Prediction of Surge in Centrifugal Compressors of Automotive Turbochargers Using Convolutional Neural Networks

Research Proposal

November 2020

CDL7416-2021

Carl Wilson [U0370630]

u0370630@unimail.hud.ac.uk

1. INTRODUCTION

Turbocharger compressor surge is a complex challenge in turbomachinery design and can lead to customer dissatisfaction either through noise or, in rare cases, turbocharger failure. Surge starts will a stall within the compressor and escalates to large pressure oscillations throughout the system (Japikse & Baines, 1997). As such the surge line of a compressor map represents a hard boundary where stable and unstable operation occurs. Figure 1 illustrates a typical compressor performance map of temperature and pressure corrected mass flow [reduced mass flow] on the x axis, pressure ratio [PR] on the y axis at lines of constant temperature corrected speed [reduced speed] and contours of efficiency.

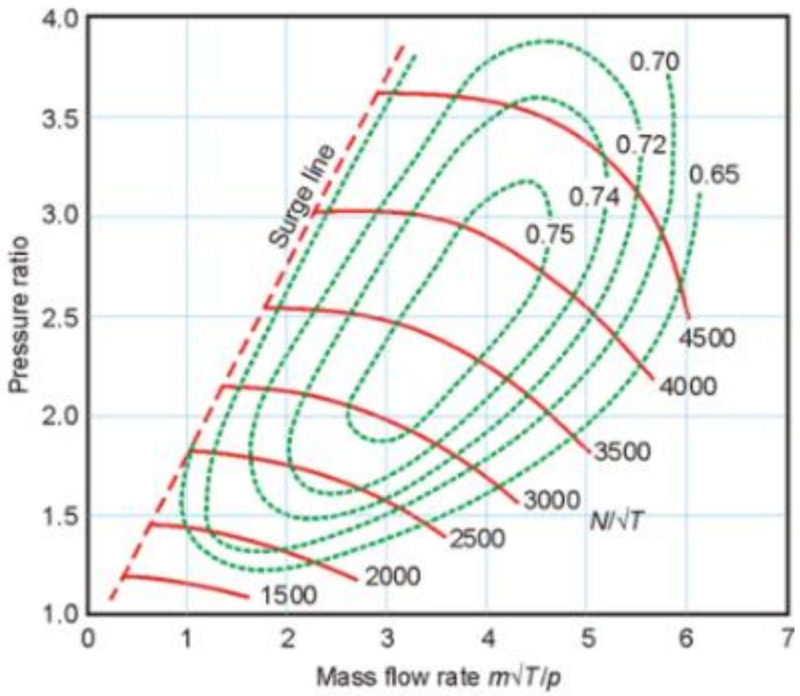

Figure 1: A typical compressor performance map (Anderson, 2019)

It is also a limitation for compressor designers when looking to better

Steady-state Computational Fluid Dynamics [CFD] is heavily relied upon by mechanical design engineers to make crucial design decisions and optimise their compressor designs. Due to the dynamic nature of surge, it can be a significant challenge to predict where surge will occur or requires a significant computational overhead and is often not truly understood until hardware is gas stand tested and a performance map, as per Figure 1, is produced.

Convolutional Neural Networks [CNNs], either used in a Variational Auto Encoder [VAE] configuration (Bhatnagar et al., 2019) and (Guo et al., 2016) or with a Generative Adversarial Network [GAN] architecture (Wu et al., 2020), have been shown to provide good approximations to flow field analysis, at a significantly reduced computational overhead and solve time when used as a Reduced Order Model [ROM] for Reynolds Averaged Naiver-Stokes [RANS] CFD approximations.

Wu et al. (2020) studied supercritical aerofoils at a fixed Mach number and Reynolds Number with low angle of attack using a GAN architecture for the model. Good approximations were produced across the training and test datasets, with some evidence of overfitting.

In the study completed by Bhatnagar et. Al (2019), Signed Distance Function [SDF] was used for geometric representation across several aerofoils and the network was trained multiple times using k-fold cross-validation to tune the hyperparameters, however there was a lack of hard data available on the training scores and the test results also indicated overfitting when comparing training flow fields and test flow fields.

This proposal is to train a CNN using supervised learning to output compressor surge predictions of temperature and pressure corrected mass flow and PR based on a 3D Computer Aided Design [CAD] geometry, in the form of a cavity model, and temperature corrected speed as inputs.

2. PROPOSAL

2.1 Aims

The aim of this proposal is to answer one key question:

Can a Convolutional Neural Network be trained to accurately predict compressor surge based on 3D CAD geometry and a given speed?

