

Introduction to Prognostics

Annual Conference of the PHM Society 2019

Matteo Corbetta

SGT Inc., NASA Ames Research Center



September 24, 2019 Scottsdale, AZ



Acknowledgments:

- ▶ Dr. Kai Goebel and the PHM Society
- ▶ Previous tutorial presenters
- ▶ SGT Inc., Diagnostics & Prognostics Group, NASA Ames
- ▶ Prof. Yongming Liu and his team for the crack growth dataset



Topics of the tutorial

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)

Your feelings during this tutorial if:

you know (some)
PHM

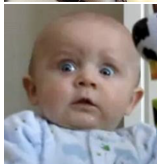
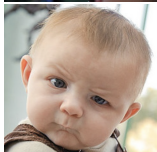
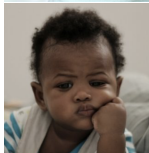
Uh? PHM?
prediction?

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)



Today's material

Download this presentation and tutorial code at:

phmsociety.org/events/conference/phm/19/tutorials

Scripts and dataset:

[*particleFilterPrediction.py*](#)

[*gpRegression.py*](#)

[*CO2data.txt*](#)

Libraries we'll use:

numpy

scipy

matplotlib

Instructions to install Python and libraries:

[*README.txt*](#)

What is prognostics?

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)



Definition

Prognostics is an engineering discipline focused on predicting the time at which a system or a component will no longer perform its intended function*

*Vachtsevanos GJ, Lewis F, Hess A, Wu B. Intelligent fault diagnosis and prognosis for engineering systems. Hoboken: Wiley; 2006 Sep.

Approaches to prognostics

Thanks to Prof. J. W. Hines, PHM Tutorial 2009

Type I: Reliability-based

$\lambda = \lambda(t)$, MTTF, MTBF, ...

Approaches to prognostics

Thanks to Prof. J. W. Hines, PHM Tutorial 2009

Type I: Reliability-based

$\lambda = \lambda(t)$, MTTF, MTBF, ...

Type II: Stress-based

E.g., proportional hazard models

$\lambda = \lambda(t, z)$, where z are "stressors"

Approaches to prognostics

Thanks to Prof. J. W. Hines, PHM Tutorial 2009

Type I: Reliability-based

$\lambda = \lambda(t)$, MTTF, MTBF, ...

Type II: Stress-based

E.g., proportional hazard models

$\lambda = \lambda(t, z)$, where z are "stressors"

Type III: Condition-based ← **what we'll see today**

Modeling individual failure mechanisms, cumulative damage models, state extrapolation, ...

Why prognostics?

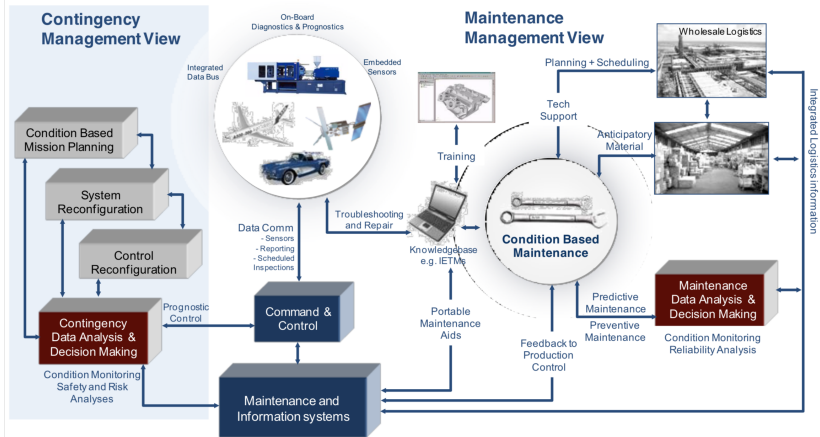
What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)

Why prognostics?



