Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-18 (2018) 518–523

Parallelizable Real-Time Algorithm for Integrated Experiment Design MPC

Xuhui Feng * Yuning Jiang * Mario Eduardo Villanueva * Boris Houska *

* *School of Information Science and Technology, ShanghaiTech University, Shanghai, China (fengxh@shanghaitech.edu.cn)*

Abstract: This paper proposes a parallelizable real-time algorithm for integrated experiment-design model predictive control (MPC). Integrated experiment design MPC is needed if a system is not observable at a tracking reference and needs to be excited on purpose in order to be able to estimate the system's states and parameters. The contribution of this paper is a real-time MPC algorithm using two processors. On the first processor an extended Kalman filter (EKF) as well as a parametric certainty-equivalent MPC controller are implemented, which can provide immediate feedback at high sampling rates. On the second processor, optimal experiment design (OED) problems are solved in parallel in order to perturb the certainty-equivalent MPC control loop improving the accuracy of the state estimator at a lower sampling rate. We show that this framework can achieve optimal tradeoffs between OED and control objectives. The approach is applied to a biochemical process in order to illustrate that the proposed controller can achieve superior control performance when compared to certainty-equivalent MPC.

Keywords: Model Predictive Control, Parallelizable Algorithms, Optimal Experiment Design.

1. INTRODUCTION

Model predictive control (MPC) is an advanced control technique, which is capable of dealing with inequality state- and control constraints (Mayne et.al. (2000); Rawlings and Mayne (2009)). Certainty-equivalent MPC controllers work well in practice if a reasonably accurate model is available (Zhu (2009)). Here, a basic requirement is that the controlled system is observable such that state and parameter estimates can be determined online, e.g., by using an extended Kalman filter (EKF) or moving horizon estimator (Diehl et.al. (2009); Rao et.al. (2001)).

For some systems this observability requirement fails to hold if no further precaution is taken. In such scenarios, for example, if the system is not observable at steady-state or if significant control excitations are needed to improve the accuracy of the online state and parameter estimates, it is advisable to augment the MPC objective function with an additional optimal experiment design (OED) objective (Fisher (1935); Pukelsheim (1993)). During the last decade there have been numerous suggestions on how to integrate OED criteria in MPC (see Yan et.al. (2005)). For example, Hovd and Bitmead (2005) suggest to augment the system dynamics by a Riccati differential equation implementing an EKF. The matrix valued state of this Kalman filter can be used to predict the variance of future state estimates, which, in turn, can be penalized in the MPC objective. Similar strategies to integrate OED criteria in MPC can be found in (Heirung et.al. (2012, 2015)). Notice that there is also a number of articles on persistently exciting MPC (Hernandez and Trodden (2015); Marafioti et.al. (2014); Mesbah et.al. (2015); Zacekova et.al. (2013)), which have appeared recently and which all propose variants to perturb the nominally optimal control input in order to increase the accuracy of future state and parameter estimates. Houska et.al. (2017) suggested a self-reflective MPC controller, which proceeds by minimizing the sum of a nominal control performance term as well as an additional term measuring the expected inherent loss of optimality of the controller (Stengel (1994)).

The above reviewed approaches have in common that they introduce additional hyperstates for predicting the accuracy of future state estimates. In particular, if the variance of future state estimates is penalized by augmenting the system dynamics with an EKF, matrix valued hyperstates have to be included in the MPC optimization problem, increasing the difficulty to solve it (Telen et.al. (2016)). Recently, in (Feng and Houska (2016)) a tailored real-time optimization algorithm has been suggested, which attempts to exploit the particular structure of MPC with integrated experiment design objectives. This approach improves the computation time of self-reflective MPC.

Therefore, the current paper proposes an algorithm, which splits the augmented MPC problem into a nominal and an experiment design part. Section 2 starts with a review of existing formulations of integrated experiment design MPC. The main contribution of this paper is presented in Section 3, where we propose to use two processors, which communicate with each other and which provide feedback at different sampling rates. On the first processor a parameterized standard MPC controller is running with high-sampling rate, while, on the other processor OED problems are solved. An associated communication scheme uses ideas from the field of augmented Lagrangian based alternating direction inexact Newton (ALADIN) methods (Houska et.al. (2016)), which ensures that both processors can find a compromise between control excitation

10.1016/j.ifacol.2018.09.372

for the purpose of improving future state estimates and nominal control performance. Section 4 compares the runtime and control performance of the algorithm for nominal and integrated-experiment design MPC by studying a challenging biochemical process that is not observable at its steady-state. Section 5 concludes the paper.

