INJECTING INTELLIGENCE INTO DRONES

Sadhana Kumar,[1] Sarah Tan Si Yu[2], Wong Ruiming Jeremy[3]

[1]Cedar Girls' Secondary School, 1 Cedar Ave, Singapore 349692
[2]Raffles Girls' School (Secondary), 2 Braddell Rise, Singapore 318871
[3]Defence Science and Technology Agency, 1 Depot Road, Singapore 109679

Abstract

Imagine numerous identical drones flying in perfect coordination for a light show, often the result of meticulous planning. Sounds feasible? Now imagine if the choreography changed on the fly—an inevitable occurrence if they were flying for logistics or transport instead. A malfunction could cause collisions—a financial and physical catastrophe. In order to live in smart cities with safe automated drones at our disposal, this is a problem we need to address. In previous works, drones were equipped with learning-based controllers. As individual agents, they cooperated with each other to reach waypoints without collision, via decentralised neural barrier certificates based upon the control barrier function (a formal method that defined and constrained the safety of the learning-based controllers). In this project, we aim to improve the safety of these agents by training them to deal with malfunctions or even malice. We draw upon the concept of Generative Adversarial Networks (GANs), where two neural networks compete with opposing objectives, forcing each other to improve in the process. Similarly, agents trained to avoid malicious drones fare better on safety than ones trained solely in cooperative environments. By combining GANs with the initial model, we successfully achieved a noticeable improvement in its safety rates.

Introduction

This project builds on a previous research paper "Learning Safe Multi-Agent Control With Decentralised Neural Barrier Certificates" [1], which details a system comprising a multi-agent control policy and control barrier functions, implemented in a decentralised fashion that can easily scale up to 1024 or more agents. Each agent can travel from their initial locations to target locations without collision and without a central coordinator. However, the system assumes that all agents are cooperative and fully functional. While the control policy can be trained for different inputs and outputs such as 2D or 3D position, velocity, and acceleration, we focus on drones controlled by acceleration commands and flying at a constant altitude such that congestion is greatest. We combine [1] with the concept of Generative Adversarial Networks (GANs) in order to train these agents to achieve a higher level of safety. GANs is a machine learning model in which two neural networks compete with each other to become more accurate in either generating or discriminating images [2]. Similarly, we train different agents to compete with each other to become safer or more dangerous in their control.

This system can be used in self-driving cars or drone delivery services and is implemented directly onboard the vehicles. As it does not require a central coordinator, it is particularly useful in situations with multiple disparate parties, such as in an unforeseen emergency, and especially where there may not be the time, resources, or even technology for a central coordinator. As an example, one could imagine such a system used for search-and-rescue with autonomous vehicles in a complex cave network where a central coordinator may not be able to communicate with all vehicles due to line-of-sight issues.

Hypothesis

Our hypothesis is that agents trained to compete with each other on safety (ie. to minimise or maximise collision rate) can be even safer than they have been trained in [1]. As an analogy, we are training agents to pick up "defensive driving skills" by introducing them to hazards on the road, such as distracted, aggressive or even dangerous drivers.

Methodology

Training 'bad actors'

Firstly, we trained dangerous agents, or 'bad actors', by reversing their idea of safety (which is to avoid collisions with one another) and in doing so, reversing their safety rate. Similar to [1], the safety rate is defined as the percentage of instances across all instances where two or more drones moved within a predefined distance from each other (the safety radius).

Initially, we tried to train bad actors to reach their own goal positions while simultaneously training to collide with other agents. However, this proved to be ineffective as unlike good actors who mostly tried to reach their goals whilst occasionally avoiding collision, bad actors always wanted to move within the safety radius of another agent. Thus, we trained them to focus only on minimising the safety rate and completely ignoring goal positions. This was successful and with bad agents, many more collisions occurred compared to that of the baseline. The baseline used here is similar to the baseline used in [1], a Linear Quadratic Regulator (LQR) controller. The LQR controller simply minimises the time required and positional error to the goal position, based upon available control outputs (such as maximum velocity or acceleration).

To compare with the safe agents, or 'good actors', trained in [1], we used a similar number of training iterations (70,000). In [1], each iteration generated a random start and goal position for every agent, and goal-reaching was rewarded while collisions were penalised. These were calculated via the cumulative remaining distance of all agents to their goal points, as well as the safety rate respectively. Contrary to this, in training bad actors we ignored the cumulative remaining distance, and reversed the safety rate penalty.

The difference in safety rate from the baseline LQR controller decreased from a positive 3.2% to a negative -5.6% from the respective LQR safety rates (seen in Fig 1.1 and 1.2).

