ALEX CINGARA
PharmEng Technology Inc,
Toronto, Ontario, Canada

SCIENTIFIC PAPER

511.52+517.9:512.64

TRANSFORMATION OF THE PROBLEM OF MULTIVARIABLE SYSTEM OF DIFERENTIAL EQUATIONS INTO THE PROBLEM OF ALGEBRAIC EQUATIONS WHICH RELATE THE OPEN AND CLOSED LOOP MATRICES AND DESIGN OF THE NEW ALGEBRAIC MIMO CONTROLLER USING THE ALGEBRAIC EQUATIONS

In this paper we are proposing the algebraization of original MIMO closed loop transfer functions by derivation of algebraic matrix equations which can be used for algebraic multivariable controller design instead the original equations. First, using closed loop transfer function with P–controller we derive two algebraic matrix equations. The open loop transfer function is: $\boldsymbol{G}(s) = (\boldsymbol{T}s + \boldsymbol{I})^{-1} \boldsymbol{K}$, *and closed loop is:* $\boldsymbol{G}_{CL}(s) = (\boldsymbol{T}_{CL}s + \boldsymbol{I})\, \boldsymbol{K}_{CL}$. *Those two equations are connected with two algebraic matrix equations: the first one is static equation* $\boldsymbol{K}_{CL} = (\boldsymbol{I} + \boldsymbol{KR})^{-1} \boldsymbol{KR}$ *which represents Closed Loop Static Matrix* $\boldsymbol{K}_{CL}$ *as a function of Open Loop Static Matrix K and Controller matrix R. The second equation is* $\boldsymbol{T}_{CL} = (\boldsymbol{I} + \boldsymbol{KR})^{-1} \boldsymbol{T}$, *represents the Closed Loop Dynamic Matrix as a function of Open Loop Static Matrix K, Open Loop Dynamic matrix T and Controller Matrix R. Those two algebraic formulas contain all the parameters of original differential equations and we can use them instead the differential equations to design the multivariable controller. Second, having two matrix formulas instead of one system of differential equations solves some problems (we don't have to solve differential equations) but creates some new problems. Those algebraic equations show functional relations between matrices only, there are no vectors, and matrices are variables and parameters. We can do only algebra of those equations; we don't know the general theory of those equations. One simple way to relate those equations with control is to define RGA for each matrix. Then those equations are showing how the RGAs of open–loop static, dynamic and controller are affecting the closed–loop static and dynamic RGA. Third, using the dynamic matrix formula we can algebraically calculate the controller gain which will make V matrix to have dynamic closed loop* $RGA(\boldsymbol{T}_{CL}) = 1$. *The controller algebraic formula is simple* $\boldsymbol{R} = \boldsymbol{K}^{-1} (\boldsymbol{T} - \boldsymbol{I})$. *This formula gives the proper structure for the controller, and it can be improved by multiplying the first formula by the scalar (r)* $\boldsymbol{R} = r^*\boldsymbol{K}^{-1} (\boldsymbol{T} - \boldsymbol{I})$. *The results are very good for* 2×2 *and* 4×4 *systems and they suggest that the algebraic design for any nxn system will work successfully as well.*

The characteristics of the multivariable linear systems are defined with the system of linear differential equations with constants coefficients. The Laplace transform converts differential equations into much simpler algebraic equations in the transform variable (s), simplifying the modeling and design.

But for the multivariable systems the Laplace transforms alone is not good enough. Those systems have new level of complexity and the new problem is that instead of scalar parameters in single–variable systems we have matrix parameters in multivariable systems.

Because of the matrix parameter problem we derived two algebraic matrix equations that connect the closed–loop matrices with open–loop matrices and controller and so transform the closed–loop Laplace transform MIMO problem to algebraic problem which is much simpler to use for design.

This new derivation is by its results similar to some kind of Second Multivariable Laplace Transform because it transforms the original given in Laplace transform matrices [functions of (s)] into Algebraic Matrix Equations, without any functions of (s).

Those equations express the static and dynamic matrices of closed–loop as a function of open–loop static and dynamic and controller matrix.

Author address: Alex Cingara, 172 Kennard Ave., North York, ON, M3H 4M7
Email: cingara@sympatico.ca
Paper received: February 21, 2008
Paper accepted: Marth 28, 2008

In the typical case we have five types of matrix parameters: open–loop static, open–loop dynamic, closed–loop static, closed–loop dynamic and controller matrix. The typical controller design will be: given the open–loop matrices and closed–loop output specifications, find the controller matrix that will make closed–loop matrices to have such values that the output specification will be as required. The design of such controller is not a simple problem.

The main concepts to relate the matrices with control problem are algebra, RGA and eigenvalues. Using algebra (Eq. (10)) we can calculate the P controller directly from the two equations. RGA is more appropriate for the static matrices but can be used for dynamic matrices also. Eigen–values are more appropriate for the dynamic matrices but they can be used for static matrices too. Both concepts are appropriate for internal matrix interactions for one matrix. In this paper we will use only algebra for design.

