

# ALTAIR

## Altair<sup>®</sup> FluxMotor<sup>®</sup> 2024

Synchronous Machines

Motor Factory – Test – Best Practices

General user information

[Altairhyperworks.com](https://www.altairhyperworks.com)

Contents

**1 Performance mapping – Sine wave – Motor – Efficiency map -----3**

**1.1 Advanced Inputs and their impacts on the test performance ----- 3**

1.1.1 Positioning and objective -----3

1.1.2 General guidelines for a balance between accuracy and computation time-----3

1.1.3 Examples demonstrating the impact of advanced inputs -----6

# 1 PERFORMANCE MAPPING – SINE WAVE – MOTOR – EFFICIENCY MAP

## 1.1 Advanced Inputs and their impacts on the test performance

### 1.1.1 Positioning and objective

The aim of this section is to provide a guide on how to set up the test with inputs providing results appropriate to one’s needs in terms of accuracy and computing time for the efficiency map test of synchronous machines including SMPM, RSM and SMWF. The following inputs, with important impact on the accuracy and computing time of the test, will be discussed in the section:

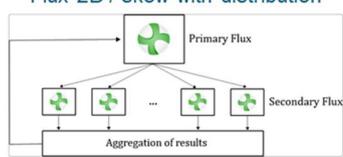
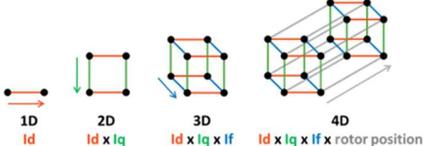
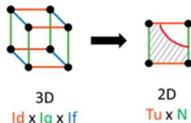
- Number of computations per electrical period
- Number of computations for  $J_d$ ,  $J_q$
- Number of computations for  $I_f$ -axis field current
- Number of computations for speed
- Number of computations for torque

### 1.1.2 General guidelines for a balance between accuracy and computation time

The efficiency map test happens in three phases that are listed below:

- FE project creation
- FE simulation for magnetic data
- Optimization in torque – speed space

Here below, the process is summarized for a quick understanding of how it is carried out and which factors can influence the overall performance in terms of accuracy and computing time. In general, the machine performance needs to be evaluated within its operational range before looking for optimal configuration of control variables for each torque-speed working point.

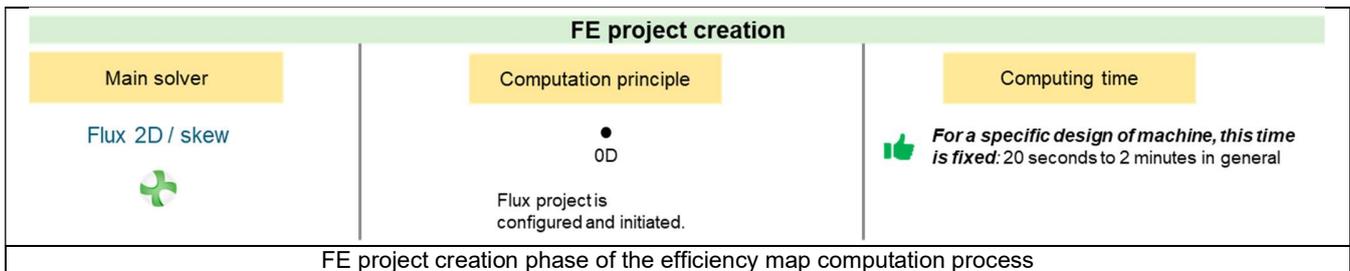
	FE project creation	FE simulation for magnetic data	Optimization in torque – speed space
Main solver	Flux 2D / skew 	Flux 2D / skew with distribution 	FluxMotor internal process 
Computation principle	0D Flux project is configured and initiated.	 <p>A FE simulation is conducted, where the number of steps equals the product of elements along each dimension. Rotor position dependency:</p> <ul style="list-style-type: none"> <li>• No → 1 rotor position, rotor iron losses are neglected</li> <li>• Yes → multi rotor position → more accurate torque and iron losses in both stator and rotor</li> </ul>	! All magnetic terms are averaged along the dimension of rotor position  <p>The optimization process to identify the configuration of control variables (<math>I_d</math>, <math>I_q</math>, <math>I_f</math>) yielding the optimal performance within the useful torque-speed space.</p>
Computing time	 <p><b>For a specific design of machine, this time is fixed:</b> 20 seconds to 2 minutes in general</p>	 <p><b>More parallel Flux instances → less computing time:</b> 5 Flux in parallel → computing time down by around 5x</p>  <p><b>More computation steps → more computing time:</b> 2x computations along all 3 current axes → computing time up by around 2<sup>3</sup> or 8x</p>	 <p><b>More <math>I_d</math>, <math>I_q</math>, and <math>I_f</math> computation steps → more computing time:</b> 2x computations along 3 current axes → computing time up by maximum 2<sup>3</sup> or 8x</p>  <p><b>More useful torque-speed computation steps → more computing time:</b> 2x computations along torque and speed axes → computing time up by maximum 2<sup>2</sup> or 4x</p>
Summary of the efficiency map computation process with SMWF as example			

