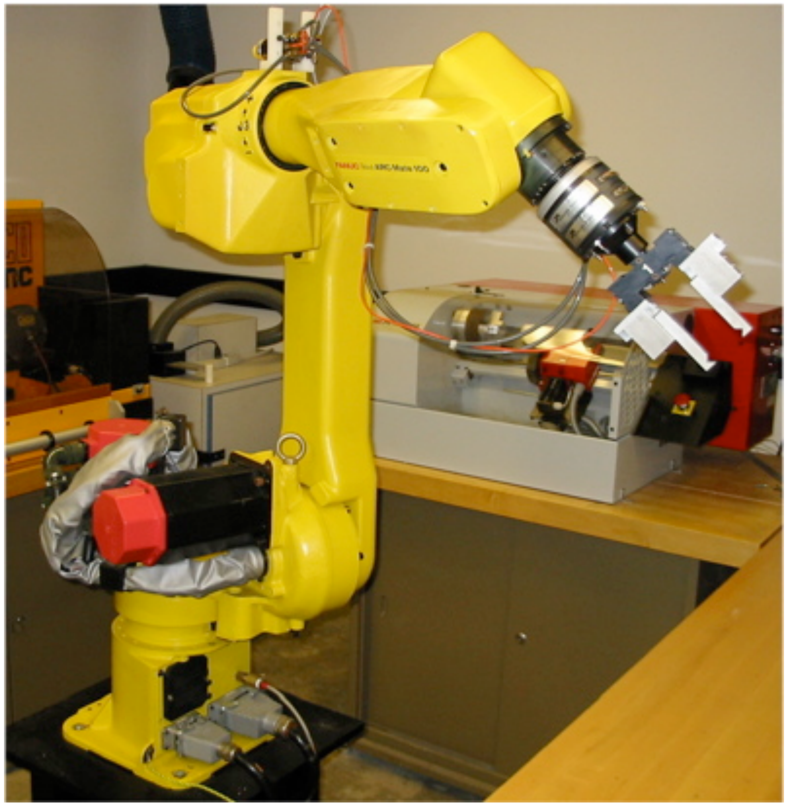
ISE 422/ME 478/ISE 522
Robotic Systems

Overview of Course

R. Van Til
Industrial & Systems Engineering Dept.
Oakland University

What kind of robots will be studied?

