# Propagation of quality requirements for mechanical assemblies

Philipp Hungerländer

March 5, 2020

Alpen-Adria-Universität Klagenfurt

# Table of contents

# Motivation

## Mechanical assemblies

- **Mech. assembly**: product consists of multiple, potentially hundreds of, subcomponents which interact with each other mostly in a non-linear manner.

# Mechanical assemblies

- **Mech. assembly**: product consists of multiple, potentially hundreds of, subcomponents which interact with each other mostly in a non-linear manner.
- **Goal**: understand how to pass down quality requirements from the mech. assembly down to the subcomponents.

# Mechanical assemblies

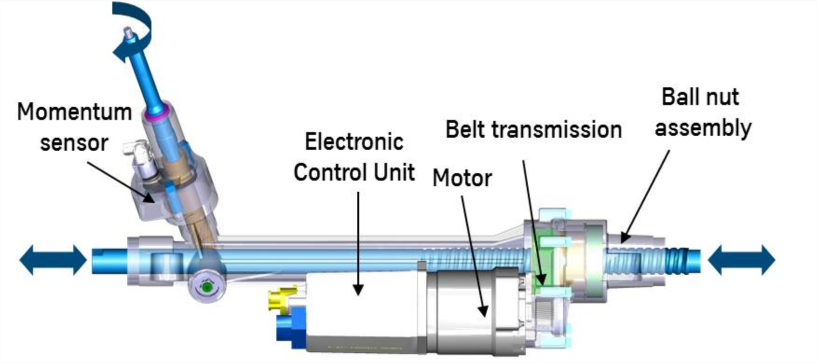
- **Mech. assembly**: product consists of multiple, potentially hundreds of, subcomponents which interact with each other mostly in a non-linear manner.
- **Goal**: understand how to pass down quality requirements from the mech. assembly down to the subcomponents.
- Building an exact mathematical model for the transfer path is impossible due to the system complexity.

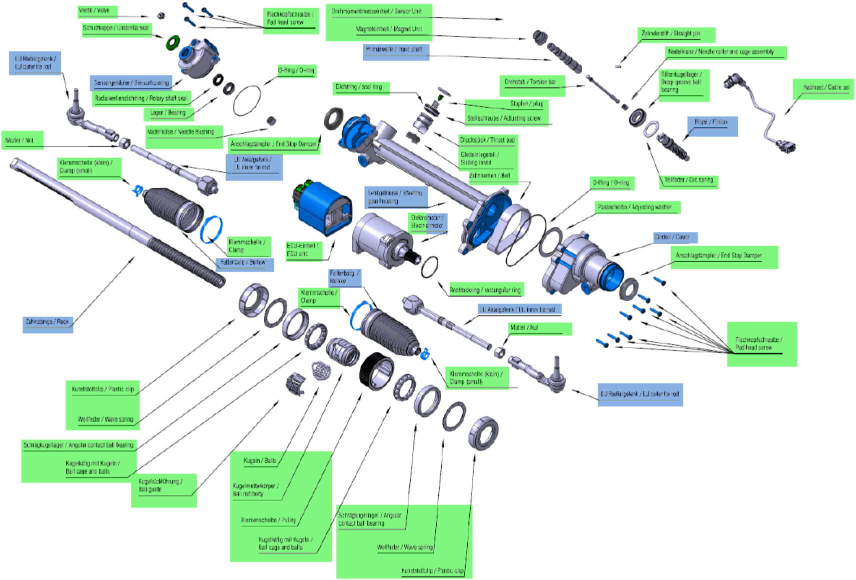
# Mechanical assemblies

- **Mech. assembly**: product consists of multiple, potentially hundreds of, subcomponents which interact with each other mostly in a non-linear manner.
- **Goal**: understand how to pass down quality requirements from the mech. assembly down to the subcomponents.
- Building an exact mathematical model for the transfer path is impossible due to the system complexity.
- **Data-driven optimization**: optimality defined over a finite dataset.

# Specific use-case: steering gear

- **Focus**: connection between the vibroacoustic behavior of steering gear (SG) and ball nut assembly (BNA), a major vibroacoustic contributor of the steering gear.
- **Goal**: understand how to pass down vibroacoustic quality requirements from the SG down to the BNA.

