



UNIVERSITY OF IOANNINA

Department of Computer Science and Engineering

Virtual Labs 2:

"Fading Processes-Channel Characterization"

Documentation

Referring to Courses:

Wireless Links

of

Computer Science and Engineering

at

University of Ioannina

Prepared by

Priftis Filippo and Prifti Joanis

at

Computer Science and Engineering

at

University of Ioannina

Portability

This project may execute in every operating system that supports octave.

Project Structure

FIRST_GUI.m: Αυτή η συνάρτηση λειτουργεί σαν “main”, δηλαδή καλεί όλες τις συναρτήσεις που είναι απαραίτητες για την τρέξι το πρόγραμμα και κατά συνέπεια και η τελική προσομοίωση. Αρχικά καλεί την show_FIRST_GUI() η οποία αρχικοποιεί και εμφανίζει το κυρίως γραφικό περιβάλλον(**βλέπε παρακάτω Εικόνα 1**). Στη συνέχεια καλεί την realCarSimulator() η οποία εμφανίζει το παράθυρο της προσομοίωσης(βλέπε παρακάτω).

FIRST_GUI_def.m: Αυτή η συνάρτηση περιέχει τις συναρτήσεις που αρχικοποιούν τα στοιχεία του κυρίως γραφικού περιβάλλοντος καθώς και τις συναρτήσεις callbacks(καλούνται κάθε φορά που ο χρήστης αλληλεπιδρά με κάποιο στοιχείο του κυρίως γραφικού περιβάλλοντος, πχ κουμπιά κτλ)για κάθε στοιχείο.

realCarSimulator.m: Αυτή η συνάρτηση δημιουργεί τον κόσμο της προσομοίωσης(αυτοκίνητο, κτήρια, δρόμος, κεραία) και επιπλέον προσομοιώνει τη διάδοση του σήματος κατά την διέλευση του αυτοκινήτου στον δρόμο μεταξύ κεραίας και αυτοκινήτου ενώ υπάρχουν ανάμεσα εμπόδια(πχ κτήρια).

showReport.m: Αυτή η συνάρτηση εμφανίζει τις τελικές αναφορές(γραφικές παραστάσεις) αφού ολοκληρωθεί πρώτα η προσομοίωση(Under Construction!).

jakes.m: The jakes fading signal creation method.

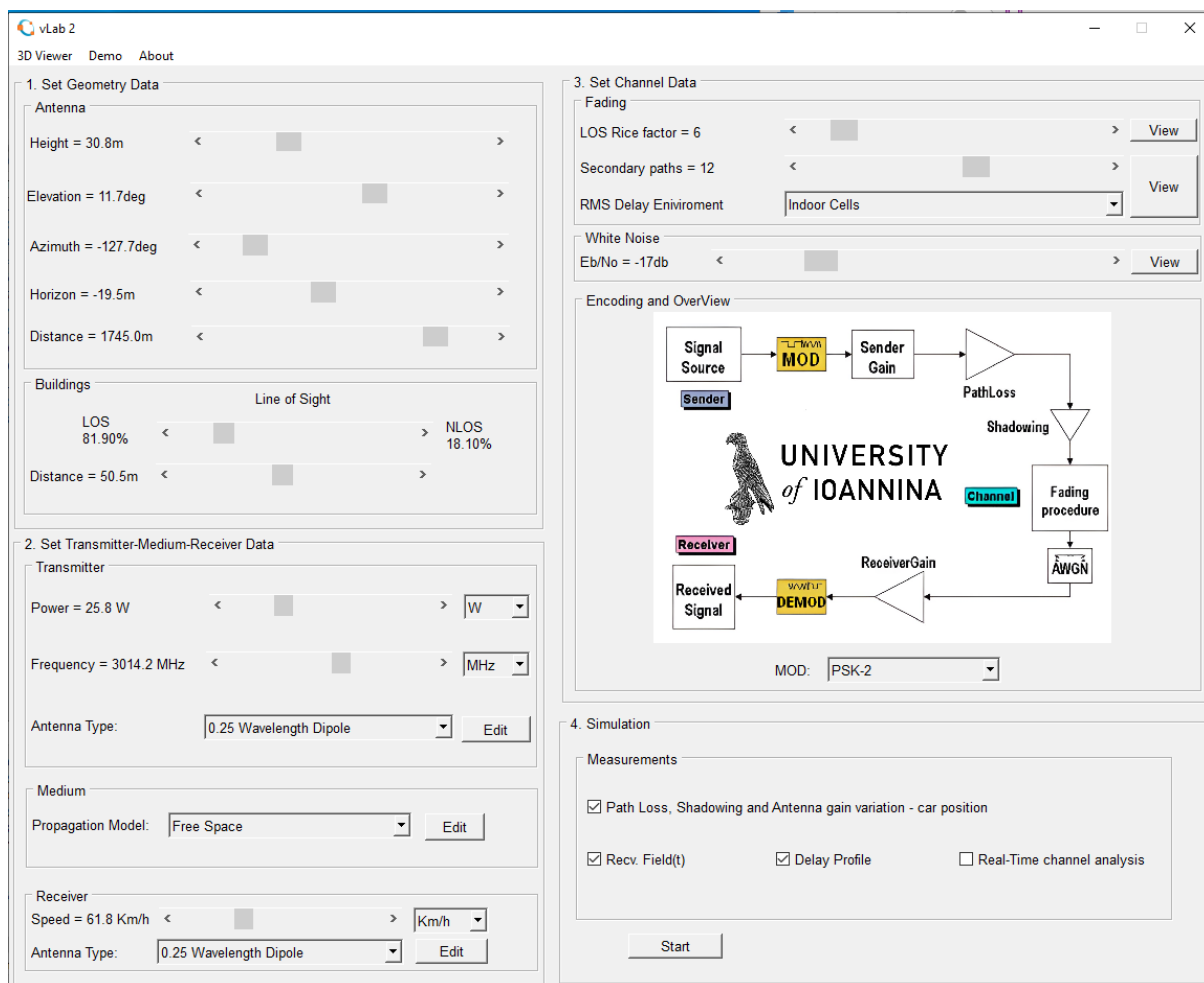
pulses.m: creates a signal of N smoothed pulses, emulating the shadowing imposed by the buildings.

