

LEARNING-BASED CLUSTERING FOR FLIGHT CONDITION RECOGNITION

Murat Şenipek, murat.senipek@tai.com.tr, Design Specialist, Turkish Aerospace Inc. (Turkey)
 Uğur Kalkan, ugur.kalkan@tai.com.tr, Design Engineer, Turkish Aerospace Inc. (Turkey)

Abstract

This paper presents flight condition recognition (FCR) algorithms for rotorcraft health and usage monitoring systems (HUMS), which are developed by using the clustering techniques of machine learning. Training and validation dataset are generated by using a generic nonlinear helicopter simulator and several flight data are obtained to train the algorithm. Gaussian Mixture Model (GMM), Neural Networks (NN) and Logistical Regression (LR) algorithms are implemented to perform FCR analyses. Validation and comparison studies are performed and results are compared in terms of accuracy, execution and training time. Finally, a detailed flight report about the flight is provided with percentages of performed flight conditions, which is used to provide feedback for health and usage monitoring systems to predict the life of the aircraft components.

1. INTRODUCTION

There is an increasing demand for rotorcrafts to perform several roles and missions, which leads them to be operated differently than they designed [1]. They may operate several different flight condition which may result in an increased level of structural fatigue. Therefore, it is crucial to monitor how the rotorcrafts are being flown and to enable efficient usage of sub-components.

Health and Usage Monitoring Systems (HUMS) are developed to determine the usage characteristics of the helicopters to predict the future damage on several components of the rotorcrafts. Furthermore, monitoring the health of a helicopter prevents possible failures of the components. HUMS, which is a widely-used system by the world-wide operators, reduces the maintenance costs and helps to monitor the fatigue critical helicopter components to increase the safe flying hours [2].

One of the fundamental operations of a standard HUMS is the flight condition monitoring. Flight condition monitoring, simply maneuver/regime monitoring, provide usage spectrum for further structural health analyses for a helicopter.

In order to monitor the flight, several types of methodologies are implemented to identify the distinct flight conditions (i.e. maneuvers) which may be related to helicopter's life to provide a usage spectrum. Usage spectrum gives information about how the helicopter is flown according to pilot's flying style, environment, operational requirements and payload. Therefore, every helicopter may undergo different structural failures according to the pilot usage.

In this work, clustering based flight condition recognition algorithms are developed. Determination of flight conditions is a simple numerical classification problem, however; similar maneuvers are not uniquely being performed for each helicopter. Therefore, it is desired to implement machine learning techniques for adaptation, comparison and validation. In this work, Gaussian Mixture Models (GMM), Neural Network (NN) and Logistical Regression (LR) based classification algorithms are employed. Learning algorithms require;

- training data to generate the parametrized models of the clustering problem,
- test data to test the accuracy of the algorithms,
- data to be clustered (i.e. flight data)

Training dataset is generated to cover all gross weight, temperature, altitude, wind, engine, mass and payload configuration by using several flight data belonging to different helicopters and mathematical simulation models [3]. Total number of maneuvers are determined to predict the required set of maneuvers for health and usage management systems. Reduced and extended set flight conditions are utilized. GMM, NN and LR algorithms are implemented and compared in terms of accuracy, training and execution time cost and complexity.

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ERF proceedings or as individual offprints from the proceedings and for inclusion in a freely accessible web-based repository.

2. METHODOLOGY

Flowchart of the applied methodology is provided in Figure 1. It starts with the generation of training dataset by the help of comprehensive mathematical models, flight simulator and real flight experience. Once a proper training set is generated, trained models for each algorithm is generated to perform flight condition recognition. Real flight or full flight simulator data is used to classify the flight conditions. Identified regimes, flight report and usage spectrum is provided to evaluate the usage

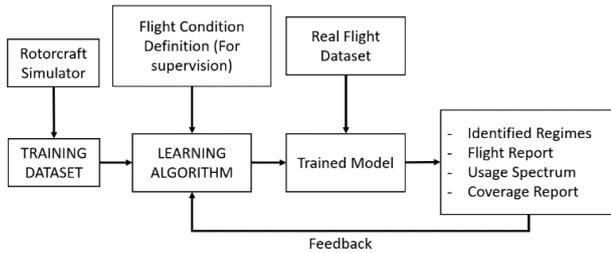


Figure 1. Flow Chart of the FCR

Clustering methodology applied in this work starts with a reduced set of maneuvers that includes 21 different flight regimes. Training, testing and flight condition recognition functions are performed successively. However, there are several other transition and detailed maneuver definitions for HUMS related analyses that ends up having 57 different flight conditions.

2.1. Flight Data and Training Dataset

Training dataset consists of similar flight conditions for the selected set of maneuvers. Dataset is generated by a previously performed flights by flight simulator, validated generic nonlinear rotorcraft simulator and the previous experience [3] [4]. In the training set the input vector \vec{x}_t is provided for the related flight condition for different altitude, air temperature, center of gravity, weight and helicopters to simulate the possible scatter of the data. Input vector \vec{x}_t is the array of numeric flight data which can be measured by a sensing system during the flight of the rotorcraft. The set of monitored flight data for our problem is provided in Table 1. Presented flight data is being monitored and recorded during the flight and filtered to provide the frequency of 20 Hertz; therefore, a total flight data is an array of 19xN size that starts with ground run, take-off, and flight, landing and shut down. Training flight data is divided into segments which belongs to the trained maneuver type for supervised learning. During the compilation of training dataset, only the limited amount of flight condition datasets is used in order to keep the complexity in an acceptable level to employ and compare different methodologies.

