Robot Control System Simulation (MATLAB Version)

Were Vincent

Vehicle Cruise Control System

1. Knowledge Point Arrangement(Mind Map) for Vehicle Cruise Control System

Function

The vehicle cruise control system automatically controls the speed of a vehicle. The system maintains a set speed as set by the driver, reducing the need for manual throttle control.

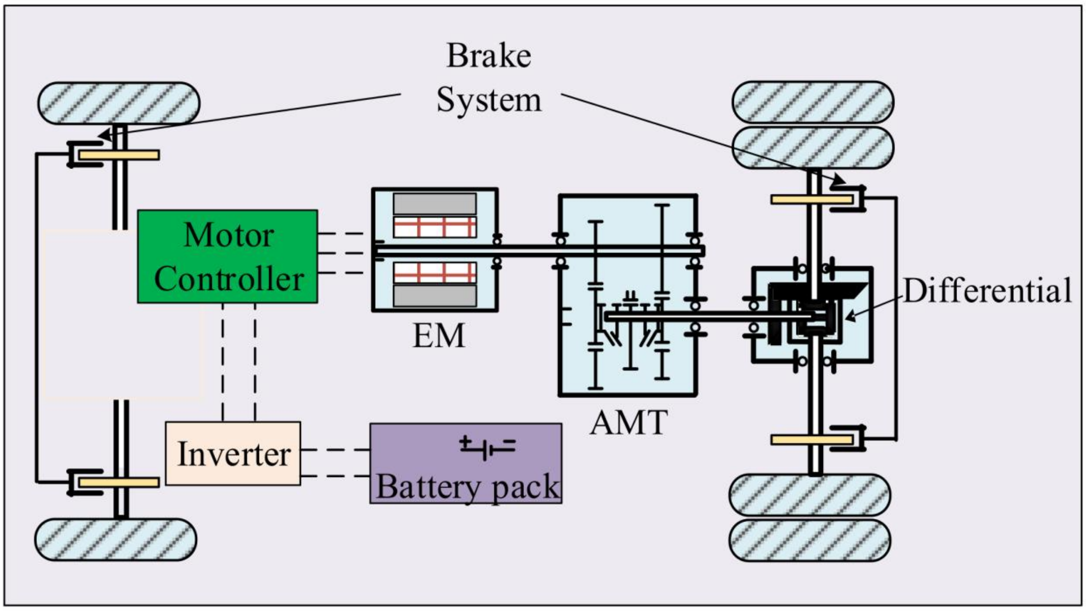
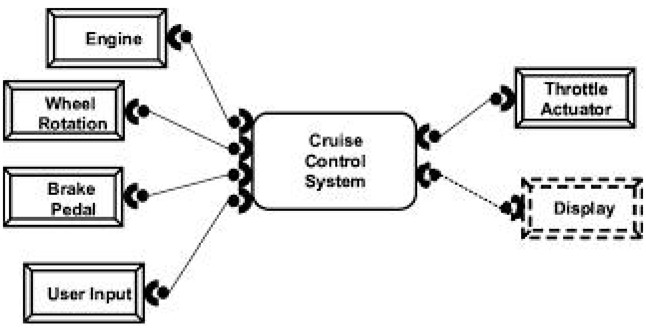


Figure: Mind map of Vehicle Cruise Control System



Key Points and Difficulties

Key Points

- Accurate modeling of vehicle dynamics and resistive forces is essential.
- Effective tuning of PID controller parameters ensures optimal performance.
- Seamless integration of Stateflow for managing different operational modes of cruise control is crucial.

Difficulties

- Ensuring system stability and a quick response to speed changes is challenging.
- Handling complex interactions between vehicle dynamics and external disturbances requires advanced techniques.
- Designing a robust control system that adapts to various driving conditions involves sophisticated modeling and simulation.

2. Control System Design

Function of the Control System

The vehicle cruise control system is designed to automatically regulate the speed of a vehicle to maintain a driver-set speed. This system reduces the need for constant manual throttle adjustments by the driver, enhancing driving comfort and efficiency. The system must adapt to varying conditions such as changes in road slope and wind resistance to ensure consistent speed maintenance.

