AURORA ROBOTICS CORE WORKSHOP 1.0

ROBOTIC ARMS: KINEMATICS

GENERAL CLASS RULES

- All mics should be muted
- I am always looking at the chat box so feel free drop questions anytime
- Feel free to use the “raise your hand” option anytime
- I recommend you have a note (could be digital), I tend to drop random knowledge casually

All slides, code and materials will be shared in the training repo. Feel free to use and share but do not modify

(I am very good with lawsuits)

ROBOTICS: NOT JUST HARDWARE

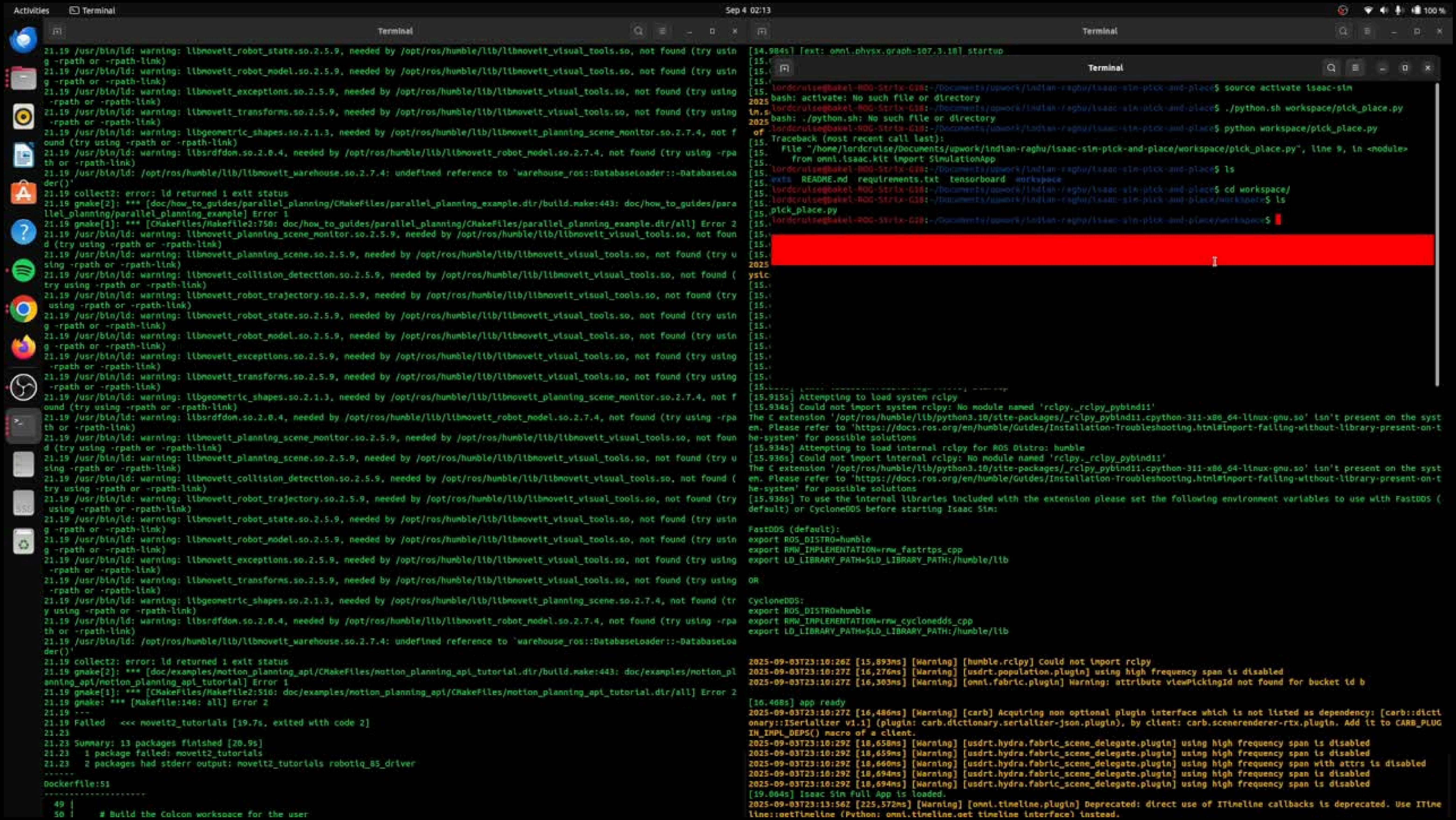
SOFTWARE IS AS HARD AS HARDWARE

Welcome to what runs the hardware of robotics. SOFTWARE

The pain and suffer of setup you already experienced
Shey body tell una last week abi?

