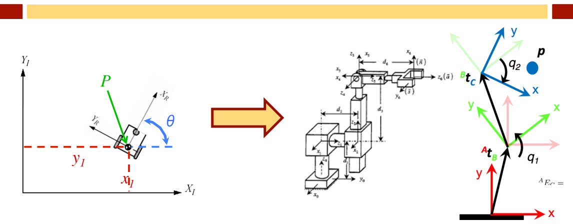
Kinematics

Manipulator Kinematics

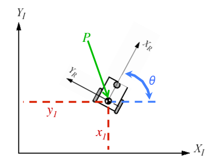


Many slides, graphics, and ideas adapted (with thanks!) from:
Siegwart, Nourbakhsh and Scaramuzza, Autonomous Mobile Robots
Renata Melamud, An Introduction to Robot Kinematics, CMU
Rick Parent, Computer Animation, Ohio State
Steve Rotenberg, Computer Animation, UCSD
Angela Sodemann, www.youtube.com/watch?v=IVjFhNv2N8o, ASU

(A final note on) Mobile Kinematics

2

- Given this setup:



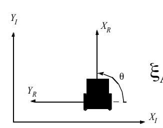
$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

- We can map $\{X_I, Y_I\}$ (global) ←→ $\{X_R, Y_R\}$ (robot)
  - Use rotation matrices and velocity vector in x, y, θ
- Why do we care so much?

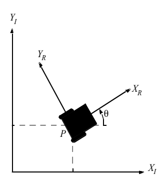
(A final note on) Mobile Kinematics

3

- Goal: take robot from A_I to B_I
  - We know where we want it in the *global* setting
  - What do we actually control? (In what frame of reference?)



$$\xi_A = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$



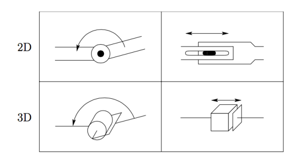
$$\xi_B = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix}$$

- Point: Convert from A_I to B_I by changing ξ_R

Manipulator Kinematics

4

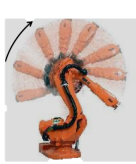
- Kinematics (possible motion of a body) for manipulator robots
  - End effector position and orientation, wrt. an arbitrary initial frame
- A manipulator is moved by changing its…
  - Joints: revolute and prismatic

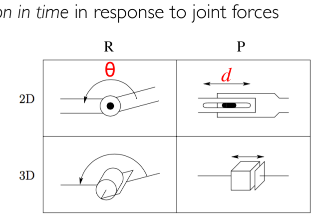


Manipulator State

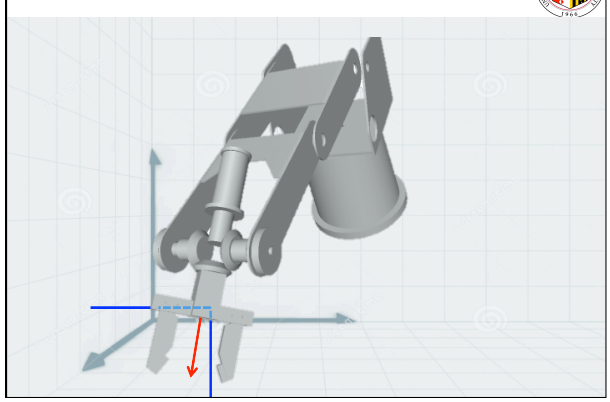
5

- Configuration: where is every point on manipulator?
  - Instantaneous description of geometry of a manipulator
- State: a set of variables which describe
  - *Change of configuration in time* in response to joint forces
    - Control inputs
    - External influences

