



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΥΠΗΡΕΣΙΕΣ ΒΑΣΙΖΟΜΕΝΕΣ ΣΤΗ ΘΕΣΗ ΤΟΥ
ΚΙΝΗΤΟΥ ΤΕΡΜΑΤΙΚΟΥ (LOCATION BASED SERVICES).
ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΕΝΤΟΠΙΣΜΟΥ ΣΤΗΡΙΖΟΜΕΝΟΥ ΣΤΗΝ
ΤΑΥΤΟΤΗΤΑ ΚΥΨΕΛΗΣ (CELL ID), ΤΗΝ ΠΡΟΠΟΡΕΙΑ
ΣΥΓΧΡΟΝΙΣΜΟΥ (TA) ΚΑΙ ΤΗΝ ΙΣΧΥ ΛΗΨΗΣ ΣΗΜΑΤΟΣ
(RXLEV)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΑΡΗ Σ. ΤΖΟΥΜΑ

Επιβλέπων : Μιχαήλ Ε. Θεολόγου
Καθηγητής Ε.Μ.Π

Αθήνα, Ιούνιος 2003



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΥΠΗΡΕΣΙΕΣ ΒΑΣΙΖΟΜΕΝΕΣ ΣΤΗ ΘΕΣΗ ΤΟΥ
ΚΙΝΗΤΟΥ ΤΕΡΜΑΤΙΚΟΥ (LOCATION BASED SERVICES).
ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΕΝΤΟΠΙΣΜΟΥ ΣΤΗΡΙΖΟΜΕΝΟΥ ΣΤΗΝ
ΤΑΥΤΟΤΗΤΑ ΚΥΨΕΛΗΣ (CELL ID), ΤΗΝ ΠΡΟΠΟΡΕΙΑ
ΣΥΓΧΡΟΝΙΣΜΟΥ (TA) ΚΑΙ ΤΗΝ ΙΣΧΥ ΛΗΨΗΣ ΣΗΜΑΤΟΣ
(RxLev)**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΑΡΗ Σ. ΤΖΟΥΜΑ

Επιβλέπων : Μιχαήλ Ε. Θεολόγου
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19^η Ιουνίου 2003.

.....
Μ. Ε. ΘΕΟΛΟΓΟΥ
ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π

.....
Γ. Ι. ΣΤΑΣΙΝΟΠΟΥΛΟΣ
ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π

.....
Ε. Δ. ΣΥΚΑΣ
ΚΑΘΗΓΗΤΗΣ Ε.Μ.Π

Αθήνα, Ιούνιος 2003

.....

ΑΡΗΣ Σ. ΤΖΟΥΜΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2003 – All rights reserved

ΠΕΡΙΛΗΨΗ

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη των τεχνολογιών εντοπισμού θέσης του κινητού τερματικού, καθώς και η πραγματοποίηση ενός αλγόριθμου εντοπισμού θέσης με χρήση πληροφοριών από το δίκτυο.

Αρχικά παρουσιάζονται όλες οι γνωστές μέθοδοι με τα πλεονεκτήματα και μειονεκτήματά τους, καθώς και τα διάφορα ήδη διαθέσιμα εμπορικά πακέτα της αγοράς.

Κατόπιν αναλύεται ο αλγόριθμος εντοπισμού θέσης που υλοποιήθηκε, με τα διάφορα στάδιά του. Ο αλγόριθμος αυτός κάνει χρήση των πληροφοριών που είναι διαθέσιμες από το δίκτυο, δηλαδή της ταυτότητας κυψέλης (cell id), της προπορείας συγχρονισμού (timing advance) και της ισχύος λήψης σήματος (rx level).

Τέλος δίνονται τα συγκεντρωτικά αποτελέσματα από τις μετρήσεις που πραγματοποιήθηκαν με βάση τον συγκεκριμένο αλγόριθμο και αξιολογείται η ακρίβεια και η απόδοσή του συγκριτικά με την απλή μέθοδο της ταυτότητας κυψέλης (cell id).

ABSTRACT

The scope of this thesis is the study of mobile location technologies and the development of a location algorithm by using network based information.

Firstly, all known location methods are presented with their advantages and disadvantages, as well as various commercial packages currently available in the market.

Afterwards, the implemented location algorithm is being analyzed, with its various stages. This algorithm makes use of information which is available through the network, such as the cell id, timing advance and the received power level.

Finally, the results of the measurements which were taken using this algorithm are given, in order to evaluate the algorithm's accuracy comparing with the simple Cell ID method.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

| | |
|--|-----------|
| 1. ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ | 13 |
| 1.1. ΚΙΝΗΤΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ | 13 |
| 1.2. ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ (Location Based Services, LBS) | 13 |
| 1.3. ΚΑΤΗΓΟΡΙΕΣ ΥΠΗΡΕΣΙΩΝ | 14 |
| 1.4. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΥΠΗΡΕΣΙΩΝ | 15 |
| 1.4.1. ΥΠΗΡΕΣΙΕΣ ΤΥΠΟΥ PULL | 15 |
| 1.4.2. ΥΠΗΡΕΣΙΕΣ ΤΥΠΟΥ PUSH | 15 |
| 1.5. ΔΥΝΑΤΟΤΗΤΑ ΓΕΩΓΡΑΦΙΚΟΥ ΠΡΟΣΔΙΟΡΙΣΜΟΥ - ΑΚΡΙΒΕΙΑ | 16 |
| 2. ΜΕΘΟΔΟΙ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ | 21 |
| 2.1. ΠΡΟΛΟΓΟΣ | 21 |
| 2.2. ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ | 21 |
| 2.3. ΔΙΑΧΩΡΙΣΜΟΣ ΜΕΘΟΔΩΝ | 21 |
| 2.4. ΜΕΘΟΔΟΙ ΒΑΣΙΣΜΕΝΕΣ ΣΤΟ ΔΙΚΤΥΟ (NETWORK BASED) | 22 |
| 2.4.1. ΑΝΑΓΝΩΡΙΣΗ ΤΑΥΤΟΤΗΤΑΣ ΚΥΨΕΛΗΣ (CELL ID) | 22 |
| 2.4.2. ΠΡΟΠΟΡΕΙΑ ΣΥΓΧΡΟΝΙΣΜΟΥ (TIMING ADVANCE, ENHANCED CELL-ID) | 23 |
| 2.4.3. ΙΣΧΥΣ ΛΗΨΗΣ ΣΗΜΑΤΟΣ (SIGNAL STRENGTH) | 26 |
| 2.4.3.a. ΜΟΝΤΕΛΑ ΡΑΔΙΟΚΑΛΥΨΗΣ | 26 |
| 2.4.3.b. ΥΛΟΠΟΙΗΣΗ ΕΝΤΟΠΙΣΜΟΥ | 28 |
| 2.4.4. ΧΡΟΝΟΣ ΑΦΙΞΗΣ (TIME OF ARRIVAL, TOA) | 30 |
| 2.4.5. ΔΙΑΦΟΡΑ ΧΡΟΝΟΥ ΑΦΙΞΗΣ (TDOA / E-OTD) | 30 |
| 2.4.6. ΓΩΝΙΑ ΛΗΨΗΣ (ANGLE OF ARRIVAL, AOA) | 32 |
| 2.5. ΜΕΘΟΔΟΙ ΒΑΣΙΣΜΕΝΕΣ ΣΤΟ Κ/Τ (MOBILE BASED) | 34 |
| 2.5.1. ΔΟΡΥΦΟΡΙΚΟΣ ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΘΕΣΗΣ (GPS) | 34 |
| 2.5.2. ΔΙΑΦΟΡΙΚΟ GPS (DGPS) | 35 |
| 2.5.3. ΥΠΟΒΟΗΘΟΥΜΕΝΟ GPS (A-GPS) | 35 |
| 2.6. ΣΥΓΚΕΝΤΡΩΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΜΕΘΟΔΩΝ | 35 |
| 3. ΥΠΑΡΧΟΥΣΑ ΤΕΧΝΟΛΟΓΙΑ | 39 |
| 3.1. ΠΡΟΛΟΓΟΣ | 39 |
| 3.2. CELLPOINT | 39 |
| 3.3. SNARTRACK | 40 |
| 3.4. Ericsson Mobile Positioning System (MPS) | 41 |
| 3.5. CPS Cursor | 42 |
| 3.6. NOKIA mPosition | 42 |
| 3.7. TRUEPOSITION Wireless Location System | 43 |
| 3.8. SIGNALSOFT Corporation | 43 |
| 3.9. BT CELLNET | 45 |
| 3.10. ΠΙΝΑΚΕΣ ΥΠΗΡΕΣΙΩΝ-ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ | 45 |

4. ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΕΝΤΟΠΙΣΜΟΥ _____ 51

| | |
|---|-----|
| 4.1. ΕΙΣΑΓΩΓΗ _____ | 51 |
| 4.3. ΙΣΤΟΡΙΑ ΑΛΓΟΡΙΘΜΟΥ _____ | 52 |
| 4.3.1. ΠΡΩΤΑ ΣΤΑΔΙΑ – ΑΛΓΟΡΙΘΜΟΣ ΓΙΑ ΤΟ RXLEV | 52 |
| 4.3.2. ΤΕΛΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ – (CELL ID, TA, RXLEV) | 55 |
| 4.4. ΑΛΓΟΡΙΘΜΟΣ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ _____ | 57 |
| 4.4.1. ΕΞΑΓΩΓΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ (PROCESS 1) | 58 |
| 4.4.1.a. Υπο-Ρουτίνα <i>Timer1_Timer()</i> | 59 |
| 4.4.1.b. Υπο-Ρουτίνα <i>SortMatrix</i> | 62 |
| 4.4.1.c. Υπο-Ρουτίνα <i>ProcessData</i> | 63 |
| 4.4.1.d. Υπο-Ρουτίνα <i>StoreValues</i> | 66 |
| 4.4.1.e. PROCESS 1 – Διάγραμμα Ροής | 67 |
| 4.4.2. ΑΝΑΖΗΤΗΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ (PROCESS 2) | 69 |
| 4.4.2.a. Τύπος και Δομή της Βάσης Δεδομένων | 69 |
| 4.4.2.b. Η Μήτρα "FinalOk" – Εφαρμογή της Αναζήτησης | 73 |
| 4.4.2.c. Υπο-Ρουτίνα <i>Databasefind</i> | 74 |
| 4.4.2.d. Υπο-Ρουτίνα <i>ProcessDist</i> | 84 |
| 4.4.2.e. Υπο-Ρουτίνα <i>GetFinalMatrix</i> | 85 |
| 4.4.2.f. PROCESS 2 - Διάγραμμα Ροής | 86 |
| 4.4.3. ΥΠΟΛΟΓΙΣΜΟΣ ΕΚΤΙΜΟΥΜΕΝΗΣ ΘΕΣΗΣ ΤΟΥ Κ/Τ (PROCESS 3) | 87 |
| 4.4.3.a. Συνάρτηση <i>MicroCheck</i> | 88 |
| 4.4.3.b. Υπο-Ρουτίνα <i>NotLocate</i> | 89 |
| 4.4.3.c. Υπο-Ρουτίνα <i>DrawMicro</i> | 91 |
| 4.4.3.d. Υπο-Ρουτίνα <i>TAHATAMatch</i> | 91 |
| 4.4.3.e. Υπο-Ρουτίνα <i>CheckNeighbourHata</i> | 94 |
| 4.4.3.f. Υπο-Ρουτίνα <i>CalcDistances</i> | 109 |
| 4.4.3.g. Υπο-Ρουτίνα <i>DrawGraph</i> | 112 |
| 4.4.3.h. PROCESS 3 – Διάγραμμα Ροής | 112 |

5. ΜΕΤΡΗΣΕΙΣ – ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ _____ 117

| | |
|--|-----|
| 5.1. ΣΧΕΤΙΚΑ ΜΕ ΤΙΣ ΜΕΤΡΗΣΕΙΣ _____ | 117 |
| 5.2. ΠΕΡΙΠΤΩΣΕΙΣ ΜΕΤΡΗΣΕΩΝ _____ | 118 |
| 5.2.1. ΜΕΤΡΗΣΗ ΜΕ ΤΗ ΜΕΓΑΛΥΤΕΡΗ ΑΚΡΙΒΕΙΑ | 118 |
| 5.2.2. ΜΕΤΡΗΣΗ ΜΕ ΤΗ ΜΙΚΡΟΤΕΡΗ ΑΚΡΙΒΕΙΑ | 119 |
| 5.2.3. ΣΦΑΛΜΑ ΣΤΗΝ ΠΕΡΙΟΧΗ ΕΜΠΙΣΤΟΣΥΝΗΣ | 120 |
| 5.2.4. ΤΟ SERVING CELL ΔΕΝ ΕΙΝΑΙ Ο ΚΟΝΤΙΝΟΤΕΡΟΣ Σ/Β | 122 |
| 5.2.5. ΠΟΛΥ ΜΕΓΑΛΗ ΑΚΡΙΒΕΙΑ ΤΗΣ ΗΑΤΑ RADIUS ΤΟΥ SERV. CELL | 123 |
| 5.2.6. ΤΟ ΠΡΟΒΛΗΜΑ ΣΤΙΣ ΚΟΝΤΙΝΕΣ ΑΠΟΣΤΑΣΕΙΣ | 124 |
| 5.2.7. ΔΙΑΦΟΡΕΣ ΠΕΡΙΠΤΩΣΕΙΣ | 125 |
| 5.3. ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ - ΑΝΑΛΥΣΗ _____ | 127 |
| 5.3.1. ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ | 127 |
| 5.3.2. ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ | 135 |

ΠΑΡΑΡΤΗΜΑ Α (ΠΑΡΟΥΣΙΑΣΗ ΚΩΔΙΚΑ) _____ 139

ΠΑΡΑΡΤΗΜΑ Β (ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ) _____ 227

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

| | | |
|-------------|-------|----|
| ΠΙΝΑΚΑΣ I | | 14 |
| ΠΙΝΑΚΑΣ II | | 17 |
| ΠΙΝΑΚΑΣ III | | 23 |
| ΠΙΝΑΚΑΣ IV | | 36 |
| ΠΙΝΑΚΑΣ V | | 46 |
| ΠΙΝΑΚΑΣ VI | | 47 |

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

| | | |
|------------|-------|-----|
| ΣΧΗΜΑ 1.1 | | 22 |
| ΣΧΗΜΑ 1.2 | | 24 |
| ΣΧΗΜΑ 1.3 | | 25 |
| ΣΧΗΜΑ 1.4 | | 29 |
| ΣΧΗΜΑ 1.5 | | 31 |
| ΣΧΗΜΑ 1.6 | | 32 |
| ΣΧΗΜΑ 4.1 | | 52 |
| ΣΧΗΜΑ 4.2 | | 53 |
| ΣΧΗΜΑ 4.3 | | 55 |
| ΣΧΗΜΑ 4.4 | | 57 |
| ΣΧΗΜΑ 4.5 | | 58 |
| ΣΧΗΜΑ 4.6 | | 68 |
| ΣΧΗΜΑ 4.7 | | 69 |
| ΣΧΗΜΑ 4.8 | | 70 |
| ΣΧΗΜΑ 4.9 | | 73 |
| ΣΧΗΜΑ 4.10 | | 86 |
| ΣΧΗΜΑ 4.11 | | 87 |
| ΣΧΗΜΑ 4.12 | | 92 |
| ΣΧΗΜΑ 4.13 | | 92 |
| ΣΧΗΜΑ 4.14 | | 93 |
| ΣΧΗΜΑ 4.15 | | 93 |
| ΣΧΗΜΑ 4.16 | | 94 |
| ΣΧΗΜΑ 4.17 | | 94 |
| ΣΧΗΜΑ 4.18 | | 96 |
| ΣΧΗΜΑ 4.19 | | 97 |
| ΣΧΗΜΑ 4.20 | | 98 |
| ΣΧΗΜΑ 4.21 | | 99 |
| ΣΧΗΜΑ 4.22 | | 99 |
| ΣΧΗΜΑ 4.23 | | 100 |

| | | |
|------------|-------|-----|
| ΣΧΗΜΑ 4.24 | | 101 |
| ΣΧΗΜΑ 4.25 | | 103 |
| ΣΧΗΜΑ 4.26 | | 104 |
| ΣΧΗΜΑ 4.27 | | 105 |
| ΣΧΗΜΑ 4.28 | | 106 |
| ΣΧΗΜΑ 4.29 | | 107 |
| ΣΧΗΜΑ 4.30 | | 108 |
| ΣΧΗΜΑ 4.31 | | 109 |
| ΣΧΗΜΑ 4.32 | | 111 |
| ΣΧΗΜΑ 4.33 | | 113 |
| ΣΧΗΜΑ 4.34 | | 114 |
| ΣΧΗΜΑ 5.1 | | 128 |
| ΣΧΗΜΑ 5.2 | | 128 |
| ΣΧΗΜΑ 5.3 | | 129 |
| ΣΧΗΜΑ 5.4 | | 130 |
| ΣΧΗΜΑ 5.5 | | 130 |
| ΣΧΗΜΑ 5.6 | | 131 |
| ΣΧΗΜΑ 5.7 | | 132 |
| ΣΧΗΜΑ 5.8 | | 132 |
| ΣΧΗΜΑ 5.9 | | 133 |
| ΣΧΗΜΑ 5.10 | | 134 |



ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ (Location Based Services)



1. ΕΙΣΑΓΩΓΗ ΣΤΙΣ ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ

1.1. ΚΙΝΗΤΕΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΕΣ

Η έξαρση που παρατηρείται τα τελευταία χρόνια στον τομέα των Κινητών Τηλεπικοινωνιών είναι ένα γεγονός αναμφισβήτητο. Η αμεσότητα, η ασφάλεια και η πληθώρα των υπηρεσιών που προσφέρουν σήμερα τα κινητά τηλέφωνα, τα έχουν καταστήσει αναπόσπαστο κομμάτι στη ζωή των ανθρώπων.

Η μεγάλη ζήτηση που υπάρχει στον τομέα αυτό από την αγορά, έχει ως αποτέλεσμα (από την πλευρά των παροχέων υπηρεσιών), την αύξηση του συναγωνισμού και την διαρκή αναζήτηση για νέες και καλύτερες υπηρεσίες. Μέσα λοιπόν σ' αυτά τα πλαίσια, μία νέα μορφή υπηρεσιών έχει εμφανιστεί

η οποία έχει να κάνει με υπηρεσίες βασισμένες στην θέση του κινητού τηλεφώνου.

1.2. ΥΠΗΡΕΣΙΕΣ ΘΕΣΗΣ (Location Based Services, LBS)

Τι σημαίνει λοιπόν Location Based Services;

Ας προσπαθήσουμε να το κατανοήσουμε βλέποντας τις παρακάτω ερωτήσεις που περιγράφουν το πρόβλημα:

"Γεια, τι κάνεις Πού βρίσκεσαι;"

"Από πού θα πάω για να φτάσω εκεί που θέλω;"

"Πού είναι το κοντινότερο πρακτορείο εφημερίδων;"

"Από πού θα πάρω το λεωφορείο για να πάω εκεί που θέλω;"

"Πού είναι το κοντινότερο σινεμά που παίζει αυτή την ταινία;"

"Είδα την ταινία, πού θα πάω για φαγητό;"

"Χάλασε το αυτοκίνητό μου, αλλά δεν ξέρω που βρίσκομαι..."

"Είμαι σε διακοπές, που να πάω και τι να δω;"

Όλες αυτές οι ερωτήσεις και τα "πού" θα μπορούν να απαντηθούν σε λίγο καιρό από τις υπηρεσίες θέσης (LBS).

Κάθε άνθρωπος θέλει να γνωρίζει όσο το δυνατόν περισσότερα για τον χώρο στον οποίο βρίσκεται και κινείται. Επίσης ενδιαφέρεται να ξέρει για το που βρίσκονται τα αγαπημένα του άτομα. Όλες οι παραπάνω "ανάγκες" έρχονται να καλυφθούν από τις "LBS", μια έννοια η οποία είναι πολύ φρέσκια στον χώρο των κινητών τηλεπικοινωνιών κι αυτό γιατί μόλις πολύ πρόσφατα άρχισε η διεθνής αγορά να ασχολείται με το θέμα.

Το μεγαλύτερο κίνητρο για την εισαγωγή τέτοιου είδους υπηρεσιών αποτέλεσε η νομική επιβολή στις ΗΠΑ έτσι ώστε μέχρι το τέλος του 2005 και σταδιακά, όλες οι καινούργιες συσκευές να μπορούν να παρέχουν πληροφορία θέσης όταν πραγματοποιείται μια επείγουσα κλήση.

Επίσης και στην Ευρώπη έχει προταθεί από την Ευρωπαϊκή Ένωση να γίνει υποχρεωτικός ο εντοπισμός για λόγους επείγουσας ανάγκης από τον Ιανουάριο του 2002 και μέχρι το 2008 να είναι συμβατές όλες οι συσκευές.

Επιπλέον, οι πολύ μεγάλες επενδύσεις στα συστήματα τρίτης γενιάς (3G) που έχουν ήδη γίνει από όλους τους οργανισμούς, επιβάλλουν την εισαγωγή νέων υπηρεσιών στα ήδη υπάρχοντα συστήματα, ώστε να υπάρξει άμεσο οικονομικό όφελος και νέοι πόροι κερδών.

1.3. ΚΑΤΗΓΟΡΙΕΣ ΥΠΗΡΕΣΙΩΝ

Από τα παραπάνω καταλαβαίνουμε ότι οι υπηρεσίες θέσης (LBS) πρόκειται να έρθουν στο προσκήνιο πολύ σύντομα. Ποιες είναι όμως οι υπηρεσίες αυτές;

Εκτός από την περίπτωση έκτακτης ανάγκης, η γνώση της γεωγραφικής θέσης του κινητού μπορεί να βοηθήσει και στην δημιουργία πολλών άλλων νέων υπηρεσιών.

Οι υπηρεσίες αυτές μπορούν να καταταχθούν σε οκτώ επιμέρους κατηγορίες όπως φαίνεται στον Πίνακα Ι.

| Κατηγορία | Περιγραφή |
|-------------------------------|--|
| Τοπικές Πληροφορίες | Τι υπάρχει κοντά (Τουριστικά μέρη, Εστιατόρια, Κινηματογράφοι), τοπικά νέα, τοπικός Χρυσός Οδηγός |
| Προσανατολισμός – Δρομολόγηση | Ενημέρωση παρούσας κίνησης, Επιλογή συντομότερου δρόμου, Καθοδήγηση δρομολογίου, Δημόσιες Μεταφορές |
| Εμπόριο | Μηνύματα (SMS) Βάση αδείας, κουπόνια από κοντινά καταστήματα, ειδικές προσφορές |
| Ασφάλεια | Η θέση μου, η θέση της οικογένειάς μου, που βρίσκεται το αυτοκίνητό μου |
| Διαχείριση πόρων | Διαχείριση προσωπικού, οχημάτων, κατανομή πόρων |
| Κοινωνικά, Παιχνίδια | Μοιράζομαι τη θέση μου με τους συναδέλφους/ φίλους μου, γνωριμία με άτομα που βρίσκονται κοντά μου, παιχνίδια αλληλεπίδρασης |
| Έκτακτη Ανάγκη | Εντοπισμός σε επείγουσα κλήση όπως ορίζεται από τις Κοινοτικές Διατάξεις. |
| Χρέωση κλήσης | Χρέωση της κλήσης ανάλογα με την τοποθεσία (Σπίτι, Δουλειά, στο Δρόμο) |

ΠΙΝΑΚΑΣ Ι

Όπως γίνεται φανερό η γνώση και μόνο της γεωγραφικής θέσης ενός κινητού τηλεφώνου μπορεί να δημιουργήσει ένα πλήθος νέων πρωτοποριακών υπηρεσιών, των οποίων η επιτυχία είναι σίγουρη, αφού η τοποθεσία είναι ένα βασικό στοιχείο που επηρεάζει τους ανθρώπους ως προς το πώς οργανώνονται και σχετίζονται με τον υπόλοιπο κόσμο.

Άλλωστε, πολλές είναι και οι έρευνες που επιβεβαιώνουν την δεδομένη επιτυχία τέτοιου είδους υπηρεσιών, όπως για παράδειγμα μια μελέτη στη Μεγάλη Βρετανία που δείχνει ότι το 65% των ερωτηθέντων χρηστών θα άλλαζε δίκτυο εάν επρόκειτο να αποκτήσει πρόσβαση σε "LBS". Το αντίστοιχο ποσοστό για τη Γερμανία και τη Γαλλία είναι 54% και 44%.

1.4. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΥΠΗΡΕΣΙΩΝ

Οι υπηρεσίες χωρίζονται επίσης σε δύο διαφορετικούς τύπους (Pull/ Push) ανάλογα με το ποια διαδικασία ακολουθείται για να ενεργοποιηθούν.

1.4.1. ΥΠΗΡΕΣΙΕΣ ΤΥΠΟΥ PULL

Βάσει αυτού του μοντέλου, ο χρήστης του κινητού αιτείται την εξυπηρέτησή του από κατάλληλη υπηρεσία σχετική με τη θέση του, κινώντας την διαδικασία προσδιορισμού του. Το αποτέλεσμα είναι η εκτίμηση της θέσης του η οποία και διατίθεται στον πάροχο της υπηρεσίας.

Παραδείγματα Μοντέλου Pull.

- **Οδηγίες Ταξιδιού:** Βρίσκομαι ΕΔΩ, πως θα πάω ΕΚΕΙ;
- **Κλήση Ταξί:** Ο κάτοχος του κινητού ειδοποιεί ότι χρειάζεται ταξί. Η κατάλληλη υπηρεσία Ταξί που λειτουργεί στην περιοχή μπορεί αυτόματα να μεταφέρει την τοποθεσία και το νούμερο τηλεφώνου του κατόχου αμέσως στο πιο κοντινό ελεύθερο Ταξί. Ο κάτοχος ειδοποιείται με μία απάντηση που λέει ότι υπάρχει κάποιο Ταξί κοντά και ότι θα τον παραλάβει σύντομα.
- **Χρυσός Οδηγός:** Που είναι το κοντινότερο 'X'; Ο χρήστης υποδεικνύει τις κατηγορίες επιχειρήσεων για τις οποίες ενδιαφέρεται και 'κατεβάζει' στο κινητό του μία λίστα με σειρά απόστασης από την τοποθεσία στην οποία βρίσκεται.

1.4.2. ΥΠΗΡΕΣΙΕΣ ΤΥΠΟΥ PUSH

Βάσει αυτού του μοντέλου, ο χρήστης του κινητού είναι προεγγεγραμμένος σε επιλεγμένες υπηρεσίες/ εφαρμογές. Ο εξυπηρετητής της συγκεκριμένης εφαρμογής εκτελεί αναζήτηση στο σύστημα προσδιορισμού θέσης που εξυπηρετεί το συγκεκριμένο δίκτυο για χρήστες που πληρούν γεωγραφικά και άλλα κριτήρια.

Παραδείγματα Μοντέλου Push.

- **Διαφημίσεις:** Μελέτες έδειξαν ότι οι χρήστες μπορεί να είναι θετικοί σε διαφημίσεις στα κινητά τους με αντάλλαγμα κάποια ηλεκτρονικά κουπόνια, ή άλλους τύπους προσφορών και βραβείων. Πολλοί καταναλωτές μπορεί να επωφεληθούν με το να επιτρέπουν στους προμηθευτές να γνωρίζουν πότε βρίσκονται σε μικρή απόσταση από αυτούς.
- **Ενημέρωση Κίνησης στο δρόμο:** Παροχείς τέτοιων υπηρεσιών, ενημερώνουν τον εγγεγραμμένο συνδρομητή για το αν πλησιάζει σε δρόμο με μεγάλη κίνηση και προτείνονται εναλλακτικές διαδρομές. Για ανθρώπους που δεν θέλουν να καθυστερούν στο δρόμο, τέτοιου είδους υπηρεσίες θα βοηθήσουν στο να φθάνουν στον προορισμό τους έγκαιρα.

1.5. ΔΥΝΑΤΟΤΗΤΑ ΓΕΩΓΡΑΦΙΚΟΥ ΠΡΟΣΔΙΟΡΙΣΜΟΥ - ΑΚΡΙΒΕΙΑ

Η γνώση της γεωγραφικής θέσης στην οποία βρίσκεται ένας κάτοχος κινητού τηλεφώνου, αποτελεί σημαντικότερη πληροφορία για την παροχή υπηρεσιών "LBS".

Στα σημερινά δίκτυα αυτή η γνώση υπάρχει σε μειωμένο βαθμό, δηλαδή το δίκτυο γνωρίζει μόνο σε ποια κυψέλη βρίσκεται ένας συνδρομητής, με την ακτίνα της κυψέλης να κυμαίνεται από 50m – 35km.

Όπως φαίνεται όμως και από τον Πίνακα II, υπάρχουν αρκετοί τύποι υπηρεσιών οι οποίοι απαιτούν μεγαλύτερη ακρίβεια. Έτσι έχουν προταθεί αρκετές νέες μέθοδοι προσδιορισμού θέσης οι οποίες θα παρουσιαστούν στο Κεφάλαιο 2.

| ΑΠΑΙΤΟΥΜΕΝΗ ΑΚΡΙΒΕΙΑ | ΤΥΠΟΙ ΥΠΗΡΕΣΙΩΝ |
|--|---|
| Ανεξάρτητες από γεωγραφική θέση | Πλειοψηφία των σημερινών υπηρεσιών κινητής τηλεφωνίας, τιμές μετοχών, σκορ αγώνων, κ.ο.κ |
| Τοπικού ενδιαφέροντος (έως 200km) | Δελτία καιρού, Ανακοινώσεις ακραίων καιρικών φαινομένων τοπικού ενδιαφέροντος, Πληροφορίες Κίνησης σε Οδικά Δίκτυα |
| Εστιασμένου Τοπικού ενδιαφέροντος (έως 20km) | Τοπικές ειδήσεις, εκδηλώσεις |

| | |
|--------------------------|--|
| Έως 1km | Διαχείριση στόλου, Συμβουλές για αποφυγή προβληματικών σημείων του οδικού δικτύου, Γεωγραφικός Χρυσός Οδηγός |
| 500m έως 1km | Αγροτικές και ημιαστικές υπηρεσίες άμεσης βοήθειας, Διαχείριση ανθρώπινου δυναμικού (πχ. Δίκτυο πωλητών), Υπηρεσίες πληροφοριών θέσης. |
| 100m (67%) 300m (95%) | Απαιτήσεις FCC(99-245) για κλήσεις επειγόντων περιστατικών που χρησιμοποιούν μεθόδους* βασισμένες στο δίκτυο. |
| 50m (67%) 150m (95%) | Απαιτήσεις FCC(99-245) για κλήσεις επειγόντων περιστατικών που χρησιμοποιούν μεθόδους* βασισμένες στο τερματικό. |
| 10m – 50m | Εντοπισμός εμπορευμάτων, Πλοήγηση, Συστήματα κατεύθυνσης με χάρτη |

ΠΙΝΑΚΑΣ ΙΙ (ETSI – TS 22.071)

*Οι μέθοδοι αναλύονται στο επόμενο κεφάλαιο



ΚΕΦΑΛΑΙΟ 2

ΜΕΘΟΔΟΙ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ



2. ΜΕΘΟΔΟΙ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ

2.1. ΠΡΟΛΟΓΟΣ

Συνολικά έχουν προταθεί διάφορες μέθοδοι εντοπισμού θέσης. Μερικές από αυτές έχουν εφαρμοστεί μόνο σε δοκιμαστικά συστήματα, ενώ άλλες έχουν προχωρήσει και διατίθενται ως εμπορικά προϊόντα και ήδη χρησιμοποιούνται. Παρόλα αυτά καμιά μέθοδος μέχρι στιγμής δεν έχει ξεχωρίσει ιδιαίτερα, αφού η καθεμιά έχει πλεονεκτήματα και μειονεκτήματα.

Εμείς σ' αυτό το κεφάλαιο θα προσπαθήσουμε να παρουσιάσουμε τις διάφορες μεθόδους, και να τις αξιολογήσουμε από πλευράς ακρίβειας, ποιότητας αλλά και απαιτήσεων.

2.2. ΑΞΙΟΛΟΓΗΣΗ ΜΕΘΟΔΩΝ

Η αξιολόγηση που θα επιχειρήσουμε έχει να κάνει με την επίδοση της κάθε μίας μεθόδου στους παρακάτω τομείς:

1. Ακρίβεια Μεθόδου
2. Μεταβολή ακρίβειας ανάλογα με το περιβάλλον διάδοσης
3. Επιβάρυνση Δικτύου υπό μορφή νέου εξοπλισμού (SW/HW) και τροποποιήσεων
4. Επιβάρυνση Κινητού Τερματικού (SW/HW)
5. Απαιτούμενος χρόνος για την εξαγωγή αποτελεσμάτων

2.3. ΔΙΑΧΩΡΙΣΜΟΣ ΜΕΘΟΔΩΝ

Οι μέθοδοι που θα παρουσιαστούν διακρίνονται σε τρεις διαφορετικές κατηγορίες ανάλογα με τις οντότητες που εμπλέκονται στον καθορισμό της θέσης.

1. Μέθοδοι βασισμένες στο Δίκτυο (Network Based):

Οι απαραίτητες μετρήσεις γίνονται από τους Base Stations και κάποιο υπεύθυνο κέντρο εντοπισμού θέσης (Location Centre) υπολογίζει την γεωγραφική θέση.

2. Μέθοδοι βασισμένες στο Κινητό Τερματικό (Mobile-Based):

Το κινητό τερματικό εκτελεί τις απαραίτητες μετρήσεις και υπολογίζει μόνο του την γεωγραφική θέση.