Design Process

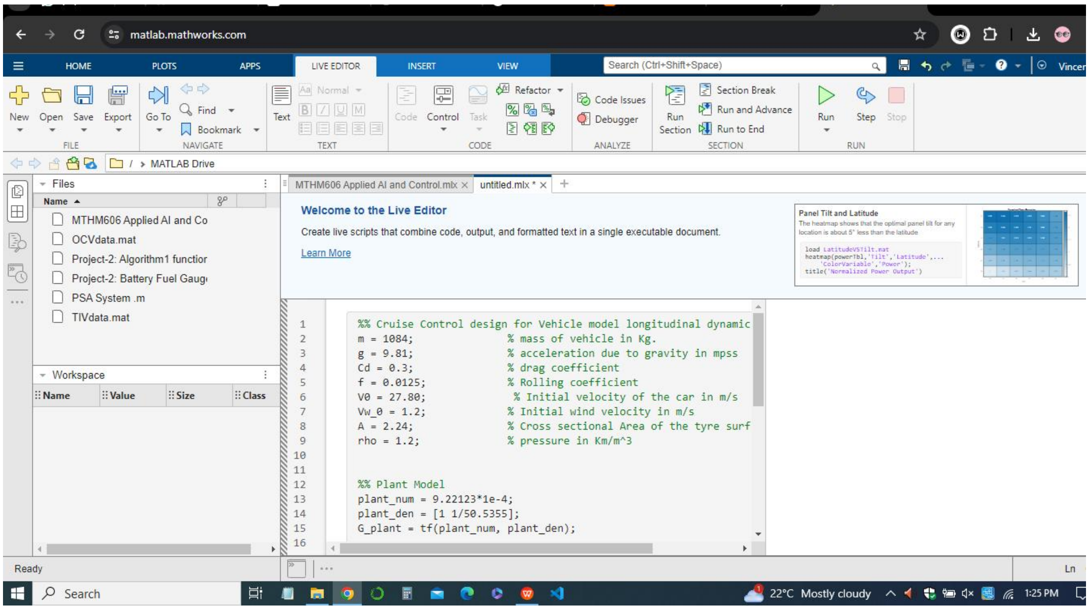
a. *Vehicle Dynamics Modeling*

- Objective- To represent the relationship between throttle input and vehicle speed.

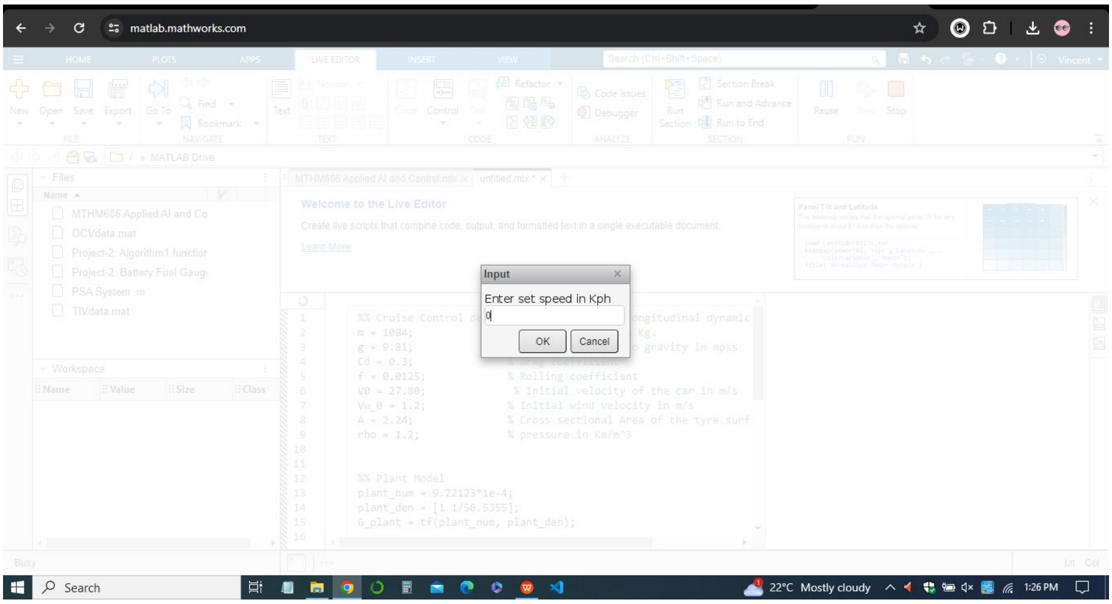
The Components

- Engine Model- Converts throttle position into engine power.
- Vehicle Model-Uses engine power to generate vehicle speed, considering resistive forces like aerodynamic drag and road slope.