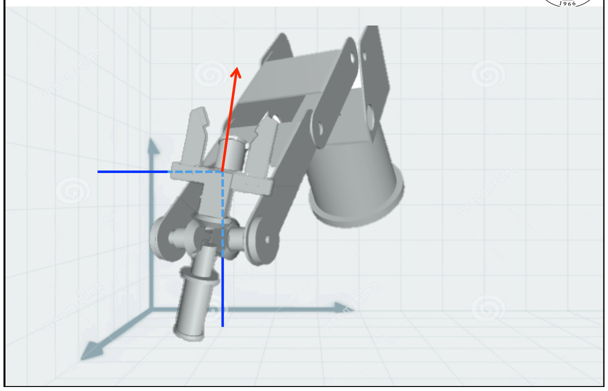




Position & Orientation

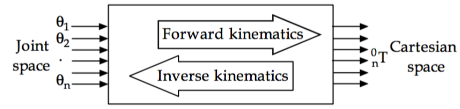


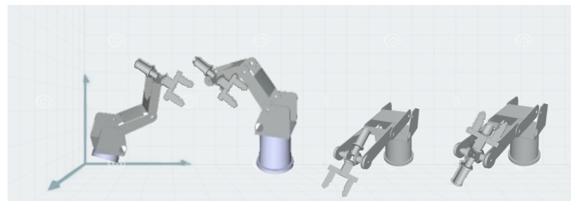
Position & Orientation



Forward Kinematics & IK

8

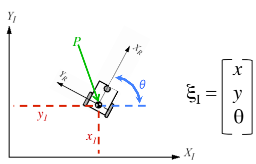


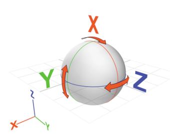


Mobile vs. Manipulator

9

- Description: how many terms…
  - …to describe planar position & orientation?
  - …to describe 3D position & orientation?
- AKA, how many
  - Degrees of freedom

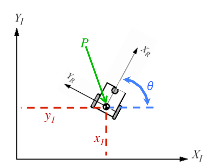




Kinematics Problem

10

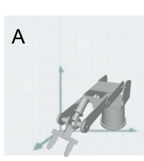
- The **state space** is the set of all possible states
- The **state** of the manipulator is:
  - A set of variables which describe changes in **configuration** over time, in response to joint forces + external forces
- Where do joint forces come from?
  - Controllers!
- So, given some set of joints, what signals do we send?
- In joint space vs. Cartesian space

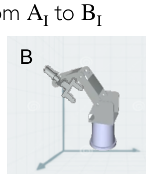


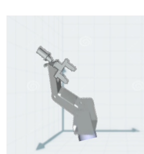
Goal

11

- Goal: take robot end effector from A_I to B_I
  - We know where we want it in the *global* setting
  - What do we actually control? (In what frame of reference?)
- Point: Convert from A_I to B_I







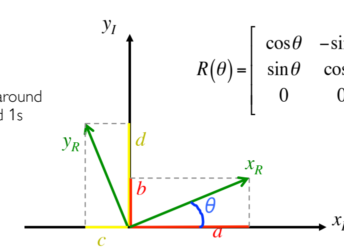
- Now a 6 ←→ 6 transformation

Review: Z Rotation Matrix*

12

- We derived this geometrically in class:
  - If we assume frame axes are of length 1
  - $a = \cos\theta$
  - $b = \sin\theta$
  - $c = -\sin\theta$
  - $d = \cos\theta$
  - Rotations around z → 0s and 1s

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

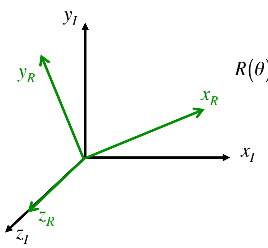


* AKA *orthogonal rotation matrix*

Review: Z Rotation Matrix*

13

- In practice, it's really this:
- Rotations around z → 0s and 1s



$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

* AKA *orthogonal rotation matrix*

Other Rotation Matrices

14

- Similarly derived from axis of rotation and trigonometric values of projections

(Last time I threw in a few 2D rotations around y
2D → a 2x2 matrix
So, don't do that)

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}$$

$$R_Y = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix}$$

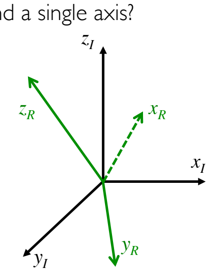
$$R_Z = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Complex Rotations

15

- What if we don't just rotate around a single axis?
- Any rotation in 3D space can be broken down into single-axis rotations
  - Given orthogonal axes
- Multiply rotation matrices!

$$R = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\gamma & 0 & sin\gamma \\ 0 & 1 & 0 \\ -sin\gamma & 0 & cos\gamma \end{bmatrix}$$

