MathWorks presents

Electromagnetic design with SyR-e

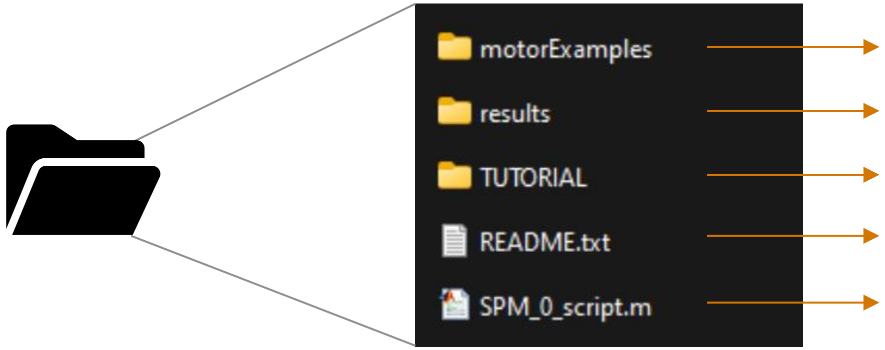
Jacopo Ferretti

MATLAB Student Ambassador

What is needed for this tutorial?

1. Matlab (of course!)
2. Syre
3. This Tutorial

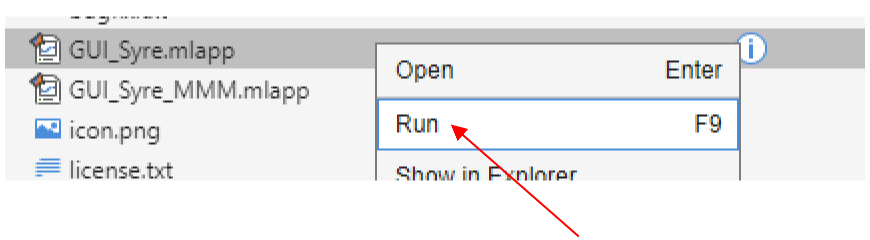
Follow these slides to build the motor. If you encounter any problems, you can use the attached files in this subfolder.



- Motor example, syre results and Simulink model
- Example optimization results
- This presentation
- General info
- Script for loading parameters into the GUI.

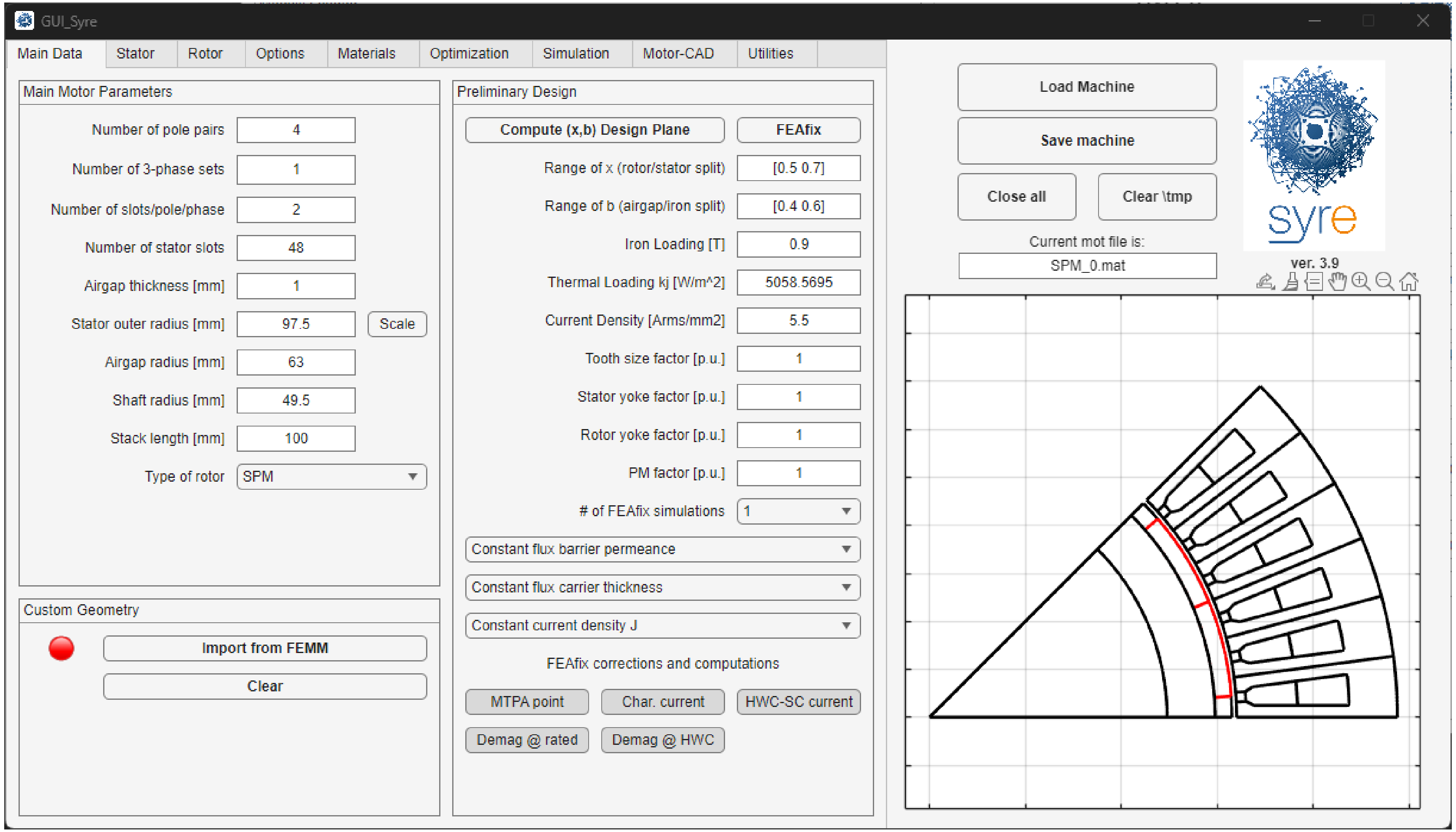
Next steps

- Download SyR-e
  https://it.mathworks.com/matlabcentral/fileexchange/158216-syr-e
- Unzip it in your favourite loaction
- Open Matlab
- Open the unzipped syre path on matlab
- Right click "GUI_Syre.mlapp"
  - Run



GUI_Syre – Welcome

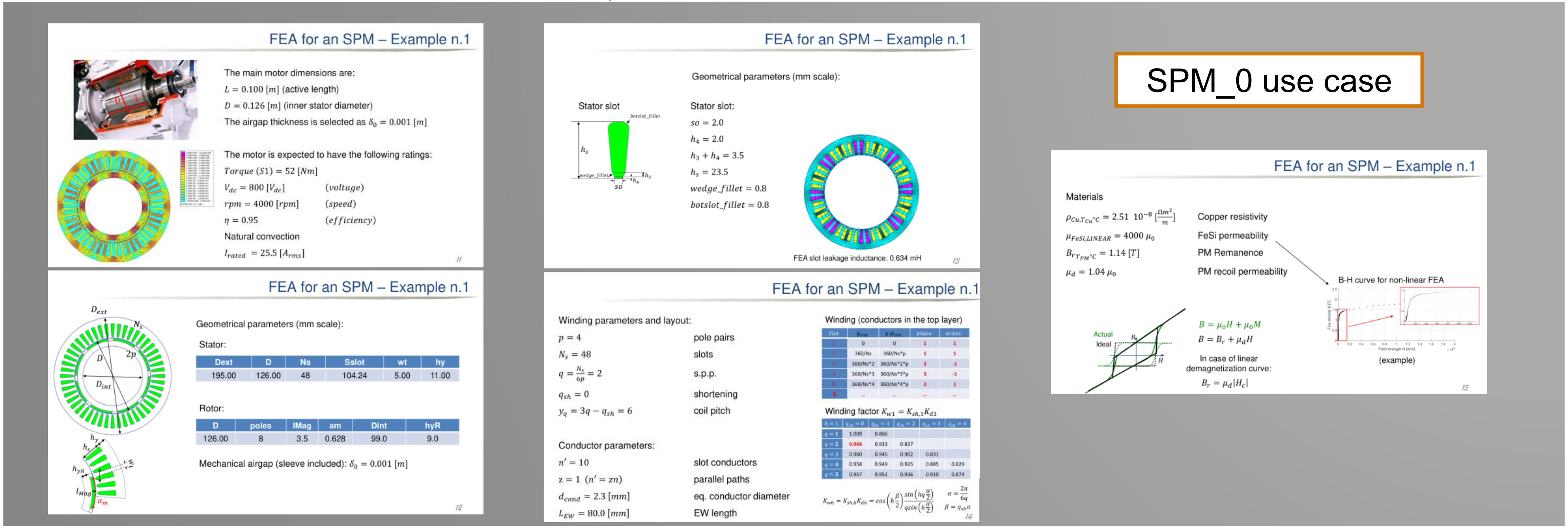
- This is the first tool! Here, you can insert the motor's characteristics, geometry, loads and materials, perform simulations and evaluate flux maps and losses.



GUI_Syre – setting up parameters

- **The second step is to load the parameters of your motor in GUI_syre.**
  - We will use the SPM motor use case presented below. This requires us to convert its values into Syre-compatible parameters using 'SPM_0_script.mat'.

How to do it?
Next slide ->



SPM_0 use case

FEA for an SPM – Example n.1

The main motor dimensions are:

$L = 0.100\ [m]$ (active length)

$D = 0.126\ [m]$ (inner stator diameter)

The airgap thickness is selected as $\delta_0 = 0.001\ [m]$

The motor is expected to have the following ratings:

$Torque\ (S1) = 52\ [Nm]$

$V_{dc} = 800\ [V_{dc}]$ $(voltage)$

$rpm = 4000\ [rpm]$ $(speed)$

$\eta = 0.95$ $(efficiency)$

Natural convection

$I_{rated} = 25.5\ [A_{rms}]$

11

FEA for an SPM – Example n.1

Geometrical parameters (mm scale):

Stator:

Dext	D	Ns	Sslot	wt	hy
195.00	126.00	48	104.24	5.00	11.00

Rotor:

D	poles	lMag	am	Dint	hyR
126.00	8	3.5	0.628	99.0	9.0

Mechanical airgap (sleeve included): $\delta_0 = 0.001\ [m]$

12

FEA for an SPM – Example n.1

Geometrical parameters (mm scale):

Stator slot

Stator slot:

$so = 2.0$

$h_4 = 2.0$

$h_3 + h_4 = 3.5$

$h_s = 23.5$

$wedge\_fillet = 0.8$

$botslot\_fillet = 0.8$

FEA slot leakage inductance: 0.634 mH

13

FEA for an SPM – Example n.1

Winding parameters and layout:

$p = 4$	pole pairs
$N_s = 48$	slots
$q = \frac{N_s}{6p} = 2$	s.p.p.
$q_{sh} = 0$	shortening
$y_q = 3q - q_{sh} = 6$	coil pitch

Conductor parameters:

$n' = 10$	slot conductors
$z = 1\ (n' = zn)$	parallel paths
$d_{cond} = 2.3\ [mm]$	eq. conductor diameter
$L_{EW} = 80.0\ [mm]$	EW length

Winding (conductors in the top layer)

Slot	φ_{slot}	$p\ \varphi_{slot}$	phase	orient.
1	0	0	1	1
2	360/Ns	360/Ns*p	1	1
3	360/Ns*2	360/Ns*2*p	3	-1
4	360/Ns*3	360/Ns*3*p	3	-1
5	360/Ns*4	360/Ns*4*p	2	1
6	...	...	...	...

