Prognostics in presence of imprecise health indicator based on computational geometry and similarity-based approach: a case study on C-MAPSS datasets

Keywords: Computational geometry, Prognostics, Health Indicator estimation, Imprecise health indicator, Similarity-based approach, CMAPSS datasets

Emmanuel RAMASSO

Seminar GDR MACS / GT S3
June 25, 2014, Paris, France
Context & Motivations
Prognostics and Health Management (PHM)

Through the **interpretation** of sensor measurements (D. Simon, 2012):

- **Health assessment** – Early signs of wear
  - Accurately assess system / component’s performance deterioration over the lifetime
  - Accurately detect and isolate any engine system and/or instrumentation malfunctions

- **Prognostics** – Remaining Useful Life (RUL)
  - Predict the state of health in future use
  - Provide a degree of confidence in predictions

**Taxonomy** of PHM approaches

- Category 1 – **Data driven** approaches
- Category 2 – **Physics-based** approaches
Data driven approaches:
- Use **sensor data** (reflecting the system’s behavior)
- Use **pattern recognition** / machine learning / data mining to:
  - Detect a « change » in the system: Mapping data → health state
  - Predict the RUL: Mapping « data → RUL » or « past data → future data »

Model/Physics-based approaches:
- Use **physical models** of the system for RUL estimation
- Modeling physics relies on **defining relationships** between
  - Degradation of components / subsystems and operational conditions
  - Multiple components (inputs, state variables, measurements)

In both cases, the « model » is a **simplified representation** of the system’s behavior: **Uncertainty management** required aiming at compensating modeling errors
Outline

Data-driven models for PHM under uncertainty

- Part 1: Turbofan engine
  - Presentation of turbofan engine and CMAPSS
  - Datasets description (complexity illustrated)

- Part 2: RUL estimation
  - Sensor data have high variability
  - No knowledge about the data generating process, complex noise

RUL-CLIPPER: Remaining Useful Life estimation based on impreCise heaLth Index modeled by simPle Polgons and similarity-basEd Reasoning
Part 1: Turbofan engine
Engine control using an **engine model** to provide guaranteed **performance** throughout the life of the engine, but engine models are **imperfect**

Model used to generate datasets for enhanced on-board turbofan PHM:

- No two engines are the same, sensors not modeled correctly, model inaccuracy during transients and at off-design operating conditions, models not updated once engine into production (design changes not always)...

- **Hybrid** modeling (analytical + empirical) techniques hold promise for capturing engine-model mismatch
Modeling physics of a turbofan engine

(S. Garg and D. Simon, 2012)

Commercial Modular Aero-Propulsion System Simulation (CMAPSS) (Frederick, D., DeCastro, J., & Litt, J., 2007)

Simulink model

Computer model

PHM in presence of uncertainties
Turbofan engine PHM
Simulated datasets on CMAPSS

- CMAPSS (Commercial Modular Aero-Propulsion System Simulation) datasets (Saxena, A., Goebel, K., Simon, D., & Eklund, N., 2008)
- Datasets were generated for prognostics development: Simulating of various operational conditions + faults injection + varying wear degree

(S. Garg and D. Simon, 2012)
Turbofan engine PHM
Simulated datasets on CMAPSS

- CMAPSS (Commercial Modular Aero-Propulsion System Simulation) datasets (Saxena, A., Goebel, K., Simon, D., & Eklund, N., 2008)
- Datasets were generated for prognostics development: Simulating of various operational conditions + faults injection + varying wear degree

