Journal of Process Control 22 (2012) 873–883

Contents lists available at SciVerse ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Self-optimizing control with active set changes

Henrik Manum, Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway

ARTICLE INFO

Article history:
Received 5 June 2011
Received in revised form 22 January 2012
Accepted 29 February 2012
Available online 12 April 2012

Keywords:
Self-optimizing control
Control Structure selection
Quadraticoptimization
Plantwide control

ABSTRACT

In this paper we extend the "nullspace method" by [1] to cover changes in active set. The extension is based on recent results from explicit model predictive control by [3]. The nullspace method is a method for selecting controlled variables, assuming that the set of active constraints does not change. With the extension presented here, we show that by applying the nullspace method for several different regions, where the regions are found by a parametric program, we can use the value of the controlled variables for each region to decide when to switch regions.

The proposed method is demonstrated on a simple model of an ammonia production plant, and the results are comparable to real-time optimization.

1. Introduction

In this paper we extend some recent results on implementation of quadratic programs to cover changes in the active set. The work is in the field of "self-optimizing control", where the focus is to select the right variables c to control, such that acceptable operation under all conditions is achieved with constant setpoints for the controlled variables [17,8].

A more direct approach for ensuring optimal operation is real-time optimization (RTO) [12]. Using RTO, the optimal values (setpoints) for the controlled variables c are computed online based on online measurements, and a model of the process [2]. In control of chemical processes, an hierarchal structure [4] is often preferred. RTO is then used to calculate setpoints c_s for the controlled variables c for the supervisory control layer. In the supervisory layer, model predictive control (MPC) [14,13] is often used.

A typical hierarchical structure is outlined in Fig. 1. In this paper, we assume that regulatory control is already implemented, and the focus is on the interaction between the optimizer and the supervisory control.

In the RTO framework, the degrees of freedom in the measurement selection or combination matrix H is not exploited as an optimization variable, while for "self-optimizing control" finding a good H is the main focus. The two approaches are therefore complementary. In this paper, H is a constant combination matrix.

One can identify at least four ways of choosing the combination matrix H for the controlled variables $c = Hy$:

1. Use $c = u_0$, i.e. open loop control. This is not expected to work very well unless the static optimization is updated frequently.
2. Use $c = y_0$, where y_0 are the presently used controlled variables in the supervisory layer. Also this choice is not expected to give a good performance unless the static optimization problem is resolved frequently.
3. Use $c = Hy$, where y is all available measurements, including u_0 and y_0. If H is chosen carefully, this choice is expected to give better performance between samples of the RTO than the choices above. In particular, H should be chosen such that even though we have large disturbances, the optimal values c_s of the controlled variables c does not change much. Such a choice of H may be beneficial in at least two ways. First, since RTO is in general a non-convex problem, the starting values for the optimization are important, and thus if the optimal values do not change much, such a choice of c should aid the success of a RTO implementation. Second, c_s can be updated less frequent and the system will work better if the RTO is out of service. In the ideal "self-optimizing" case, one identifies controlled variables such that the RTO layer may be eliminated. During the last decade several methods for finding "good" controlled variables have been developed, such as the maximum gain rule [7], exact local method [6,2], and the nullspace method [1].
4. Even more general; one may change the controlled variables c as operating conditions change. This is equivalent to letting the map H be a function of the operating conditions.

* Corresponding author. Tel.: +47 735 94154; fax: +47 735 94080.
E-mail addresses: henman@statoil.com (H. Manum), skoge@ntnu.no (S. Skogestad).

0959-1524/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2012.02.015

In this paper, we consider the last approach and develop a method for changing the combination matrix H when changes in the active set occur. The results are exact for quadratic problems, but an example of an ammonia production plant shows that the method may be applicable also to more general processes by local linearization.

The approach we use is to exploit a link between self-optimizing control and linear-quadratic explicit MPC [11]. The link is exact for quadratic approximations of the self-optimizing control problem, because then the static optimal operation problem of self-optimizing control and explicit MPC have the same equation structure.

Using parametric programming [9] and recent results from explicit MPC [3] on implementation of the optimal solution, we show that combination matrices H^i, found by using the nullspace method, can be used to track changes in the active set using only information about the measured outputs. In the multivariable case, a scalar function of the outputs is enough to track changes in the active set. We have already proposed similar results earlier, see [10,11], where we used controlled variables (invariants) from the nullspace method to track changes in the optimal active set. However, we proposed to keep track of the whole vector $c = Hy \in \mathbb{R}^{n_c}$, where n_c is the number of controlled variables. Baoticc̀ et al. [3] show that this may be simplified further by tracking the scalar function $w'Hy$ (where w is a vector that is found by a specific off-line algorithm). One of the contributions of this paper is therefore to transfer these results to the field of self-optimizing control.

The rest of the paper is organized as follows: First, we review theory from self-optimizing control and implementation of solutions to quadratic optimization problems using *descriptor functions*. Then we show how continuous piecewise-affine (PWA) descriptor functions from measurements can be constructed by using the *nullspace method*. We then discuss how to match constraints between measured constraints and constraints in the model. This *constraint matching* may in some cases have a significant effect on the economical operation of a given plant. Thereafter we collect our findings in an algorithm for design of a control structure that handles changes in the active set, and finally we show how this method can be used on an example of an ammonia production plant.

2. Background

2.1. Quadratic approximation to RTO

We consider the steady-state optimal operation problem

$$\begin{aligned} &\min_{x,u_0} J_0(x, u_0, d_0) \\ &\text{s.t.} f_0(x, u_0, d_0) = 0 \\ &\quad g_0(x, u_0, d_0) \geq 0 \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ are states, $u_0 \in \mathbb{R}^{n_{u_0}}$ are steady state degrees of freedom and $d_0 \in \mathbb{R}^{n_{d_0}}$ are disturbances. Using the model equations $f_0(x, u_0, d_0) = 0$ to formally eliminate the internal state x, we can rewrite problem (1) on the form

$$\begin{aligned} &\min_{u_0} J(u_0, d_0) \\ &\text{s.t.} g(u_0, d_0) \geq 0 \end{aligned} \tag{2}$$

2.2. Unconstrained case

Assume that for the nominal disturbance $\bar{d}_0$, the optimal input u_0^* is such that none of the inequality constraints $g(u_0^*, \bar{d}_0) \geq 0$ are exactly equal to zero (i.e., they are not active). Further, introduce the following substitutions:

$$u = u_0 - u_0^* \tag{3}$$

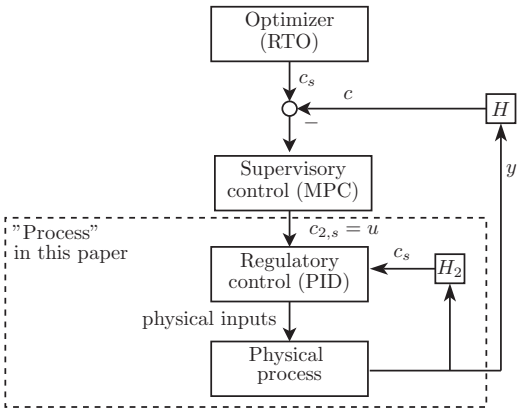

Fig. 1. Interconnection between RTO and supervisory control layer. Self-optimizing control focuses on the measurement selection or combination matrix H.

$$d = d_0 - \bar{d}_0 \tag{4}$$

$$M_u = -\nabla_{u_0} g|_{u_0^*, \bar{d}_0} \tag{5}$$

$$M_d = \nabla_{d_0} g|_{u_0^*, \bar{d}_0} \tag{6}$$

$$M = g(u_0^*, \bar{d}_0) \tag{7}$$

By a quadratic expansion of the objective function around the nominal optimum ($\nabla J(u_0^*, \bar{d}_0) = 0$) we can obtain the following quadratic approximation to problem (2), which we will use throughout the paper:

$$\begin{aligned} &\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}' & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \\ &\text{s.t.} M_u u \leq M + M_d d \end{aligned} \tag{8}$$

Here, the notation J_{uu} means the second derivative of the function J with respect to the inputs u, and so on.

2.3. Extension to constrained case

The case of a nominally constrained optimum can also be posed on the form of problem (8), which we will now demonstrate. The only difference from the unconstrained case is that we do a change of variables to "eliminate" the effect of a non-zero first derivative J_u at the optimum.

