

# Basic Kinematics: Representing Location and Motion

- Basic task- manipulate objects with high degree of precision. Primarily for arm-based robots. Precisely predict and control robot effectors. Requires that all the pieces be rigid: no rubber arms, etc.
- Approach-
  - define a mathematical space in which the robot exists
  - define a mathematical space of the robots movements
  - derive a mapping between the two spaces.
  - determine the position in space desired of the robot, and calculate how to move the robot to get there.
- Create a 3 dimensional coordinate space  $x,y,z$ . E.G.  $z$  is up,  $y$  is right &  $x$  is down to the left.
- Representations - Represent objects as collection of points of the corners.
  - For example, a cube might have corners at:  $((-1,1,1), (-2,1,1),(-1,2,1),(-2,2,1),(-1,1,2),(-2,1,2),(-1,2,2),(-2,2,2))$
- Transformations-
  - movement of object in space (definition for our purposes).
  - 2 kinds:
    - \* translation- movement in one or more directions in a straight line.
    - \* rotation- movement around a fixed axis.
      - right hand rule for positive/negative rotation.
      - 1st finger is  $x$ , 2nd is  $y$ , thumb is  $z$
      - finger points in positive direction.
      - positive rotation is counter-clockwise rotation around an axis, when the positive portion of the axis poke you in the eye.
- Transforming the location of an object means transforming the locations of ALL the points representing the object.
  - In the case of the cube, one movement results in 8 transformations.

- 3 movements results in 24 transformations.
- To move an n-cornered object through m movements requires n\*m transformations.
- To reduce the computational load, we create a *new local coordinate frame*.
  - Define a new coordinate system where one corner of the object is at the origin
  - in our cube, the corners would be at:  $((0,0,0), (-1,0,0),(0,1,0),(-1,1,0),(0,0,1),(-1,0,1),(0,1,1),(-1,1,1))$ .
  - Keep track of ONE corner in the global *reference frame*. (The original frame we described above.)
  - The origin above would be at location  $(-1,1,1)$  in the reference frame.
  - Now we perform transformations on just the one point in the reference frame. When the final destination is reached, we generate all 8 points of the cube by performing 8 transformations from the local frame, resulting in n+m total transformations.
- Review matrix multiplication.
- Basic transformations
  - Transformations will be implemented as the product of a point (represented as a column vector) and a matrix (representing the transformation) resulting in a new point. We will start in 2 dimensions so the math will be easier, and scale up to 3 dimensions later on.

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} (ax + cy) \\ (bx + dy) \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

The identity matrix does not change the point:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

- look at various matrices:
    - \* Scale- moves a point in a direction proportional to that point's distance from the origin in each dimension
    - \* Rotation- rotates a point around the origin.
- Scale in the x direction by a:

$$\begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} ax_0 \\ y_0 \end{bmatrix}$$

Scale in the y direction by d:

$$\begin{bmatrix} 1 & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_0 \\ dy_0 \end{bmatrix}$$

Scale in both x & y:

$$\begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} ax_0 \\ dy_0 \end{bmatrix}$$

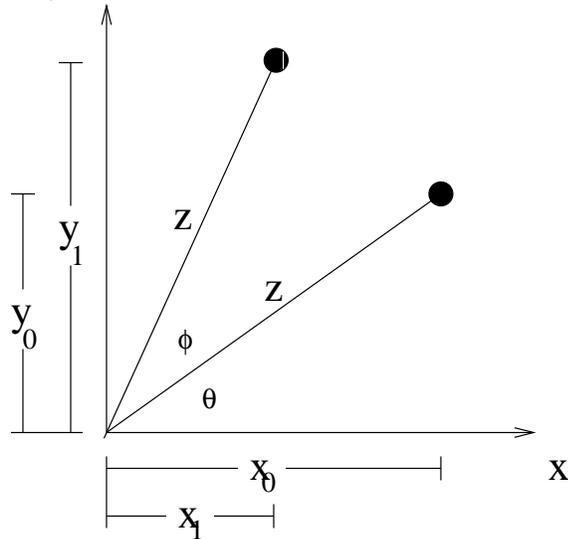
rotate  $\phi$  degrees:

$$\begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} \cos(\phi)x_0 - \sin(\phi)y_0 \\ \sin(\phi)x_0 + \cos(\phi)y_0 \end{bmatrix}$$