Table 1 Names of the General Flight Parameters

ID	Parameter Name	Units
1	Weight on Wheel	(1 or 0)
2	Indicated Airspeed	knots
3	Roll Attitude	deg
4	Pitch Attitude	deg
5	Radio Altitude	Ft
6	Rate of Climb	ft/min
7	Roll Rate	deg/sec
8	Pitch Rate	deg/sec
9	Yaw Rate	deg/sec
10	Longitudinal Acc.	m/s ²
11	Lateral Acc.	m/s ²
12	Normal Acc. (Nz)	g
13	Engine 1 Torque	%
14	Engine 2 Torque	%
15	Angle of attack	deg
16	Heading	deg
17	Ground Speed	knots
18	Latitude	deg
19	Longitude	deg

According to the listed 19 numbers of flight data all flight conditions can be determined. However, reduced dimension of data is used for the determination of the maneuvers as described in the next chapter. Sample training data for twin-engine Hover condition with critical flight parameters is plotted in Figure 2.

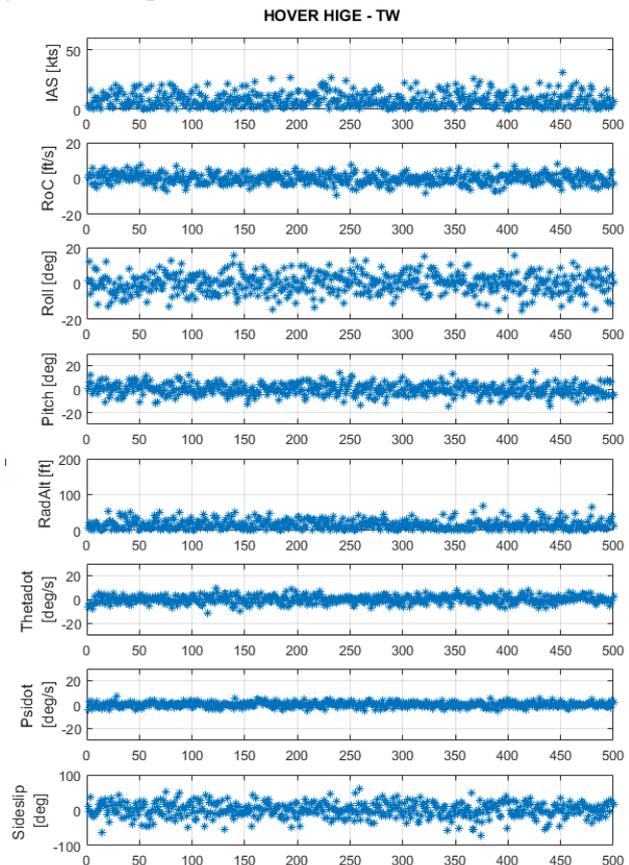


Figure 2 Training dataset for HIGE condition

In this work, flight data set is reduced to a minimum possible set covering the largest set of flight conditions is determined to increase the training speed and keep the complexity in an acceptable level. Therefore, all ground idle and taxi conditions, acceleration and deceleration flight conditions, one engine inoperative conditions and autorotational flight conditions are excluded. Reduced dimension dataset includes the variables;

- $x_t[1]$: Indicated Airspeed
- $x_t[2]$: Rate of Climb
- $x_t[3]$: Roll Attitude
- $x_t[4]$: Pitch Attitude
- $x_t[5]$: Radio Altitude
- $x_t[6]$: Pitch Rate
- $x_t[7]$: Yaw Rate
- $x_t[8]$: Sideslip Angle

Training dataset obtained from the rotorcraft simulation software and simulators are not noisy and can easily be used. However, real flight data is noisy and some filtering pre-process is required to train the model.

2.2. Definition of Flight Conditions

Reduced and extended set of flight conditions to be clustered are labeled and classified. Reduced set of flight conditions are given in Table 2. In the reduced dataset, algorithms are evaluated and compared with a generated test dataset.

Table 2 Reduced set of Flight Conditions

ID	Maneuver Name
1	Hover
2	Forward Flight @ 20 knots
3	Forward Flight @ 40 knots
4	Forward Flight @ 60 knots
5	Forward Flight @ 80 knots
6	Forward Flight @ 100 knots
7	Forward Flight @ 120 knots
8	Forward Flight @ 140 knots
9	Forward Flight @ 160 knots
10	Vertical Climb
11	Vertical Descent
12	Oblique Climb @ 60 knots
13	Oblique Climb @ 120 knots
14	Oblique Descent @ 60 knots
15	Oblique Descent @ 120 knots
16	Spot Turn Right
17	Spot Turn Left
18	CT. Right @ 60 knots
19	CT. Right @ 120 knots
20	CT. Left @ 60 knots
21	CT. Left @ 120 knots

Extended set of flight conditions are given in Table 3 that includes 57 number of different maneuvers including steady flight conditions and maneuvering flight conditions such as pull-up and push-over maneuvers.

