E7250 Laboratory exercise 5::

Linear time variant channels

Contents

Measurements of linear time variant channels

1.1 2-tone measurement of spaced-time spaced-frequency correlation

In this measurement we determine a form of the *spaced-time spaced-frequency correlation function* $\phi_H(\Delta f, \Delta t)$. In a fading radio channel the signal power fluctuates over time. Moreover, in a multi-tap channel, where signal arrives at the receiver as multiple copies separated in time, the fluctuations are frequency dependent such that different frequencies experience fading at different times. Related concept is the coherence bandwidth B_m which determines the bandwidth over which the signal remains coherent. The 2-tone measurement in this exercise is relatively simple to implement and the results of such measurement are quite easy to understand.

In a 2-tone measurement two sinusoidal waves with a frequency difference Δf are transmitted over a radio channel. At the receiver the power of each wave is sampled periodically and the data is collected. What we have then is a large number of observations of two random variables and the final task is to compute the correlation between the variables.

1.1.1 Equipment

This measurement requires the following equipment:

- Rohde & Schwarz SMBV100A vector signal generator
- Tektronix RSA6114A spectrum analyzer
- National Instruments USB GPIB adapter
- Spirent Vertex channel emulator
- Computer with software for remote controlling the spectrum analyzer

- Coaxial cables, adapters, attenuator

1.1.2 Measurement

The measurement setup is depicted in figure 1.1.

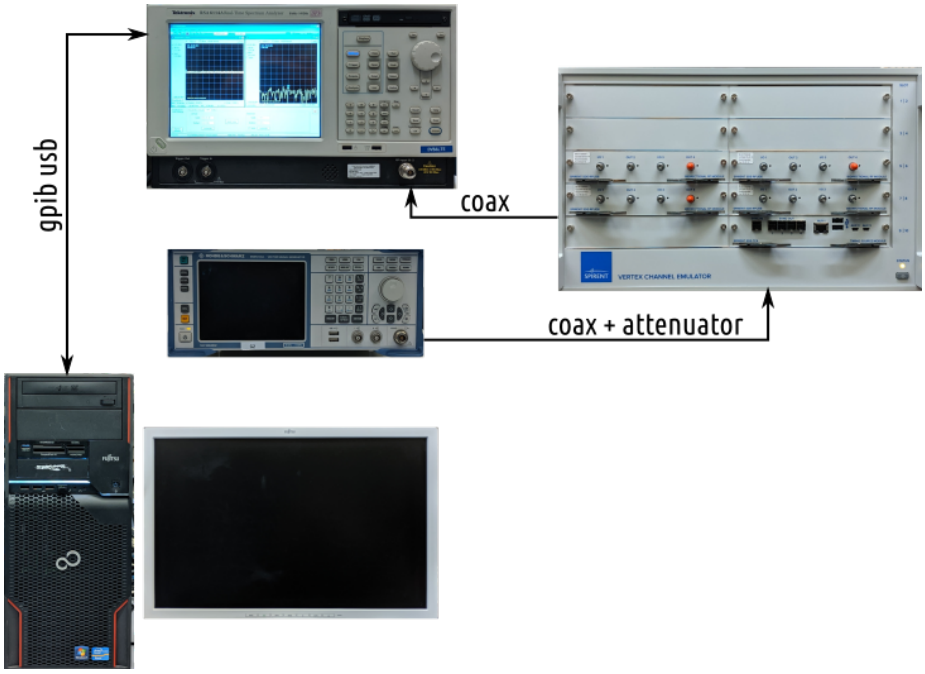

Figure 1.1: Measurement setup for 2-tone measurement. GPIB adapter plugs into the back of spectrum analyzer. A 30 dB attenuator is used between generator and channel emulator to prevent accidental damage from using too much power.