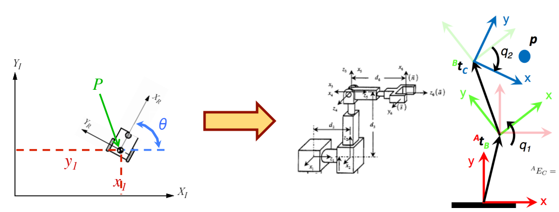


- Can do any number of rotations; just multiply out

Mobile to Manipulator

16

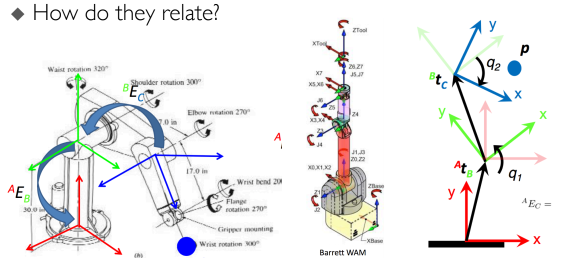
- Add a number of chained frames of reference



Multiframe Kinematics

17

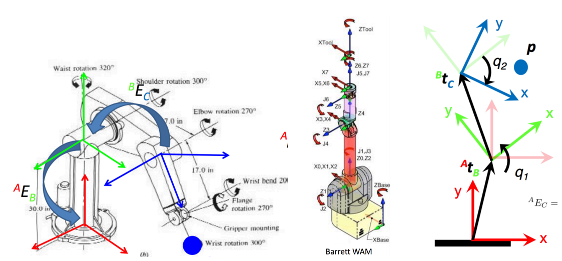
- How many frames of reference do we have?
  - We've been translating among frames based on possible motion
- How do they relate?



Kinematic Chaining

18

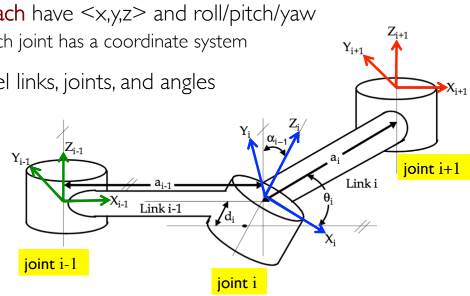
- Do you need to do every transformation?
- What do we really care about?



Describing A Manipulator

19

- Arm made up of links in a chain
- Joints each have <x,y,z> and roll/pitch/yaw
  - So, each joint has a coordinate system
- We label links, joints, and angles



Forward Kinematics

20

- We will sometimes use the vector **Φ** to represent the array of **M** joint values:

$$\mathbf{\Phi} = [\phi_1 \quad \phi_2 \quad \ldots \quad \phi_M]$$

- We will sometimes use the vector **e** to represent an array of **N** values that describe the end effector in world space:

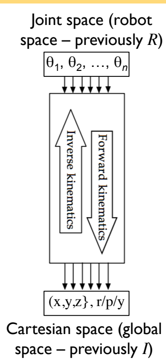
$$\mathbf{e} = [\, e_1 \quad e_2 \quad \ldots \quad e_N \,]$$

- Example: If we need end effector position and orientation, **e** would contain 6 DOFs: 3 translations and 3 rotations. If we only need end effector position, **e** would just contain the 3 translations.

Forward & Inverse

21

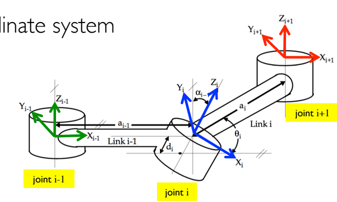
- Forward:
  - Inputs: joint angles
  - Outputs: coordinates of end-effector
- Inverse:
  - Inputs: desired coordinates of end-effector
  - Outputs: joint angles
- Inverse kinematics are tricky
  - Multiple solutions
  - No solutions
  - Dead spots



Describing A Manipulator

22

- Arm made up of links in a chain
  - How to describe each link?
  - Many choices exist
    - DH parameters, quaternions **are** widely used, Euler angles…
- Joints each have coordinate system
  - {x,y,z}, r/p/y



Forward: i → i-1

23

- We are we looking for transformation matrix T, going from frame i to frame i-1:

$$T^{i-1}_{i} \qquad (\text{or } {}^{i-1}_{\ \ i}T\,) \qquad (\text{or } {}^{i-1}T_i)$$

- Determine position and orientation of end-effector as function of displacements in joints
- Why?
  - We can multiply out along all joints

