### Flux Activate coupling Induction Motor Scalar Control

2D Multiphysics Summary

### Introduction

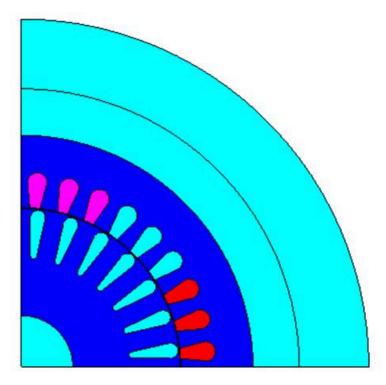
This document describes Flux-Activate coupling on the example of induction motor.

### Keywords

Applications	Flux main functions
Transient Magnetic	Co-Simulation Induction motor

### **Studied device**

Our sample problem consists of a 4-pole, 3-phase, 36-slot, 28-bar induction motor. Because of the motor's periodicity, we will model only ¼ of it (1 pole). Our model consists of 9 stator slots and 7 rotor bars. The air-gap is set to 0.25 mm.





# Aim and steps

### Aim

The aim of this coupling is to do a cosimulation between Flux and Activate. Activate serves to make the system part.

Action	Software
Device modeling	<b></b> Altair Flux™
Generate the coupling component for Activate	<b></b> Altair Flux™
Create the Flux connector	METINATE CONTRACT OF CONTRACT.
Solve the problem	ACTIVATE

### Steps

The different steps to realize this coupling are shown in the table below:

Phase	Software	Description
1	Flux	<ul> <li>Flux project preparation:</li> <li>Model description: geometry, mesh and physics</li> <li>Specific description : creation of the input/output parameters required for the coupling</li> <li>Generate the coupling component for Activate</li> </ul>
2	Activate	Activate opening: • Preparation of the Activate model
3	Activate / Flux	Solve the problem
4	Activate / Flux	Post Processing



### Flux model

The first step is to prepare the Flux project:

- The geometry, mesh the device and the physics description (materials, regions, non-mesh coils, mechanical sets...) and circuit.
- We create the input and output parameter that we use to generate the coupling element between Flux and Activate
- Generate the coupling element

The motor is driven with a 2 phase's voltage source and running at coupled load. To do it we use the scalar control. The principle of this type of control is to adjust the motor speed V

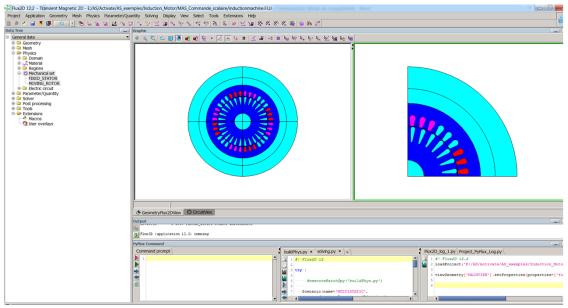
s by varying the output frequency  $f_s$  such as the magnetic state of the machine is about fixed, through the preservation of  $\Phi_s$  in a constant value, and such as the torque follows the wished law of variation according to the speed. This type of command does not allow to control the electric and magnetic transients, thus it must be operated only for laws of control with slow dynamics for which we can consider that the machine keeps on electric and magnetic steady state.

The simplicity of this type of control favors its implementation in many industrial variators conceived for these types of application. Indeed, it is characterized by a simplified structure, but requiring a speed sensor and a speed controller for position servo-control.

The simulated motor performances are used to compute shaft torque, current, speed and bar current.

### Flux model: detailed steps

In the first step we create the geometry, mesh and physics. To do it, we open the project: inductionmachine.FLU.



We create input and output parameters



Altair Flux<sup>™</sup> - 2D Example Summary: Flux Activate coupling Induction Motor Scalar Control

Flux model

**1.** Inputs:

Multi-Physical I / O Parameters: this allows, via formulas to pilot:

- the physical quantities:  $\mu$ r, Bs, Br etc.
- electrical quantities: resistance, voltage, current etc.
- mechanical quantities: position, speed, resistant force, resistive torque etc.
- **2.** Outputs:

Scalar I / O Settings: this allows, via sensors, formulas, parameters to retrieve the values

- of sensors
- of forces, couples etc.
- positions, speeds, accelerations etc.

Input parameters:

Control Contro	Bees-RES		2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2	
Entities	Modify all	V_A	V_B	V_C
B Physical parameter				
INAME *		V_A	V_B	V_C
() Comment	Initial values			
🗉 🔘 Sub types	Initial values	Parameter for multiphysical app	Parameter for multiphysical app	Parameter for multiphys
Parameter for multiphysical application		Parameter for multiphysical app	Parameter for multiphysical app	Parameter for multiphys
B Reference value *	Initial values	1.0	1.0	1.0
OK Apply Cancel				

#### Output parameters:

New Physical parameter
Name of the Physical parameter *
Current_phase1
Comment
Type of Physical parameter
Parameter defined by a formula
Expression *
I(B1) f()
OK Cancel 🚳



Altair Flux<sup>™</sup> - 2D Example Summary: Flux Activate coupling Induction Motor Scalar Control Flux model

