flight navigation systems to enable fully automated safe flight under varying weather scenarios. The work presented here discusses the challenges in designing and developing weather navigation skills and devises a framework for learning such skills from human pilots. Applying a learning-based approach offers compelling advantages: for one it alleviates the burden of explicitly programming the skills to cover every contingency, a task which is burdensome if at all possible. Moreover, aviation guidelines do not always provide detailed guidance on how to handle every possible weather scenario, leaving those details to the pilot's discretion. Learning from pilot demonstrations offers an implicit way to learn such skills in the absence of explicit rules and guidance. Finally, the skills acquired through learning from demonstrations tend to be more human-like and less robotic. This is not easily achievable through programming heuristics or tuning cost functions.

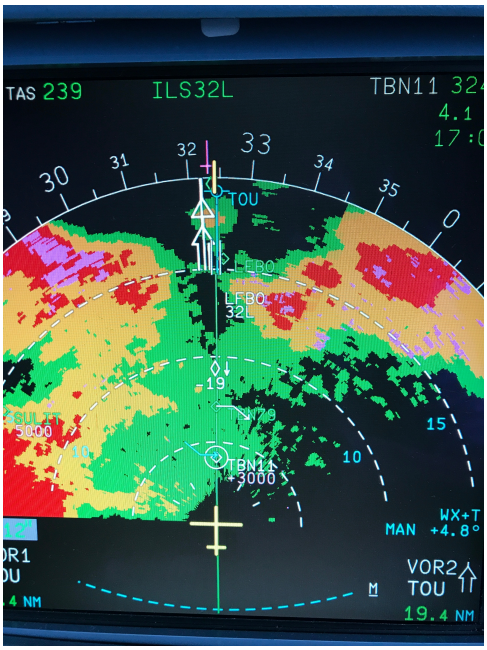


Fig. 1: Screenshot of an airborne weather radar display (source: Adobe Stock)

We propose a framework for learning weather navigation skills from human pilot demonstrations. In the proposed framework, weather avoidance is framed as an inverse optimal control problem that seeks a cost function, i.e., a mapping from weather features to a scalar value cost, under which human pilot demonstrations are optimal. The inverse optimal control is implemented as a maximum margin structured prediction problem, which has been successfully used in the past for solving navigation problems in other domains [1]. For a given weather navigation scenario, the system uses the learned cost

[1]Published and presented in *Digital Avionics Systems Conference (DASC)*, San Diego, CA, USA, September 2024

function to generate a costmap representing the traversability of the space based on the observed weather features. We then use a deterministic planner to search the costmap for the lowest cost, i.e., optimal, route that efficiently traverses the space to maneuver around adverse weather regions.

As a proof of concept we apply this framework to the problem of in-flight weather navigation using airborne radar data (Fig.1). The training examples are obtained by asking pilots to design and draw feasible routes on airborne radar images to navigate example weather regions. We then demonstrate the effectiveness of the approach by applying the cost mapping learned from these examples to plan optimal routes for a test set of examples. The proof of concept presented here focuses on simple and relatively unrealistic scenarios. As a future direction to this research, we allude to the flexibility of the underlying approach and discuss the challenges of extending it to real-world scenarios and ways to address them. We also discuss how other data sources such as NEXRAD can be leveraged to improve the learning process and achieve more enhanced skills.

The outline of the paper is as follows: in the next section we give an overview of the background and some of the related work in this area. Then we provide the inverse optimal control (IOC) problem formulation in Section III followed by the details of the Maximum Margin Planning approach to solve the IOC problem in Section IV. In Section V we explain our process for gathering training data and we discuss the results validating the performance of the proposed approach in learning weather navigation from human pilot demonstrations on a holdout test set. Finally, in Section VI we discuss some of the challenges of extending this work to include other sources of weather data and provide a few possible directions to address those challenges as the future work.

II. Background and Related Work

The work presented here spans a wide range of subjects in aviation, weather navigation, robotics, autonomy, and imitation learning. A thorough and complete review of each area and its related work is beyond the scope of this paper. Here first we focus on in-flight weather navigation, its challenges, and existing strategies for autonomous generation of avoidance maneuvers to navigate weather regions. Then we discuss some of the main work in mobile robot navigation related to the problem of route generation and challenges of generating routes in complex unstructured environments where the mapping between the environment features and planning is not trivial. We then review some of the strategies that focus on learning this mapping through demonstration and, more specifically, imitation learning.

Adverse weather can pose serious threats to the safety and comfort of a flight, and will have to be detected and avoided in a timely and safe fashion. During each phase of a flight, pilots rely on multiple different sources to gather information about the weather condition and aggregate such data to be able to decide on any flight plan changes in response to changing conditions. During the in-flight phase one of the main sources of weather data is the airborne weather radar which helps the pilots to assess the intensity of the convective weather ahead. It enables them to strategize and decide on safe and efficient maneuvers around adverse local conditions. Despite recent technological advances in weather radar, pilots are still required to perform most of the tuning and settings manually to be able to correctly interpret the information shown by the radar display. Besides the task of operating the radar features such as tilt, range, gain control, and modes, pilots interpret the intensity images shown on the radar display and accordingly decide on a safe route to navigate the weather region. We should note that pilots also rely on other sources of weather information such as looking out the window, reports from other pilots in the vicinity, ATC reports, and/or NEXRAD data. The work here focuses on using airborne radar data only. However, we will discuss the challenges of extending it to also automate the radar features settings and use of other information sources as future directions to this research.

