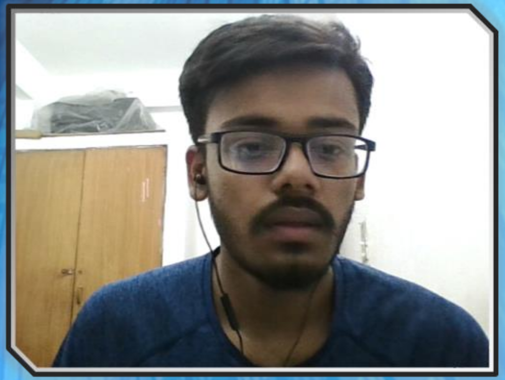
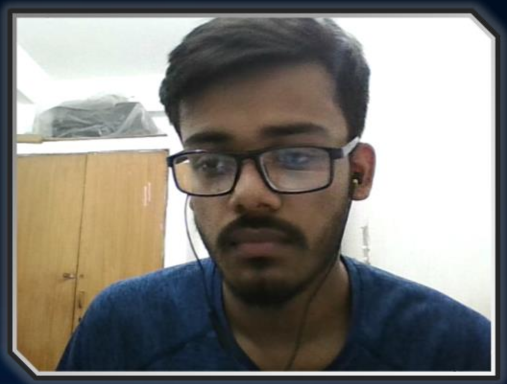
Inverse Kinematics of Three DOF Planar Robot



TO: ROSHAN KUMAR HOTA SIR, ASSISTANT PROFESSOR,
DEPARTMENT OF MECHANICAL ENGINEERING

TEAM MEMBERS:

- TANMAY TRIPATHI (20226152)
- KESHAV KRISHNA SHARMA (20226075)
- MANUDEV VERMA (20226082)



APPROACH OF THE PROBLEM :

Step 1

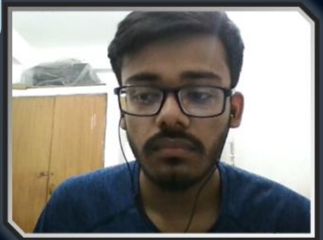
- Defining the design variables , variable bounds and constraints

