Control System Designer for JSBSim with Algorithm of Automatic Ziegler-Nichols Sustained-Oscillation Method

Tomáš Vogeltanz[1, a)]

[1]*Tomas Bata University in Zlín, Faculty of Applied Informatics*
Department of Computer and Communication Systems
nám. T.G. Masaryka 5555, 760 01 Zlín, Czech Republic

[a)]vogeltanz@utb.cz

Abstract. In this paper, the algorithm of the automatic Ziegler-Nichols sustained oscillation method is described. The algorithm is used as the essential way of PID setting in the Control System Designer for JSBSim. The algorithm is tested during a control system design of the wing leveler, and its sufficient functionality is proved. The first section introduces the Ziegler-Nichols method itself, and the description of the Control System Designer for JSBSim follows. The main section shows the algorithm which can perform the method automatically. The last section illustrates the experiment with the successful PID adjustment.

INTRODUCTION

Three mode controllers with proportional, integral, and derivative (PID) part have gained widespread industrial acceptance [1, 2]. The overriding reason for that is its simple structure which has proved to be very robust because many process control problems can be adequately and routinely solved by conventional PID-control strategies [3].

However, the design and adjustment of control systems are complex and time consuming processes which need expertise in system and control theory [1]. For simplification of this processes, Ziegler and Nichols proposed PID controller tuning methods which have been widely utilized either in the original form, or in modified forms [4, 5, 2]. There are many researches which use this method [6, 7, 8, 9, 3, 5, 2].

JSBSim [10, 11] is probably the most flexible and comprehensive flight dynamics model and its quality was verified in many interesting projects [12, 13, 14]. However, there was lack of visualization and control design tools which was solved by the Control System Designer [15, 16]. Despite a user can visualize and design a control system for JSBSim now, the adjustment of the PID controls has to be set manually. Thus, the main aim of this paper is to implement an algorithm of the Ziegler-Nichols method into Control System Designer for JSBSim.

ZIEGLER-NICHOLS SUSTAINED-OSCILLATION METHOD

The procedure of closed-loop sustained-oscillation method is as follows: [5, 7, 3]

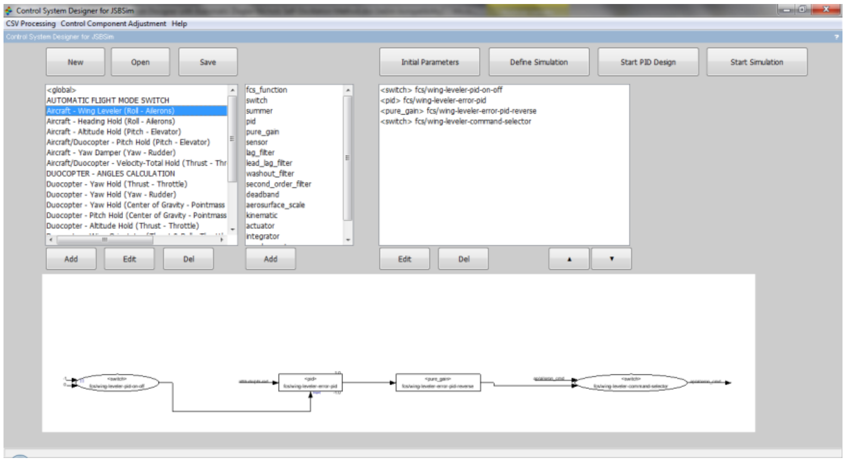
- Reduce the integrator K_i and derivative K_d gains to 0.
- Set proportional control K_p to 0.
- Increase the value of the proportional gain K_p until the point of instability is reached (i.e. sustained oscillation). At that point, the critical value of gain K_{crit}, is reached. (If the sustained oscillation does not occur, another method has to be applied)
- Measure the period of oscillation to obtain the critical time constant T_{crit}.
- Once the values for K_{crit} and T_{crit} are obtained, the PID parameters can be calculated, according to the design specifications as shown in Table 1.

