# GalFer Contest
# Evaluation rules

GalFer Contest Organising Committee

December 11, 2024

# Contents

# Chapter 1

# Contest problem definition

The main aim of GalFer Contest is to assess the new opportunities brought by the use of surrogate data-driven models of electrical machines performance. Under this viewpoint, the models and procedures proposed by the Teams will be evaluated on the basis of three criteria, two based on the numerical assessment of procedure results and one on the novelty and efficiency of the approach.
The evaluation criteria are based on:

- *interpolation*: how the surrogate model is able to reconstruct the input-output relationship on a given motor type data-set;
- *extrapolation*: how the surrogate model trained on two data-sets is able to extrapolate its prediction on a new size of motor of the same typology assessing thus the *degree of innovation* of the surrogate algorithm.
- *novelty*: how the approach is able to provide useful design insights or novel contributions to the field that can be expressed either by the definition of new methodologies or by new applications of existing methods to the motor design.

The first two rankings will be provided by the *Contest Organizing Committee* processing the results of the Teams procedures and will be supervised by the *Advisory Board*. The *novelty* evaluation will be carried out by the *Advisory Board* and will be based on *reports* provided by the Teams and describing the methodology and implementation of the of the data driven procedure. The report should be a technical document up to 5 pages in IEEE double column format [1].

### Multi-objective problem definition

The problem is described by datasets linking together a set of inputs describing the geometry, structure and supply conditions of the motor and a set of outputs made up by a series of motor performance evaluated by a set of finite element

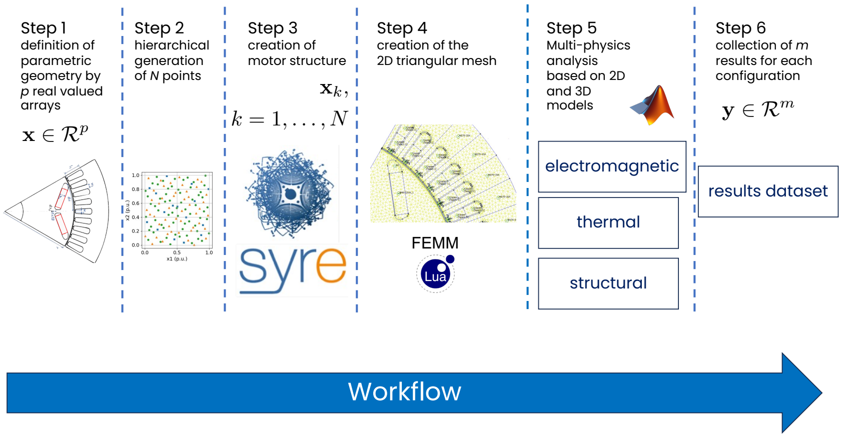



Figure 1.1: Flowchart describing the ensemble of procedures used to create the datasets.

based procedures. The procedures used to link the set of inputs $\mathbf{x}$ to the one of outputs $\mathbf{y}$ are outlined in figure 1.1. More details can be found on the GitHub page [6].

The array of outputs belongs to a 7 dimensional real space. As the aim of the surrogate problem is to address the multi-objective assessment of motor performance, to facilitate the gradual assessing of the whole procedure an increasing difficulty of the multi-objective task is considered. In particular, the following *sub-problems* will be considered:

$$
\begin{array}{l}
2D_1 \rightarrow O_1 = -T, O_2 = T_r \\
2D_2 \rightarrow O_1 = -T, O_2 = VM \\
2D_3 \rightarrow O_1 = -T, O_2 = Temp \\
3D_1 \rightarrow O_1 = -T, O_2 = VM, O_3 = Temp
\end{array}
$$

The negative value of torque $T$ is considered to have all problems looking for a minimum of each objective. Anyway, it must be remarked that **the final ranking will be performed on the full $7$-th dimensional problem**.

The datasets provided are considered *sufficient* for defining the surrogate model and data provided are considered as the *ground truth* for the Contest. Datasets should be the only source of information for the surrogate models provided by the **Teams** taking part in the Contest.

