Weak Interactions Based System Partitioning Using Integer Linear Programming

Romain Guicherd * **Paul A. Trodden** * **Andrew R. Mills** *
Visakan Kadirkamanathan *

* *Rolls-Royce University Technology Centre,*
Department of Automatic Control & Systems Engineering,
University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK
(e-mail:rguicherd1@sheffield.ac.uk; p.trodden@sheffield.ac.uk;
a.r.mills@sheffield.ac.uk; visakan@sheffield.ac.uk).

Abstract: The partitioning of a system model will condition the structure of the controller as well as its design. In order to partition a system model, one has to know what states and inputs to group together to define subsystem models. For a given partitioning, the total magnitude of the interactions between subsystem models is evaluated. Therefore, the partitioning problem seeking for weak interactions can be posed as a minimization problem. Initially, the problem is formulated as a non-linear integer minimization that is then relaxed into a linear integer programming problem. It is shown within this paper that cuts can be applied to the initial search space in order to find the least interacting partitioning; only composed of controllable subsystems. Two examples are given to demonstrate the methodology.

Keywords: Decoupling problems, linear multivariable systems, decentralized control, integer programming.

1. INTRODUCTION

Systems models are widely used in control design especially with the development of techniques such as model predictive control (Rawlings and Mayne (2009)) (Maciejowski (2002)). Systems are growing in size and complexity and they are in most cases composed of interacting subsystems (Scattolini (2009)). For these large scale systems, the design of a centralized controller can be prohibitive due to the heavy computational resources required (Mayne (2014)). Also, if the system is geographically spread out, communication delays between the centralized controller and the actuators and sensors arise. One way to solve this problem is to see the system as a concatenation of subsystems and to design local controllers for each subsystem. In a top-down approach the full model of the multivariable system is partitioned into subsystem models so that the decentralized controller can be designed. Decentralized control has been studied for decades and design procedures have been established (Šiljak (1991)) (Bakule (2008)). However, the system model partitioning problem has been overlooked, often because the system is already composed of physical subsystems. Every subsystem model is defined by a set of states and inputs. The weak interaction partitioning problem consist of defining these sets in order to minimize the coupling between subsystem models. For instance, strongly coupled subsystem models can emerge from the main system model, particularly within chemical plants (Stewart et al. (2010)) or heating systems (Moroşan et al. (2010)). The ideal partitioning of a system model would yield completely decoupled subsystem models.

Defining the subsystems of a plant has been done in different ways in the past. One of the first methods employed to couple inputs and outputs was the relative gain array (Bristol (1966)). This method is used to find the best pairing at steady state between inputs and outputs and hence to choose the most relevant input to control a given output in a multi-input multi-output system. It can be seen as a response to the industrial need to control a multi-variable process as a combination of single variable processes. The relative gain array has been extended to the block relative gain, allowing for suitable pairing for block decentralized control (Manousiouthakis et al. (1986)) (Kariwala et al. (2003)). The extension of the relative gain array allows the design of multi-variable controllers in a decentralized way. However it only links inputs and outputs together and does not provide a partitioning of the plant model. A technique similar to the relative gain array, is the Nyquist array method, allowing the design of single-input single-output controllers after rendering the model diagonaly dominant (Leininger (1979)) (Chen and Seborg (2003)). System partitioning can be performed by seeking the least interacting groups. Another technique used for system decomposition and integration is the design structure matrix also known as the dependency structure matrix or interaction matrix (Browning (2001)). This technique indicates the link between the elements it represents, moreover the links are directed. Elements along a row indicate that a contribution is provided to other elements whereas elements along a column indicates a dependency from other parts of the system. The attribution of weights within the interaction matrix is used in order to perform clustering and achieve system decomposition. Other work on decentralized control combined the controller design along with the controller topology, these two aspects are combined in the optimization function yielding a trade-off between the need for feedback links and the loss of performance compared to the centralized controller (Schuler et al. (2014)). Finally, other works have studied the actuator partitioning problem (Jamoom et al. (1998)) (Motee and Sayyar-Rodsari (2003)). To the best of the authors' knowledge the problem addressing state space model partitioning has not been studied. Therefore this partitioning approach is a standalone work, making any comparison difficult.