At first, power on all equipment and connect just the spectrum analyzer to the PC using the GPIB USB adapter. The spectrum analyzer settings will be controlled by a measurement script on the PC, so no need to set anything manually. The signal generator supplies the 2-tone measurement signal and needs to be configured manually as follows:

- Preset the instrument
- Baseband: config... → ARB... → Multi Carrier...
- Number of Carriers: 2
- Carrier Spacing: 200 KHz

- Carrier Table... → Set state to On for both carriers in table then close window
- Create and Load then close window
- In ARB window, switch State to On then close window
- Level -20 dBm, Frequency 2.45 GHz

Move on the channel emulator where a basic 2x2 MIMO configuration is opened by default. Only one path is needed, A1 → B1, and from the user interface you should find out which ports are in use for input and output for that path. Now connect a 30 dB fixed attenuator to the input port and connect the generator to the attenuator using a coaxial cable and suitable adapter. Connect output port of channel emulator to RF Input of spectrum analyzer.

Setting up the channel emulator:

- Expected input power: -50 dBm
- Output power: -20 dBm
- Frequency: 2.45 GHz
- Press Play button

The emulator is now running with the default channel model which is just a passthrough. Move on to the computer, open terminal and navigate to the *channel_lab* directory. Simultaneously open the file *2tone.py* in any text editor. Explore the measurement script and edit some variables to correspond with your measurement. For now set number of samples in variable N to 100. Save the file. Run script in terminal:

```
sudo python3 2tone.py
```

Observe the spectrum analyzer and verify that you can see the two lines in the spectrum with markers on them. This was a test run only to check that the setup is working correctly.

Now edit the channel model by clicking the Default channel model and click Edit. Make a 2 tap channel with equal power in each tap and fading type Rayleigh. Second tap is delayed by 1 μs. Set Velocity to 5 km/h.

Again observe the spectrum analyzer. Now the fading of the signal should be visible.

Use the measurement script to record 500 observations of the fading channel for five different frequency separations between 100 KHz and 1 MHz. Remember to set the carrier spacing in signal generator in each case.

1.1.3 Report tasks

- Compute the correlation between signal amplitude values for each frequency separation. You can use Pearson's correlation coefficient, available as standard function in Matlab/Octave.
- Plot the measured correlation.
- What is the approximate coherence bandwidth of the fading signal, i.e. where correlation is above 0.5.

1.2 Measurement of LCR and AFD

In this measurement we investigate the level crossing rate (LCR) and the average fade duration (AFD) in a Rayleigh fading mobile channel. LCR and AFD measure how frequently a certain threshold level is crossed and average duration of a fade below certain level, respectively. Related concept is the coherence time T_d of the channel.

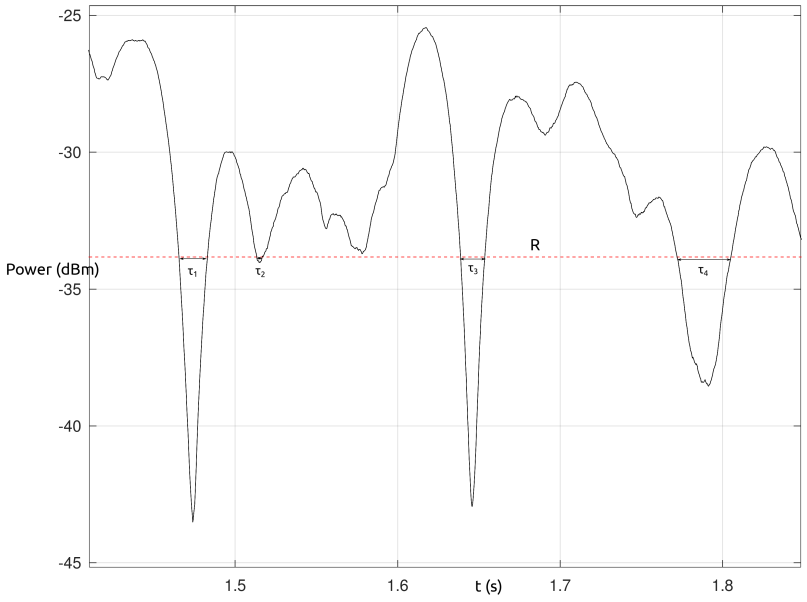

Figure 1.2: LCR is the average number of positive-going crossings per second, in other words the average number of fades per second. AFD is the average of τ_1, τ_2 ... τ_n.