# Chapter 2

# Rules for the *interpolation* phase

The interpolation capabilities will be assessed by linking the surrogate model to a multi-objective optimisation procedure on a single motor model.

## 2.1 what each Team will provide

- Each **Team** will receive a dataset for motors A ($4096$ points). Motor $A$ has a target application in full electric vehicle with a rated torque of $236\ \mathrm{Nm}$, rated power of $120\ \mathrm{kW}$ and a maximum speed $15000\ \mathrm{rpm}$.
- Geometry and supply conditions of motor $A$ are described as function of 8 input variables, whose names and ranges are reported in Table 2.1. Sampling points in design space are generated using a p.u. definition for 3 design variable $d\alpha$, $h_\mathrm{c}$ and $w_\mathrm{o}$. This procedure ensures that these design variables are independent and uniformly distributed in their variation range. The geometry generation routine [2] assigns actual lengths and angles to these variables on the basis of the other geometric variables. The resulting values are thus depending on other inputs. Both values, in p.u. and actual units, will be given in the dataset.
- Motor family $A$ is built on the assumption that the stator outer diameter $R_s$, stack length $h_c$, number of pole pairs $n_p$ and number of slots/pole/phase $n_{spp}$ are fixed at the values reported in Table 2.1.
- The dataset contains $4096$ records each row of the dataset corresponds to one motor configuration and is made up of $8$ variables and $7$ outputs.
- The 7 outputs contained in each row of the dataset are: torque $T$, torque ripple $T_r$, mass of the winding copper $M_{Cu}$, mass of the rotor magnets $M_{mag}$, power factor $\cos(\varphi)$, Von Mises equivalent stress $VM$ and maximum temperature on the winding $Temp$.

Table 2.1: Variable ranges for motor family $A$: variables $1 \div 8$ change in the dataset, while $9 \div 12$ are fixed for motor $A$.

| | Parameter | name | range |
|---|---|---|---|
| 1 | Barrier position (pu) | $d\alpha$ | [0.65 0.85] |
| 2 | Barrier width (pu) | $h_\mathrm{c}$ | [0.3 0.7] |
| 3 | Rotor radius (mm) | $r$ | [60 78] |
| 4 | Tooth width (mm) | $w_\mathrm{t}$ | [3.8 6.3] |
| 5 | Tooth length (mm) | $l_\mathrm{t}$ | [15 22.5] |
| 6 | Slot opening (pu) | $w_\mathrm{o}$ | [0.1 0.4] |
| 7 | Barrier shift (mm) | $dxIB$ | [-4 6] |
| 8 | Current phase angle (°) | $\gamma$ | [30 60] |
| 9 | Stator outer diameter (mm) | $R_s$ | 225 |
| 10 | Stack length (mm) | $h_c$ | 134 |
| 11 | Number of pole pairs | $n_p$ | 3 |
| 12 | Number of slots/pole/phase | $n_{spp}$ | 3 |

- *constraints*: the physical realisation of the electric motor imposes constraints on maximum Von Mises stress $VM < VM^{(max)} = 450\ \mathrm{MPa}$ and on maximum temperature $Temp < Temp^{(max)} = 180°$. Some of the points in the dataset contains values that are violating the constraints but they are kept to give a complete exploration of the design space. Anyway, **values violating constraints will have to be removed from the surrogate model results**.
- Each **Team** will send to the GalFer organizers one data-driven surrogate model as a procedure. Scripts can be either as a `Matlab` or a `Python` procedure (the **Procedure**).
- Each **Team** will send to the GalFer organizers a report describing the approach used and highlighting the novel contributions of their approach (the **Report**).
- The **Procedure** will have to take a 8-dimensional array in input and give as result a 7-dimensional array. If different procedures would have been devised for some of the outputs, they will have to be *bundled up* in one single function.

## 2.2 what GalFer Organizers will do

The procedure is based on the algorithm proposed in [7], Figures 2.1 and 2.2 are used to explain the procedure steps in a two dimensional Pareto space.