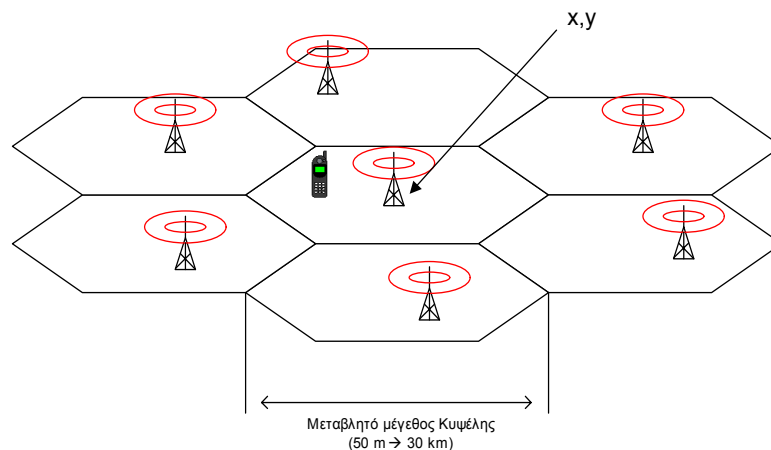
3. Μέθοδοι βοηθούμενες από το Κινητό Τερματικό (Mobile-Assisted):

Το κινητό τερματικό κάνει τις μετρήσεις και τις στέλνει στο Location Centre για να υπολογίσει αυτό την γεωγραφική θέση. Πρόκειται ουσιαστικά για μια υποκατηγορία των Network Based και Mobile Based και γι' αυτόν το λόγο δεν αναλύονται ξεχωριστά (πολλές από τις Network/Mobile Based μεθόδους μπορούν με κατάλληλες μετατροπές να θεωρηθούν Mobile-Assisted). Απλά υποδηλώνουν την συνεργασία και των δύο οντοτήτων (Κ/Τ, Δίκτυο) για την εξαγωγή αποτελεσμάτων.

2.4. ΜΕΘΟΔΟΙ ΒΑΣΙΣΜΕΝΕΣ ΣΤΟ ΔΙΚΤΥΟ (NETWORK BASED)

2.4.1. ΑΝΑΓΝΩΡΙΣΗ ΤΑΥΤΟΤΗΤΑΣ ΚΥΨΕΛΗΣ (CELL ID)

Πρόκειται για την απλούστερη μέθοδο για τον εντοπισμό ενός κινητού τηλεφώνου. Βασίζεται στην πληροφορία αναγνώρισης της κυψέλης στην οποία βρίσκεται το κινητό, η οποία είναι ήδη διαθέσιμη στο δίκτυο (Home Location Register [HLR], Visitor Location Register [VLR]). Με λίγα λόγια, η μέθοδος αυτή καθορίζει ποια είναι η κυψέλη που εξυπηρετεί το κινητό τηλέφωνο και χρησιμοποιούνται οι συντεταγμένες του Base Station (ή του κέντρου της περιοχής κάλυψης) ως μία εκτίμηση της θέσης του κινητού. Η μέθοδος αυτή αναφέρεται και ως "CGI" (Cell Global Identification) και μια σχηματική παράσταση φαίνεται παρακάτω (Σχήμα 1.1).



ΣΧΗΜΑ 1.1

Πλεονεκτήματα Μεθόδου:

- Δεν χρειάζονται υπολογισμοί για την απόκτηση της πληροφορίας Γεωγραφικής Θέσης, αφού αυτή υπάρχει ήδη.
- Ελάχιστες αλλαγές απαιτούνται στα ήδη υπάρχοντα συστήματα για την εφαρμογή αυτής της μεθόδου.
- Ο χρόνος που απαιτείται για την εξαγωγή αποτελεσμάτων είναι ελάχιστος.

Μειονεκτήματα Μεθόδου:

- Η ακρίβεια της μεθόδου εξαρτάται άμεσα από την ακτίνα της κυψέλης και κυμαίνεται από 50m περίπου για εσωτερικούς χώρους έως 30km για αγροτικές περιοχές.
- Λόγω φαινομένων διάδοσης, η κυψέλη που εξυπηρετεί το κινητό δεν είναι πάντα και η πιο κοντινή. Έτσι, η ακρίβεια δεν είναι τόσο καλή όσο θα μπορούσε να εξαχθεί απλά από την πυκνότητα των σταθμών βάσης και την γεωμετρία των κυψελών.

Στον Πίνακα III δίνονται οι διάφοροι τύποι κυψελών και οι διαστάσεις τους

| Τύπος Κυψέλης | Τοποθεσία Κεραίας | Μέγεθος Κυψέλης (km) |
|----------------------|---------------------------------------|-----------------------------|
| Large MacroCell | Πάνω από το υψηλότερο επίπεδο κτηρίων | 3 – 30 |
| Small MacroCell | Πάνω από το υψηλότερο επίπεδο κτηρίων | 1 – 3 |
| Microcell | Κοντά στο υψηλότερο επίπεδο κτηρίων | 0.1 – 1 |
| PicoCell | Κάτω από το υψηλότερο επίπεδο κτηρίων | 0.01 – 1 |
| NanoCell | Κάτω από το υψηλότερο επίπεδο κτηρίων | 0.001 – 0.01 |

ΠΙΝΑΚΑΣ III

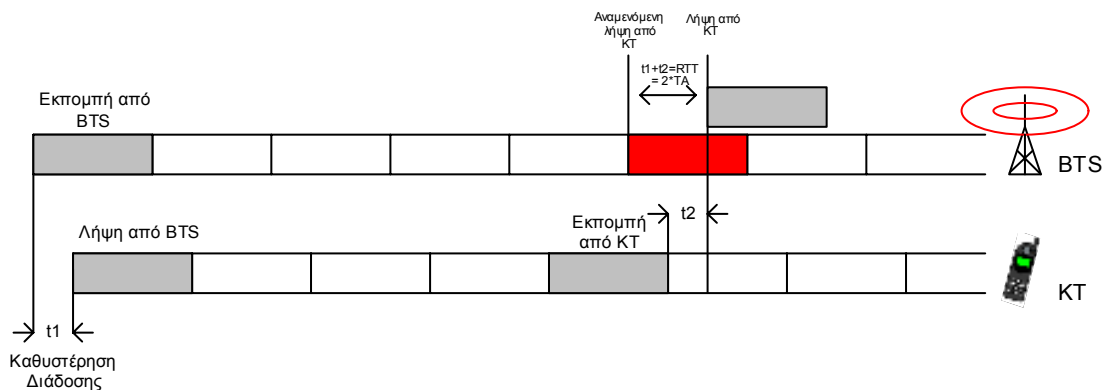
2.4.2. ΠΡΟΠΟΡΕΙΑ ΣΥΓΧΡΟΝΙΣΜΟΥ (Timing Advance, Enhanced Cell-ID)

Γνωρίζουμε ότι κάθε κανάλι στο GSM χωρίζεται σε χρονικά πλαίσια (time frames). Για να δουλέψει σωστά το σχήμα TDMA που χρησιμοποιείται στο GSM, τα time frames από το κάθε κινητό τηλέφωνο πρέπει να είναι συγχρονισμένα όταν λαμβάνονται από τον σταθμό βάσης (BTS). Αυτός ο συγχρονισμός επιτυγχάνεται χρησιμοποιώντας την έννοια του Timing Advance.

Ο βαθμός του συγχρονισμού μετριέται από τον Σταθμό Βάσης κατά την uplink μετάδοση, με το να ελέγχεται η θέση της εκπαιδευτικής ακολουθίας (training sequence). Αυτή η εκπαιδευτική ακολουθία είναι υποχρεωτική σε όλα τα πλαίσια που στέλνονται από το κινητό.

Βάσει αυτών των μετρήσεων ο Σταθμός Βάσης μπορεί να υπολογίσει το Timing Advance και να το στείλει πίσω στο κινητό κατά τη διάρκεια της πρώτης downlink μετάδοσης. Από την τιμή TA που λαμβάνει το κινητό, γνωρίζει πλέον πότε θα στείλει το πλαίσιο, έτσι ώστε να φτάσει στον Σταθμό Βάσης πλήρως συγχρονισμένο.

Το Timing Advance μας δίνει τον χρόνο που απαιτείται από το RF σήμα για να μεταδοθεί με την ταχύτητα του φωτός από το Κινητό Τερματικό στο Σταθμό Βάσης (ή αντίστροφα).



ΣΧΗΜΑ 1.2

Η τιμή του TA που στέλνεται έχει μέγεθος 6 bit, δηλαδή μας παρέχει ένα εύρος 2^6 τιμών (0 έως 63). Γνωρίζουμε ότι κατά τον υπολογισμό του TA πραγματοποιείται στρογγυλοποίηση στο πλησιέστερο bit. Η περίοδος του ενός bit είναι $\frac{48}{13} \mu\text{sec} = 3,69 \mu\text{sec}$. Επομένως, η αναλυτική ικανότητα για το round-trip-time είναι 1 bit και αντίστοιχα για τον χρόνο μετάδοσης χωρίς επιστροφή $\frac{1}{2}$ bit. Επομένως, αν λάβουμε υπόψη ότι η ταχύτητα διάδοσης είναι ίση με την ταχύτητα του φωτός ($c=3 \times 10^8 \text{ m/sec}$), τότε παίρνουμε ότι σε κάθε bit αντιστοιχούν

$$c \cdot \frac{T_b}{2} = 3 \cdot 10^8 \frac{3,69 \cdot 10^{-6}}{2} = 554 \text{m} \approx 550 \text{m}$$

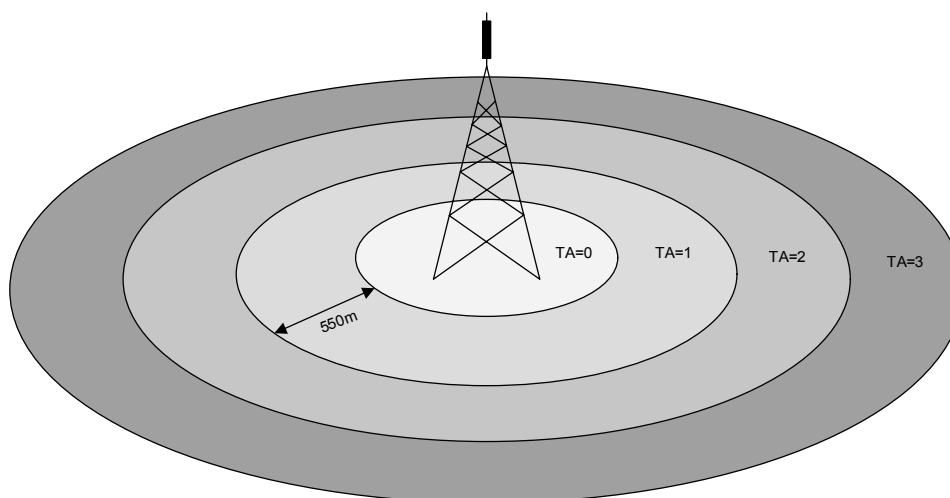
Το ίδιο συμπεραίνουμε αν κάνουμε την παρακάτω θεώρηση: Λαμβάνοντας υπόψη ότι η μέγιστη ακτίνα κυψέλης είναι 35km και χωρίζοντας αυτήν την απόσταση σε 64 βήματα (με την τιμή 63 να αντιστοιχεί στα 35km), έχουμε ότι ανά 1 βήμα TA καλύπτεται απόσταση $\approx 550 \text{m}$.

Τελικά, από τα παραπάνω συμπεραίνουμε ότι η πραγματική απόσταση (d) μεταξύ Κινητού Τερματικού και Σταθμού Βάσης θα δίνεται από τον τύπο:

$$[550 \cdot (TA)] \text{m} \leq d < [550 \cdot (TA + 1)] \text{m} \quad \text{για } TA \geq 0$$

Για την καλύτερη κατανόηση του παραπάνω τύπου ας μελετήσουμε την απεικόνιση που φαίνεται στο Σχήμα 1.3.

Όπως βλέπουμε, για κάθε τιμή TA που παίρνουμε μπορούμε να ορίσουμε μία περιοχή, στην οποία θα βρίσκεται το Κινητό Τερματικό (Κ/Τ). Για $TA=0$, αυτή η περιοχή είναι ένας κύκλος με κέντρο τον Σταθμό Βάσης και ακτίνα 550m. Για τις υπόλοιπες τιμές του TA, οι περιοχές είναι δακτύλιοι που όπως φαίνονται στο σχήμα έχουν όλοι διαφορά εξωτερικής-εσωτερικής ακτίνας ίση με 550m.

**ΣΧΗΜΑ 1.3**

Η τιμή του TA ανανεώνεται κατά την εγκατάσταση σύνδεσης φωνής/ δεδομένων ή με την εκκίνηση διαδικασίας αναζήτησης θέσης (silent call) από το BSC/ MSC αν το κινητό είναι σε idle mode.

Η μέθοδος αυτή επειδή χρησιμοποιεί και την πληροφορία του Cell-ID για την εξαγωγή αποτελεσμάτων αναφέρεται στην βιβλιογραφία και ως "Enhanced Cell-ID", δηλαδή βελτιωμένη Cell-ID. Πράγματι, τα αποτελέσματα αυτής της μεθόδου είναι καλύτερα από αυτά της απλής Cell-ID.

Πλεονεκτήματα Μεθόδου:

- Οι υπολογισμοί που χρειάζονται για την υλοποίηση της μεθόδου είναι πολύ λίγοι.
- Ελάχιστες αλλαγές απαιτούνται στα ήδη υπάρχοντα συστήματα για την εφαρμογή της.
- Ο χρόνος που απαιτείται για την εξαγωγή αποτελεσμάτων είναι ελάχιστος.
- Η ακρίβεια είναι βελτιωμένη σε σχέση με το Cell-ID και δεν επηρεάζεται τόσο πολύ από το μέγεθος της κυψέλης.

Μειονεκτήματα Μεθόδου:

- Λειτουργεί σωστά μόνο σε συνθήκες LOS (Line-Of-Sight) και τα όποια σφάλματα στην ακρίβεια οφείλονται σε συνθήκες NLOS όπου το σήμα διανύει μεγαλύτερες αποστάσεις από την πραγματική απόσταση μεταξύ Κ/Τ και BTS. Έτσι, μπορεί η πραγματική απόσταση να είναι τελικά μικρότερη από την ελάχιστη ακτίνα που υπολογίζει η μέθοδος.

2.4.3. ΙΣΧΥΣ ΛΗΨΗΣ ΣΗΜΑΤΟΣ (Signal Strength)

Το Κ/Τ μετράει τη στάθμη σήματος από όλους τους γειτονικούς Σταθμούς Βάσης και στέλνει αναφορά στον Σταθμό Βάσης που τον εξυπηρετεί. Έτσι, η πληροφορία για την στάθμη του ραδιοσήματος υπάρχει ήδη στο Δίκτυο. Παρακάτω παρουσιάζονται τα πιθανά μοντέλα για τον υπολογισμό της γεωγραφικής θέσης.

2.4.3.a. ΜΟΝΤΕΛΑ ΡΑΔΙΟΚΑΛΥΨΗΣ

i. Μοντέλο Ελευθέρου Χώρου

Το πιο απλό μοντέλο ραδιοκάλυψης είναι αυτό του Ελευθέρου Χώρου (Free Space). Βάσει αυτού του μοντέλου, ο λόγος ισχύος εκπομπής-λήψης στον κενό χώρο είναι:

$$\frac{P_R}{P_T} = G_T \cdot G_R \left[\frac{\lambda}{4\pi \cdot d} \right]^2$$

όπου

P_R : Η Ισχύς λήψης

P_T : Η Ισχύς εκπομπής

G_T : Το κέρδος της κεραίας εκπομπής

G_R : Το κέρδος της κεραίας λήψης

λ : Το μήκος κύματος

d : Η απόσταση μεταξύ πομπού και δέκτη

Βέβαια το πραγματικό περιβάλλον των Κινητών Επικοινωνιών δεν έχει καμία σχέση με τον ελεύθερο χώρο. Γι' αυτόν το λόγο έχουν προταθεί διάφορα εμπειρικά μοντέλα ραδιοκάλυψης, τα οποία χρησιμοποιούνται ανάλογα με το περιβάλλον διάδοσης.

ii. Μοντέλο Line Of Sight

Ο σκοπός αυτού του μοντέλου είναι να υπολογίζει τις απώλειες διάδοσης σε συγκεκριμένη μορφολογία εδάφους, όταν υπάρχει οπτική επαφή (LOS) μεταξύ του Κ/Τ και του Σταθμού Βάσης. Όταν αυτές οι συνθήκες ικανοποιούνται, το μοντέλο μπορεί να χρησιμοποιηθεί για τον υπολογισμό της απόστασης με δεδομένη την ισχύ λήψης.

$$L(\text{dB}) = 10 \cdot \alpha \cdot \log\left(\frac{f}{c}\right) - 10 \cdot \beta \cdot \log(4\pi \cdot d)$$

- όπου
- L : Η εξασθένιση σε dB
 - f : Η συχνότητα λειτουργίας
 - c : Η ταχύτητα του φωτός ($3 \cdot 10^8$ m/s)
 - d : Η απόσταση σε μέτρα (m)

Το μοντέλο χρησιμοποιεί έναν συντελεστή συχνότητας (α) και ένα συντελεστή εδάφους (β). Πρόκειται για ακέραιους αριθμούς και καθορίζονται εμπειρικά.

Αν η απόσταση που υπολογίζεται (d) παρουσιάζει απόκλιση x dB από την πραγματική, τότε ισχύει η σχέση

$$\frac{d+x(\text{dB})}{d} = 10^{\frac{1}{10\beta}}$$

Από αυτή την εξίσωση μπορεί να εξαχθεί μια κατανομή πιθανότητας για τη μετρούμενη απόσταση.

iii. Μοντέλο HATA

Βασισμένος στις αναφορές του OKUMURA ο HATA ανέπτυξε έναν εμπειρικό τύπο για τον υπολογισμό της απώλειας σήματος, ο οποίος είναι ο ακόλουθος:

✓ Για urban περιοχή:

$$L(\text{urban})(\text{dB}) = 69.55 + 26.16 \cdot \log(f_c) - 13.82 \cdot \log(h_{te}) - \alpha(h_{re}) + (44.9 - 6.55 \cdot \log(h_{te})) \cdot \log(d)$$

όπου ο συντελεστής διόρθωσης είναι

$$\alpha(h_{re}) = (1.1 \cdot \log(f_c) - 0.7) \cdot h_{re} - (1.56 \cdot \log(f_c) - 0.8) \text{ dB} \quad (\text{Small or medium City})$$

$$\alpha(h_{re}) = 8.29 \cdot (\log(1.54 \cdot h_{re}))^2 - 1.1 \text{ dB}, \text{ για } (f_c \leq 300 \text{ MHz})$$

$$\alpha(h_{re}) = 3.2 \cdot (\log(11.75 \cdot h_{re}))^2 - 4.97 \text{ dB}, \text{ για } (f_c \geq 300 \text{ MHz})$$

} (Large City)

✓ **Για Suburban περιοχή:**

$$L \text{ (dB)} = L \text{ (urban)} \text{ (dB)} - 2 \cdot \left[\log \left(\frac{f_c}{28} \right) \right]^2 - 5.4$$

✓ **Για open area:**

$$L \text{ (dB)} = L \text{ (urban)} \text{ (dB)} - 4.78 \cdot (\log (f_c))^2 + 18.33 \cdot \log (f_c) - 40.94$$

Στο μοντέλο του HATA, L είναι η εξασθένιση σε dB, f_c η συχνότητα λειτουργίας, h_{te} , h_{re} τα ύψη του πομπού και του δέκτη αντίστοιχα και d η απόσταση μεταξύ πομπού και δέκτη.

Το μοντέλο αυτό για να λειτουργήσει σωστά έχει κάποιες απαιτήσεις:

- Περιοχή Συχνοτήτων : 150 – 1500 MHz
- Ύψος της κεραίας του Σταθμού Βάσης: 30 – 200m
- Ύψος της κεραίας του Κινητού Τερματικού: 1 – 10m

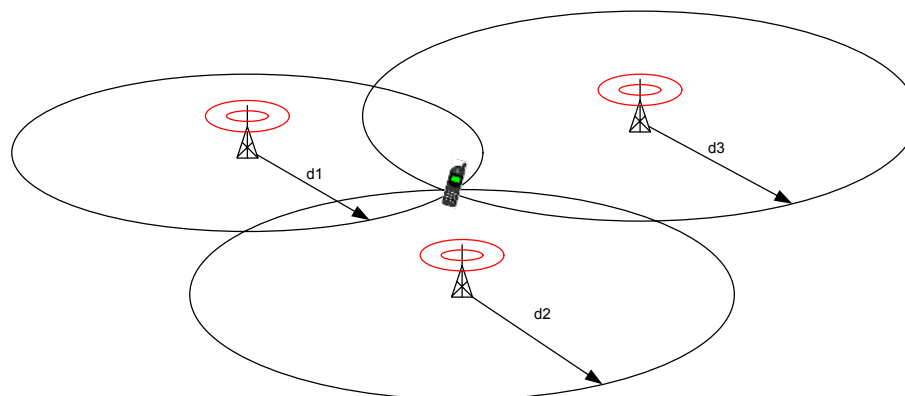
2.4.3.β. ΥΛΟΠΟΙΗΣΗ ΕΝΤΟΠΙΣΜΟΥ

Χρησιμοποιώντας λοιπόν ένα από τα υπάρχοντα μοντέλα ραδιοκάλυψης μπορούμε να βρούμε τις εκτιμώμενες αποστάσεις του Κ/Τ από τους γειτονικούς Σταθμούς Βάσης. Αυτό που μένει είναι ο εντοπισμός του Κ/Τ βάσει των υπάρχόντων εκτιμήσεων.

Μία μέθοδος φαίνεται στο Σχήμα 1.4. Γνωρίζοντας τις αποστάσεις d_1 , d_2 και d_3 , μπορούμε να βρούμε το σημείο τομής των τριών κύκλων το οποίο θα είναι και η ακριβής θέση του Κ/Τ.

Βέβαια ο προσδιορισμός του σημείου τομής δεν είναι τόσο απλός όσο φαίνεται, αφού το πιο πιθανό είναι ότι οι τρεις κύκλοι δεν θα τέμνονται στο ίδιο σημείο.

Το γεγονός ότι δεν γνωρίζουμε τις ακριβείς αποστάσεις αλλά έχουμε μόνο κάποια εκτίμηση, μας κάνει να συμπεράνουμε ότι τελικά αυτό που θα βρούμε θα είναι μία περιοχή της οποίας το χαρακτηριστικό είναι ότι θα ανήκει και στους τρεις κύκλους.

**ΣΧΗΜΑ 1.4**

Το μέγεθος της περιοχής και κατά συνέπεια η ακρίβεια της μεθόδου εξαρτάται από τους εξής παράγοντες:

- Το χρησιμοποιούμενο Μοντέλο Ραδιοκάλυψης και το Περιβάλλον
- Φαινόμενα Σκίασης (shadowing) και πολυδιαδρομικά (multipath)
- Αριθμός μετρήσεων για τον υπολογισμό των αποστάσεων.

Εκτός από αυτή, υπάρχουν και άλλες μέθοδοι σχετικές με την Ισχύ Λήψης οι οποίες βασίζονται σε πίνακες όπου καταχωρούνται μετρήσεις της λαμβανόμενης ισχύος σε διαφορετικές κατευθύνσεις και αποστάσεις. Διατηρώντας τέτοιους πίνακες είναι δυνατός ο εντοπισμός του Κ/Τ μέσω της σύγκρισης των μετρούμενων τιμών με τις ήδη αποθηκευμένες και του υπολογισμού της πιο πιθανής τοποθεσίας με τη βοήθεια κάποιου αλγόριθμου. Βέβαια τέτοιου είδους μέθοδοι αυξάνουν κατά πολύ την πολυπλοκότητα της όλης διαδικασίας, αν και καταλήγουν σε αρκετά ακριβή αποτελέσματα.

Πλεονεκτήματα Μεθόδου:

- Η πληροφορία υπάρχει ήδη στα τωρινά συστήματα.
- Ελάχιστες αλλαγές απαιτούνται στο δίκτυο για την εφαρμογή της.
- Ο χρόνος που απαιτείται για την εξαγωγή αποτελεσμάτων είναι μικρός.
- Η ακρίβεια είναι σχετικά καλή.

Μειονεκτήματα Μεθόδου:

- Είναι δύσκολο να επιτευχθεί πολύ υψηλή ακρίβεια.
- Τα φαινόμενα σκίασης και πολυδιαδρομής (Shadowing, Multipath) μειώνουν αισθητά την ακρίβεια, ειδικά σε κλειστούς χώρους, μέσα στο αυτοκίνητο κλπ.

2.4.4. ΧΡΟΝΟΣ ΑΦΙΞΗΣ (Time Of Arrival, TOA)

Η μέθοδος αυτή βασίζεται στη μέτρηση του χρόνου καθυστέρησης διάδοσης ενός γνωστού σήματος από το Κ/Τ στον Σταθμό Βάσης (ή και αντίστροφα). Έχοντας μετρήσει αυτόν τον χρόνο, η απόσταση θα δίνεται από τον τύπο

$$d_i = c \cdot t_i + c \cdot t_{\text{off},i}$$

όπου

d_i : Η υπολογιζόμενη απόσταση από τον i -Σταθμό Βάσης

c : Η ταχύτητα διάδοσης του φωτός

t_i : Ο χρόνος καθυστέρησης από τον i -Σταθμό Βάσης

$t_{\text{off},i}$: Διαφορά χρονισμού μεταξύ του ρολογιού του Κ/Τ και του ρολογιού του i -στού Σταθμού Βάσης.

Η μέτρηση της καθυστέρησης διάδοσης από τρεις Σταθμούς Βάσης, ορίζει τρεις αποστάσεις, οι οποίες με τη σειρά τους ορίζουν μονοσήμαντα ένα σημείο στον χώρο (ανάλογη περίπτωση με το Σχήμα 4 που αφορά τον εντοπισμό μέσω του Signal Strength).

Επειδή το $t_{\text{off},i}$ είναι άγνωστη μεταβλητή, το σφάλμα που εισάγεται είναι απαράδεκτο και γι' αυτό η μέθοδος αυτή απαιτεί τον συγχρονισμό όλων των Σταθμών Βάσης του δικτύου. Σ' αυτήν τη περίπτωση, η διαφορά χρονισμού δεν θα πάψει να υπάρχει, αλλά θα είναι πλέον σταθερή ανεξαρτήτως Σταθμού Βάσης ($t_{\text{off},i} = t_{\text{off}}$), εξαρτημένη όμως ακόμα από το ρολόι του Κ/Τ.

Λόγω της ύπαρξης σφάλματος εξαιτίας του t_{off} , η μέθοδος αυτή έχει εγκαταλειφθεί και την θέση της έχει πάρει η επόμενη κατά σειρά μέθοδος που θα εξετάσουμε, η οποία και απαλείφει οποιαδήποτε διαφορά χρονισμού. Επομένως δεν πρόκειται να αναλύσουμε περαιτέρω την παρούσα μέθοδο, ούτε θα παρουσιάσουμε τα πλεονεκτήματα και μειονεκτήματά της (εμπεριέχονται εν μέρει στην επόμενη μέθοδο).

2.4.5. ΔΙΑΦΟΡΑ ΧΡΟΝΟΥ ΑΦΙΞΗΣ (TDOA / E-OTD)

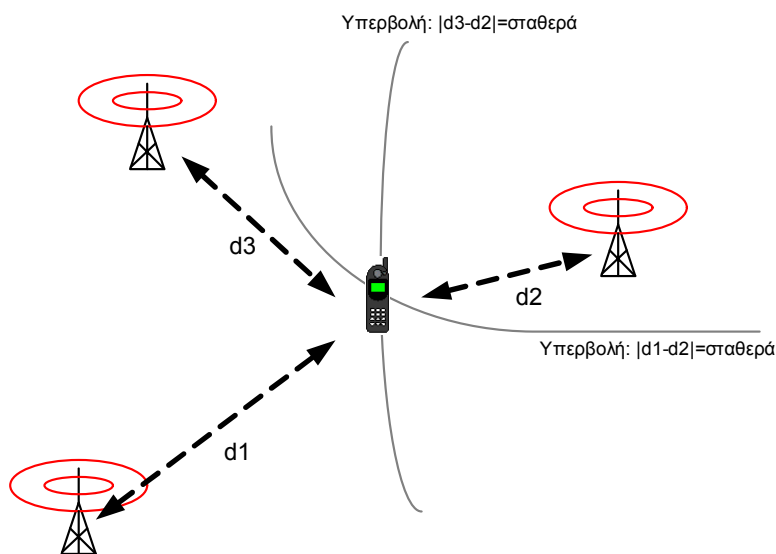
Σ' αυτή τη μέθοδο, παρατηρούνται οι διαφορές των χρόνων διάδοσης σε αντίθεση με την TOA μέθοδο όπου έχουμε απόλυτους χρόνους. Το πλεονέκτημα είναι ότι με αυτόν τον τρόπο καταφέρνουμε να απαλείψουμε την άγνωστη διαφορά χρονισμού t_{off} . Για παράδειγμα ας θεωρήσουμε ότι:

$$d_1 = c \cdot t_1 + c \cdot t_{\text{off}} \quad \text{και} \quad d_2 = c \cdot t_2 + c \cdot t_{\text{off}}$$

Έχουμε :

$$d_1 - d_2 = c \cdot (t_1 - t_2)$$

που είναι ανεξάρτητο του t_{off} .



ΣΧΗΜΑ 1.5

Υπάρχουν δύο παραλλαγές της μεθόδου "Διαφοράς Χρόνου Άφιξης":

- **TDOA** : Οι Σταθμοί Βάσης (ο καθένας ξεχωριστά) μετρούν τους χρόνους άφιξης ενός καταιγισμού από το Κ/Τ.
- **E-OTD** : Το Κ/Τ μετρά τις χρονικές διαφορές μεταξύ σημάτων από διάφορους συγχρονισμένους Σταθμούς Βάσης. (*Mobile-Assisted μέθοδος*)

Ανεξάρτητα από αν έχουμε TDOA ή E-OTD, κάθε μέτρηση χρονικής διαφοράς καθορίζει μία υπερβολή όπως φαίνεται και στο Σχήμα 5. Τρεις ανεξάρτητες υπερβολές έχουν πάντα μόνο ένα κοινό σημείο, το οποίο και θα είναι η γεωγραφική θέση του Κ/Τ. Για να έχουμε τρεις ανεξάρτητες υπερβολές, χρειαζόμαστε μετρήσεις από τέσσερις Σταθμούς Βάσης.

Αν όπως στο Σχήμα 1.5 χρησιμοποιήσουμε τρεις Σταθμούς Βάσης, τότε οι δύο ανεξάρτητες υπερβολές μπορεί να τέμνονται σε περισσότερα από ένα σημείο (συνήθως σε δύο).

Τα Specifications του GSM υποστηρίζουν τις απαιτούμενες μετρήσεις για την μέθοδο TDOA. Πρόκειται για τις παρατηρούμενες χρονικές διαφορές (Observed Time Differences, OTDs). Οι μετρήσεις OTD έχουν το πλεονέκτημα ότι δεν επιβάλλουν την διακοπή (handover) προκειμένου να γίνουν μετρήσεις από πολλαπλούς Σταθμούς Βάσης (αντίθετα με τις "Timing Advance" (TA) που χρησιμοποιούνται στην TOA). Παρόλα αυτά η ακρίβειά τους είναι μόνο 554m (διακριτικότητα bit) και ο απαραίτητος συγχρονισμός των Σταθμών Βάσης δεν είναι εγγυημένος.

Στην μέθοδο E-OTD λύνεται το πρόβλημα, αφού ο συγχρονισμός πετυχαίνεται εγκαθιστώντας ειδικές συσκευές στις θέσεις των Σταθμών Βάσης, οι οποίες μετρούν τις διαφορές χρονισμού μεταξύ των

Σ/Β. Αυτές οι πραγματικές χρονικές διαφορές (RTDs) στέλνονται μαζί με τις OTDs που μετρώνται από το Κ/Τ στο Mobile Location Center (MLC) το οποίο πραγματοποιεί τον υπολογισμό της Γεωγραφικής Θέσης.

Βλέπουμε λοιπόν ότι πρόκειται για μια μέθοδο η οποία απαιτεί την εισαγωγή καινούργιων μονάδων τόσο στο Δίκτυο (SW/HW) όσο και στο Κ/Τ (SW).

Πλεονεκτήματα Μεθόδου:

- Πολύ καλή ακρίβεια σε σχέση με τις προηγούμενες μεθόδους
- Ο χρόνος που απαιτείται για την εξαγωγή αποτελεσμάτων είναι ανεκτός.

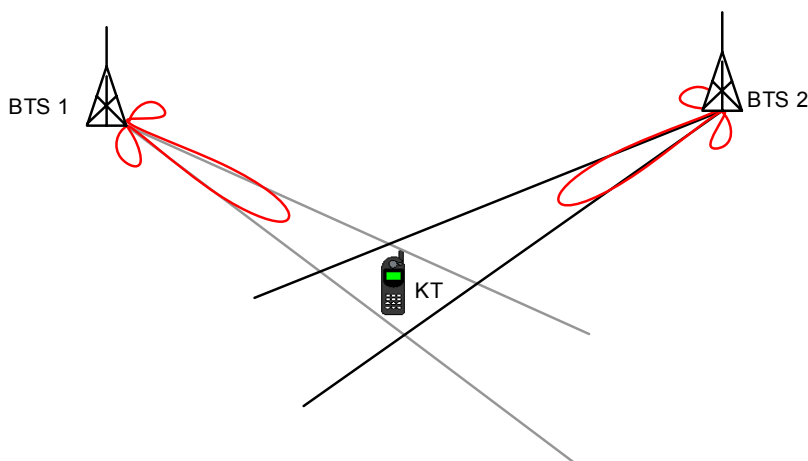
Μειονεκτήματα Μεθόδου:

- Απαιτεί την προσθήκη νέων μονάδων στο Δίκτυο και την προσαρμογή του Software των Κ/Τ.
- Εισάγεται σφάλμα όταν υπάρχει πολυδιαδρομική διάδοση (multipath propagation) και όταν δεν υπάρχει LOS συνθήκη για μερικούς Σταθμούς Βάσης.