Prior works on autonomous weather navigation are limited. Although there are a variety of tools and commercial products offering aviation weather planning and visualization, such as the ForeFlight weather tools [2], to the best of our knowledge, there is no automated functionality for in-flight dynamic (re)planning of safe routes around weather regions, in particular based on airborne radar data. The closest we came across is the Dynamic Weather Routes (DWR) tool developed by NASA [3]. DWR is a ground-based automation system that continuously and automatically analyzes in-flight aircraft in en-route airspace to find opportunities for time- and fuel-saving corrections to weather avoidance routes during departure and overflights. The focus of the tool is on saving time and fuel, and starting with the original flight plan it uses the direct route as a "reference route," and inserts up to two auxiliary waypoints as needed to find a minimum-delay route correction that avoids the weather, or optionally weather and traffic conflicts, and returns the flight to its planned route at the downstream fix. The DWR was later extended in [4] to improve arrival traffic flow as well.

Autonomous navigation is a well studied subject in robotics with decades of research that has resulted in successful implementation of some of those promising ideas in real world applications such as self-driving cars. Autonomous navigation in unstructured and complex environments is still a challenging task despite many advances in perception, for obtaining environment features, and planning, for generating plans to traverse the environment given the features extracted by the perception system.

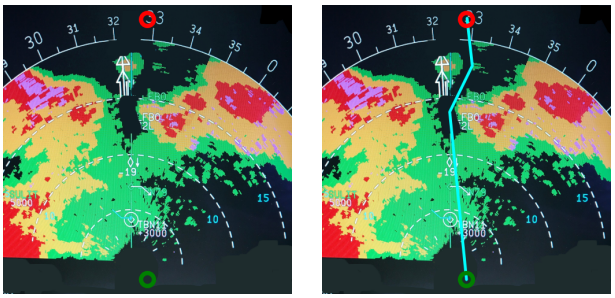
The coupling between perception and planning is one of the biggest challenges in navigation. Usually the approach taken is to encode and calculate a notion of traversability for every location of the environment and use that to inform a planner to traverse the space. Some earlier methods [5] rely on a binary classification (traversable vs. non-traversable). However, such classification was shown to be sub-optimal in dealing with more complex environments where the critical differences between features could be lost due to the binarization, hence

leading to sub-optimal plans.

The inadequacy of binary classification motivated the notion of (continuous) navigation cost functions (e.g., [6]). A cost function is a mapping from environment features to a scalar cost value. The magnitude of the cost determines the traversability of the space. The crux of the problem is then to find the ideal mapping from the environment features to cost value. This is a far harder problem compared to binary classification. The richer the environment in terms of its features, the bigger the challenge of designing a cost function that adequately encodes and captures the complexity of the environment in order to generate optimal and efficient routes. Moreover, the problem gets even harder when it is desired for the solution route to optimize different metrics (e.g., travel distance, safety, risk, time, speed). In a navigation problem it may be desired that the metric getting optimized is a combination of two or more of such metrics and finding the proper balance between them when shaping the cost function remains a challenge.

Over the years a number of techniques have been proposed for shaping and computing navigation cost functions. The reader is encouraged to refer to [7] for a detailed and extensive review of these techniques. Majority of these approaches rely on manual design and tuning of cost functions and use of engineering techniques to shape the cost function. Techniques using physical simulation, supervised classification, and learning from experience have alleviated some of the tedious aspects of manual tuning and engineering of the cost functions. However, the success of these techniques have been limited and they tend to be not easily extendable to wide variety of navigation problems and robotic systems. These challenges motivated a class of approaches known as imitation learning that try to learn to directly map perception to actions through learning from human demonstrations. Imitation learning techniques rely on the fact that the correct action is known to the human experts even though it may be very difficult to quantify the rationale for their actions. Therefore, rather than having an expert to manually tune the cost function the expert can demonstrate the correct action and the robot can tune itself in order to match the expert demonstrations.

For the purpose of this work, here we focus on a group of imitation learning methods that are rooted in the concept of Inverse Optimal Control (IOC) [8]. In contrast with optimal control methods where the objective is to find a trajectory that optimizes a known metric, IOC aims at finding a metric that explains a given trajectory that is optimal with respect to that metric. The metric could be either a reward function (as in a reinforcement learning setup) or a cost function (as in a planning problem on a costmap). Inverse reinforcement learning [9], and its successor, apprenticeship learning [10], are two applications of IOC in the framework of Markov Decision Process. Later on Maximum Marching Planning (MMP) [11] addressed some of the shortcomings of apprenticeship learning by producing a single deterministic solution and ensuring that the mismatch between planned and demonstrated behavior are bounded. MMP has been successfully used for mobile robot navigation in unstructured environments and off-road driving [1].



