Machine Learning for Autonomous Drone Racing; Assignment 1

CSM6420 | Tob31

Contents

Introduction

Autonomous drone racing is a new sport that has developed from standard drone racing the goal is to race drones on an elevated track where the drones need to pass through gates for a set number of laps. Racing autonomous drones or micro aerial vehicles represents a difficult and challenging area for the field of robotics since without human pilots they struggle with agile maneuvers and dynamic environments. The paper "Beauty and the Beast: Optimal Methods Meet Learning for Drone Racing"[1] aims to improve these short comings by utilizing machine learning and optimal control methods to allow the drones race through unseen or mapped tracks. In the case of this research the track is formed by using tent like gates which can be easily moved adding a dynamic element to the track. Many of the previous and current autonomous drones have need to have a preprogrammed 3D map of the track to guide its path following.

[Fig1]

Description of Main Problems being tackled.

The main issues this paper [1] aims to tackle is that the current drones have difficulty performing agile maneuvers in dynamic environments, the current algorithms have inconsistent computational overhead, the environmental sensors are heavily affected by changes in appearance such as varying lighting and previous successful approaches include either using large amounts of collected track data or building of precises preprogrammed 3D maps, additionally storing these 3D maps could be weight intensive as the 3D maps would be large files. The developments in this paper aim to allow the drones to quickly adapt to a dynamic environment without the need for a preprogrammed 3d map or a large data set and to increase the drone's overall agility and efficiency making this implementation practical solution for autonomous drone racing.

This solution proposed will allow the drone to estimate the location of these course gates before flying the path. During the race, the CNN (convolutional neural network) will predict the position of the closest unpassed gate and its uncertainty. The predictions us an EFK (extended Kalman filter) which will maintain the optimal gate location estimate, likely in the center of the gate. This will allow the drone to navigate the track using this predictive robotic control model with increased agility and accuracy previously unseen in autonomous robotics.

Identification of The Machine learning Solution Used

The main Machine Learning solution used for this project is a CNN (convolutional neural network) it is used to predict the position of closest gates to drone that have not yet been passed. This ML approach utilises several key methods, data input, feature extraction and data processing. In this identification section of the methods used, the approach will be summarised and described almost directly from the academic paper [1] and is not intended to be the work of this document's writer.

Perception system

Data input step: environmental images are collected from the drones forward facing camera, these images are taken as 320 x 240 RGB images and regresses both the mean and variance, each image is then associated with is relative position from the gate in front with respect the drone itself, this process is then able to set up the drone's path to the gate setting up the network for training without needing for the drone to complete a full lap of the track.

Mean = $\boxed{\tilde{\mathbf{z}}_{BG_l,pol} = [\tilde{r}, \tilde{\theta}, \tilde{\psi}, \tilde{\phi}]^\top \in \mathbb{R}^4}$ Fig1

Variance = $\boxed{\tilde{\boldsymbol{\sigma}}^2_{BG_l,pol} \in \mathbb{R}^4}$ Fig1

Feature extraction step: The CNN processes the images taken using the front facing camera and then extracts the relevant features from it. These extracted features are processed by two separate MLPs (multilayer perceptron's). The neural network then estimates the variance and mean multivariate normal distribution this is used to estimate the position of the next unpassed gate.

Training step, using the mean and variance estimation from the neural network the associated uncertainty is then calculated, this step allows the network to make more accurate decisions using the previous data input with the Kalman filter when in an uncertain position. The network is trained in two stages, In the first stage the parameters of the ML_z and CNN are denoted by θ_z and θ_{CNN} are jointly learned by minimizing the loss over the ground truth position for the images where the gates are visible.

$$\boxed{\{\boldsymbol{\theta}^*_{\text{CNN}}, \boldsymbol{\theta}^*_{\tilde{\mathbf{z}}_i}\} = \arg \min_{\boldsymbol{\theta}_{\text{CNN}}, \boldsymbol{\theta}_{\tilde{\mathbf{z}}_i}} \sum_{i=1}^{N} ||\mathbf{y}_i - \tilde{\mathbf{z}}_i||_2^2,}$$

