**MIMO control systems**

Definition **:**

It is the system that have more than one input and more than out put, where each output of the system depends on the inputs of the system ,both inputs are processed differently to form each output, as shown in following diagram.

Diagram

Description automatically generated

Comparison between MIMO & SISO systems :

|  |  |
| --- | --- |
| **MIMO SYSTEM** | **SISO SYSTEM** |
| Multi input multi output | Single input single output |
| Diagram  Description automatically generated | Diagram  Description automatically generated |
| (MIMO) systems describe processes with more than one input and more than one output which require multiple control loops. These systems can be complicated through loop interactions that result in variables with unexpected effects. | SISO control schemes are just one type of control scheme that engineers in industry use to control their process |
| This type of control system is complicated , These systems can be complicated through loop interactions that result in variables with unexpected effects. Decoupling the variables of that system will improve the control of that process. | This type of control system is simpler.  These systems use data/input from one sensor to control one output. These are the simplest to design since they correspond one sensor to one actuator. For example, temperature (TC) is used to control the valve state of v1 through a PID controller. |
| Examples of MIMO systems include heat exchangers, chemical reactors, and distillation columns | Examples of SISO control systems include   fan speed control, air conditioner temperature control |

* When to use MIMO ?

We use it when the output that required to be achieved depends upon many inputs so we need to choose suitable changes on input but carefully cause these inputs changing may affect anther output that also depends upon them .

RGA Controller

INTRO**:**One method for designing and analyzing a MIMO control scheme for a process in steady state is with a Relative Gain Array (RGA). RGA is useful for MIMO systems that can be decoupled (see the article about determining if a system can be decoupled). For systems that cannot be decoupled, model predictive control or neural networks are better choices of analysis tool than RGA.

A good MIMO control scheme for a system that can be decoupled is one that can control a process variable without greatly affecting the other process variables. It must also be stable with respect to dynamic situations, load changes, and random disturbances. The RGA provides a quantitative approach to the analysis of the interactions between the controls and the output, and thus provides a method of pairing manipulated and controlled variables to generate a control scheme.

DEFINATION:Relative Gain Array is an analytical tool used to determine the optimal input-output variable pairings for a multi-input-multi-output (MIMO) system. In other words, the RGA is a normalized form of the gain matrix that describes the impact of each control variable on the output, relative to each control variable's impact on other variables. The process interaction of open-loop and closed-loop control systems are measured for all possible input-output variable pairings. A ratio of this open-loop 'gain' to this closed-loop 'gain' is determined and the results are displayed in a matrix.

Text

Description automatically generatedThe array will be a matrix with one column for each input variable and one row for each output variable in the MIMO system. This format allows a process engineer to easily compare the relative gains associated with each input-output variable pair, and ultimately to match the input and output variables that have the biggest effect on each other while also minimizing undesired side effects.

The results of RGA: The closer the values in the RGA are to 1 the more decoupled the system is

* The maximum value in each row of the RGA determines which variables should be coupled or linked
* Also each row and each column should sum to 1

There are two main ways to calculate RGA:

1. Experimentally determine the effect of input variables on the output variables, then compile the results into an RGA matrix.( This method will generally yield the most accurate RGA matrix of the system because it is based on actual data taken from the operating control system)
2. Use a steady-state gain matrix to calculate the RGA matrix.( can be used with no experimental data. If there is no process model available, some experimental data must be taken)

**Method 1: Calculating RGA with Experiments**

This method of calculating the RGA can be used when it is possible to run experiments on each of the input-output pairings. Below is a step-by-step explanation of how to experimentally compile the RGA for a simple MIMO system.

The simplest MIMO system is one that has two inputs and two outputs. Remember that by definition, a change in one of the inputs in a MIMO system will change *both* of the outputs. This system can be expressed mathematically as written below.

y1=a11m1+a12m2y1=a11m1+a12m2

y2=a21m1+a22m2y2=a21m1+a22m2

Where *yi* is the output for loop *i*, the *m* variables are the inputs for each loop, and the *a* variables are the transfer functions. It also helps to see this system as a control diagram, as shown here.

Diagram

Description automatically generated

Clearly, both *y*1 and *y*2 are affected by both inputs (*m*1 and *m*2). Because of this, there are two choices for control. *y*1 could be controlled by *m*1, or *y*1 could be controlled by *m*2. *y*2 would then be controlled by the input that is not controlling *y*1. For a MIMO system with *n* input variables, there are *n*! possible control configurations. The question is: Which configuration will give you the best control? The answer can be determined by finding the relative gain array (RGA). There are a few ways to determine the RGA, one of which is by doing two experiments, repeated for every possible configuration. The RGA can be assembled from the data collected. These experiments are explained below using the above drawing and the configuration where *m*1 controls *y*1.

