

On Guided and Automatic Control Configuration Selection^{*†}

M. Castaño Arranz^{1,*}, W. Birk¹ and A. Kadhim¹

Abstract—This paper discusses the guided and automatic control configuration selection in large scale complex systems. Due to the trend of increased level of automation and connectedness which is promoted by the Industry 4.0 strategy and supported by technologies relating to cyber-physical systems and the industrial internet of things, selecting appropriate control strategies becomes increasingly important and complex. This is especially important as a control strategies will limit the achievable performance of the process system, and there are trade-offs between complexity of the control strategies, achievable performance, vulnerability and maintainability.

The paper reviews the state of the art of methodologies that support the practitioners in taking decisions on control strategies, where two main approaches are considered, the guided one and a fully automatic one. It is shown how both approached can be conducted and examples are used to clarify the selection process.

I. INTRODUCTION

In the design of a control system, the selection of an appropriate control strategy plays a key role in achieving desired performance characteristics. There, performance is a quite abstract property representing aspects like quality, energy/resources efficiency, security, environmental impact, only to mention some. While this task appears trivial in small scale, it becomes complex and potentially non-intuitive in large scale.

In general, the Control Structure Design task (CSD) is usually divided in two parts: a) the input-output (IO) selection and the Control Configuration Selection (CCS). The IO selection has been defined in [1] as selecting suitable variables to be manipulated by the controller and suitable variables to be supplied to the controller. The CCS consists of establishing the measurements, which are used in the calculation of each control action.

This paper focuses on the practical aspects of CCS, where Simple configurations are often favored due to being easier to design, implement and maintain, as well as more robust to plant failures (see [2]). However, complex configurations can render controllers with better performance, being therefore a trade-off between configuration simplicity and achievable performance.

According to the studies of the industrial environment in [3], control decisions have to be taken in a short time horizon, and the control strategies will frequently have to be adapted to match the market requirements. This environment leads the CSD question more towards the use of heuristics than to rigorous approaches. Strategies which are straightforward and easy to understand are preferred, since the support of a running control system can depend upon the simplicity of the strategy and its understandability.

The paper will also discuss the context of factory automation for CCS and in what way CCS enables more efficient and precise production processes. The challenges which need to be addressed to make CCS useful in the field of factory automation are also discussed.

The paper is structured as follows. The terminology of guided and automatic CCS is given in Sec. II, the preliminaries on CCS methods are given in Sec. III. The guidelines for CCS with a illustrative example are given in Sec. IV. The automatic methods for CCS are given in Sec. V. Finally, the conclusions are given in Sec. VI.

II. TERMINOLOGY

While there are established terminologies surrounding the control configuration selection area, see e.g. the survey on the control configuration selection field is given in [4], the terms *Guided Control Configuration Selection* and *Automatic Control Configuration Selection* are not generally defined and not yet fully established. In the context of this article we will use the following interpretations.

Guided Control Configuration Selection is the process of using a pre-defined procedure with a number of CCS tools that involve the user in the decision making processes for the selection of a control configuration. The procedure might be semi-automated where the user will get recommended decisions that could be directly accepted before continuing to the next step in the pre-defined procedure.

Automatic Control Configuration Selection is the process of determining a control configuration based on an algorithm that performs a number of computational steps and presents the result to the user. The result of intermediate steps in the algorithm do not necessarily be clear to the user.

In both cases it is possible that the user will get information on the confidence level for the result and the decisions that have been taken. In that way, the user receives additional decision support.

It is the belief of the authors that *Guided Control Configuration Selection* is a necessary step in the development of *Automatic Control Configuration Selection* systems, as

* This work has been partially funded by: a) the Horizon 2020 OPTi project under the Grant Agreement No. 649796, b) the Horizon 2020 DISIRE project under the Grant Agreement No. 636834, c) the WARP project from the VINNOVA SIP-PiiA postdoc program.

† Author's preprint of paper with the same title submitted to IEEE Emerging Technologies And Factory Automation (ETFA) 2017.

¹ Control Engineering Group, Div. of Signals and Systems, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, SE-971 87 Luleå, Sweden

* email: miguel.castano@ltu.se

the user needs to gain a sufficient level of trust to the automatically taken decision.

III. PRELIMINARIES ON INTERACTION MEASURES

We describe in this section a set of Interaction Measures, which are heuristic tools for CCS.

We assume in this section the availability of a model of MIMO system with n inputs and m outputs, which can be represented in state space form by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

with $A \in \mathbb{R}^{q \times q}$, $B \in \mathbb{R}^{q \times n}$, $C \in \mathbb{R}^{m \times q}$, where q is the number of states. Alternatively, the system can be represented by the transfer function $G(s) = C(sI - A)^{-1}B$.

The controllability gramian (P) and observability gramian (Q) for a model are obtained by solving the following continuous-time Lyapunov equations:

$$\begin{aligned}AP + PA^T + BB^T &= 0 \\ A^TQ + QA + C^TC &= 0\end{aligned}$$

A. Interaction Measures Based on Relative Gains

IMs based on relative gains can provide with information for the design of purely decentralized structures by evaluating the interactions of the closed loop system and give necessary conditions for the stability of the configuration.

a) Relative Gain Array: The Relative Gain Array (RGA) is defined as (see [5]):

$$RGA(G(0)) = G(0) \otimes G(0)^{-T} \quad (1)$$

where \otimes denotes element-by-element multiplication and $G(0)^{-T}$ is the inverse of the transpose $G(0)$.