Step 2

- Mathematical formulation of objective function using trigonometric equations

Step 3

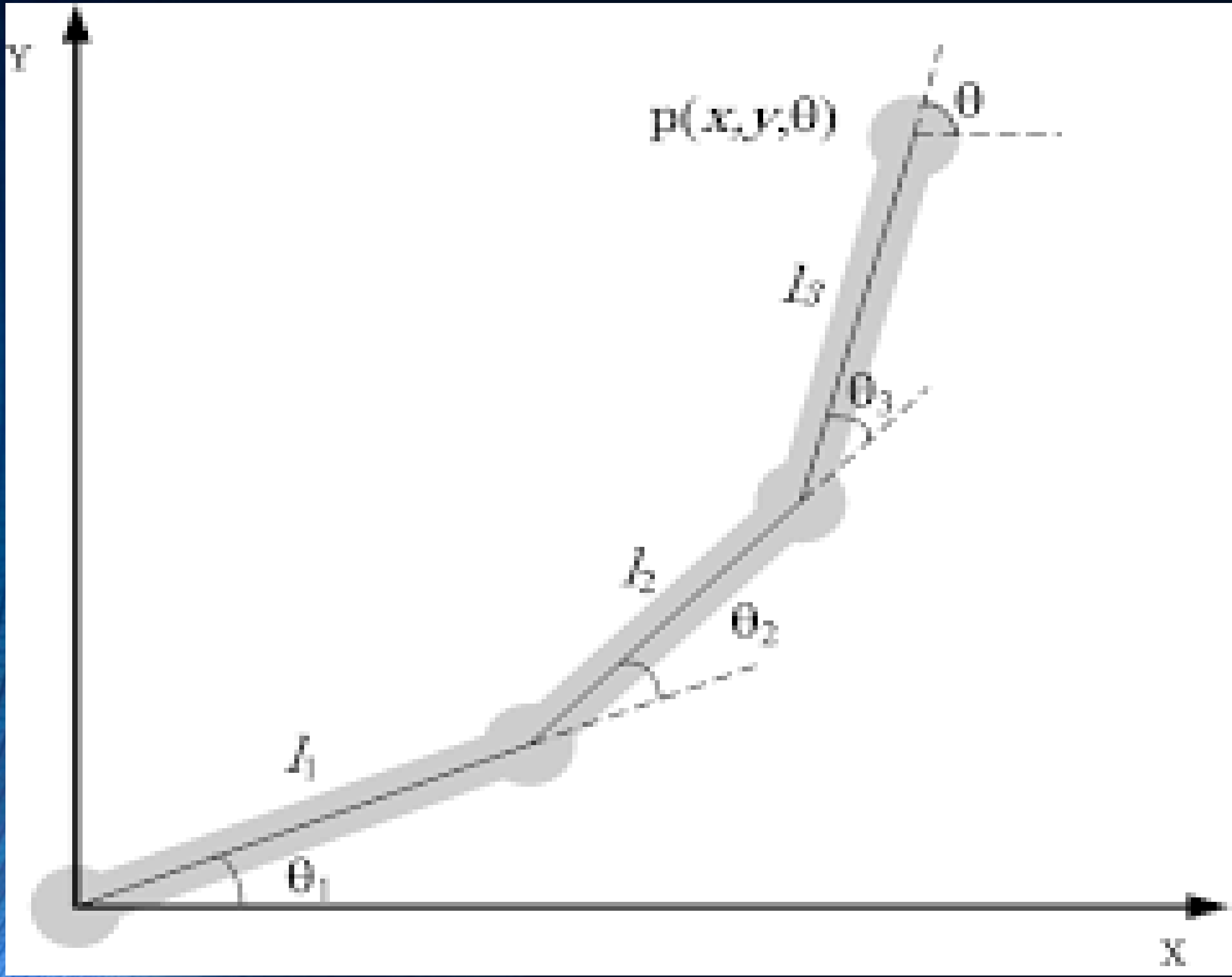
- Optimizing the objective function using specific multi variable optimization methods