TABLE 1. Ziegler–Nichols Tuning Formulas for Sustained-Oscillation Method [2, 7, 5]

Controller Type (rule name)	K_p	T_i	T_d
P (Classic)	$0.50 * K_{crit}$	-	-
PI (Classic)	$0.45 * K_{crit}$	$(1 / 1.2) * T_{crit}$	-
PID (Classic)	$0.60 * K_{crit}$	$0.5 * T_{crit}$	$0.125 * T_{crit}$
PID (Pessen Integral Rule)	$0.70 * K_{crit}$	$0.4 * T_{crit}$	$0.150 * T_{crit}$
PID (Some Overshoot)	$0.33 * K_{crit}$	$0.5 * T_{crit}$	$0.333 * T_{crit}$
PID (No Overshoot)	$0.20 * K_{crit}$	$0.5 * T_{crit}$	$0.333 * T_{crit}$

CONTROL SYSTEM DESIGNER FOR JSBSIM

The Control System Designer for JSBSim [15] is software for the design and visualization of a control system which is written in the JSBSim XML format. Furthermore, there is an option to manage simulation scripts of aircraft flights, and visualize the simulations using FlightGear [17]. The application was developed in the Scilab environment [18], version 6.0.1. The main form of the control system designer is illustrated in Fig. 1.


Control System Designer for JSBSim
CSV Processing
Control Component Adjustment
Help
Control System Designer for JSBSim
New
Open
Save
Initial Parameters
Define Simulation
Start PID Design
Start Simulation
Add
Edit
Del
Add
Edit
Del
▲
▼
<switch>
fcs/wing-leveler-pid-on-off
<pid>
fcs/wing-leveler-error-pid
<pure_gain>
fcs/wing-leveler-error-pid-reverse
<switch>
fcs/wing-leveler-command-selector

<global>
AUTOMATIC FLIGHT MODE SWITCH
Aircraft - Wing Leveler (Roll - Ailerons)
Aircraft - Heading Hold (Roll - Ailerons)
Aircraft - Altitude Hold (Pitch - Elevator)
Aircraft/Duocopter - Pitch Hold (Pitch - Elevator)
Aircraft - Yaw Damper (Yaw - Rudder)
Aircraft/Duocopter - Velocity-Total Hold (Thrust - Thr
DUOCOPTER - ANGLES CALCULATION
Duocopter - Yaw Hold (Thrust - Throttle)
Duocopter - Yaw Hold (Yaw - Rudder)
Duocopter - Yaw Hold (Center of Gravity - Pointmass
Duocopter - Pitch Hold (Center of Gravity - Pointmass
Duocopter - Altitude Hold (Thrust - Throttle)

fcs_function
switch
summer
pid
pure_gain
sensor
lag_filter
lead_lag_filter
washout_filter
second_order_filter
deadband
aerosurface_scale
kinematic
actuator
integrator

<switch> fcs/wing-leveler-pid-on-off
<pid> fcs/wing-leveler-error-pid
<pure_gain> fcs/wing-leveler-error-pid-reverse
<switch> fcs/wing-leveler-command-selector

FIGURE 1. Control System Designer for JSBSim

Algorithm of Automatic Ziegler-Nichols Sustained-Oscillation Method

The PID controller adjustment starts with the setting of initial parameters; this step influences the quality of the results and the probability of convergence. The general settings are: a path of a script file with commands for simulation, a path of an autopilot with controllers, a PID controller which will be adjusted, a path of output CSV file with results, rate of output data writing into the CSV file, a name of a JSBSim property which will be analyzed, setting of the start time and the end time for the analysis of the property progress, method of PID controller adjustment, specific parameters of selected method for PID adjustment, and optional JSBSim commands.

The specific parameters for the Ziegler-Nichols Sustained-Oscillation method are: maximum number of iterations, initial gain, gain change per iteration, gain constraint, and maximum tolerance of peak amplitudes. The algorithm for the automation of the method in the Control System Designer for JSBSim is implemented as follows:

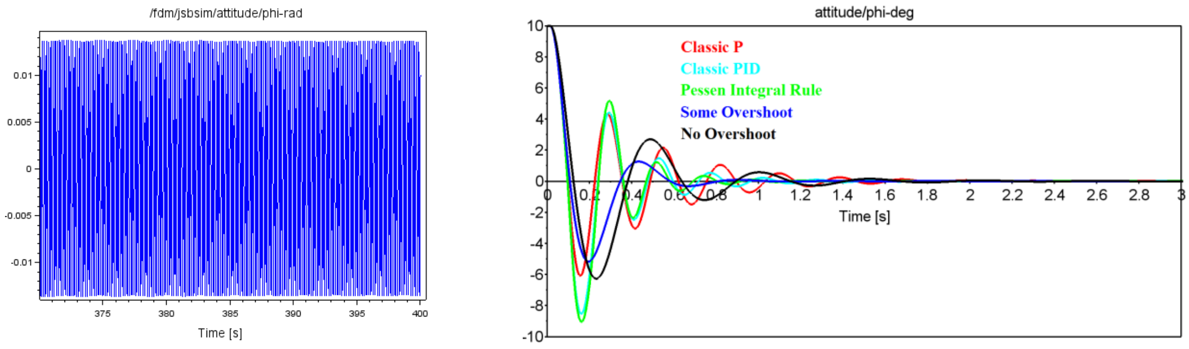
- Repeat until a maximum number of iterations or a gain constraint are reached, or until tolerance of critical gain change is lower than 0.00001.
  - At the start of iteration, an autopilot file with new PID setting is generated and saved to new folder.
  - A script file with simulation commands is executed.
  - An output CSV file generated by JSBSim is loaded.
  - Only specific range of values depending on time constraints set by user is taken.

    - The periodicity of progress of output property is analyzes. (Adequately high rate of sampling has to be adjusted – minimum recommended rate is 30 Hz; otherwise the analysis may return inaccurate results)
        - Constant, linear or piecewise linear trend from the vector of the output data are removed.
        - Amplitude peaks are found (if any) in output data.
        - Get sample period from time data (it must match the period of output data writing).
        - Distances between peaks are calculated.
        - The range of the distances of the peaks is calculated.
        - Time period are calculated using mean value of the distances and the sample period.
        - Maximum time period, which can be tolerated when the periodicity is checked, is calculated.
        - If the range of the peak distance is lower than or equal to 2 (a value from 1 to 3 is recommended), and if the time of the first peak is lower than or equal to the maximum allowed time period, and if the last peak occurs at the time when the next peak cannot follow (checked by the maximum allowed time period), and if the number of the peaks is at least 10, the data seems to be periodic.
            - Statistic information from the amplitudes of each peak is calculated.
            - If differences between amplitudes are smaller than the maximum tolerance of peak amplitudes (set by user), the data are considered to be periodic with same amplitude (i.e. the output property contains the sustained oscillation).
    - Information about the periodicity analysis of output data is written to a TXT file.
    - If the output data are periodic and the amplitude is same (with application of the user-defined maximum tolerance of peak amplitudes) for each period, the current gain is set to the previous gain, and gain change per iteration is decreased by a half of the current value. (This step is included to automatically find the gain at which the sustained oscillations occur for the first time)
    - Information about the iteration progress is written to a TXT file.
    - Increase the gain by the gain change per iteration.
    - Increase iteration number.
- If a critical gain and a critical period were found, PID controller parameters are calculated using a selected rule of the Ziegler-Nichols method.

PID CONTROLLER ADJUSTMENT USING AUTOMATIC ZIEGLER-NICHOLS SUSTAINED-OSCILLATION METHOD

This section demonstrates the PID controller adjustment using the algorithm described in the previous section. The specific parameters for the automatic Ziegler-Nichols sustained-oscillation method was set as follows: maximum number of iterations = 50, initial gain = 1, gain change per iteration = 1, gain constraint = %inf, maximum tolerance of peak amplitudes = 0.1, output rate = 100 Hz.

The control mode of the wing leveler had to be designed together with the altitude control mode which, in our case, needed three cycles of adjustment to improve the accuracy. The final critical gain for the control mode of wing leveler was analysed as ≈18.961 and the critical period was calculated as ≈0.175. Figure 2 (a) shows the sustained oscillation in the last step of the analysis, and (b) illustrates the progress of the control of the wing level.



