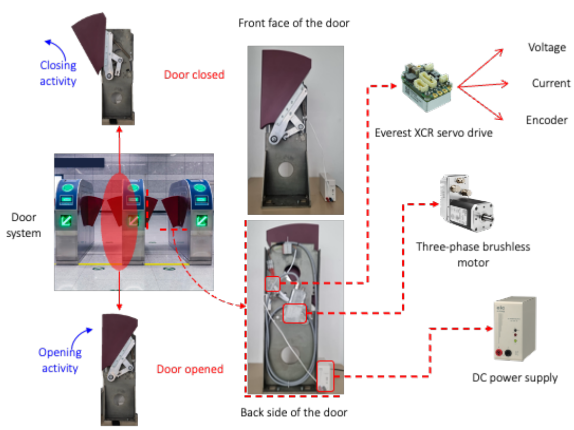
Introduction

This year's Data Challenge, a test bench for the Prognostics of Industrial Machines for System Safety and Integrated Safety (PIMSSIS) is presented. It is used for the development of prognostic algorithms for the estimation of the remaining useful life (RUL) and for the integration of algorithms for health management in the field of system security.

This test bench is a type of rotating machine (servomotor) used in a subway ticket validation door. A general overview of the different parts of the test bench is shown in Figure 1.



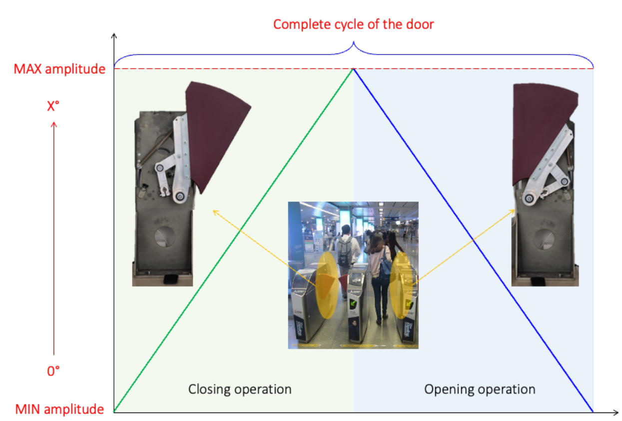


Figure 1: Overall view of the test bench. Figure 2: Illustration of closing and opening activities.

Specifically, the test bench comprises a three-phase brushless motor and a control board. The three-phase brushless motor supplied by Crouzet Motors Industry (France) is used to move the shaft of the door from position 1 (open) to position 2 (closed), i.e., for the closing activity, and reversely from position 2 to position 1 for the opening operation, as shown in Figure 2.

The movement activities of the door are carried out by the controller developed by INGENIA industry (Spain) called EVEREST XCR, which includes several microcontrollers. The XCR controller is used to set the target position of the door and to monitor and collect the data from the installed hall sensors. It can also embed the monitoring algorithm into one of the built-in microcontrollers to interact with the motor for safety and health management.

The objective of the challenge is the RUL estimation of the degradation of the door position, i.e. the activity of the motor when opening or closing, which is affected by the degradation of the position over time.

The winning teams will be asked to prepare a full manuscript which will be featured in the PHM conference proceedings and a representative will be expected to make an oral presentation at the event. The prize will be awarded at the conference social event.

Dataset description and Challenge description

Dataset

The dataset is composed of experiments in which the door of the security system performs several opening and closing operations until the door can no longer be opened.

To generate multiple failure scenarios, the XCR controller is used to inject stochastic errors into the closing activity, resulting in a reduction of the maximum position degree over time until failure. In this case, a failure scenario occurs when the closing process can no longer be executed, i.e. when the closing position of the door is less than 10% of the optimal position. For this purpose, a Python script with a baseline degradation equation (see Equation 1) is used to perform this failure simulation and then store the monitoring data from the installed sensors.

$$X_{\max}\left(t_{s_n}\right) = \left[1 - p\left(t_{s_n}\right)\right] \cdot X_{\max}\left(t_{s_{n-1}}\right)$$

$X_{max}\left(t_{s_n}\right)$ and $X_{max}\left(t_{s_{n-1}}\right)$ are the maximal position degrees of the subway door after the n_{th} and the n_{th-1} degradation.

$p\left(t_{s_n}\right)$ is the degradation percentage caused by the nth shock. It is chosen randomly from predefined ranges and reflects the severity of shocks' impact on degradation.

t_{s_n} is the occurrence time of the shocks and it reflects the different degradation rates (degradation speed) of the door.

The variation of these parameters above allows us to generate several failures representing realistic dynamic behavior of the system degradation.