First, we define a Lagrangian as:

$$\mathcal{L}(u_0, d_0, \lambda) = J(u_0, d_0) - \lambda' g(u_0, d_0) \tag{9}$$

where λ are the Lagrange multipliers. Then, we make a quadratic approximation of the nonlinear program (2) around the optimal point (u_0^*, λ^*) as [15,20]:

$$\begin{aligned} &\min_{u_0} \underbrace{\nabla_{u_0} J|_{u_0^*, \bar{d}_0}}_{J_u} (u_0 - u_0^*) + \frac{1}{2} \begin{bmatrix} u_0 - u_0^* \\ d_0 - \bar{d}_0 \end{bmatrix}' \begin{bmatrix} \mathcal{L}_{u_0 u_0} & \mathcal{L}_{u_0 d_0} \\ \mathcal{L}_{u_0 d_0}' & \mathcal{L}_{d_0 d_0} \end{bmatrix} \begin{bmatrix} u_0 - u_0^* \\ d_0 - \bar{d}_0 \end{bmatrix} \\ &\text{s.t.} g(u_0^*, \bar{d}_0) + \nabla \begin{bmatrix} u_0 \\ d_0 \end{bmatrix} g|_{u_0^*, \bar{d}_0} \begin{bmatrix} u_0 - u_0^* \\ d_0 - \bar{d}_0 \end{bmatrix} \geq 0 \end{aligned} \tag{10}$$

where we have cancelled the term $\nabla J_{d_0}(d_0 - \bar{d}_0)$ which cannot be affected by the degrees of freedom u_0. All first and second derivatives are evaluated at the nominal optimum, $(u^*, \bar{d}_0)$. Under the assumption that $\mathcal{L}_{u_0 u_0}$ is positive definite (second-order optimality

conditions) we introduce the following change of variables for the degrees of freedom u_0:

$$u = u_0 - u_0^* + \mathcal{L}_{u_0 u_0}^{-1} J_u \tag{11}$$

Note that this definition of u is not in conflict with definition (3) used for the unconstrained case, because for an unconstrained minimum $J_u = 0$ and the two definitions coincide. Now, by *defining* J_{uu}, J_{ud} and J_{dd} as

$$\left[\begin{array}{c|c} J_{uu} & J_{ud} \\ \hline J'_{ud} & J_{dd} \end{array}\right] \triangleq \left[\begin{array}{c|c} \mathcal{L}_{uu} & \mathcal{L}_{ud} \\ \hline \mathcal{L}'_{ud} & \mathcal{L}_{dd} \end{array}\right], \tag{12}$$

the nominally constrained case can be written exactly on the form of problem (8). Note again the analogy to the unconstrained case: for the unconstrained optimum the Hessian of the quadratic approximation is equal to the Hessian of the objective function at the nominal operating point, while for the constrained case the Hessian of the quadratic approximation is equal to the Hessian of the Lagrange function of the original problem at the nominal point. Further, note that the unconstrained case is a special case of the constrained case and is included here only to ease the presentation of the material, and because in the example we consider in this paper, the nominal optimum happens to be unconstrained.

Remark 1. The matrix J_{dd} is not needed and may be set to zero.

2.4. Nullspace method

The nullspace method by [1] deals with the optimal selection of linear measurement combinations as controlled variables, $c = Hy$. In a recent paper by the same authors [2], their results are interpreted more generally as deriving linear invariants for quadratic optimization problems. More specifically, a key result is the following theorem:

Theorem 1 (Linear invariants for quadratic optimization problems [2]). *Consider an unconstrained quadratic optimization problem in the variables $u \in \mathbb{R}^{n_u}$, parameterized in $d \in \mathbb{R}^{n_d}$:*

$$J^*(d) = \min_u \left\{ J(u,d) = \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ J'_{ud} & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \right\}. \tag{13}$$

In addition, there are "measurement" variables $y = G^y u + G_d^y d$. If there exists $n_y \geq n_u + n_d$ independent measurements (where "independent" means that the matrix $\tilde{G}^y = [G^y\ G_d^y]$ has full column rank), then the optimal solution to (13) has the property that there exists $n_c = n_u$ linear variable combinations (constraints) $c = Hy$ that are invariant to the disturbances d, meaning that their optimal value ($c = 0$) is independent of d. Here, H may be found from the nullspace method using $H = \mathrm{null}(F')$, where

$$F = -(G^y J_{uu}^{-1} J_{ud} - G_d^y). \tag{14}$$

2.5. Implementation of solution to parametric quadratic programs

In this section, we follow [3] unless otherwise noted. This implies that all theorems, lemmas, algorithms and definitions are taken from the reference unless otherwise noted. For the reader familiar with the results of [3], this section may be skipped.

Definition 1. Two polyhedra $P_i, P_j \in \mathbb{R}^{n_x}$ are called *neighboring polyhedra* if their interiors are disjoint and $P_i \cap P_j$ is $(n_x - 1)$-dimensional (i.e. is a common facet).

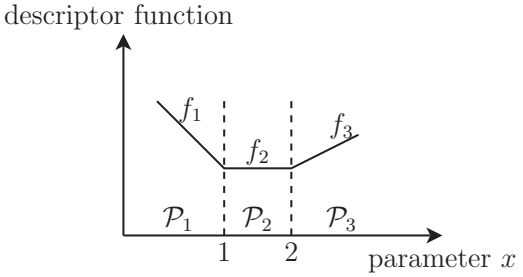

Fig. 2. A scalar descriptor function over three polyhedra.

Let $\{P_i\}_{i=1}^{N_p}$ be a polyhedral partition. For each polyhedron P_i, denote C_i as the list of all its neighbors,

$$C_i := \left\{ j \;\middle|\; \begin{array}{l} P_j \text{ is a neighbor of } P_i, \\ j = 1, \ldots, N_p, \quad j \neq i \end{array} \right\} \tag{15}$$

Throughout the paper we assume that every facet is shared by only two neighboring polyhedral partitions, i.e. that the facet-to-facet property [19] holds.

Definition 2 *(PWA descriptor function).* A scalar continuous real-valued PWA function $f : X_f \mapsto \mathbb{R}$,

$$f(x) := f_i(x) = A_i' x + B_i \quad \text{if } x \in \mathcal{P}_i, \tag{16}$$

with $A_i \in \mathbb{R}^{n_x}$, $B_i \in \mathbb{R}$, is called a *descriptor function* if

$$A_i \neq A_j, \quad \forall j \in C_i,\ i = 1, \ldots, N_p, \tag{17}$$

where $\cup_i \mathcal{P}_i = X_f \subset \mathbb{R}^{n_x}$, and C_i is the list of neighbors of $\mathcal{P}_i$.

See Fig. 2 for an example of a scalar PWA descriptor function. This kind of function can be used to track changes in the optimal active set. We can do this because the sign of $f_i(x) - f_j(x)$ changes only when the point x crosses the separating hyperplane between $\mathcal{P}_i$ and $\mathcal{P}_j$. Thus for all $x \in \mathcal{P}_i$, the difference $f_i(x) - f_j(x)$ has the same sign.

2.6. Example

In the figure, let $f_1 = -2x + 5$, $f_2 = 3$, and $f_3 = 0.5x + 2$, where we note that $f_1(1) = 3$ and $f_3(2) = 3$. Assume that the parameter or disturbance x is in $\mathcal{P}_2$ and we want to detect when x crosses into either $\mathcal{P}_1$ or $\mathcal{P}_3$ *without* measuring x itself (but we have measurements available of f_1, f_2, f_3). We can do this using the descriptor function

$$f := f_i \quad \text{if } x \in \mathcal{P}_i,\ i = 1, 2, 3.$$

For $x \in \mathcal{P}_2$ we have that $\mathrm{sign}(f_2 - f_1) = 1$ and $\mathrm{sign}(f_2 - f_3) = 1$. Now, if either $\mathrm{sign}(f_2 - f_1)$ or $\mathrm{sign}(f_2 - f_3)$ changes value, we deduce that x has moved to $\mathcal{P}_1$ or $\mathcal{P}_3$, respectively.

Definition 3 *(Ordering function).* Let $f(x)$ be a PWA descriptor function on the polyhedral partition $\{\mathcal{P}_i\}_{i=1}^{N_p}$. An ordering function $O_i(x)$ is defined as

$$O_i(x) := [O_{i,j}(x)]_{j \in C_i} \tag{18}$$

where

$$O_{i,j} = \begin{cases} +1 & \text{if } f_i(x) \geq f_j(x) \\ -1 & \text{if } f_i(x) < f_j(x) \end{cases} \tag{19}$$

with $i \in \{1, \ldots, N_p\}$, $j \in C_i$.

