

Advanced Lecture for PID Controller of Nonlinear System in Python

Dong Hwa Kim



Abstract: *PID controller is very well known in engineering areas and it has a long history. So, there are many materials such as control knowhow for application, research paper, tuning method proven through a long history. It is an important to have an advanced lecture for design and tuning as much as development. However, it is very difficult to find for teaching knowhow. Current teaching style is implementation by MATLAB. However, MATLAB S/W is quite expensive as commercial based business focusing S/W. Advanced country or rich institute can provide site license. However, it is impossible for under developing country or small institute that cannot ready because of price. So, we must find alternative S/W to teach and research for implementation. Currently, many are interested in Python because it is open source and huge communities. This paper provides teaching experience of PID controller to nonlinear system to share knowhow and develop teaching method for teacher and students, effectively.*

Keywords: *About PID tuning, Python, Nonlinear control, Control lecture.*

I. INTRODUCTION

The PID controller has been using widely in the industrial areas even there are many control theories such as optimal, multivariable, and so on. The PID controller has a long history from 1942 [35]. It means that its control theory, performance, and knowhow for operation have been proved through many areas. It also illustrates that there are many man-powers such as engineer, designer, experimenter of operation, and tuning [5-6, 11, 12, 15-19] and research material and experiment system [19-31, 11-12, 33-34].

However, it is for lecturer difficult to find how we can teach is the best way and what process or content is the best way for improvement of student's understanding even there are so many materials about PID controller because of lack of information sharing about teaching method. This teaching method is quite important to nurture an engineer for industrial company's requirement as much as the technology development. Basically, almost students do not have a background in control theory and others. Sometimes, even students of 1st grade of master course do not understand the tuning method of PID controller, exactly. Moreover, almost under-course does not teach PID tuning method or they do not

recognize how it is important in the site because PID controller theory is too simple and there are many good control theories. Additionally, almost experimental lectures have been doing by MATLAB because it is easy to implement as student and lecturers, which they can implement two or three line in MATLAB and a long history from 1970s [32].

Lecturers can explain PID control theories and tuning method easily, but students cannot understand well why it is. Even the engineers of middle class in the site cannot understand well because they have never learned deeply from University. Currently, many engineers and scientists have been interesting in Python for control engineering as alternative tool for control engineering. Python S/W was released in 1991 [1-5] and have many communities to research. It is free and can be alternatives downloadable S/W on the web instead of MATLAB. However, we cannot find lecture materials in control engineering in Python like MATLAB. Also, some professors or so do not still know on how they can approach and teach because of the traditional and friendly conception, which MATLAB is the best way from a long time ago. Most serious situation was that there was no S/W to teach and learn in control engineering before Python. When we want to use it, we have to pay so much money because MATLAB is basically commercial based S/W. Developed country or rich institute can support through site license as academic version. However, under-developing country or institute can't ready to provide. It means we must prepare alternative way.

I have been teaching control engineering using MATLAB from DOS version. There was no problem because National University support license in the site as Academy version. However, when I taught from April 2019 at ASTU (Adama Science and Technology University, Ethiopia), situation was serious. First time, I tried to teach control engineering by MATLAB as the previous way as I did in other university. But almost MATLAB module was missing in LAB PC and every PC had different version and different installation.

Of course, it was not working well. So, I have to find alternative method to teach for implementation of control engineering. I am sure other under-developing countries will be the similar situation because University cannot support expensive S/W. So, teaching is not easy.

Herein, I like to strongly suggest that you had better Python or so if you cannot have regular S/W like me.

This paper is also to provide on how we can teach control system in Python in under course and graduate course through step by step of my experience and like to share experience and develop teaching skill with others.

Manuscript received on February 08, 2020.

Revised Manuscript received on February 10, 2021.

Manuscript published on March 30, 2021.

* **Dong Hwa Kim**, Electrical power control engineering, Adama Science and Technology University, Ethiopia. Email: worldhucare@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

This teaching skill is important as much as technology development.

II. WHY WE HAVE TO USE PYTHON!

We cannot mention which S/W is better for us because each person's programming style is different. However, at least we can say this one is better than others as professor within teaching and the situation of the University (Table 1).

Table I: Python and MATLAB comparison [1-5, 13-15]

- Item	- Python	- MATLAB
- Basic concept	<ul style="list-style-type: none"> - Numeric arrays and data type - General-purpose - Free of charge based 	<ul style="list-style-type: none"> - Matrix oriented languages - Technical and Professional purpose computing - Commercial based
- Usage	<ul style="list-style-type: none"> - Web based programming - (Google Engine, etc.) 	<ul style="list-style-type: none"> - User interfaces in Computer
- Benefits	<ul style="list-style-type: none"> - All libraries Open source - Community development. 	<ul style="list-style-type: none"> - Closed source - Developer based
- Performance	<ul style="list-style-type: none"> - High-performance linear algebra, graphics, and statistics - Optimized library calls 	<ul style="list-style-type: none"> - Developer oriented program - Installing, compiling, validating for improved performance
- Academics	<ul style="list-style-type: none"> - The Python software foundation developed in the year 1991. 	<ul style="list-style-type: none"> - Basic version is in the market since the 1970s.
- Library	<ul style="list-style-type: none"> - It consists of an extensive standard library 	<ul style="list-style-type: none"> - The standard library does not contain generic programming functionality.
- Real-time Support	<ul style="list-style-type: none"> - Personalized email and phone support 	<ul style="list-style-type: none"> - No personalized real-time support
- Embedded Code	<ul style="list-style-type: none"> - No comprehensive, Automatic code generation for embedded systems. 	<ul style="list-style-type: none"> - MATLAB code generates readable, portable c and c++ code.