2. INTEGRATED OED BASED MPC

This section reviews methods for including OED criteria in the objective of an MPC controller. The focus is on discrete-time control systems of the form

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \\ \eta_k &= h(x_k) + v_k \,, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ denotes the state, $u_k \in \mathbb{R}^{n_u}$ the control input, and $\eta_k \in \mathbb{R}^{n_h}$ the measurement at time k. The functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_h}$ are assumed to be nonlinear and at least three times continuously differentiable. We are interested in analyzing OED criteria since the process noise w_k and the measurement error v_k are random variables. The system states have to be estimated frequently from the most recent measurements. As the control u_k enters f nonlinearly, the control input might influence the observability properties of the system. In the following, it is assumed that the v_ks and w_ks are uncorrelated in time with mean 0 and given variances $V \in \mathbb{S}^{n_h}_{++}$ and $W \in \mathbb{S}^{n_x}_{+}$, respectively [1].

Remark 1. Notice that in the above framework uncertain parameters can be included by introducing auxiliary state variables that are invariant with respect to k, i.e., by stacking equations of the form $p_{k+1} = p_k$ to the discrete time recursion and regarding these parameters as states.

2.1 Model Predictive Control

Certainty-equivalent MPC proceeds by solving online optimization problems of the form

$$L(u, \hat{x}) = \min_{x} \sum_{k=0}^{N-1} l(x_k, u_k) + m(x_N) \qquad \text{s.t.} \begin{cases} \forall k \in \{0, \ldots, N-1\}, \\ x_{k+1} = f(x_k, u_k), \;\; x_0 = \hat{x}, \\ c(x_k, u_k) \leq 0 \,. \end{cases} \tag{2}$$

Here, $l : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ denotes the stage cost and $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ the mixed control- and state constraints. The prediction is computed under a certainty-equivalence paradigm, i.e., as if the current state estimate $\hat{x}$ was known exactly and as if there was no process noise, $w_k = 0$. Of course, in practice, the certainty-equivalent controller only achieves reasonable control performance if the state estimates, e.g., found by an EKF, are reasonably accurate and if the controller is run in a receding horizon way; that is, the above problem is re-solved whenever a new state measurement $\hat{x}$ is available and the corresponding first element of the control sequence, denoted by u_0^*, is sent to the real process.

Remark 2. Notice that there exist many articles on how to formulate MPC problems of the above form such that suitable closed-loop stability or other closed-loop performance criteria are met (Rawlings and Mayne (2009)).

2.2 Integrated Optimal Experiment Design

For many processes the above certainty equivalent MPC in combination with an EKF works reasonably well in practice and, of course, in this case, no further modifications are needed. However, in some situations, e.g., if the MPC controller attempts to track a steady state at which the system states are not observable, EKF-MPC cascades may fail to perform well (Houska et.al. (2017); Telen et.al. (2016)). In this case, the predicted variance of future state estimates can be penalized in the objective of the MPC controller. This leads to an augmented optimization problem of the form

$$\min_{x,u,\Sigma} \sum_{k=0}^{N-1} l(x_k, u_k) + m(x_N) + \sum_{k=0}^{N} \Phi(x_k, u_k, \Sigma_k) \qquad \text{s.t.} \begin{cases} \forall k \in \{0, \ldots, N-1\}, \\ x_{k+1} = f(x_k, u_k), \;\; x_0 = \hat{x} \\ \Sigma_{k+1} = F(x_k, u_k, \Sigma_k), \;\; \Sigma_0 = \hat{\Sigma} \\ c(x_k, u_k) \leq 0 \,, \end{cases} \tag{3}$$

where Σ denotes the matrix-valued state of the EKF with

$$\begin{aligned} & F(x, u, \Sigma) := A(x, u) G(x, u, \Sigma) A(x, u)^{\mathsf{T}} + W \\ & G(x, u, \Sigma) := \\ & \Sigma - \Sigma C(x)^{\mathsf{T}} \left(C(x) \Sigma C(x)^{\mathsf{T}} + V \right)^{-1} C(x) \Sigma \,. \end{aligned} \tag{4}$$

Here,

$$A(x,\ u) = \frac{\partial f}{\partial x}(x,\ u) \quad \text{and} \quad C(x) = \frac{\partial h}{\partial x}(x)$$

denote the first order derivatives of f and h with respect to the states.