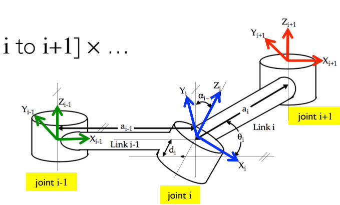
Forward Kinematics and IK

24

- Joint angles ⇆ end effector configuration
- Can string together rotations with multiplication
  - So, can get end effector rotation by ?
- Finding rotation from [joint i-1 to i] × [joint i to i+1] × …

$$\boxed{R^0_2} = R^0_1 R^1_2$$

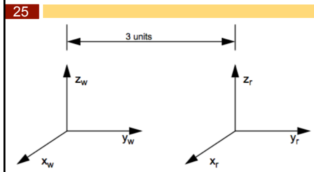
- Rotation of end effector frame, relative to base frame



www.youtube.com/watch?v=lVjFhNv2N8o

Matrices for Pure Translation

25



$$\xi_I = \begin{bmatrix} x_I \\ y_I \\ z_I \\ \theta \end{bmatrix} \quad \xi_R = \begin{bmatrix} x_R \\ y_R \\ z_R \\ \theta \end{bmatrix}$$

Origin point of R in I:

$$\begin{bmatrix} 0 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

In 3D:

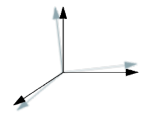
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Generally:

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matrices for Pure Rotation

26



$$\xi_I = \begin{bmatrix} x \\ y \\ z \\ \theta_I \end{bmatrix} \quad \xi_R = \begin{bmatrix} x \\ y \\ z \\ \theta_R \end{bmatrix}$$

Around z:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

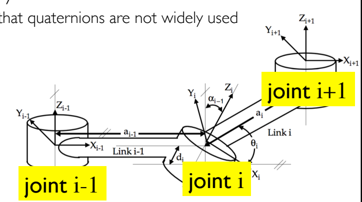
Review?
Introduction to Homogeneous Transformations & Robot Kinematics
Jennifer Kay 2005

ce.aut.ac.ir/~shiry/lecture/robotics/amr/kinematics.pdf

Describing A Manipulator

27

- Arm made up of links in a chain
  - How to describe each link?
  - Many choices exist
  - DH parameters widely used
    - Although it's not true that quaternions are not widely used
- *DH parameters*
  - Denavit-Hartenberg
  - $a_{i-1}, \alpha_{i-1}, d_i, \theta_2$



Denavit-Hartenberg Method

28

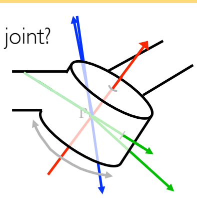
- Efficient method for finding transformation matrices

1. Set frames for all joints
   - This is actually the tricky part.
2. Calculate all DH parameters from frames
   - 4 DH parameters fully define position and orientation (not 6)
3. Populate DH parameter table
4. Populate joint-to-joint DH transformation matrices
   - Matrix for 0-1, matrix for 1-2, etc.
5. Multiply all matrices together, in order
   - 0-1 × 1-2 × 2-3 × …

Defining Frames for Joints

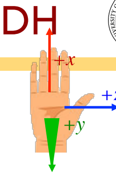
29

- What's the frame of reference for a joint?
  - Actually, completely flexible
- We usually choose:
  - 1 axis through the center of rotation/direction of displacement
  - 2 more perpendicular to that
    - Which can be any orientation!
- We can move the origin
  - P is no longer <0, 0, 0>
- To use DH method, choose frames carefully

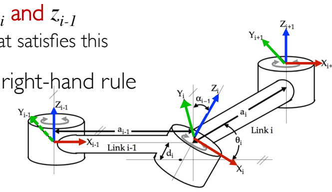


Choosing Frames for DH

30



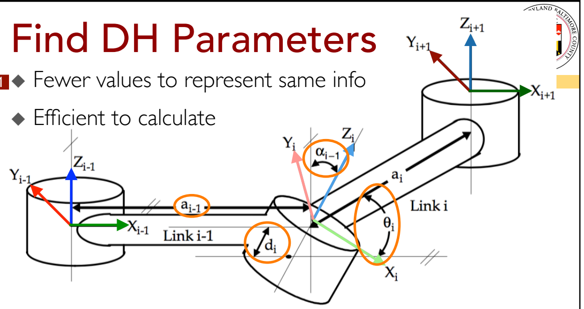
- z axis must be axis of motion
  - Rotation around z for revolute
  - Translation along z for prismatic
- x_i axis orthogonal to z_i and z_{i-1}
  - There's always a line that satisfies this
- y axis must follow the right-hand rule
  - Fingers point $+x$
  - Thumb points $+z$
  - Palm faces $+y$
- x_i axis must intersect z_{i-1} axis (may mean translating origin)



