Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Process Control 15 (2005) 183–200

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

Application of barrier function based model predictive control to an edible oil refining process

Adrian G. Wills [a,*], William P. Heath [b]

[a] School of Electrical Engineering and Computer Science, University of Newcastle, University Drive, Callaghan 2308, Australia
[b] Centre for Complex Dynamic Systems and Control, University of Newcastle, Australia

Received 3 August 2003; received in revised form 13 January 2004; accepted 13 May 2004

Abstract

We use multivariable feedback control to reduce variations in flow and back-pressure associated with two industrial separator units forming part of an edible oil refining line. This provides plant operators with a mechanism for increasing process yield without reducing product quality. Actuator limits are a natural consideration in the controller design since increased production rates require actuators to operate on or near their limits. We use recentred barrier function model predictive control in this application and obtain favourable results.

Keywords: Model predictive control; Barrier functions; Edible oil refining

1. Introduction

We describe the use of multivariable feedback control to regulate the inlet flow rate and outlet pressure of two separator units in the edible oil refining line—Fig. 1 shows a simplified process flow diagram for the refining line in question. Feedback control reduces variations in internal separator conditions which increases operator authority over process yield and product quality. In particular, operators can afford to be less conservative when setting certain parameters related to process yield without sacrificing product quality. This is a classic example of the well known benefits of feedback control—see e.g. Section 1.6 of [20]. Fig. 2 shows a conceptual view of the quality-yield trade-off and the implications of reducing process variation via feedback control.

Three process variables (PVs) were nominated for control: inlet flow to the first separator, and outlet pressure of each separator unit (see Fig. 1). These PVs are strongly influenced by three manipulated variables (MVs) namely automatic valves AV1, AV2 and AV3. In fact, the latter two automated valves were installed as part of a plant reconfiguration recommended by the authors. Manual valves (MV1 and MV2) were used in their place prior to reconfiguration.

Actuator limits were a natural consideration since it was known that higher production rates required the valves to operate on or near their maximum levels.

We chose to implement *recentred barrier function*-Model Predictive Control (abbreviated as r-MPC) to the associated control problem. The r-MPC strategy was developed by the authors in [21,22,24]. Briefly, r-MPC takes the shape of standard MPC but with a modified cost that includes a weighted barrier function; this guarantees that hard constraints are satisfied if indeed possible. The weighting on the barrier term is typically denoted by the positive scalar μ. Increasing μ has the effect of penalising points near the constraint boundary more than points away from the boundary. In effect, the controller action becomes cautious near constraint boundaries while being relatively unaffected when away from the boundary. In order to guarantee correct steady-state behaviour it is necessary to use a particular choice of barrier function. For this reason we use the so-called *gradient recentred barrier function* which is a strictly convex and smooth function whose gradient is zero at the desired steady-state set-point.

* Corresponding author. Tel.: +61-2-4921-5204; fax: +61-2-4921-6993.
E-mail addresses: onyx@ee.newcastle.edu.au (A.G. Wills), wheath@ee.newcastle.edu.au (W.P. Heath).

0959-1524/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2004.05.004

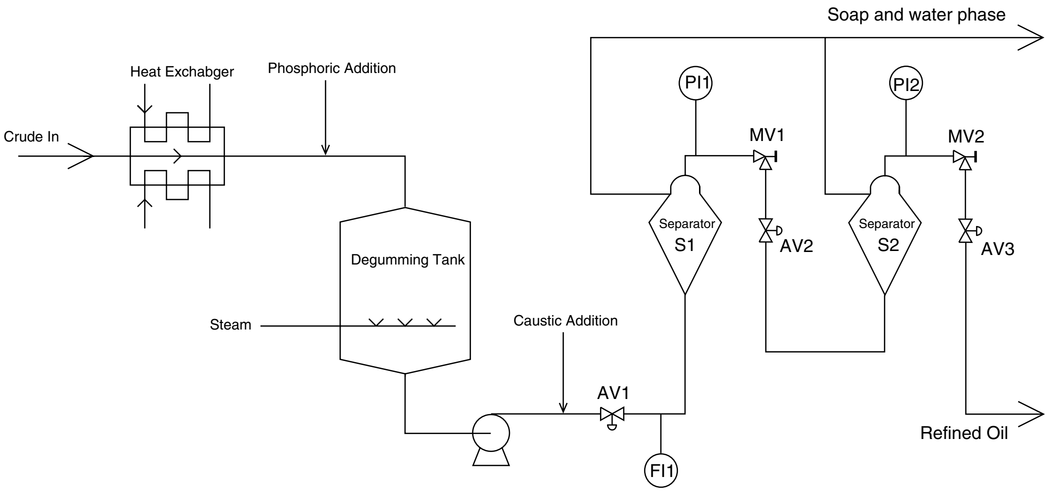


Fig. 1. Simplified flow diagram of a typical edible oil alkali refining process.

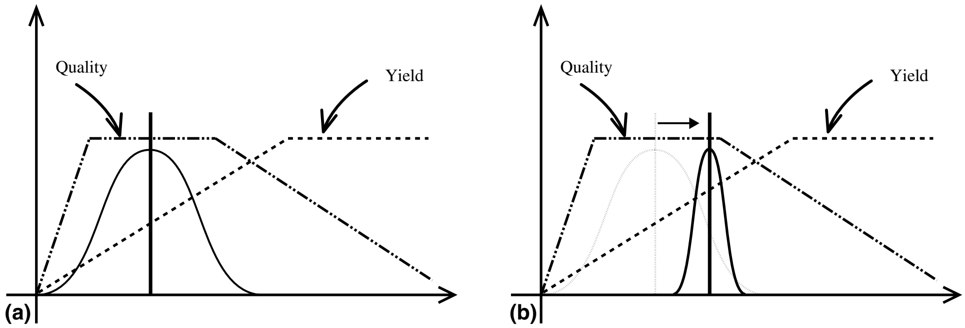


Fig. 2. Conceptual view of quality and yield trade-off. Reduced variation allows operating point closer to threshold. Feedback control can reduce variation. (a) (Open loop): This figure depicts a quality-yield curve for a plant with relatively high process variation. (b) (Closed loop): This figure depicts the same quality-yield curve with reduced process variation. Note that higher yields can be obtained without risking product quality.

Our intentions in this paper are not necessarily to advocate r-MPC as the most appropriate multivariable control strategy for this particular problem, but rather to validate this approach in an industrial setting. In fact the resulting control problem may be considered relatively straightforward; nevertheless trial results on the plant using r-MPC appear highly promising. Using the barrier weighting parameter μ as a tuning parameter to adjust control action near constraint boundaries was shown to be successful.

The remainder of this paper is structured as follows. Section 2 presents background material on edible oil refining and separation theory. Particular attention is given to separator stabilisation which is the primary beneficiary of multivariable feedback control. The control problem is described with more detail in this section. Section 3 discusses a black-box model of the system as used for control purposes. This model was obtained using standard system identification techniques. The r-MPC details are presented in Section 4. State estimation, integral action, steady-state reference calculation and computing the next control move are treated in this section. A brief discussion of tuning rules is also found in this section. Results from controlling the plant are given in Section 5 along with some practical considerations such as controller start-up and separator self-cleaning. This provides a more complete picture of the refining plant and issues involved in proper control of separator conditions. Section 6 concludes the paper.

2. Problem background

The objective of edible oil refining is to remove (or at least reduce to tolerable levels) impurities in the crude vegetable oil. Impurities are present in various forms but the major concern when refining is to reduce Free-Fatty-Acid (FFA) and Phospholipid levels. These components are directly related to oil quality and shelf-life—see e.g. [10].

A typical alkali refining line treats crude edible vegetable oil and animal fats with an alkali solution such as caustic soda. The alkali solution combines with FFAs in the oil and forms soap and water (see e.g. [10] for more details). The resulting soap/neutral oil mixture is usually subjected to some form of separation in order to remove the soap component and thus produce *refined oil*, i.e. oil with acceptable quality—in particular sufficiently small levels of FFAs.

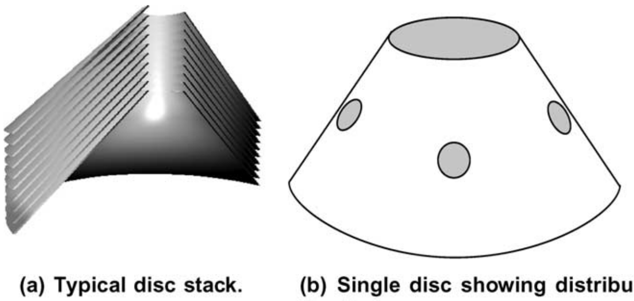
Fig. 1 presents a simplified flow diagram for the actual industrial plant in question. This particular plant incorporates another important step known as degumming. Degumming removes (or reduces) the phospholipid content. Since both alkali refining and degumming require a separation unit, these processes are combined. Notice that the flow diagram includes a second separation unit. This is common to most alkali refining plants and is often termed the water wash stage; water is added between the two separators to remove any soap remaining from the first separation stage.

The actual separators used are an Alfa-Laval disc-stack centrifuge. Fig. 3(a) shows an idealised cross-sectional view of a typical disc-stack while Fig. 3(b) shows one full disc illustrating the distribution holes common to each disc. These holes are aligned throughout the whole stack and provide passage for the inlet stream. Fig. 4 shows a conceptual view of a disc-stack centrifuge in operation.

Following various discussions with plant operators and engineers, investigations on different levels and consideration of relevant strategic factors, it was conjectured that one possible area for improvement concerned the stabilisation of separator conditions. In particular, inlet flow and light-phase outlet pressures for both separators (see Fig. 4) should be properly regulated to ensure optimal separation of soap from oil.

2.1. Separation theory and control objectives

The ensuing discussion is partially based on the booklet produced by Alfa-Laval called '*Theory of separation*' [2]. The centrifuge inlet stream comprises a mixture of oil and soapy water. The objective of separation is essentially to split the mixture into two streams; a neutral oil stream—containing a reduced amount of impurities, and the effluent stream—soapy water and other material containing as little neutral oil as possible.



(a) Typical disc stack.

(b) Single disc showing distribution holes.

Fig. 3. Cross-section of a typical disc-stack. (a) Typical disc-stack. (b) Single disc showing distribution holes.

