Gamma calculation for MTPA on a magnet machine: methodology comparison.

Cyril Favre ; Patrick Lombard - CEDRAT.

In the study of magnet machines, the research for the ultimate control is a fundamental point. For this research, **we propose three methods**, based on the use of finite elements which deliver more accurate results than traditional analytical methods. The objective here is to find the control, current and gamma angle which delivers maximum torque per amp (MTPA) for each operating point on the machine (defined by torque and speed), thereby minimising losses. The first method uses parametric resolution under Flux®. The second method uses a macro developed under Python® to find gamma delivering MPTA. Finally, the third method uses our GOT-It optimisation tool.

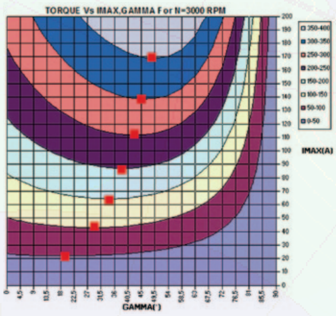

Fig.1: Torque of a magnet machine according to gamma current.

When the torque of a permanent magnet machine is plotted for a given speed, according to current and gamma value (angle between induced FEM and current), cf. figure 1, note that the gamma value which gives the maximum torque per amp is not a constant value. The three methods described below defines the optimum (red dots).

1. Flux® multiparametric calculation

Using multiparametric calculations under Flux, it is possible to make a calculation whilst varying several parameters. It is therefore possible, for a given speed, to calculate torque according to time for different current and gamma angle values. Thanks to these calculations, the gamma value for a given torque value can be deduced, allowing minimum injection of current.
Figure 2 below illustrates the geometrics described under Flux®, as well as the electrical circuit used for the multiparametric calculation:

The current sources in the Flux project are defined according to speed, time and gamma in the following way:
I_1= IMAX*Sin(OMEGA*TIME+GAMMA*Pi()/180)
I_2= IMAX*Sin(OMEGA*TIME+GAMMA*Pi()/180-2*Pi()/3)
I_3= IMAX*Sin(OMEGA*TIME+GAMMA*Pi()/180-4*Pi()/3)
With:
OMEGA = 2*PI*FREQUENCY
FREQUENCY = (SPEED/60)*(POLES/2)
SPEED= Speed of the motor in RPM, used for the definition of the overall mechanics at the given speed
POLES= Number of poles on the machine

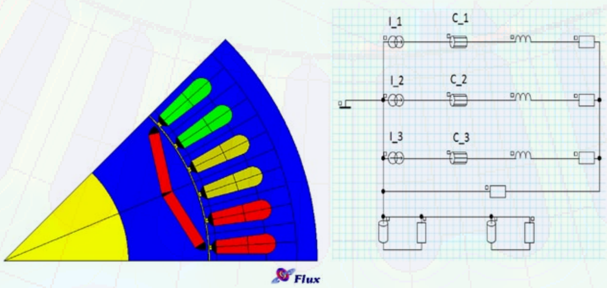

Fig. 2: Finite element model and circuit.

The results obtained from the multiparametric calculations under Flux® can be exported to Excel®, for post-processing to obtain the average torque value for each of the IMAX GAMMA parameter sets. The results obtained are shown in the graph in figure 1. In this graph, you can graphically locate gamma, which allows us to obtain the maximum torque per amp. The graph in figure 1 was constructed using 441 transitory calculations under Flux®. In this example, the optimal value of gamma, minimising the current consumed by the motor to obtain torque of 150Nm at 3,000 RPM is 35°.

2.Use of Flux® and macro in Python®

The **second method** proposed uses a macro (for a given operating torque and speed) to find the gamma value given MTPA.
Macro definition: In Flux®, a macro is a set of high level controls added to the Flux application, which groups together several controls in a given order. The macros are developed with PyFlux (Python for Flux), a language based on Python which is compatible with Java and allows customisation of your Flux version and improved efficiency of the interaction between software and user.

