
# Finding feedforward configurations using gramian based interaction measures

Downloaded from: https://research.chalmers.se, 2025-12-04 23:22 UTC

Citation for the original published paper (version of record):
Bengtsson, F., Wik, T. (2021). Finding feedforward configurations using gramian based interaction measures. Modeling, Identification and Control, 42(1): 27-35. http://dx.doi.org/10.4173/mic.2021.1.3

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Modeling, Identification and Control, Vol. 42, No. 1, 2021, pp. 27–35, ISSN 1890–1328

# Finding feedforward configurations using gramian based interaction measures

Fredrik Bengtsson  Torsten Wik

*Department of Electrical Engineering, Chalmers University of Technology, SE 412 96 Göteborg, Sweden, (email:fredben@chalmers.se, tw@chalmers.se)*

**Abstract**

A sparse control structure can be seen as a decentralised controller that is expanded to include feedforward or MIMO blocks. Here, use of the gramian based interaction measures to determine a sparse control structure with feedforward is examined. A modification to the method used today is proposed and it is demonstrated that it results in a considerable improvement. Furthermore, recently proposed modifications to scaling gramian based measures are expanded to also cover sparse control structures. We show that the method that yields the best result is when two different scaling methods are combined, using one to design a decentralized controller and another to find feedforward connections.

*Keywords:* Process control, input-output pairing, interaction matrix, control configuration selection, sparse control, feedforward.

## Introduction

A common issue in industrial process control systems is that interaction between different parts of the plant gives rise to a multiple input multiple output (MIMO) system, where the same input may affect multiple outputs, or conversely, the same output is affected by multiple inputs. One method to control a MIMO system is to divide it into subsystems of one input and one output and implement SISO controllers for each of the subsystems. This control strategy is called decentralized control and remains widely used in industry (Khaki-Sedigh and Moaveni (2009)). It has several advantages compared to implementing a MIMO controller for the entire system, as it allows the use of relatively easy to design low dimensional controllers. Moreover, it is generally less vulnerable to sensor and actuator failures than more complex control schemes that try to control the entire system with one overarching control scheme.

However, sometimes interactions between the different inputs on the output result in a decentralized control scheme yielding poor results. One solution to this is to expand the decentralized control structure to include decoupling feedforward to remove the most problematic interactions. This yields what is called a sparse controller structure. However, this requires determining which interactions that are appropriate to remove with feedforward, and which ones where implementing feedforward may create interactions that result in a worsened control outcome.

One group of measures which can be used to devise a sparse controller structure is the gramian based measures. This group includes the $\Sigma_2$ method (Birk and Medvedev (2003)), the participation matrix (PM) (Conley and Salgado (2000)) and the Hankel interaction index array (HIIA) (Wittenmark and Salgado (2002)). These methods use the controllability and observability gramians to create an interaction matrix which gives a gauge of how much each input affects each output. The interaction matrices (IMs) can then be used to devise both decentralized and sparse control structures.

While a rule of thumb on how to select a sparse control structure from the gramian based measures

doi:10.4173/mic.2021.1.3
© 2021 Norwegian Society of Automatic Control

has been presented in Conley and Salgado (2000), no deeper analysis on the subject has been presented. Here we will further examine how to best derive a sparse controller structure when using the HIIA and propose a new method for determining which signals that are appropriate for feedforward.

A new method of scaling the gramian based measures has been proposed, and it has been demonstrated that it yields improved results (Bengtsson et al. (2020)). We will here further examine the scaling of the IMs and see how it can best be adapted for the design of sparse control structures.

