Model-based & hybrid condition monitoring of mechatronic systems

2nd Public Technical Workshop
Horizon 2020 European Training Network MOIRA
07/06/2024

Konstantinos Gryllias

Department of Mechanical Engineering, KU Leuven

Outline

- Introduction
- Model based monitoring based on state-parameter estimation
- Model based monitoring based on Force estimation
- Physical model-based monitoring
- Limitations of Machine Learning & Deep Learning
- Taxonomy of Transfer Learning
- Applications
- Conclusions
- Open challenges

Condition Monitoring

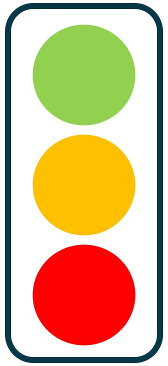









- Fault / Anomaly Detection
- Fault Diagnosis
- Prognosis / Estimation of RUL



- Healthy Operation
- Alarm
- STOP

Condition Monitoring

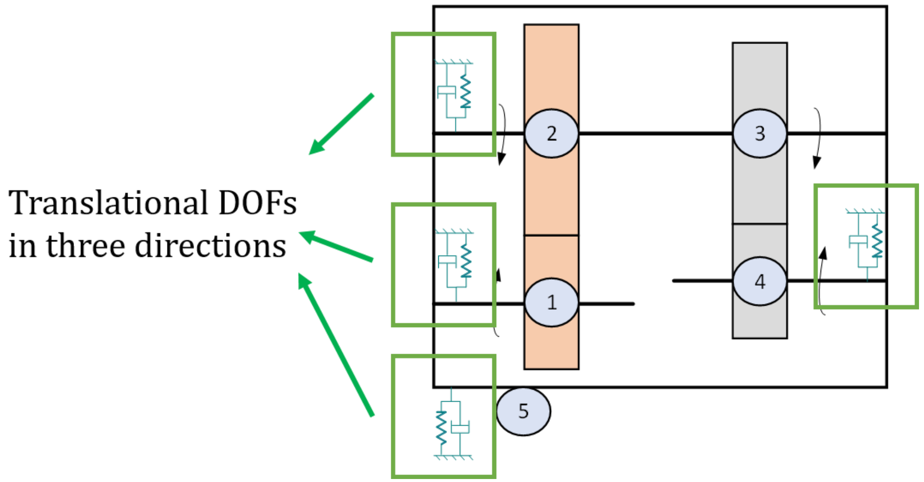
- Sensors – Data acquisition
- Monitoring Indicators / Features
  - Signal Processing
    - Fourier Analysis, Short Term Fourier Transform, Wavelets, Envelope Analysis, Cyclostationary Indicators
  - Machine Learning & Deep Learning
    - End-to-End monitoring

Monitoring via state-parameter estimation

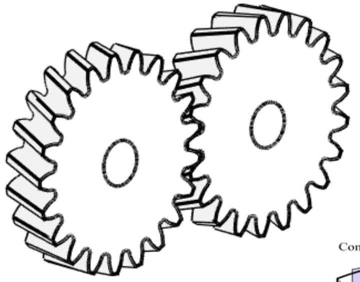
- Modelling of the component / system using some specific parameters
- State-parameter estimation and tracking of the specific parameters
- Threshold setting

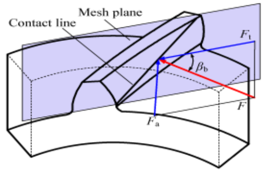
Modelling

- Model of helical or spur gears
- 1-stage
  - 11 DOF = 2 rotations + 3 translations x 2 “bearings”/shafts + 3 translations x 1 housing
- 2-stage
  - 15 DOF = 3 rotations + 3 translations x 3 “bearings”/shafts + 3 translations x 1 housing



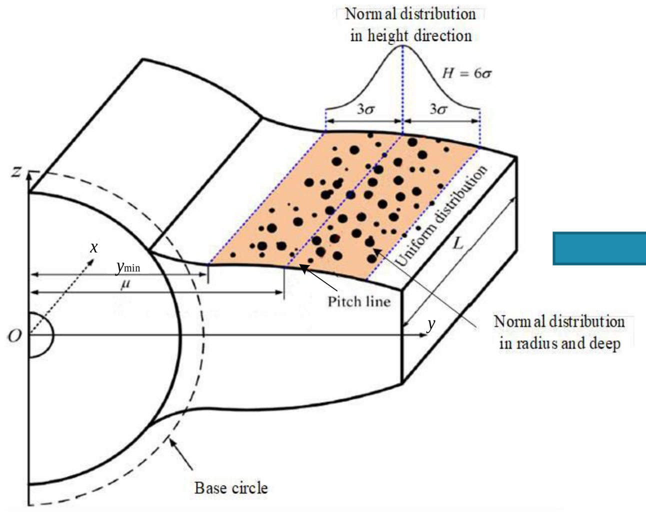
Modeling – Contact model

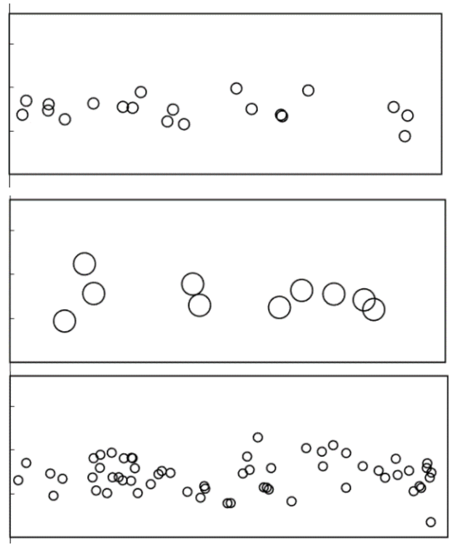




- MUTANT analytical
  1. Geometry calculation
  2. Stiffness
     - Computation along gear profile for a number of slices
     - Global stiffness (bending, shear, axial, …)
     - Local stiffness (local Hertzian contact)
  3. Analysis
     - Contact detection
     - Integration of EOMs

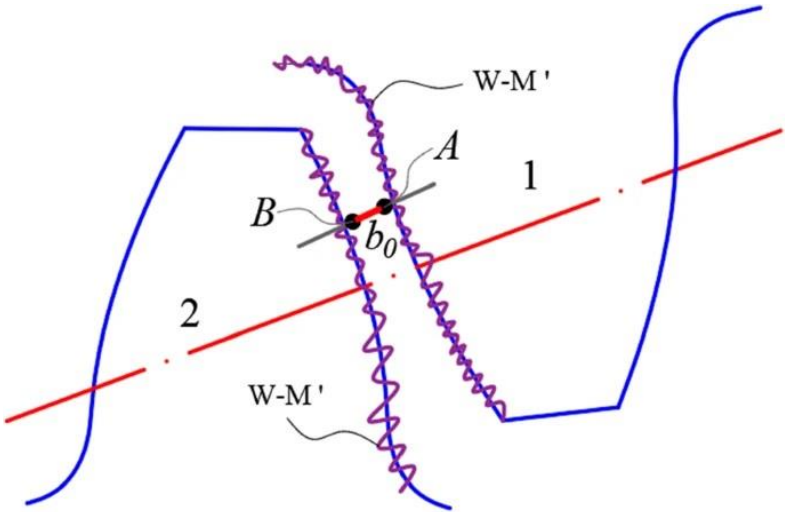
Modeling - Defects





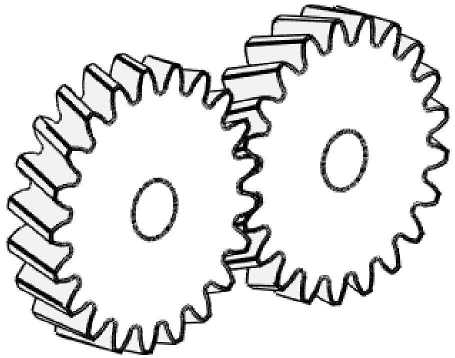
...

- Pitting defect generation
  - # of defects: deterministic (e.g. 20)
  - Location
    - Width direction: uniform distribution
    - Height direction: normal distribution (mean = pitch line)
  - Diameter
    - Normal distribution (mean = 250 μm)
  - Depth
    - Normal distribution (mean = 15 μm)



- Input in model: tooth profile modification along flank (coordinates)
- Effect
  - Contact detection (analysis part of model)
  - Stiffness calculation → limited (only significant influence for cracks)

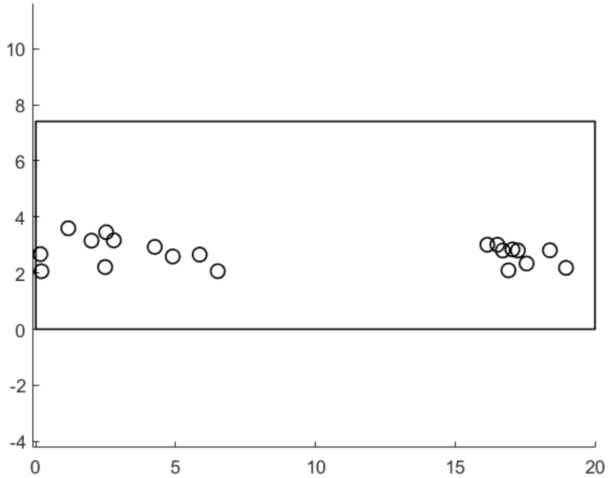
Data generation - Example

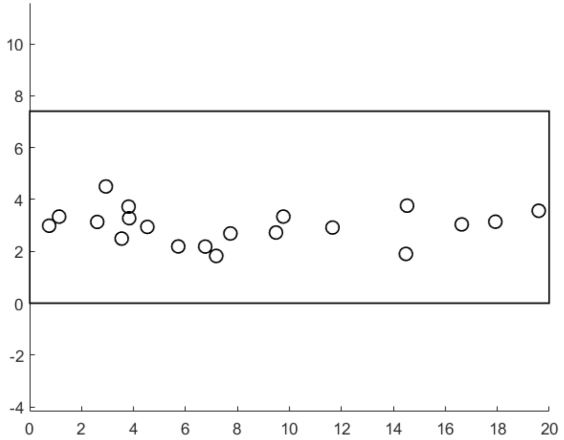


Nominal settings

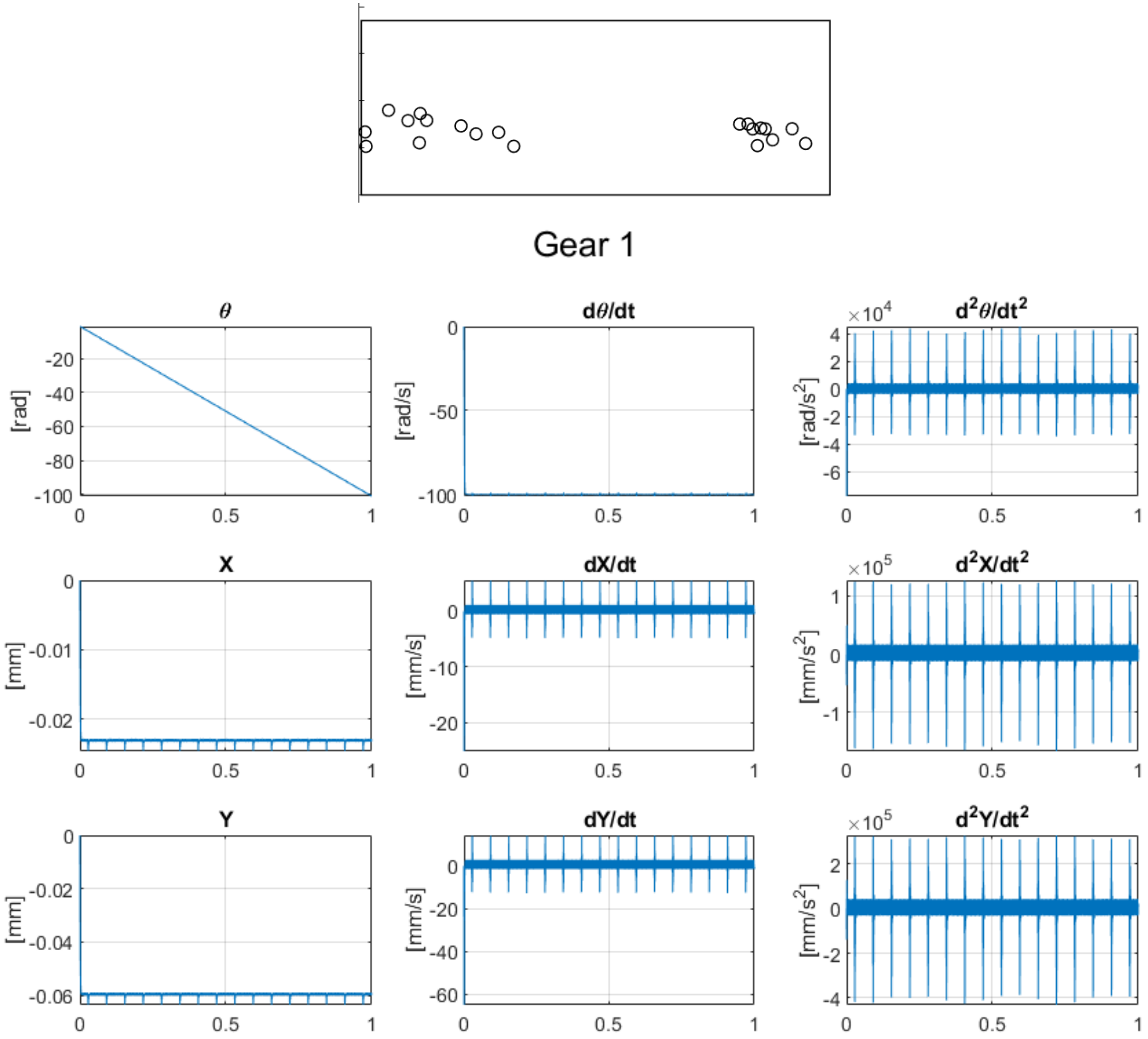
- Helical gear pair 21/21
- 20 pitting defects on gear 1, tooth 17
  - Average depth: 15 μm ($\sigma_d = 2/3\mu m$)
  - Average diameter: 250 μm ($\sigma_r = 15/3\mu m$)
  - Average height: pitch line (2.93mm, $\sigma_h = 2/3mm$)

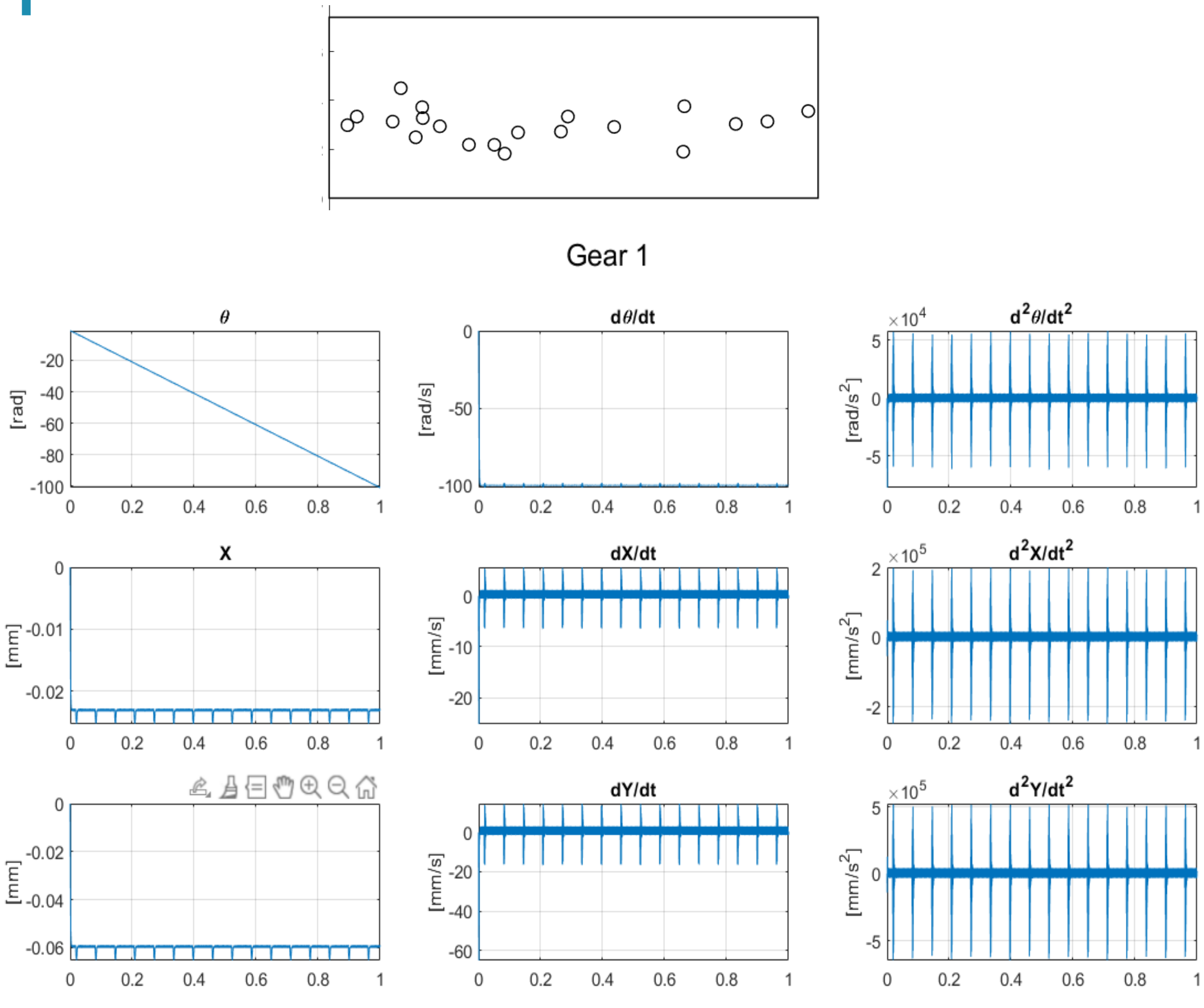
2 samples for defect generation



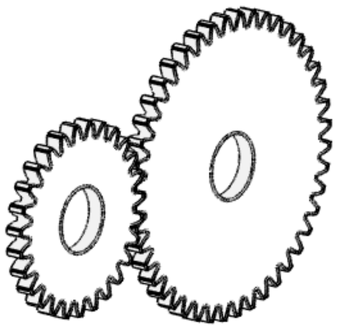


Data generation - Example

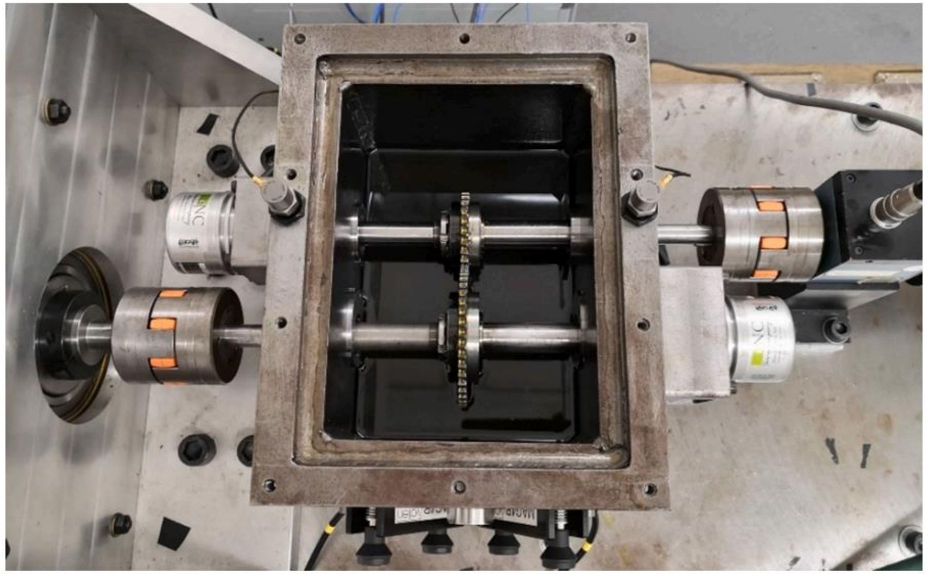


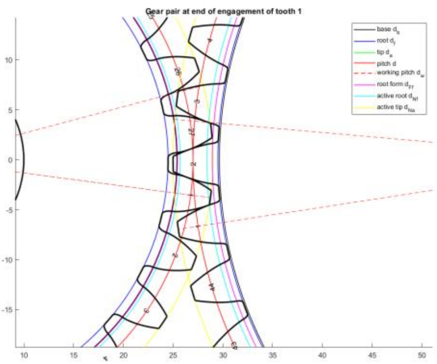


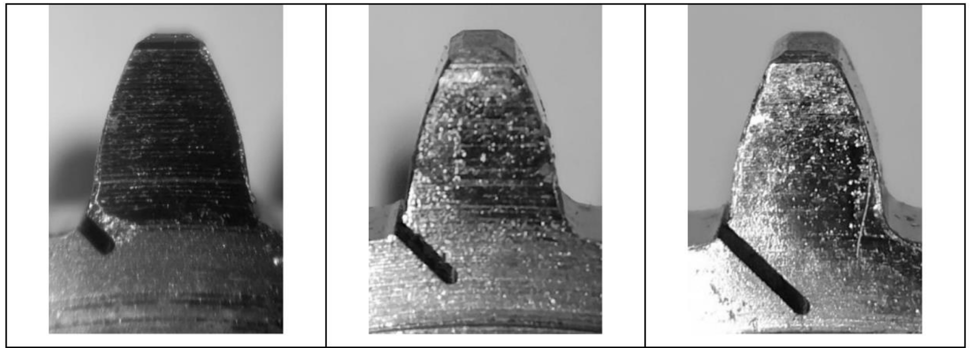
Data set – University of New South Wales



→Encoders on each shaft free end
→Accelerometer on housing
- Measurements at different loads / speeds / conditions

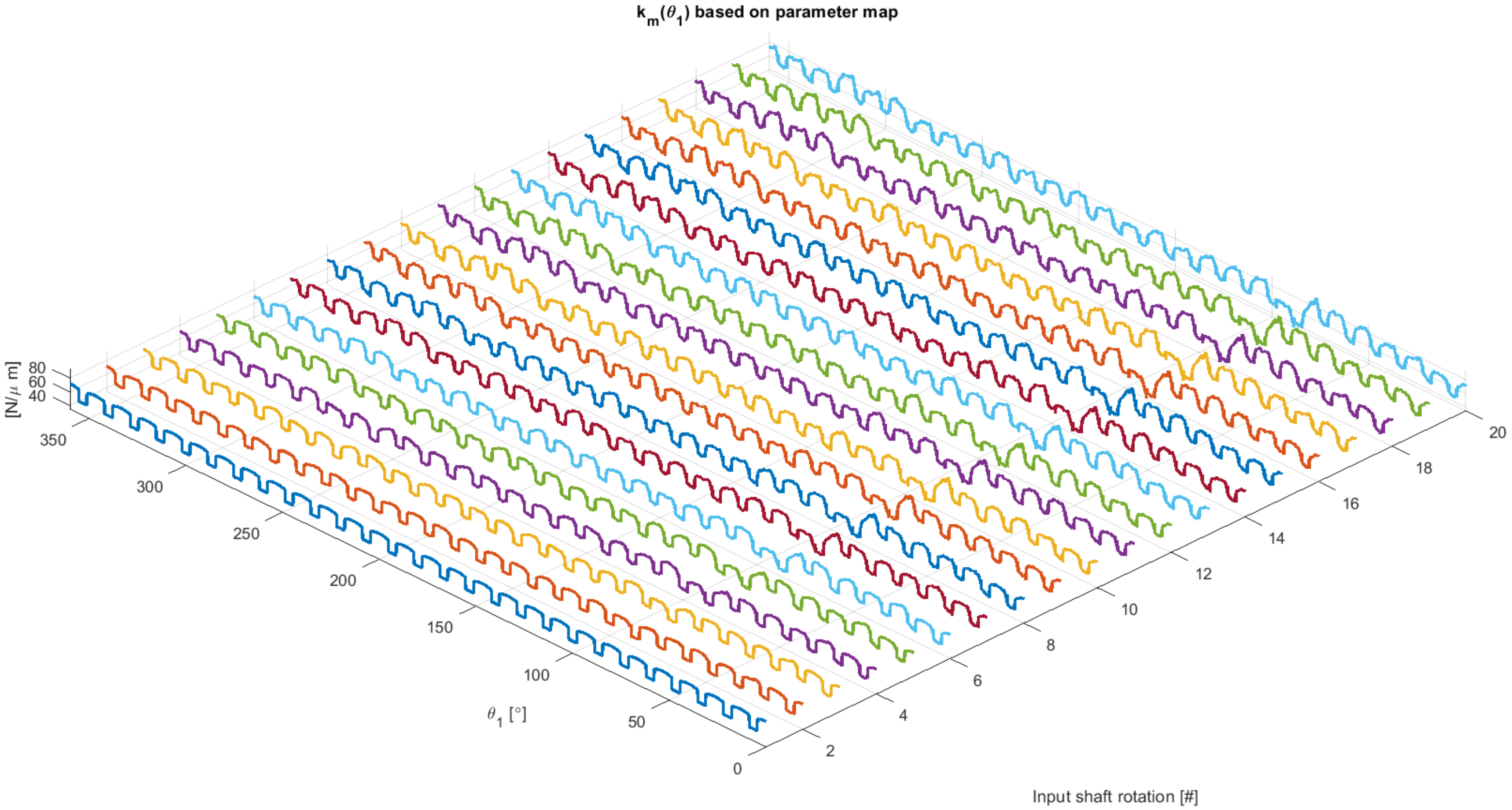


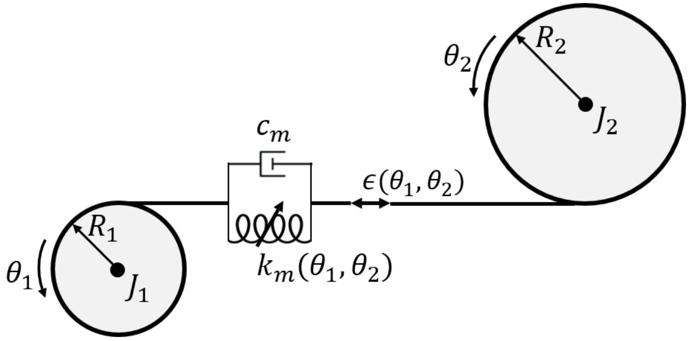


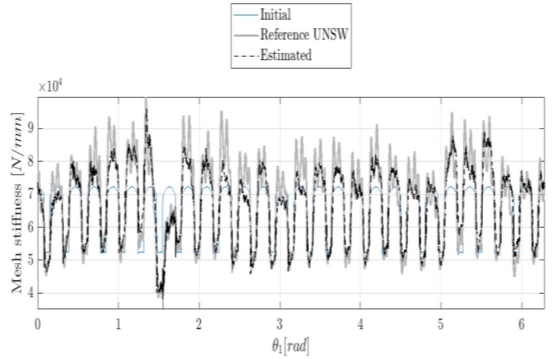


Small (19 %) | Medium (36 %) | Large (50 %)

Gear mesh stiffness estimation (120 RPM, 20 Nm, large crack)

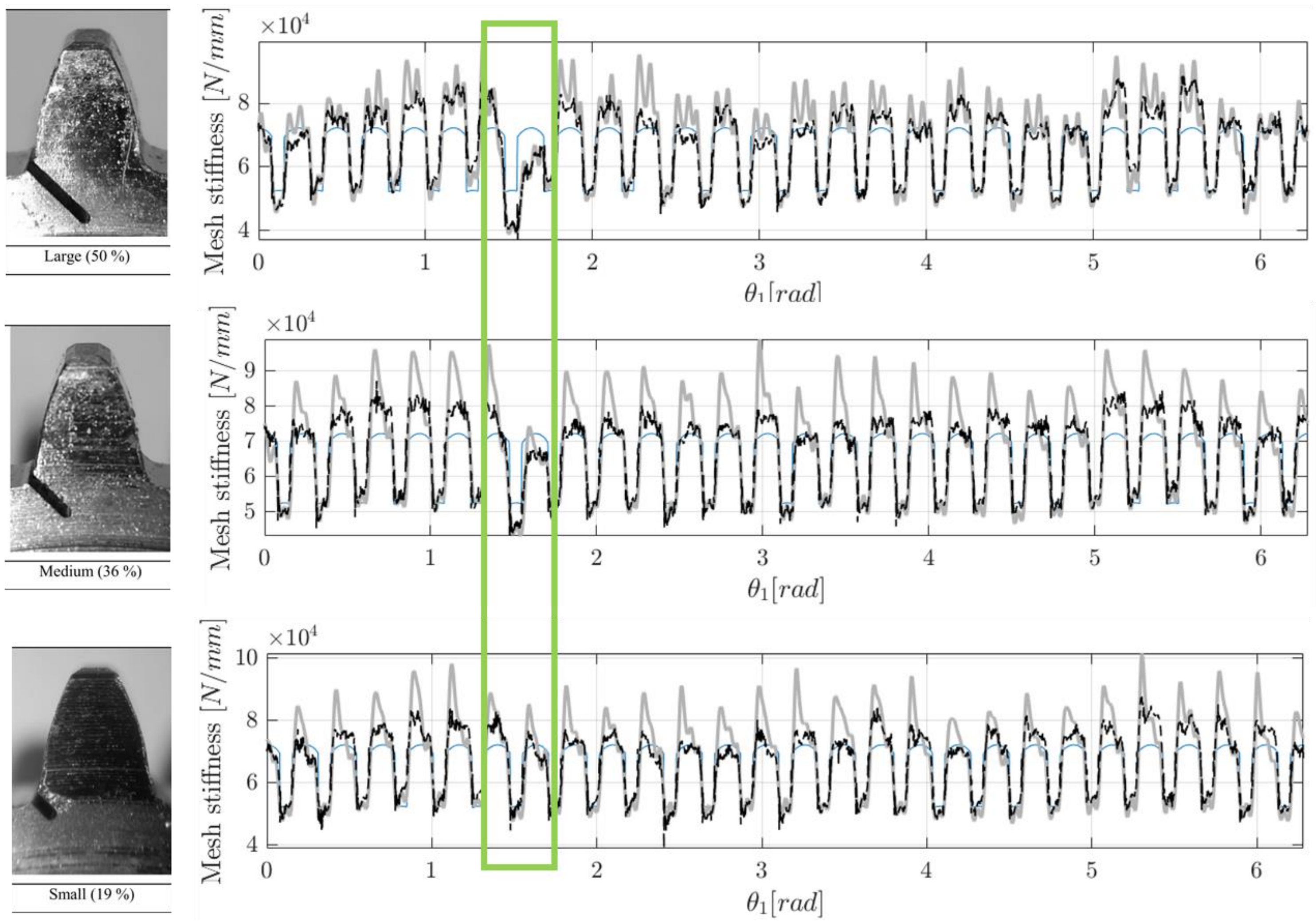


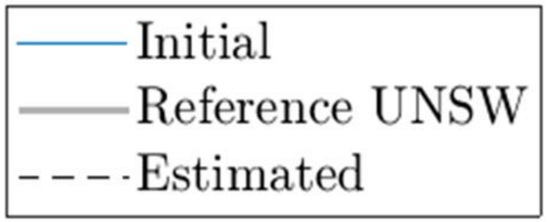


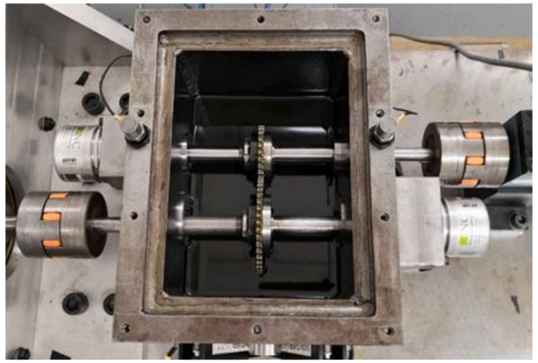


Results for crack sizes (120 RPM, 20 Nm)

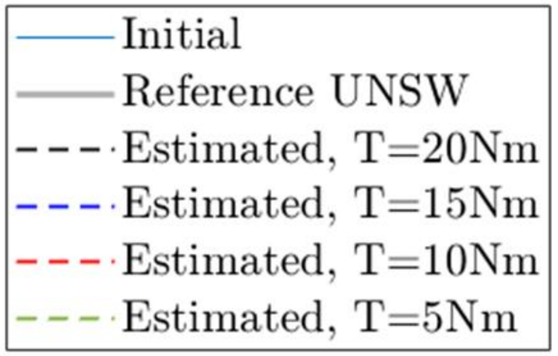
Crack size

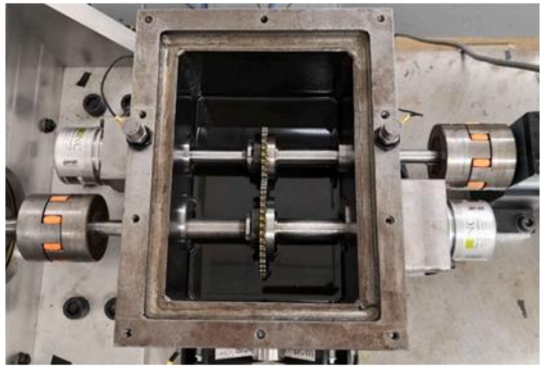


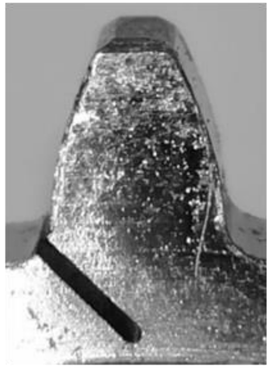




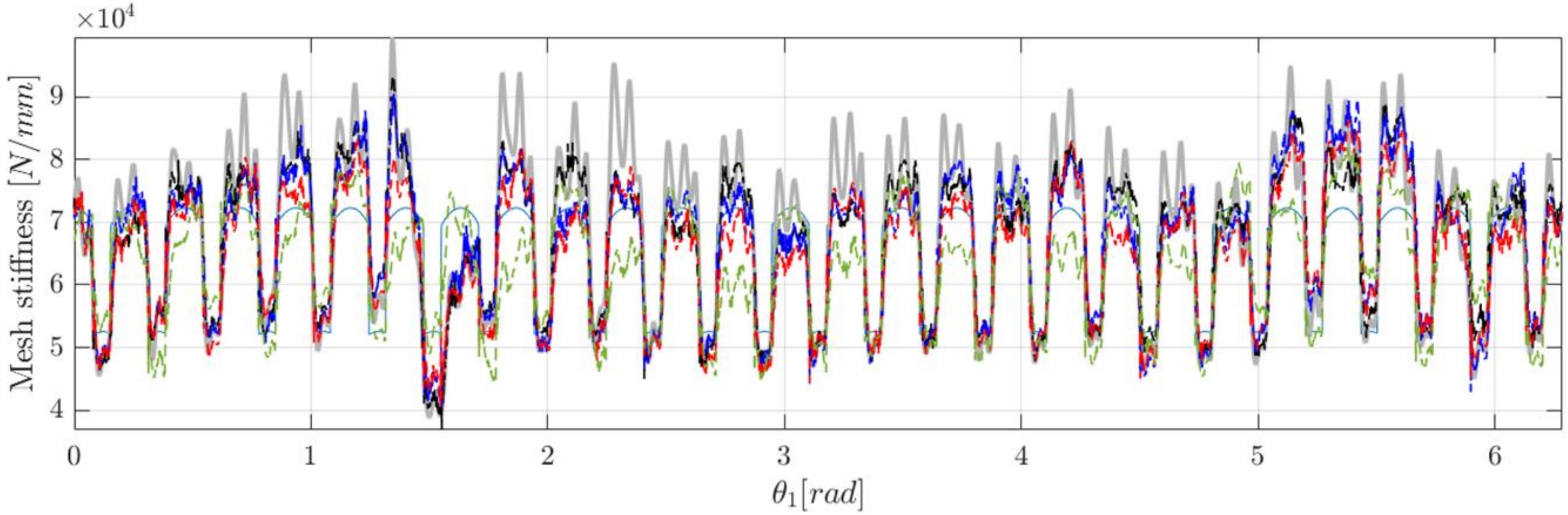
Results for different torques (120 RPM, large crack)



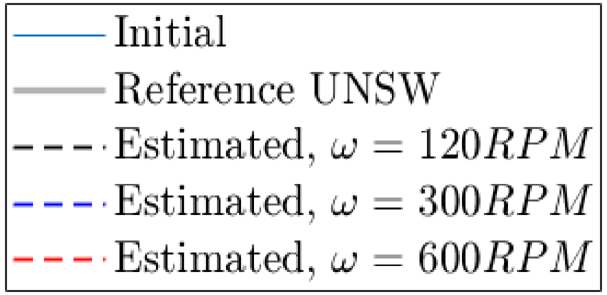


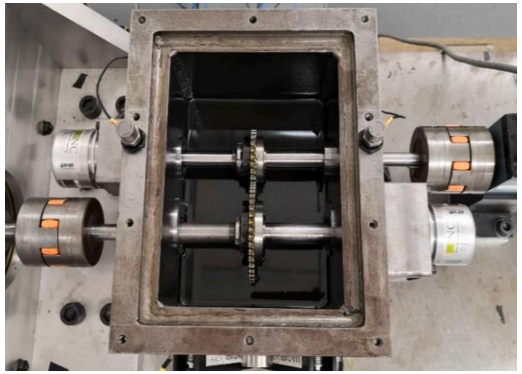


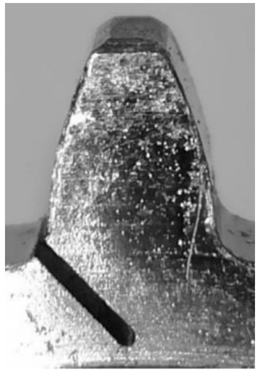
For lower torques equivalent stiffness may be a bit lower due to reduced contact surface
(+ signal to noise ↓)