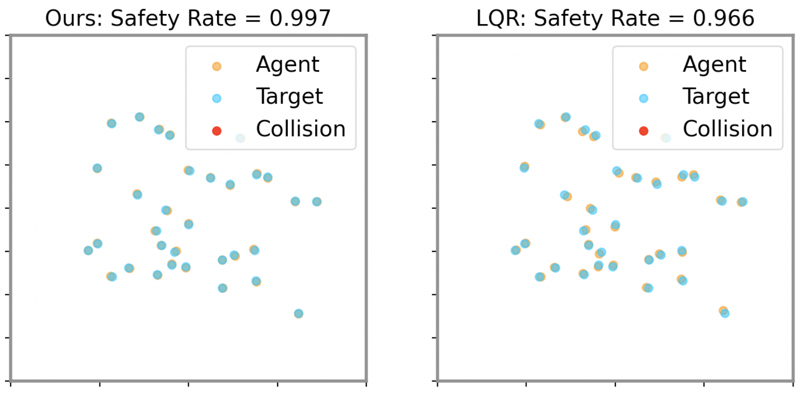

Figure 1.1: 'good_actors' (original model) with 70,000 training steps

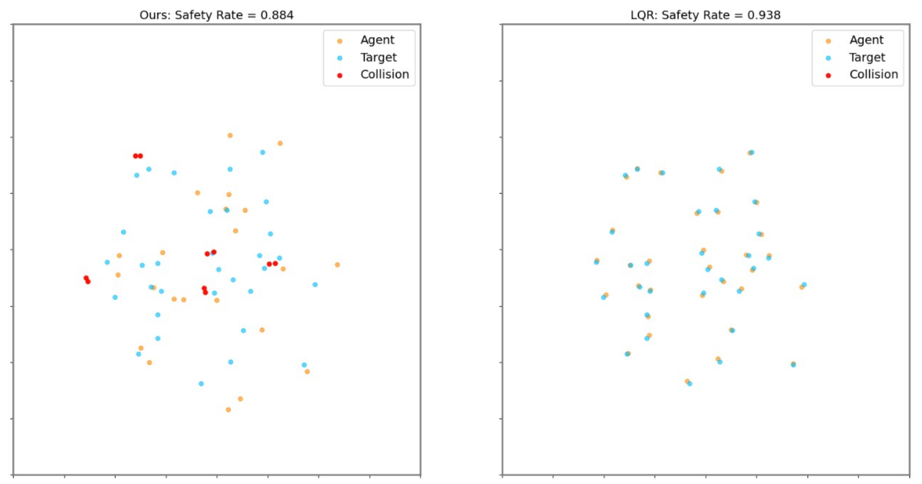

Figure 1.2: 'bad_actors' (original model) with 70,000 training steps

A possibility we thought of was retraining the 'bad_actors' model a greater number of times in order to make the bad actors' safety rate reach a minimum. However, in the case of a minimum safety rate, all the bad actors would have to be congregating at a fixed point to increase the collision rate as much as possible. In this case, the good actors could easily avoid them by learning to avoid the fixed point in which the bad actors gathered, treating the bad actors as no more than a stationary obstacle. This would defeat the purpose of our project since the model loses its competitiveness, and such behaviour would be unrealistic in a real-life scenario. Hence, we decided that the current model would suffice.

Training good actors with bad actors in the system

Next, we introduced the bad actors directly into the training environment for the good actors. The hypothesis was that this would have the effect of training the good actors to avoid each other, including the bad actors, even better than before.

When we first combined the bad actors with the good actors (Fig 2.1), we observed a safety rate of 0.939, which was 0.018 lower than the baseline of 0.957. This was expected, as the bad actors introduced were trained to collide with other agents.

