

# Improved Branch and Bound Method for Control Structure Screening

Yi Cao <sup>a,\*</sup> Prabirkumar Saha <sup>a,1</sup>

<sup>a</sup>*School of Engineering, Cranfield University, Bedford MK43 0AL, UK*

---

## Abstract

The main aim of this paper is to present an improved algorithm of “Branch and Bound” method for control structure screening. The new algorithm uses a *best-first search* approach, which is more efficient than other algorithms based on *depth-first search* approaches. Detailed explanation of the algorithms is provided in this paper along with a case study on Tennessee-Eastman process to justify the theory of Branch and Bound method. The case study uses the Hankel Singular Value to screen control structure for stabilization. The branch and bound method provides a global ranking to all possible input and output combinations. Based on this ranking an efficient control structure with least complexity for stabilizing control is detected which leads to a decentralized proportional controller.

*Key words:* Control Structure Screening, Branch and Bound method, Hankel Singular Value, Tennessee-Eastman Process, Process Control, Stability, Systems Engineering, Optimization

---

## 1 INTRODUCTION

Control structure screening has been a challenging research topic among control engineers during the last decade (Heath et al., 2000; Kookos and Perkins, 2001). It is more difficult than synthesizing a controller, specially for large sets of measurements and actuators, because design of control structure is a combinatorial problem. With a large number of candidate measurements and manipulations the number of possible combinations of inputs and outputs have

---

\* Author for correspondence (*Email:* y.cao@cranfield.ac.uk; *Fax:* +44-1234-750728)

<sup>1</sup> Current address: Department of Chemical Engineering, Indian Institute of Technology Guwahati, North Guwahati 781039, INDIA

a combinatorial growth. Thus, an approach consisting of performing a controllability analysis for each possible combination becomes time consuming (Havre and Skogestad, 1996). Tennessee-Eastman test-bed problem (Downs and Vogel, 1993) can be referred as an exemplary process due to its umpteen I/O combinations (41 measurements out of which 22 are potential outputs; 12 manipulated variables). In this case, the total number of possible input-output combinations is  $\sum_{j=1}^{12} C_j^{12} \sum_{i=1}^{22} C_i^{22} = (2^{12} - 1)(2^{22} - 1) = 17,175,670,785$ . Suppose it takes only 0.01 sec to evaluate one I/O combination, it would still take more than 5 years for an exhaustive search to reach a global conclusion. Therefore an efficient algorithm is essential to tackle the combinatorial problem.

Before discussing the importance of control structure screening, it is necessary to recapitulate the basics of control structure selection problem. Figure 1 shows the general structure of a feedback process. For a set of exogenous inputs ( $w$ ) to the process, a set of inputs ( $u$ ) and outputs ( $y$ ) are to be selected such that a compensator  $K$  can be designed to achieve cost function  $z$  within a specified limit. It is understandable that more number of  $u$  and  $y$  selected will generally yield better performance. However, it is intended to minimize the number of  $u$  and  $y$  (only necessary ones to be selected) to restrict the complexity of the feedback controller. In this perspective, control structure screening is important because of the following issues:

- Usually control structure selection is a multiobjective problem.
- Some objectives may not be suitable to be considered together mathematically
- Even a mixed cost function can be formulated, but it may not be computationally tractable.

A possible solution to the above problems is to separate the objectives into two groups: firstly only use computationally tractable indices suitable for control structure screening to reduce the candidate set, and then perform exhaustive search in the reduced set for the rest objectives.

Branch and Bound (BAB) as a general approach to solve combinatorial problem is well-known. It has been widely used in numerical algorithms for solving Mixed-Integer Linear and Nonlinear Programming (MILP and MINLP) problems (e.g. Luyben and Floudas, 1994a,b; Braatz et al., 1996) and global optimization softwares such as BARON computational system for solving non-convex optimization problems to global optimality (Gutierrez and Sahinidis, 1996; Ryoo and Sahinidis, 1996). Several researchers have demonstrated applications of BAB algorithm in Chemical Engineering and Process Control as well as in Control Structure Selection (Raman and Grossmann, 1991; Skrifvars et al., 1996). However, most of the available BAB methods are integrated in general MINLP and MILP software, and therefore is difficult to be tailored

for control structure screening.

A particular BAB method for subset selection was proposed by Narendra and Fukunaga (1977) in the context of pattern recognition. Roberts (1984) presented a FORTRAN code for the algorithm. Over the years, the algorithm was improved by several authors (Ridout, 1988; Yu and Yuan, 1993). Search speed has been considerably increased through the Improved BAB method by Chen (2003). Basically, it is based on an asymmetrical *solution tree*, whose leaves represent target subsets of optimal features, while the root represents the set of all features. The algorithm is based on the assumption that the adopted criterion function fulfills the monotonicity condition to simplify the bound estimation. Thus, any subset screening problem, with criterion that satisfies monotonicity property, can be efficiently solved using this method. In other words, this method is particularly suitable for control structure screening based on simple controllability indices. Cao et al. (1998) modified this method to use with the minimum singular value for control structure selection. This work extends this method to address the following screening issues:

- (1) The screening procedure involves multiobjective search, some objectives may not satisfy monotonicity required by BAB algorithm, for example, the right-half-plane zeros. To reduce the search space for these objectives, BAB based on a particular criterion which satisfies monotonicity should have the screening feature, *viz.*,
  - (a) searching all subsets that satisfies  $\phi \geq a$ , where  $\phi$  is the criterion function and  $a$  is a bound (Braatz et al., 1996)
  - (b) ranking all possible subsets to find a certain number of best subsets.
- (2) There should not be any restriction over either the subset-size or number of input(s) and/or output(s), *i.e.*, searching algorithm should be able to identify the best possible subset(s) for any series of pre-selected subset-sizes irrespective of whether it is square system or nonsquare.
- (3) There should be a provision for joint I/O screening.

The paper is organized as follows. The theoretical details of Branch and Bound method are discussed in the next section. It includes the basic principle of the theory, newly improved algorithm and the discussions about the special requirements of control structure screening, which are met by the new algorithm. Section 3 presents the focused criteria function, *viz.* the minimum Hankel singular value, which is used for controllability analysis for unstable plants. Application of Branch and Bound method is shown in section 4 through a case study on Tennessee-Eastman test bed process. A brief conclusion on the research finding is presented in section 5.