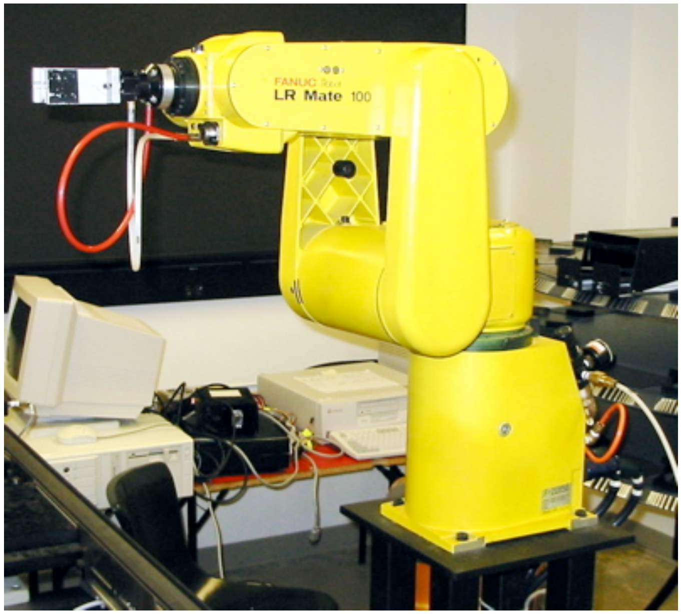


This kind

Not this kind

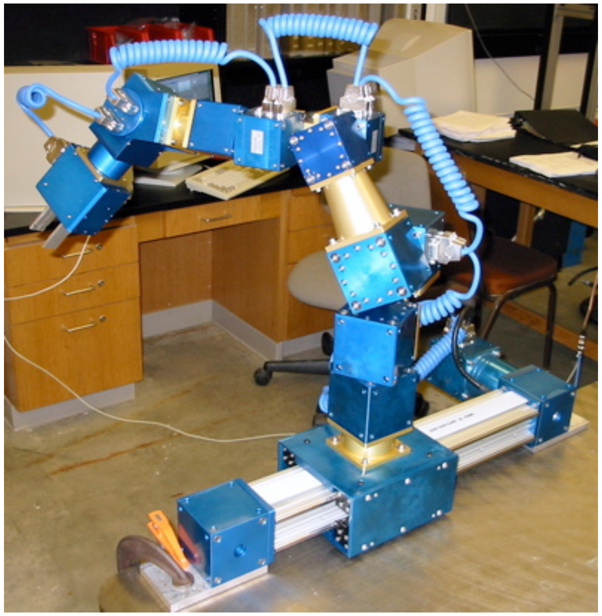
Robots Used in this Course

All robots are located in the S. & R. Sharf
Computer-Integrated Manufacturing Laboratory (room 21 SEB)