2.4.6. ΓΩΝΙΑ ΛΗΨΗΣ (Angle Of Arrival, AOA)

Η μέθοδος Γωνίας Λήψης απαιτεί την εγκατάσταση εξειδικευμένων διατάξεων κεραιών στους Σταθμούς Βάσης, οι οποίες θα πρέπει να είναι πολύ κατευθυντικές και επίσης με ικανότητα στροφής του λοβού ακτινοβολίας ανάλογα με την θέση του κινητού. Με λίγα λόγια η μέθοδος προϋποθέτει την ύπαρξη "έξυπνων κεραιών" (Smart Antennas), μιας τεχνολογίας η οποία αναπτύσσεται στα πλαίσια των Κινητών Τρίτης Γενιάς (3G).

Θεωρώντας δεδομένη την ύπαρξη τέτοιου τύπου κεραιών, ο εντοπισμός του Κ/Τ πραγματοποιείται εύκολα μέσω των γωνιών μισής ισχύος Δ_{3dB} (η γωνία που σχηματίζουν οι διευθύνσεις εκατέρωθεν της διεύθυνσης μεγίστου για τις οποίες η ένταση ακτινοβολίας είναι η μισή της μέγιστης τιμής).



ΣΧΗΜΑ 1. 6

Όπως παρατηρούμε και από το Σχήμα 1.6, για τον εντοπισμό απαιτούνται μόνο δύο Σταθμοί Βάσης. Η περιοχή εντοπισμού του Κ/Τ περικλείεται μέσα στον κλειστό χώρο που ορίζουν οι διευθύνσεις μισής ισχύος των κύριων λοβών του κάθε Σ/Β.

Πλεονεκτήματα Μεθόδου:

- Ακρίβεια παραπλήσια των μεθόδων TDOA και E-OTD
- Ο χρόνος που απαιτείται για την εξαγωγή αποτελεσμάτων είναι ελάχιστος.
- Απαιτεί μόνο δύο Σταθμούς Βάσης για τον προσδιορισμό θέσης.
- Δεν χρειάζεται κάποια μετατροπή στα Κ/Τ.

Μειονεκτήματα Μεθόδου:

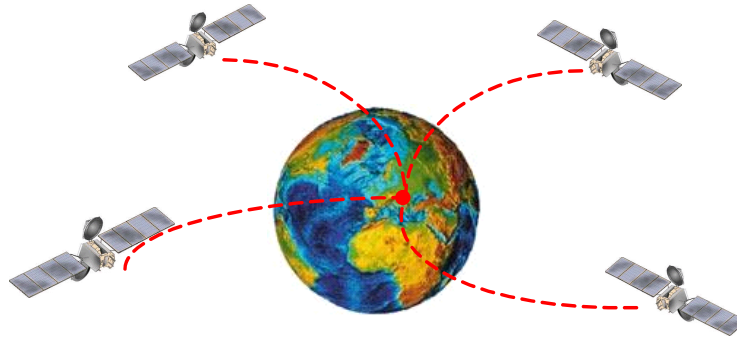
- Απαιτεί την εισαγωγή στο δίκτυο πολύ ακριβών Διατάξεων Κεραιών Λήψης.
- Δεν εγγυάται την σωστή λειτουργία σε NLOS συνθήκες

Μπορούμε λοιπόν να συμπεράνουμε ότι η συγκεκριμένη μέθοδος μπορεί να χρησιμοποιηθεί μόνο συμπληρωματικά και κυρίως σε αγροτικές και ημιαστικές περιοχές.

2.5. ΜΕΘΟΔΟΙ ΒΑΣΙΣΜΕΝΕΣ ΣΤΟ Κ/Τ (MOBILE BASED)

2.5.1. ΔΟΡΥΦΟΡΙΚΟΣ ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΘΕΣΗΣ (GPS)

Το Global Positioning System (GPS) είναι ένα σύστημα Δορυφορικής Ραδιοπλοήγησης, το οποίο λειτουργεί υπό την αιγίδα του Υπουργείου Άμυνας των ΗΠΑ. Το GPS μπορεί και παρέχει ακριβή, αξιόπιστο, συνεχή και παγκόσμιο εντοπισμό θέσης.



Στοιχεία που αποτελούν το GPS :

- 24 Δορυφόροι σε έξι ελλειπτικές τροχιές ύψους 20,2 km πάνω από την επιφάνεια της γης με κλίση κοινή, 55 μοίρες, δωδεκάωρης περιόδου. Το σύστημα είναι έτσι σχεδιασμένο, ώστε έξι τουλάχιστον δορυφόροι να είναι συνεχώς ορατοί από οποιοδήποτε σημείο της γης.
- 1 κύριος σταθμός ελέγχου που βρίσκεται στο Colorado Springs των ΗΠΑ, και πέντε μικρότεροι σταθμοί σε άλλα σημεία, που είναι υπεύθυνοι για την σωστή και απρόσκοπτη λειτουργία του συστήματος.
- Εκατομμύρια δέκτες που χρησιμοποιούνται από τον Στρατό, άλλες Υπηρεσίες, Εταιρείες, Ναυτιλιακούς Οργανισμούς, αλλά και ιδιώτες.

Η τεχνολογία του GPS είναι δεδομένη και επομένως για την χρησιμοποίησή του ως μέθοδο εντοπισμού Κ/Τ, το μόνο που απαιτείται είναι η ενσωμάτωση ενός δέκτη GPS στο Κινητό Τερματικό.

Πλεονεκτήματα Μεθόδου:

- Πάρα πολύ μεγάλη ακρίβεια (3-50m)
- Δεν απαιτείται καμιά μετατροπή στο Δίκτυο

Μειονεκτήματα Μεθόδου:

- Αναγκαία η ύπαρξη δέκτη GPS σε κάθε Κ/Τ, γεγονός που σημαίνει μεγάλο κόστος, αυξημένο βάρος και υψηλή κατανάλωση ισχύος.
- Λειτουργεί μόνο σε συνθήκες οπτικής επαφής, όχι σε κλειστούς/ στεγασμένους χώρους
- Παρουσιάζει μεγάλη καθυστέρηση όταν επιχειρείται ο εντοπισμός θέσης (1~2min).

2.5.2. ΔΙΑΦΟΡΙΚΟ GPS (DGPS)

Η μέθοδος αυτή βελτιώνει την ακρίβεια του απλού GPS. Για τον σκοπό αυτό χρησιμοποιείται ένας σταθερός δέκτης αναφοράς (Reference Receiver), ο οποίος μπορεί και διορθώνει σφάλματα πόλωσης και κοινού ρυθμού σε κινητούς δέκτες οι οποίοι βρίσκονται μέσα σε ακτίνα 100 km.

Ενώ παρέχεται σημαντική βελτίωση στην ακρίβεια, εντούτοις δεν επιδρά στις άλλες σημαντικές παραμέτρους που παρουσιάζουν πρόβλημα, όπως ο χρόνος απόκρισης, οι NLOS συνθήκες και η κατανάλωση ισχύος. Για τον σκοπό αυτό χρησιμοποιείται η παρακάτω μέθοδος.

2.5.3. ΥΠΟΒΟΗΘΟΥΜΕΝΟ GPS (A-GPS)

Η μέθοδος βασίζεται στην υποβοήθηση του χρήστη στον υπολογισμό της θέσης από το δέκτη GPS που ενσωματώνεται στο κινητό. Ο τελικός υπολογισμός της θέσης γίνεται είτε στο K/T είτε στο Mobile Location Center (MLC) του δικτύου.

Η πιο συνήθης περίπτωση περιλαμβάνει την εγκατάσταση ηλεκτρονικών διατάξεων GPS στο δέκτη, με ταυτόχρονη εγκατάσταση κατάλληλης υποδομής στο Δίκτυο.

Ένας ή περισσότεροι στατικοί δέκτες GPS κατανέμονται ομοιόμορφα στη γεωγραφική περιοχή κάλυψης του Δικτύου. Αυτοί οι δέκτες συλλέγουν δεδομένα σχετικά με τον αριθμό των ορατών Δορυφόρων, την ώρα του Συστήματος, την ολίσθηση συχνότητας κ.α. Αυτά τα δεδομένα αποστέλλονται στα K/T, τα βοηθούν να κλειδώσουν γρηγορότερα στους δορυφόρους που απαιτούνται για τον εντοπισμό τους με αποτέλεσμα να μειώνεται δραματικά ο χρόνος απόκρισης (< 30s). Επίσης μειώνεται η ελάχιστη στάθμη ισχύος λήψης, γεγονός που επιτρέπει εντοπισμό και σε NLOS συνθήκες.

2.6. ΣΥΓΚΕΝΤΡΩΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΜΕΘΟΔΩΝ

Αφού έχουμε αναλύσει όλες τις βασικές μεθόδους Προσδιορισμού Γεωγραφικής Θέσης, προχωράμε σε μια συγκεντρωτική παρουσίαση με τη βοήθεια του Πίνακα IV. Σ' αυτόν τον πίνακα παρουσιάζονται όλες οι προαναφερθείσες μέθοδοι σε συνδυασμό με τα κρίσιμα χαρακτηριστικά τους. Αυτά τα χαρακτηριστικά είναι:

- ✓ Επίδραση στο Δίκτυο
- ✓ Επίδραση στο K/T
- ✓ Ακρίβεια σε m*
- ✓ Παράγοντες που επηρεάζουν την ακρίβεια
- ✓ Χρόνος υπολογισμού θέσης*
- ✓ Πολυπλοκότητα Μεθόδου
- ✓ Σημαντικότερο Πλεονέκτημα
- ✓ Σημαντικότερο Μειονέκτημα

| ΜΕΘΟΔΟΣ | Πολυπλοκότητα | Αλλαγές Στο Δίκτυο | Αλλαγές Στο ΚΤ | Ακρίβεια | Παράγοντες που Επηρεάζουν την Ακρίβεια | Απαιτούμενος Χρόνος | Σημαντικότερο Πλεονέκτημα | Σημαντικότερο Μειονέκτημα |
|-------------------------|---|---|--|------------------------|--|--|--|---|
| CELL-ID | Ελάχιστη - Απαιτείται μόνο ένας Σταθμός Βάσης | Καμία - Η πληροφορία είναι ήδη διαθέσιμη | Καμία | 500m - 20km+ | Μέγεθος Κυψέλης Multipath Propagation | Sub Seconds | Γρήγορη και εύκολη Εφαρμογή | Πολύ μικρή Ακρίβεια |
| TA, Enhanced CID | Ελάχιστη | Καμία - Η πληροφορία είναι ήδη διαθέσιμη | Καμία | Δακτύλιος 500m - 20km+ | Συνθήκες NLOS | Sub Seconds | Γρήγορη και εύκολη Εφαρμογή | Η ελάχιστη απόσταση που υπολογίζεται δεν είναι πάντα σωστή |
| SIGNAL STRENGTH | Μικρή | Μικρή (Ανάλογα με τον Αλγόριθμο) Η πληροφορία είναι ήδη διαθέσιμη | Καμία | 200m- 1500m+ | 1) Μοντέλο Ραδιοακάλυψης που χρησιμοποιείται 2) Shadowing 3) Multipath | Sub Seconds/ Seconds (Εξαρτάται από τον αλγόριθμο) | Εύκολη Εφαρμογή με ικανοποιητική Ακρίβεια | Μεμονωμένα Φαινόμενα Multipath και Swadowing δεν προβλέπονται από τα μοντέλα Ραδιοακάλυψης |
| TOA | Μικρή | Μέτρια - Επιπλέον Μονάδες Συγχρονισμού | Καμία | 200m- 1000m+ | 1) Multipath 2) Αγνωστο toff 3) Συνθήκες NLOS | ~ 5sec | Ικανοποιητική Ακρίβεια | Υπάρχει πάντα ένα σφάλμα toff. Απαιτεί Συγχρονισμό |
| TDOA | Μικρή/Μέτρια | Μέτρια - Επιπλέον Μονάδες Συγχρονισμού | Καμία | 40m- 500m+ | 1) Multipath 2) Συνθήκες NLOS | ~ 5sec | Καλή Ακρίβεια | Απαιτεί Συγχρονισμό των Σ.Β |
| E-OTD | Μικρή/Μέτρια | Μέτρια - Επιπλέον Μονάδες Συγχρονισμού | Μικρή/Μέτρια Λογισμικό (SW) και μήμημη | 40m- 500m+ | 1) Multipath 2) Συνθήκες NLOS | ~ 5sec | Καλή Ακρίβεια | Απαιτεί Συγχρονισμό των Σ.Β και νέο Software στα ΚΤ |
| AOA | Μέτρια | Μεγάλη - Απαιτούνται πολύ ακριβείς Στοιχειοκεραιών | Καμία | 50m- 150m+ | 1) Συνθήκες NLOS 2) Γωνία μισής Ισχύος κύριου Λοβού | Sub Seconds | Χρηάζονται μόνο 2 Σταθμοί Βάσης για τον προσδιορισμό της θέσης | Απαιτεί πολύ ακριβές Στοιχειοκεραιές (Smart Antennas) |
| GPS | Μικρή | Καμία | Πολύ Μεγάλη Ενσωμάτωση Δέκτη GPS | 5m- 100m+ | 1) Συνθήκες NLOS 2) Σφάλματα Πόλωσης | 1min ~ 2min | Πολύ καλή ακρίβεια | Η ύπαρξη δέκτη GPS στο ΚΤ προκαλεί αύξηση του βάρους, του μεγέθους, της τιμής και της κατανάλωσης ισχύος της συσκευής |
| DGPS | Μικρή | Μικρή - Δέκτης Αναφοράς | Πολύ Μεγάλη Ενσωμάτωση Δέκτη GPS | 3m- 50m+ | Συνθήκες NLOS | 1min ~ 2min | Ακόμα Καλύτερη ακρίβεια από το GPS | Η ύπαρξη δέκτη GPS στο ΚΤ προκαλεί αύξηση του βάρους, του μεγέθους, της τιμής και της κατανάλωσης ισχύος της συσκευής |
| A-GPS | Μικρή/Μέτρια | Μικρή - Μετάδοση βοηθητικών δεδομένων | Πολύ Μεγάλη Ενσωμάτωση Δέκτη GPS | 3m- 50m+ | Ελάχιστοι | 5 ~ 30sec | Δυνατότητα λειτουργίας και σε NLOS συνθήκες - Μειωμένη κατανάλωση Ισχύος | Η ύπαρξη δέκτη GPS στο ΚΤ προκαλεί αύξηση του βάρους, του μεγέθους, της τιμής και της κατανάλωσης ισχύος της συσκευής |

ΠΙΝΑΚΑΣ IV



ΚΕΦΑΛΑΙΟ 3

ΥΠΑΡΧΟΥΣΑ ΤΕΧΝΟΛΟΓΙΑ



3. ΥΠΑΡΧΟΥΣΑ ΤΕΧΝΟΛΟΓΙΑ

3.1. ΠΡΟΛΟΓΟΣ

Όπως έχουμε ήδη πει, οι υπηρεσίες θέσης (LBS) είναι ένας καινούργιος τομέας ο οποίος εξελίσσεται συνεχώς. Πολλές εταιρίες έχουν μπει στο χώρο δυναμικά και τα προϊόντα τους αναπτύσσονται και βελτιώνονται με τον καιρό.

Στις επόμενες παραγράφους λοιπόν, θα επιχειρήσουμε να παρουσιάσουμε τις ήδη υπάρχουσες τεχνολογίες που παρέχουν Location Based Services. Πρόκειται για τις πιο γνωστές τεχνολογίες οι οποίες είτε εφαρμόζονται ήδη στην αγορά, είτε βρίσκονται σε δοκιμαστικό στάδιο και πρόκειται να κυκλοφορήσουν σύντομα.

3.2. CELLPOINT

www.cellpt.com

Η CellPoint παρέχει ένα άμεσα εμπορικά εφαρμόσιμο πακέτο εντοπισμού θέσης. Η τεχνολογία της CellPoint συνεργάζεται με τα τυπικά GSM τερματικά καθώς και με κινητά τηλέφωνα που υποστηρίζουν WAP. Δεν απαιτεί κάποια μετατροπή στα ψηφιακά δίκτυα ούτε επιπρόσθετα στρώματα, ενώ μπορεί να ελεγχθεί από μία κεντρική απομακρυσμένη τοποθεσία.

Είναι μία Software Based Solution, που χρησιμοποιεί το Sim Toolkit του κινητού και μία σύνδεση στο Internet. Η μέθοδος εντοπισμού βασίζεται κυρίως στο cell global identity (Cell-ID), στο timing advance (TA) και στο NMR, ενώ παρέχει μελλοντική υποστήριξη και για A-GPS, E-OTD και TDOA. Η τεχνολογία αυτή επιτρέπει εντοπισμό του κινητού μέσα σε κλειστούς χώρους (Γκαράζ, Στέγαστρα, εντός Κτηρίων), ακόμα και όταν το κινητό βρίσκεται μέσα στην τσέπη ή στον χαρτοφύλακα, περιπτώσεις όπου το GPS δεν λειτουργεί.

Πρόκειται για ένα πλήρως προσαρμόσιμο πακέτο που παρέχει αρχικά τις βασικές υπηρεσίες (με χρονικό διάστημα εφαρμογής 3 μήνες) και είναι δυνατή η αναβάθμιση και προσαρμογή (συνδυασμός τεχνολογιών εντοπισμού) ανάλογα με τις ανάγκες του παροχέα κινητών επικοινωνιών.

Επίσης δίνει την δυνατότητα και σε τρίτες Εταιρίες ανάπτυξης λογισμικού, να φτιάξουν δικές τους εφαρμογές μέσω του MLP (Mobile Location Protocol support).

Χρησιμοποιώντας την τεχνολογία εντοπισμού της CellPoint, η Σουηδική εταιρία Tele2Mobil προσφέρει αυτή τη στιγμή στους κατόχους κινητών της χώρας της ανάλογες εφαρμογές και υπηρεσίες στο Internet. Η υπηρεσία ονομάζεται "Tele2Mobil Position" και δίνει την δυνατότητα σε οργανισμούς μεταφορών, ασφαλείας, υπηρεσιών και πωλήσεων να αυξήσουν τα κέρδη τους και τον αριθμό των πελατών τους με το να καθοδηγούν τα οχήματα και το προσωπικό τους πιο αποτελεσματικά.

Άλλες εταιρίες που χρησιμοποιούν τη συγκεκριμένη τεχνολογία είναι:

1) EuroTel Praha Ltd

3) MTN (South Africa)

2) France Telecom

4) E-Plus Mobilfunk (Germany)

3.3. SNAPTRACK

www.snaptrack.com

Πρόκειται για ένα σύστημα βασισμένο στο GPS που απαιτεί έναν εξυπηρετητή από την πλευρά του δικτύου καθώς και μετατροπή στις τερματικές συσκευές. Η μέθοδος αυτή πρόσφατα αγοράστηκε από την Qualcomm.

Η τεχνολογία Enhanced Global Positioning System (EGPS) μπορεί να προσφέρει πολλά στην δημόσια ασφάλεια των συνδρομητών με το να εντοπίζονται εκτελώντας τηλέφωνα ανάγκης. Επίσης μπορεί να ενεργοποιήσει ένα μεγάλο αριθμό νέων υπηρεσιών, όπως βελτιωμένη οδική βοήθεια, καθοδήγηση στον επιθυμητό προορισμό, ευκολότερη μίσθωση Ταξί, καθοδήγηση εταιρικών οχημάτων και καινούργιες υπηρεσίες καταλόγου.

Για την δοκιμή αυτής της νέας τεχνολογίας έχει δημιουργηθεί ένα test group που αποτελείται από τις παρακάτω εταιρείες:

- 1) Vodafone AirTouch Communications PLC (UK and US)
- 2) BellSouth Mobility DCS (US)
- 3) BT Cellnet (UK)
- 4) Esat Digifone (Ireland)
- 5) France Telecom (France)
- 6) Omnitel Pronto Italia (Italy)
- 7) T-Mobil (Germany)
- 8) Telecel (Portugal)
- 9) Telefonica (Spain)

Η Motorola έχει αναλάβει να κατασκευάσει τα πρωτότυπα κινητά στα πλαίσια αυτής της δοκιμής, ενώ η SignalSoft θα παρέχει το κατάλληλο software για τις Location Based Services. Έτσι θα δημιουργηθεί ένα πλήρες άκρη-σε-άκρη πειραματικό περιβάλλον.

Το SnapTrack επιφέρει σημαντική βελτίωση σε σχέση με την απόδοση του συμβατικού GPS, χρησιμοποιώντας ειδικούς αλγόριθμους μέσα στο κινητό καθώς και με τη βοήθεια ενός σύγχρονου server software που τρέχει στο ασύρματο δίκτυο. Έτσι ενώ οι συμβατικοί δέκτες GPS χρειάζονται μερικά λεπτά για να επεξεργαστούν τα δεδομένα και να δώσουν κάποιο αποτέλεσμα, το SnapTrack μπορεί και εντοπίζει τους καλούντες μέσα σε λίγα δευτερόλεπτα.

Η NTT DoCoMo στην Ιαπωνία εφαρμόζει ήδη την τεχνολογία εμπορικά (από το 2000).

3.4. Ericsson Mobile Positioning System (MPS)

www.ericsson.se

Το σύστημα αυτό της Ericsson δεν απαιτεί μετατροπές στα GSM τηλέφωνα, πρόκειται για μία server-based λύση και επιτρέπει την εφαρμογή location services σε οποιοδήποτε GSM δίκτυο που αποτελείται από Ericsson Switching Systems. Η ERICSSON παρέχει ειδικό HLR, MSC και BSC software καθώς και το απαιτούμενο πρόσθετο Hardware.

Η καρδιά του MPS είναι το Mobile Location Centre (MLC) που επιτρέπει στις εφαρμογές να έχουν πρόσβαση στις πληροφορίες τοποθεσίας των GSM συσκευών. Επίσης είναι διαθέσιμο και ένα Application Programming Interface (API) που επιτρέπει την ανάπτυξη ανεξάρτητων εφαρμογών. Το MLC είναι επίσης υπεύθυνο και για την προστασία των εγγεγραμμένων μελών και δίνει την δυνατότητα στους χρήστες κινητών τηλεφώνων να επιλέγουν αν θέλουν ή όχι να εντοπίζονται οι συσκευές τους. Οι εφαρμογές που θα υποστηρίζονται από το σύστημα είναι πολλές και εκτός από τις υπηρεσίες έκτακτης ανάγκης, την δρομολόγηση στόλου οχημάτων και τον εντοπισμό κλεμμένου αυτοκινήτου, το σύστημα θα προσφέρει και άλλες εμπορικές υπηρεσίες όπως για παράδειγμα:

- 1) "Where am I?" καθοδήγηση,
- 2) Οδική Βοήθεια,
- 3) Τοπικά νέα,
- 4) Πληροφορίες και πρόβλεψη καιρού
- 5) Χρυσό Οδηγό ανάλογα με την τοποθεσία.

Οι μέθοδοι εντοπισμού που χρησιμοποιούνται από την ERICSSON είναι τέσσερις:

- | | | |
|----------|---|-----------------|
| a) CGI | } | Network-Based |
| b) TA | | |
| c) A-GPS | } | Mobile-Assisted |
| d) E-OTD | | |

Οι παροχείς που χρησιμοποιούν την τεχνολογία αυτή είναι:

- 1) Eesti Mobiiltelefon (Estonia)
- 2) Telia (Sweden)
- 3) FarEasTone (Taiwan)
- 4) Telenor Mobil (Norway)

3.5. CPS Cursor

www.cursor-system.com

Η Cambridge Positioning Systems (CPS) έχει δημιουργήσει ένα σύστημα που ονομάζεται Cursor το οποίο είναι ένα E-OTD σύστημα με ελάχιστη ακρίβεια 100m. Αυτό χρησιμοποιεί τα δικά του ξεχωριστά σημεία αναφοράς τα οποία βρίσκονται στην ίδια τοποθεσία με τα Base Stations.

Το Cursor προσφέρει:

- Μέση ακρίβεια 50m στα δίκτυα GSM, με ακόμα μεγαλύτερη στα δίκτυα 3G
- Αξιόπιστο και ακριβή εντοπισμό θέσης μέσα σε κτήρια, σε πυκνοκατοικημένες περιοχές και εν κινήσει.
- Ευκολία στην εφαρμογή του συστήματος στο υπάρχων δίκτυο μέσω software πακέτων που παρέχουν μία πλήρως standardized E-OTD λύση.

Η CPS έχει ήδη αδειοδοτήσει μεγάλους κατασκευαστές εξοπλισμού δικτύων και κατασκευαστές κινητών τηλεφώνων να ενσωματώσουν την τεχνολογία Cursor E-OTD, έτσι ώστε να μπορεί στο μέλλον να λειτουργεί ευρέως σε παγκόσμιο επίπεδο.

Παροχείς που χρησιμοποιούν την τεχνολογία:

- 1) SmartTone (Hong Kong)
- 2) Vodafone (UK), Trial

3.6. NOKIA mPosition

www.nokia.com

Αυτή η πρόταση από τη NOKIA είναι μία end-to-end λύση. Η πληροφορία θέσης παρέχεται ως Γεωγραφικό μήκος και πλάτος (x,y συντεταγμένες) στις εφαρμογές που τη ζητούν. Υποστηρίζονται τόσο εφαρμογές εντός ασύρματου δικτύου όσο και εκτός. Η διαδικασία προσδιορισμού μπορεί να ξεκινάει από το K/T, το δίκτυο, ή από μια εξωτερική εφαρμογή.

Υποστηρίζονται διάφορες μέθοδοι εντοπισμού:

- ✓ Cell-ID + Enhanced timing advance + additional BSS information
- ✓ Enhanced Observed Time Difference (E-OTD)
- ✓ Network assisted GPS (A-GPS) μελλοντικά

Η τεχνολογία της NOKIA παρέχει στους Operators τη δυνατότητα να ξεκινήσουν την εφαρμογή Υπηρεσιών Θέσης με ελάχιστη επένδυση.

Οι παροχείς που χρησιμοποιούν το mPosition είναι:

- 1) ONE (Austria)
- 2) Fujian Mobile Communications Company (FMCC) (China)

3.7. TRUEPOSITION Wireless Location System

www.trueposition.com

Η TruePosition παρέχει αυτή την ολοκληρωμένη hardware/software λύση, η οποία πραγματοποιεί τον εντοπισμό του Κ/Τ, επιτρέποντας έτσι την εφαρμογή διαφόρων υπηρεσιών θέσης.

Το Wireless Location System (WLS) αποτελείται από τα παρακάτω μέρη:

✓ **Wireless Location Platform**

Το WLP υπολογίζει την γεωγραφική θέση μιας συσκευής με το να συλλέγει και να επεξεργάζεται τα δεδομένα θέσης. Όταν ένα σήμα μεταδίδεται (πχ. εγκαθίσταται μια κλήση), τα Position Determining Equipment (PDE) (ή αλλιώς Location Measuring Units, LMU's) που είναι εγκατεστημένα σε διάφορους Σταθμούς Βάσης, συλλέγουν τις πληροφορίες κλήσης. Ένας Επεξεργαστής Τοποθεσίας ή Serving Mobile Location Center (SMLC) τριγωνοποιεί την θέση του καλούντα χρησιμοποιώντας τους αλγόριθμους TDOA και AOA.

✓ **Wireless Location Gateway**

Πρόκειται για το Interface που είναι υπεύθυνο για την επικοινωνία και την μετάδοση πληροφοριών μεταξύ του δικτύου και των εφαρμογών θέσης εκτός δικτύου (third party applications).

Το WLS έχει το πλεονέκτημα ότι μπορεί να εντοπίσει οποιοδήποτε υπάρχων Κ/Τ, με καλή ακρίβεια ακόμα και σε κλειστούς χώρους και πυκνοκατοικημένες περιοχές. Επίσης προσφέρει μεγάλη χωρητικότητα και μπορεί να προσαρμοστεί μελλοντικά ώστε να υποστηρίζει και νέες μεθόδους εντοπισμού.

3.8. SIGNALSOFT Corporation

www.SignalSoftCorp.com

Ουσιαστικά πρόκειται για μια εταιρεία ανάπτυξης λογισμικού, η οποία σε συνεργασία με στις εταιρίες (Applications Platform & Integrators, Location Technology, Wireless Network Platform, Content & Applications) μπορεί και παρέχει ολοκληρωμένες λύσεις LBS, δηλαδή μέθοδο εντοπισμού, απαιτούμενο Software και Hardware, εφαρμογές και υπηρεσίες.

Το μεγαλύτερο πλεονέκτημα είναι ότι ο Location Manager (Middleware) στις εταιρίες υποστηρίζει τις γνωστές μεθόδους και τις επιπλέον.

Αυτές οι μέθοδοι είναι:

- 1) TOA/TA
- 2) TDOA
- 3) AOA
- 4) A-GPS
- 5) E-OTD
- 6) Cell-ID/TA
- 7) Cell-ID/NMTC
- 8) MAP ATI

Location Manager

Ο Location Manager είναι ο software πυρήνας στις τεχνολογίες στις SignalSoft. Πρόκειται για την διέξοδο υπολογισμού στις θέσης του συνδρομητή. Οι διαθέσιμες πληροφορίες συνδυάζονται έτσι ώστε να υπολογιστεί τελικά με τη μεγαλύτερη δυνατή ακρίβεια το Γεωγραφικό μήκος και πλάτος που αντιστοιχεί στην θέση του Κ/Τ.

Location Studio

Πρόκειται για ένα λογισμικό που τρέχει μέσα στο Δίκτυο και είναι υπεύθυνο για την ασφαλή διασύνδεση μεταξύ στις location based εφαρμογής και του συνδρομητή. Έτσι διασφαλίζεται η ιδιωτικότητα (privacy), το σωστό authentication του πελάτη και παρέχεται η δυνατότητα για προσαρμοσμένη τιμολόγηση.

Οι εφαρμογές που παρέχει η SignalSoft είναι οι ακόλουθες:

- ***IN location:***

Συνδέει στις συνδρομητές απευθείας με τον πλησιέστερο εμπορικό αντιπρόσωπο. Πχ. η πληκτρολόγηση του μηνύματος *CAR συνδέει τον καλούντα με το πλησιέστερο στην περιοχή Σέρβις Αυτοκινήτων.

- ***FRIEND finder***

Επιτρέπει στις συνδρομητές να εντοπίζουν ανά πάσα στιγμή στις φίλους/ συναδέλφους στις.

- **SAFETY first**

Παρέχει την πληροφορία τοποθεσίας όταν γίνεται κλήση για έκτακτη ανάγκη.

Οι Παροχείς υπηρεσιών κινητών τηλεπικοινωνιών που χρησιμοποιούν αυτή τη τεχνολογία είναι:

- 1) Hutchison Telecommunications (Hong Kong)
- 2) Libertel (Netherlands)
- 3) VodaCom (South Africa)
- 4) Orange (Switzerland)
- 5) DiAx (Switzerland)
- 6) PTC-Era (Poland)
- 7) Πολλές εταιρείες στις ΗΠΑ (Sprint, SBC, US West, Bell Atlantic Mobile κλπ)

3.9. BT CELLNET

www.btcellnet.co.uk

Η BT Cellnet έχει φτιάξει ένα E-OTC σύστημα στο οποίο οι Base Stations συγχρονίζονται ώστε να γίνουν τα σημεία αναφοράς. Το σύστημα έχει δοκιμαστεί για δύο χρόνια στην Βορειοανατολική Αγγλία. Δεν υπάρχουν περαιτέρω πληροφορίες για το εν λόγω σύστημα.

Επίσης, αξίζει να αναφερθεί ότι η BT Cellnet ήταν η πρώτη εταιρεία κινητής τηλεφωνίας στην Αγγλία που εισήγαγε Location Based Services, χρησιμοποιώντας ένα πολύ απλό σύστημα Cell-ID. Από τον Σεπτέμβριο του 2000 προσφέρει στους συνδρομητές της φωνητικές υπηρεσίες θέσης, χρησιμοποιώντας μια database με πάνω από 130.000 καταχωρήσεις (τράπεζες, εστιατόρια, διασκέδαση κλπ.). Η υπηρεσία ονομάζεται FINDme και εντοπίζει τους χρήστες σε ακτίνα 100m για αστικές περιοχές και 15km για αγροτικές.

