A PROJECT BASED SEMINAR REPORT

ON

"AUTONOMOUS LANDING OF A DRONE USING DEEP NEURAL NETWORKS"

Submitted by

PRACHI MATE : [Exam Seat No: C22017441545]

Under the guidance of

Dr. Dipti Patil

(Associate Professor, Department of Information Technology)

DEPARTMENT OF INFORMATION TECHNOLOGY

MKSSS'S CUMMINS COLLEGE OF ENGINEERING FOR WOMEN, PUNE

2019-20

AFFILIATED TO

SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT BASED SEMINAR REPORT

ON

"AUTONOMOUS LANDING OF A DRONE USING DEEP NEURAL NETWORKS"

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY

in partial fulfillment of TE (Information Technology Engineering)

2015 Course

by

PRACHI MATE : [Exam Seat No: C22017441545]

Under the guidance of

Dr. Dipti Patil

DEPARTMENT OF INFORMATION TECHNOLOGY

MKSSS'S CUMMINS COLLEGE OF ENGINEERING FOR WOMEN, PUNE

AFFILIATED TO

SAVITRIBAI PHULE PUNE UNIVERSITY

MKSSS’s Cummins College of Engineering For Women, Pune

CERTIFICATE

This is to certify that the project based seminar report entitled “**AUTONOMOUS LANDING OF A DRONE USING DEEP NEURAL NETWORKS**” being submitted by

PRACHI MATE : [Exam Seat No: C22017441545]

is a record of bonafide work carried out by her under the supervision and guidance of Dr. Dipti Patil in partial fulfillment of the requirement for TE (Information Technology Engineering) 2015 course of Savitribai Phule Pune University, Pune in the academic year 2019-20.

Dr. Dipti Patil **Guide** **Dept. of IT**	**Dr. Anagha Kulkarni** **HOD** **Dept. of IT**

Dr. Madhuri Khambete
Director

This Project Based Seminar report has been examined by us as per the Savitribai Phule Pune University, requirements, at MKSSS’s Cummins College of Engineering For Women, Pune, on . . . .

Internal Examiner External Examiner

ABSTRACT

When landing a drone, the conventional methods often failed to overcome unknown aerodynamic forces which in turn reduces the accuracy of the controller. In this research, a nominal dynamic model with deep neural networks to improve the control performance of the autonomous drone when landing is presented. This is a two part method. The first part aims at calculating the speed of the quadrotor using background subtraction method. This method is an application of computer vision and image processing. Part two comprises of using ReLU Deep Neural Networks for comparing takeoff and landing performance of autonomous and baseline controller. Performance of the drone with different DNN capabilities is compared and experiments are performed on the drone with significant ground effects and a sharp transition where the nature of trajectory with and without spectral normalization is observed. Finally the trajectory tracking performance of autonomous drone with baseline controller has been compared where it is seen how the DNN also enables the drone to generalize the unseen data outside its training domain. Hence, the conclusion made is that the autonomous drone outperforms the baseline tracking controller in not only landing but also in cross table tracking cases.

ACKNOWLEDGEMENTS

I would like to appreciate all my team members: Deeksha Gupta, Rutuja Kolhe and Srishti Nene for their contribution to this project. Not only have they invested their time and efforts to this project but also their high levels of team spirit have kept me motivates throughout.

A special gratitude I would like to give to our project guide, Dr. Dipti Patil, for stimulating our suggestions and encouraging me, enabled me coordinate my project in the best way possible as well as guided the team in achieving the goal.

Furthermore I would like to thank Dr. Anagha Kulkarni, HOD, IT Department, CCOEW for her inspiring guidance, pertinent criticism and pragmatic suggestions which helped us sail through this project.

I sincerely appreciate the cooperation and valuable help rendered by the teaching and non- teaching staff of Department of Information Technology, CCOEW.

And above all, I am grateful to Dr. Madhuri Khambete, principal, CCOEW to provide me with a platform of opportunities where I can learn and broaden the scope of my knowledge everyday.