berfading.m:This function was used as it is from matlab. It has some modification so that it works properly on octave. Yet this function is not fully implemented and has some errors that need to be handled.

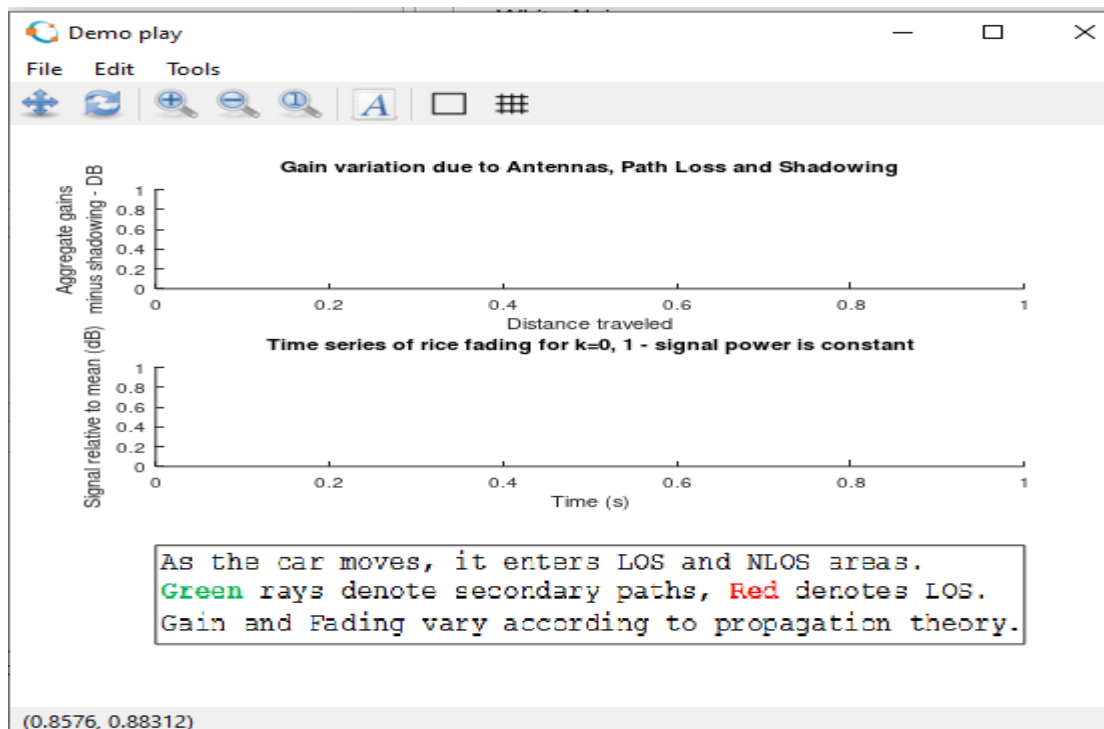
smooth.m: Αυτή η συνάρτηση είναι της matlab και λειτουργεί ακριβώς όπως και στη matlab χωρίς καμία αλλαγή.

ricepdf.m: Rice/Rician probability density function (pdf).

