ANSYS MotorCAD 2022R2 新功能介绍

新科益系统与咨询（上海）有限公司

Agenda

Motor-CAD Highlights

- ✓ High fidelity synchronous machine lab model
- ✓ NVH Enhancements
- ✓ RPC automation for Motor-CAD

Ansys Motor-CAD 2022 R2 Release Highlights

What's new in Ansys Motor-CAD?

- NVH Enhancements
- Machine type modelling improvements
- New RPC automation with Python
- Motor-CAD to Maxwell export updates
- Motor-CAD to OptiSLang export updates

What's new in Ansys Motor-CAD?

- NVH Enhancements
- Machine type modelling improvements
- New RPC automation with Python
- Motor-CAD to Maxwell export updates
- Motor-CAD to OptiSLang export updates

Infinite cylinder and Acoustic Weighting model (new!)

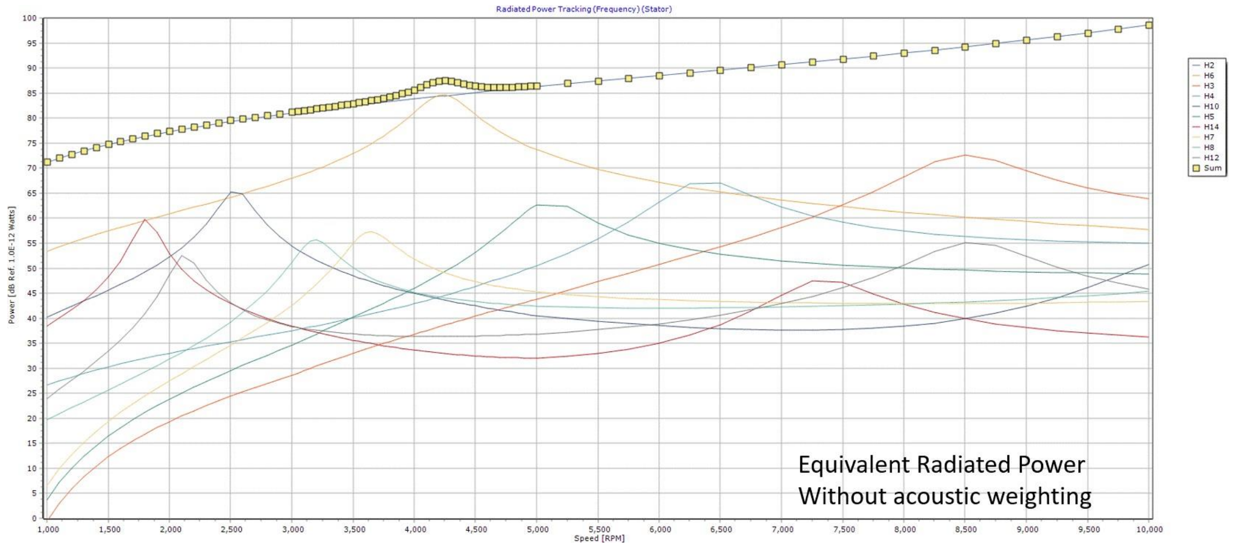
Acoustic model:

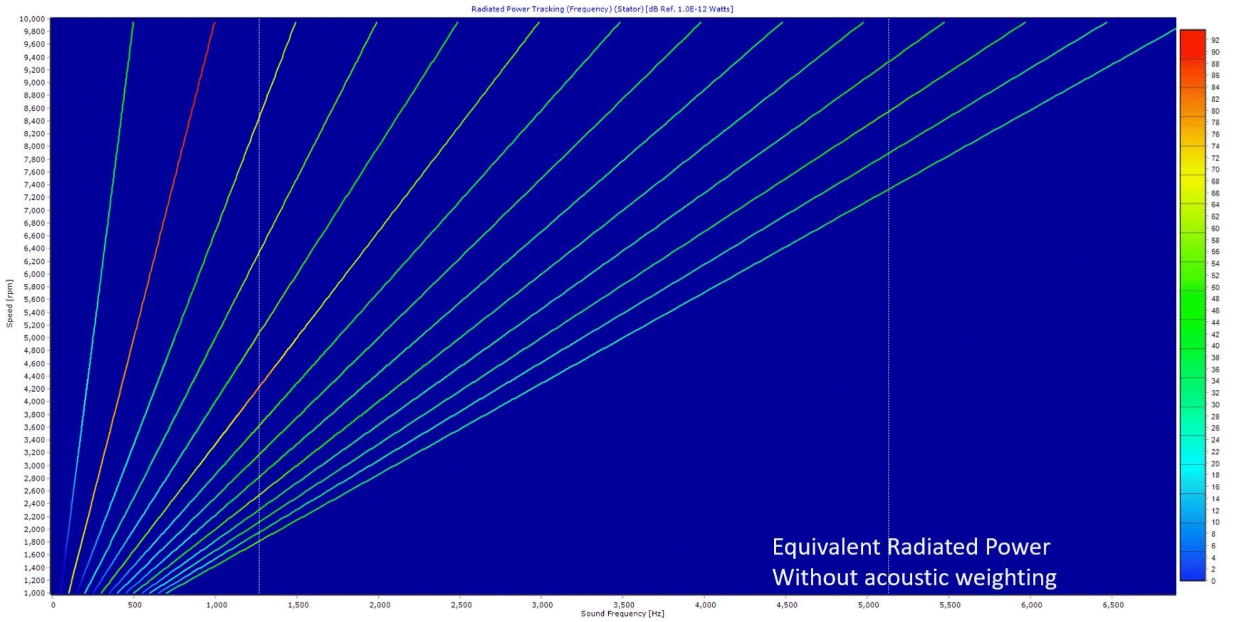
- Infinite cylinder acoustic model added, giving better prediction of low frequency sound radiation
  - This takes into account how efficiently stator vibration is converted into sound radiating away from the motor.