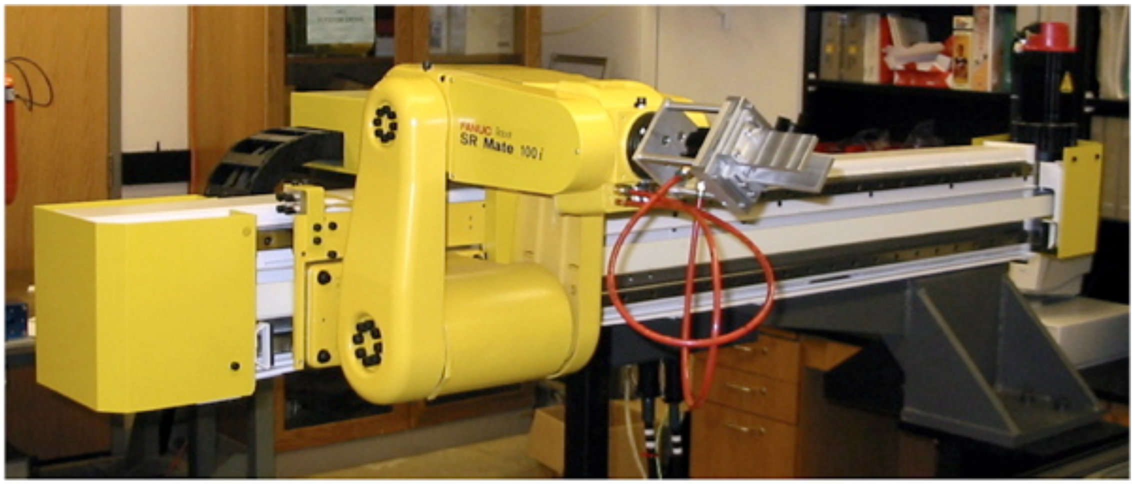
Fanuc LR Mate 100

Kuka KR3

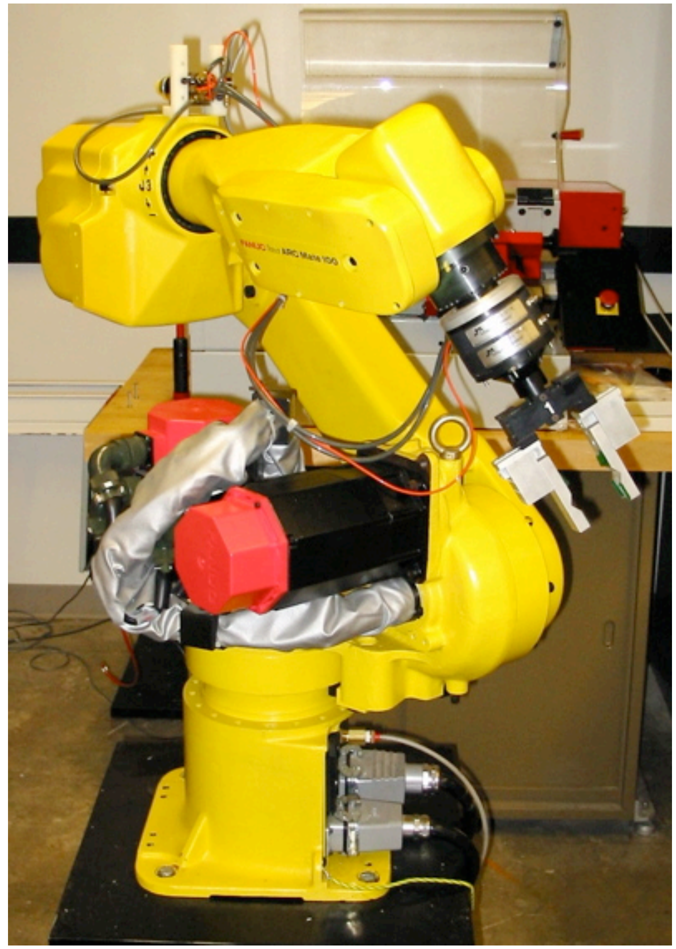


Amtec PowerCube

Robots Used in this Course



Fanuc SR Mate 100i



Fanuc ARC Mate 100

Course Outline

- Overview of robotic systems
- Robot programming
- Modeling 3-dimensional position & orientation
- Denavit-Hartenburg kinematic model
- Forward (or direct) kinematic solution
- Reverse (or inverse) kinematic solution
- Trajectory planning & Jacobian matrix
- Overview of robot dynamics

Robot Programming

- Native robot programming languages are text based.
  - Similar to Basic or C++.
  - Examples include:
    - Karol (Fanuc Robotics)
    - RAPID (ABB Robotics)
  - Few programs are written directly in these native languages.

Robot Programming

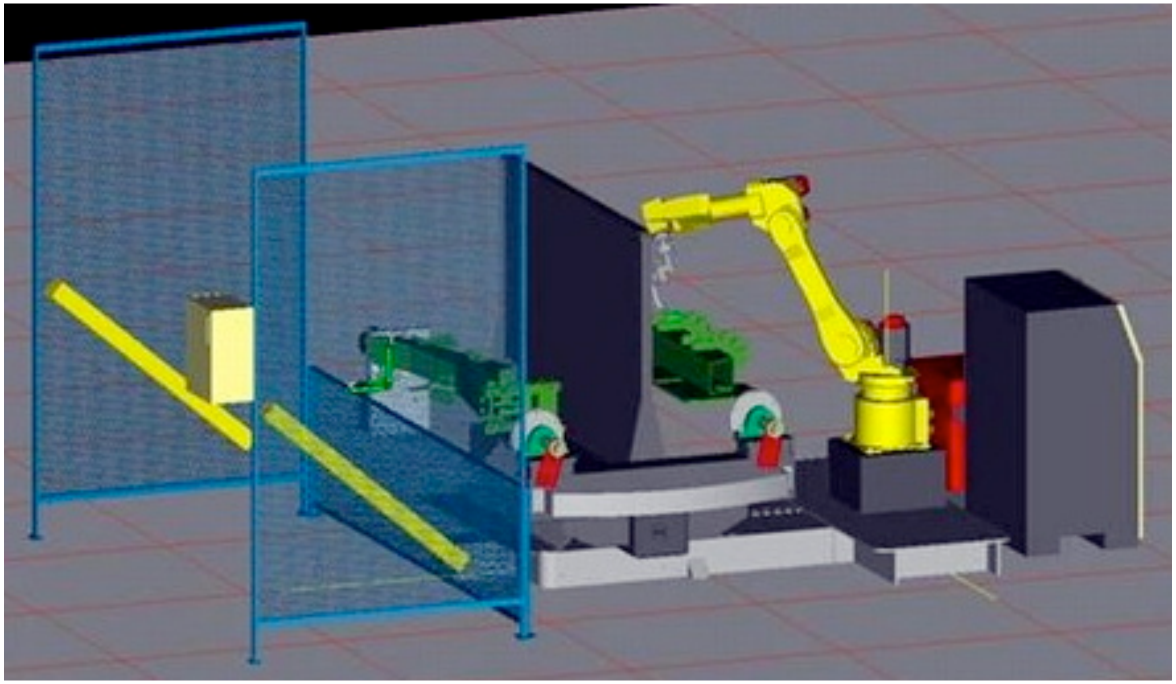
- Example of a Fanuc robot program.

```
R[3] = 1
LBL[4]
RO[1] = OFF
J  P[1]  50%  FINE
PR[2] = P[1]
PR[2,3] = PR[2,3] - 200
L  PR[2]  20%  FINE
WAIT  0.5
RO[1] = ON
WAIT  0.5
L  P[1]  20%  FINE
J  P[3]  100%  CNT50
R[3] = R[3] + 1
IF  R[3] <= 5,  JMP LBL[4]
```