TVLABS2CustomMeasurement.m: Custom TMeasurement for internal use in the VLABs2. Functions are modified to return the Gains of the receiving antenna Gr, the transmitting antenna Gs and the space loss L.



Εικόνα 1: Κυρίως παράθυρο εφαρμογής

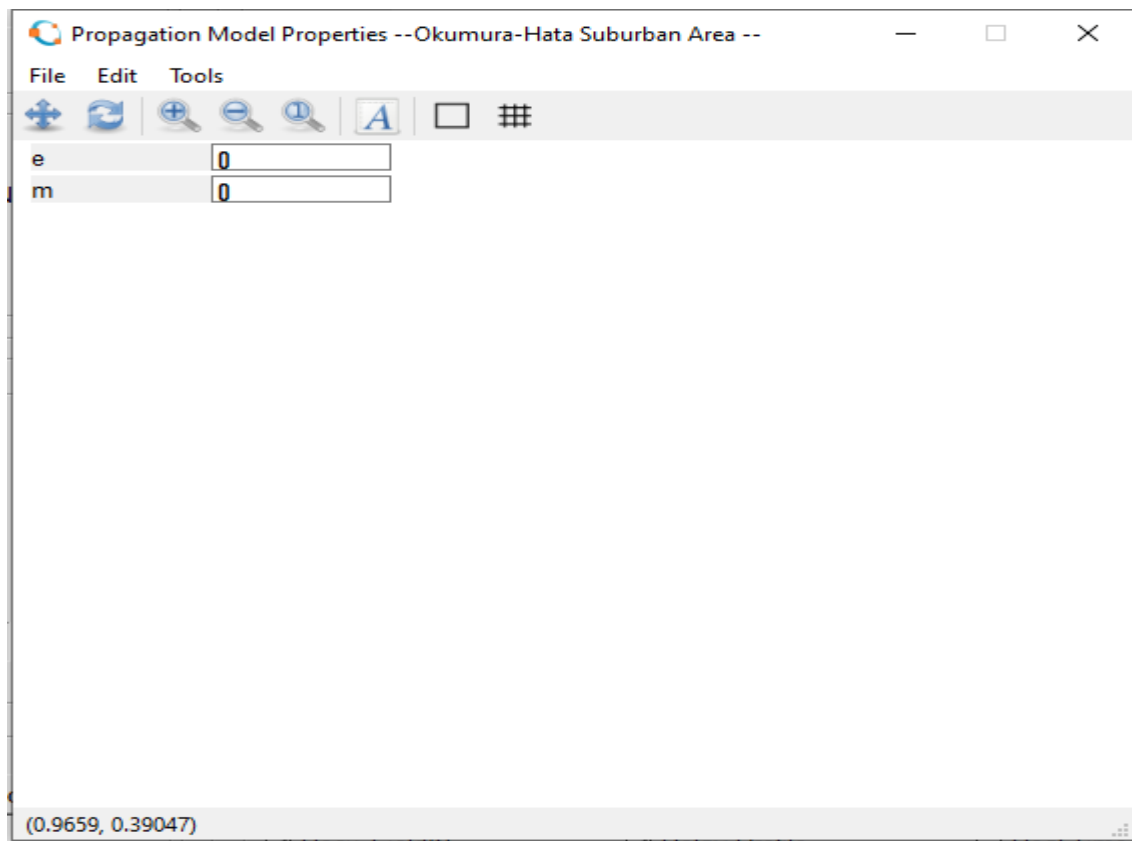


Εικόνα 2: Demo button figure

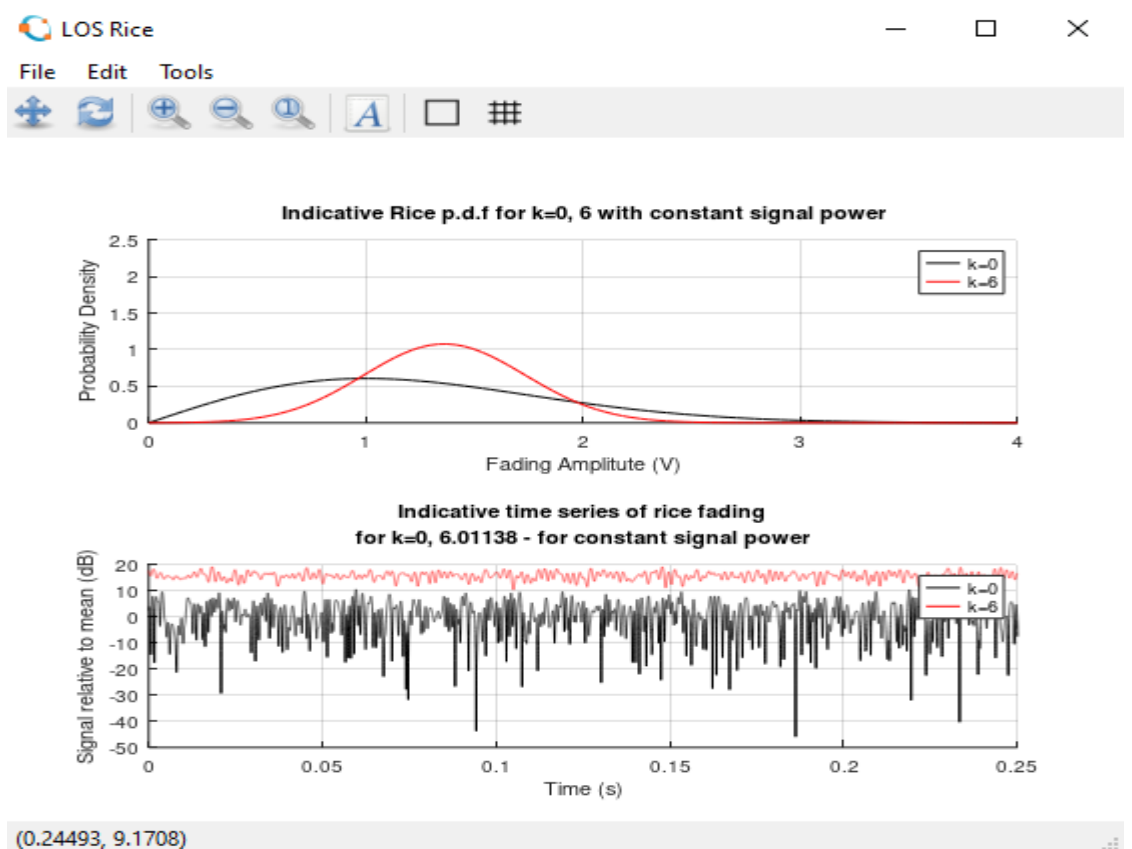