Thanks to: Dr. Abhinav Saxena

Schematic adapted from: A. Saxena, Knowledge-Based Architecture for Integrated Condition Based Maintenance of Engineering Systems, PhD Thesis, 2007.

Why prognostics?

Safety

prevent unexpected failures
minimize impact on other systems
be prepared to initiate contingency plans

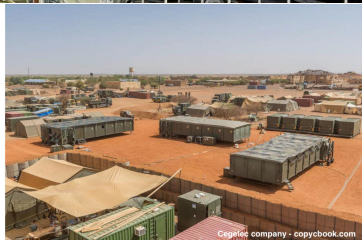


Why prognostics?



Logistics

reduce spare parts stock
logistics footprint



Why prognostics?

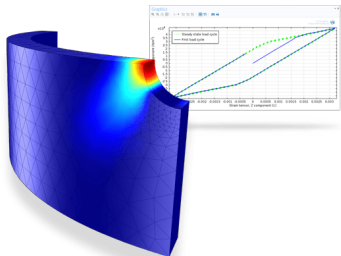


Maintenance

reduce unnecessary interventions
"Just-in-time" approach
optimize fleet maintenance



Why prognostics?



Reliability & Performance

product reputation
reduced safety factors

Why prognostics?

Safety

- prevent unexpected failures
- minimize impact onto other systems
- implement contingency plans

Logistics

- reduce spare parts stock
- logistics footprint

Maintenance

- reduce unnecessary interventions
- "Just-in-time" approach
- optimize fleet maintenance

Reliability & Performance

- product reputation
- reduced safety factors

Thanks to: Dr. N. Scott Clements. Please refer to his tutorial [PHM Tutorial 2011](#) for more information on industrial applications

Prognostic process

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)

What we are trying to predict

Future behavior

Calculate the future values of the quantities of interest to infer future behavior of the system

End-of-life

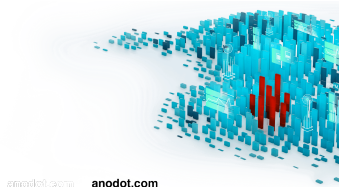
Calculate the time-to-failure or the remaining useful life (RUL) of a component/system, which current condition is known with certain confidence.

Steps of the prognostic process

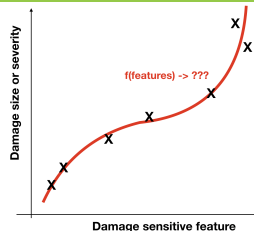
1. Anomaly / fault detection



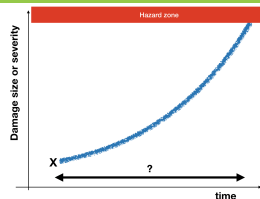
2. Identification and isolation



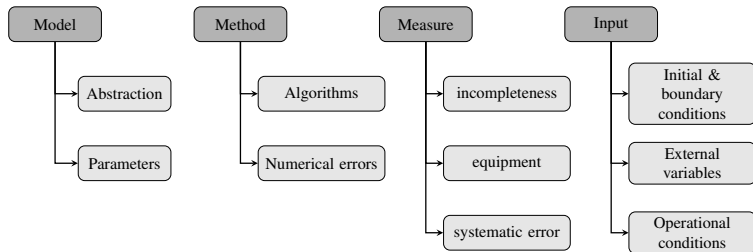
3. Quantification



4. Prognosis



Source of information (and uncertainty)



- ▶ Ground truth measures are hard to come by, and sometimes there's no term for comparison (e.g., Golden Gate bridge)
- ▶ Many times measures are noisy, corrupted by systematic errors or faulty measurement systems
- ▶ Health-related quantities are typically hard to measure (i.e., measures are intrusive or destructive)
- ▶ Some times it's simply not possible (physically or economically) to measure some variables

- ▶ Models are mere representations of reality
- ▶ Models do not (typically) accommodate all physical phenomena affecting the system. If they do, they may not be suitable for real-time applications
- ▶ They always require calibration and validation.
- ▶ They may require correction terms to be updated in real-time (every time) or to be tuned on a case-basis