Both matrix equations show how the external interaction between different open–loop matrices affect the closed loop matrices. Combination of internal and external interaction results in closed loop matrices. The external interaction can be taken into account using Matlab.

Using only those two algebraic matrix equations with RGA dynamic specification we were able to algebraically design the multivariable proportional controller with good static and dynamic characteristics.

SINGLE VARIABLE SYSTEM DECOMPOSITION

Open–Loop Transfer Function Decomposition to Static and Dynamics Part

The single variable representation of first–order linear differential equations given in the form of transfer function is:

$$G(s) = (Ts + I)^{-1} K \quad (1)$$

The advantage of this transfer function type model is that the two numbers K and T have clearly defined physical meaning, K is gain (static parameter) and T is time constant (dynamic parameter).

This transfer function can be decomposed into static part, open loop gain = K and open–loop dynamic part = (Ts + I). The static part is simple number, but the dynamic part is function of complex variable s. We will analyze the details of dynamic part latter, but now let's just say that all the dynamic parameters of the system are given in the dynamic parameter which is time constant T. For this reason we will focus our attention to this parameter.

Closed Loop Transfer Function Model

The general solution of one closed–loop linear differential equations in the form of transfer function is:

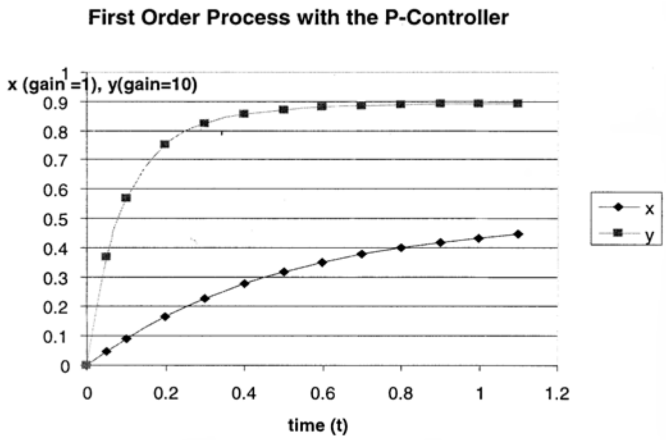

Figure 1. How the process (K = T = 1) unit response y(t) change with controller gain R, (R = 1 slower and R = 10 faster). This is the graph of the solution of differential equation unit step response.

$$G_{CL}(s) = (T_{CL}s + 1)^{-1}K_{CL} \quad (2)$$

where T_{CL} and K_{CL} are time constant and gain of closed loop.

We can see that the functional shape of the open–loop and closed–loop solution is the same, but the static and dynamic parameters are now different.

We will use this open loop–closed loop similarity to try to derive some understanding of the closed–loop solution, based on the understanding of the open–loop solution in the case of classical control approach.

In the Figure 1, we can see the effect of the increase of controller gain on the closed loop step response. The bigger the controller gain, the response is faster and closer to the unit value.

Two Closed–Loop Algebraic Equations and Algebraic Design

The closed loop transfer function matrices are connected with open loop transfer function matrices with two nonlinear Algebraic functions:

$$K_{CL} = (I + KR)^{-1}KR \quad (3)$$

$$T_{CL} = (I + KR)^{-1}T \quad (4)$$

Here we have a system of two nonlinear parametric equations that connect open–loop, controller and closed–loop characteristics.

The most important effect is to see how the change of proportional controller parameter R will change closed–loop static matrix K_{CL} (Eq. (3)) and closed loop dynamic matrix T_{CL} (Eq. (4)).

From Figure 2 we can see that with bigger controller gain R the closed loop static gain limit is converging to 1 and dynamic gain limit is converging to 0. This simple means that static gain value converges toward the wanted setup and dynamic of closed loop is getting faster.

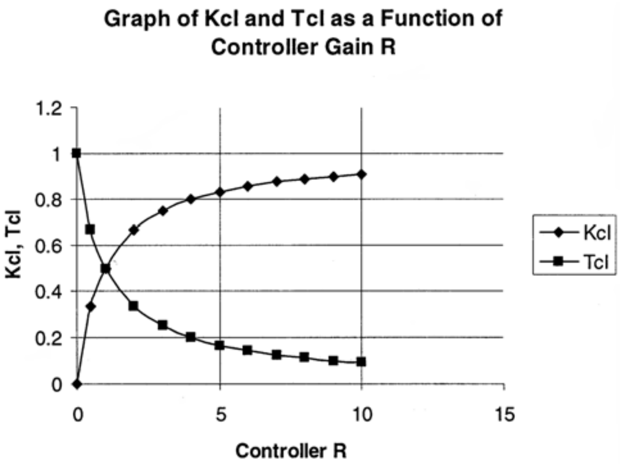

Figure 2. The graph is representing how Kcl = f(R) and Tcl = g(R). R is the adjustable parameter that is affecting static and dynamic of closed loop.