**Experiment 1**

For the first experiment, the objective is to observe the effect of *m*1 on *y*1. To do this, *all* of the loops must be open. In other words, the feedback loop is removed and the system is run manually with no control. This configuration is shown below.

A diagram of a house

Description automatically generated with low confidence

Now, since the system is under manual control, it is possible to introduce a step change Δ*m*1 while keeping *m*2 constant. Because *m*2 is held constant, the change in *y*1, Δ*y*1, can be attributed entirely to *m*1. Although a change in *m*1 will also affect *y*2, it is the relationship between *m*1 and *y*1 that must be observed at this point. Then, the gain of *y*1 with respect to *m*1 with all loops open is defined as *g*11. The calculation of *g*11 is shown below.

g11=Δy1( all - loops -open )Δm1( all − loops -open )g11=Δy1( all - loops -open )Δm1( all − loops -open )

The final objective of experiment one is obtaining the value *gij* where the controlled variable *i* is controlled by manipulated variable *j*.

**Experiment 2**

The goal of experiment two is to determine the effect of *m*2 on *y*1. To do this, loop 1 must remain open, but all other loops remain closed. This configuration is shown below.

Diagram

Description automatically generated

Now, the *same* step change that was introduced to *m*1 in experiment one must be introduced again. The difference is, now loop 2 is in *perfect control* which means that when *m*1 changes, it affects *y*2, so the control loop on loop two will adjust *m*2 in order to compensate and keep *y*2 constant. When *m*2 changes it, in turn, affects *y*1. It is this change that is being observed. The amount *y*1 changes is Δ*y*1 and the amount *m*1 was stepped up is Δ*m*1. Now:

g∗11=Δy1( only − loop − one − open )Δm1( only − loop − one − open )g11∗=Δy1( only − loop − one − open )Δm1( only − loop − one − open )

The objective of experiment two is to obtain this value of g∗ijgij∗ where controlled variable *i* is being controlled by manipulated *j*.

**Compiling the Array**

Once the experiments are run for every possible configuration, the results can be used to compile the relative gain array (RGA). To do this, we first have to find the *relative gain*. The relative gain is the ratio of g_{ij}to g^*_{ij}and is represented by \lambda_{ij}.

**Method 2: Calculating RGA with Steady-State Gain Matrix**

Sometimes it is not convenient or possible to perform the experiments described above on every control pairing possibility in the system. If that is the case, a *steady-state gain matrix* can be used to determine the RGA. If a process model is available, the steady-state gain matrix can be calculated from the model equations. If a process model is not available, the steady-state gain matrix can be calculated from experimental data (fewer experiments are required than when the RGA is calculated directly from experimental data as above). Once the steady-state gain matrix is calculated, it can be used to compute the RGA. Beware that a RGA has to have the same number of controlled variables and manipulated variables (same number of inputs and outputs) to be in an *m* x *m* matrix form.

**Process Model Available**

If a process model is available, the steady-state gain matrix relates the manipulated variables to the controlled variables according to the following equation:

y=Gmy=Gm

where yy is a vector of the controlled variables:

y=⎡⎣⎢⎢⎢⎢y1y2⋮yn⎤⎦⎥⎥⎥⎥y=[y1y2⋮yn]

mmis a vector of the manipulated variables:

m=⎡⎣⎢⎢⎢⎢m1m2⋮mn⎤⎦⎥⎥⎥⎥m=[m1m2⋮mn]

and *G* is the steady-state gain matrix:

G=⎡⎣⎢⎢⎢⎢g11g21⋮gn1g12g22gn2⋯⋯⋯g1ng2ngnn⎤⎦⎥⎥⎥⎥G=[g11g12⋯g1ng21g22⋯g2n⋮gn1gn2⋯gnn]

The values of *gij* are calculated by taking partial derivatives of the equations governing the controlled variables. Specifically:

gij=∂yi∂mjgij=∂yi∂mj

During differentiation, hold constant all manipulated variables other than *mj*. To fill in matrix *G* with numerical values, simply substitute the appropriate steady state values for the process into the expressions derived from differentiation.

Now that the steady-state gain matrix has been found, the RGA can be calculated with just a little bit more math. However, now that you have the steady state gain matrix, this might be a good time to determine if your system can even be decoupled! See the article on singular value decomposition for methods to check this. Assuming SVD gives a condition number of less than about 50, we can proceed to calculation of the RGA. First, define a matrix *R* to be the transpose of the inverse of matrix *G*:

R=(G−1)TR=(G−1)T

The calculation of the **inverse** of a matrix is quite complicated for anything larger than a 2x2 matrix. Fortunately Mathematica will do this with the following command for a square matrix *M*:

Inverse[*M*]

The **transpose** of a matrix is when the rows become columns and the columns become rows. For a square matrix, this just means reflecting across the diagonal running from the top left to the bottom right corner of the matrix. The **transpose** can be found in Mathematica with the following command:

Transpose[*M*]

The RGA can now be obtained one element at a time according to this equation:

λij=gijrijλij=gijrij

Note that this is not your usual matrix multiplication! Here you multiply corresponding elements of the *G* and *R* matrices to get the corresponding element of the RGA. This is the type of multiplication Mathematica does with the standard multiplication operator.