❖ Mathematical Model-Use differential equations to describe the vehicle dynamics.



b. *Speed Controller Design*



✧ Objective: To maintain the desired speed set by the driver.

✧ Controller Type: PID (Proportional-Integral-Derivative) controller.

✧ Tuning: Adjust the PID gains (Kp, Ki, Kd) to achieve optimal performance, ensuring quick response and minimal overshoot.

c. *Stateflow State Machine*

❖ Objective-To manage different operational modes of the cruise control system.

The States

- **Off-** System is inactive.
- **On-** System actively maintains the set speed.
- **Standby/Set-** System is ready but not maintaining speed (e.g., after a brief stop).
- Transitions- Based on driver inputs such as setting the speed, resuming, or canceling cruise control.

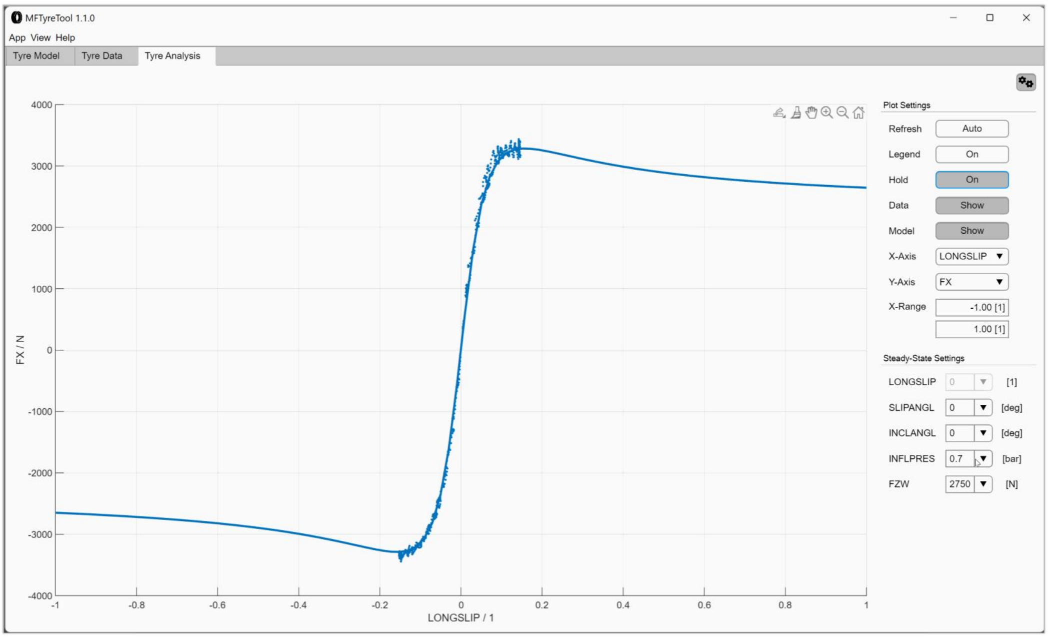
Simulink Model Integration

- Model Integration- Combine the vehicle dynamics, speed controller, and stateflow state machine in Simulink.
- Simulation Setup- Define simulation parameters and scenarios, including different road conditions and driver inputs.