We claim now that Eqs. (3) and (4) presented in the Figure 2, contain all the information needed to design the closed loop controller R that we have in the Figure 1.

In other worlds we do not need to solve differential equation to know that bigger controller gain R will produce better response. Bigger R will make static closer to 1 and dynamic faster, closer to zero.

MULTIVARIABLE DECOMPOSITION AND ALGEBRAIC DESIGN

The Linear Feedback Design of MIMO controller is mathematically much more complicated than SISO system.

To solve the above multivariable problem in the simplest possible way we propose the following Direct Matrix Analysis Approach analog to the previous single variable problem. The obtained two algebraic matrix equations can be handled for control purposes using relative gain array analysis (RGA): The method is proposed by Bristol [1] and it is intensively used because it is mathematically simple and very efficient in practice: Skogestad [2], McAvoy [3], Luyben [4] and Stephanopoulos [5]. The method is improving and extending in the work of: Arranz [6], Vaes [7], Kairwala [8], Asmar [9,10], but we did not find any method that is similar to our method in this paper.

Transfer Function Decomposition of Open–Loop Static and Dynamics

The general multivariable solution of system of linear differential equations (open–loop) in the form of multivariable transfer function is:

$$\mathbf{G}(s) = (\mathbf{T}s + \mathbf{I})^{-1}\mathbf{K} \quad (5)$$

where **T** and **K** are (nxn) matrices. The advantage of this transfer matrix type model is that the two matrices **K** and **T** have clearly defined physical meaning, static matrix and dynamic matrix.

This transfer function can be decomposed into static part, open loop gain matrix = **K** and open–loop dynamic part = (**T**s + **I**). The static part is simple matrix, but the dynamic part is more complicated.

We can say that **T** is dynamic matrix and **K** is static matrix. The solutions of characteristic equation of dynamic matrix **T** are the time constants of the system.

Closed Loop Decomposition

The general multivariable solution of system of linear differential equations (closed–loop) in the form of MV transfer function is:

$$\mathbf{G}_{CL}(s) = (\mathbf{T}_{CL}s + \mathbf{I})^{-1}\mathbf{K}_{CL} \quad (6)$$

where $\mathbf{T}_{CL}$ and $\mathbf{K}_{CL}$ are (n×n) matrices.

We can see that the functional shape of the open–loop and closed–loop solution is the same, but the static matrices are different. We will use this open–closed similarity to try to derive some understanding of the closed–loop solution, based on the understanding of the open–loop solution just in the case of single–variable approach.

Focus on Two Algebraic Matrix Functions

The general multivariable solution of system of linear differential equations with multivariable proportional controller (**R**), depends completely on two nonlinear matrix functions, which are the same as Eqs. (3) and (4), but instead of scalars we have matrices.

$$\mathbf{K}_{CL} = (\mathbf{I} + \mathbf{KR})^{-1}\mathbf{KR} \quad (7)$$

$$\mathbf{T}_{CL} = (\mathbf{I} + \mathbf{KR})^{-1}\mathbf{T} \quad (8)$$

First static equation represents the interaction between the open–loop static and the controller and the second the interaction between the open–loop static, open–loop dynamic and controller. The first equation is basically showing that if you have the controller with big enough gain the static closed–loop gain will be close to the unit matrix. The second is showing static and dynamic interaction so we will use this equation to make both static and dynamic of closed loop equal to minimum.

Having two matrix formulas instead of one system of differential equations solves some problems (Diff. Eq.) but creates some new problems. The new problem is that those equations show functional relations only between matrices, here matrices are variables and parameters. We can do only algebra of those equations; we don't know the general theory of those equations.

One concept that can help us with this new type of equations is the relative gain array, RGA (Skogestad [2] Luyben [4], Stephanopoulos [5] and McAvoy [3]), and the other one is eigenvalues. The first one is more important for the static equation and the second one for the dynamic equation. But to keep the paper simple let's use only the RGA for both equations.

If the Static Open–Loop Relative Gain is close to 1 our two multivariable static curve will be close to one static single–variable curve from Figure 2.

If the Open–Loop Dynamic RGA is close to 1 then our two multivariable dynamic curve will be close to two dynamic single–variable curve from Figure 2.

All the parameters of the system and controller are here and we will try to analyze and design the closed–loop system using only these equations, without reference to the transfer matrices.

ALGEBRAIC MIMO CONTROLLER DESIGN

Let's use the algebraic formulas for controller design. We have two equations; the first has two parameter interactions: the interaction of process open–loop static and controller gain.

The second equation shows three parameters interaction: open–loop static, dynamic and controller. Here we have all the relevant parameters.