3.10. ΠΙΝΑΚΕΣ ΥΠΗΡΕΣΙΩΝ-ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

Παρακάτω παρουσιάζουμε ενδεικτικά (Πίνακας V) μερικές ήδη υπάρχουσες υπηρεσίες που προσφέρονται από διάφορους Operators. Όπως γίνεται φανερό, οι υπηρεσίες αυτές προσφέρονται μέσω διαφόρων τρόπων (WAP, SMS, Voice) ενώ στη δεξιά στήλη του πίνακα βλέπουμε και τη χρέωση σε USD.

| Country | Network Operator | Application | Service | Service Cost |
|----------|------------------|----------------------------|--------------------|------------------------------------|
| Austria | Tele.ring | Information Services | Local Info | Normal WAP tariff |
| Canada | AT&T Wireless | Emergency Services (E-911) | Emergency Calls | 0.60 per month |
| France | Bouygues | Information Services | Where's my nearest | 0.28 per SMS message (transaction) |
| Germany | Viag Interkom | Safety | Phone Finder | 0.18 per min WAP |
| Norway | Telenor | Game | Treasure Hunt Game | 3.00 per game |
| Norway | NetCom | Buddy Finder | Buddy- Service | 0.85 per request |
| Portugal | Telecel | Team Manager | Phone tracker | 0.09 per location |
| Portugal | Telecel | Fleet Services | Phone tracker | 0.09 per location |
| UK | Vodafone | Traffic Info | Roadwatch | 0.72 per min (min) |
| UK | ZagMe | M-Coupons | Localized coupons | 0.15 per coupon SMS (transaction) |
| USA | AT&T Wireless | Emergency Services (E-911) | Emergency calls | 1.25 per month |

ΠΙΝΑΚΑΣ V

Κλείνοντας, παρουσιάζουμε στον Πίνακα VI τα συγκεντρωτικά στοιχεία που προκύπτουν από το Κεφάλαιο 3. Αναφέρουμε δηλαδή ποια τεχνολογία χρησιμοποιεί ο κάθε Operator και επίσης μερικά επιπλέον στοιχεία (όπου είναι δυνατόν).

| ΧΩΠΑ | Network Operator | Location Supplier | Υποστηριζόμενες Location Techniques* | Λεπτομέρειες |
|----------------|------------------------------|-------------------|---|---|
| Austria | ONE | Nokia | Cell-ID, Enhanced TA, BSS info, E-OTD, A-GPS | 7/3/2002 |
| China | FMCC | Nokia | Cell-ID, Enhanced TA, BSS info, E-OTD, A-GPS | mCatch/mPosition |
| Czech Republic | EuroTel | CellPoint | Cell-ID, TA, NMR, A-GPS, E-OTD, TDOA | Resource Manager (Fleet Management Service) |
| Estonia | Eesti MobilTelefon | Ericsson | CGI, TA, A-GPS, E-OTD | MPS first Customer |
| France | France Telecom Mobile | CellPoint | Cell-ID, TA, NMR, A-GPS, E-OTD, TDOA | Resource Manager (Fleet Management Service) |
| France | France Telecom Mobile | SnapTrack | EGPS | STGTG Member and GSM Trials Host |
| Germany | T-Mobil | SnapTrack | EGPS | STGTG Member and GSM Trials Host |
| Germany | E-Plus | CellPoint | Cell-ID, TA, NMR, A-GPS, E-OTD, TDOA | Mobile Location System |
| Hong Kong | Hutchison Telecommunications | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| Hong Kong | SmarTone | CPS | E-OTD | Trial of Cursor and Coverage Services |
| Ireland | Esat Digifone | SnapTrack | EGPS | |
| Italy | Omnitel Pronto Italia | SnapTrack | EGPS | STGTG Member and GSM Trials Host |
| Japan | SECOM | SnapTrack | EGPS | GpsOne launch by 1H 2001 |
| Japan | KDDI | SnapTrack | EGPS | GpsOne launch by 2H 2001 |
| Japan | NTT DoCoMo | SnapTrack | EGPS | January 2000 launch |
| Netherlands | LiberTel | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| Norway | Telenor Mobil | Ericsson | CGI, TA, A-GPS, E-OTD | |
| Poland | PTC-Era | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | First Eastern European mobile operator to invest in |
| Portugal | Telecel | SnapTrack | EGPS | STGTG Member and GSM Trials Host |
| South Africa | MTN | CellPoint | Cell-ID, TA, NMR, A-GPS, E-OTD, TDOA | |
| South Africa | Vodacom | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| Spain | Telefonica Moviles | SnapTrack | EGPS | STGTG Member and GSM Trials Host |
| Sweden | Tele2Mobil | CellPoint | Cell-ID, TA, NMR, A-GPS, E-OTD, TDOA | Resource Manager (Fleet Management Service) |
| Sweden | Telia | Ericsson | CGI, TA, A-GPS, E-OTD | |
| Switzerland | Orange | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| Switzerland | DiAx | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| Taiwan | FarEasTone | Ericsson | CGI, TA, A-GPS, E-OTD | |
| UK | Vodafone | CPS | E-OTD | Trial of CURSOR |
| UK | Vodafone AirTouch | SnapTrack | EGPS | |
| UK | BT Cellnet | SnapTrack | EGPS | |
| USA | Sprint | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| USA | SBC | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| USA | Alltel | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| USA | US West | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| USA | Bell Atlantic Mobile | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |
| USA | PrimeCo | SignalSoft | TOA/TA, TDOA, AOA, AGPS, E-OTD, Cell-ID/TA, MAP ATI | |

* Αναφέρεται στις μεθόδους που υποστηρίζει η συγκεκριμένη τεχνολογία και όχι απαραίτητα στις μεθόδους που χρησιμοποιεί ο κάθε Operator

ΠΙΝΑΚΑΣ VI



ΚΕΦΑΛΑΙΟ 4

ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΕΝΤΟΠΙΣΜΟΥ (CID-TA-RXLEV)



4. ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΕΝΤΟΠΙΣΜΟΥ

4.1. ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο θα περιγράψουμε τον αλγόριθμο εντοπισμού θέσης που εφαρμόσαμε. Ο αλγόριθμος αυτός είναι βασισμένος σε στοιχεία που μπορούν να ανακτηθούν από το δίκτυο (Network Based Solution). Αυτά τα στοιχεία είναι:

☞ Ταυτότητα Κυψέλης (Cell ID):

Ο αριθμός της τρέχουσας κυψέλης. Κάθε BTS σε μια δεδομένη περιοχή έχει ένα μοναδικό αριθμό ταυτότητάς. Γνωρίζοντας τη θέση του BTS, μπορούμε να έχουμε μια προσέγγιση της θέσης του Κινητού Τερματικού (Κ/Τ).

☞ Πρόοδος Συγχρονισμού (TA):

Ο Σταθμός Βάσης (BTS) ενημερώνει το Κινητό Τερματικό πότε πρέπει να αρχίσει τη μετάδοση έτσι ώστε φθάσει στο BTS εγκαίρως. Αυτή η παράμετρος ενημερώνεται κατά τη διάρκεια της επικοινωνίας με το δίκτυο (στέλνοντας / λαμβάνοντας SMS, τοποθετώντας μια κλήση, ζητώντας κάποια υπηρεσία, κλπ....) και κυμαίνεται από 0 έως 63.

☞ Λαμβανόμενη Ισχύς σήματος (RXLEV):

Η δύναμη του σήματος που λαμβάνεται στο κανάλι BCCH. Αυτή η πληροφορία είναι διαθέσιμη για την τρέχουσα Κυψέλη Εξυπηρέτησης (Serving Cell), καθώς επίσης και για τις Γειτονικές Κυψέλες (Neighbour Cell). Ο μέγιστος αριθμός των διαθέσιμων γειτονικών κυψελών είναι 6.

4.2. ΑΠΑΙΤΟΥΜΕΝΟΣ ΕΞΟΠΛΙΣΜΟΣ

Προκειμένου να πραγματοποιηθεί ο αλγόριθμος και να δοκιμαστεί στην πράξη, κρίθηκε αναγκαία η χρησιμοποίηση του ακόλουθου εξοπλισμού:



Φορητός Υπολογιστής Compaq® Armada, Intel® Pentium 4, Λειτουργικό Σύστημα: Microsoft® Windows 2000 Professional, Εγκατεστημένα προγράμματα: Microsoft® Office 2000, Microsoft® Visual Studio 6 Enterprise Edition. Το Visual Studio 6 μας παρείχε το αναγκαίο περιβάλλον προγραμματισμού, μέσω της Visual Basic 6, ενώ το Microsoft® Office 2000 εγκαταστήθηκε λόγω της ανάγκης ύπαρξης μιας βάσης δεδομένων για τους Σταθμούς Βάσης (Microsoft® Access 2000) και ενός μέσου αποθήκευσης των αποτελεσμάτων (Microsoft® Excel). Χρησιμοποιήθηκε φορητός και όχι Desktop υπολογιστής, αφού ήταν αναγκαία η μεταφορά και μετακίνησή του μέσα στο πεδίο μετρήσεων, για να πραγματοποιηθούν τα απαραίτητα field tests προκειμένου να βελτιστοποιηθεί ο αλγόριθμος. Ο υπολογιστής διέθετε και δύο θύρες USB για σύνδεση περιφερειακών συσκευών.

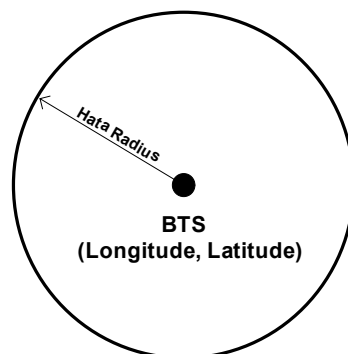
- ☞ Ένα κινητό τηλέφωνο NOKIA® 6150 που συνοδευόταν από σειριακό καλώδιο FBUS για την διασύνδεσή του με τον υπολογιστή. Στο κινητό τοποθετήθηκε μια test sim της Vodafone για την πραγματοποίηση των μετρήσεων. Τα στοιχεία του δικτύου (Cell ID, Timing Advance και RxLEV) εξαγόταν από το κινητό μέσω του Netmonitor της Nokia και εισάγονταν στον υπολογιστή μέσω του καλωδίου FBUS. Οι περαιτέρω υπολογισμοί γίνονταν στον υπολογιστή από το πρόγραμμα εντοπισμού.
- ☞ Μία συσκευή GPS, Magellan® GPS 320, που και αυτή συνοδευόταν από το αντίστοιχο σειριακό καλώδιο για τη διασύνδεση με τον υπολογιστή. Το GPS χρησιμοποιήθηκε για να γνωρίζουμε κάθε φορά την πραγματική θέση (με πολύ μικρή απόκλιση ~10m) του K/T, ώστε να μπορούμε να την συγκρίνουμε με αυτή που υπολόγιζε ο αλγόριθμος εντοπισμού. Το πρόγραμμα έχει τη δυνατότητα να ανακτά την πληροφορία θέσης από το συγκεκριμένο GPS, αλλά και από οποιοδήποτε GPS αρκεί να είναι συμβατό με το NMEA v.1.5/v.2.1 και να υπάρχει το καλώδιο για την σύνδεση με τον υπολογιστή.
- ☞ Δύο μετατροπείς Serial→USB. Αφού ο φορητός υπολογιστής δεν ήταν εφοδιασμένος με σειριακές θύρες, αλλά μόνο με USB, προμηθευτήκαμε δύο τέτοιους μετατροπείς ώστε να γίνει εφικτή η σύνδεση του K/T και του GPS με τον υπολογιστή.

4.3. ΙΣΤΟΡΙΑ ΑΛΓΟΡΙΘΜΟΥ

Κατά τη διάρκεια της εφαρμογής του αλγορίθμου, πολλές διαφορετικές μέθοδοι προτάθηκαν και εξετάστηκαν. Εδώ θα δώσουμε μια συνοπτική περιγραφή όλων των προηγούμενων μεθόδων που μας οδήγησαν στον τελικό αλγόριθμο. Αυτή η παράγραφος είναι μόνο για πληροφοριακούς σκοπούς και μπορεί να παρακαμφθεί.

4.3.1. ΠΡΩΤΑ ΣΤΑΔΙΑ – ΑΛΓΟΡΙΘΜΟΣ ΓΙΑ ΤΟ RXLEV

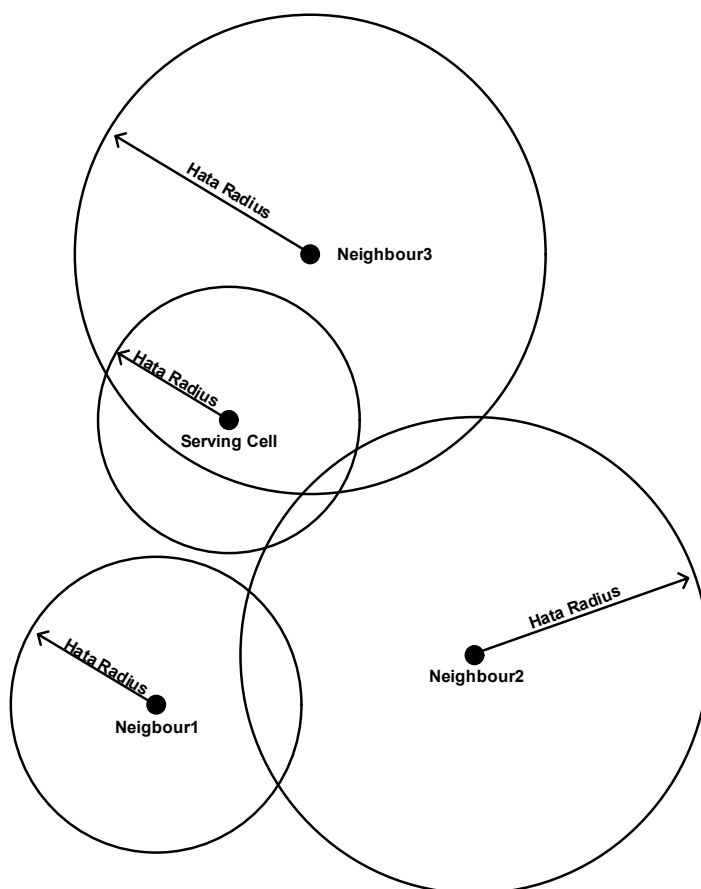
Λαμβάνοντας υπόψη το γεγονός ότι οι άλλες δύο μέθοδοι (TA, Cell ID) είναι περισσότερο τυποποιημένες, αρχίσαμε με την έρευνα της μεθόδου Ισχύος λαμβανόμενου Σήματος (RXLEV). Γνωρίζοντας την λαμβανόμενη Ισχύ των σημάτων (Εξασθένιση σήματος = Εκπεμπόμενη Ισχύς(Tx) – Λαμβανόμενη Ισχύς(Rx)), μπορούμε να υπολογίσουμε την απόσταση που διανύεται από το σήμα με τη βοήθεια του μοντέλου εξασθένισης του HATA, το οποίο έχει ήδη περιγραφεί.



ΣΧΗΜΑ 4.1

Κατά συνέπεια, μπορούμε να θεωρήσουμε ότι έχουμε ένα κύκλο με κέντρο τη γεωγραφική θέση του Σταθμού Βάσης (γεωγραφικό μήκος / γεωγραφικό πλάτος) και ακτίνα την υπολογιζόμενη απόσταση μέσω του HATA (σχήμα 4.1).

Εάν εκτελέσουμε τον ίδιο υπολογισμό για κάθε διαθέσιμο Σταθμό Βάσης (για το Serving Cell καθώς και για όλους τους διαθέσιμους Γείτονες) θα πάρουμε διάφορους κύκλους (μέγιστος αριθμός = 7) από τους οποίους άλλοι θα τέμνονται και άλλοι όχι. Μια τυχαία αναπαράσταση παρουσιάζεται στο σχήμα 4.2. Όπως είναι προφανές, δεν μπορούμε να υπολογίσουμε την θέση του Κ/Τ με τη χρησιμοποίηση των σημείων τομής των κύκλων επειδή σε μερικές περιπτώσεις οι κύκλοι δεν τέμνονται.



ΣΧΗΜΑ 4.2

Μετά από πολυάριθμες δυνατές λύσεις, καταλήξαμε στο συμπέρασμα ότι ο καλύτερος τρόπος είναι να υπολογίσουμε το σημείο όπου *το σύνολο των αποστάσεων από κάθε ακτίνα "HATA Radius" είναι ελάχιστο*.

Πιο συγκεκριμένα, ας υποθέσουμε ότι θέλουμε να υπολογίσουμε το σύνολο των αποστάσεων για ένα σημείο (X_{tp}, Y_{tp}) .

Τα σημεία θέσης των Σταθμών Βάσης είναι:

$$\begin{array}{l}
 \text{Serving Cell } (X_0, Y_0, R_0) \\
 \text{Neighbour1 } (X_1, Y_1, R_1) \\
 \text{Neighbour2 } (X_2, Y_2, R_2) \\
 \dots \\
 \dots\dots \\
 \text{NeighbourN } (X_N, Y_N, R_N)
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{Serving Cell } (X_0, Y_0, R_0) \\ \text{Neighbour1 } (X_1, Y_1, R_1) \\ \text{Neighbour2 } (X_2, Y_2, R_2) \\ \dots \\ \dots\dots \\ \text{NeighbourN } (X_N, Y_N, R_N) \end{array}} \right\}
 \begin{array}{l}
 N \leq 6 \\
 X_N, Y_N : \text{Συντεταγμένες N-οστού BTS} \\
 R_N : \text{"HATA Radius" για το N-οστό BTS}
 \end{array}$$

Η απόσταση του σημείου (X_{tp}, Y_{tp}) από την ακτίνα "HATA Radius" του N-οστού BTS είναι:

$$\text{Distance}(N) = \left| \sqrt{(X_{tp} - X_N)^2 + (Y_{tp} - Y_N)^2} - R_N \right|$$

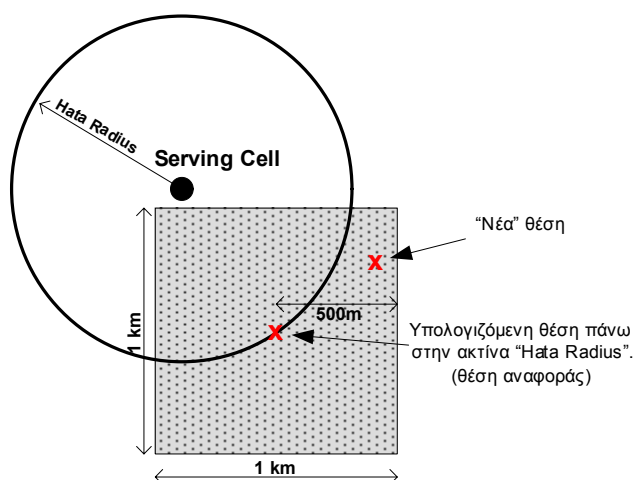
Το άθροισμα όλων των αποστάσεων είναι:

$$\text{TotalDistance} = \sum_N (k_N \cdot \text{Distance}(N))$$

Όπου k_N ένας συντελεστής που χρησιμοποιείται για να δείξει τη σημασία κάθε απόστασης. Αυτός ο συντελεστής δεν θα περιγραφεί τώρα, αλλά αργότερα όταν περιγράψουμε τον τελικό Αλγόριθμο Εντοπισμού Θέσης.

Μια εναλλακτική λύση, είναι να υπολογίζουμε το άθροισμα $[\text{Distance}(N)]^2$, αλλά αυτή η περίπτωση απορρίφθηκε επειδή μετά από δοκιμές ανακαλύψαμε ότι τα αποτελέσματα ήταν χειρότερα ακολουθώντας αυτή τη λύση. Αυτό συμβαίνει επειδή μια λανθασμένη τιμή του RXLev (λόγω φαινομένων πολλαπλής διαδρομής (multipath), ανακλάσεων, κλπ....) θα μπορούσε να οδηγήσει σε μια μεγάλη τιμή του $\text{Distance}(N)$, οπότε θα είχαμε ακόμα μεγαλύτερη τιμή του $[\text{Distance}(N)]^2$. Αυτό θα οδηγούσε τον αλγόριθμο στο να υπολογίσει μια θέση πιο κοντά στην λανθασμένη ακτίνα "HATA Radius".

Στα πρώτα στάδια, ο αλγόριθμος υπολόγιζε την ελάχιστη τιμή του TotalDistance επάνω στην ακτίνα "HATA Radius" της Κυψέλης Εξυπηρέτησης (Serving Cell). Μετά, έχοντας υπολογίσει μια θέση αναφοράς, το πρόγραμμα ανίχνευε μια έκταση 1 km^2 γύρω από αυτήν την θέση μήπως τυχόν βρει μια άλλη καλύτερη θέση (σχήμα 4.3).



ΣΧΗΜΑ 4. 3

Αργότερα συνειδητοποιήσαμε ότι αυτό δεν ήταν απαραίτητο, για δύο λόγους:

- ➔ Ο πρώτος λόγος είναι ότι η πιο αξιόπιστη μέτρηση (σχετικά με το RXLev) είναι αυτή από το Serving Cell. Στην πραγματικότητα, η αληθινή γεωγραφική θέση (που λαμβάνεται από το GPS) ήταν στις περισσότερες περιπτώσεις πολύ κοντά στην ακτίνα HATA του Serving Cell (μέση απόσταση από 30 έως 70 μέτρα).
- ➔ Ο δεύτερος λόγος είναι ότι ο αλγόριθμος έπρεπε να υπολογίσει 10.000 τιμές για το TotalDistance γεγονός που, από απόψεως υπολογιστικής αξίας, ήταν πολύ χρονοβόρο (1-2 sec).

4.3.2. ΤΕΛΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ – (Cell ID, TA, RXLev)

Ύστερα από πολλές δοκιμές, λάθη και διαφορετικές εφαρμογές του αλγόριθμου, καταφέραμε να δημιουργήσουμε έναν αλγόριθμο μεγάλης πολυπλοκότητας που συνδυάζει και τις τρεις μεθόδους (Cell ID, TA και RXLev). Ο σκοπός αυτής της παραγράφου είναι να παρουσιαστούν τα κύρια χαρακτηριστικά γνωρίσματα αυτού του αλγόριθμου.

Το πρόγραμμα αξιοποιεί πλήρως και τις τρεις ήδη αναφερθείσες μεθόδους, καθεμία για έναν διαφορετικό λόγο:

CID:

- ✓ Χρησιμοποιείται για τον καθορισμό της θέσης του Κ/Τ όταν δεν δύναται αλλιώς. Παραδείγματος χάριν, εάν δεν υπάρχει καμία πληροφορία για τις Γειτονικές Κυψέλες ή υπάρχει, αλλά οι γείτονες βρίσκονται στην ίδια τοποθεσία με το Serving Cell.

- ✓ Καθορίζει την θέση του Κ/Τ σε άλλες ειδικές περιπτώσεις. Τέτοια περίπτωση είναι όταν το Serving Cell τυγχάνει να είναι microCell και επομένως καμία περαιτέρω επεξεργασία δεν είναι απαραίτητη, αφού η περιοχή κάλυψης του microCell είναι πολύ μικρή. Έτσι η γεωγραφική θέση του microCell θεωρείται και ως θέση του Κ/Τ.

TA:

- ✓ Καθορίζει μια περιοχή εμπιστοσύνης (μια περιοχή μέσα στην οποία, το Κ/Τ θα βρίσκεται κατά πάσα πιθανότητα). Προκειμένου να κατασταθεί αυτή η περιοχή μικρότερη, εκμεταλλευόμεστε την κατευθυντικότητα της κεραίας του Serving Cell (Serving Cell's coverage area).
- ✓ Χρησιμοποιείται από κοινού με τη μέθοδο RXLev, δεδομένου ότι είναι ένας *διορθωτικός παράγοντας* για την ακτίνα HATA του Serving Cell. Επεξηγεται καλύτερα στην αναλυτική περιγραφή του αλγορίθμου.

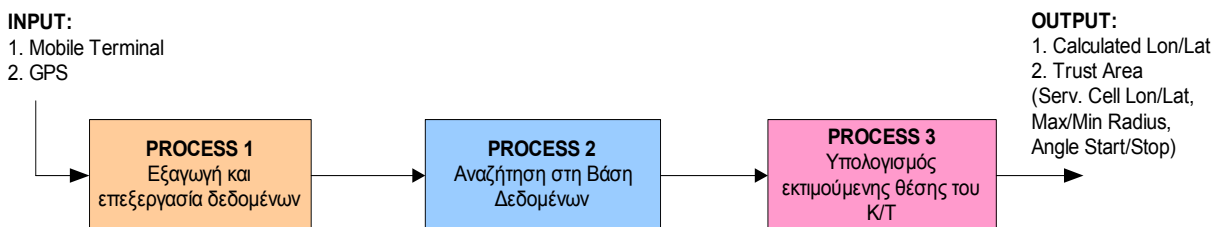
RXLev:

- ✓ Παράγει μια θέση όταν είναι δυνατόν(εάν υπάρχουν ικανοποιητικά στοιχεία). Ο αλγόριθμος είναι πιο πολύπλοκος από αυτόν που περιγράφεται στην Παράγραφο 4.2.1. Εξετάζονται διαφορετικά περιβάλλοντα μετάδοσης (HATA) για κάθε BTS (Cell), προκειμένου να ικανοποιηθούν οι απαραίτητες προϋποθέσεις (π.χ. η ακτίνα HATA του Serving Cell πρέπει να είναι μέσα στην περιοχή που ορίζει το TA). Επιπλέον, κατά την υπολογισμό των αποστάσεων, λαμβάνεται υπόψη η περιοχή κάλυψης, τόσο της κεραίας του Serving Cell όσο και των γειτονικών κυψελών (Neighbours).

Στο επόμενο κεφάλαιο δίνεται η πλήρης περιγραφή της Διαδικασίας Εντοπισμού Θέσης του Κ/Τ.

4.4. ΑΛΓΟΡΙΘΜΟΣ ΕΝΤΟΠΙΣΜΟΥ ΘΕΣΗΣ

Ο αλγόριθμος εντοπισμού θέσης αποτελείται από τρεις επιμέρους διαδικασίες (PROCESS 1, 2 & 3), όπως φαίνεται και στο σχήμα 4.4.



ΣΧΗΜΑ 4.4

Σε αυτό το κεφάλαιο θα γίνει η πλήρης περιγραφή του αλγόριθμου, χωρίς όμως να δοθεί οποιοδήποτε παράδειγμα σε γλώσσα Visual Basic. Ο πλήρης κώδικας του προγράμματος συνοδεύομενος από σχόλια, παρουσιάζεται στο ΠΑΡΑΡΤΗΜΑ Α.

Το πρόγραμμά μας «τρέχει» σε τρία διαφορετικά modes:

- **NORMAL** Mode
- **USER** Mode
- **FULL** Mode

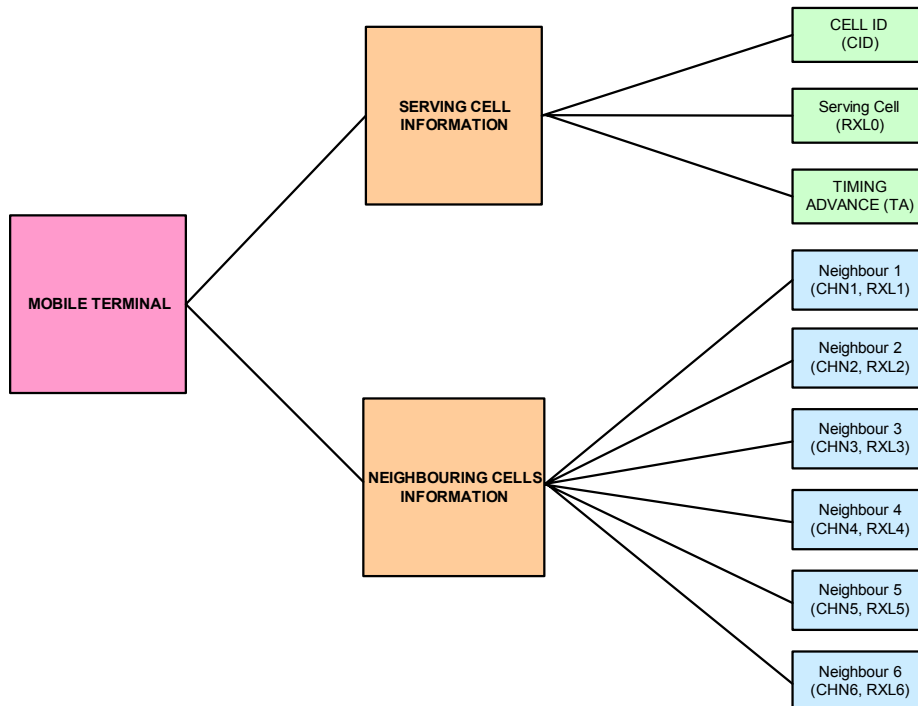
Στο "**NORMAL Mode**", το πρόγραμμα τρέχει τον αλγόριθμο εντοπισμού θέσης μόνο μια φορά. Τα αποτελέσματα παρουσιάζονται σε ένα γραφικό παράθυρο. Το Κινητό Τερματικό (Κ/Τ) είναι απαραίτητο για την εξαγωγή αποτελεσμάτων.

Στο "**USER Mode**", ο χρήστης εισάγει τα απαραίτητα στοιχεία "με το χέρι", έτσι η ύπαρξη Κινητού Τερματικού (Κ/Τ) δεν είναι απαραίτητη. Για τον λόγο αυτό, μερικές SUB-Routines δεν χρησιμοποιούνται στο "USER Mode".

Στο "**FULL Mode**", το πρόγραμμα τρέχει τον αλγόριθμο εντοπισμού θέσης συνεχώς. Υπολογίζεται η «ακρίβεια» της μεθόδου και τα αποτελέσματα σώζονται σε ένα φύλλο εργασίας του Excel. Και σε αυτήν την περίπτωση, όπως και στο "Normal Mode", το Κινητό Τερματικό (Μ/Τ) είναι απαραίτητο.

4.4.1. ΕΞΑΓΩΓΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ (PROCESS 1)

Σε αυτή τη διαδικασία, ο αλγόριθμος εντοπισμού θέσης εκτελεί την εξαγωγή όλων των απαραίτητων δεδομένων από το Κ/Τ. Για την ακρίβεια, τα στοιχεία που αποθηκεύονται για περαιτέρω επεξεργασία είναι το **“Timing Advance” (TA)**, το **Serving Cell ID (CID)** και το **RXLev**.



ΣΧΗΜΑ 4.5

Για να επιτύχουμε την επικοινωνία μεταξύ του Κ/Τ και του υπολογιστή, χρησιμοποιούμε ένα εμπορικό “ActiveX Control”, από την εταιρεία SoftwareCave (www.softwarecave.com), που ονομάζεται “Mobile FBUS 1,7 ActiveX Control”. Το Mobile FBUS μας επιτρέπει να ελέγχουμε ένα κινητό τηλέφωνο από τον υπολογιστή μας με μεγάλη ευκολία, χρησιμοποιώντας ως μέσο διασύνδεσης ένα καλώδιο FBUS.

Sub-Routines που χρησιμοποιούνται στην πρώτη διαδικασία (PROCESS 1)

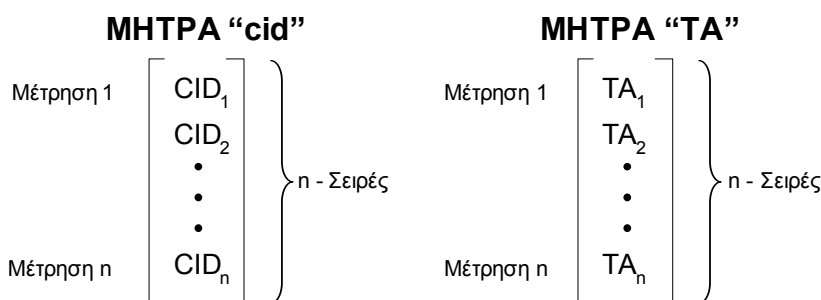
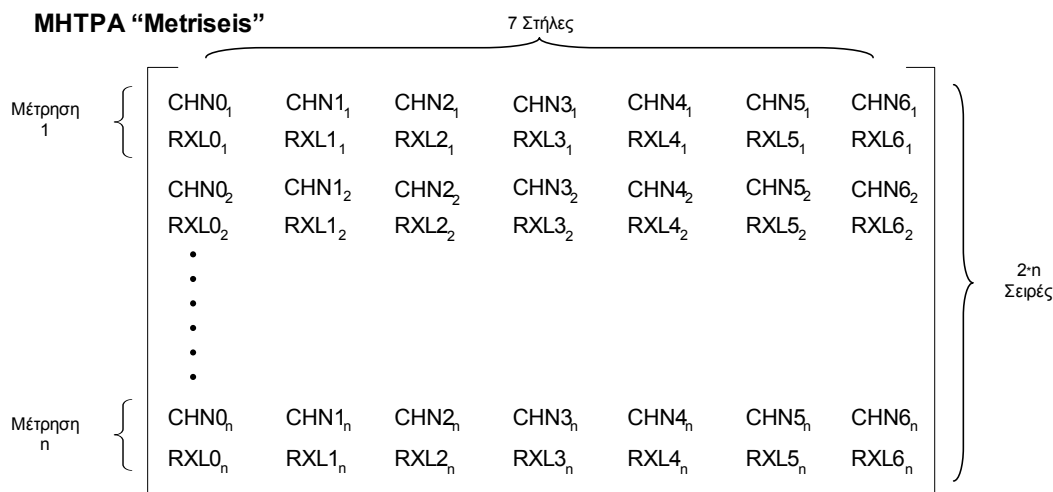
Χρησιμοποιούνται οι ακόλουθες Sub-Routines (Υπο-Ρουτίνες):

1. frmModeNormal.**Timer1_Timer()** & frmModeFull.**Timer1_Timer()**
2. ModSub.**SortMatrix**
3. ModSub.**ProcessData**
4. ModSub.**StoreValues**

4.4.1.a. Υπο-Ρουτίνα *Timer1_Timer()*

Η Υπο-Ρουτίνα **Timer1_Timer** χρησιμοποιείται μόνο στο "NORMAL" Mode και στο "FULL" Mode.

Αφού πραγματοποιηθεί η σύνδεση με το Κ/Τ, οι μετρήσεις αποθηκεύονται σε τρεις διαφορετικές Μήτρες:



MHTPA "Metriseis":

Εδώ αποθηκεύονται τα κανάλια των Σταθμών Βάσης και οι ισχύεις λήψης (RXLev). Κάθε μέτρηση αντιστοιχεί σε 2 σειρές της μήτρας(π.χ. Μέτρηση 1 → Σειρές 1&2, Μέτρηση 2 → Σειρές 3&4, κλπ.). Κάθε στήλη αντιστοιχεί και σε ένα Σταθμό Βάσης (Στήλη 1 → Κυψέλη Εξυπηρέτησης, Στήλες 2-7 → Γειτονική Κυψέλη 1-6)

Πρέπει να σημειώσουμε ότι η έκφραση (CHN1₁ = CHN1₂ = CHN1₃ =..... = CHN1_n) είναι στις περισσότερες περιπτώσεις μη αληθής. Αυτό συμβαίνει επειδή οι Γειτονικές Κυψέλες αποθηκεύονται στην μήτρα "Metriseis" ανάλογα με την τιμή του RXLEV που αντιστοιχεί στον κάθε Γείτονα. Δηλαδή ισχύει ότι:

$$RXL1_n > RXL2_n > RXL3_n > RXL4_n > RXL5_n > RXL6_n$$

Παράδειγμα 1:

Σε αυτό το παράδειγμα παρουσιάζεται η διαδικασία αποθήκευσης των δεδομένων στη Μήτρα "Metriseis". Όπως αναφέρθηκε ήδη, τα στοιχεία αποθηκεύονται κατά σειρά μέγιστης τιμής RXLEV και ΌΧΙ με βάση τον αριθμό του καναλιού (CHN) του Σταθμού Βάσης.