Winding factor $K_{w1} = K_{sh,1}K_{d1}$

$h = 1$	$q_{sh} = 0$	$q_{sh} = 1$	$q_{sh} = 2$	$q_{sh} = 3$	$q_{sh} = 4$
$q = 1$	1.000	0.866			
$q = 2$	0.966	0.933	0.837		
$q = 3$	0.960	0.945	0.902	0.831	
$q = 4$	0.958	0.949	0.925	0.885	0.829
$q = 5$	0.957	0.951	0.936	0.910	0.874

$K_{wh} = K_{sh,h}K_{dh} = cos\left(h\frac{\beta}{2}\right)\frac{sin\left(hq\frac{\alpha}{2}\right)}{q sin\left(h\frac{\alpha}{2}\right)}$ $\alpha = \frac{2\pi}{6q}$ $\beta = q_{sh}\alpha$

14

FEA for an SPM – Example n.1

Materials

$\rho_{Cu,T_{Cu}°C} = 2.51\ 10^{-8}\ [\frac{\Omega m^2}{m}]$	Copper resistivity
$\mu_{FeSi,LINEAR} = 4000\ \mu_0$	FeSi permeability
$B_{r\,T_{PM}°C} = 1.14\ [T]$	PM Remanence
$\mu_d = 1.04\ \mu_0$	PM recoil permeability

B-H curve for non-linear FEA

(example)

$B = \mu_0 H + \mu_0 M$

$B = B_r + \mu_d H$

In case of linear demagnetization curve:

$B_r = \mu_d |H_c|$

15

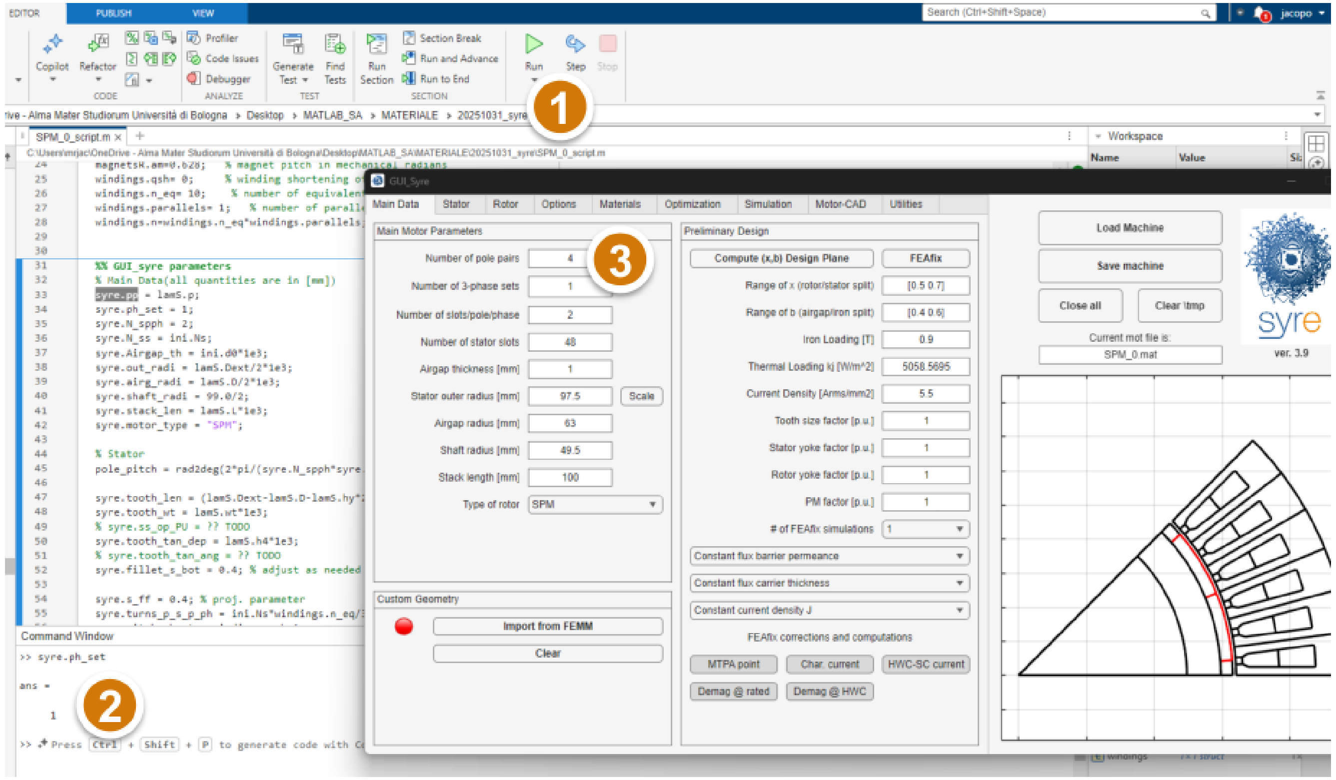
GUI_Syre – setting up parameters

Let's load the parameters in SyR-e

1. Run the script "SPM_0_script.mat"
2. Evaluate each parameter after line `%% GUI_syre parameters`
3. Paste them one by one into the corresponding box in GUI_syre

You need to insert numerical values into the boxes.

You can also directly load the example use-case by pressing 'Load Machine' and then selecting SPM_0.mat



Emotor evaluation – Single point

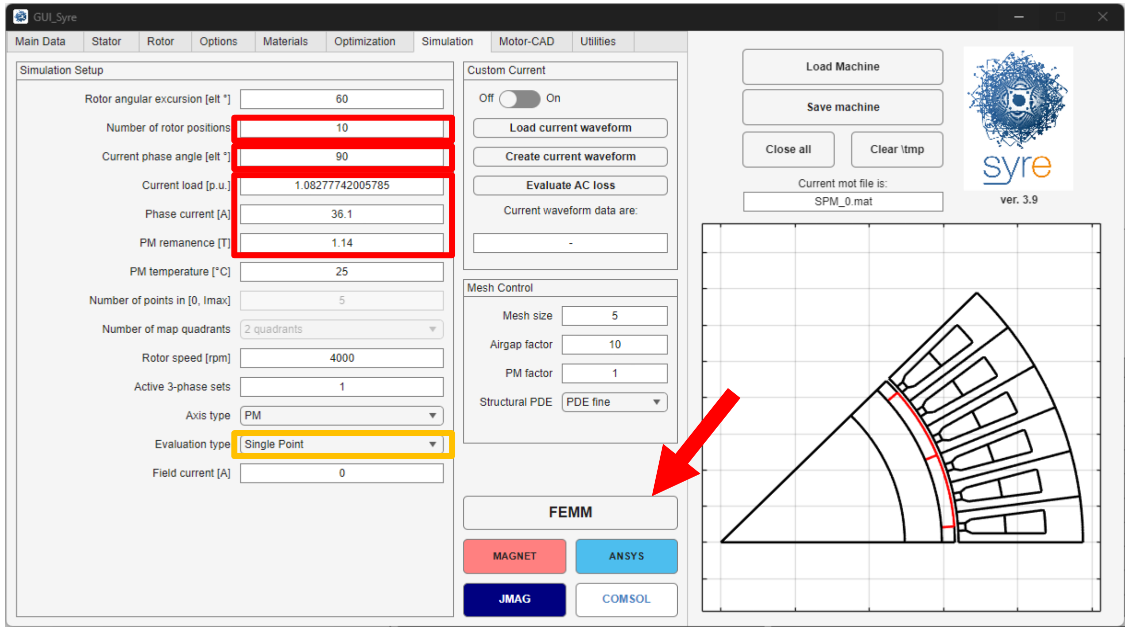
Now that the machine has been loaded, you are ready for the first evaluation.

We are going to conduct a **"single point"** evaluation.

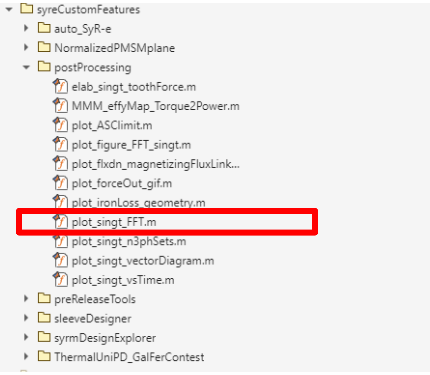
1. Save machine
2. Set number of positions -> 6 very fast! 20 accurate enough. (you choose)
3. Set current phase angle -> 90 (since it is an SPM!)
4. Set load Current (A or p.u.), PM remanence, Temperature
5. Press FEMM!

This evaluation is helpful to validate the motor's geometry in a fast way!

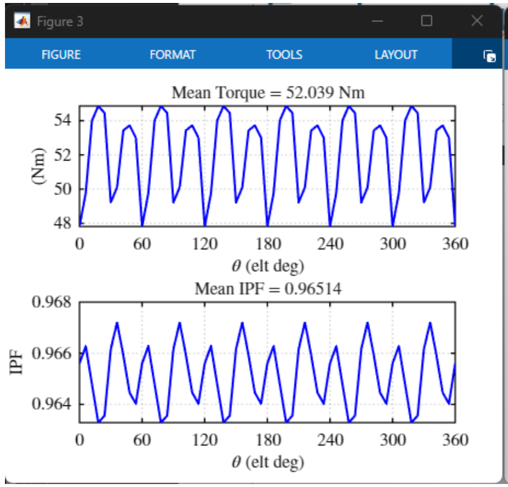
Always save the machine before running any FEMM evaluations! The simulation uses the last saved geometry.