The 'Transmitter Antenna Properties --Hertzian Dipole --' window shows a form with two input fields: 'DI' and 'Rin', both set to '0'. The window includes a menu bar with 'File', 'Edit', and 'Tools', and a toolbar with various icons.

(0.98894, 0.68694)

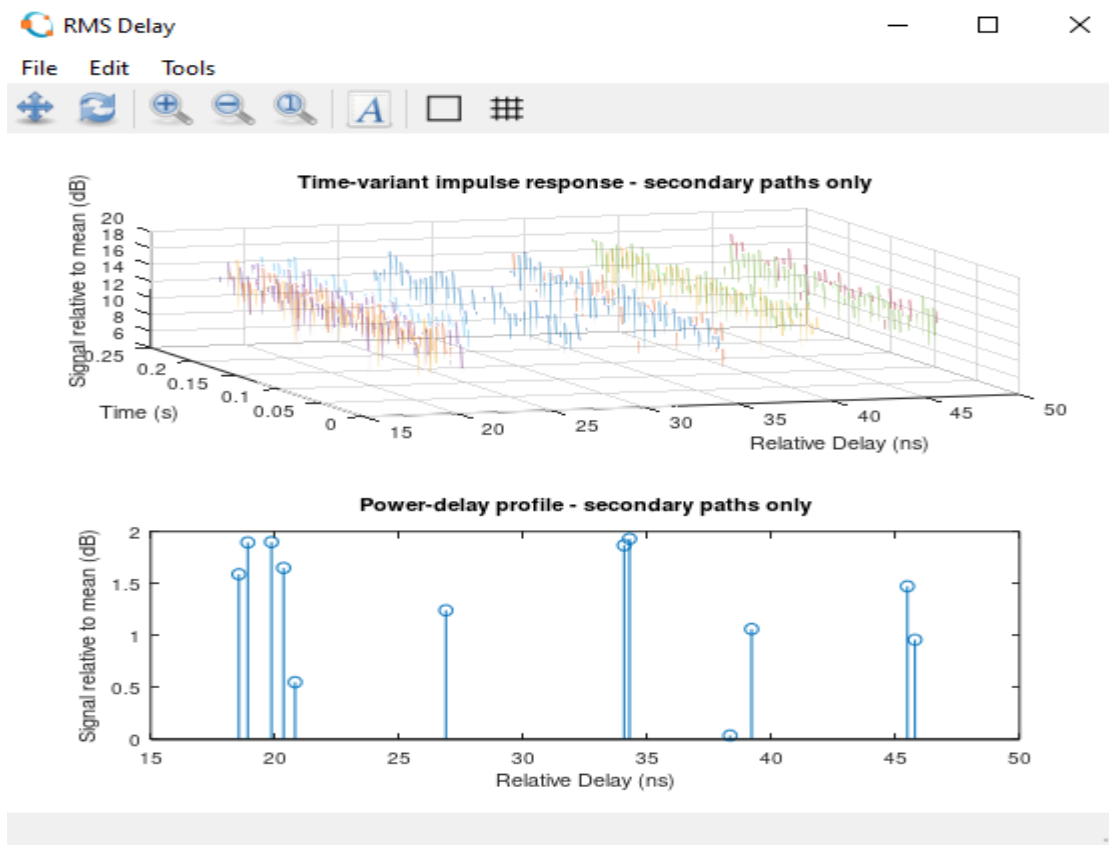
Εικόνα 3: Edit parameters from Transmitter or Receiver Antenna