Table 3 Extended set of Flight Conditions

ID	Maneuver Name
1	Hover HIGE - TW
2	Hover HIGE - Right Taxi
3	Hover HIGE - Left Taxi
4	Hover HIGE - Back Taxi
5	Hover HOGE -TW
6	Hover HOGE - Right Taxi
7	Hover HOGE - Left Taxi
8	Hover HOGE - Back Taxi
9	Forward Flight @ 20 knots
10	Forward Flight @ 40 knots
11	Forward Flight @ 50 knots
12	Forward Flight @ 70 knots
13	Forward Flight @ 90 knots
14	Forward Flight @ 120 knots
15	Forward Flight @ 130 knots
16	Forward Flight @ 140 knots
17	Climb Vertical
18	Climb Oblique
19	Climb Oblique -Right Turn
20	Climb Oblique -Left Turn
21	Descent Vertical
22	Descent Tween Engine
23	Descent TE- Right Turn
24	Descent TE- Left Turn
25	CT. Right @ <60 kts
26	CT. Right @ 30° >120 kts
27	CT. Right @ 45° >80 kts
28	CT. Right @ 60° >60 kts
29	CT. Left @ <60 kts
30	CT. Left @ 30° >120 kts
31	CT. Left @ 45° >80 kts
32	CT. Left @ 60° >60 kts
33	Bank Turn Right 30°
34	Bank Turn Right 45°
35	Bank Turn Right 60°
36	Bank Turn Left 30°
37	Bank Turn Left 45°
38	Bank Turn Left 60°
39	Spot Turn Right 20°/sec
40	Spot Turn Left 20°/sec
41	Spot Turn Right Max°/sec
42	Spot Turn Left Max°/sec
43	Take-Off TW
44	Take-Off Banked-RIGHT M
45	Take-Off Banked-RIGHT H
46	Take-Off Banked-LEFT M
47	Take-Off Banked-LEFT H
48	Take-Off Pitch UP M
49	Take-Off Pitch UP H
50	Take-Off Pitch DOWN M
51	Take-Off Pitch DOWN H
52	Pull Up Standard
53	Pull Up Right Roll
54	Pull Up Left Roll
55	Push Over Standard
56	Push Over Right Roll
57	Push Over Left Roll

3. METHOD

In this work, one reduced and one extended set of flight conditions are implemented into the clustering algorithms to assess the performance of different techniques. First technique uses Gaussian Mixture Model (GMM) for classification from the generated supervised training flight data set. Second technique utilizes the Regularized Logistic Regression and third method employs the Neural Network approach for the classification of the flight conditions.

For the limited dataset, unsupervised learning approaches are investigated and utilized with the training data in order to observe the applicability. Although training datasets are fully supervised (i.e. every maneuver is known), k-means clustering algorithm is implemented. Parameter k is selected as 21 for the reduced flight condition set to check whether the algorithm will cluster the similar algorithms that are previously identified by supervision. The results are similar about 60% of the maneuvers. The other maneuvers whose variables are close to each other are merged. Therefore, it is decided to perform supervised learning algorithms with the training dataset. Implementation is done to observe the advantages, disadvantages and applicability of the methods. Results are assessed in terms of training time, execution time and accuracy.

Same training datasets are used for all algorithms and assessments are performed by using the same dataset.

3.1. Gaussian Mixture Model

In this methodology, it is assumed that the flight variables x_t are locally Gaussian distributed [5]. This assumption defines the Gaussian mixture over the whole flight, and priors are defined according to the available usage spectrum data for the Gaussian components in order to have it Bayesian estimation. Gaussian Mixture Model (GMM) is estimated by using an Expectation Maximization (EM) algorithm. In the implemented GMM there are M (number of maneuvers) number of different Gaussian distribution with their mean and variance matrices having [1x8] and [8x8] dimensions.

Initial conditions for the EM algorithm of Gaussian Mixture Models are parametrized by a mean vector μ and a covariance matrix Σ to represent the normal probability distribution by using the training dataset.

Training state is performed for all class by using the MLE estimators of the mean vector $\hat{\mu}$ and a covariance matrix $\hat{\Sigma}$ as follows;

$$\mu_i = \frac{\sum_{t=1}^N x_i^t}{N}, i = 1, 2, \dots, n$$

And estimator $\hat{\Sigma}$ is the sample covariance matrix with entries;

$$s_i^2 = \frac{\sum_{t=1}^N (x_i^t - \mu_i)^2}{N}$$

$$s_{ij} = \frac{\sum_{t=1}^N (x_i^t - \mu_i)(x_j^t - \mu_j)}{N}$$

Then, these conditions are provided into the EM algorithm to optimize the GMM model for the provided prior probabilities. After the training is finished there are two matrices for variables and classes as follows;

$$\hat{\mu} = [8 \times M] \text{ and } \hat{\sigma} = [8 \times 8 \times M]$$

Afterwards, the algorithm classifies the flight conditions by taking the flight condition which maximizes the GMM probability density function. Since it is a multivariate case, multivariate probability density function is utilized [5]. Classification algorithm implemented in this method is described as given in Table 4.

