Embedded Model Predictive Control on a PLC Using a Primal-Dual First-Order Method for a Subsea Separation Process*

D. K. M. Kufoalor[1], S. Richter[2], L. Imsland[1], T. A. Johansen[1], M. Morari[2], and G. O. Eikrem[3]

***Abstract*— The results of a PLC implementation of embedded Model Predictive Control (MPC) for an industrial problem are presented in this paper. The embedded MPC developed is based on the linear MPC module in SEPTIC (Statoil Estimation and Prediction Tool for Identification and Control), and it combines custom ANSI C code generation with problem size reduction methods, embedded real-time considerations, and a primal-dual first-order method that provides a *fast* and *light* QP solver obtained from the FiOrdOs code generator toolbox. Since the primal-dual first-order method proposed in this paper is new in the control community, an extensive comparison study with other state-of-the-art first-order methods is conducted to underline its potential. The embedded MPC was implemented on the ABB AC500 PLC, and its performance was tested using hardware-in-the-loop simulation of Statoil's newly patented subsea compact separation process. A warm-start variant of the proposed first-order method outperforms a tailored interior-point method by a factor of 4 while occupying 40% less memory.**

I. Introduction

Model Predictive Control (MPC) has been successfully used over the years in several applications, especially in the process industry where MPC was given much attention before solid theoretical support was established for some critical aspects (e.g. stability) [1], [2]. Today, a lot of theoretical MPC literature exist due to advancements in research.

It is notable that the success of MPC in industrial applications, to a large extent, is dominated by advanced high-level process control applications based on software implementation in PC/server technology [1], [2]. However, significant progress has been made in recent years in the area of embedded Model Predictive Control, where contributions cover online approaches, which exploit MPC problem structure [3], [4], [5], and the explicit approach, which pre-computes the solution of the parameterized MPC problem offline [6], [7]. Remarkable progress has been made in the academic research and development of efficient high-speed solvers. Algorithms that target embedded platforms are available in FiOrdOs [8], FORCES [5], CVXGEN [9], qpOASES [3], and MPT3 [10], just to name a few. Earlier work on the implementation aspects of MPC on embedded hardware are reported in [11], [12], [13], [14], [15], [7], where a common goal is the efficient use of resource constrained embedded hardware. Steps towards real-time guarantees or effects of limited computation time on MPC stability and feasibility have also been made in recent research work [16], [17].

Over the years, industrial MPC design packages have developed and implemented strategies that contribute to the immense success of MPC, even in cases where academic justification was lacking or impractical [2], [1], [18]. It is therefore timely to look at viable ways of combining well-proven possibilities in industrial MPC design packages with the fast growing developments in academic research and in the process contribute to meet the need for MPC solutions on ultra-reliable industrial embedded computers.

This paper presents the results of a feasibility study covering an implementation of embedded MPC on the ABB AC500 programmable logic controller (PLC) for a subsea separation process based on the linear MPC module in SEPTIC (Statoil Estimation and Prediction Tool for Identification and Control) [18]. The embedded MPC design approach used in this work combines custom code generation with problem size reduction methods, embedded real-time considerations, and a primal-dual first-order method that provides a *fast* and *light* quadratic programming (QP) solver obtained from FiOrdOs [8]. Another contribution of this paper is to introduce a new primal-dual first-order method to the control community. The method is adapted from the computer graphics community and shows superior performance over state-of-the-art first-order methods in a comparison study.

The following sections cover essential aspects of the embedded MPC, starting with the process model (Section II), the mathematical problem formulation (Section III), the primal-dual first-order method (Section IV), and the implementation aspects considered in order to achieve a functional high-performance predictive controller in a PLC (Section V). Finally, Section VI presents and discusses the hardware-in-the-loop simulation results, followed by concluding remarks.

II. The subsea separation process

The process consists of separating a multiphase input flow of liquid (oil/water) and gas at two stages (see Fig. 1). First, a Gas-Liquid Cylindrical Cyclone (GLCC) separates the liquid and gas coarsely, and at the second stage a phase splitter and a de-liquidizer are used for finer separation.

The main objective is the control of gas volume fraction in the gas and liquid outlets. Since the outlets lead to a compressor and a pump, it is essential that the gas and liquid

[1] Department of Engineering Cybernetics, Norwegian University of Science and Technology, O.S. Bragstads plass 2D N-7491 Trondheim, Norway. {kwame.kufoalor, lars.imsland, tor.arne.johansen}@itk.ntnu.no

[2] Automatic Control Laboratory ETH Zurich, Physikstrasse 3 8092 Zurich, Switzerland. {richters, morari}@ee.ethz.ch

[3] FD ER SF, Statoil ASA, Arkitekt Ebbells veg 10, Rotvoll, Norway. gise@statoil.no

* This work was supported by the Research Council of Norway and Statoil. The research leading to these results has also benefited from collaboration within the European Union's Seventh Framework Programme under EC-GA No. 607957 TEMPO – Training in Embedded Predictive Control and Optimization.

