Remaining Useful Life estimation of a turbofan engine: an Extended Kalman Filter approach

Madelon Hulsebos

January 2018

1 Introduction

This report is part of the result of a project conducted in the context of the course Sensor Signal & Data Processing. Provided with the freedom to take on any topic of ones liking, I have chosen to dive into the problem of system failure diagnosis, prognostics and health management. More specifically, I elaborated on damage propagation modelling based on data stemming from a run-to-failure simulation with a turbofan engine[1].

All visualization and implementation related tasks were conducted in Matlab. The code is provided in three different files: `RUL_data_exploration.m`, `RUL_function_derivations.m` and `RUL_EKF_implementation_evaluation.m` along with the required (Matlab) data file `RUL_data.mat`. The code is accompanied by comments for ease of understanding. Each file can be executed separately and generates insightful figures. The code can also be found in this github repository.

Throughout the remainder of this report, figures are included to support preliminary understanding of the problem or visualize the findings and results. For all figures, green represents the actual situation or ground-truth, while blue corresponds to the estimated or approximations of some aspect.

This report starts with a description of the problem at hand in section 2, providing insight in the physical problem and the formal objective. This chapter is followed by section 3 providing the details of the chosen model and assumptions. In section 4 the details of the implementation of the model are presented along with the resulting engine state estimation on the training data. Section 5 provides more insight in the performance of the state estimation model by means of evaluation. Section 6 concludes this report with a brief conclusion.

[1] https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

2 Problem description

2.1 Physical problem

The data at hand describes the behavior of multiple turbofan engines over the number of operation cycles. These engine simulations have been observed by 21 sensors over different numbers of operation cycles under different conditions [1]. Each distinct engine is contaminated with unique sensor and process noise, but the engines all stem from the same engine fleet and therefore can be considered quite similar in this context.

In the remainder of this report, the generated data by each distinct engine is referred to as 'a run' such that we have multiple runs of approximately the same system (the engine), over different numbers of operation cycles (measurement time units).

2.2 Objective

In prognostics and health management the goal is to track the (often hidden) state of a system's component. In this project, the goal is to estimate the remaining useful lifetime of a turbofan engine. Here, the state of the system aimed to estimate is therefore the number of operating cycles it can still run until system failure, often referred to as the Remaining Useful Lifetime (RUL).

With the acquired knowledge of estimation techniques, it seemed that estimating the RUL using functional mappings from the previous RUL estimations and sensor measurements to the predicted RUL, would be a suitable approach to this problem. This approach is also identified in [2] as one of the three common approaches to estimate the RUL of turbofan engines.

3 Approach

3.1 Choice of model

I chose to implement a Kalman Filter-based (KF) method due to its optimality and computational performance; this method is based on matrix computations which are known to be efficient. Furthermore, in a survey on RUL estimation [3] it becomes clear that this type of filter is often applied to this type of problem. The main downside of deploying a KF-based estimation model as identified in [3] is that it provides point estimates. Considering the integer valued RUL labels of the provided test data set, which is used to evaluate the approach with, this is not an issue for this project.

The standard KF assumes the system is characterized by a linear system dynamics. However, as already stated in [2] the dynamics of the system at hand typically is nonlinear which becomes clear if one takes a preview at the relation between the input measurements and corresponding RUL in both images of figure 1. The KF method has been extended to fit such nonlinear systems as well, then called an Extended Kalman Filter (EKF). This model makes updates to the state and measurement vectors by means of linearizing the nonlinear function around the current estimate using Jacobian matrices. Despite it is being considered as a non-optimal method, it typically provides better results as it suits the nonlinear behavior of the system dynamics.

By means of an EKF model I try to estimate the (hidden) state st_t based on the observations stemming from sensor 4 and the previously estimated st_{t-1}. Sensor 4 was taken as its behavior seemed relatively constant over multiple runs. Not all sensors were however analysed carefully as that would be time consuming.