(a) (b)

Fig. 2: An example of weather navigation using airborne radar intensity images: (a) an example radar intensity image (source: Adobe Stock) with the current and goal locations of the airframe depicted in green and red, respectively, (b) a viable planned route (cyan) from the current to the goal location

In this work as a proof of concept we apply the MMP approach for learning how to navigate weather regions using airborne radar data. To the best of our knowledge this is the first application of MMP to solve an aviation navigation problem. Applying a learning-based approach, and particularly MMP, offers compelling advantages: for one it alleviates the burden of explicitly programming the skills to cover every contingency, a task which is burdensome if at all possible. Moreover, aviation guidelines do not always provide detailed guidance on how to handle every possible weather scenario, leaving those details to the pilot's discretion. Learning from pilot demonstrations offers an implicit way to learn such skills in the absence of explicit rules and guidance. Also, the skills acquired through learning from demonstrations tend to be more human-like and less robot-like. This is not easily achievable through programming heuristics or tuning cost functions. MMP provides an extensible framework to add more features, and allows using linear cost functions as well as non-linear cost functions. Moreover, besides static scenarios it can be applied to dynamic and unknown environments as well [12]. Finally, it allows relatively straightforward modifications (see [1]) to make it more robust and improve its generalization in the face of noisy or poor expert demonstrations.

III. Problem Definition and Formulation

A. Imitation learning problem definition

The problem of navigating weather regions using airborne radar data is defined as follows: given an image of weather radar intensity (e.g., Fig.2) and the current location of the ownship, find a safe and feasible route to reach a predefined goal location. This problem statement is simplified as compared to the real world problem. As mentioned earlier, pilots also rely on other sources when deciding on a navigation strategy. However, for the sake of this work we limit the sources of weather information to airborne radar data only. Also, we acknowledge that pilots also change and tune radar settings such as tilt, range, and gain, to maintain a good

understanding of the weather region ahead. In the proposed problem statement the radar intensity image is assumed to be the best representation of the weather ahead after the pilot manually changes and tunes the settings. We also focus on 2-D routing, though extension to 3-D would be relatively straightforward. We discuss these assumptions and ways that the current proof of concept can be extended to relax them in Section VI.

The radar intensity image is usually shown as a 4 (and sometimes 5)-color heatmap on the radar display[13]: black, means there is no discernible return (i.e., no rain or rain is too light to be detected); then green for weak return (i.e., light rain); yellow for moderate return (i.e., moderate rain); red for both strong and very strong return (i.e., heavy through extreme rain); and some displays show magenta for extremely reflective returns. Given a radar intensity image it is a relatively easy task for the pilot to decide on a viable route to navigate it. We treat each color as a feature used by the pilots for planning a route to navigate the weather regions.

In order to automate route generation one commonly used approach is to create a cost function from the discrete set of features (e.g., a linear weighted cost) and evaluate the cost at each point (pixel) of the discretized image to compute a costmap. The magnitude of the cost at any point on the costmap is a measure of its traversability in comparison with the other points. In other words, regions of the costmap with lower cost are preferable over regions with higher cost values. Given the costmap, a deterministic planner, for example A*, can be used to search for a minimum cost route through the costmap.

As elaborated in the previous section, the main challenge is to find the appropriate weighing used to combine the discrete set of features into a scalar value cost function. In this work we apply an inverse optimal control approach to learn the cost function from human pilot demonstrations. The problem then can be stated as follows: given a number of radar images and the corresponding routes demonstrated by human pilots the problem is to learn a cost function under which pilot demonstrations are optimal.

B. Problem formulation

Here we provide a formal definition for the problem and formulate the inverse control problem to be solved. The radar intensity image (discretized into grid cells) defines the state space $\mathcal{S}$ on which the planner operates to find a route (i.e., $\mathcal{S} = \mathbb{R}^2$). The feature space $\mathcal{F}$ is defined over $\mathcal{S}$, i.e, at every state (grid cell) s in $\mathcal{S}$ there exists a feature vector F_s where F_s is a 5-D vector, one dimension for each intensity level (black, green, yellow, red, and magenta). Given a radar intensity image the feature vector can be calculated for every state on the intensity image. The cost function C is a mapping from the feature space to the space of non-negative real numbers, i.e., $C : \mathcal{F} \rightarrow \mathbb{R}^+$. We adopt a linearly weighted cost function $C(F_s) = w^T F_s$ for this problem where w is the weight vector. A route R in the state space is defined as a series of state from the start s_0 to the goal s_g and the cost of a route is the sum of the cost of features at each state along the route, i.e.,

$$C(R) = \sum_{s \in R} C(F_s) = \sum_{s \in R} w^T F_s \tag{1}$$

We ask human pilots to use the radar intensity images to demonstrate viable routes and then we split the recorded demonstrations into our training and test data sets. Given a set of routes, i.e., training examples, $\mathcal{R}_e$, demonstrated by human pilots, the inverse optimal control problem to be solved is to find the weight vector w under which the cost of each pilot route is less than the cost of any other route planned for the same navigation problem with the same start and goal states as pilot route's start and goal states, respectively, i.e.,

$$\begin{aligned} \sum_{s \in \widehat{R}} w^T F_s \geq \sum_{s \in R_e} w^T F_s \\ \forall \widehat{R} \quad s.t. \quad \widehat{R} \neq R_e \end{aligned} \tag{2}$$