Note that for each polyhedral region, the length of the binary vector-function $O_i(x)$ corresponds to the number of neighbours to the region.

Theorem 2. *Let $f(x)$ be a PWA descriptor function on the polyhedral partition $\{P_i\}_{i=1}^{N_p}$. Let $\xi_i \in \mathbb{R}^{n_x}$ be any point in the interior of $\mathcal{P}_i$, and define*

$$\begin{aligned} S_{i,j} &:= O_{i,j}(\xi_i) \\ S_i &:= O_i(\xi_i), \end{aligned} \tag{20}$$

with $i = 1, \ldots, N_p$, $j \in C_i$. Then the following holds:

$$\begin{aligned} x \in \text{int}(\mathcal{P}_i) \quad &\Leftrightarrow O_{i,j}(x) = S_{i,j} \quad \forall j \in C_i \\ &\Leftrightarrow O_i(x) = S_i \end{aligned} \tag{21}$$

Theorem 2 states that the ordering function $O_i(x)$ and the vector S_i uniquely characterize $\mathcal{P}_i$. Therefore, to check on-line if the polyhedral region $\mathcal{P}_i$ contains the state x, it is sufficient to compute the binary vector $O_i(x)$ and compare it to S_i.

Vectors S_i are calculated off-line for $i = 1, \ldots, N_p$, by comparing the values of $f_i(x)$ and $f_j(x)$, $\forall j \in C_i$, in a point that belongs to $\text{int}(\mathcal{P}_i)$, for instance, the Chebysev center of $\mathcal{P}_i$.

2.7. Locating the current state

Algorithm 1 *(GLOBAL (Used for initialization and recovery)).*

$I = \{1, \ldots, N_p\}$
$i \leftarrow I$
$I = I \backslash \{i\}$, $C = C_i$
while $C \neq \emptyset$ **do**
  $j \leftarrow C$, $C = C \backslash \{j\}$
  Compute $O_{i,j}(x)$
  if $O_{i,j}(x) \neq S_{i,j}$ **then**
    if $j \notin I$ **then**
      GOTO step 2
    else
      $i = j$ and GOTO step 3
    end if
  end if
end while

Algorithm 2 *(LOCAL).*

Require Current region i
$C = C_i$ and NotLost = true
while NotLost **do**
  Compute vector $O_i(x)$
  if $O_i(x) \neq S_i$ **then**
    if the difference is at element corresponding to j only **then**
      Set $i = j$ and GOTO step 1.
    else
      Set NotLost = false.
    end if
  end if
end while

Algorithm 3 *(Main program).*

Run Algorithm 1 GLOBAL to find current region i.
while System is operational **do**
  Run Algorithm 2 LOCAL.
  if NotLost = false **then**
    Run Algorithm 1 GLOBAL to find current region i.
  end if
end while

Algorithm 1 was proposed by [3] to find the current state $x(t)$ for explicit MPC. Here, we extend this method by adding Algorithm 2 as a "local" algorithm that for the current polyhedral region only monitors the corresponding ordering function (and thus only "looks" at the neighboring regions). If one element of this vector changes sign, the algorithm updates the current region to the region *corresponding* to the element of the vector that changed sign. However, if more elements changed sign we deduce that the process did not change to a neighboring region and we must run Algorithm 1 again. This logic is covered in the main program in Algorithm 3.

2.8. Finding a scalar PWA descriptor function

A vector-valued PWA descriptor function is defined as:

Definition 4 *(Vector-valued PWA descriptor function).* A continuous vector-valued piece-wise affine (PWA) function

$$m(x) := \overline{A}_i x + \overline{B}_i \quad \text{if } x \in \mathcal{P}_i \tag{22}$$

is called a vector-valued PWA descriptor function if

$$\overline{A}_i \neq \overline{A}_j \quad \forall j \in C_i,\ i = 1, \ldots, N_p, \tag{23}$$

where $\overline{A}_i \in \mathbb{R}^{s \times n_x}$, $\overline{B}_i \in \mathbb{R}^s$, $s \in \mathbb{N}$, $s \geq 2$, and C_i is the list of neighbors of $\mathcal{P}_i$.

Next, the following Theorem gives a method for constructing a scalar PWA descriptor function from a vector-valued one.

Theorem 3 ([3]). *Let $m : \mathbb{R}^{n_x} \mapsto \mathbb{R}^s$ be a vector valued PWA descriptor function defined over a polyhedral partition $\{\mathcal{P}_i\}_{i=1}^{N_p}$. Then there exists a $w \in \mathbb{R}^s$ such that $f(x) := w'm(x)$ is a PWA descriptor function over the same polyhedral partition.*

2.9. Algorithm for finding w

For a given vector-valued PWA descriptor function, we write a set of vectors $a_k \in \mathbb{R}^s$, $\|a_k\| = 1$, $k = 1, \ldots, N_a$, by taking one (and only one) nonzero column from each matrix $(\overline{A}_i - \overline{A}_j)$, $\forall j \in C_i$, $i = 1, \ldots, N_p$. Here, $N_a := \sum_i |C_i|/2$, and $|C_i|$ denotes the cardinality of set C_i. The vector $w \in \mathbb{R}^s$ satisfying the set of equations $w'a_k \neq 0$, $k = 1, \ldots, N_a$, can be constructed using Algorithm 4.

Algorithm 4 *(Construct the vector w).*

$w \leftarrow [1, \ldots, 1]'$, $R \leftarrow 1$
while $k \leq N_a$ **do**
  $d \leftarrow w'a_k$
  if $0 \leq d \leq R$ **then**
    $w \leftarrow w + \frac{1}{2}(R - d)a_k$, $R \leftarrow \frac{1}{2}(R + d)$
  else if $-R \leq d \leq 0$ **then**
    $w \leftarrow w - \frac{1}{2}(R + d)a_k$, $R \leftarrow \frac{1}{2}(R - d)$
  end if
end while

2.10. Properties of the solution of a parametric QP

Consider again the quadratic problem (8):

$$\begin{aligned} &\min_u \frac{1}{2} \begin{bmatrix} u \\ d \end{bmatrix}' \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}' & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix} \\ &\text{s.t. } M_u u \leq M + M_d d \end{aligned}$$

From [3] we have the following properties of the solution to this problem:

Theorem 4. *Consider the parametric QP in (8) and let $J_{uu} > 0$. Then the set $\mathcal{D}$ of feasible parameters d is convex and the optimal input $u^* : \mathcal{D} \mapsto \mathbb{R}^{n_u}$ is continuous and piecewise affine.*

In addition the following Lemma is provided:

Lemma 1. *Let the optimal solution ("optimizer") be written on the form*

$$u = K_d^i d + k_d^i \quad \text{if } d \in \mathcal{P}_i \tag{24}$$

Then, for two neighboring polyhedra $\mathcal{P}_i$, $\mathcal{P}_j$ the gains $K_d^i \neq K_d^j$.

3. Measurement based descriptor function

From now on, the results are new unless otherwise noted.

Lemma 1 states that the optimizer to problem (8) can be written in the form

$$u = K_d^i d + k_d^i \quad \text{if } d \in \mathcal{P}_i. \tag{25}$$

We now want to eliminate the need of information about the disturbances d, but rather rely on plant output y_m.

Assuming a parametric solution exists, we form the following problem for a given set of active inequality constraints:

$$\min_{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}} J(u_1, u_2, d) = \frac{1}{2} \begin{bmatrix} u_1 \\ u_2 \\ d \end{bmatrix}' \begin{bmatrix} J_{u_1u_1} & J_{u_1u_2} & J_{u_1d} \\ J'_{u_1u_2} & J_{u_2u_2} & J_{u_2d} \\ J'_{u_1d} & J'_{u_2d} & J_{dd} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ d \end{bmatrix} \quad \text{subject to } M_{u_1}u_1 + M_{u_2}u_2 = M + M_d d, \tag{26}$$

where we choose u_1 (as a subset of the input vector u) such that M_{u_1} is invertible. This implies that we can write

$$u_1 = \underbrace{-{M_{u_1}}^{-1}M_{u_2}}_{K^{u_2}} u_2 + \underbrace{{M_{u_1}}^{-1}M_d}_{K^d} d + \underbrace{{M_{u_1}}^{-1}M}_{K}. \tag{27}$$