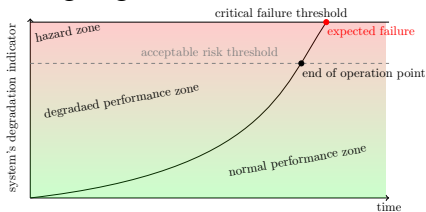
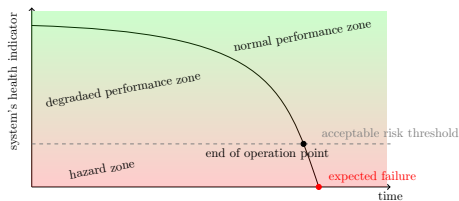
Environmental & operational conditions (input)

- ▶ Varying environmental conditions can drastically change algorithm performance (or even make algorithms useless)
- ▶ Many damage-sensitive features are also affected by operational profiles (e.g., vibrations in a wind turbine generator change with produced power)
- ▶ Environmental variables May be unknown, hard to measure or their future values hard to predict (i.e., wind speed and direction in urban environments)
- ▶ Finding causal relationships: dependencies from external factors are hard to quantify

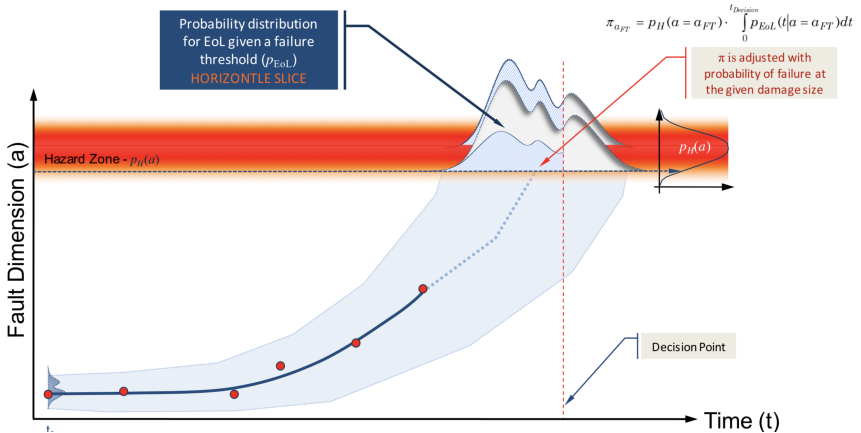
- ▶ Rounding errors or machine precision may not be negligible for the problem we're looking at
- ▶ If the algorithm goal is minimization or filtering, they may get stuck into a local minima (e.g., the results change at different runs)
- ▶ They often need tuning of parameters, or in case of data-driven methods, their performance depends on the amount of training data
- ▶ Convergence not always guaranteed

Prognosis in a cartoon

Tracking health vs. tracking degradation



Prognosis in a cartoon



Thanks to: Dr. Abhinav Saxena, GE

See his prognostics tutorial from Annual PHM Conference 2010 [here](#).

Examples (with codes)

What is prognostics?

Why prognostics?

Prognostic process

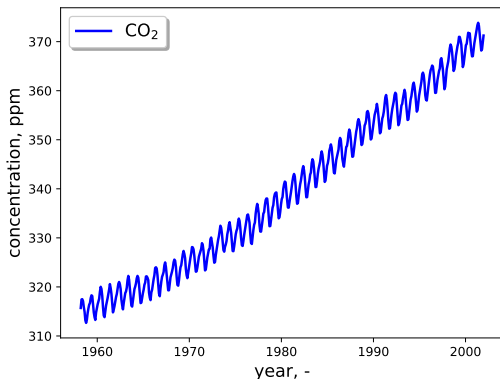
Examples (with codes)

Example 1

Data-driven CO₂ concentration prediction

CO₂ concentration prediction

Monthly average atmospheric CO₂ concentrations (in parts per million by volume, ppmv) collected at the Mauna Loa Observatory in Hawaii between 1958-2001¹.