Κατά τη διάρκεια της Μέτρησης 1:

$$\text{CHN0}_1 = 10 \rightarrow \text{RXL0}_1 = -50 \text{ dBm}$$

$$\text{CHN1}_1 = 20 \rightarrow \text{RXL1}_1 = -60 \text{ dBm}$$

$$\text{CHN2}_1 = 30 \rightarrow \text{RXL2}_1 = -70 \text{ dBm}$$

$$\text{CHN3}_1 = 40 \rightarrow \text{RXL3}_1 = -80 \text{ dBm}$$

$$\text{CHN4}_1 = 50 \rightarrow \text{RXL4}_1 = -90 \text{ dBm}$$

$$\text{CHN5}_1 = 60 \rightarrow \text{RXL5}_1 = -100 \text{ dBm}$$

$$\text{CHN6}_1 = 70 \rightarrow \text{RXL6}_1 = -110 \text{ dBm}$$

Κατά τη διάρκεια της Μέτρησης 2:

$$\text{CHN0}_2 = 10 \rightarrow \text{RXL0}_2 = -52 \text{ dBm}$$

$$\text{CHN1}_2 = 30 \rightarrow \text{RXL1}_2 = -64 \text{ dBm}$$

$$\text{CHN2}_2 = 20 \rightarrow \text{RXL2}_2 = -65 \text{ dBm}$$

$$\text{CHN3}_2 = 40 \rightarrow \text{RXL3}_2 = -78 \text{ dBm}$$

$$\text{CHN4}_2 = 50 \rightarrow \text{RXL4}_2 = -91 \text{ dBm}$$

$$\text{CHN5}_2 = 70 \rightarrow \text{RXL5}_2 = -104 \text{ dBm}$$

$$\text{CHN6}_2 = 80 \rightarrow \text{RXL6}_2 = -109 \text{ dB}$$

Κατά τη διάρκεια της Μέτρησης 3:

$$\text{CHN0}_3 = 10 \rightarrow \text{RXL0}_3 = -52 \text{ dBm}$$

$$\text{CHN1}_3 = 30 \rightarrow \text{RXL1}_3 = -65 \text{ dBm}$$

$$\text{CHN2}_3 = 20 \rightarrow \text{RXL2}_3 = -68 \text{ dBm}$$

$$\text{CHN3}_3 = 40 \rightarrow \text{RXL3}_3 = -79 \text{ dBm}$$

$$\text{CHN4}_3 = 50 \rightarrow \text{RXL4}_3 = -90 \text{ dBm}$$

$$\text{CHN5}_3 = 60 \rightarrow \text{RXL5}_3 = -105 \text{ dBm}$$

$$\text{CHN6}_3 = 70 \rightarrow \text{RXL6}_3 = -106 \text{ dBm}$$

Η Μήτρα "Metriseis" που προκύπτει τελικά, με βάση αυτές τις μετρήσεις, φαίνεται παρακάτω:

ΜΗΤΡΑ “Metriseis”

| | | | | | | | |
|-----------|-----|-----|-----|-----|-----|------|------|
| Μέτρηση 1 | 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| | -50 | -60 | -70 | -80 | -90 | -100 | -110 |
| Μέτρηση 2 | 10 | 30 | 20 | 40 | 50 | 70 | 80 |
| | -52 | -64 | -65 | -78 | -91 | -104 | -109 |
| Μέτρηση 3 | 10 | 30 | 20 | 40 | 50 | 60 | 70 |
| | -52 | -65 | -68 | -79 | -90 | -105 | -106 |

↓ ←
 Διαφορετικά Κανάλια στην ίδια στήλη Νέο Κανάλι (δεν υπάρχει στη Μέτρηση 1)

Σε αυτό το παράδειγμα μπορούμε να εστιάσουμε σε δύο ενδιαφέρουσες περιπτώσεις:

1. Στη στήλη (4) το κανάλι είναι το 40. Στη στήλη (2) έχουμε δύο κανάλια, το 20 και το 30. Κατά συνέπεια, μπορούμε να πούμε ότι σε μία τυχαία στήλη (X), η τιμή για το $CHN(X-1)_n$ δεν θα είναι πάντοτε ίδια.
2. Το κανάλι 80 (Μέτρηση 2) δεν είναι παρών στις Μετρήσεις 1&3. Το κανάλι 60 (Μετρήσεις 1&3) δεν είναι παρών στη Μέτρηση 2. Στη Μήτρα "Metriseis" έχουμε συνολικά τα ακόλουθα κανάλια που αντιστοιχούν σε διαφορετικές Κυψέλες: 10, 20, 30, 40, 50, 60, 70, 80. Συνολικός αριθμός καναλιών = 8. Άρα, μπορούμε να καταλήξουμε στο συμπέρασμα ότι ο συνολικός αριθμός των διαθέσιμων Γειτόνων μπορεί να είναι μεγαλύτερος των 6 (αφού στη στήλη (1) αποθηκεύονται οι πληροφορίες για την Κυψέλη Εξυπηρέτησης και όχι για κάποιο Γείτονα).

ΜΗΤΡΑ "cid":

Εδώ αποθηκεύεται η τιμή της Ταυτότητας Κυψέλης (Cell ID) της Κυψέλης Εξυπηρέτησης, για κάθε μέτρηση. Η τιμή αυτή μπορεί να αλλάξει κατά τη διάρκεια των μετρήσεων λόγω κάποιου Handover ($CID_1 \rightarrow \text{Handover} \rightarrow CID_2 \neq CID_1$).

ΜΗΤΡΑ "TA":

Εδώ αποθηκεύουμε την τιμή του Timing Advance (TA) για κάθε μέτρηση. Και αυτή η τιμή μπορεί να αλλάξει κατά τη διάρκεια των μετρήσεων ($TA \in [0, 1, 2, \dots, 63]$).

Μετά από την αποθήκευση των δεδομένων, καλείται η Υπο-Ρουτίνα ModSub. **SortMatrix.**

4.4.1.b. Υπο-Ρουτίνα *SortMatrix*

Η Υπο-Ρουτίνα **SortMatrix** χρησιμοποιείται μόνο στο "NORMAL" Mode και στο "FULL" Mode.

Η συγκεκριμένη Υπο-Ρουτίνα ταξινομεί την Μήτρα "Metriseis" σύμφωνα με τα στοιχεία της πρώτης σειράς.

ΠΡΟΣΟΧΗ:

Στην πρώτη στήλη δεν πραγματοποιείται ταξινόμηση, επειδή κατά τη διάρκεια της επικοινωνίας του Κ/Τ με το δίκτυο, το κανάλι που αντιστοιχεί στο Serving Cell μπορεί να πάρει και τιμές εκτός BCCH.

Εάν πάρουμε τη Μήτρα "Metriseis" του Παραδείγματος 1:

ΜΗΤΡΑ "Metriseis"

| | | | | | | |
|-----|-----|-----|-----|-----|------|------|
| 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| -50 | -60 | -70 | -80 | -90 | -100 | -110 |
| 10 | 30 | 20 | 40 | 50 | 70 | 80 |
| -52 | -64 | -65 | -78 | -91 | -104 | -109 |
| 10 | 30 | 20 | 40 | 50 | 60 | 70 |
| -52 | -65 | -68 | -79 | -90 | -105 | -106 |

→ Σειρά αναφοράς για την ταξινόμηση

Μετά την ταξινόμηση, η καινούργια Μήτρα θα έχει την ακόλουθη μορφή:

ΜΗΤΡΑ "Metriseis"

| | | | | | | |
|-----|-----|-----|-----|-----|------|------|
| 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| -50 | -60 | -70 | -80 | -90 | -100 | -110 |
| 10 | 20 | 30 | 40 | 50 | 80 | 70 |
| -52 | -65 | -64 | -78 | -91 | -109 | -104 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| -52 | -68 | -65 | -79 | -90 | -105 | -106 |

CHN 10 CHN 20 CHN 30 CHN 40 CHN 50 CHN 60
CHN 80 CHN 70

Όπως μπορούμε να διαπιστώσουμε, στη στήλη (6) έχουμε δύο διαφορετικά κανάλια (CHN60 & CHN80). Αυτό συμβαίνει επειδή στη σειρά (4) (που αναφέρεται στη Μέτρηση 2), το κανάλι CHN60

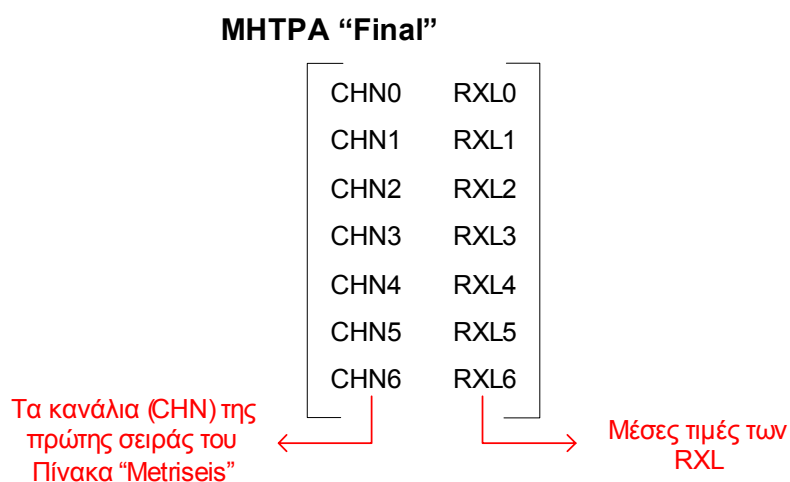
δεν είναι παρών και επομένως το κανάλι CHN80 (που δεν είναι παρόν στη Σειρά "αναφοράς"(1)) τοποθετείται στη θέση του.

4.4.1.c. Υπο-Ρουτίνα *ProcessData*

Η Υπο-Ρουτίνα **ProcessData** χρησιμοποιείται μόνο στο "NORMAL" Mode και στο "FULL" Mode.

Εδώ το πρόγραμμα ελέγχει εάν η Κυψέλη Εξυπηρέτησης (Serving Cell) έχει αλλάξει κατά τη διάρκεια των μετρήσεων. Εάν κάτι τέτοιο έχει συμβεί, το πρόγραμμα κρατά μόνο εκείνες τις μετρήσεις που έχουν την ίδια τιμή "CID" με την πρώτη μέτρηση (Ο αριθμός των μετρήσεων με το ίδιο "CID" = "SafeMeasure").

Κατόπιν, υπολογίζει τις μέσες τιμές για τις ισχύεις λήψης RXL, τις αποθηκεύει στη Μήτρα "Final",



και ταξινομεί τα στοιχεία (Bubble Sort) κατά σειρά μέγιστης τιμής του RXL (εκτός από το Serving Cell).

Τέλος, το πρόγραμμα υπολογίζει τις τελικές τιμές για το cid και το TA (οι οποίες αποθηκεύονται στις μεταβλητές "FinalCID" και "FinalTA"):

$$\text{FinalCID} = \text{cid}(1)$$

$$\text{FinalTA} = \frac{\text{TA}(1) + \text{TA}(2) + \dots + \text{TA}(\text{SafeMeasure})}{\text{SafeMeasure}}$$

Παράδειγμα 2:

Υποθέτουμε ότι οι Μήτρες "Metriseis", "cid" και "TA" μετά από την Υπο-Ρουτίνα "SortMatrix" είναι:

ΜΗΤΡΑ "Metriseis"

| | | | | | | |
|-----|-----|-----|-----|-----|------|------|
| 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| -50 | -60 | -70 | -80 | -90 | -102 | -105 |
| 10 | 20 | 30 | 40 | 50 | 80 | 70 |
| -52 | -65 | -64 | -78 | -91 | -109 | -102 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 |
| -52 | -68 | -65 | -79 | -90 | -108 | -106 |
| 20 | 10 | 30 | 40 | 50 | 60 | 70 |
| -55 | -60 | -64 | -78 | -91 | -103 | -104 |
| 20 | 10 | 30 | 40 | 50 | 60 | 70 |
| -57 | -58 | -68 | -79 | -90 | -105 | -106 |

ΜΗΤΡΑ "cid"

| | | |
|------|---|-----------|
| 1024 | → | Μέτρηση 1 |
| 1024 | → | Μέτρηση 2 |
| 1024 | → | Μέτρηση 3 |
| 1120 | → | Μέτρηση 4 |
| 1120 | → | Μέτρηση 5 |

ΜΗΤΡΑ "TA"

| | | |
|---|---|-----------|
| 0 | → | Μέτρηση 1 |
| 0 | → | Μέτρηση 2 |
| 1 | → | Μέτρηση 3 |
| 1 | → | Μέτρηση 4 |
| 2 | → | Μέτρηση 5 |

Όταν τρέχουμε την Υπο-Ρουτίνα **ProcessData** το πρόγραμμα βλέπει ότι κατά τη διάρκεια της μέτρησης 4, το Serving Cell έχει αλλάξει. Επομένως κρατά μόνο τις πρώτες 3 μετρήσεις (η μεταβλητή "SafeMeasure = 3").

Ακολουθεί ο υπολογισμός των μέσων τιμών για τα RXLEV και η ταξινόμηση της μήτρας:

$$\text{CHN10} \rightarrow \text{RXL}_{\text{mean}} = \frac{-50 - 52 - 52}{3} = -51.33 \approx -51 \text{ dBm}$$

$$\text{CHN20} \rightarrow \text{RXL}_{\text{mean}} = \frac{-60 - 65 - 68}{3} = -64.33 \approx -64 \text{ dBm}$$

$$\text{CHN30} \rightarrow \text{RXL}_{\text{mean}} = \frac{-70 - 64 - 65}{3} = -66.33 \approx -66 \text{ dBm}$$

$$\text{CHN40} \rightarrow \text{RXL}_{\text{mean}} = \frac{-80 - 78 - 79}{3} = -79 \text{ dBm}$$

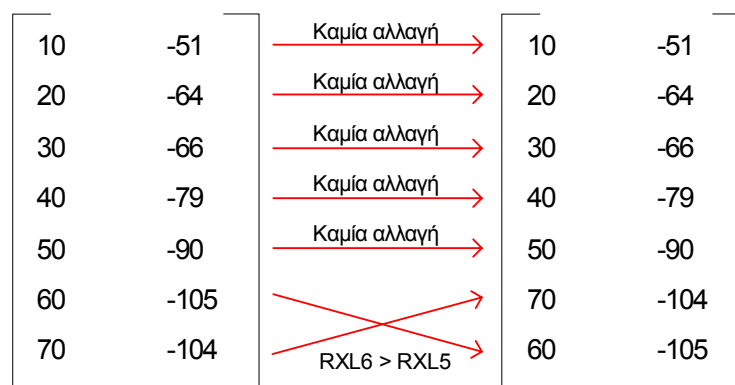
$$\text{CHN50} \rightarrow \text{RXL}_{\text{mean}} = \frac{-90 - 91 - 90}{3} = -90.33 \approx -90 \text{ dBm}$$

$$\text{CHN60} \rightarrow \text{RXL}_{\text{mean}} = \frac{-102 - 108}{2} = -105 \text{ dBm} \quad *(\mu\acute{o}\nu\omicron 2 \text{ διαθέσιμες μετρήσεις})$$

$$\text{CHN70} \rightarrow \text{RXL}_{\text{mean}} = \frac{-105 - 102 - 106}{3} = -104.33 \approx -104 \text{ dBm}$$

ΜΗΤΡΑ “Final” πριν το BubbleSort

ΜΗΤΡΑ “Final” μετά το BubbleSort



Τέλος, το πρόγραμμα υπολογίζει τα "FinalCID" και "FinalTA":

$$\text{FinalCID} = \text{cid}(1) = 1024$$

$$\text{FinalTA} = \frac{\text{TA}(1)+\text{TA}(2)+\dots+\text{TA}(\text{SafeMeasure})}{\text{SafeMeasure}} \Bigg|_{\text{SafeMeasure}=3} =$$

$$= \frac{\text{TA}(1)+\text{TA}(2)+\text{TA}(3)}{3} = \frac{0+0+1}{3} = 0.33 \approx 0$$

4.4.1.d. Υπο-Ρουτίνα **StoreValues**

Η Υπο-Ρουτίνα **StoreValues** χρησιμοποιείται μόνο στο "NORMAL" Mode και στο "FULL" Mode.

Είναι το τελευταίο βήμα πριν από την **PROCESS 2**. Αυτή η πολύ απλή Υπο-Ρουτίνα υπολογίζει την τιμή της μεταβλητής "DataMax" (Ο αριθμός διαθέσιμων Κυψελών, συμπεριλαμβανομένης και της Κυψέλης Εξυπηρέτησης) και αποθηκεύει τη Μήτρα Final() στη Μήτρα FinalOk(). Για να γίνει κατανοητή αυτή η Υπο-Ρουτίνα, ας δούμε το ακόλουθο παράδειγμα.

Παράδειγμα 3:

Υποθέτουμε ότι η Μήτρα "Final" μετά από την Υπο-Ρουτίνα **ProcessData** είναι:

ΜΗΤΡΑ "Final"

| | |
|----|-----|
| 10 | -51 |
| 20 | -64 |
| 30 | -66 |
| 40 | -79 |
| 50 | -90 |
| 0 | 0 |
| 0 | 0 |

Σε αυτή την περίπτωση, ο αριθμός των διαθέσιμων Γειτόνων είναι 4 (Η τιμή για τους Γείτονες 5&6 είναι 0, άρα θεωρούμε ότι δεν υπάρχουν).

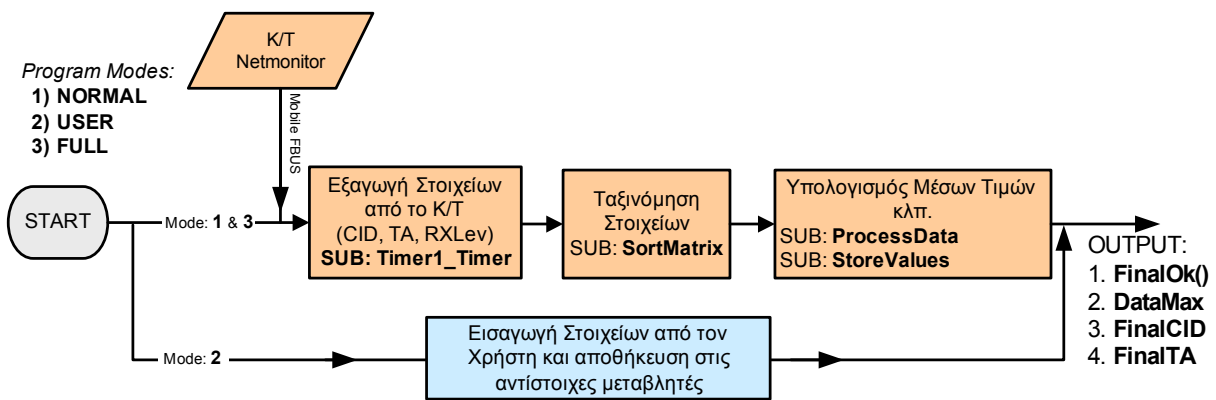
Η Υπο-Ρουτίνα **StoreValues** υπολογίζει την μεταβλητή "DataMax":

Σημείωση:

Στο "USER" Mode καμία από τις προηγούμενες Υπο-Ρουτίνες δεν χρησιμοποιείται. Όλα τα απαραίτητα στοιχεία εισάγονται από τον Χρήστη "χειροκίνητα" και αποθηκεύονται αυτόματα (στις ίδιες μεταβλητές: DataMax, FinalCID, FinalTA και FinalOk()), μόλις εκκινηθεί ο Αλγόριθμος Εντοπισμού Θέσης από τον Χρήστη.

Το διάγραμμα ροής της πρώτης Διαδικασίας, "PROCESS 1" παρουσιάζεται παρακάτω (Σχήμα 4.6).

"PROCESS 1 - ΕΞΑΓΩΓΗ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ"



ΣΧΗΜΑ 4.6

4.4.2. ΑΝΑΖΗΤΗΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ (PROCESS 2)

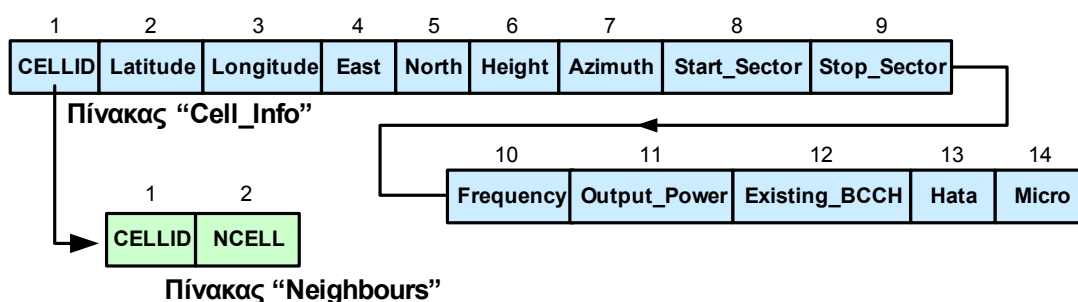
4.4.2.a. Τύπος και Δομή της Βάσης Δεδομένων

Η συγκεκριμένη Βάση Δεδομένων έχει δημιουργηθεί με τη βοήθεια του λογισμικού **"Microsoft Access 2000"**. Το αρχείο που περιέχει τη βάση δεδομένων ονομάζεται: **"LocDatabase.mdb"**.

Το LocDatabase.mdb αποτελείται από δύο πίνακες:

1. Πίνακας **"Cell_Info"**
2. Πίνακας **"Neighbours"**

Αυτοί οι δύο πίνακες είναι συσχετισμένοι μεταξύ τους, όπως φαίνεται και στο σχήμα 4.7.



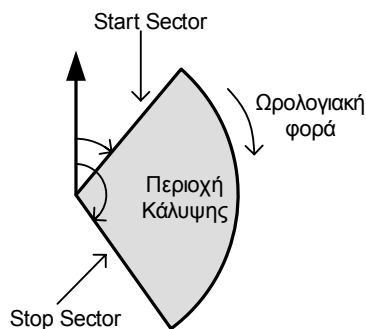
ΣΧΗΜΑ 4.7

Πίνακας Cell Info:

Κάθε σειρά αντιστοιχεί σε ένα διαφορετικό BTS, το οποίο περιγράφεται από το "πρωτεύον κλειδί" του στο πεδίο 1 (primary key: "CELLID" field).

- 1) CELLID:** Η ταυτότητα κυψέλης του Σταθμού Βάσης (BTS).
 Πριν από τον αριθμό υπάρχει ο χαρακτήρας "G" ή "D" που φανερώνει τη συχνότητα λειτουργίας ("G" → GSM900, "D" → DCS1800).
 π.χ. εάν το CID είναι 1000 και η συχνότητα 900 MHz, η τιμή στο Πεδίο 1 του πίνακα θα είναι: **"G1000"**
- 2) Latitude:** Γεωγραφικό πλάτος της θέσης του Σταθμού Βάσης (BTS) σε δεκαδική μορφή (DD.ddddd), π.χ. 38.55555°

- 3) Longitude:** Γεωγραφικό μήκος της θέσης σταθμού βάσης (BTS) σε δεκαδική μορφή (DDD.ddddd), πχ. 023.55555°.
- 4) East:** Η θέση του BTS (άξονας x) σε μορφή UTM zone34, πχ. 742000 m
- 5) North:** Η θέση του BTS (άξονας y) σε μορφή UTM zone34, πχ. 4210000 m
- 6) Height:** Το ύψος της κεραίας του BTS σε μέτρα (πχ. 25m)
- 7) Azimuth:** Το αζιμούθιο της κατεύθυνσης του κεντρικού λοβού της κεραίας σε μοίρες (πχ. 120°)
- 8) Start_Sector:**
Το αζιμούθιο του StartSector (γωνία 3dB, έναρξης της περιοχής κάλυψης) του BTS σε μοίρες (πχ. 40°), βλέπε Σχ. 4.8
- 9) Stop_Sector:**
Το αζιμούθιο του StopSector (γωνία 3dB, λήξης της περιοχής κάλυψης) του BTS σε μοίρες (πχ. 100°), βλέπε Σχ. 4.8

**ΣΧΗΜΑ 4.8**

- 10) Frequency:** Συχνότητα λειτουργίας του BTS σε MHz (900MHz, 1800MHz)

- 11) Output_Power:**

Η στάθμη εκπεμπόμενης ισχύος του BTS σε dBm (πχ. 38 dBm)

12) Existing_BCCH:

Ο αριθμός του καναλιού BCCH (BCCH: Broadcast Control Channel, το κανάλι που περιέχει ακριβείς πληροφορίες για το δίκτυο). Αυτό είναι το κανάλι που είναι εμφανές στο Κινητό Τερματικό, όταν το τελευταίο βρίσκεται σε κατάσταση αναμονής "idle mode" (δεν πραγματοποιείται κλήση, δεν αποστέλλεται SMS, κλπ.)

13) Hata:

Ο προεπιλεγμένος τύπος για το μοντέλο ραδιοκάλυψης του Hata.

Hata = 1: Open Area

Hata = 2: Suburban Area

Hata = 3: Medium-Small City

Hata = 4: Large City

14) Micro:

Αναγνωριστική τιμή για το BTS

Micro > 0: μικρο-Κυψέλη με ακτίνα κάλυψης = Micro

Micro = 0: απλή Κυψέλη

Πίνακας Neighbours:

Κάθε σειρά αντιστοιχεί σε ένα BTS, το οποίο περιγράφεται από το πεδίο 1 (CELLID).

1) CELLID:

Η ταυτότητα κυψέλης του BTS.

Εδώ το CELLID δεν έχει καθοριστεί ως πρωτεύον κλειδί, επομένως επιτρέπονται τα πολλαπλά αντίγραφα (η ίδια τιμή του CellID μπορεί να είναι παρούσα σε περισσότερες από 2 σειρές).

2) NCELL:

Η ταυτότητα κυψέλης του γειτονικού BTS (Neighbour).

Παράδειγμα 4:

Εδώ παρουσιάζουμε μια χαρακτηριστική δομή του Πίνακα "Neighbours":

| CELLID | NCELL |
|--------|-------|
| G100 | G165 |
| G100 | G200 |
| G100 | G220 |
| G100 | D1050 |
| G100 | D1030 |
| G100 | G350 |
| G100 | G190 |
| G100 | G500 |
| G200 | G320 |
| G200 | G210 |
| G200 | G100 |
| G200 | D1050 |
| G200 | G350 |
| G200 | G180 |
| G200 | G220 |

Πραγματοποίηση σύνδεσης με τη Βάση Δεδομένων

Για να επιτύχουμε τη σύνδεση, χρησιμοποιούμε ένα αρχείο "Microsoft Data Link", που το ονομάζουμε "LocDatabase.udl". Το αρχείο αυτό περιέχει τις κάτωθι πληροφορίες:

- ✓ Provider = Microsoft Jet 4.0 OLE DB Provider
- ✓ Connection = Database file location (eg. "C:\Locdatabase.mdb")

Σειρά 1 → Serving BTS

Σειρά 2 → Neighbour 1

Σειρά 3 → Neighbour 2

...

...

Σειρά "DataMax" → Neighbour ("DataMax" – 1)

Εφαρμογή της Αναζήτησης:

Και τα τρία modes του προγράμματος (NORMAL, USER και FULL) χρησιμοποιούν για τη συγκεκριμένη διαδικασία (PROCESS 2) τις ίδιες Sub-Routines, χωρίς να γίνεται καμία εξαίρεση.

Οι Υπο-Ρουτίνες (Sub-Routines) που χρησιμοποιούνται στην δεύτερη διαδικασία (PROCESS 2) είναι οι ακόλουθες:

1. ModSub.**Databasefind**
2. ModSub.**ProcessDist**
3. ModSub.**GetFinalMatrix**

4.4.2.c. Υπο-Ρουτίνα Databasefind

Η Υπο-Ρουτίνα ξεκινάει με την εύρεση των γειτόνων της Κυψέλης στην οποία αντιστοιχεί το "FinalCID". Αυτό γίνεται με το ακόλουθο ερώτημα SQL:

**"SELECT * FROM Neighbours WHERE
CELLID = 'G' & FinalCID & ' ' OR CELLID = 'D' & FinalCID & ' '"**

Το ερώτημα επιστρέφει ένα Recordset (adoRecordset) που περιέχει όλους τους εγγραμμένους γείτονες της κυψέλης "FinalCID" ("FinalCID" είναι η ταυτότητα κυψέλης που βρέθηκε κατά τη διάρκεια των μετρήσεων (βλ. PROCESS 1)). Αν δεν βρεθεί κάποιος Σταθμός Βάσης που να αντιστοιχεί στο συγκεκριμένο "FinalCID", τότε η μεταβλητή "DatabaseError" λαμβάνει τη τιμή TRUE και ο αλγόριθμος διακόπτεται. Αντίθετα, αν όλα πάνε καλά → "DatabaseError = FALSE".

Παράδειγμα 5:

Λαμβάνοντας υπόψη τον πίνακα "Neighbours" του Παραδείγματος 4 και εάν FinalCID = 100 έπεται:

Ερώτημα SQL:

"SELECT * FROM Neighbours WHERE CELLID = 'G100' OR CELLID = 'D100'"

Το Recordset που επιστρέφει είναι το ακόλουθο:

"adoRecordset"

| CELLID | NCELL |
|--------|-------|
| G100 | G165 |
| G100 | G200 |
| G100 | G220 |
| G100 | D1050 |
| G100 | D1030 |
| G100 | G350 |
| G100 | G190 |
| G100 | G500 |

Σημείωση:

Το "FinalCID" δεν περιέχει την πληροφορία για τη συχνότητα λειτουργίας("G" ή "D"). Αλλά, στο ανωτέρω Recordset, παρατηρούμε ότι υπάρχει αυτή η πληροφορία στο πεδίο CELLID. Μπορούμε λοιπόν να θεωρήσουμε γνωστή τη συχνότητα λειτουργίας μιας και δεν υπάρχει περίπτωση να έχουμε δύο διαφορετικά BTS (BTS1 & BTS2) με CELLID1 = G100 και CELLID2 = D100.

Ο επόμενος στόχος της Υπο-Ρουτίνας είναι να βρεθούν όλες οι διαθέσιμες πληροφορίες για την Κυψέλη Εξυπηρέτησης (Serving Cell) και να αποθηκευτούν στη σειρά (1) της Μήτρας FinalOk. Αυτό επιτυγχάνεται σε 3 βήματα:

1. Αποθήκευση του καινούργιου FinalCID (πληροφορία συχνότητας + Cell Identifier, πχ. G100):

```
FinalCID = adoRecordset.Fields("CELLID").Value
```

2. Ερώτημα SQL:

"SELECT * FROM Cell_Info WHERE CELLID = " & FinalCID & """

Το ερώτημα αυτό επιστρέφει ένα recordset (srvRecordset) που περιέχει όλες τις πληροφορίες για τον Σταθμό Βάσης "FinalCID".

3. Αποθήκευση των ευρεθέντων πληροφοριών στη Μήτρα FinalOk():

```
FinalOk(1, 3) = srvRecordset.Fields("CELLID").Value  
FinalOk(1, 4) = srvRecordset.Fields("Latitude").Value  
FinalOk(1, 5) = srvRecordset.Fields("Longitude").Value  
FinalOk(1, 6) = srvRecordset.Fields("Height").Value  
FinalOk(1, 7) = srvRecordset.Fields("Output_Power").Value  
FinalOk(1, 9) = srvRecordset.Fields("Micro").Value  
FinalOk(1, 10) = srvRecordset.Fields("North").Value  
FinalOk(1, 11) = srvRecordset.Fields("East").Value  
FinalOk(1, 12) = srvRecordset.Fields("Azimuth").Value  
FinalOk(1, 13) = srvRecordset.Fields("Hata").Value  
FinalOk(1, 14) = srvRecordset.Fields("Frequency").Value  
FinalOk(1, 15) = srvRecordset.Fields("Start_Sector").Value  
FinalOk(1, 16) = srvRecordset.Fields("Stop_Sector").Value
```

Προσδιορισμός Γειτονικών Κυψελών

Τώρα ο αλγόριθμος προσπαθεί να βρει την Ταυτότητα Κυψέλης για τον κάθε διαθέσιμο Γειτονικό Σταθμό Βάσης (Neighbour). Ισχύει ότι Αριθμός Γειτόνων = "DataMax" -1.

Μεταβλητές που χρησιμοποιούνται κατά τον προσδιορισμό:

"i" : Αριθμός του Σταθμού Βάσης στον οποίο αναφερόμαστε κάθε φορά
 i = 2 → Neighbour1,
 i = 3 → Neighbour2,

 i = DataMax → Neighbour(DataMax-1)

"CHNSearch": Το BCCH κανάλι του γείτονα που προσπαθούμε να προσδιορίσουμε.

