First Order Control System Using Python Technology

Palash Jain[1] and Jay Kumar Jain[2(✉)]

[1] Department of Electronics and Communication Engineering, SIRT, Bhopal, MP, India
[2] Department of Computer Applications, SIRT, Bhopal, MP, India
jayjain.research@gmail.com

Abstract. The research work highlights the application of python to practically study some basic concepts of control system engineering. The presented chapter consists of basic operations needed to perform in control system designing in python using libraries like Matplotlib, NumPy etc. Here the work presents computation and calculation of poles and zeros, Nyquist plot, bode plot, sinusoidal response, impulse response, step response on the given transfer function. And it is identified that the python code is very small and output time is very less as compared to MATLAB.

Keywords: Python · Control system · Bode plot · Impulse response · Root Locus · Poles-zeros

1 Introduction

Due to the technological advancements in the twentieth century, there is a need to fill up the gap between theory and the real world in the future engineering education system in colleges. Control system engineering is an important and specialized course for students of electronics, electrical and mechanical engineering in colleges. In this chapter, authors has introduced python programming in learning and understanding the basic & tedious concepts of control system engineering. With the help of python programming the practical problems of the control system can easily be solved in a few steps.

This chapter is structured as follows: In Sect. 2 a brief description of python and its libraries has been discussed. In Sect. 3 basics of first order control system 4 we will discuss how to use python programming in understanding and solving the problems of control system engineering & do some analysis of control system engineering. In Sect. 5, a conclusion is drawn on the basis of above analysis and study.

2 Python

Python [1] is a general purpose, dynamic, versatile and powerful programming language. It has high-level data structures and have a simple approach to object-oriented programming. Python is known for its simplicity, as it has very simple syntax and can

S. Mekhilef et al. (Eds.): ICEEE 2022, LNEE 893, pp. 152–160, 2022.
https://doi.org/10.1007/978-981-19-1742-4_12

work on any platform like windows, Mac , linux, raspberry pi etc. Python's prototyping is very quick as it runs on an interpreter system. The biggest strength of python is its vast/rich collection of open source libraries which is used in scientific computing, image processing, GUI applications, web frameworks. etc. [1] In this paper the latest version of python i.e python 3 is used along with jupyter notebook.

In the study of control systems using python, three libraries or packages are used namely NumPy, matplotlib [2] & control system package. NumPy stands for numerical python. Python numpy package contains n-dimensional array object & it is core library for scientific computing. Numpy array is in the form of rows & columns . Mathematical & logical operations can be performed using numpy. Generation of random numbers and linear algebra operations can also be performed using numpy. The combination of NumPy with SciPy (Scientific python) & matplotlib library is widely used as an alternative to MATLAB. NumPy can be installed by typing "pip install numpy" in command prompt and can be import by typing-Import numpy as np [2].

Matplotlib is a graphic plotting library used in python programming language. Simple & complex plots can be designed using matplotlib package with few commands. It is a multiplatform data visualization tool built on numpy & scipy packages. Matplotlib can be installed by typing "pip install matplotlib" in command prompt and can be import by typing-Import matplotlib.pyplot as plt. [3].

The python control system library performs operations for design and analysis of feedback control systems. Using python control system library, both time and frequency response analysis can easily be performed which includes frequency plots & analysis of linear time invariant systems. The python control system package can easily be installed by typing "!pip install –U Control" and can be import by using- Import control as co [4].

In this paper, Jupyter notebook is used in the analysis and study of control system engineering using python. Jupyter notebook is an open source application that is used for machine learning, modelling simulation, data visualization and much more. Installation of jupyter notebook is done using Anaconda distribution which includes python3, jupyter notebook and various other packages for data science and scientific computing [5].

3 Control System

Control system is a system in which the output of the system varies in coordination with the input system. Large varieties of control systems are available. Like, open loop/closed loop system, time independent/time dependent system, deterministic/non deterministic, linear/non linear etc. Here the first order linear time invariant system is taken into consideration to understand the concepts. The physical system is analyzed using its mathematical model. The transfer function is the mathematical representation of the physical system. The transfer function defines the system properties. The analysis of the control systems can be done in both time domain and frequency domain. If the output of the control system varies with respect to time, then it is called the time domain response of the control system. It includes transient response and Steady state response. The standard test signals used for analysis are impulse, step, ramp and parabolic. These signals are used to analyze the performance of the control systems in the time domain. The frequency domain analysis gives the response of a mathematical model with respect to frequency. frequency domain elaborates the stability of the system. Here BODE and NYQUIST plot analysis were discussed for frequency domain.