What will the CO₂ concentration be after 2001?

¹ credits for the idea to Rasmussen and Williams, Gaussian Processes for Machine Learning, and [Sci-kit learn](#), and NOAA for the dataset.

We use Gaussian Processes (GP) to predict the concentration over the years after 2001.

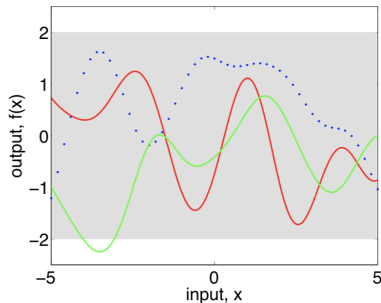
The process $f(\mathbf{x})$ is a GP if can be specified by a mean and covariance function:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

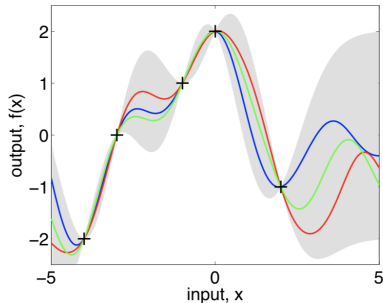
The covariance function $k(\mathbf{x}, \mathbf{x}')$ is the key containing info about time-correlations and dispersion.

Once we learn the covariance function, we can perform predictions far from training points.

GP - prior vs posterior



(a), prior



(b), posterior

Example from Rasmussen and Williams, *Gaussian Processes for Machine*

Learning, 2006.

The covariance function k is the key: To fit the CO₂ time series, we build k as a sum of elementary covariance functions:

$$k_1(x, x') = \theta_1^2 \exp\left(-\frac{1}{2} \frac{(x-x')^2}{\theta_2^2}\right) \quad \text{long-term rising trend}$$

$$k_2(x, x') = \theta_3^2 \exp\left(-\frac{(x-x')^2}{2\theta_4^2} - \frac{2 \sin^2(\pi(x-x'))}{\theta_5^2}\right) \quad \text{periodicity}$$

$$k_3(x, x') = \theta_6^2 \left(1 + \frac{(x-x')^2}{2\theta_8\theta_7^2}\right)^{-\theta_8} \quad \text{medium term irregularities}$$

$$k_4(x, x') = \theta_9^2 \exp\left(-\frac{(x_p - x_q)^2}{2\theta_{10}^2}\right) + \theta_{11}^2 \delta_{p,q} \quad \text{noise}$$

$$k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + k_4(x, x')$$

$$\theta = [\theta_1, \theta_2, \dots, \theta_{11}]$$

Find hyper-parameter vector θ

Find the hyper parameters θ that best fit the training data. We do so by maximizing the marginal likelihood $p(\mathbf{y}|X)$ in log-form:

$$\log p(\mathbf{y}|X) = -\frac{1}{2}\mathbf{y}^T (\textcolor{red}{k} + \textcolor{blue}{\sigma_n^2}I)^{-1} \mathbf{y} - \frac{1}{2} \log |\textcolor{red}{k} + \textcolor{blue}{\sigma_n^2}I| - \frac{n}{2} \log 2\pi$$

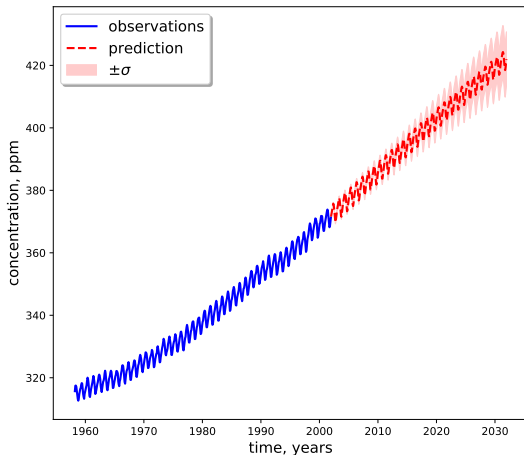
covariance function
model error (noise)

See *Rasmussen & Williams, GP for ML, 2006*.

open *gpRegression.py*. make sure *CO2data.txt* is in the same folder.