CHNSearch = FinalOk(i, 1)

Ξεκινώντας τον προσδιορισμό, το πρόγραμμα βρίσκει όλες τις Κυψέλες στον πίνακα "Cell_Info" οι οποίες ικανοποιούν την παρακάτω ισότητα:

$$\text{Existing BCCH Channel} = \text{CHNSearch}$$

Το αντίστοιχο ερώτημα SQL είναι:

**"SELECT * FROM Cell_Info
 WHERE Existing_BCCH = " & CHNSearch & " ORDER BY CELLID"**

Αυτό το ερώτημα επιστρέφει ένα Recordset (chnRecordset) που περιέχει τα αποτελέσματα. Ο αριθμός των αποτελεσμάτων είναι ίσος με τη μεταβλητή "ChnrecMax".

➔ Εάν δεν έχουμε κανένα αποτέλεσμα (ChnrecMax=0) τότε:

$$\text{FinalOk}(i, 1) = \text{"ERROR"}$$

και προχωράμε στον επόμενο Neighbour. (i = i + 1)

- Αν όμως $ChnrecMax \neq 0$, ο αλγόριθμος κρατά μόνο εκείνα τα αποτελέσματα των οποίων η ταυτότητα κυψέλης (Cell Identifier) βρίσκεται στην λίστα Γειτόνων του Serving Cell (adoRecordset, στο πεδίο NCELL).

Η εντολή σε ADO, που εκτελεί τα παραπάνω, είναι η εξής:

```
adoRecordset.Find "NCELL = 'chnRecordset.Fields("CELLID").Value'"
```

Υπάρχουν τώρα τρεις περιπτώσεις:

1) Να βρεθούν περισσότερες Κυψέλες από μία:

Ελέγχουμε ποια από αυτές απέχει λιγότερο από το Serving Cell και αφού την βρούμε, αποθηκεύουμε τις πληροφορίες της στη Μήτρα "FinalOk".

2) Να βρεθεί μόνο μία Κυψέλη:

Αποθηκεύουμε τις πληροφορίες της στη Μήτρα "FinalOk".

3) Να μην βρεθεί καμία Κυψέλη:

```
FinalOk(i, 1) = "ERROR"
```

Παράδειγμα 6:

Ας υποθέσουμε ότι έχουμε τη Μήτρα FinalOk του Παραδείγματος 3:

ΜΗΤΡΑ "FinalOk" 16 Στήλες

| | | | | | | | |
|--------------|----|-----|------|------|-------|-------|------|
| Serving Cell | 10 | -51 | void | void | | | void |
| Neighbour 1 | 20 | -64 | void | void | | | void |
| Neighbour 2 | 30 | -66 | void | void | | | void |
| Neighbour 3 | 40 | -79 | void | void | | | void |
| Neighbour 4 | 50 | -90 | void | void | | | void |

CHN RXL

} "DataMax" Σειρές

όπου,

DataMax = 5

FinalCID = 100

Επίσης, από το Παράδειγμα 5 έχουμε:

“adoRecordset”

| CELLID | NCELL |
|---------------|--------------|
| G100 | G165 |
| G100 | G200 |
| G100 | G220 |
| G100 | D1050 |
| G100 | D1030 |
| G100 | G350 |
| G100 | G190 |
| G100 | G500 |

Ένα μέρος του Πίνακα “Cell_Info” είναι το παρακάτω:

| CELLID | Latitude | Longitude | East | North | | | Existing_BCCH |
|---------------|-----------------|------------------|-------------|--------------|--------------|--------------|----------------------|
| G100 | 38.11111 | 23.71111 | 740000 | 4200000 | | | 10 |
| G200 | 38.11100 | 23.71100 | 740105 | 4200254 | | | 50 |
| G220 | 38.22222 | 23.72222 | 740100 | 4200100 | | | 30 |
| G350 | 38.11111 | 23.71111 | 740000 | 4200000 | | | 20 |
| G190 | 38.11100 | 23.71100 | 740105 | 4200254 | | | 60 |
| G500 | 38.33333 | 23.73333 | 741000 | 4200000 | | | 30 |
| G165 | 38.51045 | 23.72041 | 741254 | 4205413 | | | 70 |
| G320 | 38.50010 | 23.77001 | 740120 | 4201536 | | | 40 |
| D1050 | 38.51045 | 23.72041 | 741254 | 4205413 | | | 80 |

Η Μήτρα FinalOk, μετά τον “Προσδιορισμό του Serving Cell”, δίνεται:

ΜΗΤΡΑ “FinalOk” 16 Στήλες

| | | | | | | | |
|--------------|-----|-----|---------|----------|-----------|-------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | | |
| Neighbour 1 | 20 | -64 | void | void | | | |
| Neighbour 2 | 30 | -66 | void | void | | | |
| Neighbour 3 | 40 | -79 | void | void | | | |
| Neighbour 4 | 50 | -90 | void | void | | | |
| | CHN | RXL | Cell ID | Latitude | Longitude | | |

Εκτελείται τώρα ο “Προσδιορισμός Γειτονικών Κυψελών”:

i = 2

CHNSearch = FinalOk(2,1) = 20

Ερώτημα SQL:

"SELECT * FROM Cell_Info WHERE Existing_BCCH = 20 ORDER BY CELLID"

Το Recordset (chnRecordset) που επιστρέφεται είναι το ακόλουθο:

| CELLID | Latitude | Longitude | East | North | | | Existing_BCCH |
|--------|----------|-----------|--------|---------|-------|-------|---------------|
| G350 | 38.11111 | 23.71111 | 740000 | 4200000 | | | 20 |

ChnrecMax = 1 ≠ 0

Εντολή ADO:

adoRecordset.Find "NCELL = 'G350'"

Η κυψέλη “G350” υπάρχει στη λίστα των γειτόνων του Serving Cell (adoRecordset), οπότε οι πληροφορίες του “G350” αποθηκεύονται στη Μήτρα FinalOk:

ΜΗΤΡΑ “FinalOk”

16 Στήλες

| | | | | | | |
|--------------|-----|-----|---------|----------|-----------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | |
| Neighbour 1 | 20 | -64 | G350 | 38.11111 | 23.71111 | |
| Neighbour 2 | 30 | -66 | void | void | | |
| Neighbour 3 | 40 | -79 | void | void | | |
| Neighbour 4 | 50 | -90 | void | void | | |
| | CHN | RXL | Cell ID | Latitude | Longitude | |

i = 3

CHNSearch = FinalOk(3,1) = 30

Ερώτημα SQL:

"SELECT * FROM Cell_Info WHERE Existing_BCCH = 30 ORDER BY CELLID"

Το Recordset (chnRecordset) που επιστρέφεται είναι το ακόλουθο:

| CELLID | Latitude | Longitude | East | North | | | Existing_BCCH |
|--------|----------|-----------|---------|---------|-------|-------|---------------|
| G220 | 38.22222 | 23.72222 | 740100 | 4200100 | | | 30 |
| G500 | 38.33333 | 23.73333 | 7410000 | 4200000 | | | 30 |

ChnrecMax = 2 ≠ 0

Εντολή ADO:

adoRecordset.Find "NCELL = 'G220'"

adoRecordset.Find "NCELL = 'G500'"

Τόσο η κυψέλη "G220" όσο και η "G500" περιέχονται στη λίστα των γειτόνων του Serving Cell (adoRecordset):

$$\text{"G220" Distance} = \sqrt{(740000 - 740100)^2 + (4200000 - 4200100)^2} = 141\text{m}$$

$$\text{"G500" Distance} = \sqrt{(740000 - 741000)^2 + (4200000 - 4200000)^2} = 1000\text{m}$$

"G220" Distance < "G500" Distance

Επομένως, αποθηκεύουμε τις πληροφορίες της Κυψέλης "G220" στη Μήτρα FinalOk:

ΜΗΤΡΑ "FinalOk"

16 Στήλες

| | | | | | | |
|--------------|-----|-----|---------|----------|-----------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | |
| Neighbour 1 | 20 | -64 | G350 | 38.11111 | 23.71111 | |
| Neighbour 2 | 30 | -66 | G220 | 38.22222 | 23.72222 | |
| Neighbour 3 | 40 | -79 | void | void | | |
| Neighbour 4 | 50 | -90 | void | void | | |
| | CHN | RXL | Cell ID | Latitude | Longitude | |

i = 4

CHNSearch = FinalOk(4,1) = 40

Ερώτημα SQL:

"SELECT * FROM Cell_Info WHERE Existing_BCCH = 40 ORDER BY CELLID"

Το Recordset (chnRecordset) που επιστρέφεται είναι το ακόλουθο:

| CELLID | Latitude | Longitude | East | North | | | Existing_BCCH |
|--------|----------|-----------|--------|---------|-------|-------|---------------|
| G320 | 38.50010 | 23.77001 | 740120 | 4201536 | | | 40 |

ChnrecMax = 1 ≠ 0

Εντολή ADO:

adoRecordset.Find "NCELL = 'G320'"

Η κυψέλη "G320" δεν υπάρχει στη λίστα των γειτόνων του Serving Cell (adoRecordset), επομένως:

FinalOk(4,1) = "ERROR"

ΜΗΤΡΑ “FinalOk” 16 Στήλες

| | | | | | | |
|--------------|-------|-----|---------|----------|-----------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | |
| Neighbour 1 | 20 | -64 | G350 | 38.11111 | 23.71111 | |
| Neighbour 2 | 30 | -66 | G220 | 38.22222 | 23.72222 | |
| Neighbour 3 | ERROR | | | | | |
| Neighbour 4 | 50 | -90 | void | void | | |
| | CHN | RXL | Cell ID | Latitude | Longitude | |

i = 5

CHNSearch = FinalOk(4,1) = 50

Ερώτημα SQL:

"SELECT * FROM Cell_Info WHERE Existing_BCCH = 50 ORDER BY CELLID"

Το Recordset (chnRecordset) που επιστρέφεται είναι το ακόλουθο:

| CELLID | Latitude | Longitude | East | North | | | Existing_BCCH |
|--------|----------|-----------|--------|---------|-------|-------|---------------|
| G200 | 38.11100 | 23.71100 | 740105 | 4200254 | | | 50 |

ChnrecMax = 1 ≠ 0

Εντολή ADO:

adoRecordset.Find "NCELL = 'G200'"

Η κυψέλη "G200" υπάρχει στη λίστα των γειτόνων του Serving Cell (adoRecordset), οπότε οι πληροφορίες του "G200" αποθηκεύονται στη Μήτρα FinalOk:

ΜΗΤΡΑ “FinalOk” 16 Στήλες

| | | | | | | |
|--------------|-------|-----|---------|----------|-----------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | |
| Neighbour 1 | 20 | -64 | G350 | 38.11111 | 23.71111 | |
| Neighbour 2 | 30 | -66 | G220 | 38.22222 | 23.72222 | |
| Neighbour 3 | ERROR | | | | | |
| Neighbour 4 | 50 | -90 | G200 | 38.11100 | 23.71100 | |
| | CHN | RXL | Cell ID | Latitude | Longitude | |

4.4.2.d. Υπο-Ρουτίνα *ProcessDist*

Αυτή η Υπο-Ρουτίνα υπολογίζει τις ακτίνες "HATA Radius" που προκύπτουν για κάθε κυψέλη (Serving Cell και Γειτονικές Κυψέλες) και αποθηκεύει τα αποτελέσματα στην στήλη (8) της Μήτρας "FinalOk".

Ισχύει ότι:

$$\text{HATA Radius} = 10^{\left(\frac{((T_x - R_x) - (69.55 + 26.16 \cdot \text{Log}_{10}(f_{\text{MHz}})) - 13.82 \cdot \text{Log}_{10}(h_t) - \text{FactorA} - \text{FactorK}))}{44.9 - 6.55 \cdot \text{Log}_{10}(h_t)} \right)}$$

Όπου:

h_t : Το ύψος του σταθμού βάσης σε μέτρα → FinalOk, στήλη (6)

h_r : ύψος του δέκτη (K/T) σε μέτρα → σταθερή τιμή = 2

f_{MHz} : Η συχνότητα λειτουργίας σε Megahertz → FinalOk, στήλη (14)

T_x : Η ισχύς εκπομπής σε dBm → FinalOk, στήλη (7)

R_x : Η ισχύς λήψης σε dBm → FinalOk, στήλη (2)

FactorA και FactorK: Συντελεστές που εξαρτώνται από το περιβάλλον διάδοσης.

1) Open Area:

$$\text{FactorA} = (1.1 \cdot \text{Log}_{10}(f_{\text{MHz}}) - 0.7) \cdot h_r - (1.56 \cdot \text{Log}_{10}(f_{\text{MHz}}) - 0.8)$$

$$\text{FactorK} = 4.78 \cdot (\text{Log}_{10}(f_{\text{MHz}}))^2 - 18.33 \cdot \text{Log}_{10}(f_{\text{MHz}}) + 40.94$$

2) Suburban Area:

$$\text{FactorA} = (1.1 \cdot \text{Log}_{10}(f_{\text{MHz}}) - 0.7) \cdot h_r - (1.56 \cdot \text{Log}_{10}(f_{\text{MHz}}) - 0.8)$$

$$\text{FactorK} = 2 \cdot \left(\text{Log}_{10} \left(\frac{f_{\text{MHz}}}{28} \right) \right)^2 + 5.4$$

3) Medium-Small City

$$\text{FactorA} = (1.1 \cdot \text{Log}_{10}(f_{\text{MHz}}) - 0.7) \cdot h_r - (1.56 \cdot \text{Log}_{10}(f_{\text{MHz}}) - 0.8)$$

$$\text{FactorK} = 0$$

4) Large City

$$\text{FactorA} = 3.2 \cdot (\text{Log}_{10}(11.75 \cdot h_r))^2 - 4.97$$

$$\text{FactorK} = 0$$

4.4.2.e. Υπο-Ρουτίνα GetFinalMatrix

Σε αυτή την Υπο-Ρουτίνα, το πρόγραμμα απορρίπτει όλες τις περιπτώσεις όπου η Sub-Routine **DatabaseFind** έχει εντοπίσει "ERROR" (co-sited κυψέλες, ή κυψέλες που δεν βρέθηκαν σε έναν από τους δύο πίνακες της βάσης δεδομένων). Η Μήτρα "FinalOk" αναταξινομείται και τέλος, υπολογίζεται η καινούργια τιμή για την μεταβλητή "DataMax".

Παράδειγμα 7:

Παίρνουμε τη Μήτρα "FinalOk" του Παραδείγματος 6, που υπολογίστηκε στο τέλος της Υπο-Ρουτίνας **DatabaseFind**:

ΜΗΤΡΑ "FinalOk"

16 Στήλες

| | | | | | | |
|--------------|-------|-----|---------|----------|-----------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | |
| Neighbour 1 | 20 | -64 | G350 | 38.11111 | 23.71111 | |
| Neighbour 2 | 30 | -66 | G220 | 38.22222 | 23.72222 | |
| Neighbour 3 | ERROR | | | | | |
| Neighbour 4 | 50 | -90 | G200 | 38.11100 | 23.71100 | |
| | CHN | RXL | Cell ID | Latitude | Longitude | |

Η Υπο-Ρουτίνα **GetFinalMatrix** εντοπίζει το "ERROR" στη τέταρτη σειρά και αναταξινομεί τη Μήτρα όπως φαίνεται παρακάτω (η καινούργια τιμή της μεταβλητής "DataMax" είναι 4):

ΜΗΤΡΑ "FinalOk"

16 Στήλες

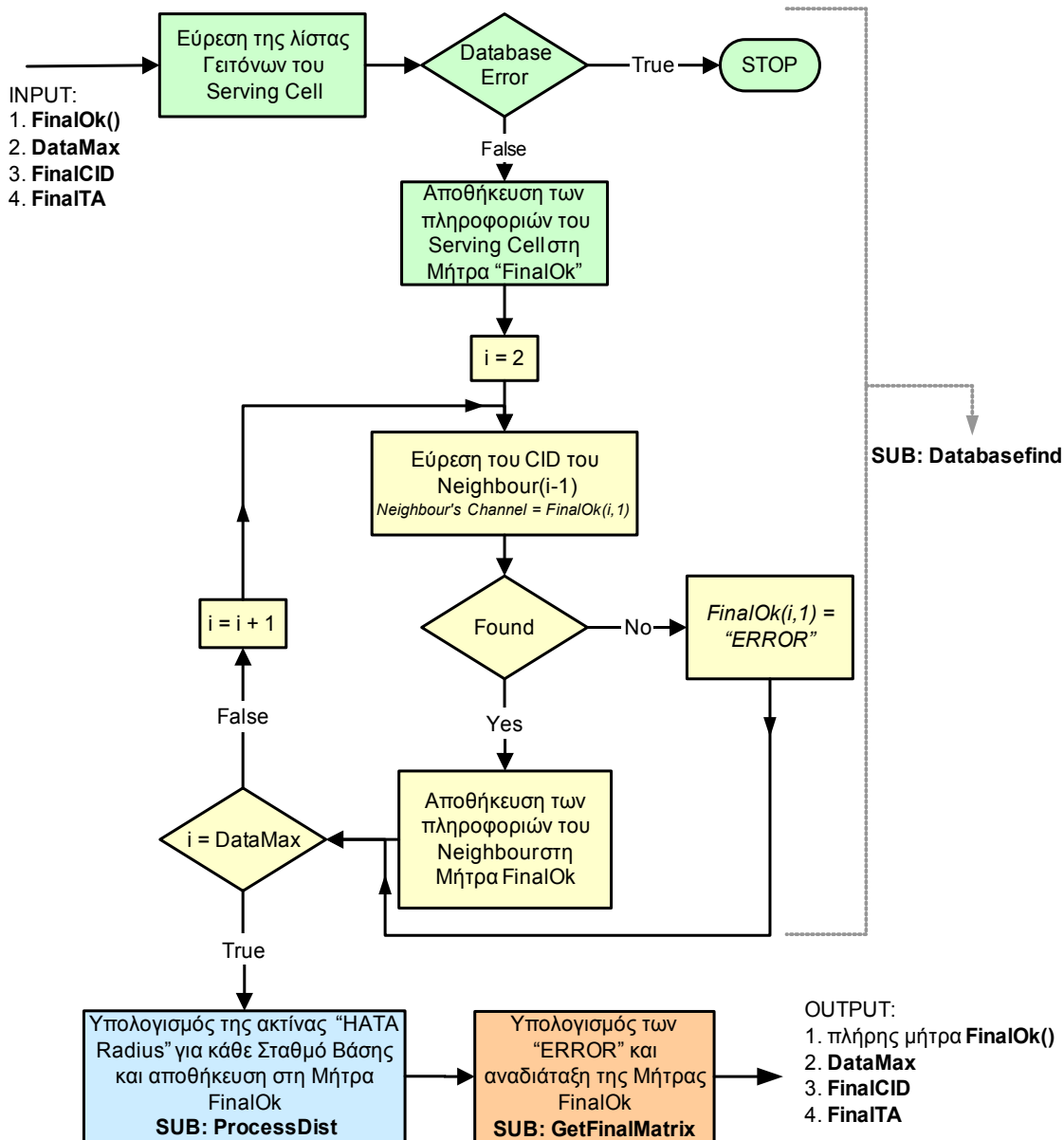
| | | | | | | |
|--------------|-----|-----|---------|----------|-----------|-------|
| Serving Cell | 10 | -51 | G100 | 38.11111 | 23.71111 | |
| Neighbour 1 | 20 | -64 | G350 | 38.11111 | 23.71111 | |
| Neighbour 2 | 30 | -66 | G220 | 38.22222 | 23.72222 | |
| Neighbour 3 | 50 | -90 | G200 | 38.11100 | 23.71100 | |
| | CHN | RXL | Cell ID | Latitude | Longitude | |

} "DataMax" = 4
Σειρές

4.4.2.f. PROCESS 2 - Διάγραμμα Ροής

Σ’ αυτό το σημείο έχει ολοκληρωθεί και η δεύτερη Διαδικασία (PROCESS 2) και το πρόγραμμα είναι έτοιμο να προχωρήσει στον υπολογισμό της εκτιμώμενης θέσης του Κ/Τ (PROCESS 3). Παρακάτω παρουσιάζουμε το διάγραμμα ροής της Αναζήτησης της Βάσης Δεδομένων (Σχήμα 4.10).

“PROCESS 2 - ΑΝΑΖΗΤΗΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ”



ΣΧΗΜΑ 4. 10

4.4.3. ΥΠΟΛΟΓΙΣΜΟΣ ΕΚΤΙΜΟΥΜΕΝΗΣ ΘΕΣΗΣ ΤΟΥ Κ/Τ (PROCESS 3)

Σ' αυτή τη Διαδικασία, ο αλγόριθμος πραγματοποιεί τον εντοπισμό του Κινητού Τερματικού (Κ/Τ). Οι παράμετροι που υπολογίζονται είναι:

1) Κατ' εκτίμηση συντεταγμένες του Κ/Τ

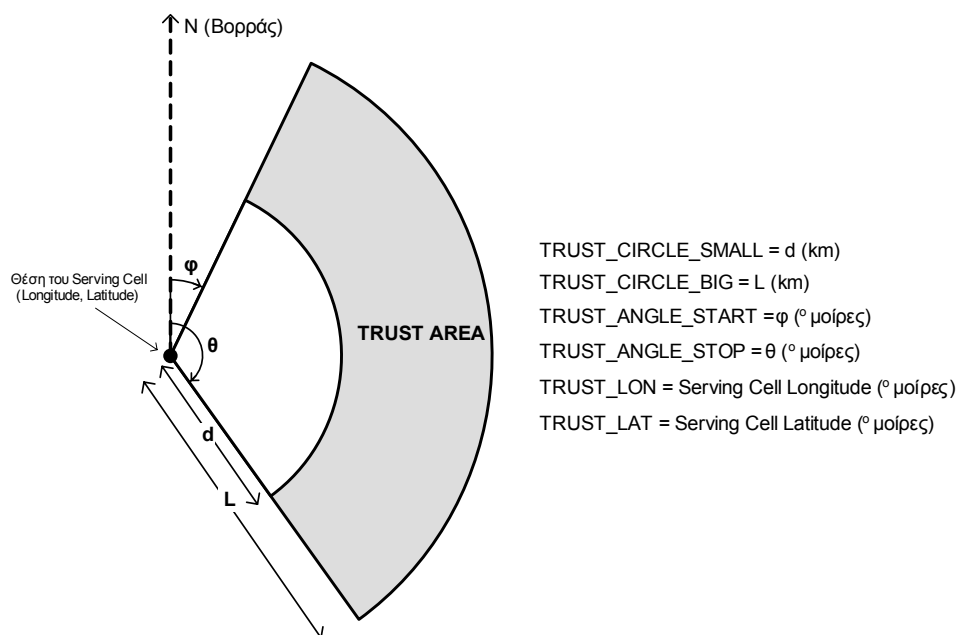
Πρόκειται για την θέση του Κ/Τ (Γεωγραφικό Μήκος/Πλάτος σε δεκαδική μορφή) η οποία υπολογίζεται από το πρόγραμμα, λαμβάνοντας υπόψη όλα τα διαθέσιμα στοιχεία. Η ακρίβεια αυτού του υπολογισμού εξαρτάται από πολυάριθμους παράγοντες, όπως είναι το περιβάλλον διάδοσης (πόλη, επαρχία, πυκνοκατοικημένη περιοχή, ρυμοτομία κλπ.), η ποιότητα των διαθέσιμων δεδομένων, η σωστά ενημερωμένη και ρυθμισμένη Βάση Δεδομένων των Σταθμών Βάσης, κλπ.

2) Περιοχή Εμπιστοσύνης (Trust Area)

Αναφέρεται στην περιοχή όπου, το Κ/Τ θα πρέπει να βρίσκεται, με χαμηλή πιθανότητα σφάλματος (μικρότερη από 5%). Αυτή η περιοχή καθορίζεται από την περιοχή κάλυψης του Serving Cell, σε συνάρτηση με την τιμή της δοσμένης κάθε φορά προόδου συγχρονισμού (TA). Κάθε τιμή του TA αντιστοιχεί και σε μια ορισμένη περιοχή). Οι παράμετροι που υπολογίζονται για να περιγράψουν την "Περιοχή Εμπιστοσύνης" (Trust Area) είναι οι ακόλουθες:

- TRUST_CIRCLE_SMALL
- TRUST_CIRCLE_BIG
- TRUST_ANGLE_START
- TRUST_ANGLE_STOP
- TRUST_LON
- TRUST_LAT

Η εξήγηση των ανωτέρω παραμέτρων δίνεται στο Σχήμα 4.11.



ΣΧΗΜΑ 4.11

Sub-Routines που χρησιμοποιούνται στην τρίτη διαδικασία (PROCESS 3)

Χρησιμοποιούνται οι ακόλουθες Sub-Routines (Υπο-Ρουτίνες):

1. ModSub.**MicroCheck (Function)**
2. ModSub.**NotLocate**
3. ModSub.**DrawMicro**
4. ModSub.**TAHATAMatch**
5. ModSub.**CheckNeighbourHata**
6. ModSub.**CalcDistances**
7. ModSub.**DrawGraph**

4.4.3.a. Συνάρτηση *MicroCheck*

Η συγκεκριμένη συνάρτηση (Function) επιστρέφει μια Boolean τιμή (TRUE/FALSE). Αν η μεταβλητή DataMax = 1 ή αν οποιοδήποτε BTS είναι χαρακτηρισμένο ως microCell (FinalOk(i, 9) > 0) τότε:

"MicroCheck = TRUE"

Σε κάθε άλλη περίπτωση:

"MicroCheck = FALSE"

Με βάση την τιμή της συνάρτησης **MicroCheck**, το πρόγραμμα εκτελεί και μία διαφορετική ομάδα από Υπο-Ρουτίνες:

MicroCheck = TRUE → ModSub.**NotLocate**

ModSub.**DrawMicro**

MicroCheck = FALSE → ModSub.**TAHATAMatch**

ModSub.**CalcDistances**

ModSub.**CheckNeighbourHata**

ModSub.**DrawGraph**

Αυτή η διαφοροποίηση γίνεται επειδή όταν **MicroCheck = TRUE**, δεν κρίνεται αναγκαίο να εφαρμοστεί ο πλήρης αλγόριθμος εντοπισμού θέσης. Η γεωγραφική θέση της Κυψέλης Εξυπηρέτησης, **Serving Cell** (στην περίπτωση όπου **DataMax = 1**) ή της *i*-οστής Κυψέλης (στην περίπτωση όπου **FinalOk(i, 9) = 0**), είναι από μόνη της αρκετή για να γίνει η εκτίμηση της θέσης του Κ/Τ.

4.4.3.b. Υπο-Ρουτίνα **NotLocate**

Αυτή η Υπο-Ρουτίνα εκτελείται μόνο όταν **MicroCheck = TRUE**. Εδώ, το πρόγραμμα παρακάμπτει τον αλγόριθμο εντοπισμού θέσης και θεωρεί την γεωγραφική θέση (γεωγραφικό πλάτος/ μήκος) του **Serving Cell** (ή του **Microcell**) ως την εκτιμώμενη θέση του Κ/Τ. Ας δούμε ποια είναι τα βήματα που ακολουθούνται σε αυτήν την Υπο-Ρουτίνα:

1. Εάν **FinalTA > 1** → **FinalTA = 1**.
2. Κατόπιν υπολογίζεται η θέση του Κινητού Τερματικού:
 Η μεταβλητή "**MicroBTS**" αναφέρεται στον Σταθμό Βάσης αναφοράς.
 π.χ.
 εάν "**DataMax = 1**" → "**MicroBTS = 1**" (**Serving Cell**)

 Αλλιώς, εάν "**FinalOk(i, 9) = 0**" → "**MicroBTS = i**"
 (**Serving Cell** εάν $i = 1$, **Neighbour (i-1)** εάν $i > 1$)

Ο αλγόριθμος κάνει την ακόλουθη υπόθεση:

Θέση Κ/Τ = Θέση MicroBTS

3. Τέλος, η Υπο-Ρουτίνα υπολογίζει την Περιοχή Εμπιστοσύνης (Trust Area):

TRUST_LON = γεωγραφικό μήκος του "MicroBTS" (= FinalOk(MicroBTS, 5))

TRUST_LAT = γεωγραφικό πλάτος του "MicroBTS" (= FinalOk(MicroBTS, 4))

TRUST_CIRCLE_BIG =

Εάν **MicroBTS = 1**: minimum {
 (0.55 * (FinalTA + 1) + 0.1)
 HATA Radius του "MicroBTS"
 (Hata Type = 0)

Εάν **MicroBTS > 1**: minimum {
 FinalOk(MicroBTS, 9)
 HATA Radius του "MicroBTS"
 (Hata Type = 0)

TRUST_CIRCLE_SMALL = 0

TRUST_ANGLE_START =

Γωνία έναρξης τη περιοχής κάλυψης του "MicroBTS"
 (FinalOk(MicroBTS, 15))

TRUST_ANGLE_STOP =

Γωνία τερματισμού της περιοχής κάλυψης του "MicroBTS"
 (FinalOk(MicroBTS, 16))

4.4.3.c. Υπο-Ρουτίνα *DrawMicro*

Αυτή η Υπο-Ρουτίνα καλείται μετά από την Υπο-Ρουτίνα **NotLocate** και πραγματοποιεί τον Γραφικό Σχεδιασμό των υπολογιζόμενων αποτελεσμάτων στη Φόρμα "**frmFinal**". Δεδομένου ότι αυτή η Υπο-Ρουτίνα δεν σχετίζεται με τον Αλγόριθμο Εντοπισμού Θέσης του Κ/Τ (χρησιμοποιείται μόνο προκειμένου να υπάρξει μια οπτική απεικόνιση των αποτελεσμάτων), δεν πρόκειται να περιγραφεί εδώ. Ο πλήρης κώδικας της συγκεκριμένης Υπο-Ρουτίνας δίνεται στο **ΠΑΡΑΡΤΗΜΑ Α**.

4.4.3.d. Υπο-Ρουτίνα *TAHATAMatch*

Αυτή η Υπο-Ρουτίνα εκτελείται μόνο όταν **MicroCheck** = FALSE. Καταρχήν, κανονικοποιείται η μεταβλητή "FinalTA" (εάν FinalTA > 1 → FinalTA = 1). Κατόπιν, πραγματοποιείται η επεξεργασία της ακτίνας "HATA Radius" της Κυψέλης Εξυπηρέτησης από κοινού με την περιοχή TA. Με λίγα λόγια, η Υπο-Ρουτίνα προσπαθεί να προσαρμόσει την ακτίνα "Hata Radius" του Serving Cell μέσα στην περιοχή του TA.

Η περιοχή του TA καθορίζεται από έναν κυκλικό δίσκο, με τα ακόλουθα χαρακτηριστικά:

- Κέντρο του κύκλου = γεωγραφικό μήκος / πλάτος της Κυψέλης Εξυπηρέτησης.
- Εξωτερική ακτίνα = $[0.55 \cdot (TA + 1)] + 0.1$ km
- Εσωτερική ακτίνα = $[0.55 \cdot (TA)] - 0.275 \cdot (TA)$ km

Παρακάτω, καταδεικνύουμε τα βήματα που ακολουθούνται από την Υπο-Ρουτίνα **TAHATAMatch**:

ΒΗΜΑ 1:

Εάν FinalTA > 1 → FinalTA = 1.

Αυτό γίνεται, επειδή μέσα σε ένα αστικό περιβάλλον (μικρή / μεγάλη πόλη) όπου υπάρχει μεγάλη συγκέντρωση Σταθμών Βάσης, τιμές μεγαλύτερες από τη μονάδα για την Πρόοδο Συγχρονισμού δεν έχουν κάποιο σημαντικό νόημα (αφού, σύμφωνα με μετρήσεις, η συχνότητα εμφάνισης του TA > 1 είναι 7-10%).

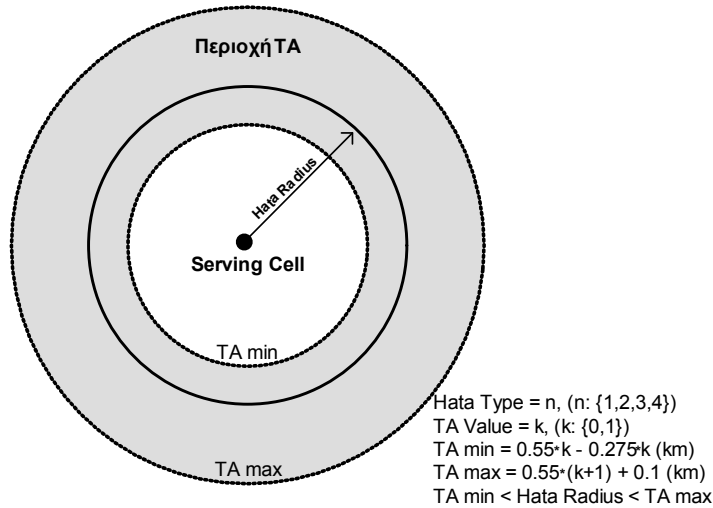
ΒΗΜΑ 2:

Προσπάθεια προσαρμογής της ακτίνας "HATA Radius" μέσα στην περιοχή του TA.

Υπάρχουν τρεις περιπτώσεις σ’ αυτό το βήμα:

⇒ **Περίπτωση 1:** Η ακτίνα “HATA Radius” ικανοποιεί την ακόλουθη ανισότητα (Σχήμα 4.12):

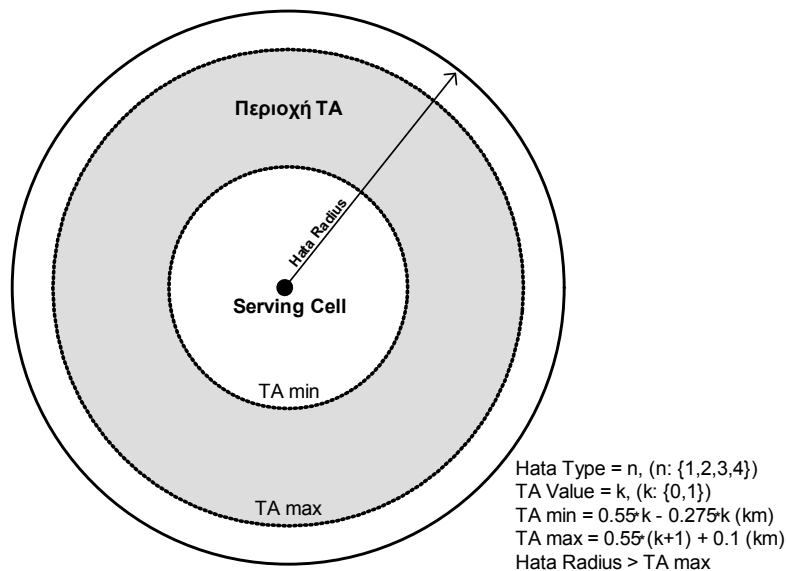
$$[0.55 \cdot (TA)] - 0.275 \cdot (TA) < \text{HATA Radius} < [0.55 \cdot (TA + 1)] + 0.1$$



ΣΧΗΜΑ 4.12

Σε αυτήν την περίπτωση κανένα επιπλέον μέτρο δεν λαμβάνεται, δεδομένου ότι η ακτίνα “HATA Radius” είναι ήδη μέσα στην περιοχή του TA.