In this paper, we propose an integer programming based approached to the problem of partitioning a system model into a set of non-overlapping but coupled subsystem models. The objective is to reduce the magnitude of the interactions between the subsystem models. Finally, cuts are added to rule out non-controllable partitionings in order for the algorithm to yield only controllable subsystem models.

The paper is organised as follows. Section 2 states the problem and section 3 introduces the required notations. Section 4 demonstrates how the problem can be relaxed into a linear integer programming problem. In section 5 the controllability cut principle is presented allowing to obtain controllable subsystems, section 6 explains the linear partitioning algorithm. In order to illustrate the algorithm section 7 includes some examples, finally section 8 concludes the paper.

Notation: For $(a,b) \in \mathbb{N}^2$ such that $a < b$, the set $[\![a;b]\!]$ defines the set containing the integers from a to b included. The operator $|.|$ is used to denote the magnitude of a complex number when applied to a complex number. When applied to a matrix the magnitude operator is applied to all the matrix elements then summed. The operator $\|.\|_2$ defines the Euclidean norm for complexes, vectors and matrices. For a set $\mathbb{N}$, the notation $\mathbb{N}^*$ defines $\mathbb{N}/\{0\}$. The superscript $^\intercal$ represents the transpose of a vector or matrix. A matrix $B \in \mathbb{R}^{n\times m}$ will be noted $(b_{ij})_{(i,j)\in[\![1;n]\!]\times[\![1;m]\!]}$ and $(B_{kl})_{(k,l)\in[\![1;N]\!]\times[\![1;M]\!]}$, respectively for the element and block notations, with N row blocks and M column blocks. For all $n \in \mathbb{N}^*$ and $A \in \mathbb{R}^{n\times n}$, $trace(A)$ denotes the sum of all the diagonal elements of A. For all $(n,m) \in (\mathbb{N}^*)^2$ and $A \in \mathbb{R}^{n\times m}$, $rank(A)$ denotes the dimension of the vector space spanned by the columns of A. For any $(i,j,k) \in (\mathbb{N}^*)^3$ and $(A,B) \in \mathbb{R}^{i\times j} \times \mathbb{R}^{i\times k}$ the notation $[A|B]$ defines the matrix obtained by concatenating A and B horizontally. For any couple of integers $(i,j) \in \mathbb{N}^2$, δ_{ij} denotes the Kronecker delta function.

2. PROBLEM STATEMENT

Given a linear time invariant controllable state space model defined by

$$\dot{x} = Ax + Bu \tag{1}$$

where the matrix A is the state matrix and the matrix B is the input matrix respectively with the appropriate sizes for N states and M inputs, therefore, $x \in \mathbb{R}^N$ and $u \in \mathbb{R}^M$. Partitioning the system model (1) consists of decomposing the inputs as well as the states into groups representing subsystems. For a given number of partitions $P \in [\![2;\min(N,M)]\!]$ and for any subsystem $p \in [\![1;P]\!]$ the model p can be expressed as follows

$$\dot{x}_p = A_{pp}x_p + B_{pp}u_p + \sum_{\substack{j=1\\j\neq p}}^{P} \left\{A_{pj}x_j + B_{pj}u_j\right\} \tag{2}$$

with for all $p \in [\![1;P]\!]$, $x_p \in \mathbb{R}^{N_p}$ and $u_p \in \mathbb{R}^{M_p}$ such that

$$\sum_{p=1}^{P} N_p = N \tag{3a}$$

$$\sum_{p=1}^{P} M_p = M \tag{3b}$$

The weak interaction partitioning problem consists of minimizing the magnitude of the right-hand side sum in (2) for the subsystems while keeping each of them controllable. A non-overlapping condition for the states and the inputs is imposed by (3). The next section presents the decision variables, the constraints as well as the interaction metric necessary to formulate the weak interactions optimization problem.

