Brushless DC Motor

Calculating Cogging Torque with MagNet

Introduction

Any rotating machine with a salient rotor and stator exhibits "cogging" torque. The reluctance of the magnetic circuit varies as a function of rotor angle. As the rotor is displaced from an initial, low-reluctance position towards a higher reluctance, a torque resisting this displacement is experienced. As the rotor passes the point of highest reluctance, the torque then tends to attract it to the next point of low reluctance, creating a periodic torque waveform.

The Brushless DC (BDC) permanent magnet motor studied here exhibits a cogging torque even without any excitations.

MagNet's 2D Static Solver can be used to calculate this cogging torque waveform. The goal of this exercise is to familiarize the user with the methods used to calculate the cogging torque.

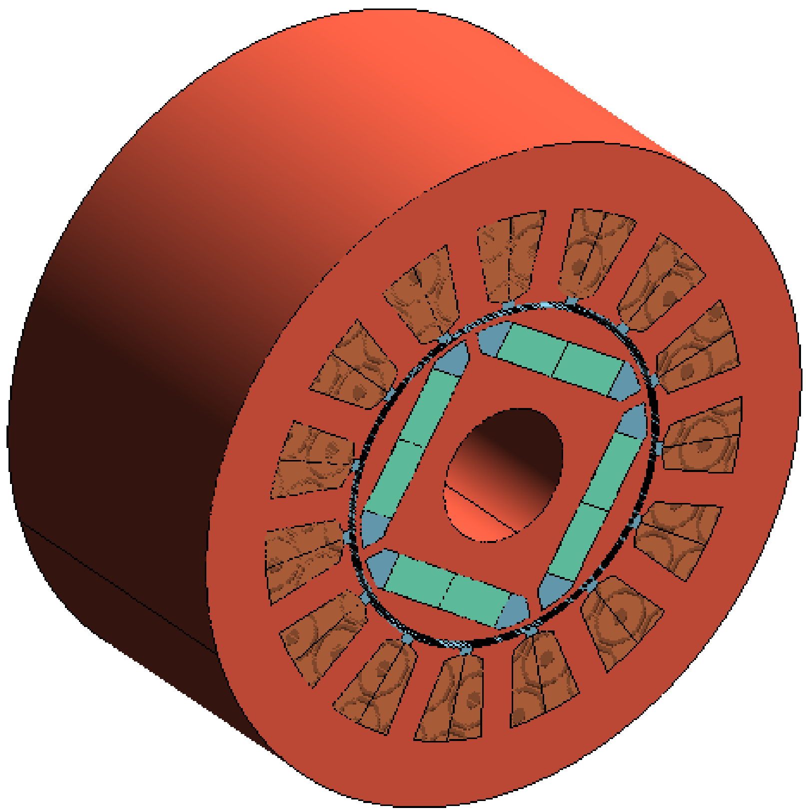

Figure 1: MagNet model of BDC motor

Creating the Model

Viewing the Mesh Refinement

Open the *BDCTutorial.mn* model and you will see that our motor is a single-barrier interior permanent magnet (IPM) machine with 15 stator slots and 4 rotor poles.

To ensure a good solution, some mesh controls have already been assigned to the model. Using *View→Initial 2D Mesh*, you can see how the mesh is refined in the air gap regions.