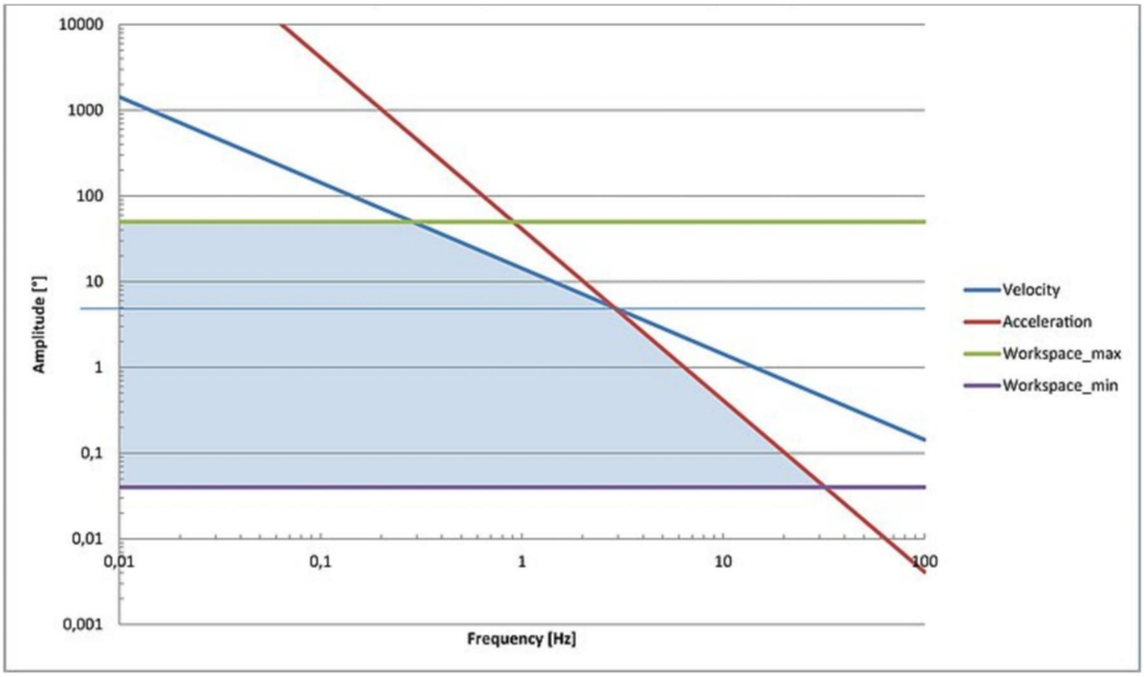
<u>Operation Results</u>

Simulation Scenarios

- ✓ Constant Speed Maintenance- The system successfully maintains the set speed on a flat road.



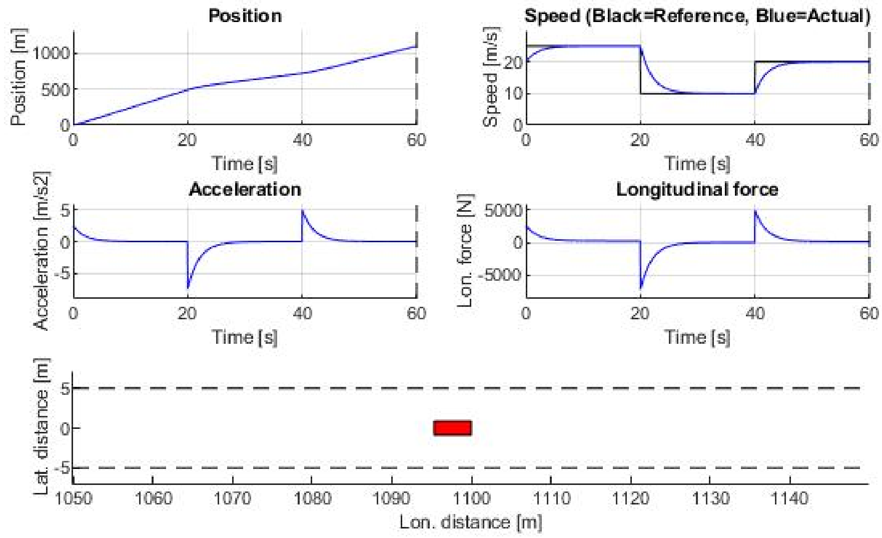
✓ Adaptation to Slopes- The system adjusts throttle input to maintain speed when encountering uphill and downhill slopes.



✓ Response to Disturbances- The system quickly corrects any deviations in speed due to external disturbances like wind resistance.

