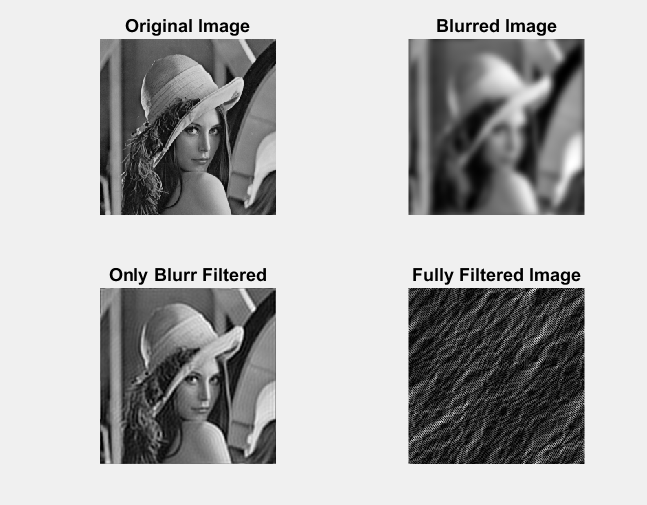
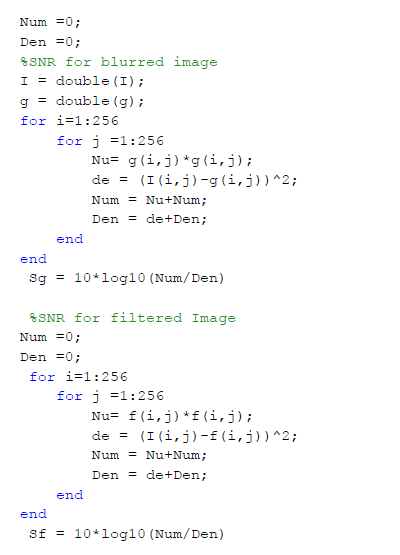
1. **Inverse Filter**

The inverse filter rarely works satisfactorily and should be used with extreme caution. One of the reasons is that the inverse filter frequency response **(G(w))** is the reciprocal of the frequency response of the given system **(H(w))** and ,even in the absence of additive noise, G(w) will not exist if H(w) has any zeros or if the values of H(W) get close to zeros as that triggers the magnitude of G(W) to tend to infinity as 1/~0 -> infinity. Therefore, to remedy this issue, an extra step should be taken to monitor the values of H(w) and set the values of G(w) to zero whenever H(w) falls below a predetermined value. However, if additive noise is present, the inverse filter will have unpredictable and disastrous effects. This is because with noise present, the recovered frequency spectrum of the inverse filter will have an additional term which is the noise spectrum (**N(w)**) divided by the frequency response of the given system **(H(w)).** Obviously, we would need this term as small as possible, however, the noise spectrum is unknown and a random variable that we can’t control. If the noise process has a high-frequency content, it will dominate the true spatial frequency content of the estimate. As a result, the SNR will be small and the noise will dominate the restored image. This is illustrated by figure 1 below. Given the original image, the input image was first blurred through a gaussian white noise (zero mean and 0.2% variance). Then, the blurred image was filtered with an inverse filter, with the precaution discussed above implemented, produced a very good result. The calculated SNR for the blurred image was 0.0591dB, approximately. As it can be predicted, the SNR for the inverse filter image is greatly improved and was calculated to be 20dB. However, when a gaussian process noise is added to the input signal, the inverse filter failed completely as shown below. The recovered image appeared distorted, with SNR of 0.8dB. It can be said the inverse filter performs poorly when an additive noise is present in the system.