4 Python Application in Control

At the education level control systems problems are mainly analyzed using MATLAB. MATLAB software provides Control Engineering Toolbox in which systematic analysis, designing, and simulation of linear control systems are performed. In control system toolbox transfer function can be specified for the system, state-space will be done, analysis of pole zero gain it's frequency response modeling can be done with ease. It provides features like step response, Bode plot, Nyquist plot etc. for analysis of behavior of system with respect to time and frequency. But the software is not freely available and affordable for individuals. This work provides an insight for the possibility of using python open source platform to perform all above mentioned tasks of MATLAB for control system engineering [6, 7].

Here the Fig. 1 below shows how to make the transfer function in Python. It also shows the calculation of poles and zeros.

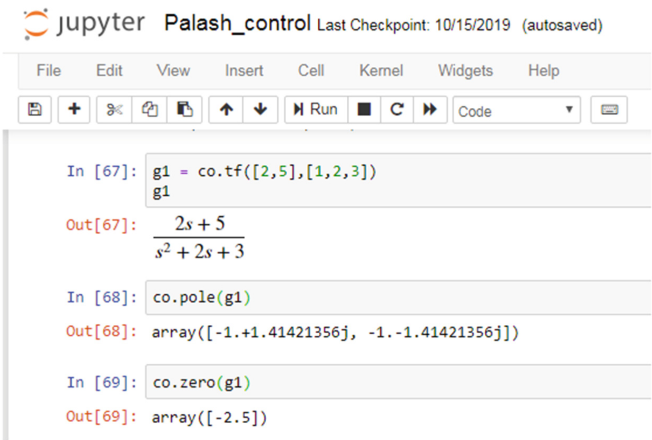

Fig. 1. Transfer function in Python

Figure 2 below shows the pole zero plot of the 2^{nd} order transfer function. Here poles are represented by 'x' and zeros are represented by 'o'. Given transfer function has 2 poles and 1 zero.

Figure 3 below shows the Step response of the transfer function using Python. When unit step input is applied to the given transfer function the output shows transient state for a few seconds, then shows steady state response.

$$\text{Transfer function, T.F} = \frac{2s+5}{s^2+2s+3}$$

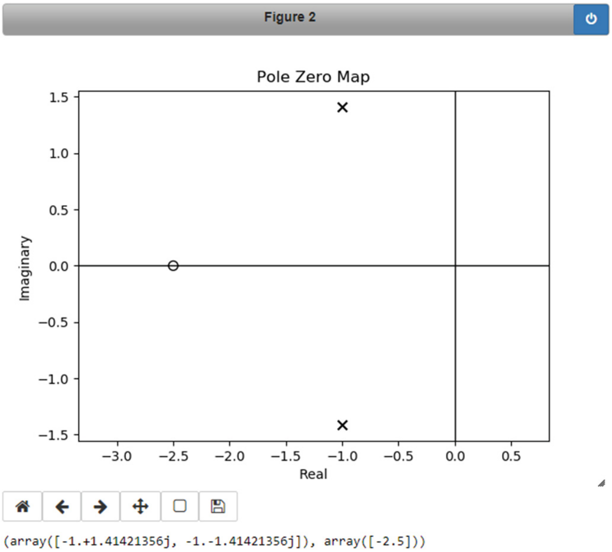

Fig. 2. Pole zero plot of the 2nd order transfer function

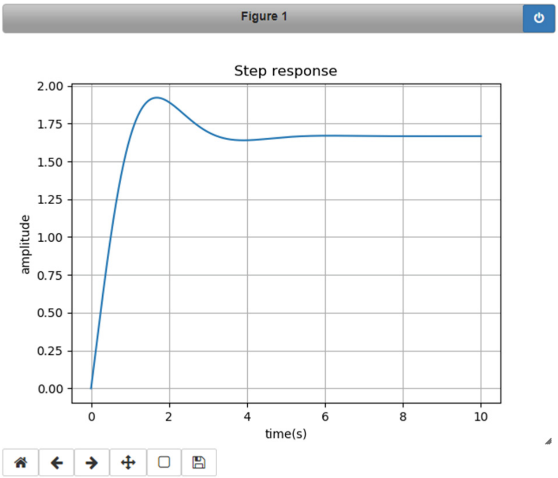

Fig. 3. Step response of the transfer function using Python

Figure 4 below shows the Impulse response of the transfer function using Python. Impulse response is basically inverse Laplace transform of the transfer function, which is a single command operation in python.