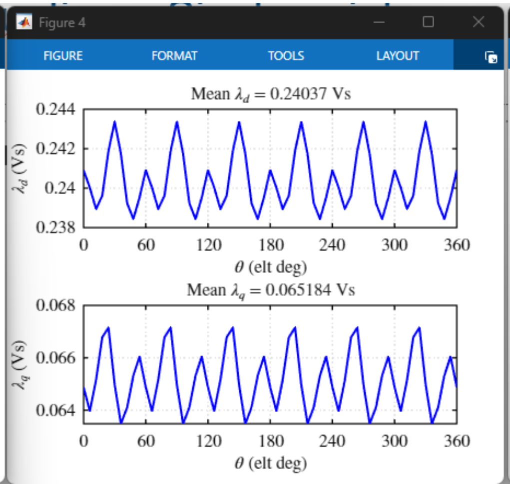


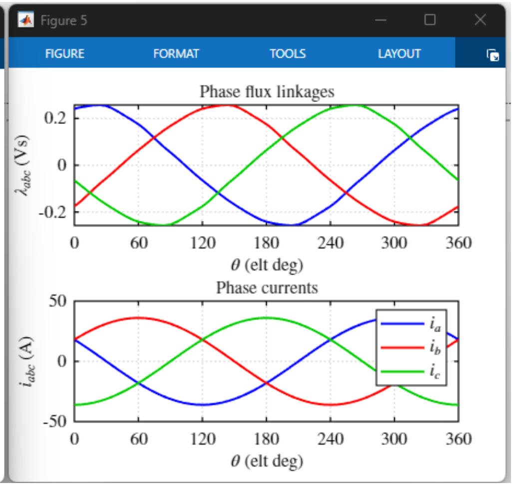
Emotor evaluation – Single point

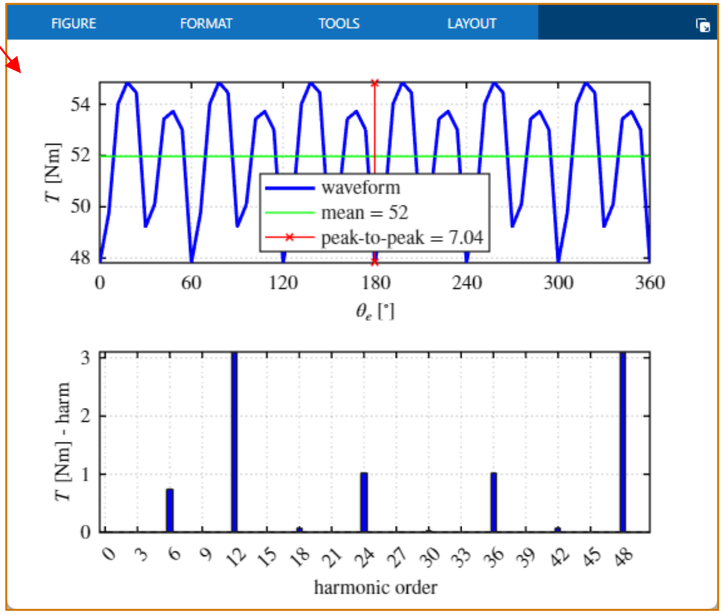


- A lot of figures pop out!
- **You can find more interesting figures using the function "plot_singt_FFT"** (run it in command window)
  - After running it, select the motor's single point result. You will find it in the following folder: MOTORXX_results/FEA results/T_eval_XX_ZZ_Yydeg/MOTORXX_OpPointResults.mat
    - There are other interesting functions to discover in "postprocessing”







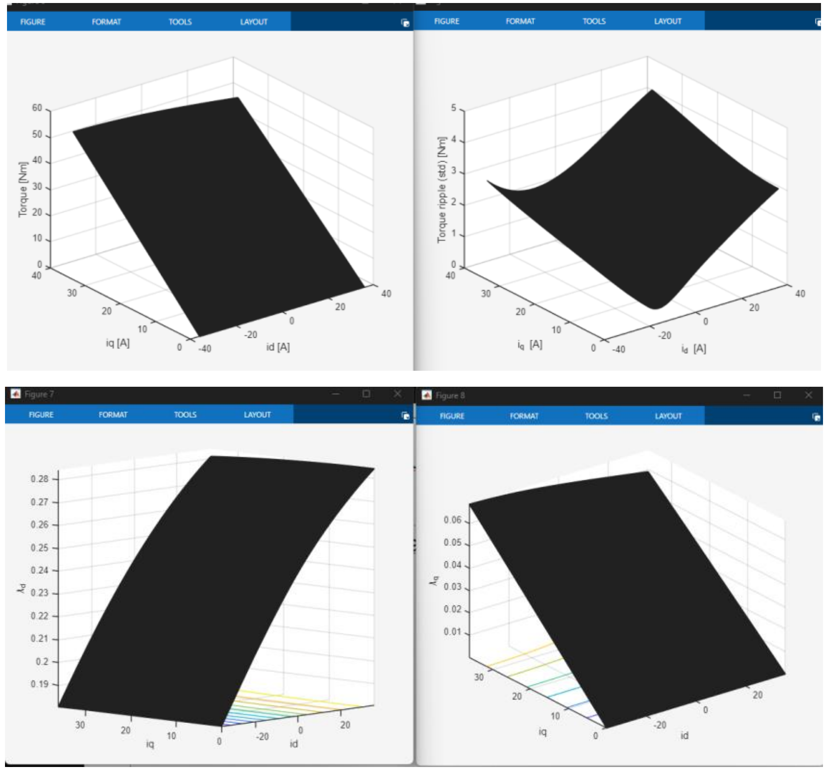


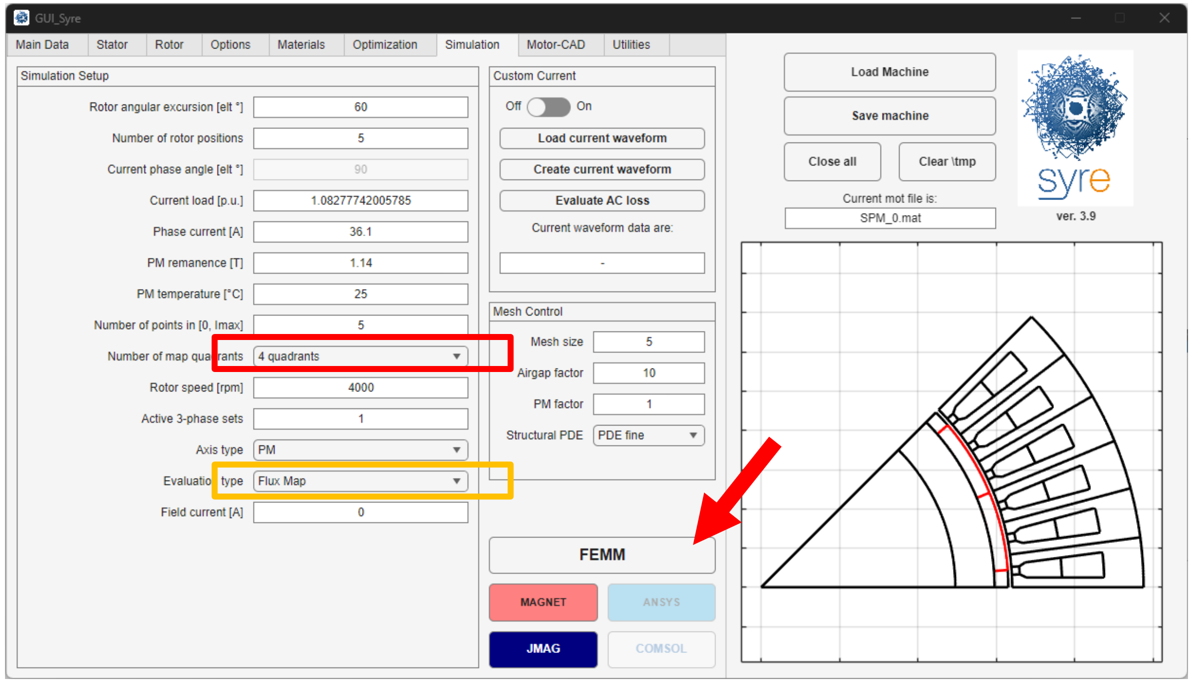
Emotor evaluation – Flux Map

Now evaluate the flux maps

– (Useful for the next passage, GUI_syre_MMM.mlapp and simulink)

- 2 quadrants -> have a rapid look at the maps.
- 4 quadrants -> Better compatibility with Simulink. But it will take longer!



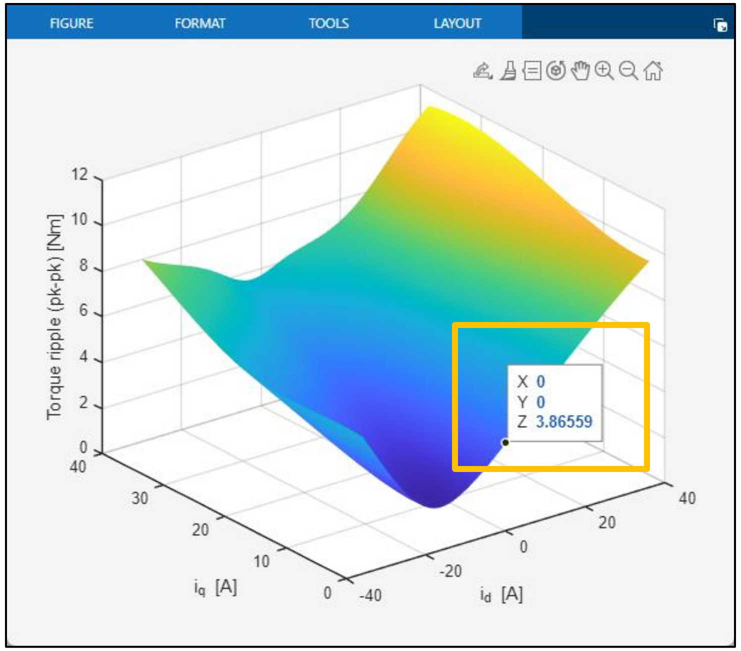


Optimization – problem statement

During the evaluation we can see an interesting output

- High torque ripple! 3.8Nm

dTppsurf.fig



To see the plot with colours, use this script:

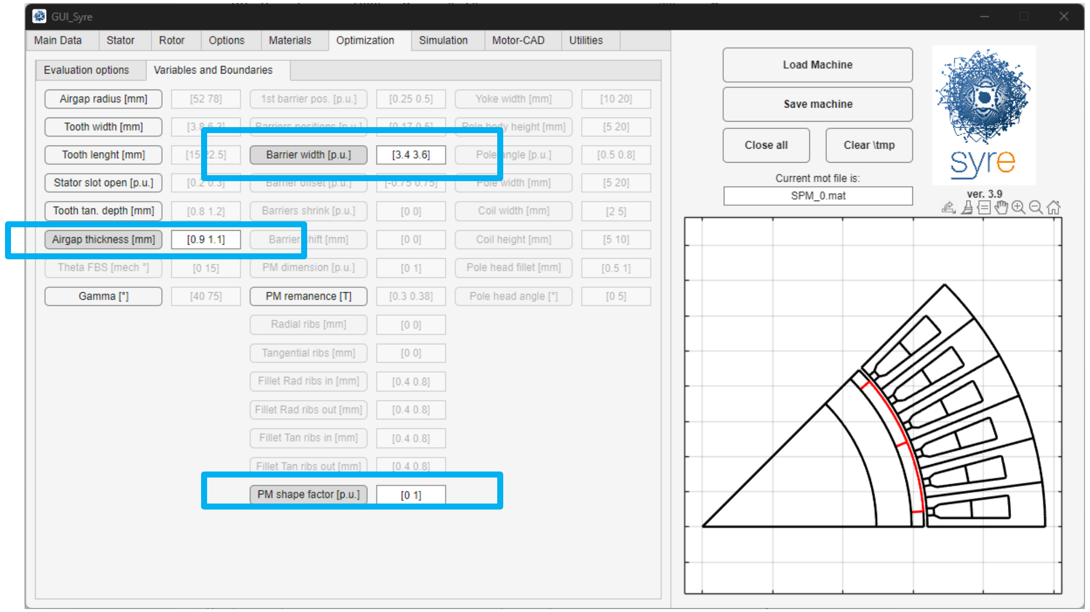
```
surf_objs = findobj(fig, 'Type', 'surface’);
for i = 1:length(surf_objs)
  set(surf_objs(i), 'EdgeColor', 'none’);
end
```