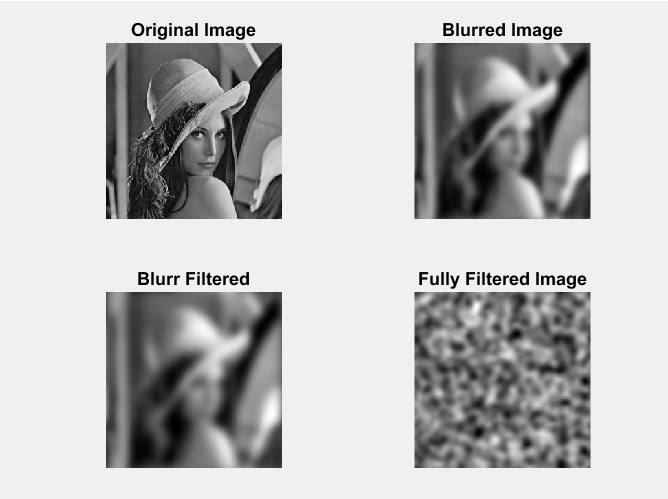
**Figure 1: Inverse Filter**

**SNR Calculation MATLAB**



1. **Least Squares Filter**

The least square filter basically finds an estimate image which minimizes the sum of the squared errors between the actual measured image (blurred image) and the predicted image. In other words, the algorithm attempts to find a matrix **H** which tries to minimize the squared length of the error matrix when convolved with the blurred image. Therefore, if this H matrix is invertible, we are basically getting the inverse filter in effect. However, if it is not invertible, we use the Moore-Penrose inverse to approximate the inverse of H which optimizes the estimation. The advantages of using this filter is that, one, it is easy to conceptualize and, second, it doesn’t make any explicit assumptions about the statistics of the noise or a priori probability of specific solution vectors. However, the least square solution treats everything in the image the same. Meaning, it doesn’t take into account any image properties or features we may wish to preserve in our restored image. Moreover, there is no guarantee that the term (HTH)-1 (in the Moore-Penrose inverse) exists although it can be approximated through singular value decomposition. Therefore, as a result of these factors, the least square filter is expected to perform poorly even when filtering a blurred image. As exhibited in the figure below, the least square filter hasn’t done much as far as deblurring the image, and this is reflected in the SNR value. As mentioned in the ‘Inverse Filter’ section, the SNR for the blurred image is calculated to be 0.0591dB, and the SNR for the least squared filter image is calculated to be -47dB for the blur only image and -131dB for the blurred and noisy image. This is believed to be due to the (HTH)-1 term in the Moore-Penrose inverse. If the inverse of this term doesn’t exist, it has to be approximated as mentioned above, which will lead to an inaccurate estimation of the restored image. In comparison to the inverse filter, the least square filter performed poorly as the SNR value for the inverse filtered image was 20dB. However, it is also important to point out that the least square filter was not able to do anything when process noise is added to the blurred image similar to the inverse filter.



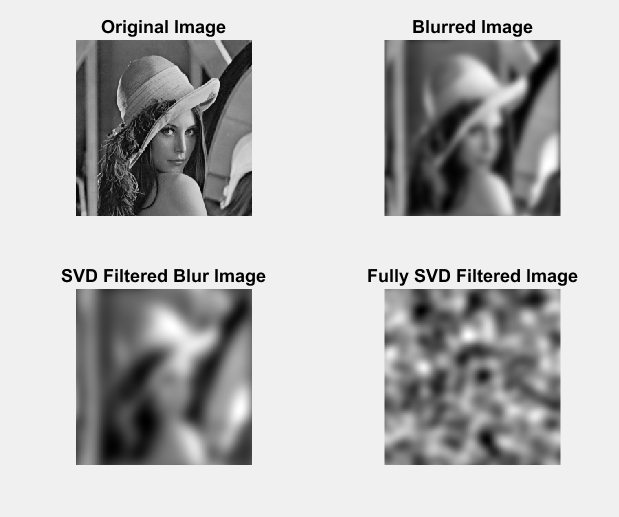
**Figure 2: Least Square Filter**

1. **SVD Pseudo Inverse**

The SVD is the optimal matrix decomposition in a least square sense that it packs the maximum signal energy into as few coefficients as possible. The SVD approach is basically another method of estimating the inverse of the PSF function through spectral decomposition. The blur matrix is decomposed into its left and right eigenvectors along with its diagonal components. The SVD approach is a very simple and can be a very effective algorithm for the right choice of the blur matrix (Block Toeplitz with Toeplitz matrix). However, since we don’t have control of the frequency impulse response (blur matrix) of a system in real life scenarios, a rotationally symmetric gaussian blur matrix is used to perform the analysis here. As it can be observed from the figure below, the SVD output resembles the Least Square filter above. This is due to the S-1 term in the equation below.

[U S V] = SVD (PSF);

Specifically, since the PSF used was not invertible, the diagonal matrix S has zero eigenvalues except its first diagonal element. Due to this its inverse has to be estimated similar to the least square method discussed above; hence the similar outputs. The SNR results are also similar to the Least Square implementation.