Table 4 GMM FCR Algorithm definition

Initialize with the trained parameters of $\hat{\mu}$ and $\hat{\sigma}$ For all $x^t \in X$ For all Maneuvers C_j Select $ID_{ij} = \text{argmax}(p(x_i C_j))$
--

3.2. Regularized Logistic Regression

Logistic Regression is another methodology that is used as a supervised learning method for FCR. Sigmoid function is used for the hypothesis h_θ (estimated probability) as shown in Figure 3.

$$h_\theta(x) = g(z) = \frac{1}{1 + e^{-z}}$$

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

n : # of variables

$$g(z) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

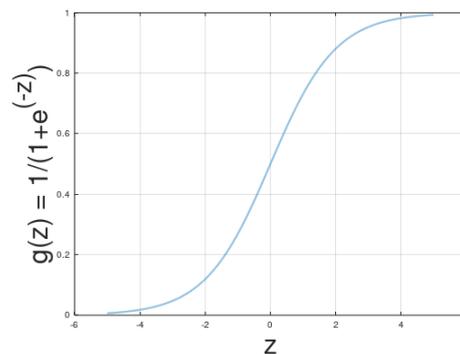


Figure 3 Sigmoid Function Used for the Hypothesis

In the training part, theta values are obtained which minimizes the cost function given below.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m [-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))]$$

i : number of training example
 m : total number of training example
 $y^{(i)}$: class of the i^{th} training example

In order to obtain simpler hypothesis and less prone to overfitting, cost function $J(\theta)$ defined is regularized

as shown below. Note that choosing λ too big results in under fitting (finding θ_1 to θ_n as zero); therefore, $\lambda=0.1$ is used in regime recognition solution.

$$J_{regularized}(\theta) = J(\theta) + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

One-vs-All multiclass classification is used with the regularized logistic regression. During the k th class training, one-vs-all approach sets $y=1$ for class k and sets 0 for the all other classes. After that $J_{regularized}$ is minimized to find the theta coefficients. After one-vs-all classifier trained, it is used to predict the regime recognition. One-vs-All prediction function picks the class for which highest probability occurs.

3.3. Neural Networks

The method is inspired from the principle working logic of the neurons of human beings and is a nonlinear learning algorithm [5]. By using this algorithm, complex relationships between the inputs and outputs can be obtained. Schematic representation of the neural network is shown in Figure 4. Input layer nodes are the flight parameters. Flight conditions to be classified are called classes (output layer nodes) shown in Table 2.

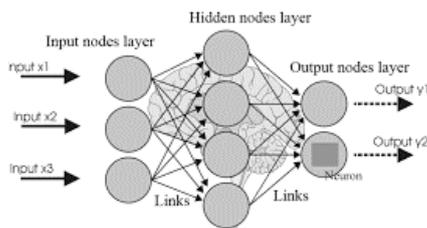


Figure 4 Representation of Neural Networks

The goal is to predict the output layer and problem is to find the θ values. Therefore, function given below is used as objective function and θ 's which minimizes the objective function are found.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[-y_k^{(i)} \log(h_{\theta}(x^{(i)}))_k \right. \\ \left. - (1 - y_k^{(i)}) \log(1 - h_{\theta}(x^{(i)}))_k \right] \\ + \frac{\lambda}{2m} \left[\sum_{j=1}^{\# \text{ hid layer node}} \sum_{k=1}^{\# \text{ inp layer node}} (\theta_{j,k}^{(1)})^2 \right. \\ \left. + \sum_{j=1}^{\# \text{ inp layer node}} \sum_{k=1}^K (\theta_{j,k}^{(2)})^2 \right]$$

K : classes

Implemented algorithm is provided as follows:

Table 5 NN FCR algorithm definition

1. Randomly initialize the θ values
For each training example :
2. Perform feedforward propagation for $a^{(2)}$, $a^{(3)}$

$$a^{(1)} = x : \text{input layer}$$

$$a^{(2)} = g(\theta^1 * a^{(1)}) : \text{hidden layer}$$

- $$a^{(3)} = g(\theta^2 * a^{(2)}) = h_{\theta}(x) : \text{output layer}$$
3. For each output in layer 3 obtain $\delta_k^{(3)} = a_k^{(3)} - y_k$
 4. For layer 2: $\delta_k^{(2)} = (\theta^{(2)})^T * \delta_k^{(3)} * g'(z^{(2)})$
 5. $\Delta^{(1)} = \Delta^{(1)} + \delta^{(1+1)}(a^{(1)})^T$
- Obtain gradient of the cost function as:
6. $D_{ij}^{(1)} = 1/m * \Delta_{ij}^{(1)} + \lambda/m * \theta_{ij}^{(1)}$ for $j > 2$
 - $D_{ij}^{(1)} = 1/m * \Delta_{ij}^{(1)}$ for $j = 1$
 7. Use an optimization method for $\min J(\theta)$, therefore; θ values are found.

Neural Network algorithm is applied for the FCR problem and runs are performed. A single hidden layer is used in the analyses. For 80k iteration of optimization, it is seen that hidden layer node number 50 gives the best result because both cost function is minimum and training accuracy is maximum as can be seen from Figure 5 and Figure 6. Furthermore, training for 80k iterations takes the shortest time for hidden layer node number 50 as can be seen from Figure 7. It was an expected result because learning is deeply modelled and as a result optimization performed more accurately and faster.