# 1. The Gramian based measures

## 1.1. Gramian based measures

The gramian based measures (PM, HIIA and $\Sigma_2$) can be calculated from a system's transfer function matrix (TFM) (Birk and Medvedev (2003); Conley and Salgado (2000); Wittenmark and Salgado (2002)). Given a TFM

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \cdots & g_{1n}(s) \\ g_{21}(s) & g_{22}(s) & & \\ \vdots & & \ddots & \\ g_{n1}(s) & & & g_{nn}(s) \end{bmatrix} \tag{1}$$

each measure generates an $n \times n$ interaction matrix (IM) $\Gamma$. For the HIIA and $\Sigma_2$ it is generated by

$$[\Gamma]_{ij} = \frac{||g_{ij}(s)||}{\sum_{kl} ||g_{kl}(s)||} \tag{2}$$

using the Hankel norm and 2-norm for the HIIA and $\Sigma_2$, respectively. The PM is derived in a similar fashion, but it uses the squared Hilbert-Schmidt norm, i.e. the IM is generated by

$$[\Gamma]_{ij} = \frac{||g_{ij}(s)||_{HS}^2}{\sum_{kl} ||g_{kl}(s)||_{HS}^2}. \tag{3}$$

Once an IM is generated, a decentralized pairing is generated by choosing the pairing (one element in each row and column) that yields the largest sum of elements from the IM. For efficient implementation in finding which pairing yields the largest sum of elements one can, for example, use the Hungarian algorithm, as in Fatehi (2011).

## 1.2. Feed forward control

Once a decentralized control structure has been found it can be expanded to include feedforward blocks. To understand how this works, we begin by examining a 3 by 3 system, i.e.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & G_{13}(s) \\ G_{21}(s) & G_{22}(s) & G_{23}(s) \\ G_{31}(s) & G_{32}(s) & G_{33}(s) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{4}$$

Let us assume that the inputs and outputs have been ordered such that our decentralized controller design has a diagonal pairing where $y_i$ is controlled by $u_i$ $\forall i$. Now, $u_1$ will also affect $y_2$ and $y_3$ by $G_{21}(s)$ and $G_{31}(s)$, respectively. If $u_1$ affects $y_3$ to such an extent that it poses a problem, this can ideally be resolved by using the feedforward

$$u_3 = u_3^* - \frac{G_{31}(s)}{G_{33}(s)} u_1, \tag{5}$$

where $u_3^*$ is the control signal from the decentralized controller and we assume $\frac{G_{31}(s)}{G_{33}(s)}$ is stable and proper. If we implement this feedforward loop we will have removed the direct effect of $u_1$ on $y_3$. However, there are other consequences of this implementation since the change of $u_3$ will also affect $y_1$ and $y_2$. If these interactions are significant the feedforward loop might do more harm than good. Having this in mind, we examine how the IM can be used to determine when feedforward might be appropriate.

Consider an interaction matrix

$$\Gamma = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1N} \\ \vdots & \ddots & \vdots \\ \gamma_{N1} & \cdots & \gamma_{NN} \end{bmatrix}. \tag{6}$$

First we choose the elements for the decentralized pairing as described previously and assume, without loss of generality, that the pairing elements are on the diagonal. After this, we look in the interaction matrix for large elements not yet selected for pairing. The current method for determining feedforward is simply to use the largest elements not selected for pairing (Conley and Salgado, 2000). However, doing so means that other potential interactions are not taken into account. For example, assume that $\gamma_{N1}$ is a large value and thus $u_1$ is a potential candidate for feedforward. However, as described in the 3 by 3 example above, this will impact $u_N$, which will not only impact $y_N$, but also the other outputs. A gauge of the size of this impact is $\sum_{i=1}^{N-1} \gamma_{iN}$. If these values are very large then the IM indicates that adding the described feedforward on $u_1$ is unwise. To determine the use of feedforward in the general case we therefore create a modified interaction matrix $\Gamma^*$, whose elements are defined by

$$\gamma_{ij}^* = \gamma_{ij} - \rho \sum_{\substack{k=1 \\ k \neq i}}^{N} \gamma_{ki}, \tag{7}$$

where $\rho$ is a tuning parameter. With this new IM, the largest elements where $i \neq j$ are chosen for feedforward until the sum of elements in $\Gamma^*$ chosen both for control and feedforward is larger than 0.7, a rule of thumb for gramian based measures (Salgado and Conley (2004)). However, as feedforward increases controller complexity it is only implemented if it seems likely that it will have a positive impact. This is determined by checking if $\gamma^*_{ij} > 0$ in which case feedforward is considered appropriate, and otherwise it is not implemented. Further precautions also have to be taken to avoid implementing an unstable or non-proper feedforward block. Note that if $\rho = 0$ the largest elements of the IM are chosen without taking into account other interactions.

