Machine Learning Applications

Winter semester 2019/2020

Simon Mehringskötter

Lecture IX

Prognosis Framework

Structure and content of the lecture

VW	Date	Content	Lead
1	18th Oct 2019	Introduction, Motivation and Organizational Information	FSR
2	25th Oct 2019	Introduction to Machine Learning I	CS
3	1st Nov 2019	Introduction to Machine Learning II	CS
4	8th Nov 2019	Introduction to Machine Learning III	CS
5	15th Nov 2019	Introduction to and Application of Machine Learning	CS
6	22nd Nov 2019	Presentation of Industrial Partner and Introduction of Practical Exam (Hackathon)	FSR/EXT
7	29th Nov 2019	**Data Understanding:** Data Acquisition, Preprocessing and Feature Engineering	FSR
8	6th Dec 2019	**Diagnostics:** Feature Engineering, Regression, Health Assessment	FSR
9	13th Dec 2019	**Prognostics:** Prediction, Remaining Useful Life, Uncertainty	FSR
10	20th Dec 2019	**System of Systems:** System Level, Systemwide Predictions, Decision Support, IoT	FSR
11	17th Jan 2020	**Predictive Maintenance:** Band Saw Application, Neural Networks for Condition Monitoring and Prediction	PTW
12	24th Jan 2020	**Predictive Quality:** Prediction of Product Quality during Machining Process	PTW
13	31st Jan 2020	**Energy Forecasting:** Electric Load Forecasting by the Example of a Machine Tool	PTW
14	7th Feb 2020	**Operational Control:** Optimized Control of Cross-Linked Energy Systems by Means of Reinforcement Learning	PTW
15	14th Feb 2020	**Recap on Lecture for Written Exam**	FSR

Structure and content of the lecture

VW	Date	Content	Lead
1	18th Oct 2019	Introduction, Motivation and Organizational Information	FSR
2	25th Oct 2019	Introduction to Machine Learning I	CS
3	1st Nov 2019	Introduction to Machine Learning II	CS
4	8th Nov 2019	Introduction to Machine Learning III	CS
5	15th Nov 2019	Introduction to and Application of Machine Learning	CS
6	22nd Nov 2019	Presentation of Industrial Partner and Introduction of Practical Exam (Hackathon)	FSR/EXT
7	29th Nov 2019	**Data Understanding:** Data Acquisition, Preprocessing and Feature Engineering	FSR
8	6th Dec 2019	**Diagnostics:** Feature Engineering, Regression, Health Assessment	FSR
9	13th Dec 2019	**Prognostics:** Prediction, Remaining Useful Life, Uncertainty	FSR
10	20th Dec 2019	**System of Systems:** System Level, Systemwide Predictions, Decision Support, IoT	FSR
11	17th Jan 2020	**Predictive Maintenance:** Band Saw Application, Neural Networks for Condition Monitoring and Prediction	PTW
12	24th Jan 2020	**Predictive Quality:** Prediction of Product Quality during Machining Process	PTW
13	31st Jan 2020	**Energy Forecasting:** Electric Load Forecasting by the Example of a Machine Tool	PTW
14	7th Feb 2020	**Recap on Lecture for Written Exam**	FSR
15	14th Feb 2020	**Operational Control:** Optimized Control of Cross-Linked Energy Systems by Means of Reinforcement Learning	PTW

What should you be able to take out of the lecture today?

- What is prognosis and when does it make sense?
- Determine health of a component and transfer curve into prognosis
- What is remaining useful life (RUL)?
- How to do a prognosis with Gaussian Process Regression?
- Can I trust in my prognosis?
- How to evaluate the prognosis performance?
- What are the challenges in real world application?

The prognosis is one of the last steps in OSA-CBM

TECHNISCHE UNIVERSITÄT DARMSTADT

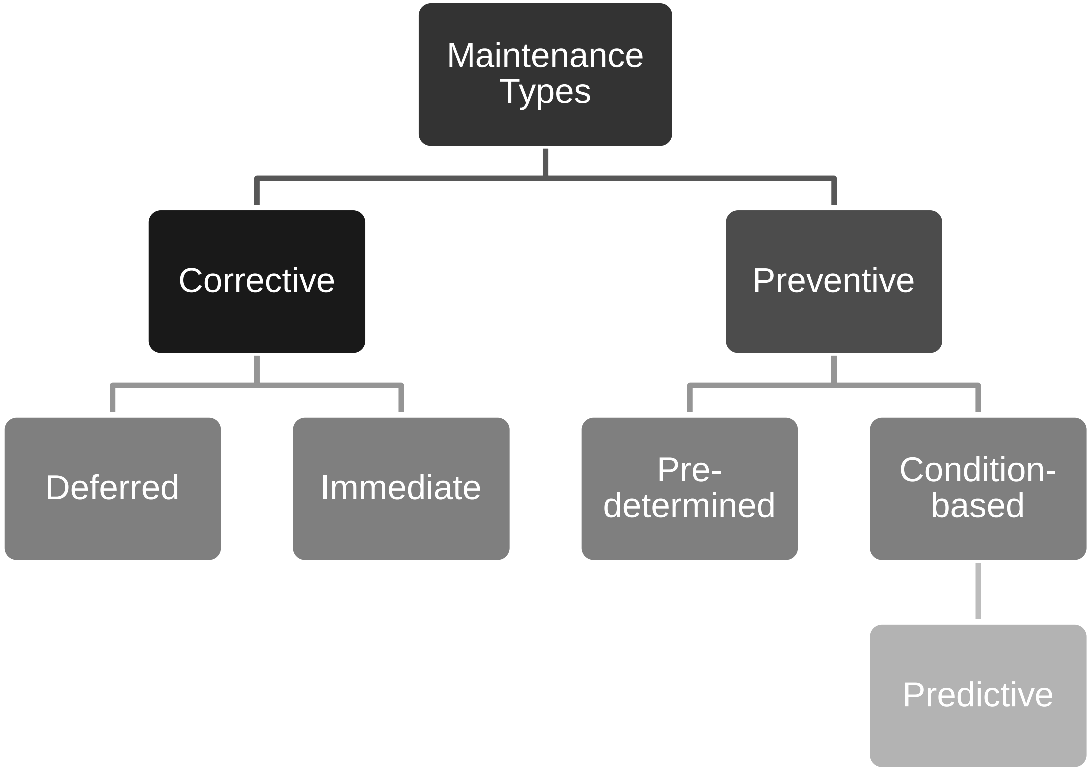
1.1	Data Acquisition (DA)
2.1	Data Manipulation (DM)
2.2	State Detection (SD)
2.3	Health Assessment (HA)
3.1	Prognostics Assessment (PA)
3.2	Advisory Generation (AG)
4.1	Presentation

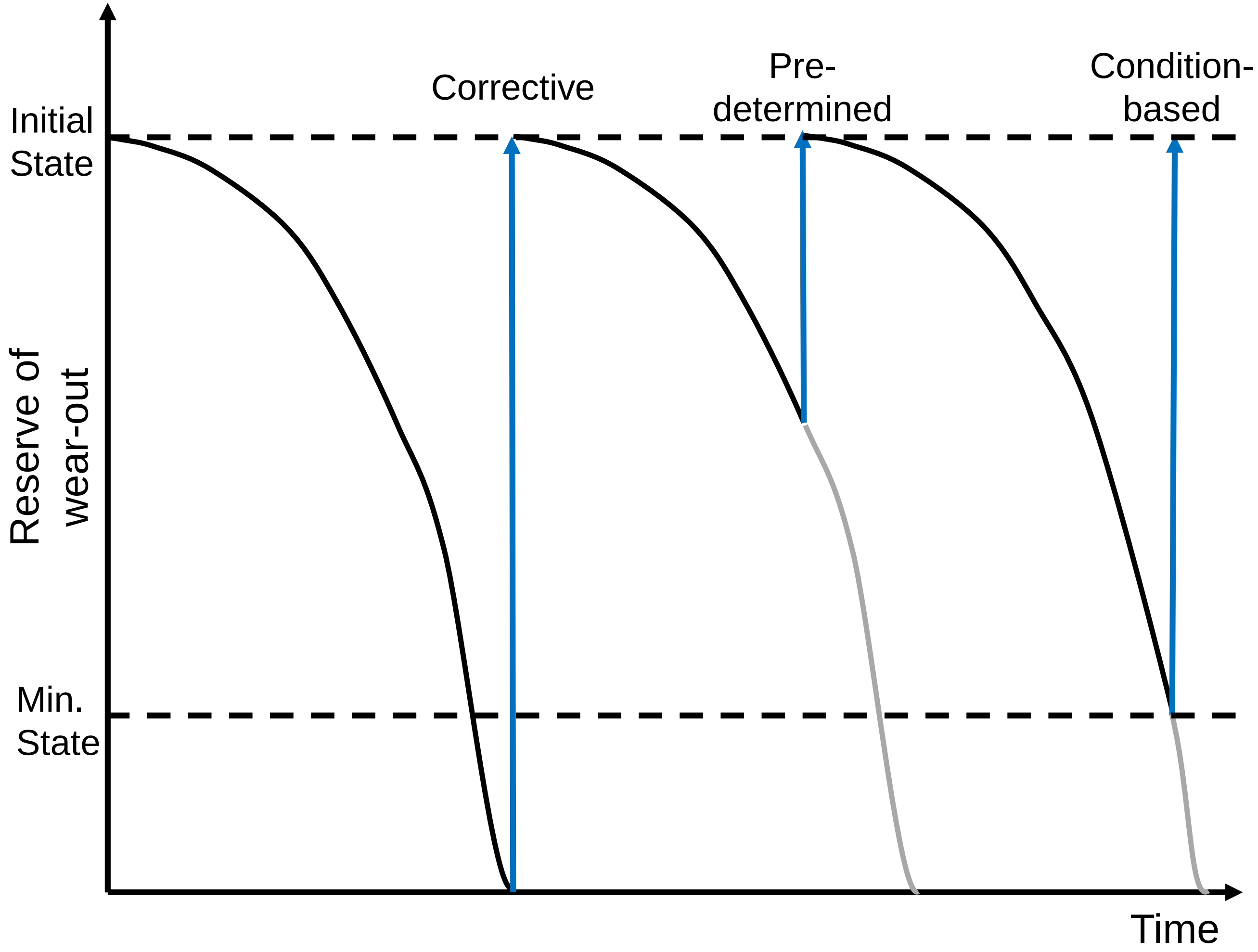
Increasing Level of Aggregation

- PA (typically) completely relies on previous steps 1.1 till 2.3
- PA aims to predict future behavior of investigated component
  How will the health of my component develop in the future?
- HA describes the health of a component (e.g. Health Index)
  What is the health status of my component?
- AG combines result of PA with system/expert knowledge
  Which actions and when should I take for the component?

THE GOAL OF PROGNOSIS

Maintenance can benefit from component‘s health prognosis

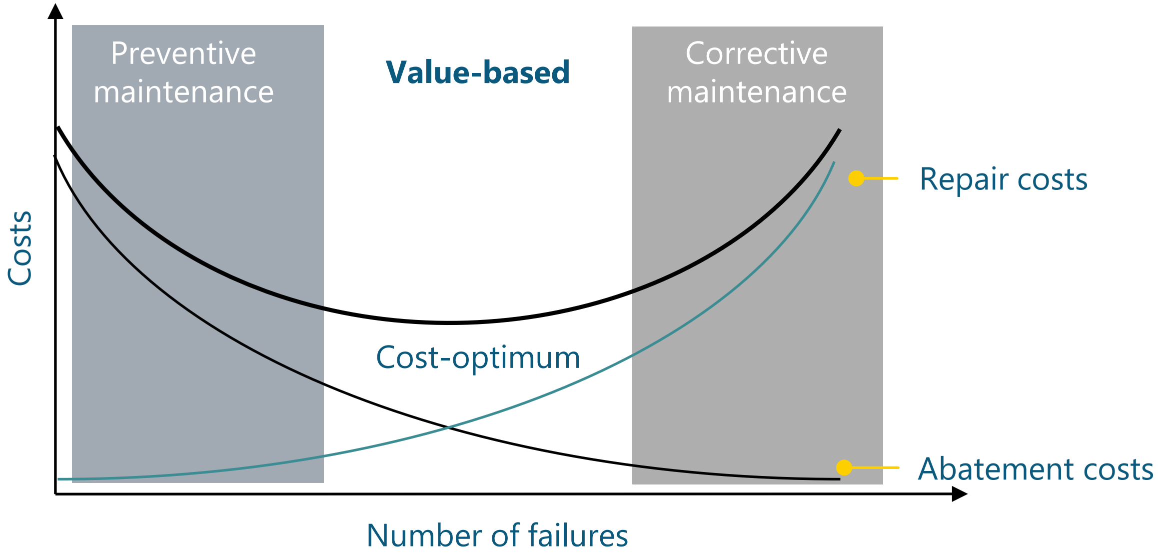




Condition-based maintenance can optimize the maintenance costs

- Model solely considers maintenance costs
- Development costs of value-based maintenance not covered
- Resulting costs from breakdown not covered → Revenue loss

➔ Each component needs to be analyzed individually



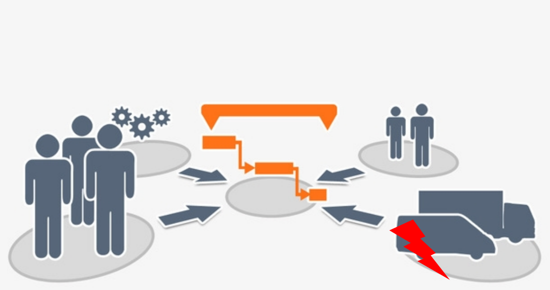
- Repair costs increases with higher number of failures
- Abatement costs increase for lower number of failures