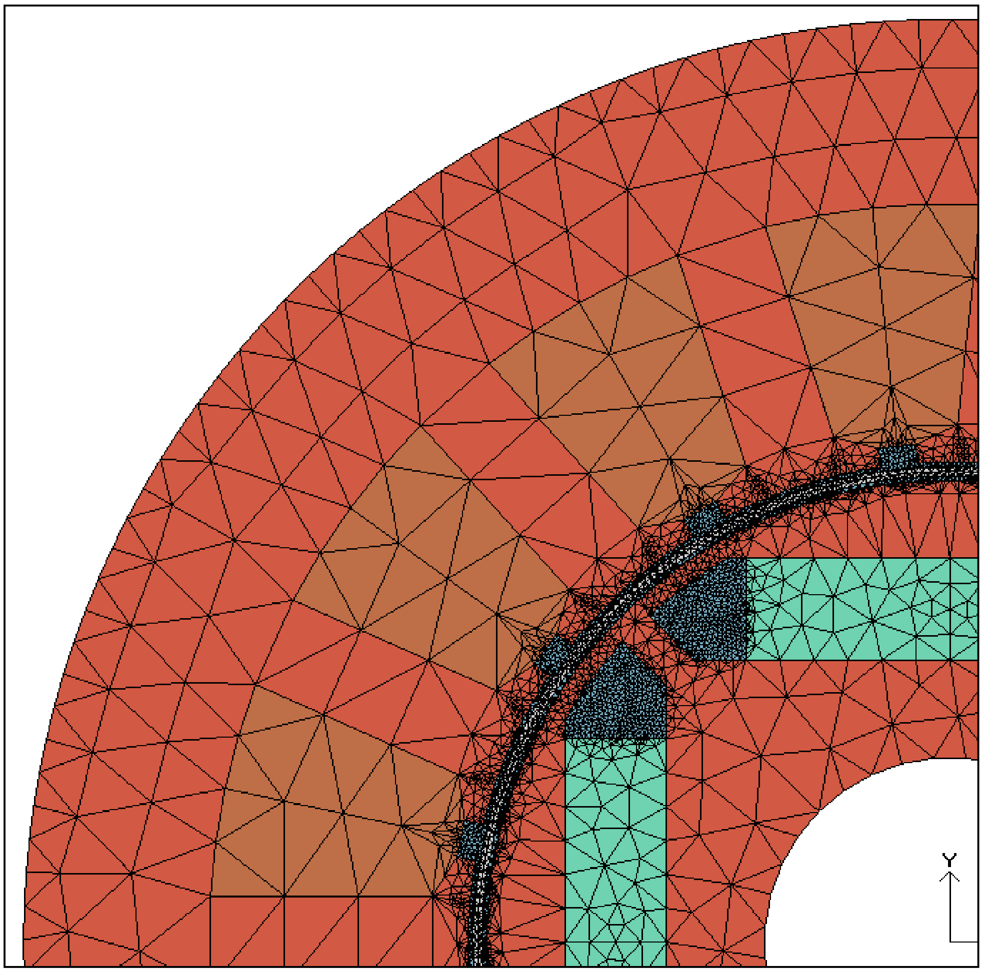

Figure 2: Quarter-model of BDC machine showing mesh refinement

Open up the Properties page for various components to see the mesh controls (specifically, the Maximum Element Size) applied.

Zoom in on the air gap region to see the special treatment applied here. For accurate torque and force calculations, air gaps in machines should be set up this way:

- Split the air gap into four layers.
- Assign the material "Virtual Air" to the layers closest to the stator and rotor.
- Assign the material "AIR" to the two other layers.

To calculate forces, MagNet must evaluate the field in the layer of AIR elements adjacent to the component. If this procedure is not done, MagNet will evaluate the field immediately adjacent to sharp corners where there is a potential for error. This method ensures that MagNet will calculate the forces and torques with accurate field values.

