

# A study on Teaching Method of Control Engineering by Using Python Based PID

Dong Hwa Kim<sup>1</sup>, Hinsermu Alemayehu<sup>2</sup>

Professor, EPCE, Adama Science and Technology University, Adama, Ethiopia<sup>1</sup>

Header of Office, EPCE, Adama Science and Technology University, Adama, Ethiopia<sup>2</sup>

**Abstract:** PID controller is so popular control engineering and it has been using in the industrial fields widely. So, it is important to have a lecture for design and theory. This paper deals with design and lecture skill through implementation of PID controller by using Python. So far, many universities and research centres have been using MATLAB to teach and learn in control engineering. However, MATLAB is commercial S/W and expensive. Therefore, developing country's universities have a very difficult because of no license. Currently, many are interesting in Python because it is open source and huge communities. It is a good tool to teach and learn as teacher and students. This paper provides on how we can teach and learn control engineering through implementation of PID controller.

**Keywords:** Python, PID Control, Control, Control Lecture.

## I. INTRODUCTION

A Proportional – Integral – Derivative (PID) controller has been widely used in the most industrial processes despite continual advances in control theory. This is not only due to the simple structure which is theoretically easy to understand but also to the fact that the tuning technique provides adequate performance in the vast majority of applications.

However, almost lecture is doing for theory and the simplified design method due to the limited S/W and design language like commercial S/W, MATLAB.

Currently, many engineers and scientists have been interesting in Python for control engineering. Python S/W was released in 1991 and have many communities to research. MATLAB S/W also has many communities for themselves. However, there are too much differences between two communities, and licenses. Also, a lot of research models and materials are written in MATLAB so far. However, MATLAB is a programming language which requires an expensive license and they do not open their theory code.

Many researchers in and outside universities can't afford this licensed S/W, which means that models are not available for students although lots of people would like to work with these MATLAB of models. Computer Aided Control System Design (CACSD) environments play a key role in teaching Control Systems in our universities. Since the late 1990, MATLAB and Simulink have been chosen as the reference CACSD for control system design. Lectures and laboratories take advantage of all the features in design and simulation offered by these products. On the other side, many universities and few companies have been purchasing these expensive licenses related to the different toolboxes required for a complete design environment. In addition, universities have to buy this package licenses of MATLAB products for their education even under course and applications in applied research of graduate course.

There are free alternatives downloadable S/W on the web, but nothing has the same quality as the cited products. However, currently Python is glowing up as one of the most promising tools since 1991. This S/W presents well a new approach to control systems based on Python and a set of additional functions for controlling real applications. In order to apply rapid prototyping methods in control design, we have to introduce this tool into lecture course from under course intensively. We can lecture well material represented by the Python Control System toolbox, which is initially developed at Caltech by Richard Murray. Most important things are that this S/W is free and there are many communities. Also, it is easy to build control theory with this language.

Most of the commands implemented in the Python Control toolbox are related to the book from Richard Murray and Carl Astrom "Feedback Systems" [4, 6, 7]. We can present and teach many control activities that can be performed using the Python packages and the Python Control toolbox. Also, Python S/W can be well presented under Windows, Mac and Linux.

Unfortunately, some professor or so do not still know on how they can approach and teach because of the traditional conception, which is MATLAB is the best and friendly from a long time ago.

This paper's aim is to provide on how we can teach control system in Python in Under course and graduate course.

I hope this material can be used usefully in the University and teachers.

## **II. PROBLEMS OF THE TRADITIONAL TOOL IN LECTURE**

### **A. Importance of Lecture for PID controller**

The basic theory of PID controller is so simple but it has been using since a long time ago. However, it cannot effectively control such a complicated or fast running system such as motor control system, since the response of a plant depends on only the three parameters (P, I, and D) and its gain has to be manually tuned by trial and error in the industrial world. Most of the PID tuning rules developed in the past years use the conventional method such as frequency-response methods [4]. This 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, it often leads to a rather oscillatory response to set-point changes as the following system [4, 7]:

- (a) Plant which the system has non-linearities such as directionally dependent actuator and plant dynamics
- (b) System that various uncertainties, such as modeling error and external disturbances, are involved in the system.

As a result of these difficulties, the PID controllers are rarely tuned optimally and the engineers need to research for a highly tuning technology. It means we have to teach application skill by case by case for students.