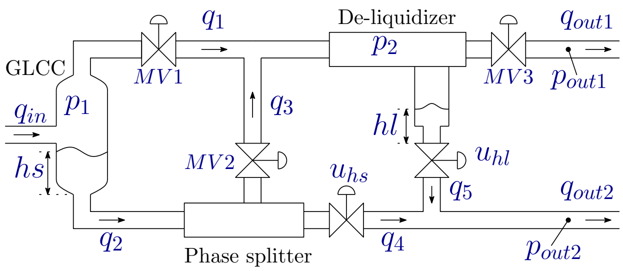

Fig. 1: The compact subsea separation process

contents are kept within their acceptable limits. It is also necessary to control the pressure in the GLCC and the de-liquidizer around their working points while respecting the physical limits of all valves. A key challenge is that, unlike most separation techniques, no buffer volumes are allowed in the subsea separation unit. Consequently, the dynamics are much faster, and disturbance effects are much more significant in the process. Due to space restrictions, the reader is referred to the patent description in [19] for further details as well as the mathematical modeling of the process.

The above process description naturally leads to a multi-variable control problem with constraints on both inputs and outputs. MPC is chosen as the preferred control method because besides its predictive behavior, constraints are explicitly taken into account requiring no additional logics and ad hoc strategies. Since the separation unit is to be placed at the sea bed, an embedded MPC solution is desirable.

III. The Embedded MPC Formulation

The MPC problem can be formulated as

$$\min \sum_{k=k_0+H_w}^{k_0+H_p} y_e(k)^T \bar{Q}_y y_e(k) + \sum_{k=k_0}^{k_0+H_u} \Delta u(k)^T \bar{P} \Delta u(k) + \rho_h^T \varepsilon_h + \rho_l^T \varepsilon_l \quad (1)$$

subject to

$$\begin{aligned}
&\underline{y} - \varepsilon_l \le y(k) \le \bar{y} + \varepsilon_h, && k \in \{k_0+H_w, \dots, k_0+H_p\},\\
&\varepsilon_h \ge 0,\ \varepsilon_l \ge 0, &&\\
&y(k) = y(k|k_0), && k \in \{k_0+H_w, \dots, k_0+H_p\},\\
&\underline{u} \le u(k) \le \bar{u}, && k \in \{k_0, \dots, k_0+H_u\},\\
&\underline{\Delta u} \le \Delta u(k) \le \overline{\Delta u}, && k \in \{k_0, \dots, k_0+H_u\},\\
&u(k) = u(k-1) + \Delta u(k), && k \in \{k_0, \dots, k_0+H_u\},
\end{aligned}$$

where k_0 is the initial time instant, H_p and H_u are the prediction and control horizons, respectively, and $H_w > 1$ specifies the number of initial steps in which the deviations from the controlled variable (CV) reference $r(k)$, $y_e(k) = y(k) - r(k)$, are not penalized. $\bar{Q}_y \succeq 0$, and $\bar{P} \succ 0$ are weighting matrices, and $\Delta u(k)$ is the change in input $u(k)$. The input u is also known as the manipulated variable (MV), and $u_e(k) = u(k) - iv(k)$ is the deviation error from the ideal (or steady state) value $iv(k)$ of the MV. The *slack* or *penalty* variables $\varepsilon_h, \varepsilon_l$, weighted by $\rho_h, \rho_l > 0$, relax the upper and lower constraints of the CVs so that infeasibility cannot occur in case of large disturbances or prediction model errors. The closed loop stability of the MPC problem can be achieved by an adequate choice of the weights $\bar{Q}_y$, $\bar{P}$ and the horizon lengths H_p and H_u.

The prediction model $y(k|k_0)$ is obtained from SEPTIC's single-input single-output (SISO) step response models that are grouped to form a sampled multi-input multi-output (MIMO) system according to the formulations in [1, §4.1.3]. A constant disturbance model is further used to introduce an integral action and thus remove steady-state control errors.