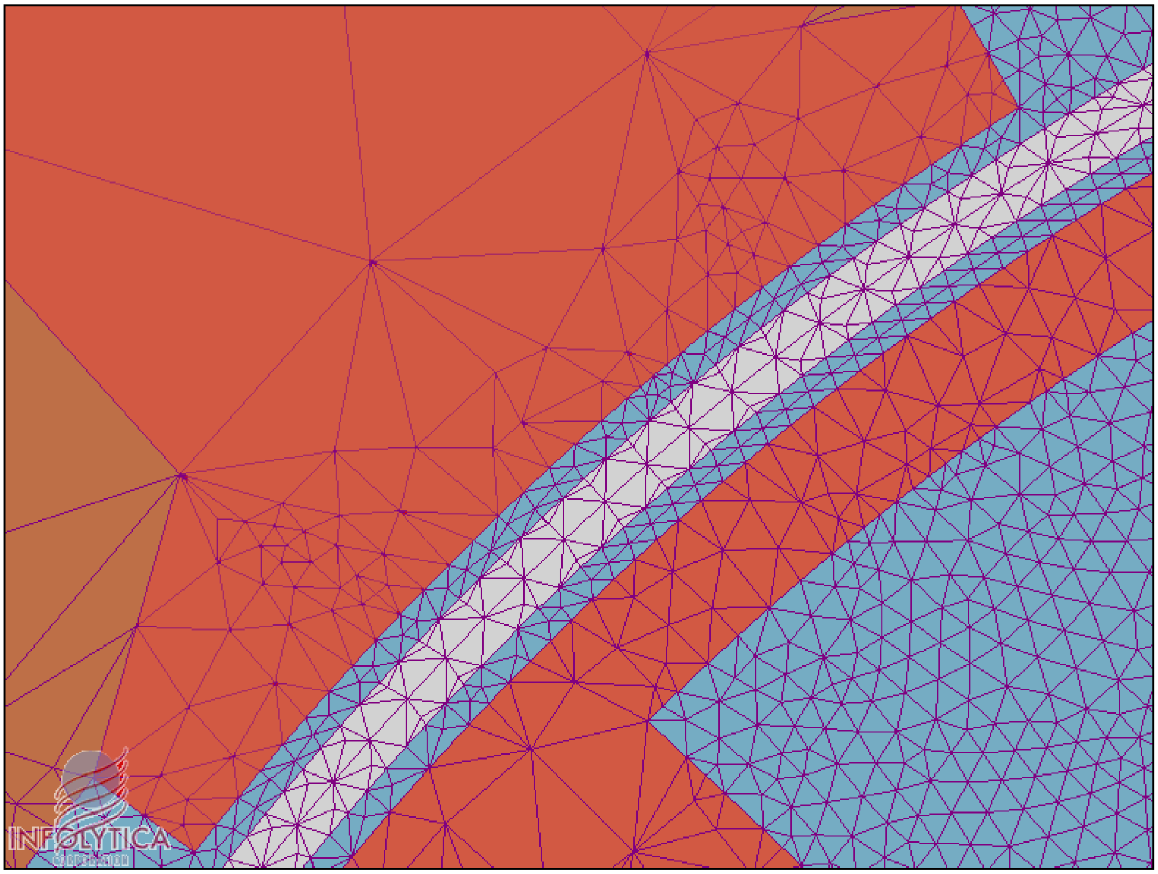

Figure 3: Air gap layout for optimal torque calculations

Assigning Parameters

In order to calculate the cogging torque, the static torque (rotor rotational speed equal to zero) is calculated for a series of rotor positions corresponding to 360 electrical degrees. At zero (and 360°) the rotor is assumed to be at a "rest" position corresponding to zero torque.

This motor has 15 stator slots and therefore a 12 degree sweep (360/15) will be the starting point for calculating the cogging torque. We are going to calculate the torque every 0.267 degrees for a total of 45 positions.

For some machines, the cogging torque periodicity is obtained simply from the geometry. For a more complex machine, the periodicity of the cogging torque waveform can be computed as follows:

$$\frac{\text{number of periods of cogging torque waveform}}{\text{stator tooth pitch rotation}} = \frac{\text{number of rotor poles}}{\text{HCF(\# of poles, \# of stator slots)}}$$

where the HCF is the highest common factor.

Example:

A permanent magnet motor has 21 stator teeth (and therefore 21 slots) and 16 permanent magnets on the rotor (giving 16 poles). The stator tooth pitch is therefore 360/21 or 17.14 degrees. The highest common factor of 21 and 16 is 1. Therefore, there are 16 periods of cogging torque every 17.14 degrees, giving a periodicity of 1.071 degrees.

For this motor, there are 15 slots and 4 poles, which implies there are four periods every 24 degrees, giving an expected periodicity of 6 degrees for the cogging torque. We are considering a 12-degree sweep, which should show two complete cycles of cogging torque.

For the cogging torque calculations, no excitation is applied to the windings, and motional effects are not considered. Therefore, the *Static 2D* solver is required.

Follow these steps to set up the parameterization that is needed.

1. In the *Object Page,* open the *Properties* page for top-level model.
2. Create the following parameters on the *Parameters* tab:

Parameter	Type	Expression
RotorAngle	Number	0, 0.2667, 0.5333, 0.8000, 1.0667, 1.3333, 1.6000, 1.8667, 2.1333, 2.4000, 2.6667, 2.9333, 3.2000, 3.4667, 3.7333, 4.0000, 4.2667, 4.5333, 4.8000, 5.0667, 5.3333, 5.6000, 5.8667, 6.1333, 6.4000, 6.6667, 6.9333, 7.2000, 7.4667, 7.7333, 8.0000, 8.2667, 8.5333, 8.8000, 9.0667, 9.3333, 9.6000, 9.8667, 10.1333, 10.4000, 10.6667, 10.9333, 11.2000, 11.4667, 11.7333