HOW IMPORTANT IS ROBOTIC ARM SIMULATION IN REAL LIFE?

REAL LIFE CASE STUDY OF AN INDIAN COMPANY



HOW IMPORTANT IS ROBOTIC ARM SIMULATION IN REAL LIFE?

- Safe testing before hardware risk
  - The company's real robot is expensive, and mistakes would be costly.
  - By simulating in Isaac Sim, we could test trajectories, workspace reach, and motion planning without buying or risking the hardware.

- Simulation helped answer: Can the arm reach all required positions?

- Will it collide with surrounding equipment?
  - This meant they could adjust link lengths, end-effector designs, or task layouts before touching or buying the physical robot arm. Or just buy another type of arm.

- Performance benchmarking
  - The company can also measure cycle time (how long it takes to complete a pick-and-place, for example).
  - This showed if their robot could meet throughput requirements before deploying on the factory floor.

HOW IMPORTANT IS ROBOTIC ARM SIMULATION IN REAL LIFE?

- Developing and debugging control code
  - The same ROS2/Isaac SDK interfaces used in simulation also connect to the real robot.
  - That means the motion planning, IK, or trajectory generation code tested in sim was later ported (sometimes directly, sometimes with minor tuning) to the real hardware.

- Digital twin foundation
  - What you built in Isaac Sim acted like a digital twin, a virtual copy of their real robot.
  - It allowed iterative improvements: tuning planners, testing different grippers, and integrating with vision systems, all in software.

HOW IMPORTANT IS ROBOTIC ARM SIMULATION IN REAL LIFE?

Do they use the simulated codes on the Real Robot?

- Yes, indirectly.
    - The motion planning scripts, kinematics solvers, and ROS2 nodes written in Isaac Sim could be deployed to the real robot with little to no modification, because Isaac Sim supports the same ROS2 interfaces used in practical robotics.

- In practice, the workflow is:
    - Develop in Isaac Sim (safe & fast).
    - Validate trajectories and code in sim.
    - Deploy the same ROS2 package to the real robot (with only calibration/tuning differences).

So, the simulation is never just “for show” neither is it the whole deal, it was a direct stepping stone to real-world deployment.

WHY ALL THESE STRESS SEF?

Before we go far, let us understand why we even need to go through all these troubles because of robotic arms

- Industrial Applications
  - Robotic arms are the backbone of manufacturing (cars, electronics, pharmaceuticals, etc.).
  - They perform repetitive, precise tasks better than humans.

- Accuracy and Repeatability
  - A robotic arm can place a chip on a circuit board hundreds of times with sub-millimeter precision.
  - Humans get tired; robots don’t.

WHY ALL THESE STRESS SEF?

- Safety
  - Arms do dangerous tasks: welding, handling chemicals, heavy lifting.
  - Keeps humans safe while productivity continues.