After lumping all decision variables of same type in problem (1) into vectors

$$\begin{aligned}
Y &= \big(y(k_0+H_w), y(k_0+H_w+1), \dots, y(k_0+H_p)\big),\\
U &= \big(u(k_0), u(k_0+1), \dots, u(k_0+H_u)\big),\\
\Delta U &= \big(\Delta u(k_0), \Delta u(k_0+1), \dots, \Delta u(k_0+H_u)\big),
\end{aligned}$$

and defining the single decision vector x as

$$x = \big(\Delta U, U, Y, \varepsilon_h, \varepsilon_l\big),$$

problem (1) can be rewritten in compact form as the QP

$$\min \left\{ \tfrac{1}{2} x^T H x + g^T x \;\middle|\; \bar{A}_i x \le \bar{b}_i,\ A_e x = b_e \right\}. \quad (2)$$

Note that vectors g, $\bar{b}_i$ and b_e as well as matrices H, $\bar{A}_i$ and A_e are easily deduced from the step response model as well as the objective and constraints in (1). Explicit expressions are omitted for the sake of saving space.

To reduce the size of decision vector x and thus enhance solver speed, we have applied common techniques such as move blocking (manipulated variables are fixed to be constant over several time-steps) and evaluation of the controlled variables $y(k)$ on a subset of $\{k_0+H_w, \dots, k_0+H_p\}$ only.

IV. The Primal-Dual First-Order Method

This section discusses the solution method for the MPC problem in (2). For the sake of a compact presentation, we first rewrite the MPC problem in standard notation as

$$\min \left\{ \tfrac{1}{2} x^T H x + g^T x \;\middle|\; x \in \mathbb{X},\ A_i x \le b_i,\ A_e x = b_e \right\}, \quad (3)$$

where $x \in \mathbb{R}^n$ is the decision vector, $b_e \in \mathbb{R}^{m_e}$ is the parameter changing in every sampling instant and $H \in \mathbb{R}^{n \times n}$ is the positive semi-definite Hessian. The convex set $\mathbb{X}$ is defined by all the inequalities in (2) such that the *projection operator*

$$\pi_{\mathbb{X}}(x) = \arg\min_{z \in \mathbb{X}} \tfrac{1}{2} \|z - x\|^2 \quad (4)$$

can be evaluated analytically or by means of an algorithm with finite convergence. For MPC problem (2), set $\mathbb{X}$ contains upper/lower bounds on ΔU, U, ε_h and ε_l. Note that projection on the penalized output constraints is not simple, in fact, solving this problem parametrically with MPT [10] leads to more than 30'000 regions. So, the inequalities involving Y are better kept as $A_i x \le b_i$, $b_i \in \mathbb{R}^{m_i}$ in (3).

For the MPC problem in this paper, we propose the preconditioned primal-dual first-order method in [20] which was originally developed for imaging applications. This method

Algorithm 1 Preconditioned primal-dual first-order method for problem (3)

Require: $\lambda_0 \in \mathbb{R}^m$, $x_0 \in \mathbb{R}^n$, $\bar{x}_0 = x_0$; A, b according to (7); preconditioner matrices $T \in \mathbb{R}^{n\times n}$, $\Sigma \in \mathbb{R}^{m\times m}$ chosen according to (8)

1: **loop**
2: $\quad \lambda_{i+1} = \pi_\Lambda\left(\lambda_i + \Sigma(A\bar{x}_i - b)\right)$ (cf. (4))
3: $\quad x_{i+1} = \arg\min_{x\in\mathbb{X}} \frac{1}{2}x^T Hx + g^T x + \frac{1}{2}\left\|x - (x_i - TA^T\lambda_{i+1})\right\|^2_{T^{-1}}$
4: $\quad \bar{x}_{i+1} = 2x_{i+1} - x_i$
5: **end loop**

operates on both primal *and* dual iterates and can be applied to solve convex problems of min-max type

$$\min_x \max_\lambda x^T A^T \lambda + \phi(x) - \theta^*(\lambda), \tag{5}$$

where x and λ are the primal and dual variables respectively. Functions ϕ and θ^* can be any extended real-valued, convex and closed functions with the important property that their *preconditioned proximity operators*, e.g.

$$\operatorname{prox}_\phi(x) = \arg\min_z \phi(z) + \tfrac{1}{2}\|z - x\|^2_{T^{-1}}, \tag{6}$$

with $T \in \mathbb{R}^{n\times n}$ being some positive definite matrix, can be evaluated in closed-form or computed efficiently. Problem (3) can be rewritten as (5) by using the definitions

$$A = \begin{bmatrix} A_e \\ A_i \end{bmatrix}, \quad \theta^*(\lambda) = \lambda^T b + \iota_\Lambda(\lambda) \text{ with } b = \begin{bmatrix} b_e \\ b_i \end{bmatrix}, \tag{7}$$

where $b \in \mathbb{R}^m$, $m = m_e + m_i$, and $\phi(x) = \frac{1}{2}x^T Hx + g^T x + \iota_\mathbb{X}(x)$. In these definitions, $\iota(\cdot)$ denotes the indicator function of the corresponding set, e.g. for the dual set Λ it is given as

$$\iota_\Lambda(\lambda) = \begin{cases} 0 & \text{if } \lambda \in \left\{(\lambda_e, \lambda_i) \in \mathbb{R}^{m_e} \times \mathbb{R}^{m_i} \,\middle|\, \lambda_i \geq 0\right\}, \\ +\infty & \text{otherwise.} \end{cases}$$