## 2 BRANCH AND BOUND METHOD

### 2.1 Principle

Let  $X_N = \{x_1, \dots, x_N\}$  be an  $N$ -element set and  $X_n$  be an  $n$ -element subset selected from  $X_N$ . There are  $C_N^n = N!/(N-n)!n!$  candidate  $n$ -element subsets  $X_n \in X_N$  for screening with criterion cost function  $\phi$ . The objective is to find an  $n$ -element subset,  $X_n^*$ , which satisfies

$$\phi(X_n^*) = \max_{X_n \subseteq X_N} \phi(X_n) \quad (1)$$

Let  $\bar{\phi}(X_N)$  be an upper bound estimation of the cost function over all possible subsets of  $X_N$ , i.e.,

$$\bar{\phi}(X_N) \geq \max_{X_n \subseteq X_N} \phi(X_n) \quad (2)$$

Therefore,

$$\bar{\phi}(X_s) \leq \bar{\phi}(X_t) \quad \text{if } X_s \subseteq X_t \quad (3)$$

Particularly, if  $\phi$  satisfies monotonicity property, i.e.,

$$\phi(X_s) \leq \phi(X_t) \quad \text{if } X_s \subseteq X_t \quad (4)$$

then estimation can be simplified as,

$$\bar{\phi}(X_N) = \phi(X_N) \quad (5)$$

Let  $B$  be a lower bound on the optimal value of  $\phi(X_n^*)$ , i.e.,

$$B \leq \phi(X_n^*) \quad (6)$$

If a  $k$ -element subset,  $X_k$ , with  $k > n$  satisfies

$$\bar{\phi}(X_k) < B \quad (7)$$

then, it can be shown by (3),

$$\bar{\phi}(X'_n) < B, \quad \forall X'_n \subseteq X_k \quad (8)$$

This means that none of the subsets of  $X_k$  can be the optimal solution, thus they can be discarded without any further evaluation. By applying this principle properly, the computational load can be reduced considerably.

The above BAB principle can be applied to any criterion, where an algorithm is required to generate an upper-bound estimation of the criterion over all possible subsets of the given set, *i.e.* (2). The simplest case is where the criteria

satisfy monotonicity (4). Hence, no upper-bound estimation is required in this case. For simplicity, the monotonicity condition is presumed in the rest of the section.

## 2.2 Branch rule

An efficient BAB algorithm requires a proper branching strategy. In a traditional BAB algorithm, branching is represented as a solution tree where each subset belongs to one and only one branch (Figure 2). However, structure of the solution tree is depend on the target subset-size. Thus, it is not suitable for a varying subset-size problem. By introducing the concepts of fixed-set and candidate-set, a new branching rule, which suitable for both fixed and variable subset-size problems, is proposed.

Consider a simple example with  $X_2 = \{x_1, x_2\}$ ,  $X_{3,1} = \{x_1, x_2, x_3\}$  and  $X_{3,2} = \{x_1, x_2, x_4\}$ . Clearly,  $X_2$  can be generated from both  $X_{3,1}$  and  $X_{3,2}$ . If  $X_2$  can only be generated from  $X_{3,1}$ , then,  $x_4$  cannot be eliminated from  $X_{3,2}$  (*fixed element*), whilst  $x_3$  must be removable from  $X_{3,1}$  (*candidate element*). More generally, for an  $n$ -element set  $X$ , the fixed-set and candidate-set of  $X$  are defined as follows.

**Definition 1 (Fixed set  $F$ )** *The fixed set,  $F \subseteq X$  is the smallest subset of  $X$ , whose elements are always included in all subsets generated from  $X$ . It has number of  $n_f$  elements.*

**Definition 2 (Candidate set  $C$ )** *The candidate set is the set of elements which are elements of  $X$ , but not elements of  $F$ , i.e.  $C = X - F$ . It has  $n_c = n - n_f$  number of elements.*

According to the definitions, a set  $X$  can be represented by a three-tuple,

$$X = (\phi, F, C)$$

The following branch rule, illustrated in Figure 3, determines how  $F$  and  $C$  are propagated into subsets.

**Definition 3 (Branch rule)** *Assume  $X = (\phi, F, C)$  and  $C = \{x_1, x_2, \dots, x_{n_c}\}$ . Let  $X^k = X - \{x_k\}$ . Then,*

$$X^k = (\phi_k, F^k, C^k), \quad F^k := F \cup \{x_1, \dots, x_{k-1}\}, \quad C^k := \{x_{k+1}, \dots, x_{n_c}\} \quad (9)$$

Starting from the full set,  $X_N = (\phi(X_N), \{\}, X^N)$  and repeatedly applying the branch rule, all subsets are properly branched.

**Lemma 4 (Branch completeness)** *The branch rule of (9) is complete without any redundancy, i.e. each subset belongs to one and only one branch.*

**PROOF.** Let  $\mathcal{S}(X)$  denote all subsets generated by  $X$ . Consider,  $1 \leq i < j \leq n_c$ . Then, according to the branch rule,  $x_i \in X_s$  if  $X_s \subseteq \mathcal{S}(X^j)$ . However,  $x_i \notin X_t$  if  $X_t \subseteq \mathcal{S}(X^i)$ . Thus,  $\mathcal{S}(X^i) \cap \mathcal{S}(X^j) = \emptyset$ , i.e. no subset belongs to more than one branch. On the other hand, the total number of second-level subsets of  $X$  is  $(n_c - 1) + (n_c - 2) + \dots + 1 + 0 = n_c(n_c - 1)/2 = C_{n_c}^2$ . Thus, the second-level subsets of  $X$  are complete and without any redundancy. It infers that all subsets of  $X$  are complete without redundancy.  $\square$

**Lemma 5** *For an  $m$ -element subset selection problem, let  $n_b$  be the number of expandable branches of  $X$ . Then,*

$$n_b = m - n_f + 1 \quad (10)$$

**PROOF.** Any set with more than  $m$  elements fixed cannot generate an  $m$ -element subset, thus can be discarded. Hence, the number of branches of a set is equal to the number of its subsets, which has their fixed-set sizes less or equal to  $m$ . If a set has  $n_b$  branches, then, according to (9), the size of  $F^{n_b}$  is  $m$ , i.e.  $m = n_f + n_b - 1$ , which leads to  $n_b = m - n_f + 1$ .  $\square$

**Remark 6** *If a node has only one branch, there is no need to evaluate intermediate nodes one by one until the terminal node. Based on this observation, Yu and Yuan (1993) developed their  $BB^+$  algorithm, which has been shown to be faster than the basic BAB algorithm. To implement the same strategy with the above branch rule, a subset with  $n_f = m$  should be directly converted to the terminal subset by eliminating  $C$ .*

**Remark 7** *The branch rule (9) is based on the element-index of  $C$ . In theory, the index-sequence is local (only relevant to the first-level subsets) and can be arbitrary. For efficiency (Ridout, 1988), the sequence of subset is ordered as  $\phi(X^1) \leq \phi(X^2) \leq \dots \leq \phi(X^{n_c})$  so that the set with least criterion value has the largest candidate set, which are more likely to be discarded although this will increase the number of criterion evaluations (number of subsets evaluated more than number of branches). For some problems, where the first-level subset criterion functions can be efficiently evaluated using an iterative formulation, subset criterion sorting will significantly reduce computational load.*

