Computational MR imaging
Laboratory 5: Image space parallel imaging

Code submission is due by 12:00 before the next Thursday lab section. Please upload your code to StudOn in a described format. Late submissions will not be accepted.

Learning objectives

- Combine multicoil images
- Reconstruct undersampled multicoil data using SENSE algorithm
- Compute g-factor and compare reconstructions to the ground truth

Before the lab: Get familiar with the functions in scipy.linalg **inv** (matrix inverse) and **pinv** (matrix pseudo-inverse), and the numpy operators **.T**(transpose) and @ (matrix multiplication).

1. **Multicoil combination**
    a. Check the data.
        i. kdata: fully-sampled k-space data (256×256×8)
        ii. sens_maps: the coil sensitivity maps (256×256×8)
        iii. noise_maps: the noise-only scan (256×8)
    b. Define coil combine algorithms.
        i. sum-of-squares
            1. Implement sos_comb method.
                a. Use np.sqrt and np.sum functions.
                b. Taks care of the axis to be summed up
        ii. matched-filter (least-squares)
            1. Implement ls_comb method without noise covariance matrix.
            2. Implement get_psi method
            3. Implement apply_spi method
                a. Think about the shapes in the matrix multiplication
                b. Use either @ or np.matmul
            4. Finish implementing ls_comb method for noise covariance matrix.
    c. Combine the multicoil images using sum-of-squares (SoS) and matched-filter with and without the coil noise covariance matrix Ψ.
    d. Plot [Complex-sum, SoS, Matched-filter with pre-whitening, Matched-filter without pre-whitening]. (Hint: to achieve complex-sum, use function "complex_sum()" in utils.py.)
    e. Discuss your results. Comment of the effect of using the noise correlation matrix.

2. **Cartesian SENSE reconstruction and g-factor**
   a. Implement multiple methods for SENSE reconstruction
      i. sense_locs
         1. locs contains $[u_1, u_2, \ldots, u_n]$
         2. Think about what how far each indices are.
         3. Consider periodic cycle of those locs indices.
      ii. sense_aliased_idx
         1. Think about how to bound the index of the aliased image within the phase encoding lines of it.
      iii. sense_sm_pinv
      iv. sense_unwrap
      v. sense_g_coef
         1. Use provided function, calc_g.
      vi. sense_recon
         1. Resemble all of component methods above in a correct order. At the same time, you need to consider the indices of the matrix.
   b. Simulate acceleration factors.
      - R=[2,3,4] along the phase-encoding direction.
      - Assume that the phase encoding direction is the row dimension (anterior-posterior).
      i. Reconstruct each undersampled data set using your SENSE implementation and compute the average g-factor.
      ii. Plot and compare PSNR and SSIM of accelerated image to that of non-accelerated image. (Use the matched filter combination as the ground truth.)
      iii. Plot the reconstructed image, reconstruction error to the matched-filter combination and g-factor map for each R.
         1. When taking the difference between two images, do not forget to normalize images. (see utils.normalization())