The algorithmic scheme of the preconditioned primal-dual first-order method from [20], adapted to problem (3), is stated in Algorithm 1. It can be shown that the sequences $\{x_i\}$, $\{\lambda_i\}$ converge to a primal/dual pair of optimizers (x^*, λ^*) of (3) if the preconditioner matrices are chosen as the diagonal matrices $\Sigma = \alpha \cdot \operatorname{diag}(\sigma)$, $\alpha \in (0,1)$, and $T = \operatorname{diag}(\tau)$ where

$$\sigma_i = \frac{1}{\sum_{j=1}^n |A_{ij}|},\ i = 1\ldots m, \text{ and } \tau_j = \frac{1}{\sum_{i=1}^m |A_{ij}|},\ j = 1\ldots n. \tag{8}$$

The preconditioner matrices Σ and T can be interpreted as 'step size matrices'. In fact, choosing them according to (8) with $\alpha \in (0,1)$ ensures that the convergence criterion

$$\|\Sigma^{\frac{1}{2}} A T^{\frac{1}{2}}\|^2 < 1 \tag{9}$$

for Algorithm 1 is satisfied (cf. [20, Lemma 2]). In general, larger step sizes lead to faster convergence, hence, constant α should be chosen close to 1. The following remarks are important for further speeding up convergence.

Remark 1: If matrix A does not change at runtime, it is possible to *rescale* the preconditioners, i.e. let

$$\nu = \|\Sigma^{\frac{1}{2}} A T^{\frac{1}{2}}\|^{-1},$$

where matrices Σ and T are chosen according to (8). The reader may verify that the rescaled preconditioner matrices

$$\widetilde{\Sigma} = \alpha \cdot \nu \cdot \Sigma \quad \text{and} \quad \widetilde{T} = \nu \cdot T$$

with $\alpha \in (0,1)$ but close to 1, satisfy the convergence criterion (9) and can only improve step sizes given by the previous preconditioner matrices since $\nu > 1$.

Remark 2: Any preconditioners Σ and T that fulfill the convergence criterion in (9) give rise to new preconditioners

$$\overline{\Sigma} = \eta \cdot \Sigma \quad \text{and} \quad \overline{T} = \eta^{-1} \cdot T$$

that, for any $\eta > 0$, also fulfill the convergence criterion. This fact can be exploited for tuning purposes, i.e. to balance primal/dual step sizes.

The crucial step in Algorithm 1 is in line 3: Only if the minimization problem in x can be solved efficiently, good performance of the primal-dual first-order method can be expected. For the MPC problem in this paper, set $\mathbb{X}$ contains upper/lower bounds on some components of x only, whereas the Hessian H is diagonal. Since the preconditioner matrix T is diagonal positive definite, the minimizer in line 3 is

$$x_{i+1} = \pi_\mathbb{X}\left(-\left(H + T^{-1}\right)^{-1}\left(g - T^{-1}(x_i - TA^T\lambda_{i+1})\right)\right),$$

where the projection on set $\mathbb{X}$ is a component-wise saturation.

A. Code Generation with FiOrdOs

FiOrdOs[1] is a Matlab toolbox that allows to transform a Matlab description of a parametric optimization problem of type (3) into a C code implementation of a dedicated first-order method. In the upcoming release of FiOrdOs, the primal-dual first-order method in Algorithm 1 will be available. The C code implementation for the MPC problem in this paper was obtained from a preliminary release.

B. Comparison Study

In the following, we present results of a comparison between the primal-dual first-order method in Algorithm 1 and other state-of-the-art first-order methods in Matlab. It turns out that for the MPC example of this paper, the proposed method has superior performance over all other methods. So, it is the first choice for an implementation on a PLC.

For a careful and fair comparison, we have created a set with 70 instances of the parameter b_e, where all of them lead to penalty variables $\varepsilon_h^* = \varepsilon_l^* = 0$ in the optimal solution. For 36 instances, no other constraint is active in the optimal solution, i.e. the optimal solution can be obtained in closed form. For 13 instances, at most three additional inequality constraints are active, and for the remaining 21 instances, 11 additional inequalities are active at the optimal solution.