The GalFer Organizers will perform the following actions:

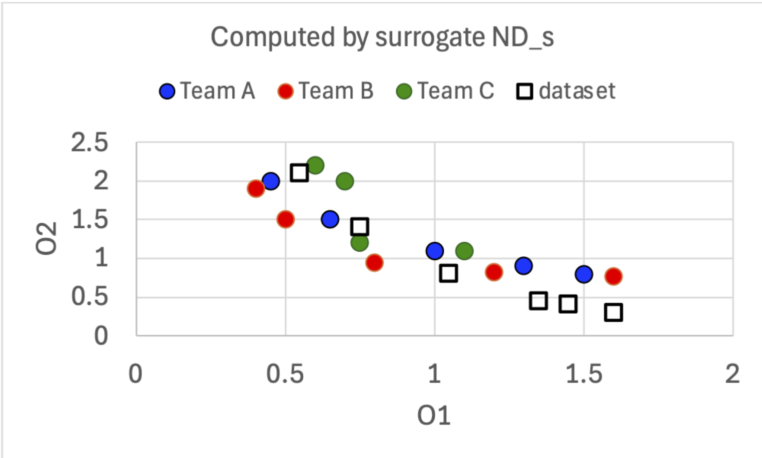
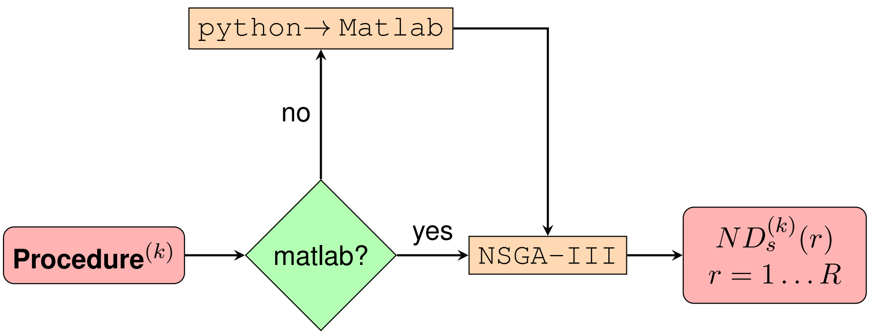



Figure 2.1: $ND_s$ collection of all non dominated points found by the **Teams** through **Procedures** (surrogate) and of all non dominated points belonging to the original dataset

1. link the **Procedures** to a Multi-Objective the optimization procedure `NSGA-III` (the **Optimizer**). The `Matlab` toolbox version of the **Optimizer** will be used as reference [5], thus `Python` written **Procedures** will be linked to the the **Optimizer** by a wrapping function [3].

2. run the **Optimizer** with a given population size ($N_{pop}$) and number of generations ($N_{gen}$) on motor $A$. **Optimizer** will be run $R = 10$ times to assess the performance distribution of the **Procedures**;



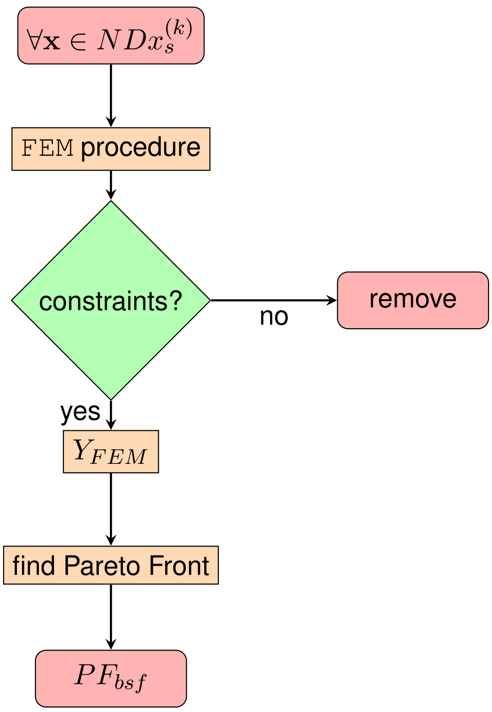

3. collect final generation of non dominated sets computed by **Procedure** and **Optimizer** in $ND_s^{(k)}(r), k = 1 \dots N_{teams}, r = 1 \dots R$;