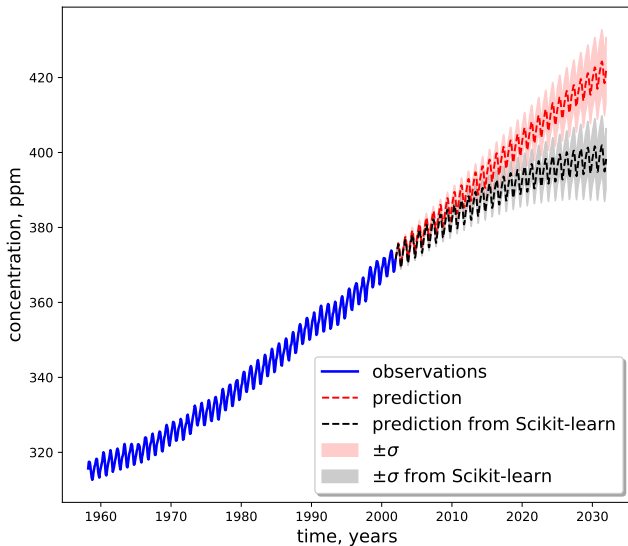
Prediction

Using (sub-)optimal parameters found via differential evolution algorithm².



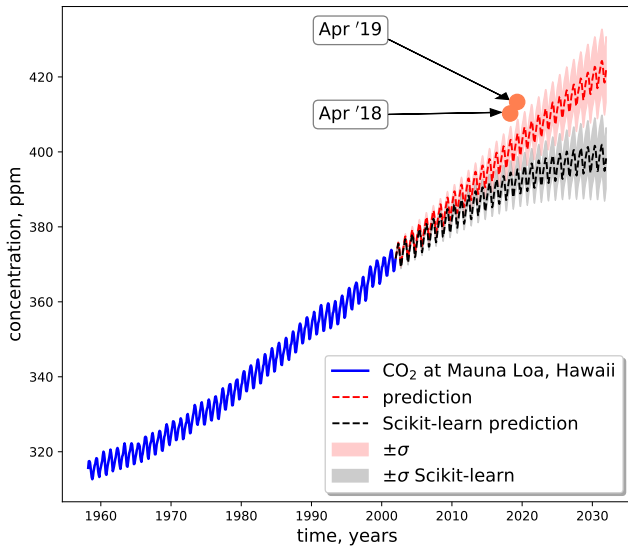
²DE should converge towards *the* optimal parameters. For this problem, different runs produced different results, suggesting either that the population size or number of iterations was too small.

Prediction



Prediction

What's the concentration today?



A few things to remember

What about model validation???

Here's some options you should try:

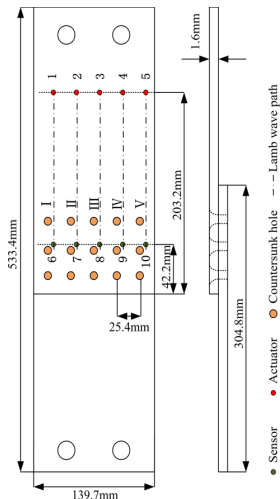
- ▶ Split dataset into training and validation
- ▶ Cross-validation with batches, leave-one-out, etc.
- ▶ Gather more data
- ▶ Try adding/removing different covariance functions

Example 2

Fatigue crack growth prognosis using particle filter

Fatigue crack growth prognosis

Data from 2019 PHM data challenge:

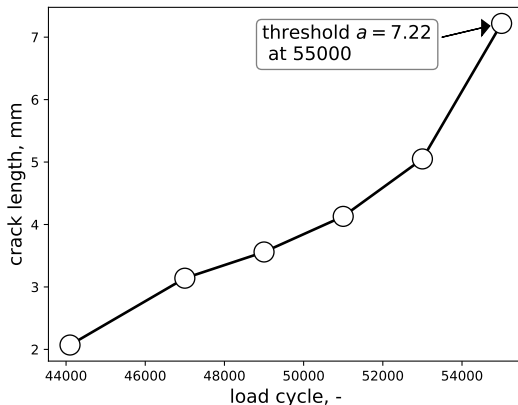


- ▶ fatigue crack growth at rivet holes
- ▶ tensile, constant amplitude fatigue loading

Thanks to: Prof. Yongming Liu and his team, ASU

Visit the 2019 PHM Data challenge website for more information.

Fatigue crack growth prognosis



Given the set of sequential measures of crack length, **can we predict the number of cycles to reach final length $a = 7.22$ mm (i.e., 55,000 load cycle)?**

Chapman-Kolmogorov and Bayesian updating

$$p(\mathbf{X}_k | \mathbf{Y}_{k-1}) = \int_{-\infty}^{\infty} p(\mathbf{X}_k | \mathbf{X}_{k-1}) p(\mathbf{X}_{k-1} | \mathbf{Y}_{k-1}) d\mathbf{X}_{k-1}$$

$$p(\mathbf{X}_k | \mathbf{Y}_k) = \frac{p(\mathbf{X}_k | \mathbf{Y}_{k-1}) p(\mathbf{Y}_k | \mathbf{X}_k)}{p(\mathbf{Y}_k | \mathbf{Y}_{k-1})}$$

Far-ahead prediction stage:

$$p(\mathbf{X}_{k+l} | \mathbf{Y}_k) = \int_{\mathcal{X}} p(\mathbf{X}_k | \mathbf{Y}_k) \left[\prod_{j=k+1}^{k+l} p(\mathbf{X}_j | \mathbf{X}_{j-1}) \right] d\mathbf{X}_{k:k+l-1}$$

Particle filtering pseudo-code

Input: $\mathbf{x}_{k-1}^{(i)}, \forall i = 1, \dots, N_s$, and y_k

Output: $p(\mathbf{X}_k | Y_k), p(\text{RUL}_k | Y_k)$

1. Approximate posterior pdf

$\mathbf{x}_k^{(i)} \sim p(\mathbf{X}_k | \mathbf{x}_{k-1}^{(i)}) \leftarrow$ propagate samples with model function

$\ell(y_k | \mathbf{x}_k^{(i)}) \leftarrow$ compute likelihood for all samples

$w_k^{(i)} \propto w_{k-1}^{(i)} \ell(z_k | \mathbf{x}_k^{(i)}) \leftarrow$ assign weights

$p(\mathbf{X}_k | Y_k) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta_{\mathbf{x}_k, \mathbf{x}_k^{(i)}} \leftarrow$ approx. posterior pdf

2. Systematic re-sampling

$\mathbf{x}_k^{(j)} \sim p(\mathbf{X}_k | Y_k) : \Pr\{\mathbf{x}_k^{(j)} = \mathbf{x}_k^{(i)}\} = w_k^{(i)}$

$w_k^{(j)} = 1/N_s \quad \forall j = 1, \dots, N_s$

3. Prognosis

for $i = 1, 2, \dots, N_s$ **do**

$l = 0$

while $\mathbf{x}_k^{(i)} \in \text{safe domain}$ **do**

$\mathbf{x}_k^{(i)} \sim p(\mathbf{X}_{k+l} | \mathbf{x}_{k+l-1}^{(i)})$

$l += 1$

end

$t_f = t_{k+l} \leftarrow$ extract time at which sample i reached threshold \mathbf{x}_{th}