$$\text{Transfer function, T.F} = \frac{2s + 7}{s^2 + 3s + 2}$$

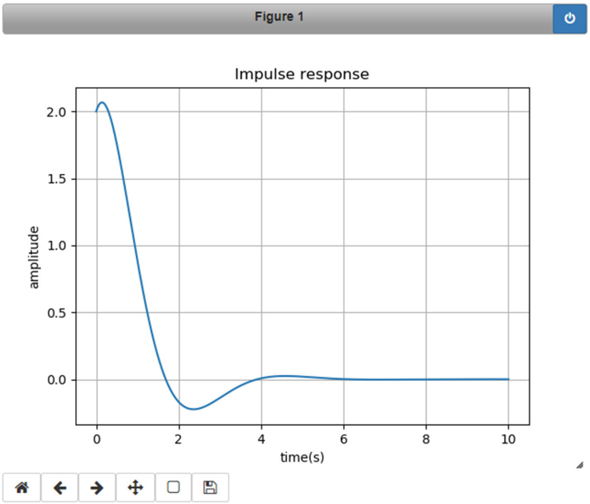

Fig. 4. Impulse response of the transfer function using Python

Figure 5 below shows the Sinusoidal response of LTI System using Python. The graph is drawn between sinusoidal input applied to the transfer function and its output in the time domain.

$$\text{Transfer function, T.F} = \frac{2s+5}{s^2+2s+3}$$

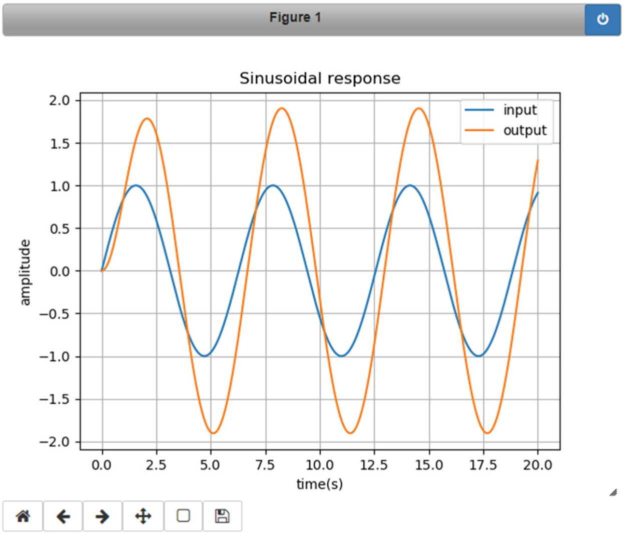

Fig. 5. Sinusoidal response of LTI system using Python

Python also facilitates the conversion of State space representation to Transfer Function & Transfer Function to state space representation. The Fig. 6 and Fig. 7 below depicts the same.

Transfer function to State space:

```
In [12]: g1
```

Out[12]: $$\frac{2s+5}{s^2+2s+3}$$

```
In [13]: g1_ss = co.tf2ss(g1)
         g1_ss

Out[13]: A = [[-2. -3.]
          [ 1.  0.]]

         B = [[1.]
          [0.]]

         C = [[2. 5.]]

         D = [[0.]]
```

Fig. 6. State space representation to transfer function

State space to Transfer function:

```
In [14]: g1_tf = co.ss2tf(g1_ss)
         g1_tf
```

Out[14]: $$\frac{2s+5}{s^2+2s+3}$$

Fig. 7. Transfer function to state space representation

Control systems behavior are analyzed using various plots like Root Locus, Nyquist and Bode plots. These plots are easily generated in Python.

Figure 8 below shows the Root locus plot, with centroid at -1 and poles below and above the imaginary axis along with asymptotes.

$$\text{Transfer function, T.F} = \frac{s}{s^3+2s^2+3s+1}$$

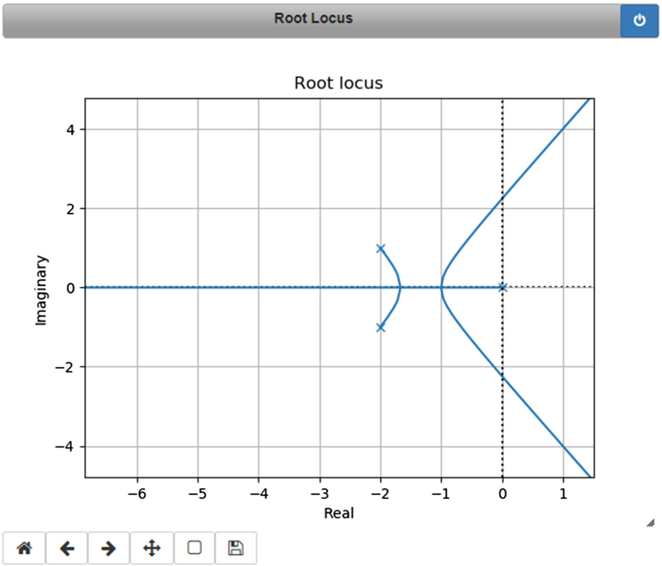

Fig. 8. Root locus plot

Figure 9 below shows the Nyquist plot, To check whether the system is stable or not.