**Sensor data**

<table>
<thead>
<tr>
<th>IDX</th>
<th>Symbol</th>
<th>Meaning</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>T2</td>
<td>Tot. temp. at fan inlet</td>
<td>R</td>
</tr>
<tr>
<td>7</td>
<td>T24</td>
<td>Tot. temp. at LPC outlet</td>
<td>R</td>
</tr>
<tr>
<td>8</td>
<td>T30</td>
<td>Tot. temp. at HPC outlet</td>
<td>R</td>
</tr>
<tr>
<td>9</td>
<td>T50</td>
<td>Tot. temp. at LPT outlet</td>
<td>R</td>
</tr>
<tr>
<td>10</td>
<td>P2</td>
<td>Pressure at fan inlet</td>
<td>psia</td>
</tr>
<tr>
<td>11</td>
<td>P15</td>
<td>Tot. pressure in bypass-duct</td>
<td>psia</td>
</tr>
<tr>
<td>12</td>
<td>P30</td>
<td>Tot. pressure at HPC outlet</td>
<td>psia</td>
</tr>
<tr>
<td>13</td>
<td>Nf</td>
<td>Physical fan speed</td>
<td>rpm</td>
</tr>
<tr>
<td>14</td>
<td>Nc</td>
<td>Physical core speed</td>
<td>rpm</td>
</tr>
<tr>
<td>15</td>
<td>epr</td>
<td>Engine pressure ratio (P50/P2)</td>
<td>–</td>
</tr>
<tr>
<td>16</td>
<td>Ps30</td>
<td>Static pressure at HPC outlet</td>
<td>psia</td>
</tr>
<tr>
<td>17</td>
<td>phi</td>
<td>Ratio of fuel flow to Ps30</td>
<td>pps/psi</td>
</tr>
<tr>
<td>18</td>
<td>NRF</td>
<td>Corrected fan speed</td>
<td>rpm</td>
</tr>
<tr>
<td>19</td>
<td>NRc</td>
<td>Corrected core speed</td>
<td>rpm</td>
</tr>
<tr>
<td>20</td>
<td>BPR</td>
<td>Bypass Ratio</td>
<td>–</td>
</tr>
<tr>
<td>21</td>
<td>farB</td>
<td>Burner fuel-air ratio</td>
<td>–</td>
</tr>
<tr>
<td>22</td>
<td>hFB</td>
<td>Bleed Enthalpy</td>
<td>–</td>
</tr>
<tr>
<td>23</td>
<td>Nf_d</td>
<td>Demanded fan speed</td>
<td>rpm</td>
</tr>
<tr>
<td>24</td>
<td>PCNfR_d</td>
<td>Demanded corrected fan speed</td>
<td>rpm</td>
</tr>
<tr>
<td>25</td>
<td>W31</td>
<td>HPT coolant bleed</td>
<td>lbm/s</td>
</tr>
<tr>
<td>26</td>
<td>W32</td>
<td>LPT coolant bleed</td>
<td>lbm/s</td>
</tr>
</tbody>
</table>

**Operating conditions**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Unit</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRA</td>
<td>Throttle Resolver Angle</td>
<td>%</td>
<td>[20, 100]</td>
</tr>
<tr>
<td>M</td>
<td>Mach number</td>
<td>Mach</td>
<td>[0, 0.84]</td>
</tr>
<tr>
<td>ALT</td>
<td>Altitude</td>
<td>Kft</td>
<td>[0, 42]</td>
</tr>
</tbody>
</table>

TRA: pilot power request
CMAPSS (Commercial Modular Aero-Propulsion System Simulation) datasets (Saxena, A., Goebel, K., Simon, D., & Eklund, N., 2008)

<table>
<thead>
<tr>
<th>Datasets characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nb. fault modes</td>
</tr>
<tr>
<td>Nb. op. conditions</td>
</tr>
<tr>
<td>Nb. training units</td>
</tr>
<tr>
<td>Nb. testing units</td>
</tr>
<tr>
<td>Minimum RUL</td>
</tr>
<tr>
<td>Maximum RUL</td>
</tr>
</tbody>
</table>

~1360 simulated turbofan run-to-failure testing data (trajectories / instances)

Datasets available since the 2008’s Int. Conference on PHM
- #1, #2, #3, #4: « Turbofan engine datasets »
- #5, #6: « Data challenge » or « PHM’08 datasets »

**Scoring function** (official performance measure called **timeliness**, to be minimized)

\[ S = \sum_{n=1}^{N} S_n \]
\[ S_n = \begin{cases} 
  e^{-d_n/10} - 1, & d_n \leq 0 \\
  e^{-d_n/13} - 1, & d_n > 0 
\end{cases} , n = 1 \ldots N \]

\[ d_n = \text{estimated RUL} - \text{true RUL} \]

During 2008-2014 period:
- ~7100 downloads of datasets #1-#4, ~2000 of datasets #5-#6
- 70 papers using those datasets (Google Scholar)
- 5 papers focused on prognostics using full training/testing datasets as originally provided by the organisers
Operating conditions can be clustered (grouped) into 6 main operating conditions (OC), (Wang 2010, Richter 2012)

For the \(i\)-th training trajectory, sensor data are grouped in each OC

**Hyp. 1:** The sensor measurements do not vary too much between two consecutive samples in the same regime \(\rightarrow\) The evolution is **locally linear**

**Hyp. 2:** The health state is **monotonically** decreasing \(\rightarrow\) An **exponential** model can be used to represent the global evolution of the health index (theoretical output)

\[
\hat{HI}_i(x_t, \theta^p) \equiv 1 - \exp \left( \frac{\log(0.05)}{0.95 \cdot T_i} \cdot t \right), \quad t \in [\sigma_1, \sigma_2].
\]

