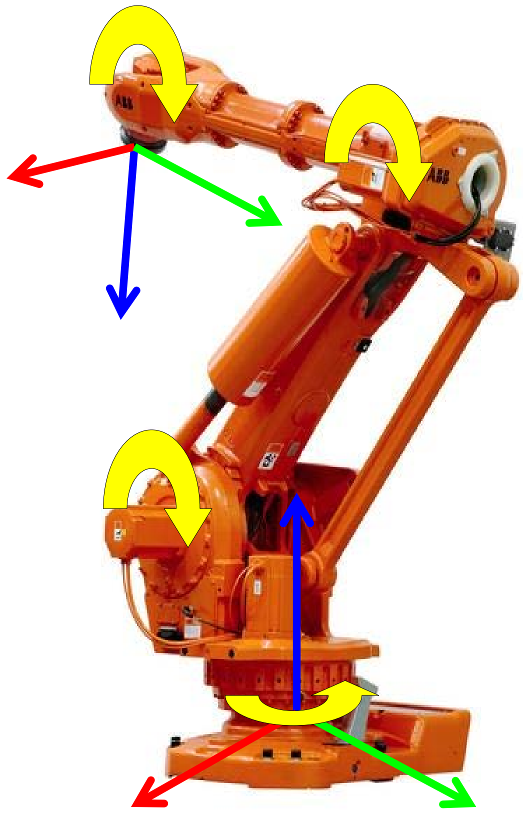
Hand-Eye Calibration

Simon Leonard
Department of Computer Science
Johns Hopkins University

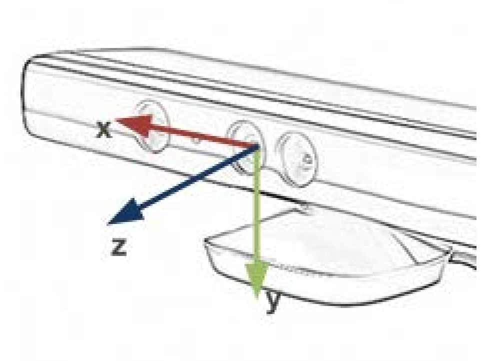
Coordinate Frames

- On robots



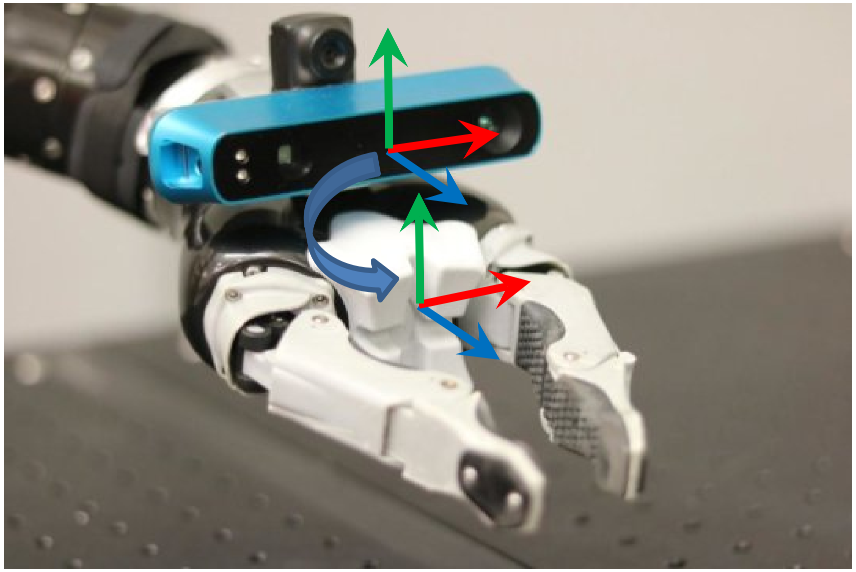
6 axes robot arm

- On sensors

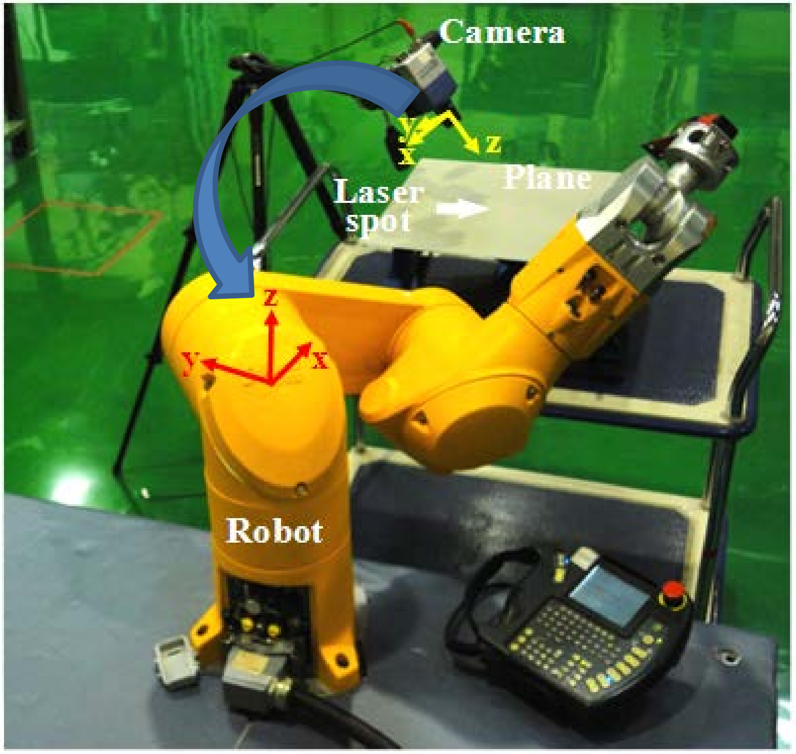


Combining Sensors and Robots

"Eye-in-Hand"



"Eye-to-Hand"

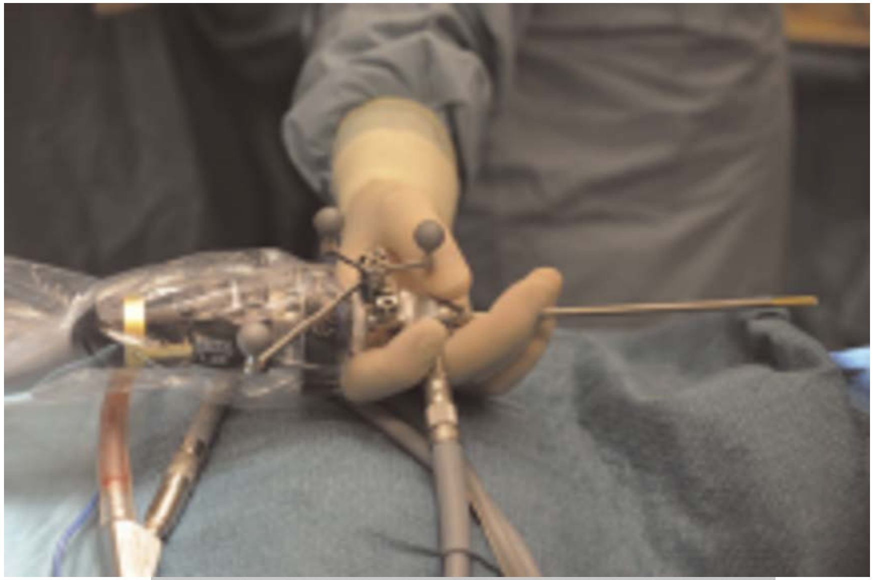


Hand-Eye Calibration

- Find the transformation between the coordinate frame of the sensor and a coordinate frame on the robot
- With few exceptions, this transformation “X” must be known if measurements from the sensor are used to control the motion of the robot

Not Just for Robots

- Use for any sensor that is tracked by some device
  - Navigation system tracks an endoscope in 3D and can "register" the position of the device to the patient's CT
  - Endoscope images are in the camera coordinate frame
  - Hand-eye calibration will relate the images of the endoscope to the patient's CT