To improve the performance of PID controller tuning for processes with changing dynamic properties, several tuning strategies for PID controller have been proposed, for example, automatic tuning PID, adaptive PID, and intelligent controller. These controllers have recalibration features to cope with little a priori knowledge and significant changes in the process dynamics [4].

However, the PID controller parameters are still computed using the classic tuning formulae and, as noted above, these do not provide good control performance in all situations, for example, for unstable systems with time delay. De Paor used the Modified Smith Predictors to cope with unstable and integrating process with long time delay.

In order to provide consistent, reliable, safe and optimal solution to industrial control problems as described above, many approaches for PID control schemes and tuning techniques have been presented. These schemes generally consist of four basic parts: model estimation, desired system specifications, optimal tuning mechanism and an online PID controller. The model estimation provides a parametric/non-parametric model for the process. The desired system specifications include a set of control requirements of the process [6]. An intelligent tuning method is also very important in PID controller for robust control with disturbance rejection function in control loop [21-33].

Therefore, we have to teach design and application method by step by step for case by case using S/W.

### **B. The problem of the traditional lecture**

After developing MATLAB, so many engineer and developer in engineering, they have been using this tool for their work. For that, they have to have license as package even they do not use many times. Also, personally, they cannot obtain tool because of high price.

The advanced country and big companies can support for employee but developing country's university cannot buy license because MATLAB TOOLBOX is too much expense. Personally, it is impossible to buy, which is 3,000\$ in 2020 for only Deep learning tool. Python is free as far as connecting Internet.

By making the first step in solving this problem, we must understand on how we can convert from MATLAB to Python. This conversion is done due to the fact that Python is open-source software which makes the model free to use for everyone.

Also, MATLAB and Python syntax (Table 1) is generally the same, which can make conversion easier. It means everyone convert the previous model designed by them to Python and ready to use it. In July 20, 2020, Python 3.8.5 was released after Python 1.4 on 25 October 1996. Therefore, they modified many problems and issues. Now, it is time we have to introduce into lecture and engineering area.

Some report that even they use version 1.0, it was found that the overall running time for version 1.0 was much larger than the original model in MATLAB. This is due to Python executing notebooks inside of one masterscript notebook, which is not very fast with Python's data structure. To solve this problem, Python model is created. Version 2.0 makes more effective use of Python's data structure and this has led to running times almost twice as fast as the original model in MATLAB.

Another benefit of conversion is that both model versions in Python are more user-friendly. Headers are possible in Python and by defining the data paths in the beginning of every masterscript. The plots created in Python are the same as in MATLAB and a more precise source and sink region can be defined at further research or be redefined.

## **III. PYTHON VS. MATLAB IN CONTROL ENGINEERING**

### **A. The differences between Python and MATLAB in Data**

Basically, Python is designed as a general-purpose language, not a numerical computing language like MATLAB. So, its philosophy and basic types are much more general focus.

TABLE I, PYTHON VS. MATLAB [2, 3].

Comparison	Python	MATLAB
Definition	Numeric arrays and data type (A high-level general-purpose programming language)	Math and Matrix oriented languages (MATLAB is the high-performance language for technical computing)
Usage	Python can be used for web programming (Zope, Google App Engine, and much more)	MATLAB allows matrix manipulations, plotting of functions and data, a creation of user interfaces
Benefit	Extensive support libraries. Open source and community development.	MATLAB allows you to test algorithms immediately without the act of compiling,
Performance	High-performance linear algebra, graphics, and statistics. Optimized library calls	Improved performance requires installing, compiling, validating, and adopting developer-oriented add-ons
Academics	It was developed by the Python software foundation in the year 1991.	MATLAB basic version is in the market since the 1970s.
Library	It consists of an extensive standard library	The standard library does not contain generic programming functionality.
Real-time support	Personalized email and phone support	No personalized real-time support
Embedded Code Generation	No comprehensive, Automatic code generation for embedded systems.	MATLAB code generates readable, portable c and c++ code.

It means everyone can understand to code and there are no arrays or matrices. Python is needed to address by the NumPy package, which provides multidimensional arrays. With NumPy, numerical computing can be calculated.

They are numbers: real, float, and complex; strings; lists and tuples, which are two types of ordered sequences; dictionaries, which are “associative arrays” or mappings; and sets, which are unordered collections of unique items.