For example, if  $\phi = 90$ , then the rotation matrix is:

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} -y_0 \\ x_0 \end{bmatrix}$$

\* Why?  
y



$$\begin{aligned} \sin(\phi + \theta) &= \frac{y_1}{z}, \\ \cos(\phi + \theta) &= \frac{x_1}{z}, \\ \sin(\phi + \theta) &= \sin \phi \cos \theta + \cos \phi \sin \theta, \\ \cos(\phi + \theta) &= \cos \phi \cos \theta - \sin \phi \sin \theta, \\ \sin \theta &= \frac{y_0}{z}, \\ \cos \theta &= \frac{x_0}{z}, \\ \frac{y_1}{z} &= \frac{x_0}{z} \sin \phi + \frac{y_0}{z} \cos \phi, \\ \frac{x_1}{z} &= \frac{x_0}{z} \cos \phi - \frac{y_0}{z} \sin \phi, \\ y_1 &= x_0 \sin \phi + y_0 \cos \phi, \\ x_1 &= x_0 \cos \phi - y_0 \sin \phi. \end{aligned}$$

- This is pretty good, but we have a problem: We don't normally want to change  $x$  by some scale, but *translate* by some *constant factor*. (i.e. we don't want  $a \cdot x$ , but rather  $a+x$ )
- The trick is to add a dimension to the point (and make all the transformation matrices  $3 \times 3$ ). Then put the translation factors in the third column:  
general translation:

$$\begin{bmatrix} 1 & 0 & m \\ 0 & 1 & n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 + m \\ y_0 + n \\ 1 \end{bmatrix}$$

We can combine this with rotation to create a general transformation matrix:

$$\begin{bmatrix} \cos(\phi) & -\sin(\phi) & m \\ \sin(\phi) & \cos(\phi) & n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi)x_0 - \sin(\phi)y_0 + m \\ \sin(\phi)x_0 + \cos(\phi)y_0 + n \\ 1 \end{bmatrix}$$

- A mathematician will tell you this is "projecting the problem to a higher dimensional space." It's nice to realize that what this means is "employ a little trick based on how matrix multiplication works." This trick is called Homogeneous Matrices.

- Three dimensional matrices.

- We are normally interested in 3 dimensional spaces.
- Scale up 2-D to the obvious 3-D matrices:

$$Trans(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(x, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(y, \phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(z, \phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To perform a series of transforms, multiply the matrices.
  - ${}^R T_N = Rot(y, 90)Trans(0, 0, a)Rot(x, 90)Trans(0, a, 0)$ .
  - Transforms from reference coordinate from to new coordinate frame.
  - Note that matrix multiplication is *NOT* commutative.
  - Each multiplication performs a single transformation, resulting in a new frame. The next transformation is with respect to that new frame!!
    - \* Rotate around the y axis in R (the reference frame), 90 degrees, resulting in a new frame  $N_1$ .
    - \* Translate along the z axis in  $N_1$ , a units, resulting in a new frame  $N_2$ .
    - \* Rotate around the x axis in  $N_2$ , 90 degrees, resulting in a new frame  $N_3$ .
    - \* Translate along the along the y axis in  $N_3$ , a units, resulting in N.
  - Note that this is not just a point, but a local frame that is moving.
  - This can be interpreted right to left as transformation that ALL took place in the reference frame:
    - \* Translate along the y axis in R, a units.
    - \* Rotate around the x axis in R, 90 degrees.
    - \* Translate along the z axis in R, a units.
    - \* Rotate around the y axis in R, 90 degrees.
  - If we were to design a series of transformation, we could either do it left to right (relative), or right to left (absolute) whichever works better for the model.
  - why would we ever need to do more than a single translation and a single rotation?