### 1) FE project creation

In this step, the computing process is set up using inputs and settings provided by the user. The Flux 2D/Skew project is configured and initiated with the appropriate scenario for execution in the subsequent phase. The distribution mode of Flux has no utility at this early stage.

The only factor impacting the computing time of this step is the design of the machine itself. Factors such as whether it is skewed, the number of poles/slots being simulated, the number of nodes utilized to depict the machine in Flux 2D/Skew, and whether conductors in slots are accounted for in Flux 2D/Skew, all play a role.

**Note:** As a general guideline, in the efficiency map, since conductors are not explicitly represented in Flux 2D/Skew, this step typically consumes a relatively short amount of time within the overall computing process, typically ranging from 20 seconds to a minute.



### 2) FE simulation for magnetic data

The Flux 2D/Skew project, previously prepared to analyze the electromagnetic performance of the machine within its defined operational range, is launched.

It's important to understand that the machine's operational range is determined by the currents supplied to it. For SMPM and RSM machines, this includes the d-axis  $J_d$  and q-axis  $J_q$  currents, and for SMWF machines, it also includes the excitation current  $I_f$ . During this phase, the supplied current values are varied within their defined ranges to evaluate the machine's electromagnetic performance across all points.

When the rotor position dependency option is enabled, an additional dimension is introduced to the evaluation of each machine. Consequently:

- For SMPM and RSM machines: Performance evaluation occurs in a 2-dimensional characteristic space of  $J_d$  and  $J_q$  when rotor position dependency is disabled, and in a 3-dimensional characteristic space of  $J_d$ ,  $J_q$ , and rotor position when enabled.
- For SMWF machines: Performance evaluation occurs in a 3-dimensional characteristic space of  $J_d$ ,  $J_q$ , and  $I_f$  when rotor position dependency is disabled, and in a 4-dimensional characteristic space of  $J_d$ ,  $J_q$ ,  $I_f$ , and rotor position when enabled.

**Note:** Increased level of details in the characteristic space requires more computations, resulting in more detailed performances, as well as longer computing times.

For example:

- If the number of computations for  $J_d$  and  $J_q$  doubles for a test with SMPM, the computation time for this phase may quadruple.
- Similarly, in the case of SMWF, doubling the number of computations for  $J_d$ ,  $J_q$ , and  $I_f$  can increase computation time by eightfold.

Additionally, when rotor position dependency is considered, doubling the number of points per electrical period in a test can double the computing time required.