**Remark 8** *In the data structure of a three-tuple  $(\phi, F, C)$ ,  $F$  and  $C$  are fixed-length integers, where each bit represents an element in the set (bit=1) or not in the set (bit=0). Thus, the branch rule can be efficiently programmed using integer operations.*

### 2.3 Pool of subsets

To solve a variable subset-size selection problem, the concept of *pool of subset* is adopted from Yu and Yuan (1992). The pool,  $\mathcal{P}$  is a dynamic queue, which consists of all active subsets to be visited. All subsets are stored in the three-tuple structure and sorted according to their criterion values. Initially,  $\mathcal{P}_0 = \{X_N\}$ . Assume at  $k$ -th iteration,

$$\begin{aligned} \mathcal{P}_k &= \{X^1, X^2, \dots, X^n\}, \\ X^i &= (\phi_i, F^i, C^i) && \text{for } i = 1, \dots, n, \\ \phi_1 &= \max(\phi_1, \dots, \phi_n). \end{aligned} \tag{11}$$

Then, the pool is dynamically updated by the following expanding and contracting rules.

**Definition 9 (Pool expanding)** *Let  $\mathcal{S}(X^1)$  be the set of all first-level subsets of  $X^1$  generated by eliminating one element from  $C^1$  using the branch rule (9). Then, the pool is expanded as*

$$\mathcal{P}_{k+1} = \mathcal{S}(X^1) \cup (\mathcal{P}_k - \{X^1\}) \tag{12}$$

**Definition 10 (Pool contracting)** *Assume the target subset size is  $m$ . Let  $n_{f,i}$  be the size of  $F^i$ ,  $\mathcal{D} = \{X^i | n_{f,i} \geq m\}$  and  $\mathcal{T} = \{(\phi(F^i), F^i, \{\}) | n_{f,i} = m\}$ . Then, the pool is contracted as*

$$\mathcal{P}_{k+1} = \mathcal{T} \cup (\mathcal{P}_k - \mathcal{D}) \tag{13}$$

The optimal subset of a pool is determined by the following theorem.

**Theorem 11 (Global optimum)** *For  $\mathcal{P}_k$  in (11), let  $n_i$  be the size of  $X^i$ , for  $i = 1, \dots, n$ . Then  $X^1$  is globally optimal among all  $n_1$ -element subsets if  $n_1 = \min(n_1, n_2, \dots, n_n)$ .*

**PROOF.** Since  $\mathcal{P}_k$  consists all active subsets and  $n_1$  is the minimum subset size among all pool members, any  $n_1$ -element subset will be a subset of one of the pool members. However,  $X^1$  has the largest criterion among all pool members. Hence, it is globally optimal among all  $n_1$ -element subsets.  $\square$

**Remark 12** *The pool algorithm can be applied to both fixed and variable subset-size selection problems. For a fixed-size problem, the size of the first subset,  $n_1$  in the current pool is compared with the target size,  $m$ . The pool will be continuously updated until  $n_1 = m$ . For a variable-size problem, the multiple target sizes,  $m_1, m_2, \dots, m_k$  are sorted in descending order. Then,*

the pool is dynamically updated to find  $n_1$  matching  $m_1$ , then  $m_2$ , until all target sizes being matched.

**Remark 13** *The pool algorithm can also be applied to find  $l$  best  $m$ -element subsets. In this case, when the first member of a pool is size of  $m$ , it will be fixed. The pool is continuously updated to find the second member with size  $m$ . And so on, until the first  $l$  members are all of size  $m$ . Alternatively, if the screening problem is to find all  $m$ -element subsets which have criterion value larger than a given threshold, then any pool members whose criterion value less than the threshold will be removed from the pool and only subsets with size larger than  $m$  will be expanded. The final pool contains the solution sets.*

**Remark 14** *The pool algorithm described above is a best-first-search approach. It is normally more efficient than traditional BAB algorithms, which are based on a depth-first-search solution tree (e.g. Cao et al., 1998) in terms of the number of subsets expanded. In a depth-first-search algorithm, to determine if a subset should be expanded, the criterion value of the subset is compared with the current best bound of the optimal solution,  $B$  in (6). However, in the best-first-search algorithm, the subset criterion value is compared with those of all pool members (including  $B$ ). Hence, the subset is less likely to be expanded than in a depth-first-search algorithm.*

**Remark 15** *Recently, Chen (2003) spotted some redundant evaluations in BAB algorithms, which can be removed to improve efficiency. That is, when a non-terminal subset,  $X$  is discarded ( $\phi(X) \leq B$ ), all natural subsets of  $X$ , which may not belong to the branch of  $X$ , can also be discarded. In the pool-based BAB algorithm, this is implemented by checking the first pool-set,  $X^1$  before expanding. If  $C^1$  has an element,  $x_k$ , which, when eliminated, will make the generated subset be a natural subset of a pool member ranked after  $B$ , then  $x_k$  will be moved from  $C^1$  to  $F^1$ .*

The pool-based branch and bound algorithm is summarized in the following procedures.

- Step 1** Define  $X_N = (\phi, \{\}, X_N)$ , a target-size array,  $M = \{m_1, m_2, \dots, m_k\}$  sorted in descending order, and the number of best subsets required,  $l$ . Initialize a subset pool,  $\mathcal{P} = \{X_N\}$  and a pointer of the target-size array,  $I = 1$ .
- Step 2** If  $I > k$  stop. Otherwise, get current target size  $m = M(I)$ ,  $I = I + 1$ , and initialize a pointer,  $L = 0$  to indicate the number of best subsets obtained.
- Step 3** Let  $\mathcal{P} = \{X_{p_1}, \dots, X_{p_n}\}$  be the current pool. Sort  $\phi(X_{p_i})$  for  $i = L + 1, \dots, n$  in descending order.
- Step 4** If  $p_{L+1} = m$ , then  $X_{p_{L+1}}$  is a new best-subset,  $L = L + 1$ . If  $L = l$ , then all best-subsets required have been found, goto Step 2. Otherwise, continue.

- Step 5** Check  $X_{p_{L+1}}$  and adjust its  $F$  and  $C$  according to Remark 15.  
**Step 6** Perform pool expanding using (12).  
**Step 7** Perform pool contracting using (13).  
**Step 8** Return to Step 3.

#### 2.4 Joint I/O screening

In most cases, due to lack of suitable tools, the importance of inputs and outputs have to be assessed separately i.e., inputs have to be screened with a output set fixed, whilst to screen outputs, an input set has to be presumed. Obviously, such a way to screen inputs and outputs cannot reveal the effect of cross-combination between inputs and outputs. As a result, global solution of the best I/O combination is not guaranteed. To solve this problem, input and output variables are considered together in the Branch and Bound search procedure via an index translation scheme. More specifically, the  $k$ -th input is the  $k$ -th variable of the joint I/O set, whilst the  $k$ -th output is the  $M + k$ -th variable in the I/O set (where  $M$  is the total number of inputs). In this way, any input-output combination can be translated as a subset of the total I/O set. To avoid a subset with pure inputs or pure output being selected, the criterion function of such a subset is defined as zero. Therefore, such a subset will never be selected by the Branch and Bound algorithm.