Figure 1.2 illustrates the concept. The threshold level R is arbitrary. It could be set to, for example, 5 or 10 dB lower than the average signal power. The knowledge of LCR and AFD can be used in designing wireless communication systems to make informed choices in modulation type, automatic gain control circuits, frame lengths, coding etc.

1.2.1 Equipment

This measurement requires the following equipment:

- Rohde & Schwarz SMBV100A vector signal generator
- Tektronix RSA6114A spectrum analyzer
- Spirent Vertex channel emulator
- Coaxial cables, adapters

1.2.2 Measurement

The measurement setup is depicted in figure 1.1 with the exception of not needing the PC with GPIB remote control interface.

Preset both the generator and spectrum analyzer to return to default settings. For signal generator use the following settings:

- Level -20 dBm, Frequency 2.45 GHz

In spectrum analyzer, use the same center frequency and a span (measurement bandwidth) of 5 MHz. Then add a new measurement window from Setup $\rightarrow$ Displays: Amplitude vs Time. This display gives a view to power of the received signal over time which is ideal for collecting the data for determining the LCR and AFD statistics. By default it operates over a relatively short time while in this measurement the interest is in collecting fading statistics over a much longer period of time. Therefore we need to adjust the sampling and analysis parameters of the instrument.

- Open Setup $\rightarrow$ Acquire and change the *Adjust* box from Auto to Acq. BW, Acq length.
- Change acquisition length to 3 seconds.
- Then open Setup $\rightarrow$ Analysis and change Analysis length to 3 seconds, Analysis offset to 0 s and Time Zero Reference to Acquisition Start.
- Adjust the Scale and Position in the Amplitude vs Time window to 3.000 s and 0.000 s.

- Finally open Setup → Settings and change in Prefs tab Max trace points to 1K.

The display updates slowly, but we are only interested in one 3 s observation for each measurement. Press the *Stop* button in upper right corner in spectrum analyzer to disable continuous measurement.

Narrowband signal, 1-tap channel

As narrowband signal we use the unmodulated carrier from signal generator, turn on the RF output. In channel emulator edit the channel model to be 1-tap Rayleigh fading with 5 km/h velocity. Initiate one measurement in spectrum analyzer by choosing *Single* under the *Run* button in upper right corner.

After a while the measurement will be ready and you can explore the Amplitude vs Time display. You can zoom in and pan the measurement on the display, it should look similar to figure 1.2.

Save the data by keeping Amplitude vs Time window active and choose Save as... from File menu. Save as type: Results export (CSV).

Wideband signal, 1-tap channel

To generate a wideband signal the carrier is modulated by a suitable baseband signal. Use the Baseband block in signal generator to define a signal with a rate of 4 megasymbols per second with, for example, QAM16 modulation. Enable the baseband block.

Initiate measurement and verify from the spectrum display that the signal has wide bandwidth. Save the trace from Amplitude vs Time display.

Narrowband signal, 2-tap channel

Disable the baseband block in signal generator. Edit channel model to be 2-tap channel with Rayleigh fading taps, 5 km/h velocity. Second tap is delayed by 1 μs.

Initiate measurement and save the trace from Amplitude vs Time display.

Wideband signal, 2-tap channel

Enable the baseband block again, initiate measurement and save the trace from Amplitude vs Time display.

1.2.3 Report tasks

The saved files contain power in dBm with some header information. File can be imported to Matlab/Octave using *importdata* function:

```
a = importdata('filename.csv', ',', 63)
```

The data can then be found in *data* field of the returned struct *a*, accessible using the dot notation:

```
a.data
```

The time axis can be figured out by considering the number of data points and start and stop time, information readable in the header lines of each file.

- Plot the power of the signal over time. What is the dynamic range? What is the average power?
- Determine the level crossing rate and average fade duration using $R = P_{avg} - 5dB$.
- Can you explain differences in LCR and AFD between 2-tap and 1-tap channels and narrowband and wideband signals? Hint: 2-tap channel is same as in 2-tone measurement with same frequency correlation properties.