➔ Optimum in the addition of both costs

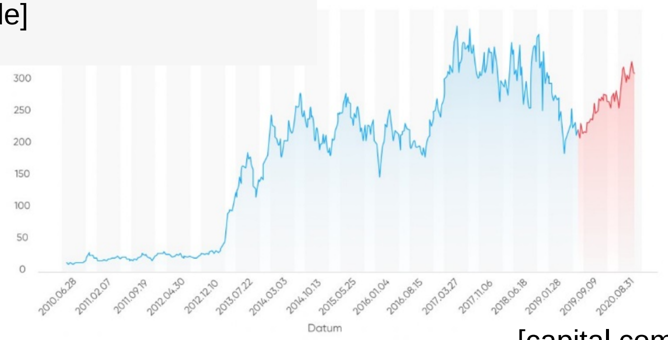
Maintenance is not the only reason for prognosis approaches

- Safety related components
  - Demand for high reliability
  - Alternative to conventional redundancy concepts
- Optimized resource planning
  - Know in advance when assets are not available
  - Know future demand for resources
- Prediction of process output
- Load prognosis
- Stock market
- …

[sfl-gmbh.de]



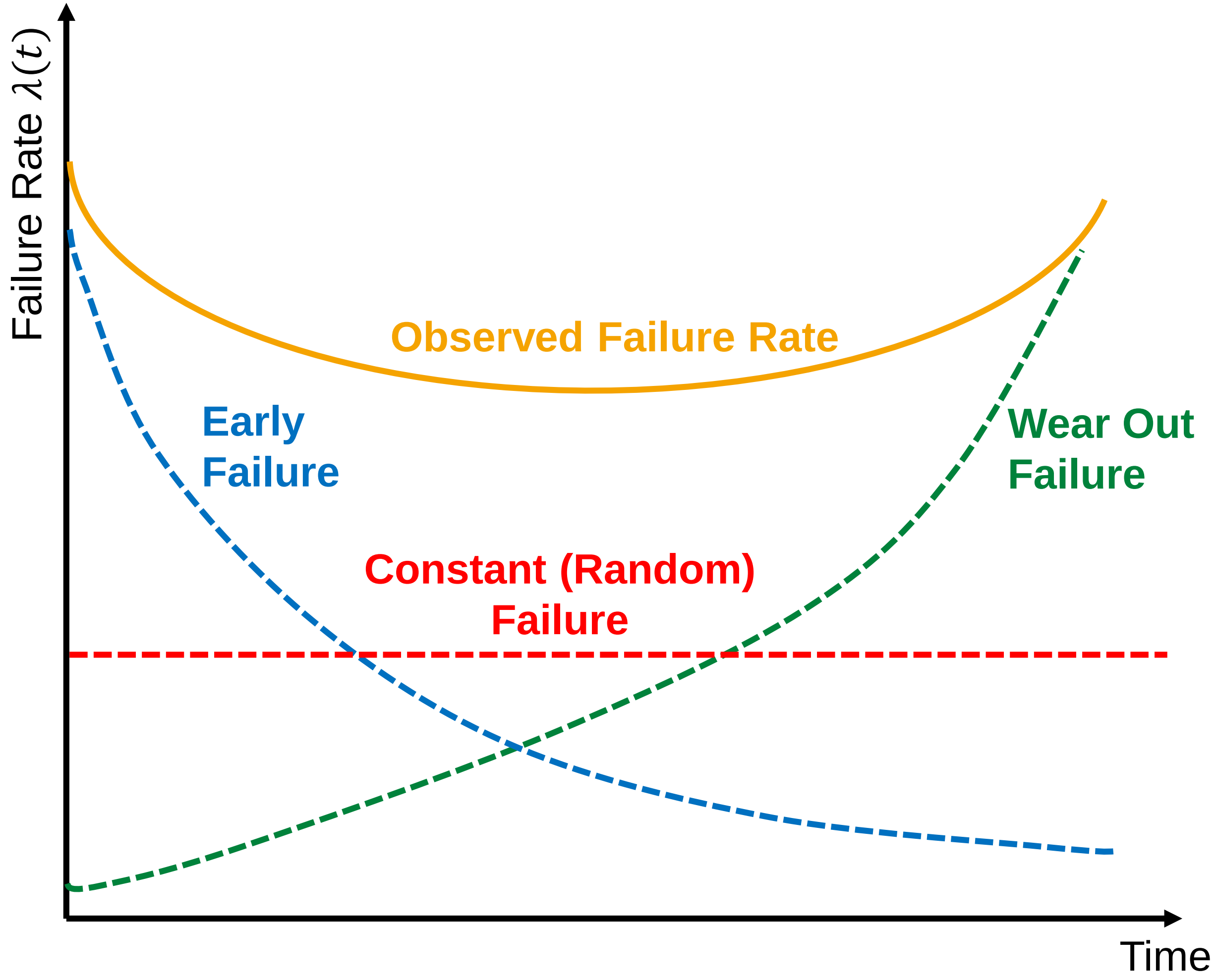
[gruenderszene.de]



[capital.com]

Prognosis of component's health is not always technical feasible

- **Early Failure**
  - Production quality
  - Wrong design
  - → Any component
- **Wear Out Failure**
  - Aging of a component
  - Degradation of component
  - → Typically mechanical components
- **Constant (Random) Failure**
  - **Unpredictable** cause
  - → Typically electronic components

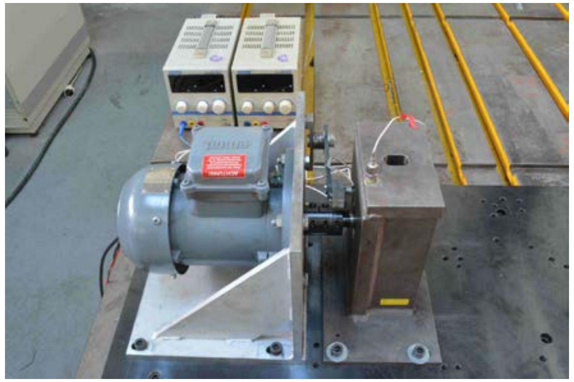


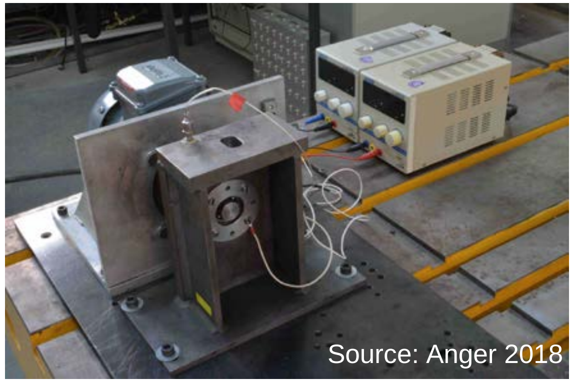
HOW TO ASSES A COMPONENT‘S HEALTH

A test rig can be used to generate run-to-failure curves

Bearing degradation

- Rotation from induction machine
- Current through bearing to accelerate degradation

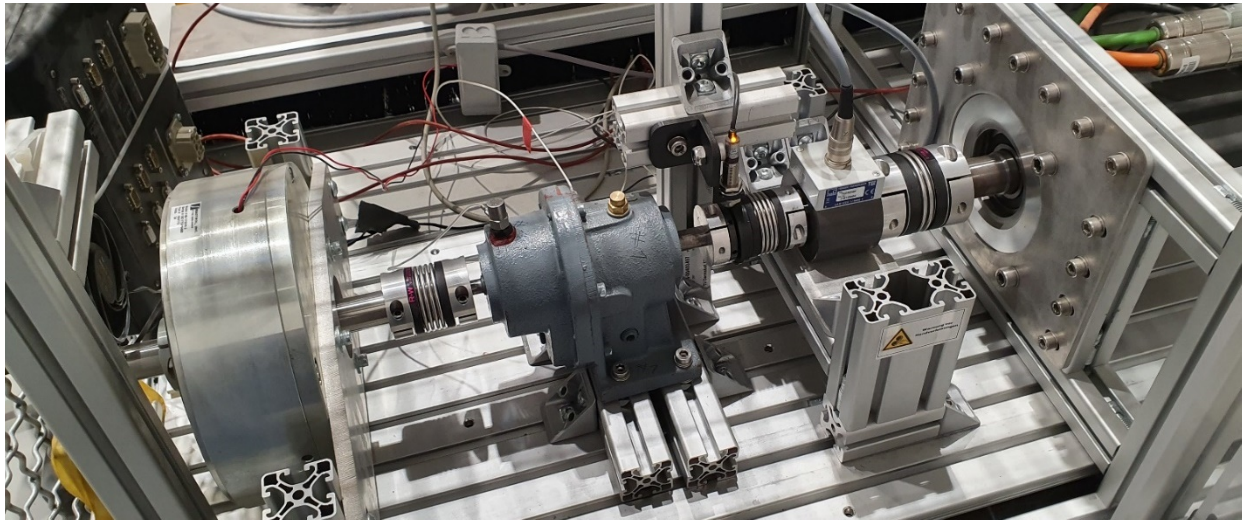




Source: Anger 2018

Gearbox degradation

- Motor for rotation
- Hysteresis brake to apply different loads

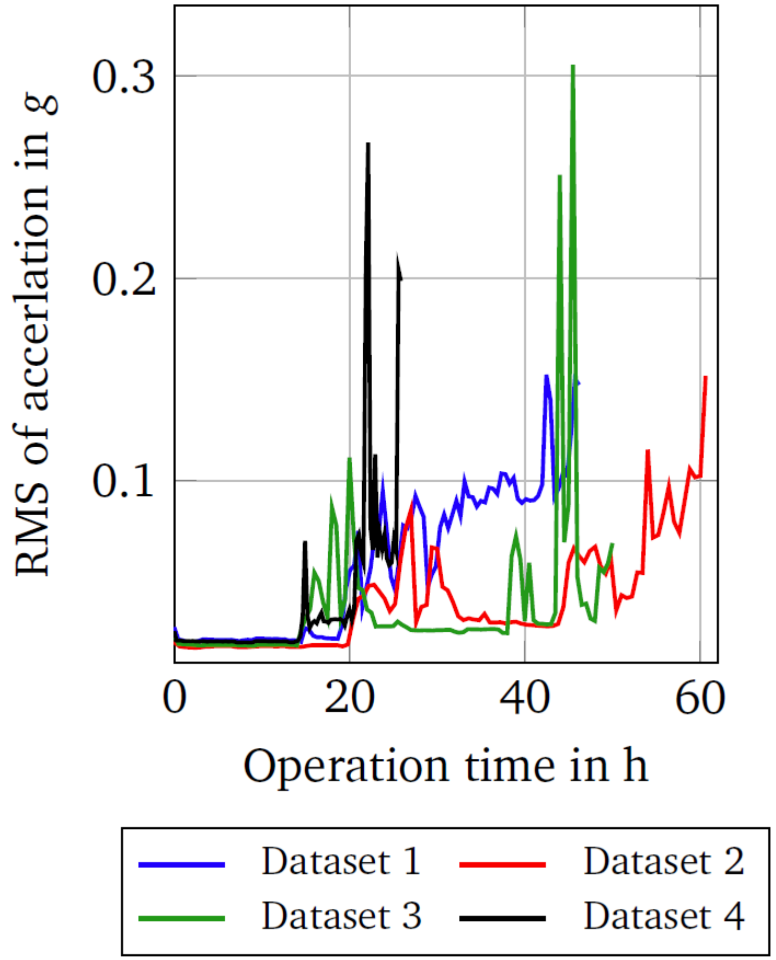


Different values of a feature may indicate a component's break down

- Operation time until breakdown differs between components
- RMS value of acceleration might differ when component actually fails
- RMS values might decrease again
  → pseudo repair

➔ Determine a component specific feature value baseline

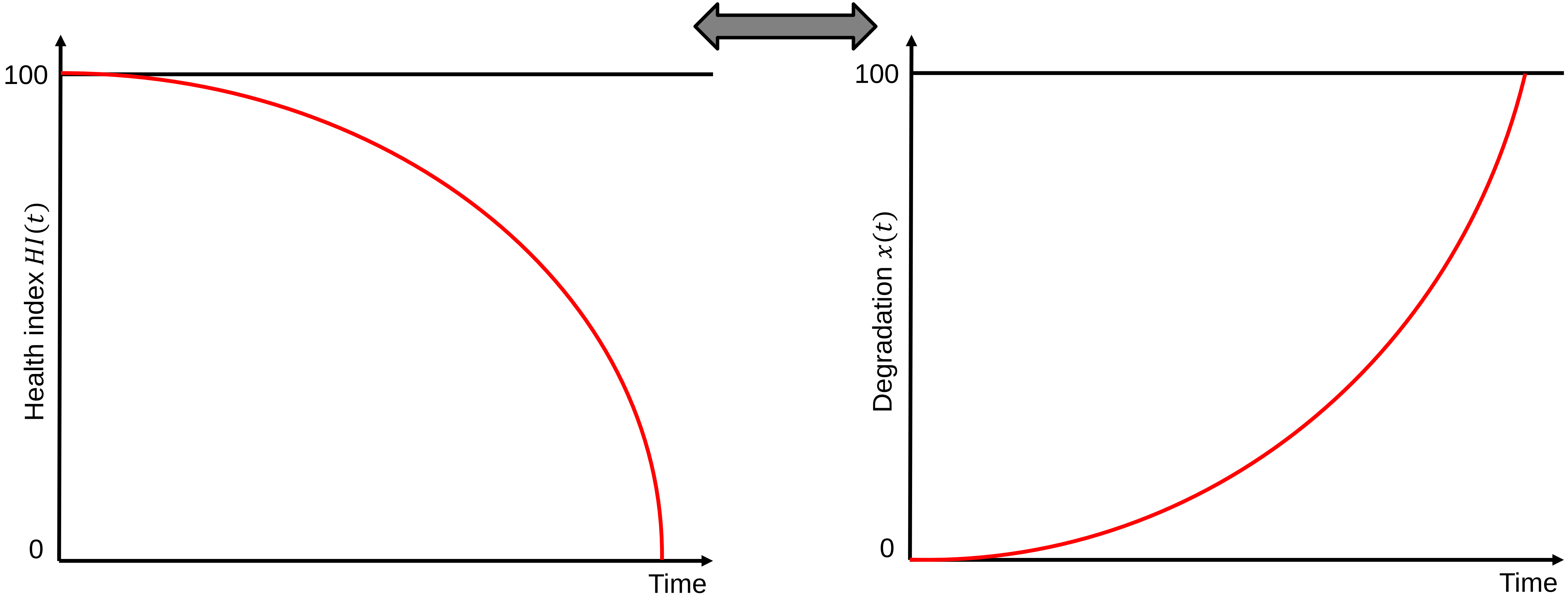
➔ Normalize values to predefined range (e.g. [0,1])



