ROOT-LOCUS METHOD FOR COMPENSATING AND PERFORMANCE TESTING MULTILOOP LINEAR CONTROL SYSTEMS

By

T. Kovács, L. Lakatos and R. Haber

Department of Automation, Technical University, Budapest

(Recieved July 10, 1973)

Presented by Prof. Dr. F. Csáki

Introduction

Design of complicated control loops is difficult, time-consuming, even if the indispensable mathematical tools are available. In such cases the existing thumb rules are seldom applicable and only under specific conditions. The availability of digital computers considerably accelerates the design because of providing possibility for examining the variations.

This paper deals with the design of linear systems only in the complex domain. Generalization of the classical root-locus method permits to examine the effect of any desired system parameter on the position of the poles in the transfer function of the feedback system. A general linear structure is assumed as a controller, that contains the classic compensators as special cases.

The control loop is assumed to be a multiloop system, so the reduction of the signal flow graph precedes the sensitivity test. A significant simplification consists in the single reduction of the signal flow graph even if different controllers are examined.

The described methods are applicable both in continuous and in discrete domain.

1. *Reduction of the signal flow graph*

Signal flow graphs are illustrative for relations of the physical systems describable by sets of linear equations. The nodes of the signal flow graph correspond to physical quantities, and the branches express the relations of the physical quantities. The signal flow graphs can readily be utilized in control theory, in this case a transfer function can be assigned to every branch. The relation between any two nodes of the signal flow graph can be determined by several methods. One of them is the node-eliminating method [1].

The essence of this method is the following. Equivalent transformations of the signal flow graph can be made so that the interaction between nodes remains unchanged if the intermediary nodes are eliminated. If in a signal flow graph all but the input and output nodes are eliminated, then the transfer function of the remaining branch will give the relation between the input and output nodes.

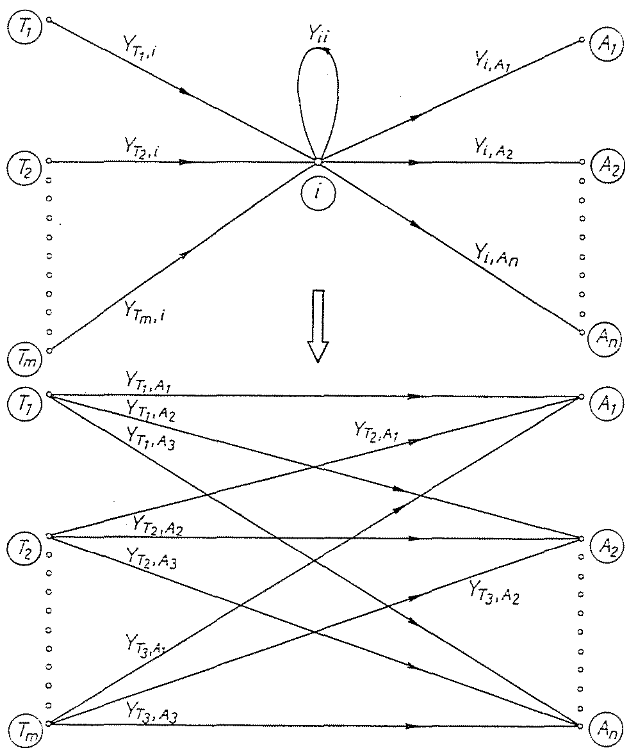

Fig. 1

A node can be eliminated as shown in Fig. 1 (here the i-th one). The transfer function of any obtained new branch:

$$Y_{T_k,A_j}(s) = \frac{Y_{T_k,i}(s)\ Y_{i,A_j}(s)}{1 - Y_{ii}(s)} . \tag{1}$$

The flow-chart of the program based on the described algorithm is seen in Fig. 2. Remind in representing the signal flow graph:

— Neither the input nor the output node can be connected to more than one branch.

— The nodes are numbered arbitrarily continuously. The input node is numbered 1 and the output node is numbered the highest (n).

— The branches are numbered arbitrarily but continuously beginning with 1.

It is noted that the resulting transfer function is not of the simplest form, since the program does not reduce the order when identical poles and zeros are in the transfer function.