All iterates of the tested methods were initialized at the origin (*cold-start*) and the stopping criterion was chosen as $\|x_i - x^*\|/\|x^*\| \leq 10^{-3}$ (in all cases, the minimizer is unique). This accuracy turned out necessary for obtaining good control performance in hardware-in-the-loop simulations.

For those methods that require a positive definite Hessian, we have computed the largest common perturbation $\delta > 0$, such that the solution x_δ^* to problem (3) with the perturbed Hessian $H_\delta = H + \delta \cdot I_n$ satisfies $\|x_\delta^* - x^*\|/\|x^*\| \leq 0.5 \cdot 10^{-3}$. The value found is $\delta = 5 \cdot 10^{-4}$, leading to a condition number of about $2 \cdot 10^5$ for the perturbed Hessian.

[1] See fiordos.ethz.ch for further details and download.

TABLE I: Performance of different first-order methods for the solution of the MPC problem in this paper. Methods with superscript $^+$ have a significantly higher per iteration cost than Algorithm 1. Methods with subscript $_\delta$ require a positive definite Hessian, i.e. they solve the perturbed problem.

Method	Parameters	min / average / max iterations	
		no restart	with restart
Algorithm 1	$\eta = 10$	137/211/347	n/a
Algorithm 1	$\eta = 1$	63/620/2496	n/a
ADMM$^+$ [21]	$\rho = 25,\ \alpha = 1.62$	129/153/197	n/a
Dual FGM$_\delta$ [17]	$L = \|AH_\delta^{-1}A^T\|$	53/3668/11621	53/3639/11412
Dual FGM$_\delta$ [22]	L_λ, L_μ from (10)	6/4365/18629	6/4089/18663
GPAD$_\delta$ [23]	$L_\Psi = \|A_i H_\delta^{-1} A_i^T\|$	all $> 100'000$	all $> 100'000$
GPAD$_\delta^+$ [22]	$L_\mu = A_i H_\delta^{-1} A_i^T + 10^{-6} I$	2/463/3702	2/1359/4546

The results of the comparison study are presented in Table I which also states the main parameters of the methods in the notation used in the cited original publications. The study encompasses the proposed method in Algorithm 1, untuned ($\eta = 1$, cf. Remark 2) and empirically tuned for good performance ($\eta = 10$). The latter is used in the benchmark implementation in Section VI-B. Note that prior to balancing the preconditioners with constant η, we follow the rescale procedure in Remark 1 in both cases.
Furthermore, we have tested the alternating direction method of multipliers (ADMM) with the splitting proposed by [21], assuming that projection on set $\mathbb{X} \cap \{x \,|\, A_i x \le b_i\}$ is 'simple'. (In fact, it is not simple, and the projection operator was evaluated with CPLEX in our study. However, the iteration count gives a baseline for comparison with our method.)
Next, we have implemented the dual fast gradient method (FGM) from [17] with the optimal step size determined by $1/L$. This method requires a strongly convex objective, hence we had to resort to the perturbed Hessian H_δ. The method in [22] is a very recent generalization of the former, using a generalized fast gradient method. Full matrix L_λ and diagonal matrix L_μ were computed via the convex SDP

$$\min\left\{\text{trace}(L_\lambda) + \text{trace}(L_\mu) \,\middle|\, \text{diag}(L_\lambda, L_\mu) \succeq AH_\delta^{-1}A^T\right\}. \quad (10)$$

Finally, results from the accelerated dual gradient projection method (GPAD) from [23] and its generalization in [22] are reported. These methods require the perturbed Hessian and dualize all inequality constraints (also the ones defining set $\mathbb{X}$). For the generalized variant of [22] we have chosen L_μ as given in Table I. This leads to a subproblem that for our MPC problem cannot be solved in closed form but requires CPLEX. Thus, the stated iteration counts must be understood as a baseline for comparison reasons only. (A diagonal L_μ leads to a closed form solution, however, we have obtained more than $25'000$ iterations in this case.) Restarting variants of the methods are implemented according to [22, Eq. (40)]. Note that both dual FGM and GPAD do not have any tuning parameters, whereas ADMM has parameters ρ and α which have been chosen empirically for good performance.

From the results in Table I we conclude that, taking into account the per iteration cost, the primal-dual first-order method performs by far best. Even the untuned version shows faster convergence than all the other methods with similar per iteration costs. For our MPC problem, the generalized FGM performs worse than the standard dual FGM. Significant improvement was only observed for those problem instances with a small number of inequalities active at the optimal solution. The more active inequalities, the worse the observed performance. Another interesting observation is that restarting can slow down convergence.