Source: Anger 2018

Health index or degradation value can be used to describe wear-out

$$x(t) = 1 - HI(t)$$

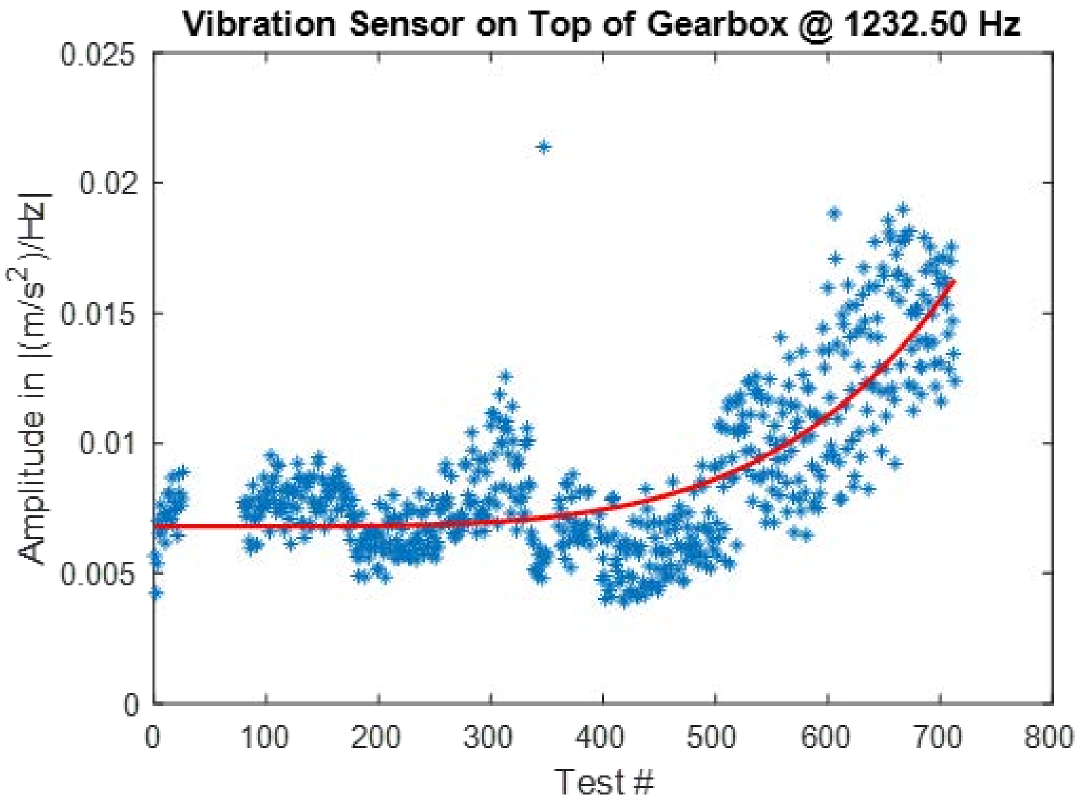


HOW TO PREDICT A COMPONENT'S REMAINING USEFUL LIFE

There are four assumptions that are ideally considered for prognosis

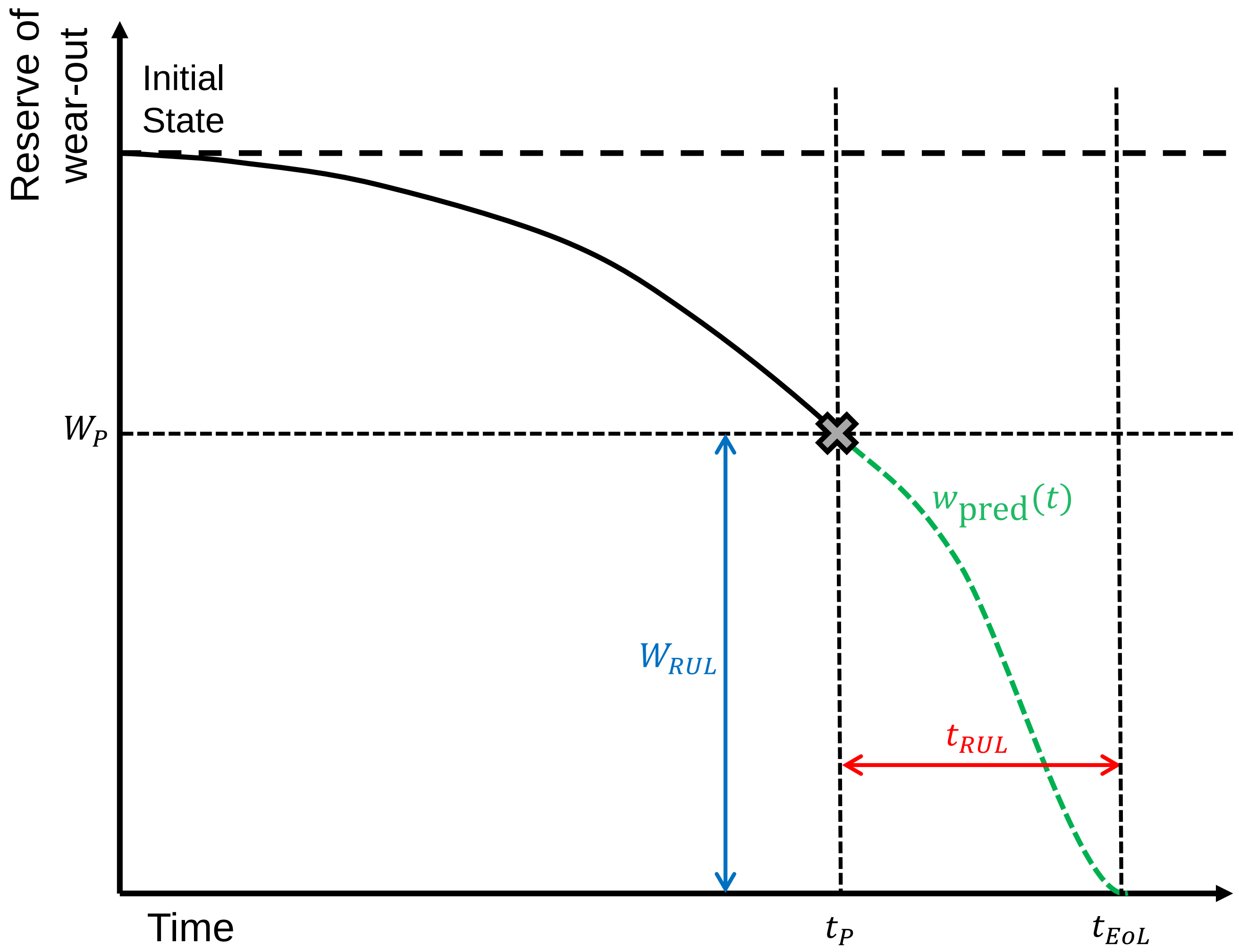
TECHNISCHE UNIVERSITÄT DARMSTADT

1. The monitored system **degrades** as a function of **use**, **time** and **environmental conditions**
2. The **aging** and **damage accumulation** is a **monotonic process**
3. Signs of **aging** are **visible before** the **failure** of the system occurs
4. Signs of **aging** can be **fitted** to a **model** to estimate the remaining useful life



Remaining Useful Life describes the remaining wear out/time/… reserve

- Remaining Useful Life (RUL) can be interpreted in different ways
  - t_{RUL} remaining useful life-time
  - W_{RUL} remaining useful life-wear-out
- Use case dependent what to predict
  - t_{RUL} if you want to know the time until a component fails → most cases
  - $w_{\text{pred}}(t)$ if you want to know how the wear-out will develop over the time

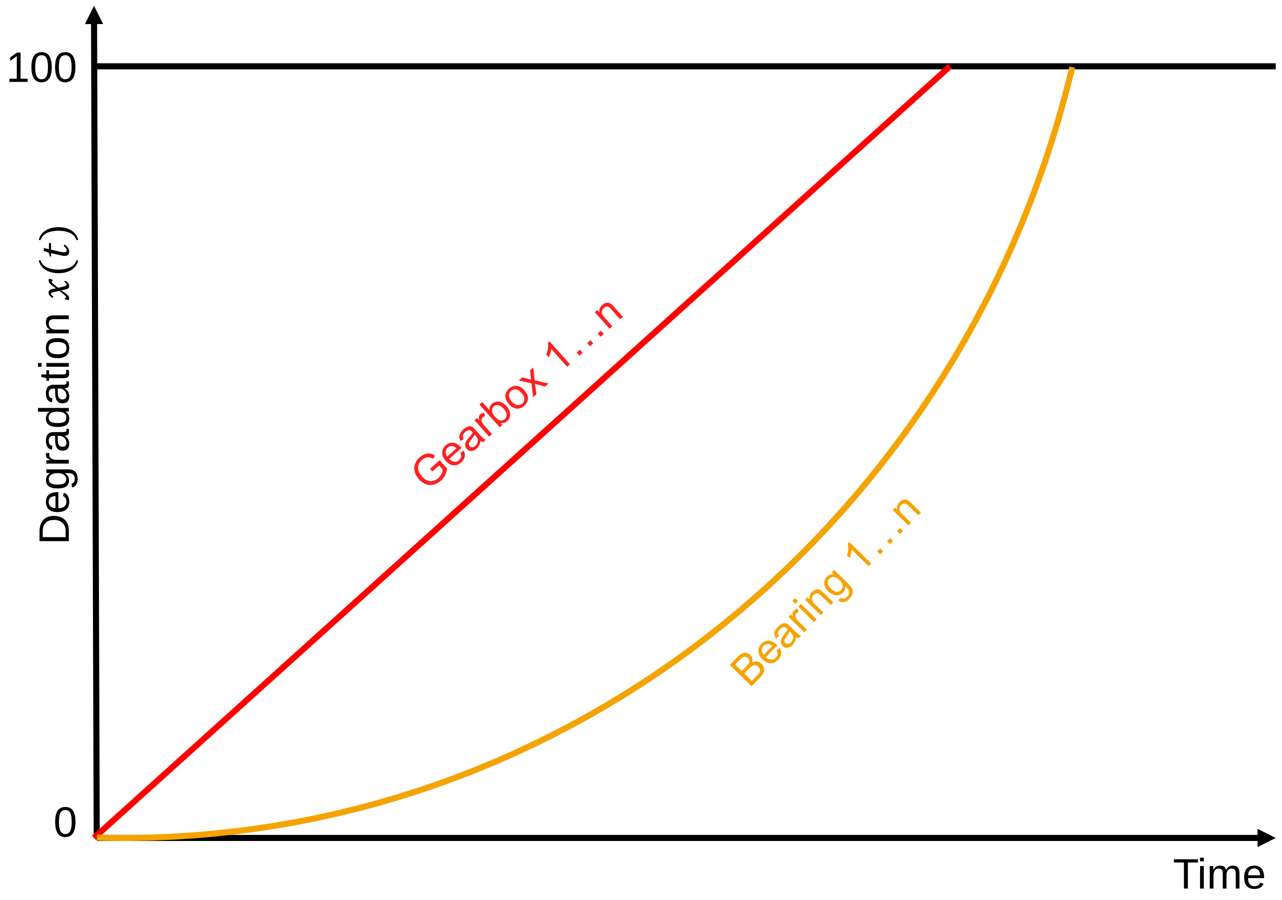


Regression that maps degradation solely over time is no prediction

TECHNISCHE UNIVERSITÄT DARMSTADT

- Simple regression when each component behaves in the same way
- $x_1(t) = x_2(t) = \cdots = x_n(t)$
- It does not matter which type of function $x(t)$ represents (linear, exponential,…)