MATLAB can be a good S/W for some areas or developer. However, in the side of teaching through experience, Python has much better advantage as:

- MATLAB is the commercial software that is employed in research and industry. It means that is business-based S/W. So, there are many limitations to access as students.
- We must access in the particular way for research material data and commands because they have commercial based S/W. Currently almost code tend to be more opened, but they are less open. Therefore, it is a very difficult to teach and learn in developing country's university unless university's support.
- Python is a high-level language and very similar to MATLAB. It means student and researcher can access and transfer easily.
- Python coding style is interpretation and an interactive prompt. Students can follow easily coding process without in-built complex numbers.

- Python is easier to make a learn and teaching using a coding for lecture.
- The best thing is open source. So, students can access easily as far as there is an Internet. With this free program, students can be a good programmer and have a motivation to study. It allows them to build start up very easily and quickly.
- Python has variety communities to share, study, solve problem. We can teach well without hesitation because of S/W in University.

III. IMPORTANCE ON LECTURE METHOD OF PID CONTROL

Basically, many understand the PID controller because the PID Controller's theory is quite simple and has a long history. But its tuning method is not easy. Therefore, many tuning methods have been developed from a long time ago [16-31]. Currently, AI (Artificial Intelligence) methods are interested in many engineer and experts. Figure 2 shows the response of PID controller depending on the value of P, P+I, and P+I+D [19, 21].

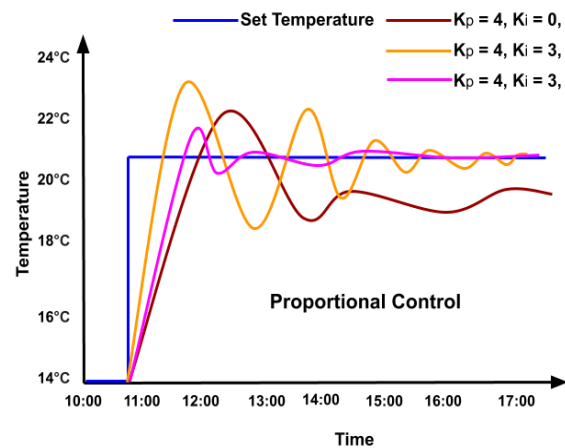


Figure 1 (a). The response of plant by the value of P, P+I, P+I+D

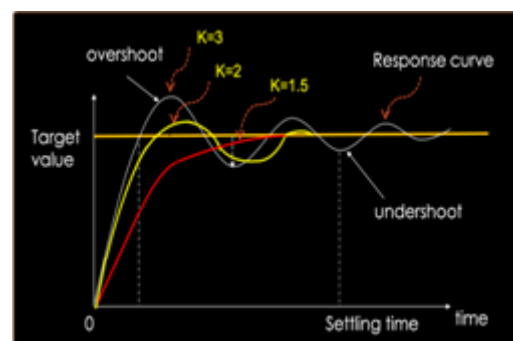


Figure 1 (b). The response of plant depending on P-value

As we can see from Figure 1, the tuning of PID controller gives an influence on the response of plant. Most of the PID tuning rules developed in the past years use the conventional method such as frequency-response methods [4], try and error, and etc. [18, 25].

The traditional method needs a highly technical experience to apply since they provide simple tuning formulae to determine the PID controller parameters. In case of the Ziegler-Nichols rule tuning technique [5, 35], it also needs try and error. Therefore, it leads to a rather oscillatory response for tuning (Figure 2 (a)). And also, we have to decide our control target between set-point control or load following control as Figure 2 (b).

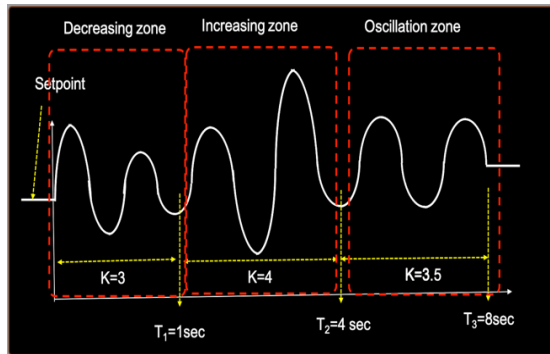


Figure 2 (a). Limited cycle Tuning (Lecture ppt)

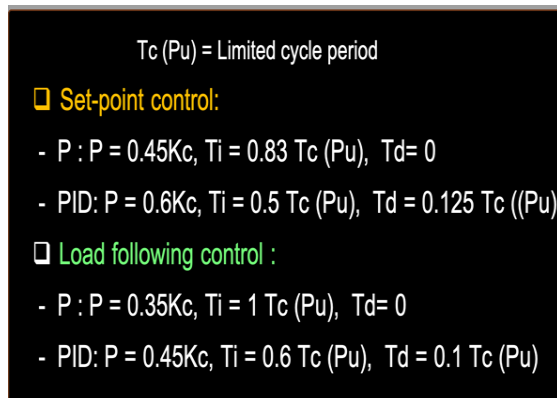


Figure 2 (b). Ziegler-Nichols Tuning value (Lecture ppt).

Figure 2 illustrates that the engineers need to research and experience for a highly tuning technology. Especially, to improve the performance of PID controller tuning for system with dynamically changing properties, many experiences are needed and this is why current AI tuning methods are interested in the name of adaptive PID or intelligent controller [27-31]. However, in many cases, the PID controller parameters are still computed using the classic tuning formulae because engineers in the site cannot understand well tuning method. Therefore, an intelligent tuning method is very important in PID controller for robust control with disturbance rejection function in control loop. Therefore, we have to teach design and application method through step by step for case by case using S/W. That is, we should share our teaching skill and develop advanced teaching method.