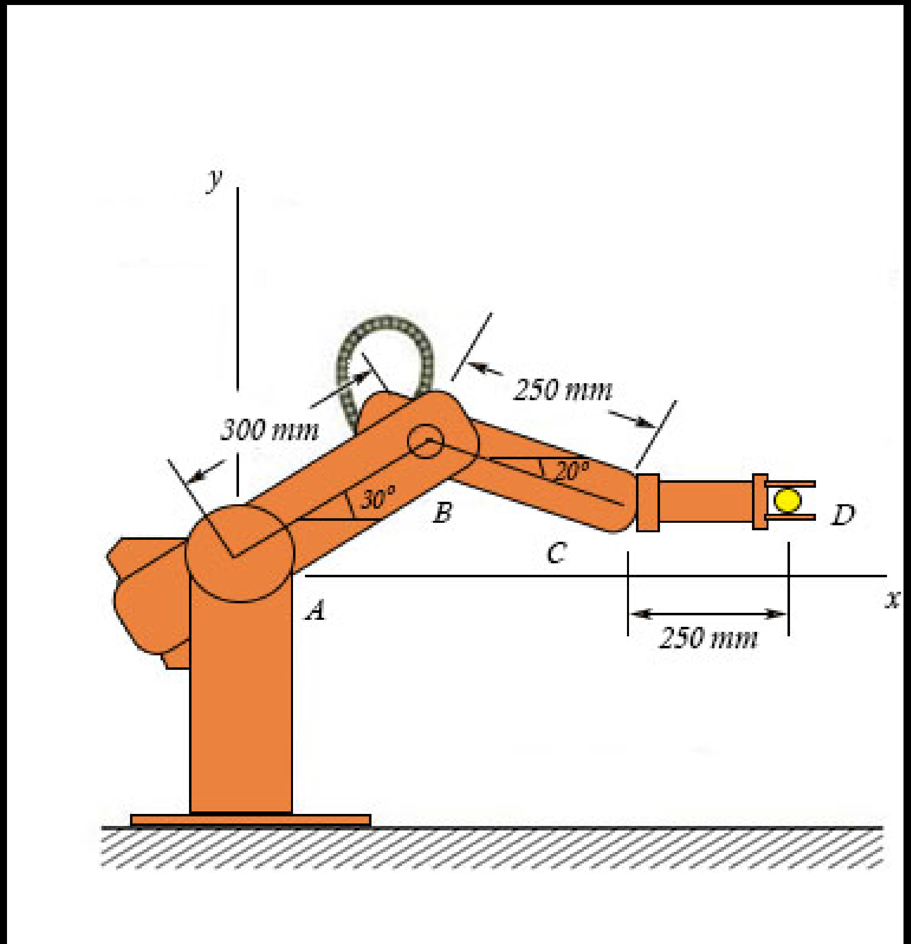
- Scalability & Efficiency
  - Once programmed, the same robot can work 24/7.
  - It scales up production without increasing human fatigue.

- Flexibility
  - Modern robotic arms can be reprogrammed for new tasks.
  - One arm can switch from assembling phones to packaging goods.

So, all this “stress” in software setup, math, and simulation is not for nothing, it’s real world.

