Introducing Python Programming for Engineering Scholars

Zahid Hussain[†], Muhammad Siyab Khan[††]

Sarhad University of Science and Information Technology, Peshawar, Pakistan

Summary

This paper introduces Python as a scientific programming language tool for the purpose of teaching kinematic analysis. Analysis of linkage contrivances is a predecessor to advance analysis of complex mechanisms. This includes analysis of linkage position, velocity and acceleration diagrams. It is much dreary and time overwhelming to draw such complex diagrams, hence, the better choice is to use modern programing modules. An assortment of programing modules is available but, in the preeminent information about the author, no one has the proficient capability to plot the required diagrams. Owing to python programming, dynamic structural support and flexible functionality, necessary programs are inscribed straightforwardly and a quick initiative is decided. The judgement of prevalent programming languages is characterized from the opinion of their expediency of usage for course learning outcomes. This paper highlights current scenario of python's progress and introducing python as an innovative programming language-teaching system based on kinematic analysis in higher education institutions. The concerns of python design based learning approach is to substitute the formal way of teaching the subject course. presented.

Key words:

kinematic analysis using computer programming, programing languages, Four bar linkage, python for engineering scholars.

1. Introduction

Kinematic analysis is the core subject of physical and mathematical science in which engineering scholars analyze position, relative velocity and acceleration diagrams for synthesis of linkage mechanism [1]. There are certain practices that offer to present kinematic analysis with mathematical and graphical approach. Those related with formulation of derivation of motion of particles to create mechanical linkage are analyzed with analytical concept [2] . However, to investigate a possible correlation among various parameters then a graphical approach is needed which is extra perceptive due to some sort of knowledge of geometry and drawing techniques for determining position, velocity and acceleration diagrams of rigid links [3].Constructing Position, Velocity and Acceleration diagrams normally devour much of time that considerably needs the exercise of a programming and

model software. Computer programming with dynamic software packages, is a tool for engineering scholars to solve problems. Majority of these packages include spreadsheets and MATLAB but owing to synthesis of mechanisms that need analysis and modeling on finite elements, require intricate mathematical reckonings [4]. Obviously there exist much of the simulation and linkage software used for the synthesis of mechanism but majority of these software is based on graphical user interface (GUI) however its demerit is that only the input data along with the final results may be visualized while the operating interfering code is veiled from the access [5]. Some of the available free and paid software versions are Linkage 3.4, Ultimate Mechanism Design, ESKAM, Mechanisms with Universal linkages, Synthesis of Planar Mechanism respectively but nearly all of them with some deficiencies are unable to present the required linkages position, velocity and acceleration polygons [6]. Keeping in views these issues, the author main focus is to develop these diagrams with application of more comprehensive and modern programming software like Python programming language that utilize a list of various modules [7]. It provides the principle to promote the study of kinematic analysis of different linkages vector diagrams with further interference for any possible alterations within input values with clear and amazing graphical results based on different parameters and variables. Additional remarkable piece of the python programing module is that it supports requirements of concerned instructors, lecturers and teachers for operative coaching and increases the scholars' interest thus improving intellectual approach.

Actually, the computer based programming languages and other sophisticated software packages may be used to teach kinematic analysis pertains to of motion of various rigid components of a system. The use of these software packages allows to test and analyze the efficiency and performance of the components and simulate the conduct of these components before the constructing stage of proposed prototype model [8]. By means of the customary attitude of education that is based on talk, walk and chalk, tries to convey information from the educator to an inactive receiver. But, due to involvement of technology, there is a rising consciousness between higher education teachers that this flair of teaching is only appropriate for coaching numerical investigation, it takes some types of limits when it concern to development, originality and combination, where dissimilar possible explanations have to be verified [9]. Therefore, it is vital to associate teaching

Manuscript received December 5, 2018
Manuscript revised December 20, 2018

by customary talks and vigorous knowledge based methods to present great inspiration [10].

In addition, dimensional analysis of various systems, the customary mathematical investigation of the calculations may be a tough and challenging task which is not probable in particular circumstances. Hence the contemporary programming techniques deliver an effectual way to support scholars in the dynamic knowledge process [11] .

In this paper, the computer-generated authentic competences are presented in teaching kinematic analysis of four bar linkage mechanism. The rest of the work is prearranged as follows. In last part the necessary python programming techniques for simulation of kinematic analysis of position, velocity and acceleration are highlighted. The major topographies and supported features of the Python language are defined in details and finally the chief conclusions of the current study are discussed.