3.2 EKF formulation

KF-based approaches rely on predictions and updates of the state and covariance matrices at cycle t. It is assumed that the system is characterized by the following set of equations (1).

$$st_t = f(st_{t-1}, z_t) + w_t \tag{1}$$

$$z_t = h(st_t) + n_t \tag{2}$$

Here, st_t represents the state, z_t the *sensor* measurement and w_t and n_t the process and observation noise respectively, at cycle t. Note, that instead of taking z_t to represent the observation of the true state w.r.t. the RUL, it is the observation of the sensor at cycle t. Though this might not be standard practice, it is not uncommon either [4] as the RUL of the system will remain unknown (hidden) and cannot be observed during operation time of the engine. In KF-based models the noise components are assumed to stem from a Gaussian distribution with 0 mean and system-dependent variance. Function f predicts the new state from the previous state estimate and the sensor measurement where function h is used to predict the new measurement from the predicted state.

The measurement vector z_t consists only of one scalar, the measurement taken at cycle t. The state vector st_t can be seen as representing the RULs at t and $t-1$ and the difference between the RULs at t and $t-1$ as shown in expression 3. In the remainder of this report, the RUL at cycle t as variable will be referred to as rul_t.

$$st_t = \begin{bmatrix} rul_t \\ rul_{t-1} \\ \Delta rul_t \end{bmatrix} \tag{3}$$

The EKF model entails two types of equations; predict and update equations. The predictions are made with respect to the state and the covariance at cycle t given the previous estimates. These predictions are generated by equations (4) and (5) respectively.

$$\hat{st}_{t|t-1} = f(\hat{st}_{t-1}, z_t) \tag{4}$$

$$\hat{P}_{t|t-1} = F_t P_{t-1} F_t^T + Q \tag{5}$$

Recall that here, the sensor measurements z_t are used as control vector to estimate st_t. Therefore z_t does not reflect the observed state of the system at cycle t but at least gives some sense of it, as we can learn the relation between rul_t

and sensor measurement z_t which will be further discussed in section 4. This relation is then captured by function $f(\cdot)$. P_t refers to the state covariance matrix, Q the process noise covariance matrix, and F_t the state transition matrix approximated by the Jacobian $\frac{\partial f}{\partial st}$ in st_{t-1}, z_t.

The EKF process concludes with a set of update steps. The main intention of this procedure is to verify whether the predictions were about right, and updates the current estimates of the state and covariance matrices according to the sensor measurement. Equations (6) to (10) describe these steps.

$$y_t = z_t - h(\hat{st}_{t|t-1}) \tag{6}$$

$$S_t = H_t \hat{P}_{t|t-1} H_t^T + n \tag{7}$$

$$K_t = \hat{P}_{t|t-1} H_t^T S_t^{-1} \tag{8}$$

$$st_t = \hat{st}_t + K_t y_t \tag{9}$$

$$P_t = (I - K_t H_t) \hat{P}_{t|t-1} \tag{10}$$

The observation function $h(\cdot)$ predicts the next sensor measurement given the predicted state. H_t is the observation matrix approximated by the Jacobian $\frac{\partial h}{\partial st}$ in $st_{t|t-1}$. S_t represents the innovation covariance, where K_t is the Kalman gain, i.e. the amount of variability expected in the actual new state and therefore the added variability to the state estimate. Equations (9) and (10) conduct the actual updates to the state and covariance matrices given the new measurement to improve the estimation in the next iteration.

Since the observation of the state will be derived from the sensor measurement instead of the state variable itself, the measurement residual as in equation (6) does not correspond to the residual of the observed and the predicted state. The consequence of this adjustment is that the residual possibly has a large range due to the unstable, noisy behavior of the measurements as can be previewed in figure 1a. Techniques to compensate for this have not been studied but it is suggested to do so to improve the approach further.

3.3 Assumptions

The following assumptions and simplifications are made to implement the EKF:

- The rul_t can be estimated from the observation of one sensor and the previously estimated rul_{t-1}. That is, the estimation can rely on a learned mapping between sensor measurements z_t and rul_{t-1} to approximate the accuracy of the predicted state by means of the observed and predicted sensor measurement residual.
- The state transition function can be described well by a simple 2^{nd} order polynomial.