THE NEW PROBLEM:
DEFINING ROBOTS IN SOFTWARE

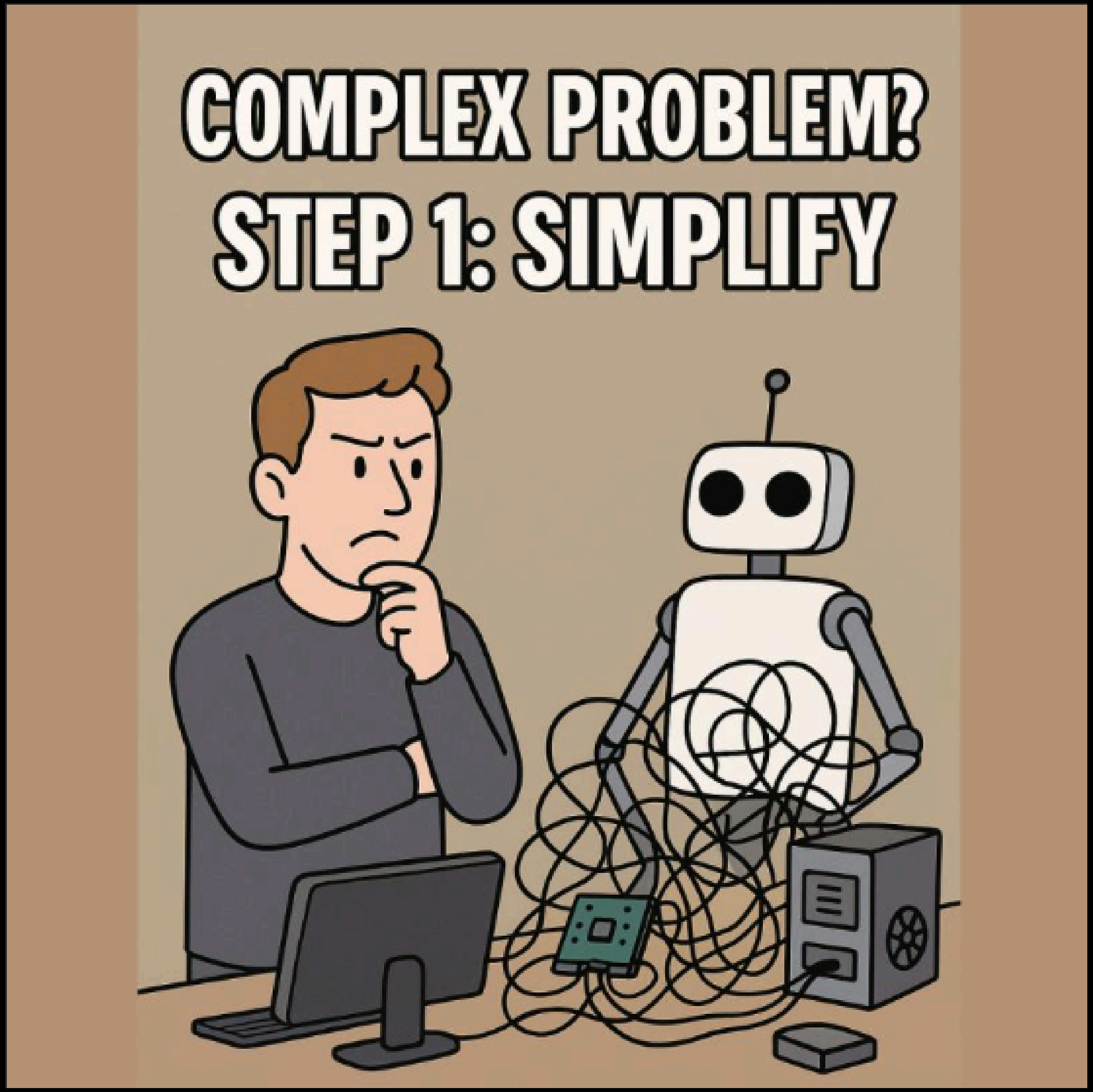
- How do we tell the computer what a robot looks like?
- From physical links to mathematical models
- Computers don’t “see” the arm – we must describe it.
- “If you were the computer, what would you need to know?”
- Links, joints, and geometry.



DEFINING WHAT A ROBOT LOOKS LIKE TO SOMETHING THAT DOES NOT UNDERSTAND WWW.GOOGLE.COM??