Deduce a regression model in each regime

\[
HI_i(x_t, \theta^p_i) = \theta^p_{i,0} + \sum_{n=1}^{q} \theta^p_{i,n} \cdot x_{t,n}
\]

This approach was used on all (six) CMAPSS datasets

Prognostics on CMAPSS
From sensor measurements to a health indicator

Theoretical model (Hyp. 2)

HI without operating conditions

HI with operating conditions

PHM in presence of uncertainties
All HI computed on dataset #2 (6 OC, 1 Fault)
Part 2: RUL estimation (RULCLIPPER algorithm)
Based on HI, usual approaches generally use a **functional form** (approx.) of the HI: locally linear, exponential, quadratic, neural networks…

These approximations are then used for **extrapolation** for RUL estimation.

Problem: **variability of measurements is high**, so that noise models / variance-based parameters (embedded in methods) do not generalise well.

A rather unusual approach is proposed: a HI is interpreted as a "geometric figure", more exactly a **planar polygon** ⇒ Imprecise Health Indicator (IHI).
Based on HI, usual approaches generally use a **functional form** (approx.) of the HI: locally linear, exponential, quadratic, neural networks…

These approximations are then used for **extrapolation** for RUL estimation.

Problem: **variability of measurements is high**, so that noise models / variance-based parameters (embedded in methods) do not generalise well.

A rather unusual approach is proposed: a HI is interpreted as a « geometric figure », more exactly a **planar polygon** ⇒ Imprecise Health Indicator (IHI).

*Health indicator*  
*Model: Polygon (set of vertices)*
RULCLIPPER Algorithm
Library of IHI

- From each training data, an IHI is built
- A library of IHI (Imprecise Health Indicator) is obtained for each dataset with known RULs

Sort the training instances with respect to the similarity

Combine the « closest » training runs (RUL fusion), to guess the RUL

PHM in presence of uncertainties
« Closest » ⇒ similarity measure, that has to work between two IHI
Solution: Area of the intersection of polygons (between testing/training data)

Polygon intersection makes use of computational geometry tools, such as generalised polygon clipping: Vatti’s algorithm, working on convex and non-convex polygons

The area of intersection is then converted into a similarity measure:

\[
A_n = \text{Area} \left( P_i \cap P_* \right) \\
R = \frac{A_n}{A_i} \quad \text{(Recall)} \\
P = \frac{A_n}{A_*} \quad \text{(Precision)}
\]

\[
F_{1,i} = 2 \frac{R}{R + P} \\
\text{Called } F_1\text{-measure, bounded in } [0,1]
\]
Illustration of recall (R) and precision (P) measures

\[ A_n = \text{Area}\left( P_i \cap P_\star \right) \]

\[ R = \frac{A_n}{A_i} \quad \text{(Recall)} \]

\[ P = \frac{A_n}{A_\star} \quad \text{(Precision)} \]

\[ F_{1,i} = 2 \frac{R \cdot P}{R + P} \]

Called F\(_{1}\)-measure, bounded in [0, 1]
RULCLIPPER Algorithm

In a nutshell

- **Inputs:**
  - Training run-to-failure data (N-dimensional) with known RUL
  - Testing data

- **Outputs:**
  - RUL of testing data
  - Confidence degree

- **Algorithm**
  - Transform each training data $i$ into $HL_i$
  - Transform $HL_i$ into $IHL_i$ (polygon) ⇒ library
  - Convert the testing data into $IHL_i$ using the i-th model
  - Compute the degree of intersection with each $IHL_i$ in the library (using Vatti’s algorithm + estimate similarity)
  
RUL = weighted sum of the RULs of the K closest training data

Confidence = Average degree of intersection
RULCLIPPER Algorithm
Evaluation of results on CMAPSS datasets

Metrics for evaluation

Accuracy measure: number of predictions with an interval, to be maximised

Too early predictions | Good predictions | Too late predictions
-13 | 0 | +10

Scoring function: penalises more the late predictions than early ones, to be minimised

Each marker has two coordinates: the sum of scores and the average of accuracy over all testing data (for each dataset). The best score was taken over ~6000 parameterisations of usual combination rules of RULs estimates.
Each marker has two coordinates: the sum of scores and the average of accuracy over all testing data (for each dataset). The best score was taken over $\sim 6000$ parameterisations of usual combination rules of RULs estimates.