The following list presents the types in more detail. The side notes next to each type show some code examples.

- Numbers are scalars. So, they are zero dimensional and do not have some shapes. This is quite different from MATLAB, which are 1-by-1 matrices. Example: real\_number = 1, float number = 1.0, complex number = 1 + 2j.

- Strings can be written with either single or double quotes and we cannot modify them. Instead of that, we can create new strings based on the contents of an existing one. Example: s1 = 'a string', s2 = "also a string", s3 = ""A (possibly) multiline string"", s3[0] returns 'A', s2[:4] returns 'also'.

Python	MATLAB
<pre> 1. import numpy as np 2. import matplotlib.pyplot as plt 3. 4. fs=[1, 2, 3] 5. all_time=np.linspace(0,2,200) 6. t=all_time[:100] 7. 8. for f in fs: 9.     y=np.sin(2*np.pi*f*t) 10.    plt.plot(t, y, label='%d Hz' % f, format(f)) 11. 12. plt.legend() 13. plt.savefig('basics_python.pdf') </pre>	<pre> 1. 2. 3. 4. fs=[1 2 3] 5. alltime=linspace(0,2,200); 6. t=alltime(1:100) 7. hold('on') 8. for f = fs 9.     y=sin(2*pi*f*t); 10.    plot(t, y, DisplayTime, sprintf('%d Hz', f)); 11. end 12. legend('show') 13. saveas(gcf, 'basics_matlab.pdf'); </pre>

Fig. 1 Comparison example of Python and MATLAB

- Lists are similar to cell arrays in MATLAB except they are only one dimensional. They can contain anything, including other lists. Even though they can contain items of different types at the same time. Example: word list = ['the', 'quick', 'brown', 'fox'], number list = [0, 1, 1, 2, 3, 5, 8].

- Tuples are lists' cousins. They are also ordered sequences. Example: point = (0, 0), also a point = 0, 0, a\_3d\_point = (0, 1, 2).

- They usually group together objects of different types, which are accessed via indexing.

- Dictionaries are similar to MATLAB structures.

- Sets are not used very often but allow for expressive “set operations”, such as intersection, union, difference, etc. Example: lights = {'red', 'yellow', 'green'}, choices = ['yes', 'no', 'yes'], unique choices = set(choices), unique\_choices is {'yes', 'no'} (Fig. 1).

Python is also one of the top coding languages. This language is required, or at least used, by the overwhelming majority of computer science courses in United States colleges and many more colleges all over the world. This means that learning Python is almost essential if one wishes to pursue any degree which requires some fundamental knowledge of coding and/or computer science practices, and especially for those who are looking to start a career in data analytics. Again, we can see the similarity between Python and MATLAB from Fig. 1.

**IV. LECTURE PROCEDURE OF PID CONTROLLER USING PYTHON**

**A. General Form of PID Controller and Theory**

Lecture introduces basic PID control theory into student as given in Equation 1, 2 and Fig.2. If possible, lecture explain PID content on why it is important and how they use in industrial are 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, lecture had better lecture I-function, D-function with graph response as Fig. 4 (P-function), Fig. 5 (the role of I-function). A typical mathematical formulation of PID Control system is given as shown in Fig. 2 and equation (1) and (2)

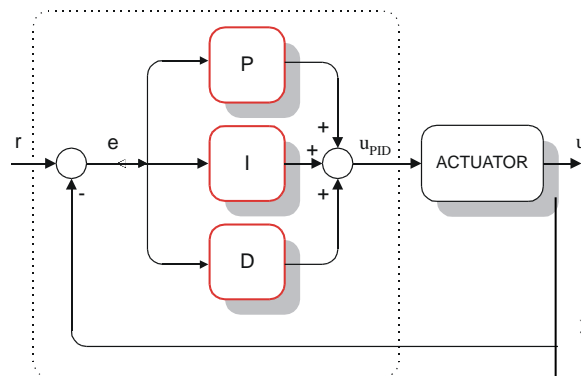


Fig. 2. The structure of PID controller.

$$u(t) = K \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \tag{1}$$

or

$$u(t) = Ke(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \tag{2}$$

where:

- $K_i = \frac{K}{T_i}$ : gain (reset) of integral part of the controller,
- $K_d = KT_d$ : gain of derivative part of the controller.