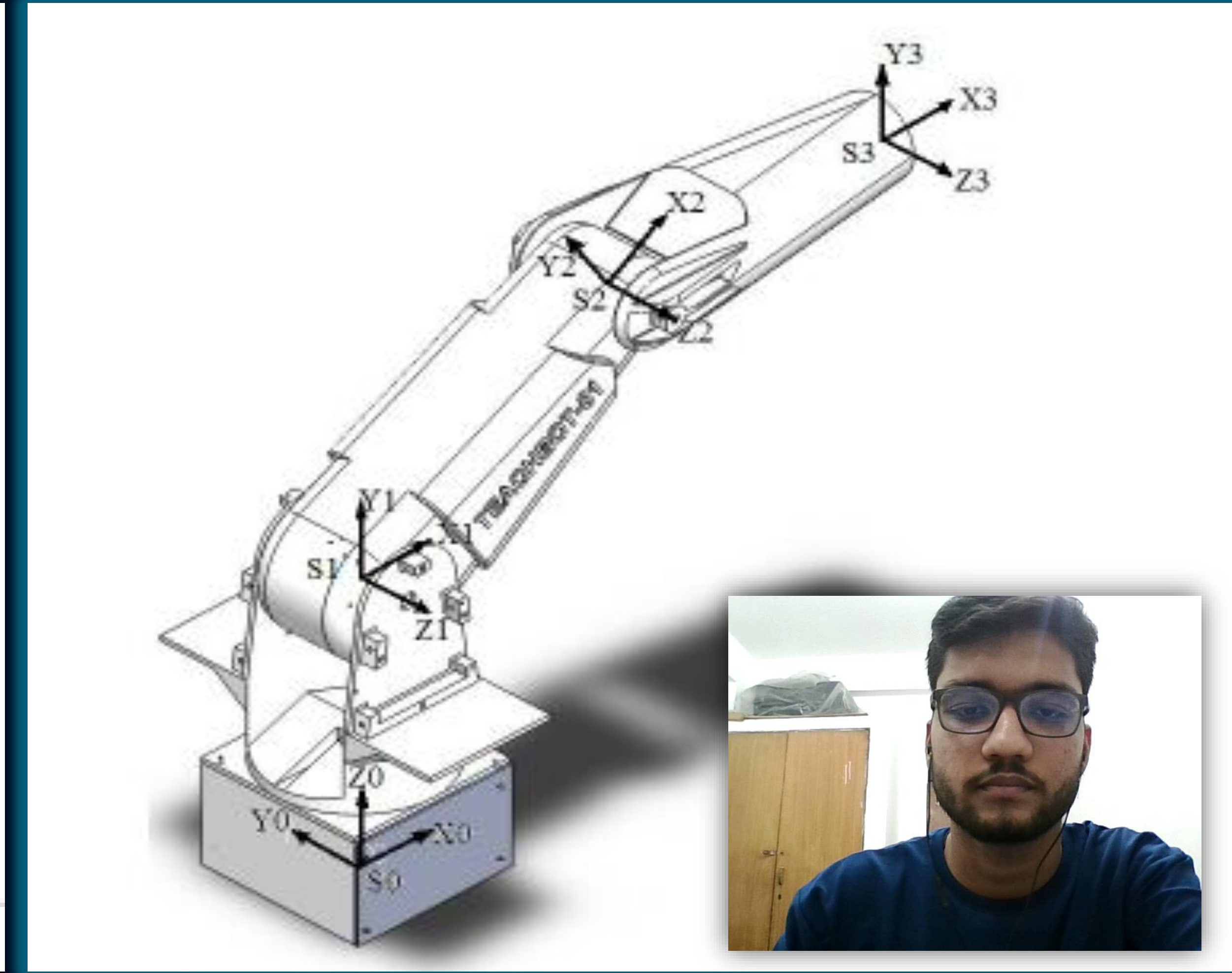


INTRODUCTION:

Inverse kinematics (IK) is a method of solving the joint variables when the end-effector position and orientation (relative to the base frame) of a 3R planar Robot and all the geometric link parameters are known. Optimizing the inverse kinematics of a 3R planar robot involves enhancing computational efficiency while ensuring accurate positioning of end-effectors. One approach is to leverage geometric techniques, such as the law of cosines and the geometric interpretation of the robot's kinematic equations, to simplify the calculation process. Additionally, utilizing numerical methods like iterative algorithms (e.g., Newton-Raphson) can refine the solutions iteratively, enhancing precision. Employing optimization algorithms such as genetic algorithms or particle swarm optimization can further fine-tune parameters for optimal performance, considering constraints such as joint limits and workspace boundaries. Moreover, incorporating efficient data structures and parallel processing techniques can expedite computations, crucial for real-time applications. By integrating these methodologies, the inverse kinematics of a 3R planar robot can be optimized to achieve a balance between computational efficiency and accuracy, facilitating smoother and more precise motion control in various applications.

DIAGRAMS





In above planar figure , let:

$x_1 = l_1 * \cos \Theta_1 + l_2 * \cos(\Theta_1 + \Theta_2) + l_3 * \cos(\Theta_1 + \Theta_2 + \Theta_3)$

$y_1 = l_1 * \sin \Theta_1 + l_2 * \sin(\Theta_1 + \Theta_2) + l_3 * \sin(\Theta_1 + \Theta_2 + \Theta_3)$

Hence objective function will be;