⇒ **Περίπτωση 2:** Ακτίνα “HATA Radius” > [0.55·(TA+1)]+0.1 (Σχήμα 4.13):

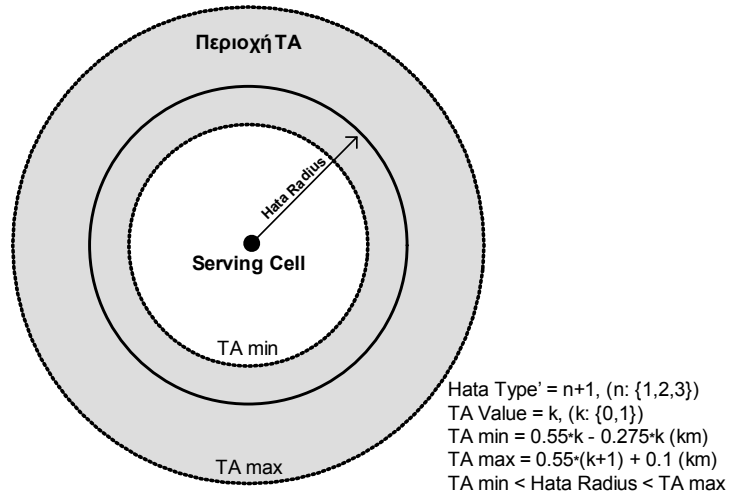


ΣΧΗΜΑ 4.13

Εδώ, υπολογίζουμε μια νέα ακτίνα “HATA Radius” που αντιστοιχεί σε Hata Type = n + 1 (n = προηγούμενο Hata Type), υπό τον περιορισμό ότι n ≤ 3.

Το βήμα επαναλαμβάνεται μέχρι:

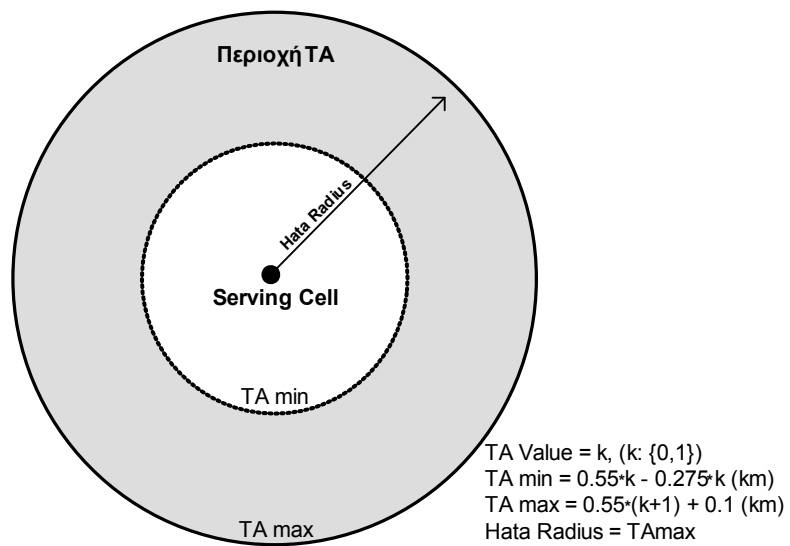
- Ακτίνα HATA < $[0.55 \cdot (TA + 1)] + 0.1$ (Σχήμα 4.14), ή



ΣΧΗΜΑ 4.14

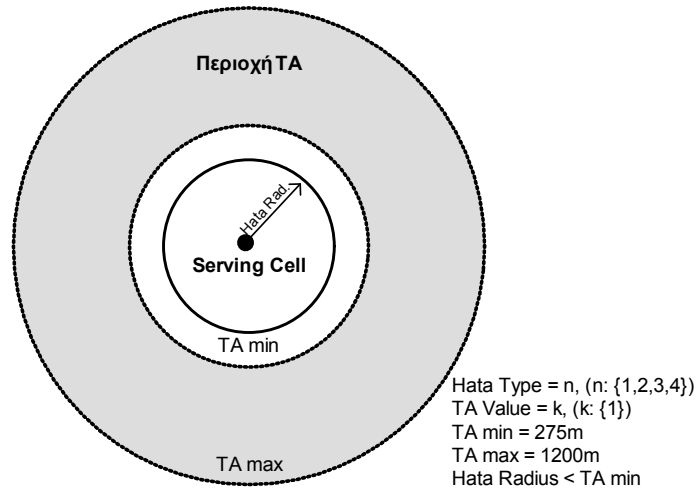
- n = 4 και η ακτίνα "HATA Radius" είναι ακόμη > $[0.55 \cdot (TA + 1)] + 0.1$. Σ' αυτήν την περίπτωση θέτουμε:

HATA Radius = TAmaz (Σχήμα 4.15)



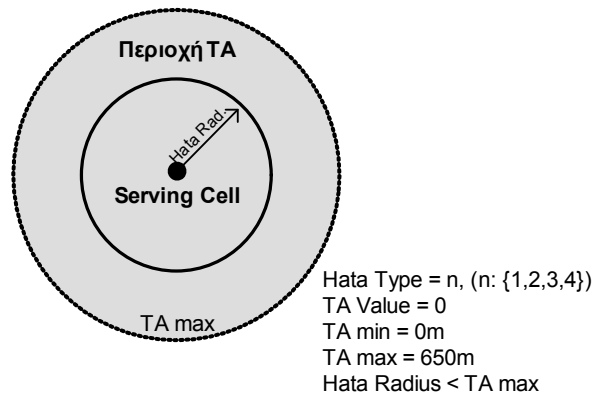
ΣΧΗΜΑ 4.15

⇒ **Περίπτωση 3:** Η ακτίνα "HATA Radius" < $[0.55 \cdot (TA)] - 0.275 \cdot (TA)$ (Σχήμα 4.16)



ΣΧΗΜΑ 4.16

Όταν συμβαίνει κάτι τέτοιο, τότε θέτουμε την μεταβλητή FinalTA = 0, έτσι ώστε η ακτίνα "HATA Radius" να περιέχεται μέσα στη καινούργια περιοχή TA Area (Σχήμα 4.17).



ΣΧΗΜΑ 4.17

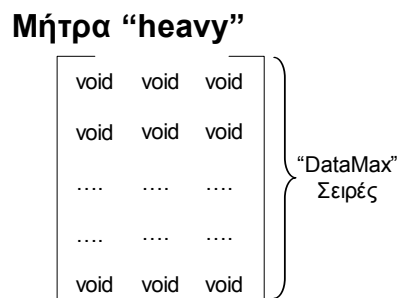
4.4.3.e. Υπο-Ρουτίνα *CheckNeighbourHata*

Η συγκεκριμένη Υπο-Ρουτίνα επεξεργάζεται την ακτίνα "HATA Radius" των γειτόνων (Neighbours) και υπολογίζει τη μήτρα "heavy", που χρησιμοποιείται από την Υπο-Ρουτίνα "CalcDistances".

Κρίσιμα Δεδομένα:

- X₀, Y₀: Οι συντεταγμένες της Κυψέλης Εξυπηρέτησης
- X₁, Y₁: Οι συντεταγμένες της Γειτονικής Κυψέλης
- R₀: Η ακτίνα "HATA Radius" της κυψέλης εξυπηρέτησης
- R₁: Η ακτίνα "HATA Radius" της γειτονικής κυψέλης
- Start_Sector: Η γωνία "Start Sector" της κυψέλης εξυπηρέτησης
- Stop_Sector: Η γωνία "Stop Sector" της κυψέλης εξυπηρέτησης

Μήτρα heavy():



Διαστάσεις Μήτρας = "**DataMax**" Σειρές x **3** Στήλες:

Κάθε σειρά της Μήτρας αντιστοιχεί σε έναν Σταθμό Βάσης:

Σειρά 1 → Κυψέλη Εξυπηρέτησης

Σειρά 2 → Γειτονική Κυψέλη "Neighbour" 1

.....

Σειρά (DataMax) → Γειτονική Κυψέλη "Neighbour" (DataMax -1)

Distance: Η απόσταση μεταξύ της Κυψέλης Εξυπηρέτησης και της Γειτονικής Κυψέλης.

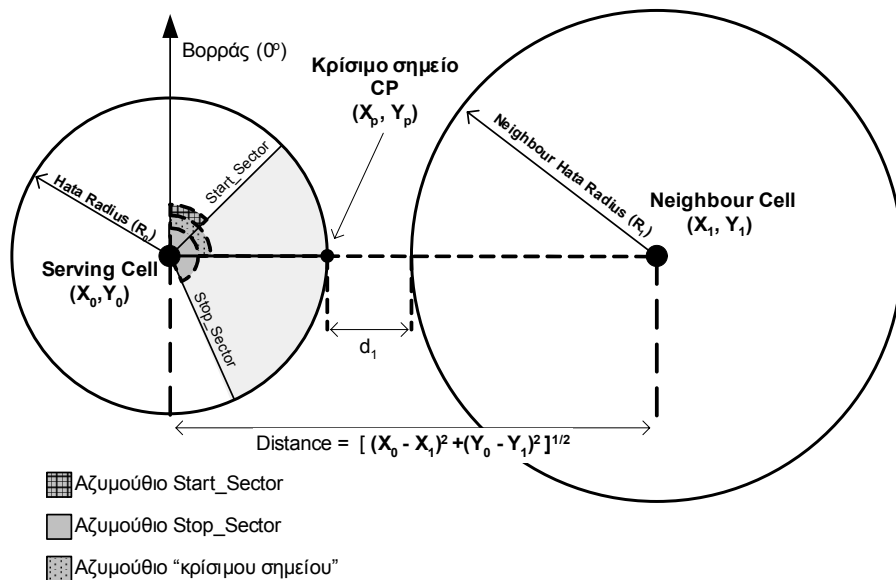
$$\text{Distance} = \sqrt{(X_0 - X_1)^2 + (Y_0 - Y_1)^2}$$

Η ακόλουθη εργασία εκτελείται για κάθε Γείτονα:

ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ:

Είναι τέσσερις οι δυνατές περιπτώσεις για τις οποίες ενδιαφερόμαστε:

⇒ **ΠΕΡΙΠΤΩΣΗ 1: Distance > R₀ + R₁**



ΣΧΗΜΑ 4.18

Όπως φαίνεται στο Σχήμα 4.18, το κρίσιμο σημείο CP(X_p, Y_p) είναι το σημείο εκείνο όπου η γραμμή που συνδέει την Κυψέλη Εξυπηρέτησης με την Γειτονική Κυψέλη, τέμνει τον κύκλο που ορίζεται από την ακτίνα "HATA Radius" της Κυψέλης Εξυπηρέτησης.

Το πρόγραμμα ελέγχει αν το αζιμούθιο του κρίσιμου σημείου (CP) βρίσκεται ανάμεσα στην περιοχή κάλυψης της Κυψέλης Εξυπηρέτησης ($\text{Start_Sector} < \text{Αζιμούθιο CP} < \text{Stop_Sector}$).

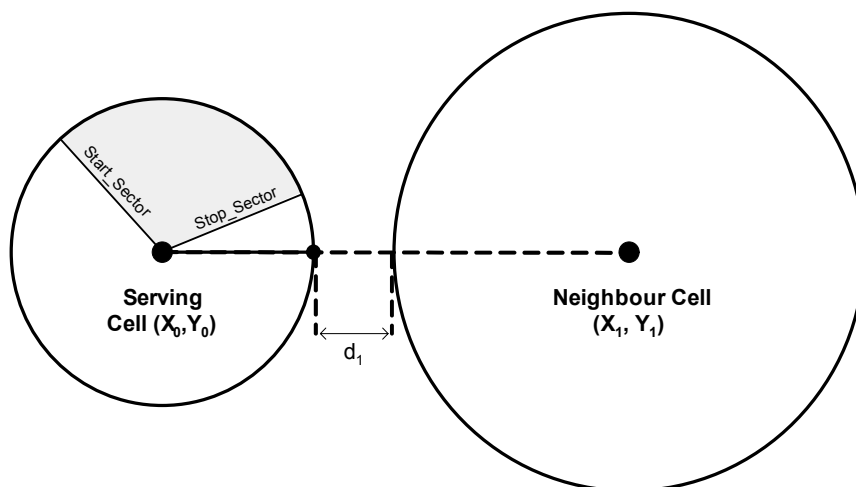
⇒ Αν αυτό είναι αληθές (Σχήμα 4.18), τότε η μήτρα "heavy" συμπληρώνεται με τα ακόλουθα στοιχεία:

heavy(i, 1) = x-συντεταγμένη του κρίσιμου σημείου CP (X_p)

heavy(i, 2) = y- συντεταγμένη του κρίσιμου σημείου CP (Y_p)

heavy(i, 3) = $\frac{10^{-3}}{d_1}$, $d_1 = \text{Distance} - R_0 - R_1$

και το πρόγραμμα προχωράει στην επόμενη Γειτονική Κυψέλη "Neighbour" ($i = i + 1$)



ΣΧΗΜΑ 4.19

↪ Αν το κρίσιμο σημείο CP βρίσκεται εξωτερικά της περιοχής κάλυψης της Κυψέλης Εξυπηρέτησης (Σχήμα 4.19), τότε:

- Αν Neighbour HATA type > 1 τότε:

$$(\text{HATA type})' = (\text{HATA type}) - 1$$

Το πρόγραμμα υπολογίζει την καινούργια ακτίνα "HATA Radius" της γειτονικής κυψέλης (που αντιστοιχεί σε $(\text{HATA type})'$) και ξανατρέχει την "**ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ**" με την καινούργια τώρα ακτίνα (μέγιστος αριθμός επαναλήψεων = 3).

- Αν Neighbour HATA type = 1 τότε:

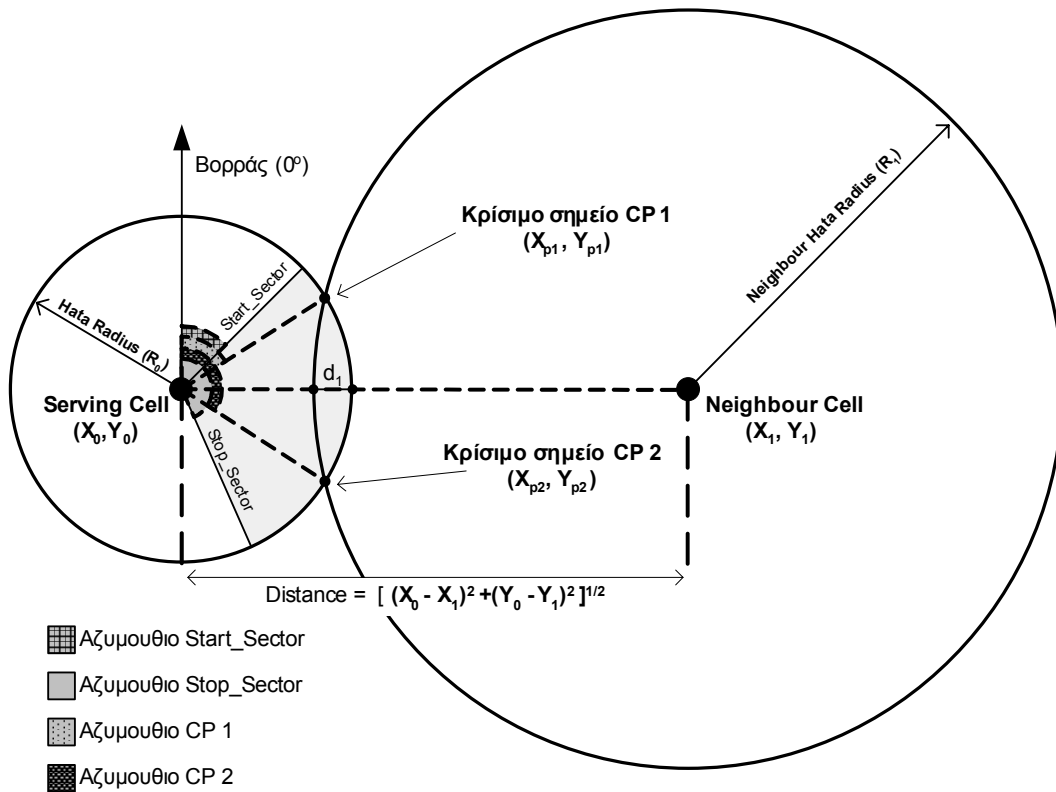
$$\text{heavy}(i, 1) = 0$$

$$\text{heavy}(i, 2) = 0$$

$$\text{heavy}(i, 3) = 0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i+1$).

⇒ ΠΕΡΙΠΤΩΣΗ 2: $Distance > R_1$ & $Distance < R_0 + R_1$



ΣΧΗΜΑ 4.20

Εδώ όπως βλέπουμε έχουμε δύο κρίσιμα σημεία CP1 (X_{p1}, Y_{p1}) και CP2 (X_{p2}, Y_{p2}), τα οποία αντιστοιχούν στα σημεία τομής των δύο κύκλων. Ο αλγόριθμος ελέγχει αν κάποιο από αυτά τα δύο σημεία ή αν και τα δύο ανήκουν στην περιοχή κάλυψης της Κυψέλης Εξυπηρέτησης.

Πιθανές Καταστάσεις:

↪ Και τα δύο κρίσιμα σημεία βρίσκονται μέσα στην περιοχή κάλυψης (Σχήμα 4.20):

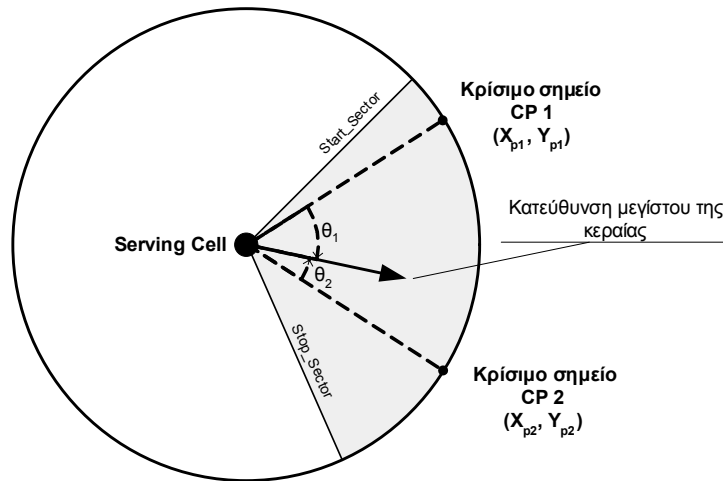
Τότε το πρόγραμμα επιλέγει το σημείο εκείνο που έχει τη μικρότερη αζιμουθιακή διαφορά από την διεύθυνση μεγίστου της κεραίας της Κυψέλης Εξυπηρέτησης (Σχήμα 4.21). Με βάση αυτό το σημείο (πρωτεύων κρίσιμο σημείο), ο αλγόριθμος συμπληρώνει τη μήτρα "heavy" με τα ακόλουθα στοιχεία:

$$heavy(i, 1) = x\text{-συντεταγμένη του πρωτεύοντος κρίσιμου σημείου } (X_{p1} \text{ ή } X_{p2})$$

$$heavy(i, 2) = y\text{-συντεταγμένη του πρωτεύοντος κρίσιμου σημείου } (Y_{p1} \text{ ή } Y_{p2})$$

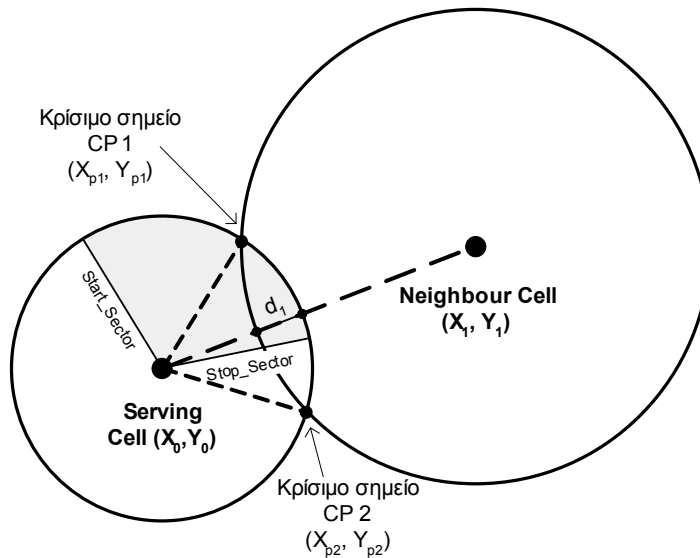
$$\text{heavy}(i, 3) = \frac{10^{-3}}{d_1}, \quad d_1 = \text{Abs}(\text{Distance} - R_0 - R_1)$$

Έπειτα, το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).



ΣΧΗΜΑ 4.21

☞ Μόνο το ένα από τα δύο κρίσιμα σημεία βρίσκεται μέσα στην περιοχή κάλυψης (Σχήμα 4.22):



ΣΧΗΜΑ 4. 22

Το πρωτεύων κρίσιμο σημείο θα είναι αυτό το οποίο περιέχεται μέσα στην περιοχή κάλυψης. Ο αλγόριθμος βασιζόμενος σ' αυτό το σημείο συμπληρώνει τη μήτρα "heavy" με τα ακόλουθα στοιχεία:

heavy(i, 1) = x- συντεταγμένη του πρωτεύοντος κρίσιμου σημείου (X_{p1} ή X_{p2})

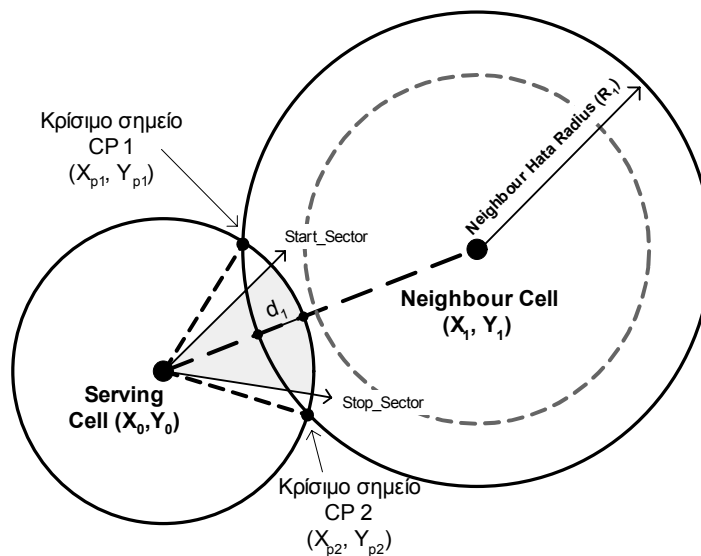
heavy(i, 2) = y- συντεταγμένη του πρωτεύοντος κρίσιμου σημείου (Y_{p1} ή Y_{p2})

$$\text{heavy}(i, 3) = \frac{8 \cdot 10^{-4}}{d_1}, \quad d_1 = \text{Abs}(\text{Distance} - R_0 - R_1)$$

Έπειτα, το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).

☞ Κανένα από τα δύο κρίσιμα σημεία δεν βρίσκεται μέσα στην περιοχή κάλυψης:

Τώρα ο αλγόριθμος θα πρέπει να αποφασίσει για τον αν θα πάρει υψηλότερη ή χαμηλότερη τιμή για το HATA type της Γειτονικής Κυψέλης.



ΣΧΗΜΑ 4.23



Αν η περιοχή κάλυψης της κυψέλης εξυπηρέτησης βρίσκεται ενδιάμεσα στα δύο κρίσιμα σημεία (κρίσιμο σημείο CP 1 → Ωρολογιακή φορά → κρίσιμο σημείο CP 2) όπως στο Σχήμα 4.23, τότε το πρόγραμμα αναθέτει στην μεταβλητή HATA type την αμέσως υψηλότερη τιμή. Έτσι τώρα, η καινούργια και μικρότερη από την προηγούμενη ακτίνα "HATA Radius" (ο κύκλος με την διακεκομμένη γραμμή) θα έχει πολλές πιθανότητες να έχει τουλάχιστον ένα κρίσιμο σημείο μέσα στην περιοχή κάλυψης:

- Αν Neighbour HATA type < 4 τότε:

$$(\text{HATA type})' = (\text{HATA type}) + 1$$

Το πρόγραμμα υπολογίζει την καινούργια ακτίνα HATA Radius της γειτονικής κυψέλης (που αντιστοιχεί στο $(\text{HATA type})'$) και ξανατρέχει την **ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ** με την καινούργια τώρα ακτίνα (μέγιστος αριθμός επαναλήψεων = 3).

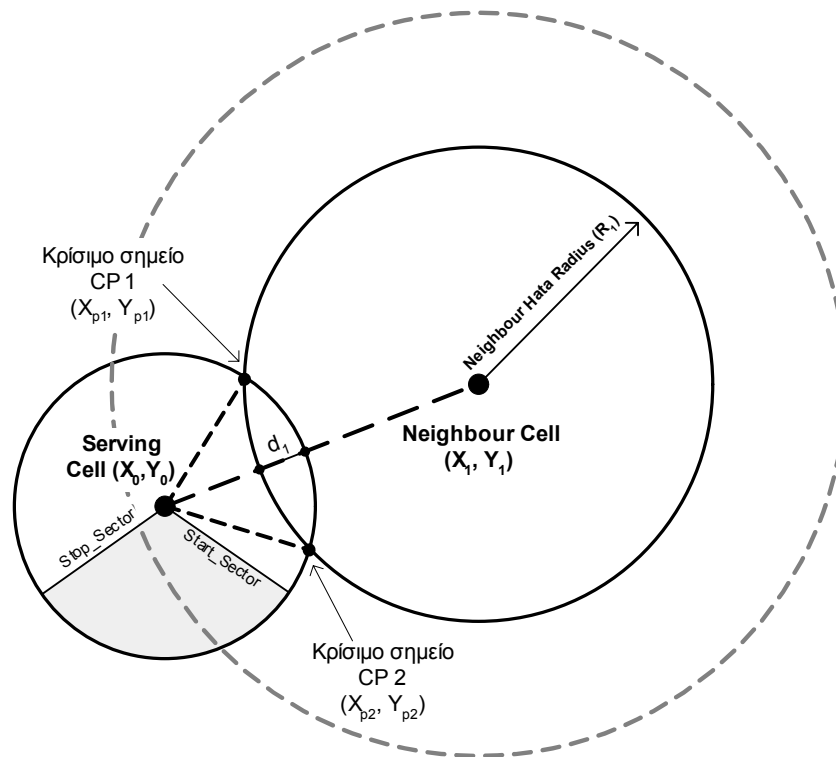
- Εάν Neighbour HATA type = 4 τότε:

$$\text{heavy}(i, 1) = 0$$

$$\text{heavy}(i, 2) = 0$$

$$\text{heavy}(i, 3) = 0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).



ΣΧΗΜΑ 4.24



Από την άλλη μεριά, αν η περιοχή κάλυψης της κυψέλης εξυπηρέτησης δεν βρίσκεται ενδιάμεσα στα δύο κρίσιμα σημεία (κρίσιμο σημείο CP 1 → Ωρολογιακή φορά → κρίσιμο σημείο CP 2) όπως στο Σχήμα 4.24, τότε το πρόγραμμα λαμβάνει την αμέσως μικρότερη τιμή για την μεταβλητή HATA type. Έτσι τώρα, η καινούργια και μεγαλύτερη από την προηγούμενη ακτίνα "HATA Radius" (ο κύκλος με την διακεκομμένη γραμμή) θα έχει πολλές πιθανότητες να έχει τουλάχιστον ένα κρίσιμο σημείο μέσα στην περιοχή κάλυψης:

- Αν Neighbour HATA type > 1 τότε:

$$(\text{HATA type})' = (\text{HATA type}) - 1$$

Το πρόγραμμα υπολογίζει την καινούργια ακτίνα "HATA Radius" της γειτονικής κυψέλης (που αντιστοιχεί στο $(\text{HATA type})'$) και ξανατρέχει την "**ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ**" με την καινούργια τώρα ακτίνα (μέγιστος αριθμός επαναλήψεων = 3).

- Αν Neighbour HATA type = 1 τότε:

$$\text{heavy}(i, 1) = 0$$

$$\text{heavy}(i, 2) = 0$$

$$\text{heavy}(i, 3) = 0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).

⇒ **ΠΕΡΙΠΤΩΣΗ 3: Distance < R₁ & Distance + R₀ > R₁**

Έτσι όπως και στην Περίπτωση 2, έχουμε δύο κρίσιμα σημεία CP1 (X_{p1}, Y_{p1}) και CP2 (X_{p2}, Y_{p2}), τα οποία αντιστοιχούν στα σημεία τομής των δύο κύκλων. Ο αλγόριθμος ελέγχει αν κάποιο από αυτά τα δύο σημεία ή αν και τα δύο ανήκουν στην περιοχή κάλυψης της Κυψέλης Εξυπηρέτησης.

Πιθανές Καταστάσεις:

☞ Και τα δύο κρίσιμα σημεία βρίσκονται μέσα στην περιοχή κάλυψης (Σχήμα 4.25):

Τότε το πρόγραμμα επιλέγει το σημείο εκείνο που έχει τη μικρότερη αζυμουθιακή διαφορά με την διεύθυνση μεγίστου της κεραίας της Κυψέλης Εξυπηρέτησης (Σχήμα 4.21). Με βάση αυτό το σημείο

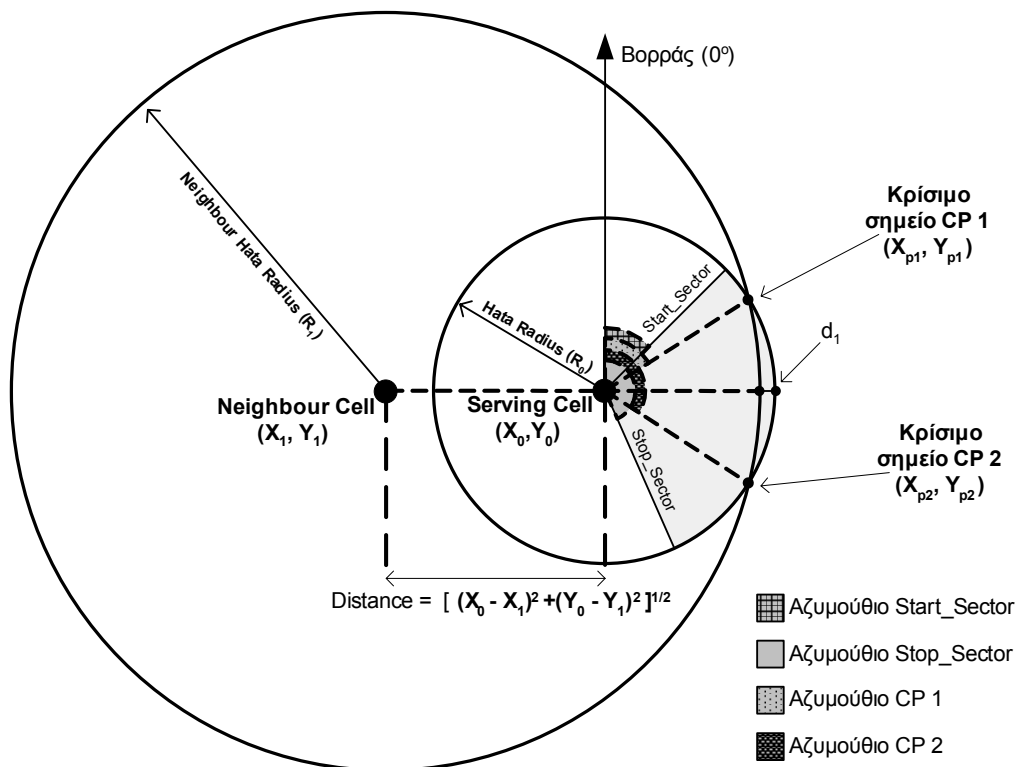
(πρωτεύων κρίσιμο σημείο), ο αλγόριθμος συμπληρώνει τη μήτρα "heavy" με τα ακόλουθα στοιχεία:

$$\text{heavy}(i, 1) = x\text{- συντεταγμένη του πρωτεύοντος κρίσιμου σημείου } (X_{p1} \text{ ή } X_{p2})$$

$$\text{heavy}(i, 2) = y\text{- συντεταγμένη του πρωτεύοντος κρίσιμου σημείου } (Y_{p1} \text{ ή } Y_{p2})$$

$$\text{heavy}(i, 3) = \frac{10^{-3}}{d_1}, \quad d_1 = \text{Distance} + R_0 - R_1$$

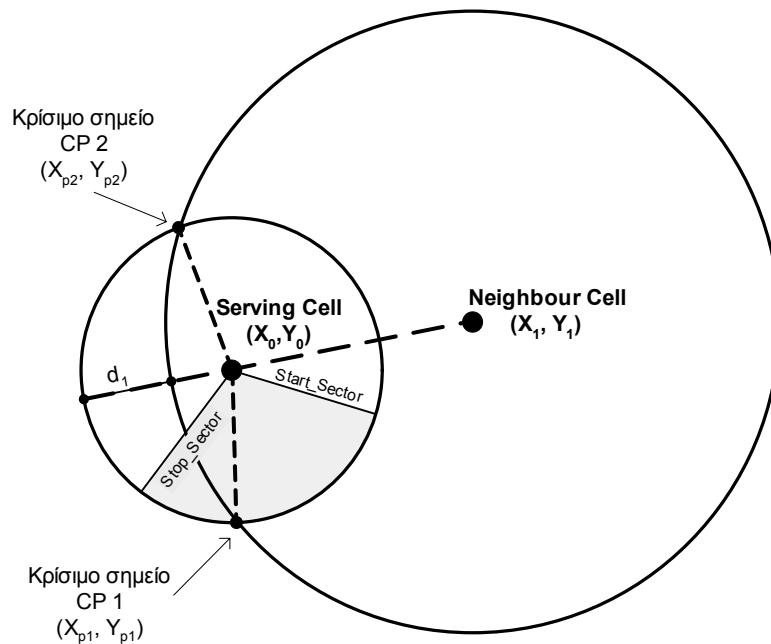
Έπειτα το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).