4. create the set of trial non dominated surrogate $ND_s$ results as:

$$ND_s = \bigcup_{\substack{k=1 \dots N_{teams} \\ r=1 \dots R}} ND_s^{(k)}(r)$$

linked with its input variables $NDx_s$. Each point in degrees of freedom space $\mathbf{x}^{(k)} \in NDx_s$ is tagged with the team model that produced it. Finally the set of the non dominated configurations belonging to the original dataset $ND_{ds}$ are added to the $ND_s$ with their $\mathbf{x}_{ds}$ and objectives $\mathbf{y}_{ds}$ (see Fig. 2.1);

5. for all points $\mathbf{x} \in NDx_s$:
   - run the FEM analysis (procedure in figure 1.1) and create a new set of points in the Pareto space $Y_{FEM}$, designed as *true values*;
   - points whose values by FEM analysis should violate the constraints on $VM$ and $Temp$ will be removed;
   - evaluate which points in $Y_{FEM}$ are non dominated, then assemble the *Pareto Front best so far* $PF_{bsf}$ (see Fig. 2.2);




6. assign to each **Team** a ranking on the basis of the following *metrics*:
   - *coverage*: defined as the ratio of the number of points $N^{(k)}$ in $PF_{bsf}$ that have been produced by the $k$-th team and the total number of points in $PF_{bsf}$:

$$m_1^{(k)} = \frac{N^{(k)}}{N_{bsf}} \tag{2.1}$$

   - Inverse Generational Distance (IGD): the average, on the $R$ runs, of IGD value evaluated considering as reference $PF_{bsf}$ and *true values*

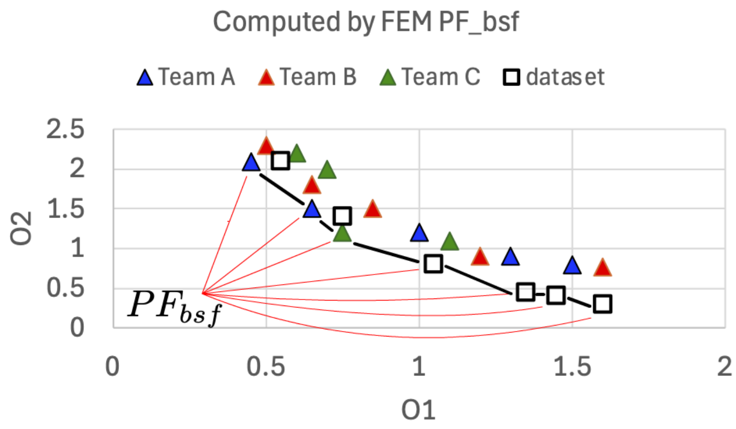


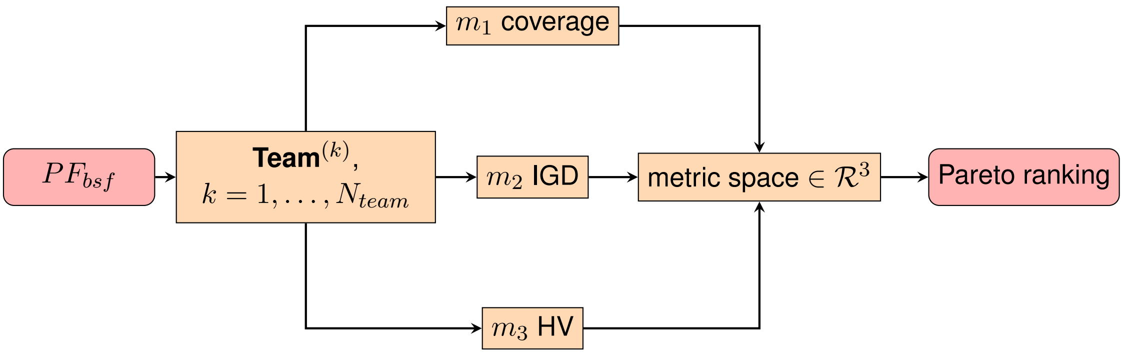

Figure 2.2: $PF_{bsf}$ obtained by FEM (*true point*): *best so far* Pareto Front with 7 points, 2 $\rightarrow$ Team A, 0 $\rightarrow$ Team B, 1 $\rightarrow$ Team C, 4 $\rightarrow$original dataset.