We do the following manipulations to get the problem in a form suitable for Theorem 1: First, we define $z = u_2 + J_{zz}^{-1}J_z$ and

$$J_{zz} = K^{u_2\prime} J_{u_1u_1} K^{u_2} + J_{u_2u_2} + 2K^{u_2\prime} J_{u_1u_2} \tag{28}$$

$$J_{zd} = K^{u_2\prime} J_{u_1u_1} K^{d} + J_{u_1u_2}{}' K^{d} + K^{u_2\prime} J_{u_1d} + J_{u_2d} \tag{29}$$

$$J_z = K' J_{u_1u_1} K^{u_2} + K' J_{u_1u_2} \tag{30}$$

With these definitions it can be shown that the objective function with the active equality constraints substituted into the objective can be written as

$$J(z, d) = \frac{1}{2} z' J_{zz} z + z' J_{zd} d. \tag{31}$$

In addition, we write the linear model as

$$\begin{bmatrix} u_1 \\ u_2 \\ y_m \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \\ G_{u_1}^{y_m} & G_{u_2}^{y_m} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ G_d^{y_m} \end{bmatrix} d \tag{32}$$

Using (27), we find that

$$\begin{bmatrix} u_1 \\ u_2 \\ y_m \end{bmatrix} = \begin{bmatrix} K^{u_2} \\ I \\ \underbrace{G_{u_1}^{y_m} K^{u_2} + G_{u_2}^{y_m}}_{=\tilde{G}_{u_2}^{y_m}} \end{bmatrix} u_2 + \begin{bmatrix} K^d \\ 0 \\ \underbrace{G_{u_1}^{y_m} K^d + G_d^{y_m}}_{=\tilde{G}_d^{y_m}} \end{bmatrix} d + \begin{bmatrix} K \\ 0 \\ G_{u_1}^{y_m} K \end{bmatrix} \tag{33}$$

$$= \underbrace{\begin{bmatrix} K^{u_2} \\ I \\ \tilde{G}_{u_2}^{y_m} \end{bmatrix}}_{G^y} (z - J_{zz}^{-1} J_z') + \underbrace{\begin{bmatrix} K^d \\ 0 \\ \tilde{G}_d^{y_m} \end{bmatrix}}_{G_d^y} d + \underbrace{\begin{bmatrix} K \\ 0 \\ G_{u_1}^{y_m} K \end{bmatrix}}_{\tilde{K}} \tag{34}$$

$$= G^y z + G_d^y d + \underbrace{\tilde{K} - G^y J_{zz}^{-1} J_z'}_{\gamma}. \tag{35}$$

Now, let $\bar{y} = y - \gamma$, where $y = [u_1' \; u_2' \; y_m']'$, and let $\bar{F} = -(G^y J_{zz}^{-1} J_{zd} - G_d^y)$. Further, let $\tilde{H}$ be a full rank matrix that fulfills $\tilde{H}F = 0$. Due to Theorem 1, $\tilde{H}$ can be chosen such that $\tilde{H}\bar{y} = \tilde{H}(y - \gamma) = 0$, hence the invariants are $c = \tilde{H}y$ with $c_s = \tilde{H}\gamma$. Due to the "extra" degrees of freedom in $\tilde{H}$ we can write the combination matrix on the form $\tilde{H} = \begin{bmatrix} I & H^{y_m} \end{bmatrix}$. The extra degrees of freedom in $\tilde{H}$ arise from the fact that if $\tilde{H}F = 0$, then also $D\tilde{H}F = 0$, for a non-singular square matrix D. The matrix D can be used to scale the entries in $\tilde{H}$, or as above to introduce an identity matrix. This is further discussed in [2]. Finally, we can show that the invariant can be written in the form:

$$u = -H^{y_m} y_m + \begin{bmatrix} (K - K^{u_2}) \\ -I \end{bmatrix} J_{zz}^{-1} J_z' + H^{y_m} [G_{u_1}^{y_m} (K - K^{u_2}) + G_{u_2}^{y_m}] J_{zz}^{-1} J_z'. \tag{36}$$

We observe that for a given set of active constraints, there is an affine optimal relationship between the input u and the measurement y_m. For several regions we can therefore pose the following optimal relationship:

$$u = K_{y_m}^i y_m + c_s^i, \quad \text{if } d \in \mathcal{P}_i, \tag{37}$$

where $K_{y_m}^i$ and c_s^i can be found by using the procedure above. The following Lemma shows that the functional relationship between measurement y_m and input u can be used as a vector-valued PWA descriptor function.

Lemma 2. *The invariants defined by* $\text{inv}^i := H^i y - c_s^i$ *can be used as a vector-valued descriptor function*

Proof. Theorem 2 states that the optimizer for problem (8) can be written in the form of Eq. (25). According to Lemma 1, for two neighboring polyhedra $\mathcal{P}_i$, $\mathcal{P}_j$, $K_d^i \neq K_d^j$, and hence the disturbance feedback law (25) is a vector-valued descriptor function.

We now consider the invariants $\text{inv}^i := H^i y - c_s^i$, and we assume that we have a perfect measurement of the input vector included in y:

$$y = \begin{bmatrix} u \\ y_m \end{bmatrix} = \begin{bmatrix} I \\ G^{y_m} \end{bmatrix} u + \begin{bmatrix} 0 \\ G_d^{y_m} \end{bmatrix} d \tag{38}$$

With this partition of y we accordingly write $H^i = [H^{u,i} \; H^{y_m,i}]$. By assumption J_{uu} is positive definite (second-order optimality conditions), and hence the optimal input u must by Theorem 2 be unique and continuous. This has the following implications: First, we can form an equivalent invariant by

$$\text{inv}_{\text{fb}}^i = u - \underbrace{(H^{u,i})^{-1} H^{y_m,i}}_{K_{y_m}^i} y_m + \underbrace{(H^{u,i})^{-1} c_s^i}_{k_{y_m}^i}. \tag{39}$$

For optimality, by Theorem 1, this invariant should be controlled to zero, hence we have the measurement feedback form

$$u = K_{y_m}^i y_m + k_{y_m}^i \quad \text{if } d \in \mathcal{P}_i, \; \forall i = 1, \ldots, N_p \tag{40}$$

Inserting the linear plant model (38), we have that

$$\begin{aligned} u &= K_{y_m}^i (G^{y_m} u + G_d^{y_m} d) + k_{y_m}^i, \\ &= K_{y_m}^i G^{y_m} u + K_{y_m} G_d^{y_m} d + k_{y_m}^i, \\ &\Rightarrow (I - K_{y_m}^i G^{y_m}) u = K_{y_m}^i G_d^{y_m} d + k_{y_m}^i, \\ &\Rightarrow u = (I - K_{y_m}^i G^{y_m})^{-1} K_{y_m}^i G_d^{y_m} d + (I - K_{y_m} G^{y_m})^{-1} k_{y_m}^i. \end{aligned} \tag{41}$$

Second, due to the uniqueness of the optimal input u, we have that the inverse of $(I - K_{y_m}^i G^{y_m})$ must exist and further that

$$(I - K_{y_m}^i G^{y_m})^{-1} K_{y_m}^i G_d^{y_m} = K_d^i \quad i = 1, \ldots, N_p \tag{42}$$

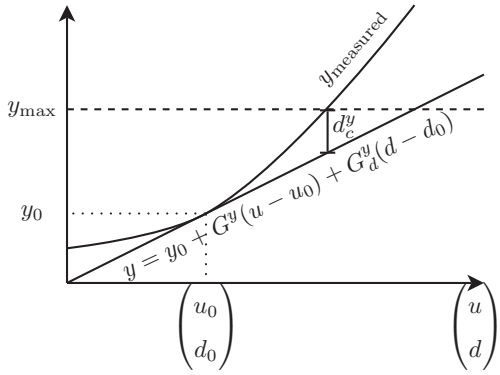

Fig. 3. Additional disturbance d_c^y to match the model with the measured constraint.

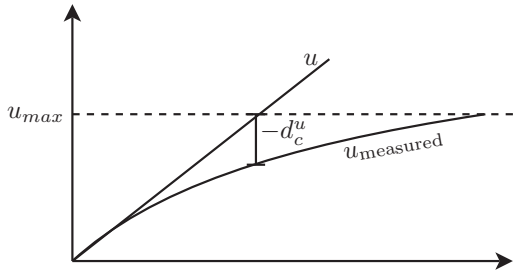

Fig. 4. Matching of an input constraint by additional disturbance d_c^u.