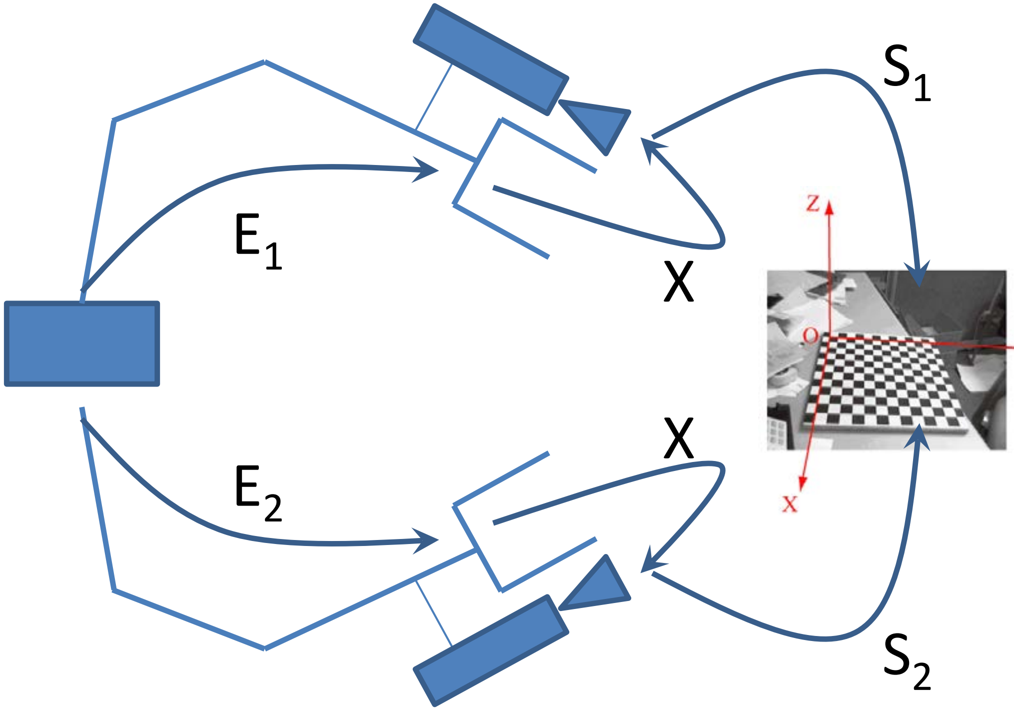


AX=XB

1. Move to robot to position E_1
2. Measure the position of the sensor S_1
3. Move the robot to position E_2
4. Measure the position of the sensor S_2

E_1 and E_2 are given by the forward kinematics

S_1 and S_2 are given by measuring the position of the sensor wrt to the “world”



AX=XB

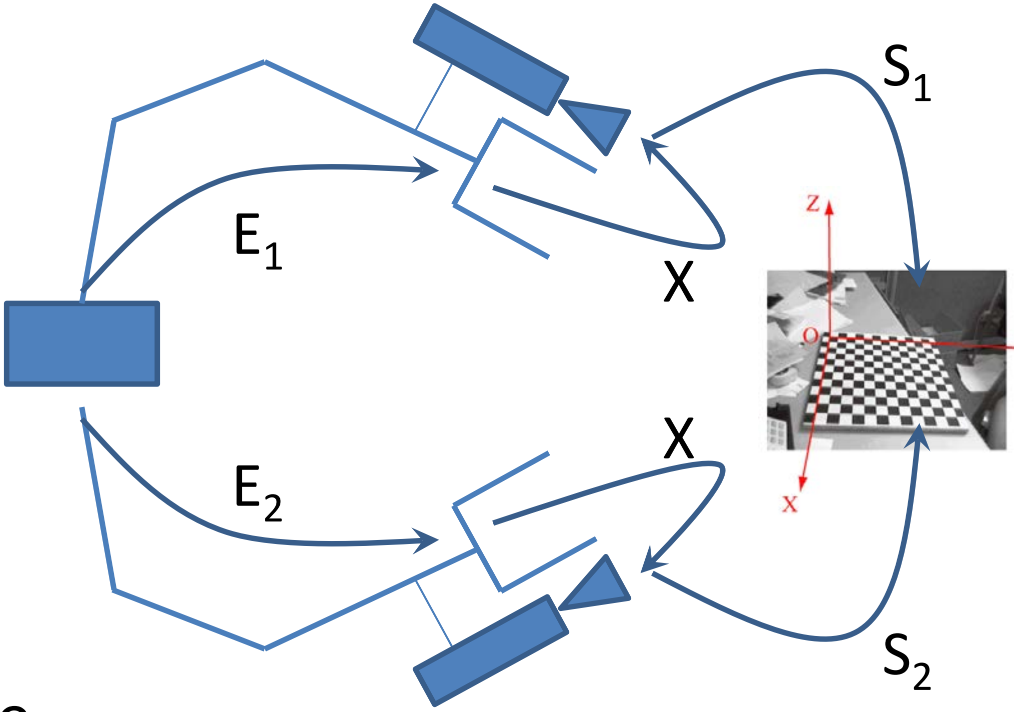
$$E_1 X S_1 = E_2 X S_2$$
$$X S_1 S_2^{-1} = E_1^{-1} E_2 X$$
$$AX = XB$$