computed by $k$-th team:

$$m_2^{(k)} = \frac{1}{R}\sum_{j=1}^{R} IGD_j^{(k)} \tag{2.2}$$

- HyperVolume (HV): the average, on the $R$ runs, of HV value evaluated considering as reference point the extrema of $PF_{bsf}$:

$$m_3^{(k)} = \frac{1}{R}\sum_{j=1}^{R} HV_j^{(k)} \tag{2.3}$$



# Chapter 3

# Ranking

The ranking is made by a two step procedure whose main aim is to apply Pareto criterion on the metrics space $(m_1, m_2, m_3) \in \mathcal{R}^3$.

The script used to perform the ranking is `GalFer_team_ranking_v2.py` and snippets of code are reported in the following.

Pareto ranking is performed by using the `pareto_plots` functions [6]. Metrics calculation are performed by using `pymoo` functions [4].

## 3.1 Ranking procedure

Data computed by the **Optimizer** linked with teams **Procedures** are read from file and processed. For sake of simplicity, in the following case, all points are considered coming from a $R = 1$ runs from $kTeam$. The extension to the case of multi-run is straightforward by applying equations (2.1), (2.2), (2.3).

1. all data points produced by **Teams** are collected in the same set ($PFteam$) and ranked to find out the `Pareto Front best so far`.

```
#
# out of all results obtained by the teams create the Pareto
    Front best so far
#
rank=pareto_plots.rank_pareto(PFteam, minimise=[True, True])
NDset=rank==1
PFbsf=PFteam[NDset]
NPFbsf=len(PFbsf)
PFbsf_team=labelTeam[NDset]
```

2. for all **Teams** results metrics are evaluated, by using the $PFbsf$ evaluated at the previous step.

```
iniz=0
for kTeam in range(Nteam):
```

```
    #innovation metric 1
    idx=PFbsf_team==kTeam
    nn=np.sum(idx)
    metric[kTeam,0]=nn/NPFbsf
    #IGD metric 2
    idx=labelTeam==kTeam
    res = PFteam[idx]
    ind = IGD(PFbsf)
    print("IGD", ind(res))
    metric[kTeam,1]=ind(res)
    #HV metric 3
    ind = HV(ref_point=ref_point)
    print("HV", ind(res))
    metric[kTeam,2]=ind(res)
```

3. all teams metric values are Pareto ranked and sorted in *fronts*;

```
rankm=pareto_plots.rank_pareto(metric, minimise=[False, True,
     False])
```

4. all **Teams** are processed starting from the first Pareto Front until all fronts are evaluated and teams ranked.

   For each $n_f$ front, a partial position $P$, considering only one metric at a time, is performed $P_m^{(k)}, m = 1, 2, 3; k \in T^{(n_f)}$. Then the arithmetic average of the ranking (score $s$) of $k$-th team with respect to each metric and for points belonging to the same front is performed:

$$s^{(k)} = \frac{1}{3}\sum_{j=1}^{3} P_j^{(k)}$$

```
for nf in range(nfront+1):
    # select teams belonging to the nf-th front
    idx=rankm==nf
    indexTeam1st=np.where(idx)[0]
    #metric ranking m1-> max, m2->min, m3->max
    #m1
    a=-metric[idx,0]
    ar=rankdata(a, method='min')
    metric_rank[idx,0]=ar
    #m2
    a=metric[idx,1]
    ar=rankdata(a, method='min')
    metric_rank[idx,1]=ar
     #m3
     a=-metric[idx,2]
     ar=rankdata(a, method='min')
     metric_rank[idx,2]=ar
    # average ranking
    rank_average=np.average(metric_rank[idx], axis=1)
```

```
    print('Teams in front',nf,(tagTeam[idx]))
    print('Teams average ranking',rank_average)
```

