International Journal of Multidisciplinary and Current Research ISSN: 2321-3124
Research Article Available at: http://ijmcr.com

Automatic Programming of Ziegler-Nichols and Strejc Methods based on Opened Loop Response

Jean Nirinarison Razafinjaka[a,b*], Andrianantenaina Tsiory Patrick[a,b], Andriamanjakony Holiarimanga Mariette[b] and Linda Christelle Mevalaza[b]

[a]Electricity Department, Higher Polytechnic School, University of Antsiranana, 201-BP O, Madagascar.
[b]Doctoral School Thematic Renewable Energies and Environment, University of Antsiranana, 201- BP O, Madagascar.

Received 15 Oct 2024, Accepted 11 Nov 2024, Available online 12 Nov 2024, **Vol.12 (Nov/Dec 2024 issue)**

Abstract

This paper is specially reserved for non-oscillatory response system. It is based on the opened loop response of the system. The principal objectives are to identify the system by its transfer function and to design the adequate controllers. For this purpose, a program is worked out to detects automatically, from measurements, the nature of the system: with new stationary state (SSNE) or with integral behavior (SIB) and to find the adequate controllers. Moreover, the methods of Ziegler - Nichols and of Strejc show a great similarity and can be combined in only one program. The results are very satisfactory and the program can be applied on real site by using, for example, LabView provided with its peripheral NI 6009.

Keywords: *PID controller, identification, design, Ziegler-Nichols, Strejc, programming.*

Introduction

The Ziegler-Nichols method to design PID controllers and Strjec's method to identify transfer function systems are both well-known methods. It is the reason that several articles are published about them [1], [2], [3]. To improve he PID controller design, many proposals are advanced as auto-tuning [4], [5], [6], [7], designing using Intelligence Articificial [8], [9, [10], [11]. In [12], Razafinjaka et *al.* develop a new method, called General Method (GM) to design PI controller.

In this paper, classical versions of both methods (Ziegler Nichols and Strejc) are used. The aim is to establish in a same program these two methods. Indeed, the two methods show a great similarity. The two methods rest on the system step response in opened loop.

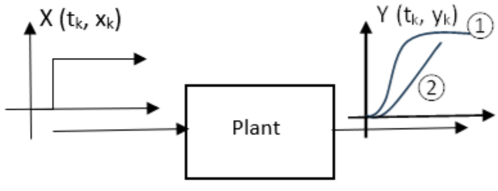


Fig. 1 Principle of the two methods

With, ① SNSS: System with New Stationary State and ② SIB: System with Integral Behavior

*Correspondant Author's ORCID ID: 0000-0000-0000-0000
DOI: https://doi.org/10.14741/ijmcr/v.12.5.4

Figure 1 above shows the basic principle. For both methods, the values of the vector time (T[t_k]) and the vector measurement (Y[yk]) are stored. Their length is N. The program is established only from these measurements. Functions giving the curves are not analytically defined.

From these measures, system function transfer and controller function transfer are calculated by using appropriated Tables. Figure 2 shows the general flow chart.

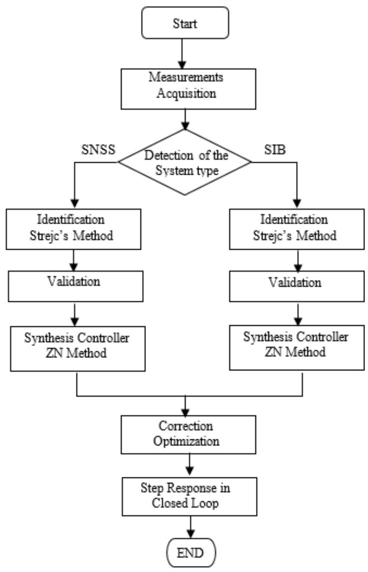


Fig. 2 General flow chart

As said above, all these various steps are combined in one program.

The organization is as follows: first, measurements acquisition is done. Then, the detection of the system type by using measurements is adopted. According to the system type (SNSS or SIB), appropriated method is chosen which allows to obtain the function transfer system (Strejc) and finally to design the selected controller (ZNOL).