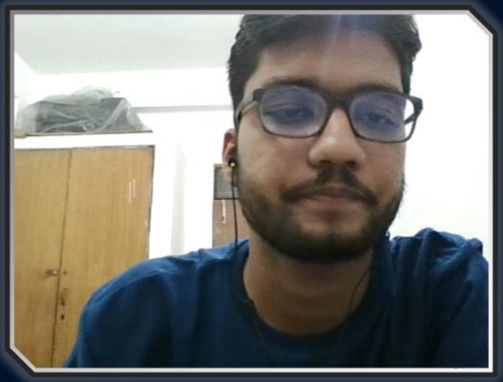
$F(\Theta_1, \Theta_2, \Theta_3) = [x - x_1]^2 + [y - y_1]^2$

Where the design variables Θ_1, Θ_2, Θ_3 are angles as shown in figure and l_1, l_2 and l_3 are lengths of robotic arms and (x_1, y_1) is the end effector position

Constraints :

$0 < \Theta_1, \Theta_2, \Theta_3 < 360^\circ$

Explanation



After formulating we get our objective function as:

$$F(\Theta_1, \Theta_2, \Theta_3)=[x-x_1]^2 +[y-y_1]^2$$

Where ,

$$x_1=l_1*\cos \Theta_1 +l_2*\cos(\Theta_1+ \Theta_2)+ l_3*\cos(\Theta_1+ \Theta_2+\Theta_3)$$

$$y_1=l_1*\sin \Theta_1 +l_2* \sin(\Theta_1+ \Theta_2)+ l_3*\sin(\Theta_1+ \Theta_2+\Theta_3)$$

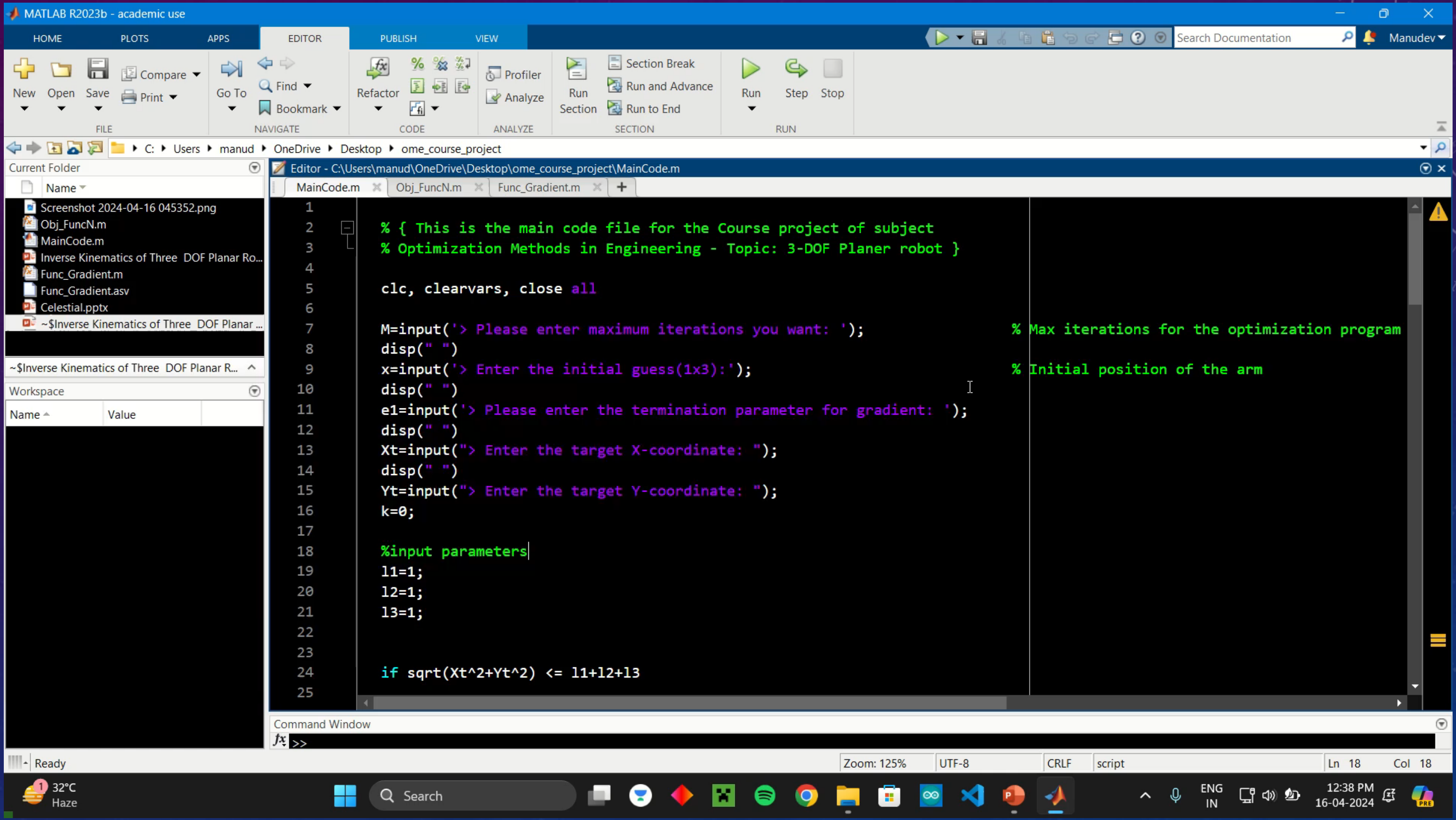
The concept behind formulating the above objective function is that the $F(\Theta_1, \Theta_2, \Theta_3)$ will be minimum when both the square quantities(that are $[x-x_1]^2$ and $[y-y_1]^2$)are zero (i.e. when $x=x_1$ and $y=y_1$, which are the desired end effector points). The idea behind squaring the difference of desired point and end effector point is that the function formed will be the square of Euclidean distance and minimizing it will reduce the error between the two quantities. Therefore we can use the multivariable optimization algorithm known as Cauchy steepest descent along with analytical univariable search method to find the minima of our objective function.