(a)

(b)

FIGURE 2. (a) Sustained Oscillation at Critical Gain; and (b) Control of ϕ Angle of mini-UAV Using Ziegler-Nichols Rules

It is obvious that the best result is provided by the modified rule of the Ziegler-Nichols method named “some overshoot”. This setting has the lowest overshoot with the settling time around 0.8 second. Consequently, this type of adjustment was chosen for the control mode of the wing leveller for our control system.

CONCLUSION

In this paper, the algorithm of the automatic Ziegler-Nichols sustained oscillation method was described and tested. The algorithm was implemented into Control System Designer for JSBSim and was successfully used to the PID adjustment of the wing leveler. The work proved that the functionality of the algorithm is sufficient and faster than the manual setting. However, the main possible upgrade should be an analysis of unstable oscillations.

ACKNOWLEDGMENTS

This work was supported by the Department of Computer and Communication Systems of Tomas Bata University.

REFERENCES

1. A. Astolfi, *Systems and Control Theory: An Introduction* (Imperial College London, UK, 2006), 158 p.
2. M. Shahrokhi, A. Zomorrodi, *Comparison of PID Controller Tuning Methods* (Sharif University of Technology, Iran, 2012), 12 p.
3. H. Rasmussen, *Automatic Tuning of PID-regulators* (Aalborg University, Denmark, 2002), 21 p., available at: http://bsiswoyo.lecture.ub.ac.id/files/2012/01/autotuningPid.pdf
4. J. G. Ziegler, N. B. Nichols, “Optimum settings for automatic controllers.” Transaction for the ASME **64**, pp. 759–768 (1942).
5. V. Kumar, A. Patra, “Application of Ziegler-Nichols Method for Tuning of PID Controller”, In: *International Conference on Recent Innovations in Science, Technology, Management and Environment* (2016), 138-149.
6. K. J. Åström, T. Hägglund, “Revisiting the Ziegler–Nichols step response method for PID control,” *Journal of Process Control* **14**(6), 635-650 (2004), DOI: 10.1016/j.jprocont.2004.01.002
7. P. Gopi, G. Suman, “A new approach for Tuning of PID Load Frequency Controller of an Interconnected Power System,” *International Journal of Modern Trends in Engineering and Research* **2**(1), 18-28 (2015).
8. F. Haugen, “Comparing PI Tuning Methods in a RealBenchmark Temperature Control System”, *Modeling, Identification and Control* **31**(3), 79-91 (2010). DOI: 10.4173/mic.2010.3.1
9. G. M. van der Zalm, *Tuning of PID-type controllers*, (Technische Universiteit Eindhoven, Netherlands, 2004), 23 p., available at: https://pure.tue.nl/ws/files/4286492/625529.pdf
10. JSBSim contributors, JSBSim Open Source Flight Dynamics Model (2019), available at: http://jsbsim.sourceforge.net/
11. J. S. Berndt, “JSBSim: An open source flight dynamics model in C++,” In: *AIAA Modeling and Simulation Technologies Conference and Exhibit* (2004), pp. 261–287.
12. O. C. Cantarelo, L. Rolland, S. O'Young, “Validation discussion of an Unmanned Aerial Vehicle (UAV) using JSBSim Flight Dynamics Model compared to MATLAB/Simulink AeroSim Blockset,” In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2016), pp. 3989–3994, DOI: 10.1109/SMC.2016.7844857
13. B. C. Reitz, “Control System Development for Autonomous Aerobatic Maneuvering with a Fixed-Wing Aircraft,” Ph.D. thesis, Auburn University, 2014.
14. J. Rugarn, “Rapid Development of Realistic UAV Simulations,” Final thesis, Linköping University, 2009.
15. T. Vogeltanz, “Development of Control System Designer for JSBSim FDM”, In: *AIP Conference Proceedings* vol. 1978 (2015), DOI: 10.1063/1.5043721
16. T. Vogeltanz, “A Survey of Free Software for the Design, Analysis, Modelling, and Simulation of an Unmanned Aerial Vehicle,” Archives of Computational Methods in Engineering **23**(3), 449–514 (2016), DOI: 10.1007/s11831-015-9147-y
17. FlightGear contributors, FlightGear Flight Simulator – sophisticated, professional, open-source (2019), available: http://www.flightgear.org
18. ESI Group, Scilab (2019), available: https://www.scilab.org/