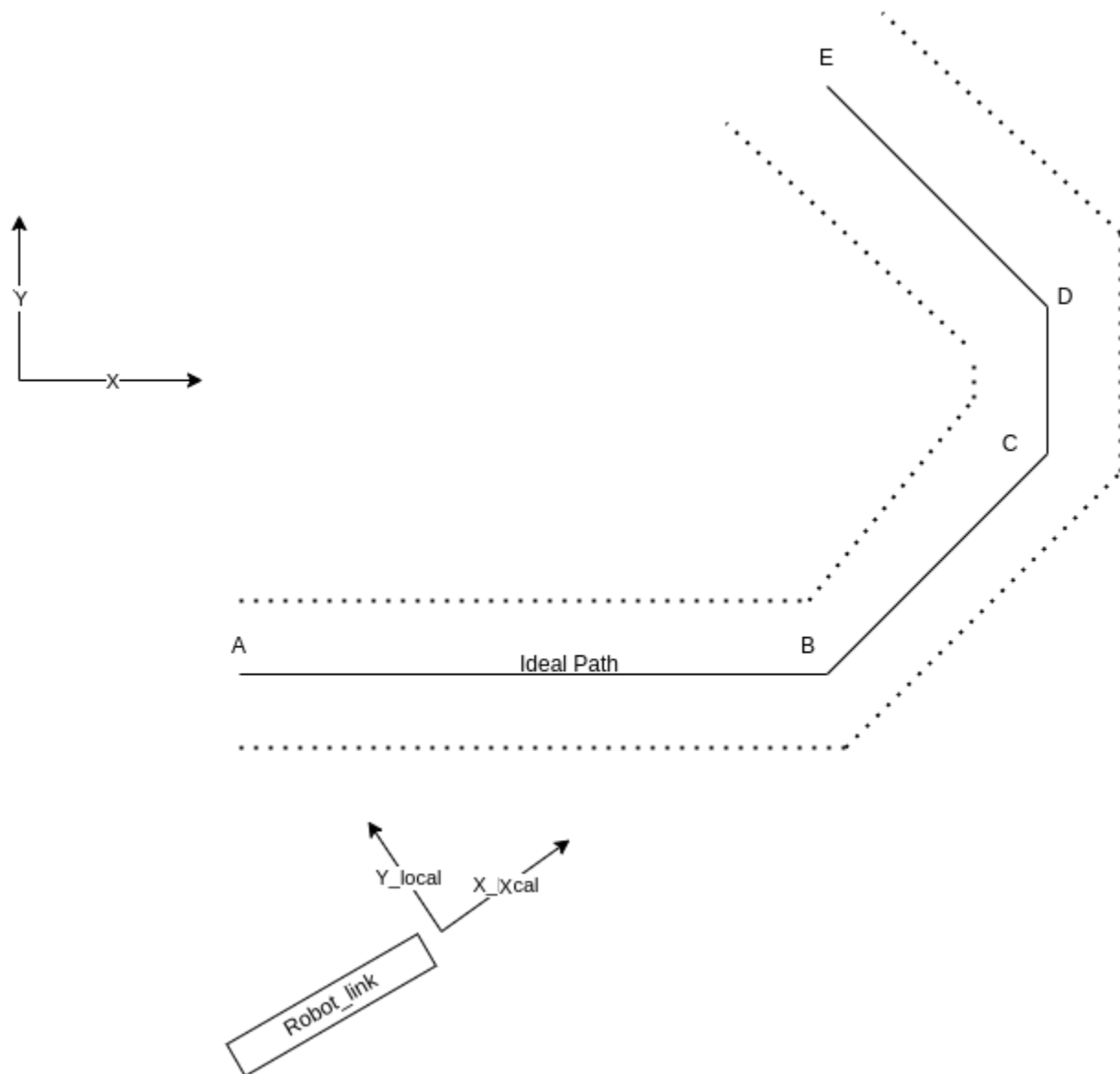


# Path Planning Technical Challenge

Develop a simple path planning algorithm that calculates the next ideal pose for a robot link in 3 dimensions. In the diagram below, we show a sample set of waypoints that show an “ideal path” that we want the robot link to be on.



- This diagram is in 2D, in reality, it will be 3 dimensions
- The ideal path is defined by waypoints “A, B, C, D, E” as a **solid line**
- The **dotted lines** show an acceptable deviation from the ideal path
  - **[ADDED]** Acceptable deviation is not fixed, this can change for different use-cases. Assume a range from 1CM to 30CM

Calculate the next pose and orientation of the robot link with the following behavior:

- The pose of the robot link should be parallel to the nearest segment of the ideal path
  - For example, if the robot is closest to segment AB, the local X axis should be parallel to AB
- The robot link can only move by a distance of 10cm at a time
- The next position of the robot link should always move further down the waypoint path until it eventually reaches the end (Waypoint E)
- **[ADDED]** Calculate the rotation between the current pose and the next pose

Presentation:

- **Write good code**
  - Treat this challenge as if it will be handed off to a client
- Use Python3, feel free to use standard libraries
- Make the system adaptable to an arbitrary number of points (A, B, C, D, E)
- Create a visualization to show the results, as well as an easy way to input points (Make your own decision on how to visualize)
- Add clear and concise documentation, note any assumptions made
- Write tests to show it in action
  - Unit tests, systems tests
- Screen record the results using OBS (<https://obsproject.com/>)
- Commit the project to github