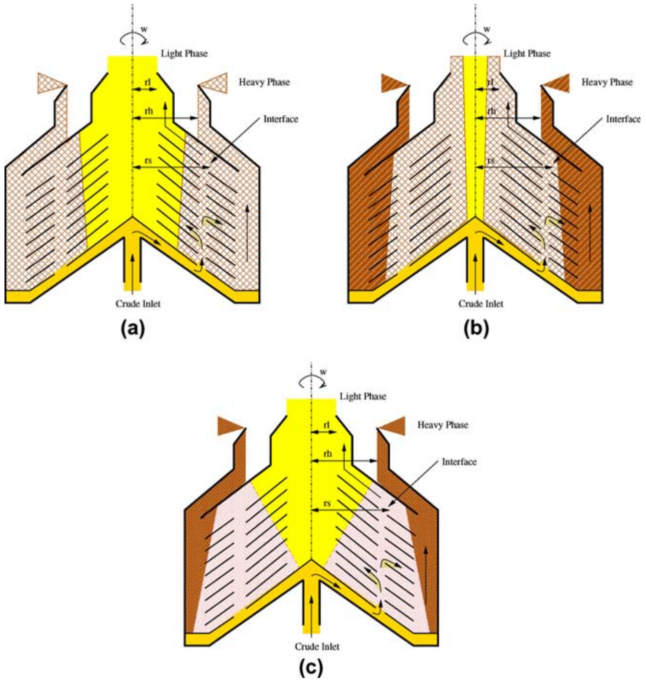


Fig. 4. Separator operation for different light phase outlet back-pressures. (a) The interface position with excessive back-pressure—notice that the light phase is pushed to the bowl periphery which may result in excessive neutral oil loss. (b) separator interface position corresponding to insufficient back-pressure—notice that the soap and water phase are not properly split which ultimately results in poor quality oil. (c) Separator with optimal interface position, i.e. in line with the distribution holes.

Stokes' Law (see e.g. Section 25.1 in [4]) may be used to derive Eq. (1), which describes the separation velocity v_g (m/s) as a function of particle size d (m), particle density ρ_p, oil density ρ_l (kg/m^3), viscosity η (kg/ms) and gravity g (m/s^2):

$$v_g = \frac{d^2(\rho_p - \rho_l)}{18\eta} g. \tag{1}$$

Generally speaking, if a liquid solution contains two primary components distinguishable by their respective densities, then a centrifugal separator may be employed to split the solution; this effectively increases g in 1. Note that oil has a lower density than soapy water and consequently the oil stream will be referred to as the light phase while the soapy water stream will be referred to as the heavy phase. The separator user's manual [1] has the following to say about achieving good separation:

> *Interface position, Liquid Seal*: When process liquid is supplied, an interface should be formed at a certain radius from the bowl centre. Practically all liquid inside this interface should consist of light phase, and all liquid outside of it heavy phase

> (liquid seal). The best separating result is obtained when the interface lies at or near the distribution holes (D2) of the bowl discs. However, the interface will not locate in this place automatically, without trimming, but must be brought into it by regulation of the feed and of the discharge of light phase, and the heavy phase respectively in the phase outlets. A reduction of the light phase discharge (increase of back-pressure p_l) will force the interface towards the bowl periphery. If the back-pressure is so high that the interface locates at D3, no separation takes place...

It is clear from this statement that in order to place the interface in the optimal position, at or near the distribution holes as shown in Fig. 4, the feed (via a flow regulation valve) and light phase discharge (via light phase back-pressure) must be regulated. The heavy phase discharge is open to the atmosphere and therefore considered to be static. This is also observed from the following expression relating the pressure at the interface, p_s, angular velocity ω, light phase density ρ_l, heavy phase density ρ_h, radius of light phase discharge r_l, radius of heavy phase discharge r_h and radius of the interface r_s—as measured from the axis of rotation,

$$p_s = \frac{\omega^2}{2}\rho_l(r_s^2 - r_l^2) = \frac{\omega^2}{2}\rho_h(r_h^2 - r_s^2), \tag{2}$$

so

$$\rho_l(r_s^2 - r_l^2) = \rho_h(r_h^2 - r_s^2). \tag{3}$$

Therefore, the control objective may be stated as: regulate inlet flow rate to the first separator (shown as S1 in Fig. 1) and light phase outlet pressure for both separators (see Fig. 1). Fig. 5 shows a block diagram of the three input three output system.

Remark 2.1. There are two manual valves labelled MV1 and MV2 in Fig. 1 which lie in series with the two automated valves AV2 and AV3 respectively. The latter pair were introduced as part of a plant reconfiguration suggested by the authors. This was to provide sufficient actuation of the light phase outlet pressure of each separator. The series connection of the two separator units poses some interesting challenges for control as discussed further in Section 5.2.

Although crude oil is filtered prior to separation, there still exists certain particulate matter (e.g. dirt and gums) that collects at the bowl periphery. These materials accumulate and consequently change the separator dynamics, ultimately causing the separator to foul due to excessive sludge. This problem is overcome by periodically opening the bowl for a very short duration to let this sludge escape; this is called a separator self-clean. Although the duration is short—in the order of milliseconds—roughly 40% of the bowl volume is lost during this operation. The result is a rather large disturbance to the separator dynamics. This issue is discussed further in Section 5.3.

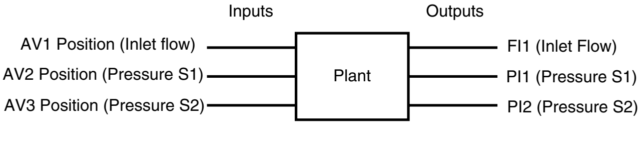

Inputs
Outputs
AV1 Position (Inlet flow)
AV2 Position (Pressure S1)
AV3 Position (Pressure S2)
Plant
FI1 (Inlet Flow)
PI1 (Pressure S1)
PI2 (Pressure S2)

Fig. 5. System inputs and outputs for control purposes.

To summarise, a complex mixture of oil, water, soap and other materials enters the separator. By utilising the difference in density between oil and soap and by increasing gravitational effect, two streams are produced; oil (light phase) and soap (heavy phase). In order to obtain the optimal split between these phases (neglecting chemical reactions and physical conditions prior to separation), the interface must be positioned at or near the distribution holes in the discs. The process variables that affect interface position are feed flow-rate and light phase back-pressure. Possible disturbances to the separator dynamics include: changes in feed flow-rate, changes in back-pressure of light phase, changes to viscosity of feed, changes to density differential between phases and finally separator self-cleaning.

3. System model

The plant is used to refine different oil types e.g. canola, coconut, sunflower and maize. Further, the plant is run at different operating points depending on initial oil quality.

In this section a three-by-three input–output model of the system is described. This model was obtained via system identification techniques. An alternative approach would be to use physical (or phenomenological) modelling where each unit operation has mass and energy balance equations. Such an approach was rejected at an early stage due to the complexity of the plant (in particular the separator unit) and the many different oil types. Nevertheless, the resulting input–output model agrees with physical intuition as discussed below.

As depicted in Fig. 5, there are three inputs and three outputs. The three inputs correspond to valve plug position (all valves are fully open at 0%). The three outputs correspond to a flow indicator (FI1) and two pressure indicators (PI1 and PI2). The system is open-loop stable under nominal operating conditions.

Assume a model with the following discrete time linear transfer function (or transfer operator, see e.g. [13]) matrix structure,

$$y(t) = G(q)u(t),$$

where q denotes the forward shift operator, i.e. $qy(t) = y(t+1)$ and $q^{-1}y(t) = y(t-1)$. The vectors y and u and matrix G are given by

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix},$$

$$G(q) = \begin{bmatrix} g_{11}(q) & g_{12}(q) & g_{13}(q) \\ g_{21}(q) & g_{22}(q) & g_{23}(q) \\ g_{31}(q) & g_{32}(q) & g_{33}(q) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix}.$$

Let u_1 denote the flow valve AV1 plug position, u_2 denote the pressure valve AV2 plug position and u_3 denote the pressure valve AV3 plug position. Similarly, let y_1 denote the flow measurement from FI1, y_2 denote the pressure measurement from PI1 and y_3 denote the pressure measurement from PI2.

Nominal operating conditions change dramatically for each oil type (the plant is capable of treating one of sixteen different oil types) and even for the same oil type with different crude properties. For example, refining canola oil with 0.5% FFA content is relatively simple compared with say a coconut oil that has 3.5% FFA content. Due to the nature of the oil and the higher FFA levels, more alkali solution is required for the coconut example then for the canola example. This means that more soap is produced and proper separation becomes both harder to achieve and more important in this case. This generally necessitates slower feed-rates which allow longer residence time in the separator and hence increase the chance of removing heavy phase material. It should be noted that some dynamical behaviour was observed to be common to more than one oil type. In particular, the model for one oil type was observed to be applicable for controlling a number of different oil types.

The following phenomena were observed and are well known to plant operators. All tests were performed on canola oil. If the system is stable and in open-loop, then some (rough) predictions can be made regarding the output response to various input changes. Consider a small step perturbation in the flow valve position u_1 (AV1). If the valve is closed by a small percentage then one might expect a decrease in flow rate and a consequential sharp fall in light phase outlet pressure (of both separators) due to the increased pressure drop across valve AV1. This behaviour was observed in practice.

Less intuitive, however, is the slow dynamic response observed after such a perturbation. It was conjectured that this slower response is due to separator dynamics—wherein the interface position finds a new equilibrium point and the soap/oil split re-establishes.