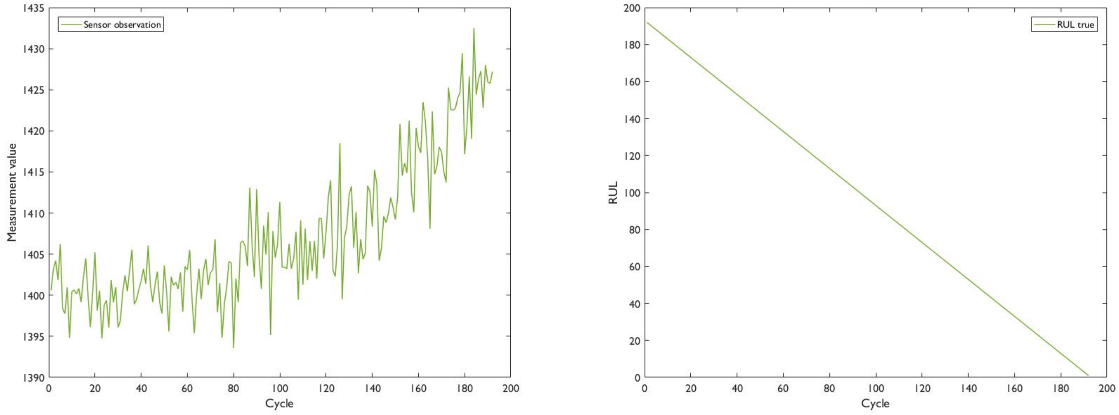
- The 2^{nd} order polynomial is derived from only one cycle of the sensor measurements in the training data instead of averaged over multiple cycles.
- The measurement/observation function to predict the measurement from the estimated $\hat{rul}_t$ can be described well by an exponential model.
- The measurement and process noise components are drawn from a Gaussian distribution with zero mean and known variance approximated from the training data.

4 Implementation Extended Kalman Filter

4.1 Input data

The data set provides 4 different subsets of measurements, each consisting of a train and test set and a true RUL label vector of the test data. I have taken the subset FD001 as this data set incorporates observations of the turbofan engine contaminated with only one fault type and the engines were run in one condition mode [5].

The FD001 training data set consists of multiple runs, each corresponding to one turbofan engine, and stops at the last cycle before the failure takes over the system behavior. An example of measurements of the first run from sensor 4 is shown in figure 1a. These measurements were taken as training data to get insights into the dynamics of the system and the relation between sensor 4 and the RUL. Figure 1b shows the true rul_t as computed from the training input data. The provided true RUL labels of the test data correspond to the RUL at the last cycle of a run in the test set where each run is stopped at an arbitrary cycle.



(a) Measurements of sensor 4 w.r.t. cycles

(b) The RUL derived from the training data

Figure 1: Input data

4.2 State transition functions

Figure 2 shows how the system dynamics is derived to come to function $f(z_t)$ for the initial estimate of the state st_t at cycle 1. The function here estimates rul_t directly from the raw measurement and provides the initial estimated (hidden) state of the system.

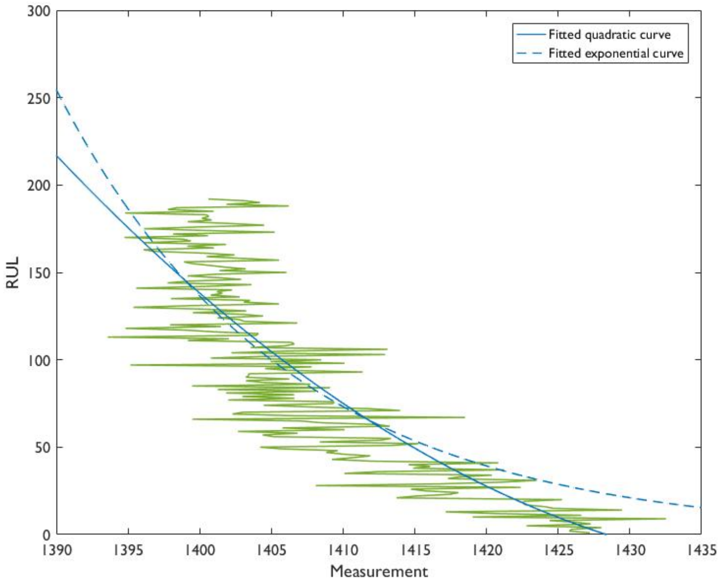


Figure 2: Fitted models on system variables

As the norm of the residual from the 2^{nd} order polynomial is smaller than from the exponential model, the 2^{nd} order polynomial was taken to construct the estimation function for the initial state. Though the RUL at $t-1$ is unknown, no significant consequence is expected when the engine is assumed to have $rul_t + 1$ cycles left at that point. However, this simplified relation between rul_t and rul_{t-1} does not necessarily correspond to reality as more factors might play a role. The difference is therefore -1. The resulting initial state prediction is shown in expression 4.2. The corresponding coefficients p_1 to p_3 can be found in table 1.