(Students Name & Signature)

Table of contents etc.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Drones are a part of technological advancements in the fields of military, logistics, transport, e-commerce and many other domains. It is known that, with manually operated drones it is seldom hard to access remote areas and to overcome this anomaly Unmanned Aerial Vehicles(UAVs) are designed. The autonomous drones provide high precision during landing and takeoff hence guaranteeing stability. Using deep learning technology the drone can be sucessfully soft landed. The feed forward deep neural networks are quite challenging as they are data hungry and can be unstable due to high dimensionality. Also they are often difficult to analyze. Hence, the training data set is divided into two parts. Part 1 (0-250s) contains operations at different heights(0.05-1.50m), part 2 (250-350s) includes random X, Y and Z motions for maximum coverage of state space .The goal is to improve landing precision of the drone by using DNNs with spectrally normalized weight matrices in each layer. The resulting controller is exponentially stable, this is achieved by introducing spectrally normalized DNNs bound by Lipschitz constant. This is why the autonomous quadrotor lands more accurately than a baseline nonlinear controller. It decreases the Z axis error from 0.13 m to 0 m and also mitigates the X-Y drift in landing up to 90%. The goal of this research is to significantly improve the control performance of the autonomous drone when landing.

CHAPTER 2

MOTIVATION, PURPOSE AND SCOPE

2.1 Motivation

Drone technology has applications in a variety of domains such as inspection, surveying & mapping, security & Emergency response, military, e-commerce, aerial photography and much more. The need for a drone operator also raises a problem of controller infrastructure and landing equipment. Hence, an automated drone system increases the efficiency and provide a seamless access to real-time data for navigation.

2.2 Purpose

Currently, drones cannot perform at full automation without supervision or manual intervention. Even the most autonomous drones require pre-programmed flight paths. The purpose of this research is for the drone to find its own speed, determine motion, detect obstacles and find the target area for landing. The ultimate goal to be achieved is for the drone to soft plan on it's own using deep learning technology.

2.3 Scope

Drones are qualified to capture aerial data which can have various applications fueling seamless data collection. Using an autonomous drone will eliminating the need of a manual pilot at the base station. It also has various applications in the field of aero-robotics in domains like mapping, surveying, monitoring and control. Also the self speed monitoring technology of the drone can be used in other non-aerial operations such as - car automation and traffic control.

CHAPTER 3

LITERATURE SURVEY

Unmanned aerial vehicles has been a fervent topic of in several domains such as military, logistics, disaster management and transport. However the lack of infrastructural assistance for landing limits the application of drones. Hence the goal of this research is to enable the functioning of a drone in an autonomous way such that it won't need a manual controller for takeoff and landing.

For landing a drone it is critical to calculate its speed and distance from the ground at every point. Using the background subtraction method for speed detection, the speed of the drone for each incoming frame can be calculated. The accuracy of the detected speed relies on how the targeted landing area is tracked in a given range when there is a presence of other elements around it. Using kalman filter across the frames for object detection can be compared against the current approach to see how it performs. Automated speed detection can be revolutionary and has a huge scope in not just UAVs but also in other transport systems.

Due to complex interactions of rotor and wind air flow with the ground, UAVs require high precision landing control. To capture complex aerodynamic interactions, DNN can be used to build an effective ground effect model. This model improves the precision of the drone landing with stability by using deep learning system. Spectrally normalized weight matrices in each layer are used to train DNNs. The generalisation performance of the DNN and also the overall control performance of the autonomous drone is evaluated. The experimental results show that the autonomous drone outperforms the baseline controller by a wide range and the spectral normalization of ReLU DNN is crucial for the trajectory tracking performance to produce smaller tracking errors. Hence, using a machine learning technology such as deep neural networks can guarantee stability for precise landing of an Unmanned Aerial Vehicle(UAV).