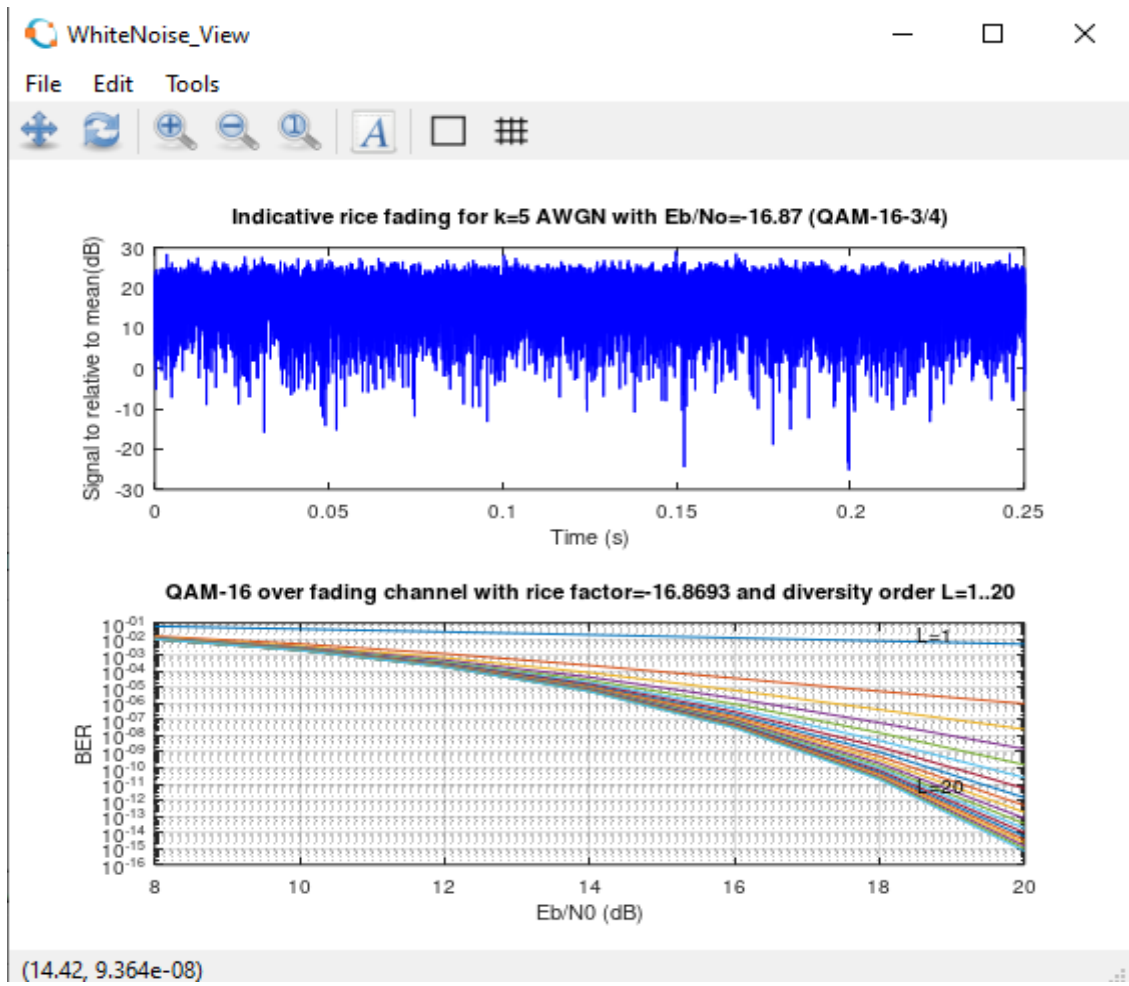
Εικόνα 4: Edit parameters from Medium Area