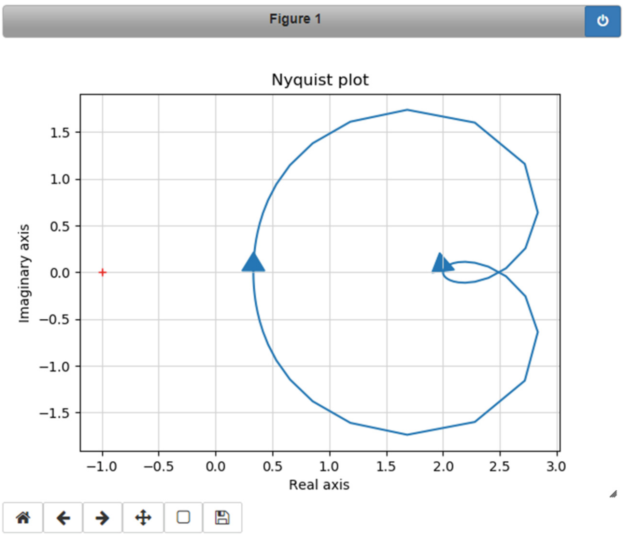

Fig. 9. Nyquist plot

Figure 10 below shows the Bode plot. The plot displays the magnitude (in dB) and phase (in degrees) of the system response as a function of frequency.

$$\text{Transfer function, T.F}\ \frac{2s^2 + 5s^2 + 12s^2 + 5s + 1}{s^2 + 2s + 3\ s^2 + 2s + 3}$$

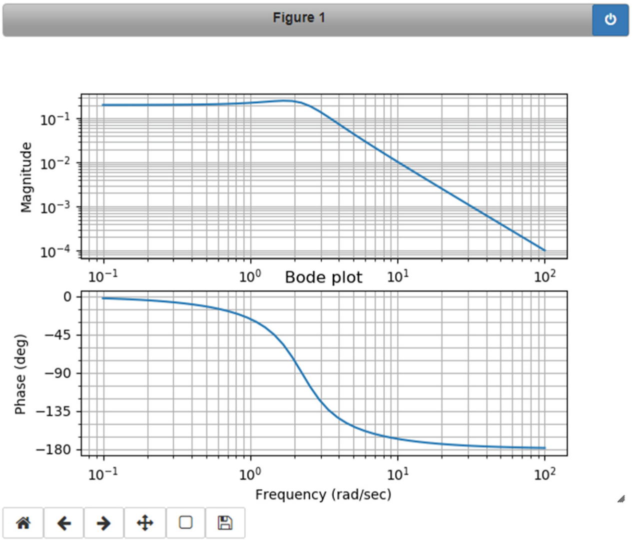

Fig. 10. Bode plot

5 Conclusion and Future Work

In this chapter, an overview study is presented on how Python can perform all the tasks which are required to be done using the Control Toolbox of MATLAB. The chapter discussed calculation of transfer function, pole-zero calculation, step response, impulse response along with various plots like Root Locus, Bode and Nyquist plot for first order and second order LTI systems.

All other functions required in control system analysis can also be explored using Python. The open source access of Python and implementation using Jupiter and empowers the researchers and students to work economically and produce the same results in Python compared to MATLAB.

Acknowledgement. We would like to thank the Institute Innovation Council, SIRT Bhopal, who have given us time to time support for this prestigious work.

References

1. Hashemian, R.: S-Plane bode plots - identifying poles and zeros in a circuit transfer function, In: 2015 IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS). IEEE (2015)

2. Ranjani, J., Sheela, A., Pandi Meena, K.: Combination of NumPy, SciPy and Matplotlib/Pylab - a good alternative methodology to MATLAB - a comparative analysis. In: 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT) IEEE (2011)
3. Teixeira, M.C.M.: A method for plotting the complementary root locus using the root-locus (positive gain) rules. IEEE Trans. Educ. **47**(3) (2004)
4. Pejovic, P.: Replotting the Nyquist plot—a new visualization proposal. In: 2019 20th International Symposium on Power Electronics (Ee), 23–26 October 2019, Novi Sad, Serbia (2019)
5. www.python.org
6. Belkhier, Y., Achour, A., Shaw, R.N., Sahraoui, W., Ghosh, A.: Adaptive linear feedback energy-based backstepping and PID control strategy for PMSG driven by a grid-connected wind turbine. In: Mekhilef, S., Favorskaya, M., Pandey, R.K., Shaw, R.N. (eds.) Innovations in Electrical and Electronic Engineering. LNEE, vol. 756, pp. 177–189. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0749-3_13
7. Jain, J.K.: Secure and energy-efficient route adjustment model for internet of things. Wireless Pers. Commun. **108**(1), 633–657 (2019)