- Equivalent Radiated Power (ERP) is a good value for comparison between motors, but can substantially overestimate noise at low frequencies, where the wavelength of the sound is much larger than the size of the motor.

Weighting:

- A weighting can be applied to acoustic results, to give a closer match to human perception of noise level

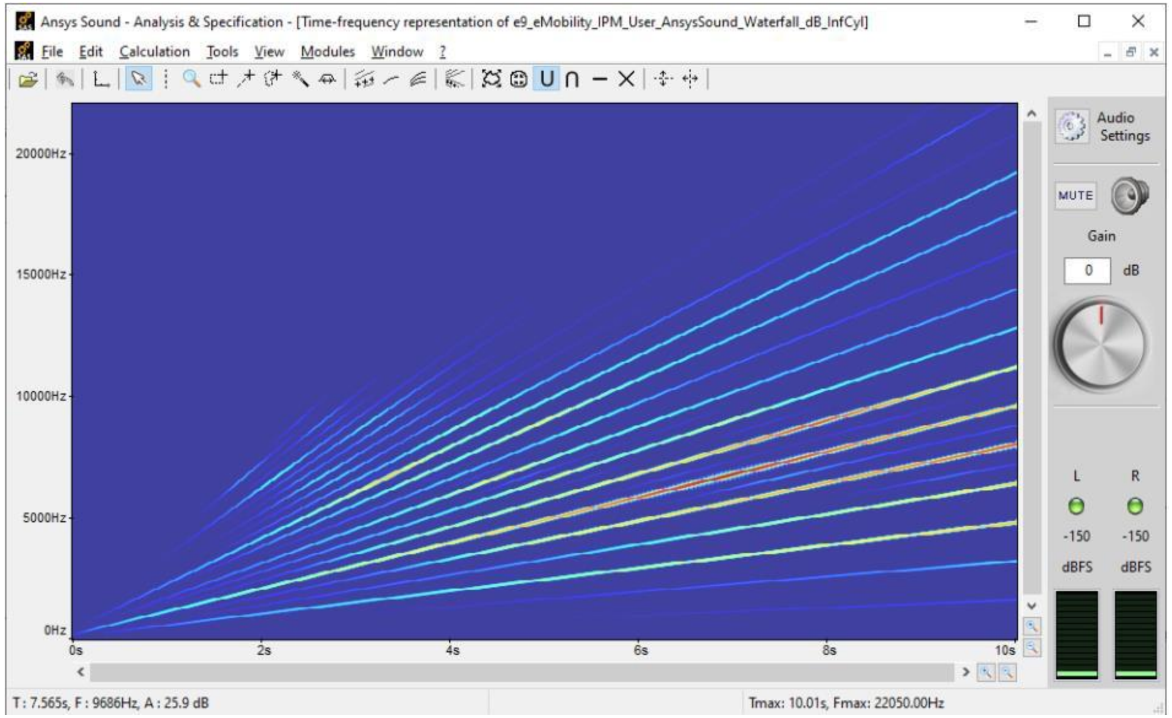




NVH: Export to Ansys Sound (new!)

Ansys Sound export:

- Motor-CAD noise predictions can be exported to Ansys Sound (SAS) in order to
  - Replay noise
  - Calculate psychoacoustic measures
  - Use Sound Composer to hear the predicted motor noise in a full acoustic environment including measured masking noise from wind and tyres



CAD-IT CONSULTANTS

What's new in Ansys Motor-CAD?