Finally the screening result can be represented graphically for criterion function vs. I/O subset-size. In this way, the screening is done both vertically (a group of subsets with the same size) as well as horizontally (subsets with different size). Wherein the best I/O combination could be determined at the point where the increase of criterion value against the increase of subset-size becomes negligible, or at the point where the controllability criterion is at a satisfactory level but the I/O subset-size is minimum.

### 3 STABILIZING CONTROL STRUCTURE SELECTION

Stabilization of an unstable plant is mandatory before designing higher level control structure of the system. Therefore, as argued by Havre and Skogestad (2003), stabilizing control usually belongs to a secondary control layer, where output performance is not an important issue because the output setpoint is normally determined by a higher level controller. Instead, the input usage is more important for stabilizing control because it directly relates to possible input saturation, which may lead to instability. Glover (1986) has proven that the minimum input usage to stabilize an unstable plant can be quantified by the minimum Hankel Singular Value (HSV) of the anti-stable projection of the

plant. This result has been applied to stabilizing control structure selection Cao and Saha (2003). However, global optimality is not guaranteed in the direct application of the minimum HSV because of the combinatorial nature of the evaluation.

In this section, an algorithm is proposed to evaluate the minimum HSV based on the BAB approach described in Section 2. This algorithm gives global ranking of minimum HSV for all possible I/O combinations without exhaustive evaluation of the index.

### 3.1 Hankel Singular Value

Consider a linear system with  $n_x$  states,  $n_y$  outputs and  $n_u$  inputs, denoted by  $G = \{A, B, C\}$ , where  $A$ ,  $B$  and  $C$  are state, input and output matrices of the state space equations.

The controllability and observability gramians,  $L_c$  and  $L_o$  of the system are symmetric matrices and satisfy the following Lyapunov equations:

$$AL_c + L_cA^T + BB^T = 0 \quad (14)$$

$$A^TL_o + L_oA + C^TC = 0 \quad (15)$$

The Hankel Singular Value,  $\underline{\sigma} = \sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_{n_x} = \bar{\sigma}$  of the system is defined as the square roots of the eigenvalues of  $L_oL_c$ , i.e.,

$$\sigma_k(G) = \sqrt{\lambda_k(L_oL_c)}; \quad k = 1, 2, \dots, n_x \quad (16)$$

For a subsystem with the  $i$ -th output and  $j$ -th input, its controllability and observability gramians and HSV are denoted as  $L_{cj}$ ,  $L_{oi}$  and  $\sigma_k^{ij}$ , respectively. Then according to (14) and (15) the following equations can be obtained:

$$L_c = \sum_{j=1}^{n_u} L_{cj} \quad L_o = \sum_{i=1}^{n_y} L_{oi} \quad (17)$$

If the system,  $G$  possesses unstable poles, then it is possible to decompose the system into stable ( $G_s$ ) and anti-stable ( $G_a$ ) projections, i.e.,  $G = G_s + G_a$  (Safanov et al., 1987). Glover (1986) studied the robust stabilization problem of a linear multivariable open-loop unstable system  $G$ . The task for a controller  $K$  to stabilize  $(G + \Delta)$  for all allowable perturbations  $\Delta$  with minimum control effort is to minimize  $\|KS\|_\infty$ , where  $S := (I - GK)^{-1}$  is the sensitivity matrix.

It has been proved by Glover (1986) that

$$\min \|KS\|_\infty = \frac{1}{\underline{\sigma}([G_a(-s)]^*)} \quad (18)$$

According to Glover's theorem, it is desired to select an input and output combination which has the largest minimum HSV. Understandably, it is possible to extract subsystems with  $m$  selected inputs and  $n$  selected outputs from  $G$ . Suppose the selected inputs and outputs have index sets,  $J = \{j_1, j_2, \dots, j_m\}$  and  $I = \{i_1, i_2, \dots, i_n\}$ , respectively. Controllability and observability gramians for any such subsystem will be

$$L_{cJ} = \sum_{j \in J} L_{cj} \quad L_{oI} = \sum_{i \in I} L_{oi} \quad (19)$$

The HSV can be calculated in a similar manner as in (16). Hence, the best combination of inputs and outputs are to be detected by comparing the minimum HSV.

There are a couple of issues involved with the HSV which are discussed below:

- *Pure integrator:* One particular issue with the HSV is that it is not well-defined for systems which have pure integrators, i.e., poles at zero. Presence of pure integrators in the system will invalidate the above theory as the matrix  $A$  in (14) and (15) will be singular. To counter this problem, a diagonal matrix, having a small constant as elements, can be added to the system matrix  $A$  before decomposing it into stable and anti-stable parts. Through this operation, the poles will be shifted slightly towards right without disturbing overall characteristics of the process. However, the pure integrators will move to the right-half-plane (RHP) of imaginary axis and will be converted into small positive poles which can later be extracted in  $G_a$ .
- *Combinatorial problem:* The selection of inputs and outputs by using the HSV is a combinatorial problem. This problem can be tackled by using the BAB algorithm described previously.

The following theorem shows that the minimum HSV satisfies the monotonicity condition (4). Therefore the BAB approach can be simplified to take this advantage as explained in Sec 2.

**Theorem 16** *If  $I_1 \subseteq I_2$  and  $J_1 \subseteq J_2$  then  $\underline{\sigma}^{I_1 J_1} \leq \underline{\sigma}^{I_2 J_2}$*

**PROOF.** If  $\underline{\sigma}^{I_2 J_2} = 0$ , then  $\underline{\sigma}^{I_1 J_1} = 0$ . For the case  $\underline{\sigma}^{I_2 J_2} \neq 0$ ,  $L_{oI_2}$  and  $L_{cJ_2}$  are symmetric and positive definite. Let  $J_1 \subseteq J_2$  and  $I_1 = I_2 = I$ , then  $L_{cJ_2} = L_{cJ_1} + L_{c(J_2-J_1)}$ . Thus,  $\lambda(L_{oI} L_{cJ_2}) = \lambda(L_{oI}^{1/2} L_{oJ_2} L_{oI}^{1/2}) = \lambda(L_{oI}^{1/2} L_{cJ_1} L_{oI}^{1/2} + L_{oI}^{1/2} L_{c(J_2-J_1)} L_{oI}^{1/2})$ . As  $L_{oI}^{1/2} L_{cJ_1} L_{oI}^{1/2}$  and  $L_{oI}^{1/2} L_{c(J_2-J_1)} L_{oI}^{1/2}$  are both symmetric,

according to Wilkinson (1965),  $\lambda(L_{oI}L_{cJ_2}) \geq \lambda(L_{oI}^{1/2}L_{cJ_1}L_{oI}^{1/2}) = \lambda(L_{oI}L_{cJ_1})$ . Similarly, it can be proved that  $\lambda(L_{oI_2}L_{cJ}) \geq \lambda(L_{oI_1}L_{cJ})$ , if  $I_1 \subseteq I_2$ . Hence  $\underline{\sigma}^{I_2J_2} \geq \underline{\sigma}^{I_2J_1} \geq \underline{\sigma}^{I_1J_1}$ .  $\square$