The Flux macro used is called «***FindOutCurrentGammaForGivenTorqueSpeed.PFM***». This macro, developed by CEDRAT, allows us, for a given operating point (Torque, Speed), to find the gamma value which gives the MTPA by performing several Flux calculations with different gamma and current values. The starting point in using this macro is a transitory magnetic flux project of a magnet machine identical to the project use for the multiparametric calculation (cf. figure 2). Input/output parameters allow us to define the physics of the project, to define the parameters used to locate gamma, and the desired level of accuracy torque and the torque target to be achieved. They also enable the macro to pilot Flux resolutions. The resolution scenario used in the Flux project is piloted by the position of the mechanics as a whole, and represents a period of torque, allowing us to reduce the resolution time for each iteration. The search for gamma given the MTPA, thanks to the macro, is performed in three stages:

1) The value of current is fixed at a calculated value, thanks to the constant of the torque of the machine. For this value, we are looking for the gamma current which gives the maximum torque. The search for gamma is done iteratively, and according to the following synoptic:
2) For the gamma value found in the previous stage, we are now looking for the value of the current given the desired torque.
3) We check the resilience of the solution (Gamma, Imax) calculated.

Results:

For the chosen operating point, the macro gives us the gamma, as well as the current giving the MTPA. For example, for a speed of 3,000 RPM and a desired torque of 150Nm, the macro finds a gamma angle of 35° and a current of 64.54 A. On the surface of the torque, according to gamma and current (figure 1), the gamma value giving MTPA is around 35° for a current of 65A.

A second advantage of this second method is the possibility of piloting Flux from the Microsoft software Excel® using VBA (cf. article CN59 p10).

(continued on page 5)

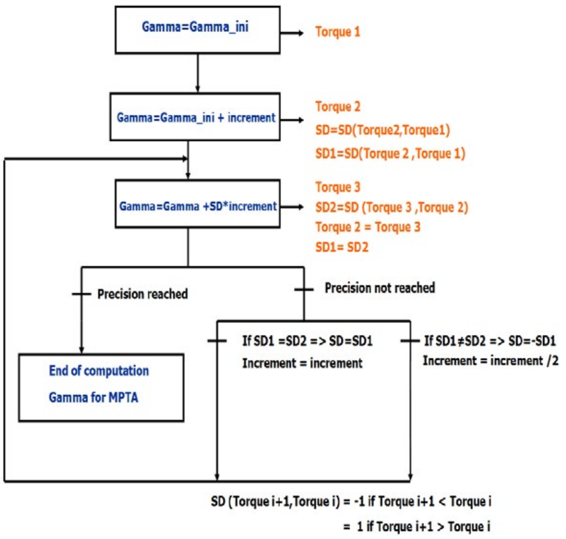

Figure 3: Synoptic use for the search for optimal gamma

This option allows us to make the gamma calculation automatic, given the MTPA for several operating points, and to create our own simplified interface under Excel®.

```
**** Summarize of Values    ****

      convergence : OK
 Objective torque : 150.0
    Result torque : 150.63315297293667
            speed : 3000.0
          current : 64.5468413321428
            gamma : 35.0

****   Ending FindOutGammaForGivenTorqueSpeed   ****
```

Fig. 4: Display of gamma and Imax found by the macro under Flux.

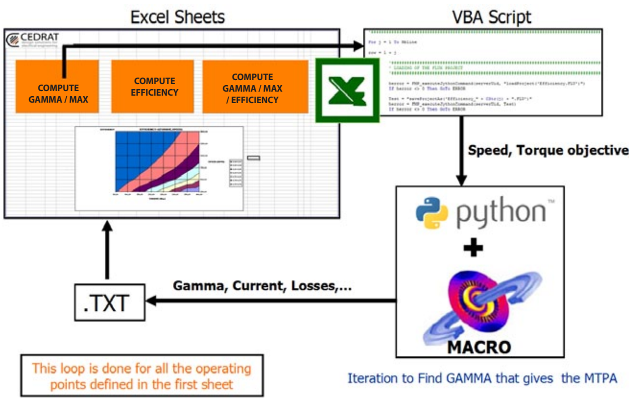

Fig. 5: Synoptic of automation of Flux calculations.