In the next section we describe how Maximum Margin Planning provides an intuitive and efficient way of finding the weight vector under the above constraint.

IV. Maximum Margin Planning

The Maximum Margin Planning [11] provides an iterative algorithm for updating the weight vector in order to solve the following optimization problem:

$$\arg\min_{w} O(w) = \lambda ||w||^2 + \sum_{s \in R_e} w^T F_s - \sum_{s \in R_*} (w^T F_s - L_e(s)) \tag{3}$$

In the above equation $\lambda ||w||^2$ is a regularization term to ensure simpler solutions, R_* is the optimal route generated by the (A*) planner using the current weight vector, and L_e is a loss function as a measure of similarity between the example route and the optimal route. The loss function gives more margin to planner routes that are dissimilar to the example routes and also helps with avoiding trivial solutions, i.e., $w = 0$ in Eq.2. We skip the details of how to derive the above equation and refer the reader to [1] for the details.

Intuitively speaking, the above optimization problem aims to reduce the difference between the cost of the example route provided by the pilot and cost of the loss-augmented optimal route generated by the deterministic planner based on the current set of weights. To solve the optimization problem a gradient descent approach can be used. However, as suggested in [1] due to the non-convexity of Eq.3, instead of gradient the sub-gradient below is used:

$$\nabla O = 2\lambda w + \sum_{s \in R_e} F_s - \sum_{s \in R_*} F_s \tag{4}$$

which intuitively indicates that the direction that the objective function $O(w)$ is most minimized is along the difference in feature counts between the example route and the optimal route. In other words, when there are more (or less) of a feature

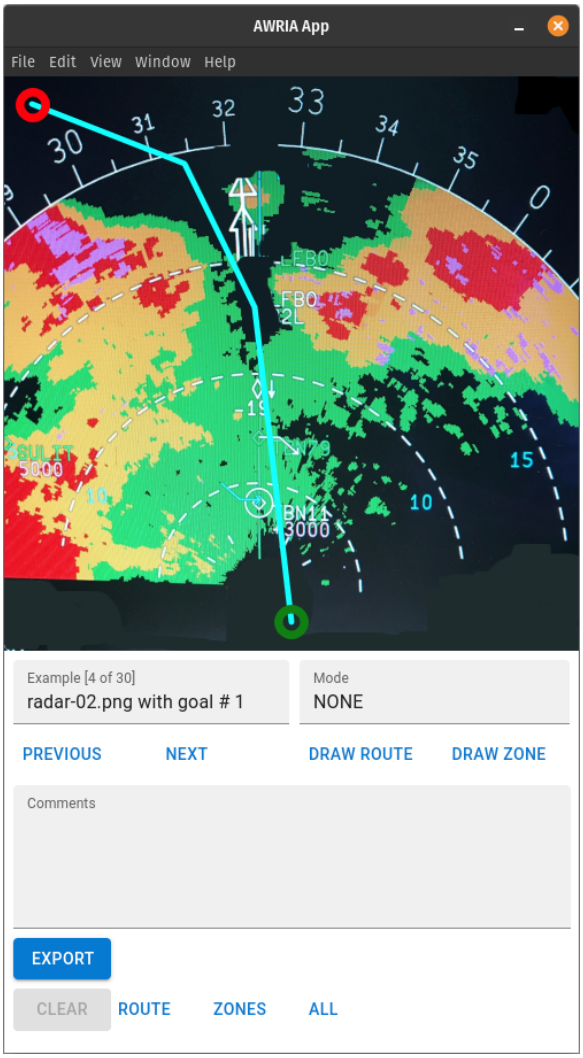


Fig. 3: A screenshot of AWRIA (Airborne Weather Radar Image Annotator) web application built to facilitate radar image annotation for generating training data

on the example route the weight for that feature is reduced (or increased).

For detailed description and analysis of the MMP we refer the reader to the original works [11] and [1]. The MMP algorithm follows a gradient descent scheme to update the weight vector iteratively: at each iteration i of the MMP the sub-gradient in Eq.5 is calculated for all the training examples (in a batch style for computational efficiency) and then the weight vector is updated given a learning rate η:

$$w_i = w_{i-1} - \eta \nabla O \tag{5}$$

V. Training and Validation Results

In this section we describe the results of training and evaluation of the MMP algorithm for solving various weather navigation problems defined using radar intensity images.

