DYNAMIC BREAKING OF ELECTRICAL MACHINES

IMPLEMENTATION OF THE DCC USING FPGA

By
Aleksandar Stojkovic & Måns Andersson

Industrial electrical engineering and automation LTH Lund 2012

Summary

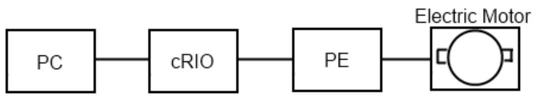
In a novel testing method called Dynamic breaking a new current controller is implemented to give better stability at high currents, high speed and sudden changes in current. The new current controller is a Direct current control, also called a DCC, which uses tolerance band to regulate current. The previously implemented current controller is a PI-controller. The DCC is a common analog current controller but in this project the DCC is implemented using an fpga, which allows high speed sampling and controlling.

Introduction

When implementing a basic DCC the current is controlled on one axis. In this case the current needs to be controlled in the x, y-frame. The reason is because both the magnetization (x-axis) and the speed (y-axis) of the motor have to be controlled. Since this DCC is implemented in a test-program the controller will have to be able to control high currents. When driving the motor with high current a big change in current can cause damage to the power electronics and in worst case even damage it. This makes it important to have good control on the current no matter what happens. The limitation of a DCC controller is often the switching frequency. In this project the goal has been to keep the switching frequency under 10 kHz, the reason for this limitation is the transistors. The measured data for the test-sequence has to be post-processed. This is done in Matlab, in this project the post-processing has been adapted to be able work with the new DCC.

Method

There are three different levels in the setup. The top level is the PC-part where the parameters are put. The cRIO is the control-unit with the FPGA. And at last the power electronics (PE), e.g. the inverter.



The chart for the implementation, where the fpga is a part of the cRIO. PE stands for Power Electronics

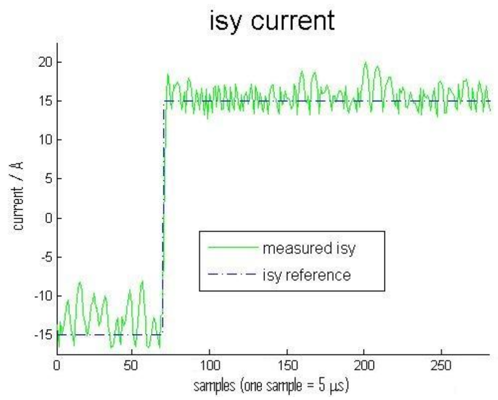
The DCC controller consists of two separate controllers running in parallel. One of the controllers will control the y-current and the other controlling the x-current. The action of both controllers is checked and by using a lookup-table an output is given. In this project two lookup-tables with different characteristics has been used to see how they will reduce the switching frequency. Both lookup-tables have the same basics. One of them is just an extended version of the other with more choices in different situations. The controller has eight different vectors that can be used to regulate the current, by selecting the values in the right way a desired value can be reached. The last step of the current controller is to make a blanking time for the transistors so that no short circuit will occur during switching.

To validate that the DCC is working it had to do a small test sequence. The same sequence was later done with the already implemented PI-controller so that some comparison can be made. The most important improvement had to be the sudden changes in current. Therefor the step-response of the two sequences will be compared. Also the general test will be compared between the two controllers. The implementation for the DCC has been done in National Instruments LabVIEW using the cRIO hardware with FPGA.

Results

The two different lookup-tables are working well with the implemented DCC. The aim was to find which of the two was better. But no one was significantly better than the other. Both of the lookup-tables had its pros and cons but the extended lookup-table had better characteristics when the motor was in standstill and for that reason it was used during the tests.

Implementation of the DCC worked well with the existing implementation. The DCC had better results in the step-response than the PI, where it reaches the reference faster and without any overshooting.

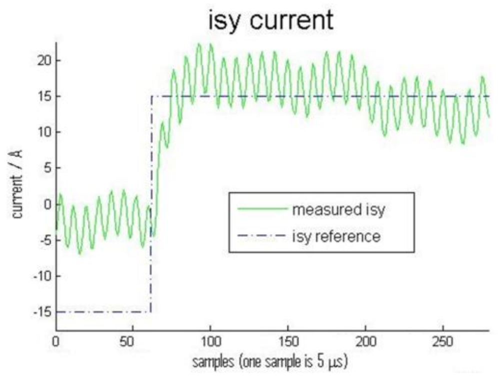


Step response of the DCC controller

Also the way it is going towards the reference is different. The DCC has a more direct way to the reference, instead of the more wobbling way that the PI-controller is doing it. The wobbling is because of the characteristics of the PI-controller e.g. the overshooting. The DCC has a more direct way of going to the reference since it is built to keep the current between the implemented tolerance bands.

Conclusion

The new current controller has shown positive results with better step-response and a good control of the current. It is a good replacement for the PI-controller. The current ripple for the DCC is similar, maybe even little better than the PI-controller and can most likely be improved. A big difference between the two controllers is the sound from the machine. The PI-controller has a constant sound, in our case 10 kHz, because it is switching with constant frequency. On the other hand the DCC controller sounds like a noise, because it switches whenever it wants the frequency is changed all the time. The noise from the DCC is louder than from the PI. This is something that could be bad for the DCC because the sound can be very annoying.



Step response of the PI-controller

If there is a lot noise in the background, the noise from the controller can blend in. The biggest advantage for the DCC is the fact that it can handle sudden changes better than the PI-controller which also was the most important improvement.