**Figure 3: SVD Pseudoinverse Filter**

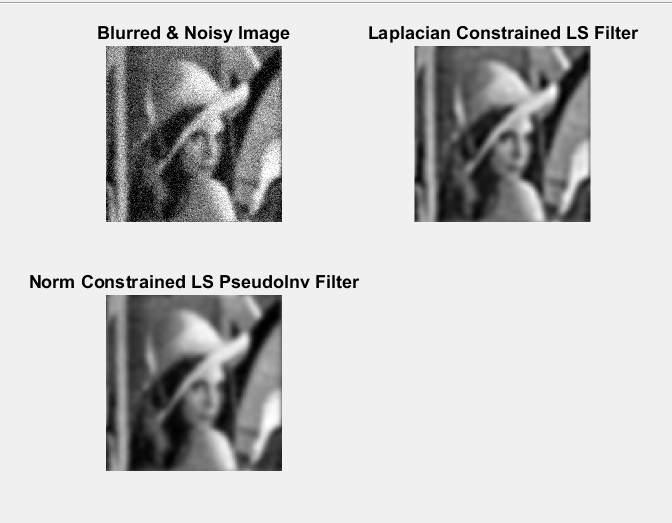
**Part (e) & (f)**

**Norm-Constrained Least-Squares Pseudo-Inverse Filter and Laplacian-Constrained Least-Squares Filter**

In constrained least-squares deconvolution, the goal is to restrict ourselves to solutions which minimize some desired quantity in the restored image, but which impose constraints on the solution space. The specific mathematical approach taken to constrained restoration is to specify a cost function which compromises tow basic parts. The first part consists of some Linear operator acting on the output distribution. The second part consists of one or more Lagrange multiplier terms which specify the constraints. Therefore, the aim is to minimize the size of the linear operator on the output distribution subject to the chosen constraints. In Laplacian constrained LS filter the linear operator chosen is a Laplacian operator, which seeks overall maximum smoothness to the image. As it can be observed from the figure below, the Laplacian Constrained LS Filter managed to filter out the process noise and sort of sharpen the image. In comparison to the LS and inverse filters, the Laplacian filter is far superior in filtering the process noise, but weak in deblurring the image.

In the case of the Norm-Constrained LS pseudo inverse filter, the linear operator to the output distribution is set to zero (through choosing the inverse Lagrange multiplier to be zero), and the filtering is optimized through the norm of the pseudoinverse of the frequency impulse response (PSF). As it can be observed from the figure below, the result of the norm-constrained LS filtering resembles the output from the Laplacian Constrained LS Filter. Both filters performed well in filtering the added noise and were unable to deblur the image effectively.

When assessing the SNRs of these filters, the SNR of the blurry and noisy signal was recorded to be -47dB; however, after passing through the Laplacian and Norm constrained filters, the SNR lowered to -83dB and -137.8dB. This is believed to be due to the smoothing (sharpening) only effect of our approach. When the algorithms smoothed out the additive noise, some image signal is also lost in the process, which we would magnify or increase the ratio of blur : signal in the image. As a result, the SNRs of the output images are lowered consequently, but this is not necessarily considered to be a poor result if we are interested in the smoothing/sharpening an image.



**Figure 3:** Constrained Least-Squares Filter

**Parts (g) & (h)**

**Covariance Constrained Least squares Parametric Wiener Filter and Linear Minimum Mean-Square Error Wiener Filters**

In the general derivation of the Covariance Constrained Wiener filter the input and output images are assumed to be a continuous stochastic field with zero mean. Then, the minimum mean-square error (MMSE) is used as the optimization criterion. In other words, the impulse response of the restoration filter is chosen to minimize the mean square restoration error. Moreover, in LMMSE version of the Wiener filter, the autocorrelation between the noise signal and the image is neglected. Meaning, the noise and image signals are assumed to be independent variables. However, in the Covariance constrained version of the Wiener filter, a frequency dependent estimate of the noise to signal power ratio is used through the autocorrelation matrices of the image and the noise signal. In other words, the noise and image signal are least square optimized through their respective covariances matrices. Basically, the LMMSE Wiener approach assumes the minimum mean-square error is stationary and ignores the constantly changing relationship between the noise and input image. It does so by just utilizing the noise to power ratio as opposed to the covariance matrices of each. Due to this, the covariance constrained Wiener filter performed better as shown in the figure below. Moreover, the SNR values of the blurred noisy image, LMMSE Wiener filtered, and Covariance Constrained Weiner Filter is shown below. As expected, the Covariance Constrained Weiner Filter produced the best SNR value.

Blurred Noisy: **SNR =** 11.4dB

LMMSE Wiener filtered: **SNR =** 12.9dB

Covariance Constrained Weiner Filter: **SNR =** 16.2dB



**Figure 7: Wiener Filter**