New Physical parameter
Name of the Physical parameter *
Current_phase2
Comment
Type of Physical parameter
Parameter defined by a formula
Expression *
I(B2) f()
😥 OK Cancel 🚳

New Physical parameter
Name of the Physical parameter *
Current_phase3
Comment
Type of Physical parameter
Parameter defined by a formula
Expression *
I(B3) f()
OK Cancel



p.5

Altair Flux<sup>™</sup> - 2D Example Summary: Flux Activate coupling Induction Motor Scalar Control Flux model

New Physical parameter
Name of the Physical parameter *
Torque
Comment
Type of Physical parameter
Parameter defined by a formula
Expression *
TorqueElecMag(MOVING_ROTOR) f()
OK Cancel 🚳

New Physical parameter
Name of the Physical parameter *
I_bar_1
Comment
Type of Physical parameter
Parameter defined by a formula
Expression *
I(BAR_1_Q1) f()
OK Cancel



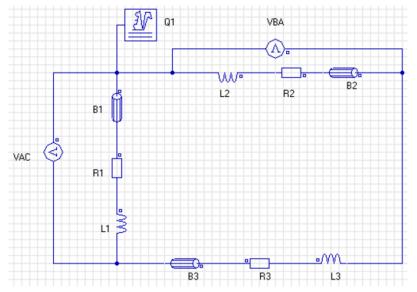
p.6

Altair Flux<sup>™</sup> - 2D Example Summary: Flux Activate coupling Induction Motor Scalar Control Flux model

New Physical parameter
Name of the Physical parameter *
I_bar_2
Comment
Type of Physical parameter
Parameter defined by a formula 👻
Expression *
I(BAR_2_Q1) f()
OK Cancel 🚳

### **Electrical circuit**

The electrical circuit is defined as:



It is composed by:

- Voltage source
- Coil conductor
- Resistor
- Inductance



In the source current we put the input parameter (I1, I2 and I3).

Generate component (solving >> generate component for Activate coupling) for Activate coupling in the same working directory.

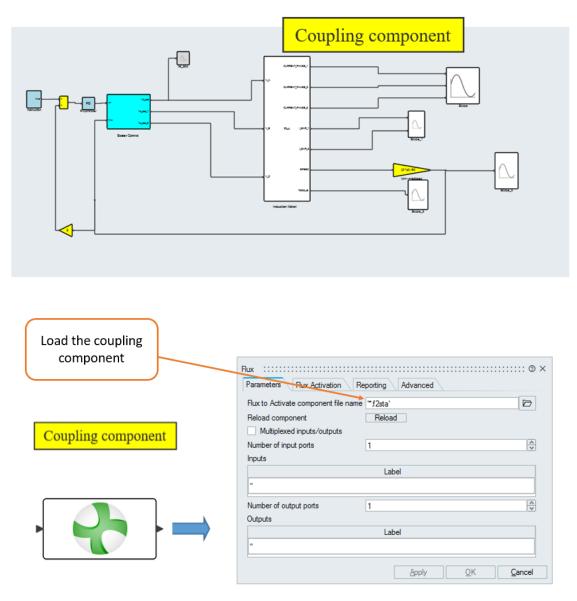
Component name * MAS_Activate Overwrite component if exist yes no Directory to save component	Name
Component input	Working directory
V A V B V C	Inputs of the component: Geometric parameters I / O parameters (scenario)
Component output CURRENT PHASE 1 CURRENT_PHASE_2 CURRENT_PHASE_3 I_BAR_1 I_BAR_2 SPEED TORQUE OK Cancel	Outputs of the component Geometric parameters I / O parameters (formula) Sensors





## Activate model

The user must prepare the Activate model by adding and characterizing the coupling block but also the blocks needed to build the desired model.



The choice of solver and the step:



### Altair Flux<sup>™</sup> - 2D Example Summary: Flux Activate coupling Induction Motor Scalar Activate model

Simulation Para	meters Zero-Crossing Solvers
Initial time	0.0
Final time	1
Real time scaling	0
Tolerance on time	auto
Activate profiling	J
	<u>O</u> K

Simulation Parameters	
Simulation Time Z	ero-Crossing Solvers
Select a solver	Forward Euler
Algebraic solver	Ida
Absolute tolerance	0.000001
Relative tolerance	0.000001
Initial step size	auto
Minimum step size	auto
(Maximum) step size	1e-3
Jacobian method	analytical
	<u>к</u>

The model includes:

- The command, which is a scalar control
- The induction machine
- A filter to soften the reference values
- A PI regulator for machine
- The output to be displayed



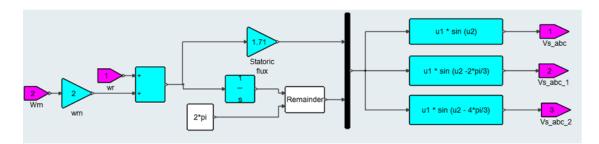


Figure 1: Command part

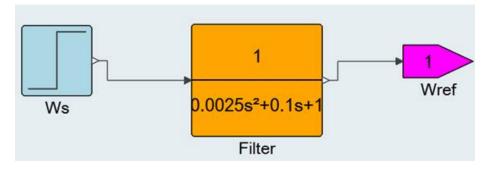


Figure 2: Subsystem of the instruction



p.11