Find DH Parameters

31

- Fewer values to represent same info
- Efficient to calculate



a_{i-1} : link length – distance Z_{i-1} and Z_i along X_i
α_{i-1} : link twist – angle Z_{i-1} and Z_i around X_i
d_i : link offset – distance X_{i-1} to X_i along Z_i
θ_i : joint angle – angle X_{i-1} and X_i around Z_i

Denavit-Hartenberg Method

32

- A way of finding transformation matrix (quickly)

1. Assign DH frames to DoFs (previous slide)
   - This takes practice.
2. Create a parameter table
   - Rows = (# frames – 1)
   - Columns = 4 (always) ← your DH parameters θ, α, a, d

	θ	α	a	d
frame 0-1	$\theta_{0\text{-}1}$	$\alpha_{0\text{-}1}$	$a_{0\text{-}1}$	$d_{0\text{-}1}$
frame 1-2	$\theta_{1\text{-}2}$	$\alpha_{1\text{-}2}$	$a_{1\text{-}2}$	$d_{1\text{-}2}$
frame 2-3	…	…	…	…

Denavit-Hartenberg Method

33

- Given parameter table,

3. Fill out transformation matrix* for each transition:

$$R_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_{i,i+1} & \sin\theta_i\sin\alpha_{i,i+1} & a_{i,i+1}\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_{i,i+1} & -\cos\theta_i\sin\alpha_{i,i+1} & a_{i,i+1}\sin\theta_i \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

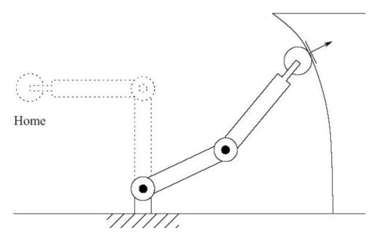
4. And multiply. Ex: $R_2^0 = R_1^0 R_2^1$

- R_2^0 is the same matrix as would be found by other methods. DH is fast and efficient.

** If you'd like the derivation of this, I'll provide a link.*

Example: Rotation in Plane

34



$$x = a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2)$$
$$y = a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2)$$
$a_i =$ the length of ith link

Transformation i to i-1

35

a_{i-1} : distance $\mathbf{Z_{i-1}}$ and $\mathbf{Z_i}$ along $\mathbf{X_i}$
α_{i-1} : angle $\mathbf{Z_{i-1}}$ and $\mathbf{Z_i}$ around $\mathbf{X_i}$
} together: *screw displacement*

$$[X_i] = \text{Trans}_{X_i}(a_{i,i+1})\,\text{Rot}_{X_i}(\alpha_{i,i+1})$$

d_i : distance $\mathbf{X_{i-1}}$ to $\mathbf{X_i}$ along $\mathbf{Z_i}$
θ_2 : angle $\mathbf{X_{i-1}}$ and $\mathbf{X_i}$ around $\mathbf{Z_i}$
} together: *screw displacement*

$$[Z_i] = \text{Trans}_{Z_i}(d_i)\,\text{Rot}_{Z_i}(\theta_i)$$

- Coordinate transformation:

$${}^{i-1}T_i = [Z_i][X_i] = \text{Trans}_{Z_i}(d_i)\,\text{Rot}_{Z_i}(\theta_i)\,\text{Trans}_{X_i}(a_{i,i+1})\,\text{Rot}_{X_i}(\alpha_{i,i+1}),$$

Transformation i to i-1

36

$$\text{Trans}_{Z_i}(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{Z_i}(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Trans}_{X_i}(a_{i,i+1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i,i+1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Rot}_{X_i}(\alpha_{i,i+1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i,i+1} & -\sin\alpha_{i,i+1} & 0 \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation in DH:

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_{i,i+1} & \sin\theta_i\sin\alpha_{i,i+1} & a_{i,i+1}\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_{i,i+1} & -\cos\theta_i\sin\alpha_{i,i+1} & a_{i,i+1}\sin\theta_i \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$