Interference statistics in wireless systems

15 March 2004- v.2.0

1.0 Introduction

Computer based simulations is one of the main tools in the research and development of mobile radio communication networks. This laboratory exercise will introduce a radio network simulation tool designed for capacity and quality evaluations of different radio network solutions.

The focus is on the interaction between all the communication links (base station - mobile station pairs) in the system. It is assumed that the performance of the individual links already has been investigated by link level simulations.). To study the interaction of the communication links the propagation between all mobile stations and all base stations need to be modelled.

We are also interested in the dynamic behaviour of the system, which requires user data traffic models as well as models for the physical movement of mobile stations connected to the system.

The purpose of the laboratory exercise is to get some hands on experience of making radio network simulations with RUNE. It also gives the opportunity to get aquainted with the basic models and performance measures commonly used.

Rune consists of a common part (runegain) that essentially models user behaviour and propagation. There is one part for TDMA (runetdma) and CDMA (runecdma). In this first encounter with rune we focus on the

The focus will be on simulation methodology rather than network design and therefore only basic radio network algorithms will be used.

1.1 Some advice

First of all RUNE is a toolbox and most of the files is just an example of how things can be implemented thus you are strongly encouraged to make changes whenever the code does not do what you want to do.

Do not change the original code. Make your own directory where you can make a copy of the file you want to change and make changes there. Most of the time it is good to give your changed file a new name to avoid confusion.

However there are cases when you want to create a file with the same name. If you want to change a file deep down in the simulator and do not want to write new calls in the code the whole way up you can overload the function. You make a new file with the same name as you want to change. Then makes sure your file is called the same way as the original.

The files in you current directory are called before any other file. The order in which files are called depends on the path, which you can see by using the ‘path’ command.

There is help in MATLAB. Type help <command> at the prompt. Usually the help is quite helpful. This is also a good way to learn about the RUNE functions.

2.0 Expected results

The results from the lab should be presented in the following way:

- Oral presentation/discussion of the results. The presentation should be 15 minutes. English or Swedish can be used.
- Written report in English.

The report should as a minimum requirement answer the questions and show the graphs required in the “result” section of each exercise. Some hints for writing the report:

- Figures should be used to support statements made in the report. Too many figures tend to confuse rather than clarify.
- If MATLAB code is written during the lab it should not be included in the report.

3.0 Laboratory instructions

3.1 Getting started

If RUNE already is installed you just have to add the paths by calling the script `setp` in the following way:

```
>> run ‘C:\Program-Files\Matlab\toolbox\rune\setp’
```

If you are performing the labs on your own computer you have to install RUNE first.

1. Download the zip-file from the course homepage.
2. Unzip the files to the directory <MATLAB>\toolbox\
3. Start MATLAB and run `setp` as described above.

3.2 Your first simulation

This task should give you a first insight in how RUNE works and which parameters there that can be changed.

RUNE Parameters

Use the default for the other parameters.

- Reuse factor: 3/9 (controlled by km, lm and sps)
- 4 channels/cell
- 50% load, i.e. 2 users per cell on average
- alpha=3.5
- sigma=8
- cell radius: 1000m
- Transmitter power 1 W
- Simulate 9 clusters (controlled by kn, ln)

Execution

Generate the parameter set by using the function `setpart`.

Modify the parameters and use `runeft` to run the simulation.

Run one snapshot and plot the cells and the mobile users, good functions to use are `plothex` and `ploth`.

Create a CDF plot of the C/I ratio in the uplink, e.g. use the function `cdfplotlow`.

Results

Plot of the snapshot and C/I.

3.3 Required number of co-channel interferers

This task should give you insight into how many clusters and cells that are required in order to give reasonable "correct" simulation results.

It should also give you a feeling of what the limitations of the computer used for the simulation is.

Finally it should make you acquainted with interference distributions.

Preparatory Task

Try to predict if the reuse factor will influence the number of clusters required to find "correct" values. Why do you think that?

Will directional antennas influence the required number of clusters? How?

RUNE Parameters

- Reuse factors: 3, 9 and 3/9
- 4 channels/cell
- 50% load, i.e. 2 users per cell on average
- alpha=3.5
- sigma=0
- raa=0
- cell radius: 400m
- Transmitter power 1 W
- Noise floor -128 dBm (to get a interference limited system)

Execution

Run RUNE snapshots for different reuse factors and for different number of cochannel interferers, i.e. number of clusters.

For each simulation determine the median C/I for the uplink in dB.

Determine how many the co-channel interferers that are required to get the median C/I within 1dB of the asymptotic median C/I. I.e. when will simulating more clusters not change the median C/I more than 1 dB.

Analytically calculating the median C/I is difficult. Thus you will have to use larger and larger systems until the value stabilises.

Results

Describe how many co-channel interferers that are needed.

3.4 *Shadow fading*

This task should give you a feeling for the effects of log-normal shadow fading.