2.2 Objectives & Measures

In order to achieve this there will be three key objectives within the project:

1. The first will be to develop the actual training data from historical compressor performance maps of actual hardware. This will involve taking the map, determining the hardware used and assembly clearances as well as the speed lines that have been run for the said geometry. Secondly, it will also involve taking the mass flow and PR data from the maps as target training and test data. Thirdly, it will also involve transforming the 3D CAD geometry into SDF format for the network to be able to interpret. They key measure for this item will be the number of training and test samples that will be produced.
2. The second key objective will be to develop the neural network for making predictions and understand the entitlement that a CNN potentially has. This will include looking at the individual layers within the neural network, the convolution patch sizes [kernel], the stride size [Guo et al. (2016) typically used a stride of 2 for their MaxPooling layers] and how aggressively the MaxPooling layers reduced the feature map to drive out the key representations. For this a coarse grid-search followed by a finer one will be completed. The key measure for this objective will be the cross-validation scores, ideally the training and validation as well as the cross-validation scores want to be in the 0.95 – 0.99 range for Mean Absolute Error [MAE]. The challenge will be balancing the right amount of reduction and number of layers to create meaningful representations but also minimise the overfitting of the training data while managing the computational overhead. The required accuracy will be agreed at the start of the project based on existing prediction methods and their accuracy.
3. The third key objective will be testing of the network for accuracy of predicting surge on unseen “clean” data and completing a statistical analysis of training and test data to understand how well the model has generalised. The key measure will be the MAE distributions for the training data when compared to the test data; the ideal would be no statistical difference.

2.3 *Key Milestones*

1. Training Data Development
    a. Selection of compressor maps for training.
    b. Creation of training and test input speed data.
    c. Creation of training and test target data.
    d. Creation of 3D cavity models and SDF input data.
2. Network Development
    a. Initial creation of CNN and test with GPU solving.
    b. Coarse grid-search and cross-validation.
    c. Fine-tuned grid-search and cross-validation.
3. Network Testing & Analysis
4. Final Report

2.4 Deliverables

The key deliverable will be a research report capturing the findings and of the application of a CNN to predict compressor surge, the training and test results as well as conclusions and recommendations for further research or applications.

Full list of deliverables:

1. Report of findings including final CNN architecture of layer numbers, shapes and activations.
2. The final model saved as a tf format file for reuse with *TensorFlow*.
3. The training and test data including .stp files of geometry and SDF data.
4. Python script for processing compressor maps into input and target data sets.

2.5 Methodology

The initial phase of the project will largely focus on additional literature reviews, ensuring the most up to date developments have been captured as well as anything that might be applicable to the use of Deep Neural Networks [DNNs] for turbocharger performance predictions. There will also be a determination and collection of compressor maps to be used for training and testing the network. The project will be limited in scope to 98mm compressors. This is a popular size and a large catalogue of product and thus training data can be made available. The data will be provided by Cummins Turbo Technologies [CTT].

The collection and development of training data will require the creation of a short script to post-process compressor map native formats into target data sets of mass flow and pressure ratio, but also for the additional input of reduced speed. This script will be developed in Python, as this language is familiar to me, is widely used and offers a number of open source libraries that can be used in this instance, particularly when it comes to developing the network. It is also quick and easy to use for developing high level scripts for laborious tasks such as this. Both *NumPy* and *Pandas* packages will be utilised for managing and shaping the data into input and target vectors [1D tensor] for the input speed and target mass flow and pressure ratio. Because each compressor map will typically have six different speeds, we can easily create six different cases for the network to train on from one compressor map. The test hardware part numbers as well as assembly configuration will also need to be output as a starting point for the geometry collection. This can be taken from the compressor map test data. The key assembly parameters of interest will be front wheel clearance, back wheel clearance and diffuser gap.

In order to create the 3D aero cavities as inputs, the test hardware will need to be replicated in CAD. The individual components will be available, however the assembly for each stage [comprising compressor wheel, housing and diffuser] will need to be created manually according the hardware list provided by the previous script. The cavity mode can then be derived from the assembly to be output as a step or .stp file. The step file format will be used as it is an open and transferable format for communicating 3D CAD geometry data between different CAD software packages, so is ideal for the data input to the geometry API.

The geometry API will likely be the most technically challenging aspect of the project and will also be developed in Python for the same reasons outlined previously. This aspect of the work will lean heavily on open-source modules. *Trimesh* will be used to import the .stp file and create the SDF 3D tensor as an input to the network.