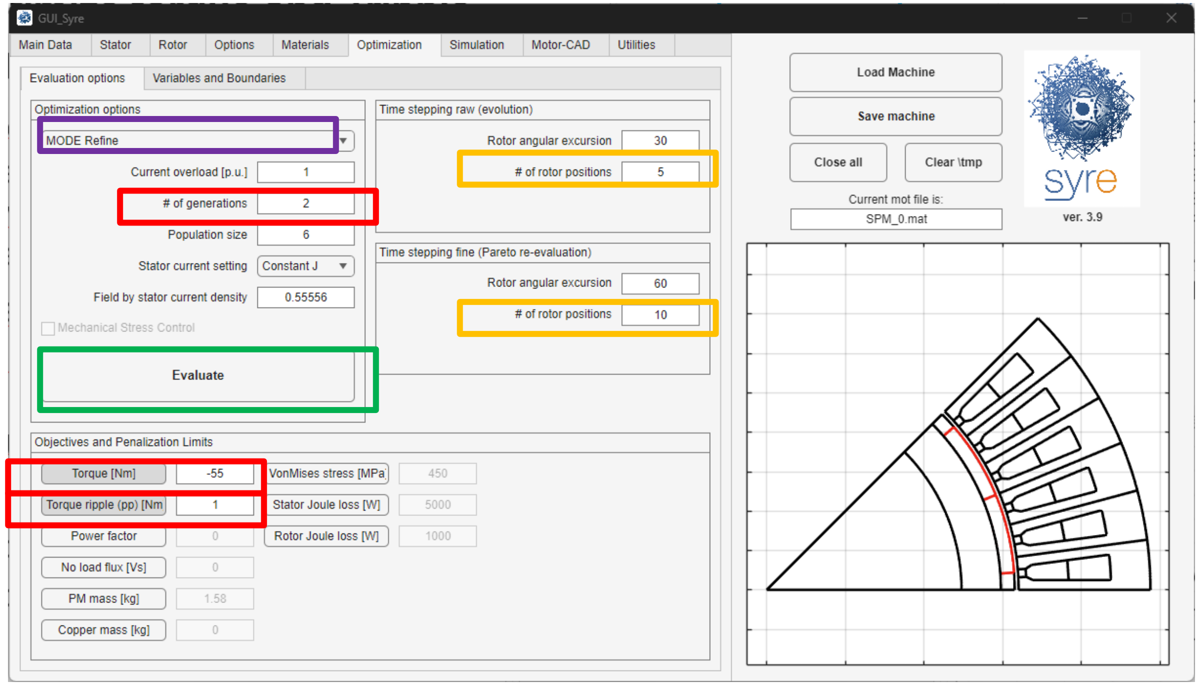
Optimization – setting

Let's optimize for torque and Tripple

- Select MODE Refine
- Set evaluation numbers
  - # generations=2
    (raw) # rorot position = 5
    (fine) # rorot position = 10
- Set variables and boundaries
  - Barrier width [3.4 3.6]
    PM shape factor [0 1]
    airgap thickness [0.9 1.1]
- Set objectives targets (at least 2)
  - Torque -55 Nm
  - Torque ripple 1 Nm
- Press EVALUATE

This is just a FAST EXAMPLE

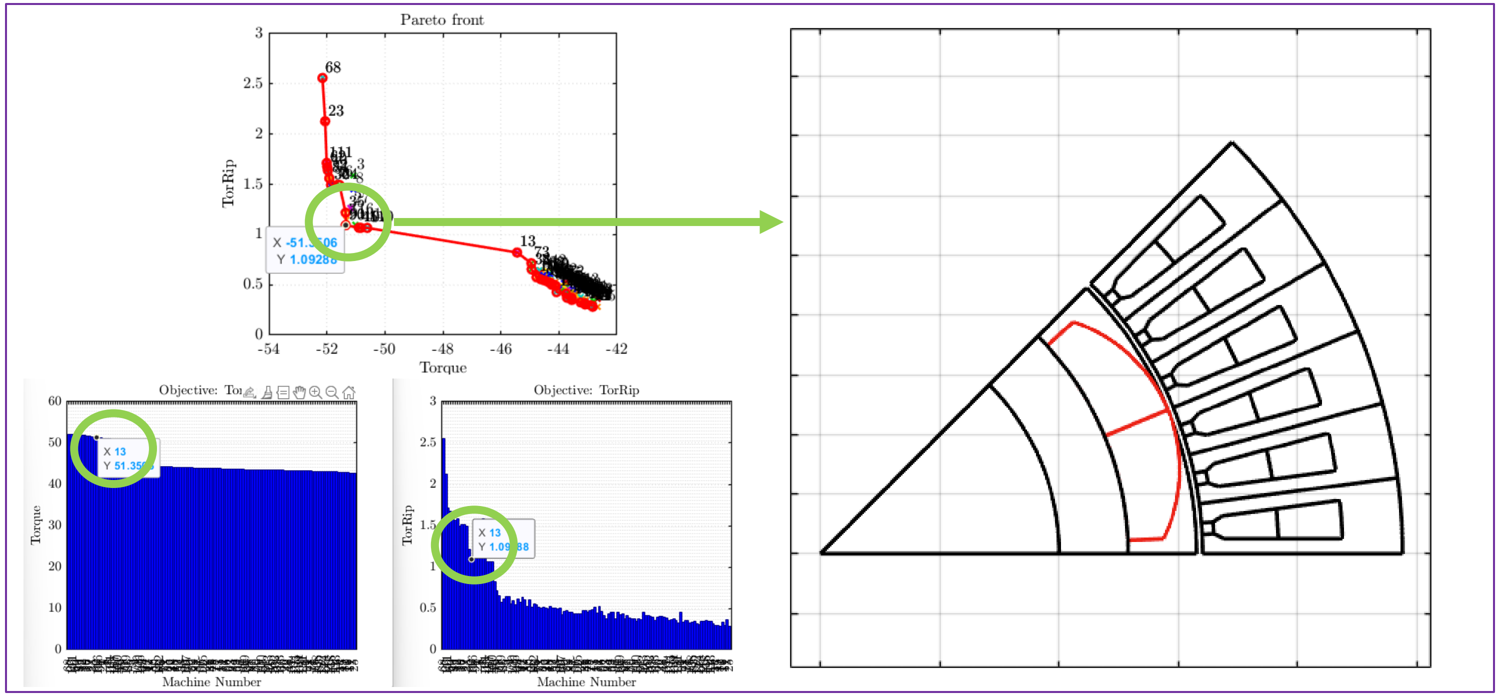




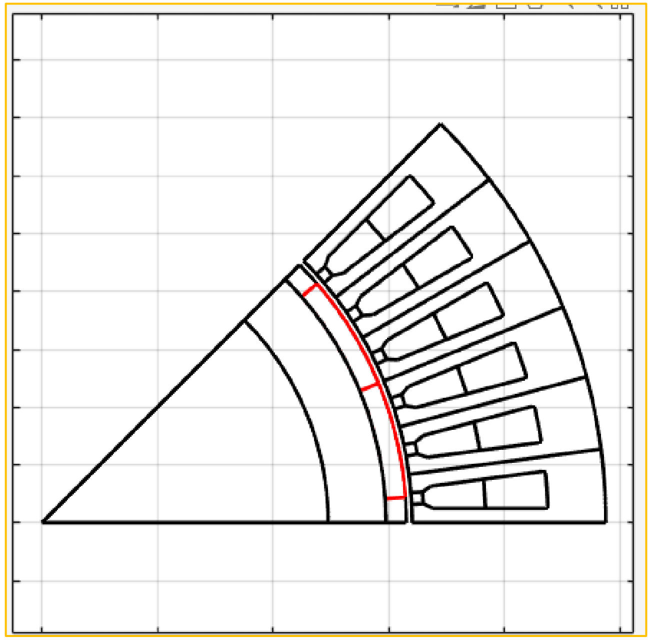
Optimization – results

- **Now you can choose your favourite motor # on the pareto front**
- The motors are saved in *results/OUT_YYYYMMDD/mot_XX.mat*
  - You can load this motor, evaluate its maps etc…

This is just an example.



Optimized #13



Original

fix #1 – syre to SPM

- You might encounter some problems when loading the motor.
  - Follow these steps:

```
Unrecognized field name "AngleSpanOfPM".

Error in back_compatibility (line 1330)
        dataSet.ALPHAdeg = dataSet.AngleSpanOfPM;
                           ^^^^^^^^^^^^^^^^^^^^^
Error in GUI_Syre/LoadMachinePushButtonPushed (line 3736)
                [dataSet,geo,per] = back_compatibility(dataSet,geo,per);
                                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Error in appdesigner.internal.service.AppManagementService/executeCallback (line 13)
        callback(appOrUserComponent, event);
        ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Error in matlab.apps.AppBase>@(source,event)executeCallback(ams,app,callback,requiresEventData,event) (line 54)
            newCallback = @(source, event)executeCallback(ams, ...
                                          ^^^^^^^^^^^^^^^^^^^^^^^^

Explain Error
Error using appdesservices.internal.interfaces.model.AbstractModel/executeUserCallback (line 282)
Error while evaluating Button PrivateButtonPushedFcn.
```

Now it woks

- Go to mfiles/back_compatibility.m
  - Modify from line **1330**

```
  dataSet.ALPHAdeg = dataSet.AngleSpanOfPM;
  dataSet.ALPHApu  = dataSet.AngleSpanOfPM/180;
end
dataSet = rmfield(dataSet,'ThicknessOfPM');
dataSet = rmfield(dataSet,'AngleSpanOfPM');
flagClear = 1;
```

