Parallel optimization in Matlab

Daniel Axehill*

September 30, 2011

One of the most popular control strategies of today is model predictive control (MPC). This type of control can be applied to linear as well as non-linear systems. A special case of non-linear systems is hybrid systems which consist of an interaction between linear dynamics and logics. The fundamental idea in MPC is to solve an optimization problem on-line in each sample which depends on the measured or estimated state of the plant. For linear MPC, this optimization problem is either of linear programming (LP) or quadratic programming (QP) type. For hybrid systems, the corresponding optimization problems are either of mixed integer linear programming (MILP) or mixed integer quadratic programming (MIQP) type. It is well-known that the mixed integer problems are often significantly harder to solve and it is especially hard to bound the worst case computational time. To decrease the computational effort on-line and to be able to give hard real-time guarantees, so-called explicit MPC can be used. Explicit MPC can be viewed as if the optimization problem is solved off-line for all possible measured or estimated states. The on-line effort is reduced to looking up the precomputed and stored solution.

In summary, MPC is often computationally demanding. This is especially true if on-line MILP or MIQP solvers are used, or if the control law is computed off-line using a parametric optimization routine. Hence, it of great relevance to incorporate new ideas that can increase the computational performance in these areas.

In order to extend the applicability of MPC, we would like to investigate the possibilities from parallelism. In particular, we would like to investigate the possible gains of using multiple computational nodes communicating over a (fast) network when solving complicated optimization problems. The purpose with this project is therefore to develop an environment in Matlab that facilitates such experiments and to demonstrate the usefulness on a relevant and interesting control related optimization problem. The main objectives in this work are

1. Develop a basic parallel branch and bound optimization framework for Matlab.
    - Review which typical parallelization needs that exist in commonly parallelized algorithms like branch and bound.
    - Investigate if existing parallelization software for Matlab, like Parallel toolbox, is suitable for this task. Furthermore, alternatives should be investigated.

*Tel: 013-284042, E-mail: daniel@isy.liu.se

- Implement a parallel solver (either on-line or explicit) in Matlab.

2. Test the framework on computationally demanding optimization problems with control applications like on-line or explicit MPC.

The project can be tailored for the student's experience and interests. For example, a student which is very interested in programming might spend most of the time developing the parallel framework. For example, it might happen that better performance can be achieved if the existing Parallel toolbox is not involved. A more optimization interested student perhaps would like to investigate how to make use of the parallel capabilities from an optimization point of view. In both cases, a working basic parallel branch and bound code for Matlab should be the outcome. Furthermore, one or more challenging examples where the software is used for high performance optimization should be investigated and used as examples of the code.

To be able to smoothly solve this task it is preferable that the person has experience with Matlab and programming. Depending on the focus of the project, knowledge in network programming might be an advantage as well as knowledge in Java programming.