2. System with New Stationary State (SNSS)

It is already said above that the two methods (ZNOL and Strejc) present a great similarity. They rest in the determination of the two pairs of parameters: (a, L) for ZNOL and (Tu, Ta) for Strejc. ZNOL gives the parameters of PID controller and Strejc's method gives the function transfer of the system. Figure 3 shows this similarity.

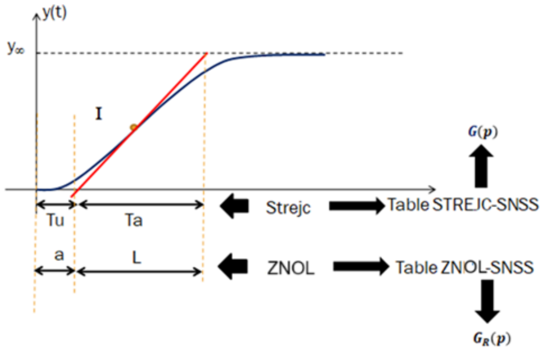


Fig.3 Similarity for the two methods

2.1 Ziegler Nichols' method in Opened Loop (ZNOL)

The ZNOL method is used to design the PID controller i.e. to determine the parameters which define the transfer function (1). The ZNOL is based on step response with non-oscillatory response. Two kinds of systems are here considered: SNSS and SIB. Ziegler-Nichols' method is an empirical one and used to determine the parameters of the PID controller. Relation (1) shows its transfer function canonical:

$$G_R(p) = g\left(1 + \frac{1}{pT_i} + pT_d\right) \quad (1)$$

In practice, a filter is added to the term D. So,

$$G_R(p) = g\left(1 + \frac{1}{pT_i} + \frac{pT_d}{1+pT_{dN}}\right). \quad (2)$$

With,

$$T_{dN} = \frac{Td}{N}, \quad N = 6 \div 20 \quad (3)$$

Typical value of N is: $N = 10$.

Relation (1) is used for the PID controller design and the PID filtered is automatically obtained by (2). Figure 4 gives the functional scheme adopted in this paper.

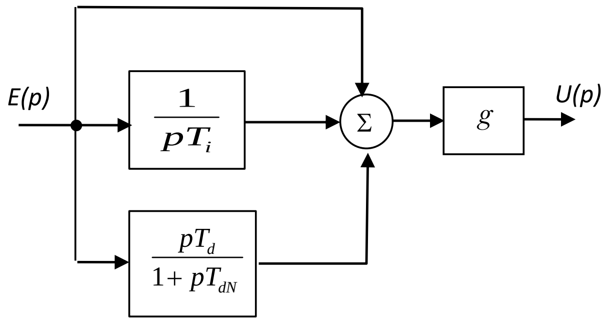


Fig. 4 Functional scheme for PID controller

2.1.1 Method of Ziegler Nichols for SNSS

For such a system, after a transient state, the response reaches a permanent state with a new finite value. Figure 5 shows an example for the step response of this kind of system.

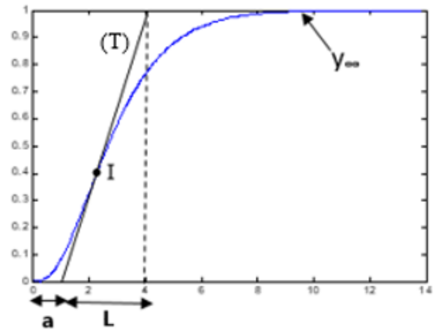


Fig. 5 Step response of SNSS

The objective is to have **a** and **L** which are the ZNOL parameters. By using (a, L), the parameters of the PID or one chosen controller are determined by using the ZNOL-Table (see Table 1). The procedure consists then in tracing the tangent (T) which passes at the inflection point I. If the function is known, the inflection point is determined by making null the second derivative. But, here, the function is not available only measurements.