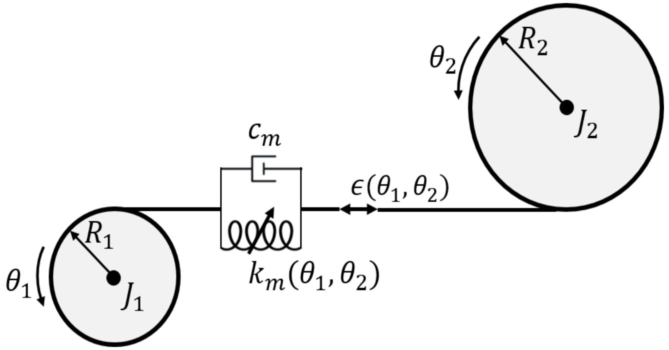


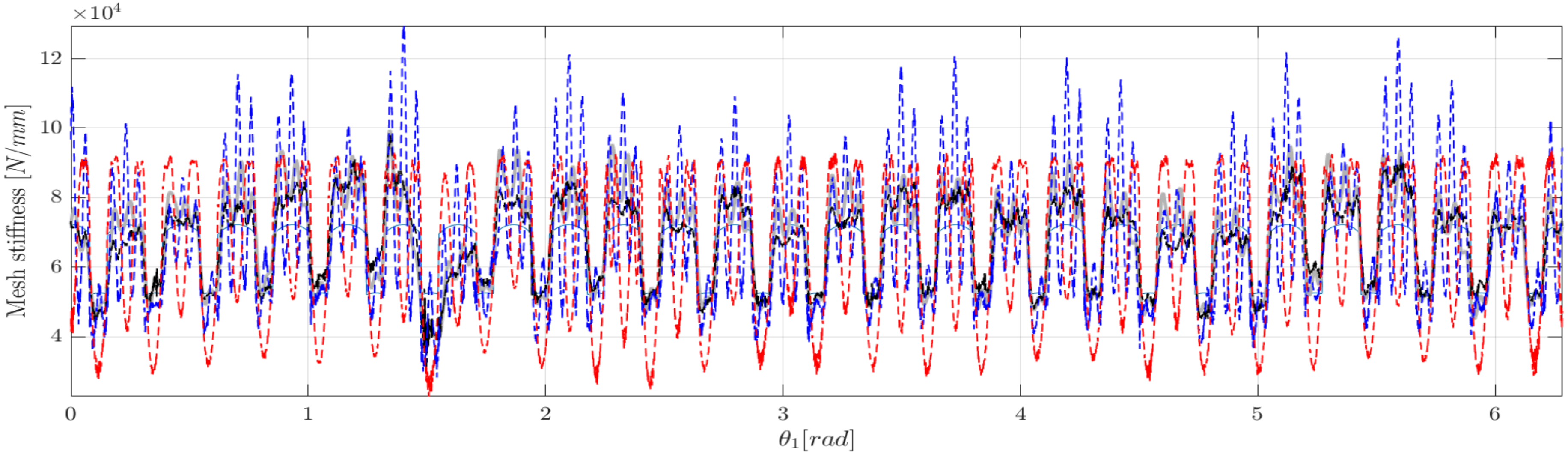
Results for different speeds – 2 DOF model (20Nm)



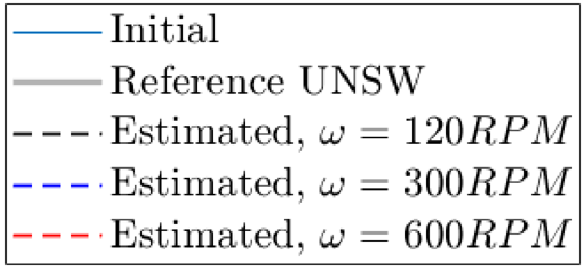


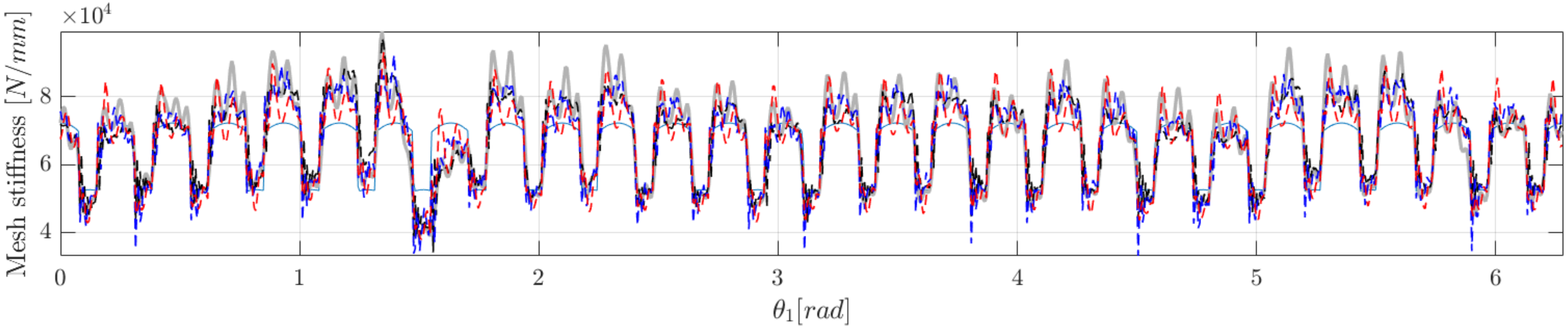


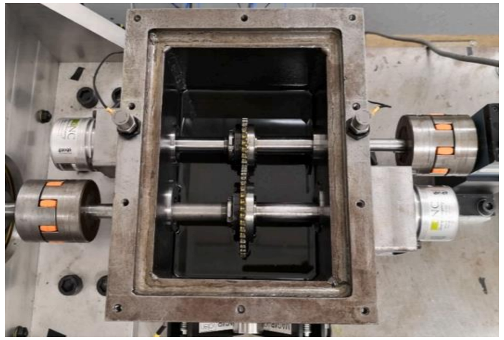


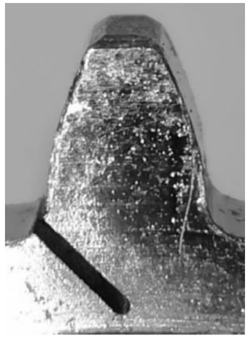


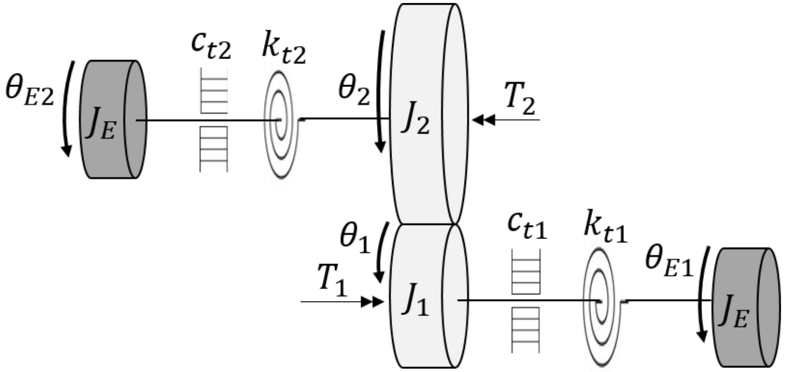
Results for different speeds – 4 DOF model (20Nm)











Estimation of force in bearings

- Attach strain gauges to high SNR locations
- Assume a general boundary condition description
- Identify the boundary condition stiffness via optimization
- Estimate the force using Virtual Sensing techniques

Approach

- Consider grounded springs at the interface of the structure.
- Spring stiffnesses are found via optimization, using the measured strain response from a known load.
- Set up the optimization problem
  - Minimizing the difference between predicted and measured strain:

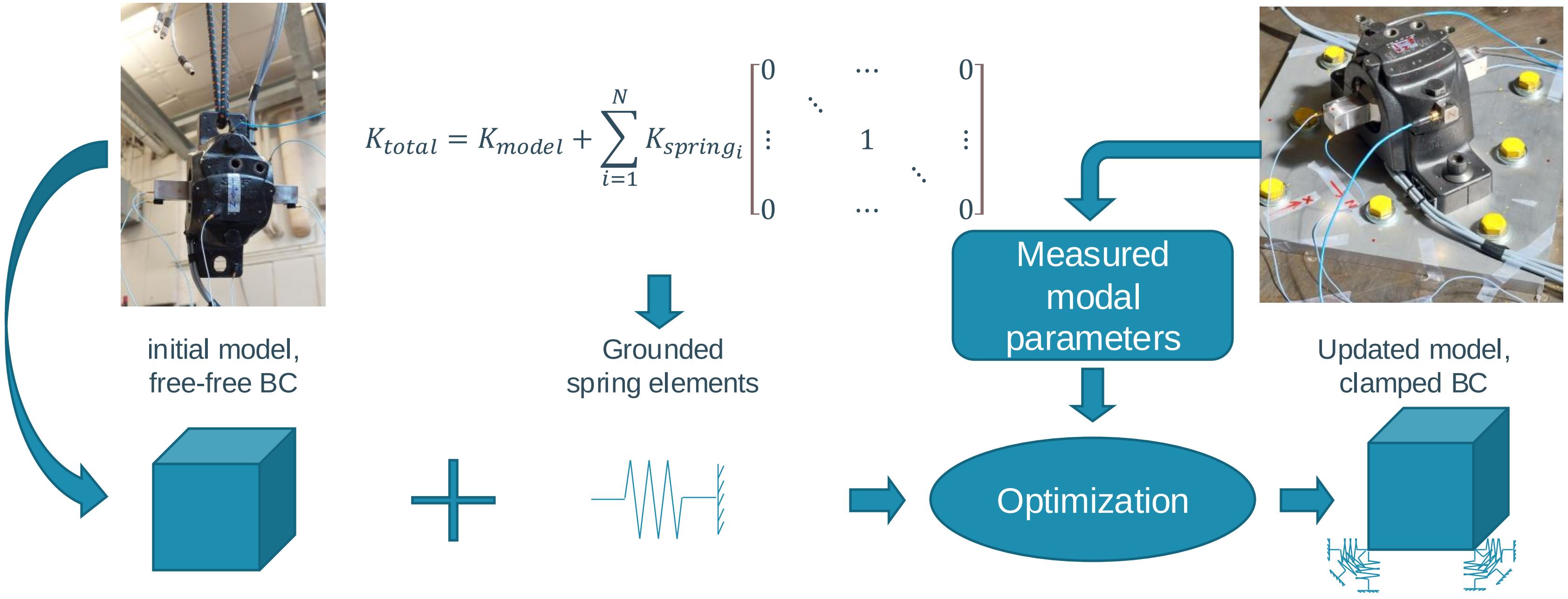
$$\arg\min_{\mathrm{k}} \sum_{i=1}^{n} (\epsilon_i - \epsilon'_i)^2$$
$$s.t.\, \epsilon_i = B_i u \,, F = K_* u$$

  - The analytic derivative of the objective function is used to speed up the heavy optimization:

$$\frac{\partial OF}{\partial k_j} = -2u^T \frac{\partial K_*}{\partial k_j} K_*^{-1} \sum_{i=1}^{n} B_i^T (B_i u - \epsilon_i)$$

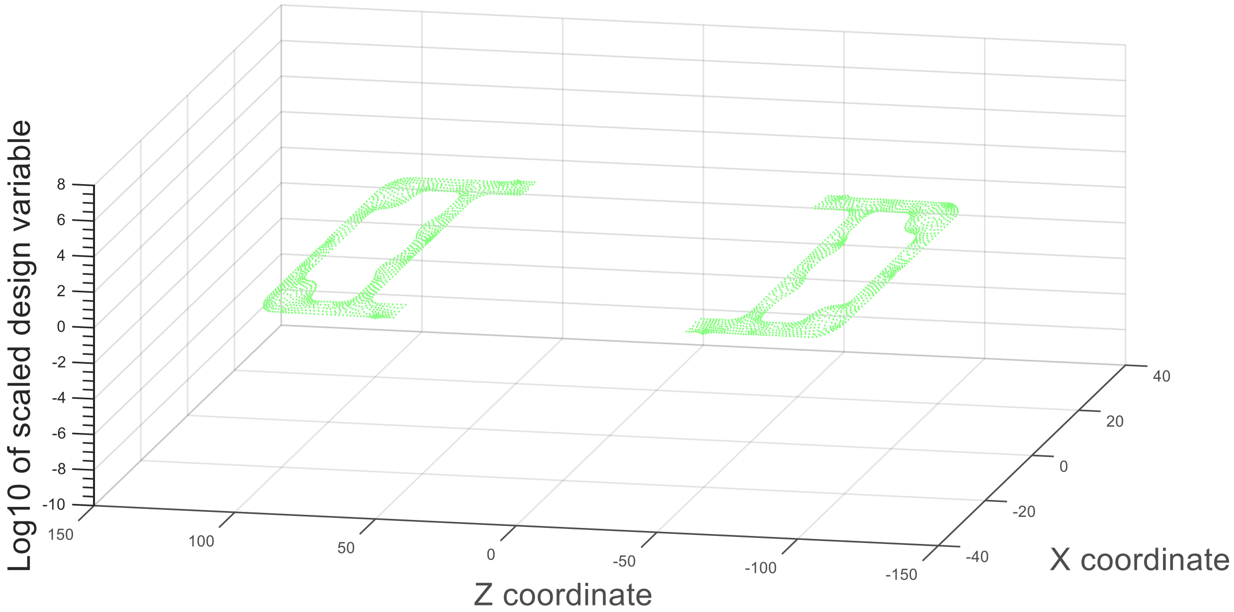
  - This is an underdetermined problem → regularization or smoothing filters are used.

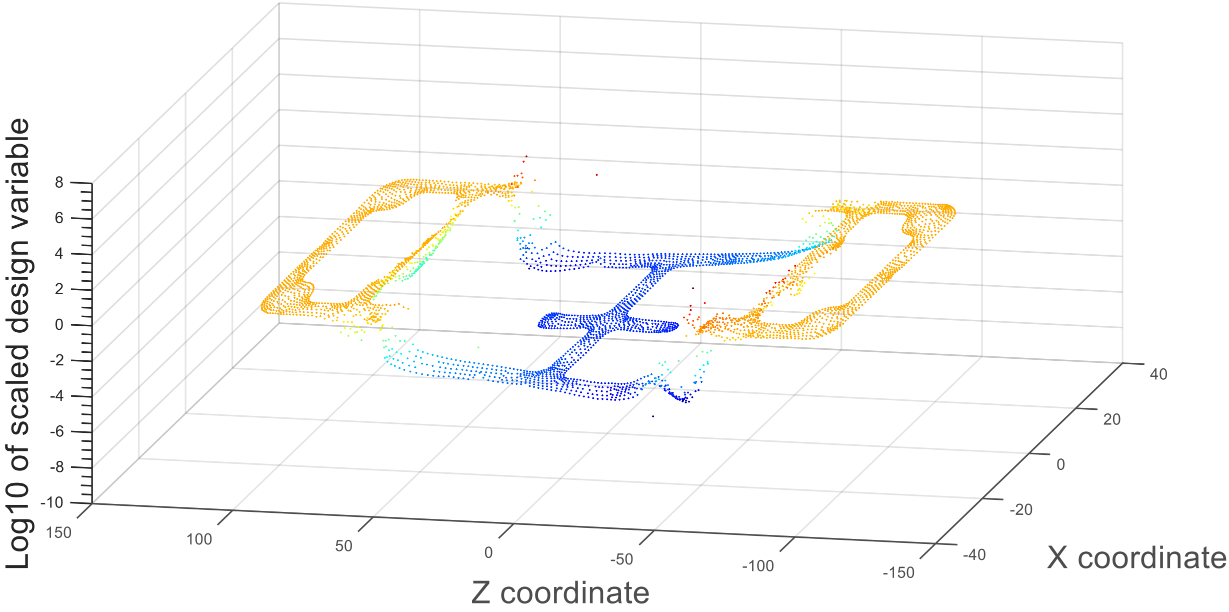
Approach



$$K_{total} = K_{model} + \sum_{i=1}^{N} K_{spring_i} \begin{bmatrix} 0 & \ddots & \cdots & & 0 \\ \vdots & & 1 & & \vdots \\ 0 & & \cdots & \ddots & 0 \end{bmatrix}$$

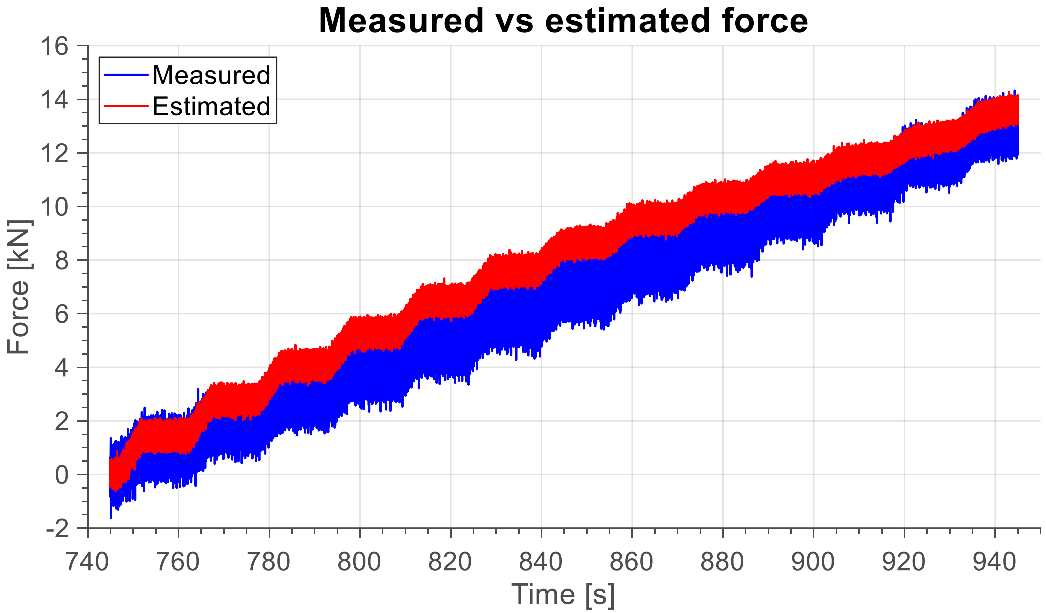
Numerical validation case with known BC

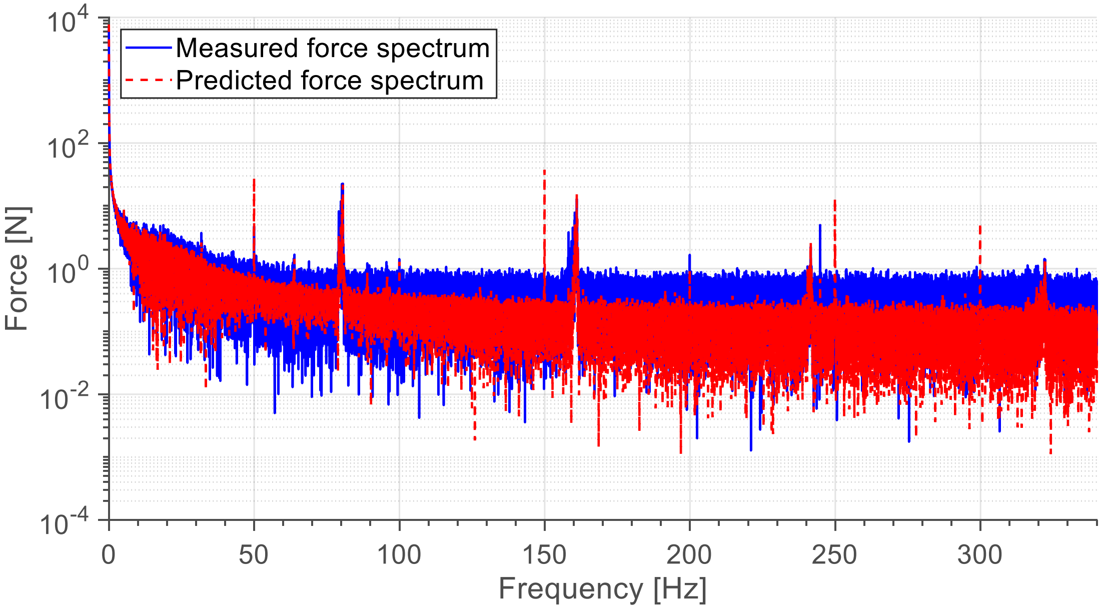




Application

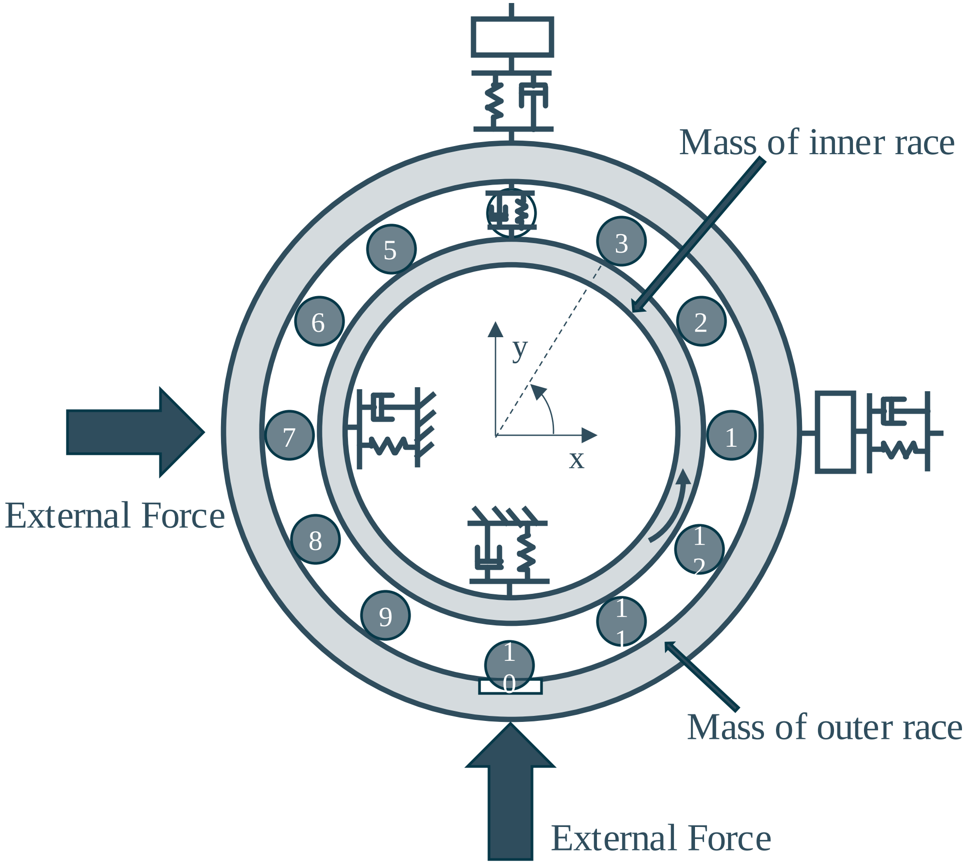
- The global stiffness distribution is predicted. This leads to an updated model that accurately predicts the strain response of the experimental case.
- The predicted input force matches with the measured input by the force cell.



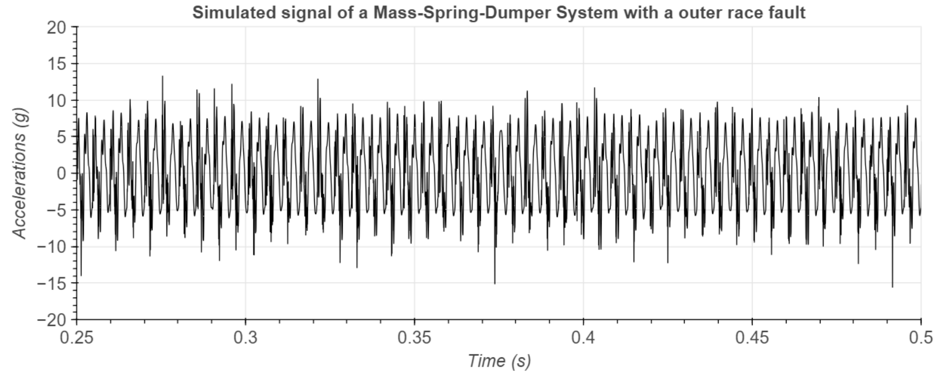


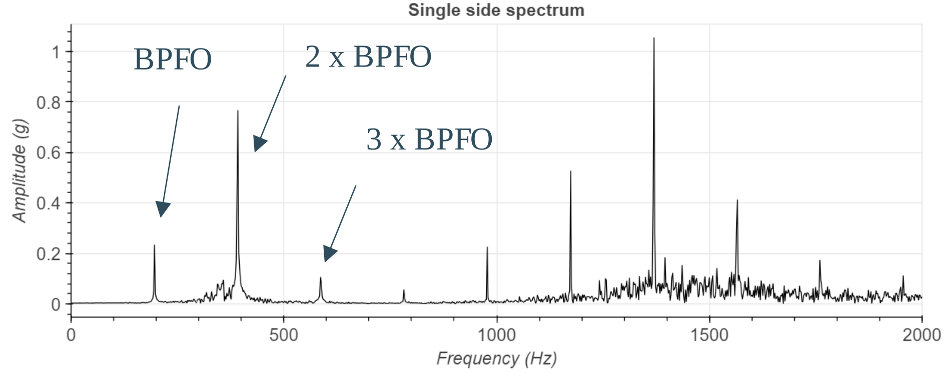
Physical model-based monitoring

Bearing Physical Model



Simulation signals



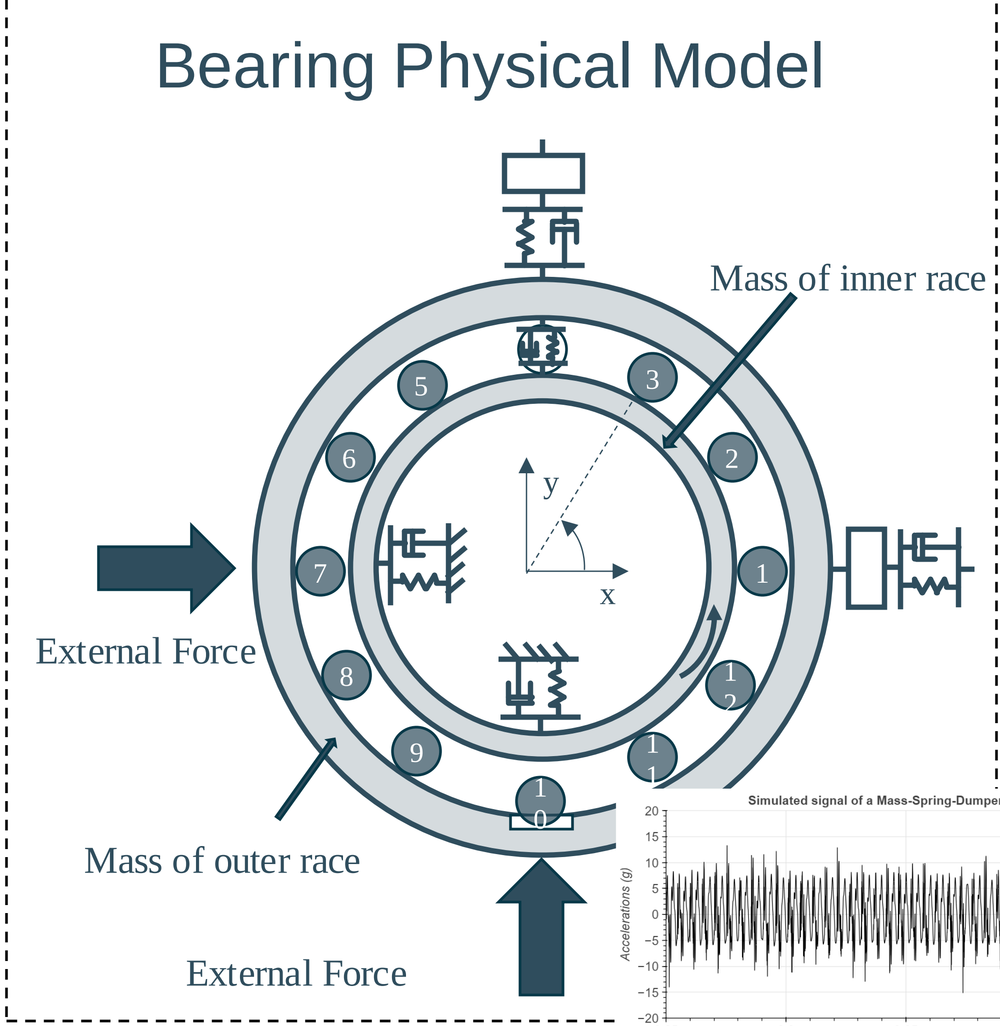


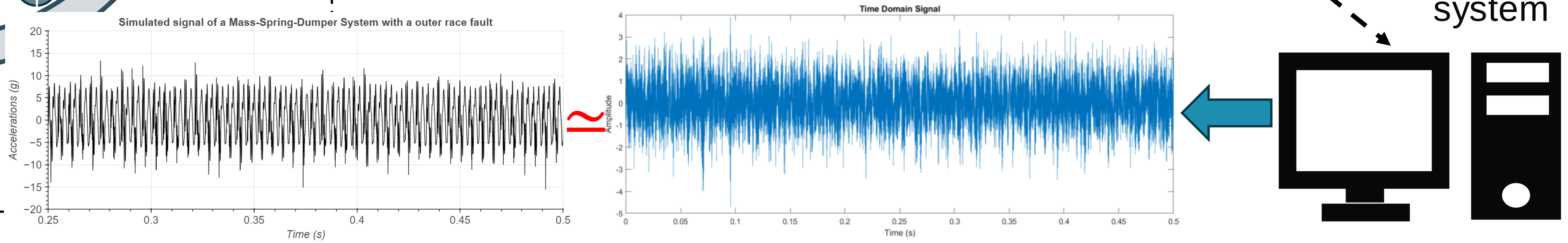
Connections?

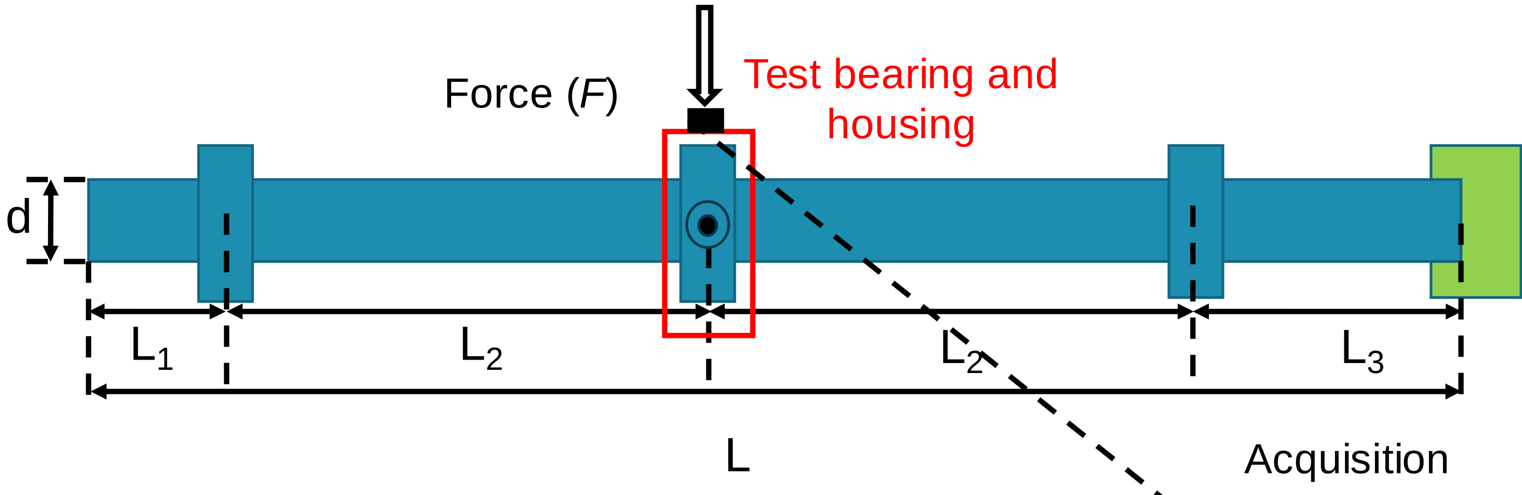
Observable vibration data

Physical model-based monitoring

Assumption 1: The energy that real defects generated in the measurements can be represented by RMS which is simulated by a Quasi-Fault physical model.





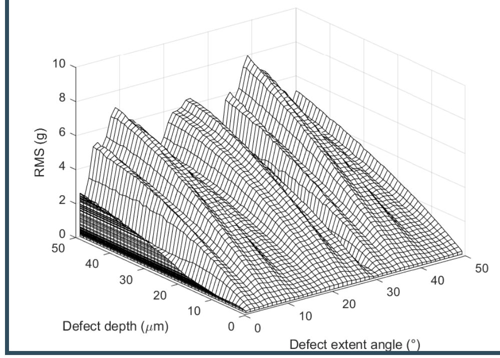


Flowchart

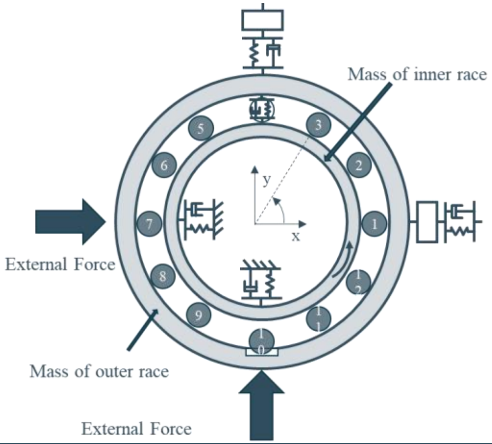
Quasi-Fault size determination

Acquisition system

RMS map

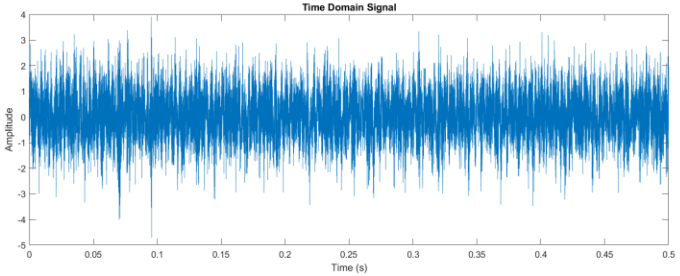


Physical model

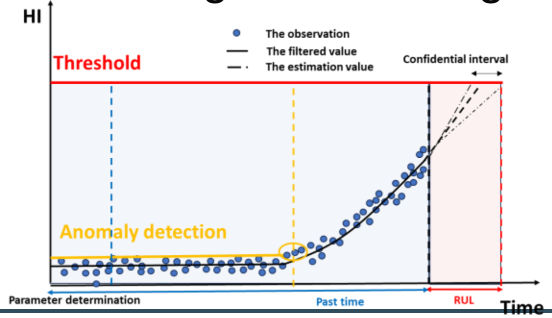


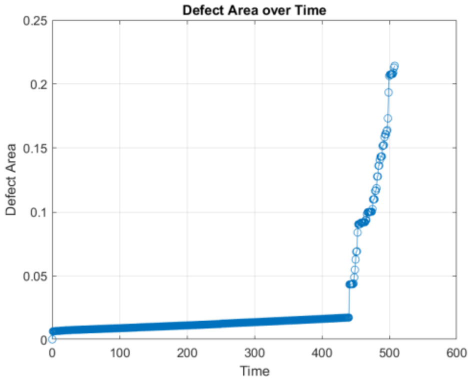
Offline

Anomaly detection



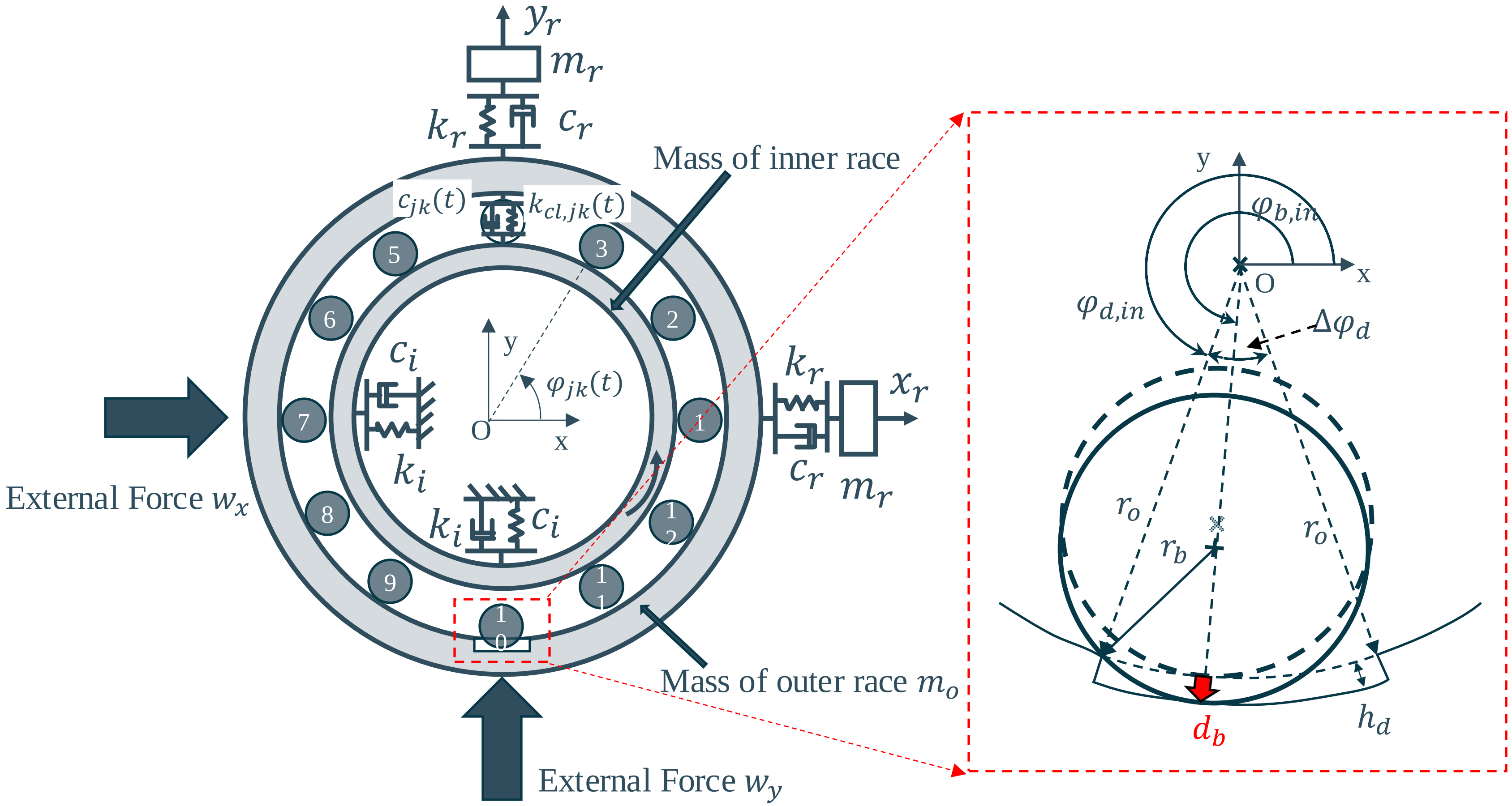
The degradation stage





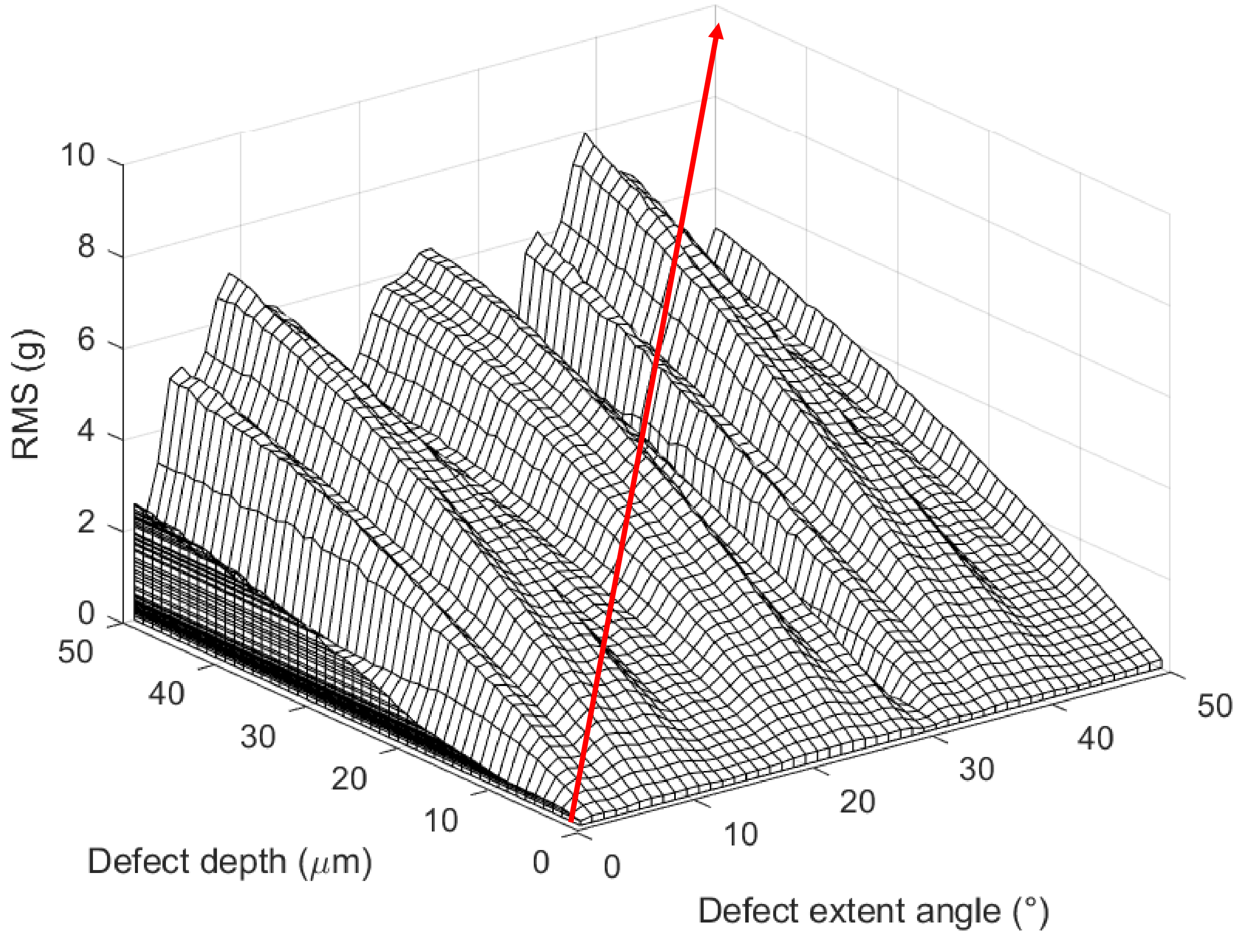
Bearing physical model

$$\boldsymbol{M}\ddot{\boldsymbol{x}}(t) + \boldsymbol{C}\dot{\boldsymbol{x}}(t) + \boldsymbol{K}\boldsymbol{x}(t) + \boldsymbol{f_c}(t) + \boldsymbol{f_d}(\boldsymbol{t}) = \boldsymbol{w}$$