➔ Since all functions are the same, there is no real prediction model

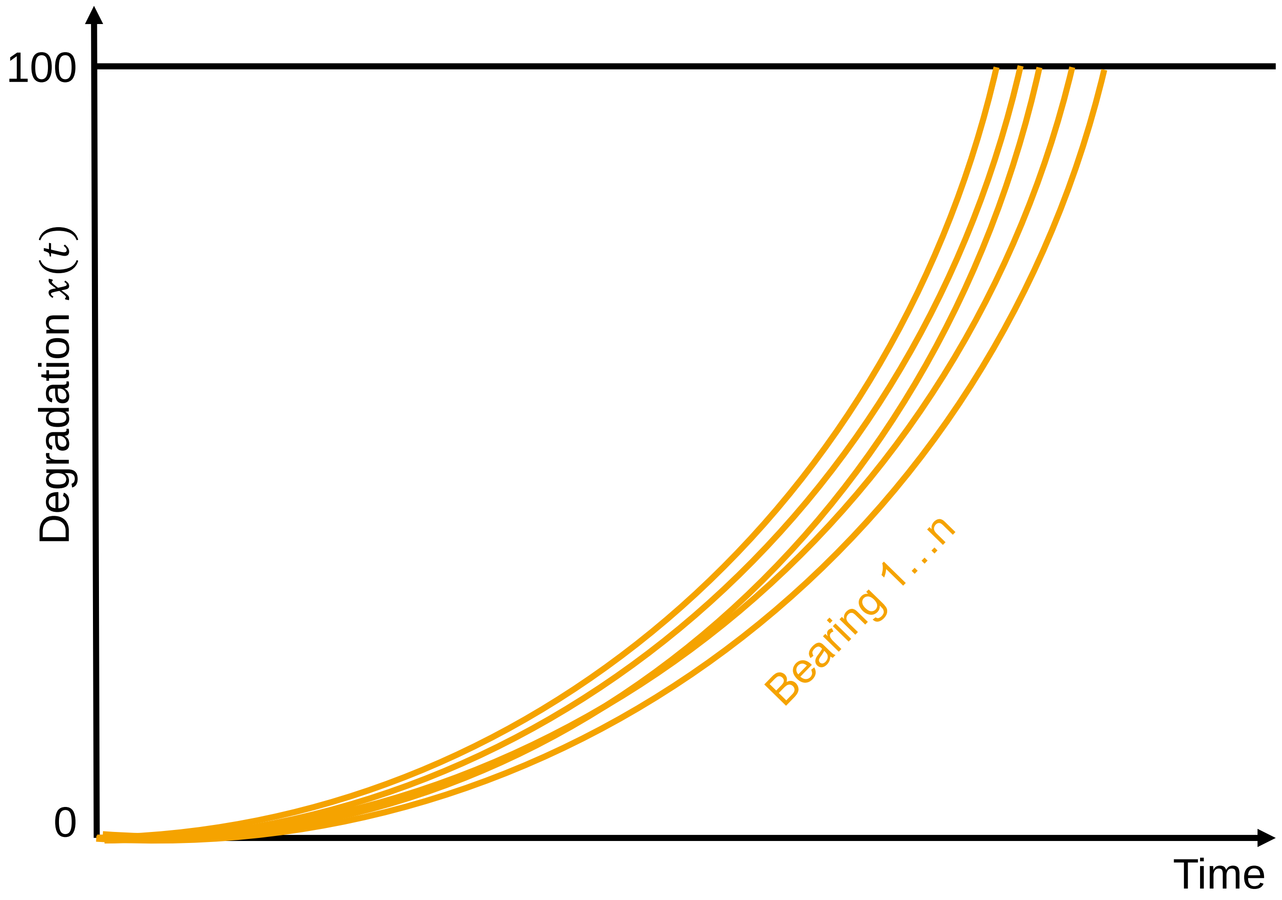


FSR

Prognosis algorithm copes with different behavior of components

- Each gearbox degrades differently
- $x_1(t) \neq x_2(t) \neq \cdots \neq x_n(t)$
- Prognosis algorithm has to incorporate further information
  - Component's degradation history
  - Further sensor values

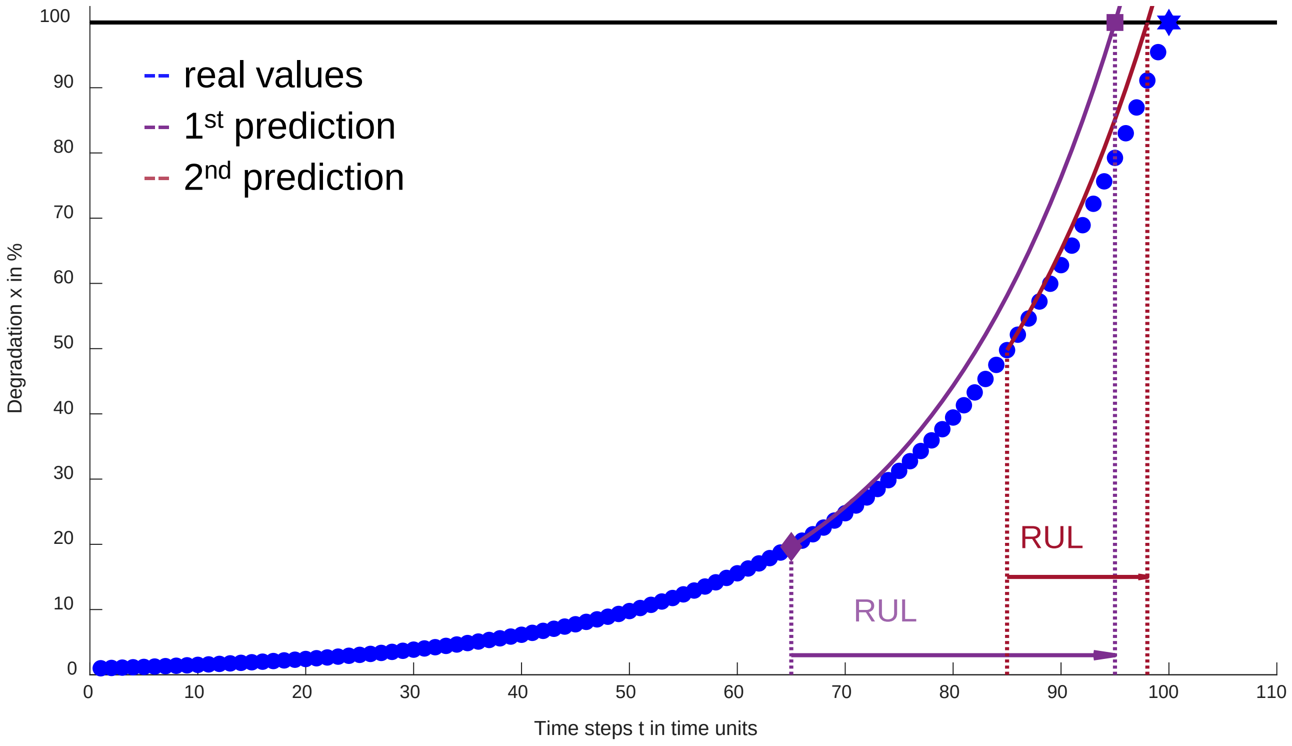
➔ Since all functions differ, a *complex* model is necessary for prediction



Prognosis learns behavior of components and predicts at different time steps

TECHNISCHE UNIVERSITÄT DARMSTADT

- Typically predictions are not accurate
- Typically later predictions tend to be closer to real end of life



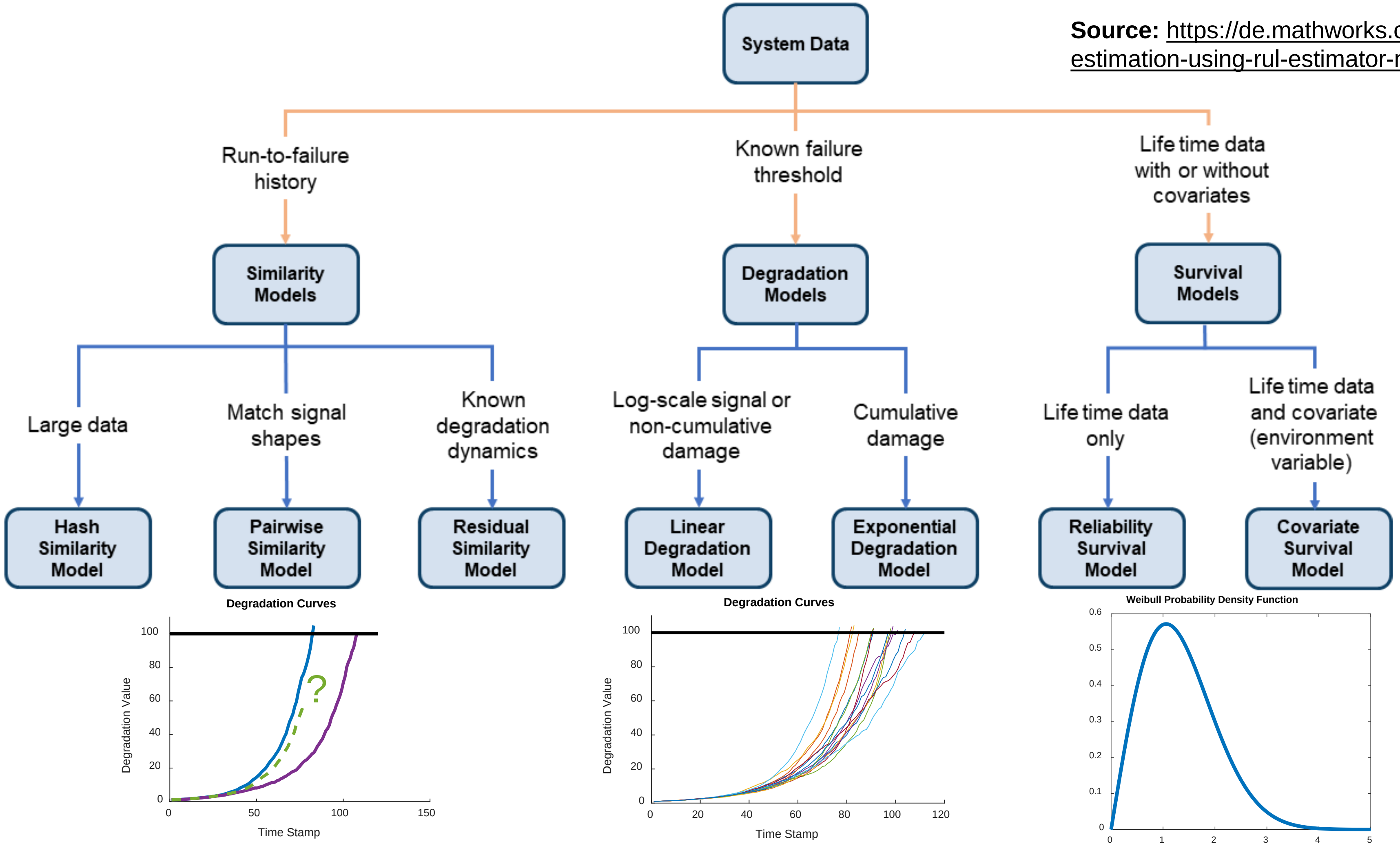
t_{EoL} → End of life time (reality)

t_{p} → Start time of prediction

t_{EoP} → End time of prediction

Different approaches according to available (historic) system data

Source: https://de.mathworks.com/help/predmaint/ug/rul-estimation-using-rul-estimator-models.html



Let‘s have a look on Gaussian Process Regression (GPR) mathematics

- Gaussian Distribution $N(\mu, \Sigma)$ ←→ Gaussian Process $GP(m(x), k(x, x'))$
- k is covariance function that is represented by a so called kernel function

$$k_{SE}(x, x') = e^{-1/2|x-x'|^2}$$

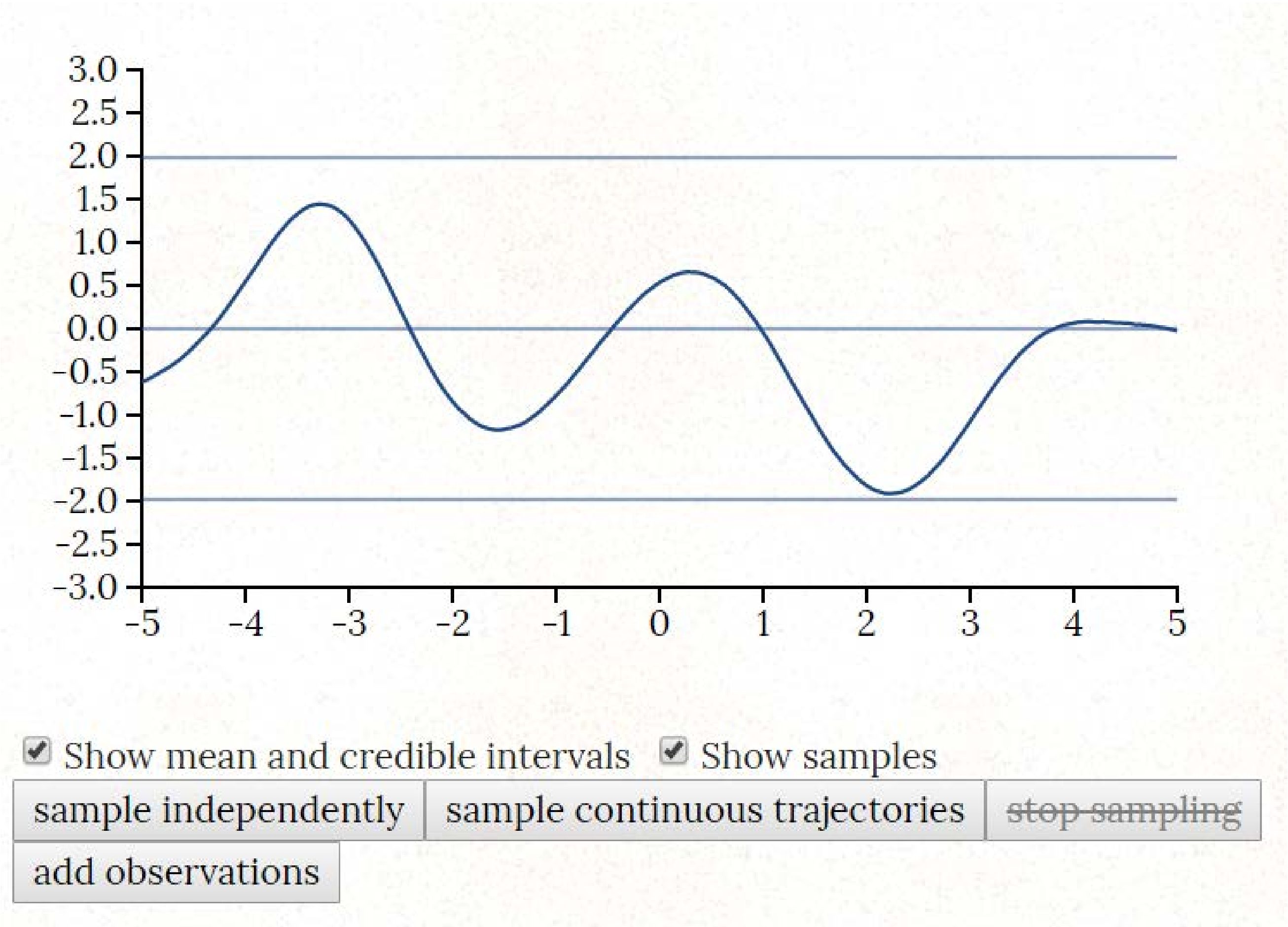
➔Gaussian Process is a random distribution of functions