Each element RGA_{ij} is can be alternatively calculated as the quotient between the gain $G_{ij}(0)$ and the gain when the loop formed by pairing $u_j - y_i$ is open while the rest of the loops are closed under perfect steady state control. The RGA is therefore a measure of the so-called 2-way interaction, which is present when control actions of one loop affect other loops which affect the original loop in turn (see [6]). A value RGA_{ij} equal to 1 means that there is no two-way interaction in a loop formed by the pairing $u_j - y_i$. The pairing rules of the RGA are: i) values of RGA_{ij} close to 1 are preferred for pairing ii) large values of the RGA_{ij} should be avoided for pairing since they are related to large interaction and ill-conditioned plants, iii) negative values of RGA_{ij} must be avoided for pairing due to instability issues.

b) Relative Interaction Array: The Relative Interaction Array is defined as:

$$RIA = \phi = (1./G \otimes G^{-T}) - 1 = (1./RGA) - 1 \quad (2)$$

There is a one to one mapping from the RGA to the RIA which implies that $RGA_{ij} = 1 \iff RIA_{ij} = 0$, $RGA < 0 \iff RIA < -1$. The RIA is often denoted as ϕ .

This mapping to the pairing rules of the RIA as: i) values of RIA_{ij} close to 0 are preferred for pairing, ii)

negative values of RGA_{ij} must be avoided for pairing due to instability issues.

An advantage of the RGA is that it is a more linear measure of two-ways interaction.

c) Partial Relative Gains: As shown by [7], the evaluation of Relative Gains and application of the conventional pairing rules might be inadequate (e.g. in triangular plants). Additional indications can be obtained by partially closing the system and recalculating the relative gains for the remaining subsystems.

Assume the reordering and partition of the system transfer matrix $G(s)$ as:

$$G = \begin{bmatrix} G_\Gamma & G_{\Gamma,\phi} \\ G_{\phi,\Gamma} & G_\phi \end{bmatrix} \quad (3)$$

where G_Γ is the subsystem formed by the inputs and outputs in open loop. G_ϕ is the subsystem formed by the inputs and outputs under perfect integral control at dc with the inverse controller $C_\phi = G_{\phi,\phi}^{-1} \cdot \frac{1}{s}$. Then the Partial Gain is the dc-gain of the open loop transfer function for the subsystem G_Γ when the the rest of the system is controlled by C_ϕ is:

$$\bar{G}_\Gamma(0) = G_\Gamma - G_{\Gamma,\phi}(0)G_\phi(0)^{-1}G_{\phi,\Gamma}(0) \quad (4)$$

For a decentralized configuration, the evaluation of Interaction Measures on the Partial Gains which results from closing different combinations of individual loops may reveal additional loop interactions which are not revealed by the original RGA and RIA.

The evaluation of the RGA on Partial Gains has been introduced in the literature under the name Partial Relative Gain (PRG). In this paper, we also apply the RIA to Partial Gains and name the result as Partial Relative Interaction Array (PRIA).

Even if there is a value in considering different combinations of PRGs or PRIAs during CCS, the sheer number of different PGs hinders the interpretation of the results and their integration in the decision making. A system of size $n \times n$ will have $n!$ possible different decentralized configurations, and each of these configurations will have $2^n - n - 1$ different arrays of PGs resulting from closing k control loops with $k = 1 \dots n - 1$. We will seek in this paper a basic exploration of algorithms for automatic decision making for supporting designers in the exploiting the use of PGs.

B. Niederlinski Index (NI)

Assuming diagonal pairing, and denoting \hat{G} as the matrix formed by the diagonal entries of G with zeros in the off-diagonal. NI was defined by [8] as:

$$NI = \det(G(0))/\det(\hat{G}(0))$$

C. Stability and Integrity Conditions

Stabilizability. A system is stabilizable for a selected pairing, if there exist a decentralized controller which can render a stable closed loop system. As mentioned above, a necessary condition for stabilizability is that $NI > 0$.

Integrity. Integrity is a property of a closed loop system. A decentralized closed-loop system has integrity if it remains stable when the gains of all the individual controllers are simultaneously detuned by a factor ϵ in the range $0 < \epsilon \leq 1$ as well as when the gains of any combination of individual controllers are set to 0. Integrity is often a desirable property, since it allows the disconnection of individual loops e.g. for maintenance/retuning or due to failures.

Integral Controllability with Integrity. Since integrity is a property of a closed-loop system, it requires that the design of a control configuration and a controller. It is therefore of interest to determine if a selected decentralized configuration can render controllability prior to the design of the controller parameters. A decentralized configuration is Integral Controllable with Integrity (ICI) if there exists a controller such that the closed-loop system has the integrity property, that is, it remains unconditionally stable and remains stable when individual controllers are arbitrarily brought in and out of service. The IMs based on relative gains provide with necessary conditions for ICI which can be used to rule out configurations with lack of integrity. There are many different necessary tests for integrity, and the more restrictive tests require a larger amount of calculations. We discuss now a basic and simple integrity/stability test which can be derived from the RGA/RIA together with the Niederlinski Index and a more restrictive test of combinatorial nature which requires a larger amount of calculations based on partial gains and which needs to be automated for practical use.

Basic Stability/Integrity test. Assuming that the transfer function matrix $G(s)$ has been reordered in such a way that the diagonal elements are selected as pairing for decentralized control, two necessary conditions for stability and integrity of a decentralized configuration are: i) $NI(G(0)) > 0$, ii) $RGA_{ii} > 0, i = 1, \dots, n$. Condition i) is a necessary condition for the stabilizability of the plant. If condition ii) is violated for any i , then at least one of the following is true: either a) the closed loop system is unstable, b) the loop related to the i -th sensor and actuator is unstable by itself, c) the system becomes unstable if the loop related by the i -th sensor and actuator is brought out of control.