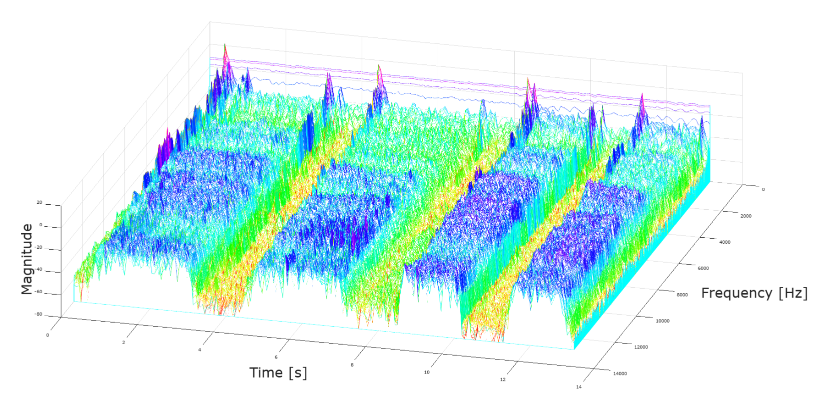




Ventil / Valve
Schutzkappe / Umbrella seal
Flachkopfschraube / Pad head screw
Drehmomentmesseinheit / Sensor Unit
Magneteinheit / Magnet Unit
Primärwelle / Input shaft
Zylinderstift / Straight pin
Nadelkranz / Needle roller and cage assembly
Rillenkugellager / Deep-groove ball bearing
Kabelset / Cable set
LU Radialgelenk / LU outer tie rod
Sensorgehäuse / Sensorhousing
Radialwellendichtring / Rotary shaft seal
Lager / Bearing
O-Ring / O-ring
Nadelhülse / Needle Bushing
Dichtring / seal ring
Drehstab / Torsion bar
Stopfen / plug
Stellschraube / Adjusting screw
Ritzel / Pinion
Mutter / Nut
Anschlagdämpfer / End Stop Damper
Druckstück / Thrust pad
Gleiteinlageteil / Sliding insert
Zahnriemen / Belt
LU Axialgelenk / LU inner tie rod
Klemmschelle (klein) / Clamp (small)
Tellfeder / Dic spring
O-Ring / O-ring
Lenkgehäuse / Steering gear housing
Passscheibe / Adjusting washer
Elektromotor / Electric motor
Deckel / Cover
Klemmschelle / Clamp
ECU-Einheit / ECU unit
Faltenbalg / Bellow
Anschlagdämpfer / End Stop Damper
Faltenbalg / Bellow
Klemmschelle / Clamp
Rechteckring / rectangular ring
LU Axialgelenk / LU inner tie rod
Zahnstange / Rack
Mutter / Nut
Flachkopfschraube / Pad head screw
Klemmschelle (klein) / Clamp (small)
LU Radialgelenk / LU outer tie rod
Kunststoffclip / Plastic clip
Wellfeder / Wave spring
Schrägkugellager / Angular contact ball bearing
Kugelkäfig mit Kugeln / Ball cage and balls
Kugelrückführung / Ball guide
Kugeln / Balls
Kugelmutterkörper / Ball nut body
Riemenscheibe / Pulley
Kugelkäfig mit Kugeln / Ball cage and balls
Schrägkugellager / Angular contact ball bearing
Wellfeder / Wave spring
Kunststoffclip / Plastic clip

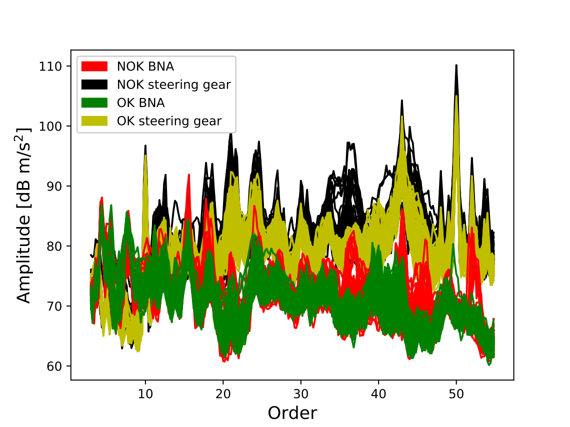
# Current quality test

Both the BNA and the SG undergo the same vibroacoustical quality test, which consists of turning the product left, then right, at two different rotational speeds.



# Current quality representation

The vibrational signals resulting from the test are considered as order spectra, where the orders represent frequencies which correspond to the rotational velocity or its multiples.



# Problem Description

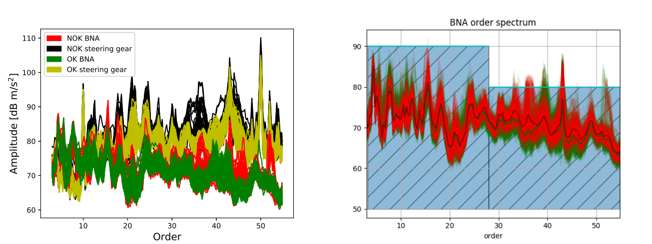
# Problem description

- Two **data matrices** containing the BNA and SG order spectra.
- Set of **binary quality labels** describing whether the SG failed its vibroacoustic test due to the BNA.
- **Goal**: build classifier which for each BNA predicts whether it will cause the corresponding SG to be faulty.
- **No-information-rate**: carefully choose accuracy metric, prevalence of good parts of 0.965.
- **Metric: Cohen Kappa** $\kappa(f) := \frac{p_o - p_e}{1 - p_e}$, where $p_o$ denotes the relative observed agreement and $p_e$ represents the hypothetical probability of chance agreement between the binary labels and the quality predictions for the BNA.
- **Metric: Business Cost** Weighted sum of false positives and false negatives, prioritizing false positive recognition with positive integer factor $\epsilon$.