$$st_1 = \begin{bmatrix} rul_t = p_1 + p_2 z_1 + p_3 z_1^2 \\ rul_t + 1 \\ -1 \end{bmatrix} \tag{11}$$

Table 1: Coefficient values for initial $f(\cdot)$

coefficient	p_1	p_2	p_3
value	$1.6369 \exp 5$	-226.5514	0.0784

From there, a function to compute the RUL over the remaining cycles is derived which takes the RUL at cycle $t-1$ and the sensor output z_t as inputs. This function was fitted in Matlab from the behavior as shown in figure 3.

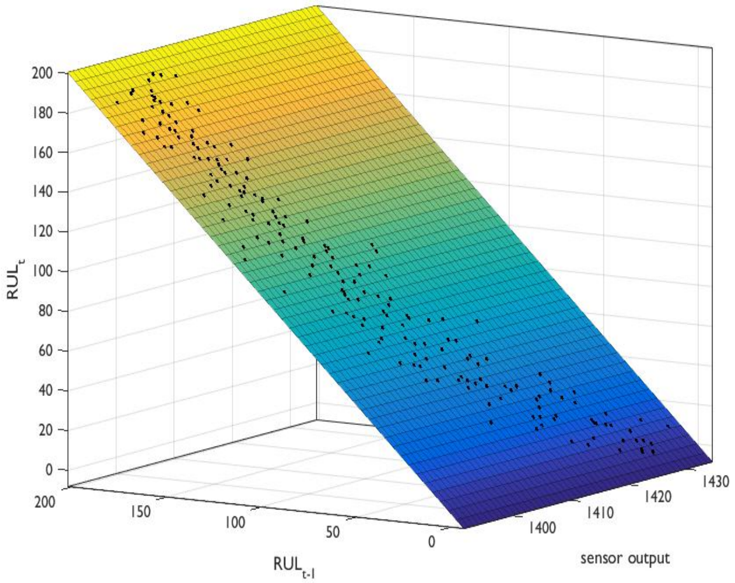


Figure 3: Relation rul_t, rul_{t-1} and the sensor output

The resulting state model is shown in expression 4.2. Table 2 provides the corresponding coefficients p_1 to p_6.

$$st_t = \begin{bmatrix} rul_t = p_1 + p_2 z_t + p_3 rul_{t-1} + p_4 (z_t)^2 + p_5 z_t rul_{t-1} + p_6 (rul_{t-1})^2 \\ rul_{t-1} \\ rul_t - rul_{t-1} \end{bmatrix} \tag{12}$$

Table 2: Coefficient values for $f(\cdot)$

coefficient	p_1	p_2	p_3	p_4	p_5	p_6
value	-1	$-3.23 \exp -13$	1	$1.135 \exp -16$	$2.978 \exp -17$	$2.579 \exp -10$

4.3 Measurement transition functions

Figure 4 shows the relation between the RUL and the sensor measurements used to derive the $h(st_t)$ function to predict the measurement given the state estimate. The dynamics is captured properly by the exponential model fitted as shown in the figure as well.