Using a Pseudo-Random-Binary-Sequence (PRBS, see e.g. [13]) to perturb the flow valve position u_1, an input–output model was constructed using standard system identification techniques for each output (step responses are shown in Fig. 6). All discrete time models are based on a one second sample interval. The model is given by

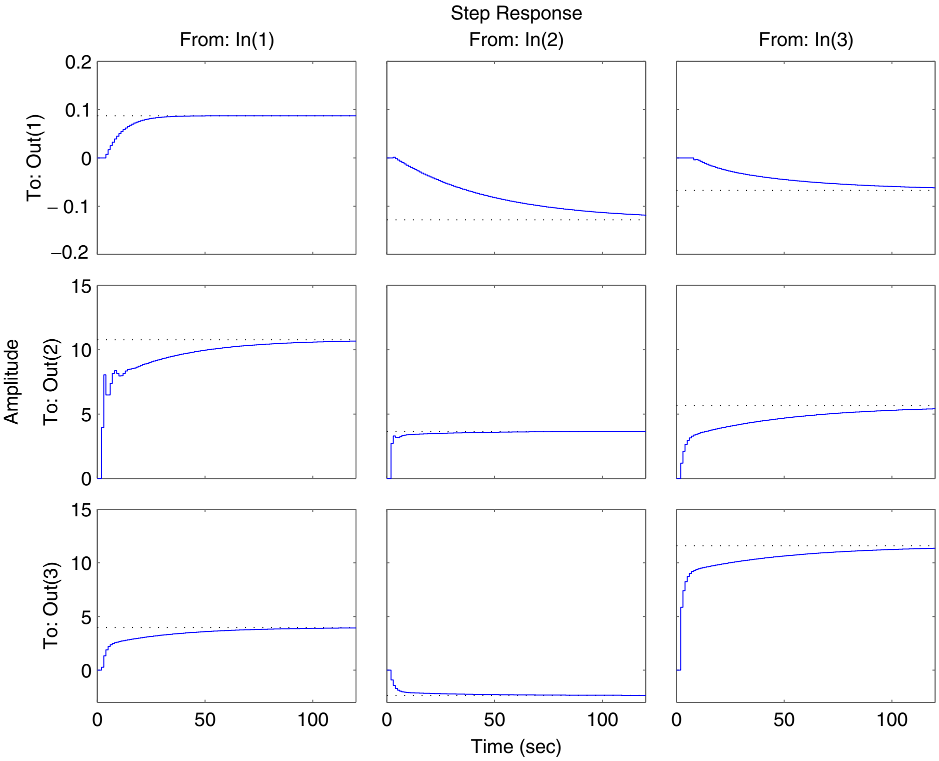


Fig. 6. Input/output step responses for 3×3 system.

$$g_{11}(q) = \frac{0.0070q^{-4} + 0.0035q^{-5}}{1 - 0.88q^{-1}},$$

$$g_{12}(q) = \frac{3.9q^{-2} - 2.9q^{-3} - 3.8q^{-4} + 6.2q^{-5} - 3.2q^{-6}}{1 - 1.8q^{-1} + 1.3q^{-2} - 0.49q^{-3}},$$

$$g_{13}(q) = \frac{0.29q^{-2} + 0.65q^{-3} - 0.88q^{-4}}{1 - 1.4q^{-1} + 0.46q^{-2}}.$$

A similar procedure was repeated for the second input (pressure valve AV2 in Fig. 1). Closing this valve by a small percentage should result in an increased pressure at the light phase outlet of the first separator and a consequential decrease in flow as indicated by FI1. The pressure response at the second separator light phase outlet should decrease due to the decrease in flow rate. An input–output model was obtained using PRBS experiments similar to before. The model is given by

$$g_{21}(q) = \frac{0.0012q^{-3} - 0.004q^{-4}}{1 - 0.98q^{-1}},$$

$$g_{22}(q) = \frac{2.7q^{-2} - 4.7q^{-3} + 2.6q^{-4} - 0.6q^{-5}}{1 - 1.9q^{-1} + 1.4q^{-2} - 0.56q^{-3} + 0.11q^{-4}},$$

$$g_{23}(q) = \frac{-0.91q^{-2} + 0.88q^{-3}}{1 - 1.5q^{-1} + 0.54q^{-2}}.$$

Again the above procedure was repeated for the last input (pressure valve AV3 in Fig. 1). Closing this valve by a small percentage should result in an increased pressure at the light phase outlet of the second separator and a consequential decrease in flow as indicated by FI1. The pressure response at the first separator light phase outlet should increase due to the increased pressure at the second separator light phase outlet. An input–output model was obtained using PRBS experiments similar to before. The model is given by

$$g_{31}(q) = \frac{10^{-3}(-4.6q^{-8} + 9.2q^{-9} - 7.1q^{-10} + 1.4q^{-11} + 0.82q^{-12})}{1 - 1.8q^{-1} + 0.83q^{-2}},$$

$$g_{32}(q) = \frac{1.2q^{-2} - 0.86q^{-3} - 0.26q^{-4}}{1 - 1.5q^{-1} + 0.54q^{-2}},$$

$$g_{33}(q) = \frac{5.9q^{-2} - 7.4q^{-3} + 1.6q^{-4}}{1 - 1.5q^{-1} + 0.53q^{-2}}.$$

Observe that both fast and slow dynamics are present (c.f. Fig. 6). The benefit of choosing a small sample interval is that fast dynamics (like pressure) are captured. However, the sample time is rather small with respect to the settling time as governed by the slow dynamics. In Section 4.5 the prediction horizon is chosen to be $N = 25$ which, although less than the settling time, is found to give acceptable results in practice.

The models include relevant time delays from each input to each output as measured from the r-MPC software (see Section 5.4 for more details on the software used). A one second delay was introduced to cater for the delayed controller action—see Section 4.1.

The above 3×3 transfer function matrix was converted into a state-space model with 47 states. This model is omitted for the sake of brevity and is henceforth referenced by the standard system matrices A, B and C.

The linear model discussed above is valid only in a small region of operation. Certain areas of operation are known to have significantly different dynamical behaviour. For example, Section 5.3 discusses a common event called separator self-cleaning. During this event, the system dynamics differ significantly from the model, but perhaps more importantly, the steady-state gain is of opposite sign—see Section 5.3 for more details. Other areas of operation in which the control action is known to be detrimental are considered in Section 5.3.

Input constraints were considered in the form of simple bounds—representing the physical limits imposed by the valve plug position. One of the primary motivations for using r-MPC for this application was the expectation that increased production rates would frequently result in operation near valve limits. Section 5 offers some actual results for operation near a constraint boundary. A more challenging task might be to represent such undesirable areas of operation using state constraints. State (and output) constraints were not considered in the industrial trials.

4. Model predictive control design

In this section we describe the r-MPC algorithm used for industrial trials. We have based the design on a state-space model structure, hence the model given in Section 3 is firstly converted to an equivalent state-space form

$$x(k+1) = Ax(k) + Bu(k), \tag{4}$$

$$y(k) = Cx(k). \tag{5}$$

In the above, $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$ with $n_x = 47$, $n_u = 3$ and $n_y = 3$. This model is used to predict the state trajectory denoted by $\mathbf{x}$ for a given input trajectory denoted by $\mathbf{u}$, where

$$\mathbf{x} = \{x_0, \dots, x_N\}, \quad \mathbf{u} = \{u_0, \dots, u_{N-1}\}.$$

In the above, N is used to denote the prediction horizon. In this paper, we are only ever concerned with the case where $\mathbf{x}$ is a function of $\mathbf{u}$ and an initial state $x_0 = \hat{x}$. Hence, given $x_0 = \hat{x}$ we have

$$x_{i+1} = Ax_i + Bu_i, \quad \text{for } i = 1, \dots, N-1.$$

Only input constraints were considered in this application. In particular, simple bounds corresponding to actuator limits were included for each actuator. This may be expressed as inclusion in the set $\mathbb{U}$ defined as

$$\mathbb{U} = \{u \in \mathbb{R}^{n_u} : \ell_l \preceq u \preceq \ell_u\}.$$

The symbol "$\preceq$" denotes element wise inequality. In this application ℓ_l and ℓ_u are given by

$$\ell_l = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \ell_u = \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}.$$

We use the following MPC cost function to calculate the next control move.

$$J(\mathbf{x},\mathbf{u}) = \frac{1}{2}\|x_N - x_N^{\mathrm{r}}\|_P^2 + \sum_{i=0}^{N-1}\left(\frac{1}{2}\|x_i - x_i^{\mathrm{r}}\|_Q^2 + \frac{1}{2}\|u_i - u_i^{\mathrm{r}}\|_R^2 + \mu B_u(u_i, u_i^{\mathrm{r}})\right).$$

In the above, $\mathbf{x}^{\mathrm{r}}$ and $\mathbf{u}^{\mathrm{r}}$ are state and input reference trajectories. The symmetric matrices Q and R are further assumed to be positive semi-definite and positive definite respectively. The positive semi-definite matrix P is chosen as the solution to the Discrete-time Algebraic Riccati Equation (DARE) corresponding to system 4 and the matrices Q and R (see e.g. [5]). The function B_u is a gradient recentred logarithmic barrier function for the constraint set $\mathbb{U}$ (see [21,22,24]) and μ is a positive weighting parameter for the barrier. The barrier B_u is given as follows. Let $L(u)$ denote the standard logarithmic barrier function for $\mathbb{U}$ given by

$$L(u) = -\sum_{j=1}^{n_u}\ln(\ell_u(j) - u(j)) - \sum_{j=1}^{n_u}\ln(u(j) - \ell_l(j)).$$

Then

$$\begin{aligned} B_u(u,u^{\mathrm{r}}) &= L(u) - L(u^{\mathrm{r}}) - \nabla L^T(u^{\mathrm{r}})(u - u^{\mathrm{r}}), \\ &= L(u) - L(u^{\mathrm{r}}) - \sum_{j=1}^{n^u}\left(\frac{u(j) - u^{\mathrm{r}}(j)}{\ell_u(j) - u(j)}\right) \\ &\quad + \sum_{j=1}^{n^u}\left(\frac{u(j) - u^{\mathrm{r}}(j)}{u(j) - \ell_l(j)}\right). \end{aligned}$$

In the above $u(j)$ and $u^{\mathrm{r}}(j)$ are used to denote the jth elements of u and u^{r} respectively. It can be observed that B_u achieves its minimum at u^{r} provided u^{r} is an interior point of $\mathbb{U}$; we address the latter issue in Section 4.3. Further, B_u becomes unbounded above as the input variable u approaches an actuator limit (i.e. the boundary of $\mathbb{U}$).

Based on the above basic setup, we developed a six step procedure to be used at each iteration of the r-MPC algorithm. The algorithm is defined in Section 4.1 and each step is detailed in the ensuing sections. In particular, state-estimation and integral action are considered in Section 4.2; steady-state reference calculation is considered in Section 4.3; calculation of the next control move is discussed in Section 4.4; parameter choices and on-line tuning rules are discussed in Section 4.5.

4.1. MPC algorithm

Model Predictive Control algorithms often contain several distinguishable sub-tasks for each iteration. A recent survey of industrial MPC packages by [19] outlines some of these tasks as: input/output measurement, state estimation, constraint prioritisation, computation of steady-state reference points, handling infeasibility and ill-conditioning issues and calculation of the next control move (to name a few). The algorithm presented below incorporates some of these tasks explicitly while others (if addressed) are discussed in the subsequent sections relating to each task.

Fig. 7 shows a simplified timing diagram for the algorithm. The sample duration—as denoted by T_{s}—is defined as the duration between successive control moves. We based the discrete time model 4 on the sample time T_{s}. The current time is denoted by t. The computation time, which may vary at each iteration, is denoted by T_{c}. We assume that the computation time includes all measurements and calculations necessary to compute the next control move.