A. Training and test datasets

We designed and developed a web-based application called AWRIA: Airborne Weather Radar Image Annotator (see Fig.3), through which pilots can easily iterate through radar intensity images and annotate them by drawing routes and adding comments, and when done export the results in image and text formats.

To build a training dataset we searched for and downloaded a set of 10 radar intensity images from various sources available on the internet. Figure 4 shows each image segmented based on the color intensities (original images are not shown for copyright reasons). After some minor clean-up we curated them into a dataset for our test pilots to annotate using the annotator app. We created 3 copies of each original image and on each copy we specified a different goal location (with the same start location in all images). In total we ended up with 30 images each representing a unique weather navigation problem: a radar intensity image, the start location, and the goal location. Then using the annotator app one of our test pilots annotated the 30 images by drawing a viable route on each image and saving the results as new images with routes overlaid. Some of the segmented images with the routes annotated by our pilots are shown in Fig.5. The routes planned by the human pilots are then used for training and validation of the learned weight vector produced by MMP algorithm.

We sliced the 30 annotated images into two sets, a training set of 20 images and the remaining as a test set containing 10 images. We left the test set aside for validation and we passed the training set through our implementation of the MMP algorithm. We used an image segmentation technique based on HSV colorspace to segment each image into 5 colored features which form the feature vector used by the MMP algorithm. The output of MMP is the learned weight vector using which a planner is expected to produce routes similar/close to human pilot routes.

B. Evaluation and validation

In order to evaluate and validate the weight vector learned by MMP, for each test image we generated a costmap using the learned weight vector. We then ran an A* planner using the generated costmap to find the optimal route from the respective start to goal locations for that training example. Fig.6 shows the routes (in red) generated using the weight vector learned by the MMP algorithm along with the corresponding pilot annotated routes (in cyan) on the costmaps (in grayscale) learned by the MMP algorithm for each image.

C. Pilots evaluations and feedback

In order to evaluate the routes produced by the learned weights we asked our test pilots to evaluate and provide feedback on the routes generated by the planner using the learned weight vector. In what follows we have categorized their feedback into 3 main groups:

1) Unnecessary or too many heading changes: one major feedback point was the fact that the learned routes make too many heading changes and seemingly in some cases unnecessary ones. This result is not due to the learning process, but is an artifact of using a greedy planner (A*) on a discretized state space, i.e., a grid. The A* planner is implemented on

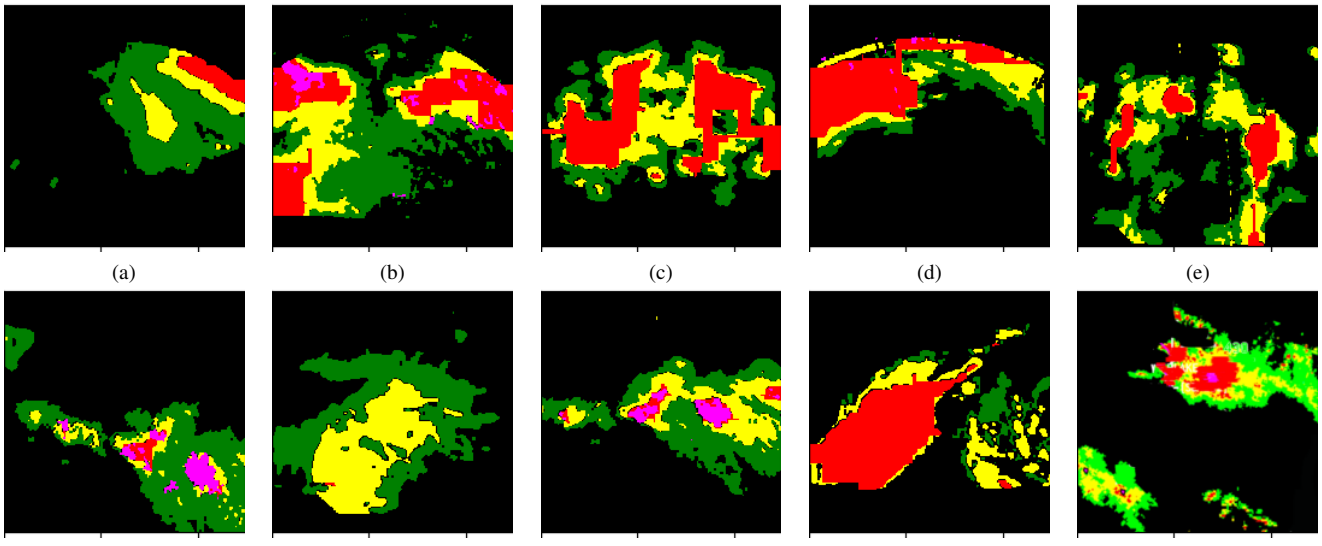


Fig. 4: Example radar images segmented based on the color intensity levels: green, yellow, red, magenta