3. WEAK INTERACTIONS PROBLEM FORMULATION

3.1 Decision variables

A decision variable is associated to the couples formed by a group p and a state i as well as a group p and an input k. All the decision variables are binary variables. They are organised in two grouping matrices, the state grouping matrix $\alpha \in [\![0;1]\!]^{P\times N}$ and the input grouping matrix $\beta \in [\![0;1]\!]^{P\times M}$. Therefore, the rows of α and β represent the P groups and the columns respectively represent the N states and the M inputs. For example with $P = 3$ and $N = 5$ a non-overlapping state grouping matrix could be

$$\alpha = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

In this example, the first three states belong to the second group, the fourth state composes the first group and the last state is in the third group. Hence, a specific partitioning is represented by a pair of state and input non-overlapping grouping matrices. The columns of the grouping matrices are composed of zeros and a single one. The one is positioned in the row representing the group where the state or input belongs respectively for a state and an input non-overlapping grouping matrix. The next subsection presents the linear constraints restricting the decision variables α and β in the integer optimization problem.

3.2 Partitioning constraints

The formulation of constraints on the decision variables is necessary in order for the algorithm to return a solution complying with the rules defining non-overlapping subsystem models.

1. Each state group contains at least a state, hence, no state group can be empty and the partitioning has the correct number of state groups
$$\forall p \in [\![1;P]\!],\ \sum_{i=1}^{N} \alpha_{pi} \geq 1 \tag{4a}$$
2. Each input group contains at least an input, hence, no input group can be empty and the partitioning has the correct number of input groups
$$\forall p \in [\![1;P]\!],\ \sum_{i=1}^{M} \beta_{pi} \geq 1 \tag{4b}$$
3. A state can be in only one state group, therefore, the multiple use of a state is prevented and the non-overlapping requirement is respected
$$\forall i \in [\![1;N]\!],\ \sum_{p=1}^{P} \alpha_{pi} \leq 1 \tag{4c}$$
4. An input can be in only one input group, therefore, the multiple use of an input is prevented and the non-overlapping requirement is respected
$$\forall i \in [\![1;M]\!],\ \sum_{p=1}^{P} \beta_{pi} \leq 1 \tag{4d}$$
5. Each state must belong to a state group, consequently, no state is left out of the optimization problem
$$\forall i \in [\![1;N]\!],\ \sum_{p=1}^{P} \alpha_{pi} \geq 1 \tag{4e}$$
6. Each input must belong to a input group, consequently, no input is left out of the optimization problem
$$\forall i \in [\![1;M]\!],\ \sum_{p=1}^{P} \beta_{pi} \geq 1 \tag{4f}$$

The constraints are expressed for the two grouping matrices, however only three different sets of constraints concern each type of grouping matrix. Because α and β are arrays of binary variables a natural implicit constraint links N, M and P.

$$1 < P \leq \min(N, M) \tag{5}$$

Subsystem interactions can come from the state matrices or the input matrices, the next subsection presents how these interactions can be formulated firstly using the block matrix form and secondly using the state space model elements.

3.3 Objective: minimizing subsystem interactions

The first part of the interactions comes from the state matrices. The subsystem model (2) presents the couplings with the other subsystems in the form of a sum, this sum can be split into the state interactions and the input interactions. For a given number of partitions $P \in [\![2;\min(N,M)]\!]$ and for any subsystem $p \in [\![1;P]\!]$ the state interactions can be expressed by

$$J_p^{state} = \sum_{\substack{j=1 \\ j\neq p}}^{P} |A_{pj}| \tag{6}$$

The expression written in block matrix form can also be represented using the state matrix elements as well as the state grouping matrix elements as follows

$$J_p^{state} = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{pi}|a_{ij}|(1-\alpha_{pj}) \tag{7}$$

The elements from the state grouping matrix are used here as boolean tests to take into account only the interactions acting on the subsystem p and coming from the other subsystem states. A similar reasoning is applied to quantify the interactions coming from the input matrices

$$J_p^{input} = \sum_{\substack{j=1 \\ j\neq p}}^{P} |B_{pj}| \tag{8}$$

In a similar fashion, (8) can also be represented using the input matrix elements as well as the elements of the state grouping matrix combined with the elements of the input grouping matrix, such that

$$J_p^{input} = \sum_{i=1}^{N}\sum_{k=1}^{M} \alpha_{pi}|b_{ik}|(1-\beta_{pk}) \tag{9}$$

Likewise, the elements from the state grouping matrix combined with the elements of the input grouping matrix are used as boolean tests to take into account only the interactions acting on the subsystem p and coming from the other subsystem inputs. After having defined the two types of interactions, the full interaction metric can be calculated. Consequently, the last step is to pose the weak interactions optimization problem like it is presented within the next subsection.