To produce the best controller we could use both equations, to simplify the design, let's try the second one because it takes into account both static and dynamics.

Algebraic Controller Design of the Best Closed Loop Dynamic Matrix

Using the static matrix model we can see from static Eq. (7) that if we have the controller with parameters big enough, the closed loop RGA will be close to the 1.

The dynamic equation is more complicated, and probably more important because of the stability. If we use the dynamic formula (8) for the controller matrix, the best controller for the system that will produce dynamic DRGA (V) = 1 will be:

$$\mathbf{R} = \mathbf{K}^{-1}(\mathbf{T} - \mathbf{I}) \tag{9}$$

This controller has the best structure of matrix **R**, but we can improve the controller by adding multiplication with the scalar constant r is given to be able to increase the parameters of the controller. So here we have the best controller for the dynamic part of the system.

$$\mathbf{R} = r^{*}\mathbf{K}^{-1}(\mathbf{T} - \mathbf{I}) \tag{10}$$

The two aspects (static, dynamic) of closed loop transfer functions are not of the same importance for each case. Each controlled unit and maybe each set of parameters have his particular priority, which has to be judged for the particular case.

Calculations of Algebraic Controller Design for some simple examples. Both static and dynamic open loop symmetric and their RGA are close to 1.

Some examples of calculation and graphs are given in the following pages. A simple Matlab program [11–13] was written for 2×2, and 4×4 systems. The more complicated cases should be solved using both matrix equations.

Case 1, The matrix data are:

$$K = \begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix} \quad T = \begin{pmatrix} 5 & 2 \\ 2 & 5 \end{pmatrix}$$

and the matrices are diagonally dominant. This static and dynamic RGA are 1.2, and the controller parameter is r = 10.

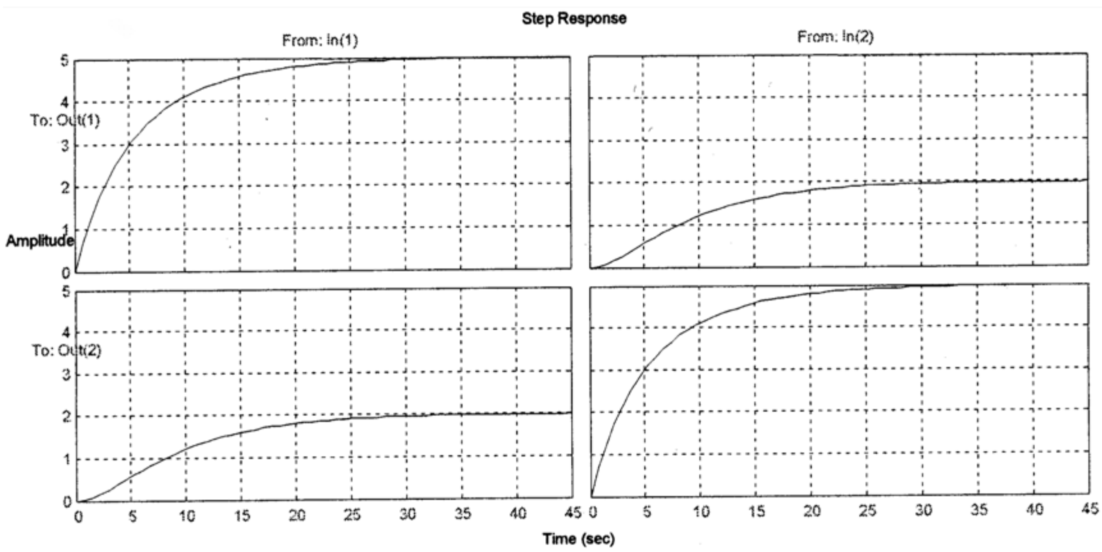

Figure 3. Open Loop MIMO, Case 1, dynamic DRGA = 1.2 close to 1, so dynamic is not difficult. Rise time is about 30 s.