Advanced Integrity test. Theorem: A system $G(s)$ of size $x \times n$ is ICI for variable pairing along the diagonal only if $[RGA(G(0))]_{ii} > 0, i = 0 \dots 1$ and the diagonal elements of all the $2^n - n - 1$ PRGs are positive, being the PRGs calculated from all the possible combinations of closing k control loops with $k = 1 \dots n - 1$.

These two tests can be formulated as a function of the RIA and PRIA instead, since $RGA_{ij} > 0 \iff RIA_{ij} > -1$.

D. Participation Matrix

The Participation Matrix (PM) was introduced in [9] as:

$$PM_{ij} \triangleq \frac{\text{trace}(P_j Q_i)}{\sum_{m,n} \text{trace}(P_m Q_n)} = \frac{\text{trace}(P_j Q_i)}{\text{trace}(PQ)} \quad (5)$$

where $\text{trace}(P_j Q_i)$ is the trace of the product of the controllability gramian related to the j th input P_j , and the

observability gramian related to the i th input Q_i (see [9]). The sum of the elements of PM adds up to 1. Each element PM_{ij} quantifies the contribution of the input-output channel G_{ij} relative to the total controllability and observability of the complete system G . The control configuration is then designed using PM by selecting the most important input-output channels. The user can comprehend the amount of the total process dynamics that the reduced model is reflecting by evaluating the closeness of the sum of the selected PM values to 1.

There are other gramian-based IMs with similar properties, which use other quantifiers instead of $\text{trace}(PQ)$. These are, the Hankel Interaction Index Array introduced in [10] which uses the Hankel-norm, and Σ_2 introduced in [11] which uses the \mathcal{H}_2 -norm.

The following heuristic rules have been formulated in [9] for the use of the PM, and are currently applied for selecting the most significant input-output channels with the use of any gramian-based IM:

Rule 1. The simplest control configuration whose total contribution (sum of considered elements in the IA) exceeds an arbitrary threshold τ is selected as candidate. This configuration considers the input-output channels with largest significance while considering at least one input-output channel in each row. Control configurations designed with $\tau \leq 0.7$ are likely to derive in satisfactory performance for systems with a few sensors and actuators. As the scales of the system increase, the value of the threshold should be reduced.

Rule 2. In a hypothetical process with r input-output channels where all the channels have the same contribution, this contribution will be equal to $1/r$. This suggests that in a more heterogeneous scenario there is no benefit from considering those input-output channels for which $IA_{ij} \ll 1/r$. The converse is also true, and the those input-output channels with $IA_{ij} \gg 1/r$ present a significant contribution in the process dynamics.

The gramian-based IMs concentrate the frequency domain properties of the system in a real valued array, and therefore it is traditional to restrict the range of frequencies of interest with e.g. the use of pre-filters (see [11]). Additionally, the gramians are not defined for systems with pure integrators but they can be analyzed by restricting the range of frequencies (see [12]).

IV. GUIDED CONTROL CONFIGURATION SELECTION

We assume in this section the availability of process models, and first describe briefly the preparation steps prior to CCS in IV-A, followed by a guided method for CCS in IV-A. An example is later given in IV-C.

A. Preparing the models: scaling, model reduction, decomposition and IO selection.

a) *Scaling:* Most of the methods for CCS are sensitive to the scaling of inputs and outputs, as well as the methods model reduction, process decomposition and I/O selection.

Appropriate scaling is usually performed by using the maximum allowed or expected change in the process variables. We will denote the maximum expected/allowed change of inputs and outputs by \bar{u}_j and \bar{y}_i and collect them in diagonal matrices of the form $D_u = \text{diag}(\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n)$ and $D_y = \text{diag}(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n)$. The resulting scaled transfer function \hat{G} is:

$$\hat{G} = D_y^{-1} \cdot G \cdot D_u \quad (6)$$

Even if scaling is principle a simple task, it is often hindered by the lack of process information. An alternative is presented in [13], where inputs and outputs are scaled by the standard deviation of the signals. This requires knowledge of only the standard deviation of the input signals, since that of the output signals can be calculated using the known model $G(s)$. For uncorrelated Gaussian excitation signals:

$$\sigma_{y_i} = \sqrt{\sum_{j=1}^n \|G_{i,j}(s)\|_2^2 \cdot \sigma_{u_j}^2} \quad (7)$$

For other kinds of excitation signals, or nonlinear systems, the system can be simply simulated and the outputs' sample variance calculated.

b) Model order reduction: The use of high order models can lead to high computational load and numerical errors e.g. calculating the gramians. Balanced truncation is a traditional approach to reduce the order of a model (see [14]). To choose the number orders to be reduced, the Hankel Singular Values (HSVs) are often used, being the number of insignificant HSVs corresponding to the number of orders to be reduced. An upper bound on the approximation error can be calculated from the sum of the disregarded Hankel Singular values [15].

Balanced truncation can be conveniently performed with the MATLAB function *balancmr* and the HSVs can be obtained with the *hsvd* function.

c) Process Decomposition: The combinatorial nature of the CCS problem makes the CCS methods hard to apply to large scale systems, and in these cases the CCS is usually preceded by a step where manipulated and controlled variables are grouped into subsystems where the number of variables is reduced to no more than approximately a couple of dozens. The resulting subsystems are composed by variables with strong mutual interconnections. The control configurations are designed within the subsystem boundaries, and the resulting controllers for the subsystems have to be appropriately combined (see [16]). Methods for such a decomposition have been proposed in [17], [18].