Table 1: ZNOL-Table to controller design for SNSS

Controller	g	T_i	T_d	Transfer Function
P	L/a	-	-	$G_R(p) = g$
PI	0,9. L/a	3,33. a	-	$G_R(p) = g\left(1 + \frac{1}{pT_i}\right)$
PID	1,2. L/a	2a	0,25a	$G_R(p) = g\left(1 + \frac{1}{pT_i} + pT_d\right)$

In this table, it can be noted that, for the PID controller,

$$T_i = 4.T_d \quad (4)$$

a/ Detection of the passage by zero

Because numeric values are only available, the detection of the passage by zero must be computed; this case can be detected when the product of two consecutive points of the second derivative is negative as shown in Fig.4. It may be noticed that calculation is precise when the difference between these two points is the minimum as possible.

Approximate derivative can be done by using the command diff(X) in MATLAB, where X is a vector of measurements. For example,

$$\begin{cases} Y = \frac{diff(X)}{h} \\ Z = \frac{diff(Y)}{h} \end{cases} \quad (5)$$

Y is the first derivative, *Z* the second derivative and *h* a constant less than 1.

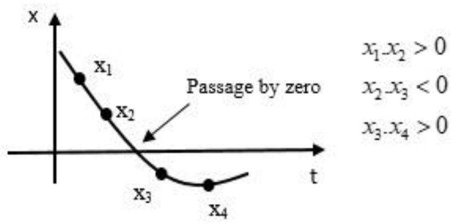


Fig. 6 Passage detection by zero

In Fig. 6, the product of two points x_2 and x_3 is negative ($x_2.x_3 < 0$), that means that there is a passage by zero. The detection is calculated by doing iterations. Directly, the value of the point which is positive is chosen.

b/ Equation of the tangent (T)

(T) is an oblique line with general equation:

$$y(t) = m.t + b \quad (6)$$

The slope ***m*** can be calculated around the coordinates of the inflection point I (x_o, t_o). The constant ***b*** can be given by noting that (T) passes by I (x_o, t_o) and inevitably by the point of its intersection A with the horizontal line of equation: y = y (∞) = y_∞ as shown in Figure 7.

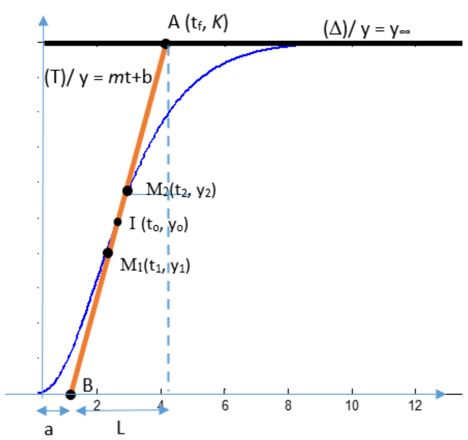


Fig. 7 Determination of the equation of the tangent (T)

At the inflexion point I, the coordinates are:

$$\begin{cases} I(t_o, y_o) \\ t_o = t(k_o) \\ y_o = y(k_o) \end{cases} \quad (7)$$

Where k_o is the rank (ko ∈ [1,N]) where the second derivative is null. To calculate the slope **m**, two points M1 (t_1, y_1) and M2 (t_2, y_2) in the vicinity of I can be used.

$$\begin{cases} m = \frac{y_2 - y_1}{t_2 - t_1} \\ y_2 = y(k_o+2);\ t_2 = t(k_o+2) \\ y_1 = y(k_o-2);\ t_1 = t(k_o-2) \end{cases} \quad (8)$$

To find the constant **b**, it is useful to remark that the tangent (T) passes through the point I. So,

$$y_o = m.t_o + b \Rightarrow b = y_o - m.t_o \quad (9)$$

c/ Determination of **a** and **L**

- The value of **a** can be detected by the passage by zero of the tangent (T). The procedure is the same as in section 2.1.1. It appears at the point B (see Fig. 7).

$$\begin{cases} a = t_B \\ t_B = t(k_B) \end{cases} \quad (10)$$

- The value of **(L+a)** is reached at:

$$\begin{cases} y_\infty = m.t_f + b; \\ t_f = t(k_f) = L + a \end{cases} \quad (11)$$

$$L = t_f - a. \quad (12)$$

Having the values of (a, L) permits to design a chosen controller according the Table 1.