Robot Programming

- Tools used to write robot programs.
  - Teach pendant.
    - » Uses a menu environment to write the robot program.
      - We will use this tool in the Sharf CIM Laboratory.
  - Off-Line Programming (OLP) tools.
    - » Uses a simulation environment to produce a robot program off-line that is then down-loaded to the robot.
      - Examples include I-GRIP (DELMIA) and ROBOGUIDE (Fanuc).

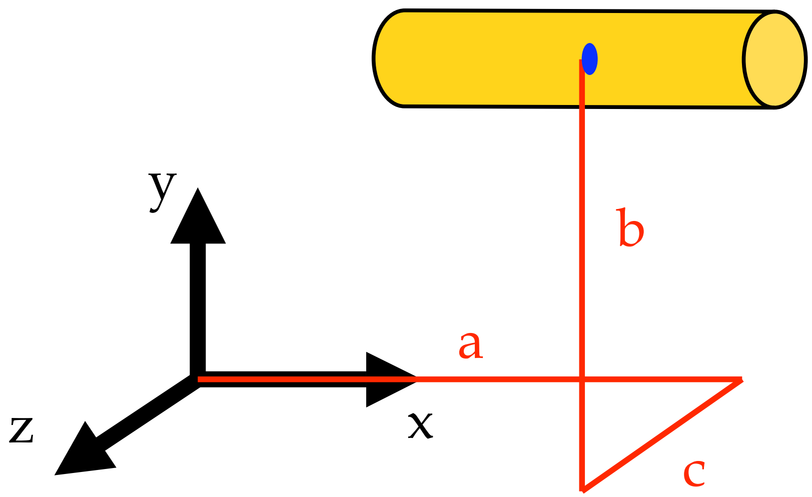
Robot Programming



Roboguide simulation

Modeling 3-Dimensional Position & Orientation

- What's the *position* of the object?

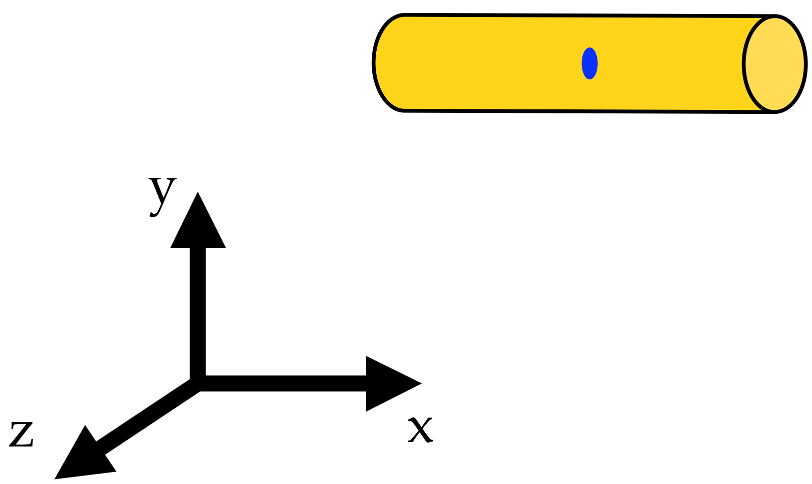


Piece of cake!