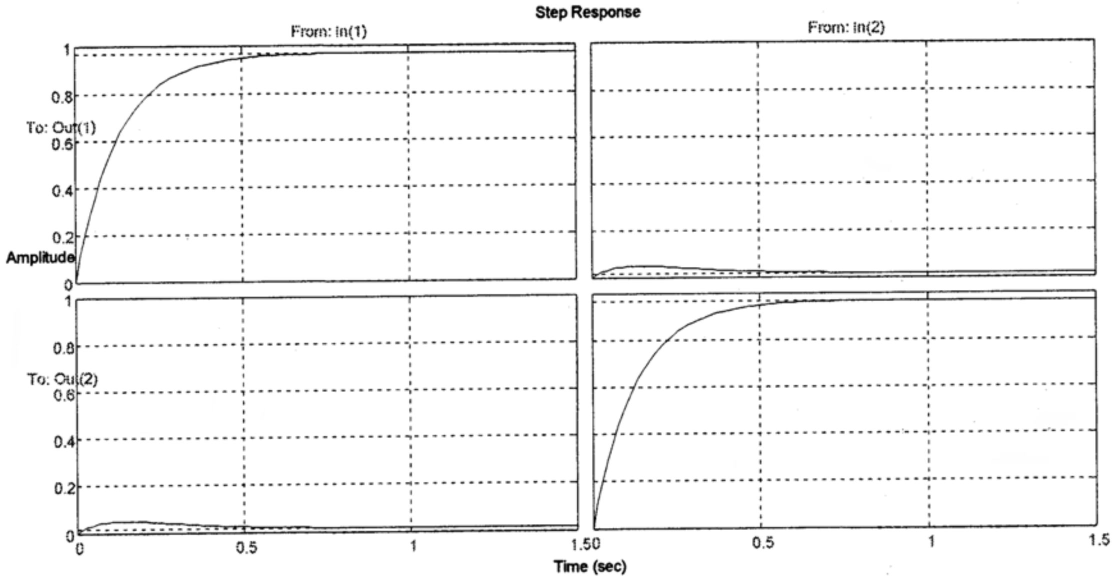

Figure 4. Closed Loop MIMO, algebraic controller design scalar parameter r = 10. Very good closed loop response with minimal static and dynamic interaction, with small r because DRGA and SRGA are close to 1. Rise time is about 0.5 s, much smaller than open loop.

Case 2, The matrix data are:

$$K=\begin{pmatrix}5 & 2\\ 2 & 5\end{pmatrix} \quad T=\begin{pmatrix}5 & 3.75\\ 3.75 & 5\end{pmatrix}$$

This case has much bigger dynamic RGA = 3.75, the dynamic is more difficult and because of this the controller parameter has to be bigger, so r = 80.

We can see that the Eq. (10) has resulted in good and very fast closed loop response. In the first case the dynamic DRGA was 1.2 and the dynamic was not difficult, but in the second case the dynamic DRGA was about 3.75 and the closed loop result is good but we had to use big value for the scalar parameter r.

Case 3. The similar results we get with 3x3 system and 4x4 is represented in the next graph.

The open loop data for matrices are:

$$K=\begin{pmatrix}5 & 1 & 1 & 1\\ 1 & 5 & 1 & 1\\ 1 & 1 & 5 & 1\\ 1 & 1 & 1 & 5\end{pmatrix} \quad T=\begin{pmatrix}5 & 3 & 3 & 3\\ 3 & 5 & 3 & 3\\ 3 & 3 & 5 & 3\\ 3 & 3 & 3 & 5\end{pmatrix}$$

The values of static and dynamic open loop RGAs are:

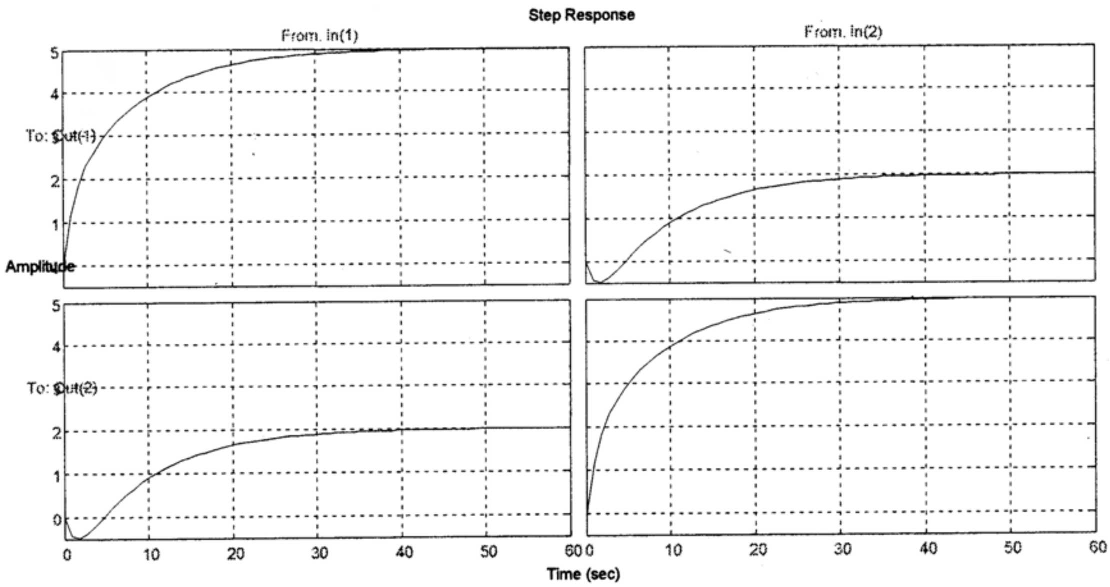

Figure 5. Open Loop MIMO Case 2, difficult dynamic RGA = 3.75. Rise time is about 40 s.