➔Gaussian Process is non-parametric: inputs are compared to training data

$$f(x) \coloneqq \text{observation} \qquad f_*(x_*) \coloneqq \text{unknown}$$

$$\begin{pmatrix} f \\ f_* \end{pmatrix} \sim N\left(0, \begin{pmatrix} k(x, x) & k(x, x_*) \\ k(x_*, x) & k(x_*, x_*) \end{pmatrix}\right)$$

Let's see a demonstration of GPR fitting process

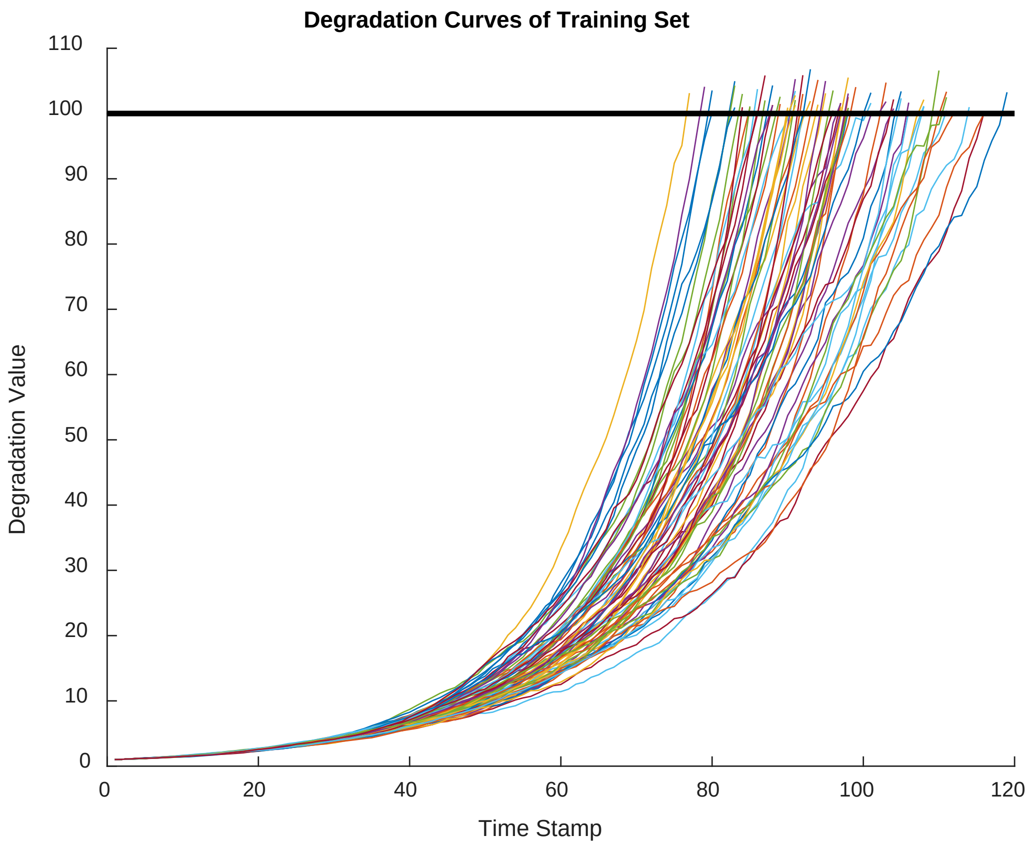


http://www.tmpl.fi/gp/

The GPR is applied to a simple degradation prognosis

- Random exponential degradation curves with noise
- Fit a gaussian process regression that has as input:
  - Current degradation: $x(t_k)$
  - Last degradation: $x(t_{k-1})$
  - Second last degradation: $x(t_{k-2})$

  and outputs:
  - Next degradation: $x(t_{k+1})$



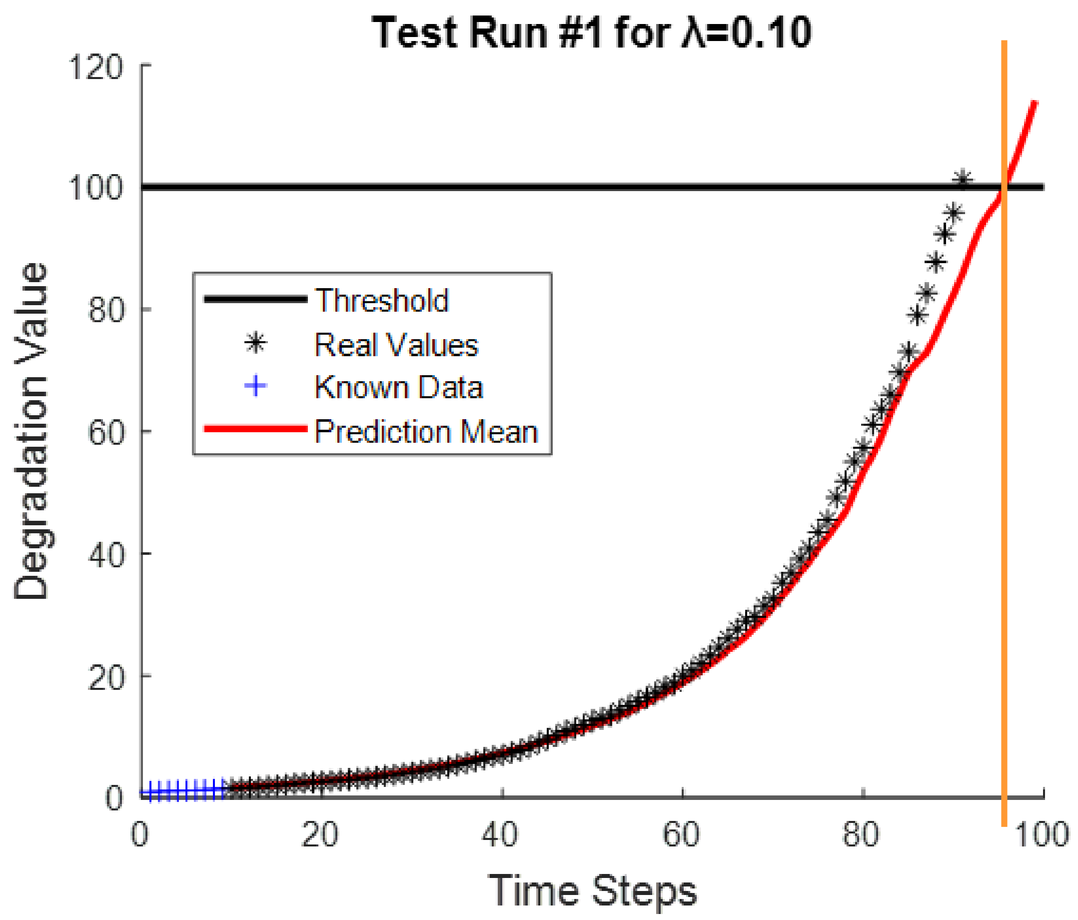
In MATALB you need only one command to train a GPR

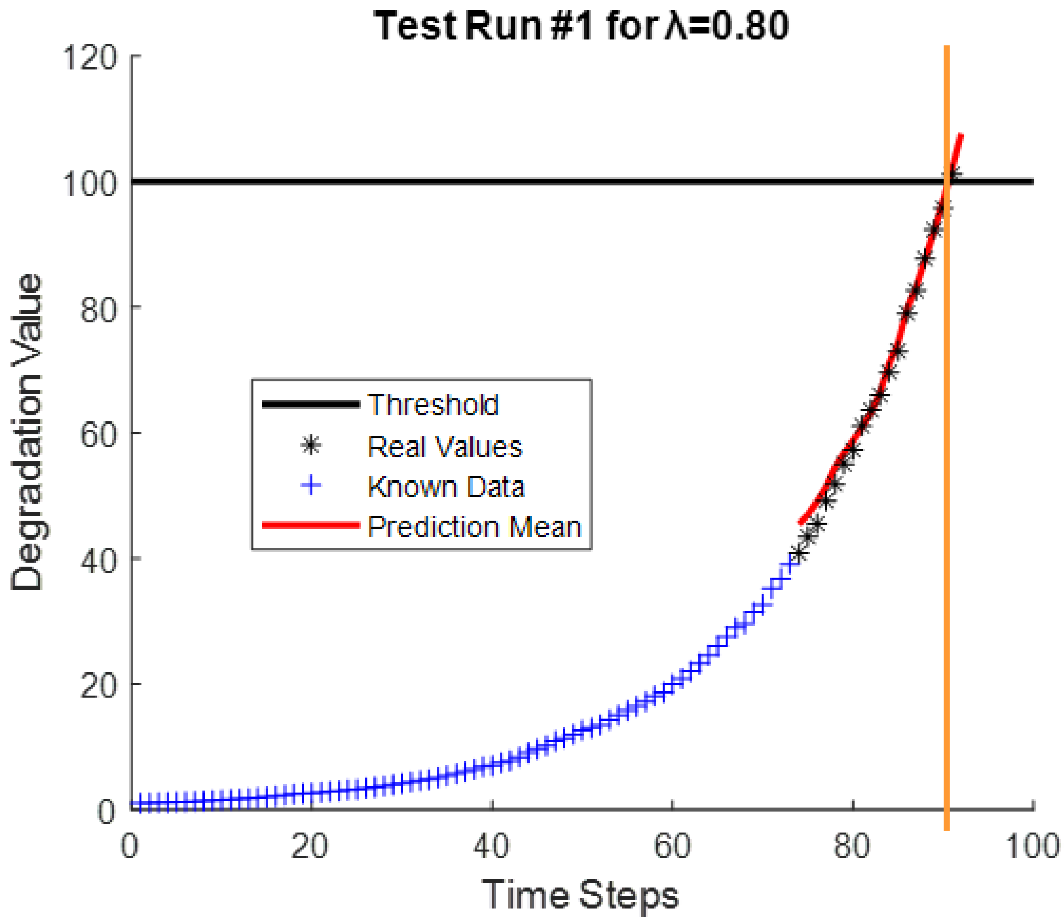
➔ Let‘s have a look into MATLAB

```
fitrgp(table2array(SimData.train.gprInputTable),SimData.train.gprOutputTable.nextDeg,...
                           'FitMethod','sd',...
                           'PredictMethod','exact',...
                           'KernelFunction',conf.gpr.KernelFunction,...
                           'BasisFunction',conf.gpr.BasisFunction,...
                           'Verbose',conf.gpr.verbose);
```

Exemplary results of an iterative GPR prediction until threshold is reached

TECHNISCHE UNIVERSITÄT DARMSTADT





UNCERTAINTY IN PREDICTION

A look into the future never gives a certain answer

- Phenomena of any prognosis like weather forecasts, stock forecasts, etc.
- Input uncertainty
  - Material properties → the reason to test more than one component
  - Initial or boundary conditions → the environment has an influence
  - Sensor uncertainty → the reason for sensor calibration
- Discretization uncertainty
  - Time steps (sample rate) → real world is continuous – information between samples is lost
  - Floating-point number precision → conversion of analog values to discretized values
- Model uncertainty
  - Representation of the real world problem → algorithm output vs. real world output

Each prognosis needs to be expressed as a probability

TECHNISCHE UNIVERSITÄT DARMSTADT

- Each prognosed degradation at time t_k is a probability distribution for $x(t_k)$
- The probability of failure is expressed as a cumulative distribution function (CDF)