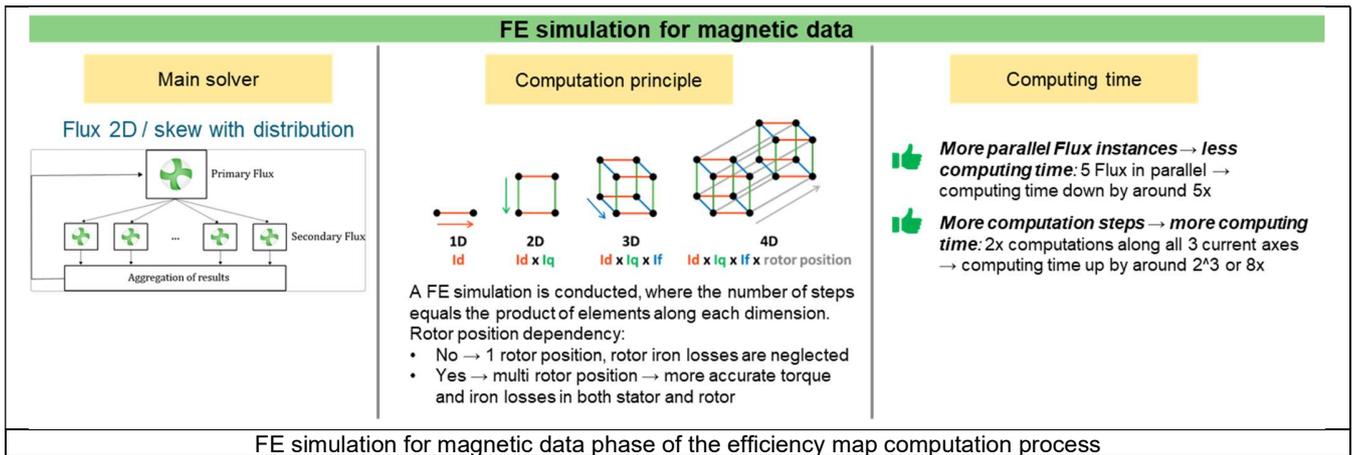
**Note:** While rotor position dependency enhances the accuracy of estimating electromagnetic torque and iron losses, it introduces a higher computational cost due to the added dimension in the characteristic space.

However, this increased cost can be compensated by:

- Using the half period option for the number of computed electric periods inputs to reduce the number of rotor positions by nearly a half (half of the number of computations per electric periods plus 1 point).
- Using the distribution mode of Flux, where computing time decreases proportionally with the number of parallel Flux instances employed. For example, running 5 Flux instances concurrently can reduce computing time by up to 5 times.

**Note:** The effectiveness of the distribution mode depends on the computer's configuration. For further details regarding distribution mode, performance, and licensing, please refer to the FluxMotor Supervisor document.

**Note:** The design of the machine also has a huge impact on the computing time of this step because the more nodes that the FE project uses to describe the machine, the more time and resources are needed to complete the FE simulation.



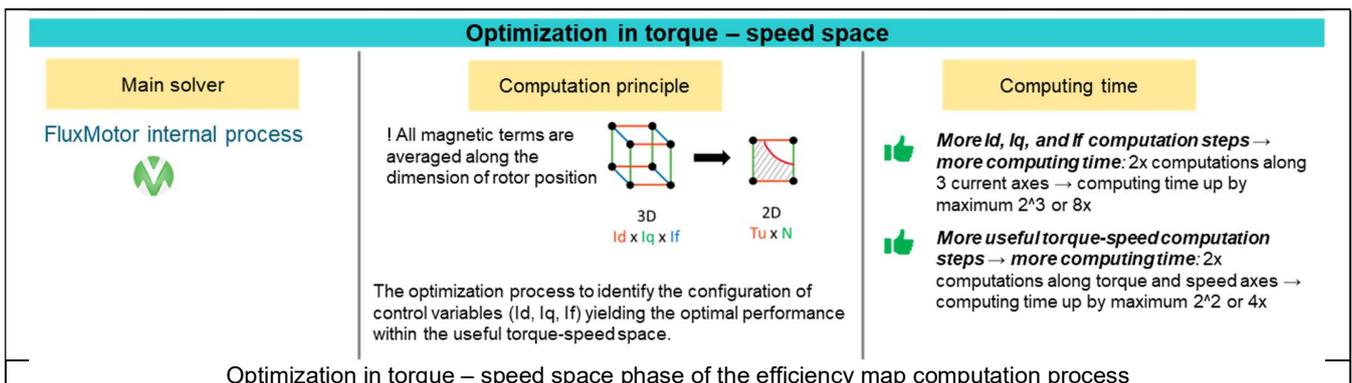
### 3) Optimization in torque – speed space

The machine's performance within its operational range was extensively assessed in the previous step, and the acquired data is utilized in an optimization process to identify optimal control variables. For detailed information on the optimization process, please refer to the section on command modes.