### 3.2 I/O Screening Procedure

In spite of these advantages, BAB method with the HSV does have a demerit, because only part of effect of RHP Zeros is identified with this method. If an RHP zero is close to an RHP pole then the HSV will be very small indicating the system to be either uncontrollable or unobservable. However, RHP zeros at other locations have little effect on the Hankel SV. Such RHP zeros do cause difficulty in stabilization of system. According to the principle of Root Locus, closed-loop poles start from open-loop poles and finish at open-loop zeros as the closed-loop gain varying. Therefore, RHP zeros will normally cause an unstable plant not possible to be stabilized by proportional-only controllers because extra poles and zeros from the controller are required to shape the Root Locus. Thus, RHP zeros should be avoided in control structure selection, if possible. However, RHP zeros cannot be directly handled by Branch and Bound method. Nevertheless, we can use Branch and Bound method to secure an overall ranking on the Hankel SV and then examine the existence of RHP zeros in those ranked structures starting from top towards bottom (considering the fact that the “mathematically” best structure is at the top) until the most top structure without any RHP zeros is found. Thus the following is the step by step procedure of control structure screening with BAB method using the minimum HSV as controllability criterion:

- Step 1:** Input and output scaling is essential for reliable analysis. Here inputs and outputs should be scaled to their possible range, *e.g.* inputs could be scaled to their nearest bound (Cao et al., 1996) whilst output might be scaled to their noise level (Skogestad and Postlethwaite, 1996).
- Step 2:** Shift poles to convert pure integrator(s) to small RHP pole(s).
- Step 3:** All unstable poles are extracted out of the process through projection method.
- Step 4:** BAB method is applied to the anti-stable sub-system to screen the best possible subsets of I/O base on the minimum HSV.
- Step 5:** The top ranked subsets are examined for possible existence of RHP zeros.
- Step 6:** A decision is reached about the best I/O combination.

## 4 APPLICATION OF BRANCH AND BOUND ALGORITHM

The Branch and Bound technique is applied to a highly-integrated chemical plant, *viz.* Tennessee-Eastman (TE) process. This process has been chosen with an intention of demonstrating usage of controllability measure discussed in the previous section, *i.e.*, the minimum *Hankel singular values*.

### 4.1 Tennessee-Eastman process

The Tennessee-Eastman test-bed problem (Downs and Vogel, 1993) involves the control of five unit operations: an exothermic two-phase reactor, a water-cooled condenser, a centrifugal compressor, a flash drum and a product stripper. The simulated plant has 41 process variables and 12 manipulated variables which are modelled with 50 state variables. The twelve manipulated variables are the four feed rates, the purge rate, the agitation rate, steam rate, condenser coolant rate, reactor coolant rate, compressor recycle, flash drum discharge rate and the stripper production rate. Out of 41 process variables there are 22 controllable outputs including level, pressure, temperature, flow and 19 composition indicators. Complete lists of input and output variables are shown in Tables 1 and 2 respectively.

Details of the process are available elsewhere (Downs and Vogel, 1993). A model of the process is generated by Ricker (2002) in Simulink software. Open loop simulation of the model indicates that the process is unstable in nature. The objective is to select the inputs and outputs in order to stabilize the complete system.

### 4.2 Input-output selection

The entire analysis has been carried out through MATLAB software. The step-by-step procedure, described in sec 3.2, is followed while doing the analysis. Scaling factors for each I/O subset-size are obtained from the model itself. Inputs are already scaled in the model within 0 to 100% range. However, an extra scaling factor, *viz.* the maximum possible deviation from its steady state value for each input variable, is incorporated. More specifically, scaling =  $\min(x, 100 - x)$ , where  $x$  is the base case value, as shown in Table 1. Whereas, the output variables are scaled using noise level found from the model, as shown in Tables 2.

The scaled process is decomposed into stable and anti-stable subsystems using STABPROJ subroutine of the Robust Control Toolbox (Safanov et al., 1987).

The anti-stable subsystem has two pairs of positive complex poles,  $3.0648 \pm 5.0837i$  and  $0.024973 \pm 0.15521i$  and two pure integrators (numerical value is of the order of  $10^{-9}$ ).

Using the Branch and Bound method, 10 sets of best I/O combinations are computed as in Figure 4. The figure shows the minimum HSV against the subset-size. Although the calculation is performed for subset-sizes from 1 through 34, HSV does not increase notably over subset-size of 20. It can be noted from Figure 4 that the minimum HSV for up to a subset-size of 4 is nearly zero. By mathematical derivation, it can be proved that it is actually zero. The significance of this result is a conclusion that, it is impossible to stabilize the plant with any combination of a total number of inputs and outputs less than 4. That means a minimum total of 5 inputs and outputs are necessary to achieve that stabilization. However, in terms of decentralized control a  $3 \times 2$  or  $2 \times 3$  system has no advantage at all compared to a  $3 \times 3$  system as far as the complexity of controller design is concerned. Hence, the minimum HSVs for subset-size 6 are ranked for the best 10 sets in Table 3.

Based on the discussions about RHP zeros in section 3.1, it is observed (Table 3) that best control structure without the presence of RHP zeros is the 4th structure in Table 3, i.e., with liquid levels in stripper and the separator (outputs 12 and 15), reactor temperature (output 9), separator and stripper outflows (inputs 7 and 8) and reactor cooling water flow (input 10). The observation is identical with the results obtained by (Cao and Saha, 2003) and similar to those suggested by Havre and Skogestad (1998) and McAvoy (1998). However, the guarantee for best I/O combination is obtained in this work through BAB technique. Moreover, this guarantee is obtained with only 16160 computations instead of possible 17175670785 in an exhaustive search. The efficiency of the improved Branch and Bound algorithm is compared with a depth-first search based algorithm (Cao et al., 1998) in Figure 5. The main advantage of the new algorithm is for a variable subset-size selection problem, where the total number of evaluations is significantly reduced due to the reuse of the pool of subsets. However, even for a fixed subset-size problems, the new algorithm is still more efficient than depth-first search approaches due to the reason explained in Remark 14. This is also shown in the comparison figure, where the total number of evaluations of the pool-based algorithm using independent pools (every time performing a fixed subset-size selection, a new pool is initialized) is still less than depth-first search algorithm.