Since both $(I - K_{y_m}^i G^{y_m})^{-1}$ and $G_d^{y_m}$ have full rank, we must have that if $K_d^i \neq K_d^j$, then $K_{y_m}^i \neq K_{y_m}^j$. Finally, since u is continuous we conclude that the function $K_{y_m}^i y_m + k_{y_m}^i$ can be used as a vector-valued PWA descriptor function. □

Remark 2. To use the nullspace method, we do not need to include a perfect measurement of u in y; it is sufficient that we have enough independent measurements $n_y \geq n_u + n_d$. Here we include u because it is then easier to prove that the resulting set of invariants can be used as a vector-valued PWA descriptor function. This means that we can use the method described in this paper to construct a descriptor function as a function of measurements y_m, and include other measurements in the controlled variable selection problem. The only requirement is that the controlled variables gives zero loss from optimality when controlled to constant setpoints c_s^i.

4. Constraint matching

The linear approximation of the constraints as used in problem (8) may, as for any model-based scheme, lead to infeasibility when used on a real plant. However, this can to some extent be accounted for if the constraints are measured. We can then simply estimate a disturbance d_c as illustrated in Fig. 3 and treat this as a measured disturbance in the problem formulation (Problem (8)). For an output constraint we then have

$$\begin{aligned} &y_{\min} \leq y_{\text{measured}} \leq y_{\max} \\ &\Updownarrow \\ &y_{\min} \leq y + d_c \leq y_{\max} \\ &\Updownarrow \\ &y_{\min} - d_c \leq y \leq y_{\max} - d_c \end{aligned}$$

One should realize that this method can (and should) also be used on important manipulated variables u, where important here means inputs that have a strong economic effect, for example inputs that are affecting the throughput of a plant. Say, that for some values of the disturbances it is optimal to implement a certain input at its maximum value, but that there is some mismatch between the model and the reality, as illustrated in Fig. 4. If a measurement of the actual value of u is available, then can this be corrected for by adding an extra disturbance d_c^u as indicated in the figure, and by using the procedure outlined above for the inputs to effectively change the value of $u_{\max}$ in the internal model of the controller.

Remark 3. The purpose of this section is to use extra measurements of active constraints in a systematic manner, to optimize the closed-loop plant performance. If there are no extra measurements available, one must rely on the model.

5. Design procedure

We summarize our findings in the following procedure that may be used to find controlled variables for an economic problem that can be approximated as a quadratic program:

1. Define the steady-state optimal operation problem, consisting of objective function $J_0(x, u, d)$, process model $f_0(x, u, d) = 0$, and operational constraints $g_0(x, u, d)$.
2. Approximate this problem around the nominal optimum as a QP by the method outlined in Section 2.1:
   (a) Formally eliminate the internal states x from the problem.
   (b) Solve the resulting optimization problem for nominal disturbance d_0 to get optimal inputs u^* and optimal Lagrange multipliers λ^*.
   (c) Approximate this problem as a QP around (u^*, λ^*).
3. Add extra disturbances d_c as illustrated in Section 4 for important constraints.
4. Solve the resulting problem as a parametric QP with disturbances d as parameters. The solution will consist of a set of polyhedral regions $\mathcal{P}_i$ in the disturbance space and a list of active constraints for each region.
5. Identify available measurements and linearize to get

$$y_m = G_m^y u + G_d^{y_m} d.$$

6. In each region (in the disturbance space), use the nullspace method of Theorem 1 to find invariants $\text{inv}^i := H^i y - c_s^i$.
7. Use Lemma 2 to make vector-valued PWA descriptor function.
8. Use Algorithm 4 to construct a scalar PWA descriptor function to be used for region detection.
9. Use Algorithm 3 for region detection.

The whole method may be automated, e.g. using the Multi-Parametric Toolbox [9].

Remark 4. (Complexity) Computation of Hessians and gradients may be cumbersome, but not fatal for implementation, as the computations are performed off-line. The on-line complexity of the method is very low, as it is based on results for explicit MPC [3]. For details regarding number of flops and memory requirement, the reader is referred to [3].

6. Example: ammonia production plant

We now develop a simple model of an ammonia synthesis loop to be used as a case study. A process flow sheet is shown in Fig. 5. The main objective is to maximize the produced ammonia in stream 4, while at the same time minimizing the compressor work. The cost function is

$$\begin{aligned} \text{profit} = {} & P_{W_{\text{feed}}} W_{\text{feed}} + P_{W_{\text{recycle}}} W_{\text{recycle}} + P_{W_{\text{cooling}}} W_{\text{cooling}} \\ & + P_{\text{NH}_3} n_4^{\text{NH}_3}, \end{aligned} \tag{43}$$

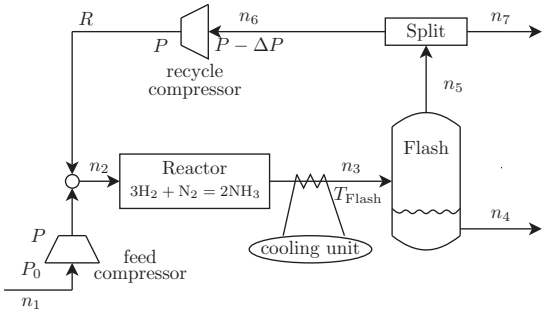

Fig. 5. Sketch of an ammonia synthesis loop.

Table 1
Prices for ammonia example.

Price variable	Value [€/unit]
$P_{W_{\text{feed}}}$	−0.4
$P_{W_{\text{recycle}}}$	−10
$P_{W_{\text{cooling}}}$	−0.5
P_{NH_3}	10^4

with prices given in Table 1. The reactor temperature is fixed, but the cooling affects the separation properties in the flash tank.

In addition there are operational constraints, namely a lower limit on the possible cooling temperature (T_{flash}) and a high limit on the recycle flow (R) in the loop.

There are three steady state degrees of freedom, which can be chosen to be the pressure P, the split factor f (recycle ratio), and the flash temperature T_{flash}.

The feed consists of a mix of H_2 and N_2, and the considered disturbances are feed rate and feed composition.

6.1. Model

The main properties of the model are:

- Equilibrium reactor.
- Henry's law (H_2, N_2) and Raoult's law (NH_3) describe the flash-tank.
- Ideal compressor works.
- Cooling work efficiency given by a Carnot factor.

The variables are the mole vector for each stream, n_i, $i=1,\ldots,n_7$ and the extent of reaction ξ. In addition we use as secondary variable x_i^j to indicate the mole fraction of component j in stream i. The components are ordered by H_2, N_2, NH_3 .

Table 2 shows a list of constants used in the model. All constants are found in the book by Skogestad [18].

The mathematical model is given below.

Table 2
Constants for the ammonia plant example.

Variable	Value	Unit
K_{eq}	$6.36\cdot 10^{-5}$	
$H_0^{H_2}$	210688	
$H_T^{H_2}$	−656	
$H_0^{N_2}$	110816	
$H_T^{N_2}$	−342	
A	4.4854	
B	926.132	
C	−32.98	
d_1^0	5.1	mole/time
d_2^0	0.8	mole fraction

6.1.1. Reactor feed

Mass balance over the feed point:

$$n_2 = n_1 + n_6 \tag{44}$$

6.1.2. Equilibrium reactor

Let $P_i = Px_3$ be the partial pressures in stream 3. The equilibrium relation is then

$$\frac{P_{\text{NH}_3}^2}{P_{\text{H}_2}^3 P_{\text{N}_2}} = K_{\text{eq}}. \tag{45}$$

Further, by using the extent of reaction ξ, we have that

$$n_3 = n_2 + S\xi, \tag{46}$$

where the stoichiometric matrix $S=[-3\ -1\ 2]'$.

6.1.3. Flash tank

We here assume Henry's law for H_2 and N_2 and Rault's law for NH_3. The K-values are given by

$$k_{\text{H}_2} = \frac{H_0^{\text{H}_2} + H_T^{\text{H}_2} T_{\text{flash}}}{P} \tag{47}$$

$$k_{\text{N}_2} = \frac{H_0^{\text{N}_2} + H_T^{\text{N}_2} T_{\text{flash}}}{P} \tag{48}$$

$$k_{\text{NH}_3} = \frac{10^{A-\frac{B}{T_{\text{flash}}+C}}}{P} \tag{49}$$

Let $K = \text{diag}(k_{\text{H}_2}, k_{\text{N}_2}, k_{\text{NH}_3})$ and we have that

$$x_5 = Kx_4. \tag{50}$$

In addition we use the Rachford–Rice equation to find the ratio $r=(\sum n_5)/(\sum n_3)$:

$$\sum_{i=\{\text{H}_2,\text{N}_2,\text{NH}_3\}} \frac{x_3^i(k_i-1)}{1+r(k_i-1)} = 0. \tag{51}$$

Now,

$$e'n_5 = re'n_3 \tag{52}$$

$$e'n_4 = (1-r)e'n_3, \tag{53}$$

where $e' = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$.