To determine the LCR and AFD the following Matlab/Octave functions can be useful: `sign`, `find`

1.3 Measurement and analysis of LTV system functions

In this exercise we measure the time-variant impulse response of mobile radio channel and compute other LTV system functions from it. Impulse response is measured using the matched filter or correlation method. A complex signal $s(t)$

is sent to the channel with complex impulse response $h(t)$. The received signal is matched filtered. The output of the matched filter is

$$
\begin{aligned}
s_{out}(t) &= s(t) \otimes h(t) \otimes \overline{s(-t)} \\
&= \left(\int_{-\infty}^{\infty} s(\tau) s(\tau - t) d\tau \right) \otimes h(t) \\
&= R_{ss}(t) \otimes h(t)
\end{aligned}
$$

The output is a convolution of the autocorrelation function of $s(t)$ and the impulse response $h(t)$. Ideal measurement signal has $R_{ss} = \delta(t)$ since then $\delta(t) \otimes h(t) = h(t)$. One good choice is a Zadoff-Chu sequence which is a Constant Amplitude Zero AutoCorrelation (CAZAC) waveform. The periodic (i.e. circular) autocorrelation of such sequence is a $\delta(t)$ function. The measurement sequence needs to be long enough to support the absolute delay spread of the channel.

1.3.1 Equipment

This measurement requires the following equipment:

- Spirent Vertex channel emulator
- Computer
- Two USRP N200 software radios
- CDA-2990 8 Channel Clock Distribution Accessory (a.k.a. Octoclock)
- Coaxial cables, adapters

1.3.2 Measurement

Diagram of the measurement setup is in figure 1.3. A clock distribution device provides 10 MHz frequency reference and a pulse per second (PPS) time reference to the software defined radio (SDR) devices. One SDR works as the transmitter and the other as receiver.

Make connections according to the diagram. Like in previous measurements, use a 30 dB attenuator at the input of channel emulator to protect the input from accidental overloading. The SDR with IP address 192.168.20.2 is the transmitter and transmits from port RF1. The receiver has IP address 192.168.30.2 receiving from port RF2.

In channel emulator use the following settings

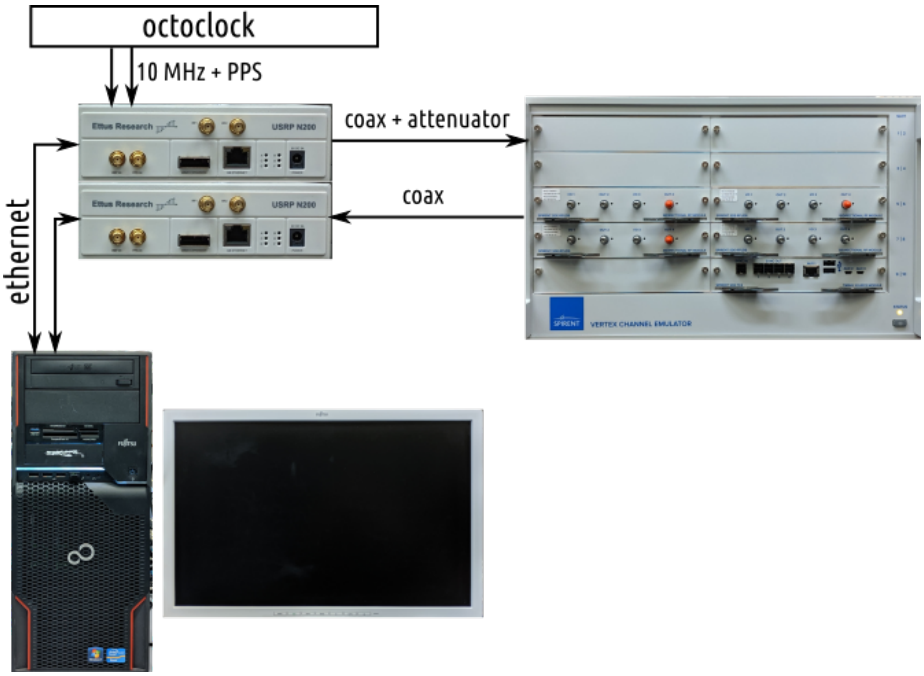

Figure 1.3: Measurement setup for impulse response measurement.