Before

```
  if isfield(dataSet,'AngleSpanOfPM’)
    dataSet.ALPHAdeg = dataSet.AngleSpanOfPM;
    dataSet.ALPHApu  = dataSet.AngleSpanOfPM/180;
    dataSet = rmfield(dataSet,'AngleSpanOfPM’);
  end
end
dataSet = rmfield(dataSet,'ThicknessOfPM');
flagClear = 1;
```

After

Demagnetization analysis

The maximum current before demagnetization can be approximated with:

$$I_{max,pk} = \frac{2p}{N}\left[\left(L_{mag} + \delta_0 \frac{\mu_d}{\mu_0}\right) k_s H_{ci,\vartheta} - \delta_0 \frac{B_r}{\mu_0}\right]$$

- $\boldsymbol{I_{max,pk}}$:Maximum allowable peak phase current [A].
- $\boldsymbol{p}$: pole pairs.
- $\boldsymbol{N}$: active conductors in series per phase.
- $\boldsymbol{L_{mag}}$: Radial thickness of the permanent magnet [m].
- $\boldsymbol{\delta_0}$: Equivalent air-gap length (incluso eventuale fattore di Carter) [m].
- $\boldsymbol{\mu_d}$**(o** $\boldsymbol{\mu_{rec}}$**)**: Recoil permeability of the magnet (typically $\approx 1.05 \cdot \mu_0$ for NdFeB).
- $\boldsymbol{\mu_0}$: Magnetic permeability of vacuum ($4\pi \times 10^{-7}$[H/m])
- $\boldsymbol{k_s}$: Safety margin coefficient (typically 0.8 to 0.9)
- $\boldsymbol{H_{ci,\vartheta}}$:Intrinsic coercive force at the operating temperatureϑ[A/m].
- $\boldsymbol{B_r}$: Remanence (residual induction) of the magnet at the operating temperature [T].

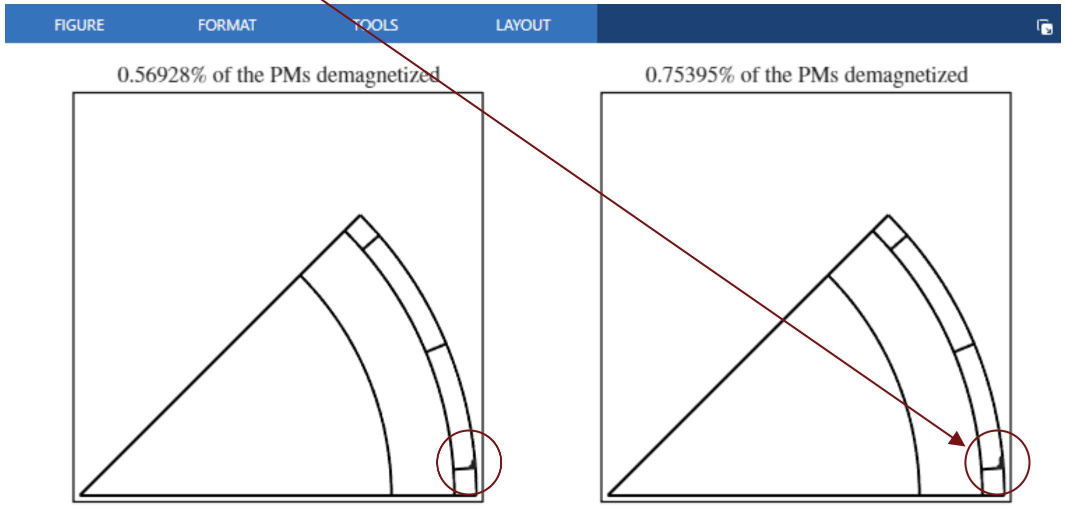
The formula gives the following demagnetisation current for the use case: [151;174] A (@140degC)

Demagnetization analysis – Use case

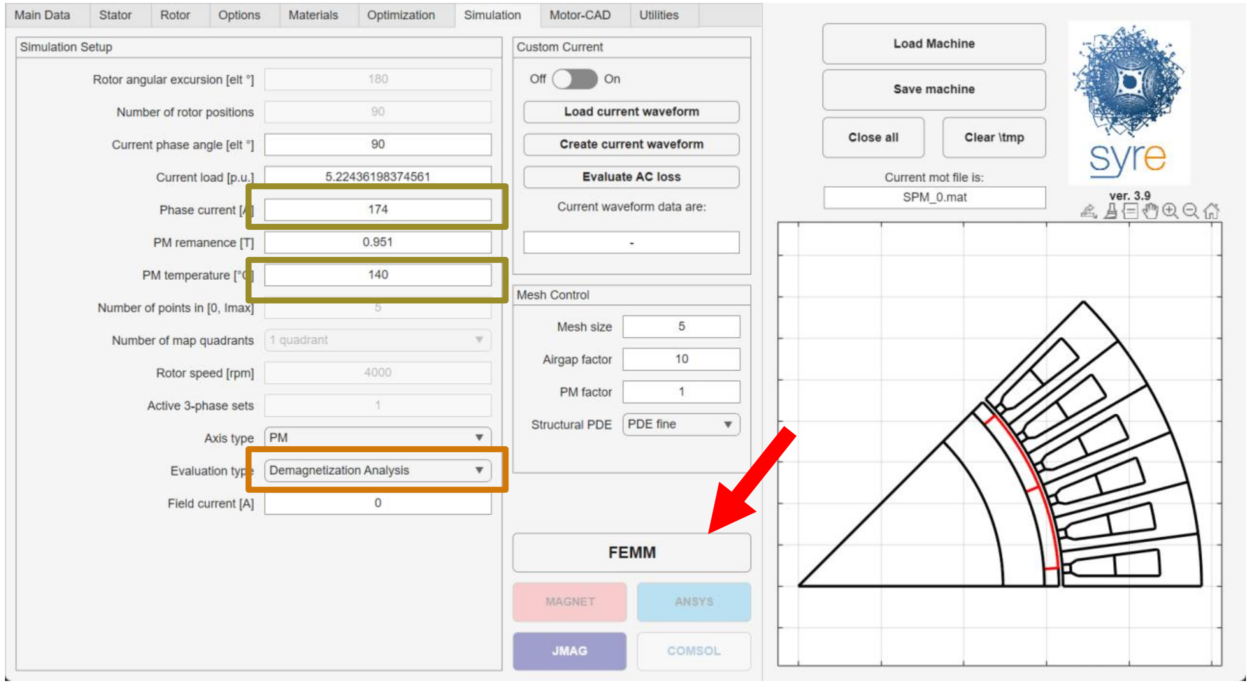
You can evaluate this also on syre:

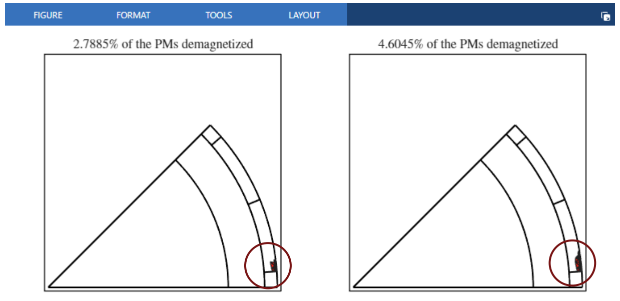
1. Select Demagnetization analysis
2. Set PM temperature, phase current
3. Press femm

Results:

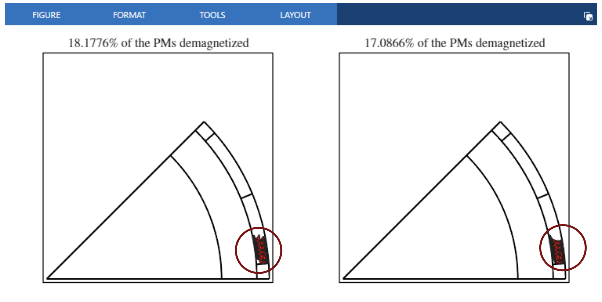


174Apk





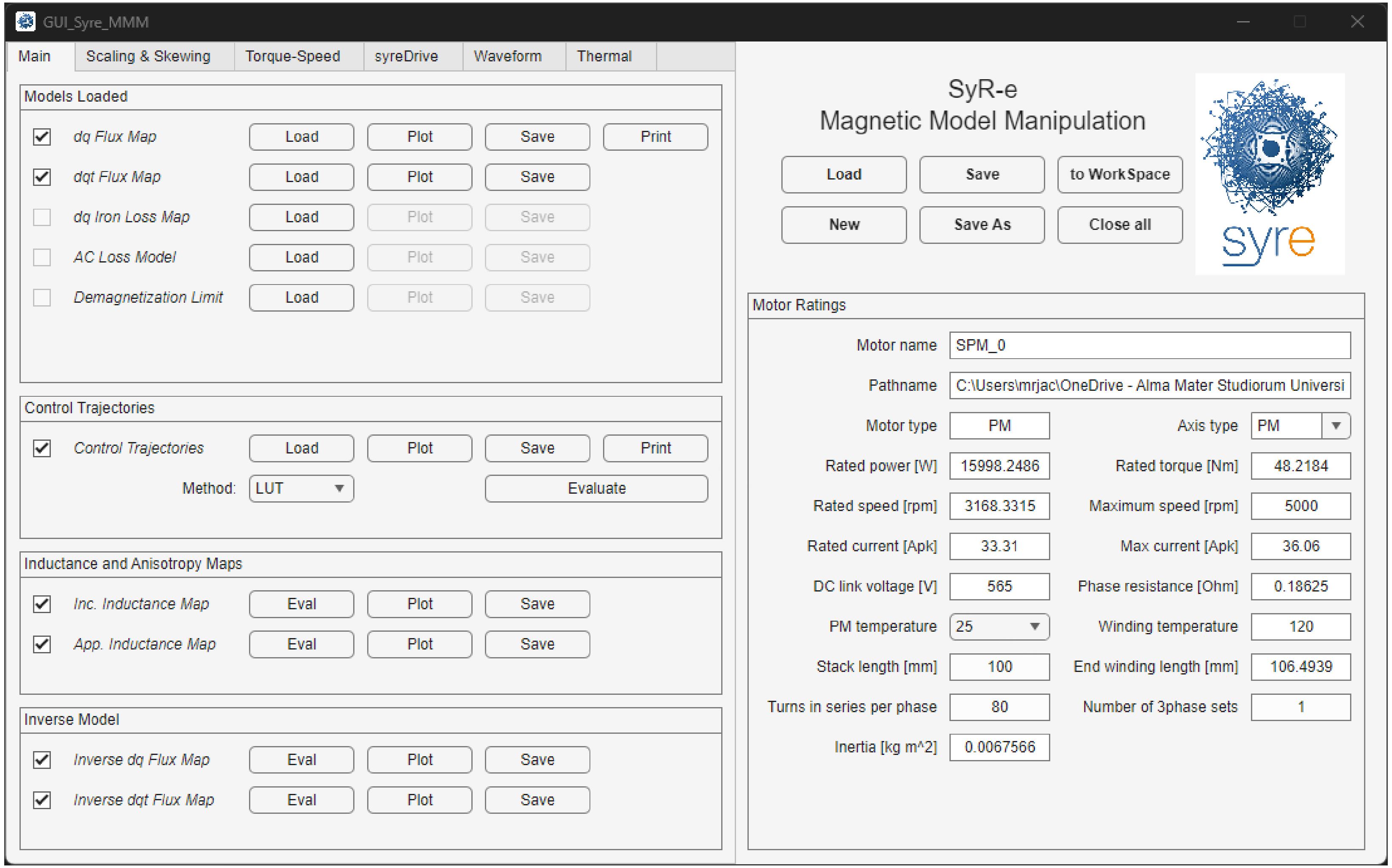
200Apk



250Apk

GUI_Syre_MMM – Welcome

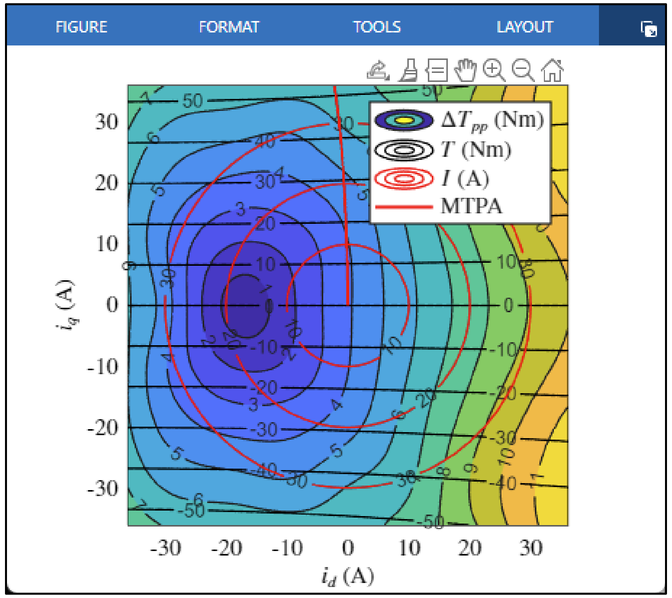
- **Next tool!** Load maps, evaluate control maps, create simulink model



Syre_MMM – Main

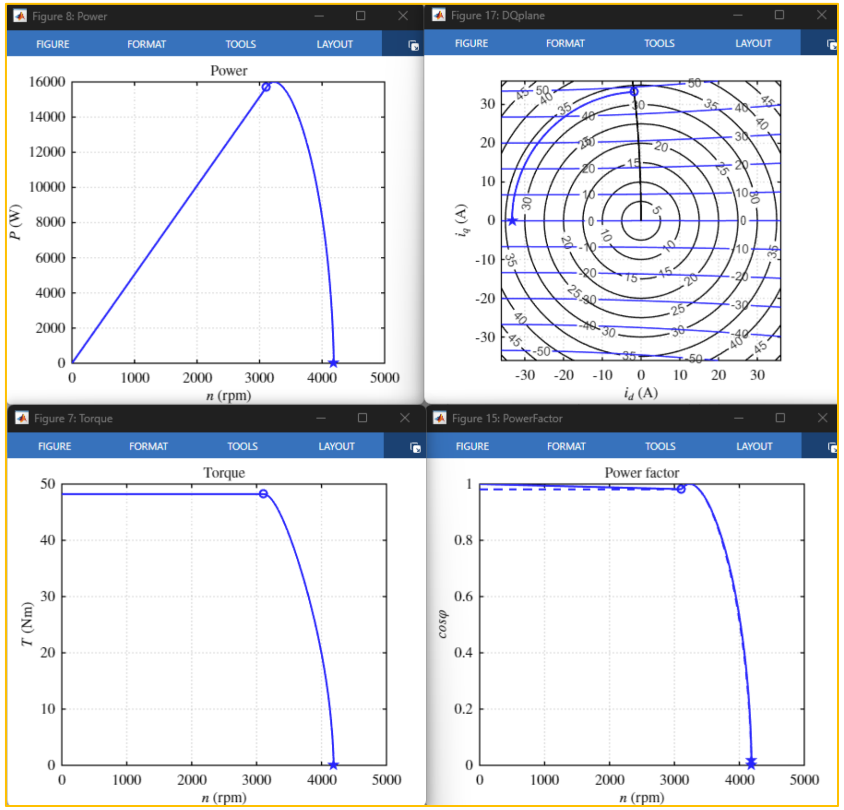
Load everything you have done from syre to syre_MMM

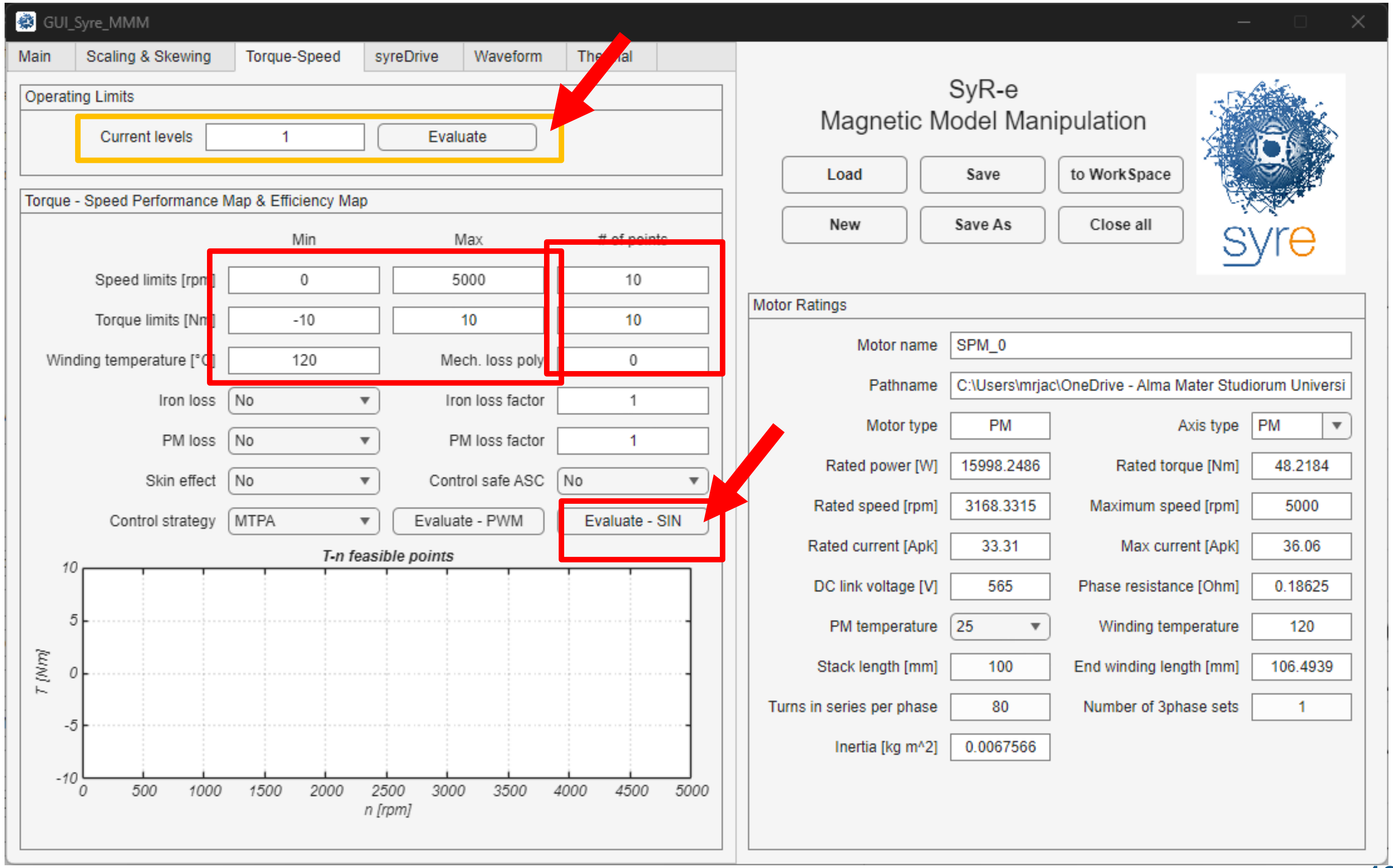
1. Load Motor
2. Load maps
    1. dq flux map
       MOTORXX_results/FEA results/\F_map_YYAYxWWAW_ZZdeg_4Q/fdfq_idiq_nXXX.mat
    2. Control Trajectories
       "Evaluate" (it will take some minutes)
    3. Inc. Inductance Map / App. Inductance Map
       "Evaluate"
    4. Incerse dq Flux Map
       "Evaluate"
3. Save it! (do it for all the maps)



Syre_MMM – Torque-Speed

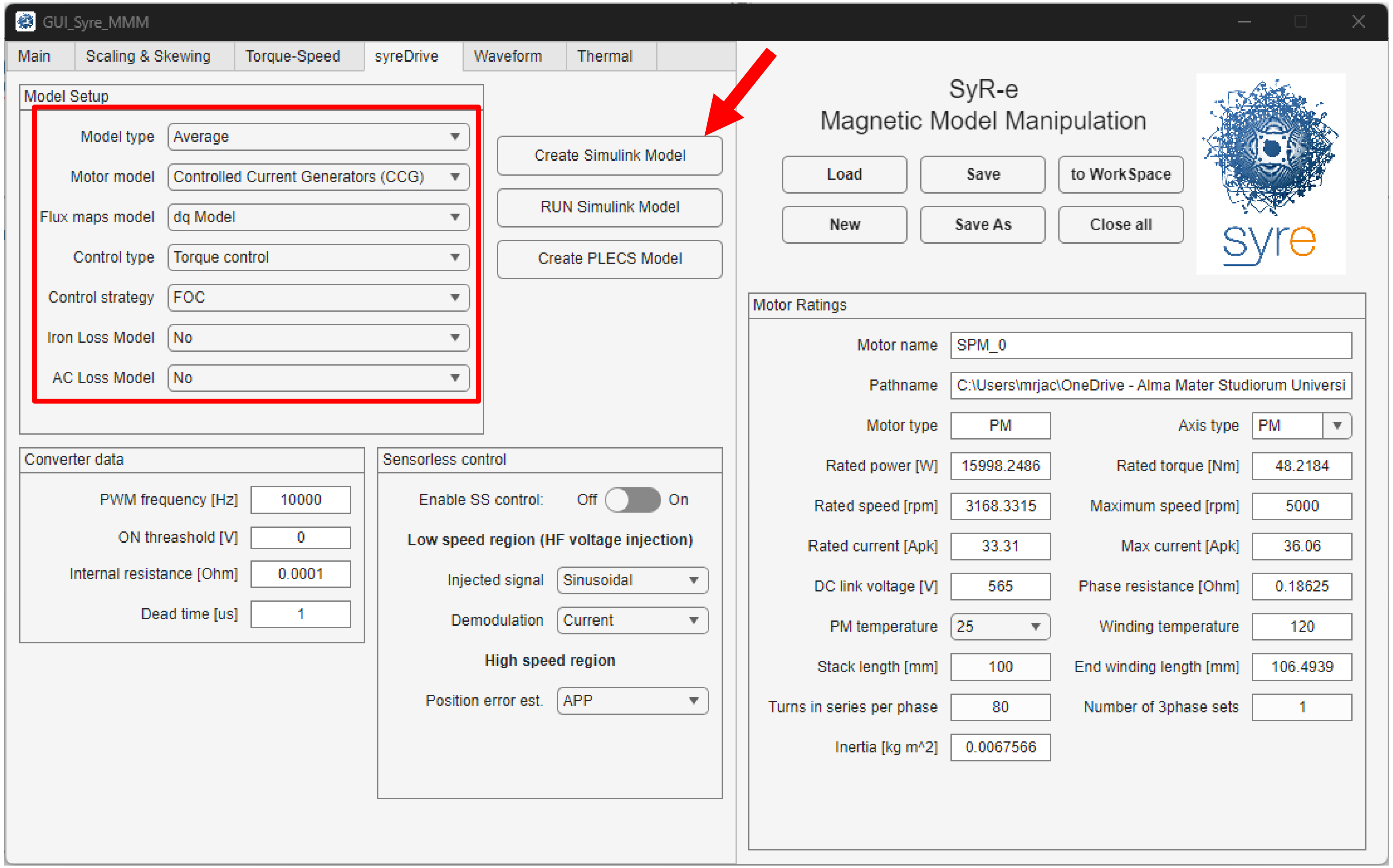
Let's evaluate the torque-speed maps & op. limits