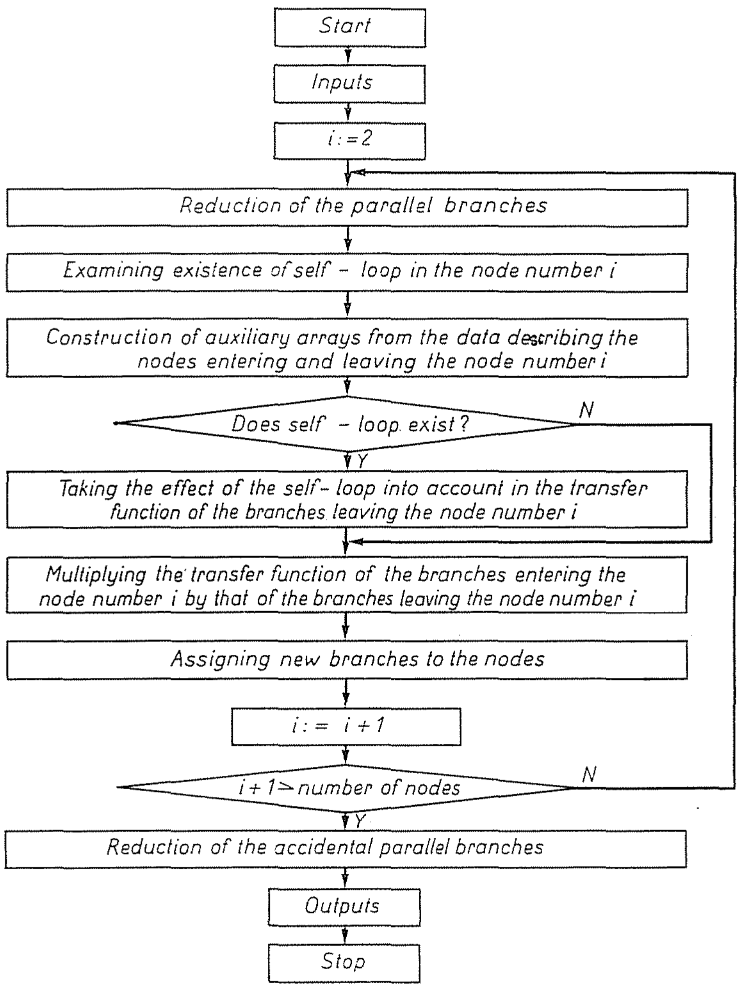

Fig. 2

2. *Reduction of the signal flow graph to four nodes*

It is often desirable to reduce the signal flow graph to four nodes, such as, in addition to input nodes, to the connecting points of the nonlinear unit Y_{nl} in the system. The described program can perform it, only the nodes are to be eliminated from 2 to $(n-3)$. The nodes 1, $(n-2)$, $(n-1)$ and n are those to be retained. It is obvious that at the beginning only one branch can be connected to each of the remaining nodes. The original signal flow graph can be simbolically represented like in Fig. 3. The general structure of the resulting signal flow graph is seen on Fig. 4. Obviously this signal flow graph contains no other branch. Namely if in the original case all only leaving branches are connected to nodes 1 and $(n-1)$, and joining branches are connected to nodes $(n-2)$ and n, then this must be true also for the resulting signal flow graph.