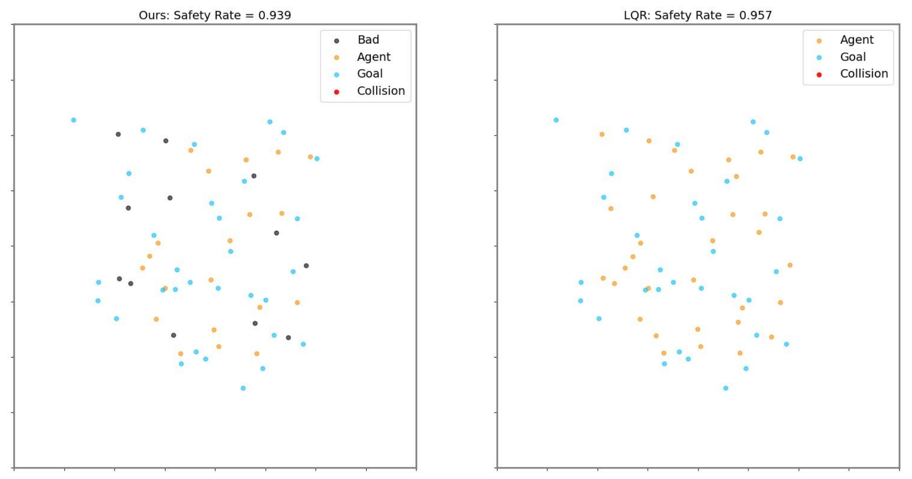

Fig 2.1: 20 original functional agents with 12 defective agents

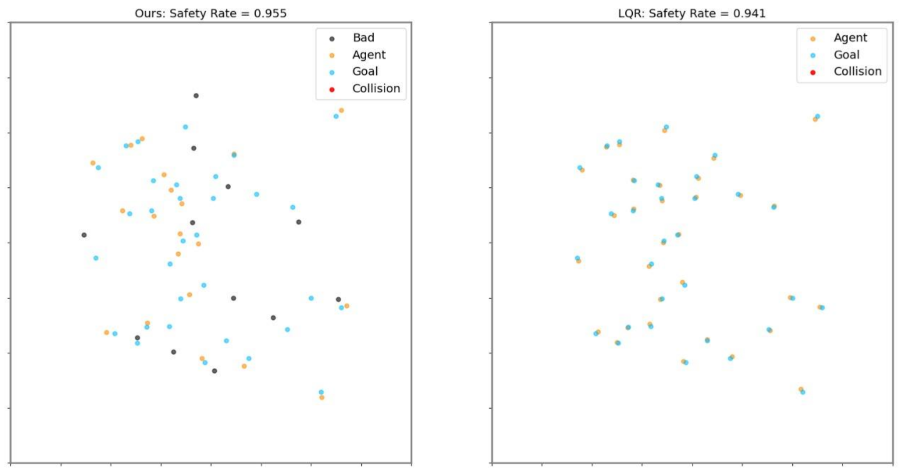

Fig 2.2: 20 retrained functional agents with 12 defective agents

As seen from the above figures, the safety rate improved from 0.018 lower than that of the LQR before training with the defective agents (Fig 2.1), to 0.014 higher afterwards (Fig 2.2). The difference in safety rate from the baseline LQR controller thus increased from a negative -1.88% to a positive 1.48% from the respective LQR safety rates (seen in Fig 2.1 and 2.2).

However, we believed that the overall increase in difference in safety rate from baseline of 3.36% (from Fig 2.1 to 2.2) could possibly be higher. In Fig 2.2, the good actors were avoiding all others as if they were bad actors, resulting in the uniform treatment of the good actors to all other actors surrounding them, where regardless of an actor being good or bad, the good actors would spend the same amount of effort avoiding them. Unfortunately due to time constraints, we were unable to further optimise the amount of effort expended by training the good actors to respond differently to bad actors, though we hope to achieve it in future work (if possible).

Conclusion

We have successfully managed to train safer agents using the concept of GANs, from 98.1% as effective as the control model (LQR) in the presence of bad actors, to 101.5% as effective after re-training with the bad actors. We have successfully engineered the original model to be applied in the real world, by allowing it to maintain its high level of performance even when faced with defective agents in the environment. During the process of engineering the original model to make it applicable to real-life, we faced some challenges. These challenges included analysing hundreds of lines of code to accurately identify which sections of the code contained the lines relevant to our project goal.

A limitation of our project is that our novel model has not yet been tested with physical drones, and therefore we are unable to accurately determine its effectiveness in the real world. For example, the real-life dynamics of drones may be affected by environmental factors such as wind. While such factors can also be addressed by learning-based controllers such as adaptive flight control of drones [3], this was not the focus of our project. Moreover, it took a long time to train each model due to the computational cost of machine learning. There were also time and resource constraints. To avoid wasting time and effort, we worked closely with our mentor to minimise mistakes in code before computational resources were used to train the models.

If given the opportunity to further develop this project in the future, we hope to make use of physical drones to test our model, and so that we can identify the more important aspects of the model to improve upon for real-world efficacy. We also hope to train good actors to identify bad actors without prior information, as real systems may not have prescient knowledge of buggy or bad actors within the system. With further research, the learning-based controller could also learn to account for different dynamics due to physical or design differences. For example, racing drones may be more manoeuvrable than cargo-lifting drones, and therefore more likely to give way to the cargo-lifting drones than vice-versa.

We believe our findings can have many real-world applications. In the military, our model can improve the lethality of unmanned aerial vehicles and serve as a precaution to prevent collisions between autonomous agents. In e-commerce and delivery services, it can serve to increase the efficiency of delivery drones or drones used for other commercial services, saving time and transportation costs. Further research can be done to model the movement in a three-dimensional space and the behaviour of trained drones in a real-life scenario with the presence of other factors such as wind and humidity. We also believe that with further exploration, the autonomous agents could be trained to vary their velocities to avoid collisions rather than only changing their directions. In a world where automation and machine learning are becoming increasingly prevalent, we believe that our findings will allow for the safer, simultaneous use of multiple autonomous agents.

Acknowledgements

We would like to thank Mr Jeremy Wong for your unwavering support and guidance throughout our project, our teachers-in-charge Dr Wu Jiang and Mr Shaun De Souza for keeping track of our progress, and the authors of [1] for inspiring us to develop our research.

Bibliography

[1]: Qin, Z., Zhang, K., Chen, Y., Chen, J., & Fan, C. (2021, April 17). [2101.05436] Learning Safe Multi-Agent Control with Decentralised Neural Barrier Certificates. arXiv.

[2]: GAN and its Applications: Everything you need to know. (2022, June 13). Daten & Wissen.

[3]: O'Connell, M., Shi, G., Shi, X., & Chung, S.-J. (2021, March 2). [2103.01932] Meta-Learning-Based Robust Adaptive Flight Control Under Uncertain Wind Conditions. arXiv.

Annex

Code: https://github.com/sarahtsy/ro015_iiid