CHAPTER 4

SYSTEM DESIGN

4.1 Block diagram of drone

The block diagram of a drone is given in the figure [7] below.The flight controller has sensors in it which receives sensory data from the surroundings. The flight controller sends signal to the speed controllers directed to the 4 motors of the quadrotor, to control the motor thrust. Cameras are installed on the drone for video streaming to the owner device and to capture images or frames when landing. GPS module and WiFi dongle is installed for location tracking and communication. Power distribution unit supplies power to the flight controller and VGPU unit.

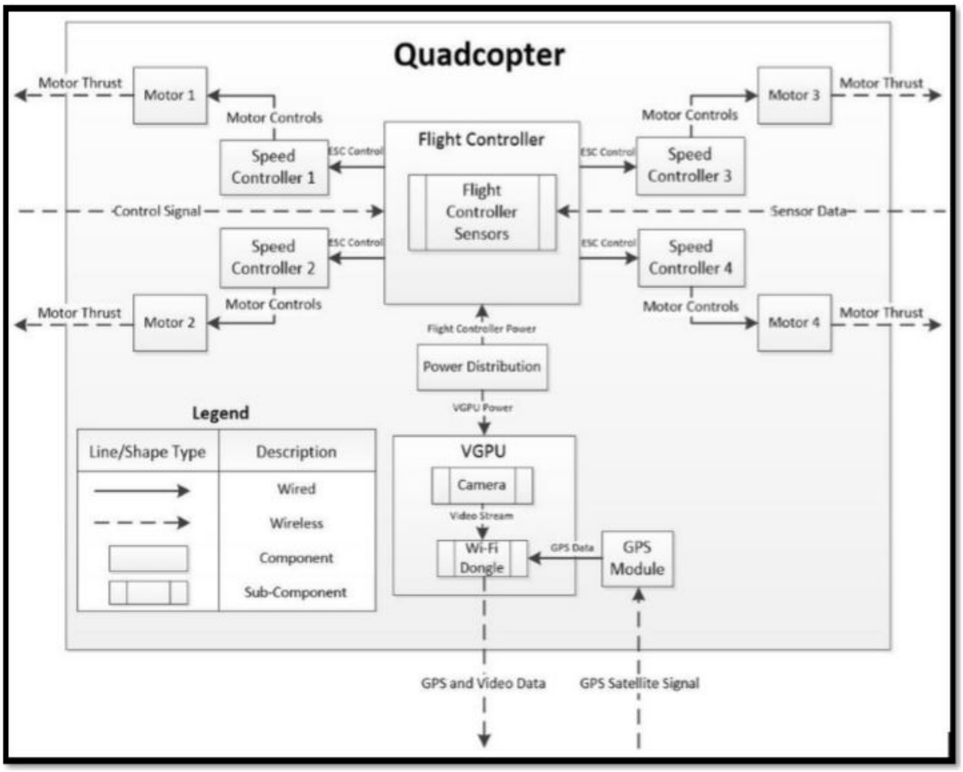


Figure 4.1: Block diagram of a drone [7]

4.2 Hardware Requirements

1. 17 cameras
2. WiFi router
3. Intel Aero drone
4. Onboard Linux computer with 2.56 GHz Intel Atom x7 processor
5. 8 reflective infrared markers
6. ARDUINO microcontroller
7. MPU-6050 (3-axis Accelerometer & Gyroscope)
8. HMC-5883L magnetometer
9. BME-280 (temperature, pressure, humidity)
10. 4 BLDC motors and ESCS
11. GPS module
12. Node MCU ESP8266

4.3 Intel Aero Drone

Figure 4.2: Aero drone [1]

CHAPTER 5

METHODOLOGY

The methodology for soft landing of a drone works in 2 parts: (i) Calculating the speed of the drone (ii) Using DNN to evaluate generalization performance

5.1 Algorithm for calculating speed of the drone using background subtraction method

Step 1: Extract images from the camera fitted on the drone when it starts landing.

Step 2: Find a target spot for landing by using background subtraction method.

Step 3: Detect a target spot in each foreground image and generate a blob object each time.

Step 4: Filter out the area in the background of the blobs or the regions outside the selected area.

Step 5: To calculate the distance of the target spot from the drone, store the frame numbers from the entry point to the current point of the drone.