See Example 1 below for a worked-out RGA calculation using this approach.

**Process Model Not Available**

In case there is no process model available and it is not feasible to determine the RGA by carrying out both experiments necessary for full experimental determination, it still may be possible to develop a steady-state gain matrix experimentally which can then be used to derive the RGA. In this case, just carry out experiment 1 as described above to determine the elements of the steady-state gain matrix:

gij=ΔyiΔmjgij=ΔyiΔmj

with all loops open.

Each element of the steady-state gain matrix can be determined this way, and then the RGA can be calculated from the steady-state gain matrix as shown in the previous section.

**Interpreting the RGA**

There are some important properties and guidelines in understanding and analyzing the RGA, and what the different values of the RGA mean:

1) **All elements of the RGA across any row, or down any column will sum up to one:**

∑i=1nλij=∑j=1nλij=1∑i=1nλij=∑j=1nλij=1

This makes calculating the RGA easier because:

* in 2X2 case, only 1 element must be calculated to determine all elements
* in 3X3 case, only 4 elements must be calculated to determine all elements

and so on.

2) The λij calculated from steady-state matrix is dimensionless and unaffected by scaling.

3) Each of the rows in the RGA represent one of the outputs. Each of the columns represent a manipulated variable.

* **If λij = 0:** The manipulated variable (mj) will have no effect on the output or the controlled variable (yi).
* **If λij = 1:** The manipulated variable mj affects the output yi without any interaction from the other control loops in the system. From the definition of λij, this implies that the gain loop with all loops open is equal to the gain loop with all other loops closed, ie: g11 = g\*11.
* **If λij < 0:** The system will be unstable whenever mj is paired with yi, and the opposite response in the actual system may occur if other loops are opened in the system.
* **If 0 < λij < 1:** This implies that other control loops (mj- yi) are interacting with the manipulated and controlled variable control loop.

Three different relationships based on λ=0.5 imply different interpretations of pairing and the RGA:

* **If λij = 0.5:** The control pairing effect is equal to the retaliatory effect of other loops.
* **If λij < 0.5:**The other control loops are influencing the control pair, and the influence from the other control loops are greater than the influence from the control pair.
* **If λij > 0.5:** This means that the control pair has a greater influence on the system than the other control loops.
* **If λij >1:**The open-loop gain of the control pair is greater than the gain with all other loops closed, ie: g11 > g\*11. The positive value of RGA indicates that the control pair is dominant in the system, but the other loops are still affecting the control pair in the opposite direction. The higher the value of λij, the more correctional effects the other control loops have on the pair.

| λij | Possible Pairing |
| --- | --- |
| λij = 0 | Avoid pairing mj with yi |
| λij = 1 | Pair mj with yi |
| λij < 0 | Avoid pairing mj with yi |
| λij = or < 0.5 | Avoid pairing mj with yi |
| λij >1 | Pair mj with yi |

**ANALYSIS OF RGA:**

The **Niederlinski Index** (NI) is a calculation used to analyze the stability of the control loop pairings using the result of the RGA, evaluated at Steady State:

NI=|G|∏ni=1giiNI=|G|∏i=1ngii

A negative NI value indicates instability in the control loop. For a 2 times2 matrix, a positive NI value indicates stability in the pairings, but this is not necessarily true for larger matrices! For matrices larger than 2 times2, a conclusion can only be drawn from a negative NI, which indicates instability. NI should not be used for systems that have time delays (dead time), as the NI stability predictions assume immediate feedback to the controller. However, the index can be adapted for systems with dead time, although that application is beyond the scope of this article.  
  
Here's an example NI calculation: given the steady state gain matrix

G=[−0.0020.0020.0010.003]G=[−0.0020.0010.0020.003]

the NI can be calculated from the following expression:

NI=(−0.002×0.003)−(0.001×0.002)−0.002×0.003=1.333NI=(−0.002×0.003)−(0.001×0.002)−0.002×0.003=1.333

Since this is a 2 times2 matrix, the positive value of the NI indicates stability in the control loop pairings.

**Optimizing a MIMO Control Scheme: Pairing Rules**

The goal of the RGA and NI analysis is to quantitatively determine the optimal variable pairing for a given process. Some basic rules to remember when attempting to obtain an optimal pairing of control loops in a system are:

* **Rule 1:** The **positive** RGA elements that are closest to 1.0 should have the corresponding manipulated and controlled variables paired. When the CN number is large, implying a less decoupled system, one should look for the max RGA elements.
* **Rule 2:** If the NI value is negative, the loop pairing for that control system configuration is unacceptable.