# Solution Approaches

# Quality zones

- Current quality examination method consists of a **binary classification** of spectral signals.
- Set tight thresholds for the BNA orders which correspond to peculiar steering gear orders (and are thus responsible for *NOK* in the steering gear test).
- **BNA OK** if its order spectrum is majorized by a predefined piecewise constant function and **NOK** if there occurs any violation of the function.
- Alternatively use quality windows which do not occupy the complete spectral grid.



# MILP Notation

- Number of windows to be found denoted by $Z$.
- Number of available order spectra in training dataset denoted by $N$.
- Spectral order grid defined as $S = [s_1, \ldots, s_K]$ for ordered integer indexes $1, \ldots, K$.
- Maximum spectral amplitude in dataset as $X_{\max} = \max_{\substack{1 \le n \le N, \\ 1 \le k \le K}} X_{n,k}$.
- Binary quality labels available as $y$.
- Window width restricted to minimum $t_{\min}$ and maximum $t_{\max}$.
- Left window borders denoted by $\alpha_i$, right window borders by $\beta_i$, corresponding upper thresholds by $\gamma_i$.
- Recognition of false positives prioritized with positive integer factor $\epsilon$.
- MILP formulation presented in [BAH19].

# MILP Formulation

$$
\begin{aligned}
\min \quad & \sum_{\substack{n\in[N] \\ y(n)=1}} \varepsilon \cdot u_n + \sum_{\substack{n\in[N] \\ y(n)=0}} v_n && && (1)\\
\text{s.t.:} \quad & 1 \leq \alpha_z \leq K - t_{\min} + 1, && \forall z \in [Z], && (2)\\
& t_{\min} + 1 \leq \beta_z \leq K + 1, && \forall z \in [Z], && (3)\\
& \alpha_z < \alpha_{z+1}, && \forall z \in [Z-1], && (4)\\
& \beta_z < \beta_{z+1}, && \forall z \in [Z-1], && (5)\\
& \beta_z \leq \alpha_{z+1}, && \forall z \in [Z-1], && (6)\\
& \alpha_z + t_{\min} \leq \beta_z, && \forall z \in [Z], && (7)\\
& \beta_z \leq \alpha_z + t_{\max}, && \forall z \in [Z], && (8)\\
& \alpha_z - M(1 - \mu_{k,z}) \leq k, && \forall k \in [K],\, z \in [Z], && (9)\\
& k < \alpha_z + M \cdot \mu_{k,z}, && \forall k \in [K],\, z \in [Z], && (10)\\
& \beta_z - M \cdot \lambda_{k,z} \leq k, && \forall k \in [K],\, z \in [Z], && (11)
\end{aligned}
$$

# MILP Formulation

$$
\begin{aligned}
& k < \beta_z + M(1-\lambda_{k,z}), && \forall k \in [K],\ z \in [Z], && (12)\\
& \mu_{k,z} + \lambda_{k,z} - 1 = \psi_{k,z}, && \forall k \in [K],\ z \in [Z], && (13)\\
& \gamma_z - M(1-\psi_{k,z}) \leq b_k, && \forall k \in [K],\ z \in [Z], && (14)\\
& b_k \leq \gamma_z + M(1-\psi_{k,z}), && \forall k \in [K],\ z \in [Z], && (15)\\
& X_{\max} + 1 - M \cdot \psi_{k,z} \leq b_k, && \forall k \in [K],\ z \in [Z], && (16)\\
& b_k \leq X_{\max} + 1 + M \cdot \psi_{k,z}, && \forall k \in [K],\ z \in [Z], && (17)\\
& X_{n,k} - M(1-g_{n,k}) \leq b_k, && \forall n \in [N],\ k \in [K],\ y(n)=1, && (18)\\
& b_k < X_{n,k} + M \cdot g_{n,k}, && \forall n \in [N],\ k \in [K],\ y(n)=1, && (19)\\
& K \cdot u_n \leq \textstyle\sum_{k=1}^{K} g_{n,k}, && \forall n \in [N], && (20)\\
& \textstyle\sum_{k=1}^{K} g_{n,k} \leq K + u_n - 1, && \forall n \in [N], && (21)\\
& b_k - M \cdot (1-h_{n,k}) < X_{n,k}, && \forall n \in [N],\ k \in [K],\ y(n)=0, && (22)
\end{aligned}
$$

# MILP Formulation

$$
\begin{aligned}
&X_{n,k} \leq b_k + M \cdot h_{n,k}, && \forall n \in [N],\ k \in [K],\ y(n) = 0, && (23)\\
&v_n \leq \textstyle\sum_{k=1}^{K} h_{n,k}, && \forall n \in [N], && (24)\\
&h_{n,k} \leq v_n, && \forall n \in [N],\ k \in [K], && (25)\\
&u_n,\ v_n \in \{0,1\}, && \forall n \in [N], && (26)\\
&g_{n,k},\ h_{n,k} \in \{0,1\}, && \forall n \in [N],\ k \in [K], && (27)\\
&\mu_{k,z},\ \lambda_{k,z},\ \psi_{k,z} \in \{0,1\}, && \forall k \in [K],\ z \in [Z], && (28)\\
&\alpha_z \in [K - t_{\min} + 1] && \forall z \in [Z], && (29)\\
&\beta_z \in \{t_{\min} + 1, \ldots, K + 1\} && \forall z \in [Z]. && (30)
\end{aligned}
$$

# GA solving approach

Pursue **alternate direction** approach, iteratively maximizing $\kappa$ w.r.t the thresholds while keeping change points fixed and vice versa [BFH19b].