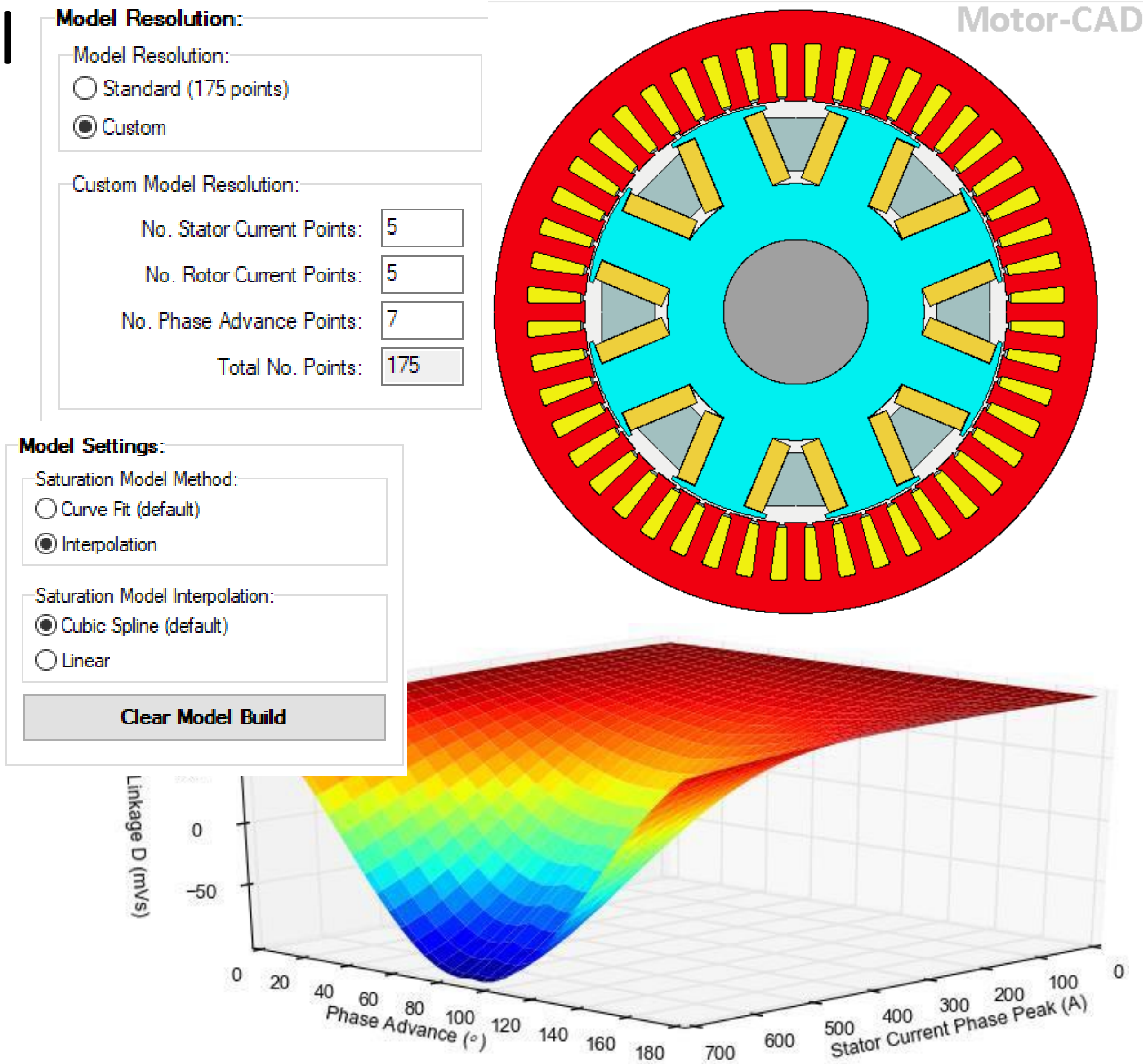
- NVH Enhancements
- Machine type modelling improvements
- New RPC automation with Python
- Motor-CAD to Maxwell export updates
- Motor-CAD to OptiSLang export updates

Synchronous machine lab model

- Model build resolution can now be specified
  - Enabling high fidelity efficiency map and drive cycle analysis
  - Interpolation methods may also be varied

- Rotor/stator loss bias ratio added
  - Enables tuning of the control strategy to shift joule losses between the rotor or the stator
  - Useful for ensuring continuous thermal
e maximised

Stator/Rotor Loss Bias Ratio 1



What’s new in Ansys Motor-CAD?

- NVH Enhancements
- Machine type modelling improvements
- New RPC automation with Python
- Motor-CAD to Maxwell export updates
- Motor-CAD to OptiSLang export updates

New Communication Method – JSON-RPC

- Replaces ActiveX for Python/MATLAB
- Motor-CAD starts an RPC server
- We can connect to this using a new Python module
- New *MotorCAD_Methods Module* for Python – installed with latest release
  - *MotorCAD* object attaches to Motor-CAD instance
  - Old Method:

```
import win32com.client
mc = win32com.client.Dispatch("MotorCAD.AppAutomation")
```