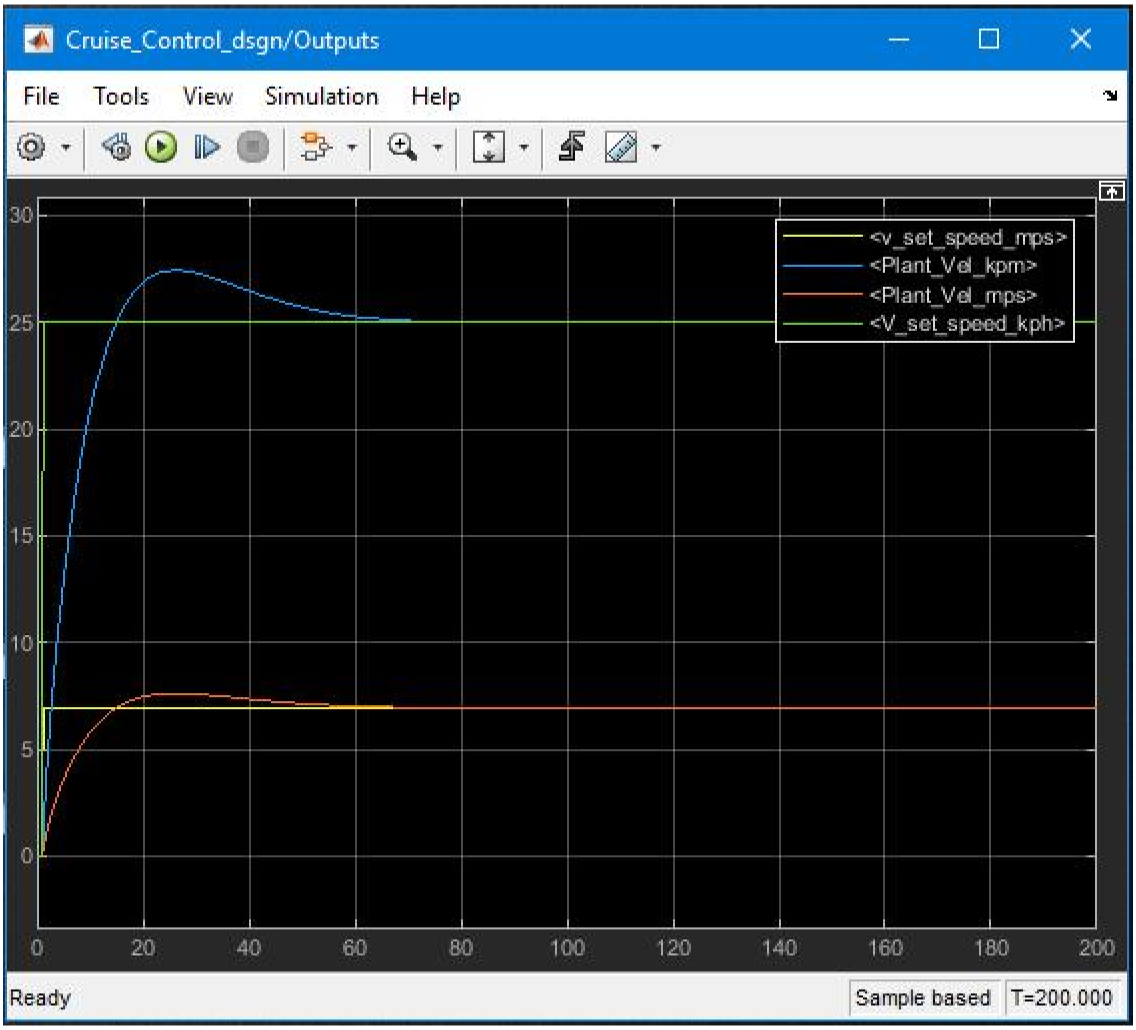
<u>Performance Metrics</u>

- *Speed vs. Time Plot*- Demonstrates how well the system maintains the desired speed over time.



- *Control Effort*- Shows the throttle adjustments made by the PID controller to maintain the set speed.
- *System Stability*- Confirms that the system remains stable and does not oscillate or overshoot excessively.

Result Analysis



- *Effectiveness*-The system maintains the desired speed accurately under various conditions.
- *Efficiency*- Minimal control effort required to counteract disturbances and maintain speed.
- *Robustness*- The system handles changes in road conditions and external disturbances effectively, ensuring a smooth and comfortable driving experience.

By following this design process, the vehicle cruise control system effectively maintains a set speed, demonstrating the practical application of control theory and Simulink modeling in automotive systems.