1) PD controller: We had better lecture where we can apply PI or PD controller because students do not understand why. Usually, PD controller can be used in control of moving objects such are flying and underwater vehicles, ships, rockets, robot, and etc. Because the effect of PD controller can stabilize quickly by sudden changes in targeting variable. Sometime, velocity control is used as sensor of heading change of moving object.

2) PI controller: PI controllers are the most often used in industry:

- Fast response of the system is not required
- Large disturbances and noise are present during operation of the process
- There is only one energy storage in process (capacitive or inductive)
- There are large transport delays in the system

Thermal processes with good thermal insulation act almost as integrators because insulation is good and thermal losses are small. In such processes, there is present process dynamics with large inertia since dynamics is slow, derivative mode is required for control of such processes. Integral mode would only already slow dynamics make more slowly.

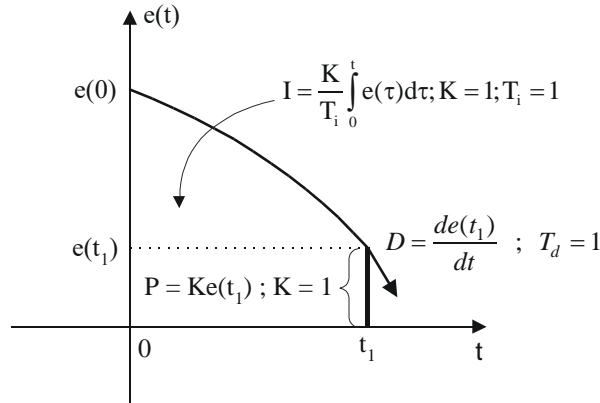


Fig. 3. PID controller control signal generation.

Fig.3 shows the output signal of PID control. Lecture must explain on how this output signal can be produced for student. Also, they should explain proportional band because many industrial controllers in the site have defined a proportional band (PB) instead of gain:

$$PB = \frac{100}{K} [\%] \tag{3}$$

Usually, a proportional band is equal  $PB = 100\%$ . Graphically characteristic of P controller is given in Fig. 4 (a) and (b). Fig. 4 (b) 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. That is, PI, PD, PID controller.

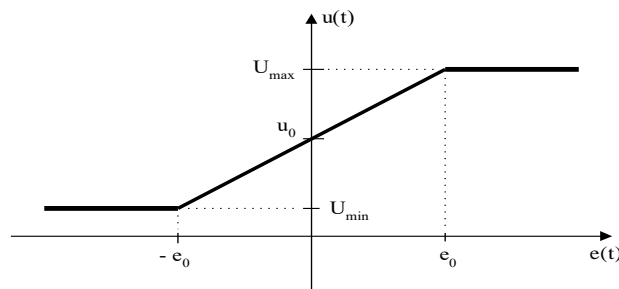


Fig. 4 (a). Graphic expression of P controller

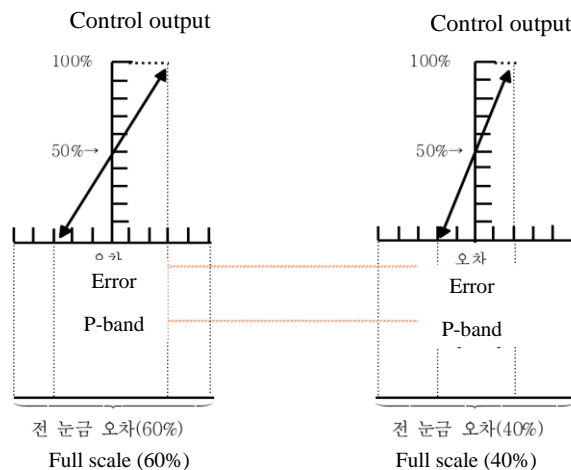


Fig. 4 (b). P-band 60[%] and 40[%]

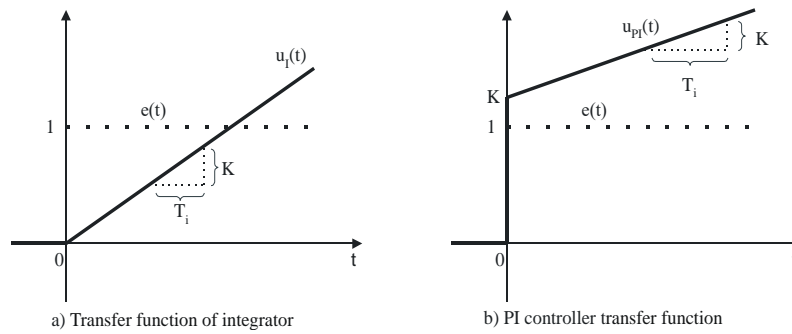


Fig. 5. Transfer functions of integrator and PI controller.