- Maximization w.r.t. change points carried out by genetic algorithm.
- Maximization w.r.t. threshold values carried out via Nelder-Mead downhill simplex method.

Repeat procedure for $k = 1, 2, \ldots$ and choose $k^*$ manually with domain experts to optimize **trade-off** between **solution quality** and **solution complexity**.

## Algorithm

**function** RUN(order spectra $X$, binary quality labels, number of change points $k$, maximum number of generations $B$)
  Initialize population $\pi[0]$ of 100 randomly generated functions in $\mathcal{P}_k$
  **for** $i \leftarrow 1, B$ **do**
    *Update current population*
    **for all** $f$ in $\pi[i-1]$ **do**
      Update threshold values for $f$ such that $\kappa_{\text{train}}(f)$ is maximized while keeping the change points constant
    **end for**
    *Breed next generation*
    $\pi[i] \leftarrow \mathit{Evolve}(\pi[i-1])$
  **end for**
  **return** Optimized piecewise constant threshold function $f^* \in \mathcal{P}_k$
**end function**

# Update

- **Keep change point positions constant, adapt threshold values**.
- Data-driven objective function, the mapping $f \mapsto \kappa_{\text{train}}(f)$ is not differentiable w.r.t to the threshold values.
- Employ the **Nelder-Mead downhill simplex** method, a **gradient-free** maximization technique.
- Stop the iteration when the increments in $\kappa_{\text{train}}$ and $f_i$ both are less than $10^{-4}$.

# Evolution

- In each iteration retain those 20% individuals with the highest fitness value for reproduction, add 5% randomly generated ones.
- Pairs of step functions are chosen from these individuals at random with replacement and crossed-over to generate offspring, refilling the population.
- A proportion of 15% of the offspring is mutated after generation.

# Cross-over

- Merge the $k$ change points of the two parent functions, run agglomerative cluster algorithm on the $2k$ positions.
- Cut resulting dendrogram so that $k$ clusters result, with the centroids of these clusters chosen as the offspring change points.
- Average the function values of the parent functions over the new intervals to obtain new values.

# Mutation

- Change point locations are randomly shifted with a probability of 0.15, under the constraint of preservation of order.
- The maximal allowed deviation is additionally constrained depending on the order.
- Use weight functions obtained via **deep canonical correlation analysis** to introduce larger change point variation in highly correlating areas of the spectra.

## Challenges

- The mathematical model of the solution as a piecewise constant function with multiple change points and covering the complete spectral order grid $S$ implicitly assumed that each order contributes information relevant for the classification and should thus be subject to a threshold.
- The Nelder-Mead downhill simplex method is computationally expensive and its gains were significantly diminished after a random mutation or after the averaging crossover mechanism, which acted as a serious perturbation.
- Crossover: if the parent solutions focus on different order intervals responsible for different root faults, the crossover technique is unlikely to find itself a root fault by clustering and averaging.

# Enhancement

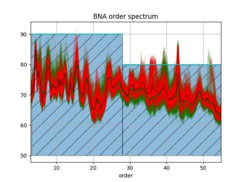
- Enhanced version of this algorithm proposed in [BAH19].
- Impose upper thresholds only on select areas of the order spectra.
- Renounce to the use of the computationally heavy Nelder-Mead method.
- More efficient crossover mechanism which selects the best windows from both parents based on their achieved cost reduction.
- Mutations are only accepted if they help reduce cost; mutation strength is diminished linearly with each generation, finally reaching a value of 0 for the last generation.

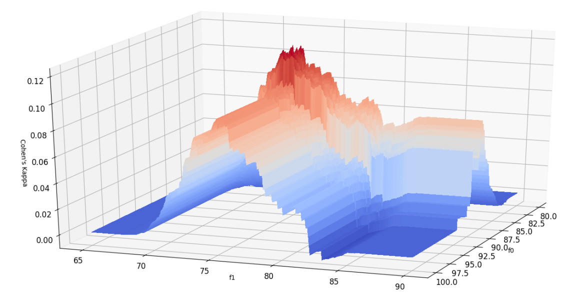
# Enhanced crossover mechanism

- Selection of two parents at random with replacement.
- For each window of each parent, it computes the confusion matrix resulting from the application of the window's upper threshold and the associated cost resulting from the number of false positives and false negatives.
- After a window is selected, other overlapping windows are disregarded.
- Confusion matrices of all other window possibilities are updated, removing from the cost calculation the order spectra already violating previous windows.
- Benefit: the best non-overlapping windows from either parent are selected.
- Benefit: the negative cost impact of a window is diminished if the false negatives were already misclassified by another window.

# Response surface

Response surface is not so smooth for current problem when using data-driven optimization.



# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.

# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.
- **Noise** is present in the data, and we may end up setting thresholds on noise.

# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.
- **Noise** is present in the data, and we may end up setting thresholds on noise.
- Necessary to find out which BNA curve patterns correlate with other patterns in the SG curves (**mutual information**).

# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.
- **Noise** is present in the data, and we may end up setting thresholds on noise.
- Necessary to find out which BNA curve patterns correlate with other patterns in the SG curves (**mutual information**).
- Consider the SG vibroacoustic behavior as another view of the BNA behavior containing heavy noise [BFH19a, BFH19b].

# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.
- **Noise** is present in the data, and we may end up setting thresholds on noise.
- Necessary to find out which BNA curve patterns correlate with other patterns in the SG curves (**mutual information**).
- Consider the SG vibroacoustic behavior as another view of the BNA behavior containing heavy noise [BFH19a, BFH19b].
- Idea: find new lower-dimensional representations of the BNA and SG curves which are highly linearly correlated.

# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.
- **Noise** is present in the data, and we may end up setting thresholds on noise.
- Necessary to find out which BNA curve patterns correlate with other patterns in the SG curves (**mutual information**).
- Consider the SG vibroacoustic behavior as another view of the BNA behavior containing heavy noise [BFH19a, BFH19b].
- Idea: find new lower-dimensional representations of the BNA and SG curves which are highly linearly correlated.
- If there is noise in either view that is uncorrelated with the other view, the learned representations do not contain the noise.

# Mutual information

- The previous approach disregards the SG curves, only uses the BNA curves and SG labels.
- **Noise** is present in the data, and we may end up setting thresholds on noise.
- Necessary to find out which BNA curve patterns correlate with other patterns in the SG curves (**mutual information**).
- Consider the SG vibroacoustic behavior as another view of the BNA behavior containing heavy noise [BFH19a, BFH19b].
- Idea: find new lower-dimensional representations of the BNA and SG curves which are highly linearly correlated.
- If there is noise in either view that is uncorrelated with the other view, the learned representations do not contain the noise.
- The maximally correlated encodings should be the most predictive of, and by, each other.

# Linear Canonical Correlation Analysis

- Assume two random vectors $(\hat{X}, \hat{Y}) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ with covariances $(\Sigma_{11}, \Sigma_{22})$ and cross-covariance $\Sigma_{12}$.
- CCA computes orthogonal sets of latent scores $(w_1^{'}\hat{X}, w_2^{'}\hat{Y})$, obtained using:

$$(w_1^*, w_2^*) = \underset{w_1, w_2}{\arg\max}\, \mathrm{corr}(w_1^{'}\hat{X}, w_2^{'}\hat{Y}) = \underset{w_1, w_2}{\arg\max} \frac{w_1^{'}\Sigma_{12}w_2}{\sqrt{w_1^{'}\Sigma_{11}w_1 w_2^{'}\Sigma_{22}w_2}}.$$

- The latent scores are maximally correlated linear combinations of the $\hat{X}$ and $\hat{Y}$ vectors.
- Projections required to have unit variance and subsequent projections to be uncorrelated with previous ones.

# Linear Canonical Correlation Analysis

- Assemble the top $k$ projection weight vectors $w_1^i$ into the columns of a matrix $A_1 \in \mathbb{R}^{n_1 \times k}$ (analog $A_2$).
- **Optimization problem** for the top $k \leq \min(n_1, n_2)$ projection pairs:

$$\begin{array}{ll} \text{maximize} & \operatorname{Tr}(A_1^{'} \Sigma_{12} A_2) \\ \text{subject to} & A_1^{'} \Sigma_{11} A_1 = A_2^{'} \Sigma_{22} A_2 = I, \end{array}$$

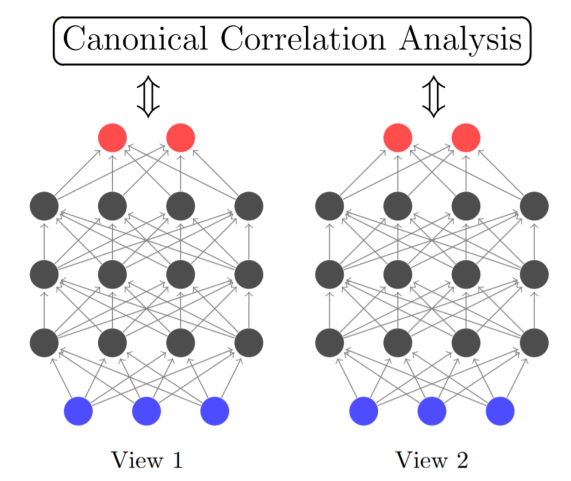
- Define $T := \Sigma_{11}^{-\frac{1}{2}} \Sigma_{12} \Sigma_{22}^{-\frac{1}{2}}$.
- **Solution**: assuming non-singularity of covariance matrices, maximal achievable correlation computed by summing the top k singular values of $T$.
- **Optimal value** encountered at $(A_1^*, A_2^*) = (\Sigma_{11}^{-\frac{1}{2}} U_k, \Sigma_{22}^{-\frac{1}{2}} V_k)$, where $U_k, V_k$ are the matrices of the first $k$ left-, respectively right-singular vectors of $T$.