6.1.4. Split

The mass balance around the split is

$$n_5 = n_6 + n_7 \tag{54}$$

In addition we have that

$$n_6 = fn_5. \tag{55}$$

For the compressors we have the following models:

6.1.5. Feed compressor

The feed compressor increases the pressure in the feed from the nominal pressure P_0 to the reactor pressure P by

$$W_{\text{feed}} = \sum(n_2)RT\ln\frac{P}{P_0} \tag{56}$$

6.1.6. Recycle compressor

This compressor should counteract pressure drop in the system by

$$W_{\text{recycle}} = \sum(n_6)RT_{\text{flash}}\ln\frac{P}{P-\Delta P} \tag{57}$$

6.1.7. Cooling water

We assume that there is cooling water free of charge that can cool the product stream down to $T_0 = 15\,^\circ\text{C} = 288$ K. For further cooling, we have to use a cooling unit with the following associated work:

$$W_{\text{sub cool}} = \sum n_3 C_P (T_0 - T)\left(\frac{T_0}{T_c} - 1\right), \tag{58}$$

$$\text{where } T_c = \frac{T_0 - T}{\ln(T_0/T)}. \tag{59}$$

This means that the overall energy usage for cooling is

$$W_{\text{cooling}} = \begin{cases} 0 & \text{if } T_{\text{flash}} > T_0 = 288\text{ K} \\ W_{\text{sub cool}} & \text{otherwise} \end{cases} \tag{60}$$

6.2. Disturbances

The only disturbances acting on the system are the feed rate (d_1) and composition (d_2). The feed stream n_1 can therefore be expressed as

$$n_1 = (d_1^0 + d_1)\begin{bmatrix} d_2^0 + d_2 \\ (1 - d_2^0) - d_2 \\ 0 \end{bmatrix}, \tag{61}$$

with $d_1^0 = 5.1$ mole/time as the nominal feed flow and $d_2^0 = 0.8$ as the nominal mole fraction of hydrogen in the feed. The disturbances are assumed to be in the set

$$\mathcal{D} = \{d \in \mathbb{R}^2 \,|\, |d_1| < 1, \quad |d_2| < 0.02\}. \tag{62}$$

This corresponds to a maximum relative change in the feed rate of about 20% and a change in the composition of about 4%.

6.3. Operational constraints

There are two operational constraints that we need to address. First, the cooling unit can only cool the reactor product to $T_{\text{flash,min}} = -7\,^\circ\text{C} = 266$ K, therefore

$$T_{\text{flash}} \geq 266\text{ K}. \tag{63}$$

We assume that this constraint can be implemented exactly, i.e. that an unbiased measurement of this temperature exists.

In addition, there is an upper bound on the maximum flow of recycle. For this constraint, we include an extra disturbance ($d_3 = d_c$) as explained in Section 4 to make sure that we satisfy the upper limit on recycle at all times. (Note that this maximum recycle constraint is motivated by the fact that the recycle compressor has a high limit on the amount of fluid it can process.) We include this correction by the following procedure: First, we find a linear model from (u, d) to n_6 on the form

$$n_6 \approx n_{6,0} + G^{n_6} u + G_d^{n_6} d, \tag{64}$$

using for instance finite differences. Then we add the "constraint matching" disturbance $d_3 = d_c$ to get the following inequality that bounds the maximum recycle in the plant:

$$R_{\text{measured}} \leq R_{\max} \Leftrightarrow e' G^{n_6} u + e' G_d^{n_6} d + 1' n_{6,0} + d_3 \leq r_{\max}.$$

Hence,

$$e' G^{n_6} u \leq (r_{\max} - e' n_{6,0}) + \begin{bmatrix} -e' G_d^{n_6} & -1 \end{bmatrix}\begin{bmatrix} d \\ d_3 \end{bmatrix}, \tag{65}$$

where d are the "economic" disturbances (d_1, d_2), and d_3 is the "constraint matching" disturbance.

Table 3
Nominal optimal inputs.

Steady state degree of freedom	Nominal optimal value
u_1: Split-factor (recycle ratio)	0.6875
u_2: Pressure P in reactor	342.17 bar
u_3: Temperature T_{flash} in flash-tank	266.53 K

6.4. Control structure selection

6.4.1. Nominal operating point

We used Tomlab™ under Matlab™ to find the nominal operating point, reported in Tables 3 and 4. The inputs u used in the sequel are deviation variables from this nominal operating point.

6.4.2. Approximation to a QP

At the nominal optimum, no constraints are active, so we can use the Hessian of the nonlinear problem (rather than using the Lagrangian) to find J_{uu} and J_{ud}. The resulting matrices are:

$$\left[\begin{array}{c|c} J_{uu} & J_{ud} \\ \hline J_{ud}' & J_{dd} \end{array}\right] = \left[\begin{array}{ccc|cc} 55720.17 & -8.01 & 2.62 & 59.53 & 340268.50 \\ -7.95 & 0.08 & -0.08 & 0.02 & 75.11 \\ 2.61 & -0.08 & 1.31 & -0.00 & 7.70 \\ \hline 59.52 & 0.02 & -0.00 & 0.04 & 15695.24 \\ 340081.83 & 74.33 & 7.73 & 15575.95 & 1702546.66 \end{array}\right] \tag{66}$$

This Hessian is found by finite differences. We observe that the matrix is not fully symmetric because of numerical inaccuracy, but it is close enough to symmetric for our purposes. We used the upper right part as J_{ud} in the calculations. The linearized constraints are:

$$\begin{bmatrix} [\max R:] \\ [\min T_{\text{flash}}:] \end{bmatrix} \underbrace{\begin{bmatrix} 9.6295 & -0.0033 & 0.0015 \\ 0 & 0 & -1.0000 \end{bmatrix}}_{M_u}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0.51 \\ 0.53 \end{bmatrix}}_{M} + \underbrace{\begin{bmatrix} -0.59 & -23.76 & -1.00 \\ 0 & 0 & 0 \end{bmatrix}}_{M_d}\begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{67}$$

In addition, there are non-negative constraints on all the compositions and a lower limit on the pressure in the system, but these constraints are not active for the disturbance space we chose to study, so we do not add them explicitly to the problem formulation.

6.4.3. Measurements for region detection

We have three disturbances, but one of them is assumed to be measured (the constraint matching for maximum recycle), hence we need to identify two measurements that we can use for region

Table 4
Nominal optimal stream data.

n_2	n_3	n_4	n_5	n_6	n_6
$\begin{bmatrix} 6.8473 \\ 1.2163 \\ 0.0314 \end{bmatrix}$	$\begin{bmatrix} 4.0455 \\ 0.2837 \\ 1.8995 \end{bmatrix}$	$\begin{bmatrix} 0.0176 \\ 0.0011 \\ 1.8589 \end{bmatrix}$	$\begin{bmatrix} 4.0279 \\ 0.2826 \\ 0.0427 \end{bmatrix}$	$\begin{bmatrix} 2.7683 \\ 0.1953 \\ 0.0304 \end{bmatrix}$	$\begin{bmatrix} 1.2586 \\ 0.0883 \\ 0.0133 \end{bmatrix}$