Notice that there are many ways to choose the functions $\Phi : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{S}^{n_x}_{+} \to \mathbb{R}$. For example, in Hovd and Bitmead (2005) it is suggested to use a weighted trace

$$\Phi(x_k, u_k, \Sigma_k) := \mathrm{Tr}\left(\alpha_k \Sigma_k\right) \,,$$

where $\alpha_k \in \mathbb{S}^{n_x}_{+}$ can be any scaling matrices that the user can choose. In principle, one could also use other standard OED criteria replacing the trace with the maximum eigenvalue (E-criterion) or a determinant (D-criterion) (Ljung (1999)). In general, the function Φ could even depend on x_k and u_k. This is for example, the case if a self-reflective MPC controller is used (Houska et.al. (2017)), which chooses Φ in such a way that the additional term in the objective of (3) can be interpreted as the controller's own expected loss of optimality in the presence of the random process noise w_k and the random measurement errors v_k. For an overview of different choices of Φ in the context of MPC we refer to Telen et.al. (2016). Notice that all these problem formulations can be written in the form (3) by defining the function Φ appropriately.

In order to avoid misunderstanding, given the large amount of articles about integrated experiment design based MPC, problem formulation (3) is not an original contribution of this paper. Rather, the contribution of this

[1] The set of real symmetric and positive semidefinite $n \times n$ matrices is denoted by $\mathbb{S}^n_{+}$. Similarly, $\mathbb{S}^n_{++}$ denotes the set of symmetric positive definite matrices.

paper is the development of an algorithm that can solve problems of the form (3) efficiently and in real-time. Notice that such algorithms are of practical relevance, since the additional matrix-valued state Σ makes (3) much more difficult to solve than the original MPC problem (2). Thus, in order to make MPC with additional OED objectives useful for practical applications, tailored algorithms and software are needed. The goal of the following section is to propose such an algorithm that exploits the structure by alternating between MPC and OED objectives.

3. ALTERNATING EXPERIMENT DESIGN AND CONTROL OBJECTIVES

This section proposes a real-time algorithm for solving (3). Here, the main idea is to split the overall problem into two interconnected optimization problem that can be solved in parallel and on two processors. The first processor solves certainty-equivalent MPC problems at a high sampling rate while the second processor solves OED problems at a lower sampling rate, in order to support the decisions of the other processor. In order to explain how this works, we introduce the shorthand notation

$$E(u,\hat{x},\hat{\Sigma}) := \min_{x} \sum_{k=0}^{N} \Phi(x_k,u_k,\Sigma_k) \\ \text{s.t.} \begin{cases} \forall k \in \{0,\ldots,N-1\}, \\ x_{k+1} = f(x_k,u_k), \;\; x_0 = \hat{x} \\ \Sigma_{k+1} = F(x_k,u_k,\Sigma_k), \;\; \Sigma_0 = \hat{\Sigma} \\ c(x_k,u_k) \leq 0 \; . \end{cases} \tag{5}$$

Notice that the function E is more expensive to evaluate than L, as the additional matrix-valued state Σ needs to be propagated forward in time. Next, the original OED based MPC problem (3) can be written in the equivalent consensus form

$$\min_{u_L,u_E} \left\{ L(u_L,\hat{x}) + E(u_E,\hat{x},\hat{\Sigma}) \right\} \quad \text{s.t.} \quad u_L = u_E \; . \tag{6}$$

The above formulation reveals a parallelizable structure, which suggests to use parallelizable control algorithms. Notice that there exists mature MPC software (see Houska et.al. (2011)), as well as OED algorithms and softwares, see, e.g., Bauer et.al. (2000). Thus, it is desirable to make use of these existing software packages by developing a splitting scheme that exploits the structure of (6). However, OED problems are (at least without further reformulation) typically non-convex (Pukelsheim (1993)). Consequently, convex splitting methods such as the alternating direction method of multipliers (ADMM) cannot be applied. Therefore, the following algorithmic developments are based on the ALADIN method (Houska et.al. (2016)), as outlined next.