Usually, they do not use derivative mode in control system with many picks or disturbance like flow system. They never use on nuclear power plant level control system because derivative mode cannot eliminate control error. However, robot arm control system should use derivative mode reacts only on change of the controller input.

### B. Lecture Process of Basic PID Controller by Python

To implement in Python, this paper modified signal as

$$MV_k = MV_{bar} + K_p e_k + K_i \sum_{k=0}^n e_k (T_k - T_{k-1}) + \frac{e_k - e_{k-1}}{T_k - T_{k-1}} \quad (4)$$

$$e_k = SP_k - PV_k, \quad (5)$$

where  $e_k$  is the difference between setpoint and measured process variable.

Users download from Python web site and enter the code as Fig. 6. Fig. 6 is implemented in Colab (Python) directly. We can see this code looks like text. So, it is easier than MATLAB code. Also, it is free of charge and you can access, as open source, any times and any place as far as there is Internet. That is why I like to strongly candidate lecture and student use this Python. Additionally, there are many communities to solve and have a solution about problem while you are working. However, MATLAB do not open source and also, we can access good knowledge because of commercial policy.

```

def PID(Kp, Ki, Kd, MV_bar=0):
    # initialize stored data
    e_prev = 0
    t_prev = -100
    I = 0

    # initial control
    MV = MV_bar

    while True:
        # yield MV, wait for new t, PV, SP
        t, PV, SP = yield MV

        # PID calculations
        e = SP - PV

        P = Kp*e
        I = I + Ki*e*(t - t_prev)
        D = Kd*(e - e_prev)/(t - t_prev)

        MV = MV_bar + P + I + D

    # update stored data for next iteration
    e_prev = e
    t_prev = t
    
```

Fig. 6. Python code (Colab)of Basic PID controller.

### C. Setpoint Weighting Control

Equation (1), (2) show the basic PID controller and Equation (4), (5) are the modified equation for coding. In this section, we can express setpoint weighting control of PID in Python. As described previous, when we change derivative term in process control applications, the results of controller change the manipulated variable (action signal) and it may require sudden and abrupt changes in valve position, process flows, or pressures, all of which can cause significant strain on very

large devices. Usually, we call that this phenomenon is called setpoint kick (or derivative kick), which is generally to be avoided.

$$MV = MV_{bar} + K_p e_p + K_i \int_0^t e_i dt + K_d \frac{de_d}{dt} \quad (6)$$

$$e_p(t) = \alpha SP - PV \quad (7a)$$

$$e_i(t) = SP - PV \quad (7b)$$

$$e_d(t) = \beta SP - PV \quad (7c)$$

```
def PID(Kp, Ki, Kd, MV_bar=0, beta=1, gamma=0):  
    # initialize stored data  
    eD_prev = 0  
    t_prev = -100  
    I = 0  
  
    # initial control  
    MV = MV_bar  
  
    while True:  
        # yield MV, wait for new t, SP, PV  
        t, SP, PV = yield MV  
  
        # PID calculations  
        P = Kp*(beta*SP - PV)  
        I = I + Ki*(SP - PV)*(t - t_prev)  
        eD = gamma*SP - PV  
        D = Kd*(eD - eD_prev)/(t - t_prev)  
        MV = MV_bar + P + I + D  
  
        # update stored data for next iteration  
        eD_prev = eD  
        t_prev = t
```

Fig. 7. Python code (Colab) of automatic mode of PID controller.

For that, we can recover by using setpoint weighting. To this end, the PID control formula is written as Equation (6), (7). We can have the effect of a different error term for each term in the control equation as Equation (7).