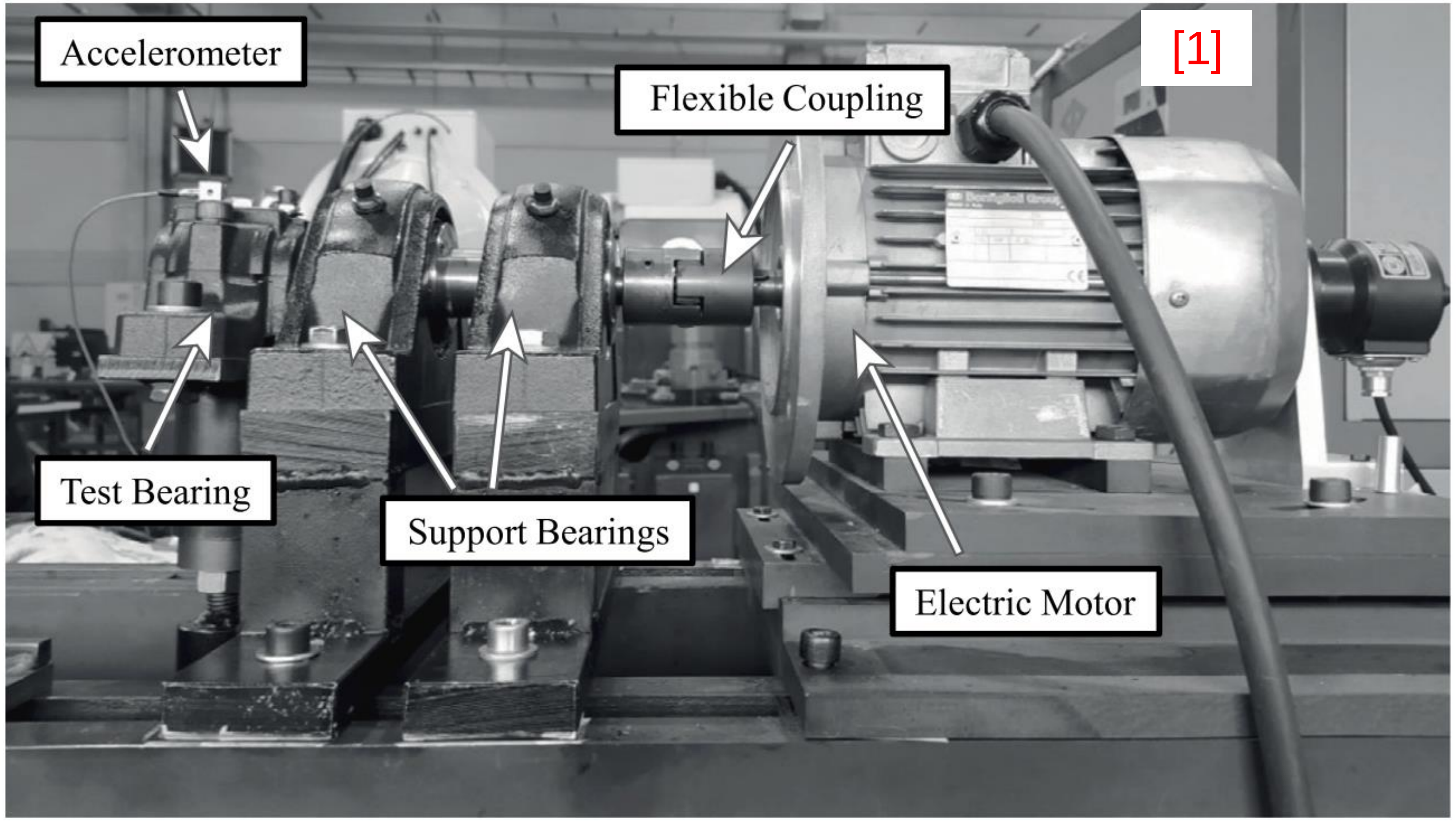
1. Determine the size of the defect.
2. Calculate the displacement of the ball when it enter in and get out of the defect.
3. Calculate the relative displacement of the whole system including inner race, outer race and resonator.
4. Calculate the contact force $\boldsymbol{f_c}(t)$ and the damping force $\boldsymbol{f_d}(\boldsymbol{t})$.
5. Based on Runge-Kutta methods, the function can be solved.

RMS MAP



1. Set the first defect area as zero.
2. Based on the observed RMS acquired from the measurements, find all the possible RMS value based on a tolerance error.
3. Find one combination of defect extent angle and defect depth where the distance from the combination point to the last one point is minimum. (Principle of Minimum Energy).
4. Remove any points which are smaller than the defect depth or extent angle.
5. Repeat 2-4, until the data acquisition is end.

Data

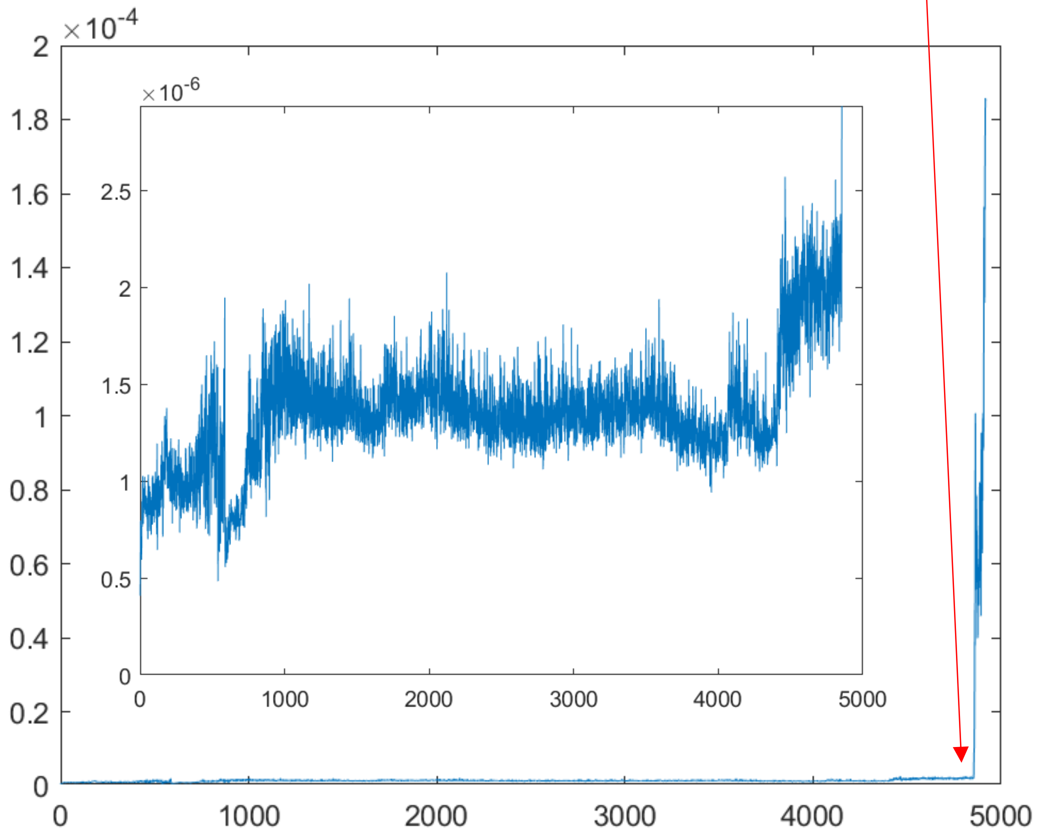


1. Speed: 2400 RPM
2. Force: 4 kN
3. Sampling frequency: 25.6 kHz
4. Duration: 5s
5. BPFO: 195.6021 Hz

[1] Gabrielli, A., Battarra, M., Mucchi, E., & Dalpiaz, G. (2024). Physics-based prognostics of rolling-element bearings: The equivalent damaged volume algorithm. Mechanical Systems and Signal Processing, 215, 111435. https://doi.org/10.1016/j.ymssp.2024.111435

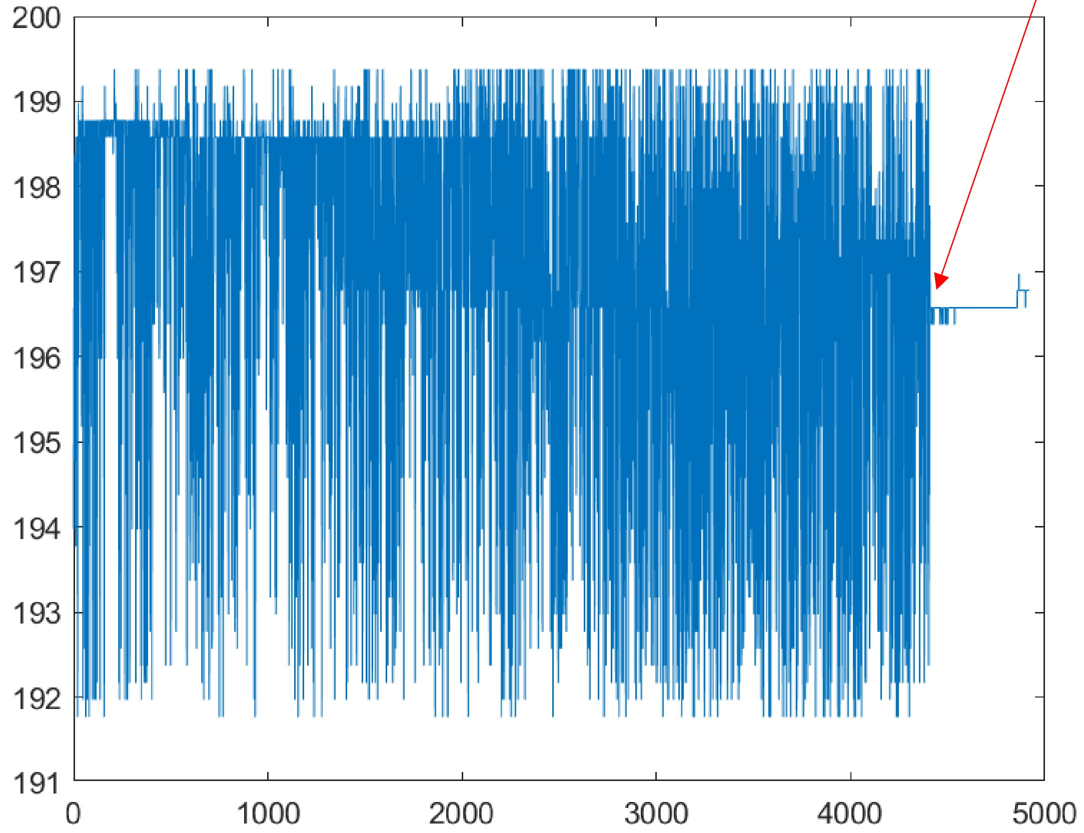
Anomaly detection

4840 (x 5 min)



The variation of maximum amplitude within [BPFO*0.98, BPFO*1.02] over time

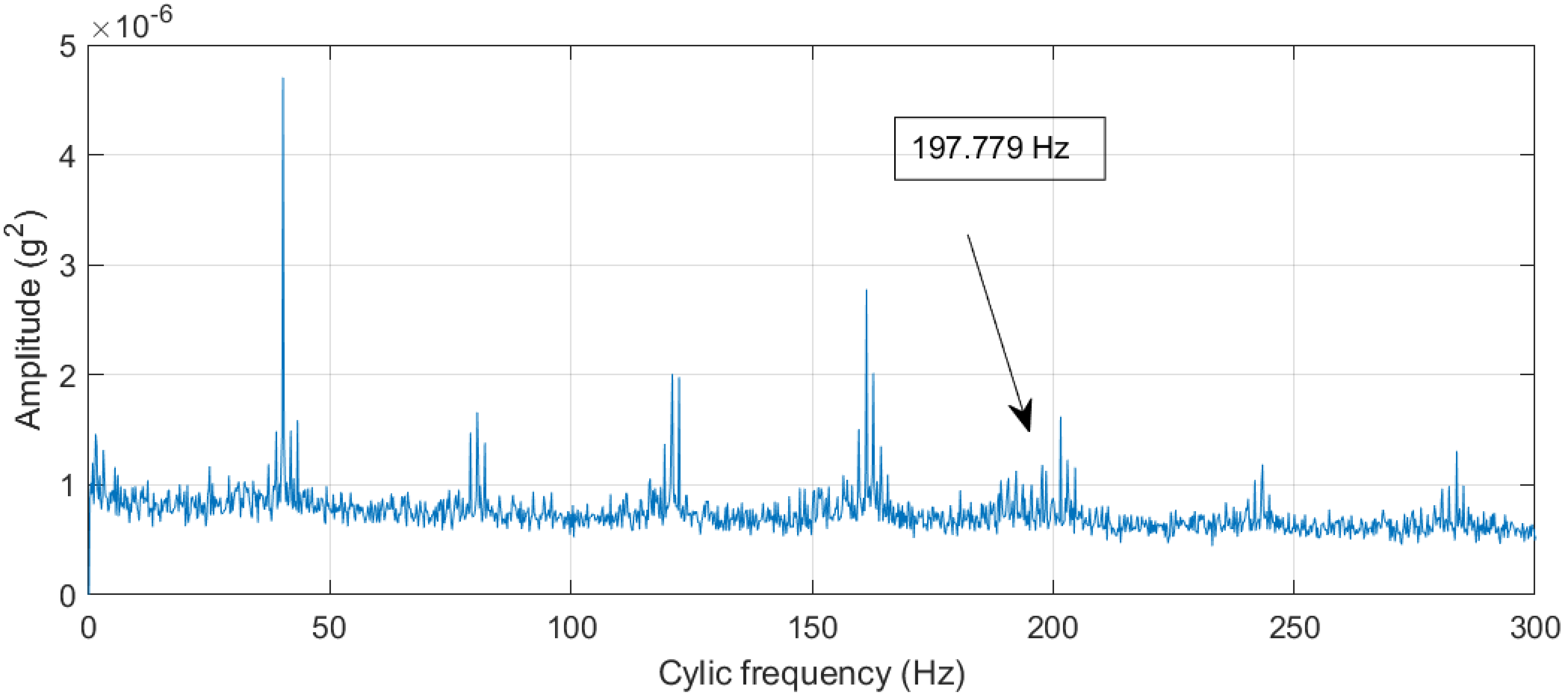
4408 (x 5 min)