3. Select the "RotorSteel#1" component and open the *Properties* page.
4. Under the *Parameters* tab, locate the parameter called **RotationAngle** and set the *Expression* to %RotorAngle %deg. This will cause the rotor steel component to be rotated by **RotorAngle** for each of the problems.
5. Repeat Step 4 for RotorAir, RotorInnerAir#1 - #8, and Magnet #1-#8. You may want to make use of the script form *ManageParameters.frm* to make this task easier.
6. Now view a few problems using the *View* page and the *Update View* button, and verify that the entire rotor structure is turning.

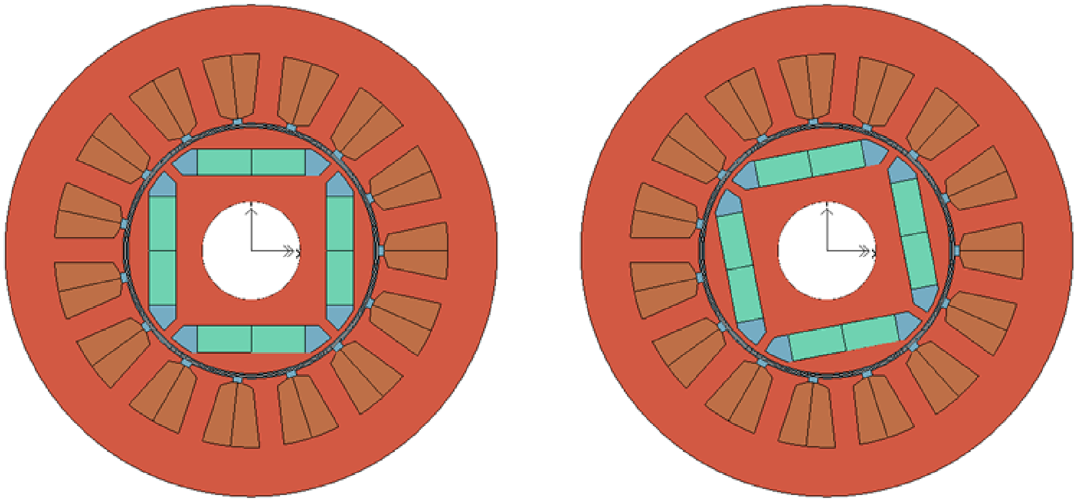

Figure 4: Models showing rotor rotation.

The model is now ready for solution.

Under the *Set Solver Options* dialog box (access with *Solve*→*Set Solver Options*), ensure the polynomial order is set to 1.

Click on *Solve*→*Static 2D*, and MagNet will solve each problem.

Post-Processing

After solving is complete, ensure that the *Post-processing Bar* is visible under the *View* menu.

Under the *Force* tab on the *Post-Processing Bar*, you can view the torque on the rotor components.

Energy | Force | Flux Linkage | Power Loss | Current

	Force		Torque	
	Vector	Magnitude	Vector	Magnitude
RotorSteel#1 + RotorAir + RotorInnerAir#1 + RotorInnerAir#2 + RotorInnerAir#3 + RotorInnerAir#4 + RotorInnerAir#5 + RotorInnerAir#6 + RotorInnerAir#7 + RotorInnerAir#8 + Magnet#1 + Magnet#2 + Magnet#3 + Magnet#4 + Magnet#5 + Magnet#6 +	(0.3521602810798, 0.007398160603448, 0)	0.3522379825495	(0, 0, 1.116296310546e-005)	1.116296310546e-005

Figure 5: Post-processing Bar

MagNet can easily create a graph of the cogging torque as a function of rotor position. Click on the *Torque Vector* column and click the *Graph Selection* button.

In order to see the cogging torque, you must look at the component of the torque vector about the Z-axis (the other two components will be zero).

The graph should look something like the one below.