(x,y,z) = (a,b,c)

Modeling 3-Dimensional Position & Orientation

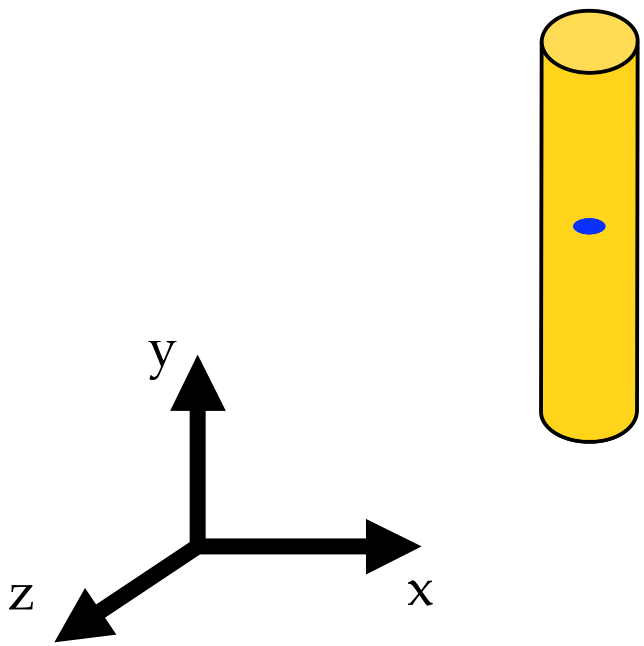
- What's the *orientation* of the object?



You might be a little rusty on this! So, we will do an extensive review concerning orientation.

Modeling 3-Dimensional Position & Orientation

- What's the *orientation* of the object?



Note this object has the same position but a different orientation. Orientation information is very important if the robot has to pick-up this part.

Modeling 3-Dimensional Position & Orientation

- The mathematical techniques used by robotic manipulators to model position and orientation are from linear transformation theory.
  - Position
    - » Modeled using vectors.
  - Orientation
    - » Modeled using matrices.

Difficult for humans to visualize. Hence, also modeled as a set of 3 angles (usually using roll, pitch, yaw angles or Euler angles).

Denivat-Hartenburg Kinematic Model

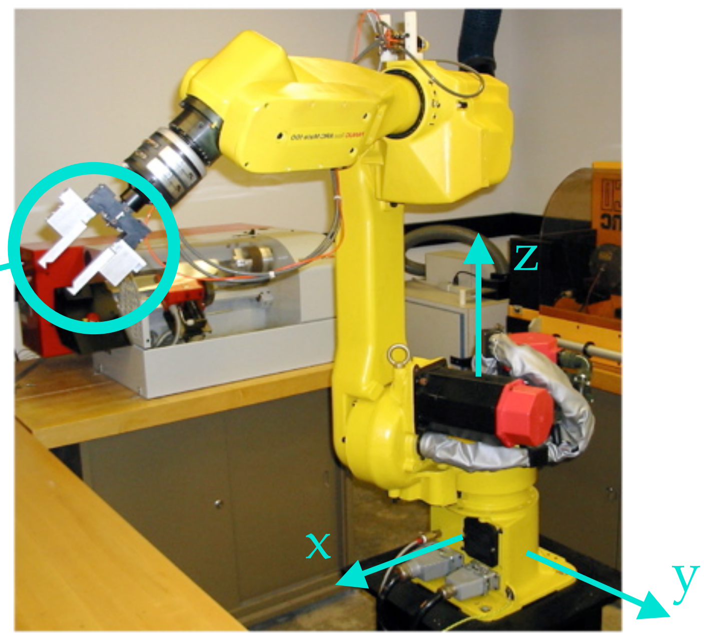
- The *D-H model* presents a standard method for developing a kinematic model of a multi-link chain of rigid bodies.
  - Each link can either rotate about, or translate along, a joint axis.
    - » First, a coordinate system is placed on each of the robot's links following the D-H rules.
    - » Next, a kinematic model is constructed using linear transformation theory techniques.