BRO HAS TO USE DNS TO TRANSLATE IT TO 142.250.190.4

SO WE NEED TO USE SOME KIND OF NUMBERS

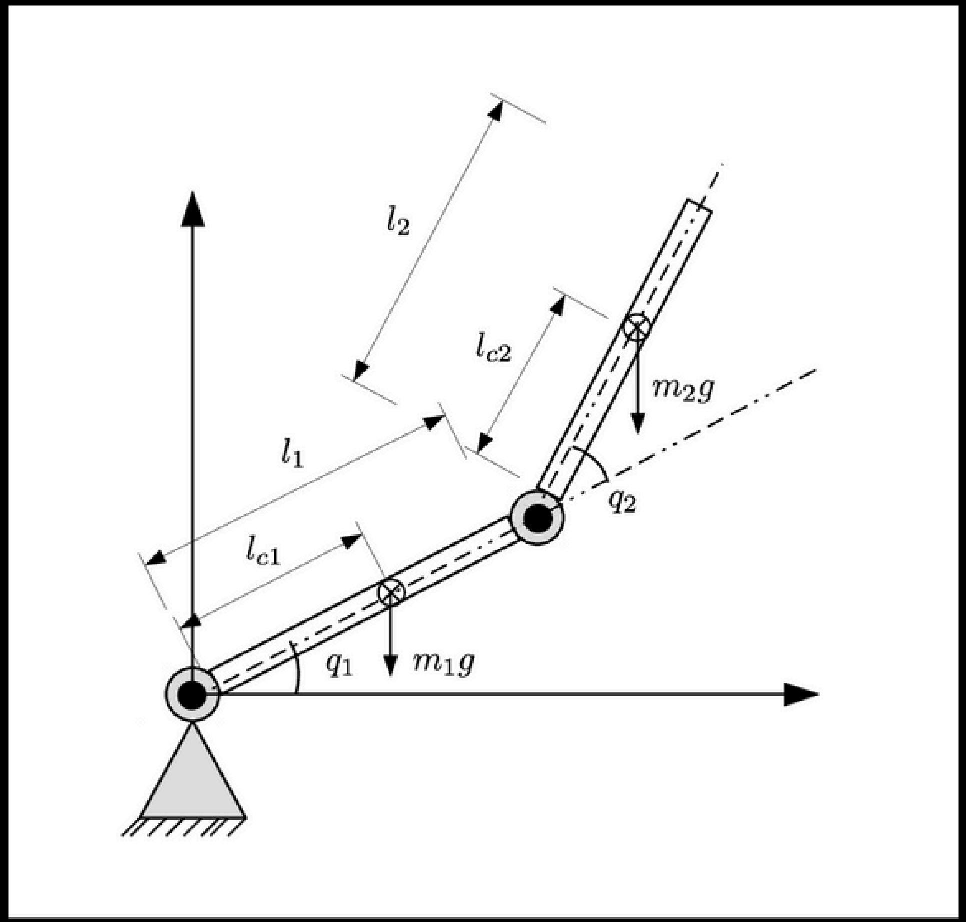


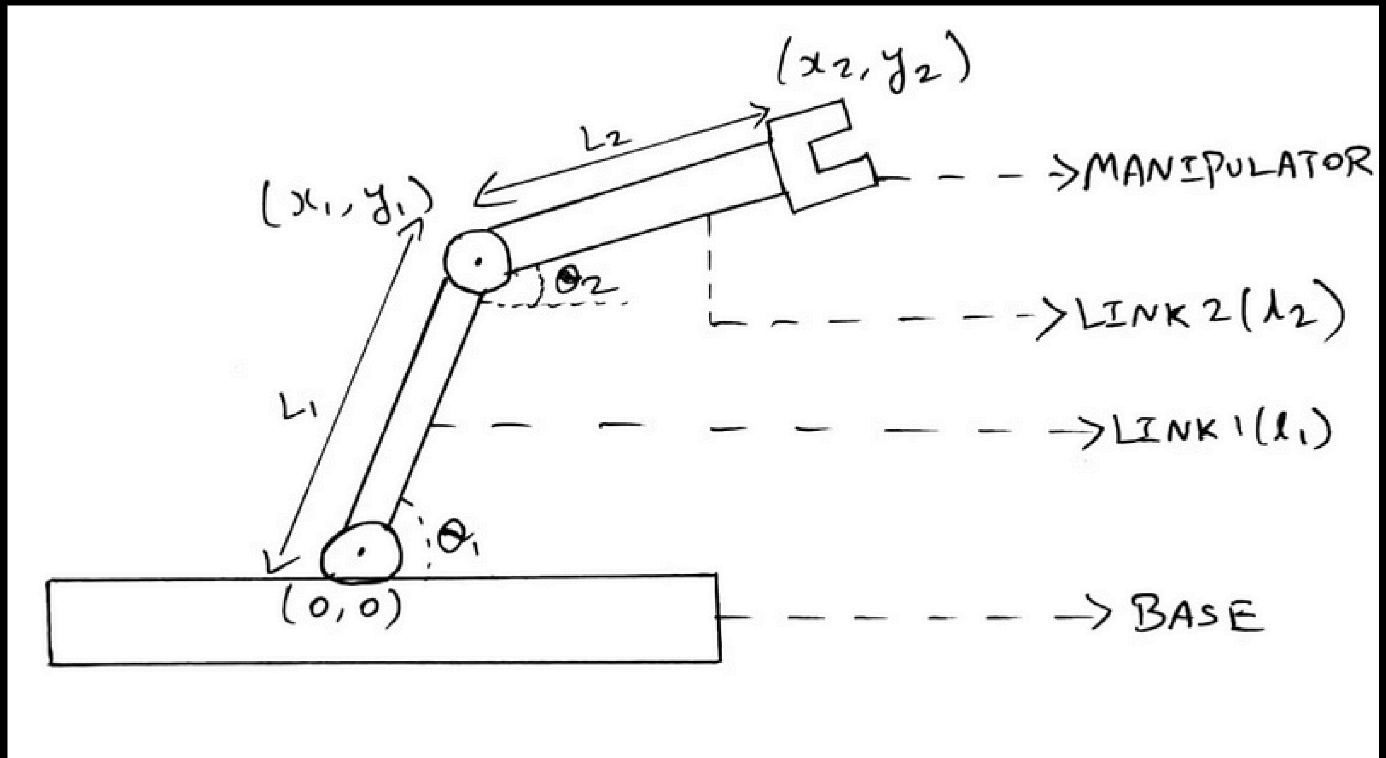
A 2D 2 LINK ROBOTIC ARM

Now, lets start simple: A simple 2D robot arm

Two links, two joints
To define this simple robot to the computer, we need to:

- Define link lengths (L1, L2).
- Define joint angles (θ1, θ2).





FORWARD KINEMATICS: CONCEPT AND MATHS

Subtitles:

- "Given the joint angles, where is the hand?"
- "The math that maps $q \to (x,y)$"

Talk about:

- The input: $\theta 1$, $\theta 2$.
- The output: (x, y) position of the end-effector.
- Write simple formulas:
  - $x = L1 \cos \theta 1 + L2 \cos(\theta 1 + \theta 2)$
  - $y = L1 \sin \theta 1 + L2 \sin(\theta 1 + \theta 2)$
- Emphasize: this is the first bridge between math and robot movement.

CODING SESSION

WE DIE HERE

INVERSE KINEMATICS: THE REVERSE PROBLEM

Subtitles:
"Given the hand’s position, what joint angles achieve it?"
"The challenge: multiple solutions or none"
Talk about:
Example: you want the hand at (x,y). How do we solve for $\theta 1$, $\theta 2$?
Show there may be 2 solutions (elbow up/elbow down).
Sometimes, no solution (target outside reach).