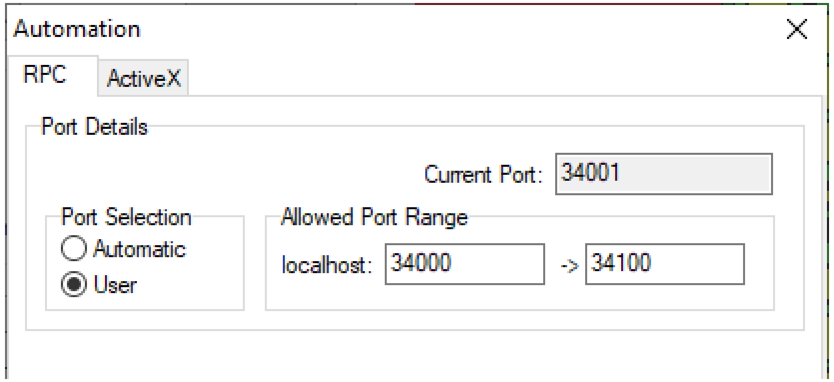
  - New Method:

```
from MotorCAD_Methods import MotorCAD
mc = MotorCAD()
```

- Syntax/behaviour can be kept identical to ActiveX

Current State of Work

- All ActiveX functions in 2022 R2
- RPC is hidden in 2022 R2
  - Enable in Defaults.INI *(RPC_Enabled)*
  - Settings in *Defaults -> Automation*



- Works with Python/MATLAB
- Docstring/error messages for most common functions

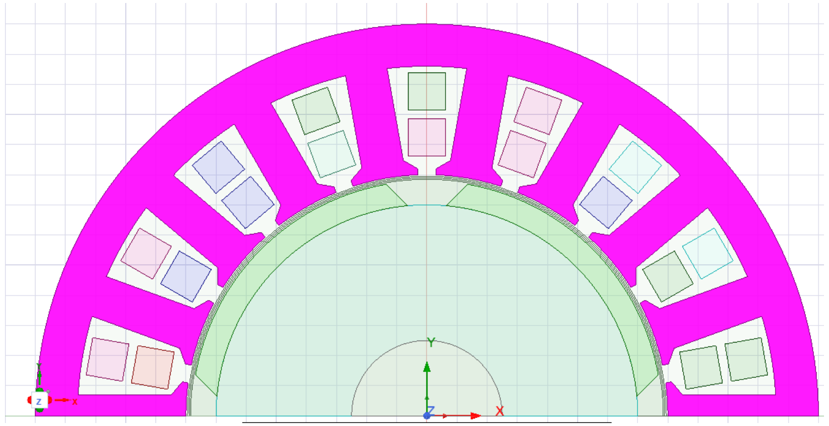
```
// Variables
function GetVariable(const variableName: string) : TJsonResult;
function GetArrayVariable(const arrayName: string; const arrayIndex: integer) : TJsonResult;
function SetVariable(const variableName :string; const variableValue : TJsonObject) : TJsonResult;
function SetArrayVariable(const arrayName: string; const arrayIndex: integer; const variableValue : TJsonObject) : TJsonResult;

// UI
function ShowMessage(const aMessage: string) : TJsonResult;
function Quit : TJsonResult;
function ShowMagneticContext : TJsonResult;
function ShowMechanicalContext : TJsonResult;
function ShowThermalContext : TJsonResult;
function DisplayScreen(const screenName : string) : TJsonResult;
function SaveScreenToFile(const screenName, fileName: string) : TJsonResult;

// Calculations
function DoMagneticCalculation : TJsonResult;
function DoSteadyStateAnalysis : TJsonResult;
function DoTransientAnalysis : TJsonResult;
function DoWeightCalculation : TJsonResult;
function DoMechanicalCalculation : TJsonResult;

// Lab
function ClearModelBuild_Lab: TJsonResult;
function SetMotorLABContext: TJsonResult;
function BuildModel_Lab: TJsonResult;
function CalculateOperatingPoint_Lab: TJsonResult;
function CalculateMagnetic_Lab: TJsonResult;
function CalculateThermal_Lab: TJsonResult;
function CalculateDutyCycle_Lab: TJsonResult;

// Geometry
function CheckIfGeometryIsValid(const editGeometry: integer): TJsonResult;

// Files
function LoadFromFile(const motFile : string) : TJsonResult;
function SaveToFile(const motFile : string) : TJsonResult;

// Internal Scripting
function LoadScript(const scriptFile : string) : TJsonResult;
function RunScript : TJsonResult;

// Graphs
function GetMagneticGraphPoint(const graphID : TJsonObject; const pointNumber : integer): TJsonResult;

// FEA
function GetPointValue(const parameter: TJsonObject; const x, y: Double): TJsonResult;

// Thermal
function GetNodeTemperature(const nodeNumber: integer): TJsonResult;
```