3.1 ALADIN as an OED-MPC splitting algorithm

By applying ALADIN to (6), a distributed algorithm is obtained that solves decoupled OED and MPC problems in parallel. Algorithm 1 presents the main algorithmic steps for a real-time variant of ALADIN, which, in contrast to the standard ALADIN algorithm, solves multiple instances of the MPC problem at a higher frequency in order to give fast feedback. The schedule of the two processors from Algorithm 1 is additionally visualized in Figure 1 highlighting that Processor 1 and Processor 2 are running their main steps in parallel. Problem (9) remains an MPC optimization problem in standard form, but augmented albeit with the state and terminal cost. Thus, it can be solved with standard MPC solvers, where the additional linear terms as well as the additional least squares term hardly introduce any additional numerical

Algorithm 1: Alternating OED-MPC

Initialization:

Initial guesses for z and λ, positive definite scaling matrices H_L and H_E, and a tuning parameter $\rho > 0$.

Schedule of Processor 1:

Step 1a (W):

Wait for $\hat{x}$ and $\hat{\Sigma}$ as sent by Processor 2.

Step 1b (OED):

Solve the augmented OED problem

$$\min_{u_E} E(u_E,\hat{x},\hat{\Sigma}) + \lambda^{\mathsf{T}} u_E + \frac{\rho}{2} \left\| u_E - z \right\|_2^2 \; . \tag{7}$$

and set $g_E = \rho(z - u_E) - \lambda$.

Step 1c (W):

Wait for the next update of u_L as sent by Processor 2 and set $g_L = \rho(z - u_L) + \lambda$.

Step 1d (c-QP):

Solve the consensus QP

$$\min_{\Delta u_E,\Delta u_L} \; \frac{1}{2}\Delta u_L^\top H_L \Delta u_L + \frac{1}{2}\Delta u_E^\top H_E \Delta u_E + g_L^\top \Delta u_L \\ + g_E^\top \Delta u_E + \lambda^\top \Delta u_L \tag{8} \\ \text{s.t.} \quad u_L + \Delta u_L = u_E - \Delta u_E \quad \mid \lambda^{\mathrm{QP}} \; ,$$

set $z = u_L + \Delta u_L$ and $\lambda = \lambda^{\mathrm{QP}}$, send (z,λ) to Processor 2, and go back to Step 1a.

Schedule of Processor 2:

Step 2a (EKF):

Wait for the measurement η and update the EKF in order to compute a new state estimate $\hat{x}$ and an associated variance $\hat{\Sigma}$.

Step 2b (MPC):

If an update for z and λ (from Processor 1) is available, update these variable. Next, solve

$$\min_{u_L} L(u_L,\hat{x}) - \lambda^{\mathsf{T}} u_L + \frac{\rho}{2} \left\| u_L - z \right\|_2^2 \; , \tag{9}$$

send the first element of the optimal control $u_0 = (u_L)_0$ to the real process, send u_L to Processor 1, and continue with Step 2a.

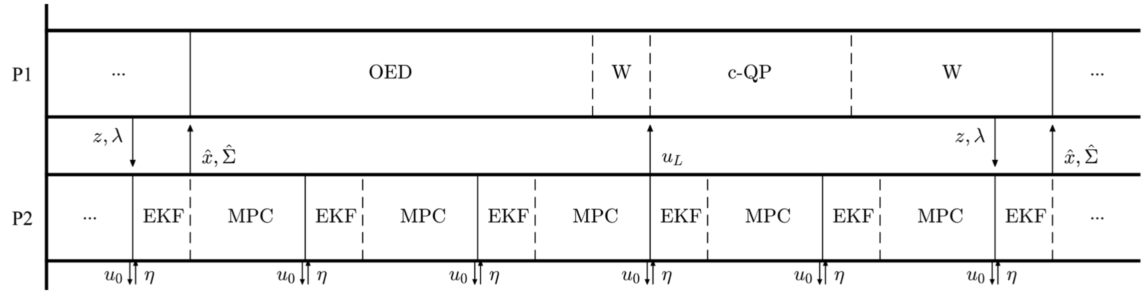

Fig. 1. Schedule for the main steps of Algorithm 1. Processor 1 and Processor 2 run in parallel.

difficulties. Consequently, Processor 2 can run with the same sampling time as a standard MPC controller for the same system, which should be considered as one of the main advantages of Algorithm 1 compared to other methods, which solve problem (3) directly.