Step 6: Calculate the number of frames lapse between the entry and current points of the drone.

Step 7: Compute the speed of the drone using the following equation-

$$s = n * d/FPS \tag{5.1}$$

Where, n = number of frames elapsed from the entry frame to the current frame

d = the mapping of the frame coordinates to real world coordinates of the video

FPS = frames per second

5.2 Algorithm to evaluate generalization performance of the drone using deep neural networks

Step one - Bench Test

First measure the mass of the drone(m), diameter of the drone(D), air density at the given point of time(ρ) and the acceleration due to gravity(g) is known. Determine thrust constant (C_T) as well as the non-dimensional thrust coefficient by performing the bench test,

$$C_{\mathrm{T}} = \frac{C_{\mathrm{T}}}{D^4} \tag{5.2}$$

C_T is a function of propeller speed (n) whose nominal value=2000RPM

Step two - Flying and various heights and collecting training data

Fly the drone at various heights to estimate disturbance force (f_a). Collect the sequence of state estimates and controlled inputs as our training dataset: {(p,v,R,u),y} where y is the value of f_a observed.Calculate f_u based on nominal C_T from the bench test. Use the following equation to calculate f_a,

$$f_{\mathrm{a}} = m\dot{v} - mg - Rf_{\mathrm{u}} \tag{5.3}$$

The training set is a continuous single trajectory at various heights. The training set is divided into two parts as follows:
Part one is used to learn ground effect.
Part two is used to learn other aerodynamic forces such as air drag.

Step three- Deep neural network prediction performance

Train a rectified linear units network(ReLU).

$$\hat{f}_{\mathrm{a}}(\zeta, u) = \hat{f}_{\mathrm{a}}(z, v, R, u) \tag{5.4}$$

where z is global height, v is global Velocity, r is altitude, u is the controlling input. Build ReLU using PyTorch. It consists of 4 hidden layers which are completely connected to each other, with input dimension equal to 12 and output dimension equal to three. To constrain Lipschitz constant of the neural network, spectral normalization is used. Compare the near ground estimation accuracy of DNN with one dimensional steady ground effect model:

$$T(n, z) = \frac{n^2}{1 - \mu(\frac{D}{8z})^2} C_{\mathrm{T}}(n) = n^2 C_{\mathrm{T}}(n_0) + \overline{f_{\mathrm{a,z}}} \tag{5.5}$$

where T is the thrust generated and n is the rotation speed.

n_0 is the idle rpm a new depends on the arrangement and number of propellers on the drone. C_T is a function of n and hence, $\overline{f}_{(a,z)}$ from T(n,z) can be derived.

CHAPTER 6

SOFTWARE TOOLS TECHNOLOGIES USED

6.1 Technology stack [8]

1. C++
2. OpenCV for image functionalities
3. PBAS and cvBlob5 open-source packages
4. Video of any format but should have FPS (frames per second) encode

6.2 Softwares:

1. Arduino 1.8.12
   Arduino is an open-source software used to write code and upload it to the AURDUINO microcontroller ATmega2560 . It runs on different operating systems like Windows, Linux and Mac. It is written in Java, based on processing. This software is built to be used with any AURDUINO board.[2]
2. Spyder 3.2.6
   Spyder is a development environment written in Python and for Python which is designed for scientists and data analysts. It offers a novel combination of comprehensive development tools with the data exploration, interactive GUI and deep inspection capabilities of a scientific package. The abilities of spyder can be extended further via its plugin system and API. Spyder is used as an IDE to run OpenCV, Tensorflow, Theano and Keras libraries. [3]

6.3 Python libraries:

1. OpenCV
   OpenCV (Open Source Computer Vision Library) is an open source library for computer vision and machine learning technologies. It was built to produce a common infrastructure for computer vision applications.[4]
2. Keras
   Keras is an open source neural-network library written in Python and is built to enable fast experimentation with deep neural networks. It is user-friendly, extensiible and modular. [5]

3. Imutils
   Imutils comprises of convinience functions to perform image processing functions like translation, skeletonization, resizing, rotation and displaying Matplotlib images.[6]

