Remaining Useful Life (RUL) Prediction with oneAPI

Using NASA's Turbofan Engine Degradation Simulation for Aircraft engines

Deepthi A J

Intel oneAPI Innovator | Intel Certified Instructor in Machine Learning using oneAPI
Ph.D. Scholar, Department of Computer Science, IICSE University

deepthiaj@live.in

Table of Contents

oneAPI

Damage Propagation Modeling for Aircraft Engine Prognostics[1]

- Prognostics is the estimation of remaining useful component life.
- Prognostics is currently at the core of systems health management.
- Reliably estimating remaining life promises considerable cost savings (for example, by avoiding unscheduled maintenance and increasing equipment usage) and operational safety improvements.
- Aircraft engines (both military and commercial), medical equipment, power plants, etc. are some of the common examples of these types of equipment.

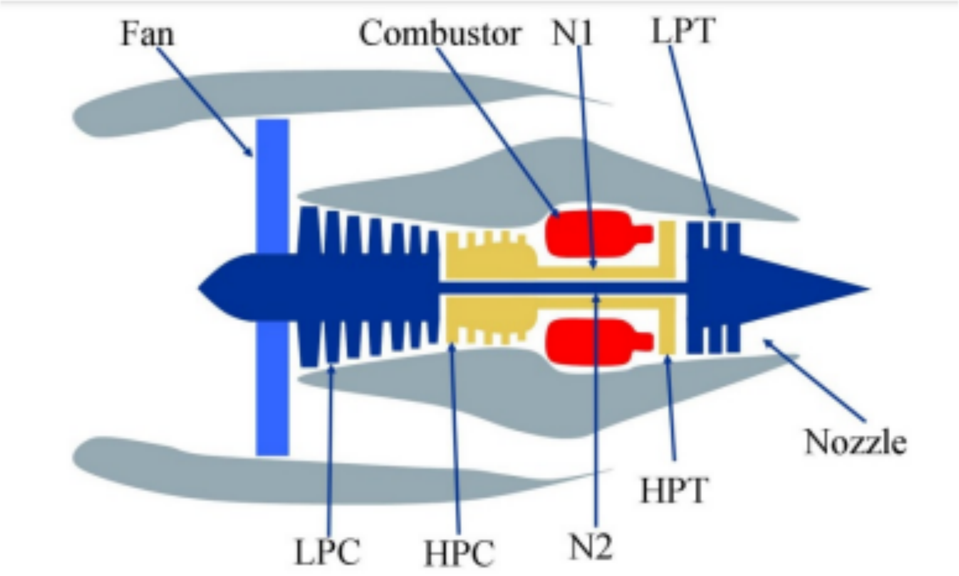
[1] **Saxena**; **Abhinav and Goebel**; **Kai and Simon**; **Don and Eklund**; **Neil (2008).**

C-MAPSS (Commercial Modular AeroPropulsion System Simulation)

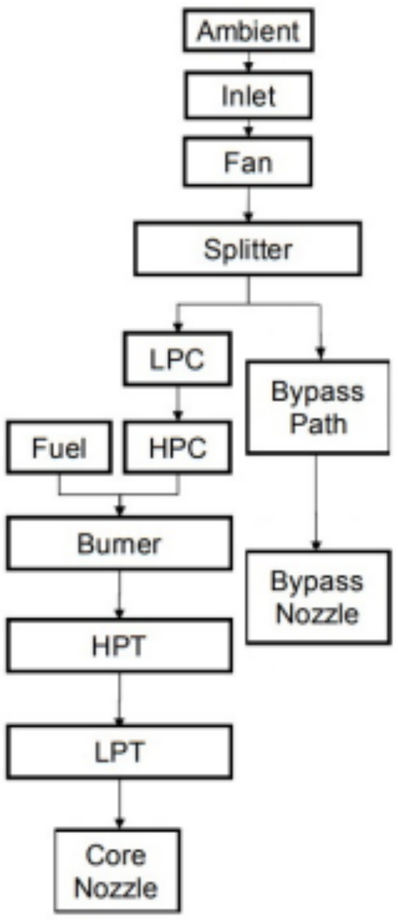
- C-MAPSS is a tool for simulating a realistic large commercial turbofan engine.
- The software is coded in the MATLAB® and Simulink® environment, and includes a number of editable input parameters that allow the user to enter specific values of his/her own choice regarding operational profile, closed-loop controllers, environmental conditions, etc.
- C-MAPSS simulates an engine model of the 90,000 lb thrust class and the package includes an atmospheric model capable of simulating operations at
  (i) altitudes ranging from sea level to 40,000 ft,
  (ii) Mach numbers from 0 to 0.90, and
  (iii) sea-level temperatures from –60 to 103 °F.