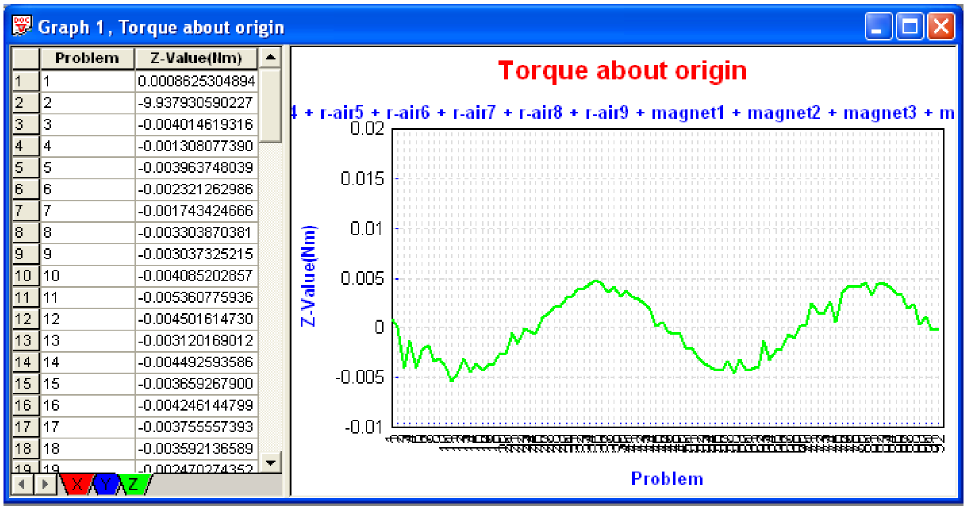

Figure 6: Cogging torque from 1st-order solution

The graph does show a periodically varying torque with a peak-to-peak value of roughly 10 mNm, however there is some noise in the curve.

Set the polynomial order to 2, and solve the problem again.

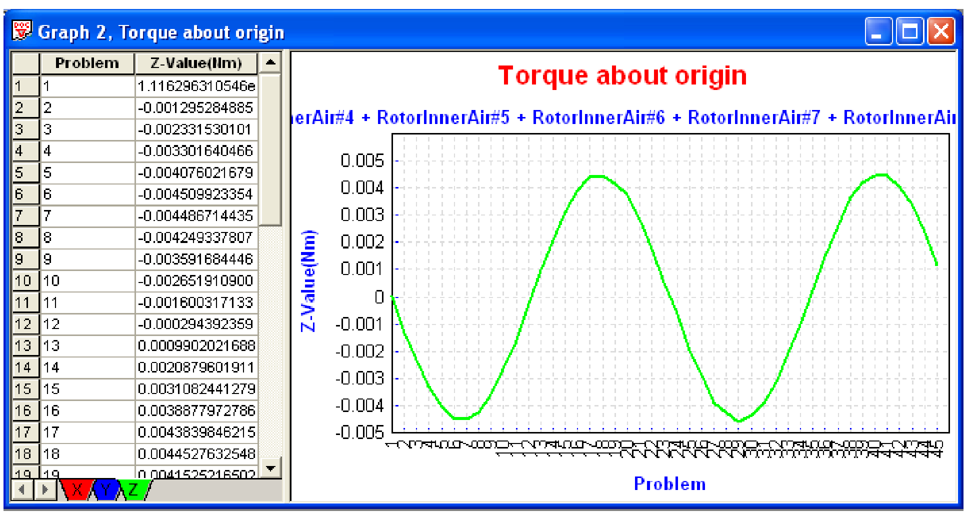

Figure 7: Cogging torque from 2nd-order solution

The second-order solution gives much smoother results, although the peak cogging torque is roughly the same (around 9 mNm). It is also verified that the period of the waveform is 6 degrees.

Conclusions

The cogging torque of a motor, due to attraction/repulsion cycles caused by the interactions of permanent magnets and a salient stator, is of interest to motor designers because it creates a torque ripple and can lead to destructive vibrations. MagNet allows for a fast and accurate calculation of the cogging torque and can help designers minimize its negative effects.

Skewing

A standard technique for the reduction of cogging torque is *skewing* of the rotor or stator—or introducing a helical twist. In the case of an interior permanent magnet machine such as this example, it is impractical to skew the rotor as the shape of the permanent magnets required becomes too complicated, and therefore the skewing of the stator will be considered.