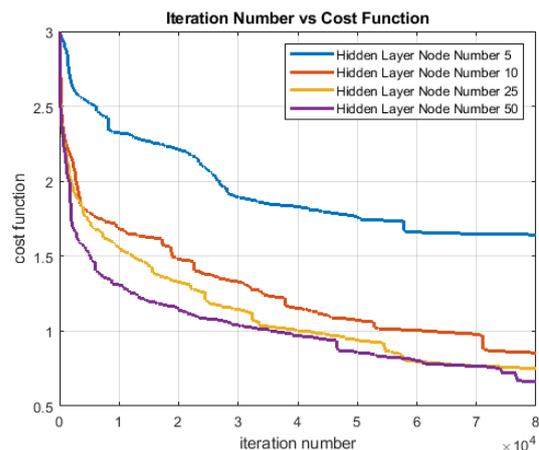


Figure 5 Neural Network Cost Function Minimization for Different Hidden Layer numbers

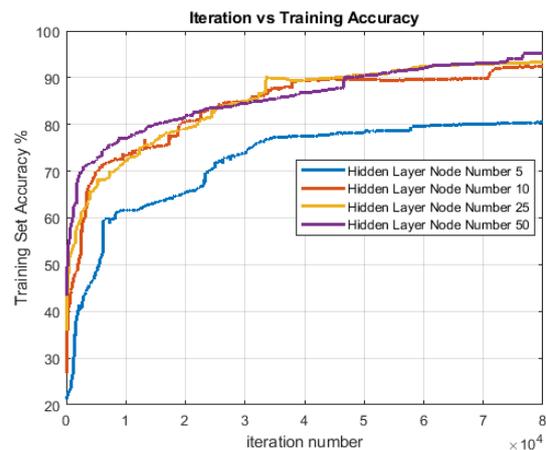


Figure 6 Neural Network Training Accuracy for Different Hidden Layer numbers

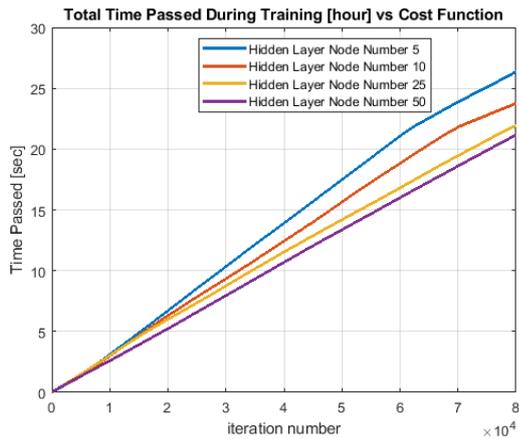


Figure 7 Neural Network Training Time for Different Hidden Layer numbers

After finding the training coefficients, estimation of the flight condition is obtained by using feedforward propagation.

4. RESULTS

In the results part, methodologies are compared in terms of training time, execution time and accuracy. Accuracy evaluation metric is defined as the ratio of successively identified flight condition duration to whole flight duration. Two sets of flight conditions are used as reduced and extended set of maneuver definitions. Results are obtained for both generated test flight data and a piloted flight from a training simulator.

After training and validation part for the reduced maneuver set, sample mission profile is generated. Results of each method is compared and flight summary is reported. In Figure 8, GMM and NN predictions are better than the LR according to regime based comparison. There is a slight difference between GMM and NN. Both of them identifies the maneuvers in an acceptable manner. On the other hand, LR decreases its accuracy %70 for some maneuvers.

According to Table 6, training time for the NN algorithm is much higher than other two. However, the execution time of NN algorithm is much lower than the other methods. This brings the advantage of using NN for real time applications such as “in-flight onboard” FCR. GMM and LR cannot be used for real time problems but can be used for huge training datasets with desktop applications which requires less training time and has high prediction accuracy. The least accurate algorithm in this case is the logical regression having 84% of accuracy.

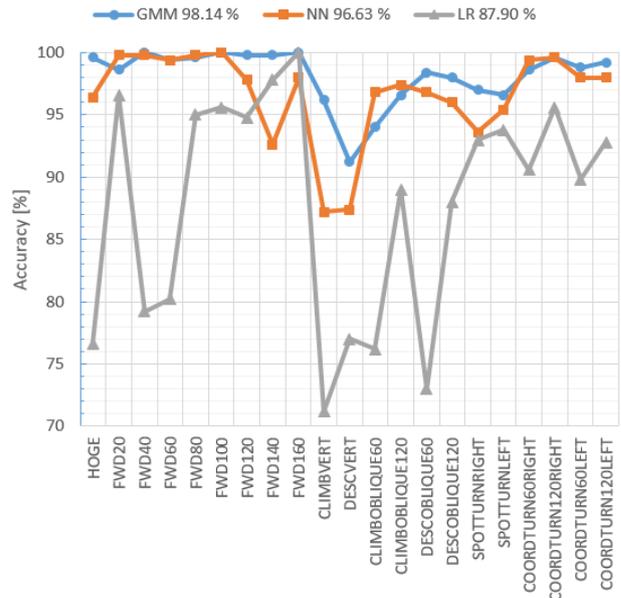


Figure 8 Comparison of all methods by a testing 10500x8 dataset

Table 6 Comparison of FCR algorithms for reduced set

	GMM	NN	LR
Training Time [sec]	2.5	76000	64
Test Time [sec]	150	0.007	1.8
Accuracy [%]	98.15	96.36	84.38

In this part a generated sample helicopter flight is clustered with the available algorithms. Performed flight conditions are summarized in Table 7 and small transition maneuvers exist between two identified flight conditions.