3.4 Weak interactions optimization problem

The overall formulation of the weak interactions optimization problem is obtained by summing the interactions coming from the states and the inputs over the P subsystems

$$J^{interaction} = \sum_{p=1}^{P} \left\{ J_p^{state} + J_p^{input} \right\} \tag{10}$$

Hence, the integer optimization problem can be formulated using the same notation employed in (7) and (9) and is expressed as follows

$$\begin{aligned}\min_{\alpha,\beta} \quad & \sum_{p=1}^{P}\sum_{i=1}^{N}\Big\{\alpha_{pi}\Big[\sum_{j=1}^{N}\{|a_{ij}|(1-\alpha_{pj})\} \\ & + \sum_{k=1}^{M}\{|b_{ik}|(1-\beta_{pk})\}\Big]\Big\} \\ \text{s. t.} \quad & (4)\end{aligned} \tag{11}$$

As it was demonstrated previously within this section the partitioning problem can be expressed as an integer optimization problem. However the cost function representing the interaction metric is non-linear, therefore the problem can be intractable. As it is presented within the next section a linear relaxation of the optimization problem (11) is made possible throughout the use of auxiliary variables.

4. LINEAR RELAXATION OF THE WEAK INTERACTIONS PROBLEM

The weak interactions optimization problem formulated previously (11) can be turned into a linear optimization problem, this is made possible due to the introduction of auxiliary variables. Replacing a product of binary variables by an auxiliary variable is a well known technique that requires the use of new linear constraints (Williams (2013)) (Bemporad and Morari (1999)) (Cavalier et al. (1990)). Two auxiliary binary variables are created along with their linear constraints. γ is the binary variable used to take into account the interactions coming from the state matrix. As it is presented in Table I and in (12), γ is linked to α throughout four constraints. Indeed, four inequalities are necessary because of the four possible outcomes for the binary product $\alpha_{pi}(1-\alpha_{pj})$ in (11).

From top to bottom within Table I, the four different cases are, first when no states belong to the group p then no interaction has to be accounted for. If the state i is not in the group p but the state j is, then no interaction is accounted for as this will be taken into account in the symmetrical case. If the state i belongs to the group p and the state j does not then an interaction is accounted for. The last case possible is when the two states i and j both belong to the group p, in this last scenario no interaction subsists as they are both in the same group.

Table 1. Auxiliary variable γ

Primary			**Auxiliary**
α_{pi}	α_{pj}	$\alpha_{pi}(1-\alpha_{pj})$	γ_{pij}
0	0	0	0
0	1	0	0
1	0	1	1
1	1	0	0

The linear constraints for the auxiliary variable γ are the following

$\forall(p;i;j) \in [\![1;P]\!] \times [\![1;N]\!] \times [\![1;N]\!]$,

$$\gamma_{pij} \leq \alpha_{pi} + \alpha_{pj} \tag{12a}$$
$$\gamma_{pij} \leq 1 + \alpha_{pi} - \alpha_{pj} \tag{12b}$$
$$\gamma_{pij} \geq \alpha_{pi} - \alpha_{pj} \tag{12c}$$
$$\gamma_{pij} \leq 2 - \alpha_{pi} - \alpha_{pj} \tag{12d}$$

In a similar way, δ is the auxiliary binary variable taking into account the interactions coming form the input matrix. This time the constraints are formed using α as well as β because the input interactions are also state dependant. The same reasoning applies to formulate the four inequalities arising from the binary product $\alpha_{pi}(1-\beta_{pk})$ in (11).

Table 2. Auxiliary variable δ

Primary			**Auxiliary**
α_{pi}	β_{pk}	$\alpha_{pi}(1-\beta_{pk})$	δ_{pik}
0	0	0	0
0	1	0	0
1	0	1	1
1	1	0	0

The four linear constraints associated to δ are represented in (13).