Syre_MMM – syreDrive

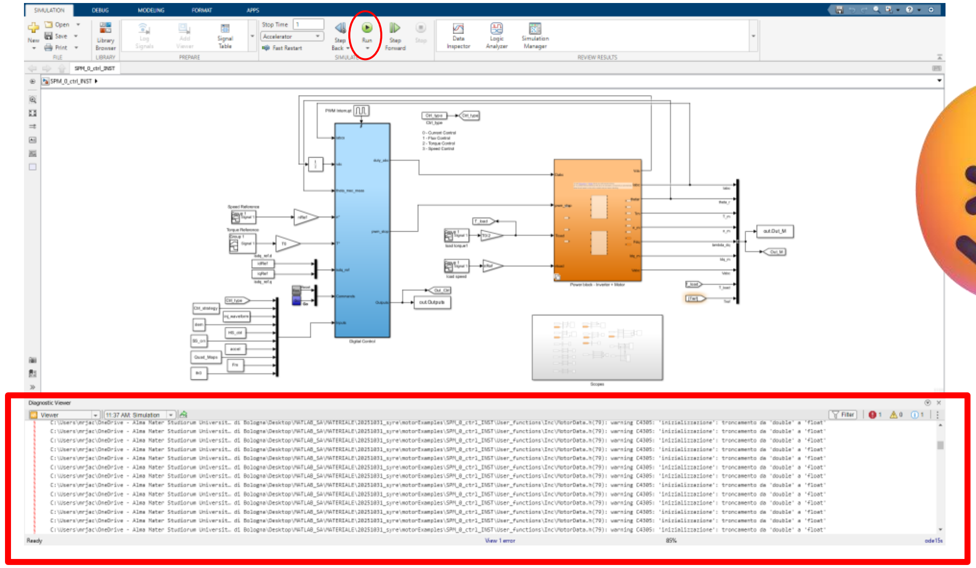
- Create simulink model (CCG or watever you prefer)

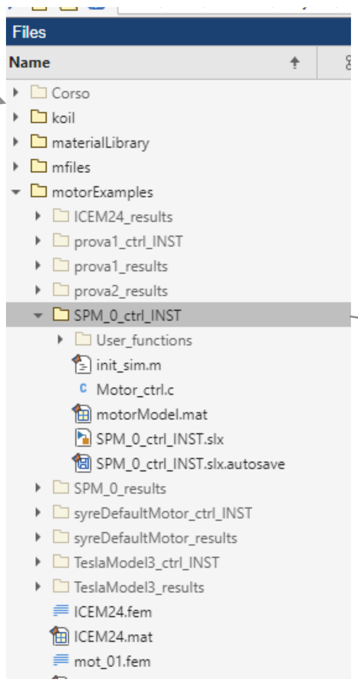


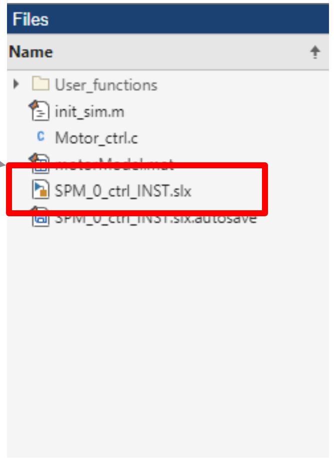
Simulink model - Initialization

Let's play with Simulink!

- Open the simulation path on matlab
  – Double click on the folder containing the model *\motorExamples\MOT_XX_ctrl_INST*
- Open the Simulink model: *MOT_XX_ctrl_INST.slx*
- Run the simulation in simulink
  – ERROR! Let's fix it!

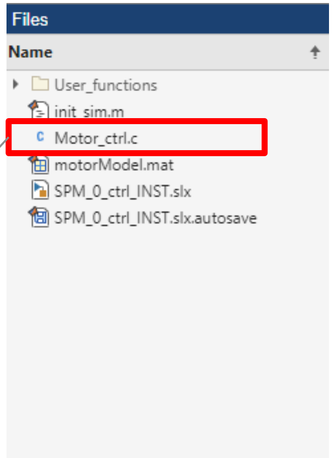


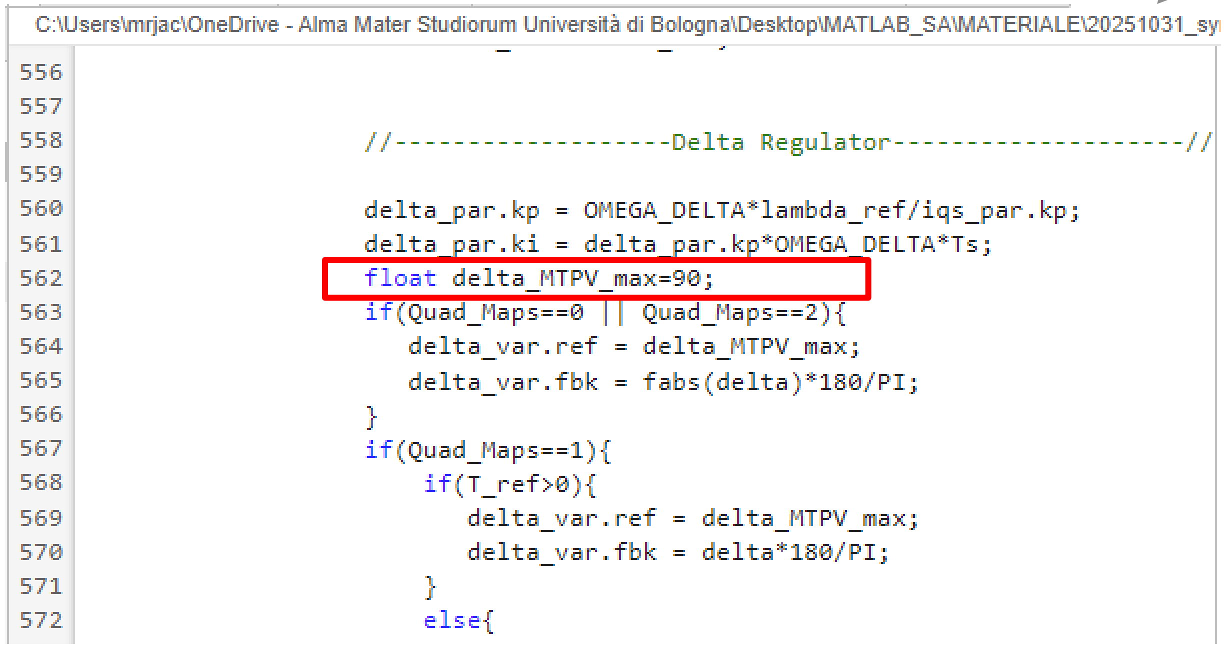




Fix #2 – control algorithm in C for Simulink

- Open Motor_ctrl.c
  - Line 562
    Add "float delta_MTPV_max=90;"
  - Save



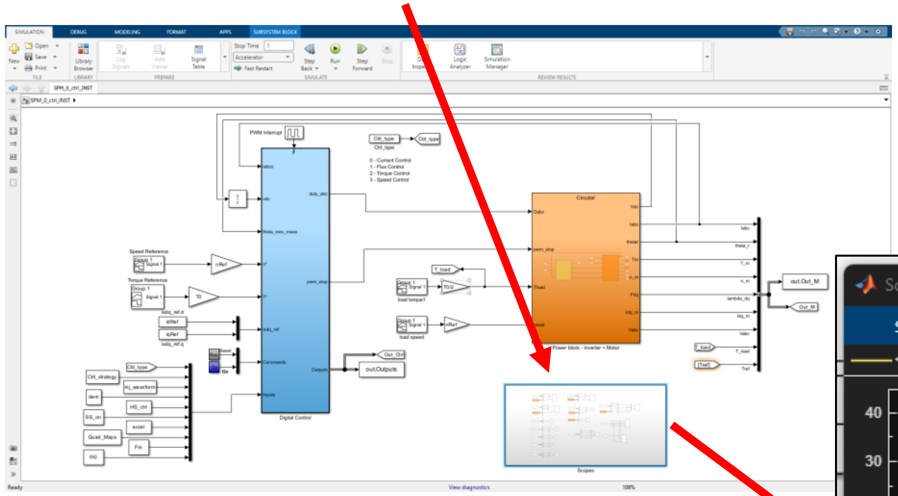

C:\Users\mrjac\OneDrive - Alma Mater Studiorum Università di Bologna\Desktop\MATLAB_SA\MATERIALE\20251031_sy

```c
556
557
558                 //------------------Delta Regulator-------------------//
559
560                 delta_par.kp = OMEGA_DELTA*lambda_ref/iqs_par.kp;
561                 delta_par.ki = delta_par.kp*OMEGA_DELTA*Ts;
562                 float delta_MTPV_max=90;
563                 if(Quad_Maps==0 || Quad_Maps==2){
564                    delta_var.ref = delta_MTPV_max;
565                    delta_var.fbk = fabs(delta)*180/PI;
566                 }
567                 if(Quad_Maps==1){
568                     if(T_ref>0){
569                         delta_var.ref = delta_MTPV_max;
570                         delta_var.fbk = delta*180/PI;
571                     }
572                     else{
```