Since the computation time may vary, it seems reasonable to stipulate a separate time—within the sample duration and after the computations have finished—at which the next control move is applied. Let this time be denoted by T_{a}. In the MPC algorithm, we measure the current output of the system and update the current state estimate $\hat{x}(k|k)$. This is used to predict the state at time $t + T_{\mathrm{a}}$ which is also used as the initial state for MPC calculations. If T_{a} is small fraction of T_{s} then care must be taken since a different model is required to give the correct estimate (for more details on this see e.g. Section 2.5 in [14] and the references therein). In what follows T_{a} is assumed equal to the sample time T_{s}. Hence, we first update the current state estimate $\hat{x}(k|k)$ given the latest measurements and then predict the state one sample in the future (denoted as $\hat{x}(k+1|k)$) and initialise the MPC calculations from this point onward.

Algorithm 1. At each time interval k, complete the following tasks,

(1) Measure the system output $y_{\mathrm{m}}(k)$ and if necessary input $u_{\mathrm{m}}(k)$.

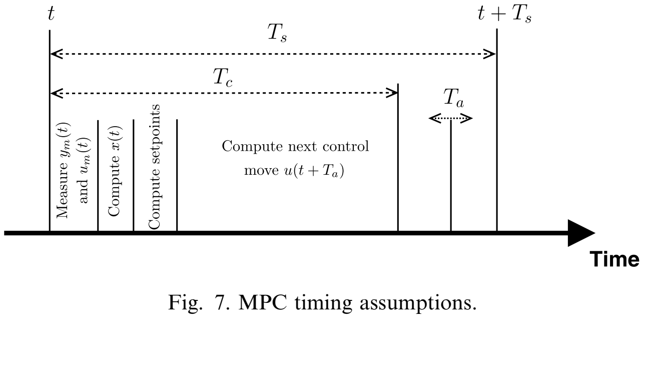

t
$t+T_s$
T_s
T_c
T_a
Measure $y_m(t)$ and $u_m(t)$
Compute $x(t)$
Compute setpoints
Compute next control move $u(t+T_a)$
Time

Fig. 7. MPC timing assumptions.

(2) Update the state estimate $\hat{x}(k+1|k)$ using the new measurements.
(3) Compute steady-state pair $(x_{\text{ss}}, u_{\text{ss}}) \in \mathbb{X}_F^o \times \mathbb{U}^o$.
(4) Compute state and input reference trajectories $\mathbf{x}^{\text{r}}$ and $\mathbf{u}^{\text{r}}$.
(5) Compute the next control move $u(k)$.
(6) Apply the calculated control move $u(k)$ at time $t + T_{\text{a}}$ to the system.

It is assumed that Step 1 takes negligible time compared with the sample duration. If this assumption is not reasonable then the corresponding delays should be accounted for in updating the state estimate $\hat{x}(k+1|k)$. Step 2 is treated in more detail in Section 4.2. Step 3 is treated in Section 4.3, Step 4 in Section 4.3.1 and Step 5 in Section 4.4.

Remark 4.1. The subscript m on $y_{\text{m}}(\cdot)$ and $u_{\text{m}}(\cdot)$ is used to denote that they are measured and thus to distinguish them from model outputs and inputs. In certain circumstances the desired control move may not be the actual measured control move applied to the actuator. In these cases it is important that the actual measured control move be used.

4.2. State estimation and integral action

In this section a state observer is designed to estimate the state in addition to a constant output disturbance term. The latter estimate provides a form of integral action for the MPC formulation considered in this paper (see e.g. [7,12,14]).

Consider the system given in (4). Further, assume that the output is corrupted by a disturbance signal $d_y(k)$. For the purposes of our analysis, $d_y(k)$ is assumed to be an integrated white noise signal. Further assume that the states are corrupted by white noise and the output measurements are corrupted likewise. With these assumptions, the system may be represented as (see Fig. 8 for a block diagram of this system),

$$x_d(k+1) = A_d x_d(k) + B_d u(k) + G w(k), \tag{6}$$

$$y_d(k) = C_d x_d(k) + v(k). \tag{7}$$

In the above, $w(k)$ and $v(k)$ are independent-identically-distributed white noise processes and

$$x_d(k) = \begin{bmatrix} x(k) \\ d_y(k) \end{bmatrix}, \quad G = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} B & 0 \\ 0 & H \end{bmatrix}, \tag{8}$$

$$A_d = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}, \quad B_d = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad C_d = [\,C \quad I\,]. \tag{9}$$

Here H is a 3×3 diagonal matrix whose elements are used as tuning parameters—see Section 4.5.

Let W denote the covariance matrix for $w(k)$ and V denote the covariance matrix for $v(k)$. Note that both W and V are diagonal; in Section 4.5 we discuss how they can be used as tuning parameters. Consider the observer structure (see e.g. [5,14]),

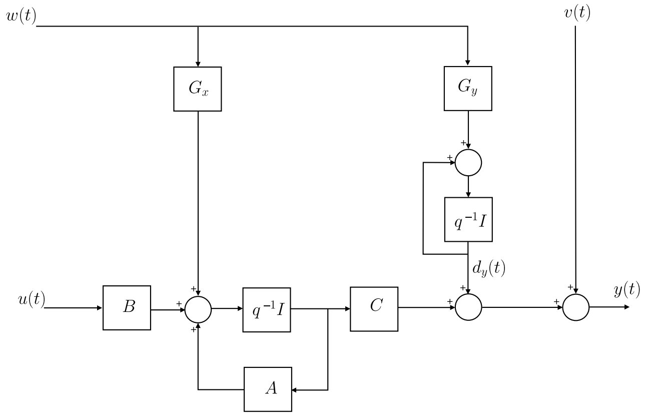


Fig. 8. Block diagram of system with various noise sources.

$$\begin{aligned} \hat{x}_d(k|k) &= \hat{x}_d(k|k-1) + L[y(k) - C_d\hat{x}_d(k|k-1)], \\ \hat{x}_d(k+1|k) &= A_d\hat{x}_d(k|k) + B_d u(k). \end{aligned} \tag{10}$$

We obtain the observer gain matrix L using the steady-state Kalman filter approach where we solve the following DARE for P (see e.g. [5]),

$$P = W + A_d\Big[P - PC_d^T(V + C_dPC_d^T)^{-1}C_dP\Big]A_d^T. \tag{11}$$

and obtain L using

$$L = PC_d^T(V + C_dPC_d^T)^{-1}.$$

4.3. Steady-state reference calculation

The MPC formulation given in Section 4 requires a strictly feasible steady-state pair. This is a common assumption for many MPC controllers in the literature [16]. Nevertheless, practical requirements often place the desired steady-state on or near the boundary of the constraint set [15,19]. Indeed the desired steady-state is often derived as the solution of a separate linear, quadratic, second-order cone or semi-definite program [11,18,19].

The following discussion offers one means for generating a strictly feasible steady-state pair. The underlying assumption is that a steady-state output reference vector r is given; this could be the result of a hierarchical optimisation problem or simply the demand as set by an operator.

Remark 4.2. It is sometimes desirable to consider a two-degree-of-freedom control structure. In this case, the output reference vector r is assumed to be the output of a pre-filter (see e.g. [14]).

Steady-state conditions can be described as follows. If system 4 reaches steady-state operation, then there exist some x_{ss} and u_{ss} such that (see e.g. [3]),

$$x_{ss} = A_d x_{ds} + B_d u_{ss}, \tag{12}$$

$$y_{ss} = C x_{ss} + d_{ys}. \tag{13}$$

In the above, d_{ys} is the steady-state estimate of the output disturbance, which in this application is simply the current estimate of the output disturbance $\hat{d}_y(k)$.

In the spirit of [18], our objective here is to find a pair (x_{ss}, u_{ss}) which satisfies (12) and minimises the distance between y_{ss} and the steady-state output reference vector r. In addition, the steady-state pair (x_{ss}, u_{ss}) should strictly satisfy the terminal and input constraints.

Remark 4.3. The steady-state disturbance term d_{ys} is given by its current estimate $\hat{d}_y(k)$, which is rarely invariant over successive time samples (see e.g. [11,19]). Consequently, the steady-state calculation problem is likely to be different—and hence necessarily must be solved—at each time sample.

A first-order low-pass pre-filter was included on each reference. Therefore, all changes made to the desired set-point via the SCADA (Supervisory Control And Data Acquisition) system are filtered before entering the r-MPC algorithm (see Section 5.4). The low-pass filters are given as follows: let $y_{ss}(i)$ denote the desired references then the ith reference as seen by the r-MPC algorithm is denoted by $r(i)$ and given by

$$r(i) = \frac{1 - \lambda_i 0.1}{1 - \lambda_i z^{-1}} y_{ss}(i).$$

Let $e_{ss}(x)$ be defined as

$$e_{ss}(x) = \frac{1}{2} \| r - (Cx + d_{ys}) \|^2_{Q_{ss}},$$

where Q_{ss} is a positive definite symmetric matrix. In the context of barrier functions, the steady-state calculation problem is naturally posed as the following optimisation problem.

$$(\mathrm{SS}_\mu): \quad (x^\mu_{ss}, u^\mu_{ss}) = \arg\min_{x,u} \quad e_{ss}(x) + \mu_{ss} B_u(u)$$
$$\text{s.t.} \qquad x = Ax + Bu.$$

The positive definite weighting matrix Q_{ss} was initially chosen as the identity matrix. However, under certain start-up conditions it was desirable to weight certain references more than others—see Section 5.2.

It is a standard result that the the above optimisation problem admits a strictly feasible solution (x^μ_{ss}, u^μ_{ss}) (see e.g. [8]). Furthermore, as the barrier weighting parameter μ_{ss} decreases, the penalty for being "near" a constraint reduces accordingly. In the limit as $\mu_{ss} \to 0$ then (x^μ_{ss}, u^μ_{ss}) converges to the solution of the following problem.

$$(\mathrm{SS}): \quad (x^*_{ss}, u^*_{ss}) = \arg\min_{x,u} \quad e_{ss}(x)$$
$$\text{s.t.} \qquad x = Ax + Bu,$$
$$u \in \mathbb{U}.$$

The difference between $e_{ss}(x^\mu_{ss})$ and $e_{ss}(x^*_{ss})$ is bounded by the duality-gap as follows (see e.g. [6]),

$$0 \leqslant e_{ss}(x^\mu_{ss}) - e_{ss}(x^*_{ss}) \leqslant M\mu_{ss}, \tag{14}$$

where M is the number of inequality constraints. For the current application $M = 6$.

From a practical perspective, μ_{ss} determines the relative weighting between $e_{ss}(\cdot)$ and the barrier term. In this way μ_{ss} may be interpreted as a tuning parameter. Furthermore, the simple bound in (14) provides a guideline for choosing μ_{ss} such that $e(x^\mu_{ss})$ is satisfactory.