Where

$A = E_1^{-1} E_2$ is the relative motion of the robot

$B = S_1 S_2^{-1}$ is the relative motion of the sensor

X is the transformation between the robot and the sensor



Solution to AX=XB

- There are dozens of ways to solve AX=XB
  - Close form solution (quaternion, angle-axis, dual quaternion, etc.)
  - Iterative (non-linear least-squares, convex optimization, etc.)
  - Probabilistic methods

Park and Martin Method

- First solve for the rotation, then solve for the translation

$$\begin{bmatrix} R_A & t_A \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_X & t_X \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_X & t_X \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_B & t_B \\ 0 & 1 \end{bmatrix}$$

The strategy is

1) Solve for the rotation R_x (closed form)

$$R_A R_X = R_X R_B$$

2) Use the solution R_X to solve for t_X using least squares

$$R_A t_X + t_A = R_X t_B + t_X$$

Solving the Rotation

A rotation matrix can be represented by the matrix exponential

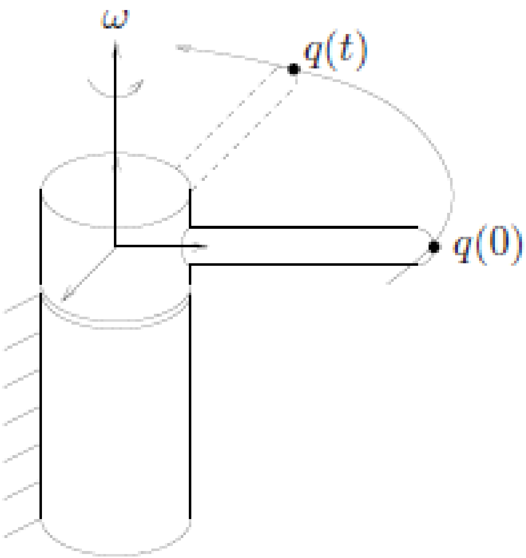
$$R(\omega, \theta) = e^{\widehat{\omega}\theta}$$

where ω is an axis of rotation and θ is a rotation angle

$$\widehat{\log(R)} = \frac{\theta}{2\sin\theta}(R - R^T)$$

with

$$\theta = \cos^{-1}\left(\frac{\text{trace}(R) - 1}{2}\right)$$



Solving the Rotation

Given the axis-angle representation of each rotation

$$\alpha_1 = \log(R_{A1}) \qquad \alpha_2 = \log(R_{A2})$$
$$\beta_1 = \log(R_{B1}) \qquad \beta_2 = \log(R_{B2})$$

we use the identity $\boxed{\alpha_i = R_X \beta_i}$ to build the system of equations

$$\mathcal{A} = R_X \mathcal{B}$$

where

$$\mathcal{A} = [\alpha_1 \quad \alpha_2 \quad \alpha_1 \times \alpha_2]$$
$$\mathcal{B} = [\beta_1 \quad \beta_2 \quad \beta_1 \times \beta_2]$$

and we get

$$\boxed{R_X = \mathcal{A}\mathcal{B}^{-1}}$$

Solving the Translation

$$R_A t_X - t_X = R_X t_B - t_A$$
$$(R_A - I) t_X = R_X t_B - t_A$$

Using A_1, A_2, B_1 and B_2 we get

$$\begin{bmatrix} R_{A_1} - I \\ R_{A_2} - I \end{bmatrix} t_X = \begin{bmatrix} R_X t_{B_1} - t_{A_1} \\ R_X t_{B_2} - t_{A_2} \end{bmatrix}$$

which can be solved by using least squares

More Than Two Data Points

- If we have A_1, …, A_N, B_1, … B_N, then we want to R_x and t_x to "best fit the data"
- We start again by solving for the rotation
- Using a polar decomposition we have

$$R_X = (M^T M)^{-1/2} M^T$$

where

$$M = \sum_{i=1}^{N} \beta_i \alpha_i^T$$

More Than Two Data Points

Note that $(M^T M)^{-1/2}$ can be computed by observing that $M^T M$ is symmetric positive definite and as such it has the diagonal factorization

$$M^T M = Q \Lambda Q^{-1}$$

and

$$(M^T M)^{-1/2} = Q \Lambda^{-1/2} Q^{-1}$$

$$= Q \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{\lambda_3}} \end{bmatrix} Q^{-1}$$

More Than Two Data Points

- Once R_X is found, we solve for t_X using least squares

$$\begin{bmatrix} I - R_{A_1} \\ \vdots \\ I - R_{A_N} \end{bmatrix} t_X = \begin{bmatrix} t_{A_1} - R_X t_{B_1} \\ \vdots \\ t_{A_N} - R_X t_{B_N} \end{bmatrix}$$