Preparatory Task

What will the distribution of the carrier be in the downlink without any shadowfading? What happens to the distribution if we apply shadow fading?

RUNE parameters

- Reuse factor: 7
- 15 channels/cell
- 60% load, i.e. 9 users per cell on average
- alpha=3.5
- sigma=0, 4 and 8
- raa=0
- Cell radius: 400 m
- Noise floor -128 dBm
- Transmitter power 1 W
- No fast fading, i.e. set par.usefastf to 0

Execution

Study the pdf for C, I and C/I for different shadow fading amplitudes in the downlink. Measure in dBm or dB.

Note that here we want to se the probability density function. There are no function that plots the pdf, but an approximation can be obtained by using the function `hist`.

Results

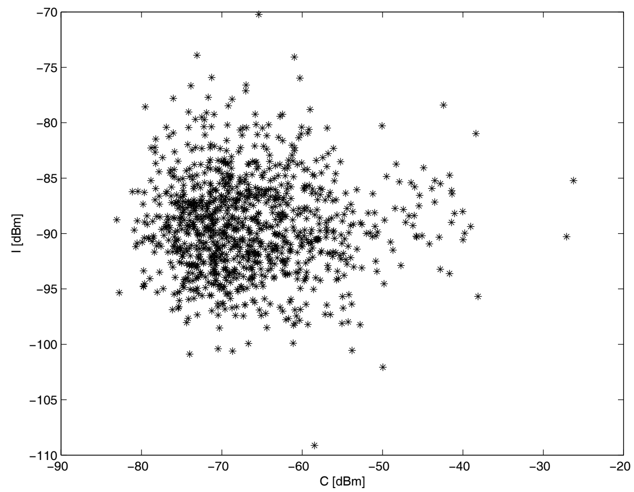
What effect does the shadow fading have on the distributions of C, I and C/I?

Is this result consistent with the theory of stochastic variables log-normal distributed variables?

3.5 *Downlink correlation*

This task should give you a feeling for downlink correlation and its effects.

This task should also familiarise you with C vs. I scatter plots. An example of a scatter plot can be found below:



This type of plots can be created using `plot(x,y,'*')` in Matlab. Note that your plots will probably not look like the one above.

RUNE parameters

- Reuse factor: 7
- 15 channels/cell
- 60% load, i.e. 9 users per cell on average
- alpha=3,5
- sigma=8
- raa= 0, 0.5, 1
- Cell radius: 400 m
- Noise floor -128 dBm (to get an interference limited system)
- Transmitter power 1 W
- No fast fading

Execution

Plot C vs. I as a scatterplot for different downlink correlation parameters. Make the plot for both uplink and downlink. Use dB scale.

Results

Explain the results.

3.6 *Wraparound*

This task illustrates how the wraparound of RUNE works.

RUNE Parameters

Use the default for the other parameters.

- Reuse factor: 3/9 (controlled by km, lm and sps)
- 4 channels/cell
- 25% load, i.e. 1 users per cell on average
- alpha=3.5
- sigma=8
- cell radius: 1000m
- Transmitter power 1 W
- Simulate 9 clusters (controlled by kn, ln)

Execution

Make a simulation of the system. Plot the mobiles and the system in the same way as in task 3.2.

Then draw a line from the mobile to the base which it is connected to. One way to do that is like this:

```
hold on;
xym=clean(res(1,1).xym);
b=clean(res(1,1).b);
for j=1:length(xym)
  s=xym(j);
  e=sys.xyb(b(j));
  plot([real(s) real(e)],[imag(s) imag(e)],'k-');
end
```

Results

Create a plot of which mobiles that are connected where. Explain how the wraparound works.

3.7 *Reuse effects*

This task should give you a feeling for the trade-off between reuse and capacity.

RUNE parameters

- Reuse 1/3, 3/9 and 4/12
- 15 channels/cell
- alpha=3.5
- sigma=0
- raa=0.5
- cell radius: 400m
- Turn off fast fading

Execution

Derive the 10% C/I level in the uplink as a function of traffic load for different reuse patterns. Deriving analytically is extremely difficult. Thus you have to run a number of simulations and plot the results.

Compare the results for 13% load for 1/3, 50% load for 3/9 and 67% load for 4/12 (same amount of mobile stations per frequency).

Results

Explain the results.

3.8 Data quality

This task illustrates the randomness of the data obtained in this type of simulations.

RUNE parameters

- Reuse 1/3
- 4 channels/cell
- Simulate 4 and 25 clusters.
- 50% load, i.e. 2 users per cell on average
- Cell radius 1000 m
- alpha=4
- sigma=8
- Transmit power 1 W

Execution

Run 10 snapshots for each number of clusters and determine the median C/I level for the downlink in each snapshot.

Calculate the mean and the variance of the obtained data.

Results

Explain the results.

Suggest different ways of reducing the variance of the data obtained. What are the benefits and drawbacks?