The maximum frequency within [BPFO*0.98, BPFO*1.02] in EES

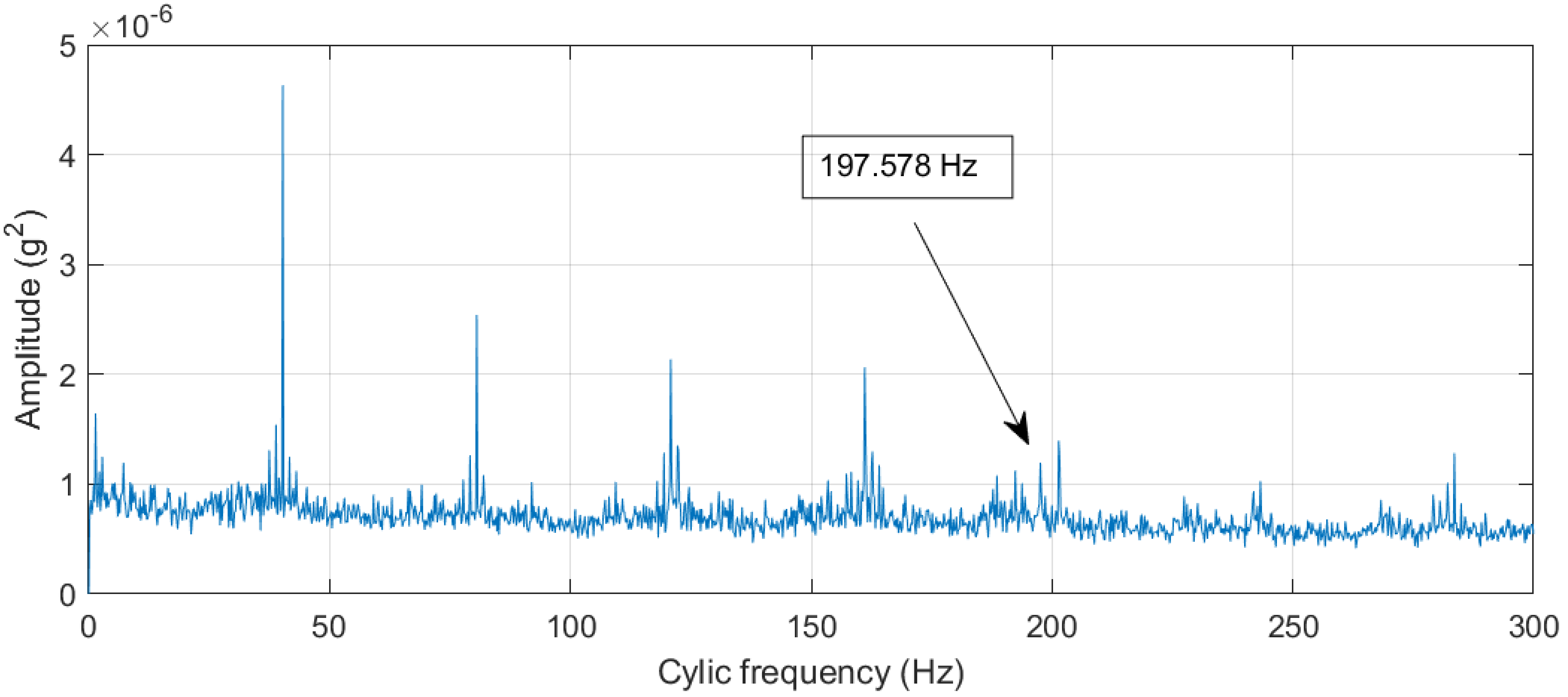
Anomaly detection

EES at the 4406 time stamp



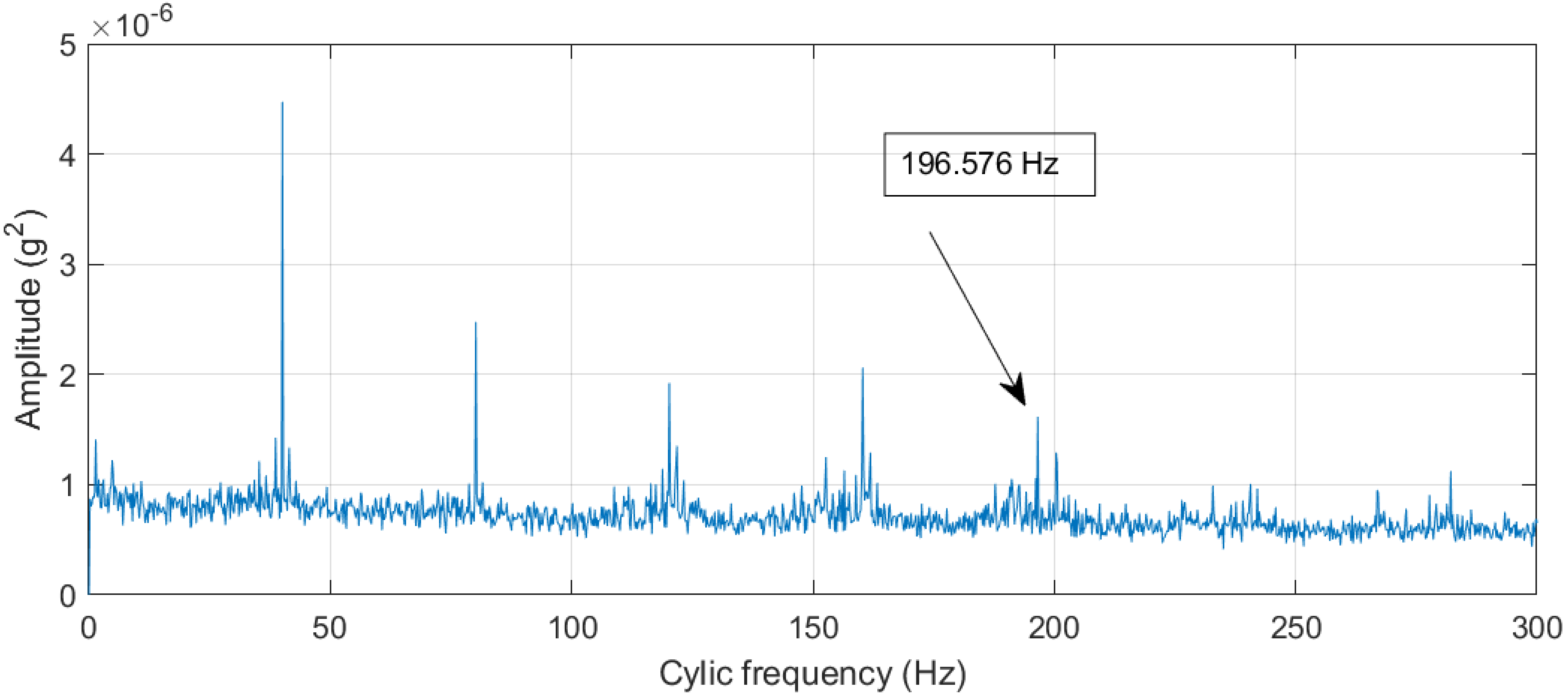
Anomaly detection

EES at the 4407 time stamp



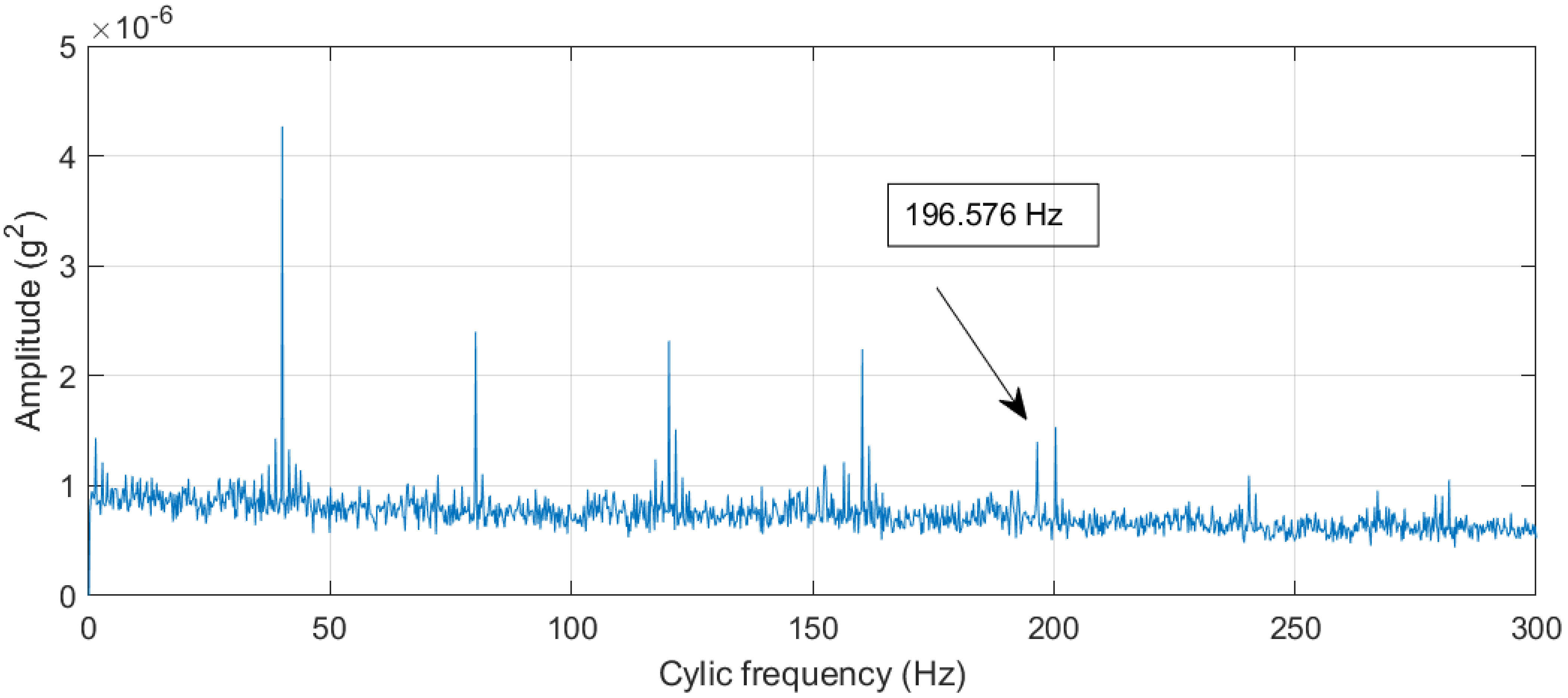
Anomaly detection

EES at the 4408 time stamp

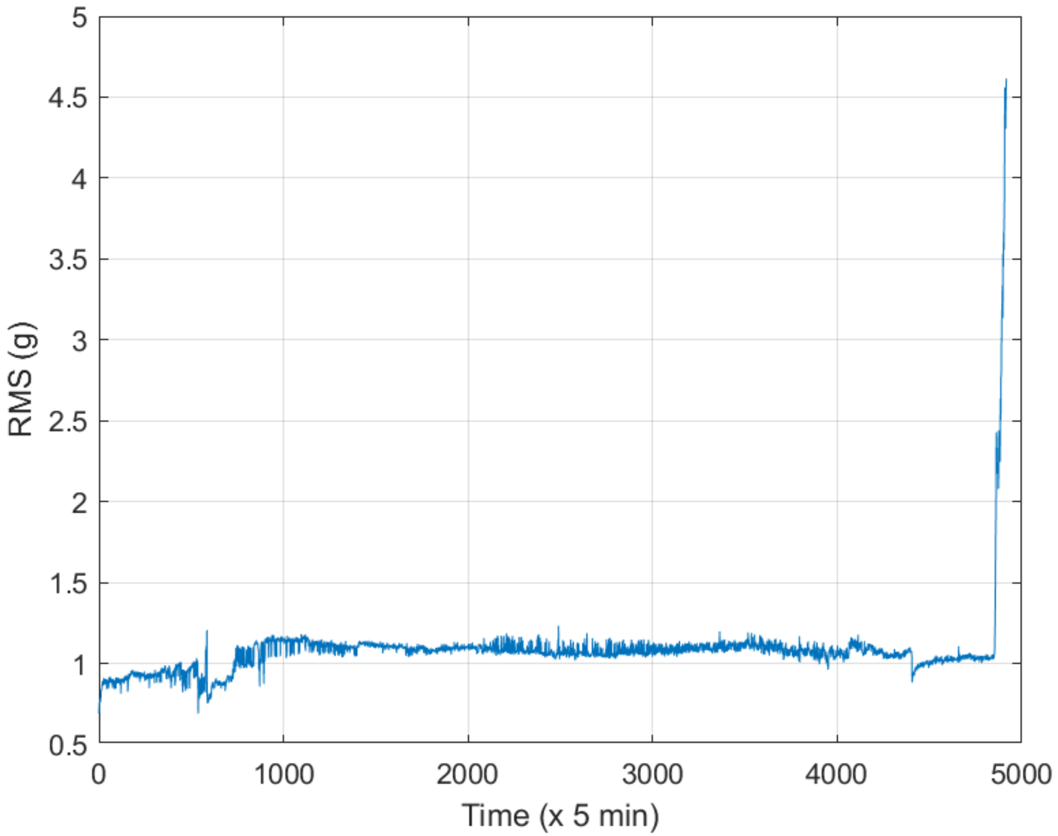


Anomaly detection

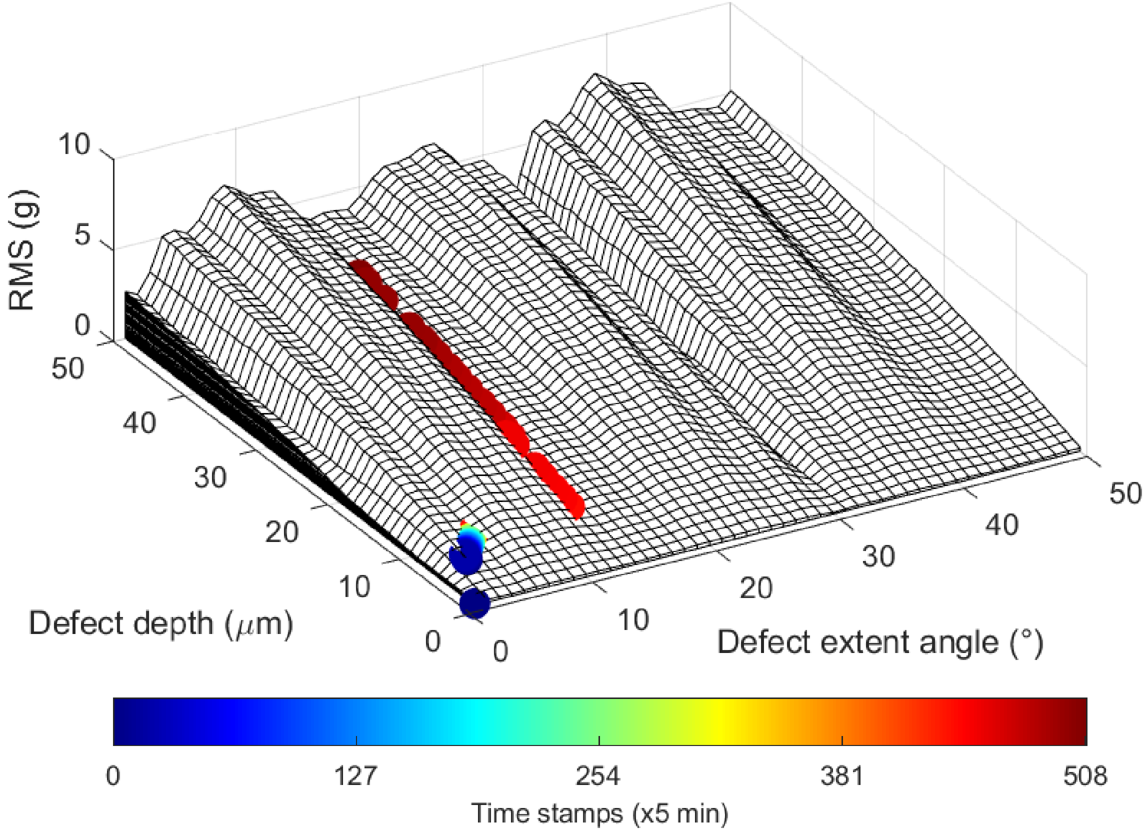
EES at the 4409 time stamp



RMS Map Search

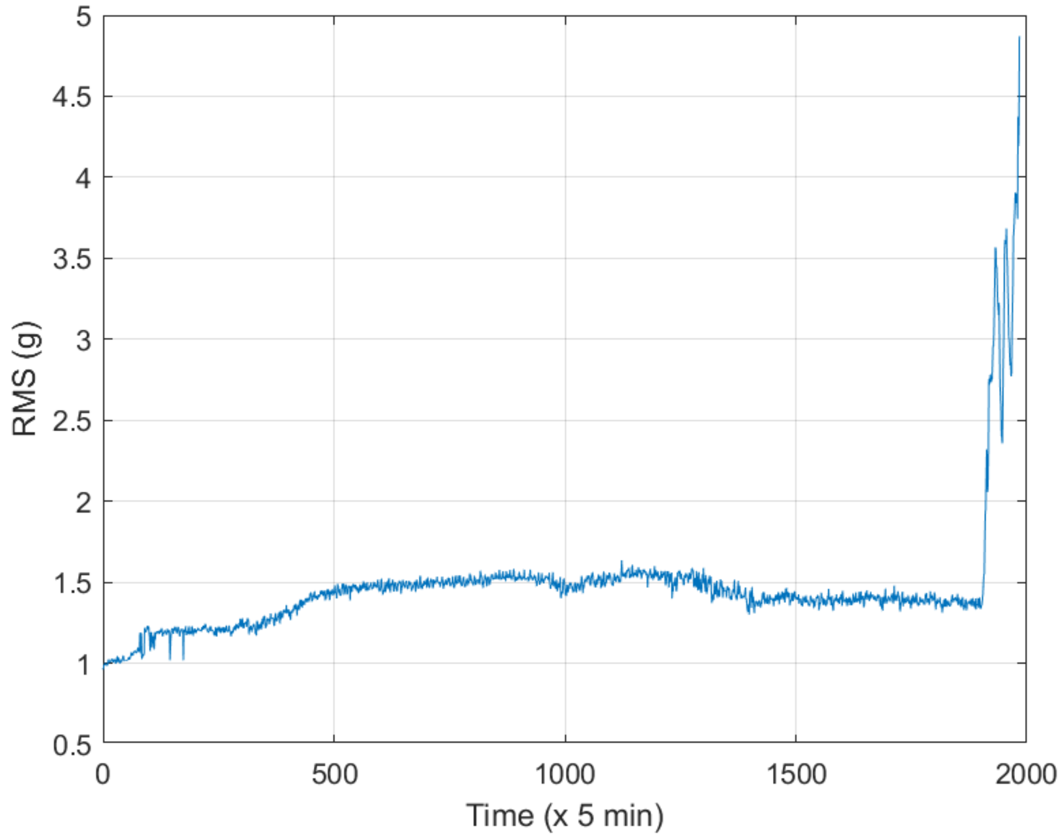


RMS of bearing 1 from measurements

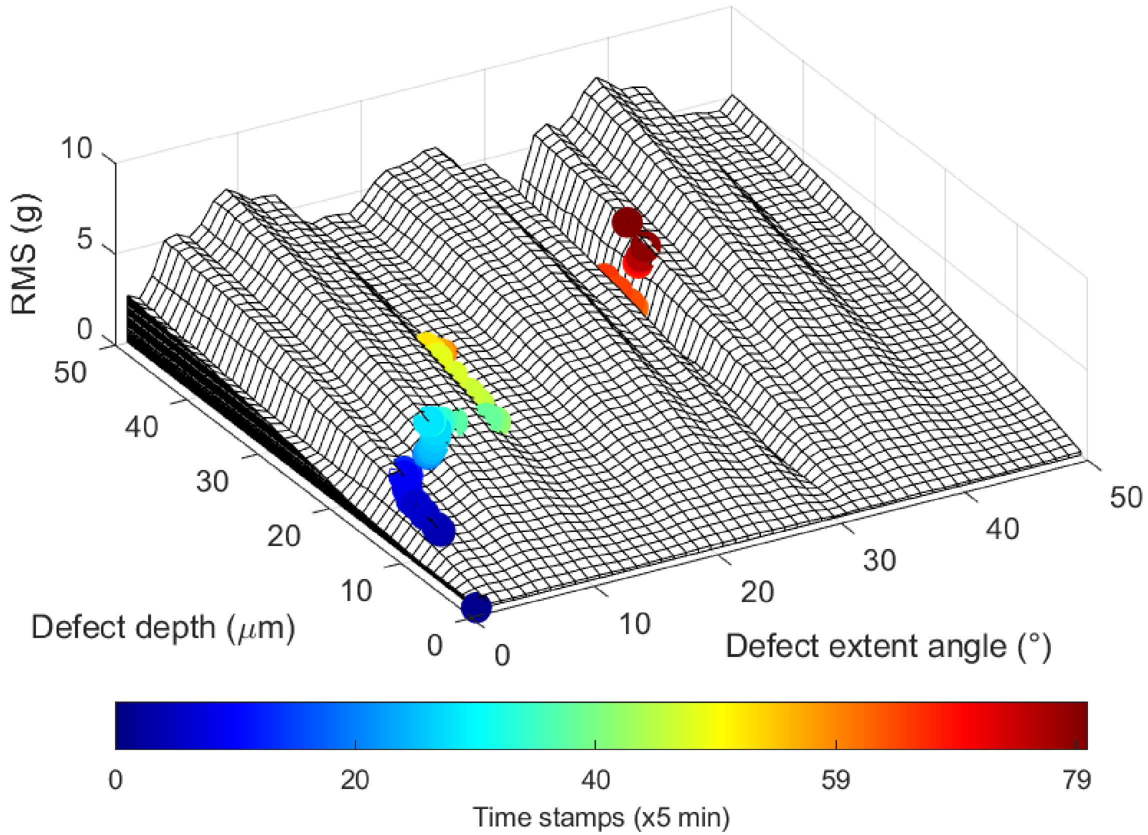


Quasi-Defect size search

RMS Map Search

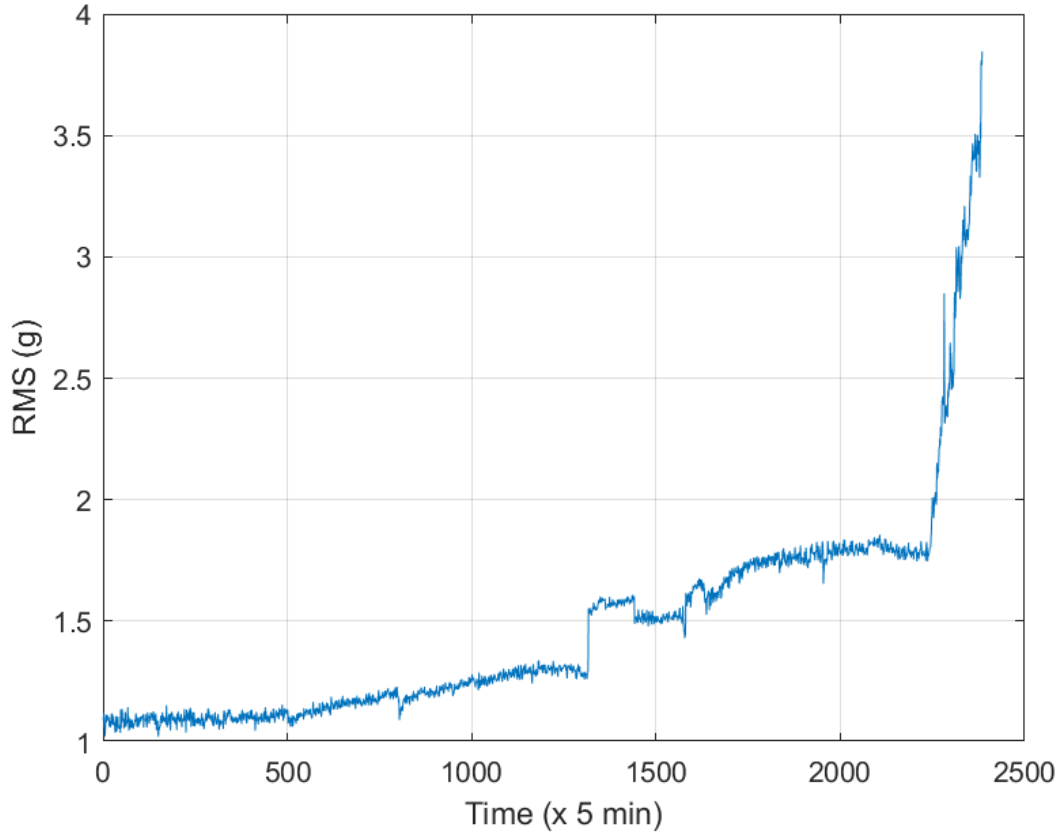


RMS of bearing 2 from measurements

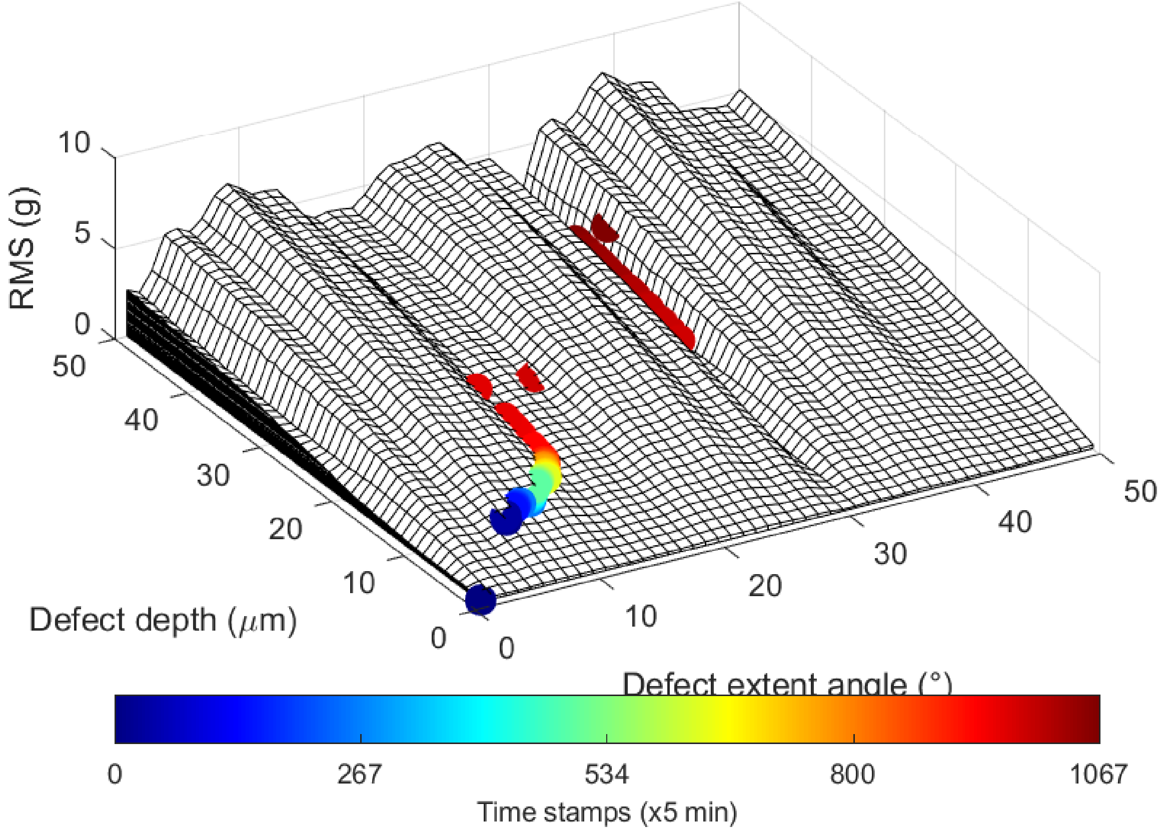


Quasi-Defect size search

RMS Map Search

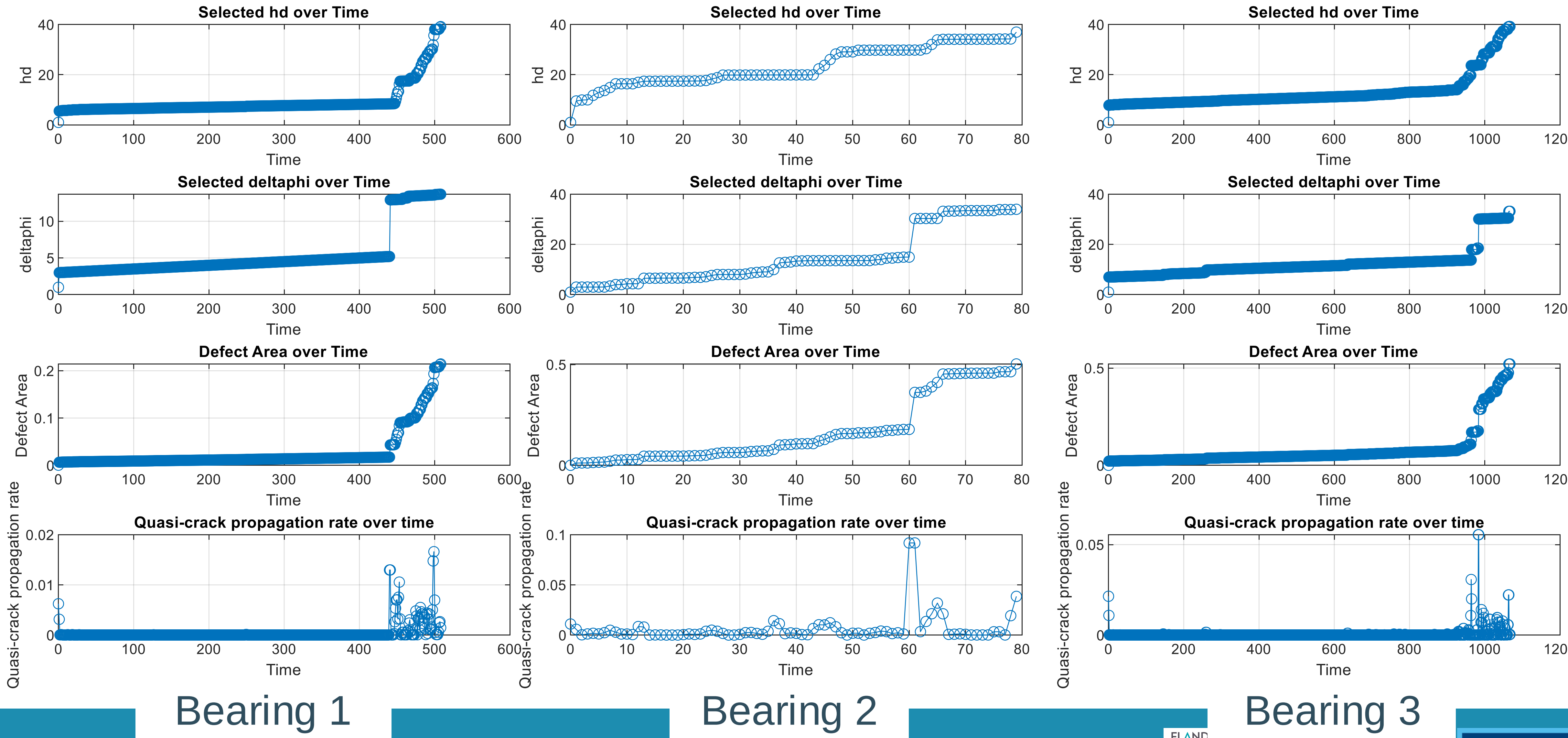


RMS of bearing 3 from measurements



Quasi-Defect size search

Quasi-defect size propagation



Monitoring – Availability of data ???

- If data is available
  - At high volumes
  - Including all possible fault types
  - Including all possible operating conditions, e.g. speeds, loads, temperatures
  - Being correctly labeled based on real ground truth
- Then a Machine Learning (ML) / Deep Learning (DL) model can be built:
  - End – to – End solution
  - Fault Detection / Diagnosis / Prognosis

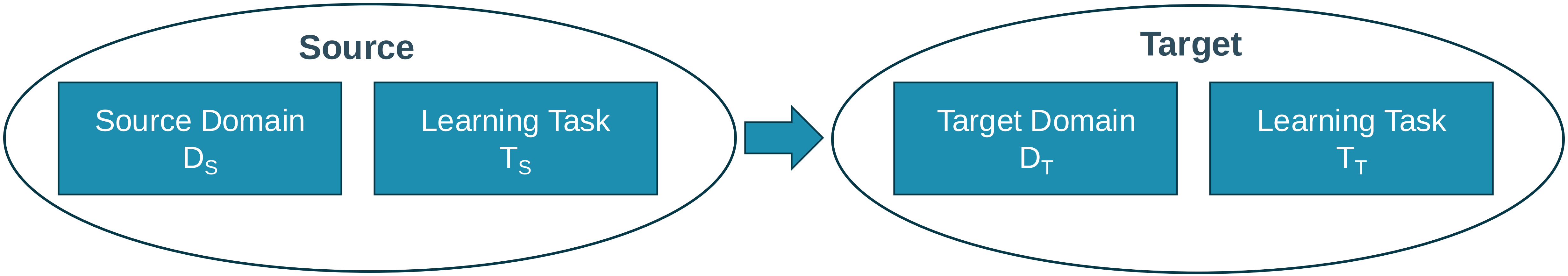
- The availability of data was promised (Big Data Era) but we are not yet there

Limitations of ML & DL

- Despite their benefits ML and DL techniques suffer several limitations:
  1. They are based on the assumption that both training and testing data are drawn from the same distribution
     - In real-world applications this **is not** necessarily **the case**
  2. They require a significant amount of historical labeled based on ground truth healthy and faulty data, covering the full life of the machine, all possible failure modes, operating and environmental conditions
     - In real-world applications **this is not feasible**
  3. The **computational cost** starting from scratch for each operating condition, for each failure mode, for each unit **is high**

Transfer Learning as a possible solution

Transfer Learning aims to improve the learning of the target predictive function $\boldsymbol{f_T(.)}$ in the Target Domain $\mathbf{D_T}$ using the knowledge captured at the Source Domain $\mathbf{D_S}$ and the Source learning Task $\mathbf{T_S}$



Taxonomy of Transfer Learning (TL)

	Domains	Tasks
Traditional ML & DL	$D_S = D_T$	$T_S = T_T$
Transductive TL	$D_S \neq D_T$	$T_S = T_T$
Inductive TL	$D_S = D_T$ OR $D_S \neq D_T$	$T_S \neq T_T$
Unsupervised TL	$D_S = D_T$ OR $D_S \neq D_T$	$T_S \neq T_T$

	Feature spaces	Labels
Homogenous Learning	$X_S = X_T$	$Y_S = Y_T$
Heterogeneous Learning	$X_S \neq X_T$ or/and	$Y_S \neq Y_T$

Taxonomy of Transfer Learning (TL)

- **Closed-set TL:** The domains have identical features spaces and labels ($X_S = X_T$, $Y_S = Y_T$)
- **Partial TL:** The label space of the target domain is a subset of the source domain's label space ($Y_T \subset Y_S$)
- **Open-set TL:** The label space of the source domain is a subset of the target domain's label space. ($Y_S \subset Y_T$), e.g. a new fault mode arises in the target domain which is not included in the fault mode set of the source domain.
- **Universal TL:** There is no prior knowledge about the label space of the source and target domain ($Y_S \neq Y_T$)

Taxonomy of Transfer Learning (TL)

- Due to the high economic and labor expenses in real-world industries, it is generally difficult for a single source to collect enough high-quality data to build an efficient data-driven predictive maintenance model in the target domain.
  - **Single Source Domain.** This technique relies on knowledge from a single source.
  - **Multiple Source Domain.** The multiple source domain transfer learning techniques transfer the knowledge from different multiple, but relevant sources.

Taxonomy of Transfer Learning (TL)

- **Transfer in the Same Machine (TSM):** The source and target domain data are collected on the same machine but under different operational conditions or working environments.
- **Transfer across Different related Machines (TDM):** The source and target domain data are collected on different but related machines (significant data distribution discrepancy).
- **Transfer from Laboratory to Real Machine (TLRM):** The source domain data is obtained from a laboratory machine. Modeling failure modes in the lab are simpler, safer, and cheaper than gathering faulty data from a real-world machine.
- **Transfer from Virtual to Real Machine (TVRM):** The source domain data is collected from a machine’s virtual model to provide transferable maintenance information for the target machine (limited real historical faulty data).

Transfer Learning as a possible solution

- **Case 1:** Available data from the Source, No data from the Target
  - A model is trained using the source data, transferred at the target and used directly at the incoming data.
    - Can be used for transfer between machines & between operating conditions
    - Usually low performance **WHY?**
    - Due to distribution change or domain shift between the two domains
- **Case 2:** Available data from the Source, Limited labeled data from the Target
  - A model is trained using the source data, transferred at the target and the last layers are retrained keeping frozen the first ones.
    - Can be used for transfer between machines & between operating conditions
    - Higher performance compared to Case 1 but very case dependent

Transfer Learning as a possible solution

- **Case 3:** Available data from the Source, Limited labeled or unlabeled data from the Target
  - Domain Adaptation techniques
  - A model is trained using the source data and the limited unlabeled data
    - Can be used for transfer between machines & between operating conditions
    - High performance

Blade ice detection in wind turbines

2 Wind Turbines:

- Turbine # 15 11/01/2015 - 01/01/2016
- Turbine # 21 11/01/2015 - 12/01/2015

Turbine #15

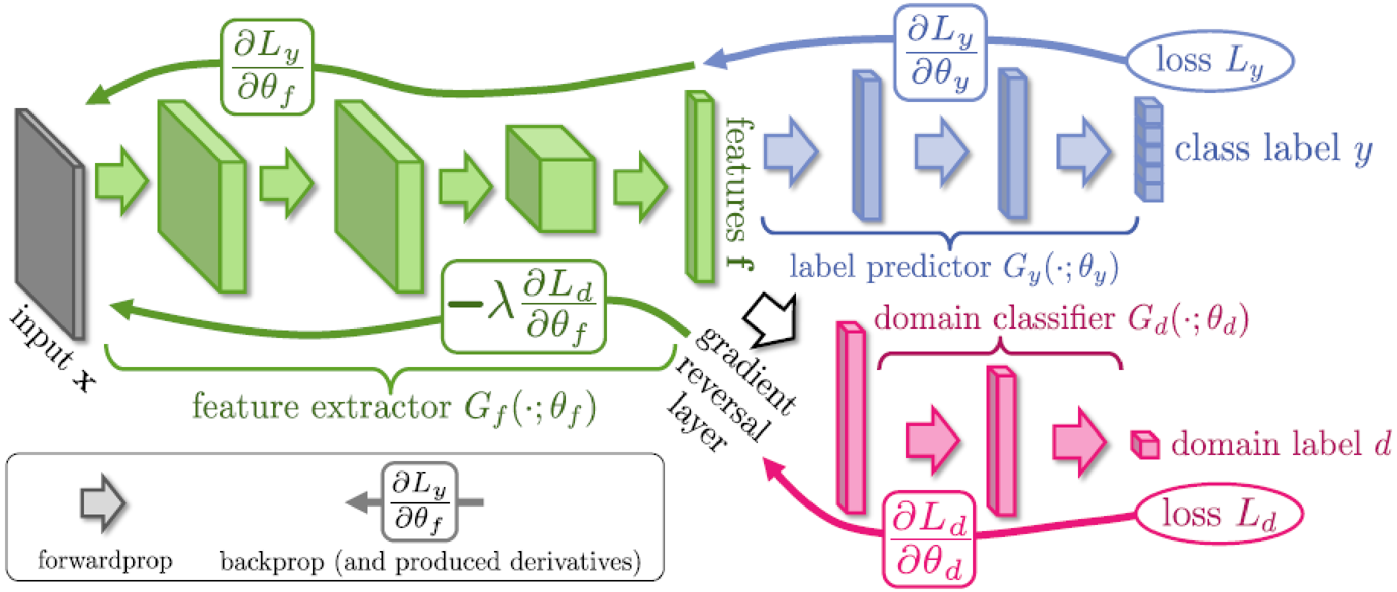
	Number of samples	Ratio
Normal	350255	88.92%
Icing	23892	6.07%
Unlabeled	19739	5.01%

Turbine #21

	Number of samples	Ratio
Normal	168930	88.68%
Icing	10683	5.58%
Unlabeled	10926	5.73%

Number	Description	Number	Description
1	Wind speed	2	Generator speed
3	Grid side active power	4	Wind direction
5	Mean wind direction	6	Yaw position
7	Yaw speed	8	Pitch1 angle
9	Pitch2 angle	10	Pitch3 angle
11	Pitch1 speed	12	Pitch2 speed
13	Pitch3 speed	14	Pitch motor 1 temperature
15	Pitch motor 2 temperature	16	Pitch motor 3 temperature
17	X-direction acceleration	18	Y-direction acceleration
19	Environment temperature	20	Cabin temperature
21	Ng5 1 temperature	22	Ng5 2 temperature
23	Ng5 3 temperature	24	Ng5 1 charger DC current
25	Ng5 1 charger DC current	26	Ng5 1 charger DC current
27	Data group identification		

Domain-Adversarial Neural Network (DANN)



Feature Extractor		
1	Convolution layer + BN layer	1*3/16, ReLU
2	Convolution layer + BN layer	1*3/32, ReLU
3	Maxpooling layer	Stride: 2
4	Convolution layer + BN layer	1*3/64, ReLU
5	Convolution layer + BN layer	1*3/128, ReLU
6	Maxpooling layer	Stride: 2
7	Convolution layer + BN layer	1*3/256, ReLU
8	Adaptive average pooling layer	1
Discriminator		
1	Fully connected layer	128, Leaky ReLU
2	Fully connected layer	64, Leaky ReLU
3	Fully connected layer	2, Sigmoid
Classifier		
1	Fully connected layer	100, ReLU
2	Fully connected layer	2

Ice detection in wind turbines

2D input (10*27)

Batch size: 128; Epoch: 50; 5 experiments
Wind turbine #15 (44181 75%, 14726 25%)
Wind turbine #21 (20641 75%, 6880 25%)

	#15 -> #21			#21 -> #15		
	2D_CNN (test on Source only)	2D_CNN (test on Target only)	2D_DANN	2D_CNN (test on Source only)	2D_CNN (test on Target only)	2D_DANN
Accuracy	0.7499 ± 0.0241	0.7156 ± 0.0265	0.8354 ± 0.0500	0.7678 ± 0.0653	0.8512 ± 0.0113	0.8675 ± 0.0068
Precision	0.7212 ± 0.0393	0.7268 ± 0.0612	0.8274 ± 0.0905	0.8080 ± 0.0924	0.8818 ± 0.0138	0.8835 ± 0.0291
Recall	0.9536 ± 0.0468	0.8795 ± 0.0786	0.9418 ± 0.0625	0.8383 ± 0.1255	0.8666 ± 0.0388	0.8978 ± 0.0489
F1	0.8195 ± 0.0091	0.7913 ± 0.0120	0.8759 ± 0.0306	0.8141 ± 0.0615	0.8735 ± 0.0131	0.8893 ± 0.0103
Score	0.7023 ± 0.0392	0.6672 ± 0.0524	**0.8068** ± 0.0716	0.7471 ± 0.0803	0.8476 ±0.0052	**0.8604** ± 0.0055

$$\text{Score} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

Ice detection in wind turbines

1D input (1*27)

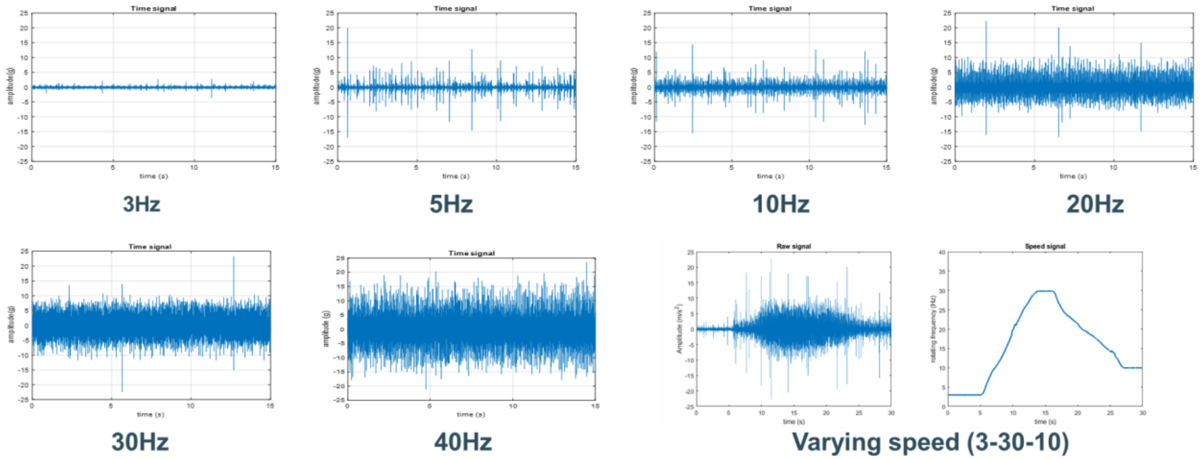
Batch size: 512; Epoch: 50; 5 experiments
Wind turbine #15 (543303 75%, 181099 25%)
Wind turbine #21 (134677 75%, 44891 25%)

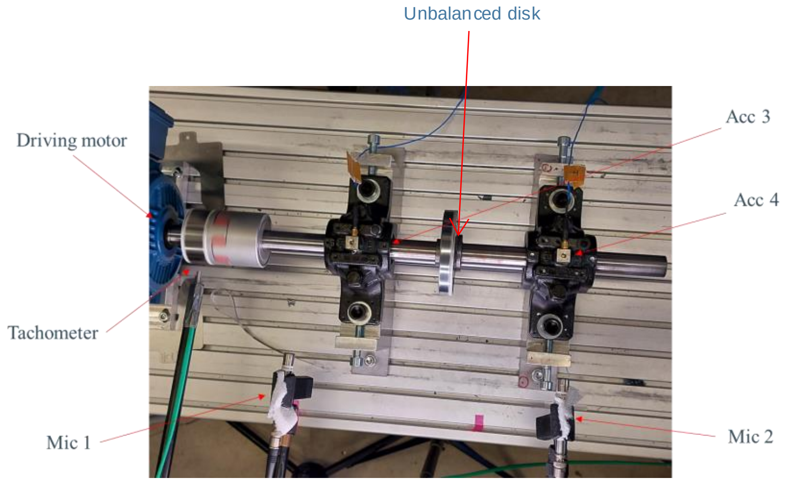
	#15 -> #21			#21 -> #15		
	1D_CNN (test on Source only)	1D_CNN (test on Target only)	1D_DANN	1D_CNN (test on Source only)	1D_CNN (test on Target only)	1D_DANN
Accuracy	0.9461 ± 0.0092	0.9388 ± 0.0093	0.9441 ± 0.0378	0.8984 ± 0.0759	0.9013 ± 0.0310	0.9349 ± 0.0252
Precision	0.9597 ± 0.0071	0.9527 ± 0.0065	0.9672 ± 0.0084	0.9712 ± 0.0138	0.9756 ± 0.0019	0.9761 ± 0.0007
Recall	0.9838 ± 0.0067	0.9839 ± 0.0138	0.9737 ± 0.0408	0.9205 ± 0.0957	0.9175 ± 0.0351	0.9538 ± 0.0269
F1	0.9715 ± 0.0049	0.9679 ± 0.0051	0.9701 ± 0.0212	0.9425 ± 0.0456	0.9454 ± 0.0181	0.9647 ± 0.0141
Score	0.6886 ± 0.0547	0.6029 ± 0.0550	**0.7242** ± 0.0710	0.7336 ± 0.0793	0.7906 ± 0.0110	**0.8058** ± 0.0137

$$\text{Score} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

Transfer Learning among different conditions

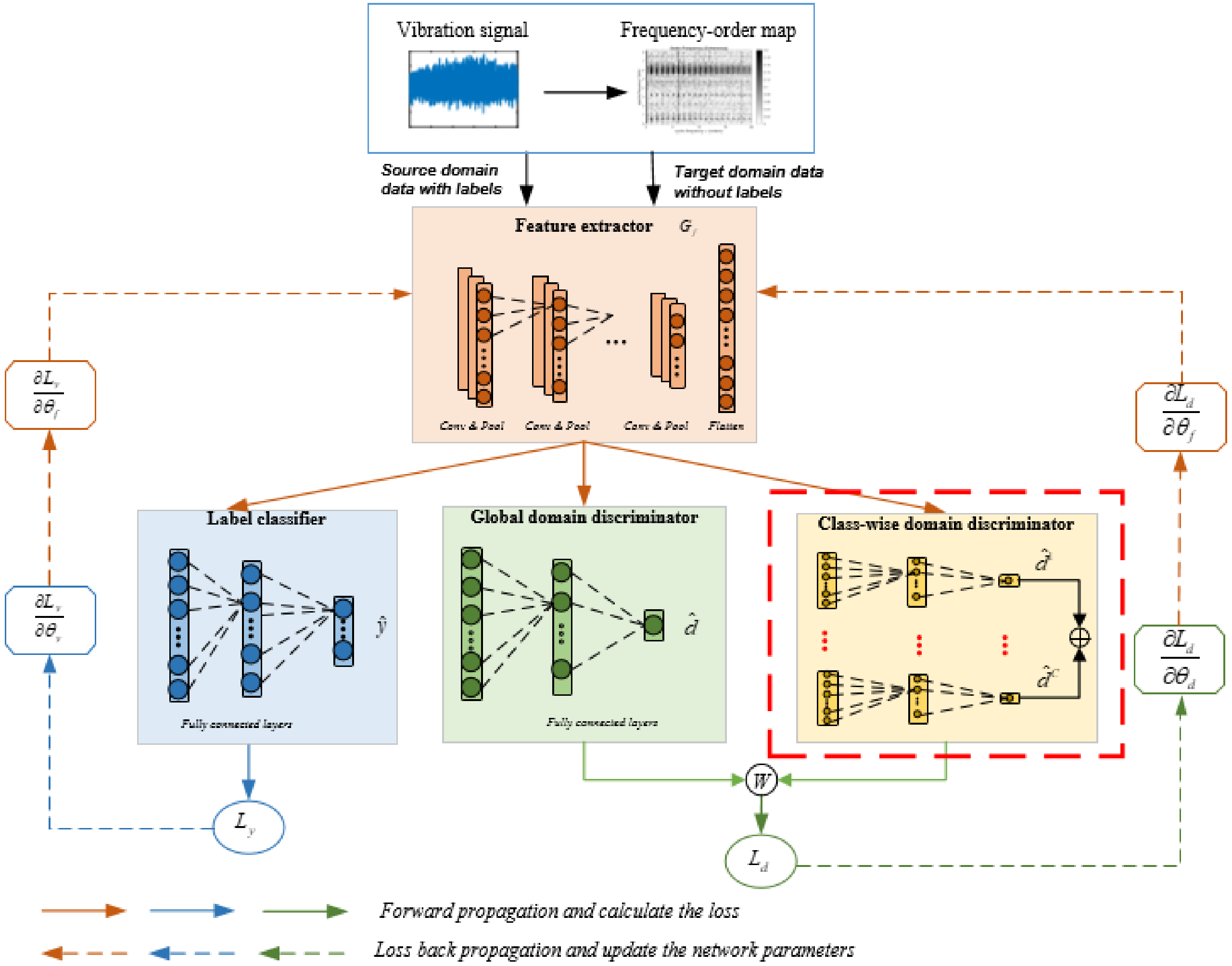
- LVL KU Leuven Test Rig



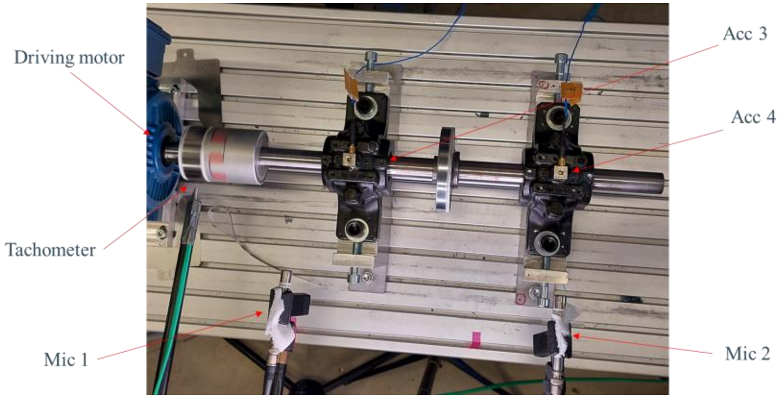


Balanced load: No bolts on the disk
Unbalanced load: One bolt on the disk

Dynamic Adversarial Adaptation Network (DAAN)



Transfer Learning among different load conditions



LVL Drivetrain (***Balanced load***) ⟷ LVL Drivetrain (***Unbalanced load***)

Accuracy [%]	CNN (Without TL)	DANN (With TL)	DAAN (With TL)
Unbalanced load → Balanced load	90.84	92.04	**99.33**
Balanced load → Unbalanced load	88.77	89.80	**98.99**

Transfer Learning among different speed conditions

LVL Drivetrain (**Speed A**) ⟷ LVL Drivetrain (**Speed B**)

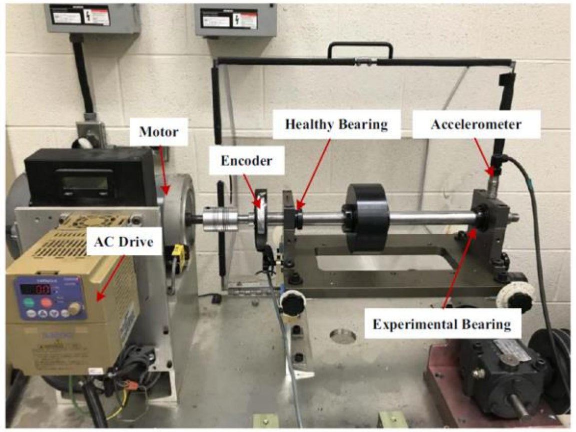
Speed	CNN (Without TL)	DANN (With TL)	DAAN (With TL)
3-5	31.11	32.50	**88.89**
3-10	47.96	40.81	**97.96**
3-20	32.97	32.98	**44.68**
3-30	34.78	32.61	**55.43**
3-40	32.32	**32.32**	31.31
5-3	64.04	65.17	**89.88**
5-10	65.31	64.29	**97.95**
5-20	30.85	67.02	**84.04**
5-30	32.61	67.39	**82.61**
5-40	32.32	**67.67**	66.66
10-3	29.21	29.21	**80.90**
10-5	66.66	64.44	**88.89**
10-20	**100.00**	**100.00**	85.10
10-30	67.39	68.48	**86.95**
10-40	46.46	67.68	**78.78**

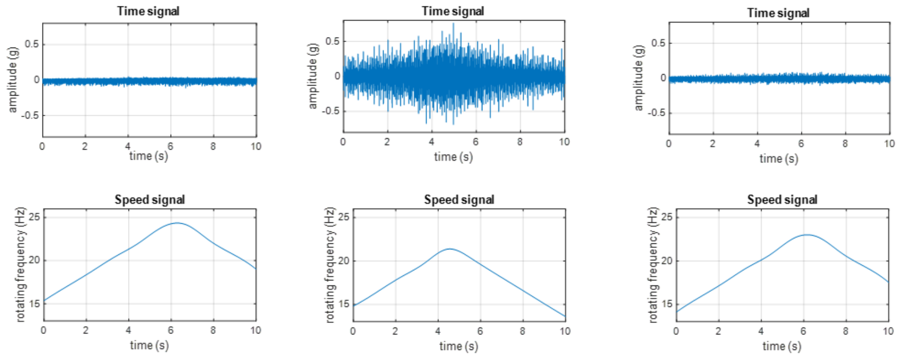
Speed	CNN (Without TL)	DANN (With TL)	DAAN (With TL)
20-3	29.21	29.21	**52.81**
20-5	51.11	62.22	**87.77**
20-10	68.36	56.12	**97.96**
20-30	92.39	**94.57**	86.96
20-40	67.67	70.71	**96.97**
30-3	35.96	34.83	**53.93**
30-5	51.11	62.22	**87.78**
30-10	52.04	51.02	**97.95**
30-20	**92.55**	89.36	85.11
30-40	87.88	92.93	**96.97**
40-3	35.96	34.83	**55.06**
40-5	55.56	40.00	**76.67**
40-10	66.33	44.89	**97.96**
40-20	**94.68**	75.53	85.11
40-30	**96.74**	90.22	86.96

Transfer Learning among test rigs/machines

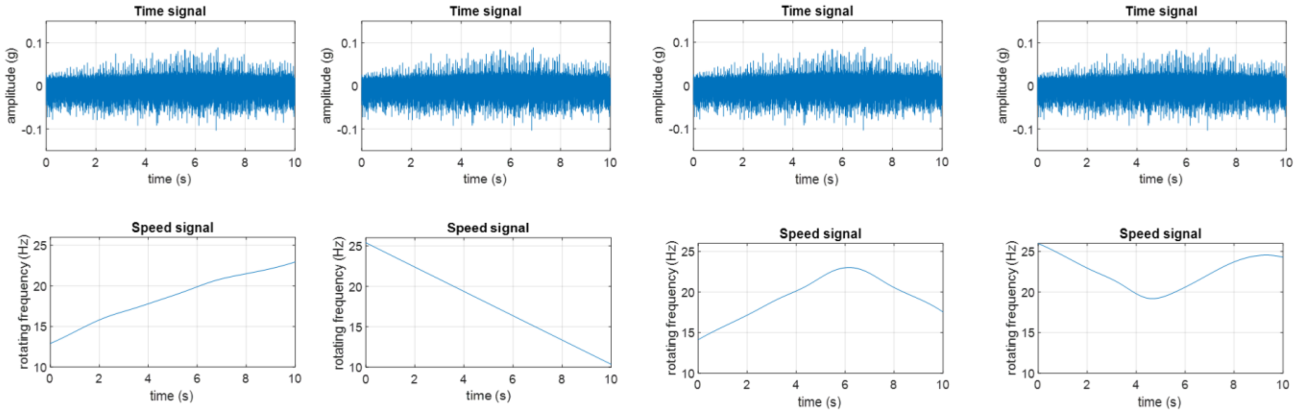
Huang Dataset, Department of Mechanical Engineering,

University of Ottawa, Ottawa, Ontario, Canada





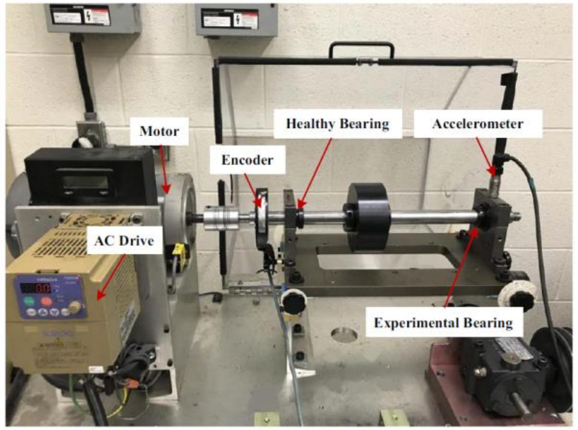
Healthy (Speed C) **Inner race fault** (Speed C) **Outer race fault** (Speed C)



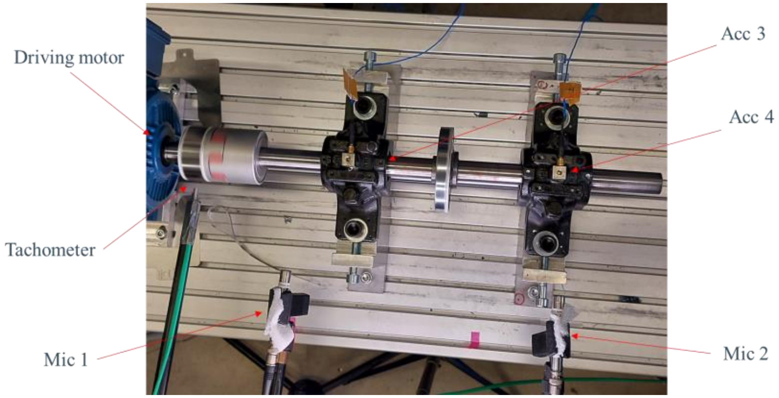
Outer race fault (Speed **A**) Outer race fault (Speed **B**) Outer race fault (Speed **C**) Outer race fault (Speed **D**)

Transfer Learning among test rigs/machines

- When fault types match between the two test rigs



2 classes: healthy; inner race fault;

