Embedded Model Predictive Control: Moving an Industrial PC-based MPC to an Embedded Platform

D. K. Kufoalor [*] **V. Aaker** [**] **T. A. Johansen** [*] **L. Imsland** [*]
G. O. Eikrem [***]

[*] *Department of Engineering Cybernetics, Norwegian University of Science and Technology, O.S. Bragstads plass 2D 7491 Trondheim, Norway. (e-mails: {kwame.kufoalor, tor.arne.johansen, lars.imsland}@itk.ntnu.no).*
[**] *Aker Solutions ASA, Oslo, Norway. (e-mail: vegardaak@gmail.com)*
[***] *FD ER SF, Statoil ASA, Arkitekt Ebbells veg 10, Rotvoll, Norway. (e-mail: gise@statoil.no)*

Abstract:
An embedded Model Predictive Controller (eMPC) based on the linear MPC module in SEPTIC (Statoil Estimation and Prediction Tool for Identification and Control) has been developed, and it is intended to facilitate the MPC application on an embedded controller. The control design approach illustrates a viable way of migrating from an industrial PC-based MPC technology to an embedded platform, and also covers the transformations and the real-time considerations necessary to achieve a functional high-performance predictive controller on a low cost embedded hardware, with limited computational resources.

The eMPC software design is based mainly on automatic code generation consisting of four main stages. The MPC application is designed at the first stage in SEPTIC, producing a configuration (config) file on which the eMPC design is based. At the second stage, a custom configuration file for the Quadratic Programming (QP) problem solver and a C code for the eMPC are generated based on the SEPTIC config file. The QP solver config file is used in a custom C code generator to produce the QP solver, which is then incorporated into the generated eMPC code in the third design stage. The third stage prepares the eMPC code with required target specific code, including code for communication and the embedded platform environment initialization and setup routines. Target specific memory management code and memory allocation for large data structures are also included at the third stage. The complete eMPC code is then compiled and linked using a compiler toolchain at the final stage.

The eMPC was implemented for rapid prototyping on the Ethernut 5 open source hardware. The concept was demonstrated using a real MPC-application developed for a prototype of a new industrial process that requires an advanced embedded multivariable, constraint handling control solution, and tested using a process simulator. The results confirm the viability of the embedded MPC design approach. The ultimate goal of this work is to enable the use of eMPC in challenging industrial applications, and also contribute to meet the need for MPC solutions on ultra-reliable industrial embedded computers. This presentation reports on a first step towards this objective, including developments and implementation of the eMPC concept on the ABB AC500 PM592-ETH industrial approved hardware.

Keywords: Embedded systems, Real-time control, Model Predictive Control.