# Deep CCA

- Assume training size of $m$ and final encoding length of $o$ in the final layer of each neural network.
- Introduce $H_1 \in \mathbb{R}^{o \times m}, H_2 \in \mathbb{R}^{o \times m}$ as matrices resulting from each neural networks.
- Use centered data matrices $\bar{H}_1 = H_1 - \frac{1}{m} H_1 \mathbf{1}$ and $\bar{H}_2$ to estimate the covariance matrices $\hat{\Sigma}_{11} = \frac{1}{m-1} \bar{H}_1 \bar{H}_1^{'} + r_1 I$, $\hat{\Sigma}_{22}$ and cross-covariance $\hat{\Sigma}_{12} = \frac{1}{m-1} \bar{H}_1 \bar{H}_2^{'}$, where $r_1 > 0, r_2 > 0$ denote regularization constants.
- Correlation: sum top k singular values of T, if o=k then we have $\mathrm{corr}(H_1, H_2) = \mathrm{Tr}\,(T^{'} T)^{\frac{1}{2}}$.

# Deep CCA

New encodings **not ordered** by explained correlation like in linear CCA, **perform linear CCA on top** of outputs of the neural networks.



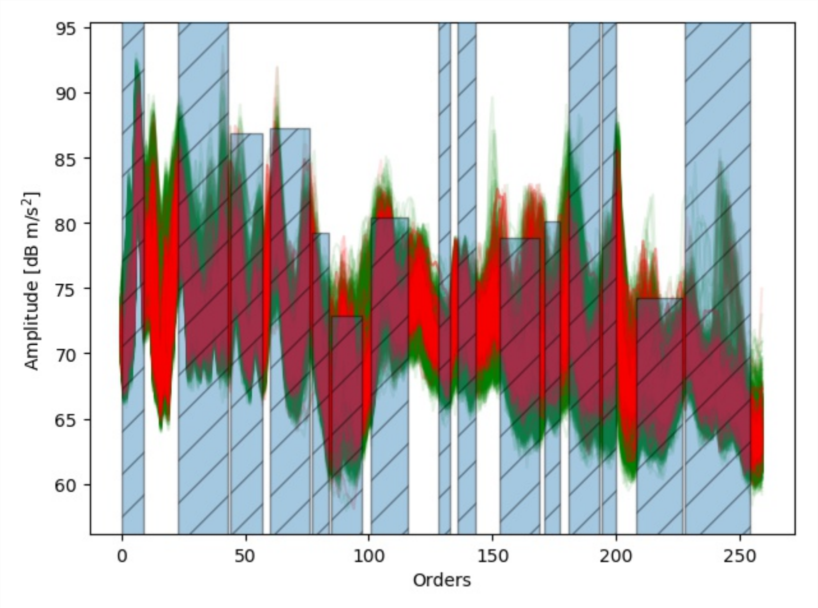
# Computational Experiments and Results

# MILP results for data subsets

| | Number of spectra | Time [s] | | GA parameters | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | MILP | GA | *a* | *b* | *c* | *d* | *e* | *f* |
| Optimal value: 0<br>Number of windows: 3 | 10 | 302.43 | 14.53 | 0.4 | 0.1 | 0.5 | 0.7 | 100 | 100 |
| | 16 | 31274.52 | 17.26 | 0.4 | 0.1 | 0.5 | 0.7 | 100 | 100 |
| | 20 | 5397.47 | 16.05 | 0.2 | 0.5 | 0.3 | 0.8 | 100 | 100 |
| | 24 | 5876.58 | 17.08 | 0.4 | 0.2 | 0.4 | 0.7 | 100 | 100 |
| Optimal value: 1<br>Number of windows: 3 | 10 | 5405.22 | 46.76 | 0.3 | 0.3 | 0.4 | 0.7 | 200 | 100 |
| | 16 | 65968.40 | 14.31 | 0.4 | 0.3 | 0.3 | 0.5 | 100 | 100 |
| | 18 | 66945.91 | 120.86 | 0.3 | 0.2 | 0.5 | 0.4 | 500 | 200 |
| | 30 | 175994.50 | 57.62 | 0.3 | 0.3 | 0.4 | 0.7 | 200 | 200 |
| Optimal value: 0<br>Number of windows: 5 | 10 | 207.54 | 16.58 | 0.4 | 0.1 | 0.5 | 0.7 | 100 | 100 |
| | 12 | 56644.21 | 58.83 | 0.3 | 0.2 | 0.5 | 0.7 | 200 | 200 |
| | 16 | 37592.43 | 1463.14 | 0.4 | 0.3 | 0.3 | 0.8 | 1000 | 1000 |
| Optimal value: 1<br>Number of windows: 5 | 20 | 57721.81 | 90.02 | 0.4 | 0.3 | 0.3 | 0.7 | 300 | 200 |

**Table 1:** Comparison of the results of the MILP approach and of the genetic algorithm (GA) in terms of solution quality and computation time on the 12 distinct datasets described in [BAH19] using either 3 or 5 quality windows.