2. Review of kinematic analysis

2.1 Position Analysis

In this regard, an analytical approach as presented in Figure1 through which the expected results will enable to explore the required position and the point of reference of every link. The input variables to work with are illustrated with required angles [12] . Assuming a four bar mechanism with

Link 1- ground or fixed link;
Link 2 - crank or rotator link;
Link 3 - connecting rod or oscillating link and
Link 4 - follower or slider link

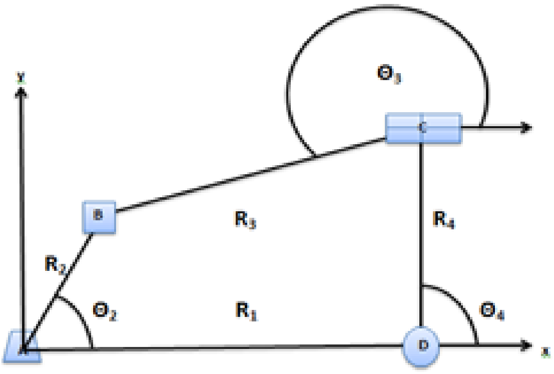


Fig.1 Configuration of positions of four bar linkage

Link 4 and link 1 both have an offset with ground with their vector R4.Further assuming that the crank after being energizing has gained a new position with turning angle θ2while the remaining angles formed and gained their new position by the crank turning as shown in Figure 1.Developing trajectory loop equations for obtaining an absolute position of each individual pinned joint centers as link 1 is frame or fixed then absolute positions are considered those links which have motion with respect to the other links.

$$R1 + R2 = R3 + R4 \quad (1)$$

Defining the position of points from equation set (1):

$$R1 = R2 \times cos * \theta 2 + cos\ R3 \times * \theta 3 \quad (2)$$

$$R4 = R2 \times sin * \theta 2 + R3 \times sin * \theta 3 \quad (3)$$

$$\theta 3 = sin^{-1} \frac{R4 - R2 * sin * \theta 2}{R3} \quad (4)$$

$$R3 = \frac{R1 - R2 * cos * \theta 2}{cos * \theta 3} \quad (5)$$

However new position of turning angle of connecting rod i.e link3and new position of slider link 4 can be analyzed using equation sets (3) and (2) respectively. A new oriented diagram of corresponding position is pinched though this way as shown in Figure 2.

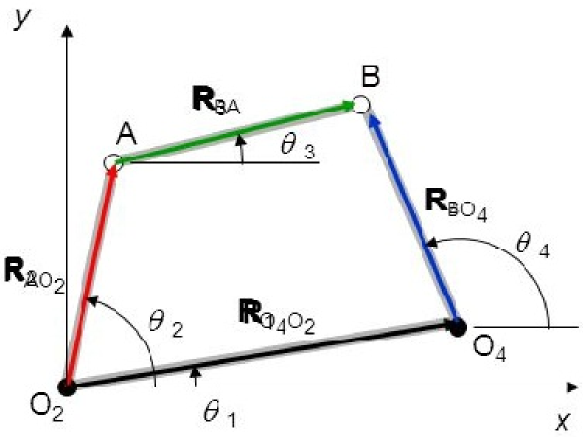


Fig. 2 New oriented position of linkages

2.2 Velocity Analysis

In this regard the relative position of every corresponding link is necessary to be determine as the required rotatory motion of the individual link with reverence to the other becomes important as this vector will always be vertical to the specific link [11]. However, if a vibratory motion is experienced then the orientation of its corresponding relative vector will be allocated through the trend of the applied motion. For rotary link 1crank, a velocity vector is shown in Figure 3 for representing its required extent of velocity and direction however it is observed that the orientation of the relative velocity vector of connecting rod i.e Link 3 is acting perpendicularly on it while the orientation of velocity vector of the fourth and final link i.e follower is acting along its sliding motion [13] . Through this combination a polygon of velocity vector is attained by these junctures related points generated from these two different directions of motions and the corresponding sides of the polygon show the required velocity vectors that explain about the direction of the velocity which is calculated on the basis of its magnitude. To work with Python and to successfully obtained the required mechanism the scale of the initial linear velocity has been calculated at the end point of the corresponding linkage hence the product of these two parameters yields the linear velocity of each individual link. Let ω2 as angular velocity of the crank that is Link 2 and R2 as Effective Length of the Crank (AB), then the magnitude of the linear velocity observed at point B may become equal with respect to corresponding point A:

$$V2 = \omega 2 \times R2 \tag{6}$$

Now with reference to Figure 3 for slider mechanism let's drawing the velocity vector representing with ab normally acting to points A and B and having corresponding linear velocity as *V2*. Next, in order to analyze the polygon of velocity, let's draw bc and dc the corresponding velocity vectors which are exactly same to the required velocity of kinematic links 3 and 4 acting normally to lines BC and DC respectively. Let: ω_3 – Angular Velocity of Connecting Rod i.e Link 3 (bc) and R_3 - Effective Length of the corresponding Link (BC), then the magnitude of the angular velocity observed at point B and C may be:

$$\omega 3 = \frac{bc}{R3} \tag{7}$$

While the corresponding linear velocity of the follower link is given by:

$$V4 = DC \tag{8}$$

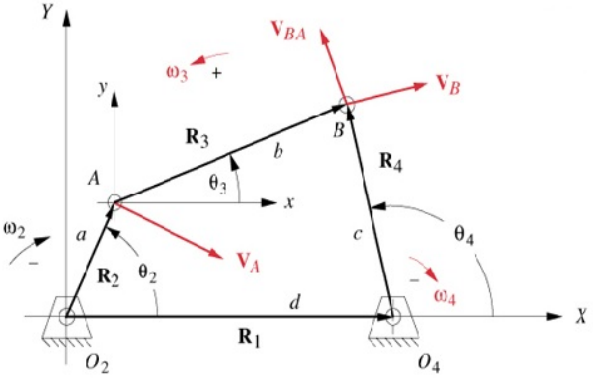


Fig.3 Velocity vector diagram

2.3 Acceleration Analysis

The acceleration analysis involves with linear acceleration of each rigid link and for instance there are two components in which one acts radially while the other one tangentially [13]. Assuming that A2 (radial) and A2 (tangential) are the two consecutive constituents of acceleration pertaining to Link 2. Determining those two components of acceleration.

$$A2\ (\text{radial}) = (\omega 2)^2 \text{ x } R2 \tag{9}$$

$$A2\ (\text{tangential}) = (\alpha 2)^2 \text{ x } R2 \tag{10}$$

Form equations (9) and (10) the resultant required acceleration pertaining to Link 2 is obtained.

$$\alpha 2 = \sqrt{A2(radial)^2 + A2(tangential)^2} \tag{11}$$

It is also noticeable that Slider (Link 4) moves with linear velocity hence it creates a component of tangential acceleration. However, for Link 3 this component will be:

$$A3\ (\text{radial}) = (\omega 3)^2 \text{ x } R3 \tag{12}$$

The oriented acceleration vector diagram as shown in Figure 4 represents the corresponding acceleration vectors i.e aa' & ab' constructed along the normal line to AB with A2(radial) and A2(tangential) as magnitude. Using the same strategy, here is acceleration vectors i.e bb' constructed along the line to BC with A3(radial) as the magnitude. In order to finally construct the required acceleration polygon velocity vector b'c along with cd are represented normal to line BC and DC separately. To obtain the linear acceleration for Link 3 (a3) and for Link 4 it would be(a4) respectively:

$$A4 = dc \tag{13}$$

$$A3 = bc \tag{14}$$

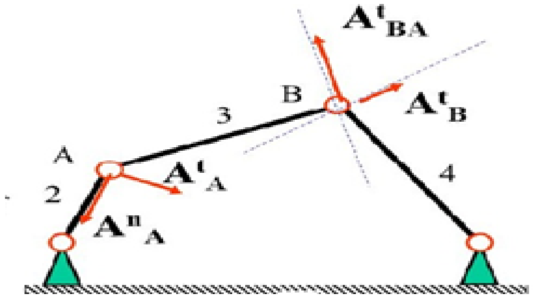


Fig. 4 Slider mechanism Acceleration vector diagram

3. Choosing best programming language:

In fact, engineering scholars use the principles of science and mathematics to find solutions of a certain system design, development, testing and managing production. They also model corporeal systems and technological processes in the physical domain as mathematical models and forecast their behavior. This aptitude to simulate actual systems or processes with mathematical approach is basically a primary fragment. Obviously the available programming languages used for solution of mathematical complications comprise of MATLAB, Fortran and Scilab [17]. The acceptance of Scilab and MATLAB is principally due to the quick response of results since they possess an unpretentious syntax, using matrix for data edifice with exclusive libraries and 2D and 3D data presentation but these programs lack some of the needs of the engineering scholar. Similarly, by considering various demerits of Pascal language which reduces its capability in physical projects, and, as a consequence, dwindling of its admiration, selecting a general and necessitated language makes intellect. [18]. Next significant measure is the technical capabilities of respectively programming language for teaching, especially at novice level. [19]. Let's have a review to choose a specific language out of the following.