To illustrate, let us assume that the door moves from position 0 to 190 when closing and returns to 0 when opening. Note that these values correspond to the nominal operation of the door. Initially, the Python script randomly generates the time of the first occurrence of a position deterioration, e.g., t_{s_n1}= 4800s from a range of [3600, 7200] s. This time t_{s_n1} represents the moment when an anomaly occurs and causes a reduction in the actual closing position. Here too, the position deterioration $p\left(t_{s_n1}\right)$ is selected randomly from a range [2, 14] % according to a percentage. For example, if $p\left(t_{s_n1}\right)$=5 %, then the closing position changes from 190 to 180.5.

The triggering threshold of position is fixed to 10% of the optimal closing position and it corresponds to the time when we consider the door as failed because under this value the XCR detects that the closing position is too low and consequently cannot move the door anymore.

The failure scenario described above is executed under three different operating conditions multiple times. Each operating condition describes the velocity, acceleration and deceleration in the closing activity. This latter activity allows the collection of dynamic run-to-failure data to test the performance of prognostic algorithms. The overall data set is collected and summarized in Table 1.

In detail, 16 monitoring parameters are collected in .csv files, and each file contains 600 samples of closing or opening activity. Note that two successive files of closing and opening activities constitute a cycle.

This dataset is part of a single study carried out by Dr. Moncef SOUALHI, Dr. T. P. Khanh NGUYEN and Prof. Kamal MEDJAHER (https://doi.org/10.1016/j.compind.2022.103766). In this study, a first approach with some performance metrics is proposed. It also shows and highlights the practical difficulties in developing PHM algorithms for RUL estimation.

Challenge description

The objective of this data challenge is to predict when the door system will stop working, creating a model able to accurately determine the Remain Useful life (RUL) of the door opening system. The model must be able to predict the RUL without knowing the operating condition, simulating the typical scenario of a real system in which the operating system is unknown.

For this, the teams will have the following experiments for training:

Model training: 48 experiments describing the run to failure of the door closing system are released and used as training datasets. The dataset includes experiments with all operation conditions. For each operating condition, 16 experiments are reported. Table 2 describes the training dataset.

Operating conditions of the PIMSSIS platform				
Velocity	Acceleration	Deceleration	Experiments	Acquisition parameters
15	18	18	16 scenarios	Hardware: INGENIA servomotor File extension: .csv Time: 600 samples/file
10	15	15	16 scenarios	
15	18	15	16 scenarios	

Table 2: Synthesis of the different operating conditions of the platform.

Model testing: 18 experiments describing the run-to failure of the door closing system are used as test datasets for participant. The datasets include operating conditions that were available during the training phase. For each operating condition, for testing 6 experiments will be used.

Collected data (.csv files) Description							
Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
Time	POS_REF	POS_FBK	VEL_FBK	VEL_FBK	FBK_DIGHALL	FBK_DIGENC1	DRV_PROT_VBUS
Duration of acquisition	Reference of desired position	Feedback of actual position	Reference of desired velocity	Feedback of actual velocity	Feedback of digital hall	Feedback of digital encoder	Voltage bus level of the driver
Column 9	Column 10	Column 11	Column 12	Column 13	Column 14	Column 15	Column 16
MOT_PROT_TEMP	FBK_CUR_A	FBK_CUR_B	FBK_CUR_C	DRV_PROT_TEMP	FBK_VOL_A	FBK_VOL_B	FBK_VOL_C
Temperature of the motor	Feedback current A	Feedback current B	Feedback current C	Temperature of the driver circuitry	Feedback voltage A	Feedback voltage A	Feedback voltage C

Table 1: Synthesis of the different monitoring measurements with referent command data.

Data Repository Overview and Scoring

1. Data Structure Overview

This section provides a clear and concise overview of the data organization for the PHME 2026 challenge, to facilitate understanding and usage.

- **General Organization**

The data is organized into two main folders:

-**Train**: contains the training data, organized into subfolders named `Train_1`, `Train_2`, etc.
-**Test**: contains the test data to evaluate your algorithms and submit solutions, with the same structure (`Test_1`, `Test_2`, etc.).

Each subfolder corresponds to a separate experiment (run to failure) or trajectory, i.e., a complete usage sequence of the system from beginning of life to end of life.

- **Contents of a Subfolder**

Let's consider a Train subfolder example content:

  - Measurement files, named according to the pattern `F_1_XXXXX_Opening.csv` and `F_1_XXXXX_Closing.csv`, where two successive files (closing and opening) represent a cycle.
  - Each file contains about 600 rows (variable depending on the cycle), no header, with 16 numerical columns separated by semicolons (`;`).
  - The columns correspond to different physical quantities (see the proposal for details).
  - A RUL file named `F_1_RUL.csv`, which contains a single column (no header): each row gives the Remaining Useful Life (RUL) value for the corresponding cycle. Note that a successive closing and opening activities have same value of RUL.