**Note:** This optimization step is executed in the backend of FluxMotor, with computing time unaffected by the machine's design. However, two factors can influence the computing time at this stage:

- The number of computations for  $J_d$ ,  $J_q$ , and  $I_f$  (applicable to SMWF machines): Since the volume of data processed in the optimization process is directly proportional to the number of elements in the machine's characteristic space, doubling the number of elements along each axis ( $J_d$ ,  $J_q$ , and  $I_f$ ) can result in an eightfold increase in computing time for SMWF and a fourfold increase for SMPM. For sure, this action brings more accurate data for the optimization process.
- The number of computations for torque and speed: Increasing the number of computations for these parameters entails optimizing more torque-speed working points. However, since the optimization process at each point begins with values from the closest point, a higher level of detail in the torque-speed space facilitates quicker convergence at each point. While increasing the number of computations for torque and speed generally lengthens computing time in this final step, the relationship is not strictly proportional.

**Note:** The choice for the rotor position dependency does not impact the computing time of this step because all electromagnetic data is averaged along the rotor position axis if the rotor position dependency is enabled. The organization of data with or without rotor position dependency is the same for the optimization step. Only the accuracy of the estimation of electromagnetic torque and iron losses are impacted.



### 1.1.3 Examples demonstrating the impact of advanced inputs

Note: All results presented in the following examples were conducted using a laptop featuring the following configuration: Core i7 Gen 12 2.10 GHz processor and 64 GB of RAM. It's important to highlight that during the testing process, all intensive tasks were halted to ensure that computing time measurements were not influenced.

However, it's crucial to understand that computing time can vary significantly depending on the setup of the computer used for the analysis. Factors such as hardware specifications, background processes, and system configuration can all impact computing performance and, consequently, computing time.

Therefore, while the results of this benchmark study provide valuable insights into computing time under specific conditions, users should be aware that actual computing times may differ based on their individual computer setups.

#### 1) Accuracy versus computing time in case of a SMPM

In this example, the efficiency map of a SMPM is analyzed under various configurations of input parameters along the current axis (Jd, Jq) and rotor position axis within the characteristic space. The goal is to illustrate how these input configurations influence the accuracy and computational time of the efficiency test.

Let's consider a SMPM with the following specifications:

- 8 poles
- 48 slots
- 4000 nodes represented in the FE project of Flux 2D.

In this instance, the distribution mode of Flux is activated with 5 Flux instances running in parallel.

Additionally, for computations involving rotor position dependency, the half period computation mode is selected.

And for the final torque – speed space, each axis is discretized by 30 elements so that the interpolation used for the comparison process can be evaluated with the most accurate.

Six test configurations are compared, each denoted by the rule: MSX-Y-Z-W, where:

- X indicates the presence of rotor position dependency (1 for No, 2 for Yes).
- Y represents the number of computations along the Jd axis.
- Z represents the number of computations along the Jq axis (Y equals Z).
- W denotes the number of rotor positions (W equals 1 if X equals 1 or if there is no rotor position dependency).

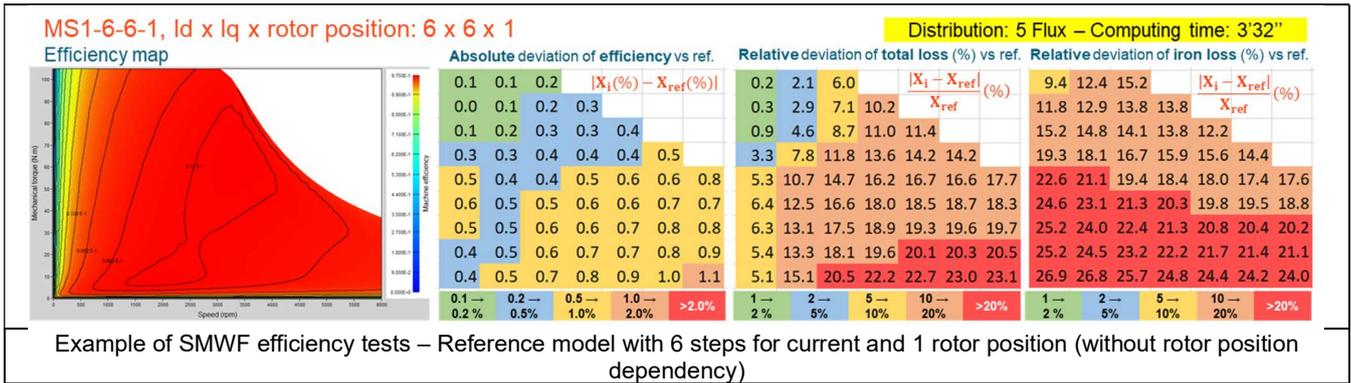
The six configurations are as follows:

- MS2-10-10-100: This serves as the reference for comparison, offering high-resolution data in the characteristic space, ensuring highly reliable computation of iron losses, total losses, and efficiency.
- MS2-10-10-40: To observe the impact of reducing the number of rotor positions.
- MS2-6-6-40: To observe the impact of reducing elements along the current axis.
- MS2-6-6-30 and MS2-6-6-20: To explore the effect of decreasing the number of rotor positions to its lower limit.
- MS1-6-6-1: To observe the impact of neglecting rotor position dependency.

Each model is compared to the reference model, analyzing efficiency, total losses, and iron losses within the map according to the following criteria:

- Efficiency: Deviation calculated as the absolute difference between the efficiency percentages of the actual and reference models.
- Total losses: Deviation calculated as the absolute difference between the total losses of the actual and reference models, divided by the total losses of the reference model, and expressed as a percentage.
- Iron losses: Deviation calculated as the absolute difference between the iron losses of the actual and reference models, divided by the iron losses of the reference model, and expressed as a percentage.





Observations:

- Computation times range from 3 minutes to approximately 30 minutes, progressively decreasing from the first to the sixth configuration as the number of computations decreases.
- All models yield nearly identical efficiency maps, with deviations consistently minimal (less than 1% of efficiency) as evident in the second figure of each model. However, deviations increase from configuration 2 to 6 due to reductions in elements along current axes and the number of rotor positions.
- Closer examination of total losses and iron losses reveals the impact of the number of rotor positions considered in the computation. Fewer positions lead to greater deviation compared to the reference. For instance, with only 20 positions, iron losses deviation can reach up to 20% in most areas, whereas with 40 positions, accuracy improves significantly.
- Utilizing six elements along current axes and 40 rotor positions appears to reach a balance between accuracy and computation time.
- Models without rotor position dependency exhibit poor estimations of iron losses **in this particular machine**. However, if only the efficiency map is of concern, this limitation may be acceptable.

2) Recommended inputs for SMPM and RSM with distribution mode disabled

Below is an example for SMPM. Let's consider a SMPM with the following specifications:

- 8 poles
- 48 slots
- 4000 nodes represented in the FE project of Flux 2D.

In this instance, the distribution mode of Flux is deactivated, so only 1 Flux instance is run.

Additionally, for computations involving rotor position dependency, the half period computation mode is selected.

Thus, the number of rotor positions per electric period varies as follows:

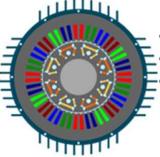
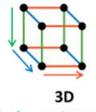
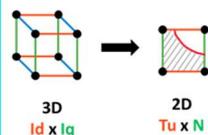
- For 21, 7, or 26 rotor positions, there are respectively 40, 13, and 50 positions per electric period.

For each configuration of rotor position dependency (Yes/No), three sets of inputs are provided, each with different priorities for accuracy and computing time:

- The recommended setting offers highly accurate results within an acceptable computing time, taking up to 2 minutes without rotor position dependency and 20 minutes with it.
- The basic setting provides preliminary results indicating performance trends with acceptable accuracy in the shortest computing time, taking up to 2 minutes without rotor position dependency and 8 minutes with it.
- The advanced setting delivers very accurate results but requires significant computing time, taking up to 5 minutes without rotor position dependency and 75 minutes with it. Memory issues may arise as finer characteristic space requires more memory resources. Refer to FluxMotor Supervisor help for guide on how to setup memory configuration of FluxMotor.

**Note:** As a general guideline, it's recommended to perform 6 computations along each current axis and have 40 rotor positions per electric period when the rotor position dependency mode is enabled. With the half period computation mode, only 21 points are computed.