IV. ADVANCED ECTURE OF PID CONTROL FOR NONLINEAR SYSTEM IN PYTHON

A. PID Controller Theory and Lecture

First, lecture introduces basic PID control theory into student as given in Equation 1 and Figure 3 and 4. Lecture explain PID content on why it is important and how they use in industrial to give them motivation. In my experience,

almost student do not understand on how they implement and why they study. Especially, students do not know the conception of the role of I-function, P-band. Before PID, lecturer had better lecture I-function, D-function with graph response as Figure 5 (P-function). A mathematical formulation of PID control system is given as shown in Figure 4 and equation (1).

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1a)$$

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \right) \quad (1b)$$

$$u(t) = e(t) \left(K_p + \frac{1}{sT_i} + sT_d \right) \quad (1c)$$

Figure 3 and 4 shows PID controller structure and its transfer function (Lecture ppt). Whenever I had a lecture, almost students could not understand block-diagram of bottom of Figure 3. How do they express and what is the real plant. So, lecturer had better explain on how we can match between block-diagram and real plant. In my case, I use this motor control system and block-diagram because they can understand motor.

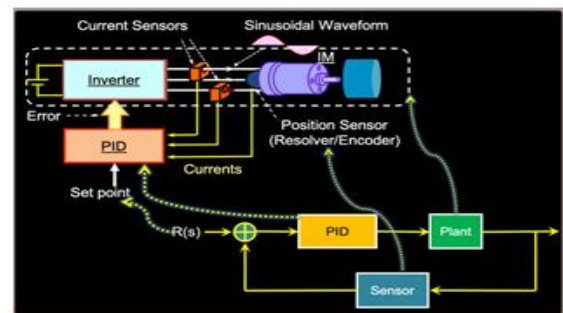


Figure 3. The block-diagram of PID controller and plan (Lecture ppt)

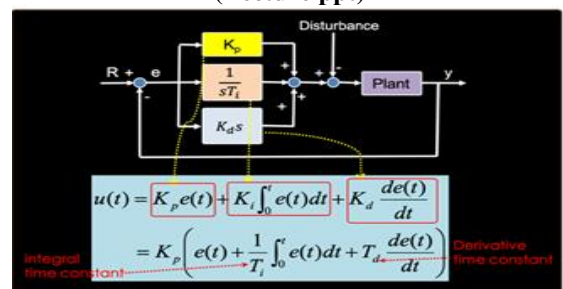


Figure 4. The structure of PID controller and transfer function (Lecture ppt)

B. PD controller

Lecture must explain the role of D-function because students do not understand and cannot apply PI or PD controller. Lecture also should have a lecture about PD controller. Usually, PD controller can be used in control of moving objects such as flying and underwater vehicles, ships, rockets, robot, and etc.

But it cannot be used in the system such as steam process, flow control system, level control because large disturbances and noise are present during operation of the process by D-function.

At this point lecture must explain why the effect of PD controller can stabilize quickly by sudden changes in moving system such a robot system.

C. PI controller

PI controller is used in many processes such as boiler level, steam process, flow process, etc. However, it cannot be used in robot and mechanical system, which is required the fast response. Thermal processes generally use derivative mode because thermal has slow response characteristics by a good thermal insulation. That is, integral mode would only already slow dynamics make more slowly because such processes have delayed dynamics with large inertia.

D. Proportional Band

Figure 5 shows the output signal of PID control. Lecture must explain on how this output signal can be produced for student (Lecture ppt). Also, they should explain proportional band because many industrial controllers in the site have defined a proportional band (PB) instead of gain,

$$PB = \frac{100}{K} [\%] \quad (2)$$

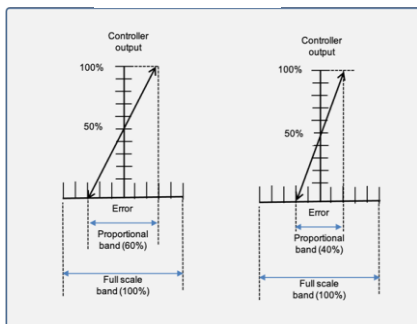


Figure 5.

P-band 60[%] and 40[%].

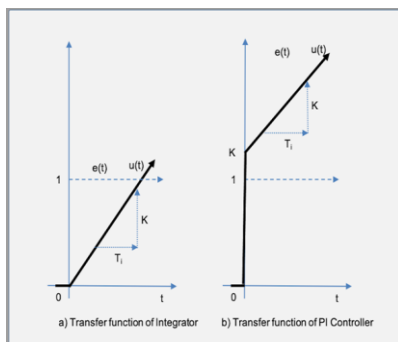


Figure 6. Transfer functions of integrator

Usually, a proportional band is equal $PB = 100\%$. Graphically characteristic of P controller is given in Figure 5 and Figure 6.

Figure 5 illustrates well band's concepts with 60% and 40%. If the controller has a narrow band, the output signal of controller increases faster than wide band. That is an important for controller's tuning when it is combined with I, D, PI, PD, PID controller.

Usually, students cannot understand about proportional band and output through my teaching experience.

E. PID controller

PID controller can be expressed by equation 1 and equation of the Figure 4 (lecture ppt material). This controller is used in the almost industrial system except typical areas described in the PI and PD. This paper focuses on this area.

F. Lecture on Importance of Tuning Technology

PID controller tuning is main topic among the PID controller theory. There are so many tuning methods to systems [16-31]. The response of PID controller is quite different from tuning method as shown in Figure 1 and Figure 2 (Lecture ppt) as shown before. This paper focuses on chemical nonlinear system as lecture example. Currently, many are interested in AI tuning methods such as fuzzy, neural network, deep learning, PSO, etc. (Figure 11).