A convenient approach for the decomposition is the use of the Dulmage-Mendelsohn decomposition given in [19], which is readily available for MATLAB users under the function *dmperm*.

d) Input-Output Selection: The Input-Output (IO) selection involves decisions on the number, place and type of actuators and sensors. This selection affects the performance, complexity and costs of the control system (see [1]).

The IO selection is a problem of combinatorial nature, and it was proposed by [20], to use the minimum eigenvalue

of the controllability and observability gramians to evaluate which of the combinations of inputs and outputs are more adequate. However, models are often created with a high order than the needed to describe the physical properties of the process, and each extra state will generate additional eigenvalues of low value. Even if model reduction is performed, the truncation is performed by selecting a threshold for removing states with low controllability/observability. Therefore, the minimum eigenvalue is often not a direct property of the process itself, and is highly influenced by the order selected to represent the system. We propose here to use the maximum eigenvalue instead.

The controllability gramian can be used for the selection of p inputs by evaluating the controllability of the reduced systems resulting from choosing any p -combination of inputs chosen from the n total number of inputs. The reduced B matrix (B_p), resulting from the optimal selection of p actuators is obtained from the following integer optimization:

$$B_p = \arg. \max_{B_i = B(i,:), i \in \binom{n}{p}} \bar{\lambda}(P(A, B_i, C))$$

where $\bar{\lambda}(P(A, B_i, C))$ is the maximum eigenvalue ($\bar{\lambda}$) of the controllability matrix $P(A, B_i, C)$ of the minimal realization of the reduced system formed by picking the columns of B indicated by the vector i . The vector i belongs to the set $\binom{n}{p}$ of p -combinations of the first n integers.

Similarly, the reduced C matrix (C_q), resulting from the optimal selection of q sensors from the total m sensors is obtained from the following program:

$$C_q = \arg. \max_{C_j = B(:,j), j \in \binom{m}{q}} \bar{\lambda}Q(A, B, C_i)$$

B. Procedure for Control Configuration Selection using Interaction Measures

The following procedure can be used for selecting control structures based on the IMs.

Step 1. Seek a decentralized control structure using methods based on relative gains, i.e. either the RGA or the RIA.

Step 2. Check necessary conditions for the stabilizability and ICI integrity of the decentralized control structure using at least the basic stability/integrity test defined in Sec. III-C.

Step 3. Design and test a decentralized controller. If the performance is found to be acceptable, proceed to the integration of the controller. Otherwise continue to Step 4 and increase the complexity of the controller.

Step 4. Calculate a gramian-based Interaction Measures, e.g. PM. Apply Rule 1 in Sec.III-D by starting with the decentralized configuration previously designed and add input-output channels in crescent order of significance until a designed threshold τ is reached.

Step 5. Review the sparse configuration by using *Rule 2* in Sec.III-D. This is done by adding or removing input-output channels which are suspected to present a significant or insignificant contribution respectively. In both cases the designer has to judge if the increase or decrease of the complexity of the configuration is justified by the increase or decrease in the total dynamic contribution.

Step 6. Design a (sparse) controller and test it in simulations prior to implementation. Iterate in this procedure if needed.

C. Guided Example: Secondary Heating System

We follow now the steps given for the guided CCS with a case study on a Secondary Heating System (SHS).

The SHS has been introduced in [21]. The transfer function of the process permuting the second and third outputs is:

$$G(s) = \begin{pmatrix} \frac{0.0179}{s} & 0 & \frac{-0.0016}{s} & 0 & \frac{-0.0104}{s} \\ 0 & \frac{0.00071}{s} & \frac{0.00038}{s} & \frac{-0.0012}{s} & 0 \\ 0 & \frac{-0.01}{710s+1} & \frac{0.11}{615s+1} & 0 & 0 \\ 0 & \frac{0.08}{1015s+1} & \frac{-0.08}{817s+1} & \frac{-0.36}{730s+1} & \frac{-0.36}{604s+1} \\ 0 & \frac{-0.00047}{s} & \frac{0.00054}{s} & \frac{0.0012}{s} & \frac{0.0024}{s} \end{pmatrix} \quad (8)$$

Preparing the process models To find appropriate scaling, we assume that the inputs are excited with uncorrelated sequences of flat unitary psd within the frequencies $[10^{-3.0065}/10, 10^{-2.781} \dot{1}0]Hz$ and measure the standard deviation of the outputs resulting in the input scaling matrix $D_u = \text{diag}(1, 1, 1, 1, 1)$ and output scaling matrix $D_y = [0.4168, 0.0235, 0.0022, 0.0114, 0.0535]$. The scaled transfer function is $\hat{G}(s) = D_y^{-1} \cdot G \cdot D_u$.

Step 1. Design of a decentralized Controller using Relative Gains. This process has pure integrators, which are a problem for the calculation of Interaction Measures. A direct calculation of the RGA is not possible due to the infinite dc-gain. However, the s coefficients representing the integrators are common divisors of rows, which means that they can be factorized as scaling coefficients of the rows and simply canceled out, since the RGA is insensitive to scaling. The resulting calculation of the RGA for the SHS is:

$$RGA = \begin{pmatrix} \boxed{1} & 0 & 0 & 0 & 0 \\ 0 & \boxed{1.0432} & 0.0508 & -0.0939 & 0 \\ 0 & 0.0499 & \boxed{0.9501} & 0 & 0 \\ 0 & -0.7836 & 0.0712 & \boxed{2.1879} & -0.4755 \\ 0 & 0.6906 & -0.0721 & -1.0939 & \boxed{1.4755} \end{pmatrix} \quad (9)$$

The RGA indicates therefore the diagonal pairing by selecting positive element with proximity to 1.

