Introduction

Advanced control algorithms provide cutting edge performances despite their heavy computational load. Recently, engineers have tried to apply them on application fields requiring very demanding cycle times (as short as a few hundreds of microseconds) and multi-rate systems. On the other end, embedded processors have expanded their computing power by providing heterogeneous resources on a single chip by having multiple cores or even an FPGA.

Theory/Method/Hypothesis

The goal of this project was to assess the benefits, if there are any, of distributing the execution of complex real-life control solution on multiple cores. Distributing the computation on multiple cores will reduce the load on a single core, but only if the additional load from the runtime hasn't got too much overhead. Additionally, the distribution of computational load can also reduce the makespan if the overhead is again not too high.

The control applications used in this thesis all had three tasks running at different rates that communicate with each other. The figure below shows the top level view for the architecture that was developed for running these tasks.

Results

Three different methods were compared to assess which gives the best results. In a first step the code is run sequentially on a single core, then the different control tasks are run sequentially, but on different cores. Finally, all control tasks are also run in parallel on multiple cores.

While the load was distributed more evenly on both processors, the inter-core communication for running control tasks on multiple cores in parallel added too much overhead. The best results were obtained when the different control tasks were run sequentially on different cores.

Conclusion/Perspectives

The schedule for the parallel implementation was provided by another project [1] and was developed before knowing the test platform. For this reason we suppose that the schedule is not optimal and thus gives worse results. If work shall be continued on this project, the first step to take is to modify the scheduler to make it correspond to the test platform.

Acknowledgments

I would first like to express my sincere gratitude to Professor Beuchat René from the EPFL for supervising my master thesis. Furthermore, I would like to thank Wolfgang Reisner, Ashutosh Mishra and Alessandro Zanarini from ABB for supervising the work I did during the thesis.

References