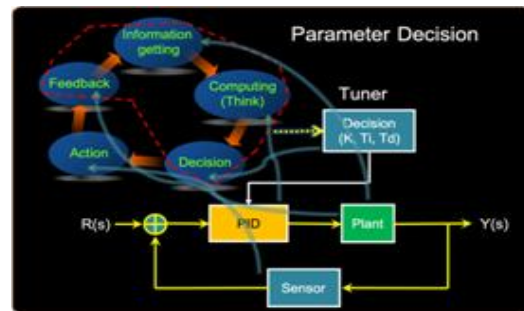


Figure 7. Tuning process of PID controller (Lecture ppt)

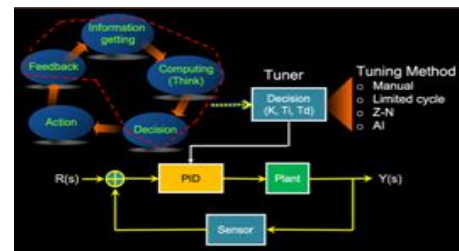


Figure 8. Tuning method of PID controller (Lecture ppt)

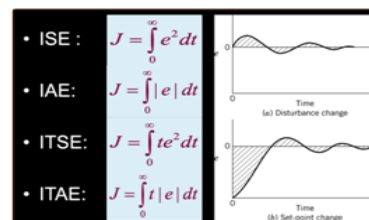


Figure 9. Performance index

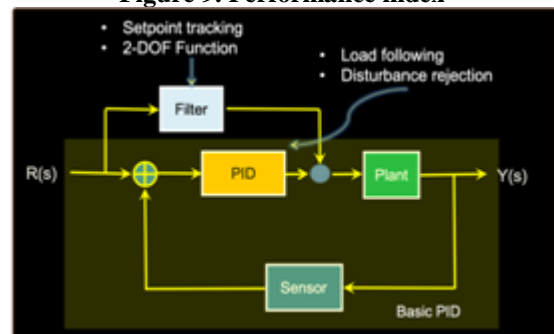


Figure 10. 2-DOF-PID controller (Lecture ppt)

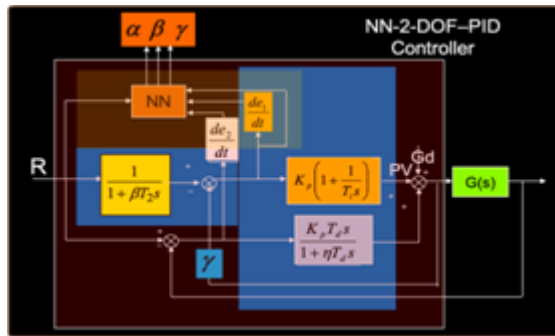


Figure 11. Tuning of 2-DOF-PID controller (Lecture ppt)
G. Lecture on 2-DOF PID controller

Many cases, they can touch the basic PID controller. However, few cases are explained 2-DOF PID controller because they think it is enough to teach as the basic PID theory [27]. However, as we can see from figure 2 (b), tuning target is quite different depending on set point tracking and load following. For this recovering, we use 2-DOF PID controller. To teach effectively, lecture introduces this control theory and tuning method. There are several types in 2-DOF PID controller. But this paper shows only Figure 10 as an example.

H. Lecture on Performance Index

The response of the PID controller is quite different from performance index (Figure 10, lecture ppt). Lecture should introduce the meaning of performance index and on how it is different in response. Of course, at this point, if lecturer can show in simulation result or ppt, students can understand well.

V. ADVANCED LECTURE ON PID CONTROL IN PYTHON

A. Lecture on Implementation of linear System by Python

The implementation of PID can be expressed in many ways (MATLAB, C++, C, Labview, etc.) because PID controller has a long history. In this paper, we focus on Python because there are many advantages in engineering and price as I already describe.

B. Linear System Response

Before lecture, lecturer has to introduce general linear system as an example because almost student do not understand concept of nonlinear system and relation between linear system and nonlinear system.

C. Initial installation

Even students installed already, they should operate install again for safety because the [Python Control Systems Library](#) is do not provide unfortunately, a standard part of most standard Python distributions. On most systems, the following commands should perform the required one-time installation of the necessary software. And then students can obtain as Figure 12 [1-4].

D. Library installation

For the safety of operation, all students should install library (Figure 13). Of course, when you operate simple code, you can do it well. However, if you want to operate control area, you have to install. There are some errors without installation of this library [1-4].

E. Demo-1 (1st Order System)

This paper aim is to provide experience on how we can teach well for student. So, to help understanding of students,

lecture had better show demo of first order system as Figure 14.

F. Demo-2 (2nd Order System)

Lecture show Python command and response of 2nd order system as shown in Figure 15. This demo main aim is to show system and command of Python.

```
[1] !pip install slycot
    !pip install control
```

```
!pip install slycot
!pip install control

Collecting slycot
  Downloading
  https://files.pythonhosted.org/packages
  /85/21/4e7110462f3529b2fbcff8a519b61bf6
  4e0604b8fcb9a07649c9bed9d7a/slycot-0.4
  .0.0.tar.gz (1.5MB)
  1.6MB 5.3MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ...
  done
  Preparing wheel metadata ... done
Requirement already satisfied: numpy in
  /usr/local/lib/python3.6/dist-packages
  (from slycot) (1.19.5)
Building wheels for collected packages:
  slycot
  Building wheel for slycot (PEP 517) ...
  done
  Created wheel for slycot:
  filename=slycot-0.4.0-cp36-cp36m-linux_
  x86_64.whl size=1417832
  sha256=493d3ae3d1c6c2152f3e03b73805ee4d
  a1376743519580e09951cc83e2626bfb8
  Stored in directory:
  /root/.cache/pip/wheels/a2/46/56/f82cbb
  2fd06556f4f3952a2eb2396e8fd29264ffecba
  ad3cf
  Successfully built slycot
Installing collected packages: slycot
Successfully installed slycot-0.4.0
Collecting control
  Downloading
  https://files.pythonhosted.org/packages
  /e9/7f/e903340dde3ad2be21578828a23d80b4
  7f4163443310882b05da96fba5f4/control-0.
  8.4.tar.gz (261kB)
  266kB 4.2MB/s
Requirement already satisfied: numpy in
  /usr/local/lib/python3.6/dist-packages
  (from control) (1.19.5)
Requirement already satisfied: scipy in
  /usr/local/lib/python3.6/dist-packages
  (from control) (1.4.1)
Requirement already satisfied: matplotlib
  in
```