2.2 Strejc's method for SNSS

Strejc's method is used to determine the system transfer function. Strejc proposes the form as follows:

$$G(p) = g\frac{e^{-pT_o}}{(1+pT)^n} \quad (13)$$

With *g*, the static gain, *T* the constant time, ***n*** the system order and T_o the time delay.

The identification consists to determine these parameters by using the Table 2 proposed by Strejc. Fig. 8 shows the curve to determine (Tu, Ta). As said above, it is the same as the parameters (a, L) of Zigler-Nichols.

$$\begin{cases} Tu = a \\ Ta = L \end{cases} \quad (14)$$

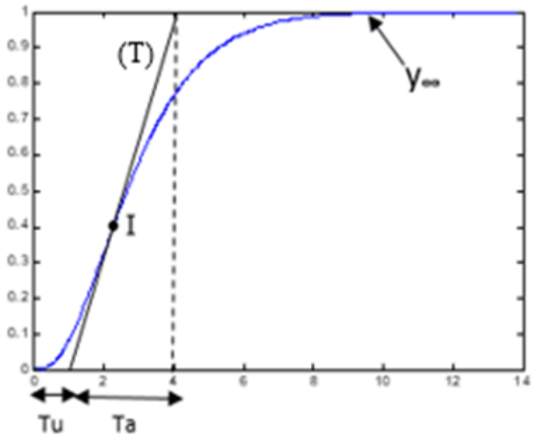


Fig.8 Step response for SNSS

Table 2: Strejc's table for SNSS

n	Tu/Ta	Tu/T	Ta/T
1	0	0	1
2	0.104	0.282	2.718
3	0.218	0.805	3.695
4	0.319	1.425	4.465
5	0.410	2.100	5.119
6	0.493	2.811	5.699
7	0.570	3.549	6.226
8	0.642	4.307	6.711
9	0.709	5.081	7.164
10	0.773	5.869	7.590

By having (Tu, Ta), the parameters are calculated using the Table 2. The procedure is as follows:

- Determination of Tu and Ta. The real value of Tu is: $Tu = Tu_R$
- Forming the ration $r = Tu_R/Ta$
- Taking the value of **n** immediately lower in the Table 2. (**$n = n_L$**)
- This lower value of **n_L** gives Tu_i
- The constant time T is calculated by using **n_L** and by the ration Ta/T
- The time delay is: $To = Tu_R - Tu_i$.
- The static gain is: g = Dy / Dx.

In Fig. 2 showing the general flowchart, Strejc's identification is done first, then he validation because correction may be brought. ZNOL is applied at the end.

3. System with Integral Behavior (SIB)

The method rests always in step response in opened loop. For this kind of system, there is no finite final value. The response increases infinitely with the time.

3.1 ZNOL for SIB

For ZNOL, Fig. 8 shows how to obtain (a, L) necessary to determine the PID controller using Table 1. The tangent (T) is easy to be obtained from the experimental curve.

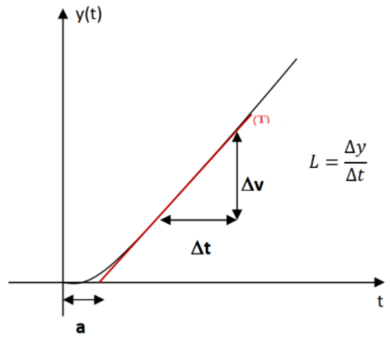


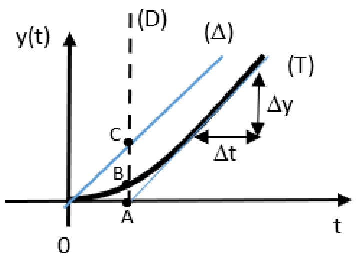
Fig.8 Determination of (a, L) for SIB with ZNOL

3.2 Strejc's method for SIB

For this system, Strejc proposes a transfer function as:

$$G(p) = \frac{1}{p}\frac{e^{-pTo}}{(1+pT)^n} \tag{15}$$

With **n,** the system order, T the time constant and **To** the time delay.



n	AB/AC
1	0,368
2	0,271
3	0,224
4	0,195
5	0,175

Fig.9 Strejc's method for SIB and corresponding Table

As in the ZNOL method, the tangent (T) is here obtained by the experimental curve. (D) is a line parallel to (T) but passing through the origin. The following equalities are given:

$$\begin{cases} A = (T) \cap (0, t) \\ B = (D) \cap (Cy) \\ C = (D) \cap (\Delta) \\ T_o = \overline{OA} \end{cases} \tag{16}$$

With (Cy), the experimental curve and contains the experimental measurements.

