

Regulate Current with User Subroutine Using Groovy Language in Flux[®]

Yassine Salhi ; Patrick Lombard - CEDRAT.

For Switched Reluctance Motor, a specific command of current is often used with a chopper in order to decrease the current ripple or hysteresis band. We propose to see in this example (see Figure 1) how to implement such a command in Flux using groovy language.

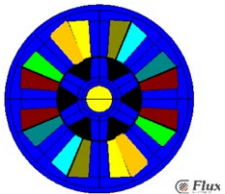


Figure 1: Geometry of the SRM.

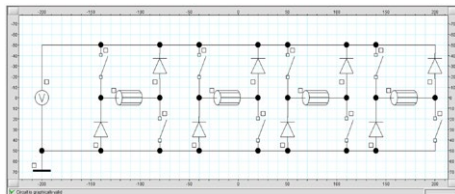


Figure 2: Associated circuit of the SRM.

It is a type of stepper motor: the movement of the rotor is get by alternately exciting pair of motor poles. We will see how to regulate current by using a user subroutine in groovy language in the associated circuit below. On our case the SRM is driven by a voltage supply of 12V and the speed is 500 rpm.

In the electric circuit, a specific command will be associated to the switches. The use of an I/O parameter defined by a user function written in groovy will allow introducing conditions. Starting from a predefined list of parameters (here current, maximum and minimum value of current, number of switch), the groovy function can define if the switch will be on or off.

The use of user subroutines can extend the scope of application of Flux, they provide the opportunity to define new physical properties based on settings chosen by user. In this case, we use it to customize I/O parameters. You can also use it to define specific material law (specific magnet orientation for instance).

We can find the groovy file in the folder persistent of the SRM example provided in the Flux 12 supervisor. To use it, we just have to paste it in the persistent of the current Flux project we use. The goal is to have a current between a minimum and maximum set as on the figure 3.

Without the user subroutine, currents are leading to very high level of saturations, which can be seen when we check the flux density (see figure 4).

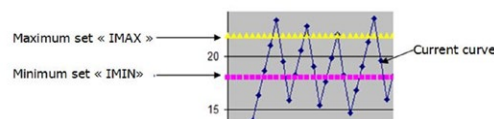


Figure 3: Current obtained with the regulation.

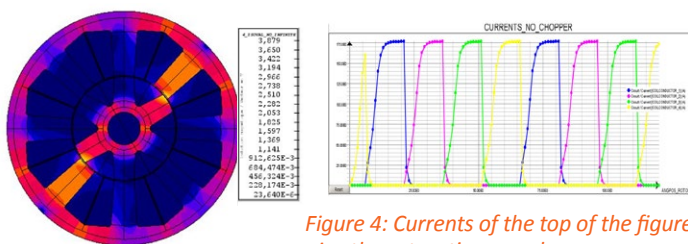


Figure 4: Currents of the top of the figure give the saturation seen here.

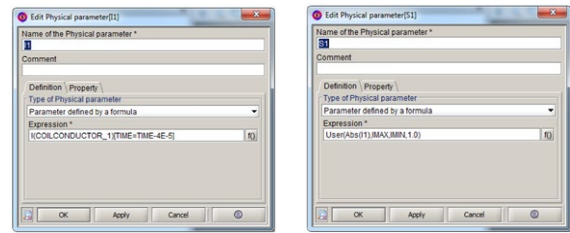


Figure 5: Parameters to regulate the current on the first coil.

To create the hysteresis band control, the user function needs 4 parameters:

- Ix is the current in the coil "x" to be regulated (value at previous time step)
- IMAX is the maximum limit
- IMIN is the minimum limit
- "x" is the number corresponding to the coil to be regulated.

The result of the user function will be 0 or 1 and is used to drive on or off state of switches.

Let's take I1 as example: the first picture of the figure 5 allows computing the derivative of the current, the second one is needed to keep the current between IMAX and IMIN: it is the control of I1 on the switch 1 between IMAX and IMIN. If we have for example a positive derivative of current (so the current is increasing) and a current value over "IMAX" the switch 1 is going to open itself in order to be under the upper set point. After that, if the derivative is negative (so the current is decreasing), and below IMIN, the switch is going to close itself in order to be above the lower set point.

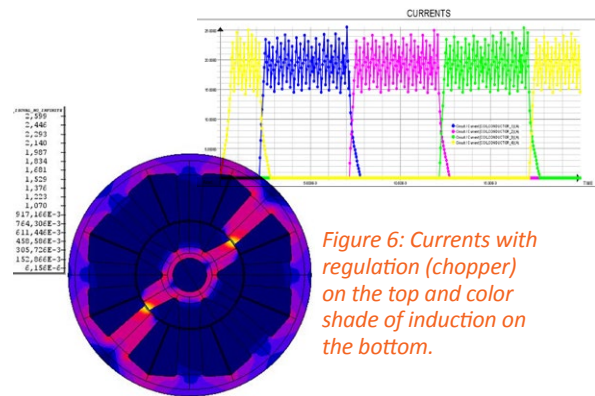


Figure 6: Currents with regulation (chopper) on the top and color shade of induction on the bottom.

After the creation of parameters for current regulation of each coil, we obtain the figure 6, on this case IMAX is 22A and IMIN 18A. Current can be out of maximum and minimum set because switches are turned off after the limit has been passed.

To reduce extrema values, you can decrease value of the time step. On the color shade of the bottom of figure 6, saturation is more in classical range of values.

Conclusion

In this paper, we have seen how to customize I/O parameters using Groovy language. You can have more information about the regulation of current using Groovy function on the example of switched reluctance motor available in the category "2D examples" from the supervisor of Flux 12.

It can also be used to customize hysteretic or non-hysteretic properties of materials, specific magnet orientation. Thus, we are able to model non standard phenomena. It is a very interesting tool which extends the possibilities of Flux.

A user subroutine can extend the scope of application of Flux, they provide the opportunity to define new physical properties based on settings chosen by user.