It is important to observe that an equilibrium point is generated regardless of the value of μ_{ss} (greater than zero). Furthermore, if the closed-loop system is stable and reaches steady-state, then MPC with recentred barrier functions, guarantees that the steady-state value—(x^μ_{ss}, u^μ_{ss})—is attained (see e.g. [21]). This is not guaranteed with a more general choice of barrier function.

Remark 4.4. It should be noted that for μ_{ss} infinitesimally small, the gradient recentred barrier may become ill-conditioned since the gradient is large near the constraint boundaries. However, in practice there appears to be no observable benefit for choosing μ_{ss} infinitesimally small. For the industrial trials presented in Section 5, μ_{ss} was chosen in the order of 10^{-4}. This generally results in a steady-state pair close to, but not on, the constraint boundary. There were no ill-conditioning effects observed in practice.

With the controller running, plant operators can enter the desired set-points into the SCADA system. The controller then filters these references and calculates the steady-state set-points for the dynamic optimisation stage. This is discussed in Section 4.5.

4.3.1. Reference trajectory synthesis

The state and input reference sequences (or trajectories) $\mathbf{x}^r$ and $\mathbf{u}^r$ offer another degree of freedom in the MPC formulation. Future events may be captured in $\mathbf{x}^r$ and $\mathbf{u}^r$—providing a form of feedforward. Such considerations are beyond the scope of this paper. The reference sequences are defined as

$$\mathbf{x}^r = \begin{bmatrix} x^\mu_{ss} \\ \vdots \\ x^\mu_{ss} \end{bmatrix}, \quad \mathbf{u}^r = \begin{bmatrix} u^\mu_{ss} \\ \vdots \\ u^\mu_{ss} \end{bmatrix}.$$

4.4. Next control move calculation

Recall from Algorithm 1 that the beginning of each iteration involves a prediction of the state estimate given new measurement information, denoted by $\hat{x}(k+1|k)$. The predicted state is then used as an initial state for the calculating the next control move as follows.

Definition 4.1 (*Next Control Move Calculation*). At each time interval k, given the state and input reference trajectories $\mathbf{x}^{\mathrm{r}}$ and $\mathbf{u}^{\mathrm{r}}$, together with an initial state $\hat{x}(k+1|k)$, then solve the following optimisation problem.

$$\begin{aligned}(\text{MPC}):\quad & (\mathbf{x}^*(k),\mathbf{u}^*(k))\\ &= \arg\min_{\mathbf{x},\mathbf{u}} \; J(\mathbf{x},\mathbf{u})\\ \text{s.t.}\quad & x_0 = \hat{x}(k+1|k),\\ & x_{i+1} = Ax_i + Bu_i, \quad i = 0,\dots,N-1.\end{aligned}$$

The next control move $u(k)$ is defined to equal $u_0^*(k)$.

Treating $\mathbf{x}$ and $\mathbf{u}$ as vectors, J can be expressed in a more convenient form as

$$J(\mathbf{x},\mathbf{u}) = \frac{1}{2}\|\mathbf{x}-\mathbf{x}^{\mathrm{r}}\|_{\bar{Q}}^2 + \frac{1}{2}\|\mathbf{u}-\mathbf{u}^{\mathrm{r}}\|_{\bar{R}}^2 + \mu B_{\mathbf{u}}(\mathbf{u},\mathbf{u}^{\mathrm{r}}).$$

In the above, $\bar{Q}$ is given by the block diagonal matrix $\bar{Q} = \mathrm{diag}\{Q,\dots,Q,P\}$ and $\bar{R}$ is given similarly by $\bar{R} = \mathrm{diag}\{R,\dots,R\}$. The recentred barrier function $B_{\mathbf{u}}(\cdot)$ is given by

$$B_{\mathbf{u}}(\mathbf{u},\mathbf{u}^{\mathrm{r}}) = \sum_{i=0}^{N-1} B_u(u_i,u_i^{\mathrm{r}}).$$

The system dynamics can be expressed as

$$\bar{A}\mathbf{x} + \bar{B}\mathbf{u} = b(\hat{x}), \tag{15}$$

where $\bar{A}$ is an $(N+1)n_x \times (N+1)n_x$ matrix, $\bar{B}$ is an $(N+1)n_x \times Nn_u$ matrix and b is an $(N+1)n_x$ vector given respectively by

$$\bar{A} = \begin{bmatrix} I & 0 & 0 & \cdots & 0\\ -A & I & 0 & \cdots & 0\\ \vdots & & \ddots & & \vdots\\ 0 & \cdots & & -A & I\end{bmatrix},$$

$$\bar{B} = \begin{bmatrix} 0 & 0 & \cdots & 0\\ -B & 0 & \cdots & 0\\ \vdots & & \ddots & \vdots\\ 0 & \cdots & & -B\end{bmatrix}, \quad b(\hat{x}) = \begin{bmatrix}\hat{x}\\ 0\\ \vdots\\ 0\end{bmatrix}.$$

Since $\bar{A}$ has full rank, then Eq. (15) may be equivalently stated as

$$\mathbf{x} = \bar{A}^{-1}(b(\hat{x}) - \bar{B}\mathbf{u}). \tag{16}$$

Expanding (16) gives the following expression for $\mathbf{x}$,

$$\mathbf{x} = \Lambda\hat{x} + \Phi\mathbf{u}, \tag{17}$$

where Λ and Φ are given by,

$$\Lambda = \begin{bmatrix} I\\ A\\ \vdots\\ A^N\end{bmatrix}, \quad \Phi = \begin{bmatrix} 0 & \cdots & \cdots & 0\\ B & 0 & \cdots & 0\\ AB & B & \cdots & 0\\ \vdots & & \ddots & \\ A^{N-1}B & A^{N-2}B & \cdots & B\end{bmatrix}.$$

Using relation (17), (MPC) can be expressed as,

Definition 4.2. Given an initial state $\hat{x}(k+1|k)$, solve the following minimisation problem,

$$\begin{aligned}(\text{MPC}_I):\quad & \mathbf{u}^*(k)\\ &= \min_{\mathbf{u}} \frac{1}{2}\|\Lambda\hat{x}(k+1|k) + \Phi\mathbf{u} - \mathbf{x}^{\mathrm{r}}\|_{\bar{Q}}^2 + \frac{1}{2}\|\mathbf{u}-\mathbf{u}^{\mathrm{r}}\|_{\bar{R}}^2\\ &\quad + \mu B_{\mathbf{u}}(\mathbf{u},\mathbf{u}^{\mathrm{r}}).\end{aligned}$$

The strictly convex optimisation problem described in (MPC_I) can be solved using unconstrained optimisation techniques such as a damped Newton method (see e.g. [9]). Typically, such an algorithm must be initialised with a strictly feasible point, which in this application is quite straightforward since the constraints are simple bounds. Care must be taken to ensure that the iterates remain on the interior of the constraint set. If the barrier weighting parameter is close to zero, then the corresponding Hessian matrix of J can become ill-conditioned (see e.g. [25]). In [21,23], the (MPC_I) problem is reformulated to coincide with a point on the central-path of an associated quadratic program and numerically stable and computationally efficient algorithms are presented for solving the latter class of problems. These algorithms are based in modern interior-point methods but are modified to stop on the central-path associated with the barrier weighting parameter μ. For the industrial trials, we use a modified primal–dual predictor–corrector method based on Mehrotra's heuristics [17] (see Chapter 3 in [21] for more details).

4.5. Parameter choice and on-line tuning

Recall from Section 2 that the control objective is to regulate the inlet flow to separator one and the light phase outlet pressures on both separators.

There is an abundance of possible tuning parameters available in the current r-MPC formulation. This includes two new parameters μ_{ss} and μ which are barrier weighting terms. Given that the r-MPC structure is fixed then tuning can be restricted to the following parameters:

- Controller terms:
  - ○ μ_{ss}—steady-state barrier weighting (a positive scalar).
  - ○ μ—dynamic barrier weighting (a positive scalar).
  - ○ Q—state weighting matrix (positive semi-definite and symmetric).

  - ○ R—input weighting matrix (positive definite and symmetric).
  - ○ P—terminal state weighting matrix (positive semi-definite and symmetric).
  - ○ N—prediction horizon.
- Observer terms:
  - ○ W—state noise covariance matrix (positive definite and symmetric).
  - ○ V—output noise covariance matrix (positive definite and symmetric).
  - ○ G—state noise gain matrix.

The parameters μ_{ss}, μ, R, W and V were reserved for on-line tuning.

From Section 4.3, the steady-state barrier weighting was chosen as $\mu_{ss} = 10^{-4}$. The dynamic barrier weighting was initially chosen as the same value but other values are considered in Section 5.

To reduce the degree of freedom in tuning parameters, the state weighting matrix Q was chosen as $Q = C^T Q_y C$, with Q_y being a diagonal matrix representing weights on each output. The weights on each output were chosen such that deviations in flow are much more heavily penalised (this corresponds to the first output). Constant flow is important because chemical dosage units are manually set according to the flow reference value. The output weighting matrix Q_y was chosen as

$$Q_y = \mathrm{diag}(500, 1, 1). \tag{18}$$

The input weighting matrix R was chosen as a diagonal matrix with strictly positive entries. Each weight value was chosen such that deviations in flow valve position are more heavily penalised. The input weighting matrix was chosen as

$$R = \mathrm{diag}(100, 10, 10). \tag{19}$$

The terminal weighting matrix P was chosen as the solution to the DARE for the above state weighting matrix Q and input weighting matrix R.

It has been shown (see e.g. [18]) that when the terminal state lies inside some invariant terminal constraint set and is penalised in a correct manner, then the resulting finite length problem coincides with the infinite horizon problem. Mayne et al. [16] point out that this condition is implicitly satisfied with a sufficiently long prediction horizon. However, no terminal constraint set was included in this trial. The prediction horizon N was chosen initially as $N = 25$. This choice reflects a practical tradeoff between robustness (inherent to longer horizons) and computational cost (which grows with horizon length). Larger values up-to $N = 100$ were also tried, but with no obvious signs of improved performance the value of $N = 25$ was settled.

Apart from the dynamic barrier weighting term μ and the input weighting matrix R, all other controller parameters were essentially unmodified throughout the tuning procedure. In fact, the majority of tuning was performed with the observer parameters W, V and H—see Section 4.2 for their corresponding structure. The importance of observer dynamics with regards to MPC tuning and in particular to disturbance rejection has been noted for example by Maciejowski [14]. Our observations from industrial trials were strongly aligned with this perspective.