Fig1

The second stage, the training data is set to include all the images that do not have a visible gate in them in addition to ones that do. Only the parameters $\theta_{\sigma 2}$ for the subnetwork $MLP_{\sigma 2}$ are trained in this stage however the other weights are kept fixed and do not change. The loss function is minimised creating a "negative log-likelihood a multivariate normal distribution this has an uncorrelated covariance"[2]. The loss function was proposed in the another paper titled "Estimating the mean and variance of the target probability distribution." [3] Using this method for mean variance estimation solves one of the main issues targeted with this paper which is that it is a computationally efficient method for calculating the uncertainty estimate. This method was referenced in the from another source [4]

$$\boxed{-\log p\left(\mathbf{y} \mid \tilde{\mathbf{z}}_i, \tilde{\boldsymbol{\sigma}}^2\right) \propto \sum_{j=1}^{4} \log \tilde{\sigma}_j^2 + \frac{(y_j - \tilde{z}_j)^2}{\tilde{\sigma}_j^2}.}$$

Fig1

Data generation step: The images collected from the drones forward facing camera system associated relative positions based on it own and its distance from the gate. The drone or quadrotor uses it onboard state estimation pipeline to generate the training data. The platform is initialized to a known position that is relative to a gate, this step then iterates when the environment is changing, and the drone is collecting images as it moves. The perception only needs to estimate the relative position in relation to the next gate thus a single gate placed in varying environment is sufficient, the perception system is never trained using the environmental image data that it will be tested on later. This allows the drone to complete the track without the need for it have traversed it before.

Mapping system

The mapping system corrects the gate estimates with the measurements from the CNN, this allows for gate displacement and accumulation VIO drift to be compensated. It has two main stages the measurement assignment stage and the EKF stage.

Measurement assignment: A map of all the gates is maintained corresponding to its angular position from the drone."[1] The output from the perception is used to update the position of the next forward gate", this measurement is then transformed to the odometry frame and assigned to the next gate. If the gate is not the closest forward gate, then it will discard it. The next gate is monitored by detecting of the drone as traversed that specific gate.

EKF (Extended Kalman Filter): To integrate the neural networks predictions efficiently on to the map including the prior knowledge of gate traversals, each gate has its own EKF and at each step the prediction calculated the measurements and the variance.

Planning and control system

Waypoint Generation: Two waypoints are generated for each gate in the map one for the center of the gate and the other for the drone. A path from each waypoint is then linearly interpolated, this path is then referenced by the controller for the drone to follow.

Model Predictive Control: The control problem is formulated as a quadratic optimisation problem solved using sequential quadratic programming.

$$
\begin{aligned}
&\min_{\mathbf{u}} \int_{t_0}^{t_f} \left(\bar{\mathbf{x}}_t^\top(t)\mathbf{Q}\bar{\mathbf{x}}_t(t) + \bar{\mathbf{u}}_t^\top(t)\mathbf{R}\bar{\mathbf{u}}_t(t) \right) dt \\
&\qquad \bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_r(t) \quad \bar{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_r(t) \\
&\text{subject to} \quad \mathbf{r}(\mathbf{x},\mathbf{u}) = 0 \qquad \mathbf{h}(\mathbf{x},\mathbf{u}) \leq 0.
\end{aligned}
$$

Fig1

States x and inputs u are given weightings with a positive diagonal matrix Q and R respectively to their references.