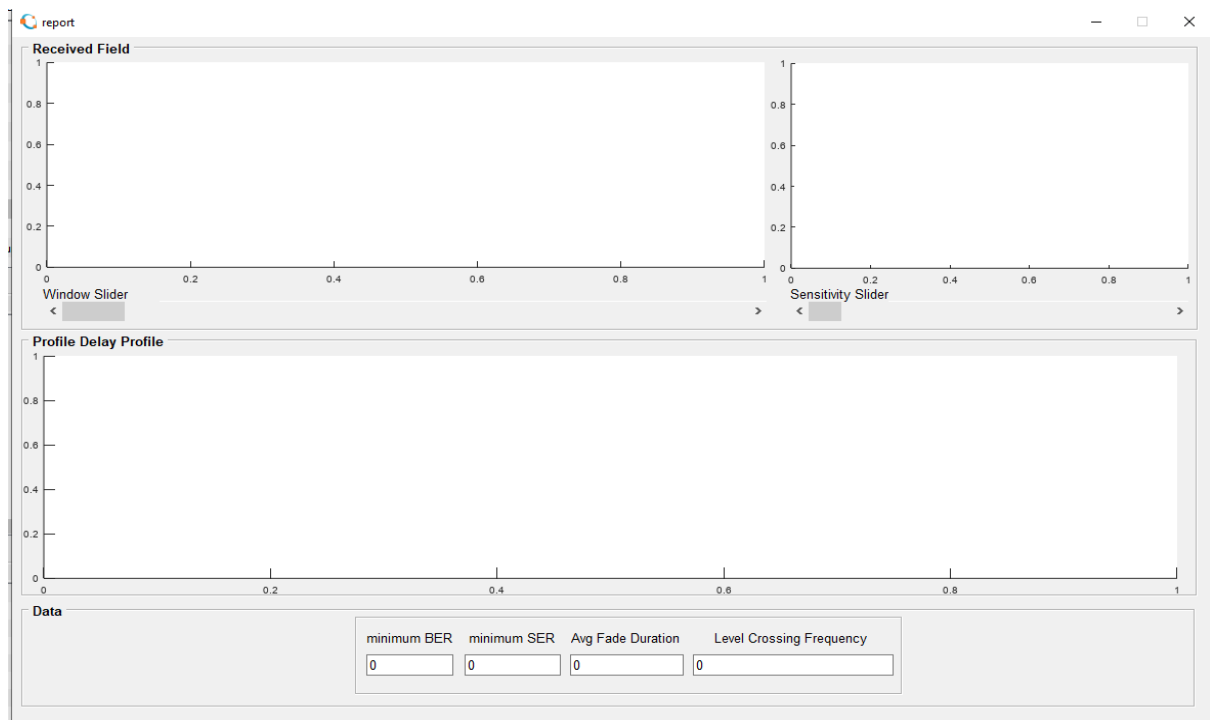
Εικόνα 5: View button figure of LOS Rice



Εικόνα 6: View button figure of RMS Delay



Εικόνα 7: View button figure of White Noise



Εικόνα 8: Final report

Ροή προγράμματος

Αρχικά, η εκτέλεση της εφαρμογής ξεκινάει από την συνάρτηση FIRST_GUI(βρίσκεται στο FIRST_GUI.m) η οποία με την σειρά της καλεί την συνάρτηση FIRST_GUI_def(βρίσκεται στο FIRST_GUI_def.m) ώστε να εμφανιστεί το κυρίως παράθυρο(**Εικόνα 1**) της εφαρμογής με τις επιλογές παραμετροποίησης του πειράματος καθώς και το παράθυρο με το αυτοκίνητο.