## 1.3. Delays

Continuous time gramian based measures struggle to appropriately deal with delays. This occurs because the $\Sigma_2$ method is completely unaffected by delays, while the Hankel singular values of systems with delays are problematic, as continuous time systems with delays are of infinite order. One solution for this is to discretize the system and implement the methods on the discrete time system as discussed in Salgado and Conley (2004). A pairing found on the discrete time system can then be implemented on the continuous time system. Note that when implementing decoupling feedforward on systems with delays some decouplings may not be possible as they would be non-causal.

## 1.4. Scaling of the IMs

An issue with the gramian based methods is that their interaction matrices are affected by the scaling of the inputs and outputs such that different scalings may yield different results. Generally, this is resolved by scaling the inputs and outputs from 0 to 1, setting zero to the lowest value they are likely to reach and 1 to the highest value (Salgado and Conley (2004)). However, this scaling is at times insufficient, and we will present a few ways in which the IMs can be rescaled for improved results, as discussed in Bengtsson et al. (2020).

### 1.4.1. Row or Column scaling

Each column in the IM corresponds to the interactions from one input, while each row corresponds to the interactions affecting one output. If one column contains significantly less interaction than the other columns (as may be the case if one input is relatively poorly suited for control), little importance will be given to the decision of which output should be controlled by this input. This may lead to a poor input-output pairing. One way to resolve this would be to normalize the columns, that is to divide the elements in each column of the IM by the corresponding column sum. This will ensure that when conducting the pairing algorithm, equal importance is given to each input. In the new IM the scaled elements would become

$$[\Gamma_c]_{ij} = \frac{[\Gamma]_{ij}}{\sum_{k=1}^{N}[\Gamma]_{kj}}, \tag{8}$$

where $\Gamma_c$ is an interaction matrix with normalized columns. If we instead wish to ensure that equal importance is given to each output, we can instead normalize the rows, which gives an interaction measure defined by

$$[\Gamma_r]_{ij} = \frac{[\Gamma]_{ij}}{\sum_{k=1}^{N}[\Gamma]_{ik}}. \tag{9}$$

### 1.4.2. Choosing between row and column scaling

It may be difficult to determine if it is preferable to scale by rows or columns. Therefore we have proposed an approach to scaling that tries to determine which is the most appropriate for a given IM. In this approach the column sums and row sums were first calculated. If the smallest sum is a row sum, then the rows are scaled, and otherwise the columns are scaled. This approach will be referred to as row/column scaling.

### 1.4.3. Sinkhorn-Knopp algorithm

By scaling the columns or rows we can guarantee that equal importance is given to either each input or each output when determining a pairing. If we, however, wish to have both the columns and rows scaled we can use the Sinkhorn-Knopp (SK) algorithm. This algorithm combines row and column scaling by alternating between normalizing the rows and normalizing the columns. In cases where the matrix can be made to have positive elements on the diagonal (as is always the case with gramian based measures) this algorithm is guaranteed to converge to a matrix that will have both rows and columns normalized (Sinkhorn and Knopp (1967)). In Bengtsson et al. (2020) it is shown that when designing decentralized controllers using Sinkhorn-Knopp scaling generally yields the best result.

Scaling the IMs with the Sinkhorn-Knopp algorithm has the additional benefit of removing the impact of input and output scaling on the IM altogether. Using the Sinkhorn-Knopp algorithm to scale the system will yield the same IM, regardless of what the original scaling of the system was.