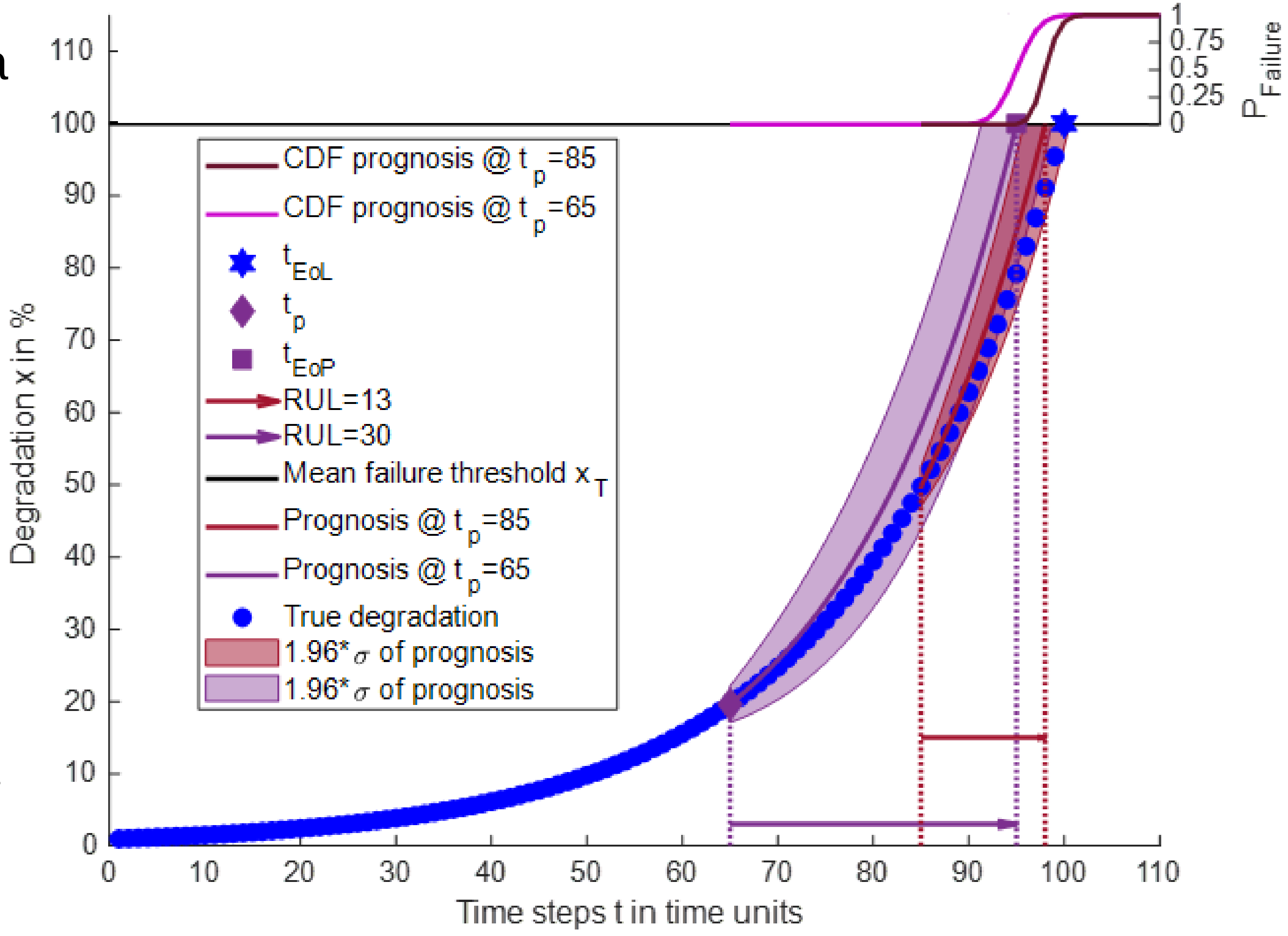
→ Reaching the threshold at time t_k is given with a probability

→ How to calculate the RUL?

  → Mean of CDF reflects the expectation value of discrete distribution

  → Median of CDF reflects a probability of 50 % that a component will have failed until that time

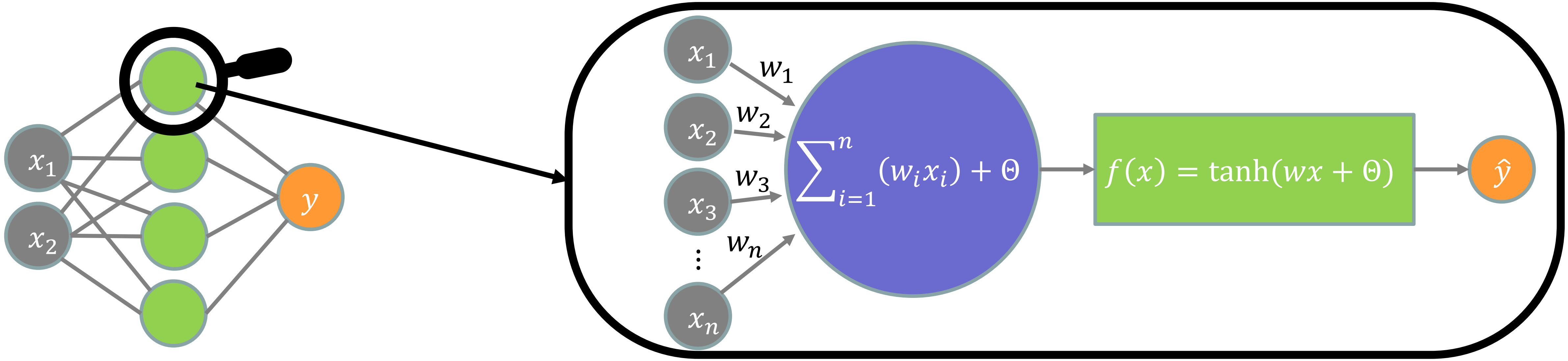
  → Specify a distinct probability value for the CDF



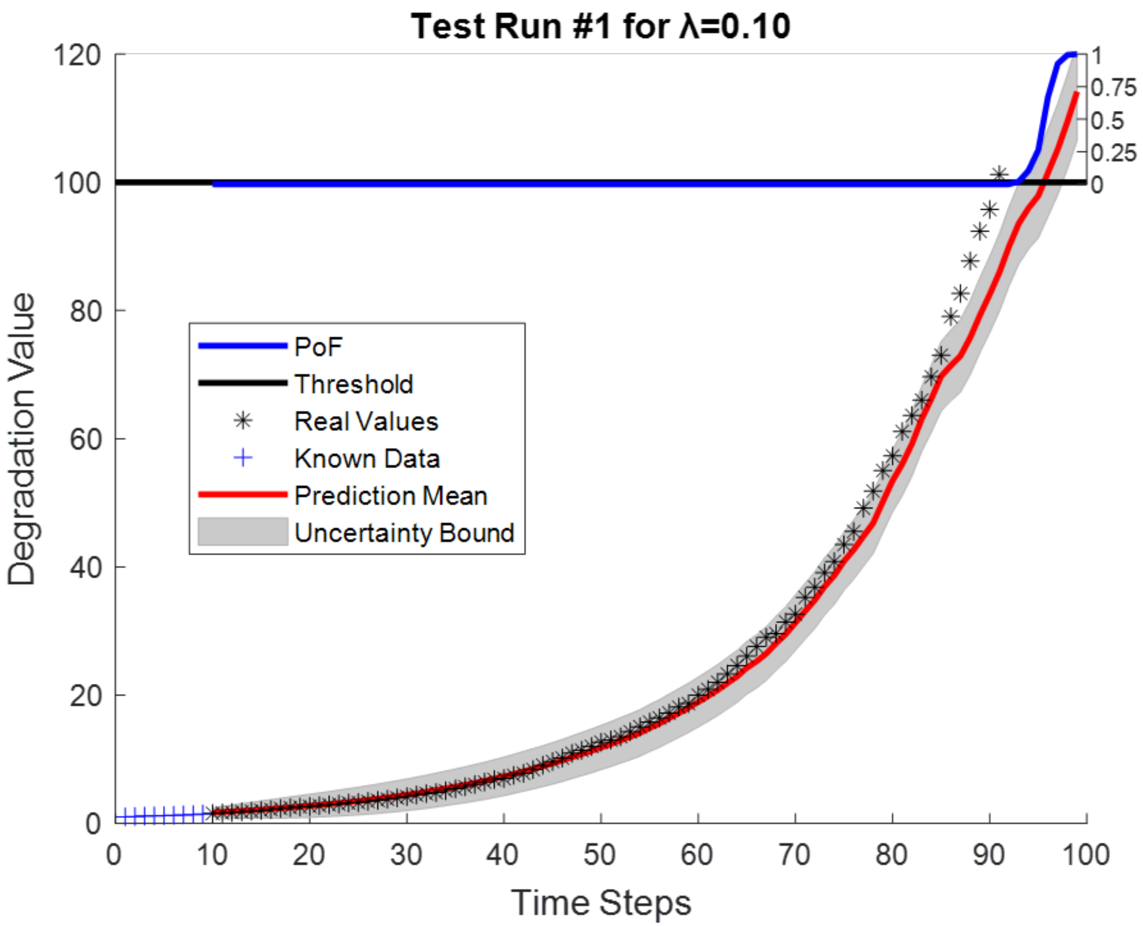
GPR inherently outputs an uncertainty boundary, but not all methods do

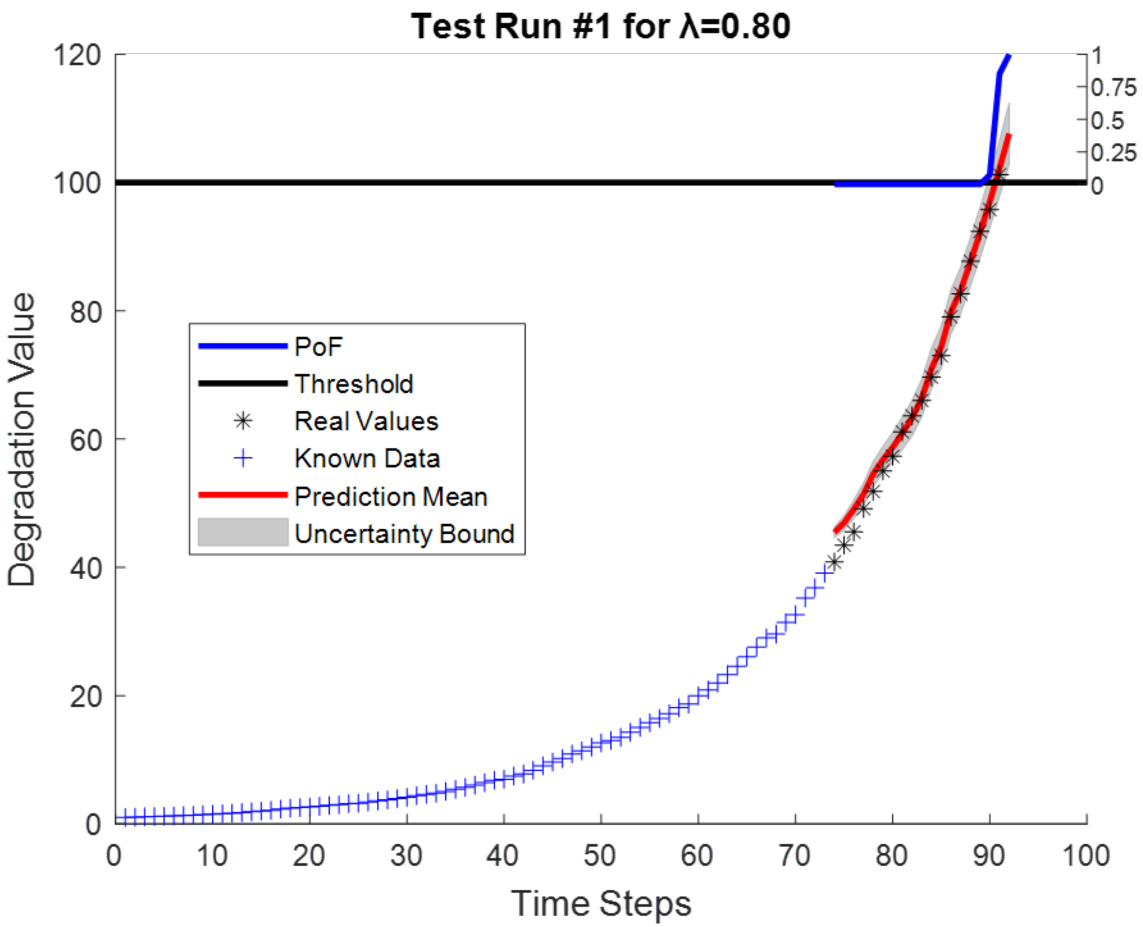
- Covariance function delivers uncertainty $\begin{pmatrix} f \\ f_* \end{pmatrix} \sim N\left(0, \begin{pmatrix} k(x,x) & k(x,x_*) \\ k(x_*,x) & k(x_*,x_*) \end{pmatrix}\right)$
- Not all methods output uncertainty

→ Use additional methods: Bootstrapping, Particle Filter, Monte Carlo,…



The same degradation curves as previously but with uncertainty

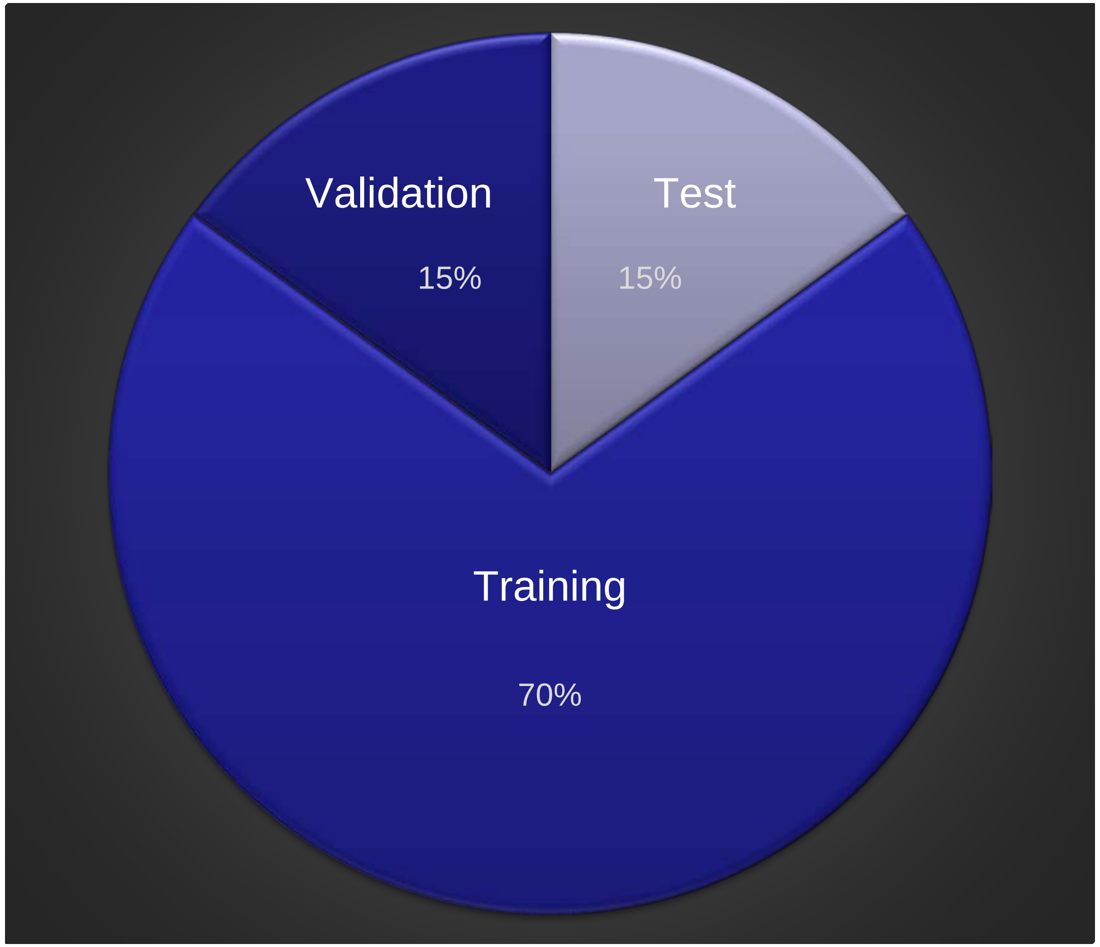




HOW TO EVALUATE THE PROGNOSTIC PERFORMANCE

The dataset is split into training, validation and testing

- **Training:** the data that is used to train an algorithm (e.g. Neural Network)
- **Validation:** the data that is used to optimize the parameters
- **Test:** the data that is used to test the trained model – never seen by the algorithm before