The closed loop simulation results are shown in Figure 6. It is observed that stabilization of the process is achieved with three unit-gain proportional controllers. Simulation results are obtained without any disturbance but with measurement noise.

## 5 CONCLUSION

An improved Branch and Bound algorithm based on the best-first search approach is proposed. The new algorithm is more efficient than other depth-first search based algorithms. Application of the improved Branch and Bound method is demonstrated in this work for I/O screening using a controllability criterion, *viz.*, the minimum Hankel singular value. The results, obtained through simulation, show that the proposed method inherits all the advantages of the original branch and bound method. Nevertheless, it demonstrates a few more merits than the original in terms of efficiency and subset-size screening. A screened set of I/O combinations is more useful than a single set of best combination because the screened set provides more options for control structure selection if the mathematically best combination leads to RHP zeros. A case study with Tennessee-Eastman test bed problem shows that it is possible to stabilize the open-loop unstable plant with minimum control effort, by using three decentralized P-controllers with unit gains. It has also been proved that it is impossible to stabilize the plant with total number of inputs and outputs less than four.

## 6 ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support provided by the EPSRC, UK (Grant number GR/R 57324) for this research work.

## References

- Braatz, R., Lee, J., Morari, M., 1996. Screening plant designs and control structures for uncertain systems. *Comput. Chem. Eng.* 20 (4), 463–468.
- Cao, Y., Biss, D., Perkins, J. D., 1996. Comparison of two techniques for assessing input-output controllability in the presence of manipulated variable constraints. *Computers Chem. Engng.* 20 (4), 337–346.
- Cao, Y., Rossiter, D., Owens, D. H., 1998. Globally optimal control structure selection using branch and bound method. In: *Preprints of DYCOPS'5*. Corfu, Greece, pp. 183–188.
- Cao, Y., Saha, P., Jun 2003. Control structure selection for unstable processes using Hankel singular value. In: *13th European Symposium on Computer Aided Process Engineering (ESCAPE-13)*. Elsevier Science, Lappeenranta, Finland, pp. 383–388.
- Chen, X., 2003. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters* 24, 1925–1933.

- Downs, J. J., Vogel, E. F., 1993. A plant-wide industrial process control problem. *Computers Chem. Engng.* 17 (3), 245–255.
- Glover, K., 1986. Robust stabilization of linear-multivariable systems - relations to approximation. *International Journal of Control* 43, 741–766.
- Gutierrez, R., Sahinidis, N., 1996. A branch-and-bound approach for machine screening in just-in-time manufacturing systems. *International Journal of Production Research* 34 (3), 797–818.
- Havre, K., Skogestad, K., 1998. Selection of variables for regulatory control using pole vectors. In: *Preprints of DYCOPS'5*. Corfu, Greece, pp. 614–619.
- Havre, K., Skogestad, S., 1996. Input-output selection and partial control. In: *Proceedings of IFAC'96 13th World Congress*. Vol. M. San Francisco, CA, USA, pp. 181–186.
- Havre, K., Skogestad, S., 2003. Screening of variables for stabilizing control using pole vectors. *IEEE Transaction on Automatic Control* 48 (8), 1393–1398.
- Heath, J., Kookos, I., Perkins, J., 2000. Process control structure selection based on economics. *AICHE J.* 46 (10), 1998–2016.
- Kookos, I., Perkins, J., 2001. An algorithm for simultaneous process design and control. *Ind. Eng. Chem. Res.* 40, 4079–4088.
- Luyben, M., Floudas, C., 1994a. Analyzing the interaction of design and control .1. a multiobjective framework and application to binary distillation synthesis. *Comput. Chem. Eng.* 18 (10), 933–969.
- Luyben, M., Floudas, C., 1994b. Analyzing the interaction of design and control .2. reactor separator recycle system. *Comput. Chem. Eng.* 18 (10), 971–994.
- McAvoy, T., 1998. A methodology for screening level control structures in plantwide control systems. *Computers and Chemical Engineering* 22 (11), 1543–1552.
- Narendra, P. M., Fukunaga, K., 1977. A branch and bound algorithm for feature subset selection. *IEEE Transaction on Computers* C-26 (9), 917–922.
- Raman, R., Grossmann, I., 1991. Relation between minlp modelling and logical interface for chemical process synthesis. *Computers and Chemical Engineering* 15 (2), 73–84.
- Ricker, N., 2002. New simulink models of two decentralized control strategies. *Tennessee Eastman Challenge Archive*, <http://depts.washington.edu/control/LARRY/TE/download.html>.
- Ridout, M. S., 1988. An improved branch and bound algorithm for feature subset selection. *Applied Statistics* 37, 139–147.
- Roberts, S. J., 1984. A branch and bound algorithm for determining the optimal feature subset of given size. *Applied Statistics* 33, 236–241.
- Ryoo, H., Sahinidis, N., 1996. A branch-and-bound approach to global optimization. *Journal of Global Optimization* 8 (2), 107–139.
- Safanov, M., Jonckheere, E., Verma, M., Limebeer, D., 1987. Synthesis of positive real multivariable feedback systems. *International Journal of Control*

- 45 (3), 817–842.
- Skogestad, S., Postlethwaite, I., 1996. *Multivariable Feedback Control. Analysis and Design*. John Wiley & Sons, Chichester.
- Skrifvars, H., Harjunkoski, I., Westerlund, T., Kravanja, Z., Pörn, R., 1996. Comparison of different minlp methods applied on certain chemical engineering problems. *Computers and Chemical Engineering* 20, S333–S338.
- Wilkinson, J., 1965. *The Algebraic Eigenvalue Problem*. Claredon Press, Oxford, England.
- Yu, B., Yuan, B., 1992. A dynamic selection algorithm for globally optimal subsets. *Engineering Applications of Artificial Intelligence* 5 (5), 457–462.
- Yu, B., Yuan, B., 1993. A more efficient branch and bound algorithm for feature selection. *Pattern Recognition* 26 (6), 883–889.



