

DOCUMENTATION

For

BEAMFORMER TEST TOOL SUITE

Contents

| | |
|--|----|
| Installation and Execution..... | 4 |
| Noise to Signal C+..... | 4 |
| Delay Generator C+..... | 4 |
| Fractal Delay Application Python..... | 4 |
| Noise Generator Python | 5 |
| Subfile Generator C+..... | 5 |
| Functionality | 6 |
| Noise to Signal C+..... | 6 |
| float* readFile(char* filename, int sample_size, int channels) | 6 |
| Delay Generator C+..... | 6 |
| void checkFlags(struct arguments arguments)..... | 6 |
| coord** parseCoordFile(char* filename) | 6 |
| float* calcDelay(coord** coordinates, double elevation, double azimuth)..... | 7 |
| char* generateOutputFilename() | 7 |
| bool generateRandomDelays(char* outputFilename) | 7 |
| bool generateModelledDelays(char* outputFilename, char* coordinateFile, double elevation, double azimuth) | 8 |
| Fractal Delay Application | 8 |
| resample(signal, sample_size, original_sample_size)..... | 8 |
| generate_gauss(sample_size, magnitude=127) | 9 |
| generate_sine(frequency, baseline, sample_size, amplitude=127, phase=None) | 9 |
| read_delay_file(filename="delays.csv") | 10 |
| delay_signal(signal, delay) | 10 |
| apply_delay(signal, delay)..... | 10 |
| splice_signal(signal, step)..... | 11 |
| reset_output_file(filename)..... | 11 |
| write_signal(signal, filename)..... | 11 |
| write_original_signals(signal, sampled_signal, filename) | 12 |

| | |
|-------------------|----|
| Contribution..... | 12 |
|-------------------|----|

Installation and Execution

Noise to Signal C+

1. open path `.../FrontEnd/src/executables/apply_noise_to_signal/` in terminal
2. run `"make"`
4. run `"./delay_generator"`

Type of files

`main.c` – main file to execute the functions

`noise_to_signal.c` – all signal functions are stored here to be executed by `main.c`

`noise_to_signal.h` – header file

`makefile` - build file

`readme` - guide to run

Delay Generator C+

1. open path `.../FrontEnd/src/executables/delay_generator/` in terminal
2. make sure having `delays.csv` in the same directory
3. run `"make"`
4. run `"./delay_generator"`

Type of files

`main.c` – main file to execute the functions

`delay_generator.c` – all signal functions are stored here to be executed by `main.c`

`delay_generator.h` – header file

`delays.csv` – csv file that store all the delays value

`makefile` - build file

`readme` - guide to run

Fractal Delay Application Python

1. open path `.../FrontEnd/src/executables/fractal_delay_application/` in terminal

2. make sure having delays.csv in the same directory
3. run "python main.py" to generate noise.csv file

Type of files

main.py – main file to execute the functions (optimized version)

utils.py – main function (old version)

delays.csv – csv file that store all the delays value

Noise Generator Python

1. open path .../FrontEnd/noise_generator/ in terminal
2. run "python main.py" to generate noise.csv file

Type of files

main.py - main file to execute

Subfile Generator C+

1. open path .../FrontEnd/subfile_generator/ in terminal
2. make sure having configfile.csv, delays.csv, signals.csv, noise.csv in the same directory
3. run "make"
4. run "./subfile_generator"

Type of files

frannor.c - library that give normally distributed gaussian random float

frannor.h - define fields and functions

subfile_generator.c - functions to load files and combine to final .subfile

makefile - build file

readme - guide to run

delays.csv - contain delays values

noise.csv - contain noise values

configfile.csv - contain settings values

signal.csv - contain signal values

Functionality

Noise to Signal C+

```
float* readFile(char* filename, int sample_size, int channels)
```