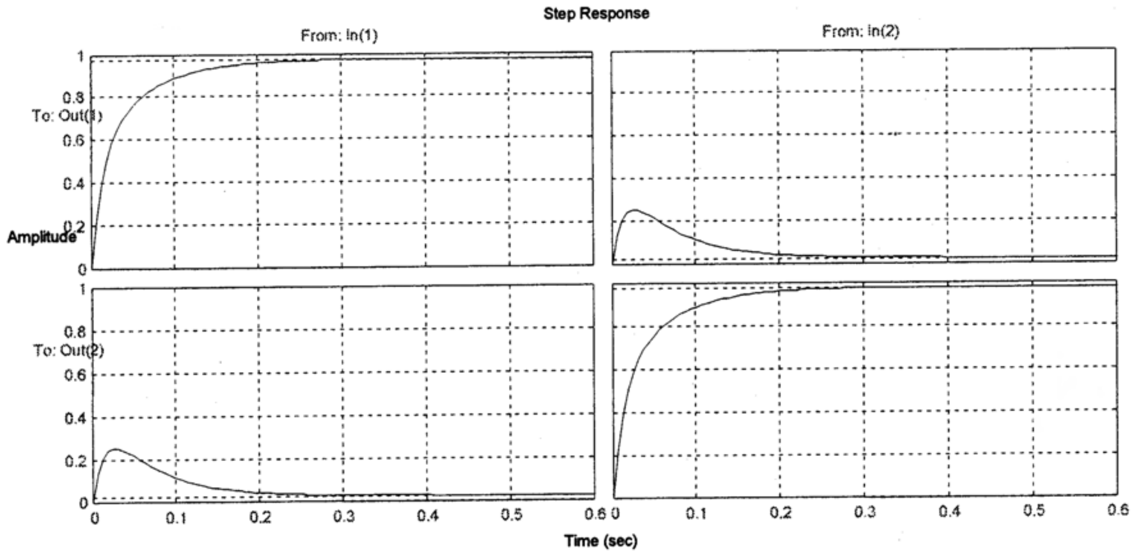

Figure 6. Closed Loop, Controller designed using Algebraic method, controller scalar parameter r = 80. For difficult open loop dynamic with DRGA>>1 the parameter r has to be very big to get the good closed loop response. Rise time is about 0.4 s, much smaller than open loop.

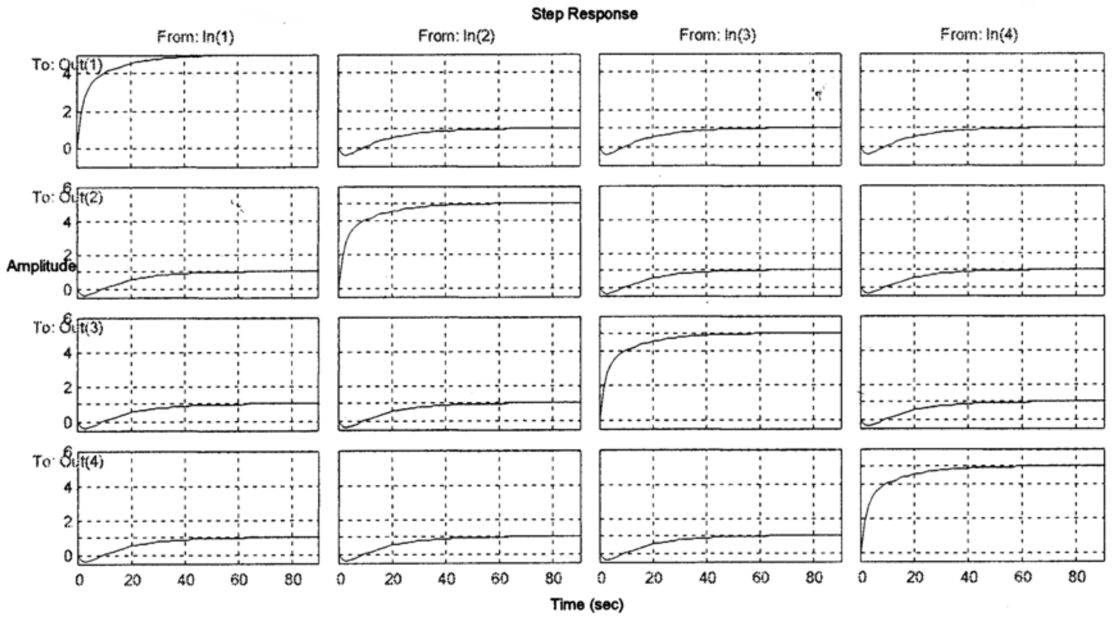

Figure 7. Open loop graphs of 4×4 system. Rise time is about 40 s.

$$RGA(K)=\begin{pmatrix}1.09 & -0.03 & -0.03 & -0.03\\ -0.03 & 1.09 & -0.03 & -0.03\\ -0.03 & -0.03 & 1.09 & -0.03\\ -0.03 & -0.03 & -0.03 & 1.09\end{pmatrix}$$

$$RGA(T)=\begin{pmatrix}1.96 & -0.032 & -0.032 & -0.03\\ -0.032 & 1.96 & -0.032 & -0.032\\ -0.032 & -0.032 & 1.96 & -0.032\\ -0.032 & -0.032 & -0.032 & 1.96\end{pmatrix}$$

So the RGA(K) is unit matrix and RGA(T) is approximately diagonal.