Table 7 Sample 58 minutes of Generated Test Flight

Leg	Duration	Maneuver Name
1	5 min	Hover
2	2 min	Transition
3	10 min	Forward Flight @ 60 knots
4	1 min	Transition
5	3 min	Oblique Climb @ 60 knots
6	2 min	Transition
7	15 min	Forward Flight @ 120 knots
8	1 min	Transition
9	4 min	Oblique Descent @ 120 knots
10	2 min	Transition
11	2 min	Turn Right @ 60 knots
12	2 min	Transition
13	3 min	Forward Flight @ 20 knots
14	1 min	Transition
15	5 min	Vertical Descent
Total	58 min	

Figure 9 illustrates the flight conditions at each flight leg except transitions.

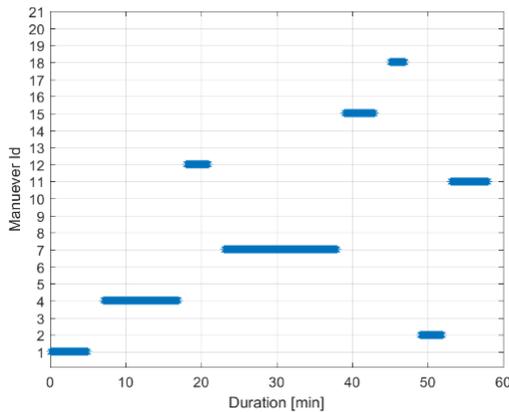


Figure 9 Mission Profile Actual Maneuver Data

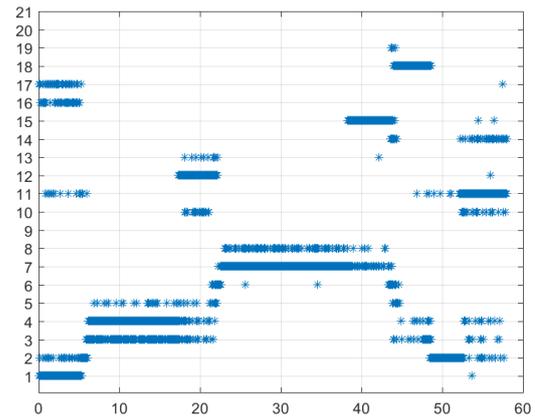


Figure 12 LR Mission Profile FCR

All methods are compared for the flight data and Figure 10, Figure 11 and Figure 12 are plotted to present the results of the algorithms. Each algorithm determines almost all labeled conditions with high accuracy. Scattered data belongs to the instantaneous flight conditions during the transition maneuver from one labeled flight condition to another. Total flight duration is about 58 minutes of time dependent data. Accuracy of the algorithms GMM, NN and LR are **98.15%**, **96.36%** and **84.38%** respectively for 21 flight conditions. As expected from the training results, the most accurate methodology is the GMM and it is followed by NN and LR.

Detailed flight summary is one of the key outputs of these algorithms which will be used to generate usage spectrum of the helicopter when whole flight usage details are accumulated during the life of the helicopter. Flight summary of 21 flight cases is provided in Table 8 for all algorithms.

Table 8 Flight summary and performed flight condition percentages

TEST [%]	GMM [%]	NN [%]	LR [%]	Maneuver Name
8.62	8.78	8.62	6.43	Hover
5.17	6.19	7.17	8.43	FWD @ 20
0.00	1.31	1.72	6.45	FWD @ 40
17.24	17.79	17.93	14.93	FWD @ 60
25.86	25.98	25.67	26.10	FWD @ 120 knots
0.00	0.02	0.28	2.03	FWD @ 140 knots
0.00	3.19	0.91	0.86	Vertical Climb
8.62	9.97	8.62	8.28	Vertical Descent
5.17	7.53	8.17	6.21	Climb @ 60
0.00	1.55	1.67	0.21	Climb @ 120
0.00	0.41	0.52	1.07	Descent @ 60
6.90	9.60	10.03	8.19	Descent @ 120
0.00	0.24	0.50	0.95	Spot Turn Right
0.00	0.21	0.38	0.98	Spot Turn Left
3.45	6.72	6.71	6.36	CT Right @ 60