C-MAPSS contd..

- The package also includes a power management system that allows the engine to be operated over a wide range of thrust levels throughout the full range of flight conditions.
- In addition, the built-in control system consists of a fanspeed controller and a set of regulators and limiters.



C-MAPSS contd..



C-MAPSS contd..

Symbol	*Description*	*Units*
Parameters available to participants as sensor data		
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s
Parameters for calculating the Health Index		
T48 (EGT)	Total temperature at HPT outlet	°R
SmFan	Fan stall margin	--
SmLPC	LPC stall margin	--
SmHPC	HPC stall margin	--

oneAPI

RUL Prediction from NASA's Turbofan Engine Degradation Simulation for Aircraft engines using oneAPI

- GitHub Repository for RUL Prediction with oneAPI
  https://github.com/Deepthi-AJ/RUL-Prediction-from-NASA-s-Dataset-for-Aircraft-engines-using-Intel-oneAPI.git
- Streamlit Deployment with oneAPI
  https://github.com/Deepthi-AJ/RUL-Prediction-from-NASA-s-Dataset-for-Aircraft-engines-using-Intel-oneAPI.git
- oneAPI can maximize computation performance with power optimization.

Results & Conclusion

Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) Difference `-20.178567848752888`

Prediction of RUL using XGBoost model

Training time: 40.80 seconds

RMSE: `30.786741709841618`

Prediction of RUL using daal4py model

Training time: 0.01 seconds

RMSE: `50.965309558594505`

oneAPI

Results & Conclusion

Prediction of RUL using XGBoost model

Training time: 40.80 seconds

RMSE: `30.786741709841618`

Prediction of RUL using daal4py model

Training time: 0.01 seconds

RMSE: `50.965309558594505`

Visualizations

Performance: XGBoost Prediction vs. daal4py Prediction

oneAPI

Why oneAPI ?

- oneAPI is an open, cross-industry, standards-based, unified, multiarchitecture, multi-vendor programming model that delivers a common developer experience across accelerator architectures – for faster application performance, more productivity, and greater innovation.
- The oneAPI initiative encourages collaboration on the oneAPI specification and compatible oneAPI implementations across the ecosystem.
- **Reference:** https://www.oneapi.io/

Call to Action

- Implement this for accurately predicting the remaining useful life (RUL) of aero-engines is of great significance for improving the reliability and safety of aero-engine systems.
- This study with oneAPI can address higher performance utilizing lower power consumption that can contribute towards longer lifespan of the space vehicles due to the similarities between the space shuttle and standard commercial aircraft, especially modern commercial aircraft.

Key takeaway

- The empirical knowledge-based methods for RUL estimation that requires industry experts to use extensive prior knowledge to establish the corresponding knowledge base can be benefitted from using oneAPI.

Reference

Liu, L., Wang, L. Yu, Z. Remaining Useful Life Estimation of Aircraft Engines Based on Deep Convolution Neural Network and LightGBM Combination Model. Int J Comput Intell Syst 14, 165 (2021). https://doi.org/10.1007/s44196-021-00020-1

Further Discussion if any Clarifications Required

Thank You!