The integral term offers the same condition as before, which has the function of gap reducing in order to force the process variable  $PV$  to be equal to the setpoint  $SP$  at steady-state. The error terms for the proportional and derivative terms, however, include constants  $\alpha$  and  $\beta$  which are used to modify the response to setpoint changes. Commonly, the tuning is to set  $\beta=0$  which entirely eliminates derivative action based on change in the setpoint. This tuning approach offer a good result in process control because it eliminates the 'derivative term' associated with a quick change in setpoint. The term  $\alpha$  is generally tuned to meet the specific application requirements of setpoint and setpoint tracking is not a high priority when setting  $\alpha=0$  is a reasonable starting point.

#### D. Implementation of Setpoint Weighting Control in Python

Fig. 8 shows controller's implemented code in Python. We can see Python is a very simple for lecture and learn as the beginner from this Fig. 8. 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 developing country's University. There is no license and PC condition is not good as student. Herein, I like to strongly candidate change to Python to teach easily and convenient for student. That is a duty as professor.

#### V. CONCLUSION

PID controller is still quite important because it is using industrial area widely and there are many 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 implementation of PID controller in Python (Colab). So many universities and research centers have been using only MATLAB to teach and research so far. Of course, MATLAB is a good tool for control engineering to teach and learn in University.

```
def PID(Kp, Ki, Kd, MV_bar=0, beta=1, gamma=0):  
    # initialize stored data  
    I = 0  
    eD = -100  
  
    # initial control  
    MV = MV_bar  
  
    while True:  
        # yield MV, wait for new t, SP, PV  
        t, PV, SP = yield MV  
  
        # PID calculations  
        P = Kp*(beta*SP - PV)  
        I = I + Ki*(SP - PV)*(t - t_prev)  
        eD = gamma*SP - PV  
        D = Kd*(eD - eD_prev)/(t - t_prev)  
        MV = MV_bar + P + I + D  
  
        # update stored data for next iteration  
        eD_prev = eD  
        t_prev = t
```

Fig. 8. Python code (Colab) of setpoint weighing control of PID controller.

However, Python has better advantage as:

- MATLAB is the commercial software that is employed in research and industry. It is the example of high-level scripting and also named as 4th generation language.
  - MATLAB is particular about the way that data must be read in and commands. This is a somewhat expected problem, as software. Currently almost code tend to be more open but is less open. Therefore, it is a very difficult to teach and learn in developing country's university except university's support.
  - Python is a high-level language and very similar to MATLAB
  - It is interpreted and has an interactive prompt. It allows dynamic typing and provides automatic memory management (and comes with 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. With this free program, students can be a good programmer and have a motivation to study. It allows them to start up very quickly.
  - Python has variety communities to share, study, solve problem.
- We can teach well without hesitation because of S/W in University.

## 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 formular," Proc. Inst. Elect. Eng., Vol. 138, pt. D, pp. 111-118, 1991.
- [7]. K. J. Anstrom and T. Haggglud, "Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica, Vol. 20, pp. 645-651, 1984.
- [8]. VirtualBox. [Online]. Available: <https://www.virtualbox.org>
- [9]. Download Anaconda. [Online]. Available: <http://continuum.io/downloads>
- [10]. Obtaining NumPy and SciPy libraries. [Online]. Available: <http://www.scipy.org/scipylib/download.html>
- [11]. Python Control toolbox. [Online]. Available: <https://github.com/python-control/python-control>
- [12]. NumPy for Matlab Users. [Online]. Available: [http://wiki.scipy.org/NumPy for Matlab Users](http://wiki.scipy.org/NumPy%20for%20Matlab%20Users)
- [13]. David J. Pine. Introduction to Python for Science. [Online]. Available: <https://github.com/djpine/pyman>
- [14]. Matplotlib. [Online]. Available: <http://matplotlib.org/>
- [15]. [Online]. Available: <http://docs.scipy.org/doc/scipy->
- [16]. SymPy Tutorial. [Online]. Available: <http://docs.sympy.org/dev/tutorial/index.html>