3. Robot Modeling: Six-Axis Linkage Robot

Objective

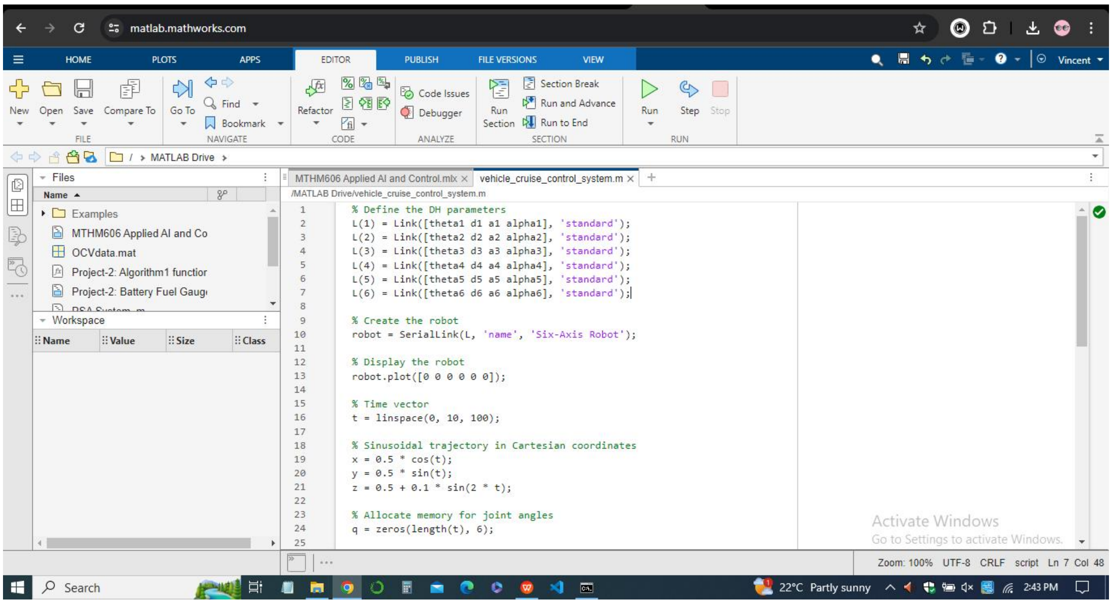
To model a six-axis linkage robot using the Denavit-Hartenberg (DH) parameter method and simulate the end effector drawing a sinusoidal trajectory. This involves establishing the DH parameters, performing DH modeling, and outputting the robot model and trajectory in MATLAB.

DH Parameters

Joint	a (m)	alpha (rad)	d (m)	theta (rad)
1	0.5	0	0.5	theta1
2	0.3	-1.5708	0	theta2
3	0.2	0	0.6	theta3
4	0.4	-1.5708	0	theta4
5	0.2	1.5708	0	theta5
6	0.1	-1.5708	0.1	theta6