V. The Embedded Platform

PLCs are found in many industrial control systems today due to their operational robustness, suitability in harsh environments, and the availability of multiple I/O arrangements and several standard industrial communication protocols that usually form an integrated part of PLCs. Their use is popular when it comes to relay replacement logics and simple control algorithms which are easily implemented using the classic IEC 61131-3 PLC programming languages. However, the software resource availability and development in real-time programming languages such as C/Real-time POSIX and Real-time Java make the IEC 61131-3 languages less attractive for implementing complex control algorithms. In recent times, support for general purpose programming languages such as C/C++ is emerging as an integrated part of the PLC software environment, and therefore making the PLC attractive for embedded MPC applications.

The target platform is the ABB AC500 PM592-ETH PLC, and the PLC software development tool used is the ABB PS501 Control Builder Plus 2.3, which is based on the CoDeSys automation platform technology. Programming and runtime support is offered for ANSI C89 and C99 code integrated into a PLC software/runtime architecture. A C code application in the PLC is therefore defined as an IEC 61131-3 application that has at least one function or function block written in C. To compile the C code part of the PLC application, the GNU GCC 4.7.0 compiler toolchain is offered with limited compiler/optimization options. Linking against external libraries is not supported, implying that a library-free MPC code is preferred.

The practical feasibility of MPC on a PLC depends immensely on the processing power and memory requirements of the given MPC problem and the selected solution approach/strategy. The AC500 PM592-ETH PLC has a Freescale™ G2_LE implementation of the MPC603e microprocessor that runs at 400 MHz. The MPC603e is a RISC CPU with a dedicated hardware floating point unit (FPU), which is fully IEEE 754-compliant for both single and double precision. The PLC is also equipped with 4MB RAM for user program memory and 4MB integrated user data memory.

VI. Embedded MPC Results on the PLC

A. Test Setup and PLC Implementation

The embedded MPC application is based on the design specifications obtained from the linear MPC configuration (config) of the subsea separation process in SEPTIC. An automatic C code generator is used to extract process and MPC design data from the SEPTIC config files, and custom QP config file/data and online measurement update functions are produced based on the MPC formulation outlined in Section III. A C code generator (e.g. FiOrdOs) is used to obtain the QP solver code based on the custom QP config file/data. The complete MPC code is implemented in the PLC software development environment as a C function block. The resulting embedded MPC program has a dedicated task that starts initialization and runs a main loop consisting of

- a function for loading initial conditions for the MPC,
- functions to send and receive data and scheduling tags,
- functions for calculating unmeasured disturbances, and
- a custom QP solver with a warm-starting routine.

Further details of the embedded MPC design approach and some preliminary results are reported in [24].

The MPC for the subsea separator has 4 CVs with 10 evaluation points each, 3 MVs (cf. Fig. 1), each with 6 move blocking indices, 2 measured process disturbances (DVs), and 6 slack variables. In total, the problem contains 58 equality constraints, 138 inequality constraints, and 82 decision variables. The valves labeled u_{hs} and u_{hl} in Fig. 1 are controlled by dedicated controllers that provide safety level control for the liquid levels hs and hl.

In this paper embedded MPC based on CVXGEN's primal-dual interior-point method (IP) is used as the benchmark for comparison with the MPC implementation based on the proposed primal-dual first-order method in Algorithm 1. CVXGEN provides a library-free custom C code generation framework that is suitable for the standard QP formulation in this study (no reformulations required). Moreover, the optimality and numerical stability parameters provided by CVXGEN make it possible to tune the QP solver for its "best" solution for the PLC application in a straightforward manner. The following *strict* optimality and numerical stability parameters were used for the CVXGEN QP solver: *duality gap* $= 10^{-4}$, *constraint residual* $= 10^{-6}$, *max iterations* $= 25$, *KKT regularization* $= 10^{-7}$, and *refine steps* $= 1$. The control performance results of the IP based MPC are referred to as the *benchmark* results in the following sections. CVXGEN recommends the use of cold-start since the solver's performance does not improve significantly with warm-start. Therefore only cold-start results are provided for the benchmark controller.

The hardware-in-the-loop test setup consists of the PLC implementation of the embedded MPC in closed loop with a SEPTIC process simulator developed for the subsea compact separation unit. For testing the control performance a *hydrodynamic slugging* scenario is assumed. Slugging occurs when gas flows faster than the liquid in the inlet pipe, resulting in waves on the liquid surface that grow large enough to fill the pipe completely. This situation is considered as a worst case scenario in the inlet flow of the compact separator. The inlet flow sequence used to simulate the slugging case consists of the two fast-changing process disturbances shown in Fig. 2.