Pascal: Obviously is considered a decent language to teach programming, but unfortunately utilization of this programming language in practical submissions is on the diminish, hence novel based programming languages are substituting it. Furthermore, current programming settings are either outmoded or branded. Another substantial motive to trash Pascal language is a poor inspiration of scholar to study it owing to the real-world unfeasibility to use it in their projects [20].

C++: Still considered one of the favorite and widely used programming language due to its convenient and user friendly environment to teach basic abilities of system programming to computer programmers. However special teaching skills and knowledge is required to teach specialists, but not novices [21].

PHP: Hypertext Preprocessor (PHP) is basically a scripting programming language and specifically designed for general purpose and Web development as well. In spite of its reputation, the language is not very stylish and topographies syntax illogicalities, non-synchronization in meaning names and a lively typing of certain variables with a likelihood to change a specific type. The language use C type syntax only for web programming purpose hence not much appropriate to teach programming to beginners.

JavaScript: JavaScript is typically used as implanting language for access of programs to various objects and normally uses in web browsers. However, some demerits associated with this language is its visibility to everyone that is the java script code is always available to anyone. No matter how fast JavaScript construe, its document object model (DOM) experience much slow response and its single error automatically stop to render with complete site. Hence not proper to teach programming to scholars because of the individualities of its use [22].

Python: For cooperative purposes, python is considered a high-level programming language, with a stress on a developer's output and readability of syntax code. Due to its dynamic features and compatibility opportunities like active typing, complete self-examination and multi-threaded reckonings provision, it will be stared as an opponent to substitute Pascal language.

4. Python Programing Language Module

Python is renowned, comprehensively acknowledged and easy to recognize programing language due to its vigorously typing with great number of built-in data types and simple syntax. Its features like READ, EVALUATE, PRINT LOOP (REPL) is comparable to those with Scilab, and MATLAB thus making it tremendously valuable and instinctive for engineering scholars [23]. Figure 5 illustrates three basic Python operating system has three main modules: The input module followed by processing and the output module. The input describes the user interface for mechanical scheme which is pure text built. The system script produced is interpreted into abstract syntax for creating the required equations of linkage motion with the help of processing module[24]. During the analysis step module, the set of given equations is abridging and decreases and permits them for necessary generation of perspectives code which generates the output profiles. The graphical portions module calculates the required positions and locations of the linkage elements of mechanism.

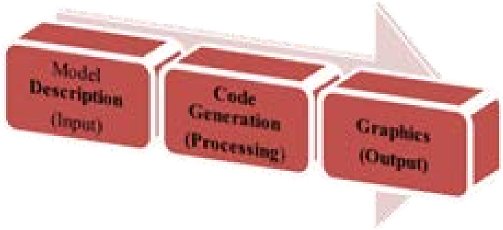


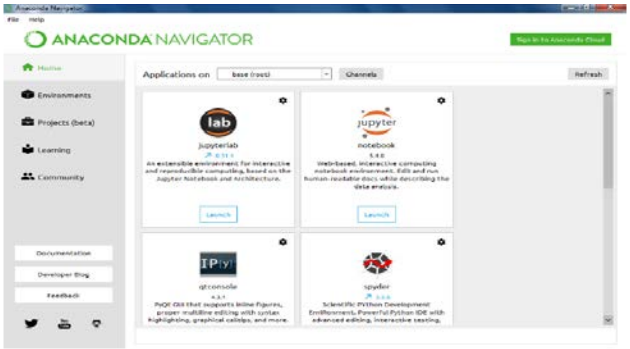
Fig. 5 Python operating systems

Python programming integrates the multipart tasks into slight interrelated ones. Basic function of python modules is to make the improvement and debugging process easy, and may be merely introduced to permit the use of numerous utilities [25]. Some of the Python modules are:

- CPython is the standard version of Pythons executed in C++ and in Python as modules.
- NumPy – Numerical Python, outfits the multi-dimensional array.
- SciPy – Scientific Python, provides a large number of numerical methods.
- Sympy – Symbolic Python.
- Matplotlib – Mathematical Plotting Library that generates the plotting utilities.