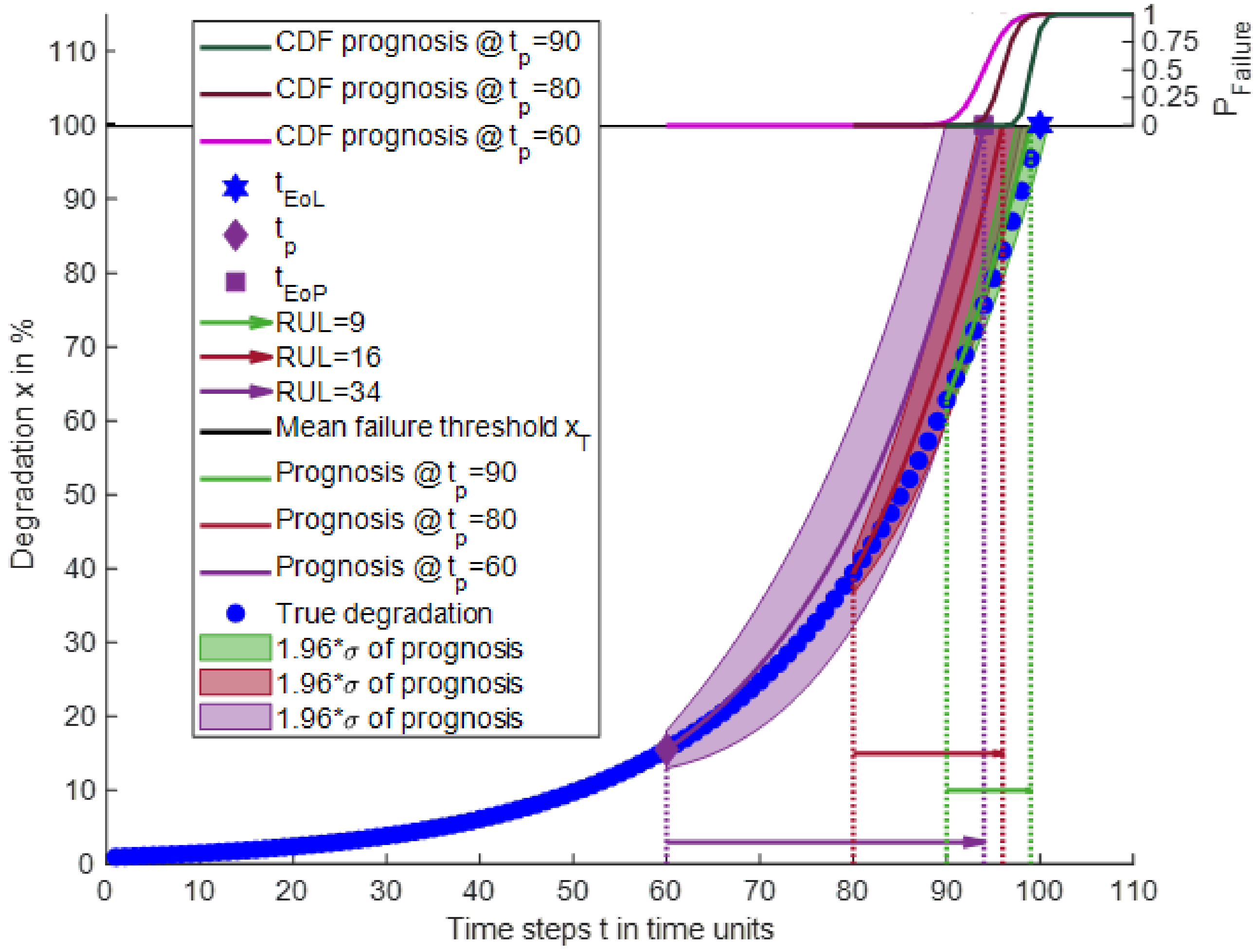
Test data is used to predict from different time steps

- For simplification and comparison reason normalized time steps are used

$$\lambda = {t_P}/{t_{EoL}}$$

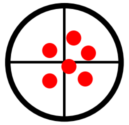
- Normalization equalizes the different lifetimes
- Typical values are

$$\lambda = 0.1, 0.2, \dots, 0.9$$



Different performance metrics exist to compare prognostic's accuracy and precision

TECHNISCHE UNIVERSITÄT DARMSTADT



Accuracy based metrics

- **Error** $\Delta(i) = r_*(i) - r(i)$

  $r(i)$: RUL estimate at time t_i

  $r_*(i)$: True RUL at time t_i

  → represents the deviation

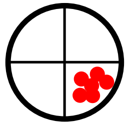
- **Mean absolute percentage error**

  $$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{100\Delta(i)}{r_*(i)}\right|$$

  → unit free metric

  → weighs small errors differently to big errors

  → heavier penalty on forecasts that exceed the actual than on those that are less



Precision based metrics

- **Sample standard deviation**

  $$SSD = \sqrt{\frac{\sum_{i=1}^{N}(\Delta(i)-\mu_\Delta)}{N-1}} \quad \text{with } \mu_\Delta\text{: mean of errors}$$

  → measures dispersion/spread of the error

  → normal distribution is assumed

- **Mean absolute deviation**

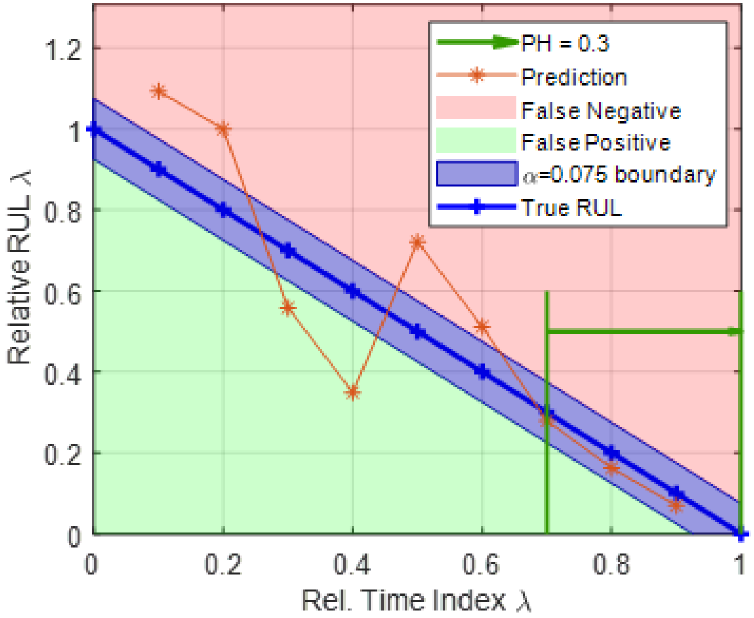
  $$MAD = \frac{1}{N}\sum_{i=1}^{N}\left|\Delta(i) - \text{median}\big(\Delta(i)\big)\right|$$

  → estimator of dispersion/spread of the error

  → can be used for small number of data

Different performance metrics exist to compare prognosis results

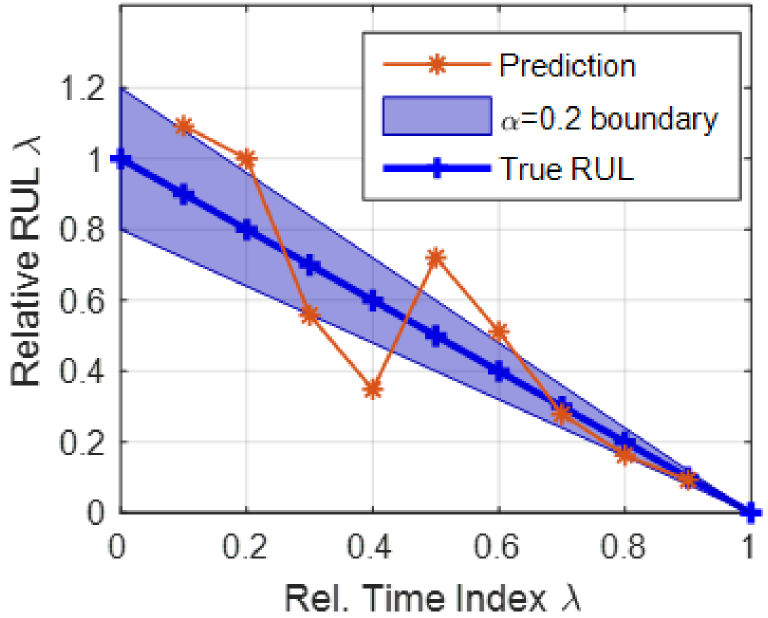
Prognostic Horizon



→ $1 - \lambda$ when the predictions stay inside boundary

- **False Positive:** unacceptable early predictions
- **False Negative:** unacceptable late predictions

$\alpha - \lambda$ Accuracy



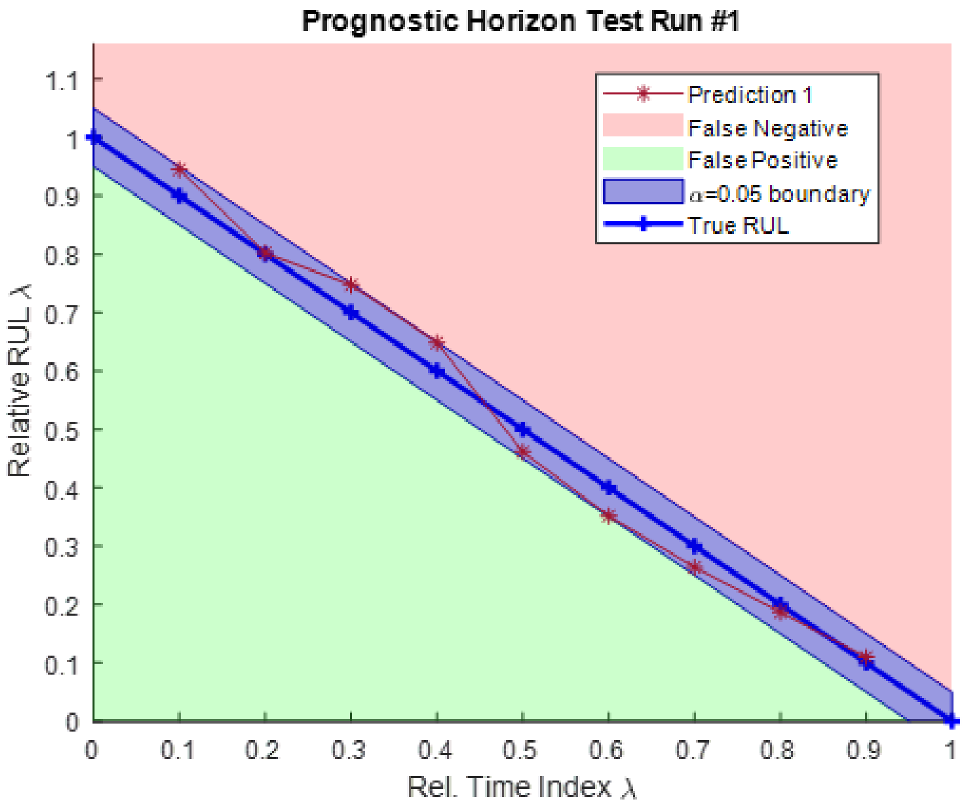
→ Similar to Prognostic Horizon metric

→ Cone shape penalizes big errors for later predictions

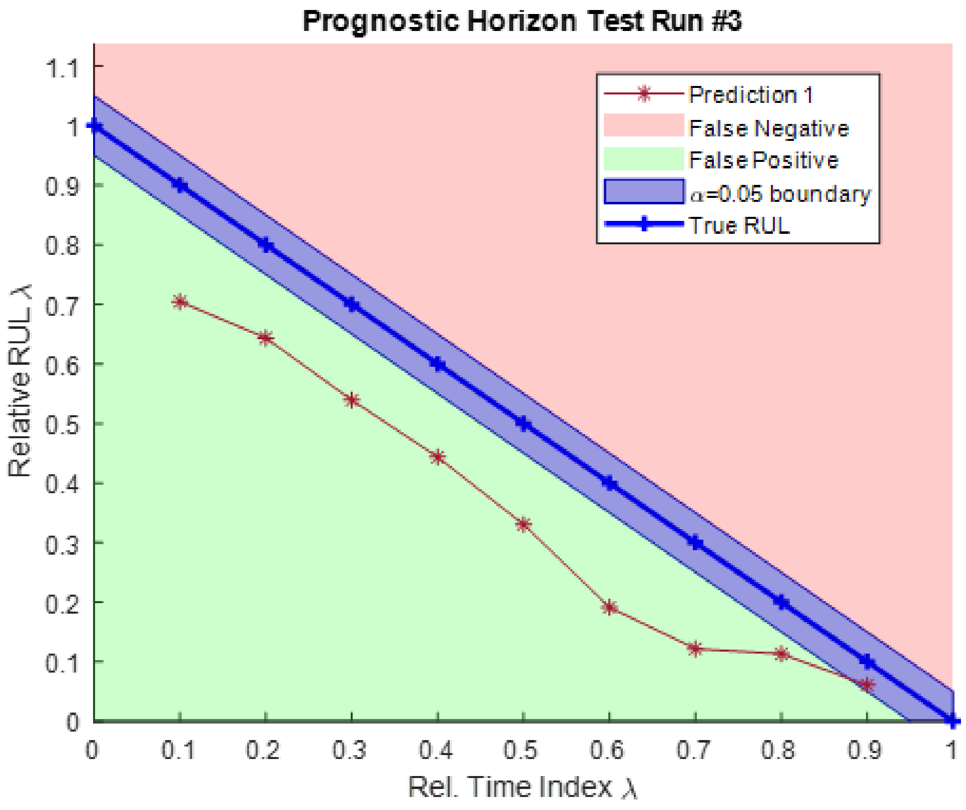
→ Cone shape considers convergence

The performance of predictions can vary between test runs

TECHNISCHE UNIVERSITÄT DARMSTADT



- Accurate prediction (PH = 0.5)
- Good result



- Prediction is inaccurate (PH = 0.1)
- Bad result
- **Safe result** (False Positive)