6.4 Machine Learning Architecture:

Machine Learning is a subset of computing and deep Learning which falls under the umbrella of artificial intelligence. In deep neural networks data flows from input end within the output player and cannot go backwards. The links between the layers are directed forwards and it never touches the same node again. The outputs are obtained by supervised learning through back propagation. Here, a hidden 4-layer rectified linear units with neural network is used. The Rectified Linear Units function is,

$$f(x) = \max(0, x) \tag{6.1}$$

This function is applied element wise to the output of any another function and can replace all activation functions except the readout layer.

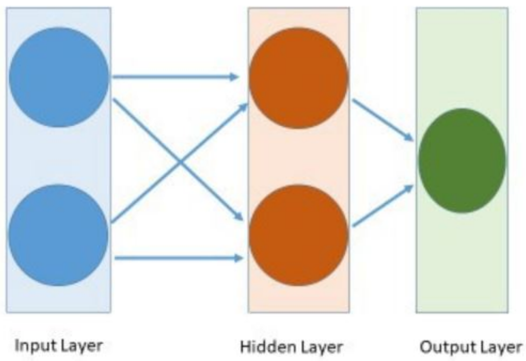


Figure 6.1: Deep Neural Network [10]

CHAPTER 7

TESTING

7.1 Comparing take off and landing performance

The figure 7.1 shows the comparison between the takeoff and landing performance of baseline controller and the controller using deep neural networks. The lines represent the mean error and the shaded areas represent the standard deviation of 10 trajectories.

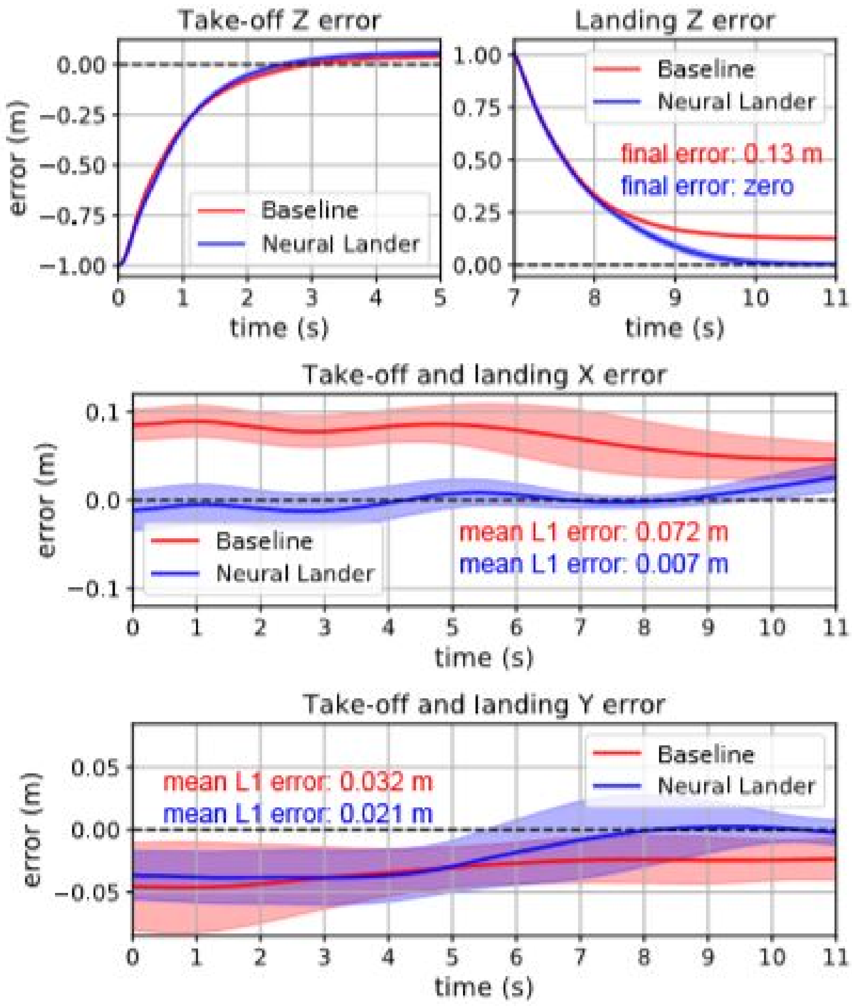


Figure 7.1: Take off and landing comparison [11]

The following conclusions can be made about the benefits of using autonomous drones:

1. It helps the drone land on the ground smoothly and precisely. Where as, due to the ground effect, the baseline controller barely achieves zero terminal height.