The constant time T is:

$$T = \frac{T_o}{n} \tag{17}$$

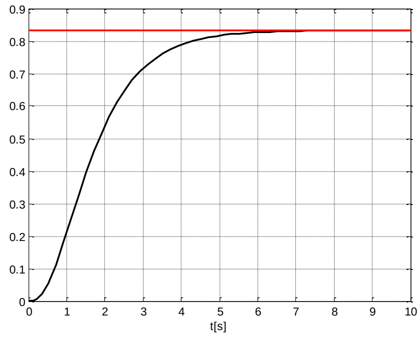
The system order **n** is chosen according the ration: $\frac{\overline{AB}}{\overline{AC}}$ Generally, it is practical to choose a system with non-high order.

4. Results and Discussions

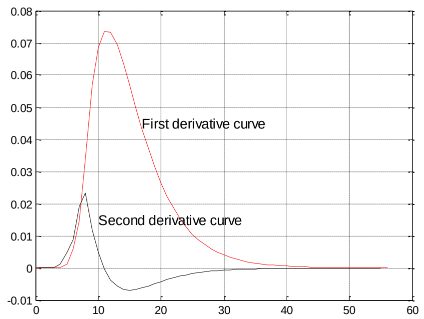
The following figures show the results obtained by the program. Figures 10 are dedicated for SNSS including

ZNOL and Strejc methods and Figures 11 for SIB. All the figures are presented chronologically as shown by the general flowchart in Figure 2.

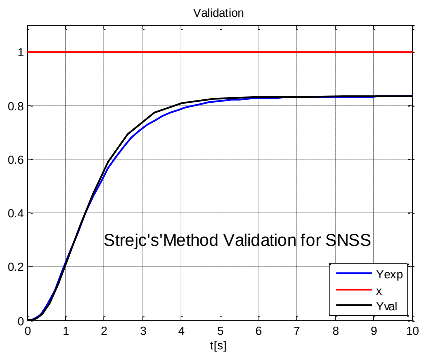
4.1 SNSS



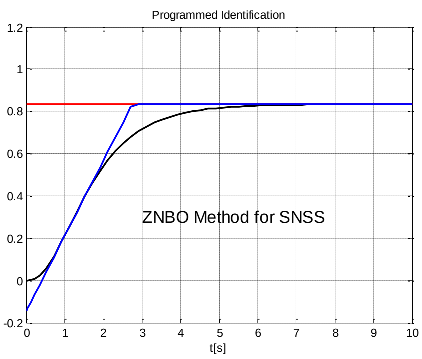
(a) Experimental measurements



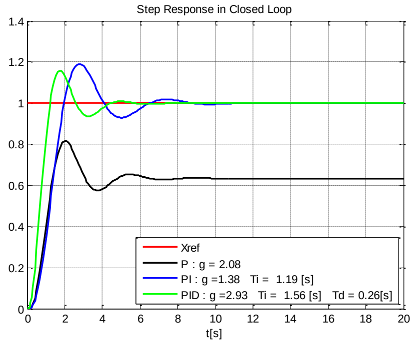
(b) First and derivative curves



(c) Identification validation



(d) ZNOL for SNSS



(e) Step response in closed loop after correction of controller parameters

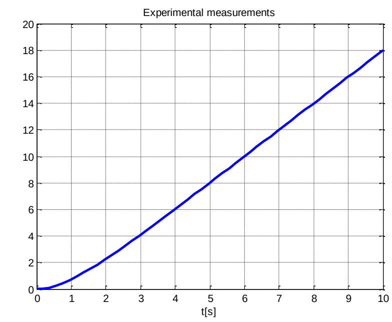
Fig. 10 Results for SNSS

Figure 11.a gives the output measurements according the vector time. It is not necessary to stock the input. Figure 11.b presents the first and second derivative curves calculated in section (2.1.1). The identification validation is important. In Figure 11.c, the curve with correction is directly adopted and compared with the experimental one.