Results

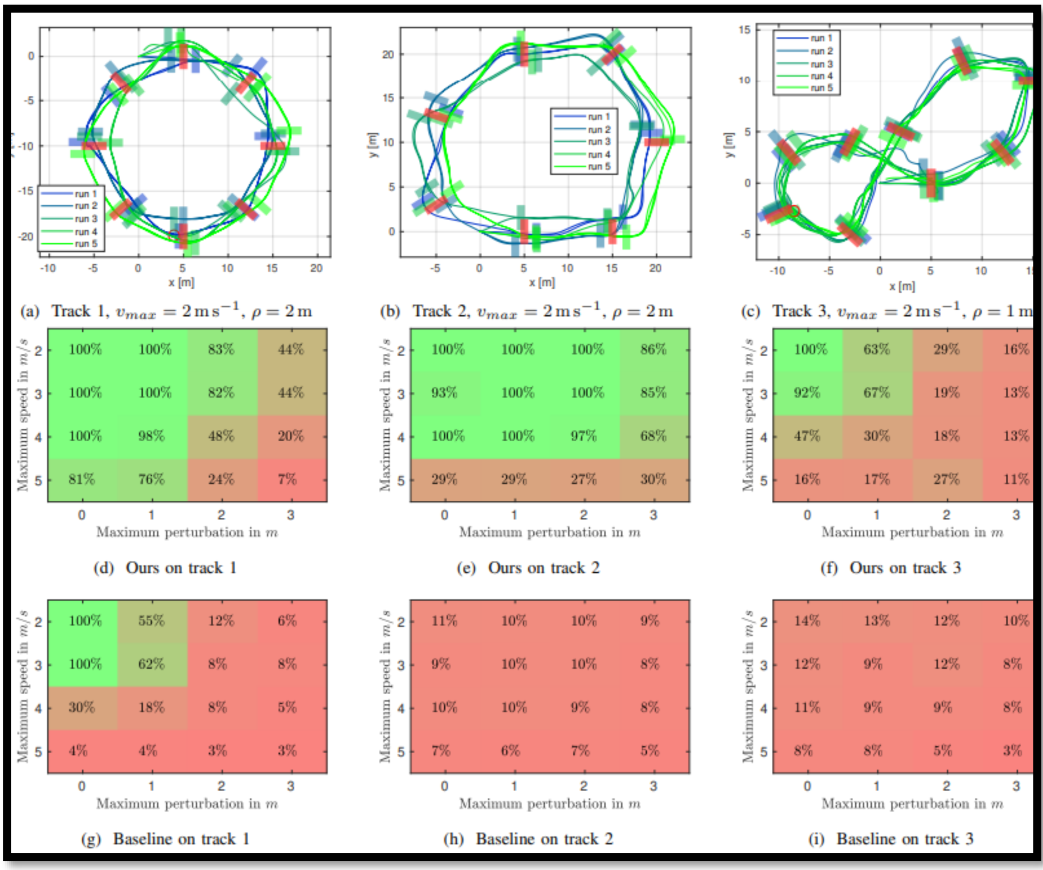

Fig1

The results above were simulated using the Gazebo software using a dynamic or moveable gate structure this meant that the gates would be moved by the researchers during the run to make the track more dynamic and difficult for the drone to complete. The top row is the simulated results using the main machine learning method and the bottom row is a purely visual method without machine learning. The Machine Learning method was far superior for completing the track however the figure of 8 track still presented some issues for the drone.

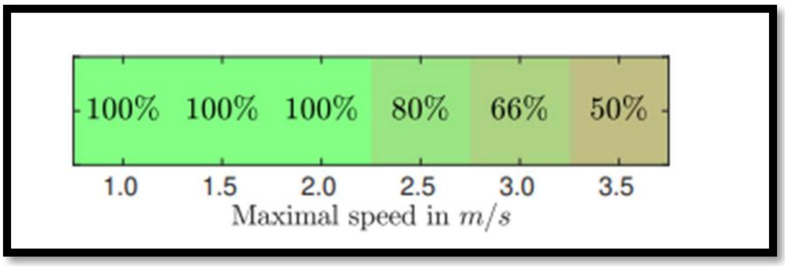

Fig1

The physical results above aligned with the simulated results in relation to the maximum optimal speed of the drone.