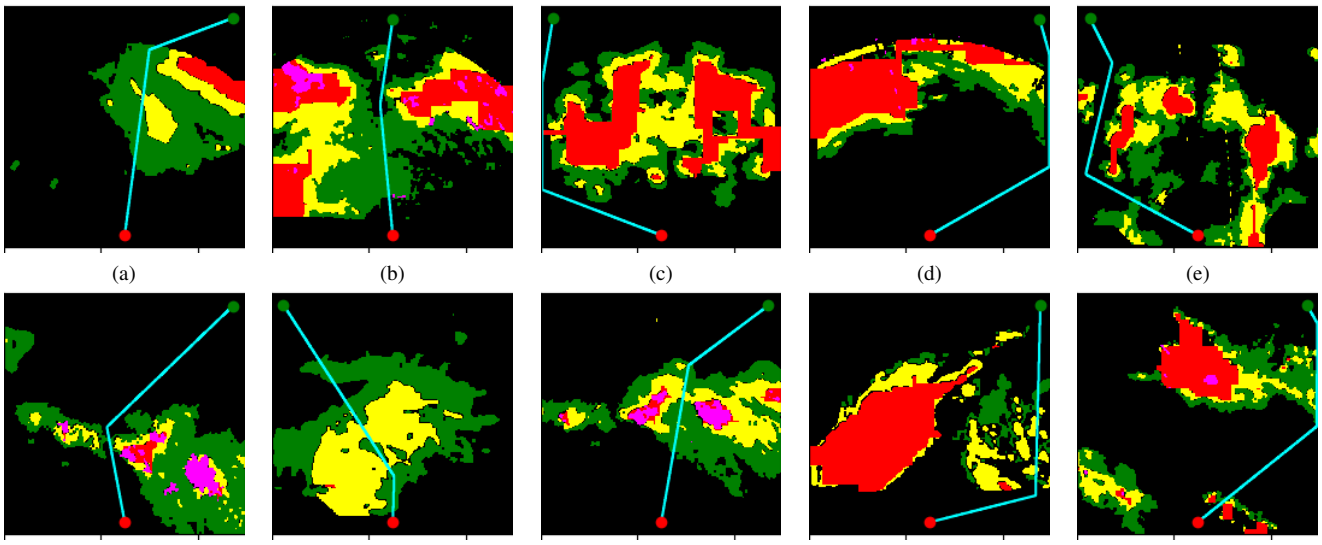


Fig. 5: Segmented radar images annotated with routes (in cyan) by a human pilot using AWRIA app

a grid with 8-neighbor connection, making it restricted to move only up, down, right, left, diagonally up-left/right, or diagonally down-left/right. Consequently, the implemented A* is not an any-angle planner [14] and this leads to sub-optimal and zig-zag routes as is more clearly visible in some of the figures in Fig.6, for example (h) and (i). Moreover, due to the greedy nature of A* it makes frequent heading changes as soon as a lower cost neighbor is immediately available. One solution to this problem, and a future work to this research, is to adapt an any-angle planner, e.g., Field D* [15] which uses interpolation during each vertex expansion to find near-optimal paths through regular, nonuniform cost grids. An intermediate solution is to apply a post shortcutting step [16] to further smooth the route by replacing segments of the route with shorter/direct shortcuts with a slight, yet bounded, increase in the overall cost of the route in some cases. The shortcutting algorithm we implemented starts off from the beginning of the route and tests whether any shortcut can be made to any of the points along the route where a heading change has happened. The search for the end point starts from the last point on the route and moves backward. The condition for a shortcut to be added is that the cost of the shortcut is less (or up to a threshold larger) than the original segment of the route being shortcutted. Where a shortcut is viable the route gets updated with the shortcut and the algorithm continues to run from the end point of the shortcut over the remaining of the route proceeding the shortcut segment. The shortcutting technique here is a brute force approach and continues until the search for a shortcut reaches the end of the route and all possible pairs of points (with heading changes) on the route