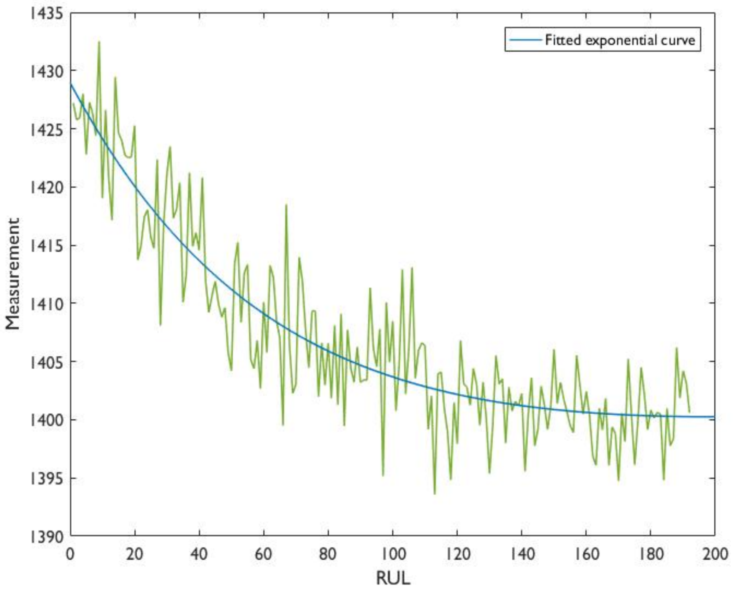


Figure 4: Relation sensor output and rul_t

$h(\cdot)$ only takes the current estimate of rul_t, where the resulting function is presented in expression (13). The corresponding coefficients are shown in table 3.

$$z_t = a \exp(b\, rul_t) + c \exp(d\, rul_t) \tag{13}$$

Table 3: Coefficient values for $h(\cdot)$

coefficient	a	b	c	d
value	35.2447	-0.0155	1.3937e3	1.7718e-5

4.4 State transition and measurement matrices

The state transition and measurement matrices F_t and H_t are defined as the linearised approximations at t from the functions $f(\cdot)$ and $h(\cdot)$. The Jacobian matrices F_t and H_t are shown in expressions (14) and (15). Each entry reflects the partial derivative of $f(\cdot)$ or $h(\cdot)$ with respect to each of the state variables (rul_t, rul_{t-1} and Δrul_t). The coefficients p_1 to p_6 remain as in table 2, where coefficients a to d are as in 3.

$$F_t = \begin{bmatrix} 0 & p_3 + p_5\, z_t + 2\, p_6\, rul_{t-1} & p_2 + 2\, p_4\, z_t + p_5 rul_{t-1} \\ 0 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix} \tag{14}$$

$$H_t = \begin{bmatrix} b \cdot a \exp\left(b \cdot rul_t\right) & 0 & 0 \end{bmatrix} \tag{15}$$

4.5 Resulting estimation given training data

Putting it all together, the implemented EKF can be deployed to estimate the RUL of the training data to inspect its performance. Figure 5 shows the estimated and true RUL over the 192 training cycles from which the EKF was formulated.

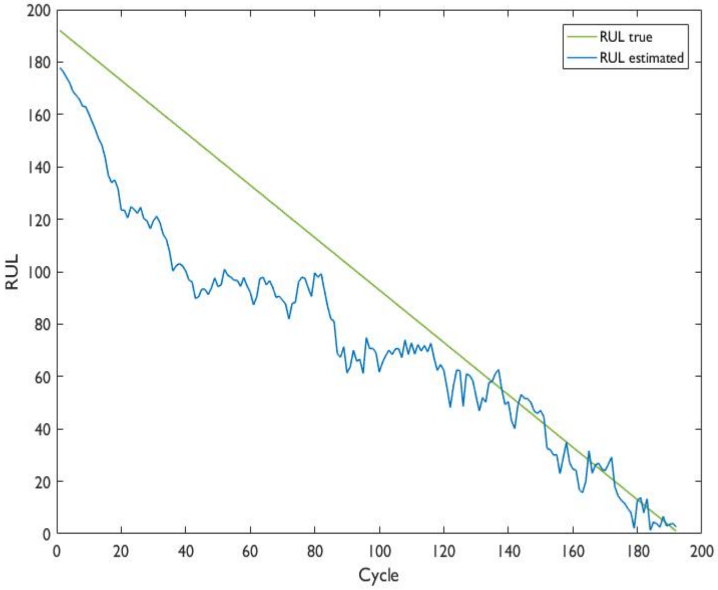


Figure 5: True rul_t and an estimation by means of the EKF over the training data

5 Evaluation

5.1 Evaluation setup

The data set also comprises labelled test data, which is used to evaluate the approach. Each test run, out of the 100 in total, is accompanied by just one integer label which reflects the RUL at the end of the run. The range of the RUL labels is [7, 145] [1], i.e. the run often does not end close to the end of the lifetime of the engine.

The performance of the EKF is evaluated in comparison to a Particle Filter (PF), which partly relies on the same derived functions. The PF is a nonlinear model as well, but approximates the posterior density by taking lots of particles as if they are sampled from the actual density. The likelihood of the particles are computed from the observations made. Based on the likelihood the particles are resampled.