**Note:** See the next example to understand the power of distribution mode in reducing computing time.

FE project creation		+	FE simulation for magnetic data		+	Optimization in torque – speed space		=	Total
 <p>8 poles 48 slots 4k nodes in Flux 2D</p> <p>0.3 mins</p>			 <p>3D ld x lq x rotor position</p> <p>Distribution Mode: <b>No</b> 1 Flux</p>			 <p>3D ld x lq</p> <p>2D Tu x N</p>			
Rotor position dependency: <b>NO</b>	<b>Recommended</b>	6 x 6 x 1	→ 1.4 mins	6 x 6	⇒ 15 x 7	→ 0.4 mins	<b>2.1 mins</b>		
	Basic	5 x 5 x 1	→ 1.1 mins	5 x 5	⇒ 10 x 10	→ 0.3 mins	1.7 mins		
	Advanced	10 x 10 x 1	→ 2.9 mins	10 x 10	⇒ 20 x 20	→ 1.8 mins	5.0 mins		
Rotor position dependency: <b>YES</b>	<b>Recommended</b>	6 x 6 x 40	→ 19.4 mins	6 x 6	⇒ 15 x 7	→ 0.4 mins	<b>20.1 mins</b>		
	Basic	5 x 5 x 13	→ 6.6 mins	5 x 5	⇒ 10 x 10	→ 0.3 mins	7.2 mins		
	Advanced	10 x 10 x 50	→ 74.3 mins	10 x 10	⇒ 20 x 20	→ 2.1 mins	76.5 mins		

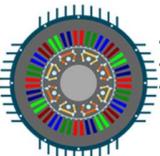
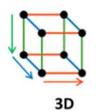
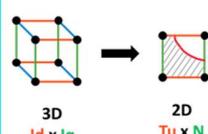
**Note:** Half electric period mode is recommended to divide the total number of rotor positions by almost twice as following

No. computed elec. periods: 1/2  
 No. comp. / elec. period: 40

$\frac{1}{2} \times \text{No. comp. per elec. period} + 1$

PC configuration: Core i7 Gen 12 2.10 GHz, 64 GB of RAM

Recommended inputs for SMPM efficiency map tests with distribution mode disabled

FE project creation		+	FE simulation for magnetic data		+	Optimization in torque – speed space		=	Total
 <p>8 poles 48 slots 4k nodes in Flux 2D</p> <p>0.3 mins</p>			 <p>3D ld x lq x rotor position</p> <p>Distribution Mode: <b>Yes</b> 5 Flux in parallel</p>			 <p>3D ld x lq</p> <p>2D Tu x N</p>			
Rotor position dependency: <b>NO</b>	<b>Recommended</b>	6 x 6 x 1	→ 1.1 mins	6 x 6	⇒ 15 x 7	→ 0.4 mins	<b>1.8 mins</b>		
	Basic	5 x 5 x 1	→ 1.0 mins	5 x 5	⇒ 10 x 10	→ 0.3 mins	1.5 mins		
	Advanced	10 x 10 x 1	→ 1.5 mins	10 x 10	⇒ 20 x 20	→ 1.8 mins	3.6 mins		
Rotor position dependency: <b>YES</b>	<b>Recommended</b>	6 x 6 x 40	→ 3.7 mins	6 x 6	⇒ 15 x 7	→ 0.4 mins	<b>4.4 mins</b>		
	Basic	5 x 5 x 13	→ 1.8 mins	5 x 5	⇒ 10 x 10	→ 0.3 mins	2.4 mins		
	Advanced	10 x 10 x 50	→ 10.1 mins	10 x 10	⇒ 20 x 20	→ 1.8 mins	12.2 mins		

**Note:** Half electric period mode is recommended to divide the total number of rotor positions by almost twice as following

No. computed elec. periods: 1/2  
 No. comp. / elec. period: 40

$\frac{1}{2} \times \text{No. comp. per elec. period} + 1$

PC configuration: Core i7 Gen 12 2.10 GHz, 64 GB of RAM

Recommended inputs for SMPM efficiency map tests with distribution mode enabled

3) Recommended inputs for SMWF with distribution mode enabled

Below is an example for SMWF. Let's consider a SMWM with the following specifications:

- 4 poles
- 48 slots
- 12000 nodes represented in the FE project of Flux 2D.

In this instance, the distribution mode of Flux is activated with 5 Flux instances running in parallel.

Additionally, for computations involving rotor position dependency, the half period computation mode is selected.