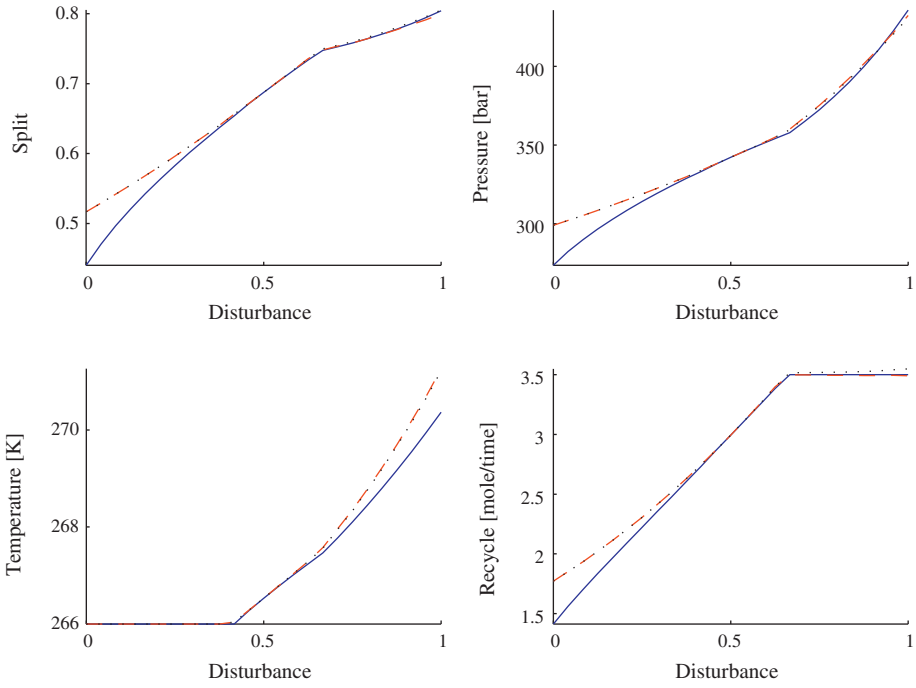

Fig. 6. Calculated steady-state degrees for freedom for the ammonia example for various disturbances. In addition we have plotted the resulting recycle R. The blue line represents the RTO, the red dashed line is the approach of this paper, and the black dotted line is the approach of this paper *without* constraint matching. The disturbance axis represent traversing the disturbance space from $d_{\text{start}} = (-1, 0.02)$ to $d_{\text{end}} = (1, -0.02)$ in a straight line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

detection (see Theorem 1). Since we have no measurement noise and the goal here is to demonstrate how to use this methodology, we simply chose the two first entries of the stream-vector n_2 as measurements, that is the flow of H_2 and N_2 in the reactor feed. This gives the following "measurements" y (in deviation variables) that we use for region detection:

$$y = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ n_2^{H_2} \\ n_2^{N_2} \\ d_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 9.6773 & -0.0024 & 0.0002 \\ -0.1411 & -0.0008 & 0.0002 \\ 0 & 0 & 0 \end{bmatrix}}_{G^y} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1.3468 & 33.8399 & 0 \\ 0.2371 & -10.3119 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{G_d^y} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \tag{68}$$

where the matrices G^y and G_d^y come from linearization around the nominal optimal point.

6.4.4. Parametric solution

Using the "Multi-Parametric Toolbox" (MPT) [9], we identify three regions for the solution of the QP-approximation, which are described in Table 5. We used three parameters in the optimization; (d_1, d_2) with search space defined in equation (62), and in addition $-0.5 < d_3 < 0.5$.

6.4.5. Invariants and vector-valued PWA descriptor function

Using the nullspace method in each region (as described in Lemma 2), we get the following invariants:

$$\text{inv} = -u + \begin{cases} \begin{bmatrix} -0.28 & 1.60 & 0 \\ -76.19 & 427.31 & 0 \\ -4.51 & 25.31 & 0 \end{bmatrix} \begin{bmatrix} y_m \\ d_3 \end{bmatrix} & \text{if } d \in \mathcal{P}_1 \\ \begin{bmatrix} -0.15 & 0.23 & -0.24 \\ -54.97 & 210.37 & -39.94 \\ -3.10 & 10.98 & -2.61 \end{bmatrix} \begin{bmatrix} y_m \\ d_3 \end{bmatrix} + \begin{bmatrix} 0.12 \\ 20.21 \\ 1.32 \end{bmatrix} & \text{if } d \in \mathcal{P}_2 \\ \begin{bmatrix} -0.28 & 1.58 & 0 \\ -70.74 & 396.73 & 0 \\ -0.00 & 0.00 & 0 \end{bmatrix} \begin{bmatrix} y_m \\ d_3 \end{bmatrix} + \begin{bmatrix} -0.0004 \\ -0.6408 \\ -0.5305 \end{bmatrix} & \text{if } d \in \mathcal{P}_3 \end{cases} \tag{69}$$

In order to check the calculations the reader is referred to Section 3.

Remark 5. The invariants "inv" are usually denoted controlled variables "c" in the self-optimizing control terminology. Note however that one may choose a different matrix $H = [H^u \; H^{y_m}]$ for actual

Table 5
Regions of the parametric solution to the QP-approximation.

Region	Description
1	Unconstrained "nominal" region.
2	$R = R_{\max}$ (maximum throughput of the recycle compressor.)
3	$T_{\text{flash}} = T_{\text{flash,min}}$ (cooling unit cannot decrease temperature further.)

Table 6
Neighbors and correct signs for the scalar PWA descriptor $f(y_m)$ as defined in Eq. (70).

Region	Neighbor(s)	$\text{sign}(f_i - f_j)$
1	(2, 3)	$(-1, 1)$
2	1	-1
3	1	1

control; the main focus of this work is to show how the invariants may be used to develop a law for region switching.

Next, using Algorithm 4, we identify the following function which can be used for tracking changes in the active set:

$$f(y_m) := \begin{cases} f_1 = \begin{bmatrix} -81.0 & 454.2 & 0 \end{bmatrix} \begin{bmatrix} y_m \\ d \end{bmatrix} & \text{if } d \in \mathcal{P}_1 \\ f_2 = \begin{bmatrix} -58.2 & 221.6 & -42.8 \end{bmatrix} \begin{bmatrix} y_m \\ d \end{bmatrix} + 21.7 & \text{if } d \in \mathcal{P}_2 \\ f_3 = \begin{bmatrix} -71.0 & 398.3 & 0 \end{bmatrix} \begin{bmatrix} y_m \\ d \end{bmatrix} - 1.18 & \text{if } d \in \mathcal{P}_3 \end{cases} \tag{70}$$

Table 6 shows neighbors and correct signs for the functions f_i in equation (70).

6.5. Simulation results

Fig. 6 shows the result of simulating the proposed control structure for a range of disturbances corresponding to the search-space of the parametric program, that is $-1 \le d_1 \le 1$ and $-0.02 \le d_2 \le 0.02$. The figure shows a traversal of the disturbance space from $d_{\text{start}} = (-1, 0.02)$ to $d_{\text{end}} = (1, -0.02)$ by following a straight line. We chose this representation because this direction was the direction where the methods differed the most. For comparison we have solved the original nonlinear program for the same disturbances.

From Fig. 6 we observe that the two methods give similar results, especially around the nominal disturbance. Fig. 7 shows the difference in cost functions scaled with the absolute value of the optimal cost. We observe that the difference in this metric is quite small, and less than 1% for the cases studied. Probably this will be an "acceptable loss" and we therefore have an implementation that is close to optimal but simple, which is exactly in the spirit of "self-optimizing control."

Fig. 8 shows the estimated nonlinear correction d_3. At saturation of the recycle the actual value is 3.5 mole/time, so the error in

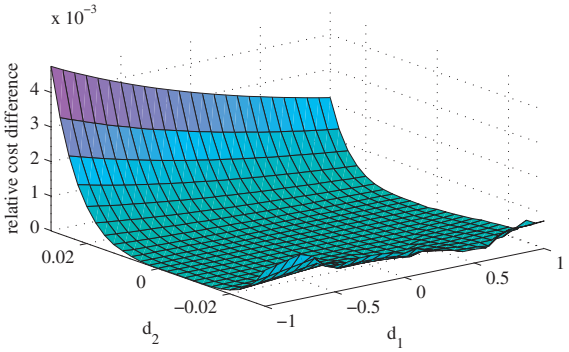

Fig. 7. Relative difference in cost functions with constraint matching implemented.

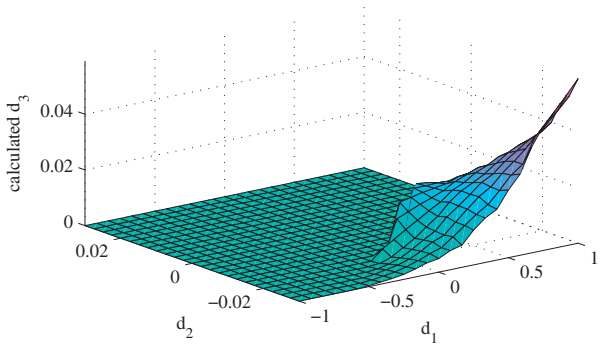

Fig. 8. Constraint matching term d_3.

predicted recycle by the linear model is about 1.4% at maximum. We also simulated the system without this disturbance included as a measurement, and then the constraint on maximum recycle was violated with about 1%. This can be observed from Fig. 6, by observing that the black dotted line is *slightly* above the red line.