Packets of markers represent the results on $\sim 500$ combinations of input sensors used to compute the health indicators.
Increasing complexity of datasets $\Rightarrow$ Decreasing performances (increasing of scores and decreasing of accuracies)
Dataset #4: most complex (6 OC, 2 faults) ⇒ Bad (high) score, bad (low) accuracy
RULCLIPPER Algorithm

Results

Without taking operating conditions (OC) into account to compute the HI: worse scores!

Dataset #4: most complex (6 OC, 2 faults), higher score, lower accuracy

By taking OC into account to compute the HI: much better scores!
Comparing with #4: OC are more influent than fault modes on the degradation prediction (for RULCLIPPER)

Dataset #2: Complex (6OC, 1 fault), higher score, lower accuracy
RULCLIPPER Algorithm

Results

Comparing with #2: Performing well on #2 would result in « good » performance on the data challenge (?)

Dataset #5 (data challenge training): Complex (6OC, 1 fault), mid score, mid accuracy
Confirm that OC are more influential than fault modes on the degradation prediction (for RULCLIPPER).

Datasets #1 and #3: less complex (1 OC, 1 or 2 faults), lower (better) score, higher (better) accuracy.
For the ~500 combinations of sensors, about ~6000 parameterisations of standard combination rules of RULs were tested.

For each input sensor subset (among ~500) used to compute the HI:

B=(Bx,By): select the combination rule yielding the lowest score (By), Bx is the corresponding accuracy.

A=(Ax,Ay): select another combination rule with the best accuracy Ay for which the score falls within the interval [By; Ay~\=By+25%]

Then A and B defines a rectangle in the score-accuracy plane.
RULCLIPPER Algorithm

Sensitivity of input sensors for HI estimation

- For the ~500 combinations, estimate and cumulate all rectangles
The selection of the right subset of sensors is critical if the dataset is complex (number of OC, number of fault modes)

#1: The accumulation of rectangles gives form to a «cube»
- Most of subsets of sensors yield good results
- Easy to parameterise the algorithms
The selection of the right subset of sensors is critical if the dataset is complex (number of OC, number of fault modes)

#2: Operating conditions yield two « clouds », one with a high peak:
- The spread means that the input sensors become critical to compute the HI
- The peaks is the position of the best parameterisations
The selection of the right subset of sensors is critical if the dataset is complex (number of OC, number of fault modes)

#3: The second fault mode do not affect too much the results:
- translation of the « cloud » to the upper left hand-side compared to #1
- The peak is higher meaning that the subset is more critical than #1
The selection of the right subset of sensors is critical if the dataset is complex (number of OC, number of fault modes)

#4: For the most difficult dataset, four «clouds» appear: The combination of fault modes and operating conditions greatly increases the sensitivity of sensor subsets on the performances
RULCLIPPER Algorithm
Comparison with the state-of-the-art – All datasets

Scores obtained in 2008

<table>
<thead>
<tr>
<th>Algo. (pseudo.) / Data</th>
<th>#5(_T)</th>
<th>#5(_V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>heracles (1)</td>
<td>737 (3rd)</td>
<td>5636 (1st)</td>
</tr>
<tr>
<td>FOH (2)</td>
<td>512 (2nd)</td>
<td>6691 (2nd)</td>
</tr>
<tr>
<td>LP (3)</td>
<td>n.a.</td>
<td>25921</td>
</tr>
<tr>
<td>sunbea</td>
<td>436.8 (1st)</td>
<td>54437 (22nd)</td>
</tr>
<tr>
<td>bobosir</td>
<td>1263</td>
<td>8637</td>
</tr>
<tr>
<td>L6</td>
<td>1051</td>
<td>9530</td>
</tr>
<tr>
<td>GoNavy</td>
<td>1075</td>
<td>10571</td>
</tr>
<tr>
<td>beck1903</td>
<td>1049</td>
<td>14275</td>
</tr>
<tr>
<td>Sentient</td>
<td>809</td>
<td>19148</td>
</tr>
<tr>
<td>A</td>
<td>975</td>
<td>20471</td>
</tr>
<tr>
<td>mjhutk</td>
<td>2430</td>
<td>30861</td>
</tr>
<tr>
<td>RelRes</td>
<td>1966</td>
<td>35863</td>
</tr>
<tr>
<td>phmnrc</td>
<td>2399</td>
<td>35953</td>
</tr>
<tr>
<td>SuperSiegel</td>
<td>1139</td>
<td>154999</td>
</tr>
</tbody>
</table>