Step 2. Check stability/integrity tests. The Niederlinski Index is also scaling independent, and its calculation for the diagonal pairing leads to; $Ni(G(0)) = 9.8509 > 0$. Together with the fact that positive RGA elements have been selected, this satisfies the necessary conditions in the basic integrity/stability test.

Step 3. Design and test a decentralized controller. A well-tuned decentralized controller for the SHS is composed by the following SISO controllers:

$$C_{11} = 0.54817(s + 0.001966)/s; C_{22} = 13.82(s + 0.001966)/s; \\ C_{33} = 53.095(s + 0.003716)/s; C_{44} = -13.245(s + 0.006829)/s; \\ C_{55} = 4.0884(s + 0.001966)/s;$$

These SISO controllers have been designed independently for each of the transfer functions in the diagonal of $G(s)$ assuming that they are individual independent systems, and the designed step responses for each of the individual control loops are depicted with continuous lines in Fig. 1. The interactions between the loops will lead to performance degradation and differences between the designed step responses

and those of the complete system, which are represented with dashed line. For the loop corresponding to C_{11} the designed and complete system's responses coincide due to the fact that there is no two-way loop interaction as indicated by the corresponding value of the RGA being equal to 1. The loops which present a larger difference between the designed and final system's responses are those related to values of the RGA with larger deviations from 1, i.e. the loops related to C_{44} and C_{55} ,

Whilst the designed decentralized controller is stable, there is a significant amount of loop interaction. Some significant perturbations between control loops are depicted in Fig. 3, where the steps responses from CL_{ij} denote the perturbation on the output y_i when a unitary step is placed for the reference of output y_j .

We proceed to the design of a sparse controller in order to reduce the loop interaction.

Step 4. Design a sparse controller using gramian-based IMs We restrict the calculation of PM to a frequency of interest due to the existence of integrators (see [21]). Multiplying the system by a Butterworth filter of second order with bandpass $[10^{-3.0065}/10, 10^{-2.781} \dot{1}0]Hz$ and calculating PM results in:

$$PM(\hat{G}(s)) = \begin{pmatrix} \boxed{0.1705} & 0 & 0.0014 & 0 & \boxed{0.0576} \\ 0 & \boxed{0.0844} & 0.0242 & \boxed{0.2411} & 0 \\ 0 & 0.0007 & \boxed{0.0954} & 0 & 0 \\ 0 & 0.0013 & 0.0016 & \boxed{0.0343} & 0.0384 \\ 0 & 0.0071 & 0.0094 & 0.0465 & \boxed{0.1861} \end{pmatrix} \quad (10)$$

The sum of the diagonal elements of PM relates to the designed decentralized structure and is:

$$\sum_{i=1}^n (PM_{ii}) = 0.5707 \quad (11)$$

Which is below the threshold of $\tau = 0.7$. Adding the next element in order significance PM_{24} results in:

$$\sum_{i=1}^n (PM_{ii}) + PM_{24} = 0.8118 \quad (12)$$

Which is above the threshold of $\tau = 0.7$.

Step 5. Revise the Configuration. Since the system has $5 \times 5 = 25$ input-output channels, the average channel contribution is $1/25 = 0.04$. The configuration related to Eq. (12) has a contribution larger than $\tau = 0.7$, however it is neglecting a channel with a contribution significantly above the average channel contribution: $PM_{15} = 0.0576$. We then decide to add the channel G_{15} to the reduced model. The sum of the PM elements of the selected input-output channels is:

$$\sum_{i=1}^n (PM_{ii}) + PM_{24} + PM_{15} = 0.8694 \quad (13)$$

Step 6. Design and test a sparse controller. We increase now the complexity of the previously designed decentralized controller by adding two feed-forward elements related to G_{24} and G_{25} . These two feed-forward controllers. The transfer function G_{15} will create a loop perturbation on the loop formed by $u_1 - y_1$ from the loop formed by $u_5 - y_5$.

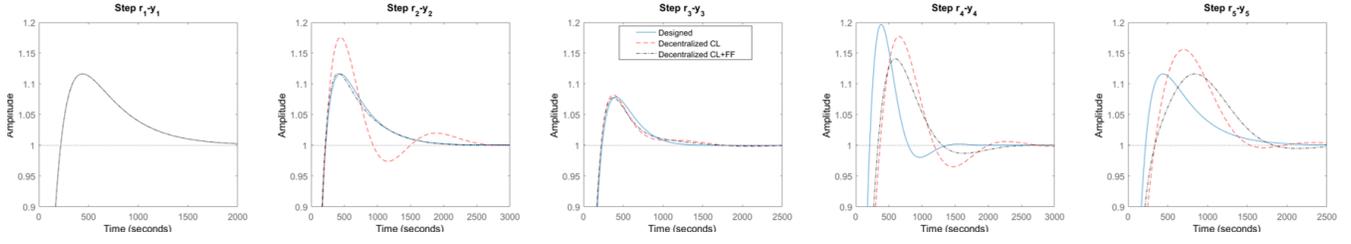


Fig. 1. Decentralized Controller for the SHS. Continuous: designed response considering SISO independent systems. Dashed: resulting step response in the interacting system with decentralized control. Dashed-dotted: step response of the closed loop system aggregating a decentralized controller and feed-forward actions.