And the values of closed loop RGAs are:

$$RGA(H)=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$RGA(V)=\begin{pmatrix}1.05 & -0.018 & -0.018 & -0.018\\ -0.018 & 1.05 & -0.018 & -0.018\\ -0.018 & -0.018 & 1.05 & -0.018\\ -0.018 & -0.018 & -0.018 & 1.05\end{pmatrix}$$

The result for this case is very good, but the open–loop static and dynamic RGAs are not difficult, they are close to the value of 1.

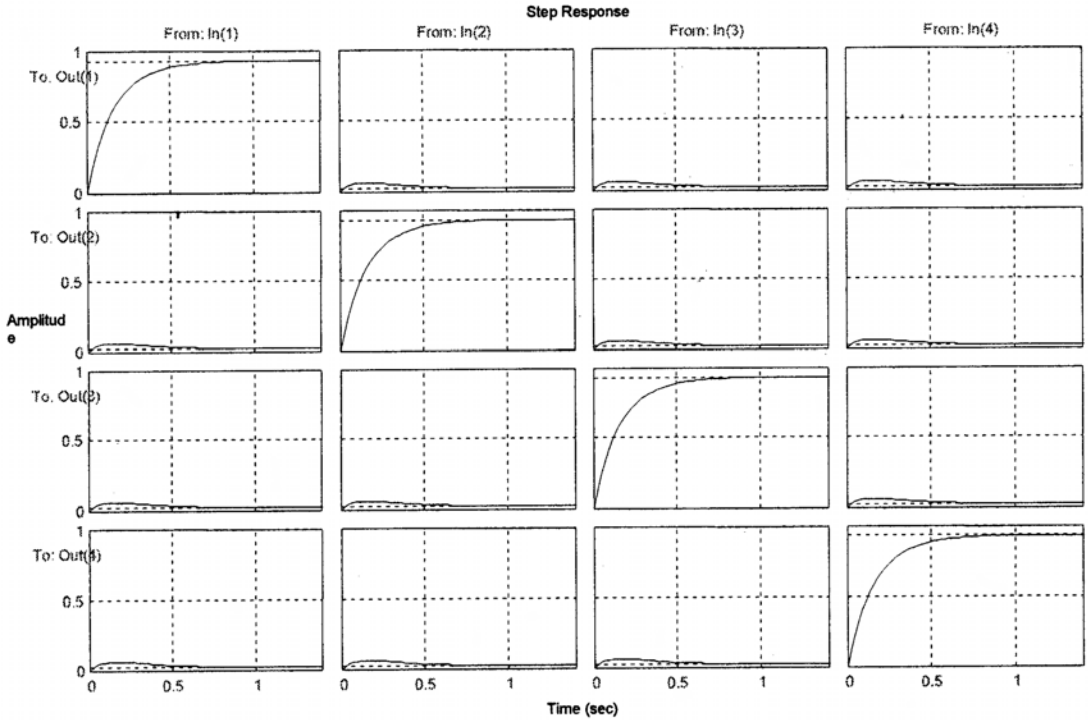

Figure 8. Closed loop graphs. Rise time is reduced to 0.5 s and both static RGA(H) and dynamic RGA(V) are practically equal to 1.

For the more difficult values of RGAs we should probably use both algebraic formulas and make some trade–off between static and dynamic closed loop RGAs.

CONCLUSIONS

A new algebraic multivariable controller design is proposed to solve the control of multivariable systems.

First, two algebraic matrix equations are derived that connect static and dynamic parts of open and closed control loops.

Second, the problem with these equations is that they are not simple linear algebra systems; they are functional relations between matrices only. The only way to understand these equations for the purpose of process control is to use RGA of matrices. In the equations we have four types of matrices so we have four types of RGAs: open–loop static, (OLS–RGA($\mathbf{K}$)) open–loop dynamic (OLD–RGA($\mathbf{T}$)), closed–loop static (CLS–RGA($\mathbf{K}_{CL}$)) and closed–loop dynamic (CLD–RGA($\mathbf{T}$)).