## 2. Methods of analysis

To properly evaluate the feedforward methods we have used the MIMO model generator described in Bengtsson and Wik (2017) to generate 200 continuous linear MIMO-systems using the settings given in Appendix 1. We then generated sparse control configurations using the different gramian based measures, using all the described methods of scaling, and with values of $\rho$ ranging from 0 to 5. For each control configuration lambda-tuned PI feedback controllers were implemented, along with decoupling feedforward. Time delays were dealt with using second order Padé approximation, and if implementing feedforward would require a non-proper transfer function it would simply not be implemented. The systems were then tested for reference steps as well as for disturbances. The first type of disturbance tested was a step disturbance placed on each of the inputs, so a disturbance $d_1$ on $u_1$ would for example give:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1N}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2N}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1}(s) & G_{N2}(s) & \cdots & G_{NN}(s) \end{bmatrix} \begin{bmatrix} u_1 + d_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \tag{10}$$

Such disturbances will henceforth be referred to as input disturbances and they were tested separately for each input.

The second type of disturbance that was tested was a step disturbance on $u$ for only one of the outputs, for example

$$\begin{aligned} y_1 &= G_{11}(s)(u_1 + d_1) + G_{12}(s)u_2 + ... + G_{1N}(s)u_N \\ y_2 &= G_{21}(s)u_1 + G_{22}(s)u_2 + ... + G_{2N}(s)u_N \\ &\vdots \end{aligned} \tag{11}$$

where the disturbance, henceforth referred to as an individual disturbance, is only on the transfer function from $u_1$ to $y_1$, in this case. This disturbance was tested on each transfer function in the TFM. Once the tests were conducted we evaluated how well the different scaling methods worked for reference steps and for disturbances. For the evaluation a cost was defined as being the squared deviation from the reference for 2000 time units after the reference step, or the input disturbance. Having calculated this cost for each IM, each IM is given a score defined as

$$S = \frac{c_{min}}{c}, \tag{12}$$

where $S$ is the score of the IM, $c$ is its cost, and $c_{min}$ is the lowest cost of all IMs for the system. The score was set to zero if the control scheme yielded unstable results. This measure normalizes the scores for each system to be between 0 and 1, ensuring that the results on different systems are comparable.

## 3. Results

The average score for each method as a function of $\rho$, when using the HIIA, is shown in Figure 1. Similiar results where found when using the PM and $\Sigma_2$ methods and are therefore omitted. The first thing that one may observe from the figure is that with input disturbances, adding feedforward decreases the average score. This is not entirely surprising and can be understood by examining a simple 2 by 2 system

$$\begin{aligned} y_1 &= G_{11}(u_1 + d_1) + G_{12}u_2 \\ y_2 &= G_{21}(u_1 + d_1) + G_{22}u_2 \end{aligned} \tag{13}$$

Now a solution to counteract the disturbance $d_1$ is to set $u_1 = -d_1$, which will remove the impact of the disturbance on $y_1$ and $y_2$ . Now if we add feedforward to $u_2$ to remove the impact of $u_1$, i.e.

$$u_2 = u_2^* - G_{22}^{-1}G_{21}u_1 \tag{14}$$

the new system becomes

$$\begin{aligned} y_1 &= G_{11}(u_1 + d_1) + G_{12}u_2 \\ y_2 &= G_{21}d_1 + G_{22}u_2^* \end{aligned} \tag{15}$$

Now, setting $u_1 = -d_1$ only eliminates the effect of $d_1$ on $y_1$, but not on $y_2$. So this indicates that implementing feedforward for these kind of disturbances can be unwise, which is supported by the tests.

For the other cases, however, feedforward yields improved results. One can also see that $\rho = 0$, which represents the usual way of finding elements for feedforward, does not yield the best results. This is Thus, an improvement can indeed be found by using the proposed modified interaction matrix for finding elements for feedforward.

What can also be seen is that although SK was found to be superior as a scaling method for regular input output pairing (Bengtsson et al. (2020)), other scaling methods score better for feedforward determination, especially for aggresive feedforward schemes with $\rho = 0$, in which case the SK-algorithm produces a worse result then with no feedforward for individual disturbances .