Thanks to piloting of Flux by Excel, gamma is calculated automatically for several operating points, which are defined beforehand by the user under Excel (for a speed of 3,000 RPM). Using the macro, Flux performs only 67 calculations to calculate the gamma for 7 operating points (cf. figure 6). The results obtained with the macro are coherent with the torque surface according to gamma and Imax (figure 1).

Torque (Nm)	50	100	150	200	250	300	350
Gamma compute (°)	20	27	35	40	40	45	47
Courant (A)	21.80	42.91	64.54	87.21	111.66	138.60	168.55
Number of iterations	11	10	9	9	9	9	10

Fig. 6: Gamma and Imax found using the macro.

3. Flux optimization project with GOT-It

The **third method** used involves our optimization tool, GOT-It. In this method, the Flux® project is identical to the project used for the multiparametrics calculation (cf. figure 2). The coupling component Flux - GOT-It is generated under Flux. We define as component inputs speed, gamma and current. The component outputs are torque and voltage. Under GOT-It, the defined piloting parameters are gamma and current, whilst the objective defined is the desired torque. We also add an objective to minimise input current. The optimization algorithm used is the SSO algorithm.

Results:

For a target torque of 50Nm, the current and gamma values found by GOT-IT are as follows (figure 7).

Index	GAMMA	IMAX	OBJ_TORQUE	OBJ_IMAX	CTRT_VOLTAGE	V_OBTAIN	EFFICIENCY	COUPLE_OBJECTIF
0	20	21.8	0.0321933420547964	0.109	-0.2561566396017667	353.3255961891608	0.978986819795967	50
1	67.7	37						
2	73.9	41.4						
3	90	0.6						

Fig. 7: Gamma and Imax found using GOT-It for a target torque of 50Nm and a speed of 3,000 RPM.

The results obtained for the different operating points (N=3,000 RPM) are provided in the following table:

Torque (Nm)	50	100	150	200	250	300	350
Gamma compute (°)	20	24	34	32.3	37.5	44.5	47.6
Courant (A)	21.8	41.8	63	86	110.8	136.6	166.6

Fig. 8: Gamma and Imax found using GOT-It.

The solution using GOT-It also makes calculations automatic, and indeed, GOT-It projects can be piloted by Python. The creation of a .bat file allowing the launch of the Python file, allows the value of the target torque sought to be modified by GOT-It, and therefore to make the search for gamma, given the MTPA for several operating points, to be made automatic.

Conclusion - Comparison of the three methods

This article presents three methods that can be implemented to calculate gamma, to obtain maximum torque per amp (MTPA). Each of the three methods presented above gives very similar results. However, the ease of implementation and number of calculations necessary to find the optimum will vary according to the method used:

- **Parametric calculation under Flux** is the easiest method to implement to find optimum gamma. However, it requires a large number of calculations for accuracy. In our example, 441 transitory Flux calculations were necessary to plot the graph in figure 1.
- **The second method, using the Flux macro**, requires knowledge of Python language, but it substantially reduces the number of Flux calculations needed to find the optimum gamma. It took sixty-one transitory Flux calculations to find the optimum gamma for seven operating points.
- **The third method, using GOT-It optimisation**, is simple to use, thanks to the Flux/GOT-It combination available. By choosing the right targets and constraints, it is easy for GOT-It to find the optimum gamma, whilst minimising the number of Flux calculations, thanks to substitute functions.

We've also seen in this article the possibilities of making a series of calculations automatic by using Flux driving with Excel, or indeed the option of running GOT-It using Python language and a .bat file.