A possible compensation is to feed-forward the value of the control action u_5 to u_1 , which will therefore require additional on u_1 . As we can see from the first column in Eq. (10), the actuator u_1 only has impact on its corresponding output y_1 and therefore extra actuation with a feed-forward action will not derive in additional interactions between loops. Similarly, the compensation for G_{24} can be resolved by feeding-forward the control action u_4 to u_2 , since u_2 has main impact on its related output y_2 and low impact on the rest of the outputs (see second column in Eq. (10)). A simplified diagram illustrating the location of these feed-forward actions is given in Fig. 2. These feed-forwards can be designed as:

$$FF_{24} = -\hat{G}(2,4)/\hat{G}(2,2) = 1.69; FF_{15} = -\hat{G}(1,5)/\hat{G}(1,1) = 0.581$$

The final closed loop transfer function $CLFF$ with the sparse controller resulting from aggregating the decentralized controller and the feed-forward actions is:

$$CLFF = \hat{G}(I + FF) \cdot \hat{C}(I + \hat{G}(I + FF) \cdot \hat{C})^{-1} \quad (14)$$

where $FF_{24} = 1.69$ and $FF_{15} = 0.581$ and the rest of the elements of FF are 0. \hat{C} is the scaled decentralized controller: $\hat{C} = D_u^{-1} \cdot C \cdot D_y$.

The step responses of each of the loops are depicted in the dashed-dotted line in Fig. 1, where it can be observed that the feed-forward compensations are able to bring the responses closer to the designed ones in comparison with the decentralized controller. The loop perturbations for the decentralized controller depicted in 3 are now depicted in Fig. 4 for the sparse controller with feed-forward actions. It can be observed that there is still a significant p perturbation on y_4 for a step on the reference for y_5 (see CL_{45} and $CLFF_{45}$), but the rest of the perturbations have been very significantly reduced.

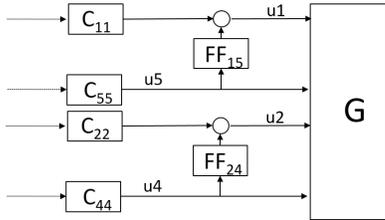


Fig. 2. Simplified diagram showing the location of the feed-forward actions.

V. AUTOMATIC CONTROL CONFIGURATION SELECTION

Decentralized configurations has to be sought for among $n!$ possible candidates in $n \times n$ system . If we consider

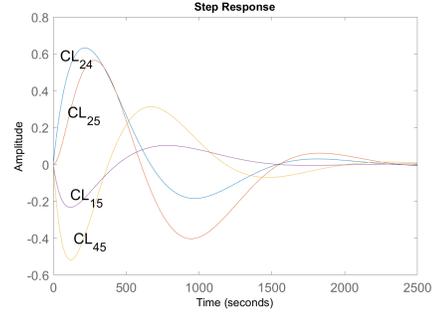


Fig. 3. Some loop perturbations of the decentralized controller.

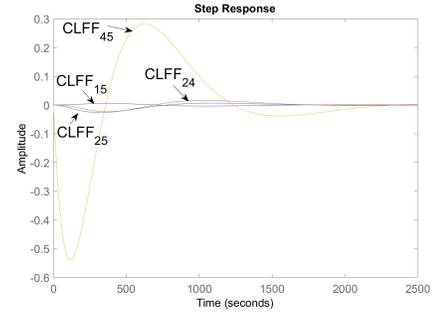


Fig. 4. Some loop perturbations of the decentralized controller with Feed-Forward addition.

any combination of the input-output channels, the number of possible configurations is around $29 \cdot 10^6$ for a 5×5 system as Secondary Heating System considered in this paper. Although some configurations could be discarded for different reasons, selecting adequate configuration manually among the feasible configurations is usually not an easy task even with the use of IMs. This leads to the recent interest in the development of algorithms for automatic selection using IMs, see [22][23] and [24].

A. Automatic Pairing using Relative Gains

In this section, an automated selection of the best configuration following the RIA pairing rules is demonstrated. The RIA has been chosen for: i) its pairing rules take into consideration the stability, integrity and robustness conditions as well as the minimization of the overall interaction, ii) it is a more linear measure of interaction than the RGA, iii) its pairing rules can be formulated as an assignment problem

(AP) which can be solved using linear programming (LP) methods without the need of combinatorial solutions.

This assignment problem is formulated as (see [25]):

$$\begin{aligned} \min_{x_{ij}} \quad & \sum_{i=1}^n \sum_{j=1}^n |\phi_{ij}| x_{ij} \quad (15) \\ \text{subject to} \quad & \sum_{j=1}^n x_{ij} = 1 \text{ for } i = 1, \dots, n \\ & \text{and } \sum_{i=1}^n x_{ij} = 1 \text{ for } j = 1, \dots, n \\ \text{where } x_{ij} = \quad & \begin{cases} 0 & \text{for all } i, j \text{ correspond to } \phi_{ij} \notin \Omega \\ \{0, 1\} & \text{otherwise} \end{cases} \end{aligned}$$

where $\Omega = \{\phi_{ij} : \phi_{ij} > -1\}$. This problem minimizes the sum of the selected values of the RIA subject to the conditions that one and only one element per row and column has to be selected, and that no negative RIA values are selected. Possible methods to solve this LP problem are the Hungarian algorithm used in [25] and the Push-Pull Algorithm used in [26]. The advantage of the Push-Pull algorithm is that bounds can be obtained on the perturbations of the RIA elements which are allowed to guarantee that the same solution is attained.