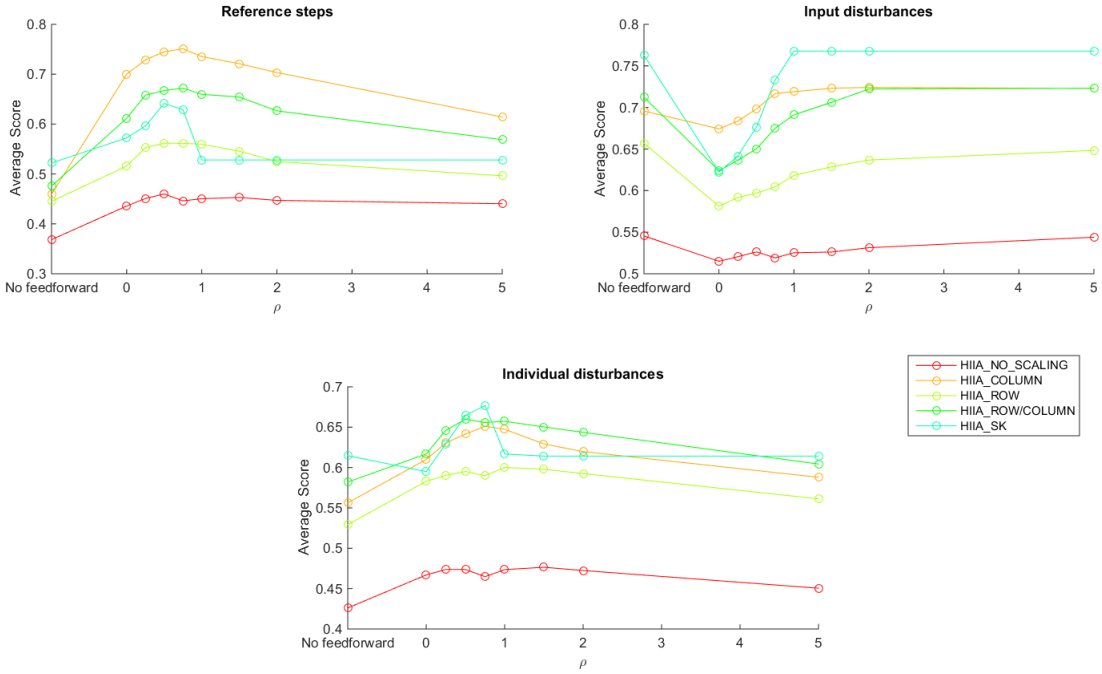


Figure 1: Scores for different $\rho$ and for different reference and disturbances steps for pairings derived using the HIIA

## 4. Adapting the Sinkhorn-Knopp scaling algorithm for sparse control

As shown, the Sinkhorn-Knopp scaled systems did not result in as large improvement as the other scaling methods when determining feedforward. A reason for this is that using Sinkhorn-Knopp scaling may remove information from the IM useful for the design of sparse controllers. Take, for example, a 3 by 3 system,

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & G_{13}(s) \\ 0 & G_{22}(s) & G_{23}(s) \\ 0 & G_{32}(s) & G_{33}(s) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \quad (16)$$

If we use one of the gramian based measures to find an IM it will result in the following:

$$\Gamma = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ 0 & \gamma_{22} & \gamma_{23} \\ 0 & \gamma_{32} & \gamma_{33} \end{bmatrix}. \quad (17)$$

Scaling this with the Sinkhorn-Knopp algorithm will ensure that both the rows and columns are normalized, resulting in the following IM:

$$\Gamma_{SK} = \begin{bmatrix} \gamma_{11} & 0 & 0 \\ 0 & \gamma_{22}^* & \gamma_{23}^* \\ 0 & \gamma_{32}^* & \gamma_{33}^* \end{bmatrix}. \quad (18)$$

As can be seen the elements $\gamma_{12}$ and $\gamma_{13}$ in this interaction measure become zero. This means that if the Sinkhorn-Knopp scaled IM is used to find elements for feedforward it will disregard the possibility of adding feedforward to $u_1$. However, this is not desirable as $\gamma_{12}$ or $\gamma_{13}$ may indicate that feedforward on $u_1$ is appropriate. Furthermore with aggressive feedforward strategies, disregarding viable candidates for feedforward may cause the scheme to instead chose poor candidates for feedforward, especially if nothing is done to check the viability of the candidates (as is the case if $\rho = 0$). To resolve this we propose a hybrid method, where Sinkhorn-Knopp scaling is used to design a decentralized controller, and one of the other scaling methods is then used to determine what signals are appropriate for feedforward.