For the network development and training the *Keras API* and *TensorFlow* back-end will be utilised, again using Python. *TensorFlow* offers the capacity to train neural networks a Graphics Processing Unit [GPU], which will be essential when working with 3D data, which can provide up to 10x faster training times (Chollet, 2017). Keras provides a highly configurable API which can be used to quickly construct as well and modify and network architectures, including CNNs.

Once the network is built and the data is prepared, a broad sweep of neural network layer sizes will be trained across using a grid-search of sizes and hyper-parameters at a relatively coarse setting, along with k-fold cross-validation to understand which network provides high accuracy with the least amount of overfitting. This is the phase of the project where the bulk of the analysis will be completed, understanding which architecture and settings work best. One the coarse settings have been determined another grid-search and k-fold validation will be run with finer setting to finalise the network.

Provided the network has achieved a high level of accuracy in predicting surge for the training data and generalisation appears good through the cross-validation, the test data will be introduced in order to assess the model and conclude the research.

3. PROJECT PLAN

3.1 Plan Summary

The project is highly sequential, initially starting with the literature review as well as selecting the input compressor maps, followed by development of the training data set including the manual work to develop the aero cavities. Figure 3 specifically shows the key, high level, phases of the project including how much time it has been provided to create the 3D CAD assemblies. Once the training data is prepared and ready, the actual network development and optimisation through grid searching across key hyperparameters and controls of the network as well as cross-validation. The purpose being to develop a network that generalises the best. Finally the testing, analysis and reporting will conclude the project.

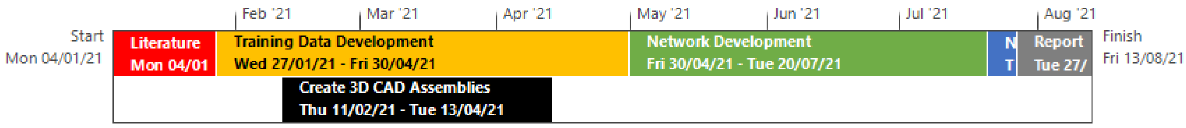

Figure 3: Project timeline.

3.2 Gantt Chart

Figure 4 provides a Gantt chart for the project steps, timing and resource requirements.

3.3 Critical Path

Due to the nature and interconnectivity of tasks within the project, there is very little room for slack. One opportunity to provide some breathing room would be to outsource the 3D CAD geometry assemblies to a CAD engineer, they would likely be quicker and the project timing could be reshuffled so that the 3D CAD task kicked off earlier while the literature review and script development and network design could happen in parallel.

4. RISK ANALYSIS

4.1 Risk Register

The project has been designed from the outset to minimise risk, ensuring the majority of the tasks are within the capability of the researcher and that the tools and resources required are readily available. There are however several risks that still require active mitigation and management in order to ensure the success of the project.

ID	Risk	Category	Probability	Impact	Risk	Mitigation	Probability	Impact	Risk
1	Time required to create 3D CAD assemblies too long / don't have experience to turn models around according to project schedule.	Project	9	9	81	A	3	3	9
2	Model training time too high / GPU hardware not powerful enough.	Project	3	9	27	B	3	3	9
3	Noise within training data due to geometry or test variation.	Project	9	3	27	C	3	3	9
4	Sharing confidential geometry and test data from CTT.	Legal	9	9	81	D	1	1	1
5	Managing storage of confidential geometry and test data from CTT.	Professional	9	9	81	E	1	9	9

4.2 Mitigation Plan

A. The time required to produce the 3D CAD assemblies will be a challenge. While relatively straight forward, the task is manual and laborious. It is also not the researcher's area of expertise or strength. Initially 10 weeks has been provided in the plan to complete this task. If after two weeks it is clear the only a limited amount of test data will become available, then support would be requested from a dedicated CAD engineer to ensure all assemblies are completed in time.
B. The initial plan is to use an nVidia GTX1060 with 1280 CUDA cores on a dedicated machine. This has historically been powerful enough to cope with training models in an acceptable time. Additionally, time has been allowed within the plan for some overrun. However, should this not be powerful enough a state of the art GTX3080 with 8704 CUDA cores can be made available at short notice. Should this also not be powerful enough then AWS instances could be considered as a back-up. However, this will have its own challenges with data

security and so if it is to be considered a real contingency plan – would need to effectively be on a parallel path.