After 2008

<table>
<thead>
<tr>
<th>Algo. (pseudo.)</th>
<th>#5(_T)</th>
<th>#5(_V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULCLIPPER</td>
<td>752</td>
<td>11572</td>
</tr>
<tr>
<td>SBL (P. Wang, Youn, &amp; Hu, 2012)</td>
<td>1139</td>
<td>n.a.</td>
</tr>
<tr>
<td>DW (Hu, Youn, Wang, &amp; Yoon, 2012)</td>
<td>1304</td>
<td>n.a.</td>
</tr>
<tr>
<td>OW (Hu et al., 2012)</td>
<td>1349</td>
<td>n.a.</td>
</tr>
<tr>
<td>MLP (Riad, Elminir, &amp; Elattar, 2010)</td>
<td>1540</td>
<td>n.a.</td>
</tr>
<tr>
<td>AW (Hu et al., 2012)</td>
<td>1863</td>
<td>n.a.</td>
</tr>
<tr>
<td>SVM-SBI (Hu et al., 2012)</td>
<td>2047</td>
<td>n.a.</td>
</tr>
<tr>
<td>RVM-SBI (Hu et al., 2012)</td>
<td>2230</td>
<td>n.a.</td>
</tr>
<tr>
<td>EXP-SBI (Hu et al., 2012)</td>
<td>2282</td>
<td>n.a.</td>
</tr>
<tr>
<td>GPM3 (Coble, 2010)</td>
<td>2500</td>
<td>n.a.</td>
</tr>
<tr>
<td>RNN (Hu et al., 2012)</td>
<td>4390</td>
<td>n.a.</td>
</tr>
<tr>
<td>REG2 (Riad et al., 2010)</td>
<td>6877</td>
<td>n.a.</td>
</tr>
<tr>
<td>GPM2B (Coble, 2010)</td>
<td>19200</td>
<td>n.a.</td>
</tr>
<tr>
<td>GPM2 (Coble, 2010)</td>
<td>20600</td>
<td>n.a.</td>
</tr>
<tr>
<td>GPM1 (Coble, 2010)</td>
<td>22500</td>
<td>n.a.</td>
</tr>
<tr>
<td>QUAD (Hu et al., 2012)</td>
<td>53846</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Better results have been obtained when considering performance measures individually (e.g. 70% on #1 vs 53% in previous work), but here only the scoring function was considered for ranking.
RULCLIPPER Algorithm

Comparison with the state-of-the-art – Data challenge

PHM in presence of uncertainties
Conclusions

The RULCLIPPER algorithm is proposed to estimate the RUL of engineering systems where the health indicators are imprecise:

- RULCLIPPER is made of elements inspired from the computational geometry community and relies on the adaptation of case based reasoning to manage imprecision in training and testing instances.
- It is an original and efficient approach for RUL estimation, validated and compared to past work using the six datasets coming from the turbofan engine simulator (C-MAPSS), including the so-called:
  - turbofan datasets (four datasets)
  - and the data challenge (two datasets)

These datasets are considered as complex due to the presence of fault modes and operating conditions.

In addition to RULCLIPPER:

- A method was proposed to estimate the health indicator
- The selection of the most relevant sensors was tackled.
- Information fusion rules / ensembles were studied to combine RULs
Perspectives

- As for all similarity-based matching algorithms (T. Wang, 2010; E. Ramasso et al. 2012, 2013), the computational cost associated to sort instances is the most important limitation, but:
  - In tests, an i7-vPro 8-cores CPU with tasks parallelisation was used leading low computational costs: Loading and formatting data (from NASA) + HI estimation took ~2 sec.; Intersection with an IHI of length 220 units (among longest ones) took only ~3.8 msec!
  - A procedure can be used to make polygons convex (approximations), that could drastically reduces computational time
  - Computational geometry has become a very active field in particular to improve memory and time requirements, with applications in multimedia (computer graphics such as games) for which CUDA implementations on processor arrays (using graphic cards) were proposed. With such implementations, real-time and anytime prognostics can be performed.

- The extension of RULCLIPPER to multiple health indices is also under study, in particular by using polytopes.
Thanks for your attention

Thanks to the organising committee

This work has been carried out in the Laboratory of Excellence ACTION through the program “Investments for the future” managed by the National Agency for Research (reference ANR-11-LABX-01-01).

Contact: emmanuel.ramasso@femto-st.fr
http://members.femto-st.fr/emmanuel-ramasso/
References

- S. Garg and D. Simon, Challenges in Aircraft Engine Control and Gas Path Health Management, 2012 Turbo Expo

See also references included in [Investigating computational geometry for failure prognostics] to appear in [PHM-Europe 2014 conference] and [International Journal on Prognostics and Health Management, 2014]