## 5. Evaluation of hybrid methods

We tested the proposed measure, using the Sinkhorn-Knopp algorithm to design a decentralized controller

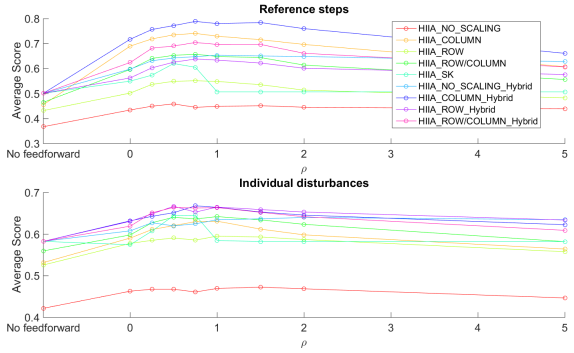


Figure 2: Average scores for different $\rho$ and for difference reference steps and disturbances

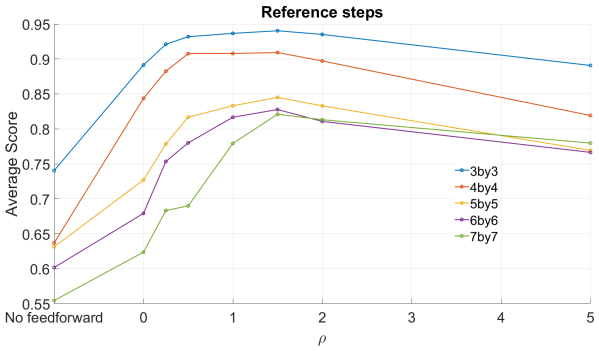


Figure 3: Scores for different $\rho$ and for reference steps using hybrid column scaling for systems with a different number of inputs and outputs

and then using one of the other scaling methods (or no scaling) to find elements for feedforward. The same test was carried out as before, except that we did not test input disturbances here, as we have already established that feedforward does not yield improvements in this case. The result is shown in Figure 2.

As can be seen, using Sinkhorn-Knopp scaling to design decentralized PI controllers, and then column scaling to find feedforward elements yields the best result, at least for reference steps. For individual disturbances, most of the hybrid methods seem to yield comparable results, with no method of scaling clearly preferable when finding elements for feedforward.

It can also be seen that, generally, values of $\rho$ ranging from $0.5 - 1$ seemed to yield the best results when implementing feedforward. To investigate if this is affected by system size, we generate TFMs of size 3 by 3 to 7 by 7 (450 of each type). We use the same MIMO generator settings as before, except with the changes specified in Table 1 (necessary due to the different system size). We then test the different systems with reference steps, using hybrid column scaling. From this we get the results presented in Figure 3 in which we can see that for increasing system sizes, larger values of $\rho$ are generally required.

Table 1: New settings when investigating different $\rho$ for differently sized systems

| | Inputs affecting each output | |
|---|---|---|
| System | Min | Max |
| 3by3 | 3 | 3 |
| 4by4 | 3 | 4 |
| 5by5 | 3 | 5 |
| 6by6 | 3 | 5 |
| 7by7 | 3 | 5 |

## 6. Comparisons with other methods

A well known method for designing sparse controllers is the RNGA-RGA-NI method by Shen et al. (2010). They also suggest a method for sparse controller design based on equivalent transfer functions (ETF). For comparison with our method we will examine the FS configuration of a Heat-integrated distillation column (Chiang and Luyben, 1988) .