As the data set stems from a public repository, many other approaches can be found in the literature. Hence, the approach is compared to those approaches as well if possible. The data set was constructed for hosting a system prognostics competition in 2008 which is evaluated in [2]. The results attained during the competition together with the results from an alternative KF-based approach as reported in a paper by Lim et al. [4], are used to compare the EKF and baseline PF with.

5.2 Evaluation metrics

To evaluate the performance, I have taken two metrics into account; the Root of the Mean Squared Error (RMSE) and a score introduced for the specific application of remaining useful lifetime estimation in [1]. The adapted score is introduced to make late predictions penalized more heavily than early predictions as the latter is preferred.

These metrics are defined by equations (16) and (17), where rul_t reflects the true RUL at cycle t and $\hat{rul}_t$ the estimated RUL. Note, that the numerators in equation (17) to compute the asymmetric score, 13 and 10 respectively, are reversed as opposed to what is stated in [1] as it seems they made an error in switching these values. Figure 6 illustrates the evaluation behavior of these metrics with respect to the error.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(rul_t - \hat{rul}_t)^{\,2}} \tag{16}$$

$$\text{score} = \begin{cases} \sum_{i=1}^{N} \exp -\frac{\hat{rul}_t - rul_t}{13} - 1 & \text{if } d < 0 \\ \sum_{i=1}^{N} \exp \frac{\hat{rul}_t - rul_t}{10} - 1 & \text{if } d \geq 0 \end{cases} \tag{17}$$

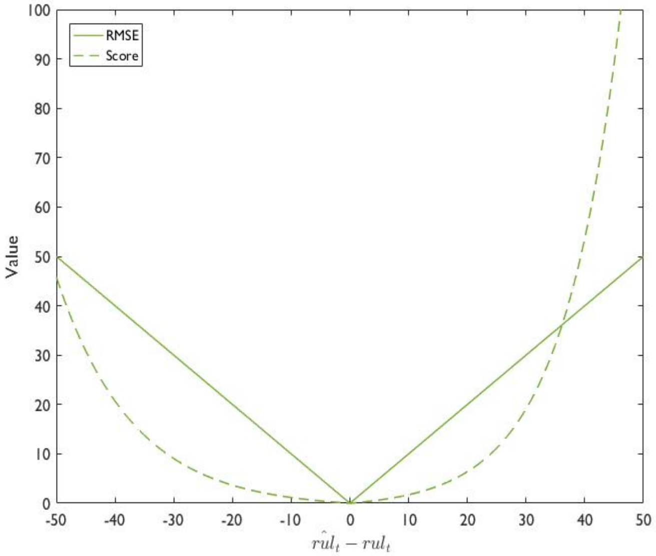


Figure 6: Evaluation behavior of metrics

Finally, the EKF is compared to the PF with respect to its runtime performance. As stated in 3, a KF-based approach was chosen as it is expected to be a faster algorithm than the PF while attaining a similar level of accuracy. The PF introduces a trade-off effect between the runtime and the estimation accuracy. That is, the estimation accuracy of the PF improves if more particles are added, at the cost of the runtime performance. The runtime performance is therefore evaluated accordingly.