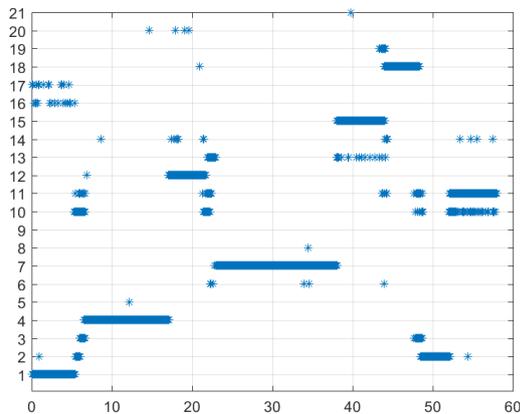


Figure 10 GMM Mission Profile FCR

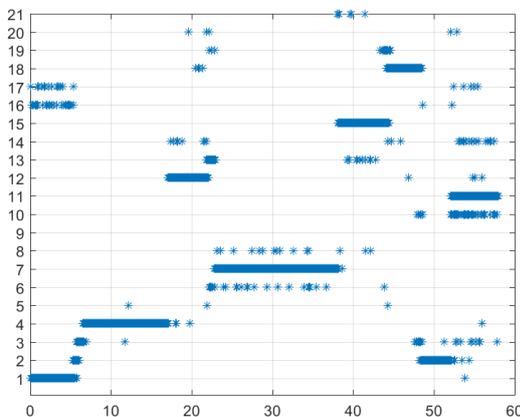


Figure 11 NN Mission Profile FCR

After performing the reduced dataset, the same training procedures are done for the 57 different maneuvers and it is seen that the estimation accuracies are decreased which is an expected situation since the problem domain is extended and there are similar flight conditions which may easily be confused. According to the results, GMM, NN and LR accuracies are 91.47%, 69.2% and 55.2% respectively for 57 flight condition. Detailed accuracy results for each flight condition is plotted in Figure 13. It can be observed that GMM provides an accuracy greater than 80% almost all flight conditions. However, NN requires more time for training or increased number of hidden layers. As the number of maneuvers increased, accuracy of the applied NN model is reduced.

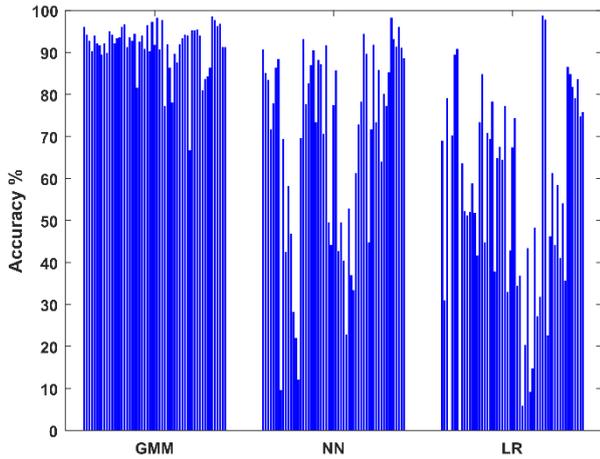


Figure 13 Percentage of accuracy chart for all methodologies for 57 different flight conditions

In the last part, full flight data from take-off is clustered by using the trained models for 57 flight conditions. Flight data belongs to the training flight simulator and flown by the pilots. Flight summary in terms of flight variables vs. time is provided in Figure 14.

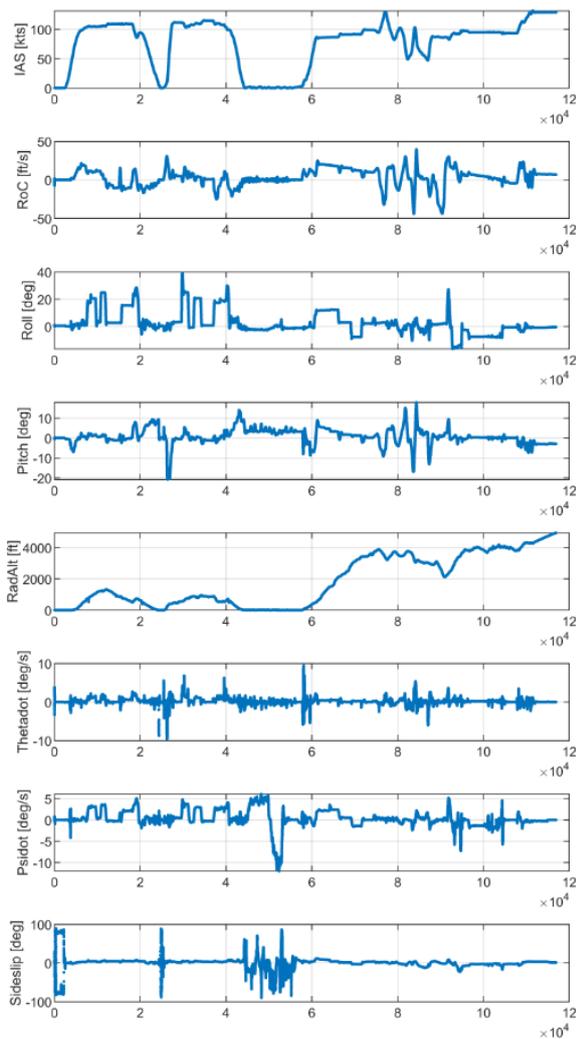


Figure 14 Simulator sample flight data

In Table 9 flight summary and percentages of the performed flight conditions are provided. Most of the flight conditions are clustered consistently by all the algorithms. However, there are some algorithms such as vertical climb and oblique climb that provides different usage percentage. Usage spectrum is obtained by collecting similar usage data for all flights.