Example. The RIA for the SHS is:

$$RIA = \begin{pmatrix} 0 & \infty & -\infty & \infty & -\infty \\ \infty & -0.0414 & 18.7018 & -11.6445 & \infty \\ -\infty & 19.0536 & 0.0525 & -\infty & -\infty \\ \infty & -2.2761 & 13.0375 & -0.5429 & -3.1030 \\ \infty & 0.4481 & -14.8642 & -1.9141 & -0.3223 \end{pmatrix} \quad (16)$$

By applying the Push-Pull algorithm on $|\Phi|$, the best configuration suggested is $(y_1 - u_1), (y_2 - u_3), (y_3 - u_2), (y_4 - u_4)$ and $(y_5 - u_5)$. This is the decentralized configuration for which SISO controllers have been designed in the guided CCS example.

B. Automatic pairing with integrity using Partial Relative Gains

An advantage of the previously discussed automated pairing method is that can be efficiently solved using linear programming, however it only tests the second condition of the basic stability/integrity test. The value of NI has to be manually tested on the resulting configuration for completing the basic test.

Checking the advanced integrity test cannot be formulated as a LP problem, and can only be done in a combinatorial fashion. In coordination with this publication, we have released the code for the algorithm to find all the possible decentralized configurations which satisfy the advanced integrity test for ICI¹. This algorithm visits individually all the possible decentralized configurations. For each decentralized configuration, all the PRIAs are visited, and the configuration is discarded as soon as a PRIA leads to a negative number along the diagonal. As long as the PRIAs remain positive for a decentralized configuration, we aggregate the trace of all the visited PRIAs. The aggregated value of the PRIAs is more reliable measure of interaction, as discussed by [7].

¹See the function *findICI* at the MATLAB File Exchange

Example. A seek of decentralized configurations satisfying the advance ICI test is performed on the SHS. There are only two decentralized configurations which fulfill the ICI test: Configuration 1 is the same as the obtained in the guided example and has a aggregated trace of the PRIAs of 10.8160. Configuration 2 is the formed by the pairing $u_1 - y_1, u_2 - y_3, u_3 - y_2, u_4 - y_4, u_5 - y_5$ and has an aggregated trace of the PRIAs of 301.9832. Configuration 1) is preferred for having a much lower level of loop interaction.

C. Automatic Control Configuration Selection using gramian-based IMs

Following the rules for the use of the gramian-based IMs described in III-D, we are interested in formulating a program for finding the configuration Θ which satisfies the following:

- Θ has the minimum number of channels p which can lead a contribution larger than τ .
- Θ is the configuration with p channels which has the largest contribution.
- Θ has full structural rank in order to guarantee structural controllability.
- Θ includes all the channels with a contribution larger than an arbitrary threshold Δ_1 .
- Θ does not include any channel with a contribution smaller than a threshold Δ_2 .

These conditions can be formulated as the integer program: footnotesize

$$\begin{aligned} \Theta = \arg \max_{\Theta_{ij} \in \{0,1\}} \quad & -\|\Theta\|_0 + \|\Theta \circ IA\|_1 \\ \text{subject to:} \quad & \|\Theta \circ IA\|_1 > \tau \\ & \text{sprank}(\Theta) = \min(m, n) \\ & \text{if } IA_{ij} > \Delta_1 \rightarrow \Theta_{ij} = 1 \\ & \text{if } IA_{ij} < \Delta_2 \rightarrow \Theta_{ij} = 0 \quad (17) \end{aligned}$$

The thresholds Δ_1 and Δ_2 represent the consideration of Rule 2 in the method.

This program first finds the smallest number of channels $p = \|\Theta\|_0$ which can derive in a contribution $\|\Theta \circ IA\|_1 > \tau$, and then finds the structure θ with p channels which maximizes the contribution $\|\Theta \circ IA\|_1$. This is due to the fact that p is an integer and $\|\Theta \circ IA\|_1 \leq 1$.

Solving this integer program by testing all the possible configurations can lead to a burdensome computational effort. However, the number of configurations to be evaluated can dramatically be reduced by considering only the configurations which satisfy the three last inequalities in Eq. (17). More explicitly, the last two inequalities can first be used to fix values of Θ , limiting the number of candidate configurations to be generated, and later we can screen the candidate configurations to preserve only those with full structural rank.

Example The application of this algorithm for the SHS using PM in Eq. (10) with $\tau = 0.7$, $\delta_1 = 0.05$ and $\delta_2 = 0.004$ result in the sparse configuration with the PM elements marked in Eq. (10). This is the configuration for

which an sparse controller has been designed in the guided CCS example.

VI. DISCUSSION AND CONCLUSIONS

Selection of control configuration in large scale systems has been a research topic in the area of process control and automation for a long time. Nevertheless, real life applications of the methodologies still rely on manual work and a very high skill level of automation or process engineers.

In factory automation the low control of continuous processes can impose limitations on achievable quality and cycle times. These unit processes need to be equally efficient as the surrounding discrete production processes, to achieve adequate overall performance. An advantage in this context is the trend of modeling which renders virtual twins of continuous and discrete production processes to a degree which was not possible prior to the beginning of 2000.

Since control configuration selection tools largely depend the availability of (dynamic) models of the processes, the virtual twins of the production processes enable the easy application of these tools.

A remaining challenge is the automation of the selection process and the support of the control engineers in the decision making. Most of the available methodologies require deep insight of the engineer in the limitations of the methods and their indications. Moreover, there are only very few software tools available which enable the engineer to focus on the selection process instead of implementing models and perform manual analysis work. A good step in that direction is the ProMoVis tool which provides a visualization and numerous CCS methods at the disposal of the engineer (see [26]).

Despite this progress automated and guided methods need to be validated and further developed for real-life applications. Factory automation cases are essential, as they provide different production paradigms as well as hierarchical aspects that need to be considered in the selection process. It is the belief of the authors that CCS methods need to be further developed for that context and to be able to deal with complete production processes.