4.1 Anaconda

Anaconda IDE (Integrated Development Environment) as shown in Figure 6 presents some appropriate and comprehensive means to start a working atmosphere for a scholar as well as for a practiced designer. Anaconda IDE offerings a command streak interpreter provided with radical landscapes. Some of them are syntax accomplishment, keyword importance followed by a folder browser. Anaconda presents the current and much more characteristics just in a meek and instinctive way. Further, Pandas outfits the Data Frame not only make it possible to write programs for image processing, machine learning and computer vision, but also easy and fun.



An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

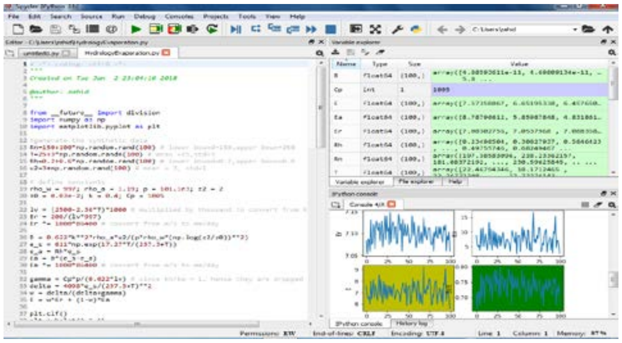
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing,

Fig. 6 Main anaconda IDE view

4.2 Spyder

Spyder is basically an open source program access integrated development environment (IDE) flexible platform for developing scientific programming modules using Python language. Initially spyder assimilates with wide number of protuberant and amazing software packages in the scientific Python pile, that fully include additional libraries like NumPy, Editor, SciPy, I Python, Matplotlib, pandas,and Cython, followed by other extra open sources capabilities. Spyder integration development environment module was initially shaped and established in 2009 by Pierre Raybaut, however since 2012 it has been upgraded on regular basis and unceasingly enhanced by a team of scientific based community developers. Spyder has open extension, support and various plugins for interactive tools for data visualization , inspection and implants specific programming code and self-examination instruments which are available at cross platform using Anaconda on Windows. The Spyder IDE as shown in Figure 7 delivers a convenient way for working with Python files. Its name originates from Scientific Python Development EnviRonment and considered as a major components of the anaconda distribution. It resents three panes:

- ➢ Editor: used for writing and editing program supported with Run option for results.
- ➢ IPython Console: used for writing Python provisions using console window.
- ➢ Help Explorer, variables, and File.



```
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt
```

```
plt.clf()
```

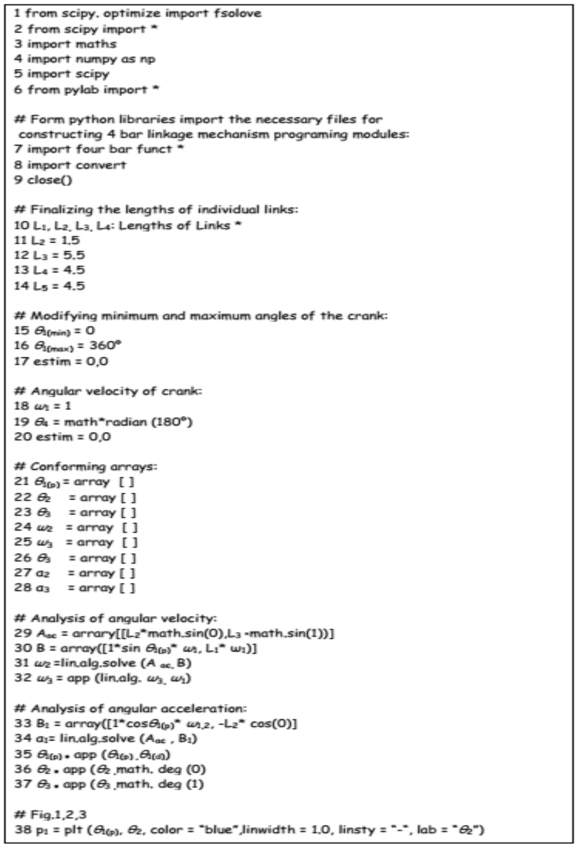
Fig. 7 Spyder IDE shell view

4.3 Jupyter

Jupyter Notebook is another enormous assistance of Python for academics. It is a web based investigative programming environment for Python. Code can be generated and executed in parallel while output in the form of both text and graphics displayed under the cell. Similarly, the notebook may also be converted to PDF or HTML formats [26]. It is feasible to evaluate scholar's assignments by a web facility that can automatically assess the code and show if the required code qualify all tests assigned by the instructor. Jupyter Lab which is another expansion of the Notebook into a full IDE and allocates superior programming efficiency [27].