CODING SESSION

- AGAIN, WE DIE HERE!

ADVANCED PROBLEMS:
3 LINKS ROBOTIC ARM

- OGHENEEE

MOTION PLANNING

- “It’s not enough to reach – we must move smoothly.”
- “Joint paths vs Cartesian paths”

COLLISIONS: THE REAL WORLD CONSTRAINT

- “Robots don’t move in empty space”
- “We must avoid hitting themselves and the world”

Talk about:

- Self-collision (elbow hitting forearm).
- Environment collision (arm hitting a wall or table).

PATH PLANNING WITH COLLISIONS

- “Finding a safe and feasible path”
- “Planning = Kinematics + Collision Checking”

Talk about:

- Planner explores possible paths (graph/tree search).
- Rejects those in collision.
- Outputs trajectory that is both kinematically valid & collision-free.
- Example: pick-and-place with obstacles.

OTHER KEY CONCEPTS IN ROBOT ARMS

"Beyond kinematics and planning"

Talk about:

Dynamics: how torques/forces generate motion.

Control: PID, model-based control, torque control.

Sensors: encoders, force/torque sensors, vision.

Grippers/End-effectors: the "hands" of robots.

Autonomy layers: perception → planning → control → execution.

ADVANCED CONCEPTS

Please look it up yourself.

ASSIGNMENT

How to Linkedin

(Brief chat before we install)

THANK YOU FOR LISTENING