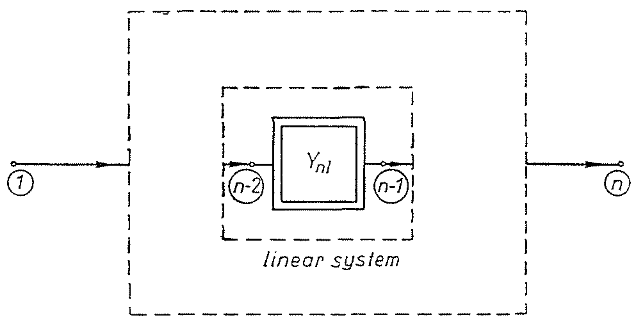

Fig. 3

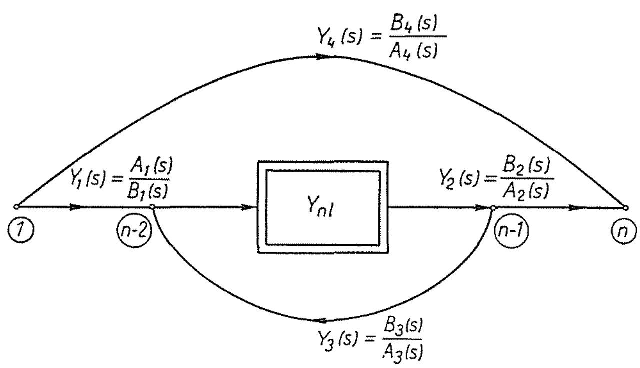

Fig. 4

3. *Determination of the root-locus diagram*

The classic method finds the poles of the single loop feedback system as a function of the overall gain K (Fig. 5):

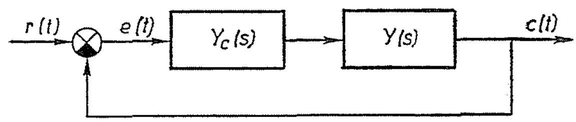

Fig. 5

$$Y(s) = \frac{h(s)}{g(s)} e^{-s\tau} \tag{2}$$

$$Y_c(s) = K. \tag{3}$$

The characteristic equation is of the following form:

$$F(s) = g(s) + Kh(s)e^{-s\tau} = 0. \tag{4}$$

This equation can be solved by several methods. The well-known root-locus plotting rules [2] are useless, because Eq. (4) can be solved by whatever mathematical root-finding program for different K values [3]. These methods are disadvantageous by not indicating in advance for what value to find the roots, although only a part of the complex plane meeting the performance requirements is of interest.

KRALL and FORNARO eliminated K to give conditions under which $s = x+jy$ is root of the equation [4]:

$$\Phi(x, y) = \cos \tau y \sum_{k=0}^{\infty} \frac{(-1)^k y^{2k+1}}{(2k+1)!} \sum_{i=0}^{2k+1} \binom{2k+1}{i} (-1)^{2k+1-i} h^{(i)}(x) \times$$
$$\times g^{(2k+1-i)}(x) - \sin \tau y \sum_{k=0}^{\infty} \frac{(-1)^k y^{2k}}{(2k)!} \sum_{i=0}^{2k} \binom{2k}{i} (-1)^{2k-i} h^{(i)}(x) \times \tag{5}$$
$$\times g^{(2k-i)}(x) = 0.$$

Thus after indicating on the complex plane a strip defined by $X_{min}-X_{max}$ and $Y_{min}-Y_{max}$ the poles within this rectangle can be found at any desired accuracy. Beginning the computation with the discrete division of x, the eventual vertical asymptote is to be found by other means. Then the overall gain can be computed in the related point of the root-locus diagram:

$$K = e^{\tau x} \,|\, g(s) \,|^2 \cos \tau y / Re(h(s)\overline{g(s)}) \tag{6}$$

where the upper line denotes conjugation.

The program based on the described algorithm has already been published [5].

4. *Root-locus diagram as a function of the compensating term*

The root-locus method is generalized by locating the poles of the feedback system versus the parameters of the compensating unit.

In the case when every parameter of the controller is at most a linear function of the tested parameter, and the transfer function of the process contains no dead time ($\tau = 0$), then the task can be solved with the program prepared for the parameter as overall gain, see item 3. Let the controller in Fig. 5 be:

$$Y_c(s) = \frac{BO(s) + BP(s)}{AO(s) + AP(s)}. \tag{7}$$

Here $BO(s)$ and $AO(s)$ are independent of the examined parameter, while $BP(s)$ and $AP(s)$ are its linear functions:

$$BP(s) = p \; BO^*(s) \tag{8}$$

$$AP(s) = p \; AO^*(s) \tag{9}$$

where $BO^*(s)$ and $AO^*(s)$ are independent of p.