---[MAIN CODE]---

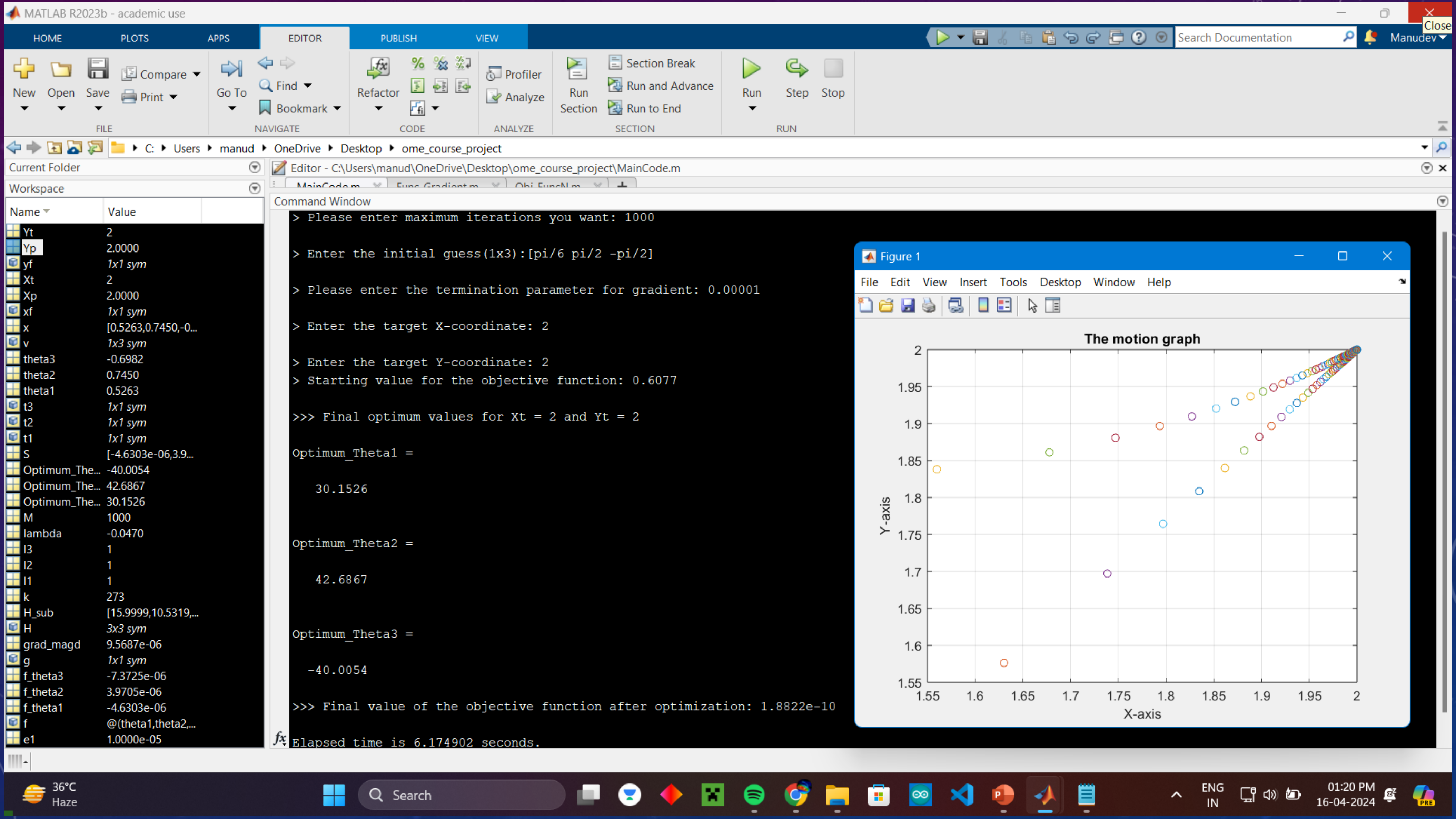
[THIS SECTION CONSISTS MAINLY OF THE CODE AND PROCESSING]

[WORK FOR THE OPTIMIZATION]

---[CODE EXPLANATION]---



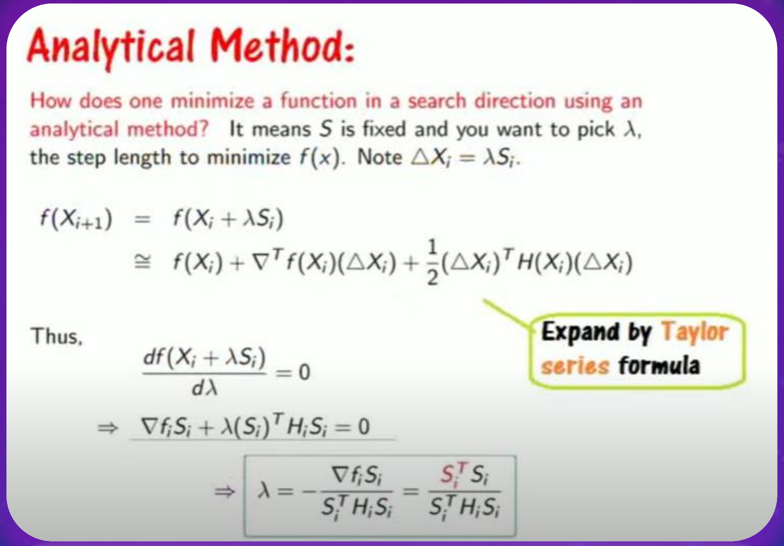
[SCREENSHOT OF A SAMPLE RUN]



REFERENCES:

- www.mathworks.com
- https://in.mathworks.com/help/matlab/
- https://www.youtube.com/
- Book: Optimization Methods in Engineering
  - Kalyanmoy Deb

- Lambda value via analytical method :
- www.google.com



THANK YOU!