$$G(s) = \begin{bmatrix} \frac{4.45}{(14s+1)(4s+1)} & \frac{-7.4}{(16s+1)(4s+1)} & 0 & \frac{0.35}{(25.7s+1)(2s+1)} \\ \frac{17-3e^{-0.9s}}{(17s+1)(0.5s+1)} & \frac{-41}{(21s+1)(s+1)} & 0 & \frac{9.2e^{-0.3s}}{20s+1} \\ \frac{0.22e^{-1.2s}}{(17.5s+1)(4s+1)} & \frac{-4.66}{(13s+1)(4s+1)} & \frac{3.6}{(13s+1)(4s+1)} & \frac{0.042(78s+1)}{(21s+1)(11.6s+1)(3s+1)} \\ \frac{1.82e^{-s}}{(21s+1)(s+1)} & \frac{-34.5}{(20s+1)(s+1)} & \frac{12.2e^{-0.9s}}{(18.85s+1)(s+1)} & \frac{-6.92e^{-0.6s}}{(15s+1)(4s+1)} \end{bmatrix} \tag{19}$$

We test three control configuration selections and control design strategies. The first is the combination of SK and column scaled HIIA control configuration method with $\rho = 0.5$ and with lambda tuned controllers. This is compared with the configuration recommended by the RNGA-RGA-NI, evaluated both using lambda tuning with decoupling, as well as with the controller design method for sparse control based on ETFs, as described in Shen et al. (2010). The different methods are evaluated in the same way as before, testing them for reference steps and individual disturbances. The results are presented in Table 2.

As can be seen from Table 2, the modified HIIA configuration outperforms the RNGA configurations. The improvement for reference steps is quite modest, since the RNGA configuration along with lambda tuning yielded nearly the same cost. For input disturbances, though, the improvements are more significant.

If we expand this comparision and use the MIMO generator to test 150 randomly generated systems with the same specifications as before we get the results shown in Table 3. Now, the RNGA-RGA-NI method

Table 2: A comparison of the costs for different control configuration methods on the Heat integrated distillation column

| Control configuration | Reference step | Input disturbance |
|---|---|---|
| Decentralized controller | 323 | 1919 |
| HIIA SK/column scaling | 184 | 1805 |
| RNGA-Lambda | 192 | 2897 |
| RNGA-ETF | 685 | 14011 |

is specially designed on first order plus dead time systems in mind. If we test only those systems we get the results in Table 4.

As can be seen from Table 3 and Table 4 the proposed hybrid method outperforms the other methods considerably for randomly generated systems. This illustrates the main strength of this method, namely that it reliably gives stable control schemes for a wide variety of systems. This results in a very good score when testing on a large number of systems, even though for some specific systems other methods may yield better results.

Table 3: Average score of the solutions of different control configuration methods for 150 randomly generated systems with specifications as specified by Table 5.

| Control configuration | Reference step | Input disturbance |
|---|---|---|
| HIIA SK/column scaling | 0.78 | 0.81 |
| RNGA-Lambda | 0.48 | 0.66 |
| RNGA-ETF | 0.26 | 0.24 |

Table 4: Average score of the solutions of different control configuration methods for 150 randomly generated first order and dead time systems

| Control configuration | Reference step | Input disturbance |
|---|---|---|
| HIIA SK/column scaling | 0.91 | 0.86 |
| RNGA-Lambda | 0.67 | 0.73 |
| RNGA-ETF | 0.32 | 0.29 |

## 7. Conclusion

We have proposed a new method for how to determine feedforward control based on gramian based interaction measures. More specifically, we have examined feedforward implementation using the HIIA, PM and $\Sigma_2$ interaction measures, and found that the best results were found when using Sinkhorn-Knopp scaling to find the decentralized control scheme, and then use column scaling to find elements for feedforward.

Furthermore, we demonstrated the need to take into account the entire interaction matrix when choosing elements for feedforward, not only choosing the element representing the largest interaction. This was done using a factor $\rho$, and in general, values of $\rho$ ranging from $0.5 - 1.5$ seemed to yield the best results when implementing feedforward for 5 by 5 systems with the specified specifications, while increasing the number of inputs and outputs necessitates a larger $\rho$ to acquire good results for reference following.

The proposed method was compared to other methods of sparse control design, and it was found to do well in comparison, both for a model of a distillation column and for randomly generated systems.