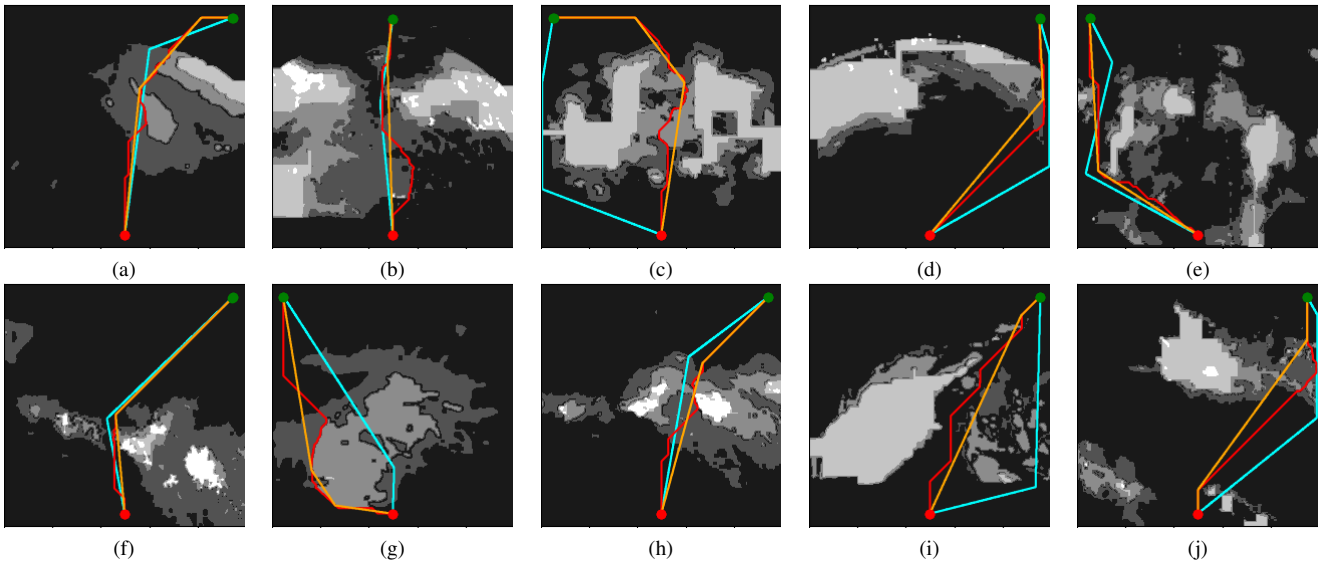


Fig. 6: Validation results on the test dataset: pilot annotated routes (in cyan), the routes generated using learned weights (in red), and the shortcutted version of the learned routes (in orange) overlaid on the learned costmaps shown in grayscale

are explored. In Fig.6 the shortcutted learned routes are shown in orange. As shown and confirmed by our test pilot, there is a noticeable reduction in the number of heading changes in most cases (for example, compare red and oranges routes in Fig.6(h), or 6(i)).

2) No safety buffer close to the danger zones: another feedback from pilots was the fact that in some cases the learned routes are skirting the boundary of red/magenta regions and getting too close to the zones with adverse weather (for example see the red/orange routes in Fig.6(e) where they get very close to the red spots on the left side of the image while the pilot route steered away). FAA guidelines also advise to avoid by at least 20mi any thunderstorm that is identified as severe or is giving an intense, heavy, or extreme radar return [17]. This is due to the fact that while radar returns provide information about precipitation, there are other dangerous weather conditions not shown on radar that may extend a few miles away from the edge of a strong radar return. For example, strong updrafts and downdrafts associated with convective systems and hazardous turbulence may extend to as much as 20mi from the strong return edge. The lack of buffer with respect to the danger zones is another artifact of the greedy nature of the A* planner, but could also be an indication for the lack of a feature as a measure of closeness to danger zones so it can be learned from pilot routes. Eroding (i.e., padding) the danger zones could be a temporary solution to this problem. However, the principled way of handling this is to add a feature to the feature vector F_s that measures the distance to the closest red/magenta region from every state s. Then the weight for this feature will be automatically learned using the MMP framework without any extra work. This could be a rather simple but effective improvement for the future work.

3) Overly cautious routes: in a few cases it was observed that the learned routes are overly cautious trying to avoid green regions at the cost of increasing the length of the route (for example see Fig.6(a)) or adding extra heading changes (for example see Fig.6(b)(g)). In some cases this behavior is an artifact of the imbalance between the length of the route and the overall feature cost of the route. The shortcutting technique suggested above would resolve some of these cases (e.g., Fig.6(b)). There is also the possibility that since in the training examples there were not many instances of pilot routes going through green/yellow regions the algorithm did not get much opportunity for learning to lower the gains of the green/yellow features further. Including more of such examples during the training could help with verifying this hypothesis.

VI. Conclusions and Future Work

The work presented here is a proof of concept for learning weather navigation using airborne weather radar data. We formulated the problem as an inverse control problem which seeks to find a cost function that explains the training examples provided by human pilots. We implemented and applied the Maximum Margin Planning algorithm as an elegant way of solving the optimization problem for finding the weighting between the features encoded in radar intensity images. We evaluated and validated the effectiveness of the proposed framework on datasets consisting of radar intensity images annotated in-house by our test pilots. The observations and feedback received from the test pilots shed some light on some directions the work here can be extended and improved towards building a real-world weather navigation component as part of an autonomous flight system, for example:

A. Including other sources of weather data

The MMP formulation enables adding more features as part of its optimization framework. For example, as mentioned

earlier, in addition to airborne radar data, pilots make observations about the weather by looking out the window, as well as receiving reports from other aircraft in the vicinity, ATC, or NEXRAD. These can be added as features to the feature vector defined at every state. The challenges for perceiving and digesting such information into vectorized features is a separate problem, and an essential pre-requisite.