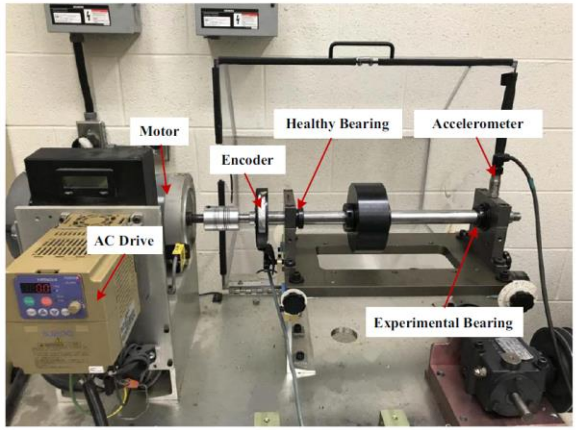


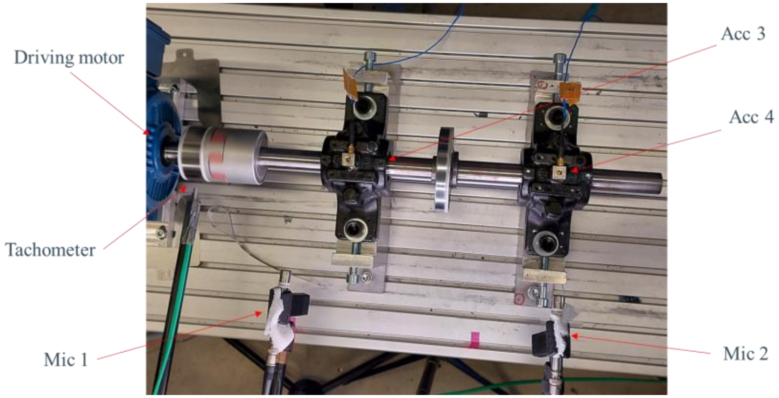
2 classes: healthy; inner race fault;

Accuracy [%]	CNN (Without TL)	DANN (With TL)	DAAN (With TL)
Huang → LVL	52.56	52.56	**88.78**
LVL → Huang	50.00	50.00	**99.54**

Transfer Learning among test rigs/machines

- When fault types do not match between the two test rigs



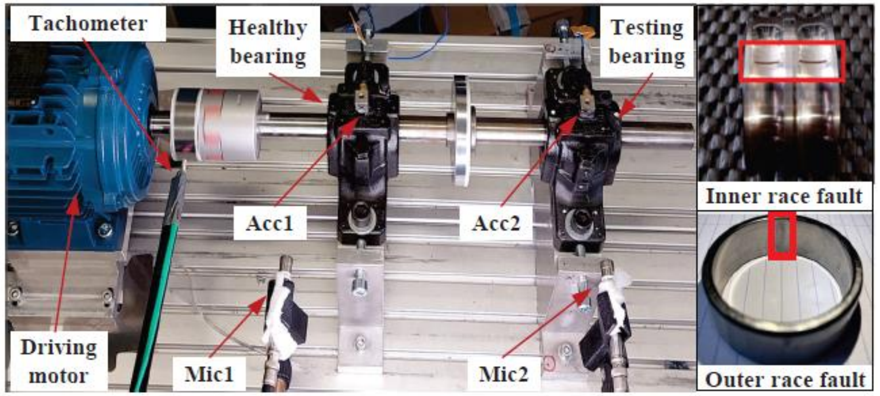


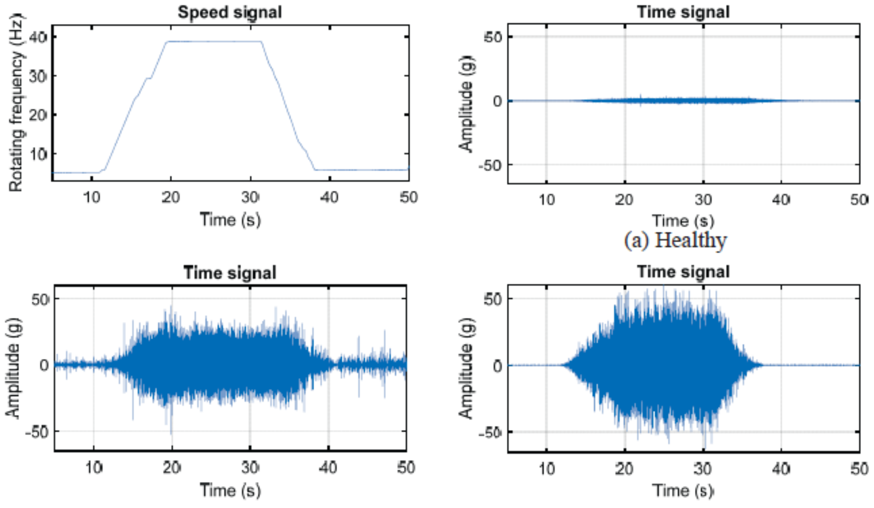
3 classes: healthy; inner race fault; outer race fault

3 classes: healthy; inner race fault 1; inner race fault 2

Accuracy [%]	CNN (Without TL)	DANN (With TL)	DAAN (With TL)
Huang → LVL	35.27	53.37	**55.88**
LVL → Huang	33.70	**69.91**	64.97

Transfer Learning among test rigs/machines

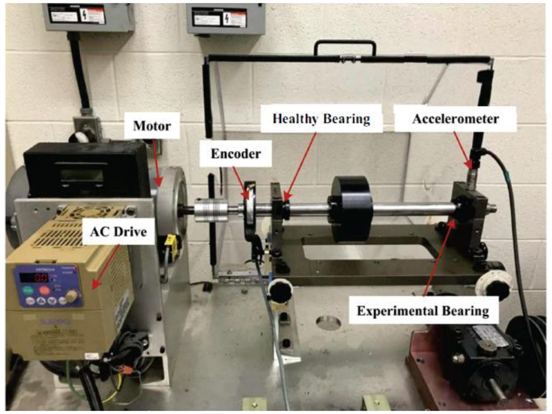


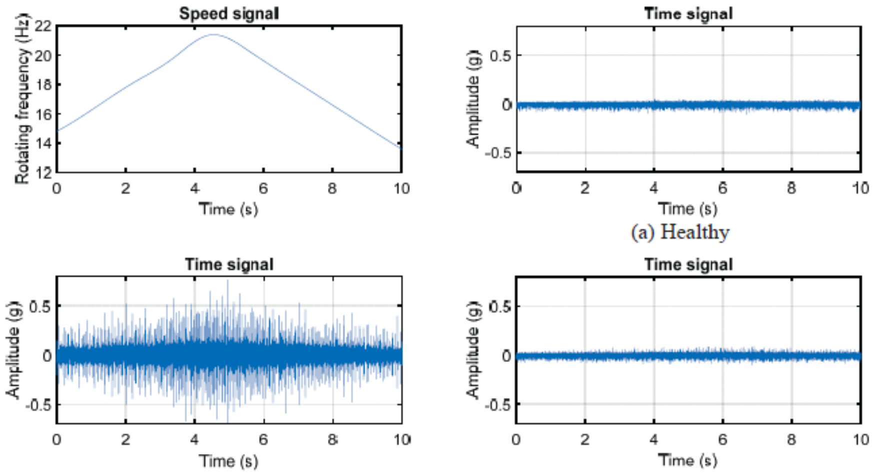


(a) Healthy

(b) Inner race fault

(c) Outer race fault





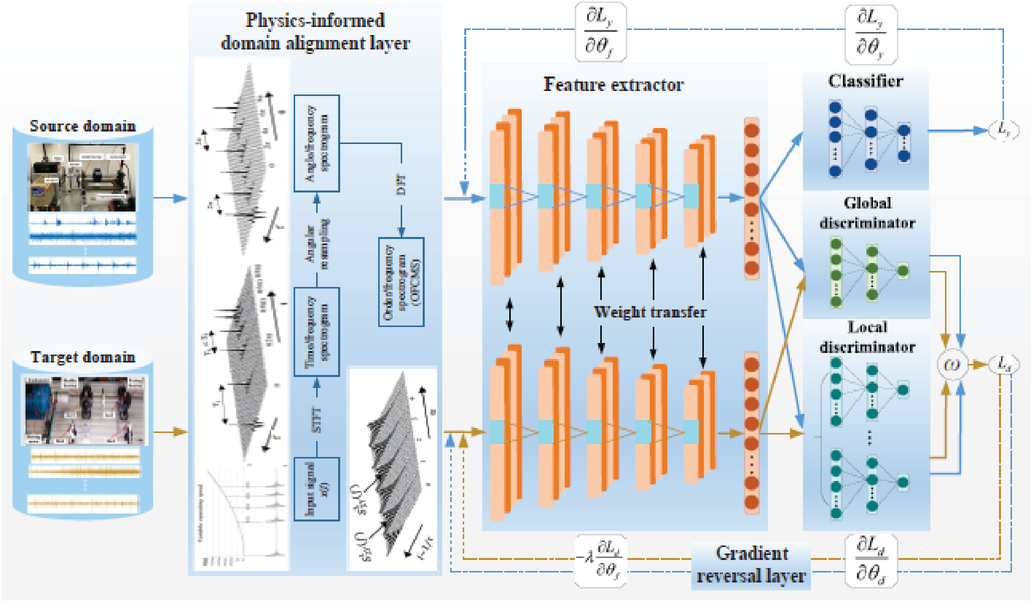
(a) Healthy

(b) Inner race fault

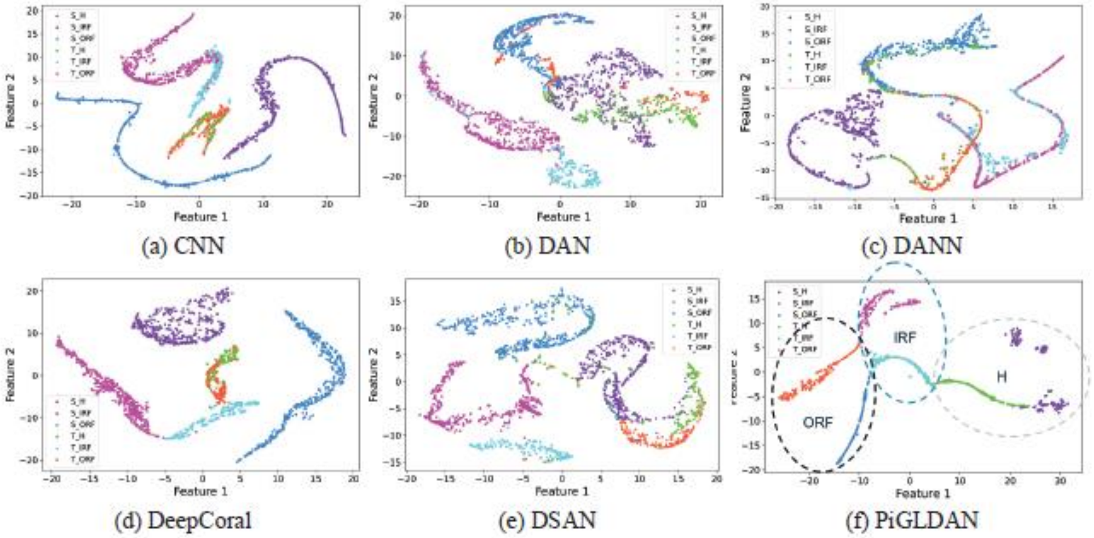
(c) Outer race fault

Transfer Learning among test rigs/machines

Physics-Informed Global Local Domain Adaptation Network

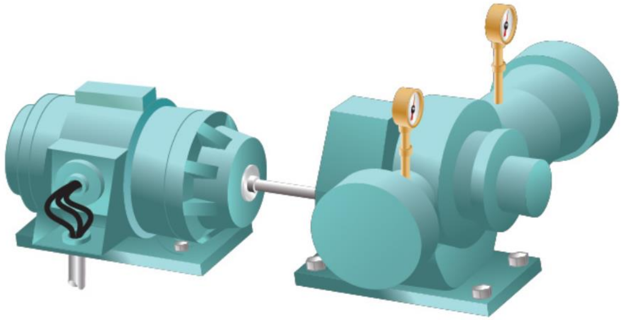


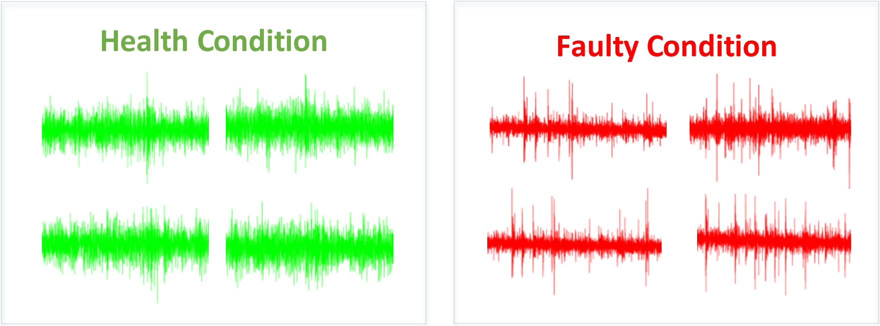
Tasks	Metrics	CNN	DAN	DANN	DeepCoral	DSAN	PiGLDAN
LVL→Huang	Accuracy (%)	64.72±2.66	64.05±20.18	62.61±7.14	75.01±11.11	57.54±23.09	**88.62±9.59**
	F1 score (%)	58.38±7.46	62.31±20.96	54.27±7.75	74.88±10.85	54.93±24.07	**88.55±9.64**
	MCC (%)	54.45±3.18	47.76±30.58	52.79±12.71	63.08±16.73	37.61±34.46	**83.04±14.25**
Huang→LVL	Accuracy (%)	51.66±15.37	63.97±12.62	39.72±29.64	62.86±8.53	60.93±9.16	**70.19±5.96**
	F1 score (%)	43.46±20.40	63.70±12.24	26.38±11.66	60.00±11.19	59.05±11.66	**69.06±6.18**
	MCC (%)	31.25±23.80	46.86±18.90	11.82±15.20	46.17±11.09	42.22±13.58	**58.17±7.90**



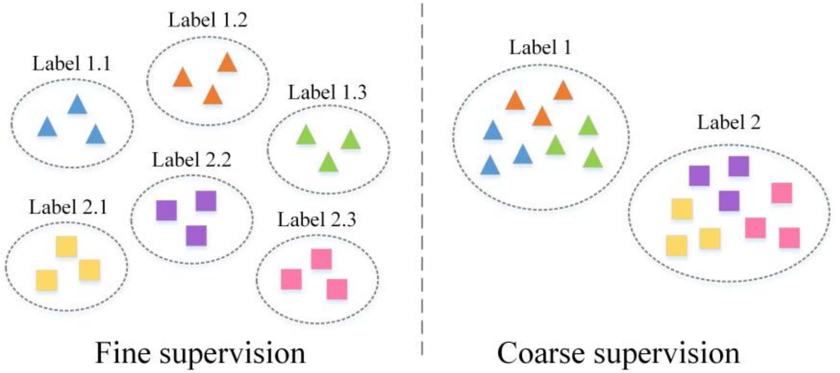
Simulation-driven Domain Adaptation for Rolling Element Bearing Diagnosis

- **Challenge 1:** Insufficient training data in real industry especially for faulty cases



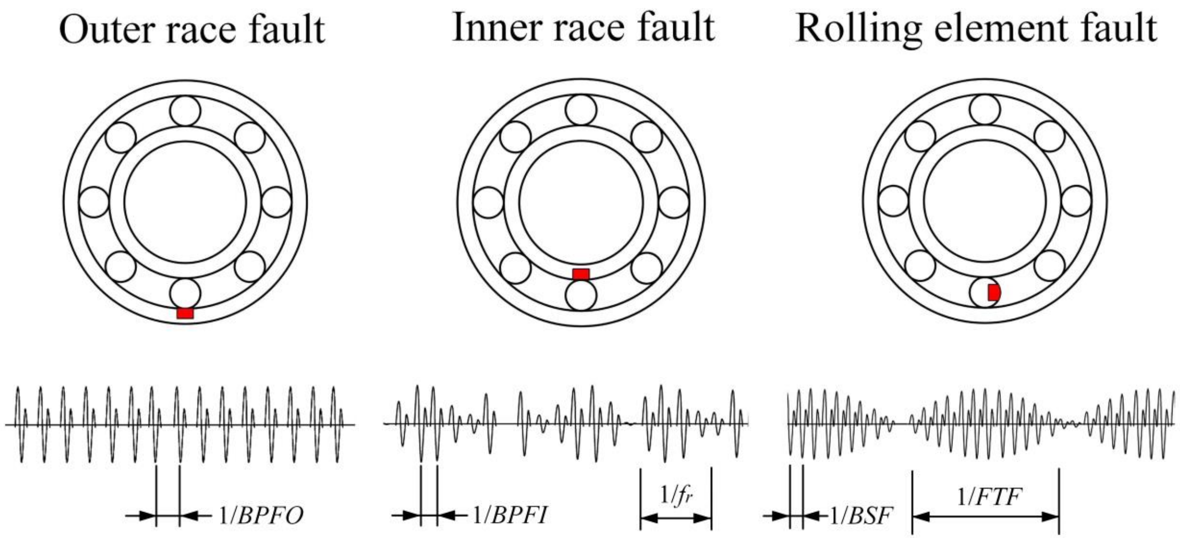


- **Challenge 2:** Category mismatch during transfer learning



Simulation model

- A bearing phenomenological model is used to generate faulty bearing vibration responses
- The model could simulate signals with different fault locations



MacFadden bearing model:

$$d(t) = d_o \sum_{k=-\infty}^{\infty} \delta(t - kT_d)$$

$$q(\theta) = q_0 \left(1 - \frac{\epsilon}{2}\left[1 - cos(\theta)\right]\right)^n$$

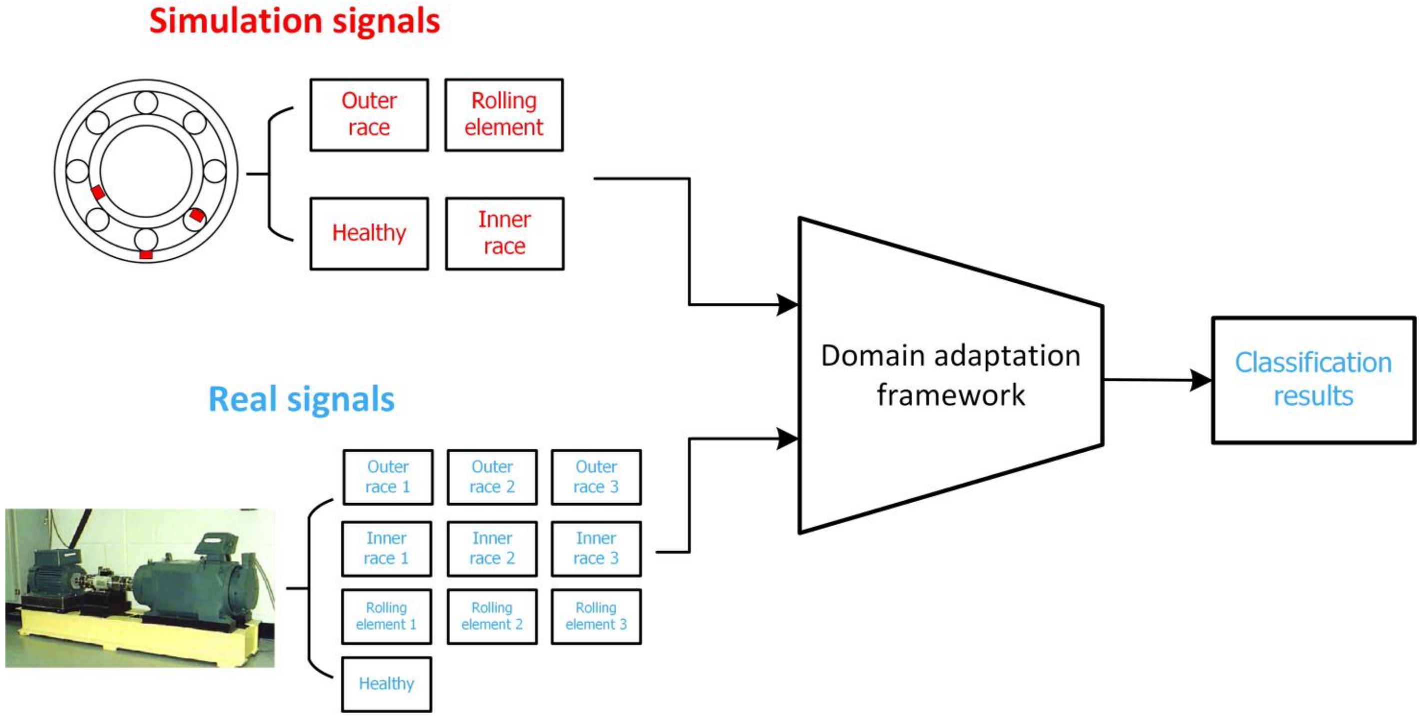
$$q(t) = \begin{cases} q_0 \left(1 - \frac{\epsilon}{2}\left[1 - cos(\theta)\right]\right)^n & \text{for } |\theta| < \theta_{max} \\ 0 & \text{elsewhere} \end{cases}$$

$$e(t) = \begin{cases} e^{\frac{-t}{T_e}} & \text{for } t > 0 \\ 0 & \text{elsewhere} \end{cases}$$

$$v(t) = \left[d(t) \cdot q(t) \cdot a(t)\right] * e(t)$$

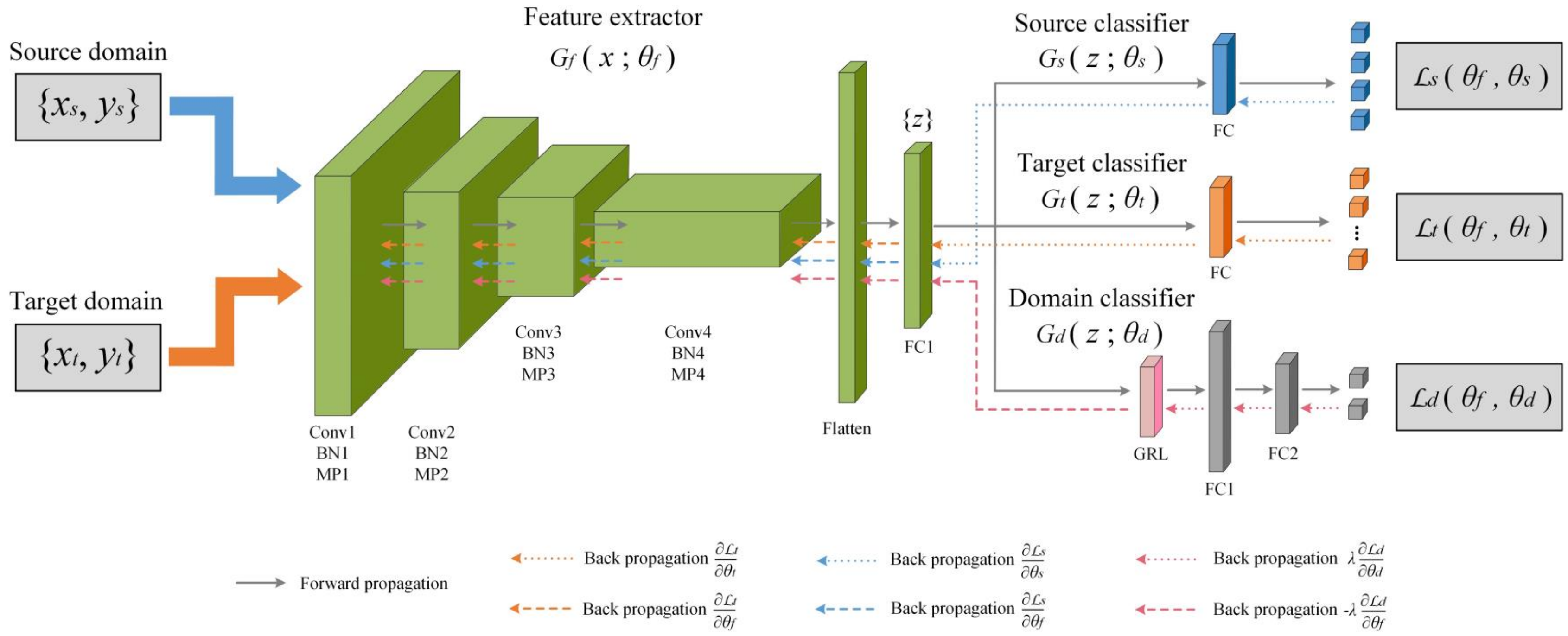
Simulation-driven domain adaptation

- A bearing phenomenological model is utilized to generate simulated signals with coarse labels: **healthy, inner race fault, outer race fault and rolling element fault**
- Real signals are under fine supervision with more categories based on severity, damage distribution, damage type etc.



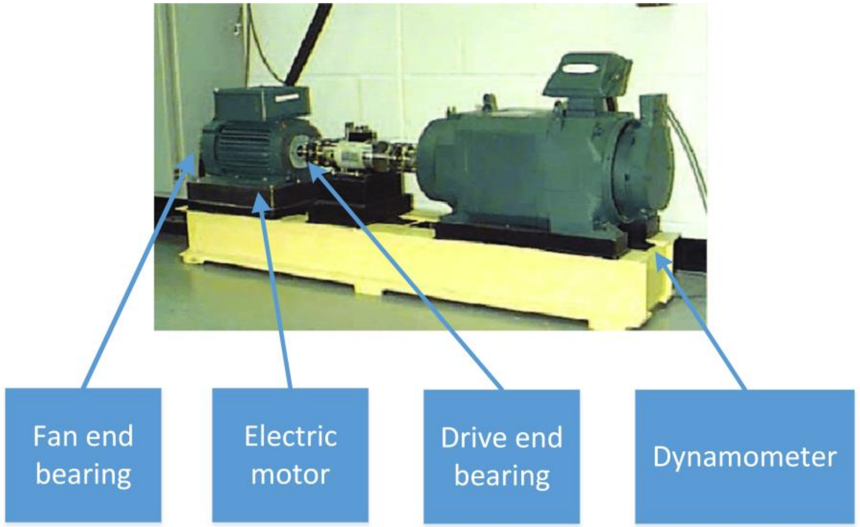
Simulation-driven domain adaptation

- The simulated signals are used as the source domain in the transfer learning model
- A new network architecture is proposed which can simultaneously deal with coarse supervised source and fine supervised target



Application: Case Western Reserve University

- The data is labelled in **10 fine categories** under 4 operating conditions



- Operating conditions

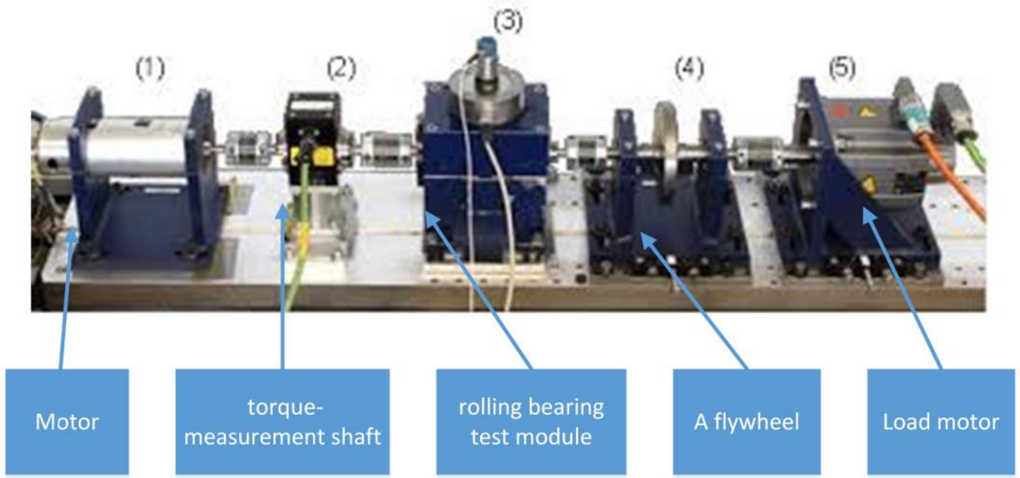
	Condition 1	Condition 2	Condition 3	Condition 4
Speed (rpm)	1797	1772	1750	1730
Load (HP)	0	1	2	3
Total samples	664	664	664	665

- Labels of the dataset

Class label	1	2	3	4	5
Fault type	N	OF	IF	REF	OF
Fault diameter (inch)	0	0.007	0.007	0.007	0.014
Class label	6	7	8	9	10
Fault type	IF	REF	OF	IF	REF
Fault diameter (inch)	0.0014	0.014	0.021	0.021	0.021

Application: Paderborn University

- The data is also labelled in **10 fine categories** under 4 operating conditions



- Operating conditions

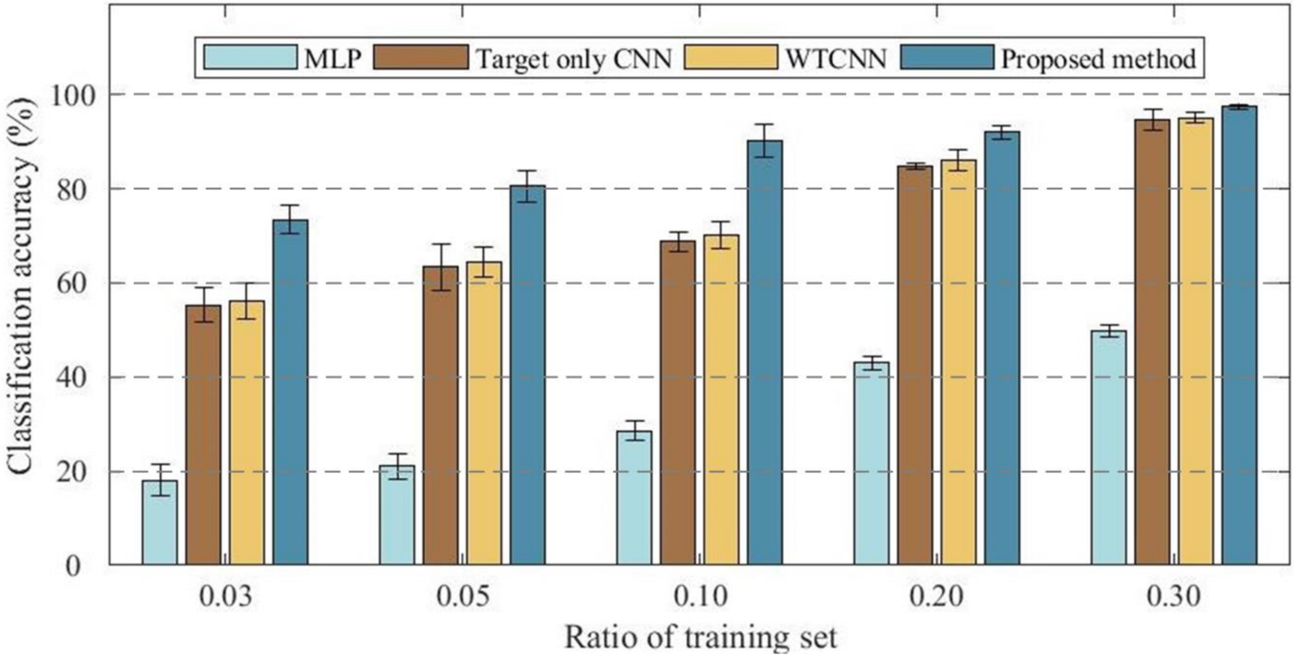
	Condition 1	Condition 2	Condition 3	Condition 4
Speed (rpm)	1500	900	1500	1500
Load Torque (Nm)	0.7	0.7	0.1	0.7
Radial force (N)	1000	1000	1000	400
Total samples	1280	1291	1307	1280

- Labels of the dataset

Class label	1	2	3	4	5
Fault type Bearing code	N K002	OF KA04	OF KA15	OF KA16	OF KA22
Class label	6	7	8	9	10
Fault type Bearing code	OF KA30	IF KI14	IF KI16	IF KI18	IF KI21

Results: CWRU dataset

- Comparison to non-transfer models
  - 3 non-transfer models: MLP, Target only CNN, CNN with 2D inputs (WTCNN)
  - Ratio of training set is selected from **0.03** (20 real samples) to **0.30** (199 real samples)

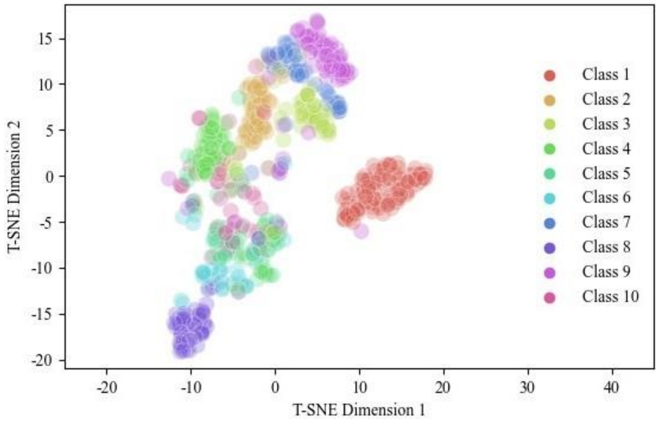


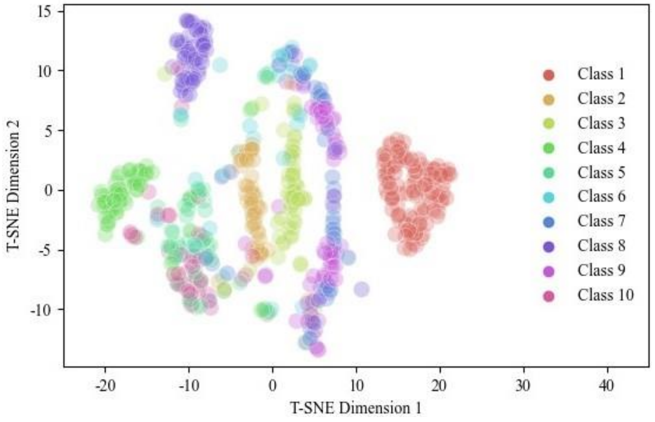
The proposed method outperforms the non-transfer learning models with small ratio of training set using CWRU dataset.

Results: CWRU dataset

- Feature visualization using t-SNE

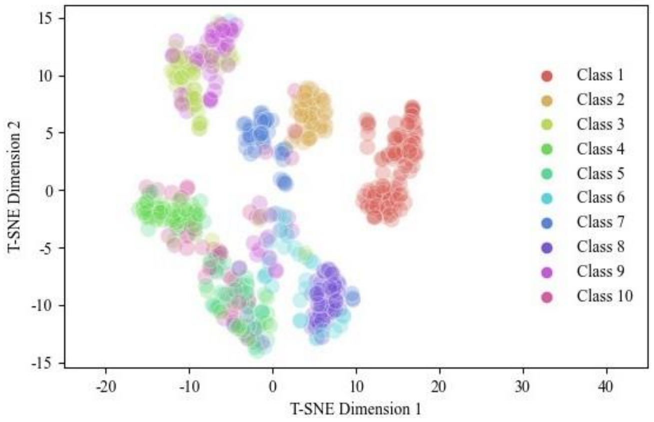
MLP

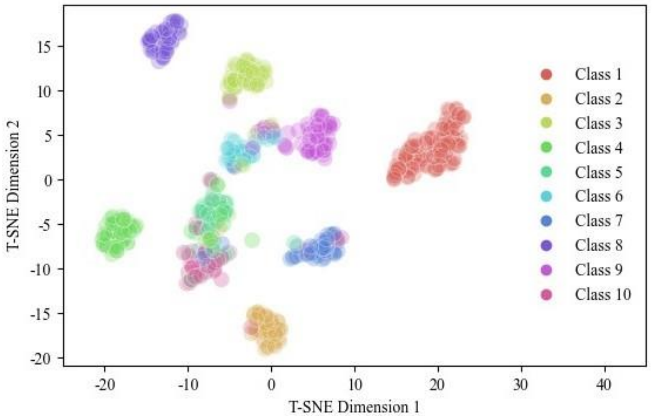




Target only CNN

WTCNN



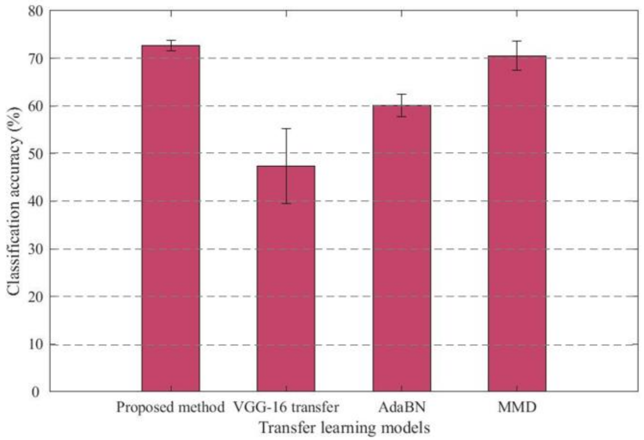


Proposed method

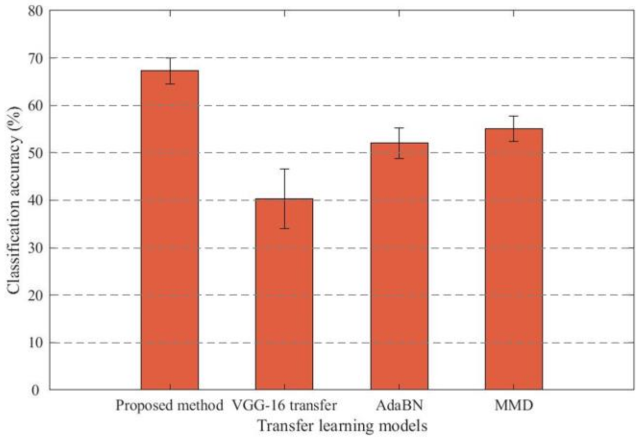
A clear clustering of features are presented using the t-SNE for the proposed method

Results: PU dataset

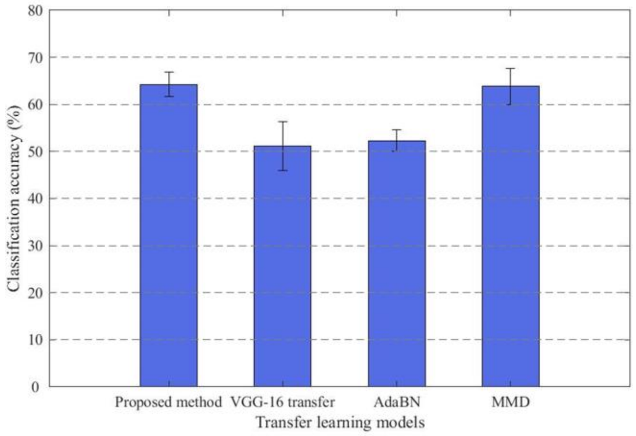
- Comparison to domain adaptation models
  - 3 state-of-the-art transfer learning models: VGG-16 transfer, AdaBN, MMD
  - Simulation-real against real-real transfer for different operating conditions



Condition 1



Condition 2



Condition 3

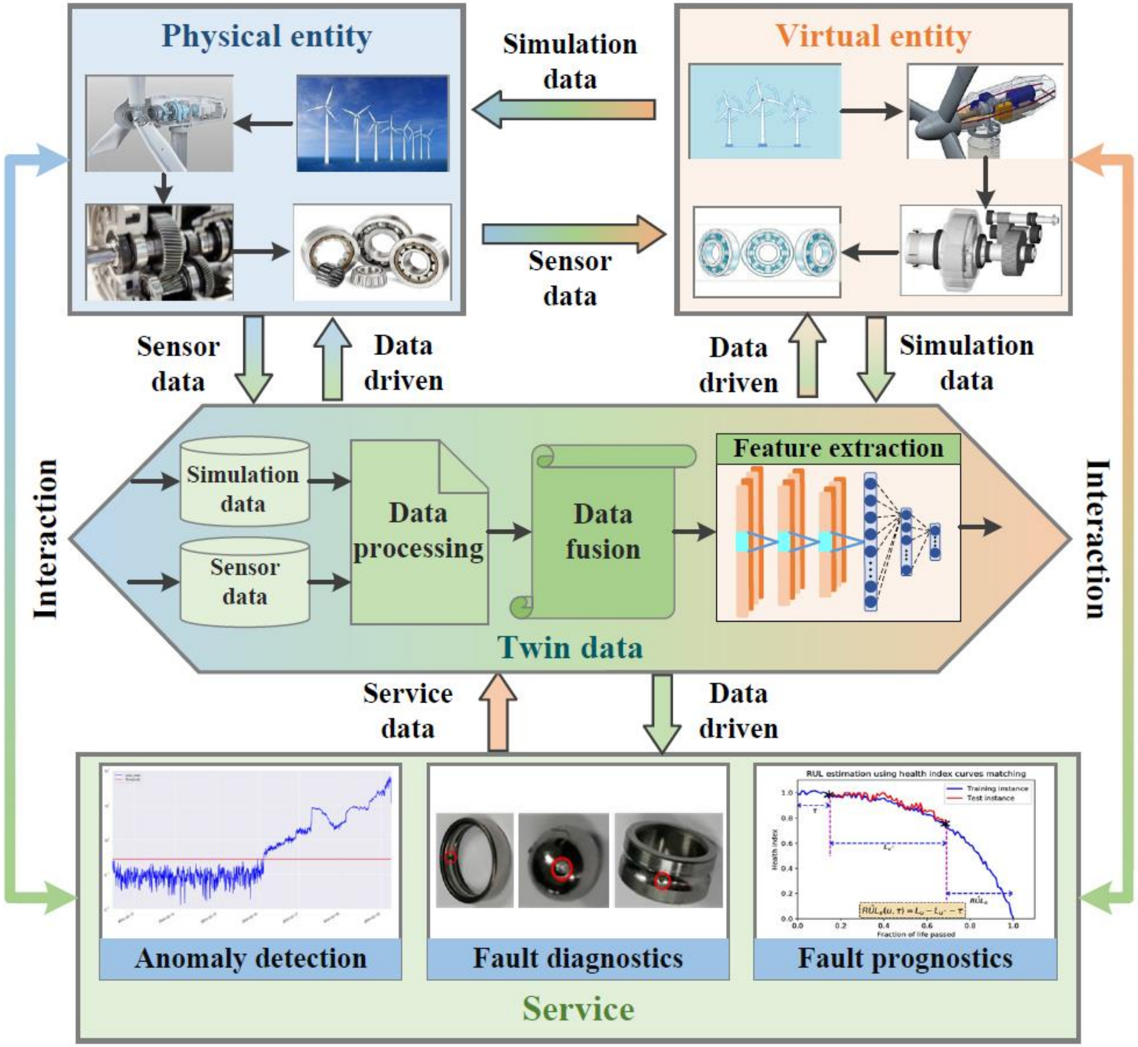
The proposed method shows high classification accuracy training with only 0.03 of the real data (40 real samples) using PU dataset.