$\forall(p;i;k) \in [\![1;P]\!] \times [\![1;N]\!] \times [\![1;M]\!]$,

$$\delta_{pik} \leq \alpha_{pi} + \beta_{pk} \tag{13a}$$
$$\delta_{pik} \leq 1 + \alpha_{pi} - \beta_{pk} \tag{13b}$$
$$\delta_{pik} \geq \alpha_{pi} - \beta_{pk} \tag{13c}$$
$$\delta_{pik} \leq 2 - \alpha_{pi} - \beta_{pk} \tag{13d}$$

Both auxiliary variables have three indexes and can be represented by cubic arrays of binary variables, their respective sizes are $P \times N \times N$ for the state auxiliary variable γ and $P \times N \times M$ for the input auxiliary variable δ. In addition to the constraints presented in (12) and (13), both auxiliary variables have to be composed only of binary variables.

The optimization problem presented in (11) is reformulated into the linear integer optimization problem presented in (14), obtained by replacing the binary products by the auxiliary variables along with their constraints.

$$\begin{aligned}\min_{\alpha,\beta} \quad & \sum_{p=1}^{P}\sum_{i=1}^{N}\Big\{\sum_{j=1}^{N}\{\gamma_{pij}|a_{ij}|\} + \sum_{k=1}^{M}\{\delta_{pik}|b_{ik}|\}\Big\} \\ \text{s. t.} \quad & (4),(12),(13)\end{aligned} \tag{14}$$

As it was presented within this section, the use of two auxiliary variables enables the linearisation of the optimization problem. The minimization problem presented in (14) because of the use of primary and auxiliary binary variables allows to trade non-linearities for an increase in variable size. Nonetheless the complexity of the optimization problem can be reduced by exploiting the structure of the plant model and by creating only the auxiliary variables where the state space model elements are not equal to zero. For instance, if $a_{ij} = 0$ there is no need to create $(\gamma_{pij})_{p\in[\![1;P]\!]}$, similarly if $b_{ik} = 0$ with $(\delta_{pik})_{p\in[\![1;P]\!]}$. The next section presents the notion of controllability cuts reducing the search space in order to obtain only controllable subsystems.

5. CONTROLLABILITY CUT

Running the previous optimization problem will yield P subsystems presenting the least amount of interactions, unfortunately no information is given concerning their controllability. The state space model of any subsystem p, represented without the couplings coming from the other subsystems can be rewritten from (2) as follows

$$\forall p \in [\![1; P]\!],\, \dot{x}_p = A_{pp}x_p + B_{pp}u_p \tag{15}$$

The controllability of any given subsystem model p yielded by the optimization problem can be checked by verifying that the controllability matrix C_p defined in (16) has full row rank.

$$\forall p \in [\![1; P]\!],\, C_p = [B_{pp}|A_{pp}B_{pp}|\dots|A_{pp}^{N_p-1}B_{pp}] \tag{16}$$

Therefore, at the end of the optimization process, a controllability test is performed for each subsystem model, testing that the set of equalities given in (17) holds.

$$\forall p \in [\![1; P]\!],\, rank(C_p) = N_p \tag{17}$$

As one can see the controllability matrices C_p as well as the integers N_p representing the number of states in subsystem p are results of the optimization and are not known a priori. Therefore implementing constraints within the linear integer optimization problem in order to restrain the solutions to the set of controllable subsystems is a tremendously difficult task. However, applying controllability cuts to the search space recursively and a posteriori is possible.

Every time a non-controllable partitioning is achieved new linear constraints are added to the existing ones in order to reduce the search space by cutting the non-controllable partitionings out with an affine hyperplane. The principle of cutting solutions out of the search space is similar to the Gomory cuts (Gomory (1958)) where cuts are used to discard solutions that are not integer. Controllability cuts are applied from the root node and are valid for the entire search tree, hence cuts lifting methods are not necessary in this case (Balas et al. (1996)).

Grouping matrices can be seen as a concatenation of basis vectors, such that

$$\alpha = [e_{i_1}|e_{i_2}|\dots|e_{i_k}|\dots|e_{i_N}]_{i_k \in [\![1;P]\!]} \tag{18a}$$
$$\beta = [e_{i_1}|e_{i_2}|\dots|e_{i_k}|\dots|e_{i_M}]_{i_k \in [\![1;P]\!]} \tag{18b}$$

with $(e_{i_k})_{i_k \in [\![1;P]\!]}$ the canonical orthonormal basis of $\mathbb{R}^P$. Subsequently the square of their 2-norm is equal to respectively N and M as it is calculated in (19) (20).