Description

Read the file for the delay value

Parameters

char – char file name

sample_size – integer sample_size

channels – integer number of channels

What the function Returns

Return nothing

Delay Generator C+

```
void checkFlags(struct arguments arguments)
```

Description

Check the default parameters value, if there is no error, proceed to execute

Parameters

Arguments – Struct arguments

What the function Returns

Return nothing

```
coord** parseCoordFile(char* filename)
```

Description

Get the coordinate for the longitude and latitude value from the file

Parameters

filename – char file name

What the function Returns

Return the coordinate value from the file

float* calcDelay(coord** coordinates, double elevation, double azimuth)

Description

Parameters

What the function Returns

char* generateOutputFilename()

Description

Generate the timestamp of the output filename

Parameters

Nothing

What the function Returns

Return the file

bool generateRandomDelays(char* outputFilename)

Description

Generate the random delay and then write it to a file

Parameters

outputFilename – char file name

What the function Returns

Return true if the file has been successfully written, else false

bool generateModelledDelays(char* outputFilename, char* coordinateFile, double elevation, double azimuth)

Description

Generate the delay model from the coordinate

Parameters

outputFilename – char file name

coordinateFile – char coordinates value

elevation – double value of the elevation

azimuth – double value of the azimuth

What the function Returns

Return true if the delays have been calculated, else false

Fractal Delay Application

resample(signal, sample_size, original_sample_size)

Description

Resample the signal value

Parameters

signal – the signal value

sample_size – the sample size value

original_sample_size – the original sample size value

What the function Returns

Return the resampled signal values


```
generate_gauss(sample_size, magnitude=127)
```

Description

Generate the gaussian white noise signal

Parameters

sample_size – the sample size of the signal

magnitude – the noise magnitude

What the function Returns

Return the signal wave

```
generate_impulse(duration, baseline, sample_size, amplitude=127)
```

Description

Generate the impulse signal and then plot the graph out

Parameters

duration – the duration of the impulse wave

baseline – the baseline of the impulse wave

sample_size – the signal sample size

amplitude – the amplification of the impulse wave

What the function Returns

Return nothing

```
generate_sine(frequency, baseline, sample_size, amplitude=127, phase=None)
```

Description

Generate the sinusoidal wave and plot the graph out

Parameters

frequency – the frequency of the sine wave

baseline – the baseline of the wave

sample_size – the signal sample size

amplitude – the amplification of the sine wave

phase – the phase mode for the phasing of the sine wave

What the function Returns

Return the y-axis value of the graph

```
read_delay_file(filename="delays.csv")
```

Description

Read the delay value from the file

Parameters

filename – the file name

What the function Returns

Return the delay values

```
delay_signal(signal, delay)
```

Description

Get the signal with the delay value and then calculate it

Parameters

signal – the signal value

delay – the delay value

What the function Returns

Return the calculated delay value for the signal

```
apply_delay(signal, delay)
```

Description

Apply the delay value to the signal and shift it

Parameters

signal – the signal value

delay – the delay value

What the function Returns

Return the shifted value of the signal

`splice_signal(signal, step)`

Description

Splice the signal based on the step

Parameters

signal – the signal value

step – the amount of step requires to splice

What the function Returns

Return the spliced signal

`reset_output_file(filename)`

Description

Reset the file by removing the old path name

Parameters

filename – the file name

What the function Returns

Return nothing

`write_signal(signal, filename)`

Description

Write the generated signal into the file

Parameters

signal – the signal value

filename – the file name

What the function Returns

Return nothing

write_original_signals(signal, sampled_signal, filename)

Description

Write the original signals to the file

Parameters

signal – the signal value

sampled_signal – the sampled signal value

filename – the file name

What the function Returns

Return nothing

Contribution

| NAME | SECTION |
|-----------------|--|
| Chuin Jet Ong | Installation & Execution 1. Apply Noise to Signal 2. Delay Generator 3. Fractal Delay Application |
| | Functionality 1. Apply Noise to Signal 2. Delay Generator 3. Fractal Delay Application |
| Phi Long Nguyen | Installation & Execution 1. Noise Generator 2. Subfile Generator |
| | Functionality (Comments and Description are added in the source code itself) 1. Noise Generator 2. Subfile Generator |
| | |