To conclude, control configuration selection methods have matured largely and are now able to deal with many different categories of processes. Moreover, there are initial methodologies for guided and automatic control configuration selection available. Some of these methods are presented in this paper and their limitations are discussed. A number of development directions are indicated and should be addressed in future research and development efforts.

REFERENCES

- [1] M. van de Wal and B. de Jager, "A review of methods for input/output selection," *Automatica*, vol. 37, no. 4, pp. 487 – 510, 2001.
- [2] D. Šiljak, "Decentralized control and computations: status and prospects," *Annual Reviews in Control*, vol. 20, no. 0, pp. 131 – 141, 1996.
- [3] J. Downs, *Plantwide Control: Recent Developments and Applications. Chapter 2. Industrial Perspective on Plantwide Control*, G. P. Rangaiah and V. Kariwala, Eds. John Wiley & Sons, Inc., 2012.
- [4] M. Castano Arranz, W. Birk, and G. Nikolakopoulos, "A survey on control configuration selection and new challenges in relation to wireless sensor and actuator networks," in *IFAC World Congress 2017, July 9-15, Toulouse, France.*, 2017.
- [5] E. Bristol, "On a new measure of interaction for multi-variable process control," *IEEE Transactions on Automatic Control*, vol. AC-11, no. 1, pp. 133 – 134, 1966.
- [6] Z.-X. Zhu, "Loop decomposition and dynamic interaction analysis of decentralized control systems," *Chemical Engineering Science*, vol. 51, no. 12, pp. 3325–3335, 1996.
- [7] K. E. Häggblom, "Control structure analysis by partial relative gains," in *Proc. of the 36th Conference on Decision & Control, San Diego, USA*, December 1997, pp. 2623–2624.
- [8] A. Niederlinski, "A heuristic approach to the design of linear multi-variable interacting control systems," *Automatica*, vol. 7, no. 6, pp. 691 – 701, 1971.
- [9] M. E. Salgado and A. Conley, "Mimo interaction measure and controller structure selection," *International Journal of Control*, vol. 77, no. 4, pp. 367–383, 2004.
- [10] B. Wittenmark and M. E. Salgado, "Hankel-norm based interaction measure for input-output pairing," in *Proc. of the 2002 IFAC World Congress, Barcelona*, 2002.
- [11] W. Birk and A. Medvedev, "A note on gramian-based interaction measures," in *Proc. of the European Control Conference 2003, University of Cambridge, UK*, 2003.
- [12] M. Castaño, W. Birk, and P. Asplund, "Control configuration selection for integrating processes using graphs," in *2015 IEEE Conference on Control Applications (CCA)*, Sept 2015, pp. 1606–1611.
- [13] M. Castaño and W. Birk, "New methods for interaction analysis of complex processes using weighted graphs," *Journal of Process Control*, vol. 22, no. 1, pp. 280 – 295, 2012.
- [14] M. G. Safonov and R. Chiang, "A schur method for balanced-truncation model reduction," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 729–733, 1989.
- [15] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their $1, \infty$ -error bounds," *International journal of control*, vol. 39, no. 6, pp. 1115–1193, 1984.
- [16] G. S. Manfred Morari, Yaman Arkun, "Studies in the synthesis of control structures for chemical processes. Part I: Formulation of the Problem. Process Decomposition and the Classification of the Control Tasks. Analysis of the Optimizing Control Structures." *AICHE Journal*, vol. 26, no. 2, pp. 220–231, 1980.
- [17] M. E. Sezer and D. D. Šiljak, "Nested ε -decompositions and clustering of complex systems," *Automatica*, vol. 22, no. 3, pp. 321–331, 1986.
- [18] A. I. Zečević and D. D. Šiljak, "A decomposition-based control strategy for large, sparse dynamic systems," *Mathematical problems in engineering*, vol. 1, pp. 33–48, 2005.
- [19] A. Pothén and C.-J. Fan, "Computing the block triangular form of a sparse matrix," *ACM Transactions on Mathematical Software (TOMS)*, vol. 16, no. 4, pp. 303–324, 1990.
- [20] D. Georges, "The use of observability and controllability gramians or functions for optimal sensor and actuator location in finite-dimensional systems," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*, vol. 4, Dec 1995, pp. 3319–3324 vol.4.
- [21] M. Castaño, W. Birk, and P. Asplund, "Control configuration selection for integrating processes using graphs," in *2015 IEEE Conference on Control Applications (CCA)*, Sept 2015, pp. 1606–1611.
- [22] I. Kookos and A. Lygeros, "An algorithmic method for control structure selection based on the RGA and RIA interaction measures," *Chemical Engineering Research and Design*, vol. 76, no. 4, pp. 458–464, 1998.
- [23] V. Kariwala and Y. Cao, "Branch and bound method for multiobjective pairing selection," *Automatica*, vol. 46, no. 5, 2010.
- [24] A. Fatehi and A. Shariati, "Automatic pairing of mimo plants using normalized rga," in *Control & Automation, 2007. MED'07. Mediterranean Conference on. IEEE*, 2007, pp. 1–6.
- [25] A. M. Kadhim, M. Castaño Arranz, and W. Birk, "Automated control configuration selection considering system uncertainties," *Industrial & Engineering Chemistry Research*, 2017.
- [26] W. Birk, M. Castaño, and A. Johansson, "An application software for visualization and control configuration selection of interconnected processes," *Control Engineering Practice*, vol. 26, no. 0, pp. 188 – 200, 2014.