Figure 12. Message of installation

Collecting slycot

Downloading

<https://files.pythonhosted.org/packages/85/21/4e7110462f3529b2fbcff8a519b61bf64e0604b8fcbe9a07649c9bed9d7a/slycot-0.4.0.0.tar.gz> (1.5MB)

1.6MB 8.7MB/s

Installing build dependencies ... done

Getting requirements to build wheel ... done

Preparing wheel metadata ... done

Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from slycot) (1.19.5)

Building wheels for collected packages: slycot

Building wheel for slycot (PEP 517) ... done

Created wheel for slycot:

filename=slycot-0.4.0-cp36-cp36m-linux_x86_64.whl
size=1417833
sha256=a02e5a28ea33fdf4e44cdacdef30bcf3d3e5a
a704a52557aa9b6e4cad5af94b9

Stored in directory:

/root/.cache/pip/wheels/a2/46/56/f82cbb2fd06556f4f3
952a2eb2396e8fd29264ffecbaad3cf

Successfully built slycot

Installing collected packages: slycot

Successfully installed slycot-0.4.0

Collecting control

Downloading

<https://files.pythonhosted.org/packages/e9/7f/e903340dde3ad2be21578828a23d80b47f4163443310882b05da96fba5f4/control-0.8.4.tar.gz> (261kB)

266kB 8.5MB/s

Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from control) (1.19.5)

Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from control) (1.4.1)

Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-packages (from control) (3.2.2)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->control) (1.3.1)

Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->control) (2.8.1)

Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-packages (from matplotlib->control) (0.10.0)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->control) (2.4.7)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from python-dateutil->matplotlib->control) (1.15.0)

Building wheels for collected packages: control

Building wheel for control (setup.py) ... done

Created wheel for control:

filename=control-0.8.4-py2.py3-none-any.whl
size=276947
sha256=f40cda1730919ba423872ead53e65f4356fad
5f3ee024d219b8ee6e17fc26a9d

Stored in directory:

Figure 13: Library installation

```
G = control.tf([5.3],[4., 1])
print(G)
```

$$\frac{5.3}{4s + 1}$$

```
y,t = control.step(G)
plt.plot(t,y)
plt.xlabel('Time')
plt.title('Step Response')
```

Figure 14 (a). Python code for transfer function command (1st order)

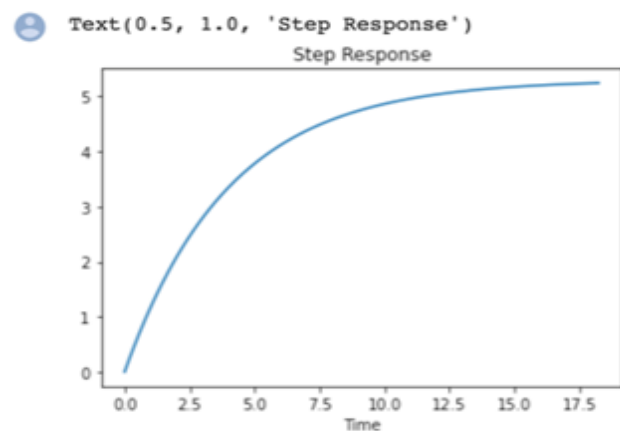


Figure 14 (b). Python code for transfer function command (1st order)

$$G(s) = \frac{3}{4s^2 + s + 1}$$

```
R = control.tf([5.],[5.,1.,1.])
print(R)

y,t = control.step(R)
plt.plot(t,y)
plt.xlabel('Time')
plt.title('Response of a Second Order System')
```