Thus, the number of rotor positions per electric period varies as follows:

- For 21, 7, or 26 rotor positions, there are respectively 40, 13, and 50 positions per electric period.

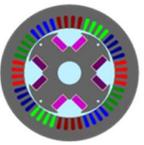
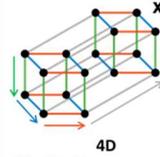
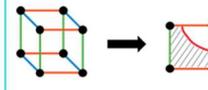
And because the SMWF benefits from the advantages of both the SMPM and RSM, the 1st and 2nd quadrants are selected in the backend for the FE simulation step. Therefore, the number of steps for  $I_d$  varies as follows:

- For 11, 9, or 19  $I_d$  values, there are respectively 6, 5, and 10  $I_d$  values per quadrant.

For each configuration of rotor position dependency (Yes/No), three sets of inputs are provided, each with different priorities for accuracy and computing time:

- The recommended setting offers highly accurate results within an acceptable computing time, taking up to 10 minutes without rotor position dependency and 80 minutes with it.
- The basic setting provides preliminary results indicating performance trends with acceptable accuracy in the shortest computing time, taking up to 5 minutes without rotor position dependency and 20 minutes with it.
- The advanced setting delivers very accurate results but requires significant computing time, taking up to 60 minutes without rotor position dependency and 480 minutes with it. Memory issues may arise as finer characteristic space requires more memory resources. Refer to FluxMotor Supervisor help for guide on how to setup memory configuration of FluxMotor.

**Note:** As a general guideline, it's recommended to perform 6 computations along each current axis and have 40 rotor positions per electric period when the rotor position dependency mode is enabled. With the half period computation mode, only 21 points are computed.

FE project creation		+	FE simulation for magnetic data		+	Optimization in torque – speed space		=	Total	
 <ul style="list-style-type: none"> <li>• 4 poles</li> <li>• 48 slots</li> <li>• 12k nodes in Flux 2D</li> </ul> <p><b>0.4 mins</b></p>			 <p><b>Distribution Mode: Yes</b> <b>5 Flux in parallel</b></p> <p>4D <math>I_d \times I_q \times I_f \times \text{rotor position}</math></p>			 <p>3D <math>I_d \times I_q \times I_f</math></p> <p>2D <math>T_u \times N</math></p>				
Rotor position dependency: <b>NO</b>	<b>Recommended</b>	$6 \times 6 \times 6 \times 1$	→ 5.9 mins		$6 \times 6 \times 6$	⇒ $15 \times 7$	→ 2.6 mins		<b>8.9 mins</b>	
	Basic	$5 \times 5 \times 5 \times 1$	→ 3.2 mins		$5 \times 5 \times 5$	⇒ $10 \times 10$	→ 1.2 mins		4.8 mins	
	Advanced	$10 \times 10 \times 10 \times 1$	→ 20.6 mins		$10 \times 10 \times 10$	⇒ $20 \times 20$	→ 35 mins		56.0 mins	
Rotor position dependency: <b>YES</b>	<b>Recommended</b>	$6 \times 6 \times 6 \times 40$	→ 74.2 mins		$6 \times 6 \times 6$	⇒ $15 \times 7$	→ 3.2 mins		<b>77.8 mins</b>	
	Basic	$5 \times 5 \times 5 \times 13$	→ 15.8 mins		$5 \times 5 \times 5$	⇒ $10 \times 10$	→ 1.6 mins		17.8 mins	
	Advanced	$10 \times 10 \times 10 \times 50$	→ 445.2 mins		$10 \times 10 \times 10$	⇒ $20 \times 20$	→ 35.0 mins		480.6 mins	

**Note 1:** Half electric period mode is recommended to divide the total number of rotor positions by almost twice as following  $\frac{1}{2} \times \text{No. comp. per elec. period} + 1$

**Note 2:** Quadrants 1<sup>st</sup> and 2<sup>nd</sup> are used for the FE simulation step, therefore the number of steps for  $I_d$  is provided as following  $2 \times \text{No. comp. for } I_d, I_q - 1$

PC configuration: Core i7 Gen 12 2.10 GHz, 64 GB of RAM

Recommended inputs for SMWF efficiency map tests with distribution mode enabled