$\text{RUL}_k^{(i)} = t_f - t_k \leftarrow$ extract remaining useful life for sample i

end

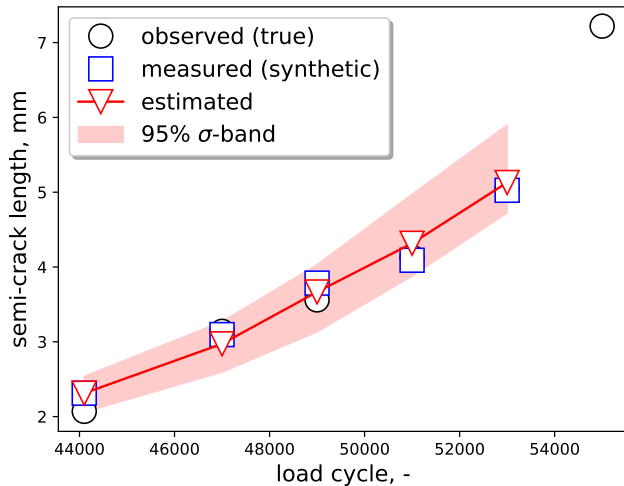
$\mathbf{x} = [a, \log C, m]^T$	augmented state vector
$\mathbf{z} \rightarrow z = a + \epsilon_g$	unbiased, noisy measures
$\mathbf{u} \rightarrow u = \Delta S = 95 \text{ MPa}$	applied stress range ($R \approx 0.05$)
$\boldsymbol{\theta} = [\log C, m]^T$	state model parameter vector
$\boldsymbol{\epsilon}_f = [e^\omega, \epsilon_{\log C}, \epsilon_m]^T$	state model error

where:

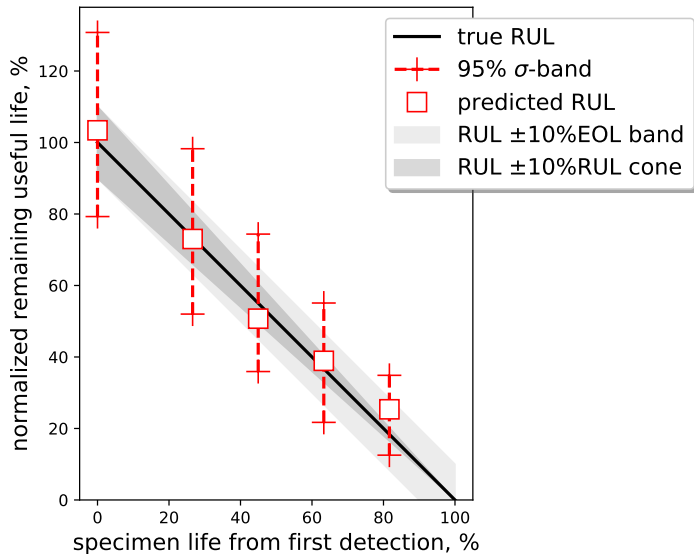
$$\omega \sim \mathcal{N}\left(-\frac{\sigma_\omega^2}{2}, \sigma_\omega^2\right), \quad [\epsilon_{\log C}, \epsilon_m] \sim \mathcal{MVN}(\mathbf{0}, \Sigma_\theta), \quad \epsilon_g \sim \mathcal{N}(0, \sigma_g^2)$$

open *particleFilterPrediction.py*

Prediction



Prediction



A few things to remember

- ▶ The model error (or process noise) e^ω has that form for a reason. Please see *Corbetta et al. MSSP 2018, 104; 305:322*
- ▶ Try to implement Kernel smoothing instead of artificial dynamics for better performance (see *Liu J, West M. In Sequential Monte Carlo methods in practice 2001; 197:223 Springer, NY.*)
- ▶ Using unbounded processes to estimate bounded parameters usually results in poor performance

Prognostics Center of Excellence (PCoE)

Web page:

<http://prognostics.nasa.gov>

Data repository:

<https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>