5. all the $n$ **Teams** belonging to $T^{(n_f)}$ are thus ranked from $1$ to $n$ positions. Tie positions are handled so that **Teams** having the same score are put in the same ranking order.

```
    npos=1
    ar=rankdata(rank_average, method='min')
    indTeam=np.sort(rank_average)
    counter=0
    while counter < len(ar):
        for kk in range(len(ar)):
            if(ar[kk]==npos): #tie handling
                rank_final[counter+nstart,0]=npos+nstart
                rank_final[counter+nstart,1]=indexTeam1st[kk]+1
                rank_final[counter+nstart,2]=nf
                rank_final[kk+nstart,3]=indTeam[kk]
                counter=counter+1
        npos=npos+1
    nstart=nstart+len(ar)
    print('nstart',nstart)
```

6. The **team** (or **teams** in case of a tie) getting the highest score is (are) the winner(s).

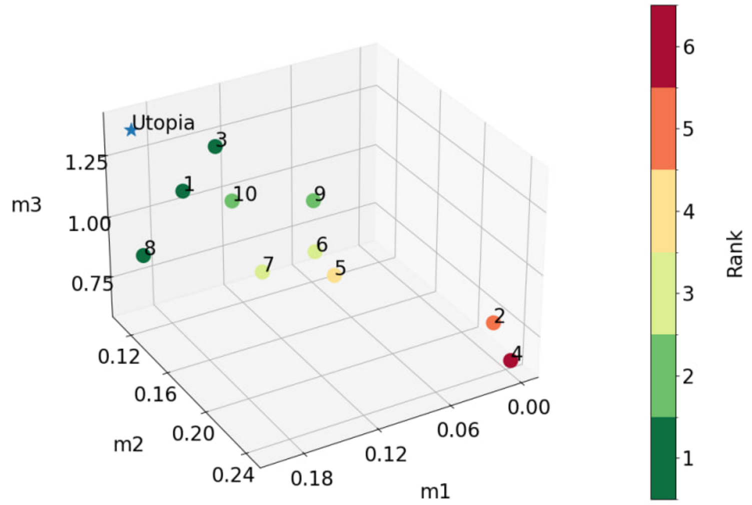



Figure 3.1: Plot of the metrics obtained by ten teams with team number as label. Each team belongs to a Pareto front that is highlighted by the symbol color.

## 3.2 Numerical example

A set of 10 teams is simulated and, for sake of simplicity, this example is limited to $R = 1$ runs.

The metric values for $IGD$ are computed by using the $PF_{bsf}$ and comparing it with the final Pareto Front produced by each team.

The computed metric values are reported in table 3.1.

The *Utopia* point is defined as the maximum value of $m_1$, minimum of $m_2$ and maximum of $m_3$ and is equal to $(0.2000, 0.1092, 1.3697)$.

Following the computation of metrics, the Pareto ranking in the metric space

Table 3.1: Metric values for each team

| team | $m_1$ | $m_2$ | $m_3$ |
|---|---|---|---|
| 1 | 0.1600 | 0.1092 | 1.0687 |
| 2 | 0.0000 | 0.2279 | 0.7309 |
| 3 | 0.1600 | 0.1451 | 1.3697 |
| 4 | 0.0000 | 0.2439 | 0.6365 |
| 5 | 0.0800 | 0.1660 | 0.8119 |
| 6 | 0.0800 | 0.1468 | 0.8404 |
| 7 | 0.1200 | 0.1410 | 0.7911 |
| 8 | 0.2000 | 0.1175 | 0.8876 |
| 9 | 0.0800 | 0.1453 | 1.0467 |
| 10 | 0.1200 | 0.1101 | 0.9770 |

Table 3.2: Metric position $P$ for teams belonging to the first front

| team | $P_1$ | $P_2$ | $P_3$ | rank average $s$ |
|---|---|---|---|---|
| 1 | 2 | 1 | 2 | 1.6667 |
| 3 | 2 | 3 | 1 | 2.0000 |
| 8 | 1 | 2 | 3 | 2.0000 |