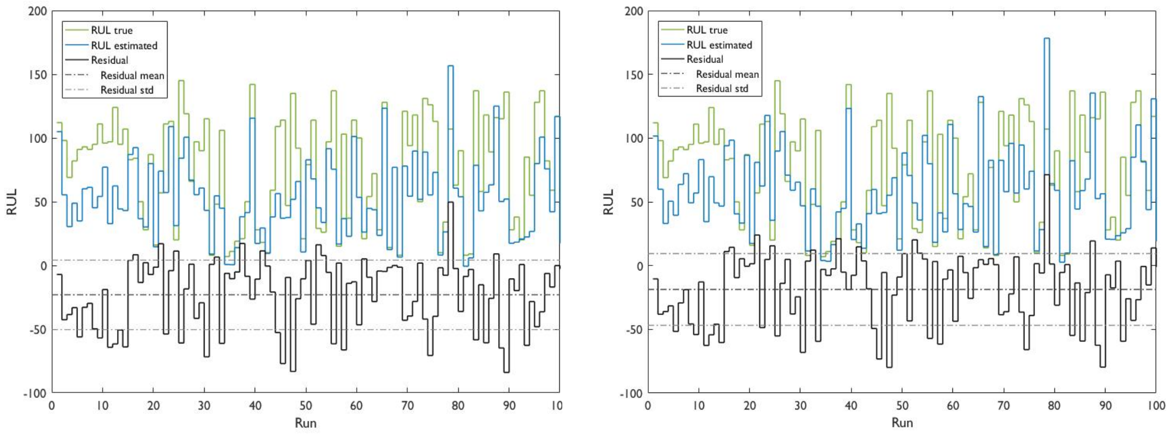
5.3 Results

The true RULs and the ones estimated by the EKF and PF are shown in figure 7. Additionally, the residuals are shown with corresponding mean and standard deviation. The mean of the residual attained by the EKF is -22.09 which reflects the tendency of the EKF to make (too) early predictions. Looking at the PF, its residual mean of -18.72 indicates this method estimates RULs later than with the EKF.

Table 4 shows the performance of the EKF and PF evaluated with respect to the metrics. The results of the PF presented are averaged over 10 runs, with 5000 particles.

Table 4: EKF performance on metrics

Metric	RMSE	Score
EKF	35.01	4081.0
PF	33.67	4151.6



(a) Extended Kalman Filter

(b) Particle Filter

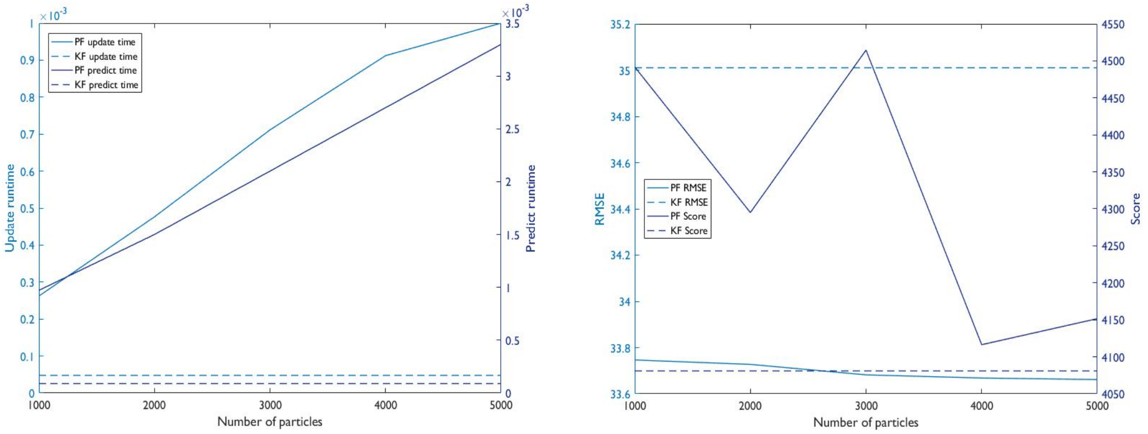
Figure 7: Comparison between true and estimated RUL

Considering the RMSEs and the range of [7, 145] of the true RUL, the current approach performs surprisingly well despite the numerous simplifications made. As the RMSE is normalized over the number of runs, the performance as reflected by the RMSE can also be compared to approaches in [2]. The performances of the 10 winning approaches as measured by the RMSE range from 23.38 to 32.86 [2]. Clearly, the attained RMSE of 35.01 is not too far off. Nevertheless, the PF attains a score of 33.67 which is even closer to the winning approaches considering this metric.

The performances on the adapted score metric are harder to compare to approaches based on the other data sets, as this metric is not normalized over the number of runs. Fortunately, in [4] some algorithms are presented on the same test set (FD001). The performance of an ensemble of standard linear KFs attains a score of 5000. Compared to the attained score of 4081.0, it can be concluded that the presented simplified but nonlinear EKF is quite effective. As already indicated by the means of the residuals, the PF makes later predictions than the EKF on average. This is also reflected by the resulting score of 4328.4. Taking the objective of this study into account, it can be concluded that the EKF outperforms the PF on that account. If, however, the preference of early-predictions would not have been an important aspect in the evaluation, the PF would be a better approach to adopt considering its RMSE.

Nevertheless, from the results in [4] it can be concluded that an ensemble of neural networks is likely to outperform the PF as well as the EKF implementation.