The state noise covariance matrix is given by

$$W = \mathrm{diag}(1, 5, 5, 1, 5, 5) \tag{20}$$

and the output noise covariance matrix is given by

$$V = \mathrm{diag}(0.1, 0.1, 0.1). \tag{21}$$

The matrix H (see the end of Section 4.2) is given by

$$H = \mathrm{diag}(0.04, 1, 1). \tag{22}$$

Lastly, the first-order low-pass reference filters described in Section 4.3 were chosen with the following parameters $\lambda_1 = 0.9$, $\lambda_2 = 0.7$ and $\lambda_3 = 0.7$.

The above tuning parameters were determined after numerous on-line trials and by considering various physical aspects of the plant. In the next section, some results from actual trials are presented. Unless otherwise stated, it can be assumed that the parameters given above are used.

5. Results and implementation

In this section, some results which illustrate r-MPC and some implementation issues are presented. Section 5.4 provides a brief description of the software and hardware implementation.

5.1. Controller performance

Recall that our objective is to reduce process variations about the operating point (see Section 1 and Fig. 2). Fig. 9 shows the process variables without r-MPC running while Fig. 10 shows the process variables with the controller running. Note, although these two samples were taken at different times, the oil being refined was the same in both cases and came from the same crude oil tank. The scaling for both figures is the same (aside from a DC offset term which is not important here). Using r-MPC in place of the existing configuration, and neglecting separator self-cleans (see Section 5.3), the standard deviation of S1 inlet flow was increased by 10% and the standard deviations of S1 and S2 back-pressure were decreased by 76% and 78% respectively.

Fig. 11 shows the system with various step reference changes on separator one and two. Step reference changes on inlet flow were kept to a minimum due to quality assurance considerations. This figure illustrates

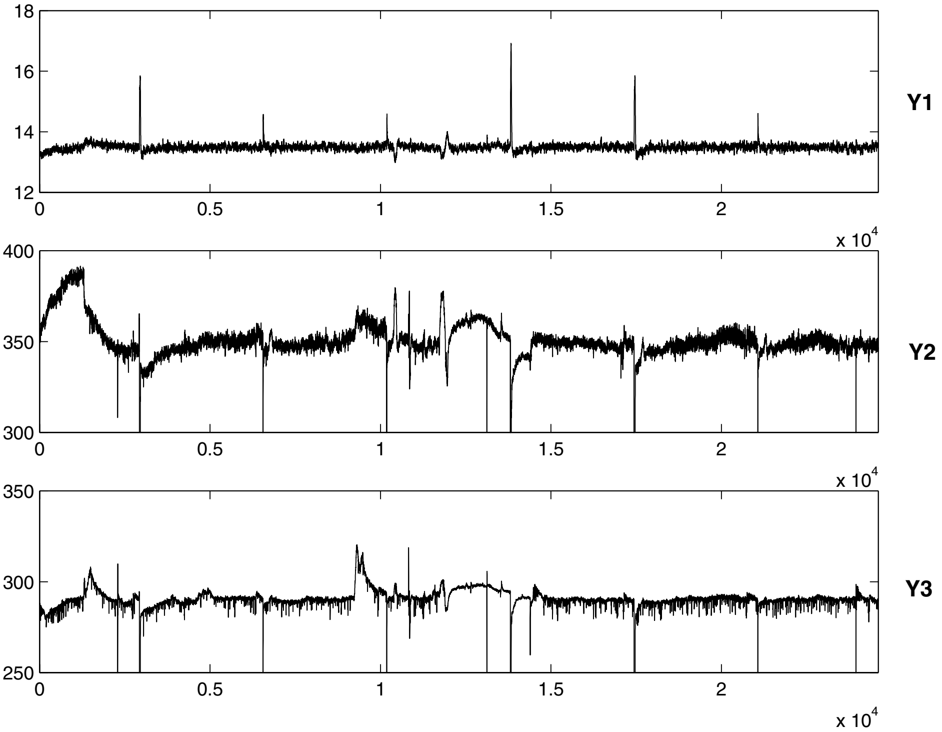


Fig. 9. System response over a period of approximately seven hours without the controller running . Note that a PI control loop is running for inlet flow regulation which is the original configuration.

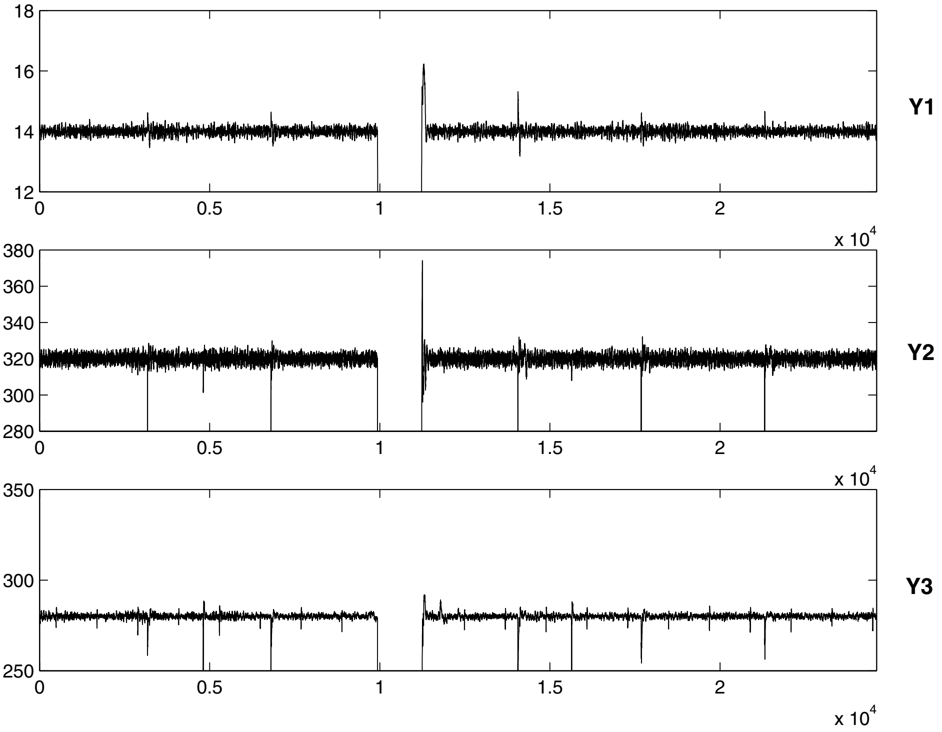


Fig. 10. System response over a period of approximately seven hours with the controller running. Note the visible reduction in process variation.

the benefit of MIMO control with respect to decoupling the various outputs.

Fig. 12 shows the system operating near a constraint boundary. This situation was anticipated at an early

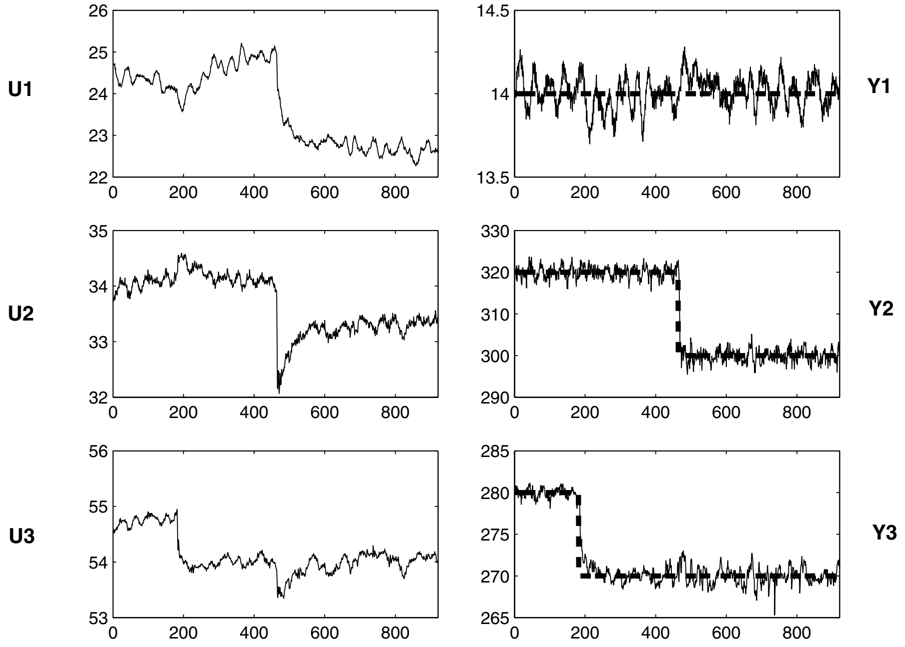


Fig. 11. System response to a step reference change for separator two (at around 200 s) and separator one (at around 450 s). The responses were deemed to be adequately decoupled.

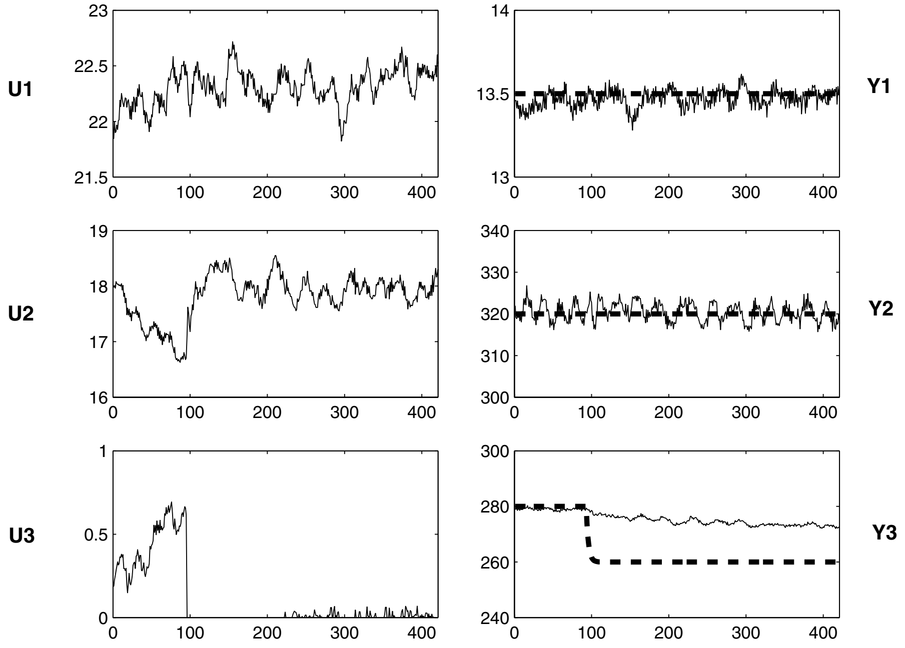


Fig. 12. System response to a step reference change for separator two. Note that the new set-point cannot be tracked without offset, and r-MPC pushes valve AV3 (almost) into the fully open position. Note that $\mu_{ss} = 10^{-4}$ gives acceptable results. Indeed, it seems reasonable to expect that the effects of reducing μ_{ss} even further would be indistinguishable in this situation.

stage (see Section 1). Note that separator two light-phase outlet pressure cannot track the reference since valve AV3 is hitting a constraint. Further, note that $\mu_{ss} = 10^{-4}$ which is deemed to be adequate—see Section 4.5.