Table 3.3: Final ranking

| ranking | team | front | rank average $s$ |
|---|---|---|---|
| 1 | 1 | 1 | 1.6667 |
| 2 | 3 | 1 | 2.0000 |
| 2 | 8 | 1 | 2.0000 |
| 4 | 10 | 2 | 1.3333 |
| 5 | 9 | 2 | 1.6667 |
| 6 | 7 | 3 | 1.3333 |
| 7 | 6 | 3 | 1.6667 |
| 8 | 5 | 4 | 1.0000 |
| 9 | 2 | 5 | 1.0000 |
| 10 | 4 | 6 | 1.0000 |

is performed. The teams belonging to the first Pareto Front ($front = 1$) are: $(1, 3, 8)$. Their positioning $P$ in each metric is reported in table 3.2 together with the average score $s$.

The teams average ranking are $(1.6667, 2.0000, 2.0000)$, the lower the better. Team $1$ is thus the winner, where $3$ and $8$ are equal merit in the second place.

The **Teams** points in the metric space are reported in figure 3.1. Their *metric rankings* are in table 3.3 where the proposed procedure is iterated for all fronts in the metric space.

# Chapter 4

# Rules for the *extrapolation* phase

The extrapolation capabilities of the **procedures** will be assessed by training them of two *families* of electric motors and guessing the performance of a third motor. All motor families have the same *IPM V-type* structure.

The values chosen for the three motor families are reported in Table 4.1.

Complete datasets for motor families $A$ and $B$ $(4096$ points) will be provided, limited dimension dataset ($256$ points) will be provided for motor $C$.

The geometrical and structure data peculiar of each motor family will be added to the $8$-th dimensional dataset provided for motor $A$ in the *interpolation* phase. A complete list of input parameters and ranges are reported in table 4.2.

## 4.1 what each Team will provide

Each **Team** will:

- elaborate a data-driven surrogate model in the $12$-dimensional space of degrees of freedom (input) for the $7$ outputs of the datasets that can surrogate all motor families;
- as in the *interpolation* step, each **Team** will send to the GalFer organizers one surrogate data-driven model that can work for both motor models $A$, $B$ and $C$. If different procedures have been devised for some of the outputs,

Table 4.1: External values and rated reference performance.

| Family | | A | B | C |
|---|---|---|---|---|
| *target* | | Full electric | Full hybrid | Mild hybrid |
| Rated torque | [Nm] | 236 | 87 | 165 |
| Rated power | [kW] | 120 | 36.9 | 42 |
| Max. speed | [rpm] | 15000 | 13500 | 12000 |

Table 4.2: Variables sweep lower and upper ranges for motors $A$, $B$ and $C$.

| | Parameter | name | $A$ | $B$ | $C$ |
|---|---|---|---|---|---|
| 1 | Barrier position (pu) | $d\alpha$ | [0.65 0.85] | [0.4 0.85] | - |
| 2 | Barrier width (pu) | $h_{\mathrm{c}}$ | [0.3 0.7] | [0.3 0.7] | - |
| 3 | Rotor radius (mm) | $r$ | [60 78] | [72 90] | - |
| 4 | Tooth width (mm) | $w_{\mathrm{t}}$ | [3.8 6.3] | [5 9] | - |
| 5 | Tooth length (mm) | $l_{\mathrm{t}}$ | [15 22.5] | [12.36] | - |
| 6 | Slot opening (pu) | $w_{\mathrm{o}}$ | [0.1 0.4] | [0.1 0.4] | - |
| 7 | Barrier shift (mm) | $dxIB$ | [-4 6] | [-4 8] | - |
| 8 | Current phase angle (°) | $\gamma$ | [30 60] | [35 75] | - |
| 9 | Stator outer diameter (mm) | $R_s$ | 225 | 264 | - |
| 10 | Stack length (mm) | $h_c$ | 134 | 50 | - |
| 11 | Number of pole pairs | $n_p$ | 3 | 4 | - |
| 12 | Number of slots/pole/phase | $n_{spp}$ | 3 | 2 | - |

they will have to be *bundled up* in one single function.