# Optimal thresholds of the enhanced GA



# Results of the enhanced GA

| | Minimum window length | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5 | | 10 | | 20 | | 30 | | 40 | | 50 | |
| Number of windows | Cost | Time [s] | Cost | Time [s] | Cost | Time [s] | Cost | Time [s] | Cost | Time [s] | Cost | Time [s] |
| 1 | 1388.67 | 138.41 | 1235.17 | 81.64 | 1230.33 | 104.29 | 1229.17 | 95.07 | 1266.50 | 84.33 | 1222.83 | 83.06 |
| 2 | 1095.83 | 166.98 | 1089.67 | 140.43 | 1099.50 | 212.09 | 1091.83 | 176.93 | 1091.83 | 158.96 | 1102.17 | 152.93 |
| 3 | 1058.67 | 247.61 | 1074.67 | 220.12 | 1075.17 | 362.42 | 1088.50 | 281.47 | 1092.33 | 258.08 | 1096.17 | 260.85 |
| 4 | 1011.50 | 295.41 | 1021.33 | 287.71 | 1031.00 | 511.69 | 1020.83 | 325.29 | 1050.83 | 327.59 | 1113.17 | 331.27 |
| 5 | 997.83 | 357.72 | 1037.50 | 347.34 | 1030.67 | 623.63 | 1057.00 | 353.56 | 1108.83 | 397.18 | 1090.33 | 329.29 |
| 6 | 956.17 | 436.35 | 988.83 | 424.36 | 1011.83 | 868.24 | 1056.17 | 497.46 | 1059.83 | 451.06 | | |
| 7 | 947.00 | 524.56 | 948.17 | 506.52 | 1034.67 | 839.72 | 1007.33 | 535.40 | | | | |
| 8 | 939.33 | 607.68 | 946.00 | 570.59 | 1019.17 | 808.69 | 1057.00 | 524.85 | | | | |
| 9 | 895.50 | 721.20 | 928.50 | 796.97 | 984.17 | 936.60 | | | | | | |
| 10 | 844.17 | 795.09 | 886.67 | 1254.24 | 1018.00 | 1106.82 | | | | | | |
| 11 | 869.17 | 859.47 | 887.67 | 1587.75 | 1003.50 | 1030.41 | | | | | | |
| 12 | 849.50 | 924.95 | 938.17 | 1454.65 | 1009.50 | 1091.94 | | | | | | |
| 13 | 857.67 | 1014.22 | 900.17 | 1445.38 | | | | | | | | |
| 14 | 805.17 | 1092.32 | 944.33 | 1993.66 | | | | | | | | |
| 15 | 801.50 | 1184.60 | 879.33 | 1574.93 | | | | | | | | |

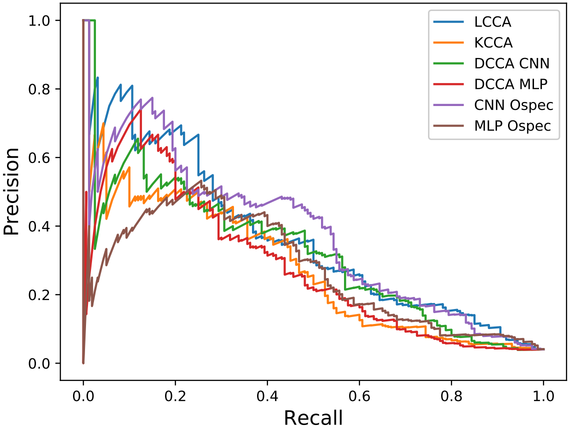
**Table 2:** For each combination of required number of quality windows and minimum window length, the results give insights into the obtained solution quality in terms of incurred cost on the test data and computation time on the training data. The best solution features a cost of 801.50, which implies a scrap cost reduction of 49.91% with respect to the currently employed BNA quality classification method.

# Results CCA and neural network approaches

| | LCCA | | KCCA | | DCCA MLP | | DCCA CNN | | MLP Ospec | | CNN Ospec | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sup. | Semi | Sup. | Semi | Sup. | Semi | Sup. | Semi | Sup. | Semi | Sup. | Semi |
| Cohen Kappa | 0.23 | 0.35 | 0.27 | 0.36 | 0.31 | 0.32 | 0.33 | 0.39 | 0.37 | 0.39 | 0.43 | 0.45 |
| AUPR | 0.22 | 0.37 | 0.20 | 0.28 | 0.29 | 0.28 | 0.34 | 0.32 | 0.33 | 0.27 | 0.41 | 0.38 |
| AUROC | 0.79 | 0.87 | 0.72 | 0.78 | 0.75 | 0.76 | 0.81 | 0.81 | 0.87 | 0.84 | 0.88 | 0.86 |
| Matthews C.C. | 0.25 | 0.36 | 0.27 | 0.36 | 0.32 | 0.33 | 0.34 | 0.39 | 0.38 | 0.39 | 0.43 | 0.45 |
| Specificity | 0.18 | 0.40 | 0.31 | 0.36 | 0.28 | 0.29 | 0.46 | 0.48 | 0.51 | 0.40 | 0.43 | 0.47 |
| Sensitivity | 0.99 | 0.97 | 0.97 | 0.98 | 0.98 | 0.98 | 0.96 | 0.96 | 0.96 | 0.98 | 0.98 | 0.98 |
| Youden J | 0.17 | 0.37 | 0.28 | 0.34 | 0.26 | 0.28 | 0.41 | 0.45 | 0.46 | 0.38 | 0.41 | 0.45 |

**Table 3:** Performance of the different methods, ranging from the CCA components being classified by SVM to the MLP and CNN Neural Networks learning directly on the order spectra [BHF19].