Figs. 13 and 14 show the response to step reference changes on separator two. An artificial constraint was placed on valve AV2 in order to demonstrate the effect of varying μ. These figures confirm our intuition that small weighting values ($\mu = 10^{-4}$ as shown in Fig. 13)

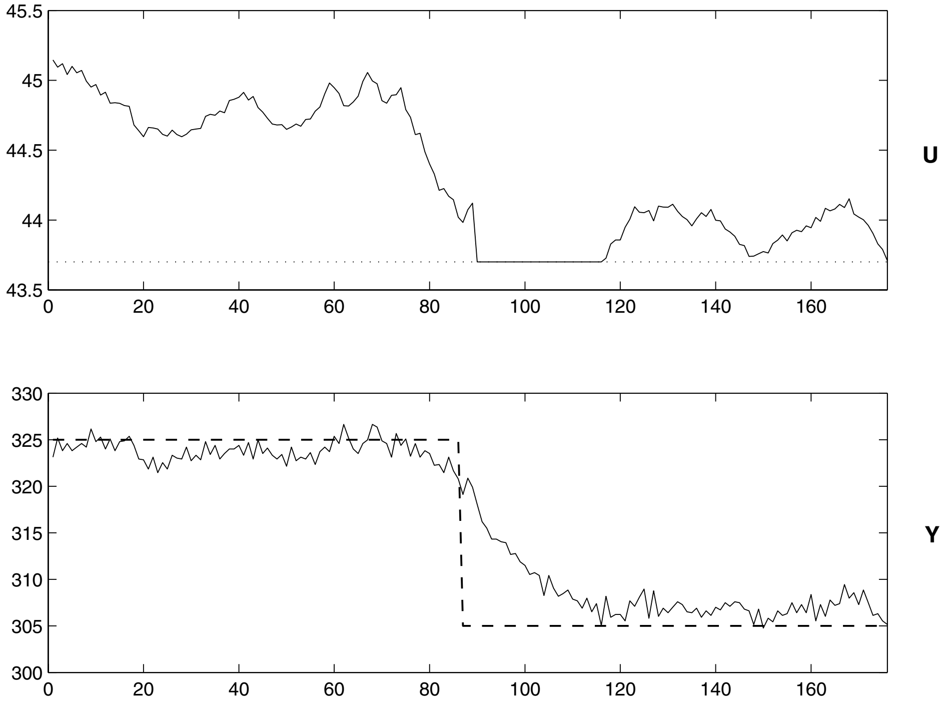


Fig. 13. AV2 valve position and separator one light-phase outlet pressure for a step reference change with $\mu_{ss} = 10^{-4}$ and $\mu = 10^{-4}$. Note that the controller response has a switching style movement—from off to on the constraint.

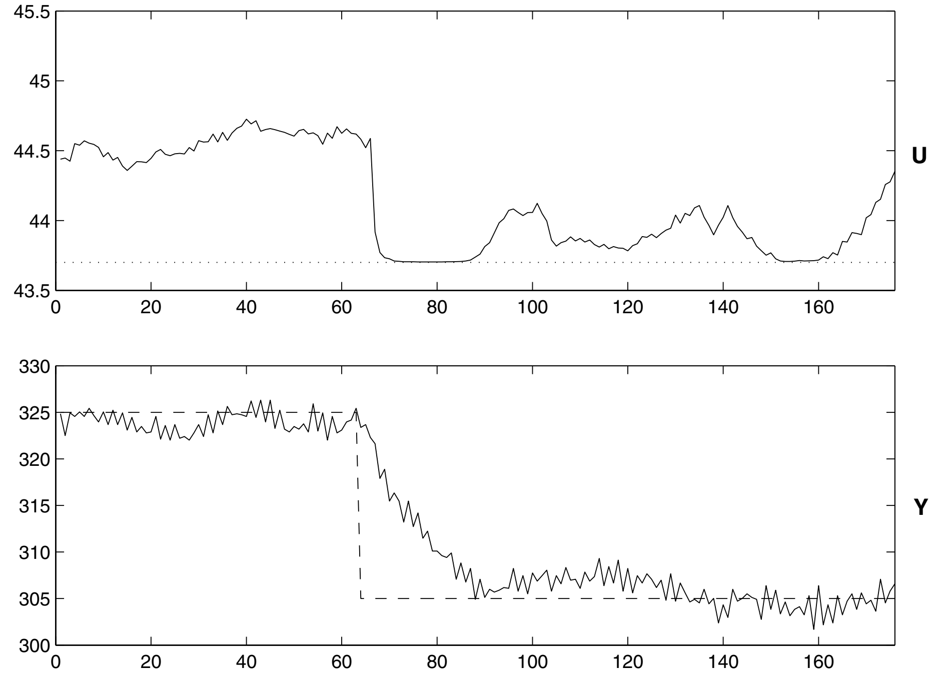


Fig. 14. AV2 valve position and separator one light-phase outlet pressure for a step reference change with $\mu_{ss} = 10^{-4}$ and $\mu = 1$. Note that the controller response now has a smooth style movement—gradually approaching the constraint boundary.

generate a switching style control action—i.e. either on or off the constraint—while larger values of weighting parameter ($\mu = 1$ as shown in Fig. 14) result in smoother control action near constraint boundaries, without significantly changing behaviour away from constraint boundaries.

Based on these trials, our industrial partners commissioned us to install the control software on the main

SCADA computer. This is now fully operational and the company estimates an eight month return on investment based on one oil type alone (there are 16 different oil types).

5.2. Manual to automatic operation

As mentioned in Remark 2.1 the plant re-configuration included the installation of two automated control valves. Fig. 1 shows these valves as AV2 and AV3 which are in series with manual valves MV1 and MV2 respectively. The original proposal recommended that the automated control valves be fitted with manual hand wheels and replace the manual valves. Included in this proposal was a valve position indicator so that switching from manual to automatic operation could be achieved in a parsimonious manner.

One of the challenges for the control implementation was switching from manual to automatic mode. During manual mode, AV2 and AV3 are kept fully open and the light phase outlet pressures are regulated manually with MV1 and MV2 respectively. The flow is regulated with a PI loop around control valve AV1. In automatic mode, the r-MPC controller handles plug positions for AV1, AV2 and AV3.

To switch from manual to automatic mode, the following procedure was used:

(1) Hand control of AV1, AV2 and AV3 over to r-MPC.
(2) Gradually move manual valve plug position on separator one (MV1) into fully open position.
(3) Gradually move manual valve plug position on separator two (MV2) into fully open position.

Fig. 15 shows the system response to this procedure. The above procedure is made plausible because of the inherent constraint handling capabilities of r-MPC. It can be observed from Fig. 15 that the automatic valves are placed on constraint boundaries during this procedure.

5.3. Separator self-cleaning

An inevitable component of crude edible oil is particulate matter such as dirt and gums etc. Unfortunately, this particulate matter collects inside the separator unit at the bowl periphery and can (if untreated) ultimately cause the unit to fail. As such, typical operation of a separator unit includes a periodic *self-cleaning* process, whereby this internal bowl actually opens for a brief duration allowing the particulate matter to escape. Although this operation completes in the order of milliseconds, almost 40% of the separator bowl volume is lost during this time. As might be expected, this results in a significant disturbance to inlet flow and outlet pressure of the separator unit (and other units). Self-cleaning is responsible for most of the spikes in Figs. 9 and 10.

Immediately after this operation, the separator bowl is partially empty and the inlet flow tends to surge whilst

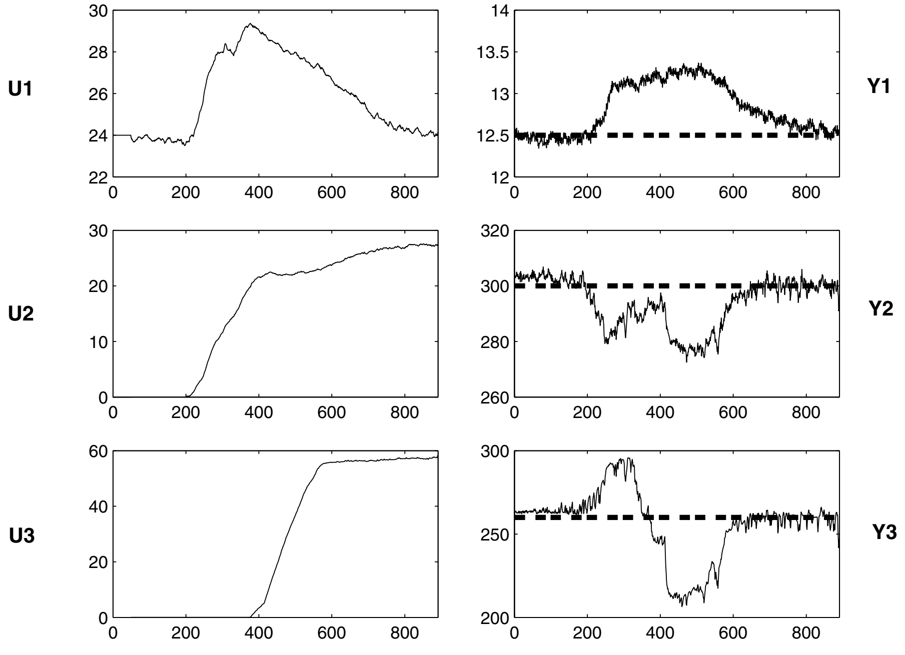


Fig. 15. The strategy used for switching from manual to automatic operation. Note that the controller is switched on at around 50 s and after 200 s the manual valve MV1 is gradually moved into the fully open position. Note that this disturbance is detected by the controller and is compensated for. At around 400 s the same procedure is repeated for manual valve MV2. Again the controller compensates for this disturbance.

the outlet pressure tends to atmospheric pressure (due to no outlet flow). If not considered, the controller would (and does) try to close the light phase outlet pressure valve in order to maintain the reference pressure. This has potentially dire consequences because with this valve closed, no light phase flow will prevail and all the oil (plus soap) will be lost in the heavy phase outlet. Therefore, unacceptable losses will be incurred.