Every familiar classic controller (P, PI, PD, PID, phase lead, phase lag, phase lead and lag) can be assigned to the outlined compensating unit being linear function of the tested parameter.

In case of PI controller:

$$Y = A_p \left(1 + \frac{1}{sT_I}\right) = \frac{A_p + A_p T_s}{sT_I} . \tag{10}$$

Versus A_p (T_I = constant):

$$\begin{aligned} BO(s) &= 0 , \qquad & BP(s) &= 1 + sT_I , \\ AO(s) &= sT_I , \qquad & AP(s) &= 0 . \end{aligned} \tag{11}$$

Versus T_I (A_p = constant):

$$\begin{aligned} BO(s) &= A_p , \qquad & BP(s) &= sA_p ; \\ AO(s) &= 0 , \qquad & AP(s) &= s . \end{aligned} \tag{12}$$

The poles of the control loop in Fig. 5 with a controller according to Eq. (7) can be computed from the following equation:

$$\frac{BO(s) + BP(s)}{AO(s) + AP(s)} \, Y(s) = -1 . \tag{13}$$

From Eq. (8) and Eq. (9) for $\tau = 0$ we obtain:

$$[BO(s)h(s) + AO(s)g(s)] + p[BO^*(s)h(s) + AO^*(s)g(s)] = 0 . \tag{14}$$

Eq. (14) is seen to be analogous to Eq. (4), thus the program described in item 3 for the root-locus diagram versus the overall gain can be applied as a function of optional compensating parameters fulfilling the described stipulations.

5. *The root-locus diagram of multiloop systems as a function of the parameters of the compensating units*

The simplest train of thoughts is to reduce the complicated no dead time linear system to an element with a single transfer function for different values of compensating parameters by means of the program reducing the signal flow graph in item 1, then finding the roots of the denominator of this fraction by a mathematical root-finding program [3]. This procedure requires in addition to presuming the parameters many repetitions of reducing the signal flow graph.

It is essentially simpler to reduce the signal flow graph as shown in item 2, omitting the controller from the reduction. The general representation of the reduced signal flow graph is seen on Fig. 4, mentally replacing the non-linear element by the controller ($Y_{nl} \Rightarrow Y_c(s)$). The poles are obtained from the equation:

$$Y_c(s)\,\frac{B_3(s)}{A_3(s)} = 1. \tag{15}$$

The roots of $A_1(s)$, $A_2(s)$, and $A_n(s)$ do not give further information, provided the system contained an external feedback. In the opposite case these transfer functions give poles independent of the parameter.

Supposing that the transfer function of the parameter, in Eq. (7) further simplifications are possible. Arranging the equation:

$$[-BO(s)B_3(s) + AO(s)A_3(s)] + p[-BO^*(s)B_3(s) + AO^*(s)A_3(s)] = 0\,. \tag{16}$$

Also this equation can be solved by the program for the root-locus diagram described in item 3, with the overall gain as parameter.

6. *Examples*

Besides the algorithms for partial problems a comprehensive program has also been made including the reduction of the signal flow graph in item 2, computation of polynomials dependent and independent of the parameter of the characteristic equation (16) of item 5, and activizing the program for the root-locus diagram in item 3. A program is available for plotting the results.

The statements in items 1 to 5 are illustrated by some examples.

Example 1

The equivalent reduction of the signal flow graph in Fig. 6 for the input and output nodes is the following transfer function:

$$Y(s) = \frac{4s^4 + 17s^3 + 18s^2 + 7s + 0.875}{8s^5 + 36s^4 + 43.5s^3 + 24.75s^2 + 7_s + 0.75}\,.$$