Τα γραφικά στοιχεία του κυρίως παραθύρου(**Εικόνα 1**) αρχικοποιούνται στην συνάρτηση show_FIRST_GUI() με την μορφή struct το καθένα.

Στο κυρίως παράθυρο μπορούν να δωθούν τιμές για το πείραμα από τους αντίστοιχους sliders όπως επίσης και τα drop down menus . Για τους sliders και τα πεδία αυτά υπάρχουν αντίστοιχα για το καθένα μία συνάρτηση(callback, δηλαδή που εκτελείται κάθε φορά που ο χρήστης αλληλεπιδρά με το στοιχείο του γραφικού περιβάλλοντος) που είναι υπεύθυνη να παίρνει την τιμή αυτή και να ενημερώνει το αντίστοιχο label στα αριστερά ώστε να μπορεί να βλέπει ο χρήστης την τιμή που έχει ο slider εκείνη την στιγμή όπως επίσης και την μονάδα μέτρησης. Οι συναρτήσεις αυτές είναι της μορφής GUIcomponent_dolt().

Στην συνέχεια ο χρήστης μπορεί να παραμετροποιήσει τα πεδία της κεραίας του αποστολέα(Transmitter) η/και του παραλήπτη(Receiver) που είναι επιλεγμένη εκείνη τη στιγμή στο drop down menu πατώντας το κουμπί Edit. Μόλις πατηθεί το κουμπί αυτό καλείται η συνάρτηση TransmitAndReceiv_Edit_dolt() η οποία ελέγχει ποιος τύπος κεραίας είναι επιλεγμένος εκείνη την στιγμή και καλεί με την σειρά της την συνάρτηση setFigContext() η οποία καθορίζει το ή τα πεδία της αντίστοιχης κεραίας και εμφανίζει ένα νέο παράθυρο(**Εικόνα 3**) με αυτά τα πεδία. Με παρόμοιο τρόπο λειτουργεί το Edit στο Medium panel με την διαφορά ότι η συνάρτηση που καλείται είναι η Medium_Edit_dolt() που με την σειρά της καλεί την setPropagationModel() η οποία καθορίζει τα αντίστοιχα πεδία και τα εμφανίζει αντίστοιχα σε ένα παράθυρο(**Εικόνα 4**).

Σημαντική παρατήρηση 1(TODO): Όσον αφορά τα πεδία που εμφανίζονται πατώντας τα αντίστοιχα κουμπιά Edit δεν τα χρησιμοποιούμε κάπου ενώ θα έπρεπε ώστε να γίνει σωστά η προσομοίωση.

Επιπλέον, ο χρήστης μπορεί να πατήσει κάποιο κουμπί View και να δει γραφικές παραστάσεις σχετικά με το πείραμα(**Εικόνες 5-6-7**). Συγκεκριμένα κάθε φορά που θα πατηθεί κάποιο από τα κουμπιά view καλείται η αντίστοιχη View_dolt() συνάρτηση.

Σημείωση: Όλες οι συναρτήσεις των κουμπιών View είναι της μορφής όνομα_View_dolt(), όπου όνομα=LOS ή RMS ή White_Noise.

Σημαντική παρατήρηση 2(TODO): Από το panel Encoding and OverView τα στοιχεία του MOD drop down menu δεν χρησιμοποιούνται κάπου ενώ θα έπρεπε.

Σημαντική παρατήρηση 3(TODO): Από το panel Simulation/Measurements τα checkboxes(Path Loss, Shadowing and Antenna gain variation – car position, Recv. Field(t), Delay Profile, Real-Time channel analysis) δεν χρησιμοποιούνται κάπου ενώ θα έπρεπε. Έχει υλοποιηθεί το πρωτότυπο της συνάρτησης για το κάθε checkbox η οποία καλείται κανονικά

κάθε φορά που ο χρήστης κάνει κλικ (check/uncheck) πράγμα το οποίο σημαίνει ότι μπορεί να χρησιμοποιηθεί μελλοντικά από άλλον προγραμματιστή.

