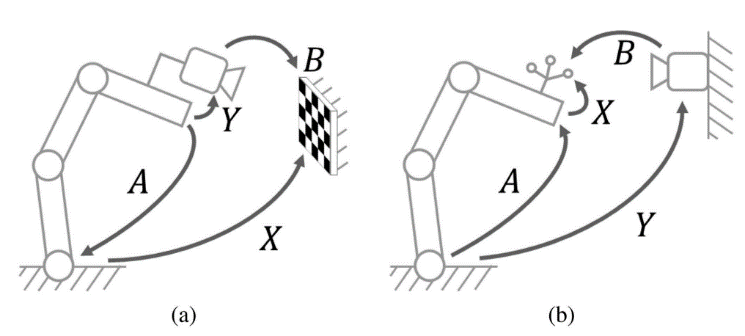
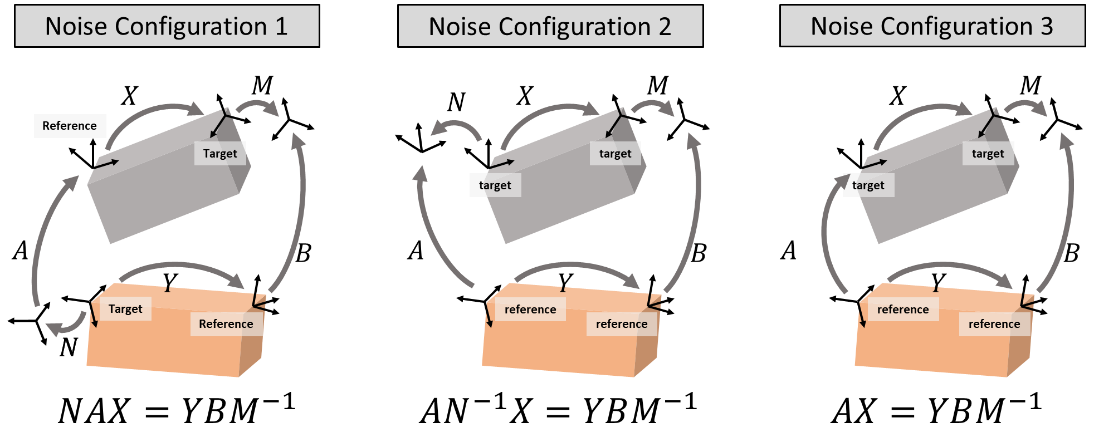
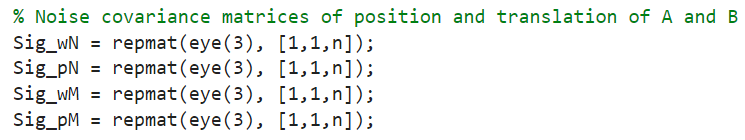
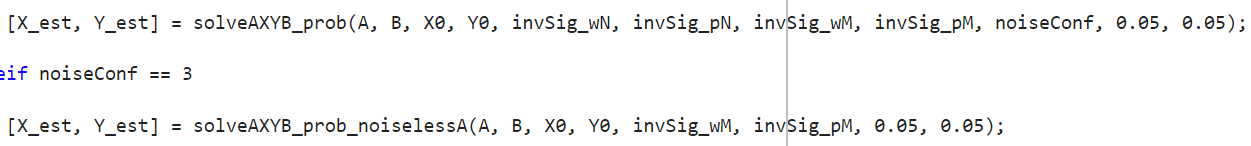
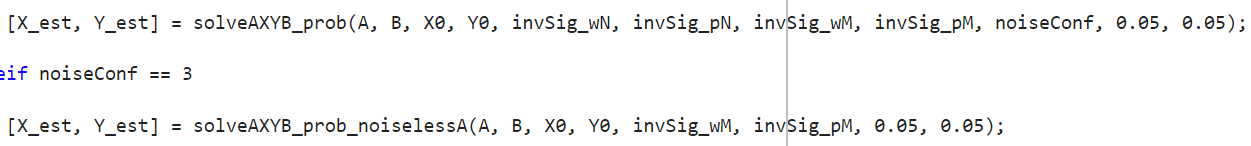
Probabilistic Robot-World and Hand-Eye Calibration AX=YB

This code solves for the calibration problem AX=YB by utilizing system noise properties. We strongly recommend reading our paper if you want to understand the details: <https://ieeexplore.ieee.org/abstract/document/9931998>

1. If you have limited information about system noise properties, please use existing methods:
   1. Use distance minimization methods, or equivalently, loop-closure error minimization methods.
   2. A recommended one is implemented here. Use **solveAXYB\_sgo()**.
   3. You have two measurements. Define A as less noisy one and B as more noisy one with right-translated noise.
      1. This is the hidden underlying assumption of all distance (or error) minimization methods.
      2. Find the details in our paper: <https://ieeexplore.ieee.org/abstract/document/9931998>
      3. See examples below:  
         

If you have good information about your system noise, you can leverage that information to achieve the best calibration result using our code. The following steps explain how to use our code based on **main\_example1.m**.

1. Find out your Noise Configuration:  
   
   1. Be cautious in the locations of noise transformations N and M. Their locations depend on the arrangement of sensors, which determines the reference and target frames.
   2. If one measurement is very accurate (e.g., commercial robot arms), use Noise Configuration 3.
2. Set the noise covariances of A and B.
   1. Each transformation has rotation error and position error, each of which is represented with 3X3 covariance matrix.
   2. Each measurement can be assigned with a different covariance matric.
   3. In the example below, all covariances are set as identity matrices. This assumes equal noise levels over all directions and all measurements.  
      
3. An initial guess is needed. One can be acquired by applying any existing method. For example, **solveAXYB\_sgo()** can be used.
   1. If Noise Configuration is 1 or 2, use the following function:  
      
   2. If Noise Configuration is 3, use the following function:  
      

\* *Note that the covariances are inverted. See* ***main\_example1.m*** *for more details.*