Table 9 Flight summary and performed flight condition percentages

GMM [%]	NN [%]	LR [%]	Maneuver Name
13.3	16.2	16.1	Hover HIGE - TW
0.0	0.0	0.0	Hover HIGE - Right Taxi
0.8	0.0	0.0	Hover HIGE - Left Taxi
2.0	0.0	0.7	Hover HIGE - Back Taxi
0.0	0.0	0.3	Forward Flight @ 20 knots
0.0	0.0	0.0	Forward Flight @ 40 knots
0.0	0.0	0.0	Forward Flight @ 50 knots
0.4	0.0	2.2	Forward Flight @ 70 knots
1.1	0.9	0.7	Forward Flight @ 90 knots
0.4	0.7	0.9	Forward Flight @ 120 knots
2.3	1.5	1.2	Forward Flight @ 130 knots
1.6	4.6	1.9	Forward Flight @ 140 knots
20.9	7.9	19.3	Climb Vertical
6.4	1.0	18.1	Climb Oblique
5.4	5.0	6.3	Climb Oblique -Right Turn
0.0	8.7	0.4	Climb Oblique -Left Turn
0.8	0.9	0.4	Descent Vertical
15.5	19.0	17.7	Descent Tween Engine
8.3	8.2	0.3	Descent TE- Right Turn
1.0	1.5	0.0	CT. Turn Right @ 30° >120 kts
0.1	0.0	0.0	CT. Turn Right @ 45° >80 kts
8.8	10.9	10.2	Bank Turn Right 30°
3.0	2.8	0.0	Bank Turn Right 45°
0.0	0.7	0.0	Spot Turn Left 20°/sec
0.0	0.0	0.0	Spot Turn Right Max°/sec
0.0	3.1	0.0	Take-Off Pitch DOWN M
4.0	1.9	0.2	Take-Off Pitch DOWN H
0.0	0.0	0.0	Pull Up Standard
0.0	0.0	0.0	Push Over Standard

Categorized usage percentage is plotted in Figure 15 to identify that whether the differently classified maneuvers are among the similar maneuvers or not.

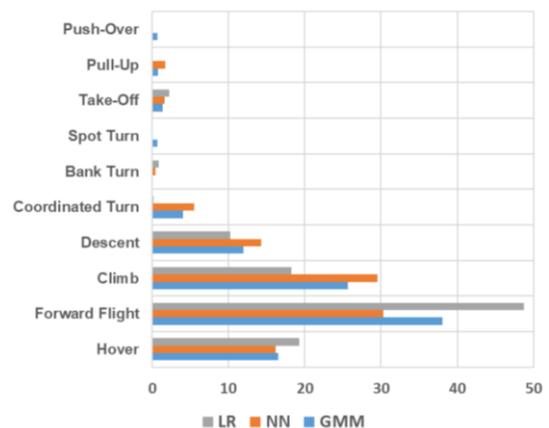


Figure 15 Reduced flight conditions and percentages

It is observed that forward flight and climb cases have the most variation between the clustering algorithms. Finally, 57 flight condition is trained and training and execution times are presented in Table 10.

Table 10 Training and Execution time comparison of 57 FCR algorithms for a simulator flight data

	GMM	NN	LR
Training Time [sec]	104	1 week	600
Test Time [sec]	1621	0.1	10

5. CONCLUSION

In conclusion, three different methodologies which are NN, GMM and LR are implemented to the regime recognition problem. It is seen that for 21 flight maneuvers, GMM and NN has 98.15% and 96.36% prediction accuracy respectively whereas LR has 84.37%. When the number of flight maneuvers are increased to 57, prediction accuracies of all the methods decrease. GMM, NN and LR accuracies become 91.47%, 69.2% and 55.2% respectively.

NN can be accepted as the most suitable algorithm to identify the flight conditions in real time for on-board HUMS systems since it has the fastest execution speed. However, NN and GMM gives almost the same prediction accuracy when trained with proper training dataset. Therefore, if the prediction time is not important GMM can also be used for the ground based applications. Furthermore, training time of the GMM algorithm is relatively low when compared with the other algorithms and can be useful with the massive training data.

As a future work, NN can be trained for multiple number of hidden layers and for different hidden node numbers. Sensitivity analysis should be done for the FCR problem defined and training accuracy can be improved.

6. REFERENCES

- [1] C. Cheung, A. L. Rubio and J. J. Valdes, "Data driven classification of CH-146 manoeuvres using MEMS-IMU sensor system," in *Annual Forum Proceedings*.
- [2] K. F. Fraser, "An Overview of Health and Usage Monitoring Systems for Military Helicopters," Australia, 1994.
- [3] M. Şenipek, "Development of an Object Oriented Design, Analysis and Simulation Software for a Generic Air Vehicle," METU Aerospace Engineering, Ankara, 2017.
- [4] A. Yucekayali, M. Senipek and Y. Ortakaya, "Development of Flight Condition Recognition Algorithms Based on Rotorcraft Simulator," in *AIAC*, Ankara, 2015.
- [5] E. Alpaydın, *Introduction to Machine Learning - Second Edition*, 2010.
- [6] S. Wu, E. Bechhoefer and D. He, "A practical regime recognition approach for HUMS applications," in *63rd Annual Forum of the American Helicopter Society*, 2007.
- [7] H. David, W. Shenliang and E. Bechhoefer, "A Regime Recognition Algorithm for Helicopter Usage Monitoring," in *Aerospace Technologies Advancement*.