Figure 15 (a). Python code for 2nd Transfer function command

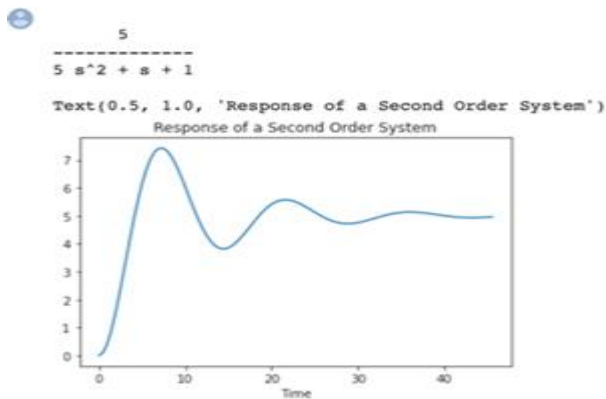


Figure 15 (b). Step response of 2nd order transfer function by Python

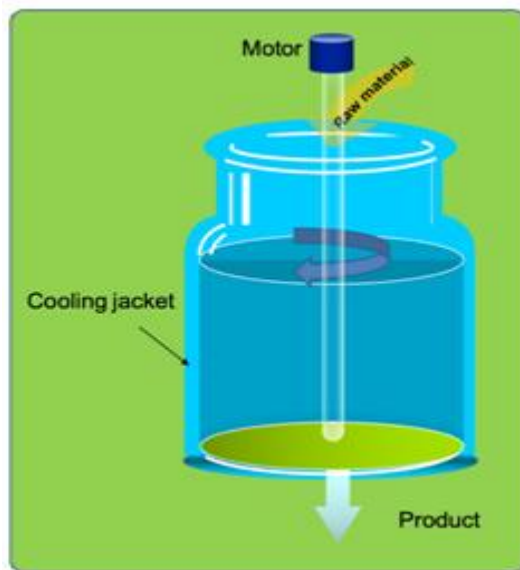


Figure 16. Nonlinear system of cooling jacket

G. Advanced Lecture of Nonlinear System by Python

1) Nonlinear Dynamic Equation of Cooling Jacket

Table1 design parameter of cooling jacket

Quantity	Symbol	Value	Units
Activation Energy	E_a	72,750	J/gmol
Arrhenius pre-exponential	k_0	7.2×10^{10}	1/min
Gas Constant	R	8.314	J/gmol/K
Reactor Volume	V	100	liters
Density	ρ	1000	g/liter
Heat Capacity	C_p	0.239	J/g/K
Enthalpy of Reaction		-50,000	J/gmol
Heat Transfer Coefficient	UA	50,000	J/min/K
Feed flowrate	q	100	liters/min
Feed concentration	C_a	1	gmol/liter
Feed temperature	T_f	350	K
Initial concentration	$c_{A,0}$	0.5	gmol/liter

Initial temperature	T_0	350	K
Coolant temperature	T_c	300	K

Figure 16 shows the block-diagram of cooling jacket used for control simulation of nonlinear system in this paper [4]. The temperature of this system is controlled by water flow rate and temperature of cooling water. Temperature level gives an influence on chemical reaction of jacket inside. Chemical material is important as products. Therefore, cooling water flow rate and temperature influence on product directly. That is, PID tuning gives an influence to produce quality. Lecture must illustrate this situation to all students, and then they cannot understand without any situation explanation. The parameter of this jacket is integrated below. To express PID controller's tuning effectively, Lecture explain dynamic equation. Lecture should illustrate dynamic equation simply because this lecture focuses on PID tuning.

The model of this jacket system can be expressed by energy balance of cooling water and chemical reactor system [26]. Therefore, energy balance equation can be expressed by equation 2 and 3.

$$\frac{dc_a}{dt} = q(c_{ai} - c_a) - Vkc_a \quad (2)$$

$$V\rho c_p \frac{dT}{dt} = \omega c_p (T_i - T) + (-\delta H_r Vkc_a + UA(T_c - T)) \quad (3)$$

And then this system should be normalized for controller design.

$$\frac{dc_a}{dt} = \frac{q}{V} (C_{ai} - C_a) - kC_a \quad (4)$$

$$\frac{dT}{dt} = \frac{q}{V} (T_i - T) + \frac{-\delta H_r}{\rho c_p} kC_a + \frac{UA}{V\rho c_p} (T_c - T) \quad (5)$$

This cooling jacket is to control chemical reaction through temperature of the cooling water jacket. Therefore, reaction results are different depending on cooling water speed and temperature. Figure 17 shows Python code and graph for the initial condition of jacket [2, 26].

2) PID control of Nonlinear System of Cooling Jacket in Python

Figure 18 illustrates concentration, temperature, and flow rate by P, I, D of PID controller. Figure 18 (a) is response of PID at $K=40$, $I=80$, $D=0$ and Figure 18 (b) shows response of PID at $K=50$, $I=70$, $D=0$. Figure 18 (c) is response of PID at $K=50$, $I=70$, $D=0$ and Figure 18 (d) presents response of PID at $K=50$, $I=70$, $D=5$. Figure 18 (e) is response of PID at $K=50$, $I=70$, $D=7$. From these figures, we can see impact of D-function is big than others. It means D-function increases abruptly in the noise process.

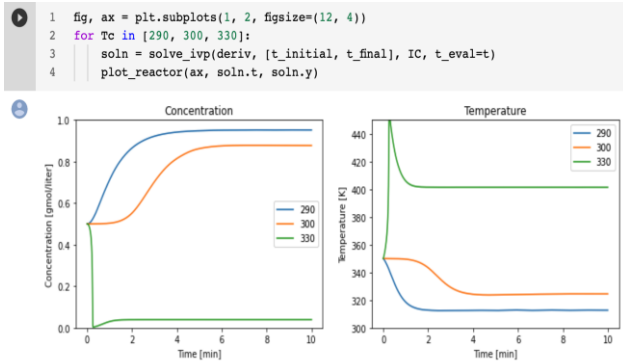


Figure 17. Initial condition of cooling jacket system [1-5]

Figure 19 shows tuning code of controller implemented in Python. We can see Python is a very simple for lecture and learn. Personally, I have been teaching for under course and graduate course in the university as control engineering since a long time ago. Of course, University had a license for 25 students in one class and there are many classes in engineering departments. And also, there was no choice to alternative method. However, currently, it is very difficult to use MATLAB in under developing country's University. There is no license and PC condition is not good for student.

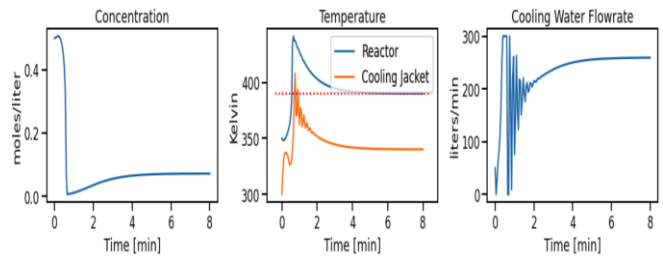


Figure 18 (d). Response of PID at K=50, I=70, D=5

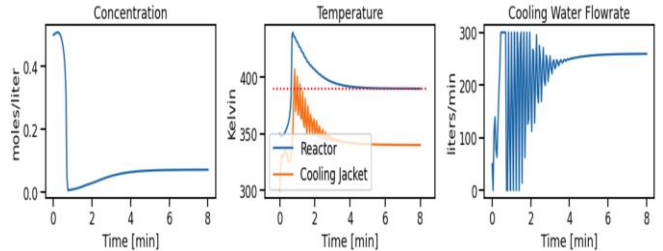


Figure 18 (e). Response of PID at K=50, I=70, D=7

Herein, I like to strongly recommend for change to Python to teach easily and convenient for student. That is a duty as professor.