## 4.2 what GalFer Organizers will do

The **Procedures** will be evaluated on the performance of motor $C$[1], following the same ranking procedure detailed in the *interpolation* phase.

[1] Detailed data for motor $C$ will be released later accordingly to the timeline publicly available on the [6] page.

# Chapter 5

# Rules for the *novelty* phase

The **Reports** produced by the **Teams** will be evaluated by the *Advisory Board* mainly on the basis of the following criteria:

- novelty of the data-driven proposed approach or of the adoption of existing methodologies to the electric motor design application.
- ability of the approach to provide useful design insights going beyond traditional techniques.
- efficient use of the training data provided and of the devised surrogate model.

# Chapter 6

# Awards

The three categories for awards defined in the previous chapters must share a budget available to the Contest by contributes in cash by different sponsors. At the time of today (December 11, 2024), a reasonable budget is around $9000$ €.
The budget is subdivided in shares for each categories (see figure 6.1):

- $35$ % is devoted to the *extrapolation* awards
- $25$ % is devoted to the *interpolation* awards
- $30$ % is devoted to the *novelty* awards

In turn, each of the allotment for category has to be divided in amount for first, second and third prize. Subdividing the allotment as: second prize equal to half of the first one and third prize half of the second, an hypothesis of prizes is reported in table
Money prizes are reserved only for academic teams.
One team can be recipient of only one prize, in case of a position leading to multiple prizes it will take the highest among the possible ones.

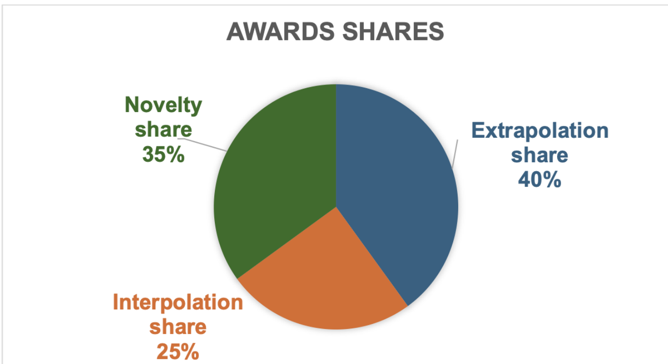



Figure 6.1: Subdivision of awards amount among categories.

Table 6.1: Prizes in money for the three categories under the hypothesis of a prize budget of 9000 €.

| | Extrapolation | Interpolation | Novelty |
|---|---|---|---|
| total for cat. | 3600 | 2250 | 3150 |
| 1st | 2057 | 1286 | 1800 |
| 2nd | 1029 | 643 | 900 |
| 3rd | 514 | 321 | 450 |

# Bibliography


[1] Manuscript Templates for Conference Proceedings https://www.ieee.org/conferences/publishing/templates.html, (2024), Online; accessed 25-Nov-2024.

[2] SyR-e Synchronous Reluctance – evolution design, https://github.com/SyR-e/syre_public, (2024), Online; accessed 25-Nov-2024.

[3] Directly Call Python Functionality from MATLAB, https://ch.mathworks.com/help/matlab/matlab_external/ways-to-call-python-from-matlab.html, (2024), Online; accessed 10-Oct-2024.

[4] Multi-objective Optimization in Python, https://pymoo.org/algorithms/index.html, (2024), Online; accessed 31-Oct-2024.

[5] Evolutionary multi-objective optimization platform https://ch.mathworks.com/matlabcentral/fileexchange/105260-platemo, (2024), Online; accessed 08-Nov-2024.

[6] Computer Aided Design of ElectroMagnetic Apparatus, https://github.com/cadema-PoliTO, (2024), Online; accessed 31-Oct-2024.

[7] Amin Ibrahim, Azam Asilian Bidgoli, Shahryar Rahnamayan, Kalyanmoy Deb, "A Novel Pareto-optimal Ranking Method for Comparing Multi-objective Optimization Algorithms" https://arxiv.org/abs/2411.17999