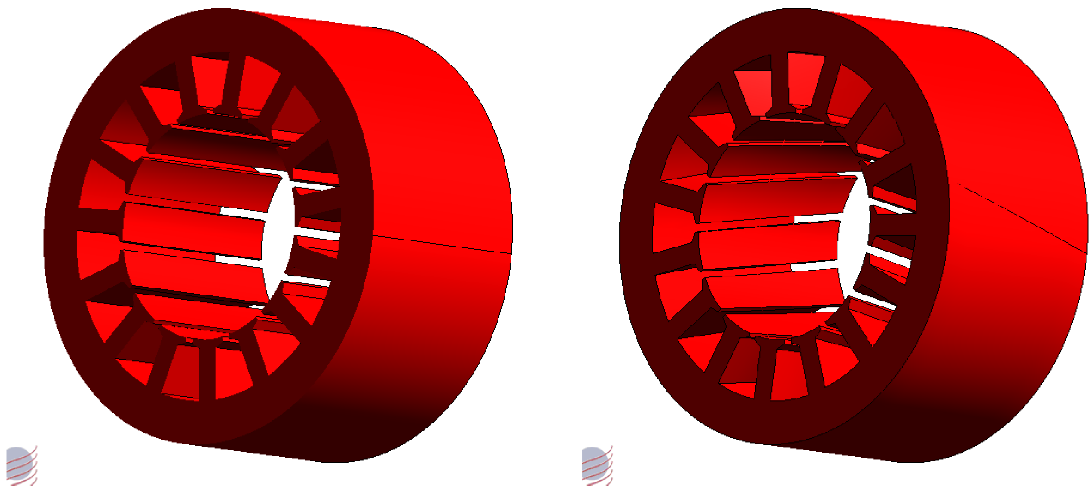

Figure 8 : Straight and skewed stator stacks

Skewing is technically a 3D problem, however, a method has been devised for obtaining approximate results using only 2D simulations.

The following details the procedure to obtain the cogging torque-angle curve of a skewed rotor machine, from a 2D model.

Let L be the thickness (in the z-direction) of the machine. The procedure is based on the fact that the actual 3D motor with skewed rotor can be imagined to correspond to a stack of N un-skewed machines having a thickness of L/N, each one with a slightly different relative position between the rotor and the stator, thus mimicking the actual 3D skew. This can be seen as a "discrete" approximation of a continually skewed machine. The torque-angle curve of the actual 3D motor is therefore approximated as the sum of the torque-angle curves of each of the virtual "thin motors".

The first curve is obtained from a 2D model parameterized at various rotor angles, e.g. 0, a, $2a$, $3a$, ..., $(M-1)a$, a being the angular increment between each rotor position. This curve would correspond to the torque-angle curve of the first virtual "thin machine" of thickness L/N, with the rotor at a given start angle. The data should be such as to cover an integral number of cycles, minus one point.

The second curve would correspond to that of the second virtual "thin machine" of thickness L/N, with the rotor at the (previous start angle + a) degrees. This second torque-angle curve does not have to be calculated in *MagNet*. It is simply a shifted (or rather circularly permutated) version of the first curve, and can be obtained easily in an *Excel* worksheet.

The third torque-angle curve is likewise calculated in *Excel* from the second curve, and so forth.

All these torque-angle curves are then added. If the 2D simulations giving the first torque-angle curve were performed on a machine of thickness *L*, then the summation of torque-angle curves must be divided by *N*.

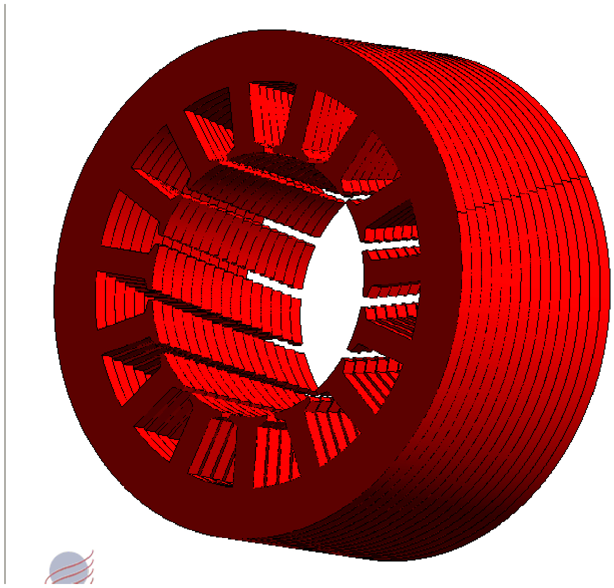

Figure 9: Approximation of skewed motor by piecewise-skewed stator.