- Expected input power: -30 dBm
- Output power: -30 dBm
- Frequency: 2.45 GHz

In PC the measurement script *channel_sounding.py* is found in directory *channel_lab*. Open script in editor and check that the measurement frequency *tx_freq* is set to 2450e6, sampling rate *tx_rate* is set to 10e6 and duration of measurement *T_rx* is set to 2.5. Variable *channel_sampling_rate* controls how often the sounding signal is transmitted. With a rate of 200 Hz we get 500 samplings of the impulse response over 2.5 seconds. In general the channel sampling rate should be significantly higher than the rate of change of the channel.

The channel emulator includes a collection of different channels, usually representing typical radio channels in different environments. To choose from the collection, click the current channel model name (probably Default or Unsaved profile), scroll down the list and select More... You can browse these to get an idea what kind of channels are used for testing wireless communication systems. For each channel you can see the power delay profile (PDP) plotted.

For this exercise choose first under GSM the channel TU Low 2 (6 Paths, 3

km/h). Take a screenshot or picture of the power delay profile. Press Play in channel emulator to activate the emulation. Choose a different filename in the measurement script for each measurement and save changes to the script. Run the script with

```
sudo python3 -i channel_sounding.py
```

After the measurement the python interpreter remains in interactive mode. You can display some figures by writing

```
plt.show()
```

The plot shows a short segment of the received signal correlated with the measurement sequence. Zoom in to one sample of the impulse response. If everything is right you should see roughly the same six channel taps as in the channel model. Note that each sampling of the impulse response seems to show three copies of the impulse response. That is because the waveform sent to the channel was actually the original sequence in three copies. The reasoning is that first and last copies help to take advantage of the *periodic* autocorrelation property of the measurement sequence, effective for the middle sequence. Therefore only the middle part is the valid measurement of impulse response.

The saved file contains the received samples and the measurement sequence (one sequence, not three).

Now select another channel model, LTE EPA5 and take a screenshot or picture of its power delay profile. Perform the measurement and observe the impulse response again. How does the impulse response look compared to the power delay profile visible in channel emulator? Can you explain why it looks different?

Report tasks

Time-variant impulse response

- Compute the matched filtered output. It can be computed using either `conv` or `xcorr` functions. Variable 'seq' is the original measurement sequence, for `conv` you need time reversed and conjugated version. You should get the same data that was plotted during the measurement. Verify by plotting small part of the output, convert to dB for plotting.
- Look where the valid middle part of the first channel response is located. See figure 2.1 for example. You could choose for example +/- 50 samples around the first channel tap, or even smaller window if the channel response fits inside.
- Construct an array of the impulse responses, in other words, the time-variant impulse response $h(\tau, t)$. First column of array is the first impulse response, second column is the second impulse response and so on. You know the locations of the impulse responses in the filtered/correlated output by considering the location of the first impulse response and knowing the repetition rate of 200 Hz, sampling rate of 10 MHz and there is 2.4 seconds of samples. There can be some clever indexing trick to achieve it in one line of code and if not, at least a for-loop should do. As result you should have 100x480 matrix, if the length of impulse response is 100 samples.
- Plot the absolute value of the impulse response for first 50 impulse responses. Example is given in figure 2.2.

PDP, delay spreads

- Compute and plot power delay profile (PDP), i.e. average power of each tap of the time-variant impulse response, for each measured channel. Normalize so that strongest tap has 0 dB.