C. There will naturally be some noise in the compressor map data due to geometric variation [within design tolerances] as well as gas stand repeatability. To minimise this historical gauge repeatability data will be assessed during compressor map selection, specifically considering the gas stands capability to reproduce the surge line over time. Compressor maps from before historical data collection will not be considered for the study.
D. The geometry and compressor map data from CTT will be proprietary. As such a Non-Disclosure Agreement [NDA] will need to be put in place prior to sharing data. This can be a lengthy process, so it is recommended this task start ahead of literature review with a target to be completed and in place for map selection.
E. Managing the proprietary data will be a challenge. In order to mitigate this risk all data will be kept on the CTT network – which the researcher will have access to. For the model training the dedicated machine mentioned in mitigation item B can be brought inside the CTT network to ensure data security. The researcher will also need to remain vigilant in ensuring not proprietary data is shared through Box, OneDrive or e-mail accidentally.

5. CONCLUSION

This proposal has outlined how a CNN could be trained to predict compressor surge in terms of pressure ratio and reduced mass flow based when provided 3D CAD geometry and reduced speed. This is based on making use of an existing database of compressor data and 3D CAD geometry and details the method by which this existing dataset can be wrangled into a format suitable for a CNN.

A detailed schedule of activities with clear aims, objectives, deliverables and measurables have been provided along with a risk register and mitigation plans.

It is expected that the introduction of such a model would allow compressor designers to better optimise compressor geometries to suit individual application needs but can also act as a tool to determine the key parameters and their effect on compressor surge through the use of a Design of Experiments.

	Task Name	Duration	Start	Finish	Predecessors	Resource Names	Resource
1	**Literature Review**	**20 days**	**Mon 04/01/21**	**Wed 27/01/21**			
2	Refresh Existing Literature on CNNs	4 wks	Mon 04/01/21	Wed 27/01/21		Carl Wilson[50%]	Library & Summon
3	Review Literature on Surge Prediction	4 wks	Mon 04/01/21	Wed 27/01/21		Carl Wilson[50%]	Library & Summon
4	**Training Data Development**	**77 days**	**Wed 27/01/21**	**Fri 30/04/21**			
5	Select Compressor Maps	5 days	Wed 27/01/21	Tue 02/02/21	2,3	Carl Wilson	Compressor Map Database
6	Create and Test Map Processing Script	5 days	Tue 02/02/21	Tue 09/02/21	5	Carl Wilson	Python & Pandas
7	Run Script	2 days	Tue 09/02/21	Thu 11/02/21	6	Carl Wilson	Python & Pandas
8	Input Speed Data Ready	0 days	Thu 11/02/21	Thu 11/02/21	7		
9	Input Hardware Details Ready	0 days	Thu 11/02/21	Thu 11/02/21	8		
10	Target Surge Data Ready	0 days	Thu 11/02/21	Thu 11/02/21	9		
11	Create 3D CAD Assemblies	10 wks	Thu 11/02/21	Tue 13/04/21	9	Carl Wilson	PTC Creo
12	Create and Test SDF Script	10 days	Tue 13/04/21	Mon 26/04/21	11	Carl Wilson	Python, Numpy, Trimesh
13	Run Script	5 days	Mon 26/04/21	Fri 30/04/21	12	Carl Wilson	Python, Numpy, Trimesh
14	Input Geometry Data Ready	0 days	Fri 30/04/21	Fri 30/04/21	13		
15	**Network Development**	**65 days**	**Fri 30/04/21**	**Tue 20/07/21**			
16	Build & Test Initial Network with GPU	3 days	Fri 30/04/21	Wed 05/05/21	8,10,14	Carl Wilson	Python, TensorFlow, GPU [GTX1060]
17	Configure for Grid-search	7 days	Wed 05/05/21	Thu 13/05/21	16	Carl Wilson	Python, TensorFlow
18	Complete Coarse Grid-search with CV	25 days	Thu 13/05/21	Mon 14/06/21	17	Carl Wilson	Python, TensorFlow, GPU [GTX1060]
19	Analyse Results	5 days	Mon 14/06/21	Fri 18/06/21	18	Carl Wilson	Python, Matplotlib, Pandas
20	Coarse Grid-search Complete	0 days	Fri 18/06/21	Fri 18/06/21	19		
21	Complete Refined Grid-search	15 days	Fri 18/06/21	Thu 08/07/21	20	Carl Wilson	Python, TensorFlow, GPU [GTX1060]
22	Analyse Results	5 days	Thu 08/07/21	Wed 14/07/21	21	Carl Wilson	Python, Matplotlib, Pandas
23	Refined Grid-search Complete	0 days	Wed 14/07/21	Wed 14/07/21	22		
24	Train Final Network	5 days	Wed 14/07/21	Tue 20/07/21	23	Carl Wilson	Python, TensorFlow, GPU [GTX1060]
25	Network Training Complete	0 days	Tue 20/07/21	Tue 20/07/21	24		
26	**Network Testing**	**5 days**	**Tue 20/07/21**	**Tue 27/07/21**			
27	Run Predictions on Test Data	2 days	Tue 20/07/21	Thu 22/07/21	25	Carl Wilson	Python, TensorFlow, GPU [GTX1060]
28	Analyse Results	3 days	Thu 22/07/21	Tue 27/07/21	27	Carl Wilson	
29	Report	15 days	Tue 27/07/21	Fri 13/08/21	28	Carl Wilson	MS Word & PowerPoint
30	Project Complete	0 days	Fri 13/08/21	Fri 13/08/21	29		