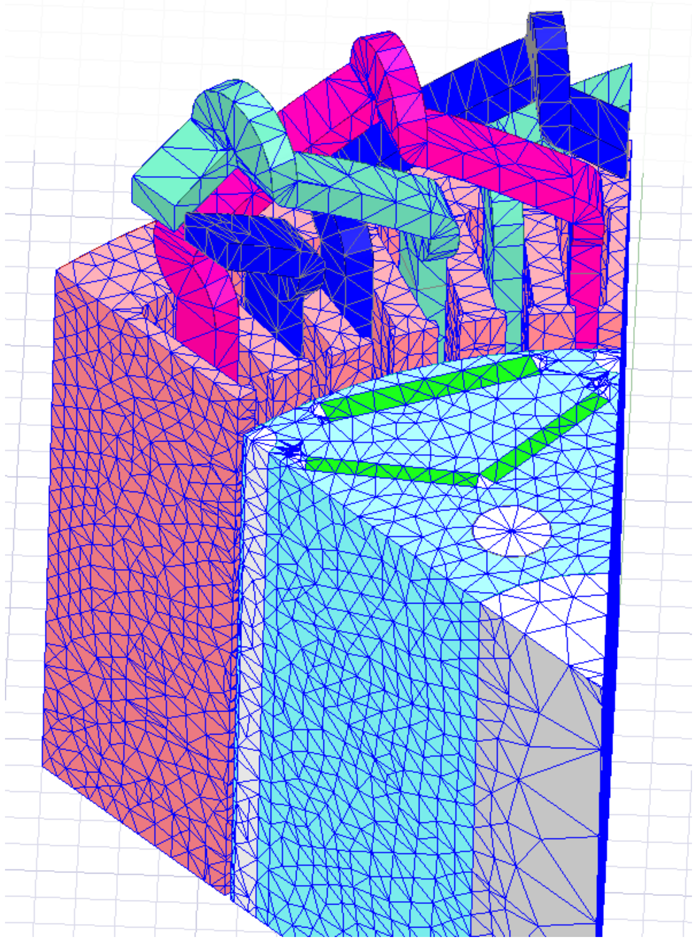
What's new in Ansys Motor-CAD?

- NVH Enhancements
- Machine type modelling improvements
- New RPC automation with Python
- Motor-CAD to Maxwell export updates
- Motor-CAD to OptiSLang export updates

UDP’s

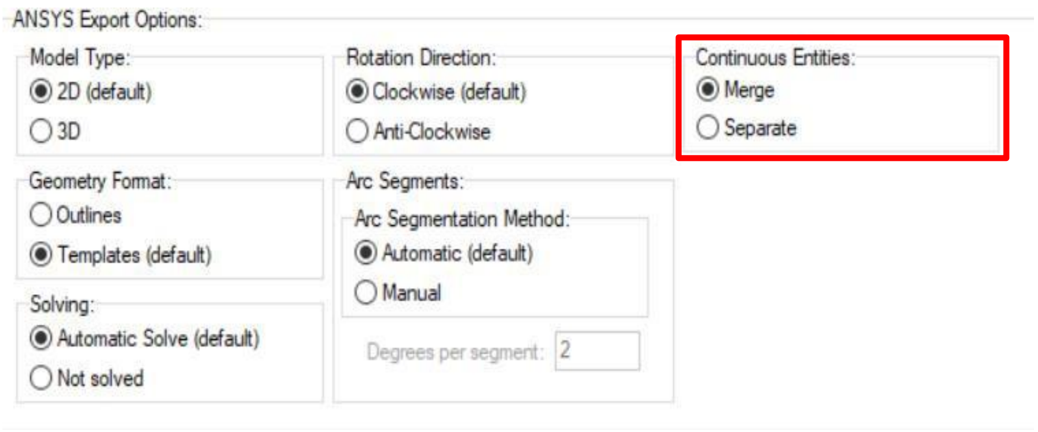
- Added new UDP’s :
  - BPM - Surface Radial and Surface Parallel (rotor/magnets)
  - Stator Parallel Tooth and Parallel Tooth Square Base
  - SRM – Added stator pole taper angle into existing UDP

