



Homework assignment 7: MIMO

Due date: 2017-12-06

1. Implement a simple MIMO system

Implement a MIMO simulation. Assume that the channel matrix is known to both transmitter and receiver. Use approaches similar to previous simulations: vary the signal-to-noise ratio and plot symbol error rates. You could also introduce a channel estimation error by adding some randomization to the channel matrix, using a different matrix for actual transmission and for pre- and postprocessing.

Make sure to run enough iterations! For each SNR value, iterate over bits and iterate over channel realizations. A channel matrix would comprise random values for each channel, e.g., from a Rayleigh fading channel. Noise is added for each receive antenna separately.

Look at the symmetric case of $n_t = n_r$. Send, in each timeslot, different symbols over each of the n_t antennas.

For an additional insight, plot the resulting symbol error rate against observed condition numbers. What do you observe?

Hints:

- In case you use Numpy, look at the function `numpy.linalg.svd` for the singular value decomposition. It does not exactly return what we want; the second return value is not a matrix, but that is easy to fix.
- A simple way to get a Rayleigh fading channel coefficient is to multiply an exponential random variate with a uniform random variate (between 0 and 2π) that is multiplied by i and used as an argument to the exponential function.

Do that for n_t by n_r matrices, and you have your channel matrix with a single line of code in numpy (the `numpy.random` functions can produce matrices, which you can feed into the exponential function and multiply, elementwise)

- Functions to generate data, modulate and demodulate data to symbols have been used in previous code examples. You can reuse them. (Use, e.g., a QPSK modulation)
- The rest of the code is just plugging these functions together. The main creative part is to understand the man page of the svd function, really.