2. with DNN, the drone learns about the additional aerodynamic forces such as air drag. Hence, it can alleviate drifts observed in the x-y plane

7.2 Comparing drone performance with different DNN capacities

Figure 7.2 shows the performance of the drone after testing it with different DNN capacities

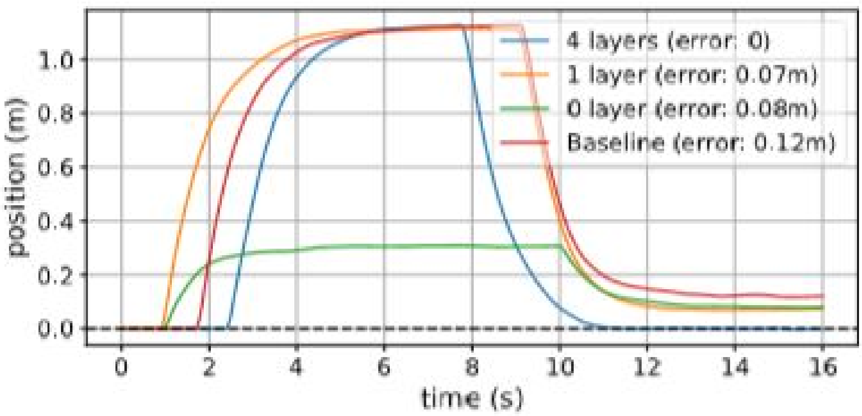


Figure 7.2: Comparison between different DNN capabilities [11]

Value of unknown disturbance force f_a for Baseline model is,

$$\hat{f}_a = 0 \tag{7.1}$$

Value of unknown disturbance force f_a for 0-layer DNN is,

$$\hat{f}_a = b \tag{7.2}$$

Value of unknown disturbance force f_a for 1-layer DNN is,

$$\hat{f}_a = Ax + b \tag{7.3}$$

Hence it can be concluded that, as compared to the baseline model, 1 layer model decreases the z error but it is still not enough to land a drone. Also the 0 layer model generates significant error during takeoff.

7.3 Trajectory Tracking Performance

The DNN algorithm cannot handle complicated situations where physics based aerodynamic modelling is more difficult. To model complex dynamics, experiments over a table with significant ground effects and a sharp transition,were performed to collect data set. The ReLU DNN is trained into a new model with input features: (p,v,R,u). The nature of drone without spectral normalization is observed. The lack of spectral normalization can result in crashing of the drone due to the controller outputs being unexpected. The figure 7.3 shows the heat maps with and without spectral normalization of ReLU network. It is seen that the spectrally normalized DNN shows a clear boundary of the table.