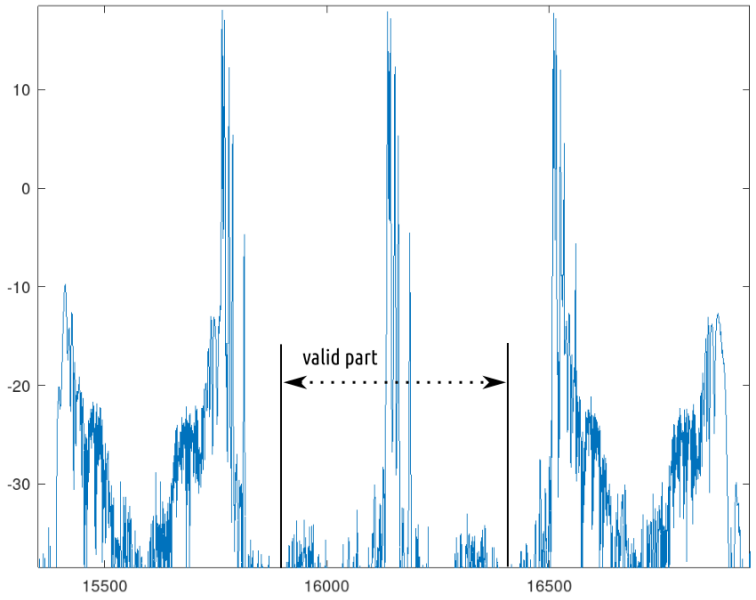

Figure 2.1: Valid part of impulse response is in the middle of each matched-filtered measurement burst.

- Compare the PDPs to the PDPs displayed in channel emulator. What differences do you find?
- From PDP, calculate absolute delay spread and rms delay spread. Give answers in microseconds.

Time-variant channel response, delay-Doppler spread function

- Construct the delay-Doppler spread function $h(\tau, \phi)$ from the time-variant channel response.
- Plot in one figure the Doppler spectrum of each channel tap (i.e. at the delays τ_k where the peaks in PDP are). How much is the Doppler spread?
- Use the Doppler spread to estimate mobile velocity when carrier frequency is 2.45 GHz.

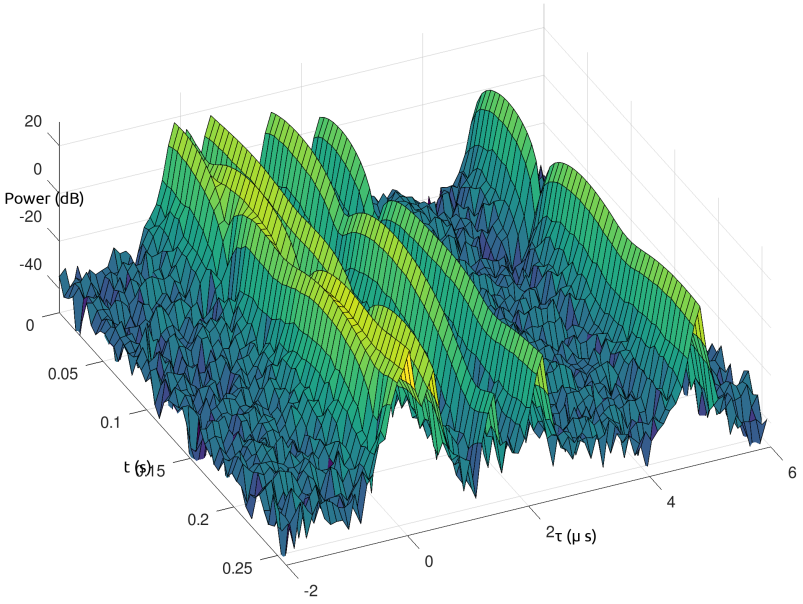

Figure 2.2: 50 snapshots of time-variant impulse response. Plotted using `surf` function.

Estimated statistical LTV system functions

- Construct the time-variant channel transfer function $H(f,t) = F_\tau\{h(\tau,t)\}$ using `fft` across the τ dimension.
- Estimate and plot the frequency correlation function $\phi_H(\Delta f, 0)$. How to estimate? Use `xcov` or `xcorr` to take autocovariance/autocorrelation along f axis of $H(f,t)$ and average them along t axis. What is the coherence bandwidth?
- Estimate and plot the time correlation function $\phi_H(0, \Delta t)$. What is the coherence time?
- What is relation between coherence bandwidth and delay spreads?
- What is relation between coherence time and Doppler spread?

Matlab/Octave tips

The frequency axis in output of `fft` starts goes from 0 Hz to $F_s/2$ and then from $-F_s/2$ to near 0 Hz again. Use `fftshift` function to shift the frequency axis so that it goes from $-F_s/2$ to $F_s/2$ with 0 Hz in the middle. It will be easier to view figures of complex spectra that way.

Use scaling option `'coeff'` in `xcov` or `xcorr` so that the result with zero shift is normalized to 1.