5. Methodology

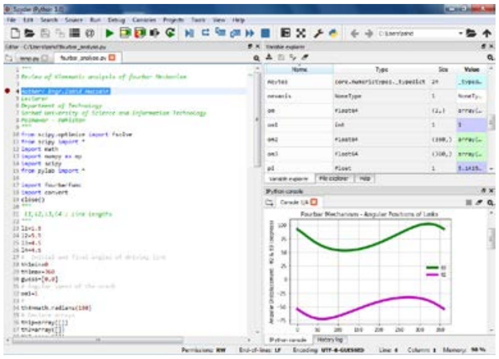
Core drive of the current work is to write a Python program for presenting position, velocity and acceleration diagrams using equations 1 to 14 related to a four bar linkage mechanism especially for pedagogics persistence. Initializing Programming module using python library to simulate a four bar linkage and introduce the command for handling kinematic loops using complete source code. As usual, start to create a model system and adding various parameters. Here, they are the lengths, masses and inertia of the bars to be connected:



```
from scipy. optimize import fsolove
from scipy import *
import maths
import numpy as np
import scipy
from pylab import *

# Form python libraries import the necessary files for
 constructing 4 bar linkage mechanism programing modules:
import four bar funct *
import convert
close()

# Finalizing the lengths of individual links:
L1, L2, L3, L4: Lengths of Links *
L2 = 1.5
L3 = 5.5
L4 = 4.5
L5 = 4.5

# Modifying minimum and maximum angles of the crank:
θ1(min) = 0
θ1(max) = 360°
estim = 0,0

# Angular velocity of crank:
ω1 = 1
θ4 = math*radian (180°)
estim = 0,0

# Conforming arrays:
θ1(p) = array  [ ]
θ2    = array [ ]
θ3    = array [ ]
ω2   = array  [ ]
ω3   = array  [ ]
θ3    = array [ ]
a2    = array [ ]
a3    = array [ ]

# Analysis of angular velocity:
Aac = arrary[[L2*math.sin(0),L3 -math.sin(1))]
B = array([1*sin θ1(p)* ω1, L1* ω1)]
ω2 =lin.alg.solve (A ac, B)
ω3 = app (lin.alg. ω3, ω1)

# Analysis of angular acceleration:
B1 = array([1*cosθ1(p)* ω1,2, -L2* cos(0)]
a1= lin.alg.solve (Aac , B1)
θ1(p) . app (θ1(p) , θ1(d))
θ2 . app (θ2 ,math. deg (0)
θ3 . app (θ3 ,math. deg (1)

# Fig.1,2,3
p1 = plt (θ1(p), θ2, color = "blue",linwidth = 1.0, linsty = "-", lab = "θ2")
```

6. Results

The required mechanisms carefully modelled carefully and independently in Python program shell however final analyses and results have been executed using spyder integration simulation module. The computed results were associated at distinct locations for some response constraints and establish to be similar with the final results acquired from manual computation. After inserting the script in spyder editing shell and running the program, the output will look like as shown in Figure 8.



```
from scipy.optimize import fsolve
from scipy import *
import math
import numpy as np
import scipy
from pylab import *

import fourbarfunc
import convert
close()
```

Fig. 8 Spyder explorer variable explorer pane.

Figure 9, 10 and 11 shows the IPython console pane in which the graphical output by provoking the input variables for required position, velocity and acceleration diagrams of four bar linkage mechanism.