Algorithm 1 can be interpreted as a tailored variant of the generic ALADIN algorithm that has been proposed in (Houska et.al. (2016)). Consequently, the local contraction proof of the ALADIN iterates from Houska et.al. (2016) can be applied one-to-one, thus the iterates of Algorithm 1 contract in a local neighborhood of a primal dual solution (u^*, λ^*) solution of (6), i.e., as long as $(z, \lambda) \approx (u^*, \lambda^*)$, and under the assumption that there is no noise present (see also Feng and Houska (2016)). In this paper, the matrices H_L and H_E are considered as scaling matrices, which can in practice be pre-computed by using a pre-conditioner, and ρ is kept constant. More details regarding how to choose the scaling and tuning factors of ALADIN can be found in Houska et.al. (2016). Of course, in practice, Algorithm 1 has to be run in the presence of process noise, but an empirical observation is that Algorithm 1 still controls the system reasonably well in this case, although a more detailed stability and robustness analysis of this empirical observation is beyond the scope of this paper.

Remark 3. As mentioned above, the run-time complexity of most solvers that can deal with problem (9) is of order $\mathbf{O}(Nn_x^3)$, as this problem has the same size as standard MPC problems. On the other hand, solving problem (7) with a generic solver usually leads to an algorithm with run-time complexity $\mathbf{O}(Nn_x^6)$ as the Kalman filter states are matrix-valued. However, it turns out that one can develop tailored algorithm that can solve this problem with run-time complexity $\mathbf{O}(Nn_x^3)$, too, by exploiting the particular structure of the algebraic Riccati recursion (Bittanti et.al. (1991)), as for example discussed by Telen et.al. (2013) or also, in another variant, in Feng and Houska (2016). If such advanced OED solvers are used, the OED solver takes in practice still longer than the MPC solver, but the run-time ratio between the two solvers does not depend on the number of states.

Remark 4. One could also imagine real-time variants of Algorithm 1. E.g., instead of solving the decoupled augmented OED and MPC problems to optimality one could apply one SQP step per iteration following the classical real-time scheme as pioneered by Diehl et.al. (2002) in order to solve the decoupled NLPs only approximately.

Remark 5. Notice that the EKF may be inaccurate for nonlinear system in the presence of large noise, but the proposed parallelizable algorithm can also be in combination with the sigma-point approach (see Kawohl (2007)), which approximates Σ more accurately.

4. NUMERICAL RESULTS

4.1 Dynamic system model

We consider a controlled chemical reactor with given dynamics

$$\forall t \in [0, T], \quad \dot{\chi}(t) = g(\chi(t), \nu(t)), \quad \chi(0) = x\,, \tag{10}$$

where its right-hand expression is given by

$$g(\chi, \nu) = \begin{pmatrix} -(D + k_1)\chi_1 + k_2\chi_2\chi_3 + \nu_1 \\ -D\chi_2 - k_2\chi_2\chi_3 + k_1\chi_1 + \nu_2 \\ -D\chi_3 - k_2\chi_2\chi_3 + \nu_3 \end{pmatrix}.$$

Here, χ $[\frac{\mathrm{g}}{\mathrm{L}}]$ denotes concentrations of three substances, ν $[\frac{\mathrm{g}}{\mathrm{L \cdot s}}]$ the feeding rates, k_1 and k_2 $[\frac{1}{\mathrm{s}}]$ the reaction rates, and D $[\frac{1}{\mathrm{s}}]$ the dilution rate. In the following, $f(x, u) = \chi(\tau)$ denotes the solution of the associated ODE for a small step-size $\tau > 0$ and a piecewise constant control input. In this case study, a Runge-Kutta integrator of order 4 is used to evaluate the function f with high numerical accuracy. Moreover, the stage cost l, the Mayer term m and the constraint c are given by

$$l(x_k, u_k) = \frac{1}{2}\|x_k - x_{\mathrm{ref}}\|_Q^2 + \frac{1}{2}\|u_k - u_{\mathrm{ref}}\|_R^2\,,$$

$$m(x_N) = \frac{1}{2}\|x_N - x_{\mathrm{ref}}\|_{Q_N}^2\,,$$

as well as $c(x_k, u_k) = \underline{u} - u_k$. Here, we assume that only the first state can be measured, $h(x) = x_1$. The numerical values for the MPC horizon length, references, and other parameters can be found in Table 1.