B. Extension to routing in 3-D

Expanding this work to 3-D is rather straightforward. The main challenge is to expand the state representation to 3-D (instead of a 2-D grid) and calculate the feature values at each 3-D state during annotation and training. The MMP algorithm does not make any assumption about the dimensionality of the state space and the A* planner used here can be simply extended to 3-D. The main challenge here is to create a 3-D radar intensity image which requires operating the tilt feature of the radar. In some more advanced radars (e.g., Honeywell IntuVue RDR-4000) such a function is automated and can be used to build a 3-D intensity image.

C. Dynamic planning and re-planning

We acknowledge the fact that in this work we treated the radar data as static images. Although such an assumption does not mesh with reality (due to weather changes, radar setting changes performed by the pilot, or ownship motion) the approach used here for learning the weights and building the cost function is still applicable to real scenarios. However, due to the dynamic nature of the weather and changes in the state of the radar and/or the ownship it would not be practical to generate a fixed route and stick to it for execution until the end. Instead we envision a dynamic planning and re-planning approach, i.e., receding horizon planning and execution. In other words, at every cycle of planning and execution we only execute a small part of the route and then in the next cycle a new route is generated (with the most recently updated information and state) and the execution continues on the new route until the next cycle when a newer route is generated. With such a scheme it can be ensured that at each cycle we use the latest information based on the current state of the environment and ownship.

D. Handling radar shadows

Another interesting future direction to this work is to look into avoiding radar shadows, i.e., regions of the space behind the severe weather regions where there is no reflectivity due to high attenuation of the radar signal. Attenuation may exist when a storm cell absorbs or reflects all of the radio signals sent by the radar system. Attenuation may prevent the radar from detecting additional cells that might lie behind the first cell. Pilots are expected to have good understanding of radar shadows when looking at the radar images and avoid routing through them. Introducing a feature to denote such regions of the radar image as forbidden (i.e., very costly) is a straightforward way to handle them in the optimization problem and finding their appropriate weight in the cost function. Of course, that is given that these regions can be detected and identified on the images, which is an interesting research problem by itself.

ACKNOWLEDGMENT

We are grateful to Merlin's test pilots for their valuable feedback and discussions and their help with generating training examples and evaluating the results. We would like also to thank Merlin's Advanced Capabilities Team for their feedback during the research and development phases of this work.

REFERENCES

[1] Silver D, Bagnell JA, Stentz A., *Learning from Demonstration for Autonomous Navigation in Complex Unstructured Terrain*, The International Journal of Robotics Research, 2010;29(12):1565-1592.

[2] *Foreflight Preflight & In-flight Weather Planning*, https://foreflight.com/products/foreflight-mobile/weather/

[3] McNally, D., Sheth, K., Gong, C., Love, J., Lee, C. H., Sahlman, S., and Cheng, J. , *Dynamic Weather Routes: A Weather Avoidance System for Near-Term Trajectory Based Operations*, 28th International Congress of the Aeronautical Sciences, 23-28 September 2012.

[4] McNally, D., Gong, C., Lee, C. H., *Dynamic Arrival Routes: A Trajectory-Based Weather Avoidance System for Merging Arrivals and Metering*, 15th AIAA Aviation Technology, Integration, and Operations Conference, 22-26 June 2015

[5] Olin, Karen E. and David Y. Tseng., *Autonomous cross-country navigation: an integrated perception and planning system*, IEEE Expert 6 (1991): 16-30

[6] Stentz, A., Bares, J., Pilarski, T., and Stager, D., *The crusher system for autonomous navigation*, In AUVSIs Unmanned Systems, 2007

[7] 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

[8] Kalman, R., *When is a linear control system optimal?*, Trans. ASME, J. Basic Engrg., 86:51–60. 1964

[9] Ng, A. Y. and Russell, S., *Algorithms for inverse reinforcement learning*, In Proc. 17th International Conf. on Machine Learning, 2000

[10] Abbeel, P. and Ng, A. Y., *Apprenticeship learning via inverse reinforcement learning*, In International Conference on Machine learning, 2004

[11] Ratliff, N., Bagnell, J. A., and Zinkevich, M., *Maximum margin planning*, In International Conference on Machine Learning, 2006

[12] Silver, D., Bagnell, J. A., and Stentz, A., *Perceptual interpretation for autonomous navigation through dynamic imitation learning*, International Symposium on Robotics Research, 2009

[13] Trammell, A., *Convective weather flying (2nd edition)*, course manual, 2017

[14] Nash, A., Koenig, S., *Any-Angle Path Planning*, AI Magazine, 34(4), 85-107, 2013

[15] Ferguson, D., Stentz, A., *Field D*: An Interpolation-Based Path Planner and Replanner*, Proceedings of the International Symposium on Robotics Research, 2005.

[16] Geraerts, R., Overmars, M. H., *Creating high-quality paths for motion planning*, The International Journal of Robotics Research, 26(8), 845-863, 2007

[17] *FAA-H-8083-28: Aviation Weather Handbook*, Last updated: December 22, 2022