A sampling frequency of 1 Hz was used, requiring real-time computational time much less than a second.

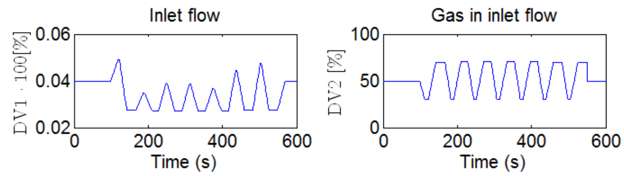

Fig. 2: Inlet flow of the hydrodynamic slugging case

TABLE II: Real-time closed-loop results on the PLC for 600 time steps of the subsea compact separation process. Abbreviation *sp* denotes single precision floating point.

QP Solver	**Time** (ms) avg./max	**Iterations** avg./max	**Mean Square Error** CV1/CV2/CV3/CV4	**C / PLC** Code (MB)
1: IP-cold	72.2/84.9	15/18	0.04/0.008/2.68/0.31	0.96/2.16
2: IP-cold,*sp*	63.8/65.4	18/18	0.04/0.008/2.35/0.31	0.92/2.14
3: Alg.1-cold	114.6/116.4	785/785	0.04/0.008/2.56/0.31	0.56/1.35
4: Alg.1-cold,*sp*	102.9/104.7	785/785	0.04/0.008/2.20/0.33	0.54/1.33
5: Alg.1-warm	18.2/19.8	100/100	0.02/0.003/2.81/0.28	0.56/1.35
6: Alg.1-warm,*sp*	15.3/16.9	100/100	0.02/0.003/2.96/0.31	0.54/1.33

B. Results for the Primal-Dual First-Order Method

FiOrdOs was used to generate a library-free C code that implements the proposed first-order primal-dual method outlined in Algorithm 1. The control performance of the resulting embedded MPC program that runs on the AC500 PLC is summarized in Table II and Fig. 3. Tests 1–4 in Table II represent baseline tests for cold-start performance of the IP and the first-order method, where the Mean Square Error (MSE) values of the CVs were used as the control performance measure. The deviations from the pressure setpoint values were used for calculating the MSE of CV1 and CV2, while the measured gas contents in the process outlets were used for calculating the MSE of CV3 and CV4.

The maximum iteration limit for the first-order method that provides a similar control performance as the IP benchmark was found to be 785 iterations. Although this means about 45 times more iterations than the IP solver, the computation times are close (only 40% increase for double precision, 60% increase for single precision (*sp*)). However, the first-order method can benefit substantially from warm-starting from the shifted previous solution. Tests 5–6 in Table II indicate that about 6 times computational speed-up is achieved. Moreover, the corresponding MSE values of the CVs in Table II show an improvement in control performance. Note that warm-starting for IP is not supported by CVXGEN and also, warm-starting for IP requires more effort than in the first-order case. For more details on warm-starting in MPC we refer the reader to [25] where speed-ups of 1–5 times are reported.

Table II also lists the compiled MPC code size ('C') and the overall PLC program size ('PLC'). The numbers underline that a significantly *light* program is obtained when using the proposed first-order method. The difference in the C code and the PLC program size is an indication of the extra memory required to incorporate the C code into the PLC software structure defined in the AC500 592-ETH target.

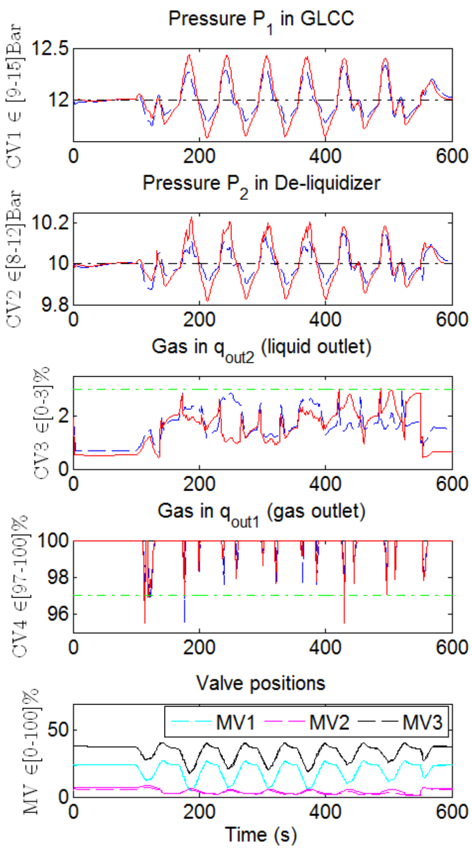

Fig. 3: Warm-start control performance of the first-order primal-dual method — —, IP *benchmark* results ——

VII. Conclusions

This paper provides a viable approach to achieve a real-time predictive controller on an ultra-reliable industrial hardware and, furthermore, motivates the use of first-order methods in embedded MPC. Essential aspects of a PLC implementation of embedded MPC for an industrial problem are covered, where automatic code generation, problem size reduction methods, embedded real-time considerations, and the proposed primal-dual first-order method are highlighted as key elements that contribute to the successful application. The results of the comparison study on first-order methods emphasize the potential of the proposed primal-dual method. The PLC implementation of the warm-start variant was found to outperform a tailored interior-point method by a factor of 4 while occupying 40% less memory.