$$\|\alpha\|_2^2 = trace(\alpha^\intercal\alpha) = \sum_{k=1}^{N} e_{i_k}^\intercal . e_{i_k} = \sum_{k=1}^{N} \delta_{i_k i_k} = N \tag{19}$$

$$\|\beta\|_2^2 = trace(\beta^\intercal\beta) = \sum_{k=1}^{M} e_{i_k}^\intercal . e_{i_k} = \sum_{k=1}^{M} \delta_{i_k i_k} = M \tag{20}$$

The set of non-overlapping grouping matrices of size $P \times N$ and respecting the constraints (4a) (4c) (4e) will be referred to as $\mathbb{G}_{PN}$ with $\mathbb{G}_{PN} \subset [\![0; 1]\!]^{P\times N}$. For a given non-controllable optimal partitioning denoted by α^{nc^*} and β^{nc^*}, and for any couple of non-overlapping grouping matrices α and β, the inequalities (21) and (22) hold.

$$\forall(\alpha^{nc^*}, \alpha) \in \mathbb{G}_{PN}^2,$$
$$trace(\alpha^{nc^*\intercal}\alpha) = \sum_{k=1}^{N} e_{i_{k^{nc^*}}}^\intercal . e_{i_k} = \sum_{k=1}^{N} \delta_{i_{k^{nc^*}} i_k} \leq N \tag{21}$$

$$\forall(\beta^{nc^*}, \beta) \in \mathbb{G}_{PM}^2,$$
$$trace(\beta^{nc^*\intercal}\beta) = \sum_{k=1}^{M} e_{i_{k^{nc^*}}}^\intercal . e_{i_k} = \sum_{k=1}^{M} \delta_{i_{k^{nc^*}} i_k} \leq M \tag{22}$$

In addition, the upper bound is only reached in (21) when $\alpha = \alpha^{nc^*}$ and in (22) when $\beta = \beta^{nc^*}$. Consequently there exists a natural way of constructing affine cutting hyperplanes (23) when a non-controllable optimal partitioning $(\alpha^{nc^*}, \beta^{nc^*})$ is obtained.

$$\forall(\alpha, \beta) \in \mathbb{G}_{PN} \times \mathbb{G}_{PM},\, (\alpha, \beta) \neq (\alpha^{nc^*}, \beta^{nc^*})$$
$$\Leftrightarrow trace(\alpha^{nc^*\intercal}\alpha) + trace(\beta^{nc^*\intercal}\beta) \leq N + M - 1 \tag{23a}$$
$$\Leftrightarrow \sum_{p=1}^{P} \left\{ \sum_{i=1}^{N} \{\alpha_{pi}^{nc^*}\alpha_{pi}\} + \sum_{k=1}^{M} \{\beta_{pk}^{nc^*}\beta_{pk}\} \right\}$$
$$\leq N + M - 1 \tag{23b}$$

Every time a non-controllable partitioning is obtained a controllability cut (23) is added to the linear constraints before the optimization is computed again. Therefore, the previous non-controllable optimal partitioning can no longer be reached as it is now excluded from the search space. However because the groups are not ordered the same result can be achieved again simply by swapping the rows of α and β, leading to another representation of the same partitioning. Indeed, without any order constraints on the groups, $P!$ identical representations of a single partitioning are possible.

Different techniques can be employed to make the optimization more efficient. First of all, it would be possible to constraint the grouping matrices in order to rank the different groups as it has been done with move blocking matrices (Cagienard et al. (2007)). In this case only one representation per partitioning would be possible. The second solution would be to perform $P!$ controllability cuts every time a non-controllable optimal solution is obtained. Therefore all the possible non-controllable representations of a partitioning would be removed from the search space simultaneously. It is the latter solution that has been implemented in the weak interactions partitioning algorithm. Every time an optimal non-controllable solution is encountered $P!$ controllability cuts are added to the linear set of existing constraints. The optimization is ran iteratively until the least interacting controllable partitioning is found. The next section presents how the partitioning algorithm is built using the linear relaxation as well as the controllability cut technique.