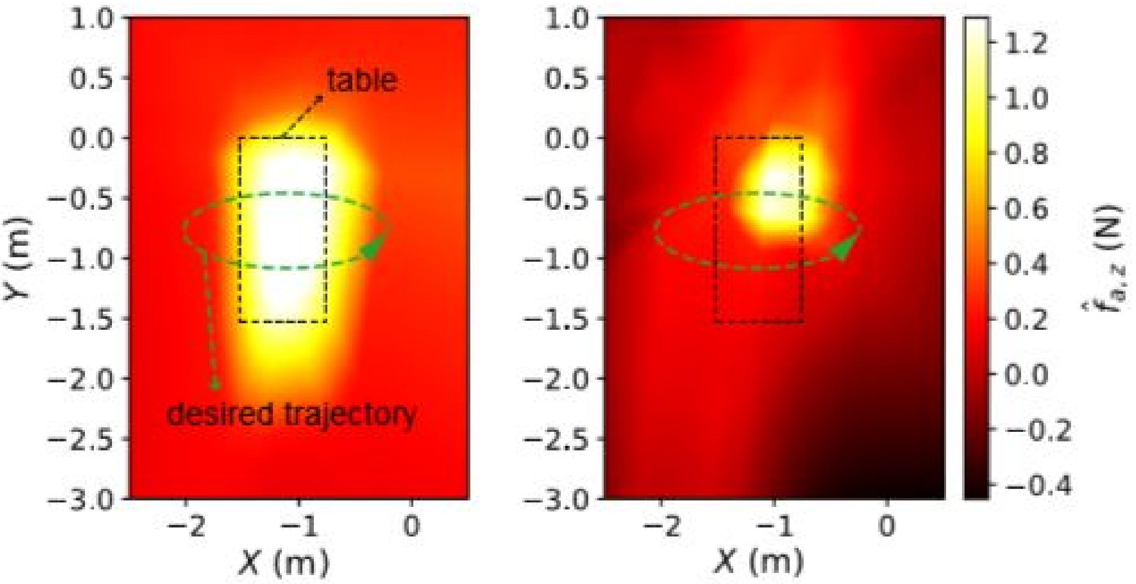
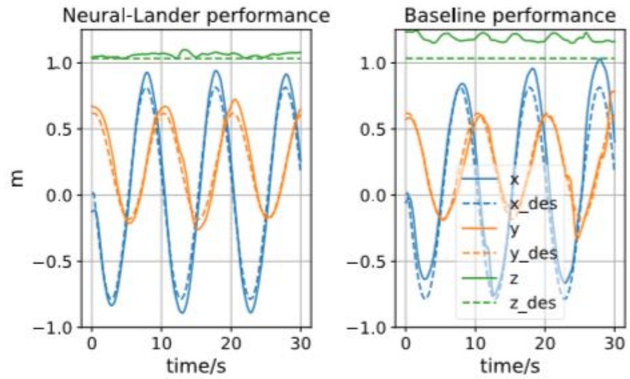


Figure 7.3:
(Left) ReLU DNN network with spectral normalization
(Right) ReLU DNN network without spectral normalization [11]

7.4 Comparing the trajectory tracking performance of autonomous drone with baseline controller

The autonomous drone with DNN overshadows the baseline controller in all the X,Y,Z axes. The figure7.4 shows the tracking performance and statistics of autonomous and baseline controller.



Mean X error	**0.079m**	0.126m
Mean Y error	**0.052m**	0.061m
Mean Z error	**0.027m**	0.153m
Z variance	**0.014m**	0.026m

Figure 7.4: Tracking performance and statistics [11]

The height variance at the edge of the table was seen in autonomous drone as changes were captured in the controller with ground effects as drone flew over the table. Hence, much smaller tracking errors are seen in autonomous drone which uses spectrally normalized DNN.

CHAPTER 8

CONCLUSION

The deep learning based controller significantly improves the controller performance of the drone. From each test case scenario, we can conclude that the autonomous drone using ReLU outperforms the manually controlled drone. Also, using a fully connected 4-layer hidden DNN has almost zero error during take-off and landing of the drone where as 1-layer and 0-layer DNNs show a significant error. It has also been observed that using spectrally normalized ReLU network has better trajectory tracking performance as compared to ReLU network without spectral normalization. Hence, to sum up, the main benefits of an autonomous drone are:

1. The autonomous drone provides more accurate estimates with less error percentage than theoretical ground effect model.
2. It outperforms conventional drones on X, Y and Z axes as it can apprehend the ground effects as well as additional trivial aerodynamics.
3. The stability of the controller is guaranteed and it also employs generalization to unseen domains.
4. Unmanned Aerial Vehicles (UAVs) have a greater span of movement that manned aircrafts. The drone is able to fly at lower and more distinct angles allowing them to access traditionally non-navigable areas .[9]

CHAPTER 9

FUTURE ENHANCEMENT

In the future the capabilities of the drone can be increased by handling unseen disturbances such as wind fan array. The next generation drones can have built-in safeguards, smart accurate sensors, compliance tech and would be available with self regulating technology. Search drones will have applications in domains like transport, military, logistics and e-commerce. Persistently developing technology will ensure mass adoption of drones in the future .