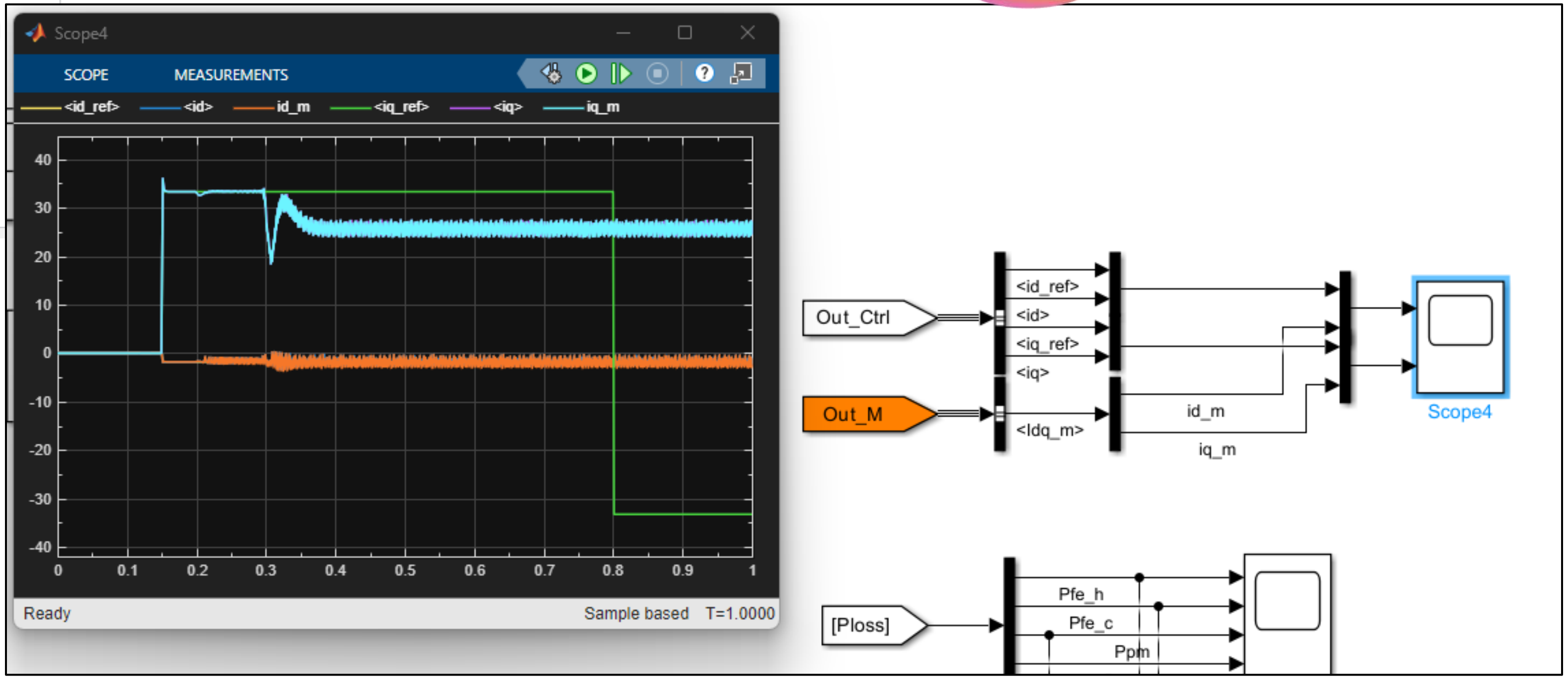
Now it woks

Simulink model – Results #1

- Investigate the "scopes" section

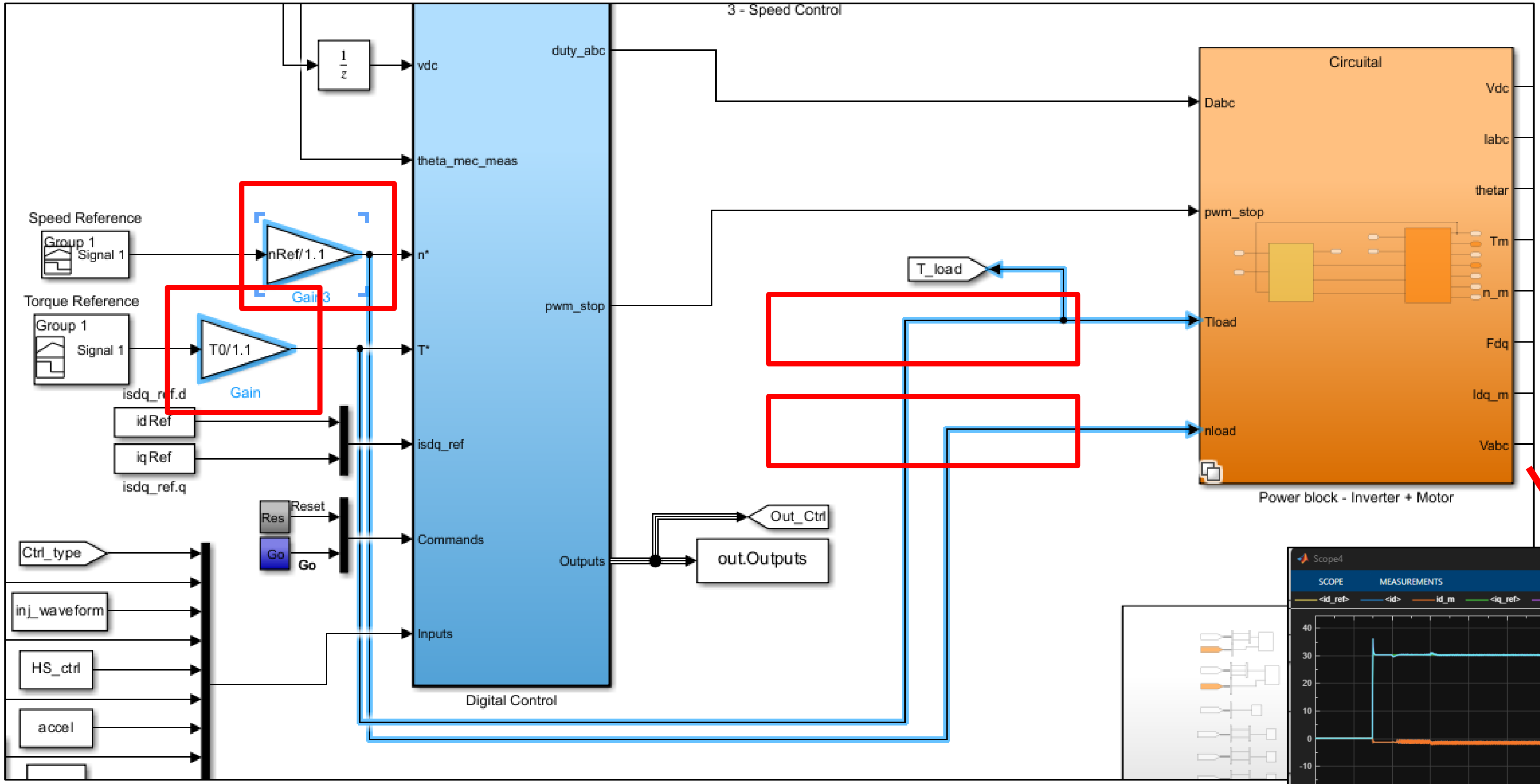


What is that??
Not very nice…

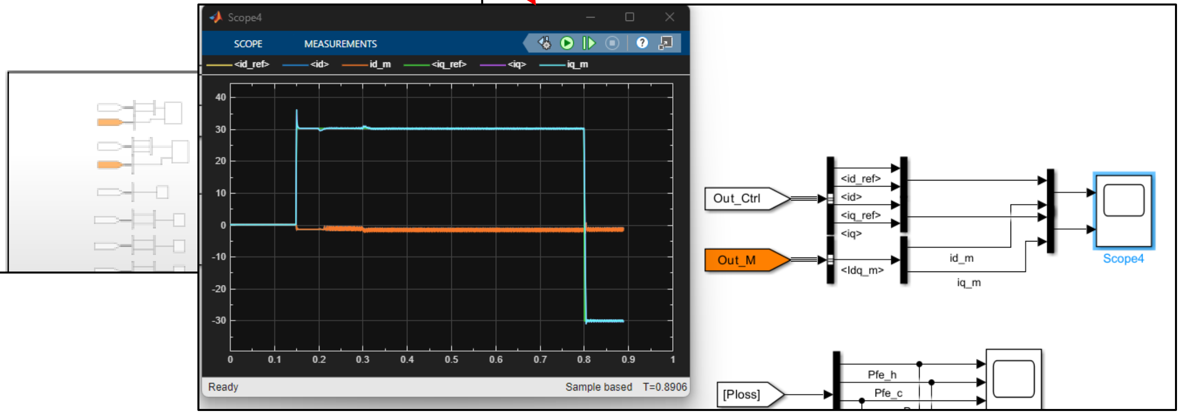


Simulink model – Results #2

- Reduce a bit nRef and T0, and short circuit their output to the power block



Now it woks

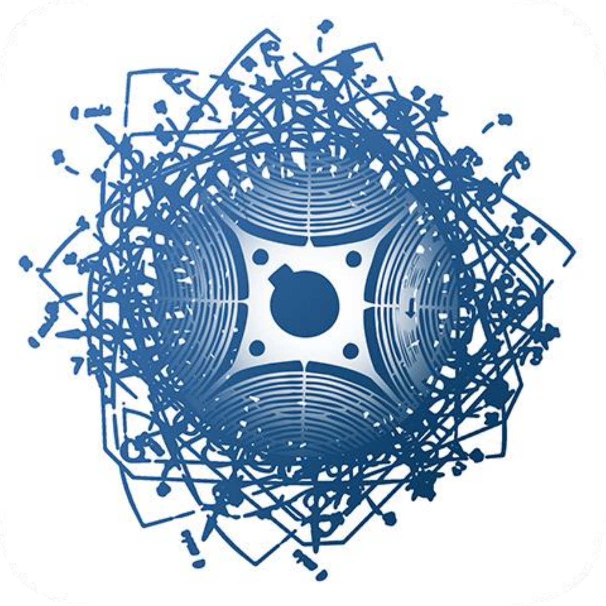


Simulink model – Results #3

- Play with inputs and see the results.
- Now you can export the motor and inverter model in any other simulink!

Main tackeaways

- Evaluation of a motor, knowing its geometry, materials, loads…
- Geometry optimization.
- MTPA/MTPV Map creation.
- Model for simulation in simulink/simscape.
- Export models to your favourite motor design/CAD software



Contacts

- Jacopo Ferretti
  - jacopo.ferretti5@unibo.it
- Paolo Panarese
  - ppanarese@mathworks.com
- Simone Ferrari (SyR-e)
  - simone.ferrari@polito.it