6. PARTITIONING ALGORITHM

The algorithm implemented to perform the system partitioning takes the main state space model as well as the group number P as inputs. It returns the state and input grouping matrices α and β once one of the least interacting controllable partitioning is reached. The algorithm can be described by steps as it is computed. The first step is to build the linear constraints that will be used for the primary and auxiliary variables. Then the optimization is performed yielding the subsystem models as well as their controllability matrices. The last step of the algorithm is to add the appropriate set of controllability cuts if the optimal partitioning presents at least one uncontrollable subsystem as well as to go back to the previous step. Otherwise, the algorithm returns the least interacting controllable partitioning as a final result if the optimal partitioning obtained has all its subsystem models controllable.

The weak interaction partitioning problem is a 0-1 integer linear programming problem. The partitioning algorithm is presented below in algorithm 1.

Algorithm 1 Partitioning algorithm

```
Input  : $A, B, P$
Output: $\alpha, \beta$
while $\exists p \in [\![1; P]\!]$, $rank(C_p) \neq N_p$ do
    if $\exists(\alpha^{nc^*}, \beta^{nc^*})$ then
        Add controllability cuts (23)
        Run the optimization problem (14) subject to (23)
        Extract the subsystem state space models and compute: $\forall p \in [\![1; P]\!]$, $C_p$
    else
        Run the optimization problem (14)
        Extract the subsystem state space models and compute: $\forall p \in [\![1; P]\!]$, $C_p$
    end
end
```

On the very first loop iteration, no solution exists, therefore, the optimization is performed and the first grouping matrices are obtained. The next step is to check that every subsystem is controllable by verifying that (17) holds. If at least one of the subsystems is not controllable then controllability cuts are added to the constraints and the optimization can start again using the reduced search space. The algorithm finishes when the least interacting controllable partitioning comprising P groups is found or when no controllable partitioning can be established.

7. EXAMPLES

This section presents some examples that were used to test the partitioning algorithm. The first example presented was used only to test the linear optimization and does not require any controllability cuts. However the second example was used specifically to demonstrate the controllability check and the controllability cuts.

7.1 Example without controllability cuts

The first example tested is the state space model of a military engine, the Pratt and Whitney $F100$ taken from (Jaw and Mattingly (2009)). The algorithm was used with the parameters given in (24). The reader can see that the whole system is already controllable even before performing any kind of partitioning.

The partitioning obtained can be guessed due to the presence of zeros but also because of the presence of large elements in the matrices. The two grouping matrices resulting from the optimization are given in (25). It is important to notice that this first example does not need any controllability cut.

$$\alpha = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{25a}$$

$$\beta = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{25b}$$

This example has been run on a standard desktop with an execution time of $0.54s$.

7.2 Example involving controllability cuts

The second example is defined such that

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \tag{26a}$$

$$B = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{26b}$$

$$P = 3 \tag{26c}$$

The first 19 iterations of the algorithm result in non-controllable partitionings. After performing 114 controllability cuts, being $19 \times 3!$, the least interacting controllable partitioning is obtained (27).

$$\alpha = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{27a}$$

$$\beta = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{27b}$$

For three groups each controllability cut has to be performed 6 times in order to take into account all the possible permutations. On a standard desktop the total running time was $15.4s$.

8. CONCLUSION

An integer linear programming approached has been presented within this paper to tackle the problem of partitioning a system model into non-overlapping subsystem models. Auxiliary binary variables have been introduced in order to linearise the objective function. Finally, a method similar to Gomory cuts has been implemented in order to

$$A = \begin{pmatrix} -0.3245 \times 10^{1} & -0.2158 \times 10^{1} & -0.9155 \times 10^{3} & 0.5731 \times 10^{0} & 0.1342 \times 10^{3} \\ 0.1642 \times 10^{1} & -0.5941 \times 10^{1} & -0.2816 \times 10^{3} & 0.1897 \times 10^{0} & 0.5705 \times 10^{2} \\ 0.1685 \times 10^{-1} & -0.2554 \times 10^{-1} & -0.1003 \times 10^{2} & 0.7994 \times 10^{-2} & 0.5807 \times 10^{0} \\ 0 & 0 & 0 & -0.1 \times 10^{2} & 0 \\ -0.2163 \times 10^{1} & 0.6862 \times 10^{1} & 0.7405 \times 10^{3} & 0.1195 \times 10^{1} & -0.1715 \times 10^{3} \end{pmatrix} \tag{24a}$$