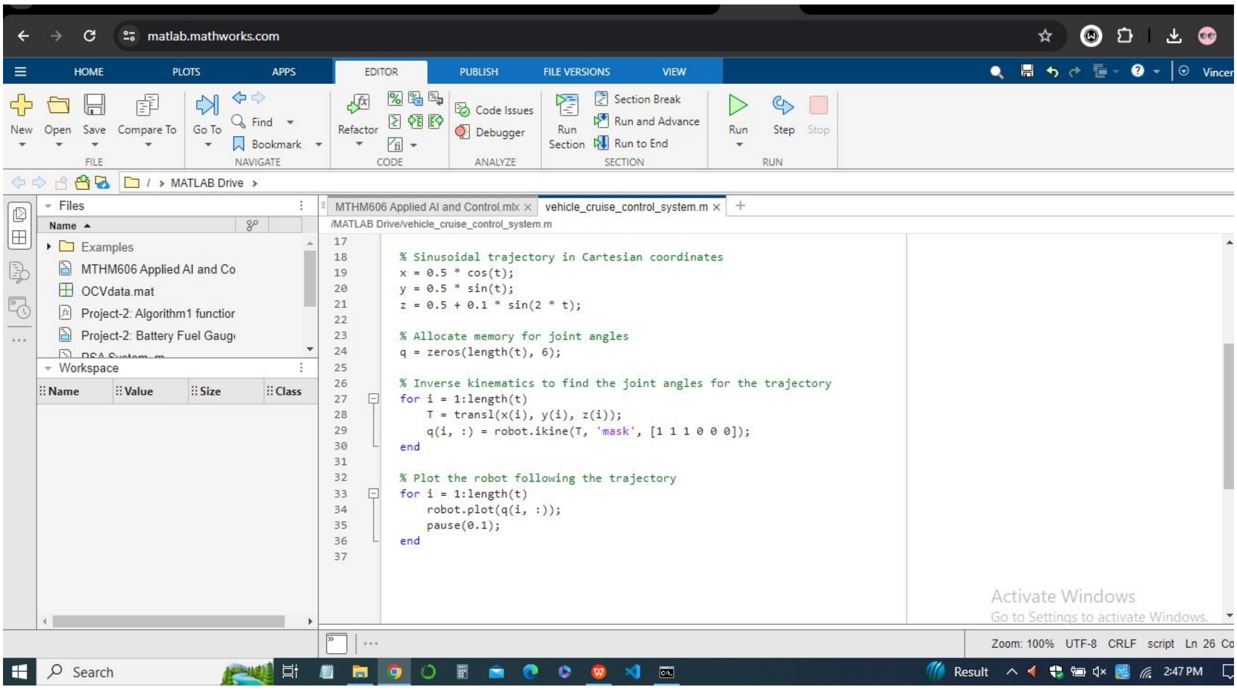
MATLAB DH Modeling Process

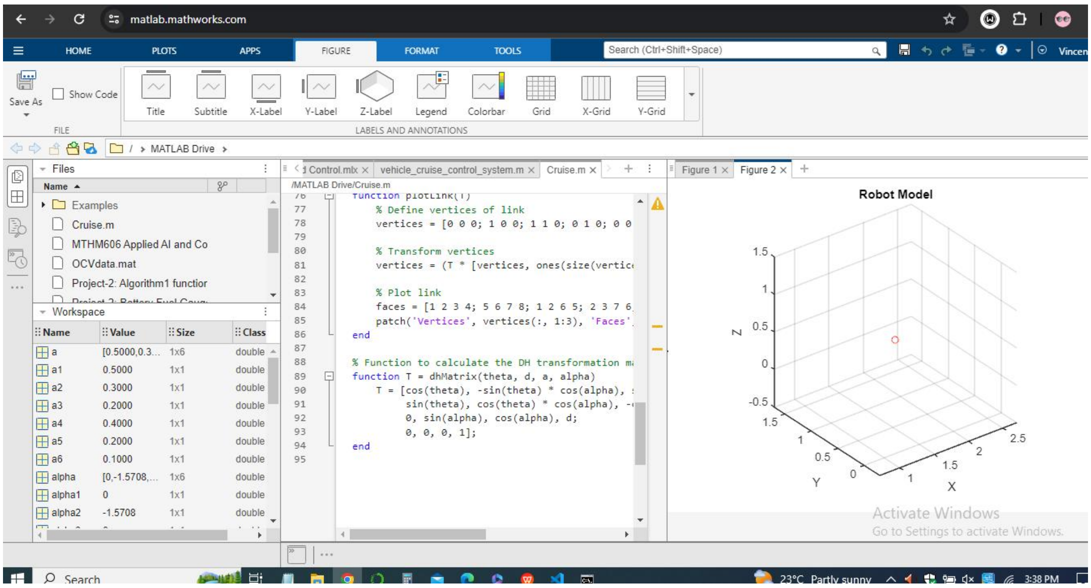
- Define the Robot Links- Use the SerialLink class from the Robotics System Toolbox in MATLAB to define each link of the robot based on the DH parameters.
- Assemble the Robot- Combine the links to form the complete robot model



Sinusoidal Trajectory

- The trajectory for the end effector was designed to follow a sinusoidal path.
- Inverse kinematics was used to calculate the required joint angles for the end effector to follow this trajectory.





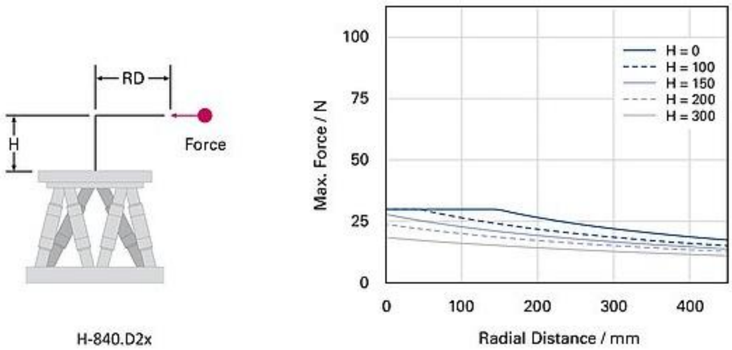


Figure: Six-axis linkage robot arm of Vehicle cruise control system