All measurement files and the RUL file are synchronized: each measurement row has an associated RUL value.

- **Reference to Official Documentation**

The columns in the measurement files correspond to specific physical variables, described in the official challenge documentation (proposal). It is essential to refer to it to correctly interpret each column.

- **Submission format**

  - The submission file should contain 2 columns, column 1 is the test ID and column 2 is the RUL results corresponding, below an example of the file content:

1	2567
1	2567
1	2566
1	2566
.	.
.	.
.	.
1	1
2	2098
2	2098
2	2097
2	2097
.	.
.	.
.	.
19	400
19	400
19	399
19	399
.	.
.	.
.	.
19	1

  - The format is universal: semicolon as separator, no header, and strict synchronization of rows between Test file measurements and RUL.

- Tips and Key Points
  - Always check that the number of rows in the measurement files and the RUL file is identical.
  - Use the official documentation to interpret the columns.
  - Evaluation and processing scripts must respect this structure to ensure valid results.

2. Scoring results

The PHME 2026 challenge rewards algorithms that remain accurate across the entire RUL trajectory. Each submission is compared to the official True_RUL.csv; our evaluation pipeline computes per-test metrics, normalizes them, and produces a final score between 0 and 1. Scores close to 1 indicate stable, predictive models able to keep the error under control throughout the timeline.

- Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum(y_{true} - y_{pred})^2}$$

penalizes large deviations between predicted and true RUL.

- Highlights large deviations and punishes outliers.
- Sensitive to noise but ensures critical failures are captured.

- Precision (Mean Relative Error)

$$Precision = 100 \times \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\left(\frac{|y_{true,i} - y_{pred,i}|}{y_{true,i}} \leq 0.1\right)$$

It measures the percentage of time steps whose relative error stays within ±10%.

- Expressed in %, easy to interpret for stakeholders.
- Guarantees fairness when RUL magnitudes differ between tests by scaling with the true value.

- Prognostic Horizon (PH)

$$PH = \frac{t_{EOF} - t_{\alpha}}{t_{EOF}}$$

with $t\alpha$ first index where relative error > 20%

- Measures how long the model stays within acceptable tolerance.
- Values near 1 mean the algorithm keeps predictions within ±20% almost to the end of life.

- Normalization

Raw RMSE or Precision may have very different scales, making it hard to combine them directly. Normalization converts each metric to the same [0, 1] range so that:

- Perfect behavior maps to 1.
- Extremely poor behavior tends toward 0.
- The final score remains interpretable and comparable across teams and datasets.

- Normalization Details

We use the fitted function f(m) to map raw values to [0, 1]. Parameters stem from the calibration objectives (e.g., f(0)=1, f(57)=0.9, f(3000)=0.1).

$$f(m) = \frac{a}{m^b + c} + d$$

Parameter meaning:

- a control the amplitude: larger a lift the initial value and accelerates the decay.
- b shapes the curvature: higher b makes the drop steeper after small errors.
- c shifts the curve horizontally; it dictates how fast the denominator grows and thus when the score leaves the plateau near 1.

- d is the floor value (asymptote when the error → ∞). Here d = 0, so the score tends toward 0 for very large errors.
- RMSE parameters: a=4443.76, b=1.53, c=4443.76, d=0.
- Precision parameters: same as RMSE since the targets are identical.
- PH: already normalized by design; no extra mapping needed.

This function ensures a smooth decay: small errors remain near 1 while very large errors quickly drop below 0.1, stressing robustness.

Final Score Interpretation

The official script combines the normalized components as:

$$Score = \frac{RMSE_{norm} + Precision_{norm} + \alpha \times PH_{norm}}{2 + \alpha}$$

- $\alpha = 2$ gives more weight to PH, emphasizing long-term predictive stability.
- Rough guide: score > 0.9 = excellent, 0.7–0.9 = good, 0.5–0.7 = average, <0.5 = needs improvement

Prize

To be announced soon

Relevant Dates

Dead line of submission: March 31st

Teams

Collaboration is encouraged and teams may comprise students and professionals from single or multiple organizations. There is no requirement for team size.

The winning teams will be selected and awarded contingent upon:

- Having at least one member of the team register and attend the PHM 2026 Conference.
- Submitting a peer-reviewed conference paper.
- Presenting the analysis results and techniques employed at the conference.

The organizers of the competition reserve the right to modify these rules and disqualify any team for any efforts it deems inconsistent with fair and open practices.