Results

- Comparison to domain adaptation models

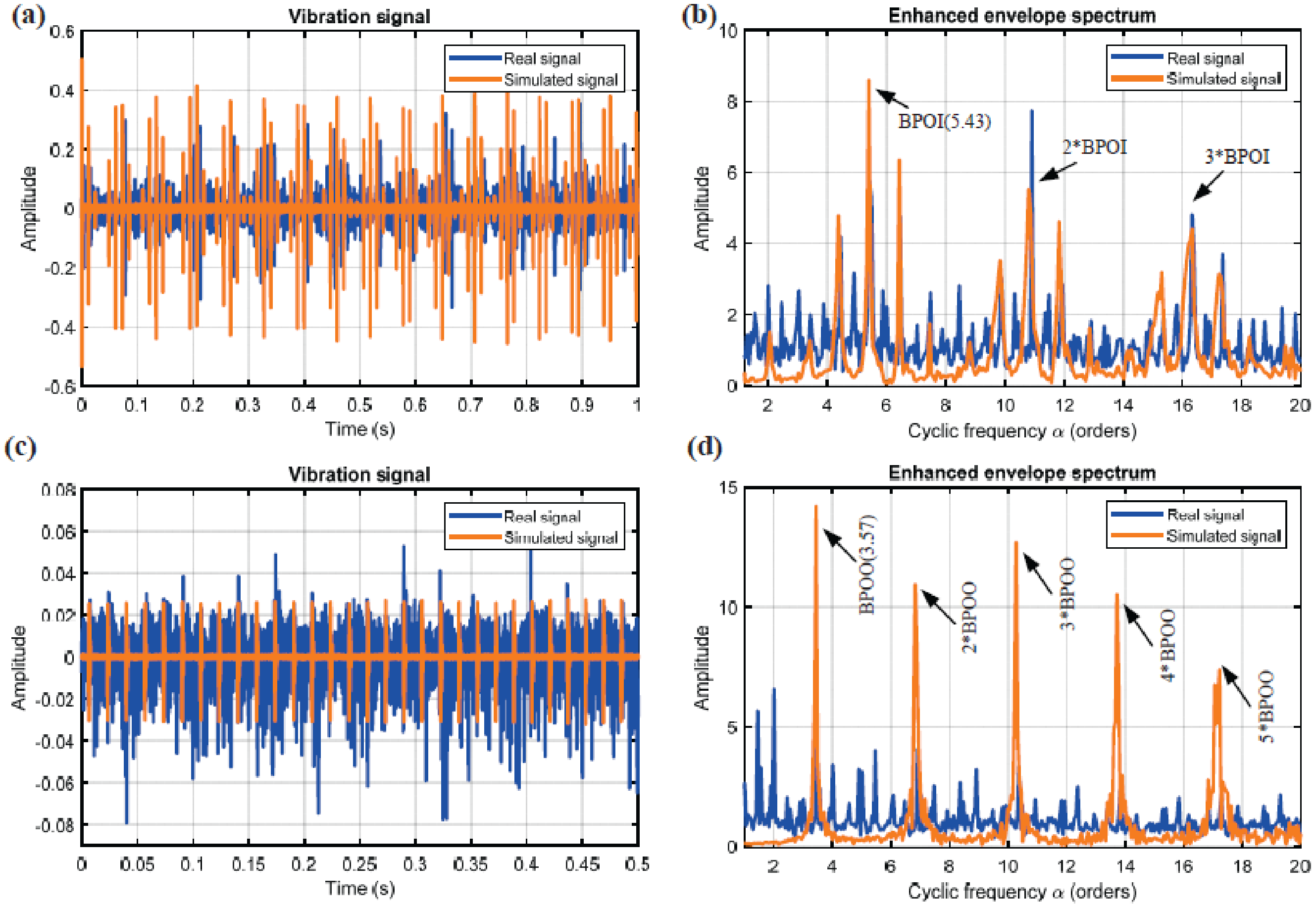
Model	Source data	Target data	Ratio of training set					
			0.05	0.10	0.20	0.30	0.40	0.50
Proposed method	Simulation	Condition 1	68.16±2.38	71.08±2.03	74.04±3.89	77.28±2.31	86.72±1.22	94.06±1.12
	Simulation	Condition 2	71.70±4.07	72.04±3.26	79.32±3.82	81.34±2.49	90.21±1.02	96.43±0.75
	Simulation	Condition 3	73.03±2.48	76.92±3.46	78.72±2.02	84.39±1.23	88.04±1.58	95.52±1.18
VGG-16 transfer	ImageNet	Condition 1	56.29±6.98	57.28±7.87	60.57±2.04	73.82±4.19	86.90±5.02	90.43±2.54
	ImageNet	Condition 2	51.46±3.06	60.20±5.42	71.12±5.07	76.13±1.86	83.25±5.76	88.22±2.09
	ImageNet	Condition 3	46.62±7.76	50.20±3.40	62.39±1.04	67.23±1.20	80.25±1.57	86.27±1.04
AdaBN	Condition 4	Condition 1	62.11±2.84	67.53±2.07	73.87±2.45	76.63±1.36	83.19±0.63	91.52±1.43
	Condition 4	Condition 2	54.73±4.83	61.02±2.59	73.35±1.51	79.38±1.05	85.62±1.21	93.70±1.09
	Condition 4	Condition 3	71.42±2.65	76.05±2.23	76.15±3.27	82.39±1.49	88.55±1.87	93.91±1.75
MMD	Condition 4	Condition 1	65.03±2.63	70.20±3.78	73.25±3.98	74.06±2.18	83.81±1.70	92.14±1.55
	Condition 4	Condition 2	63.34±4.09	67.85±2.83	75.18±3.41	79.33±2.02	86.78±0.95	91.54±1.48
	Condition 4	Condition 3	70.08±3.98	76.06±3.26	80.17±1.87	84.52±1.94	87.74±1.77	94.28±1.02

Digital Twin Framework

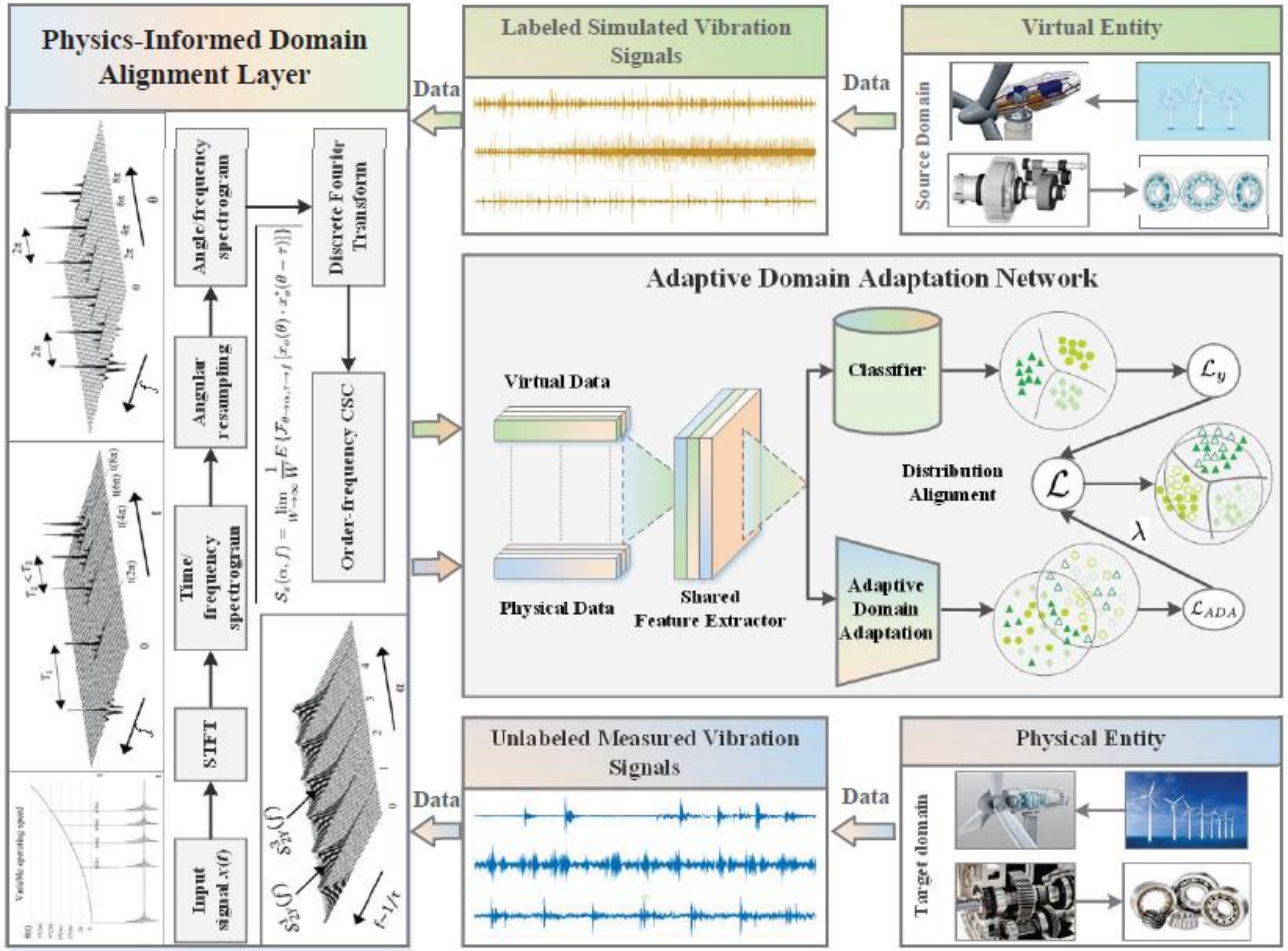


Virtual model

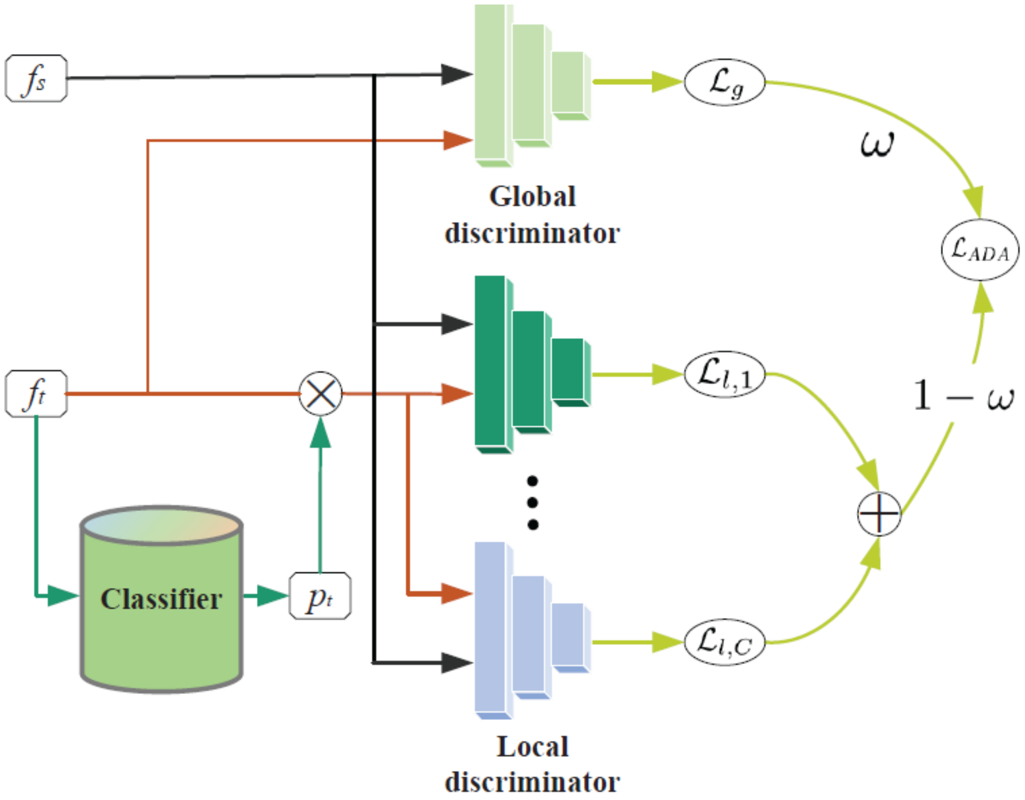
- Phenomenological model of a rolling element bearing
- How accurate should be the model?