62

Example 2

The reduction of the signal flow graph in Fig. 7 to four nodes using notations in Fig. 4:

$$Y_1(s) = \frac{s^4+10s^3+37s^2+60s+36}{s^4+10s^3+38s^2+65s+42}$$

$$Y_2(s) = \frac{s^2+5s+6}{s^3+6s^2+12s+7}$$

$$Y_3(s) = \frac{-s^2-5s-6}{s^4+8s^3+24s^2+31s+14}$$

$$Y_4(s) = \frac{s^2+5s+6}{s^3+8s^2+22s+21} .$$

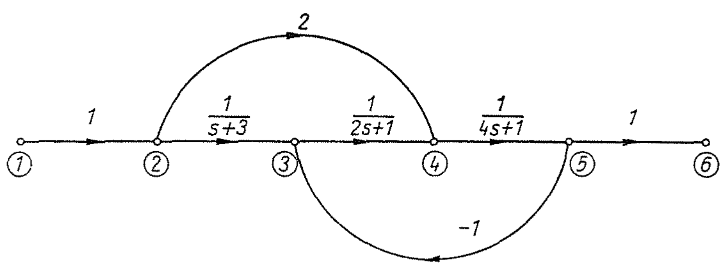

Fig. 6

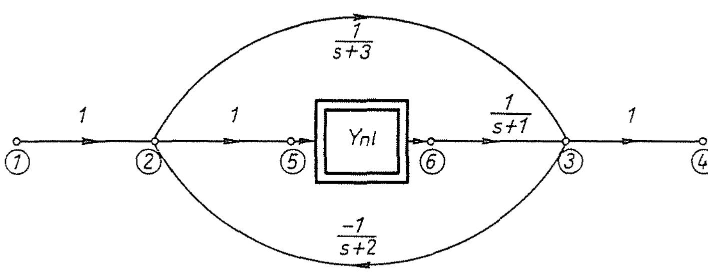

Fig. 7

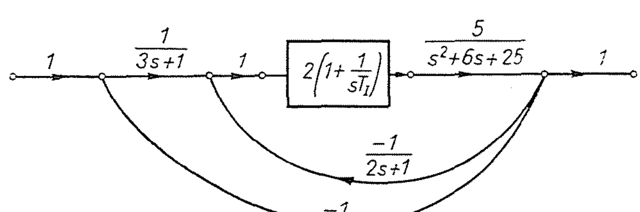

Fig. 8

63

Example 3

The root-locus diagram of the structure in Fig. 8 is seen in Fig. 9.

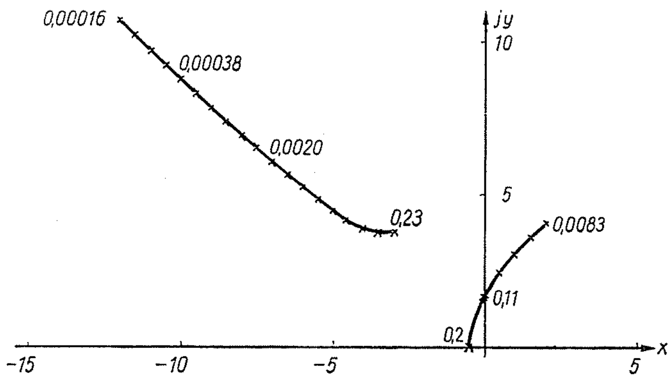

Fig. 9

Summary

This paper describes an algorithm for the reduction of linear, single-variable, multiloop systems containing no dead time. The system is represented by its signal flow graph and the reduction is accomplished for the input and output nodes. The reduction for four nodes has also been presented. A program has been prepared for computing the generalized root-locus diagram depending on linear compensating elements for these systems.

References

1. WALLACH, Y.: Zur analyse linearer Regelkreise mit Hilfe von digitalen Rechnenmaschinen. Regelungstechnik, Heft 7, 1967.
2. CSÁKI, F.: Dynamics of control systems. Akadémiai Kiadó, Budapest, 1966.
3. GYÜRKI, I.: Computer analysis and synthesis of control systems using the root-locus method.* Mérés és Automatika, XII/12. 1968.
4. KRALL, A. M.—FORNARO, R.: An Algorithm for Generating Root-locus Diagrams. CACM, Vol. 10/3, 1967.
5. KOVÁCS, T.: An Algol Program for Generating Root-locus Diagrams. Periodica Polytechnica, El. Eng., Vol. 13/Nr. 1—2, 1969.

* In Hungarian.

Dr. Tivadar KOVÁCS
Lóránt LAKATOS H-1521 Budapest,
Róbert HABER