Kinematic Solutions

- Forward Kinematic Solution (FKS).
  - Given the robot's joint variables, the FKS computes the resulting position and orientation of the tool.

<u>Fanuc ARC Mate 100</u>

Select joint angles (θ_1, θ_2, θ_3, θ_4, θ_5, θ_6), then FKS computes location of the tool.



Kinematic Solutions

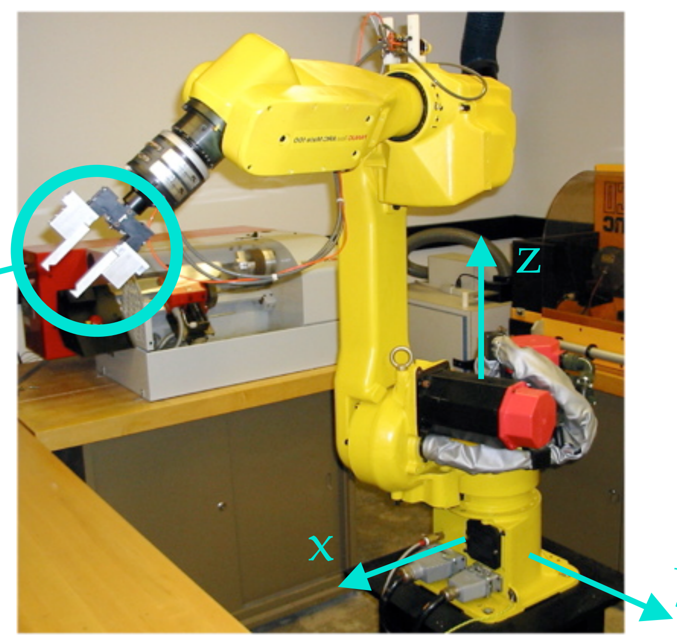
- Forward Kinematic Solution (FKS).
  - The FKS is straightforward to construct using the D-H rules.
  - The FKS consists of a 4×4 matrix that is a function of the robot's joint variables.
    - Each joint variable is either an angle or a displacement.
  - The FKS is also called the *direct kinematic solution*.

Kinematic Solutions

- Reverse Kinematic Solution (RKS).
  - Given a desired position and orientation for the robot's tool, the RKS computes the resulting joint variables.

Fanuc ARCmate 100

Select location for the tool, then RKS computes joint angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ needed to get there.



Kinematic Solutions

- Reverse Kinematic Solution (RKS).
  - It is constructed from the robot's FKS.
  - It consists of an algorithm that contains both mathematical equations and logical statements.
  - The RKS is also called the *inverse kinematic solution*.

Trajectory Planning

- The objective in *trajectory planning* is to make a robot follow (or track) a specified:

position, velocity & acceleration — Called a *trajectory*

  - Trajectory planning is used in two domains:
    1. Joint space.
    2. Cartesian space.

Trajectory Planning - Joint Space

- How to make each individual joint follow a specified position, velocity and acceleration.
  - If the robot has N joints, then the robot controller must make each joint (i = 1, ..., N) follow specified:

$$\theta_i(t),\ \dot{\theta}_i(t)\ \&\ \ddot{\theta}_i(t)\ \text{ - if joint i rotates}$$

or

$$d_i(t),\ \dot{d}_i(t)\ \&\ \ddot{d}_i(t)\ \text{ - if joint i translates}$$

This is not a difficult problem!

Trajectory Planning - Cartesian Space

- How to make both the position and orientation of the robot's end-effector track a specified trajectory in Cartesian space.
  - Requires the controller to coordinate the motion of *all* N joints on the robot.
  - Applications include arc welding, spray painting and applying sealant.

This is a difficult problem!

Trajectory Planning - Cartesian Space

- Methods commonly used in robotics to solve the problem of trajectory planning in Cartesian space include:
  - Point-to-point approach.
  - Jacobian matrix.

Robotic Dynamics

- Robots are extremely complex dynamic systems.
  - ISE 422/ME 478/ISE 522 presents a short overview of robot dynamics.
  - SYS 623 presents a detailed analysis of robot dynamics.