$$B = \begin{pmatrix} 0.1432 \times 10^{-1} & -0.3553 \times 10^{3} & -0.9906 \times 10^{2} & -0.1549 \times 10^{2} & 0.222 \times 10^{5} \\ 0.2871 \times 10^{0} & 0.7286 \times 10^{3} & 0.2514 \times 10^{2} & -0.6487 \times 10^{2} & 0.8122 \times 10^{4} \\ -0.2469 \times 10^{-2} & -0.103 \times 10^{3} & 0.6333 \times 10^{0} & -0.3213 \times 10^{0} & -0.7418 \times 10^{2} \\ 0.1 \times 10^{2} & 0 & 0 & 0 & 0 \\ -0.1311 \times 10^{0} & 0.3295 \times 10^{3} & -0.25 \times 10^{2} & 0.6257 \times 10^{2} & -0.6445 \times 10^{5} \end{pmatrix} \tag{24b}$$

$$P = 2 \tag{24c}$$

rule out the least interacting partitionnings including at least one non-controllable subsystem model. Because the partitioning problem is a combinatorial problem, the size of the search space increases very rapidly with the size of the system and the number of groups. Therefore the computational cost is important for large scale systems. Future work will address studying the performance of the algorithm as well as its complexity.

REFERENCES

Bakule, L. (2008). Decentralized control: An overview. *Annual Reviews in Control*, 32, 87–98.

Balas, E., Ceria, S., Cornuéjols, G., and Natraj, N. (1996). Gomory cuts revisited. *Operations Research Letters*, 19, 1–9.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35, 407–427.

Bristol, E.H. (1966). On a new measure of interaction for multivariable process control. *IEEE Transactions on Automatic Control*, 11, 133–134.

Browning, T.R. (2001). Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*, 48, 292–306.

Cagienard, R., Grieder, P., Kerrigan, E.C., and Morari, M. (2007). Move blocking strategies in receding horizon control. *Journal of Process Control*, 17, 563–570.

Cavalier, T.M., Pardalos, P.M., and Soyster, A.L. (1990). Modeling and integer programming techniques applied to propositional calculus. *Computers & Operations Research*, 17, 561–570.

Chen, D. and Seborg, D.E. (2003). Design of decentralized pi control systems based on Nyquist stability analysis. *Journal of Process Control*, 13, 27–39.

Gomory, R.E. (1958). Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64, 275–278.

Jamoom, M.B., Feron, E., and McConley, M.W. (1998). Optimal distributed actuator control grouping scheme. *Proceedings of the* 37th *IEEE Conference on Decision & Control*, 2, 1900–1905.

Jaw, L.C. and Mattingly, J.D. (2009). *Aircraft Engine Controls: Design, System Analysis, and Health Monitoring.* AIAA, Reston.

Kariwala, V., Forbes, J.F., and Meadows, E.S. (2003). Block relative gain: properties and pairing rules. *Industrial & Engineering Chemistry Research*, 42, 4564–4574.

Leininger, G.G. (1979). Diagonal dominance for multivariable Nyquist array methods using function minimisation. *Automatica*, 15, 339–345.

Maciejowski, J.M. (2002). *Predictive control with constraints.* Prentice Hall, Harlow.

Manousiouthakis, V., Savage, R., and Arkun, Y. (1986). Synthesis of decentralized process control structures using the concept of block relative gain. *American Institute of Chemical Engineers*, 32, 991–1003.

Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50, 2967–2986.

Moroşan, P.D., Bourdais, R., Dumur, D., and Buisson, J. (2010). Building temperature regulation using a distributed model predictive control. *Energy and Buildings*, 42, 1445–1452.

Motee, N. and Sayyar-Rodsari, B. (2003). Optimal partitioning in distributed model predictive control. *Proceedings of the American Control Conference*, 6, 5300–5305.

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

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, 19, 723–731.

Schuler, S., Münz, U., and Allgöwer, F. (2014). Decentralized state feedback control for interconnected systems with application to power systems. *Journal of Process Control*, 24, 379–388.

Stewart, B.T., Venkat, A.N., Rawlings, J.B., Wright, S.J., and Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, 59, 460–469.

Šiljak, D.D. (1991). *Decentralized control of complex systems.* Academic Press Inc., San Diego.

Williams, P.H. (2013). *Model building in mathematical programming.* Wiley, Chichester.