## References

Bengtsson, F. and Wik, T. A multiple input, multiple output model generator. Technical report, Department of Signals and Systems, Chalmers University of Technology, 2017. URL https://research.chalmers.se/publication/253490.

Bengtsson, F., Wik, T., and Svensson, E. Resolving issues of scaling for gramian based input-output pairing methods. *International Journal of Control*, 2020. doi:10.1080/00207179.2020.1812725.

Birk, W. and Medvedev, A. A note on gramian-based interaction measures. In *Proceedings of European Control Conference (ECC)*. pages 2625–2630, 2003. doi:10.23919/ECC.2003.7086437.

Chiang, T. P. and Luyben, W. L. Comparison of the dynamic performances of three heat-integrated distillation configurations. *Industrial & Engineering Chemistry Research*, 1988. 27(1):99–104. doi:10.1021/ie00073a019.

Conley, A. and Salgado, M. E. Gramian based interaction measure. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 5. pages 5020–5022, 2000. doi:10.1109/CDC.2001.914730.

Fatehi, A. Automatic pairing of large scale MIMO plants using normalised RGA. *International Journal of Modelling, Identification and Control*, 2011. 14(1-2):37–45. doi:10.1504/IJMIC.2011.042338.

Khaki-Sedigh, A. and Moaveni, B. *Control configuration selection for multivariable plants*. Springer, 2009. doi:10.1007/978-3-642-03193-9.

Salgado, M. E. and Conley, A. MIMO interaction measure and controller structure selection. *International Journal of Control*, 2004. 77(4):367–383. doi:10.1080/00207170420001976 31.

Shen, Y., Cai, W.-J., and Li, S. Multivariable process control: Decentralized, decoupling, or sparse? *Industrial & Engineering Chemistry Research*, 2010. 49(2):761–771. doi:10.1021/ie901453z.

Sinkhorn, R. and Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 1967. 21(2):343–348. doi:10.2140/pjm.1967.21.343.

Wittenmark, B. and Salgado, M. E. Hankel-norm based interaction measure for input-output pairing. In *Proceedings of the 2002 IFAC World Congress*, volume 139. pages 429–434, 2002. doi:10.3182/20020721-6-ES-1901.01625.

# A. MIMO model generator settings

Table 5: Table showing the MIMO model generator (Bengtsson and Wik (2017)) settings

| Parameter | Value |
|---|---|
| **Size** | |
| Number of inputs | 5 |
| Number of outputs | 5 |
| Minimum number of inputs affecting each output | 3 |
| Maximum number of inputs affecting each output | 5 |
| Minimum transfer function order | 1 |
| Maximum transfer function order | 3 |
| Minimum relative degree | 1 |
| Maximum relative degree | 3 |
| **Dynamics** | |
| Maximum static gain | 1000 |
| Minimum pole time constant | 1 |
| Maximum pole time constant | 10 |
| Minimum damping for complex poles | 0.1 |
| Distinct time constants | false |
| Basing zeros time constants on poles when possible | true |
| Maximum overshoot percentage | 10 |
| Maximum undershoot percentage | 25 |
| Tolerance when determining overshoot/undershoot | 0.01 |
| Factor used to determine minimum time constant | 100 |
| **Poles and Zeros** | |
| Maximum number of unstable poles | 0 |
| Minimum number of unstable poles | 0 |
| Maximum number of purely imaginary pole pairs | 0 |
| Minimum number of purely imaginary pole pairs | 0 |
| Percentage of unstable poles which are complex | 0 |
| Percentage of stable poles which are complex | 20 |
| Percentage of transfer functions with single integrators | 0 |
| Percentage of transfer functions with double integrators | 0 |
| Percentage of transfer functions with derivatives | 0 |
| Maximum number of non-minimum phase zeros | 4 |
| Minimum number of non-minimum phase zeros | 0 |
| **Delay** | |
| Percentage of transfer functions with delay | 10 |
| Minimum Delay | 0 |
| Maximum Delay | 0.5 |
| Padé approximation order | 2 |

CC BY