Σημαντική παρατήρηση 4(TODO): Όταν πατηθεί το κουμπί Start εκτελείται επαναληπτικά η συνάρτηση `realCarSimulator()` και ανάλογα με τις παραμέτρους που έχει δώσει ο χρήστης στους slider της απόστασης και LOS αρχικοποιούνται το «πάχος» των κτηρίων και η απόσταση τους από τον δρόμο. Επιπλέον, μέσω της συνάρτησης `realCarSimulator()` μεταβάλλεται διαρκώς η θέση του αυτοκινήτου στον δρόμο(μέσω του τρίτου ορίσματος το οποίο χρησιμοποιείται στον υπολογισμό της νέας θέσης του αυτοκινήτου) ώστε να πραγματοποιείται η κίνηση. Τέλος, καλείται η `showReport()` η οποία είναι υπεύθυνη για την εμφάνιση των τελικών γραφημάτων-αποτελεσμάτων. **Ο ΛΟΓΟΣ** που υπάρχει η for-loop εδώ μέσα είναι καθαρά γιατί κάθε φορά ζωγραφίζεται από το μηδέν η προσομοίωση ώστε να φαίνεται η κίνηση του αυτοκινήτου. Καλό θα ήταν να μην υπάρχει εκεί και να υλοποιούνταν με κάποιον τρόπο μέσα στην `realCarSimulator()` και επιπλέον θα εξοικονομούσαν υπολογιστικούς πόροι .

Σημαντική παρατήρηση 5(TODO): Στην `realCarSimulator()` αρχικά ζωγραφίζονται σε ένα figure 6 κτήρια(3 δεξιά και 3 αριστερά), ένας δρόμος, μια κεραία και ένα αυτοκίνητο. Οι σχεδίαση αυτών έγινε καθαρά με μαθηματική μοντελοποίηση με την χρήση μαθηματικών συναρτήσεων όπως επίσης και τη χρήση `meshgrid`. Για να προκύψει αυτό το αποτέλεσμα έγιναν πολλά πειράματα με το «χέρι» δοκιμάζοντας διάφορους αριθμούς μέχρι να βγει το επιθυμητό αποτέλεσμα. Προτείνουμε να προστεθούν και άλλες παράμετροι στο πρωτότυπο της συνάρτησης `realCarSimulator()` ώστε να μπορεί να μεταβάλλεται και η κεραία και να φαίνεται το αποτέλεσμα στο figure. Επιπλέον, προτείνουμε να μουν textures στα αντικείμενα ώστε να είναι και πιο κατανοητό στον χρήστη.

Σημαντική παρατήρηση 6(TODO): Η `showFinalReport()`(**Εικόνα 8**) καλείται στο τέλος της προσομοίωσης και αυτό που κάνει είναι να εμφανίζει τα συγκεντρωτικά αποτελέσματα σε ένα νέο figure υπό την μορφή γραφικών παραστάσεων. Κανονικά θα έπρεπε να συλλέγει όλες τις παραμέτρους που όρισε ο χρήστης ώστε να κάνει τους τελικούς υπολογισμούς και να εμφανίσει τις κατάλληλες γραφικές παραστάσεις πράγμα το οποίο δεν υλοποιήσαμε(**Εικόνα 8**).

Σημαντική παρατήρηση 7(TODO): Όταν ο χρήστης πατάει κάποιο κουμπί View ή περιμένει να αποτελέσματα της προσομοίωσης κάποια View ανάλογα και με τον υπολογιστή του χρήστη αργούν λίγο να εμφανιστούν.

Τερματισμός της εφαρμογής:

- 1) Ο χρήστης μπορεί να κλείσει το παράθυρο με το αυτοκίνητο πριν πατήσει το κουμπί Start και απλά θα κλείσει αυτό το παράθυρο. Το κυρίως παράθυρο θα παραμείνει ανοικτό.
- 2) Αν ο χρήστης κλείσει το κυρίως παράθυρο τότε τερματίζει όλη η λειτουργία του προγράμματος μέσω της συνάρτησης `myCloseReq()` που βρίσκεται στο `FIRST_GUI_def.m`
- 3) Αν ο χρήστης έχει ξεκινήσει την προσομοίωση(πάτησε το κουμπί Start) και πατήσει το Χ(κλείσιμο) στο παράθυρο με το αυτοκίνητο ενώ εκτελείται η προσομοίωση τότε θα κλείσει αυτό το παράθυρο μέσω της συνάρτησης `myCarCloseReq()` που βρίσκεται στο

FIRST_GUI_def.m, το κυρίως παράθυρο θα παραμείνει ανοικτό ενώ θα εμφανιστεί το παράθυρο με τα συγκεντρωτικά αποτελέσματα.

- 4) Κατά την διάρκεια της προσομοίωσης ο χρήστης δεν μπορεί να πειράξει ή να κλείσει το κυρίως παράθυρο παρά μόνο αν κλείσει το παράθυρο με το αυτοκίνητο.