APPENDIX A

RELU DEEP NEURAL NETWORKS

ReLU you stands for rectified linear units and is used because it is simple, fast and empirically works well. Before training a neural network, it's weights must be initialized to random values. Initializing weights to small random values centered on the zero is done using ReLU. Hence, by default half of the units in the network will output is zero value. The parameterized weights when input x is functonally mapped to output f(x,$theta$) are:

$$\theta = W^1, ..., W^{(L+1)} \tag{A.1}$$

Hence,

$$f(x, \theta) = W^{(L+1)}\phi(W^L(\phi(W^{(L\text{-}1)}(...\phi(W^1x)...))) \tag{A.2}$$

where, element wise ReLU function is the activation function.

$$\phi(\cdot) = max(\cdot, 0) \tag{A.3}$$

Due to high sensitivity on the curvature of training objectives, deep neural networks are unstable. To eliminate this issue spectral normalization is used in the network.

APPENDIX B

SPECTRAL NORMALIZATION

To stabilize DNN by constraining Lipschitz constant, spectral normalization is used. This is done in order to generalize the DNN which in turn provides stability to the algorithm. The ReLU network can be bound by spectral normalization of each layer with Lipschitz constant

$$g^l(x) = \phi(W^l x) \tag{B.1}$$

For weight matrices in each layer, spectrum normalization can be applied as:

$$\bar{W} = \frac{W}{\sigma(W)}.\gamma^{\frac{1}{L+1}} \tag{B.2}$$

Where, $gamma$ = Intended Lipschitz constant for DNN. Consider a layered ReLU network $f(x, theta)$ with output dimension without activation function.

$$||f||_{\text{Lip}} \leq \gamma \tag{B.3}$$

is satisfied by the entire layer's Lipschitz constant with spectrally normalize parameters,

$$\bar{\theta} = \bar{W}^1, ..., \bar{W}^{L+1} \tag{B.4}$$

Bibliography

[1] https://www.bhphotovideo.com/c/product/1365179-REG/intel_954400_aero_ready_to_fly.html. Accessed on 2020-03-31.

[2] https://www.arduino.cc/en/main/software. Accessed on 2020-03-31.

[3] https://www.spyder-ide.org/. Accessed on 2020-03-31.

[4] https://opencv.org/about/. Accessed on 2020-03-31.

[5] https://g.co/kgs/RpZm6F. Accessed on 2020-03-31.

[6] https://pypi.org/project/imutils/. Accessed on 2020-03-31.

[7] Suvam Kumar Sar Angshuman Bhattacharjee Arghya Hazra. "QUADCOPTER CONTROL USING ARDUINO MICROCONTROLLER". In: (May 2018).

[8] Suhas Gulur Ramakrishna Bipra De Ghanshyam Malu. "Speed Detection Of Moving Vehicle In A Video". In: (May 2016).

[9] Divya Joshi. *Drone technology uses and applications for commercial, industrial and military drones in 2020 and the future*. https://www.businessinsider.in/tech/news/drone-technology-uses-and-applications-for-commercial-industrial-and-military-drones-in-2020-and-the-future/articleshow/72874958.cms. Accessed on 2020-03-31. Dec. 2019.

[10] SPRH LABS. *Understanding Deep Learning: DNN, RNN, LSTM, CNN and R-CNN*. https://medium.com/@sprhlabs/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff. Accessed on 2020-03-31. Mar. 2019.

[11] Guanya Shi et al. "Neural Lander: Stable Drone Landing Control Using Learned Dynamics". In: May 2019, pp. 9784–9790. DOI: 10.1109/ICRA.2019.8794351.