Finally, the runtime of the EKF is compared to that of the PF which has been averaged over 10 runs. As the number of particles influences both the runtime and the estimation performance of the PF, figure 8 also shows this trade-off effect given the performance metrics (RMSE and the score) for different numbers of particles. Figure 8a shows how the EKF is faster for both the predict as well as the update step. In figure 8b it can be seen that the EKF yields a lower score while the RMSE of the PF is lower. Taking both figures into account it can be seen how the estimation accuracy of the PF increases with an increasing number of particles at the cost of a lower runtime performance.



(a) Runtime over # particles

(b) Performance on metrics over # particles

Figure 8: Trade-off between estimation accuracy and runtime performance

A final note, is that in [1] it is noted that it is much harder to predict the remaining cycles at a point farther away from the end of the failure than close to it. As the test data consists of many test runs stopping far before degradation – the average number of remaining cycles in the test data is 75.5 – it seems that this might indeed influence the performance of the EKF, as well as the PF.

5.4 Future improvements

As already hinted throughout the report, I can think of numerous potentially significant improvements of the current EKF approach to make it more competitive to the approaches as in [2].

Starting with the significant simplifications, the resulting estimation is expected to improve if the state transition and measurement functions are derived from the average dynamics of the system. That is, 100 engine runs are available for each sensor until the failure occurs. These runs can be used to attain a more stable description of the system behavior, instead of relying fully on one measurement series corresponding to one run.

Another enhancement likely to improve the estimations is by incorporating a subset of sensors in the EKF. For this project, the RUL is estimated from only a single sensor, sensor 4. Incorporating multiple sensors is likely to improve the estimation as it can make the estimation model more robust. In [2] a commonly exploited subset of sensors is presented.

Also, the function f used to estimate the initial RUL state of the turbofan engine seems crucial for the results. As can be seen in figures 2 and 5, it is likely that this initial function underestimates the actual RUL as the residuals are relatively high in the early cycles of the run. Taking more time to optimize the estimation of the initial RUL from the first sensor measurement is expected to improve the results significantly.

It is also interesting to evaluate the influence of the operational setting of the turbofan. This information is included in the data set but enhancing the model with this data is considered out of scope of this project.

Finally, though not actually an improvement, but potentially a critical aspect to study further, is whether significant implications have been ignored after taking the raw sensor measurements as being representative of the observed state in the update procedure of the EKF. It is not hard to imagine that a normalization component should have been included to align the two sources of information, i.e. the measurement residual in equation (6) and the updated state estimate in equation (9).

6 Conclusion

In this report, an approach to the estimation of the Remaining Useful Lifetime of a turbofan engine is presented. The approach is based on an Extended Kalman Filter (EKF), where the sensor measurements from only one sensor over one run were used to derive the required functions and models.

The details of the implementation are provided accompanied by the resulting Remaining Useful Lifetime estimations of the model on the data used for training. The approach is evaluated using publicly available test data of which the labels are provided as well. The performance of the approach is evaluated by means of the RMSE and an adapted score metric for the specific problem of Remaining Useful Lifetime estimation. It concludes with an overview of aspects of the approach that can be improved further.

The evaluation of the approach indicates that the model performs reasonably well considering the simplifications and assumptions made. It has also been shown how the EKF attains a lower score (which is preferred) than a Particle Filter (PF) while it has a lower runtime as well. The PF attains a lower RMSE, though makes more late predictions. Even though most concurrent approaches as for instance reviewed in [2] and [4] perform better, the EKF shows high competitive potential if the implementation is improved as suggested.

References

[1] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pp. 1–9, IEEE, 2008.

[2] E. Ramasso and A. Saxena, "Performance benchmarking and analysis of prognostic methods for CMAPSS datasets," *International Journal of Prognostics and Health Management*, vol. 5, no. 2, pp. 1–15, 2014.

[3] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation – a review on the statistical data driven approaches," *European journal of operational research*, vol. 213, no. 1, pp. 1–14, 2011.

[4] P. Lim, C. K. Goh, K. C. Tan, and P. Dutta, "Estimation of remaining useful life based on switching kalman filter neural network ensemble," tech. rep., Rolls Royce Singapore Singapore Singapore, 2014.

[5] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set." http://ti.arc.nasa.gov/project/prognostic-data-repository, 2008.