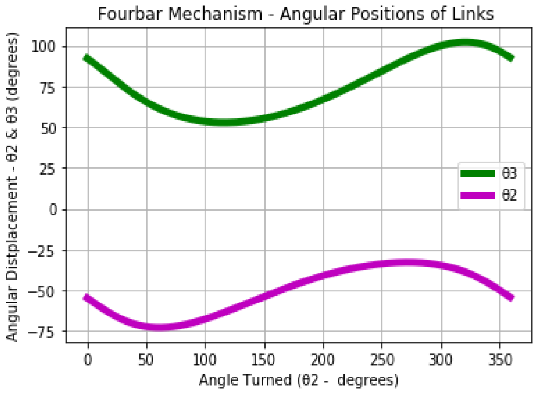


Fig. 9 Spyder IDE IPython console output of position analysis

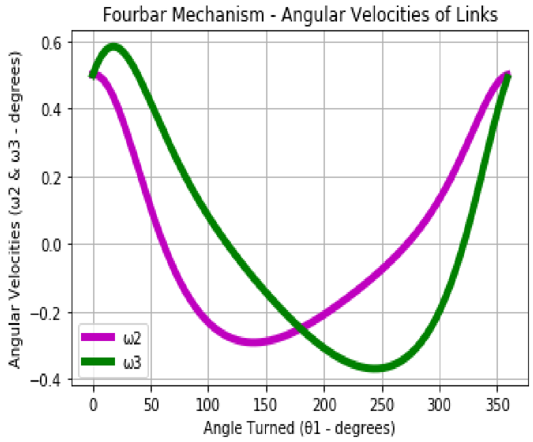


Fig. 10 Spyder IDE IPython console output of velocity analysis

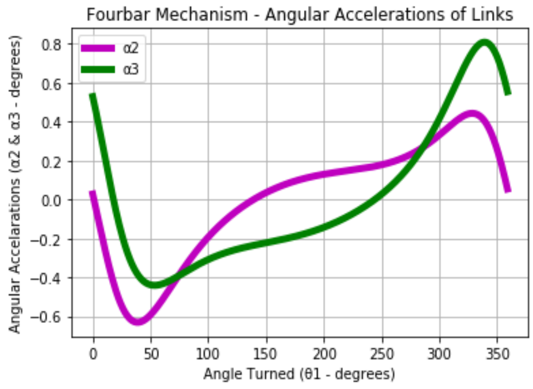


Fig. 11 Spyder IDE IPython console output of acceleration analysis

5. Conclusion

Presently the practice of Pascal programming language to teach engineering scholars at begging level has become antiquated while Python language that is more influential and modest, is recommended to be applied for various drives. Python assures to be an innovative language to look forward to in the educational and scientific fields. Due to its wide programming modules, learning can be ongoing without any groundwork hence Python has effectively achieved courtesy around the world owing perfect sophistication and pointed development bends particularly for effortlessness of teaching as well as learning. It well serves the basic needs of engineering scholars for kinematic analysis of synthesis. Due to availability of a great number of software developers, Python has proved irritating collections of programming modules for various scientific and educational problems. Hence, Python educated engineers may satisfy the straining of modern job, which stresses on careful path. In this work kinematic analysis of linkage position, velocity and Acceleration diagrams were studied using python programming language. Similarly, simulation necessary for illustration of vectors analysis were presented that might be used by an instructor with formal way of teaching. Programming knowledge based teachers and instructors with strong awareness of Python language are constantly in petition, and they are sound remunerated. Therefore, those who grasped Python, can seek to start decent positions in data processing, information technologies and computer science based companies. In this paper, an effort to validate the computer-generated authentic capabilities regarding teaching of kinematics analysis was presented. In wide-ranging, engineering scholars studying this course subject for future project building, teaching have communicated improved appreciative of the topic with the incorporation of related software and programming tools. It is extremely inspiring, for concerned learners, to experience some simulations and to perceive the outcome of all the precise formulae that they settled before, they texture that all the established determination is essential and they also observe the usefulness of these results. The basic aim of this paper was to establish the possible potentials of the presentation of a suitable programming language using Python based approach that use various graphical programming modules in the course subject of kinematic analysis of a complex system.

References

[1] S. Rattan, “Theory of machines,” 2014.
[2] A. G. and A. K. Malik, Theory of mechanisms and machines, 3rd ed. New Delhi: East-West, 2006.
[3] R. Norton, “Design of machinery: an introduction to the synthesis and analysis of mechanisms and machines,” 1999.

[4] S. Patel, D. P.-I. J. E. S. I. Technol, and undefined 2013, “Dynamic Analysis of Quick Return Mechanism Using MATLAB,” ijesit.com.

[5] R. P. Podhorodeski, S. B. Nokleby, and J. D. Wittchen, “Quick-Return Mechanism Design and Analysis Projects,” Int. J. Mech. Eng. Educ., vol. 32, no. 2, pp. 100–114, Apr. 2004.

[6] J. Cabrera, A. Simon, M. P.-M. and M. theory, and undefined 2002, “Optimal synthesis of mechanisms with genetic algorithms,” Elsevier.

[7] E. C. Kinzel, J. P. Schmiedeler, and G. R. Pennock, “Kinematic Synthesis for Finitely Separated Positions Using Geometric Constraint Programming,” J. Mech. Des., vol. 128, no. 5, p. 1070, 2006.