- [17]. Kane's Method in Physics/Mechanics. [Online]. Available: <http://docs.sympy.org/0.7.5/modules/physics/mechanics/kane.html>
- [18]. Kane's Method & Lagrange's Method (Docstrings). Available: [http://docs.sympy.org/latest/modules/physics/mechanics/api/kane\\_lagrange.html](http://docs.sympy.org/latest/modules/physics/mechanics/api/kane_lagrange.html)
- [19]. P. C. M. . T. R. Kane. Motion Variables Leading to Efficient Equations of Motions. [Online]. Available: [http://www2.mae.ufl.edu/fregly/PDFs/efficient\\_generalized\\_speeds.pdf](http://www2.mae.ufl.edu/fregly/PDFs/efficient_generalized_speeds.pdf).
- [20]. M.Bubak et al, A Comparison of C, MATLAB, & Python as Teaching Languages in Engineering, ICCS 2004, LNCS 3039, pp. 1210–1217, 2004.
- [21]. Dong Hwa Kim, "Robust tuning of PID controllers with disturbance rejection using bacterial foraging-based optimization," WSEAS Transaction on systems Vol. 3, No. 9, 2004, pp. 2834-2840.
- [22]. Dong Hwa Kim, Jae Hoon Cho, "Robust PID using Gain/Phase margin and advanced immune algorithm," WSEAS Transaction on systems Vol. 3, No. 9, 2004, pp. 2841-2851.
- [23]. Dong Hwa Kim, "Robust tuning of embedded intelligent PID controller for induction motor using bacterial foraging-based optimization (Best paper candidate)," Lecture Notes in Computer Science Proceeding of Springer (SCI) –LNCS, ICES2004, pp. 137-142, Dec. 9-10, 2004, Zhejiang University.
- [24]. Dong Hwa Kim, Jin Ill Park, "Intelligent Tuning of PID Controller For AVR System Using a Hybrid GA-PSO Approach," Lecture Notes in Computer Science Proceeding of Springer (SCI), May 12-15, 2005, Atlanta.
- [25]. Dong Hwa Kim, Jae Hoon Cho, "Adaptive Tuning of PID Controller For Multivariable System Using Bacterial Foraging Based Optimization," Lecture Notes in Computer Science Proceeding of Springer (SCI), June 12-15, 2005, Poland.
- [26]. Dong Hwa Kim, "Robust Intelligent Tuning of PID Controller for Multivariable System Using Clonal Selection and Fuzzy Logic," Lecture Notes in Computer Science Proceeding of Springer (SCI), September 12-15, 2005, KES 2005, Melbourne, Australia.
- [27]. Dong Hwa Kim, Jin Ill Park, "Intelligent PID Controller Tuning of AVR System Using GA and PSO," IEEE 2005 Intelligent computing, Lecture Notes in Computer Science Proceeding of Springer (SCI), Aug. 23-26, 2005.
- [28]. Dong Hwa Kim, "Decentralized PID Controller Tuning for Multivariable Process Using Multiobjective Optimization Based on Bacterial Foraging," IEEE 2005 Intelligent computing, Lecture Notes in Computer Science Proceeding of Springer (SCI), Aug. 23-26, 2005, China.
- [29]. Dong Hwa Kim, Jae Hoon Cho, "Robust Tuning of PID Controller Using Bacterial-Foraging-Based Optimization," JACIII, Vol. No. 6, pp. 2436, Sept. 22, 2005.
- [30]. Dong Hwa Kim, "Decentralised PID Controller Tuning for Multivariable Process Using Multiobjective Optimization Based on Bacterial Foraging," Dynamics of Continuous, Discrete & Impulsive Systems: Series B Applications & Algorithm (SCI), Vol. 14(SI), Jun, pp. 33-40, 2007.
- [31]. 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.
- [32]. Dong Hwa, Jae Hoon Cho, "Advanced Intelligence Tuning Using Hybrid of Clonal Selection and Genetic Algorithm, GM and PM", International Journal of Computational Intelligence and Applications, Vol. 14, No. 1, 2015.
- [33]. 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.

## BIOGRAPHIES



**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. 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). He is current Prof. at Electrical Power and Control Eng. Adama Science and Tech. Uni., Ethiopia (<http://worldhumancare.wixsite.kimsite>).



**Hinsermu Alemayehu** was born in Ethiopia, Fincha city in 1988. I received the B.Sc. in Electrical and Electronics Technology in from ASTU (Adama Science and Technology University) in 2008 and M.Sc degree in Control Engineering from Tianjin University of Technology and Education in 2011. He joined EPCE, ASTU. He is currently working as a lecturer and Department head. Main areas of research interest are control systems, robotics and automation and power electronics and drives. Currently he is a member of the Ethiopian Society of Electrical Engineers.