CS 8790: Solution to assignment 3

Chanmann Lim

September 24, 2014

Report:

Accessing the calibration of the IDEK's new 3D sensor requires the studies of the expected value and covariance matrix of the given 100k measurements in comparison to the ground truth and the error covariance matrix given by the engineers designing the hardware.

$$Ground\ truth = [12.9 \quad 130.4 \quad 23.5]$$

$$Error\ covariance\ matrix(R_{True}) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

In this case, the computed expected value of the measurements is $[12.895992 \quad 130.398454 \quad 23.494780]$ which will round up to the ground truth. Thus, the sensor is properly calibrated and unbiased. Next, calculating the covariance matrix of the measurements which is the expected square error matrix

$$R = \frac{1}{n-1} \sum_{i=1}^{n} \begin{bmatrix} \bar{x_i} \\ \bar{y_i} \\ \bar{z_i} \end{bmatrix} \begin{bmatrix} \bar{x_i} & \bar{y_i} & \bar{z_i} \end{bmatrix}$$

$$R = \begin{bmatrix} 0.9974 & 0.9991 & 1.0001 \\ 0.9991 & 2.0037 & 2.0084 \\ 1.0001 & 2.0084 & 3.0114 \end{bmatrix}$$

Finally, the expected covariance matrix must be conservative in other words it must be smaller than or equal to the true covariance matrix given above.

$$\boxed{Eig(R_{True} - R) >= 0}$$

$$Eig(R_{True} - R) = Eig\left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} - \begin{bmatrix} 0.9974 & 0.9991 & 1.0001 \\ 0.9991 & 2.0037 & 2.0084 \\ 1.0001 & 2.0084 & 3.0114 \end{bmatrix} \right)$$

$$Eig(R_{True} - R) = \begin{bmatrix} -0.016873 \\ 0.001233 \\ 0.003055 \end{bmatrix}$$

However, the eigenvalues of the $(R_{True} - R)$ turns out to contain one negative value -0.016873 which indicates that the given covariance matrix is not valid.

Code result:

$$Expected\ value = \begin{bmatrix} 12.895992 & 130.398454 & 23.494780 \end{bmatrix}$$
$$Eigenvalues = \begin{bmatrix} -0.016873 & 0.001233 & 0.003055 \end{bmatrix}$$

Appendix:

assignment_3.m

```
data_size = 100000;

% read data file
fid = fopen('A3-MeasurementData.bin');
measurement_data = fread(fid, [3, data_size], 'float');
fclose(fid);

% calculate covariance matrix
expected_value = mean(measurement_data, 2);
%% use ones() to generate the consistent dimensions for matrix subtraction
measurement_error = measurement_data - expected_value * ones(1, data_size);
%% the sum of many n by 1 matrice M * M' = [M list] * [M list]'
measurement_covariance = measurement_error * measurement_error' / (data_size - 1);

% verify covariance
sensor_covariance = [1 1 1 ; 1 2 2 ; 1 2 3];
eigenvalues = eig(sensor_covariance - measurement_covariance);
fprintf('Expected value = [%f %f %f]\n', expected_value);
fprintf('Eigenvalues = [%f %f %f]\n', eigenvalues);
```