References

[1] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson and Prentice Hall, 2002.

[2] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[3] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust Nonlinear Control*, vol. 18, pp. 816–830, July 2008.

[4] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[5] A. Domahidi, A. Zgraggen, M. Zeilinger, M. Morari, and C. Jones, "Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control," in *IEEE Conference on Decision and Control*, Maui, HI, USA, Dec. 2012, pp. 668 – 674.

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[7] G. Valencia-Palomo and J. A. Rossiter, "Efficient suboptimal parametric solutions to predictive control for PLC applications," *Control Engineering Practice*, vol. 19, no. 7, pp. 732–743, 2011.

[8] F. Ullmann, "A Matlab toolbox for C-code generation for first order methods," Master's thesis, ETH Zurich, 2011.

[9] J. Mattingley and S. Boyd, "CVXGEN: A Code Generator for Embedded Convex Optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[10] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510.

[11] P. Zometa, M. Kögel, T. Faulwasser, and R. Findeisen, "Implementation aspects of model predictive control for embedded systems," in *2012 American Control Conference*. Fairmont Queen Elizabeth, Montreal, Canada: AACC, June 2012, pp. 263–275.

[12] A. G. Wills, G. Knagge, and B. Ninness, "Fast Linear Model Predictive Control via Custom Integrated Circuit Architecture," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, pp. 59–71, 2012.

[13] J. L. Jerez, K. V. Ling, G. A. Constantinides, and E. C. Kerrigan, "Model predictive control for deeply pipelined field-programmable gate array implementation: algorithms and circuitry," *IET Control Theory and Applications*, vol. 6, no. 8, pp. 1029–1041, July 2012.

[14] B. Huyck, H. J. Ferreau, M. Diehl, J. D. Brabanter, J. F. M. V. Impe, B. D. Moor, and F. Logist, "Towards Online Model Predictive Control on a Programmable Logic Controller: Practical Considerations," *Mathematical Problems in Engineering*, vol. 2012, pp. 1–20, 2012.

[15] A. Roldao-Lopes, A. Shahzad, G. Constantinides, and E. Kerrigan, "More Flops or More Precision? Accuracy Parameterizable Linear Equation Solvers for Model Predictive Control," in *FCCM '09*, April 2009, pp. 209–216.

[16] C. N. Jones, A. Domahidi, M. Morari, S. Richter, F. Ullmann, and M. N. Zeilinger, "Fast Predictive Control: Real-Time Computation and Certification," in *IFAC Conference on Nonlinear Model Predictive Control*, Noordwijkerhout, the Netherlands, August 2012, pp. 94–98.

[17] S. Richter, C. N. Jones, and M. Morari, "Certification Aspects of the Fast Gradient Method for Solving the Dual of Parametric Convex Programs," *Mathematical Methods of Operations Research*, vol. 77, no. 3, pp. 305–321, Jan. 2013.

[18] S. Strand and J. Sagli, "MPC in Statoil – advantages with in-house technology," in *International Symposium on Advanced Control of Chemical Processes (ADCHEM)*, Hong Kong, 2003, pp. 97–103.

[19] J. Høydal, O. Kristiansen, G. O. Eikrem, and K. Fjalestad, "Method and system for fluid separation with an integrated control system," Patent WO2 013 091 719 A1, 06 27, 2013.

[20] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *IEEE International Conference on Computer Vision*, 2011, pp. 1762–1769.

[21] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A Splitting Method for Optimal Control," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2432–2442, Nov. 2013.

[22] P. Giselsson, "Improving Fast Dual Ascent for MPC - Part II: The Embedded Case," Dec. 2013, version 1.

[23] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 18–33, 2014.

[24] V. Aaker, "Embedded MPC for a Subsea Separation Process," Master's thesis, Department of Engineering Cybernetics, NTNU, June 2012.

[25] A. Shahzad and P. J. Goulart, "A new hot-start interior-point method for model predictive control," in *Proceedings of the 18th IFAC World Congress Milano (Italy)*, vol. 18, no. 1, 2011.