The function *predict* in MATLAB performs a prediction step

TECHNISCHE UNIVERSITÄT DARMSTADT

➔ Let‘s have a look into MATLAB

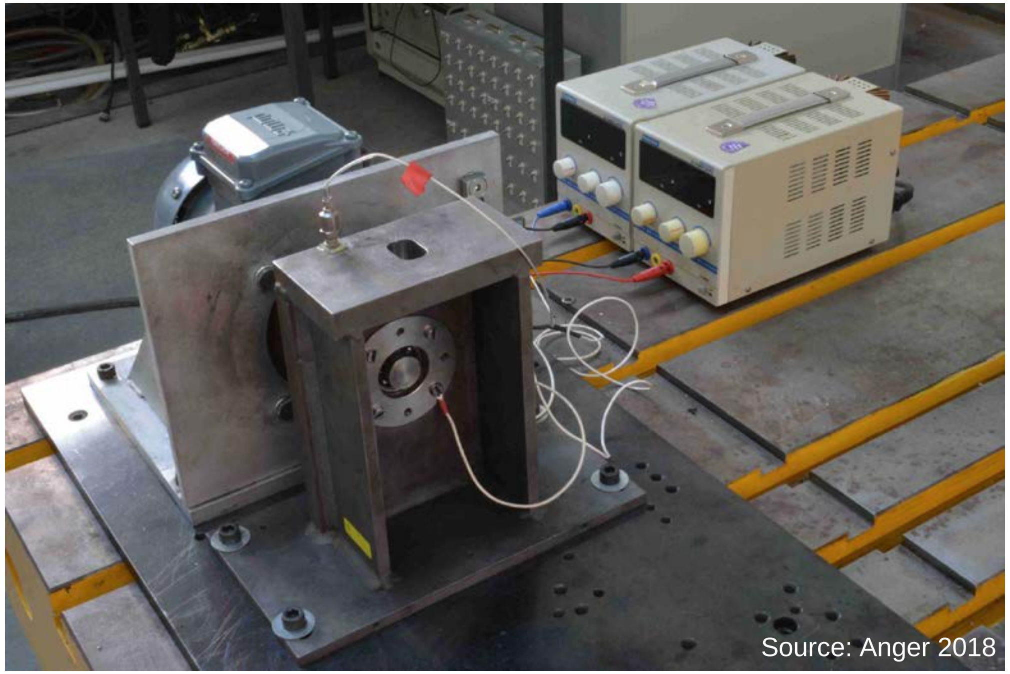
```
%% do the prediction
[y_predict(end+1,:), y_predictStdCurr] = predict(gprMdl.gprMdl,x_predict(end,:));
%% add up the variance of the prognosis
if isempty(y_predictStd)
        y_predictStd(1,:) = sqrt(y_predictStdCurr.^2 + conf.prog.baseStd.^2);
else
        y_predictStd(end+1,:) = sqrt(y_predictStd(end,:).^2 + y_predictStdCurr.^2);
end
```

CHALLENGES IN REAL WORLD APPLICATION

Prediction in laboratory environment is easier than in real world application

Laboratory environment

- Run-to-failure → accelerated degradation
- Controlled environment
- Different algorithms developed and approved



Source: Anger 2018

Real environment

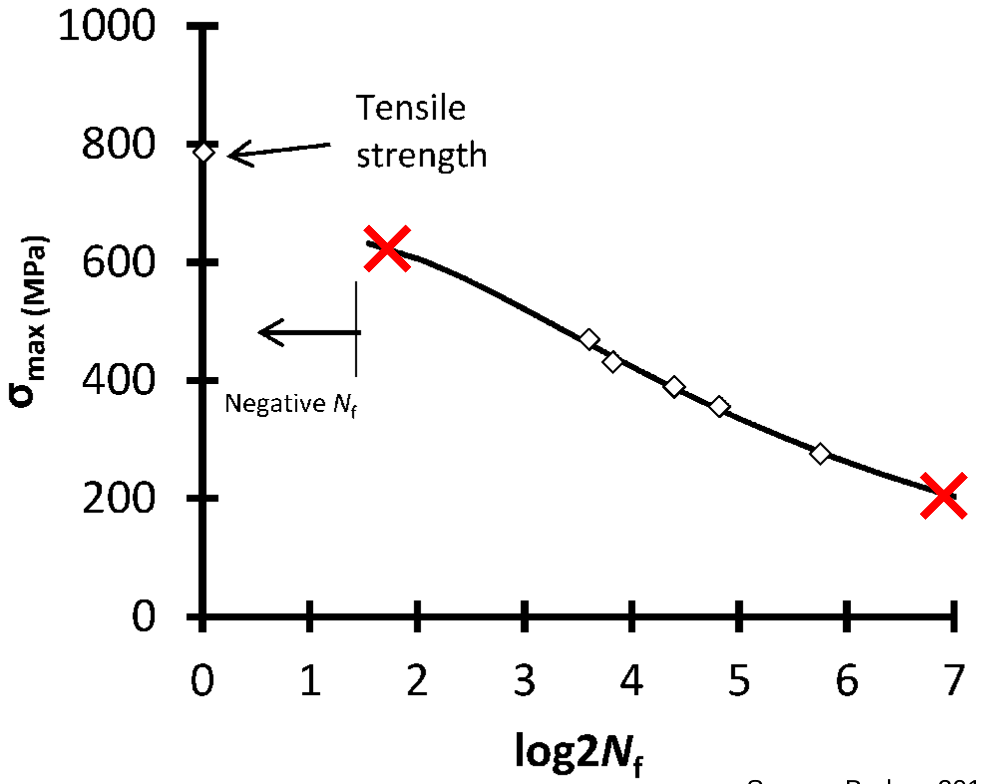
- No run-to-failure → limited data
- Environmental influences (vibration, weather,…)
- Algorithm transfer from labor is challenging

Source: EnBW

The higher the load, the higher the wear and vice versa

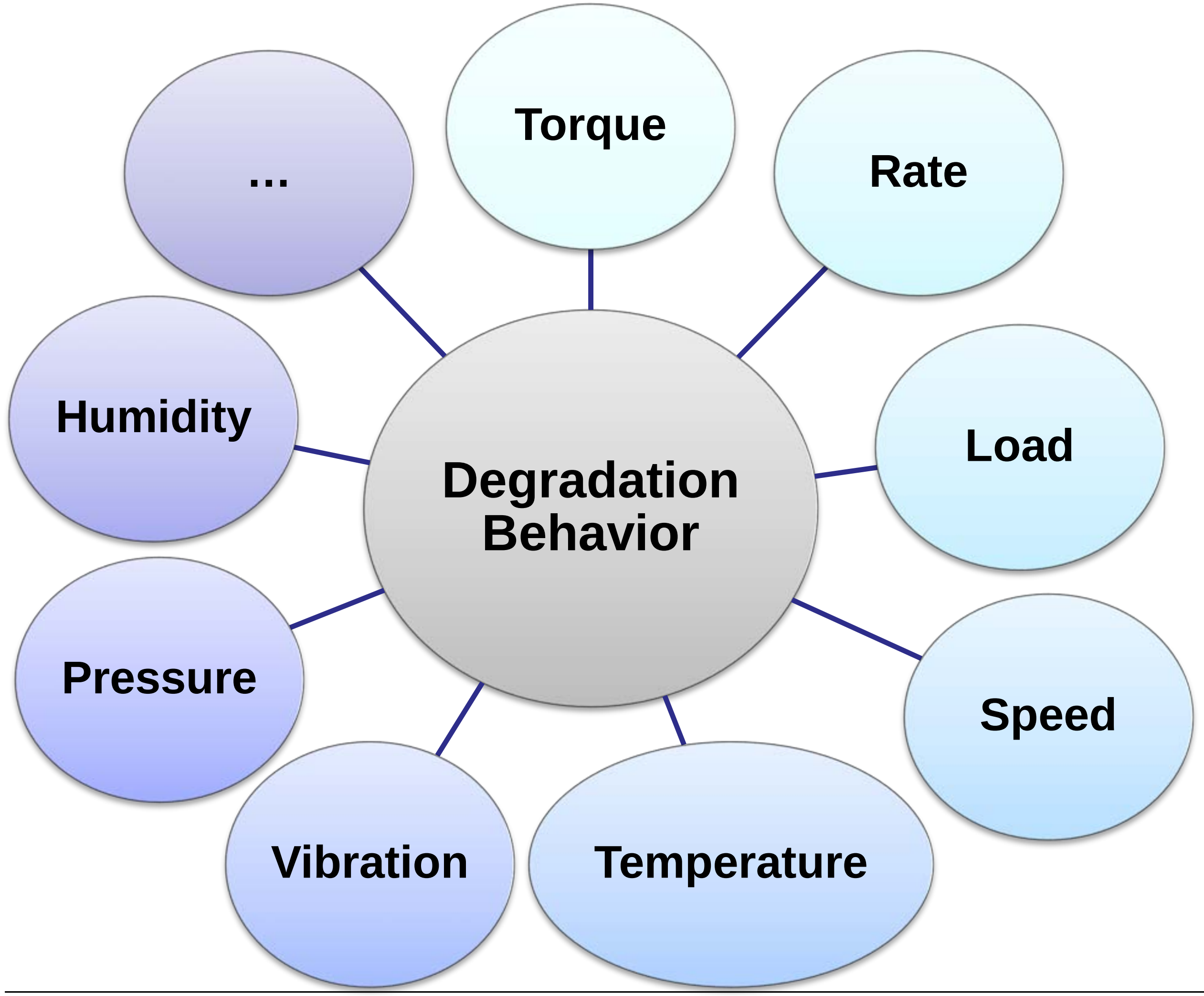
- Applied load determines the maximum number of cycles
- Well known from typical robustness models
  - Wöhler curve
  - Miner‘s rule
  - …

Wöhler-Curve



Source: Burhan 2018

There is a wide range of parameters that can effect degradation



- Parameter‘s relevance depends on usage scenario
- In case influence of parameter variation should be considered, additional measurement is required
- Planned conditions need to be known for a degradation prediction

What to take with you?

LEARNING OUTCOMES

Key Findings

- Prognosis only makes sense for components that show degradation behavior
- Creating a Health Indicator can be challenging
- RUL is typically interpreted as a time
- Different approaches are available to predict the RUL / degradation
- Consideration of the uncertainty is essential
- There are common metrics that can be used to evaluate the prognosis performance
- Application of prognosis algorithms in real world environment is challenging

References

- ISO-13374-1 Condition monitoring and diagnostics of machines — Data processing, communication and presentation — Part 1: General guidelines
- http://www.mimosa.org/mimosa-osa-cbm/
- DIN EN 13306 – Instandhaltung – Begriffe der Instandhaltung
- DIN 31051 – Grundlagen der Instandhaltung
- Carl Edward Rasmussen and Christopher K. I. Williams, The MIT Press, 2006 (http://www.gaussianprocess.org/gpml/)
- Dewey, H. Heath and DeVries, Derek R. and Hyde, Scott R.: Uncertainty Quantification in Prognostic Health Management Systems; 2019 IEEE Aerospace Conference
- A. Saxena and J. Celaya and E. Balaban and K. Goebel and B. Saha and S. Saha and M. Schwabacher: Metrics for evaluating performance of prognostic techniques; 2008 International Conference on Prognostics and Health Management
- A. Saxena and J. Celaya and B. Saha and S. Saha and K. Goebel: Evaluating algorithm performance metrics tailored for prognostics; 2009 IEEE Aerospace conference
- Christoph Anger: Hidden semi-Markov Models for Predictive Maintenance of Rotating Elements; PhdThesis 2018; TU Darmstadt
- I. Burhan and H. S. Kim, "S-N Curve Models for Composite Materials Characterisation: An Evaluative Review," Journal of Composites Science, vol. 2, no. 3, 2018.
- S. Uckun, K. Goebel, and P. J. Lucas, "Standardizing research methods for prognostics," International Conference on Prognostics and Health Management, 2008.


THANK U