VI. CONCLUSIONS

PID controller is still quite important because it is using industrial area widely and there are many material and references to use and tuning methods for many plant characteristics. So, it is important to have a lecture by using S/W. However, they use MATLAB S/W because of limited S/W. This paper focuses on how we can teach well implementation and tuning of PID controller using alternative S/W, Python. To teach well, information sharing about knowhow of teaching is important as much as technology development. So far, many universities and research centers have been using only MATLAB to teach and research because of the traditional way and familiar with that S/W.

However, the philosophy of S/W development is quite different as we can see from table 1. So, for the future, 4th industrial revolution, even primary students should code by themselves. However, it is very difficult for primary students by using MATLAB because this S/W focuses on high level engineering and commercial business.

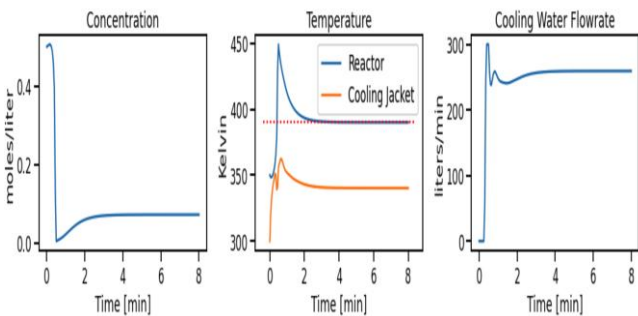


Figure 18 (a). Response of PID at K=40, I=80, D=0

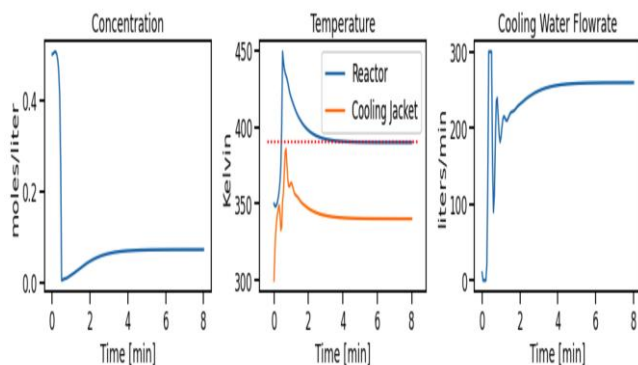


Figure 18 (b). Response of PID at K=50, I=70, D=0

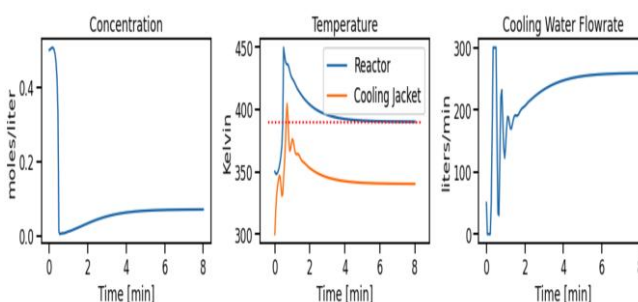


Figure 18 (c). Response of PID at K=50, I=70, D=0


```

1 # setpoint
2 Tsp = 390
3
4 # set initial conditions and cooling flow
5 IC = [C0,T0,Tcf]
6
7 # do simulation at fixed time steps dt
8 dt = 0.05
9 ti = 0.0
10 tf = 8.0
11
12 # control saturation
13 qc_min = 0 # minimum possible coolant flowrate
14 qc_max = 300 # maximum possible coolant flowrate
15 def sat(qc): # function to return feasible value of qc
16     return max(qc_min,min(qc_max,qc))
17
18 # control parameters
19 kp = 50
20 ki = 50
21 kd = 8
22 beta = 0
23 gamma = 0
24
25 # create python list to log results
26 log = []
27
28 # start simulation
29 c,T,Tc = IC
30 qc = 150
31
32 eP = beta*Tsp - T
33 eD = gamma*Tsp - T
34 eD_ = eD_
35
36 for t in np.linspace(ti,tf,int((tf-ti)/dt)+1):
37     # PID control calculations
38     eP = beta*Tsp - T
39     eI = Tsp - T
40     eD = gamma*Tsp - T
41     qc = kp*(eP - eP_) + ki*dt*eI + kd*(eD - 2*eD_ + eD_)/dt
42     qc = sat(qc)
43
44     # log data and update state
45     log.append([t,c,T,Tc,qc])
46     c,T,Tc = odeint(deriv,[c,T,Tc],[t,t+dt])[-1] # start at t, find state at t + dt
47
48     # save data for PID calculations
49     eD_,eD,eP_ = eD,eD,eP
50
51 qplot(log)

```

Figure 19. Python code for PID controller tuning of nonlinear control system

In case of Python, it was developed for general public purpose and free of charge use by foundation. Additionally, coding system is easy for young children. It means that primary students can code well and they do not need pay.

This philosophy is quite important for building of 4th industrial revolution infrastructure because manpower is the best solution for that.