Main issues for consideration

For this application of the machine learning solution several technical issues need to be considered. Looking at the results data from the physical simulation the drone speed was an issue the experiment only saw competitive success rates at around 1ms-1 to 2.5ms-1 when going to 3.5ms-1 the success rate dropped to 50%, for comparison typical human controlled racing drones can reach speeds of more than 44ms-1.

The drone was able to successfully handle to dynamic aspect of the gate displacement however when at less than or equal to two meters way the drone experienced a significant drop in performance, the drone was not quick enough to react to changes in the environment without sufficient distance and time.

The use of the CNN and the Extended Kalman filters is extremely computationally expensive which can lead to delays in the estimation calculations made. This aspect will require the network architecture and the systems inferences to be optimised.

The environmental uncertainty's such as wind speed, lighting changes and other types of sensor noise as well as the opponent's behavior in the race can have an effect on the drone's performance, since the system was trained with only one drone going through the track on an indoor track with constant lighting and wind speed. Drone collision avoidance was also not a factor that these researchers considered when attempting this work, the drone system would require a hybrid reactive control system to accommodate for potential collision with other racing drones.

The Lack of sufficient training data was also an issue, large and diverse amounts of training data are needed to train these models to generalise well and allow for the drone to experience and fly new tracks and environmental conditions it hasn't experienced before, example of this is when the model encounters the figure of eight track, the model was trained to handle this type of track, since the model gets confused when entering the crossing section believing the other gate next to it is the closest on the path to follow since it is closest to the drone.

An Alternative Machine Learning Approach

A hybrid LSTM Long Short-Term Memory network approach which is a type on RNN recurrent neural network could be used as an alternative for improving the drone's agility and prediction capabilities. In a recent paper titled "On-board Deep-learning-based Unmanned Aerial Vehicle Fault Cause Detection and Identification" by V. Sadhu et al[2], this method was used to develop a novel architecture for fault cause identification and detection which was 90% accurate for detection and 85% accurate for accuracy and combined these were 99% accurate for the simulation data[2]. In this identification section of alternative methods used, the approach will be summarised and described almost directly from the academic paper [1] and is not intended to be the work of this document's writer.

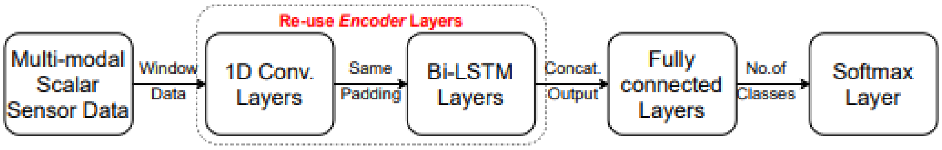

Fig4

The LSTM approach.

The bidirectional LSTM for anomaly detection autoencoder approach is able to take in and analyse temporal data from the drones' sensors so that it can understand the patterns that indicate the drone's normal operation compared to its operation when experiencing faults, this process while different can be used to make the drone racing algorithm more efficient, this method has two stages Encoding and Decoding.

In the encoding stage the 1 dimensional convolution layers capture the spatial correlations between the data taken from the sensor channels, the bidirectional nature of this allows the model to capture these patterns from over time from both directions which is beneficial when working with temporal/time-series data as the predictions of the drones future state depends on both what it will experience and what is has experienced.

In the decoding stage the goal of the model is to reconstruct the input data from the encoding stage in reverse, so it incorporates a 1-dimensional deconvolutional layer. The model assumes that is has been trained on normal operational time data, when the drone encounters an anomaly that it hasn't encountered during training there is a higher error rate.