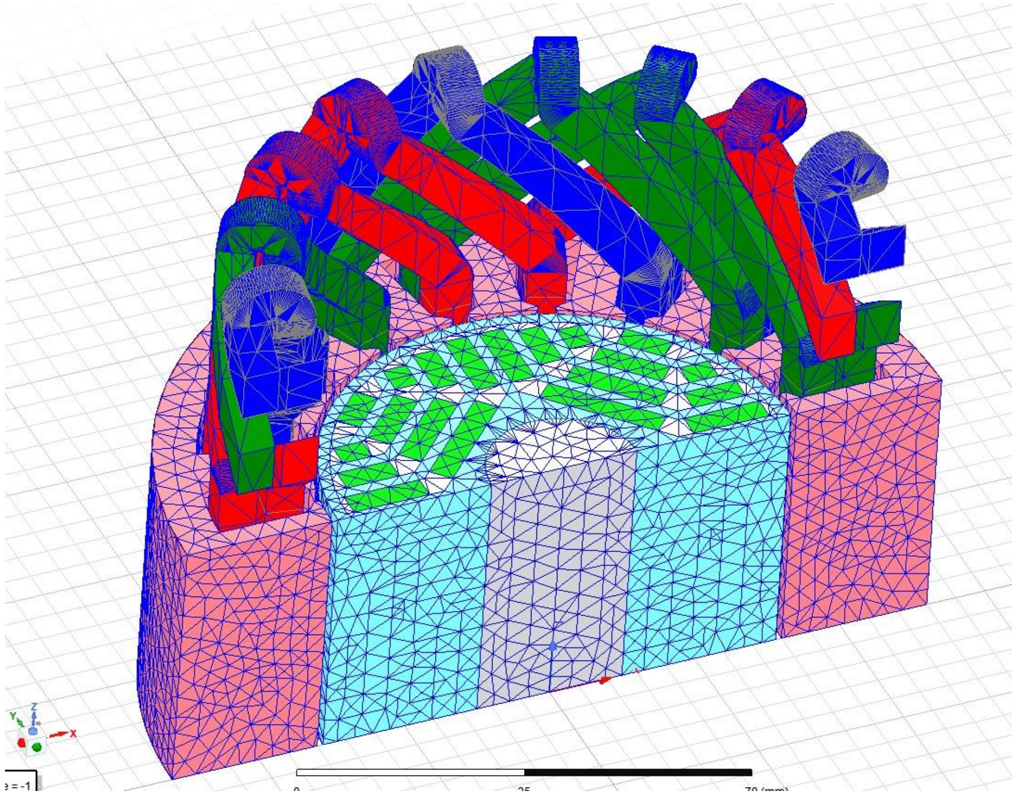




New Functionality

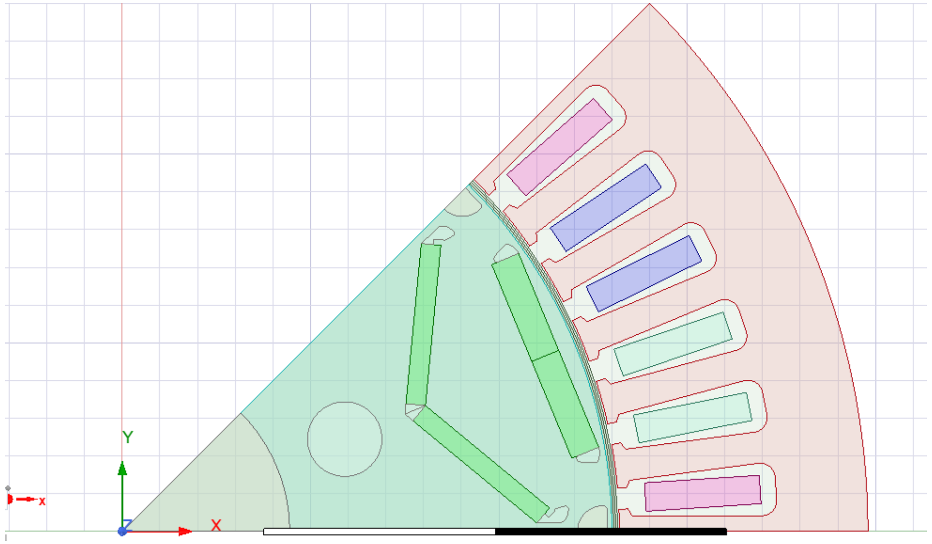
- Ability to enable or disable "Continuous Entities", which merges continuous entities (arc or line) within the polyline region.





Improved Functionality

- Improvements to polyline detection and creation when using imported DXF’s.
- Ability to replace arc entities within DXF geometries which are not valid within Maxwell, e.g. small arc entities replaced using a line entity.