REFERENCES

1. <https://www.educba.com/python-vs-matlab/>
2. <https://www.educba.com/software-development/courses/?source=footer>
3. Software Development Course - All in One Bundle (<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)
4. Become a Python Developer (<https://www.educba.com/software-development/courses/python-certi>)
5. Ziegler and Nichols, "Optimum setting for automatic controllers," Transaction ASME, Nov. pp. 759-768, 1942.
6. C. C. Hang and K. J. Astrom, "Refinements of the Ziegler-Nichols tuning formula," Proc. Inst. Elect. Eng., Vol. 138, pt. D, pp. 111-118, 1991.
7. Download Anaconda. [Online]. Available: <http://continuum.io/downloads>
8. Obtaining NumPy and SciPy libraries. [Online]. Available: <http://www.scipy.org/scipylib/download.html>
9. Python Control toolbox. [Online]. Available: <https://github.com/python-control/python-control>
10. NumPy for Matlab Users. [Online]. Available: [http://wiki.scipy.org/NumPy for Matlab Users](http://wiki.scipy.org/NumPy%20for%20Matlab%20Users)

11. Dong Hwa Kim, "Experimental Research of Intelligent Multivariable 2-DOF PID Control System for DCS," International Journal of Systems Applications, Engineering & Development, July 2013, pp. 148-157.
12. Dong Hwa Kim, "Getachew Teshome", Dawit Dubela**, Yosef Dentamo**, Hinsermu Alemayehu, "Optimal Conversion of DC-DC Converter Considered Optimal Switching Time and Optimal Switching Mode of PWM by Fuzzy Based PID Tuning", IJITEE, pp. 1-6, March 2020.
13. Dong Hwa Kim, "A Study on Improving Lecture Skill and Implementation of Anti-reset and Bampless Using 2-DOF-PID Controller and Python IARJSET, Vol. 7, Issue 10, October 2020, 2394-1588.
14. Dong Hwa Kim, Hinsermu, "A study on Teaching Method of Control Engineering by Using Python Based PID", IARJSET, Vol. 7, Issue 9, September 2020, 2394-1588.
15. Dong Hwa Kim, "Advanced Lecture Skill of Fuzzy Control in Python", IARJSET, Vol. 7, Issue 10, October 2020, 2394-1588.
16. C. H. Lee and C. C. Teng, "A Novel Robust PID Controllers Design by Fuzzy Neural Network," Asian Journal of Control, Vol. 4, No. 4, pp. 433-438, 2002.
17. [J. X. Xu, Y. M. Pok, C. Liu, and C. C. Hang, "Tuning and Analysis of a Fuzzy PI Controller Based on Gain and Phase Margins," IEEE Trans. on Systems, Man, and Cybernetics- Part A: Systems and Humans, Vol. 28, No. 5, pp. 685-691, 1998.
18. Kraus, T.W., & Mayron, T.J., "Self-tuning PID controllers based on a pattern recognition approach," Control Engineering Practice, 106-111, 1984.
19. Zhuang, M. and D. P. Atherton, "Automatic tuning of optimum PID controllers," IEE Proc. Part D, 14, 216-224, 1993.
20. C. H. Lee and C. C. Teng, "Tuning PID Controller of Unstable Processes: A Fuzzy Neural Network Approach," Fuzzy Sets and Systems, Vol. 128, No.1, pp. 95-106, 2002.
21. Dong Hwa Kim, "Tuning of a PID controller using an artificial immune network model and fuzzy set" IFSA, July 28, Vancouver, 1998.
22. S. Matsumura, Adaptive control for the steam temperature of thermal power plants," Proceedings the 1993 IEEE on Control applications," PP. 1105 - 1109, Sept. 1998.
23. Teng Fong-Chwee, "Self-tuning PID controllers dor dead time process," IEEE Trans., Vol. 35, No. 1, pp. 119-125, 1988.
24. Ya-Gang Wang, "PI tuning for processes with dead time," AACC2000, Chicago, Illinois, June 2000.
25. W. K. Ho, "PID tuning for unstable process based on gain and phase-margin specifications," IEE Proc. Control Theory Appl. vol. 45, no. 5, pp. 392-396, Sept. 1998.
26. Nichols, "Instrumentation for process flow engineering," Technonics publishing company, 1987.
27. Dong Hwa Kim, "Tuning of 2 - DOF PID controller by immune algorithm," IEEE international conference on evolutionary computation, Hawaii, May 12 - 17, 2002.
28. Dong Hwa Kim, "Auto-tuning of reference model based PID controller using immune algorithm," IEEE international conference on evolutionary computation, Hawaii, May 12 - 17, 2002.
29. Dong Hwa Kim, "Comparison of PID Controller Tuning of Power Plant Using Immune and genetic algorithm. Measurements and Applications," Ligano, Switzerland, 29-31 July 2003.
30. Dong Hwa Kim, "Robust PID controller tuning multiobjective optimization based on clonal selection of immune algorithm," Proc. Int. Conf. Knowledge-based intelligent information and engineering systems. Springer-Verlag. pp. 50-56, 2004.
31. Zwi-Lee Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System", IEEE Trans. Energy Con. Vol. 19, pp. 384-391, No. 2, June 2004.
32. MATLAB TOOL BOX Manual.
33. B. Stuart, "Development of PID controller," IEEE control systems, vol. pp. 58-62, Dec.1993.
34. Y. Stephen, "A laboratory course on fuzzy control," IEEE Trans. on Education, vol. 42, no. 1, pp. 15-21, May 1998.
35. https://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method

AUTHORS PROFILE



Dong Hwa Kim Ph.D: Dept. of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering (AI Application for Automatic control), TIT (Tokyo Institute of Technology), Tokyo, Japan. He was the Hanbat National University (Dean, Prof., S. Korea), He has experience in many University, overseas as Prof. He was NCP of EU-FP7 (EU-Framework Program, ICT). He had keynote speak at several international conference and

University. He has 200 papers in Journal and conferences. He was editor of IJCIR (International Journal of Computational Intelligence) and He is reviewing IEEE and other's Journal. He is current Prof. at Electrical Power and Control Eng. Adama Science and Tech. Uni., Ethiopia (<http://worldhumancare.wixsite.kimsite>).

- Home page: www.worldhumancare.wixsite.com/kimsite
- Contact: worldhucare@gmail.com, 82-10-4899-1170
- Research citations: https://www.researchgate.net/profile/Dong_Kim53