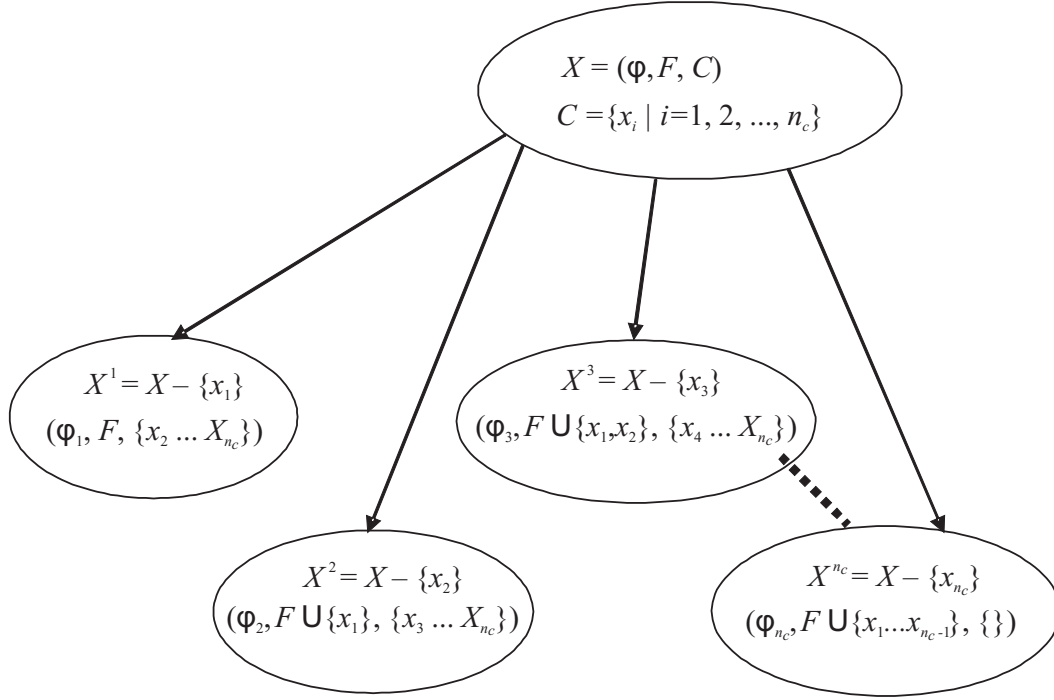


Fig. 3. Fixed-set and candidate-set propagation in subsets

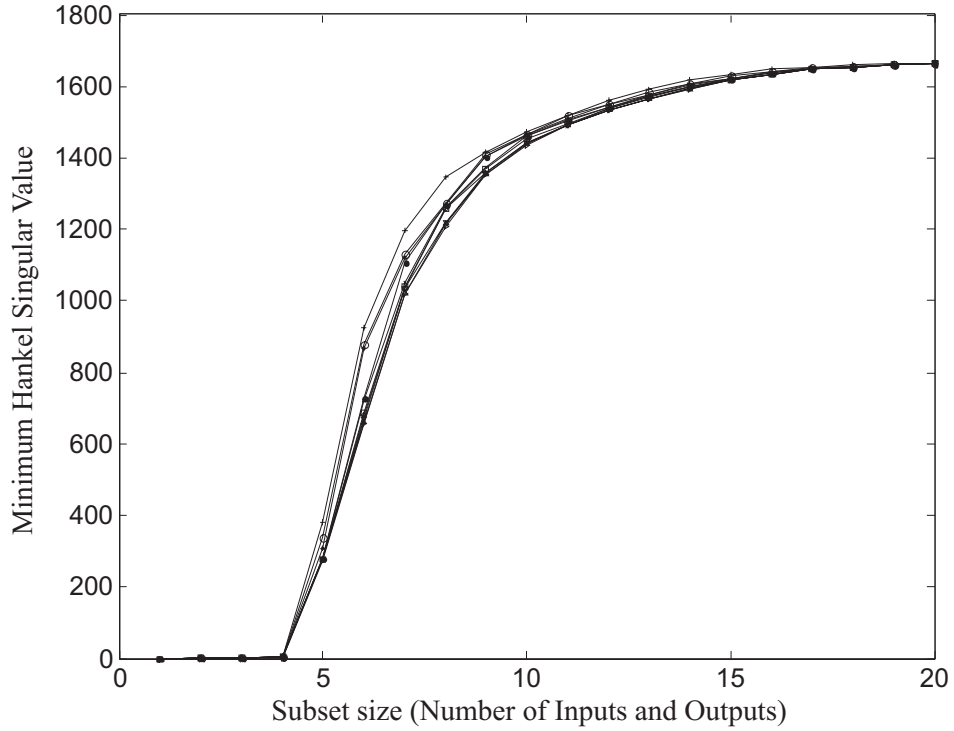


Fig. 4. HSV vs subset-size for 10 best cases of I/O combination

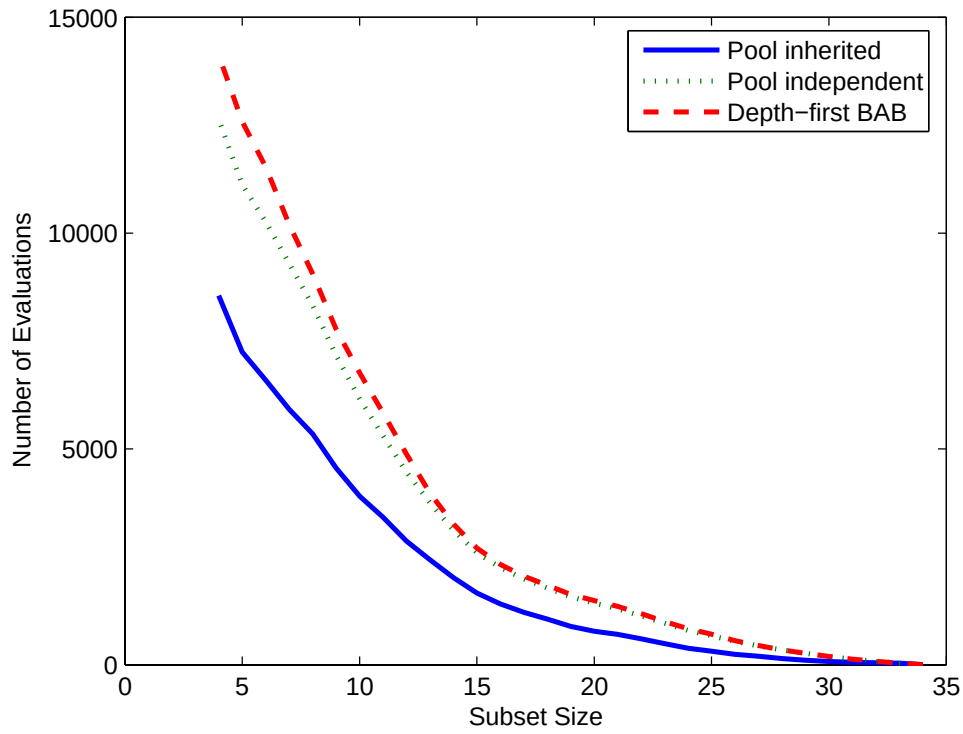


Fig. 5. Total number of evaluations executed by different algorithms

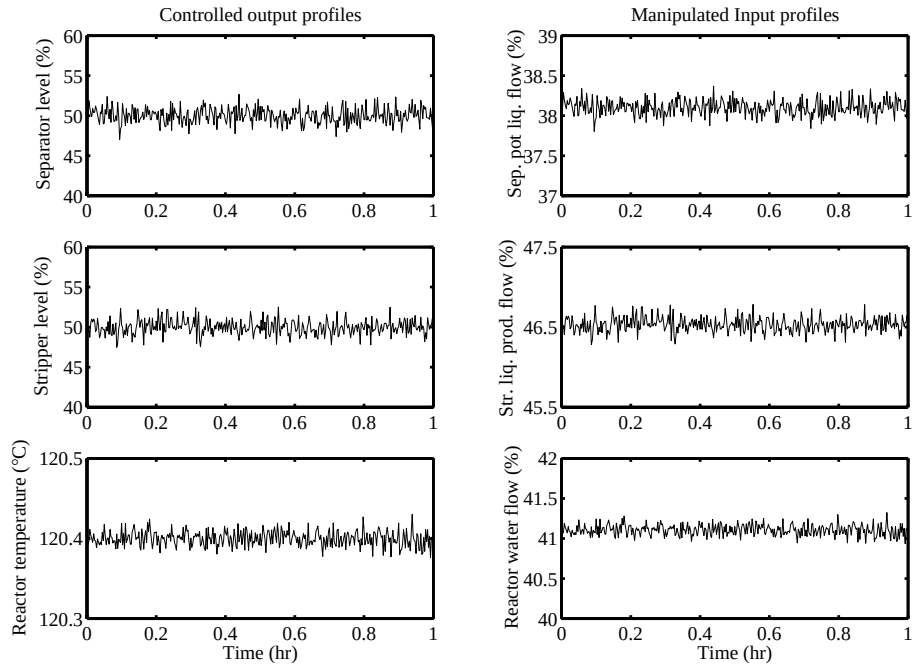


Fig. 6. Closed-loop response of TE plant using stabilizing P-only controllers

Table 1  
Tennessee-Eastman process manipulated variables

| Variable<br>description      | Variable<br>number | Base case<br>value (%) | Scaling<br>(%) |
|------------------------------|--------------------|------------------------|----------------|
| D feed flow                  | XMV 1              | 63.053                 | 36.947         |
| E feed flow                  | XMV 2              | 53.980                 | 46.020         |
| A feed flow                  | XMV 3              | 24.644                 | 24.644         |
| A and C feed flow            | XMV 4              | 61.302                 | 38.698         |
| Compressor recycle valve     | XMV 5              | 22.210                 | 22.210         |
| Purge valve                  | XMV 6              | 40.064                 | 40.064         |
| Separator pot liquid flow    | XMV 7              | 38.100                 | 38.100         |
| Stripper liquid product flow | XMV 8              | 46.534                 | 46.534         |
| Stripper steam valve         | XMV 9              | 47.446                 | 47.446         |
| Reactor cooling water flow   | XMV 10             | 41.106                 | 41.106         |
| Condenser cooling water flow | XMV 11             | 18.114                 | 18.114         |
| Agitator speed               | XMV 12             | 50.000                 | 50.000         |

Table 2  
Tennessee-Eastman process output variables

| Variable<br>description             | Variable<br>number | Base<br>case<br>value | Scaling<br>(noise<br>level) | Unit              |
|-------------------------------------|--------------------|-----------------------|-----------------------------|-------------------|
| A feed                              | XMEAS 1            | 0.2502                | 0.0012                      | Kscm/h            |
| D feed                              | XMEAS 2            | 3664                  | 18                          | Kg/h              |
| E feed                              | XMEAS 3            | 4509.3                | 22                          | Kg/h              |
| A and C feed                        | XMEAS 4            | 9.3477                | 0.05                        | Kscm/h            |
| Recycle flow                        | XMEAS 5            | 26.902                | 0.2                         | Kscm/h            |
| Reactor feed rate                   | XMEAS 6            | 42.339                | 0.21                        | Kscm/h            |
| Reactor pressure                    | XMEAS 7            | 2705                  | 0.3                         | KPa gauge         |
| Reactor level                       | XMEAS 8            | 75                    | 0.5                         | %                 |
| Reactor temperature                 | XMEAS 9            | 120.4                 | 0.01                        | °C                |