4.2 Implementation details

Algorithm 1 for this particular case study has been implemented by using the algorithmic differentiation software `CasADi` (Andersson et.al. (2012)) in combination with automatic C-code generation. Moreover, we use the software `qpOASES` as a QP solver (Ferreau et.al. (2008)). All the results below are obtained on a macOS Sierra operating system with two 3.3 GHz Intel Core i7 processors and 16 GB, 2133 MHz LPDDR3.

4.3 Control performance

Figure 2 shows a comparison of the closed-loop trajectories that are obtained by running Algorithm 1 and certainty-

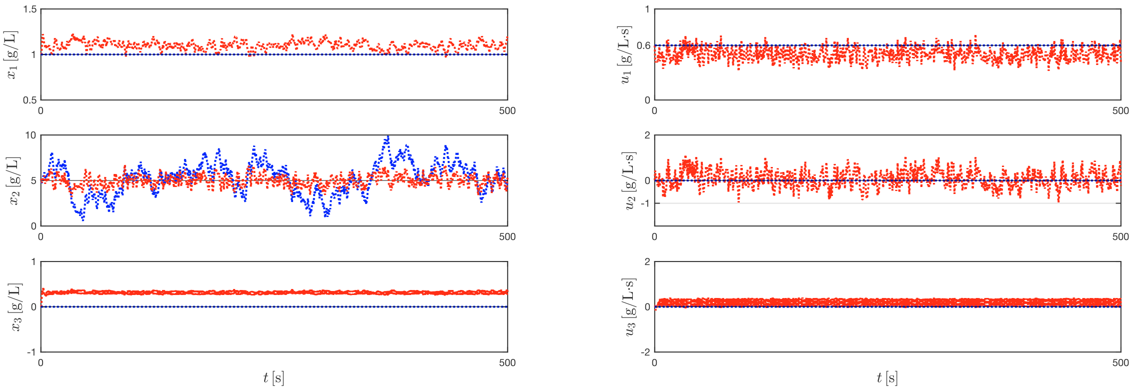

Fig. 2. [Discrete-time system] Closed-loop state trajectories x with random noise as well as the associated closed-loop input profile u obtained by running the alternating OED-MPC Algorithm 1 (red, dotted), by running the certainty-equivalent MPC (blue, dotted), and the reference states x_{ref} and controls u_{ref} (black, solid).

Name	Symbol	Value
dilution rate	D	$0.1\ [\frac{1}{\mathrm{s}}]$
reaction rates	$k_1,\ k_2$	$0.1,\ 0.5\ [\frac{1}{\mathrm{s}}]$
discrete-time step-size	τ	0.5 [s]
MPC horizon length	N	10
initial state estimate	$\hat{y}$	$[1.0,\ 5.0,\ 0.0]^\mathsf{T}\ [\frac{\mathrm{g}}{\mathrm{L}}]$
initial state variance	$\widehat{\Sigma}$	$\mathrm{diag}\,(0,\ 0,\ 0)\ [\frac{\mathrm{g}^2}{\mathrm{L}^2}]$
state reference	x_{ref}	$[1.0,\ 5.0,\ 0.0]^\mathsf{T}\ [\frac{\mathrm{g}}{\mathrm{L}}]$
control reference	u_{ref}	$[0.6,\ 0.0,\ 0.0]^\mathsf{T}\ [\frac{\mathrm{g}}{\mathrm{L\cdot s}}]$
lower control bound	$\underline{u}$	$[0,\ -1,\ -1]^\mathsf{T}\ [\frac{\mathrm{g}}{\mathrm{L\cdot s}}]$
measurement error variance	V	$2.5 \cdot 10^{-4}\ [\frac{\mathrm{g}^2}{\mathrm{L}^2}]$
process noise variance	W	$\mathrm{diag}\,(0,\ 2.56,\ 0)\ [\frac{\mathrm{g}^2}{\mathrm{L}^2}]$
state weighting matrix	$Q,\ Q_N$	$\mathrm{diag}\,(1,\ 1,\ 1)$
control weighting matrix	R	$\mathrm{diag}\,(1,\ 1,\ 5)\ [\mathrm{s}^2]$
penalty term	ρ	10^3