After the anomaly detection stage an LSTM classifier with a CNN is then used to identify what the fault to the drone is specifically, The convolutional layers extract the special features here, each channel is processed using multiple kernels creating feature maps for the next layer, the output is passed into the LSTM layers which capture temporal and dynamic changes. This model is able to detect when the drone changes from its normal operations but also identify what is wrong with its current operation so it can return to normal. The data ends into a softmax layer which calculates the probability of different classes. This identification stage only happens if the anomaly from the previous stage has been scored to be a high enough priority. This identification allows the drone to take appropriate actions need to fix the situation.

The benefits of this approach.

Convolutional neural networks are beneficial when used to process spatial data like RGB image inputs as well as other sensor readings making them perfect for this application. Furthermore, the Bi-LSTM approach used in this study is specialised so that it can process sequential data as well as temporal data. This hybrid architecture is beneficial for drone racing as it will be able to process and extract the relevant features then quickly calculate the measurements and variance in relation to the gates position from the drone in real time. This model should be robust and highly accurate, being able to work with unlabeled data in an unsupervised manner, as well as needing little training for new track environments once the model has been trained before.

Limitations and Opportunities for advancing the applications systems.

The main limitation with the current system application is that it relies on the data input from only the RGB 320x240 images taken from the drones forward cameras as well as the training environment being only a single basic track loop without any complex sections. This training approach does not capture sufficient detail that is needed to train the drones system of these highly complex and dynamic track environments additionally, the performance requirements of the drones hard-ware limit the advanced complexity of the machine learning system approach that can be loaded on to the drone as the weight the components effect the drone is ability to fly but also its eligibility to fly in the racing competitions.

There are a few opportunities for advancing the current system application which could be implemented one of these could be added more advanced sensors to the drones sensory array like lidars or even inferred sensors so that the drone can get a better understanding of its environment especially when that environment is dynamic. Integrating more computationally efficiently specialised hardware with improved machine learning algorithms which are optimised for speed on the specialized hardware would allow the drone to quickly adapt to changes in the environment as it will update its calculated estimates faster. Training the machine learning model on a more diverse and dynamic track layout that better represent the kinds of tracks the drone will race will allow the drone to complete these complex laps with minimal training.

Conclusion

The paper "Beauty and the Beast;" presents an advanced and comprehensive development for autonomous robotics in drone racing, utilizing an affective machine learning strategy and optimal control. While the proposed approach is effective for increasing the drone agility and accuracy at speeds up to 2.5ms-1, there are further opportunities for improvement and current limitations in the approach that could use further development to correct. For example, the drone is slow to reactive to a dynamic environment an alternative machine learning approach like LSTM coupled with more advanced sensor input could greatly improve the drone's adaptability to the dynamic environment and its overall agility and accuracy, however this method could require more computational processing power. The possible applications and development of the method from the paper could include improved Mars rover path planning, Manufacturing on a factory line, delivery of parcels, surveillance, military and defense. Autonomous path planning for Mars rovers is particularly important for furthering science as controlling these rovers is almost impossible in real time and many rovers are often lost in dust storms, thus the machine learning approaches referenced in this document could be used to improve the efficiency and longevity of current rover systems extending their research lifecycle.

References

[1] E. Kaufmann *et al.*, 'Beauty and the beast: Optimal methods meet learning for drone racing', in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 690–696. doi: 10.1109/ICRA.2019.8793631.

[2] V. Sadhu, S. Zonouz, and D. Pompili, 'On-board Deep-learning-based Unmanned Aerial Vehicle Fault Cause Detection and Identification', Apr. 2020, [Online]. Available: http://arxiv.org/abs/2005.00336

[3] IEEE Neural Networks Council. and Fla.) IEEE World Congress on Computational Intelligence (1994 : Orlando, *Estimating the Mean and Variance of the Target Probability Distribution*. IEEE Neural Networks Council, 1994.

[4] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, 'Comprehensive review of neural network-based prediction intervals and new advances', *IEEE Transactions on Neural Networks*, vol. 22, no. 9. pp. 1341–1356, Sep. 2011. doi: 10.1109/TNN.2011.2162110.