To overcome this problem, the following strategy was adopted: since the time of a self-clean operation is known in advance, all valve positions are frozen just before the operation and held constant until adequate flows and pressures are regained, at which point the controller will become active again. This essentially mimics the way a plant operator handles the same scenario, albeit with faster response.

As a further comment, it was beneficial to let the reference values equal the actual measured outputs during the self-clean process. When the output measurements have all returned to within some acceptable tolerance of the actual desired set-points, the current reference values are passed through the prefilter (the same used for set-point filtering) and control action resumes as normal. This can be seen in Fig. 16.

5.4. Software and hardware hierarchy

The r-MPC algorithm was coded in C++ as a stand alone set of libraries. Initial industrial trials were aided with a graphical interface to this library using the UNAC industrial control software package (see http://www.unac.com.au/). This package is not used in the fully commissioned r-MPC algorithm. The algorithm runs on a 500 MHz Intel Pentium III and the entire processing time each iteration takes less than 0.1 s.

Fig. 17 shows a schematic of the hardware structure involved. The SCADA system provides an interface for plant operators to important variables such as flow and

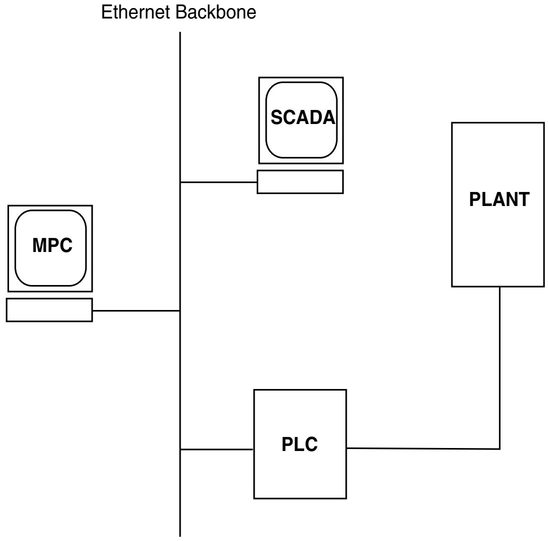


Fig. 17. Schematic representation of the hardware.

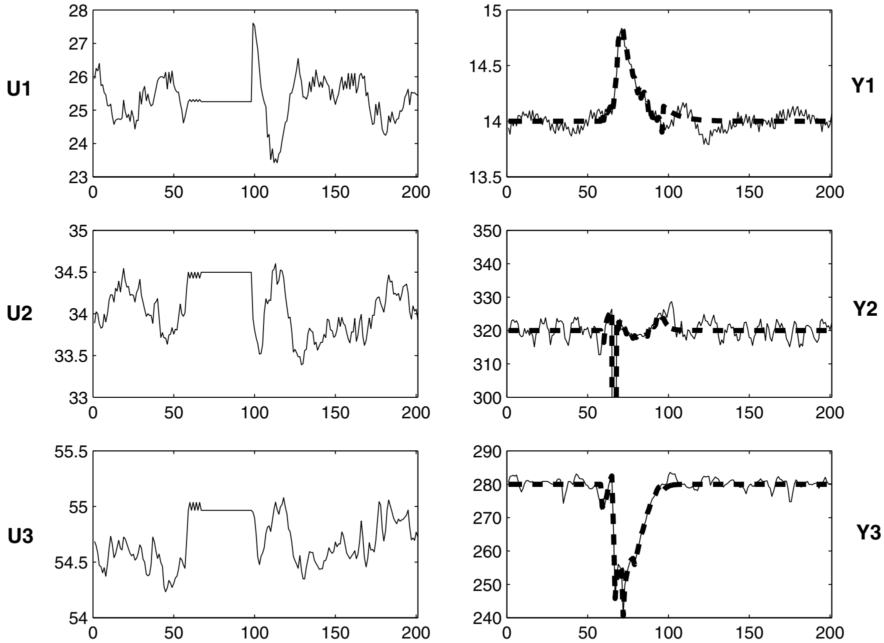


Fig. 16. The self-clean strategy where valve positions are held constant over a pre-specified duration beginning just before the self-clean and holding until pressures and flows are adequate. Notice the sudden decrease in separator one pressure at around 65 s, this is precisely when the self-clean occurs (the pressure actually drops to zero, but for clarity the image is cropped). Further, notice the resulting disturbance to inlet flow and separator two pressure. During the self-clean operation, references are set to the measured outputs. When the controller is reactivated (at around 100 s), the references are passed from their current values to their desired set-points via a low pass filter (the same filter used to filter reference changes).

pressure references. The PLC (Programmable Logic Controller) system provides an interface between actuator and sensor signals to the rest of the hardware system. The PLC also has numerous logic statements for discrete event type control and offers some basic PID control algorithms.

Various software and hardware safeguards were incorporated during the industrial trials. For example, it is important that the r-MPC computer has a seamless connection to the PLC. As a typical safeguard, a heartbeat signal is transmitted between the r-MPC computer and the PLC so that plant operators could be alerted to any communications loss. As another example, should the optimisation software fail to produce a result in time then the previous input is used. An alarm is sent to the operators if this this situation arises too frequently.

6. Conclusion

We have discussed the application of r-MPC to a novel industrial control problem. The particular plant in question is an edible oil refining line. The motivation for incorporating feedback control is to give plant operators a means to balance the tradeoff between quality and yield. Refined oil quality and refinery yield can be related to certain process conditions and in the absence of feedback control these process conditions can exhibit high variations. An effect of high variations is that meeting product quality requirements (an issue of utmost importance in this application) is achieved at the cost of reduced process yield.

With the introduction of multivariable feedback control, in the form of r-MPC, the standard deviation of S1 inlet flow was increased by 10% and the standard deviations of S1 back-pressure and S2 back-pressure were decreased by 76% and 78% respectively. More recently a modified version of the controller was fully commissioned on the same plant with an estimated eight month return on investment using figures from one oil type alone (there are 16 different oil types).

Physical constraints in the form of actuator limits were taken into consideration within the r-MPC strategy and their inclusion was observed to be beneficial. One interesting example of this was seen during the manual to automatic switching of the controller strategy. Here, manual control (where r-MPC is not running) is switched to automatic control (where r-MPC is running). The controller is able to exploit constraints to aid this transition.

Tuning of the controller was achieved partly on-line and partly off-line. The most influential tuning parameters, however, were seen to be those related to observer dynamics. The other notable parameter was the barrier weighting term μ. The intuitive effect of μ on controller dynamics was demonstrated, i.e. increasing the barrier weighting term resulted in more cautious controller action near constraint boundaries, while the controller performed as usual away from constraint boundaries.

Acknowledgements

The authors would like to acknowledge the help, both technical and financial, of Peter Cuthbert, Richard Cuthbert, Greg Ryan and John Templeman from Meadow Lea Pty. Ltd. West Footscray, Australia. We thank Prof. Michel Perrier for his help; particularly in terms of technical discussions and early proof reading. We thank the CRC for International Food Manufacture and Packaging Science and the Centre for Integrated Dynamics And Control (CIDAC) for their financial and administrative support. We would also like to thank the anonymous reviewers for their helpful suggestions.

References

[1] Alfa-Laval, Instruction book—Separator SRPX 314HGV-14CH, Alfa-Laval—Instruction Book, 1990.

[2] Alfa-Laval, Theory of separation, Alfa-Laval—Booklet, 1990.

[3] K. Åström, B. Wittenmark, Computer Controlled Systems—Theory and Design, second ed., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1990.

[4] P.W. Atkins, Physical Chemistry, third ed., Oxford University Press, Oxford, 1986.

[5] R.R. Bitmead, M. Gevers, V. Wertz, Adaptive Optimal Control—The Thinking Man's GPC, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1990.

[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[7] D.W. Clarke, C. Mohtadi, P.S. Tuffs, Generalized predictive control, parts 1 and 2, Automatica 23 (2) (1987) 137–148.

[8] A.V. Fiacco, G.P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, John Wiley & Sons Inc., New York, 1968.

[9] P.E. Gill, W. Murray, M.H. Wright, Practical Optimization, Academic Press, London, 1981.

[10] Y. Hui, Bailey's Industrial Oil & Fat Products, John Wiley & Sons Inc., New York, 1996.

[11] D.E. Kassmann, T.A. Badgwell, R.B. Hawkins, Robust steady-state target calculation for model predictive control, AIChE Journal 46 (5) (2000) 1007–1024.

[12] H. Kwakernaak, R. Sivan, Linear Optimal Control Systems, John Wiley & Sons, Inc., New York, 1972.

[13] L. Ljung, System Identification: Theory for the User, second ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1999.

[14] J.M. Maciejowski, Predictive Control with Constraints, Pearson Education Limited, Harlow, Essex, 2002.

[15] D.Q. Mayne, Nonlinear model predictive control: an assessment, in: AIChE Symposium Series, 5th International Symposium on Chemical Process Control, vol. 93, 1997, pp. 217–231.

[16] D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: stability and optimality, Automatica 36 (2000) 789–814.

[17] S. Mehrotra, On the implementation of a primal–dual interior-point method, SIAM Journal on Optimization 2 (1992) 575–601.

[18] K.R. Muske, J.B. Rawlings, Model predictive control with linear models, AIChE Journal 39 (2) (1993) 262–287.

[19] S.J. Qin, T.A. Badgewell, A survey of industrial model predictive control technology, Control Engineering Practice 11 (2003) 733–764.

[20] D.E. Seborg, T.F. Edgar, E.A. Mellichamp, Process Dynamics and Control, John Wiley & Sons Inc., New York, 1989.

[21] A.G. Wills, Barrier function based model predictive control, Ph.D. thesis, School of Electrical Engineering and Computer Science, University of Newcastle, Australia, 2003.

[22] A.G. Wills, W.P. Heath, A recentred barrier for constrained receding horizon control, in: Proceedings of the 2002 American Control Conference, vol. 5, Anchorage, Alaska, 2002, pp. 4177–4182.

[23] A.G. Wills, W.P. Heath, Using a modified predictor–corrector algorithm for model predictive control, in: Proceedings of the 15th IFAC World Congress on Automatic Control. Barcelona, Spain, 21–26 July 2002.

[24] A.G. Wills, W.P. Heath, Barrier function based model predictive control, Automatica 40 (8) (2004) 1415–1422.

[25] M.H. Wright, Interior methods for constrained optimization, in: A. Iserles (Ed.), Acta Numerica 1992, Cambridge University Press, New York, USA, 1992, pp. 341–407.