Table 1. Parameter values.

equivalent MPC (solely running the control solver without any assistance from the OED solver) on the above case study, respectively. Here, the function Φ is a weighted trace term using the self-reflective weighted A-criterion as proposed in Houska et.al. (2017).

Notice that for this particular case study the system is not observable at its steady states x_{ref}, since only the first state component of the system can be measured. Consequently, the controller finds an optimal tradeoff between estimation accuracy and tracking performance by steering the system to a steady-state that is observable, but not exactly equal to x_{ref}, as one would expect from an integrated OED based MPC controller. Their average control performance illustrates the difference between two controllers. By performing the closed-loop simulation for a sufficiently long time M, the average performance of the certainty-equivalent MPC controller is

$$\frac{1}{M}\sum_{k=1}^{M} l(x_i^{\mathrm{MPC}}, u_i^{\mathrm{MPC}}) \approx 1.62 \left[\frac{\mathrm{g}^2}{\mathrm{L}^2}\right]$$

and the corresponding value of performing Algorithm 1 is

$$\frac{1}{M}\sum_{k=1}^{M} l(x_i^{\mathrm{OED\text{-}MPC}}, u_i^{\mathrm{OED\text{-}MPC}}) \approx 0.53 \left[\frac{\mathrm{g}^2}{\mathrm{L}^2}\right]$$

for randomly generated uniformly distributed uncertainty. Consequently, the proposed integrated OED-MPC controller achieves over three times better performance than certainty-equivalent MPC.

4.4 Run-time performance

Although the weighted A-criterion comprises 18 states and 3 controls in total for this particular case, the runtime is in the microsecond range by combining the proposed parallelizable algorithm with the acceleration scheme from Feng and Houska (2016), which exploits the particular structure of integrated experiment design problem.

Processor 1 (OED processor)	CPU time [µs]	%
Step 1a (W)	–	–
Step 1b (OED)	40	42
Step 1c (W)	–	–
Step 1d (c-QP)	55	58
Total time	95	100
Processor 2 (control processor)	CPU time [µs]	%
Step 2a (EKF)	≤ 1	≈ 4
Step 2b (MPC)	24	96
Total time	25	100

Table 2. Run-time associated with the different steps of Algorithm 1 for both processors.

Table 2 summarizes the run-time of different steps of Algorithm 1. The decoupled problems (7) and (9) are solved by using real-time iterations as discussed in Remark 4. The time for solving both the OED and the coupled QP is about 95µs, which corresponds to around four times of the sampling time of the certainty-equivalent MPC loop. The waiting times in Step 1a and 1c were set to 0.

5. CONCLUSION

This paper has presented the alternating MPC-OED Algorithm 1, which finds optimal tradeoffs between OED criteria and MPC control performance objectives. The main contribution of this tailored algorithm is that it uses two processors: Processor 2 runs at the same sampling rate as standard MPC without additional OED objectives would do. Processor 1 is used to support the decisions of Processor 2 at a lower sampling rate by perturbing the gradient of the MPC controller in order to improve the accuracy of the state estimates in the presence of random process and measurement noise. A case study has illustrated the practical advantages of this algorithm in terms of both run-time as well as control performance. A more detailed robustness and stability analysis supporting the algorithmic developments will be part of future work.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support by the National Science Foundation China (NSFC), Nr. 61473185, as well as ShanghaiTech University, Grant-Nr. F-0203-14-012.

REFERENCES

Andersson, J., Akesson, J., and Diehl, M. (2012). CasADi: A Symbolic Package for Automatic Differentiation and Optimal Control. *Recent Advances in Algorithmic Differentiation*, 297–307, Springer, 2012.