7. Discussion

In this paper, we use the descriptor function defined by [3] to implement the solution of a parametric quadratic program. Our main contribution is to relate descriptor functions to implementation of static optimization problems. In particular for quadratic problems, we show that we can identify descriptor functions based on "linear measurements" $y = G^y u + G_d^y d$ by using the nullspace method in each active constraint region. As a result, we can make a list of constant setpoint policies, one for each region of the problem at hand, and a simple method for how to change between these policies, based on the outputs only.

We have only considered steady state, and in fact the region detection scheme assumes that the system is at steady state at all times. This assumption will of course not be valid for real processes, and therefore dynamic studies of control policies with dynamic controller should be conducted before application.

7.1. Measurement noise

Similar to the standard results for explicit MPC, which is the basis for this paper, measurement noise is not directly included. However, since we here deal with the steady-state control layer, sensitivity to noise may be reduced by using i.e. a Kalman filter.

In order to reduce sensitivity of noise further, note that we should choose measurements y such that the gain from u to y is large [2].

As for the implementation of active constraints, one needs to include back-off in order to ensure feasibility if measurement noise is expected to degrade the performance of the control system [5].

As a future research problem, one may consider to use the exact local method [2], which handles measurement noise explicitly, in conjunction with the method developed in this paper.

7.2. Static part of MPC

Model predictive control (MPC) is usually implemented with a static optimization problem that adjusts the setpoints of the controlled variables such that feasibility of the dynamic problem is

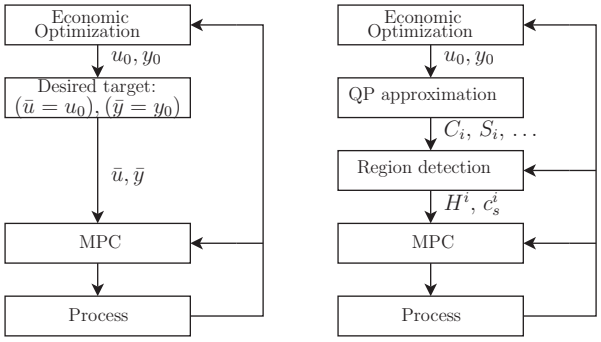

Fig. 9. Left: Typical situation when self-optimizing control is not implemented, the controller should track the optimal values (u_0, y_0) from the real-time optimization (RTO). Right: A possible implementation of self-optimizing control with an RTO layer above. The controller is tracking controlled variables that should give an acceptably small loss from optimality between RTO updates when disturbances occur. Note that "Process", as defined in Fig. 1, consists of both the physical process and the regulatory control layer.

guaranteed. The problem is often referred to as "target calculation", and may have the following structure [16]:

$$\min_{x_s,u_s,\eta} \frac{1}{2}(\eta' W_s \eta + (u_s - \bar{u})R_s(u_s - \bar{u})) + q_s'\eta$$

s.t.

$$\begin{bmatrix} I - A & -B & 0 \\ C & 0 & I \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ \eta \end{bmatrix} \left\{ \begin{matrix} = \\ \geq \\ \leq \end{matrix} \right\} \begin{bmatrix} Bd \\ \bar{y} - p \\ \bar{y} - p \end{bmatrix}$$

$$\eta \geq 0$$

$$u_{\min} \leq Du_s \leq u_{\max}$$

$$y_{\min} \leq Cx_s + p \leq y_{\max}$$

Here $\bar{y}$ and $\bar{u}$ are desired (assumed economically optimal) values for the measurements and inputs, while η is a slack variable.

However, our method may also guarantee feasibility if we can estimate (by using "constraint matching") the deviation from predicted and actual value of the output constraints. This is because the controlled variables $c = Hy$ are by construction feasible at their setpoints c_s (also for the *actual plant* when "constraint matching" is used). Hence, the method presented in this paper may be used as an alternative to the steady-state part of the MPC, with the benefit of improved economic performance of the plant.

A similar idea is presented in [20], but the authors do not consider "feedback implementations" on the form of controlling $c = Hy$ to a setpoint c_s, rather they consider an open-loop implementation of the static problem. Similar to what we do in this paper, Ying and Joseph also suggest to use the Hessian of the Lagrangian of a quadratic approximation of the RTO as a quadratic weight in the feasibility problem of the MPC.

7.3. Using self-optimizing control with RTO

In the example we assumed that the feed composition could change with about 4%. Optimal economic operation of the plant was found to be a strong function of this disturbance, which is also clear from e.g. Fig. 6 where one observes that one has to change the inputs considerably when the feed composition changes. For larger disturbances in the feed composition, say 10%, the self-optimizing control scheme will generate inputs that are quite far away from the optimal inputs, and there will be a significant loss. In these situations it would be fruitful to *update* the self-optimizing control policy by using an RTO layer (economic optimization) above the self-optimizing layer. A flow-sheet of a possible implementation is shown in Fig. 9.

In the figure we show both a typical scheme where an economic optimization layer sends a desired target value to the control layer, and a situation where one implements the scheme presented in this paper. The scheme on the right hand side of the figure would typically be interesting if the economic optimization is updated every now and then (assuming that a fast update is too difficult). Such a scheme should be able to handle larger disturbances in the feed composition and still have an acceptably small loss from optimality.

8. Conclusion

Based on a recent contribution by [3] we have presented a generalization of the nullspace method [1] to include changes in the optimal active set. The method has been demonstrated on a model of an ammonia production facility. We identified three different regions of operation, and the method was comparable in performance to real-time optimization of the same plant.

References

[1] V. Alstad, S. Skogestad, Null space method for selecting optimal measurement combinations as controlled variables, Ind. Eng. Chem. Res. 46 (2007) 846–853.
[2] V. Alstad, S. Skogestad, E. Hori, Optimal measurement combinations as controlled variables, J. Process Control 19 (1) (2009) 138–148.
[3] M. Baoticc̀, F. Borrelli, A. Bemporad, M. Morari, Efficient on-line computation of constrained optimal control, SIAM J. Control Optim. 47 (5) (2008) 2470–2489.
[4] W. Findeisen, F. Bailey, M. Bryds, K. Malinowski, P. Tatjewski, A. Wozniak, Control and Coordination in Hierarchical Systems, John Wiley & Sons, 1980.
[5] M. Govatsmark, S. Skogestad, Selection of controlled variables and robust setpoints, Ind. Eng. Chem. Res. 44 (7) (2005) 2207–2217.
[6] I.J. Halvorsen, S. Skogestad, J.C. Morud, V. Alstad, Optimal selection of controlled variables, Ind. Eng. Chem. Res. 42 (2003) 3273–3284.
[7] E. Hori, S. Skogestad, Selection of controlled variables: maximum gain rule and combination of measurements, Ind. Eng. Chem. Res. 47 (2008) 9465–9471.
[8] V. Kariwala, Y. Cao, S. Janardhanan, Local self-optimizing control with average loss minimization, Ind. Eng. Chem. Res. 47 (2008) 1150–1158.
[9] M. Kvasnica, P. Grieder, M. Baotić, Multi-Parametric Toolbox (MPT), 2004. http://control.ee.ethz.ch/ mpt/.
[10] H. Manum, S. Narasimhan, S. Skogestad, Explicit MPC with output feedback using self-optimizing control, in: Proceedings of IFAC World Conference, Seoul, Korea, 2008.
[11] H. Manum, S. Narasimhan, S. Skogestad, A new approach to explicit MPC using self-optimizing control, in: Proceedings of American Control Conference, Seattle, USA, 2008.
[12] T. Marlin, A. Hrymak, Real-time operations optimization of continuous processes, in: Proceedings of Chemical Process Control-V, Tahoe City, Nevada, USA, 1996.
[13] 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.
[14] M. Morari, J.H. Lee, Model predictive control: past, present and future, Comput. Chem. Eng. 23 (1999) 667–682.
[15] J. Nocedal, S.J. Wright, Numerical Optimization, Springer Series in Operations Research, 1999.
[16] J. Rawlings, Tutorial overview of model predictive control, IEEE Control Syst. Mag. 20 (June (3)) (2000) 38–52.
[17] S. Skogestad, Plantwide control: the search for the self-optimizing control structure, J. Process Control 10 (2000) 487–507.
[18] S. Skogestad, Chemical and Energy Process Engineering, CRC Press, 2009.
[19] J. Spjøtvold, E.C. Kerrigan, C.N. Jones, P. Tøndel, T.A. Johansen, One the facet-to-facet property of solutions to convex parametric quadratic programs, Automatica 42 (2006) 2209–2214.
[20] C. Ying, B. Joseph, Performance and stability analysis of LP-MPC and QP-MPC cascade control systems, AIChE J. 45 (7) (1999) 1521–1534.