$$T = \begin{bmatrix} \cos(90) & 0 & \sin(90) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(90) & 0 & \cos(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(90) & -\sin(90) & 0 \\ 0 & \sin(90) & \cos(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 2a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0 & 1 & 0 & 2a \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- We will describe the location and orientation of an object by its position in its local frame.
- We describe the location and orientation of a local frame by the transformation from the reference frame to the local frame.

$${}^R p = {}^R T_N {}^N p$$

$${}^R p = \begin{bmatrix} 0 & 1 & 0 & 2a \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2a \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- Things we can tell about the transformation matrix:
  - \* the last column is the same as the location of the transformation from (0,0).
  - \* The first column tells us that the new x direction will be in the direction of -z in R.
  - \* the second column tells us that the new y will be in the direction of x in R.
  - \* the third column tells us that the new z will be in the direction of -y in R.
- In general, if we know a point's location in the new frame (x,y,z) and want to know it's location in the original reference frame, we can:

$${}^R p = \begin{bmatrix} 0 & 1 & 0 & 2a \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} y + 2a \\ -z \\ -x \\ 1 \end{bmatrix}$$

- Inverse transformations

- Sometimes we know the location of a point in the reference frame:  ${}^Rq$  but would like to know the the location of the point in some other fram  ${}^Nq$ . For example to find a position of an object relative to a hand, given that we know the location of both the object and the hand in the reference frame.
- If the know the transform to the new frame (the hand), we can do this with the inverse transformation.

$${}^R T_N {}^N q = {}^R q$$

$${}^N q = {}^R T_N^{-1} {}^R q$$

where  $T^{-1}$  is the inverse of  $T$ ,  $TT^{-1} = I$ , the identity matrix.

- The notation I'm using is handing both for figuring out problems as well taking a verbal description and converting to equations. The key is cancelling the superscripts and subscripts.
- Inverting a matrix in general is hard, but fortunately, we have constraints to help us.

- \* The upper left 3x3 matrix describes the rotations of the cordinate frame.  
RRN

- \* To reverse the rotations of the coordinate frame, you just need to transpose the matrix.

- \* The right hand column describes the origin of the new frame w/r/t the reference frame  ${}^R P_N$ .

- \* To reverse this, we note that  ${}^R P_N = -{}^R R_N {}^N P_R$

- \* And  ${}^R R_N = {}^N R_P^T$

- \* **BIG Extra Credit assignment:** Prove that this is the case.

- \* The result is that:

$$\begin{bmatrix} x_x & y_x & z_x & p_x \\ x_y & y_y & z_y & p_y \\ x_z & y_z & z_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} x_x & x_y & x_z & -p_x x_x - p_y x_y - p_z x_z \\ y_x & y_y & y_z & -p_x y_x - p_y y_y - p_z y_z \\ z_x & z_y & z_z & -p_x z_x - p_y z_y - p_z z_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- An example:

- Suppose we know the transformations from the reference frame to the robot, from the robot to the hand as well as from the reference frame to an object to be grasped and from the object frame to a point on the object where it is supposed to be grasped. We want to know what is the transformation from the hand to the grasping point on the object?

- We can set up an equation of transformations to the grasping point on the object, throught the object, and through the robot:

$${}^W T_O {}^O T_G = {}^W T_R {}^R T_H {}^H T_G \text{ (show that cancellation makes equation balance).}$$

Where:

${}^W T_O$  is the transform from the world reference frame to the object,

${}^O T_G$  is the transform from the object to the grasping point on the object,

${}^W T_R$  is the transform from the world to the robot,

${}^R T_H$  is the transform from the robot to the hand,

${}^H T_G$  is the unknown transform from the hand to the grasping point on the object.

Solve:

$${}^R T_H^{-1} {}^W T_R^{-1} {}^W T_O {}^O T_G = {}^H T_G$$