Bauer, I., Bock, H.G., Körkel, S., and Schlöder, J.P. (2000). Numerical methods for optimum experimental design in DAE systems. *Journal of Computational and Applied Mathematics*, 120: 1–25, 2000.

Bittanti, S., Colaneri, P., and De Nicolao, G. (1991). *The periodic Riccati equation.* In *Willems Bittanti*, Laub, editor, The Riccati Equation. Springer, 1991.

Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4): 577–585, 2002.

Diehl, M., Ferreau, H.J., and Haverbeke, N. (2009). Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation, *Nonlinear Model Predictive Control*, 384: 391–417, 2009.

Feng, X. and Houska, B. (2016). Real-time algorithm for self-reflective model predictive control. *Journal of Process Control*, 2017, `https://doi.org/10.1016/j.jprocont.2017.10.003`.

Fisher, R. (1935). *The design of experiments.* Oliver & Boyd, 1935.

Ferreau, H.J., Bock, H.G., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

Heirung, T.A.N., Ydstie, B.E., and Foss, B. (2012). Towards Dual MPC. *In Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*, 45(17): 502–507, 2012.

Heirung, T.A.N., Foss, B., and Ydstie, B.E. (2015). MPC-based dual control with online experiment design. *Journal of Process Control*, 32: 64–76, 2015.

Hernandez, B. and Trodden, P. (2015). Persistently exciting tube MPC. *American Control Conference (ACC)*, IEEE, 2016.

Hovd, M. and Bitmead, R.R. (2005). Interaction between control and state estimation in nonlinear MPC. *Modeling, Identification, and Control*, 26(3):165–174, 2005.

Houska, B., Ferreau, H.J., and Diehl, M. (2011). An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 47(10): 2279-2285, 2011.

Houska, B., Frasch, J., and Diehl, M. (2016). An Augmented Lagrangian Based Algorithm for Distributed Non-Convex Optimization. *SIAM Journal on Optimization*, 26(2):1101–1127, 2016.

Houska, B., Telen, D., Logist, F., and Van Impe, J. (2017). Self-reflective model predictive control. *SIAM Journal on Control and Optimization*, 55(5): 2959–2980, 2017.

Kawohl, M., Heine, T., and King, R. (2007). A new approach for robust model predictive control of biological production processes. *Chemical Engineering Science*, 62(18–20): 5212-5215, 2007.

Ljung, L. (1999). *System Identification: Theory for the User.* Prentice Hall, 1999.

Marafioti, G., Bitmead, R.R., and Hovd, M. (2014). Persistently exciting model predictive control. *International Journal of Adaptive Control and Signal Processing*, 28(6): 536–552, 2014.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.Q. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6): 789–814, 2000.

Mesbah, A., Bombois, X., Forgione, M., Hjalmarsson, H., and Van den Hof, P.M.J. (2015). Least costly closed-loop performance diagnosis and plant re-identification. *International Journal of Control*, 88(11): 2264–2276, 2015.

Pukelsheim, F. (1993). *Optimal design of Experiments.* John Wiley & Sons, Inc., New York, 1993.

Rao, C.V., Rawlings, J.B., and Lee, J.H. (2001). *Constrained linear state estimation - a moving horizon approach* Automatica, 37:1619–1628, 2001.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design.* Madison, WI: Nob Hill Publishing, 2009.

Stengel, R. (1994). *Optimal Control and Estimation.* Dover Publications, New York, 1994.

Telen, D., Houska, B., Logist, F., Van Derlinden, E., Diehl, M., and Van Impe, J. (2013). Optimal Experiment Design under Process Noise. *Journal of Process Control*, 23: 613–629, 2013.

Telen, D., Houska, B., Vallerio, M., Logist, F., and Van Impe, J. (2016). A study of integrated experiment design for NMPC applied to the Droop model. *Chemical Engineering Science*, 160: 370–383, 2016.

Yan, J. and Bitmead, R.R. (2005). Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41(4): 595–604, 2005.

Zacekova, E., Privara, S., and Pcolka, M. (2013). Persistent excitation condition within the dual control framework. *Journal of Process Control*, 23(9): 1270–1280, 2013.

Zhu, Y. (2009). System identification for process control: recent experience and outlook. *International Journal of Modelling*, Identification and Control, 6(2): 89–103, 2009.