CAD-IT CONSULTANTS

Overall benefits

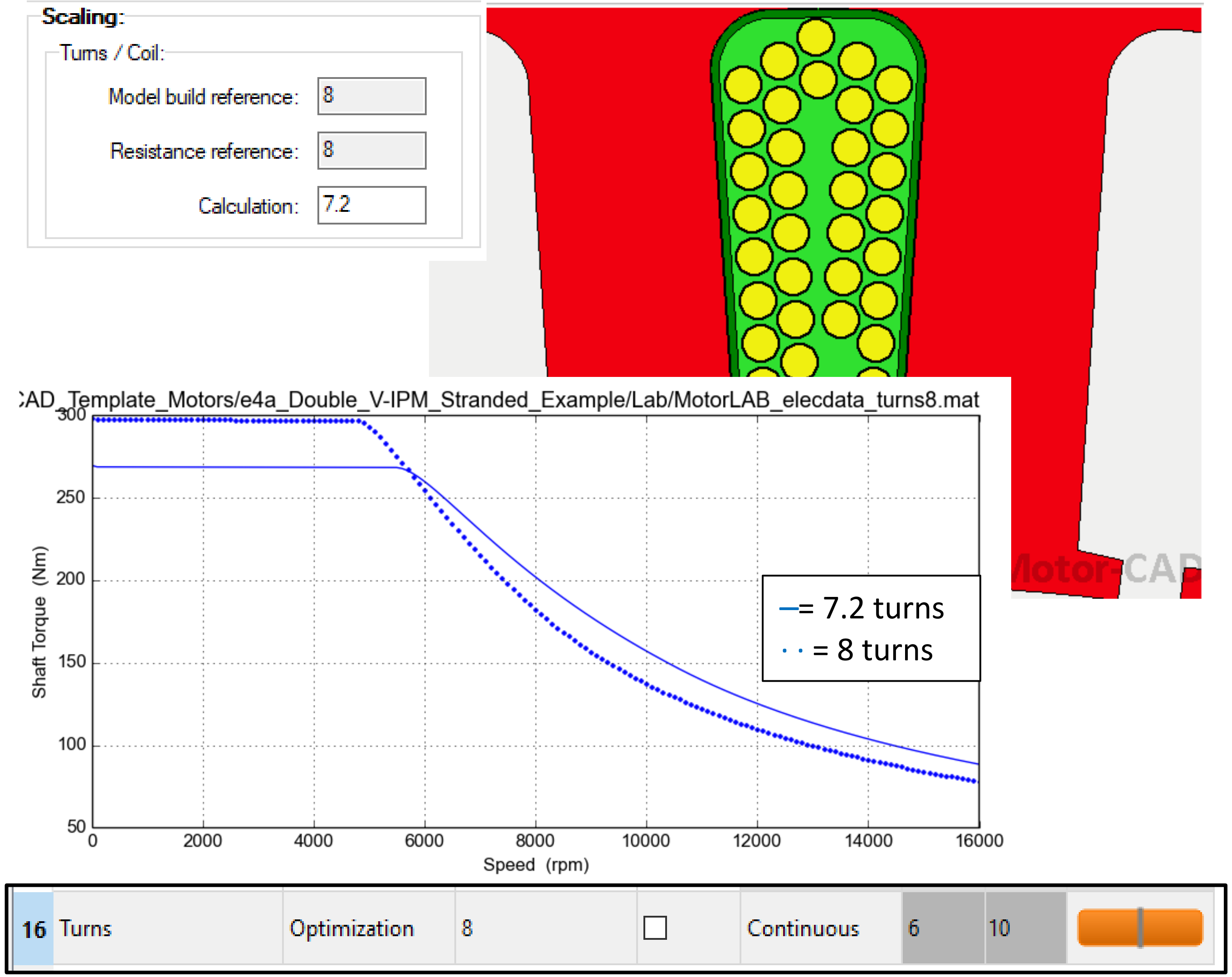
- Higher UDP coverage provides faster meshing times with the ability to use parameterised geometries within Maxwell.
- Decreased exported model generation times when using continuous entity merging and/or UDP's.
- Improved polyline detection and creation for DXF based models allows for accurate transfer of customised geometries into Maxwell.
- Ability to export and solve 3D models to accurately determine end effects etc using Finite Element solvers.
- Ability to use Maxwell Clone Meshing (User must select this option within Maxwell)

What's new in Ansys Motor-CAD?

- NVH Enhancements
- Machine type modelling improvements
- New RPC automation with Python
- Motor-CAD to Maxwell export updates
- Motor-CAD to OptiSLang export updates

Motor-CAD - optiSLang Export – Winding turns as input

- User can include the number of turns per slot as an input for optimisations using stranded winding.
- The Lab module turns scaling feature, allows non integer values to be calculated.
- This turns scaling enables a continuous range to be used at the sensitivity analysis stage, easing MOP generation and improving accuracy.