| Purge rate                          | XMEAS 10           | 0.33712               | 0.0017                      | Kscm/h            |
| Product separator temperature       | XMEAS 11           | 80.109                | 0.01                        | °C                |
| Product separator level             | XMEAS 12           | 50                    | 1                           | %                 |
| Product separator pressure          | XMEAS 13           | 2633.7                | 0.3                         | KPa gauge         |
| Product separator underflow         | XMEAS 14           | 25.16                 | 0.125                       | M <sup>3</sup> /h |
| Stripper level                      | XMEAS 15           | 50                    | 1                           | %                 |
| Stripper pressure                   | XMEAS 16           | 3102.2                | 0.3                         | KPa gauge         |
| Stripper underflow                  | XMEAS 17           | 22.949                | 0.115                       | M <sup>3</sup> /h |
| Stripper temperature                | XMEAS 18           | 65.731                | 0.01                        | °C                |
| Stripper steam flow                 | XMEAS 19           | 230.31                | 1.15                        | Kg/h              |
| Compressor work                     | XMEAS 20           | 341.43                | 0.2                         | KW                |
| Reactor cooling water outlet temp   | XMEAS 21           | 94.599                | 0.01                        | °C                |
| Separator cooling water outlet temp | XMEAS 22           | 77.297                | 0.01                        | °C                |

Table 3  
Controllability indices of 10 best I/O set of size 6

| Set | Input No. | Output No. | $\underline{\sigma}$ | RHP zeros                        |
|-----|-----------|------------|----------------------|----------------------------------|
| 1   | 1, 8, 10  | 9, 12, 15  | 926.65               | 0.152 and 109.522                |
| 2   | 4, 8, 10  | 9, 12, 15  | 878.57               | 0.1103 and $5.6057 \pm 10.5721i$ |
| 3   | 2, 8, 10  | 9, 12, 15  | 868.89               | 0.1571 and 29.2248               |
| 4   | 7, 8, 10  | 9, 12, 15  | 721.47               | None                             |
| 5   | 4, 8, 10  | 11, 12, 15 | 688.34               |                                  |
| 6   | 2, 8, 10  | 11, 12, 15 | 681.30               |                                  |
| 7   | 7, 8, 10  | 11, 12, 15 | 673.45               |                                  |
| 8   | 2, 4, 10  | 11, 12, 15 | 660.25               |                                  |
| 9   | 2, 4, 10  | 9, 12, 15  | 658.35               |                                  |
| 10  | 1, 8, 10  | 12, 15, 21 | 610.55               |                                  |



**DSPACE**

<https://dspace.org/>

## **Improved branch and bound method for control structure screening**

**Cao, Yi; Saha, Prabirkumar**

**2005-03-01T00:00:00Z**

Elsevier Science B.V., Amsterdam.

<http://dx.doi.org/10.1016/j.ces.2004.10.025>

Yi Cao and Prabirkumar Saha, Improved branch and bound method for control structure screening, Chemical Engineering Science, Volume 60, Issue 6, March 2005, Pages 1555-1564.