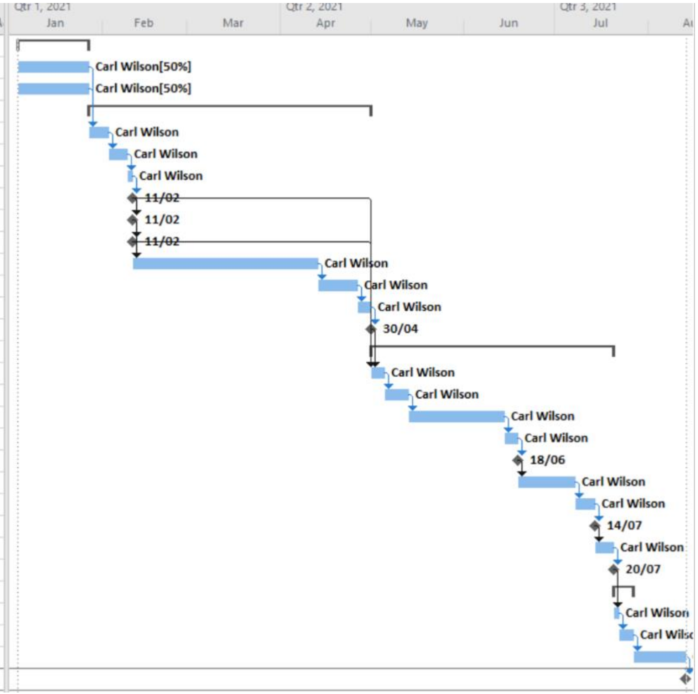

Figure 4: Gannt Chart

6. REFERENCES

Anderson, M. R. (2019). *Performance Corrections for Compressor Maps*. Retrieved 12/11/2020 from https://www.conceptsnrec.com/blog/performance-corrections-for-compressor-maps

Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., & Kaushik, S. (2019). Prediction of aerodynamic flow fields using convolutional neural networks. *Computational mechanics, 64*(2), 525-545. https://doi.org/10.1007/s00466-019-01740-0

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Co.

Guo, X., Li, W., & Iorio, F. (2016). *Convolutional Neural Networks for Steady Flow Approximation* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA. https://doi.org/10.1145/2939672.2939738

Japikse, D., & Baines, N. C. (1997). *Introduction to turbomachinery*. Concepts ETI. http://hud.summon.serialssolutions.com/2.0.0/link/0/eLvHCXMwjV1La8MwDBZ9XFYY7MmepYeRy9bS2k5cD8oofdDB1l0ydgxx7GyFNYG-YD92_2Wy26ylbNBLIEGyQI4sS9YnA1BSq1e31gRNFBdxJCLuSs2bVEuPE6IaodeQceQZKHFXDJ_FU5v5bzn4zqAxH3NVCz_H4R9YB4NaNVWOX9k9Uw-j1fG6yTIxNnjtBhgA-__MsJ1MDKpbI2O1TKBbY7RuW2I17n5zhb2hM11Elq7jvzjz2ug9SSfafNCBFW-oZpO5dhYoYy31PjN1p59OImSwC1grxn9K56FIcJeF9ri_Zgg6bd9EjYJSYi7_clfdgbJ3dJ-laRqiQhjDx4YT7B9AwQAjDiGnkyMobTQyPIbyoyl4V8tOtJVZWkFHJtOxLdRE3Z3Abb_ndwbVjaGDVeIo2FYLOYVCkib6DCp1SWIhlKIct4_abUpznhfRmEvPVVzpc7jZZcSL3cguYW_ZNdZkPq6gaLV4DXnkKW_r8Acfw7w-

Wu, H., Liu, X., An, W., Chen, S., & Lyu, H. (2020). A deep learning approach for efficiently and accurately evaluating the flow field of supercritical airfoils. *Computers & fluids, 198*, 104393. https://doi.org/10.1016/j.compfluid.2019.104393