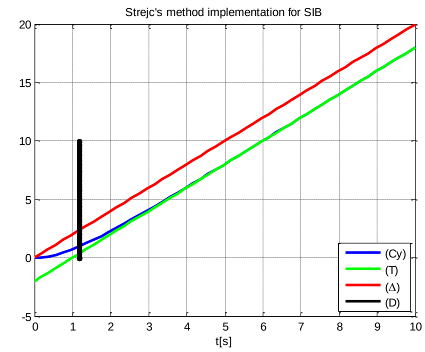
It is well-known that ZNOL method leads to a high overshoot and in Figure 11.d, parameters adjustment is already made. Corrections are brought on the integral constant time $\mathbf{T_i}$ and or on the gain **g**. For SNSS, P controller is not adapted; it is better to use PI or PID controllers.

4.2 SIB

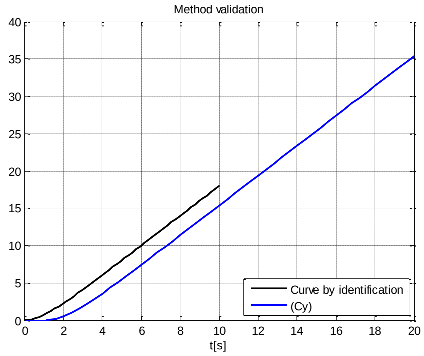
Figures 12 show the results for SIB.



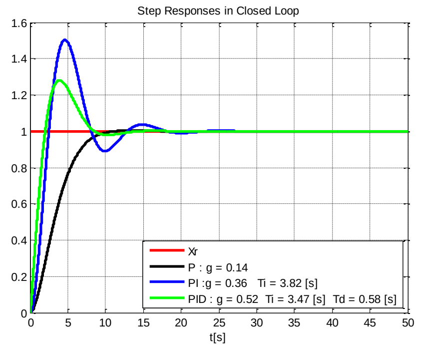
(a) Experimental measurements for SIB



(b Strejc's method for SIB



(c) Validation



(d) Step responses in closed loop

Fig. 12 Results for SIB

All these steps are programmed automatically. In Figure 11-c, there is still a difference but it is seen that acting on the constant time T doesn't bring more improvement. Because of the type of the system, P controller leads to a null position error. The more the gain of these controllers is higher, the more the overshoot (D1%) increases.

Conclusion

This paper is dedicated to programming automatically the ZN and Strejc methods based on an opened loop response. The objective is to do in one program the two methods. The obtained results show that it is possible and can be easily used in experimental site. Here, Broïda's method is not presented. The program may be also used in student practical works. Optimization of the of PID controllers by will be adopted with advanced method as Genetic Algorithm, PSO and so on. The objective is always to include them in one program which can run automatically.

References

[1] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, Trans. ASME, 64, pp. 759-768, 1942.

[2] K.J. Åström, T. Hägglund, PID Controllers: Theory, Design and Tuning, Instrument Society of America, 2nd edition, ISBN 1-55617-516-7, 1995.

[3] Jean N. Razafinjaka, Automatisation des synthèses des régulateurs standards et polynomiaux – Applications à la commande adaptative, Thèse ESP – EPFL, Lausanne, 1991.

[4] K. Johan Aström, T. Hägglund, Automatic Tuning of PID Controllers, Instrument Society of America, 1988.

[5] F. Mudry, Ajustage des Paramètres des Régulateurs PID, EIVD, Département d'Electricité et Informatique, 2002

[6] Shamsuzzoha M., Skogestad S., The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning, Journal Process Control, vol 20, pp.1120-1234 -2010.

[7] Rachid Oubrahim Auto-réglage PID par structure unifiée et performante, Thèse de Doctorat, Université de Metz, 1988.