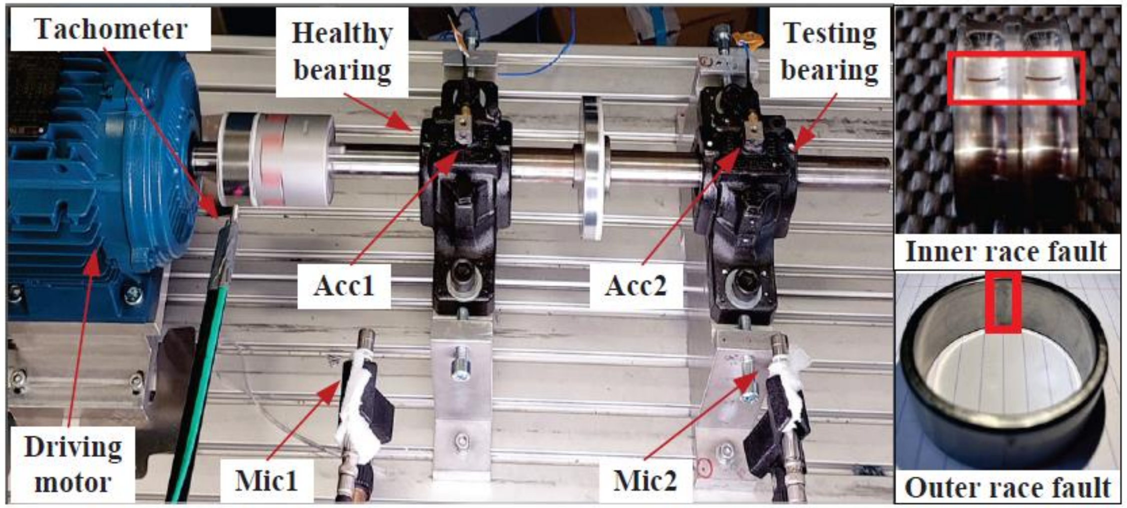
Physics-Driven Cross Domain DT

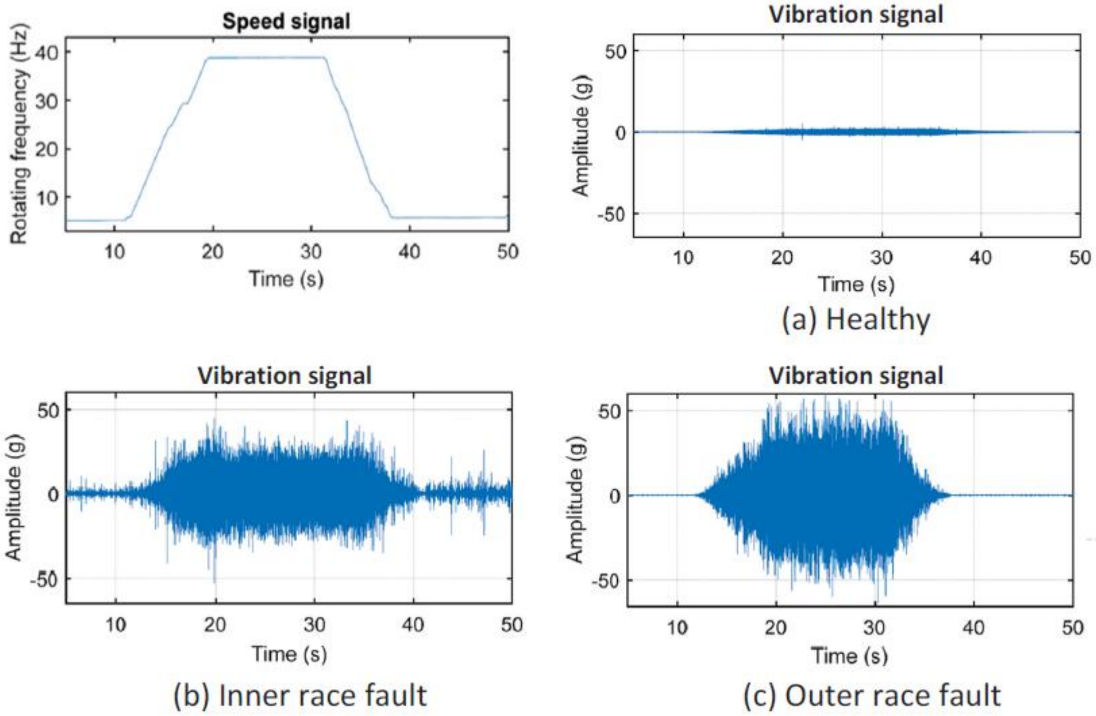


Adaptive Domain Adaptation module

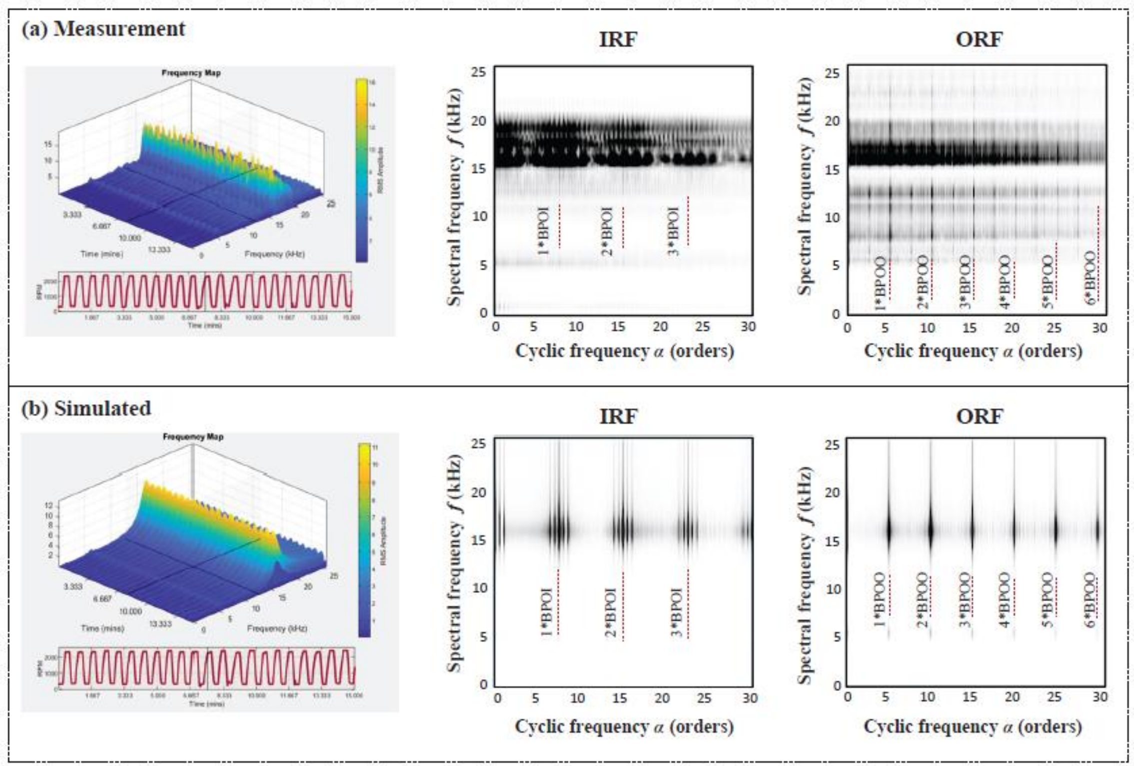


Application: LVL

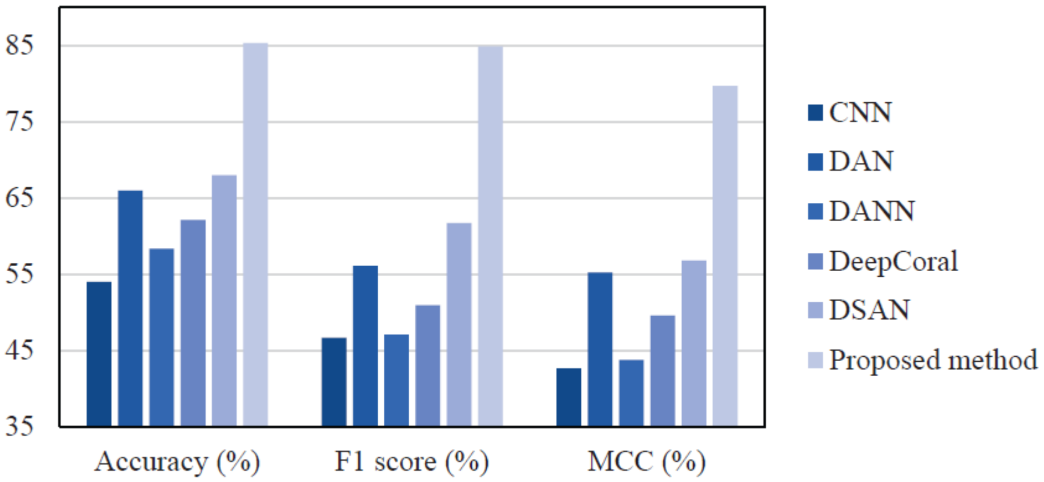


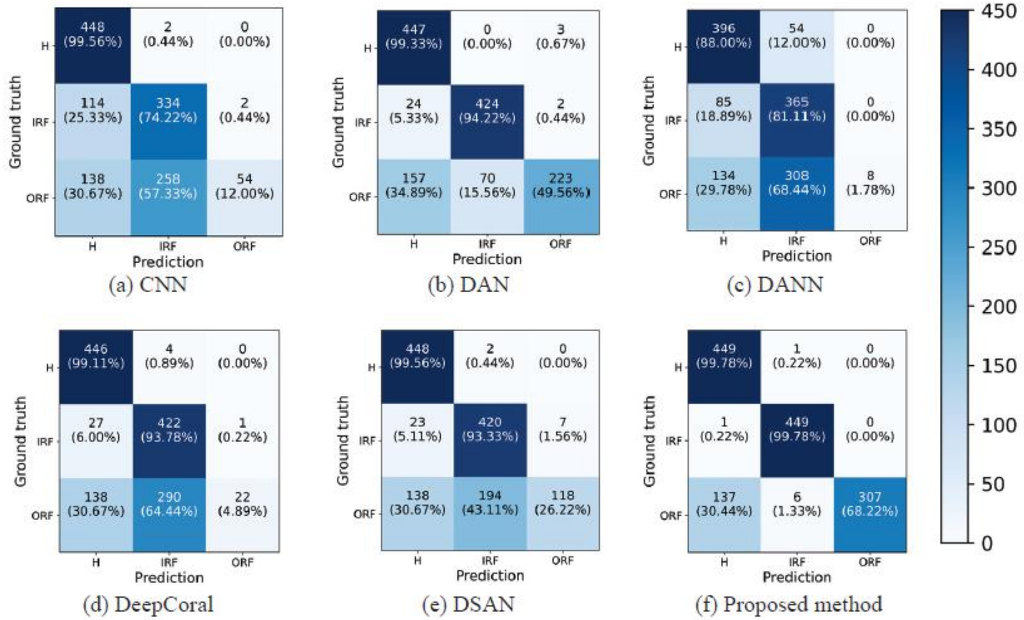


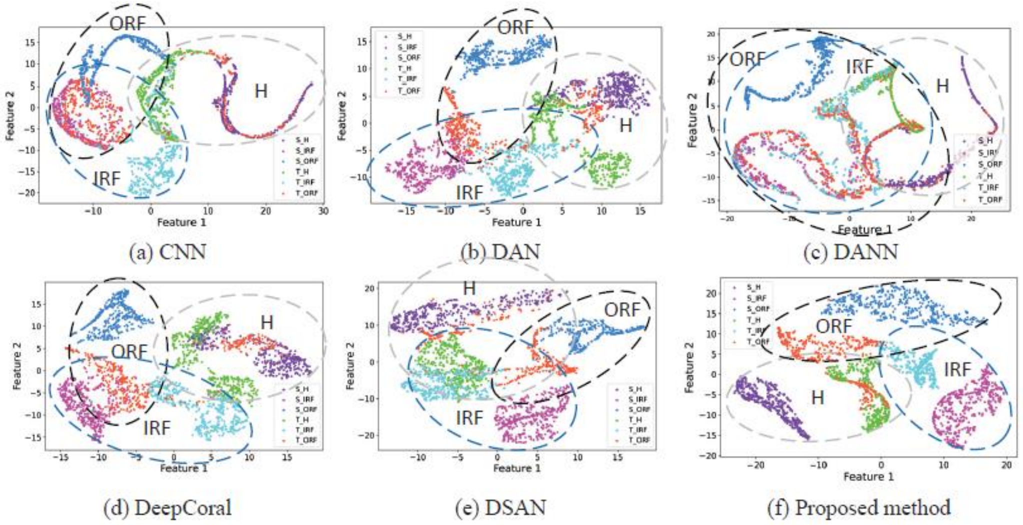
Application: LVL

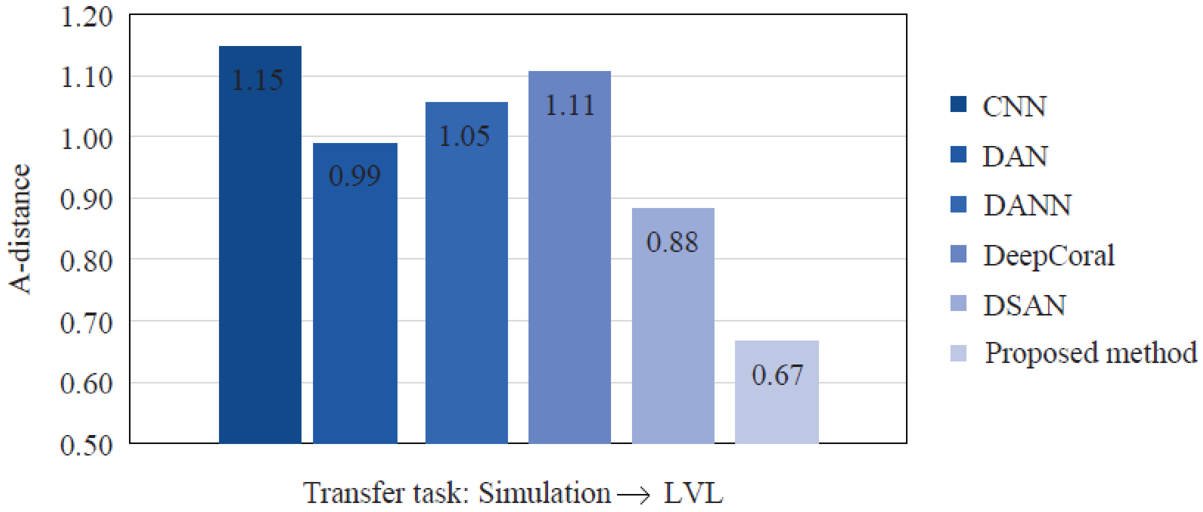


Application: LVL

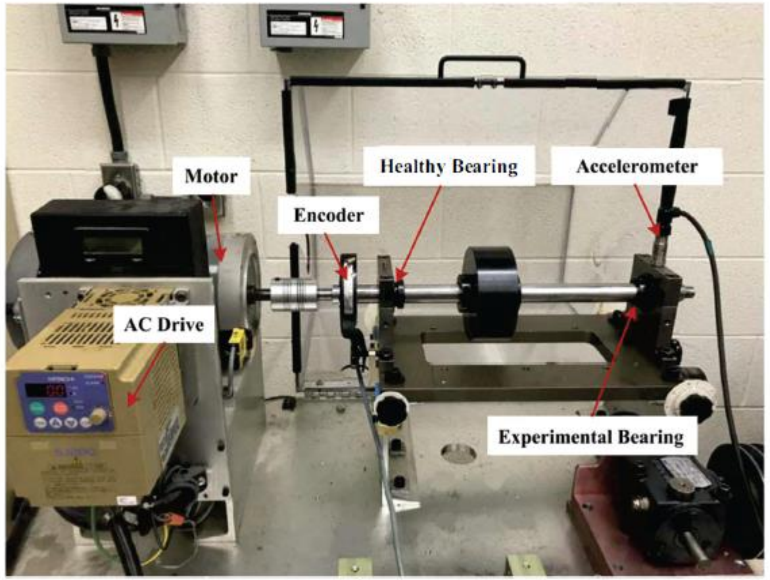


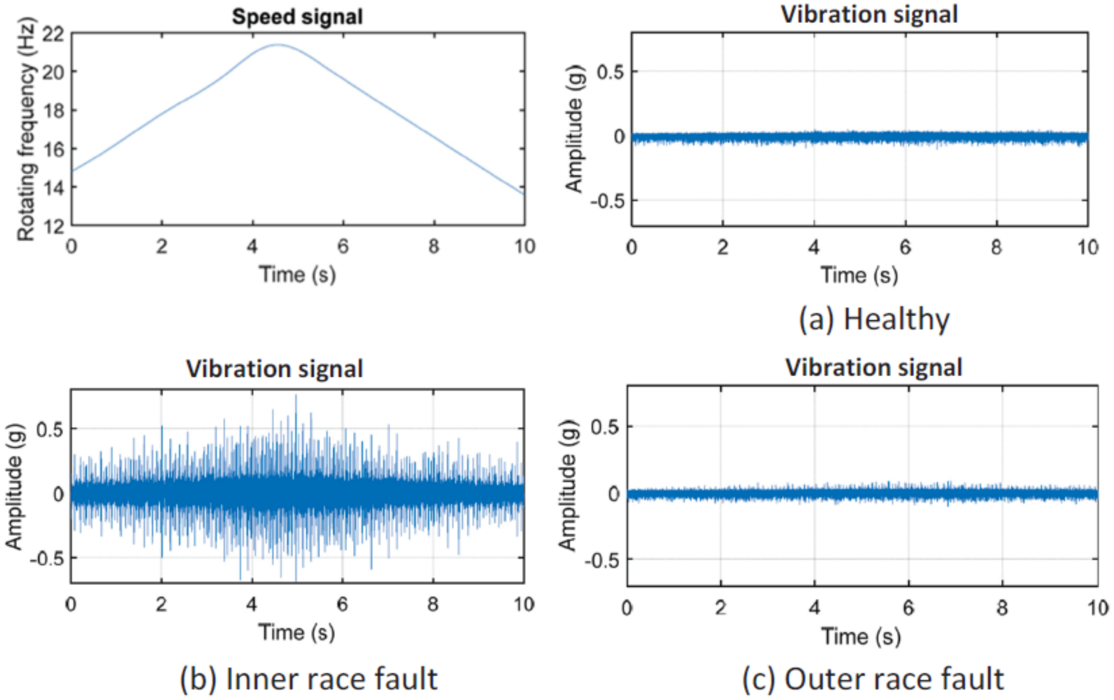




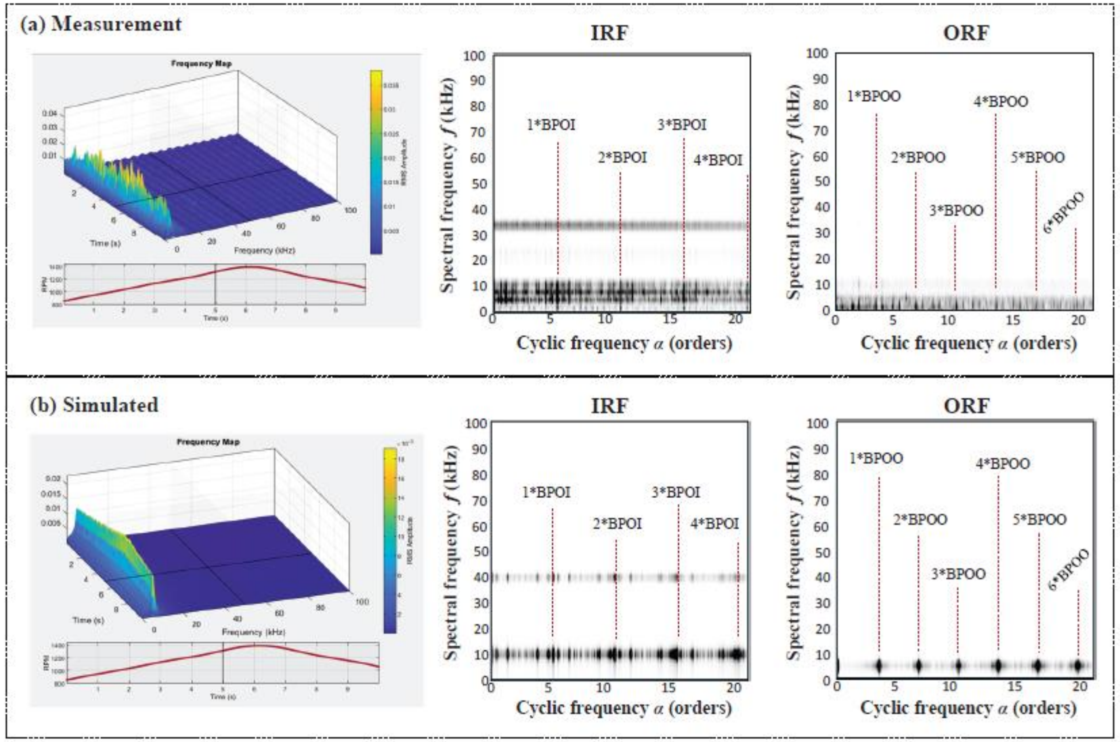


Application: Ottawa University

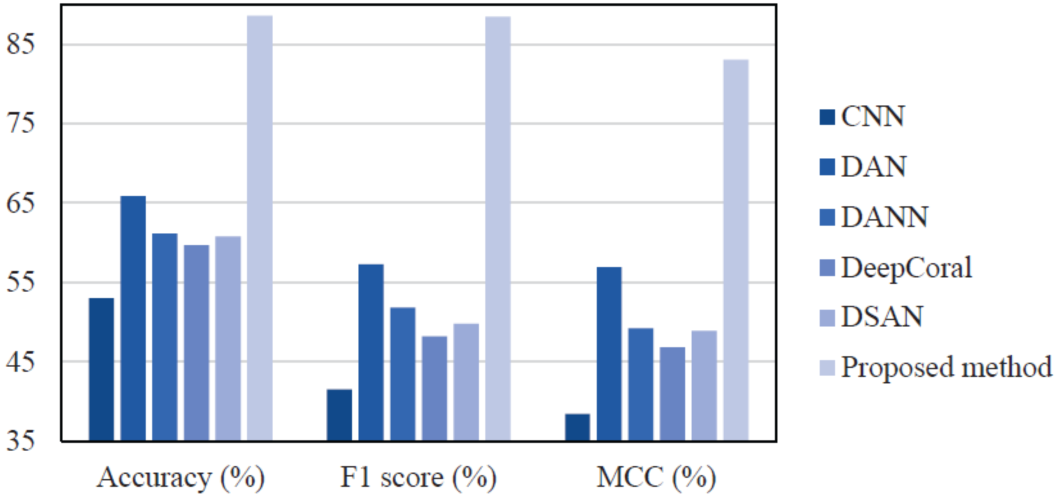


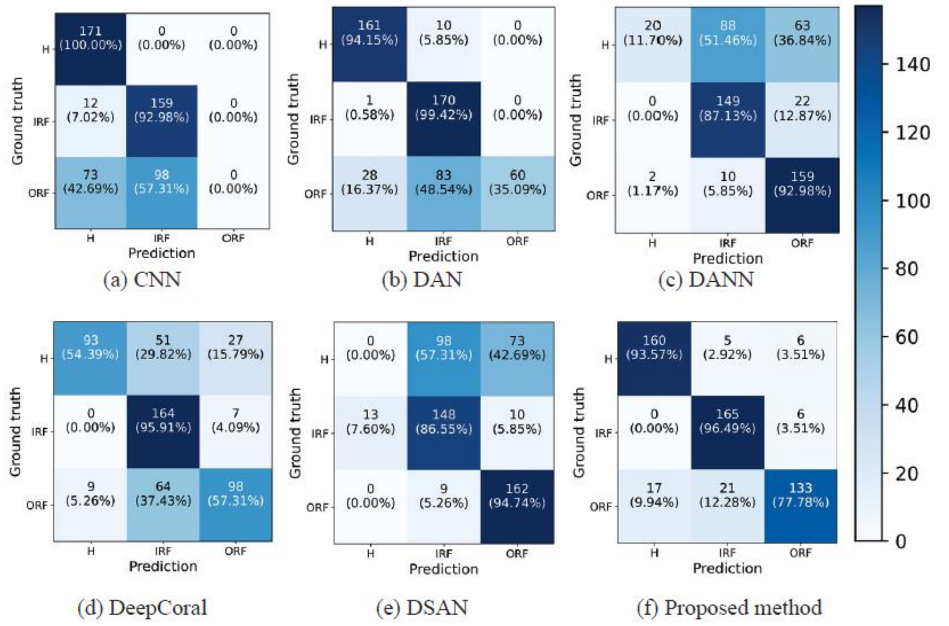


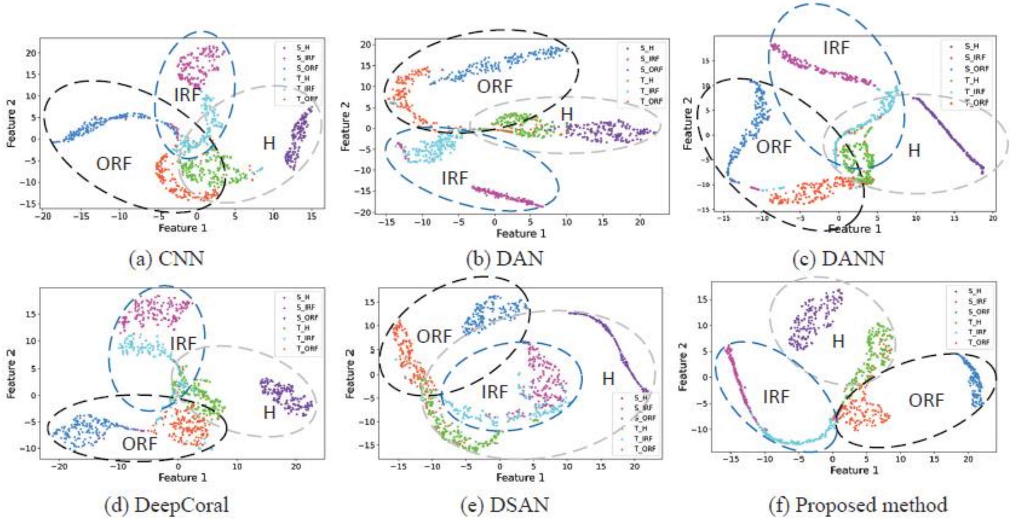
Application: Ottawa University

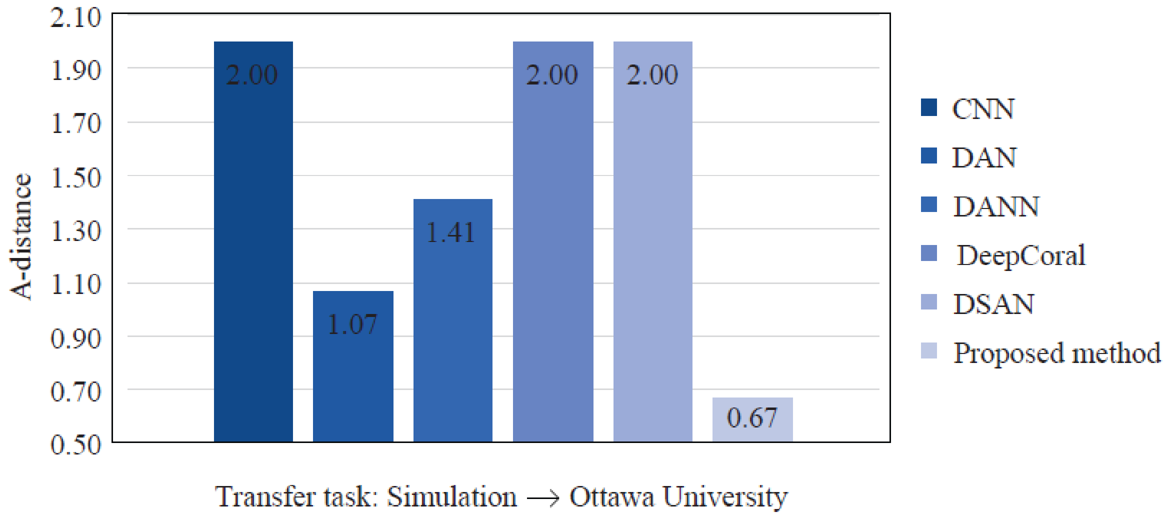


Application: Ottawa University







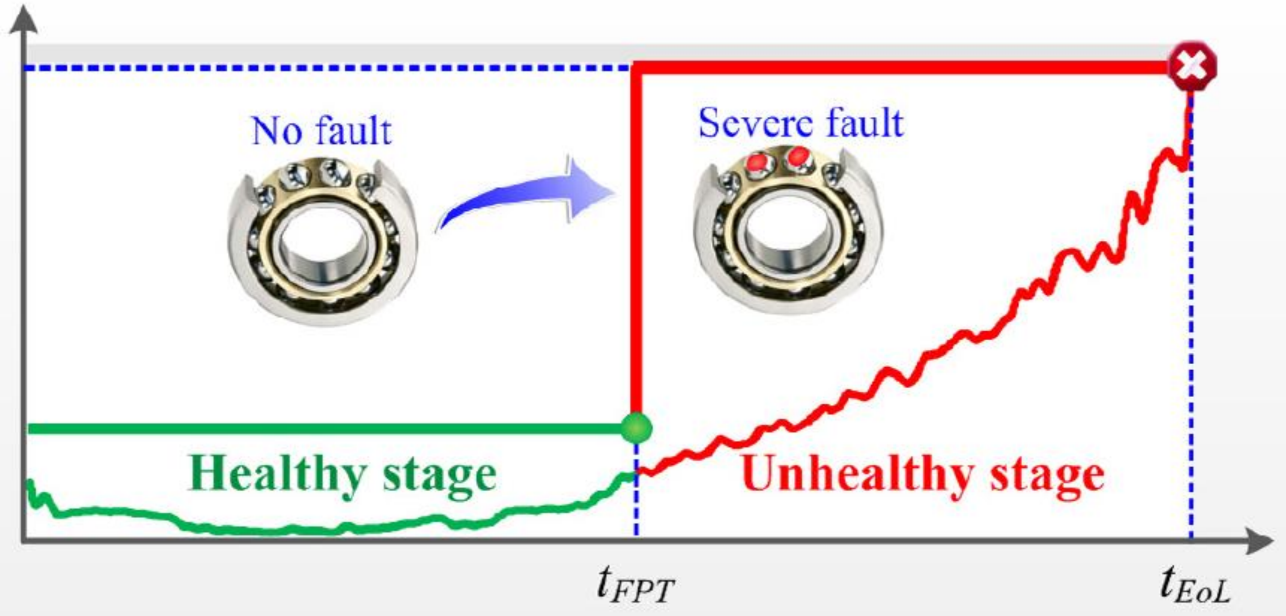


Estimation of remaining useful life: Context-aware machine learning

- Prognosis
  - Predict the future state of a component using the available information or experiences

- Remaining Useful Life (RUL)
  - Remaining time until the component can no longer operate in the desired way (failure)

- Accurate RUL prediction can reduce costs by minimizing unexpected failures and exploiting the whole lifespan of components



Introduction

- There are different approaches to estimating the RUL
- Deep Learning has shown interesting results due to its ability to model processes with high complexity

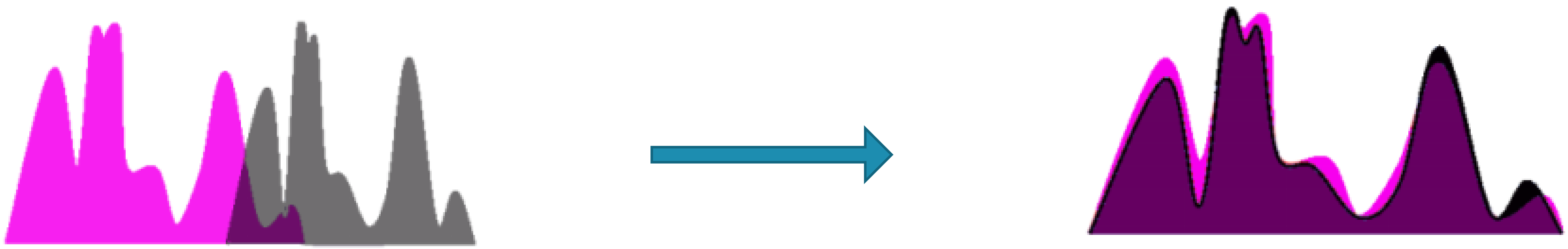
?

- Data hungry
- Collecting big dataset is:
  - Laborious
  - Of high cost

- Simulation dataset:
  - Dynamic models
  - Phenomenological models

Introduction

- Simulated dataset is not exactly like the real ones
- There is a domain shift between simulated signals (source domain) and real ones (target domain)
- **Transfer Learning** has been used to reduce the gap between domains



Introduction

- Varying speed operating condition:
  - can be seen in industrial robots, wind turbines, servo motors, etc.
  - is another challenge for the model performance in RUL prediction
- Tachometer signals can be used as a “Context” to improve the performance

> “Context can be defined as any information about working conditions such as load, temperature, and speed that has a significant effect on the equipment’s behavior”

Methodology

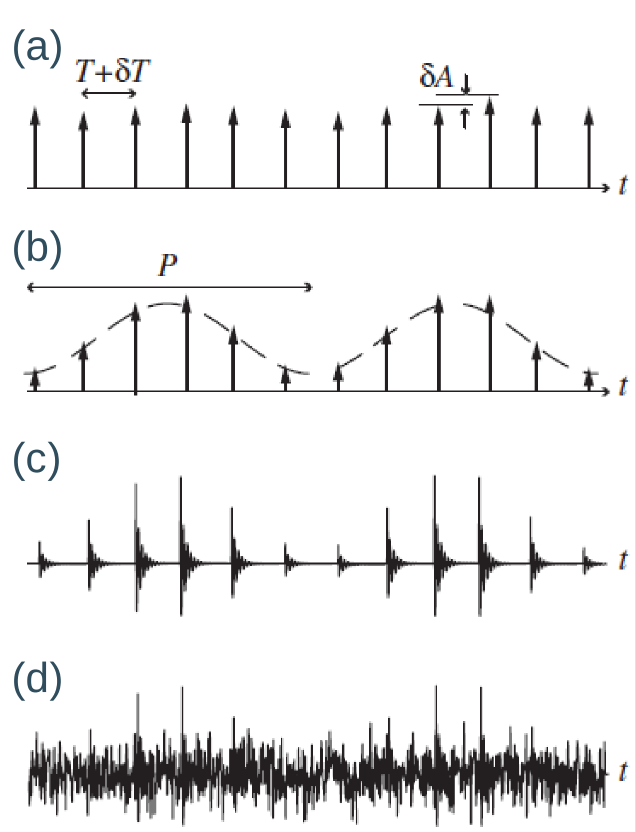
- Utilize phenomenological model to generate signals
- Mitigate the influence of insufficient data availability for training

a) Impacts in the certain intervals (depending on the type of fault)

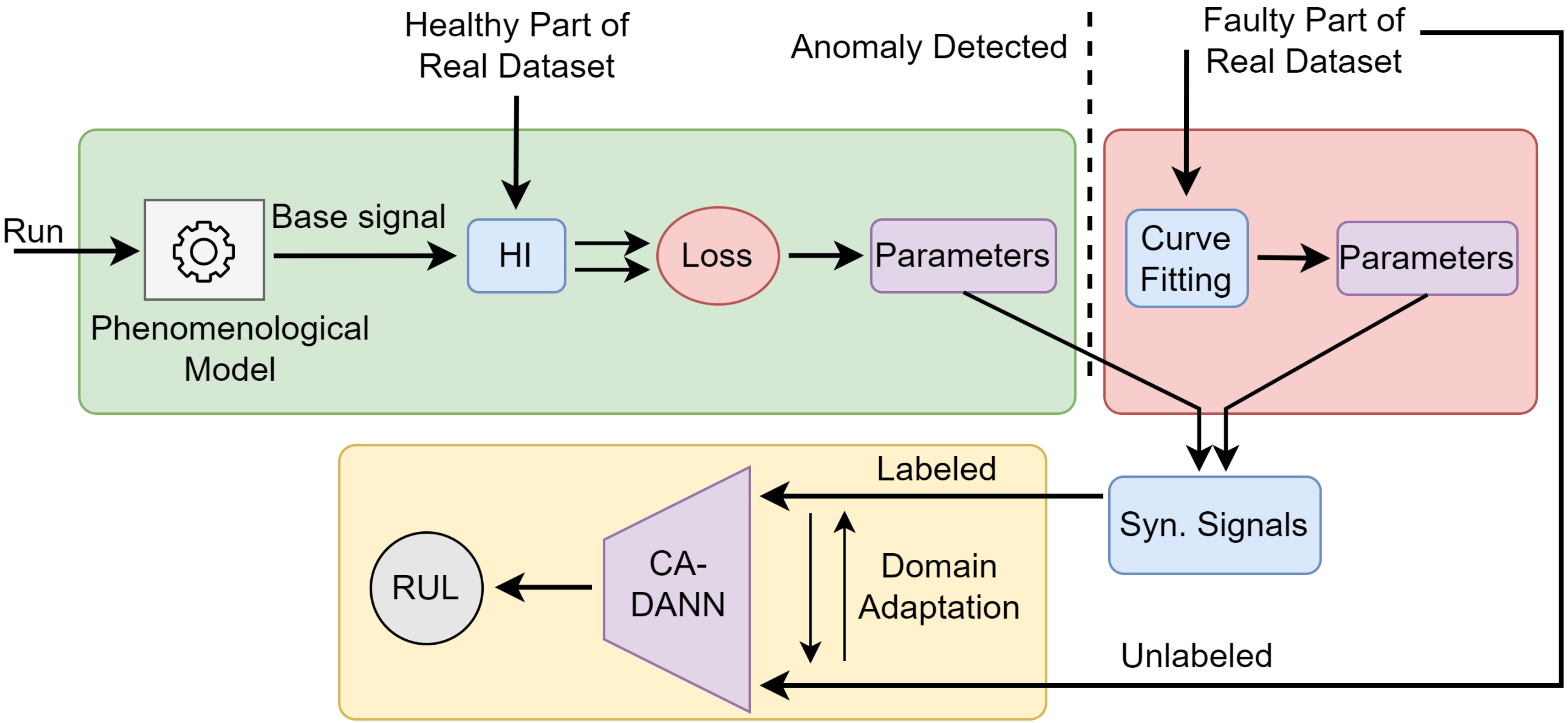
b) Load modulation (if applicable)

c) Excite the equivalent SDOF model with the impacts

d) Add noise



Methodology



Methodology

- Adapt the synthetic signals using two modifier functions **R_i** and D_i
- **Periodic stepwise speed profile** is assumed as varying speed conditions

Adapted signal

Unadopted signal by phenomenological model

$$\boldsymbol{S}_{s,i} = \boldsymbol{S}_{b,i} . (\boldsymbol{R}_i \times D_i)$$

Represents damage

$$D_i = \begin{cases} 1 & t_i < t_{anomaly} \\ e^{a(t_i - t_{anomaly})} & t_i \geq t_{anomaly} \end{cases}$$

Represents speed influence

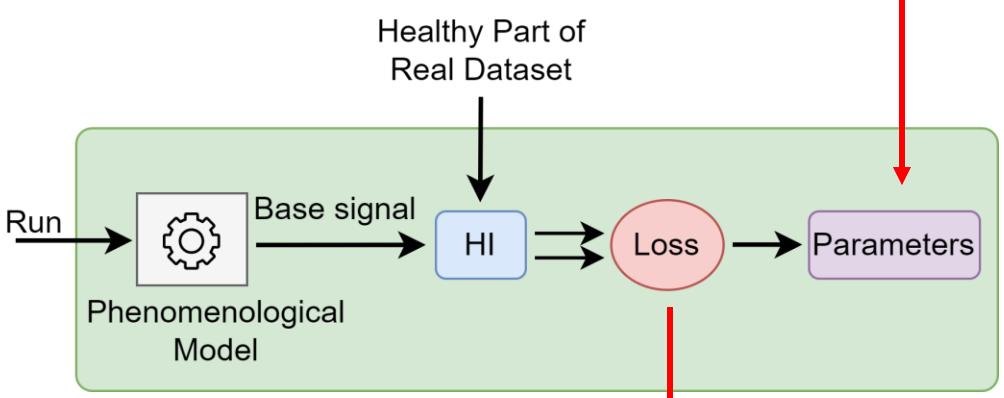
$$\boldsymbol{R}_i = c_j \; . \; \boldsymbol{rpm}_i$$

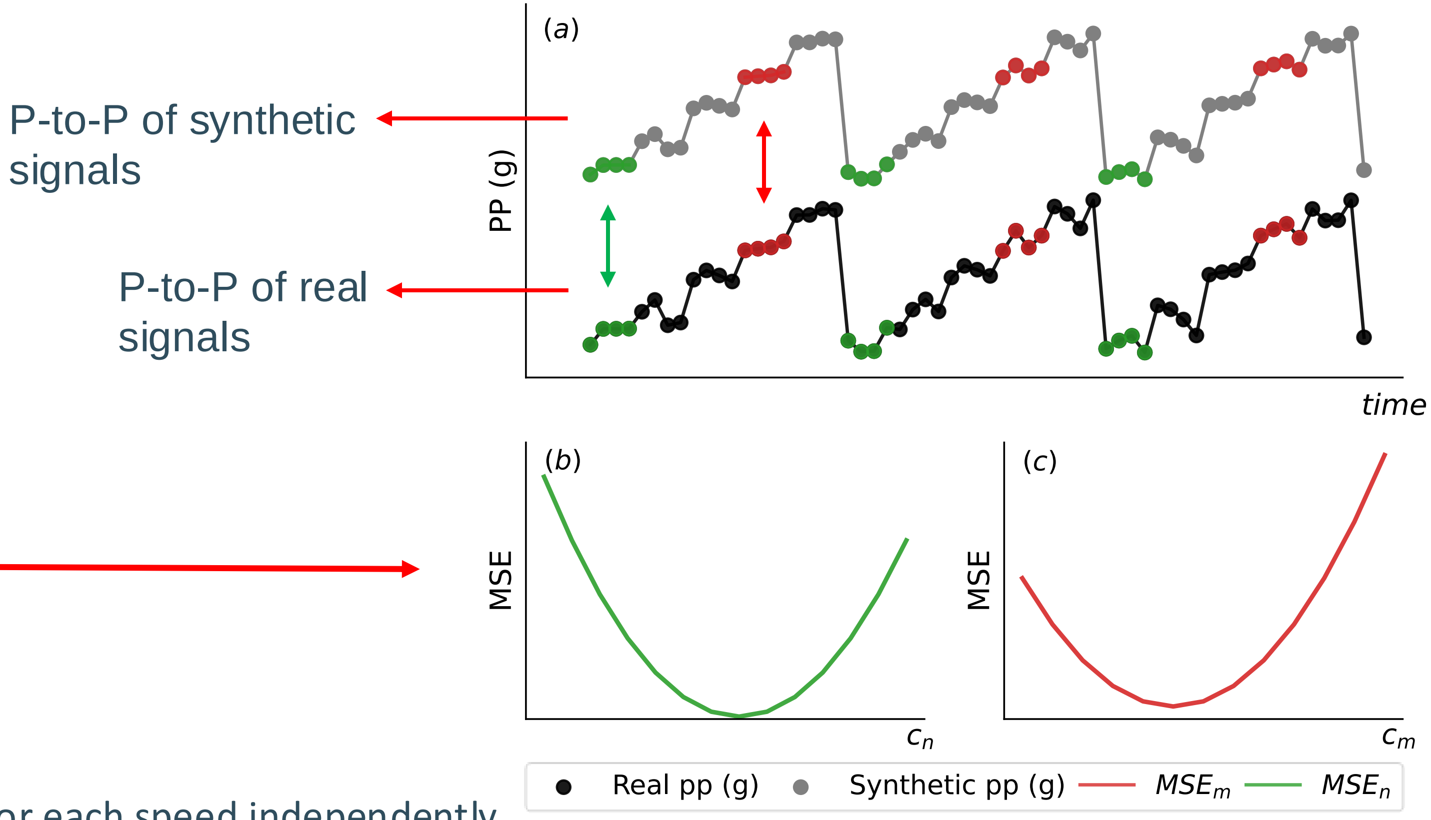
Should be found for each speed independently

i: measurement index
j: speed index

Methodology

- ➢ Find c_j in a way that peak-to-peak of real and synthetic signals should be as close as possible

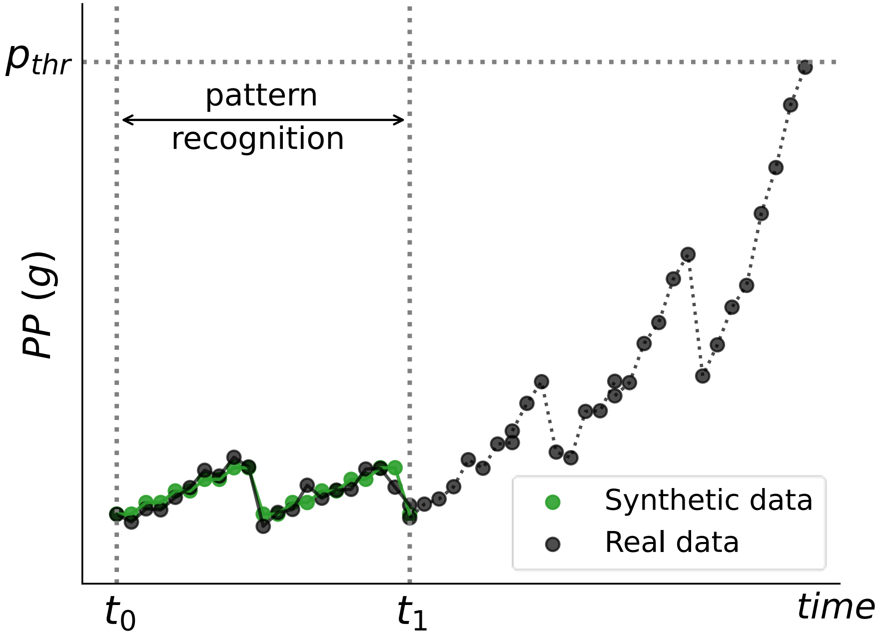




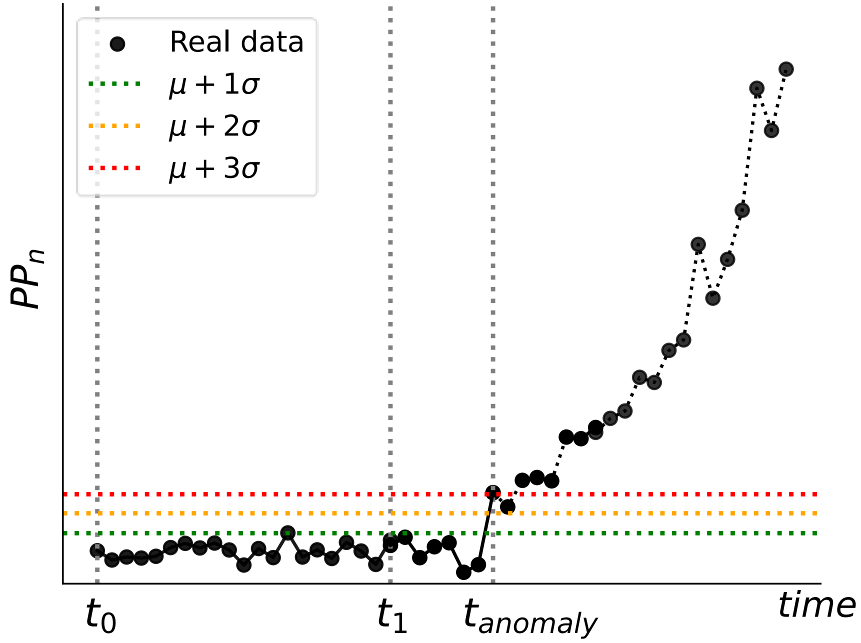
Find $\boldsymbol{c_j}$ corresponding to the minimum error for each speed independently

Methodology

- We can now generate healthy signals similar to the real healthy signals
- $PP_n = \frac{PP_r}{PP_s}$ Form a new health indicator
- Normalized Peak-to-peak is used for **anomaly detection**



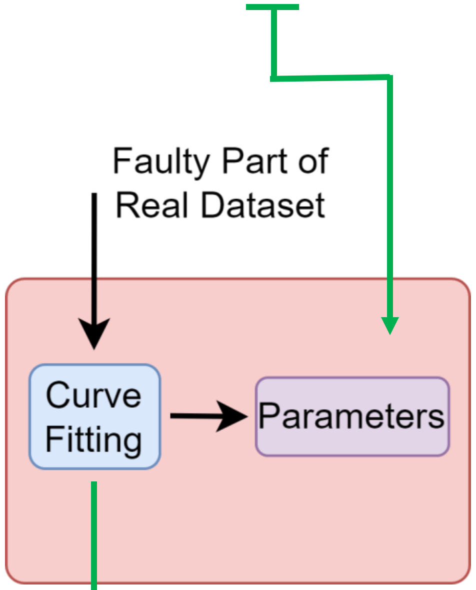
Peak-to-peak

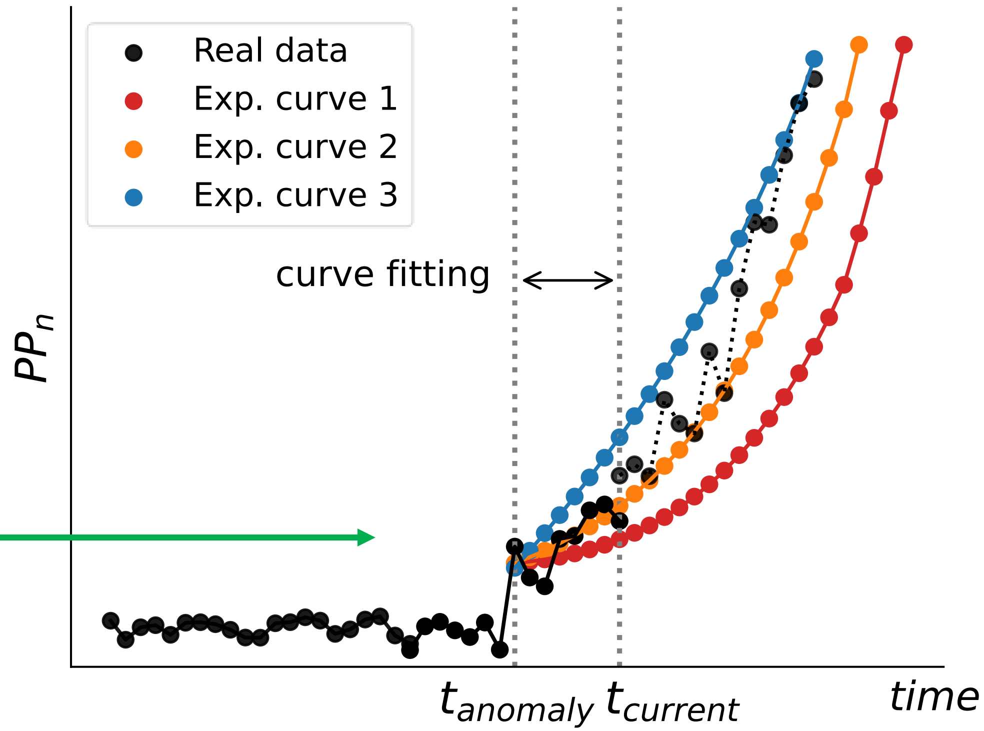


Normalized peak-to-peak

Methodology

➢ Find D_i using curve-fitting on the normalized peak-to-peak of the real dataset after the anomaly



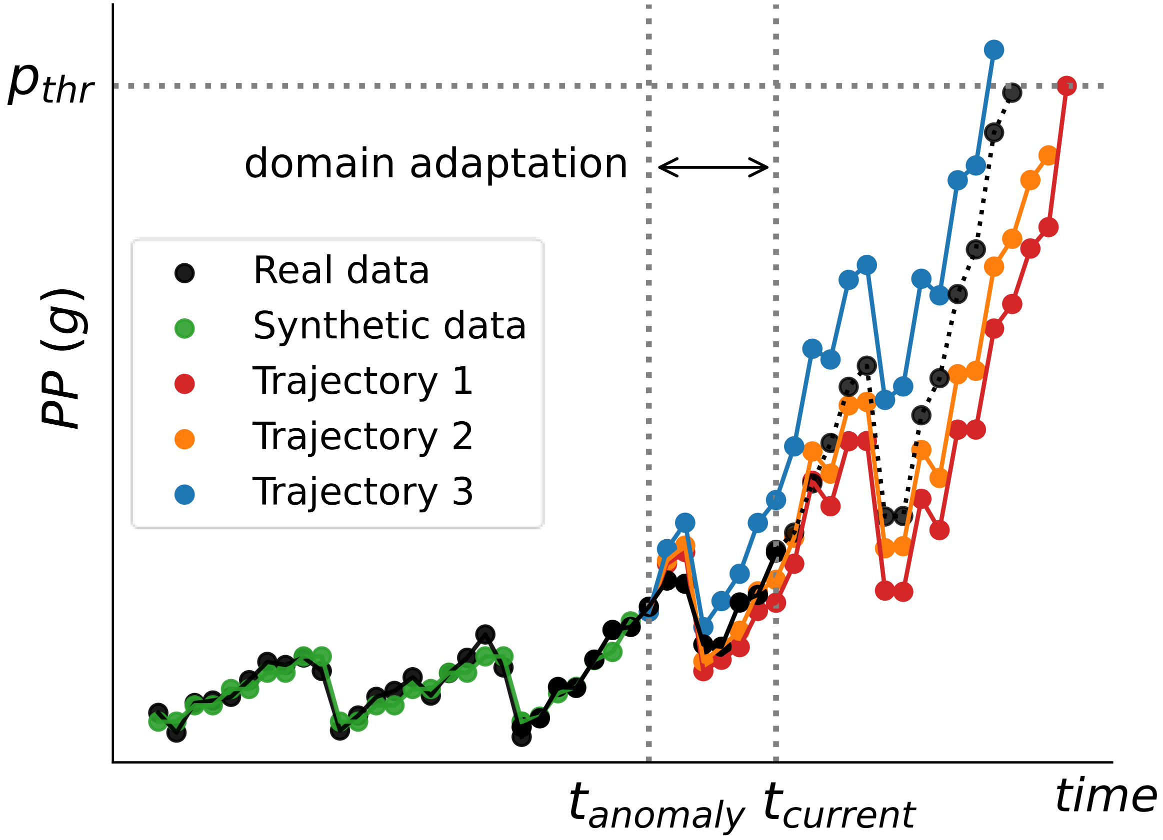


$$D_i = \begin{cases} 1 & t_i < t_{anomaly} \\ e^{a(t_i - t_{anomaly})} & t_i \geq t_{anomaly} \end{cases}$$

With a slight change in a several trajectories will be created

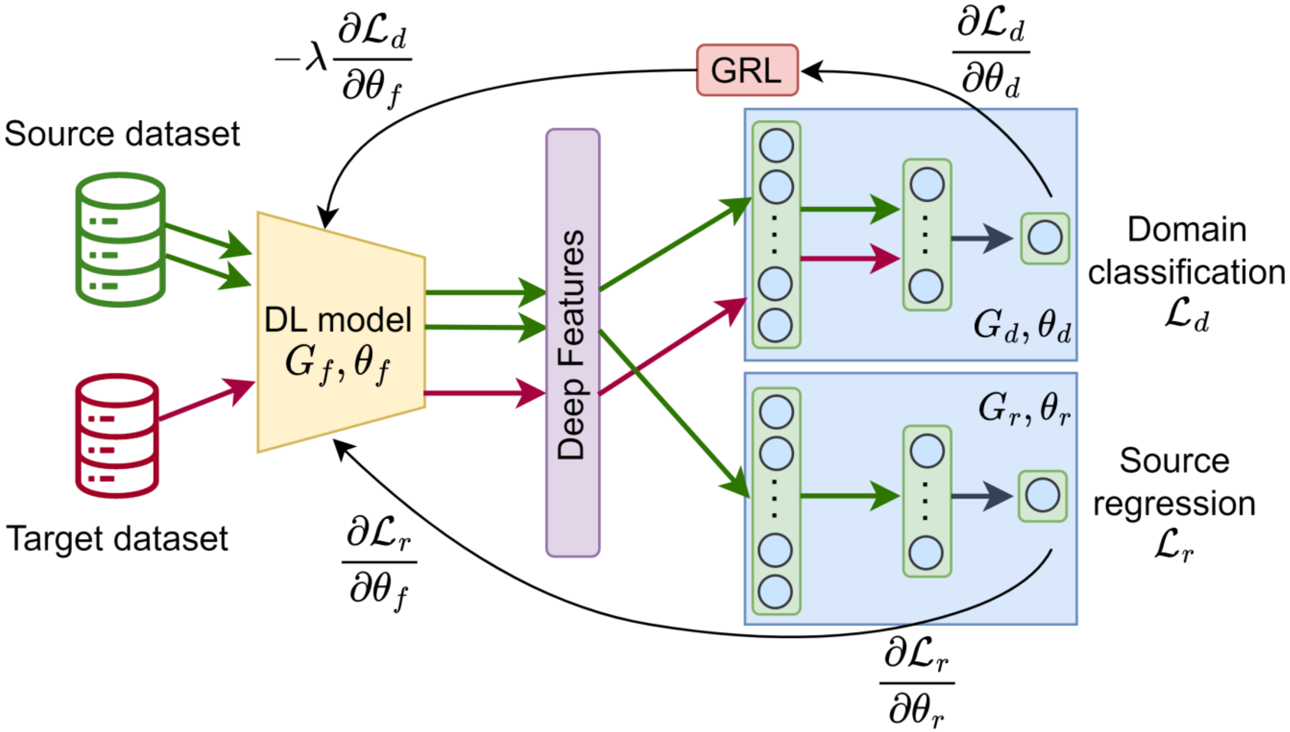
Methodology

- Putting all together, several synthetic run-to-failure datasets have been created
- Input of the ML model is the raw signals obtained by this approach



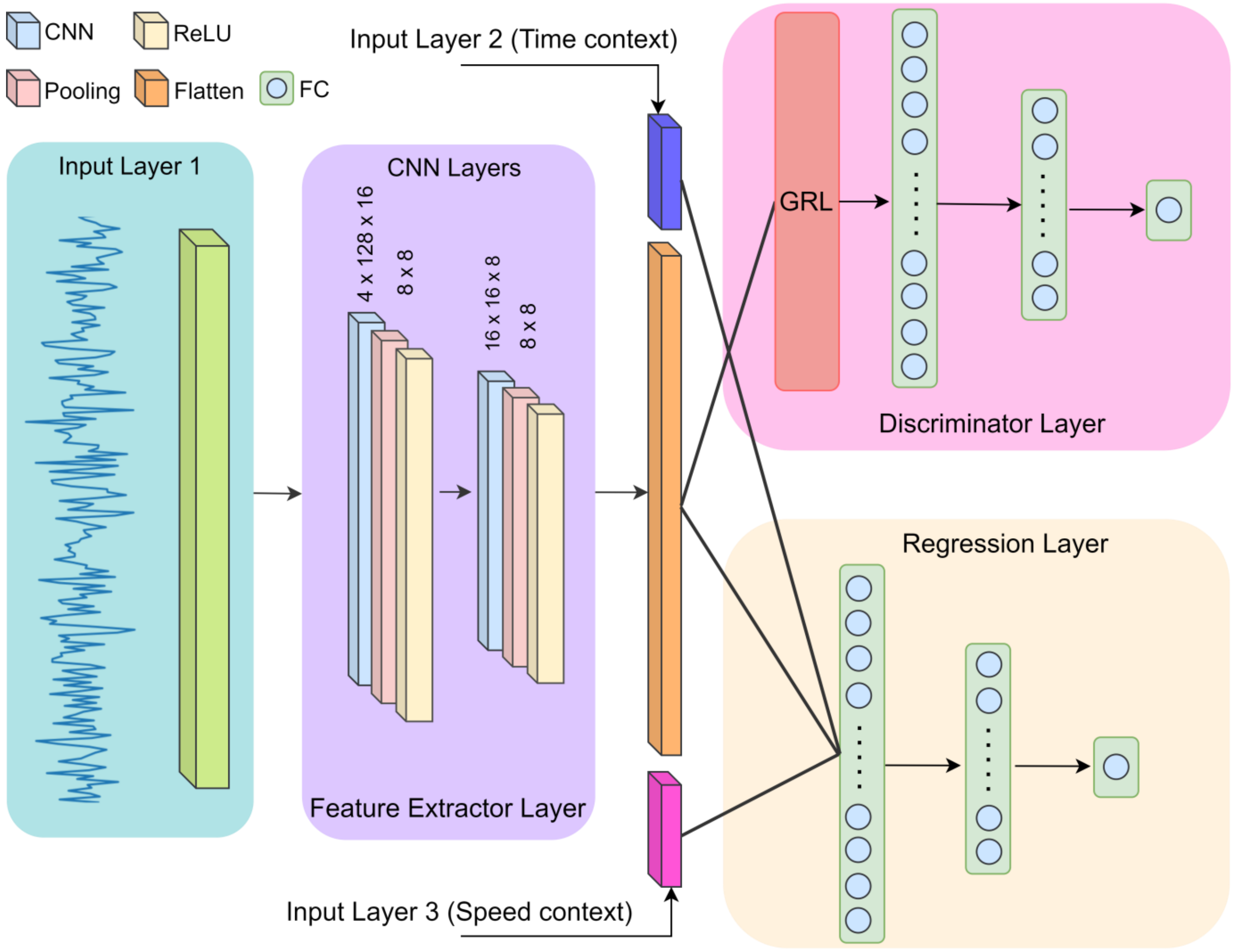
Methodology

- Domain Adversarial Neural Network (DANN) is used for domain adaptation
- Source dataset: synthetic run-to-failure data
- Target dataset: real data



Methodology

- ➢ Input 1: Raw signals
- ➢ Input 2: Time information
- ➢ Input 3: Speed information



Proposed architecture based on DANN model

Methodology

➢ Inspired by NLP, the order of measurements (time) and speed can be encoded as 1-D vectors

> "We chose this function because we hypothesized it would allow the model to easily learn to attend by relative positions, since for any fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos}"