There are two benefits in using 2D for this case:

The 2D system does not see the skewed rotor and hence the cogging torques are much larger and as a result easier to obtain accurately;

The 2D system already treats the laminations correctly where as in 3D, the laminations should be modeled as anisotropic.

Infolytica has produced a tool to perform this calculation, which can be downloaded from our website.

The tool allows the user to enter the desired skew angle in degrees, and the number of poles and slots. The tool will then generate a graph of the cogging torque for both the original and skewed machines.

First, click the "Link to MagNet" button. The tool will verify the model and extract a list of bodies in the model. The tool will require the user to select which body in the model represents the rotor. It will attempt to determine this information automatically, but the user should confirm the choice before producing the graphs. Click "Graph" to generate graphs of the cogging torque before and after skewing.

The skewed stator reduces the cogging torque significantly. This piecewise-skewing method gives reasonable results in a small fraction of the time that would be required for a 3D analysis.

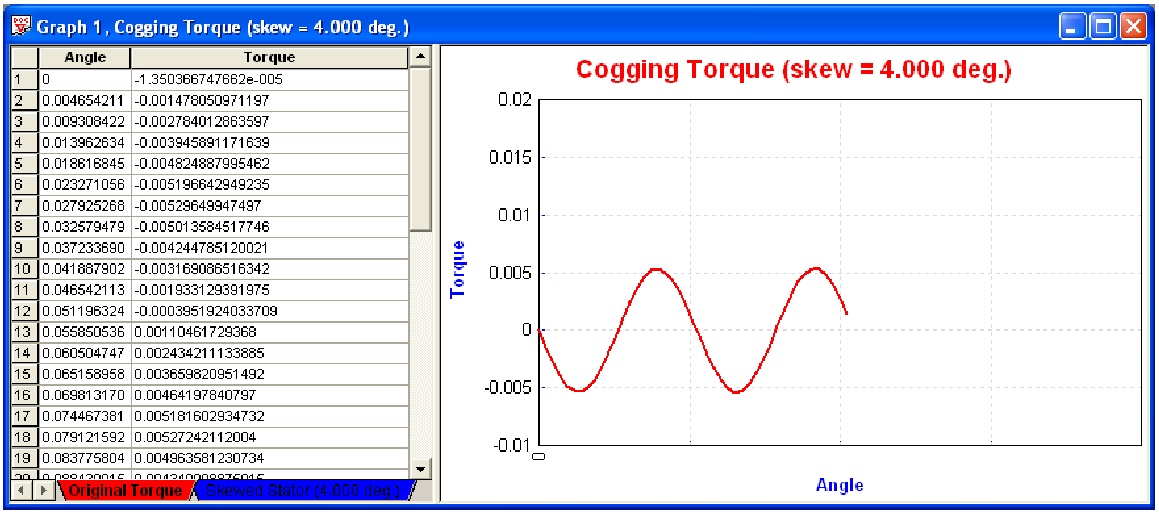

Figure 10: Magnet Cogging Torque for Brushless DC Machine, with and without Skew