[8] A. Akay, “The Renaissance Engineer: Educating Engineers in a Post-9/11 World,” Eur. J. Eng. Educ., vol. 28, no. 2, pp. 145–150, 2003.

[9] A. W. P. Piela, T. Epperly, K. Westerberg, “ASCEND. An object-oriented computer environment for modeling and analysis: the modeling language,” Comput. Chem. Eng., vol. 15, no. 1, pp. 53–72, 1991.

[10] N. A. van H.-J. R.M. Lima, D. Carvalho, A.M. Flores, “Case study on project led education in engineering: students and teachers’ perceptions,” Eur. J. Eng. Educ., vol. 32, no. 3, pp. 1–11, 2007.

[11] A. Kapse and D. H.-I. J. of E. R. and, “A Generalized Approach for Measurement of Performance of Planar Mechanism Using Relative Velocity Method,” Citeseer.

[12] A. Smaili and N. Diab, “Optimum synthesis of hybrid-task mechanisms using ant-gradient search method,” Mech. Mach. Theory, vol. 42, no. 1, pp. 115–130, Jan. 2007.

[13] R. Arnay, J. Hernández-Aceituno, E. González, and L. Acosta, “Teaching kinematics with interactive schematics and 3D models,” Comput. Appl. Eng. Educ., vol. 25, no. 3, pp. 420–429, May 2017.

[14] H. P. Langtangen, “Introduction to GUI Programming,” 2004, pp. 205–273.

[15] A. Abbas and S. Sarfraz, “Developing a Prototype to Translate Text and Speech to Pakistan Sign Language With Bilingual Subtitles: A Framework,” J. Educ. Technol. Syst., vol. 47, no. 2, pp. 248–266, Dec. 2018.

[16] M. S. Farooq, S. A. Khan, F. Ahmad, S. Islam, and A. Abid, “An Evaluation Framework and Comparative Analysis of the Widely Used First Programming Languages,” PLoS One, vol. 9, no. 2, p. e88941, Feb. 2014.

[17] P. Cherelle, V. Grosu, A. M.-… Engineering, and undefined 2014, “Design and validation of the ankle mimicking prosthetic (AMP-) foot 2.0,” ieeexplore.ieee.org.

[18] A. Hirst, J. Johnson, M. Petre, … B. P.-A. L. and, and undefined 2003, “What is the best programming environment/language for teaching robotics using Lego Mindstorms?,” Springer.

[19] A. J. Hirst, J. Johnson, M. Petre, B. A. Price, and M. Richards, “What is the best programming environment/language for teaching robotics using Lego Mindstorms?,” Artif. Life Robot., vol. 7, no. 3, pp. 124–131, Sep. 2003.

[20] A. N. Habermann, “Critical comments on the programming language Pascal,” Acta Inform., vol. 3, no. 1, pp. 47–57, 1973.

[21] B. Stroustrup, The C++ programming language. 2000.

[22] P. Stothard, “The Sequence Manipulation Suite: JavaScript Programs for Analyzing and Formatting Protein and DNA Sequences,” Biotechniques, vol. 28, no. 6, pp. 1102–1104, Jun. 2000.

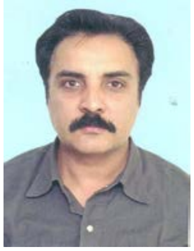
[23] M. S.-J. M. G. Model and undefined 1999, “Python: a programming language for software integration and development,” pdfs.semanticscholar.org.

[24] A. Rees, F. García-Peñalvo, I. Jormanainen, and M. Tuul, “An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers,” 2016.

[25] M. Lutz and M. Lutz, Programming python. 1996.

[26] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. G.- ELPUB, and undefined 2016, “Jupyter Notebooks-a publishing format for reproducible computational workflows.,” books.google.com.

[27] M. Ragan-Kelley, F. Perez, … B. G.-A. F. M., and undefined 2014, “The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication.,” adsabs.harvard.edu.



Zahid Hussain is currently performing his academic services in Faculty of Engineering and Technology in Sarhad University of Science and IT Peshawar, Pakistan. Graduated in 1999 from Voronezh State University of Engineering and Technology (Russian Federation) in Mechanical Engineering with Master of Science in Engineering Management. He is the life time member of Pakistan Engineering Council with "Professional Engineer" Title. He has more than 15 years of Professional Experience in various nationals and multinationals organizations in the field of Design, Production, Maintenance and Consultant.