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

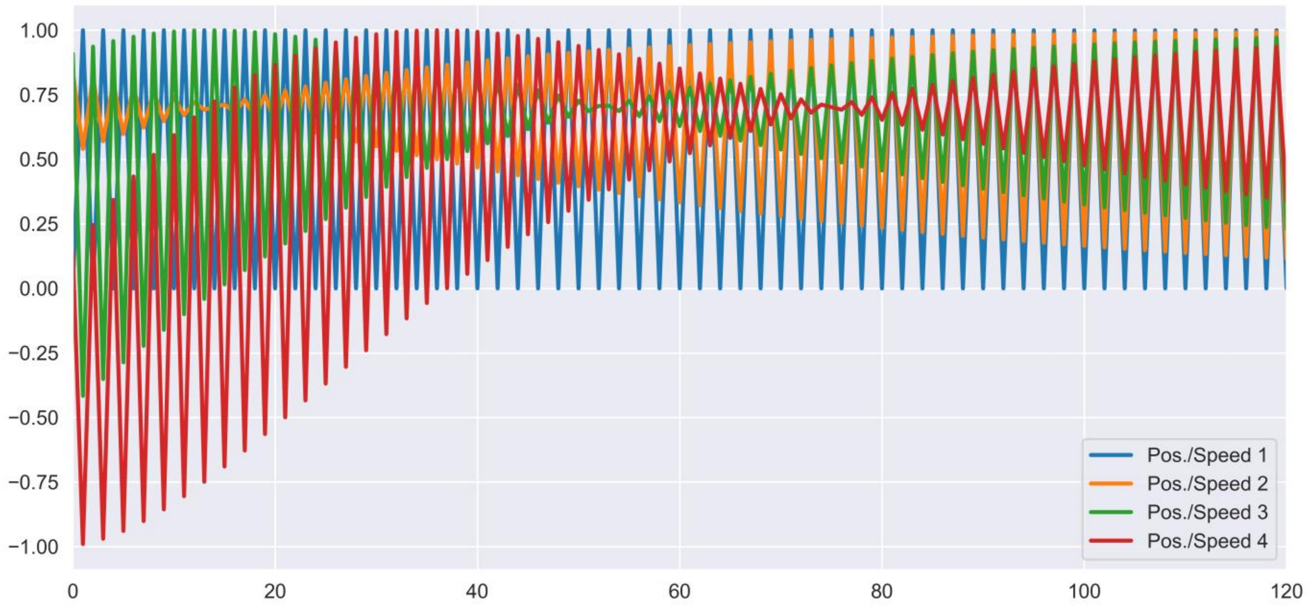
$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

For i = 0, 1, …, $\frac{d_{model}}{2} - 1$

Example for $pos = 1$:

$$\left[PE_{(1,0)}, PE_{(1,1)}, \ldots, PE_{(1,\, d_{model} - 1)}\right]$$

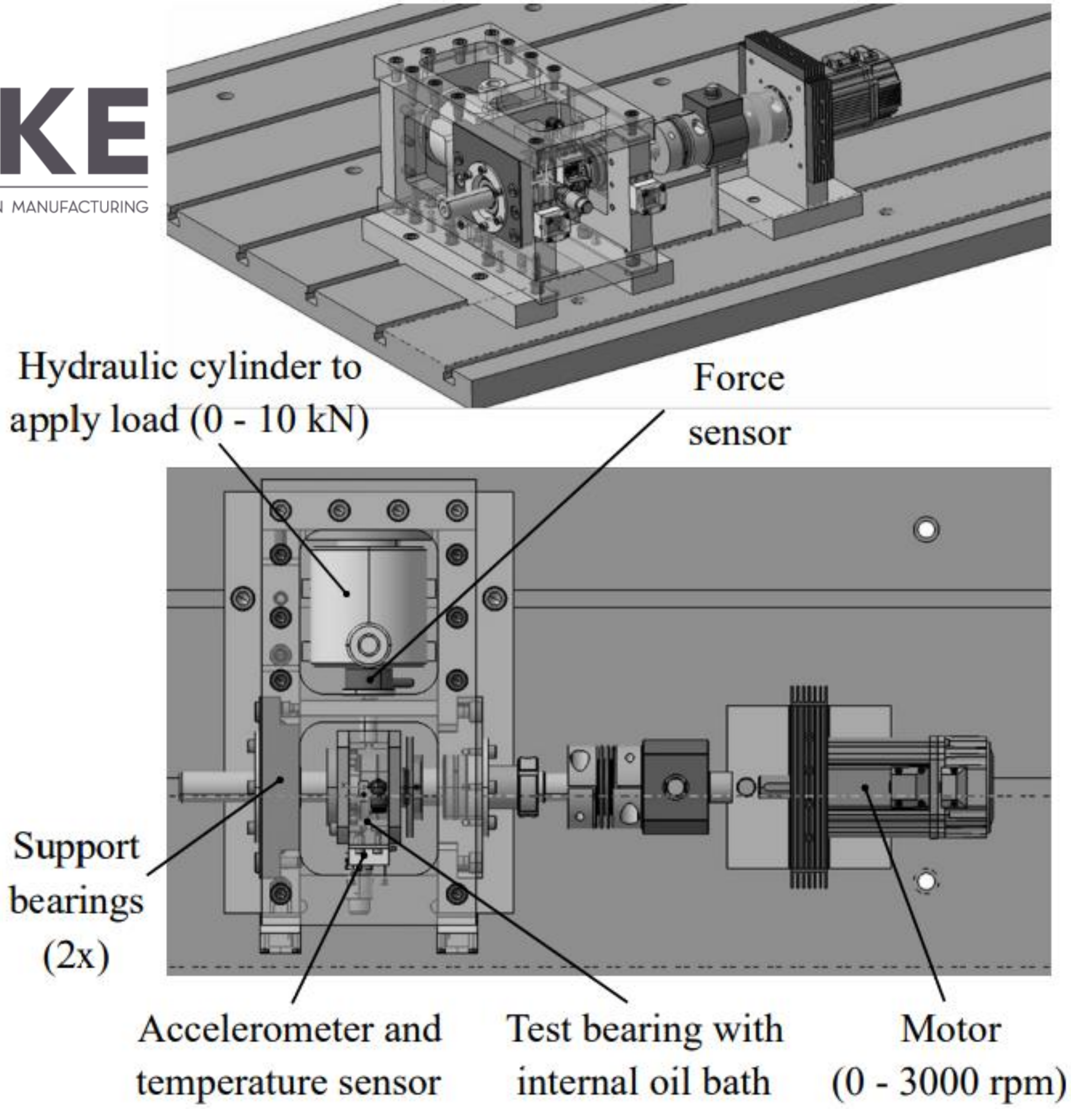
d_{model}: hyper parameter (selected 24)



Application

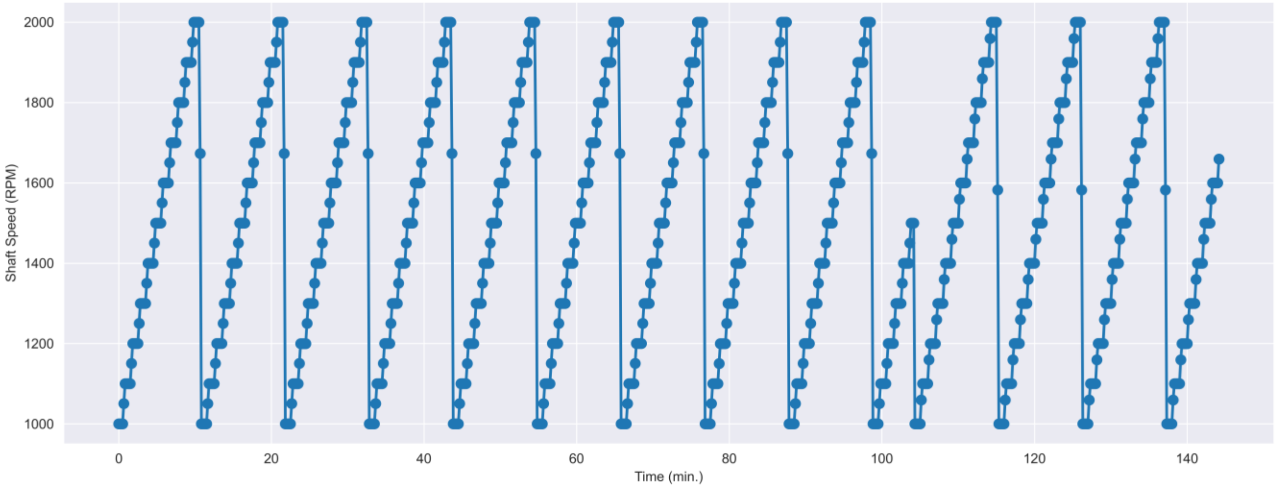
- Smart Maintenance datasets
  - Run-to-failure tests of rolling element bearings
  - Vibration signal sampling rate: 50 kHz
  - Signals are captured continuously
  - Test bearings: 6205-C-TVH from FAG

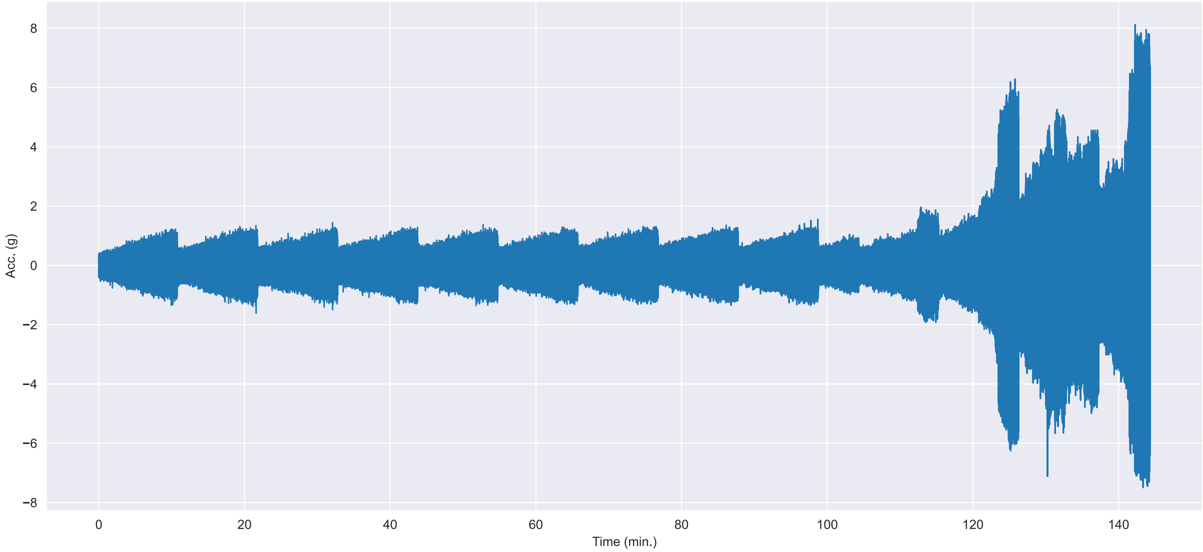
- Tests were stopped due to different criteria
  - Temperature
  - Test duration
  - Peak-to-peak of vibration exceeds 20g



Application

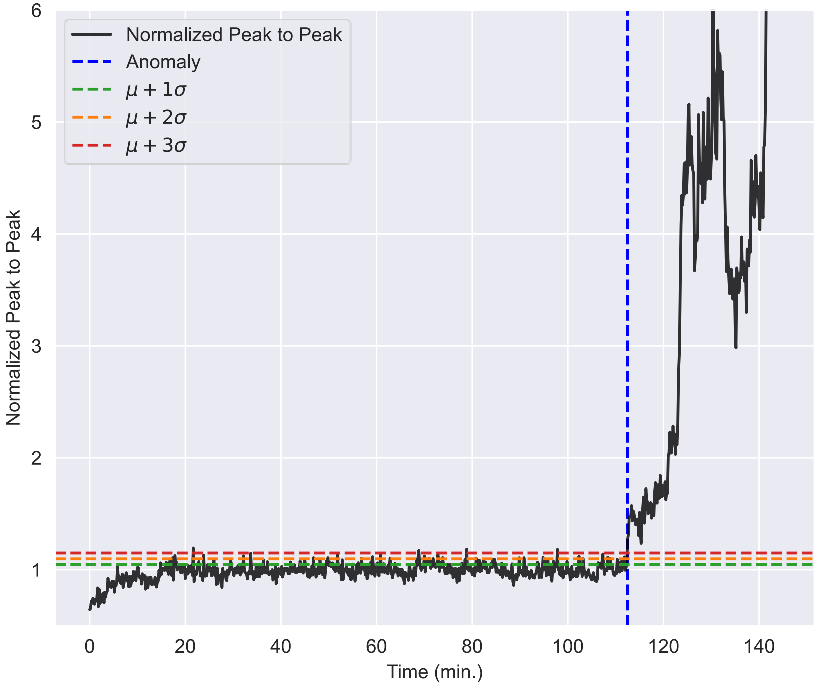
- Smart Maintenance dataset
  - Run-to-failure tests of bearings under varying speed operating conditions
  - Speed is changing stepwise between 1000rpm and 2000rpm with the increment of 100rpm
  - Each step is maintained for 60 seconds
- 6 run-to-failure tests
  - A new EoL threshold of peak-to-peak=15g has been defined to have a consistent dataset



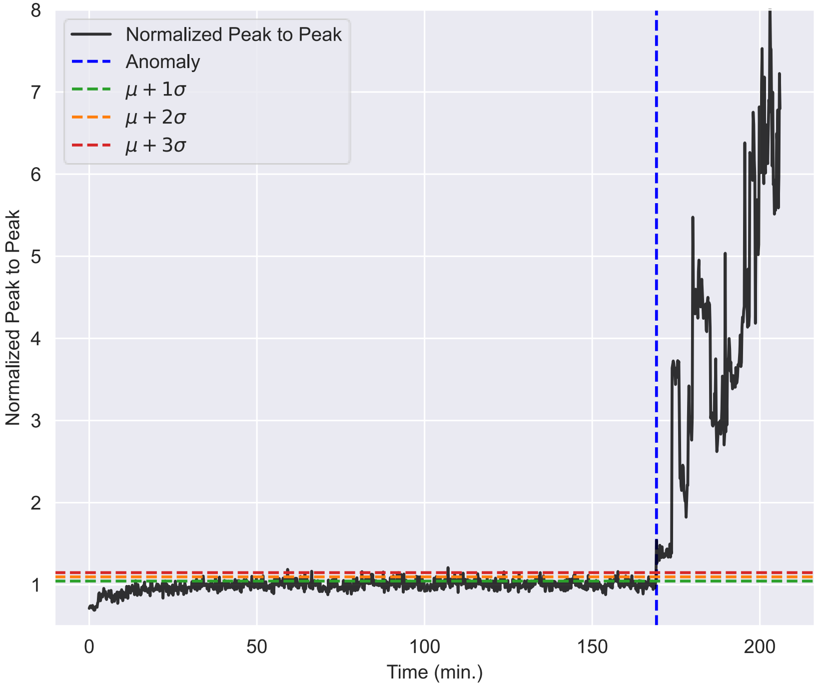


Results

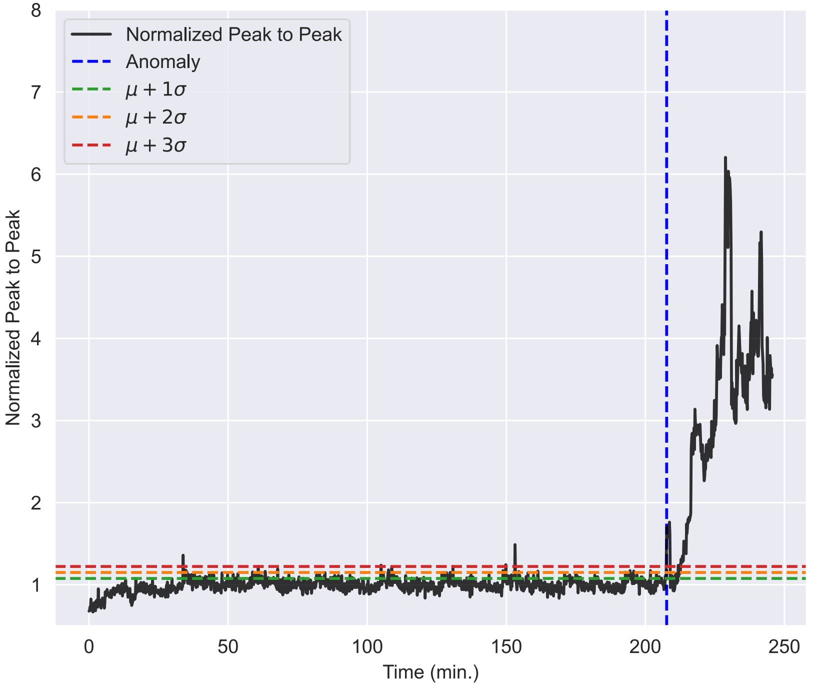
- Anomaly detection criterion
  - 3 consecutive points above the highest threshold



A148



A150

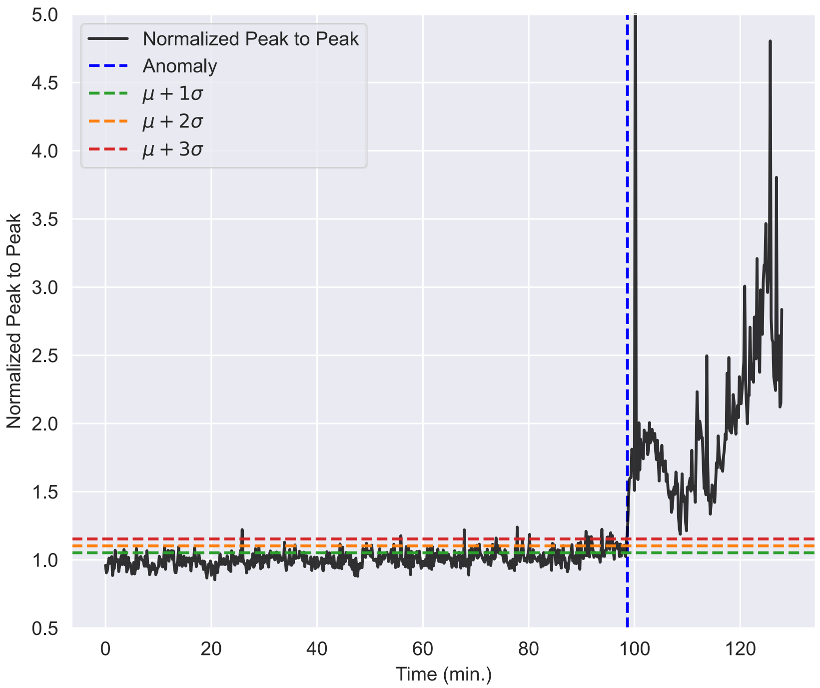


A153

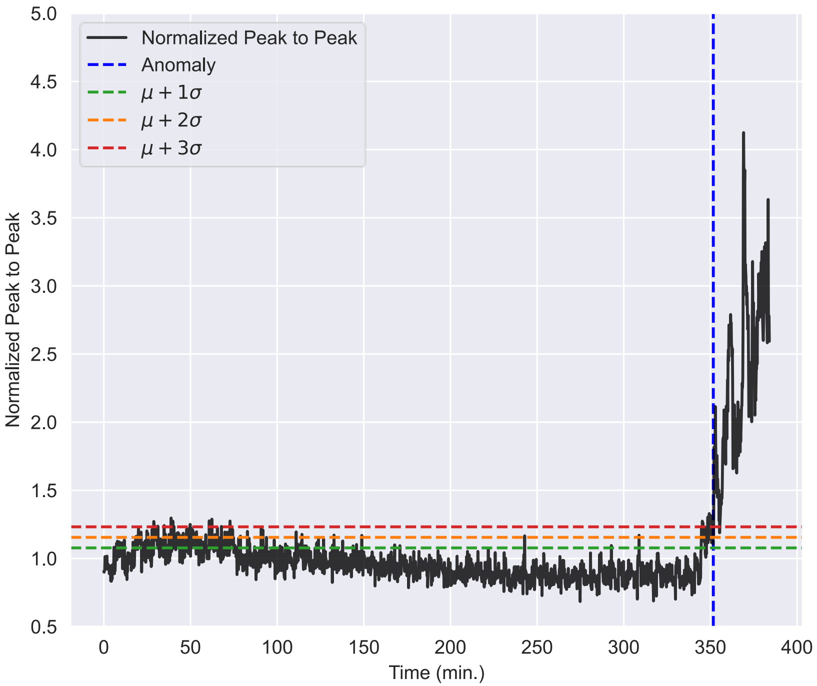
Department of Mechanical Engineering, LMSD - Mecha(tro)nic System Dynamics

Results

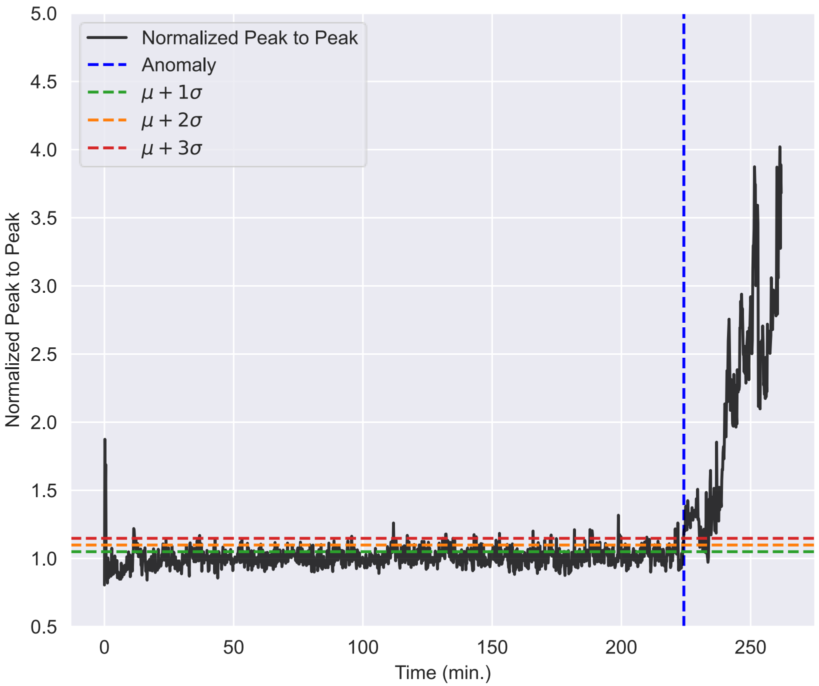
- Anomaly detection criterion
  - 3 consecutive points above the highest threshold



A154



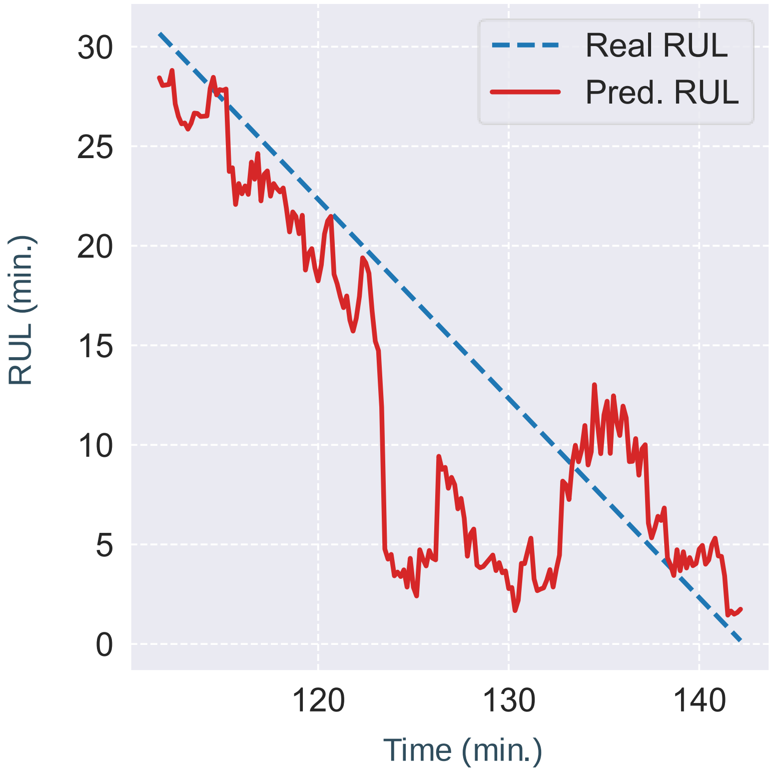
A155



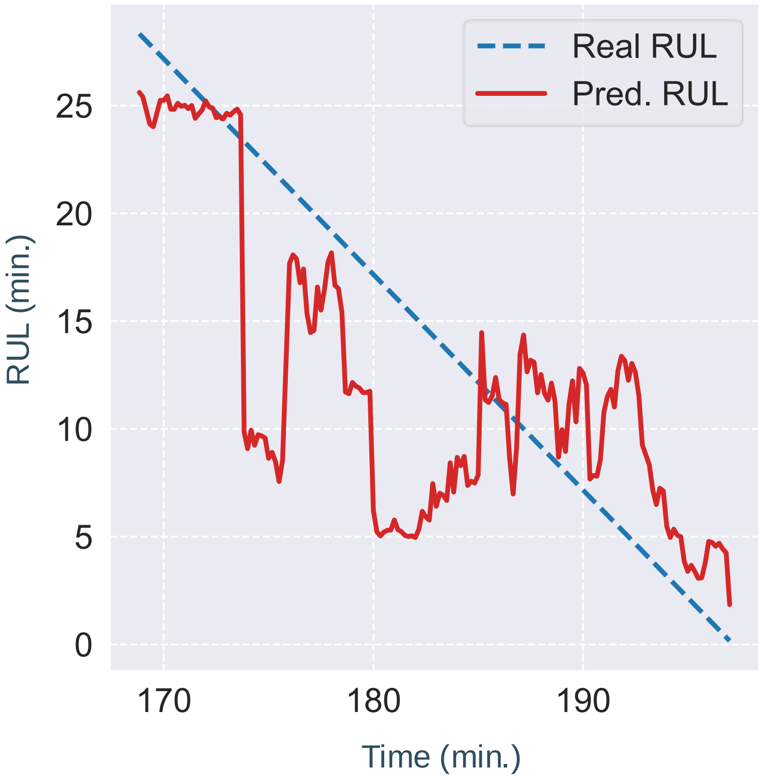
A156

Results

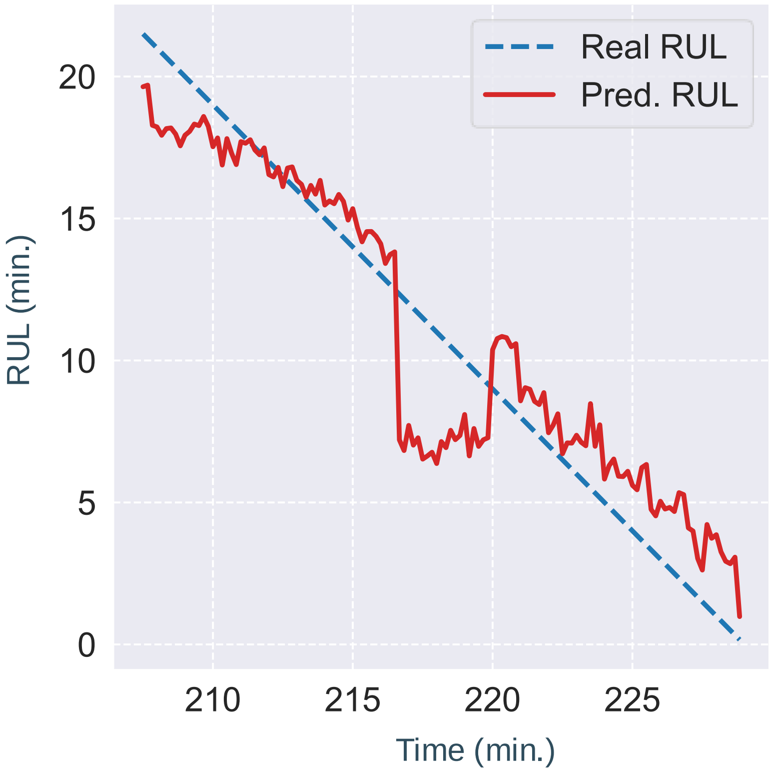
➢ RUL prediction



A148



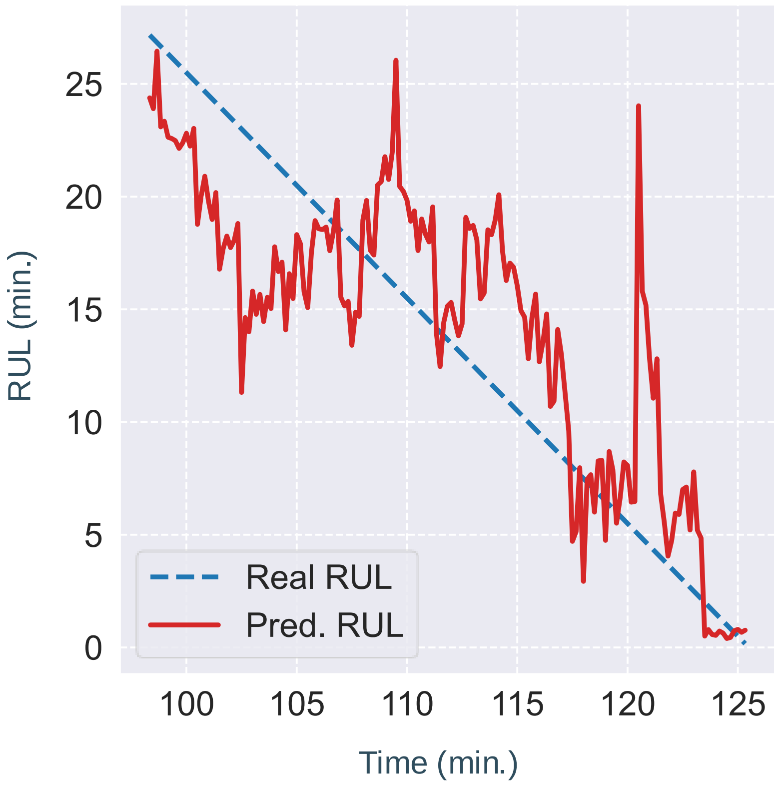
A150



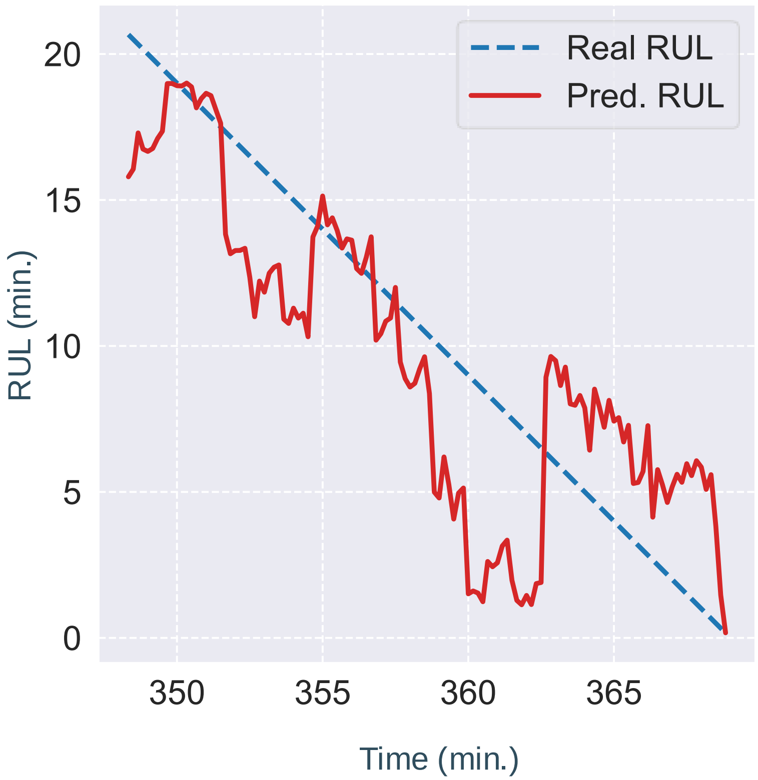
A153

Results

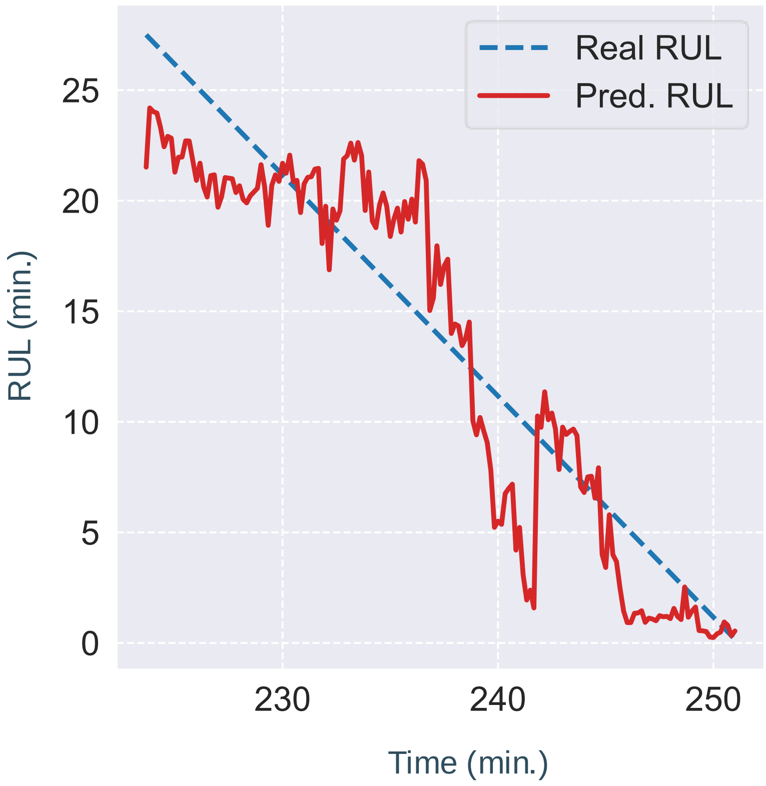
➢ RUL prediction



A154



A155



A156

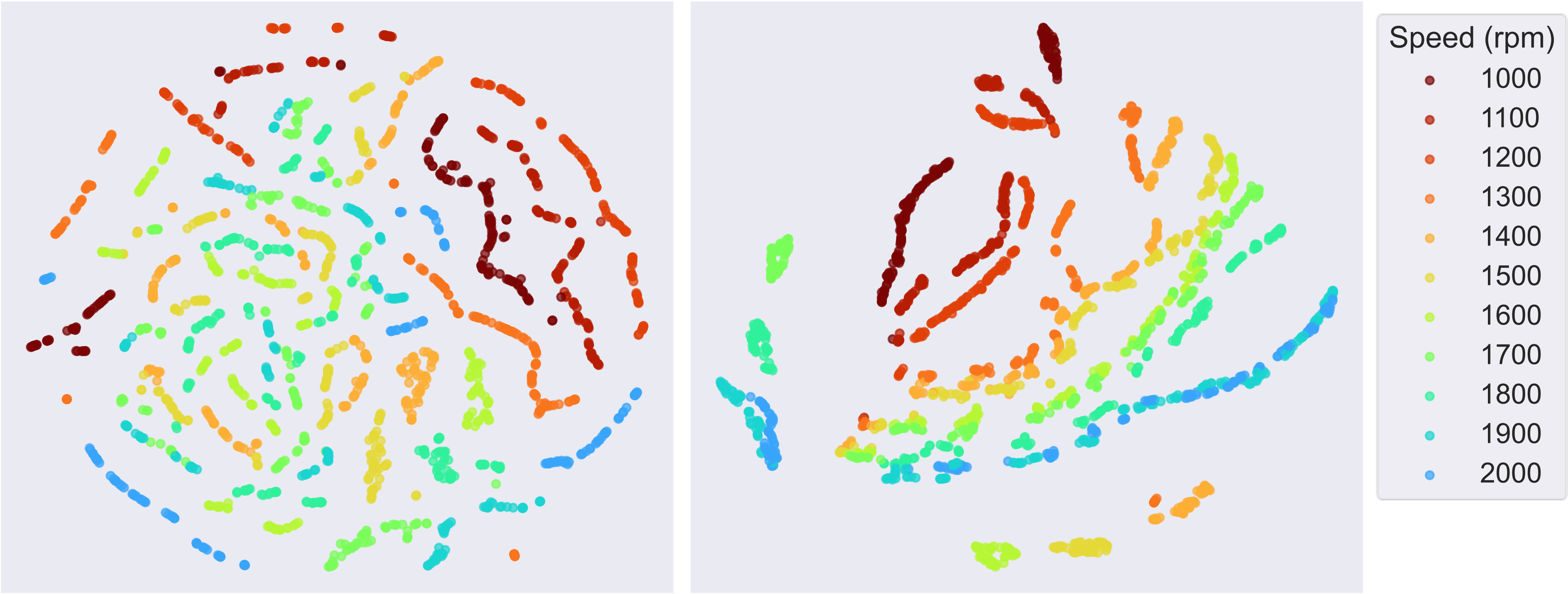
Results

MSE and MAE of the predicted RUL of the SM bearings (in minutes)

Bearing	Error	CNN	DANN	CA-DANN (time context)	**CA-DANN**
A148	MSE	6.22	6.76	6.68	**6.02**
	MAE	4.68	4.91	5.14	**4.51**
A150	MSE	7.46	6.45	7.04	**6.03**
	MAE	6.12	5.42	5.75	**4.71**
A153	MSE	3.80	3.44	3.17	**2.13**
	MAE	3.19	2.65	2.83	**1.77**
A154	MSE	6.69	6.81	6.39	**4.72**
	MAE	5.57	5.60	4.73	**3.81**
A155	MSE	6.97	5.94	6.78	**4.20**
	MAE	5.44	4.79	5.19	**3.58**
A156	MSE	5.44	5.49	4.51	**3.12**
	MAE	4.40	4.79	3.89	**2.58**

Results

- Visualize the deep features of the second to the last layer of regressor part by t-SNE
- CA-DANN makes the feature space aware of the different operating conditions



(a) DANN

(b) CA-DANN

Conclusions

- Transfer Learning has recently emerged as a powerful AI technique
- Leverages knowledge acquired for the source domain to cope with the lack of data, especially faulty data, in the target domain
- The amount of computation power and time can be drastically decreased by leveraging pre learned knowledge from various source domains
- Transfer can be realized between operating conditions and machines
- Simulation-to-real transfer learning can help in solving the data scarcity problem

Open challenges

- How to avoid negative transfer learning?
- When the transfer improves the results?
- How to select the sources domain?
- How complicated should be a virtual model?
- How can the methodologies be applied at a system level?

Acknowledgments

- Flanders Make SBO DGTwin Prediction
- Research Foundation – Flanders (FWO) SBO Robustify
- Flanders AI Research Program
- China Scholarship Council
- European Commission MOIRA MSCA

lmsd_kuleuven

lmsd-kuleuven

lmsd.kuleuven

lmsd-kuleuven

lmsd_kuleuven

KU Leuven Mecha(tro)nic System Dynamics (LMSD)

konstantinos.gryllias@kuleuven.be

www.mech.kuleuven.be/lmsd

VACANCIES www.mech.kuleuven.be/lmsd-joboffers

FLANDERS MAKE LMSD Mecha(tro)nic System Dynamics KU LEUVEN

Thank you