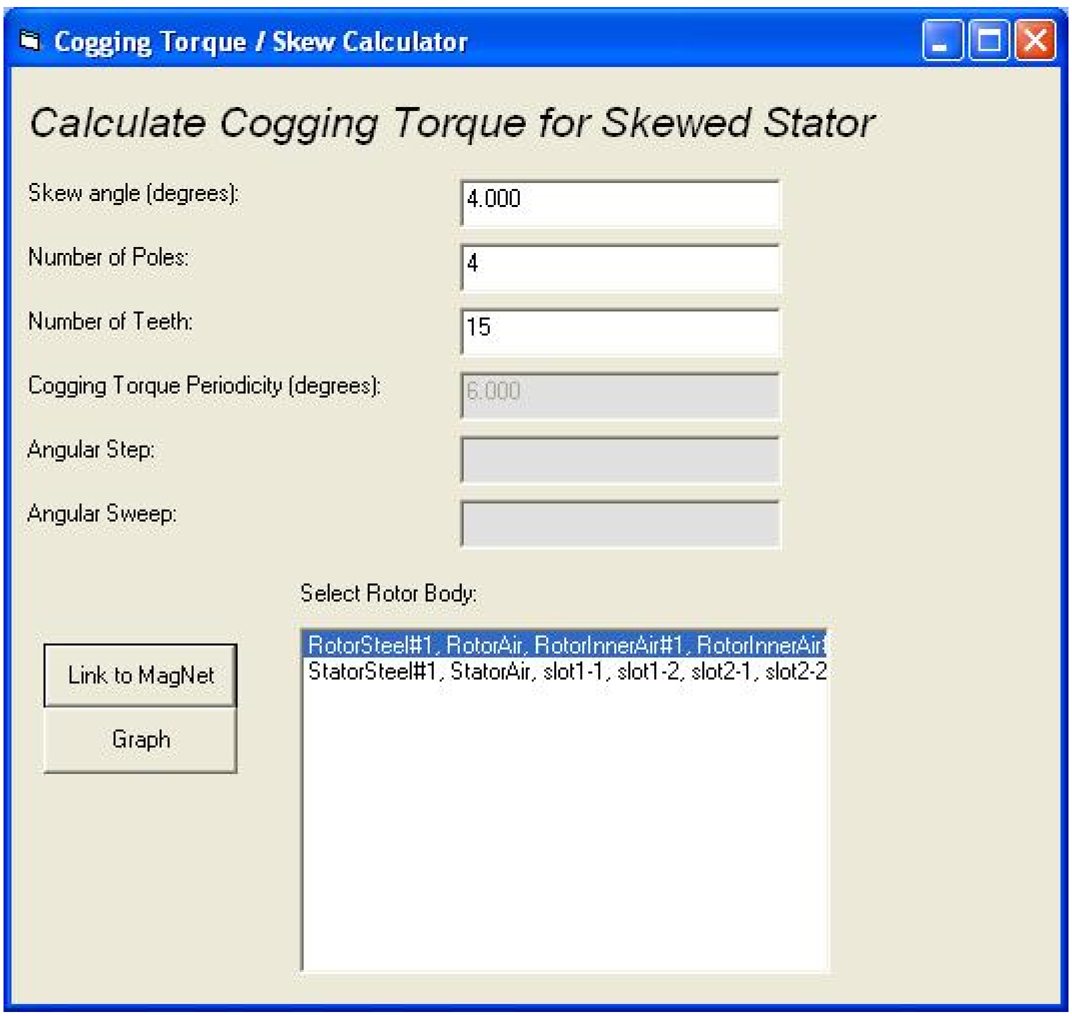

Figure 11: Infolytica Cogging Torque and Skew Tool

Appendix

In the setup of the parameters, the rotor angles were entered explicitly. If it is desirable to calculate the cogging torque over multiple sweeps, or for different step sizes, the parameters can be set up this way, such that the angles are calculated based on the total angular sweep and the number of steps used.

Parameter	Type	Expression
RotSteps	Number	45
Sweep	Number	12
RotMechStep	Number	%Sweep / %RotSteps
RotorAngle	Number	0, 1 * %RotMechStep, 2 * %RotMechStep, 3 * %RotMechStep, 4 * %RotMechStep, 5 * %RotMechStep, 6 * %RotMechStep, 7 * %RotMechStep, 8 * %RotMechStep, 9 * %RotMechStep, 10 * %RotMechStep, 11 * %RotMechStep, 12 * %RotMechStep, 13 * %RotMechStep, 14 * %RotMechStep, 15 * %RotMechStep, 16 * %RotMechStep, 17 * %RotMechStep, 18 * %RotMechStep, 19 * %RotMechStep, 20 * %RotMechStep, 21 * %RotMechStep, 22 * %RotMechStep, 23 * %RotMechStep, 24 * %RotMechStep, 25 * %RotMechStep, 26 * %RotMechStep, 27 * %RotMechStep, 28 * %RotMechStep, 29 * %RotMechStep, 30 * %RotMechStep, 31 * %RotMechStep, 32 * %RotMechStep, 33 * %RotMechStep, 34 * %RotMechStep, 35 * %RotMechStep, 36 * %RotMechStep, 37 * %RotMechStep, 38 * %RotMechStep, 39 * %RotMechStep, 40 * %RotMechStep, 41 * %RotMechStep, 42 * %RotMechStep, 43 * %RotMechStep, 44 * %RotMechStep