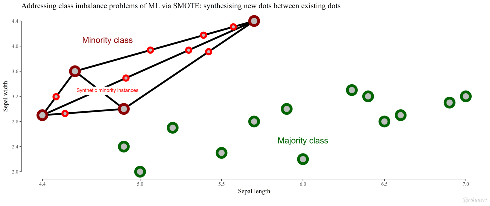
# Recall Precision



# Further work

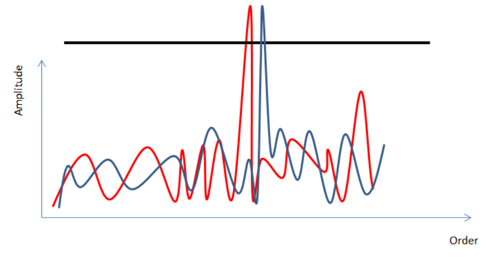
# Heavily imbalanced data

- **Performance:** metrics which consider the imbalance ratio.
- **Learning:** assign higher weights to existing underrepresented class samples or rebalance dataset.
- **Rebalancing:** Oversampling, downsampling or combination of both.
- **Select methods:** SMOTE, SMOTE-Tomek, SMOTE-ENN.
- **Synthetic samples**: on the line btw. two data points.



# Synthetic samples

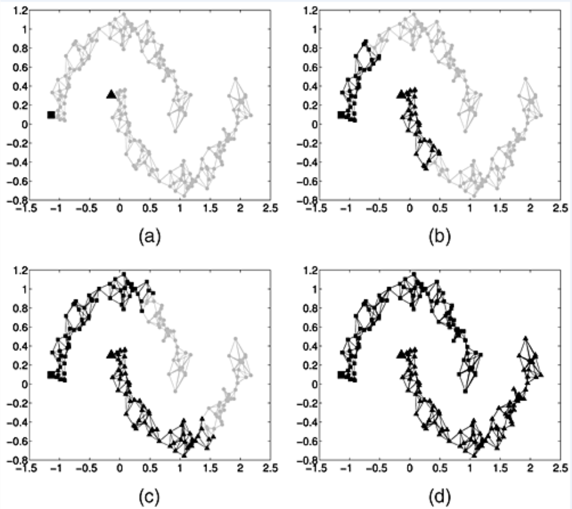
- **Problem:** synthetic samples on line btw. two data points adequate for discrete/distinct features.
- **Measurement curves:** underlie measurement noise, x-coordinate of feature responsible for quality label may vary slightly.
- Both 'parents' $x, y$ feature a peak, but at minimally different x-axis coordinate.
- **Problem are not the synthetic samples, but their labels.**



# Label Synthetic samples

- **Idea:** generate synthetic samples as usual, but allow their labels to be learned.
- **Label Spreading:** Zhou et al. - Learning with local and global consistency (2004).
- Possible to use learned labels only for unlabeled synthetic samples, or also for original labeled data.
- Alternative for dealing with noisy samples instead of rule-based removal decisions.
- Usage of 'clamping' factor $\alpha$ affects probability of a sample adopting the information from its neighbours instead of its initial label.

# Label Spreading



# Conclusion

# Recapitulation

- Introduced the quality propagation problem for mechanical assemblies.

# Recapitulation

- Introduced the quality propagation problem for mechanical assemblies.
- Presented a first solution approach based on MILP and genetic algorithms.

# Recapitulation

- Introduced the quality propagation problem for mechanical assemblies.
- Presented a first solution approach based on MILP and genetic algorithms.
- Proposed a second solution framework based on mutual information between measurement curves.

# Recapitulation

- Introduced the quality propagation problem for mechanical assemblies.
- Presented a first solution approach based on MILP and genetic algorithms.
- Proposed a second solution framework based on mutual information between measurement curves.
- Presented the weaknesses of rebalancing algorithms for curves and a potential solution approach.

Paul Alexandru Bucur, Philipp Armbrust, and Philipp Hungerländer, *On the propagation of quality requirements for mechanical assemblies in industrial manufacturing*, Manuscript in revision., 2019.

Paul Alexandru Bucur, Klaus Frick, and Philipp Hungerländer, *Correlation Analysis Between the Vibroacoustic Behavior of Steering Gear and Ball Nut Assemblies in the Automotive Industry*, EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization (Cham), Springer International Publishing, 2019, pp. 1253–1262.

Paul Alexandru Bucur, Klaus Frick, and Philipp Hungerländer, *Predicting the Vibroacoustic Quality of Steering Gears*, Operations Research Proceedings 2018 (Cham), Springer International Publishing, 2019, pp. 309–315.

Paul Alexandru Bucur, Philipp Hungerländer, and Klaus Frick, *Quality classification methods for ball nut assemblies in a multi-view setting*, Mechanical Systems and Signal Processing **132** (2019), no. C, 72–83.

**Questions?**