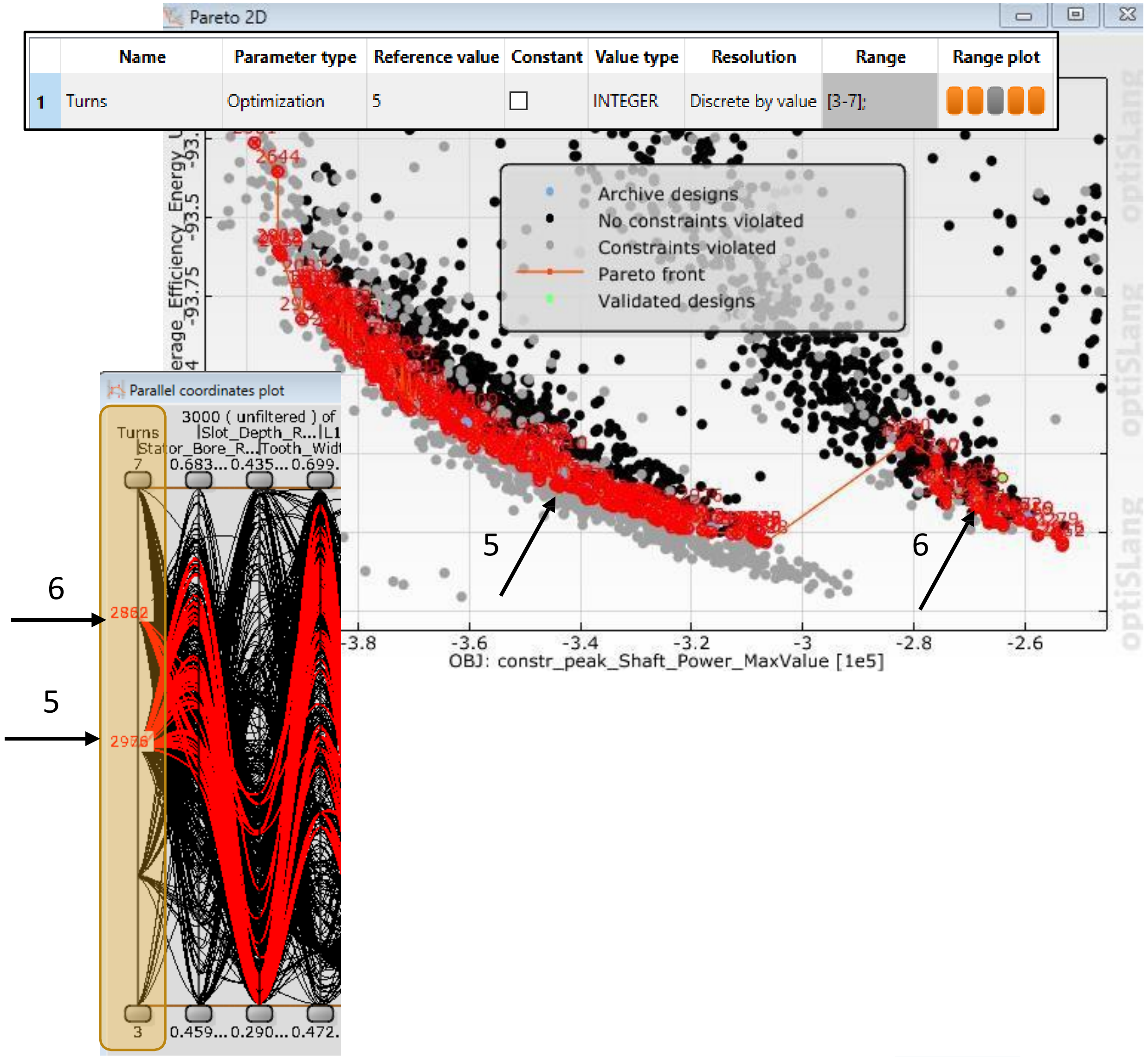


Motor-CAD - optiSLang Export – Winding turns as input

- At the optimisation stage, a discrete, integer range may be set to ensure only feasible windings are used.

- Here two distinct pareto fronts are present at two different values of winding turns.

- Available for stranded BPM motor types. It's a key parameter to be included, improving functionality for many users.



CAD-IT CONSULTANTS

Motor-CAD - optiSLang Export – Updated tutorial

- A new section has been added to the end of the Motor-CAD to optiSLang integrated export tutorial.

- It provides guidance and examples on how to customise the script, to alter inputs, outputs or do analysis, not currently included within the integrated export.

- This allows complete freedom to users, allowing them to combine the benefits of exported script with the ability to fully customise the optimisation.

Motor-CAD Software Tutorial:

Advanced Multiphysics optimisation of electric machines with Motor-CAD and optiSLang software tools

Application to an IPM traction motor

The script is shown below with the three new lines added (lines 193,194):

```
187     ### Handle housing scaling if applicable
188     success, statorDiameter - mcApp.GetVariable('Stator_Lam_Dia')
189     success, housingDiameter - mcApp.GetVariable('Housing_Dia')
190     housingThickness = housingDiameter - statorDiameter
191
192     ### Set number of magnet segments dependent on Active Length
193     Single_Mag_Length = 25
194     Mag_Segments = int(i_Active_Length/Single_Mag_Length)
195
196     ### Set parameters (do not change this comment)
197     mcApp.SetVariable('Ratio_Bore', i_Stator_Bore_Ratio)
198     mcApp.SetArrayVariable('RatioArray_PoleArc', 0, i_L1_Pole_Arc_Ratio)
199     mcApp.SetArrayVariable('MagnetThickness_Array', 0, i_L1_Magnet_Thickness)
200     mcApp.SetArrayVariable('PoleVAngle_Array', 0, i_L1_Pole_V_Angle)
201     mcApp.SetVariable('Ratio_SlotDepth_ParallelSlot', i_Slot_Depth_Ratio)
202     mcApp.SetVariable('Ratio_SlotWidth', i_Slot_Width_Ratio)
203     mcApp.SetVariable("Stator_Lam_Length", i_Active_Length)
204     mcApp.SetVariable("Rotor_Lam_Length", i_Active_Length)
205     mcApp.SetVariable("Motor_Length", i_Active_Length + motorExtension)
206     mcApp.SetVariable("Magnet_Length", i_Active_Length)
```

CAD-IT CONSULTANTS

新科益工程仿真中心

咨询邮箱：ansyssupport@cadit.com.cn

公司网址：http://www.cadit.com.cn