Third, using the dynamic matrix formula we can easy calculate the controller gain which will make $\mathbf{T}_{CL}$ matrix to have dynamic closed loop RGA($\mathbf{T}_{CL}$) = 1. The controller algebraic formula is simple $\mathbf{R} = \mathbf{K}^{-1}(\mathbf{T} - \mathbf{I})$. This formula gives the proper structure for the controller, and it can be improved by multiplying the first formula by the scalar (r) $\mathbf{R} = r^*\mathbf{K}^{-1}(\mathbf{T} - \mathbf{I})$.

The second formula was used with excellent success for 2×2, 3×3 and 4×4 systems.

The results are very good and they suggest that the general case for the 2×2 and 4×4 systems. Using the dynamic matrix equation produced closed loop results which have the RGA of both static and dynamic practically equal to 1.

APPLICATIONS

1. Apply the new algebraic multivariable controller design to derive controllers for various MIMO systems with SRGA and DRGA close to 1.

2. Derive the general procedure for more complicated cases where static and dynamic open–loop RGAs have all possible values from big negative to values between –1 and +1 and to the big positive values.

3. Apply algebraic design for plantwide control: Apply traditional techniques such as RGA, eigenvalues and singular value analysis and MPC to the two matrix equations for plantwide control analysis and control.

REFERENCES

[1] E.H. Bristol, On a new measure of interaction for multivariable process control, IEEE Trans. Autom. Control, AC–11, (1966) 133.

[2] S. Skogestad, I. Postlethwaithe, Multivariable Feedback Control: Analysis and Design, 2nd ed. John Wiley & Sons, Chichester, 2007.

[3] T. Mc Avoy, Y. Arkun, R. Chen, D. Robinson, D. Schnelle, A new approach to defining a dynamic rela-

tive gain, Control Engineering Practice, Pergamon, **11** (2003) 907–914.

[4] W.L. Luyben, Process Modeling, Simulation and Control for Chemical Engineers, McGraw–Hill, 1990, Part VI Multivariable Processes.

[5] G. Stephanopoulos, Chemical process Control, An Introduction to Theory and Practice, Prentice–Hall International 1984, Part VI, Multivariable Processes.

[6] M. Arranz, Sensitivity of Variable Pairing in Multivariable Process Control to Model Uncertainties, Lulea University of Technology, 2007.

[7] D. Vaes, Optimal Static Decoupling for Multivariable Control Design, PhD Thesis, Catholic University Leuven, Dec. 2005.

[8] V. Kariwala, Multi–loop Controller Synthesis and Performance Analysis, PhD Thesis, University of Alberta, Canada, 2004.

[9] B. Asmar, Control of a Two–Stage Refrigeration System, PhD Thesis University of Nottingham, 1999.

[10] B. Asmar, A General Method to Calculate Input–Output Gains and the Relative Gain Array for Integrating Processes, Computers and Chemical Engineering, MSc Thesis, University of Nottingham, 1995.

[11] D. McMahon, Matlab DeMystified, McGraw Hill, New York, 2007.

[12] Marlin, Process Control Course, 2002, Chapter 20, RGA Bonus Relative Gain Measure of Interaction.

[13] B. Shanian, M. Hassul, Control System Design Using Matlab, Prentice hall, Englewood Cliffs, New Jersey, 1993.

IZVOD

TRANSFORMISANJE PROBLEMA MULTIVARIJALNOG SISTEMA DIFERENCIJALNIH JEDNAČINA ZA POTREBE UPRAVLJANJA U PROBLEM ALGEBARSKIH JEDNAČINA KOJE POVEZUJU MATRICE OTVORENOG I ZATVORENOG KOLA I PROJEKTOVANJE NOVOG MIMO REGULATORA KORISTEĆI ALGEBARSKE JEDNAČINE

(Naučni rad)

Aleksandar Cingara
PharmEng Technology Inc., Toronto, Ontario

Prikazan je metod uproščavanja sistema linearnih diferencijalnih jednačina izvođenjem dve matrične algebarske jednačine koje povezuju statičke i dinamičke matrice otvorenog i zatvorenog kola. Ove dve algebarske jednačine sadrže sve parametre potrebne za projektovanje proporcionalnog regulatora za multivarijabilni sistem. Projektovanje regulatora pomoću dve algebarske jednačine je daleko jednostavnije nego projektovanje regulatora koristeći sistem linearnih diferencijalnih jednačina. Jednostavan multivarijabilni Matlab program može da projektuje multivarijabilni proporcionalni regulator bilo koga reda. Metoda je veoma jednostavna tako da se verovatno može koristiti i za projektovanje postrojenja ili čitavih fabrika, koje zahtevaju matrice sa velikim brojem kolona i vrsta.

Ključne reči: Multivarijabilni sistemi • Algebarske jednačine • Sistemi upravljanja • Projektovanje regulatora •

Key words: Multivariable control • MIMO control • Algebric matrix equations • Control systems • Controller design •