ΣΧΗΜΑ 4.25

☞ Μόνο το ένα από τα δύο κρίσιμα σημεία βρίσκεται μέσα στην περιοχή κάλυψης (Σχήμα 4.26):

Το πρωτεύων κρίσιμο σημείο θα είναι αυτό το οποίο περιέχεται μέσα στην περιοχή κάλυψης. Ο αλγόριθμος βασιζόμενος σ' αυτό το σημείο συμπληρώνει τη μήτρα "heavy" με τα ακόλουθα στοιχεία:



ΣΧΗΜΑ 4.26

heavy(i, 1) = x- συντεταγμένη του πρωτεύοντος κρίσιμου σημείου (X_{p1} ή X_{p2})

heavy(i, 2) = y- συντεταγμένη του πρωτεύοντος κρίσιμου σημείου (Y_{p1} ή Y_{p2})

$$\text{heavy}(i, 3) = \frac{8 \cdot 10^{-4}}{d_1}, \quad d_1 = \text{Distance} + R_0 - R_1$$

Έπειτα το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" (i = i + 1).

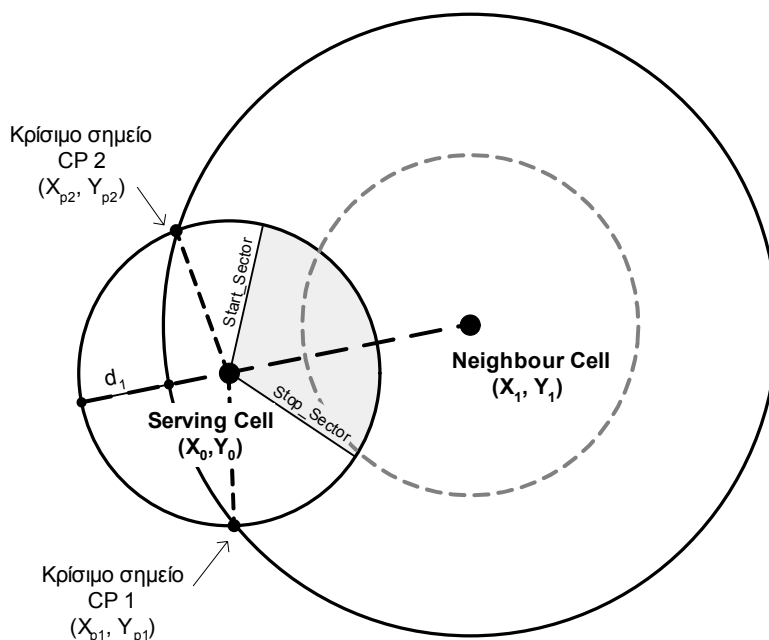
☞ *Κανένα από τα δύο κρίσιμα σημεία δεν βρίσκεται μέσα στην περιοχή κάλυψης:*

Τώρα ο αλγόριθμος θα πρέπει να αποφασίσει για τον αν θα πάρει υψηλότερη ή χαμηλότερη τιμή για το HATA type της Γειτονικής Κυψέλης.



αν η περιοχή κάλυψης της κυψέλης εξυπηρέτησης δεν βρίσκεται ενδιάμεσα στα δύο κρίσιμα σημεία (κρίσιμο σημείο CP 1 → Ωρολογιακή φορά → κρίσιμο σημείο CP 2) όπως στο Σχήμα 4.27, τότε το πρόγραμμα λαμβάνει την αμέσως μεγαλύτερη τιμή για την μεταβλητή HATA type. Έτσι τώρα, η καινούργια και μικρότερη από την προηγούμενη ακτίνα "HATA Radius" (ο

κύκλος με την διακεκομμένη γραμμή) θα έχει πολλές πιθανότητες να έχει τουλάχιστον ένα κρίσιμο σημείο μέσα στην περιοχή κάλυψης:



ΣΧΗΜΑ 4.27

- Αν Neighbour HATA type < 4 τότε:

$$(\text{HATA type})' = (\text{HATA type}) + 1$$

Το πρόγραμμα υπολογίζει την καινούργια ακτίνα "HATA Radius" της γειτονικής κυψέλης (που αντιστοιχεί στο (HATA type)') και ξανατρέχει την **"ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ"** με την καινούργια τώρα ακτίνα (μέγιστος αριθμός επαναλήψεων = 3).

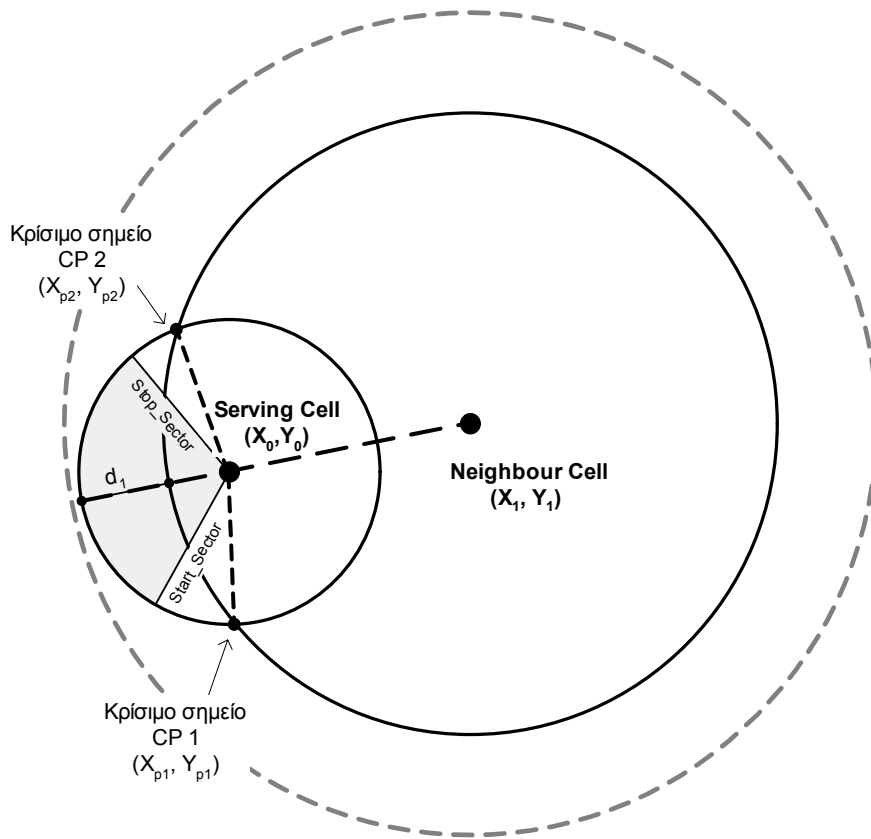
- Αν Neighbour HATA type = 4 τότε:

$$\text{heavy}(i, 1) = 0$$

$$\text{heavy}(i, 2) = 0$$

$$\text{heavy}(i, 3) = 0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).



ΣΧΗΜΑ 4.28



Εάν όμως η περιοχή κάλυψης της κυψέλης εξυπηρέτησης βρίσκεται ενδιάμεσα στα δύο κρίσιμα σημεία (κρίσιμο σημείο CP 1 → Ωρολογιακή φορά → κρίσιμο σημείο CP 2) όπως στο Σχήμα 4.28, τότε το πρόγραμμα λαμβάνει την αμέσως μικρότερη τιμή για την μεταβλητή HATA type. Έτσι τώρα, η καινούργια και μεγαλύτερη από την προηγούμενη ακτίνα "HATA Radius" (ο κύκλος με την διακεκομμένη γραμμή) θα έχει πολλές πιθανότητες να έχει τουλάχιστον ένα κρίσιμο σημείο μέσα στην περιοχή κάλυψης:

- Αν Neighbour HATA type > 1 τότε:

$$(\text{HATA type})' = (\text{HATA type}) - 1$$

Το πρόγραμμα υπολογίζει την καινούργια ακτίνα HATA Radius της γειτονικής κυψέλης (που αντιστοιχεί στο $(\text{HATA type})'$) και ξανατρέχει την "**ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ**" με την καινούργια τώρα ακτίνα (μέγιστος αριθμός επαναλήψεων = 3).

- Αν Neighbour HATA type = 1 τότε:

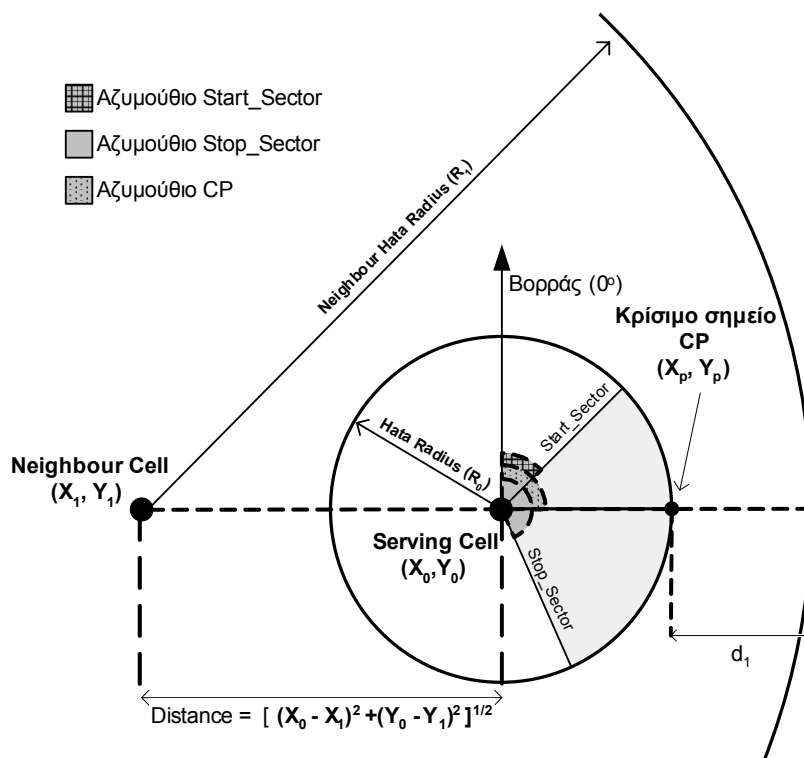
$$\text{heavy}(i, 1) = 0$$

$$\text{heavy}(i, 2) = 0$$

$$\text{heavy}(i, 3) = 0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).

⇒ **ΠΕΡΙΠΤΩΣΗ 4: $\text{Distance} + R_0 < R_1$**



ΣΧΗΜΑ 4.29

Όπως φαίνεται στο Σχήμα 4.29, το κρίσιμο σημείο CP (X_p, Y_p) είναι το σημείο εκείνο όπου η προέκταση της γραμμής που συνδέει την Κυψέλη Εξυπηρέτησης με την Γειτονική Κυψέλη, τέμνει τον κύκλο που ορίζεται από την ακτίνα "HATA Radius" της Κυψέλης Εξυπηρέτησης.

Το πρόγραμμα ελέγχει αν το αζιμούθιο του κρίσιμου σημείου (CP) βρίσκεται μέσα στην περιοχή κάλυψης της Κυψέλης Εξυπηρέτησης ($\text{Start_Sector} < \text{Αζιμούθιο CP} < \text{Stop_Sector}$).

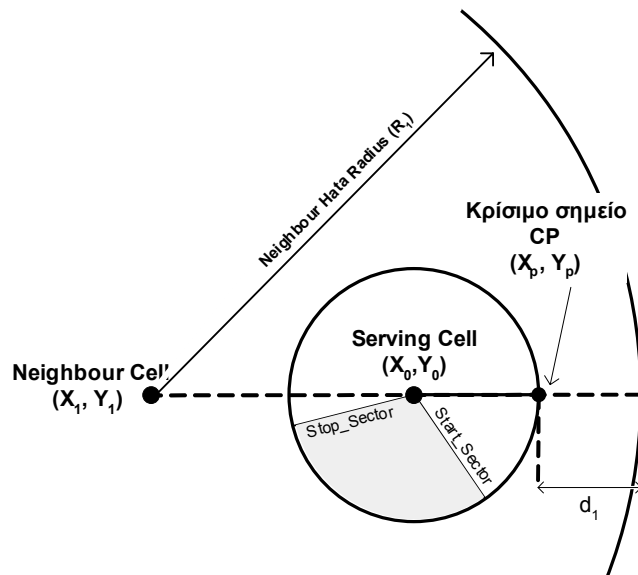
- ↳ Αν αυτό είναι αληθές (Σχήμα 4.29), τότε η μήτρα "heavy" συμπληρώνεται με τα ακόλουθα στοιχεία:

$$\text{heavy}(i, 1) = \text{x-συντεταγμένη του κρίσιμου σημείου CP } (X_p)$$

$$\text{heavy}(i, 2) = \text{y- συντεταγμένη του κρίσιμου σημείου CP } (Y_p)$$

$$\text{heavy}(i, 3) = \frac{10^{-3}}{d_1}, \quad d_1 = R_1 - \text{Distance} - R_0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i + 1$).



ΣΧΗΜΑ 4.30

- ↳ Αν το κρίσιμο σημείο CP βρίσκεται εξωτερικά της περιοχής κάλυψης της Κυψέλης Εξυπηρέτησης (Σχήμα 4.30), τότε:

- Αν Neighbour HATA type < 4 τότε:

$$(\text{HATA type})' = (\text{HATA type}) + 1$$

Το πρόγραμμα υπολογίζει την καινούργια ακτίνα "HATA Radius" της γειτονικής κυψέλης (που αντιστοιχεί σε $(\text{HATA type})'$) και ξανατρέχει την **"ΕΠΕΞΕΡΓΑΣΙΑ ΓΕΙΤΟΝΙΚΩΝ ΚΥΨΕΛΩΝ"** με την καινούργια τώρα ακτίνα (μέγιστος αριθμός επαναλήψεων = 3).

- Αν Neighbour HATA type = 4 τότε:

$$\text{heavy}(i, 1) = 0$$

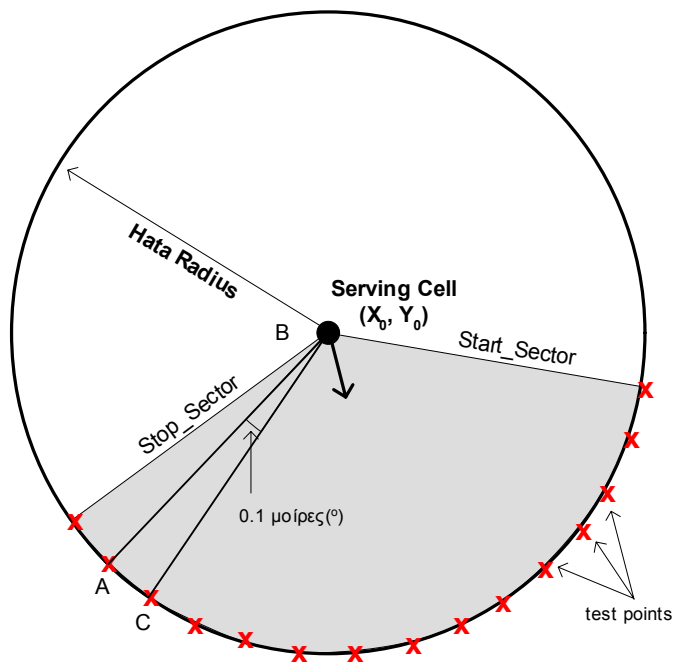
$$\text{heavy}(i, 2) = 0$$

$$\text{heavy}(i, 3) = 0$$

και το πρόγραμμα προχωράει στην επόμενη γειτονική κυψέλη "Neighbour" ($i = i+1$).

4.4.3.f. Υπο-Ρουτίνα *CalcDistances*

Αυτή η Υπο-Ρουτίνα υπολογίζει τη θέση του Κ/Τ. Ο αλγόριθμος υποθέτει ότι η θέση του κινητού τερματικού βρίσκεται κάπου επάνω στην ακτίνα "HATA Radius" της Κυψέλης Εξυπηρέτησης και μέσα στην περιοχή κάλυψης. Επομένως, παίρνει ως σημεία δοκιμής (test points) τα σημεία που παρουσιάζονται στο Σχήμα 4.31.



ΣΧΗΜΑ 4.31

Η γωνία που σχηματίζουν δύο διαδοχικά test points και η θέση της κυψέλης εξυπηρέτησης είναι:

$$(\hat{A B C}) = 0.1 \text{ μοίρες } (^{\circ})$$

Κατά συνέπεια, η απόσταση μεταξύ δύο διαδοχικών test points είναι:

$$d = R_0 \cdot 1.75 \cdot 10^{-3} \text{ m}$$

πχ. αν $R_0 = 500\text{m} \rightarrow d = 0.875 \text{ m}$

Προκειμένου να αποφασίσει ο αλγόριθμος ποιο από τα test points πρόκειται να ληφθεί ως η πιθανότερη θέση του K/T, χρησιμοποιούμε την ακόλουθη συνάρτηση:

$$\text{Θέση του K/T} = [\text{θέση του Test Point}(k), k: D(k) = \min\{D(1), D(2), \dots, D(j), D(N)\}]$$

N: ο αριθμός των Test Points

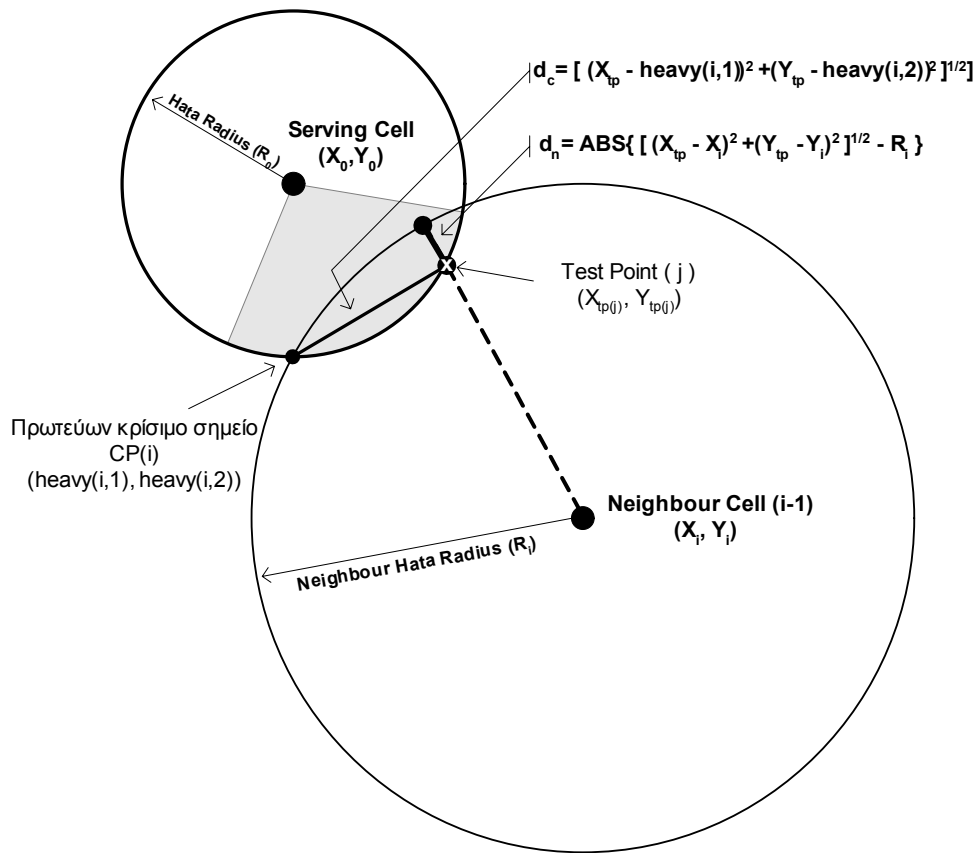
$$D(j) = \sum_{i=2}^{\text{DataMax}} \text{heavy}(i, 3) \cdot S(i, j)$$

$$S(i, j) = d_c(i, j) + d_n(i, j)$$

$$d_c(i, j) = \sqrt{(X_{\text{tp}(j)} - \text{heavy}(i, 1))^2 + (Y_{\text{tp}(j)} - \text{heavy}(i, 2))^2} \quad (\text{Σχήμα 4.32})$$

$$d_n(i, j) = \left| \sqrt{(X_{\text{tp}(j)} - X_i)^2 + (Y_{\text{tp}(j)} - Y_i)^2} - R_i \right| \quad (\text{Σχήμα 4.32})$$

Η κατ' εκτίμηση θέση του K/T υπολογίζεται αρχικά σε συντεταγμένες UTM Easting/Northing, ζώνης 34 (CalcX, CalcY). Επομένως, ο αλγόριθμος πρέπει να μετατρέψει αυτές τις συντεταγμένες σε γεωγραφικό μήκος και το γεωγραφικό πλάτος του συστήματος WGS84. Για το σκοπό αυτό χρησιμοποιείται το module UTMWGS84 (πλήρης περιγραφή του module γίνεται στο ΠΑΡΑΡΤΗΜΑ Α).



ΣΧΗΜΑ 4.32

Αφού έχει ήδη υπολογίσει την κατ' εκτίμηση θέση του Κ/Τ, το πρόγραμμα προχωράει στον καθορισμό της Περιοχής Εμπιστοσύνης (Trust Area):

$$TRUST_LON = \text{Γεωγραφικό μήκος της Κυψέλης Εξυπηρέτησης (=FinalOk(1, 5))}$$

$$TRUST_LAT = \text{Γεωγραφικό πλάτος της Κυψέλης Εξυπηρέτησης (=FinalOk(1, 4))}$$

$$TRUST_CIRCLE_BIG = \text{minimum} \left\{ \begin{array}{l} \text{"HATA Radius" της κυψέλης εξυπηρέτησης} \\ \text{(Hata Type = 0)} \\ (0.55 * (FinalTA + 1) + 0.1) \end{array} \right.$$

$$\text{TRUST_CIRCLE_SMALL} = \begin{cases} 0 & , \text{av FinalTA} = 0 \\ 0.55 * (\text{FinalTA}) - 0.275 & , \text{av FinalTA} \neq 0 \end{cases}$$

TRUST_ANGLE_START = To Start Sector της κυψέλης εξυπηρέτησης
(FinalOk(1, 15))

TRUST_ANGLE_STOP = To Stop Sector of της κυψέλης εξυπηρέτησης
(FinalOk(1, 16))

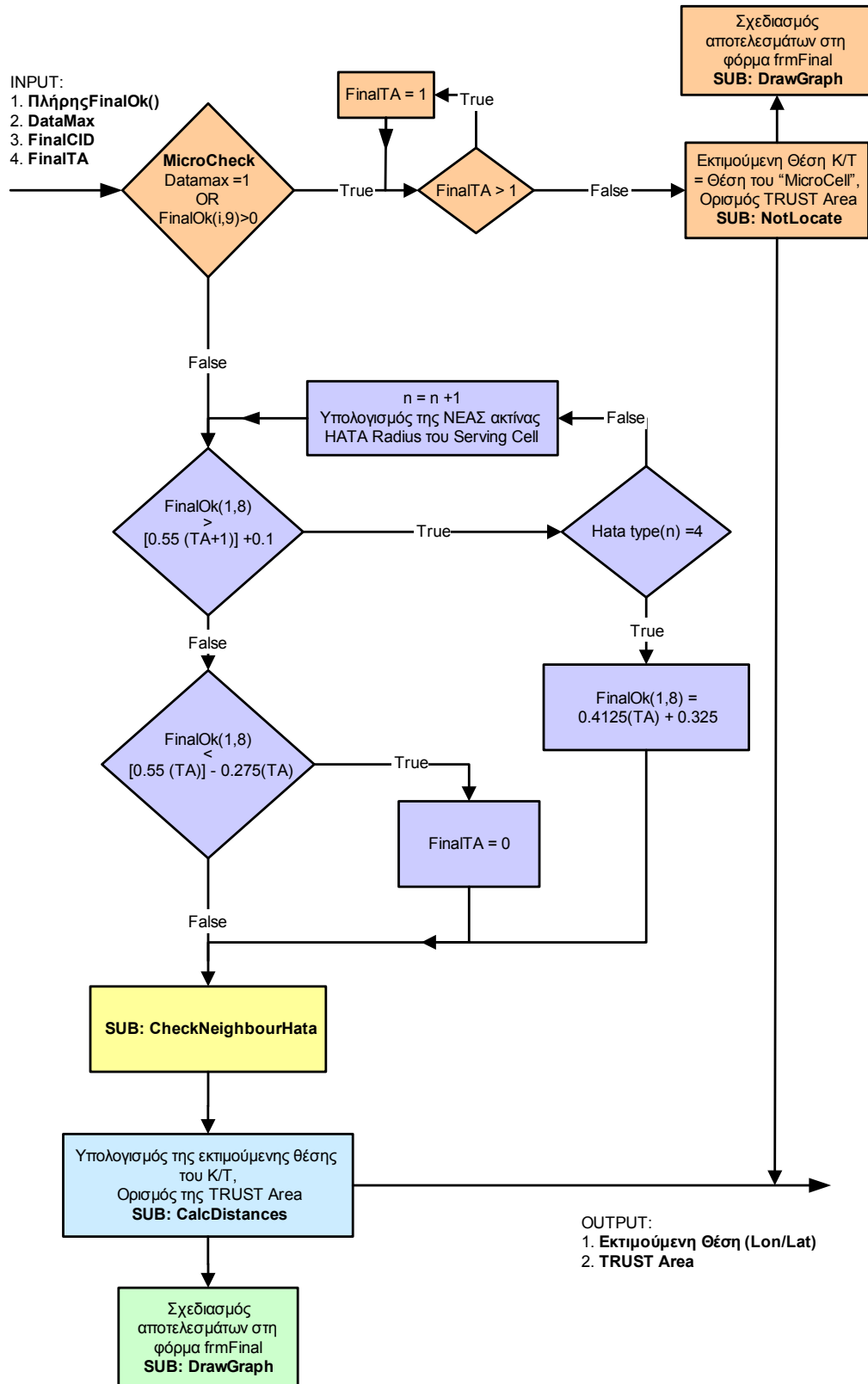
4.4.3.g. Υπο-Ρουτίνα *DrawGraph*

Αυτή η Υπο-Ρουτίνα καλείται μετά από την Υπο-Ρουτίνα **CalcDistances** και πραγματοποιεί τον γραφικό σχεδιασμό των υπολογισμένων αποτελεσμάτων στη Φόρμα "**frmFinal**". Δεδομένου ότι αυτή η Υπο-Ρουτίνα δεν σχετίζεται με τον Αλγόριθμο Εντοπισμού Θέσης του Κ/Τ (χρησιμοποιείται μόνο προκειμένου να υπάρξει μια οπτική απεικόνιση των αποτελεσμάτων), δεν πρόκειται να περιγραφεί εδώ. Ο πλήρης κώδικας αυτής της Υπο-Ρουτίνας δίνεται στο **ΠΑΡΑΡΤΗΜΑ Α**.

4.4.3.h. **PROCESS 3 – Διάγραμμα Ροής**

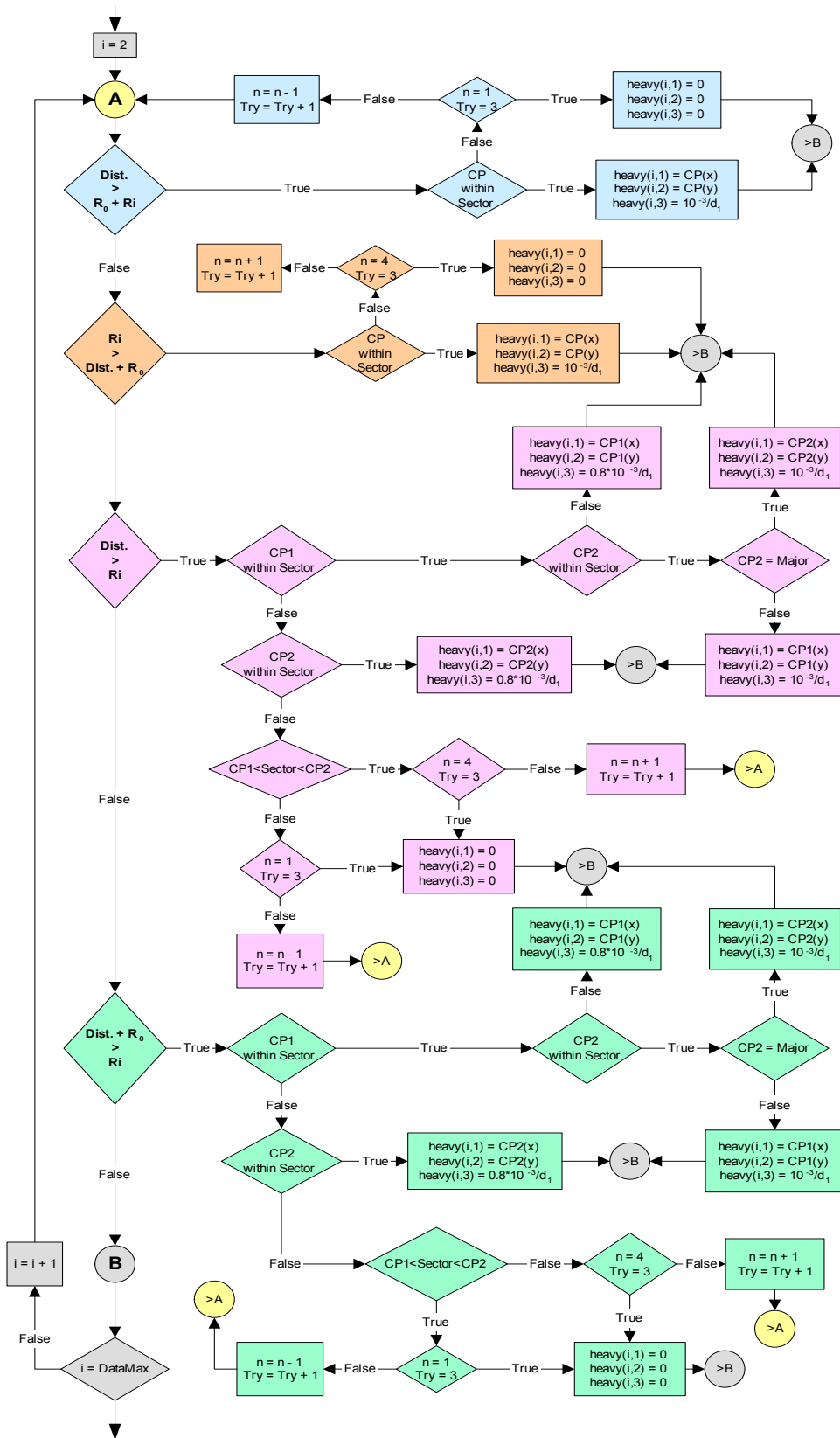
Εδώ ολοκληρώθηκε και η τελευταία Διαδικασία (PROCESS 3). Παρακάτω παρουσιάζουμε το διάγραμμα ροής του Υπολογισμού Εκτιμούμενης Θέσης του Κ/Τ (Σχήμα 4.33, 4.34). Στο Κεφάλαιο που ακολουθεί θα επιχειρηθεί να γίνει μια παρουσίαση των αποτελεσμάτων που πήραμε εφαρμόζοντας τη μέθοδο στην πράξη, καθώς και η γενικότερη αξιολόγηση της Μεθόδου Εντοπισμού Θέσης.

“PROCESS 3 - ΥΠΟΛΟΓΙΣΜΟΣ ΕΚΤΙΜΟΥΜΕΝΗΣ ΘΕΣΗΣ ΤΟΥ Κ/Τ”



ΣΧΗΜΑ 4. 33

Υπο-Ρουτίνα *CheckNeighbourHata*



ΣΧΗΜΑ 4. 34



ΚΕΦΑΛΑΙΟ 5

ΜΕΤΡΗΣΕΙΣ - ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ



5. ΜΕΤΡΗΣΕΙΣ – ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1. ΣΧΕΤΙΚΑ ΜΕ ΤΙΣ ΜΕΤΡΗΣΕΙΣ

Για τον έλεγχο της αξιοπιστίας του συγκεκριμένου αλγόριθμου εντοπισμού, κρίθηκε αναγκαία η διεξαγωγή μετρήσεων σε πραγματικό περιβάλλον. Για το σκοπό αυτό χρησιμοποιήθηκε το δίκτυο της Vodafone στην περιοχή της Λυκόβρυσης Αττικής, όπου υπήρχαν και τα απαραίτητα δεδομένα των Σταθμών Βάσης για την Βάση Δεδομένων. Η περιοχή αυτή είναι μια ημιαστική/αστική περιοχή με αρκετές βιομηχανίες/εργοστάσια και κτήρια που σε καμία περίπτωση δεν θα μπορούσε να χαρακτηριστεί σαν το κέντρο μιας μεγάλης αστικής πόλης, όπως η Αθήνα.

Οι μετρήσεις πραγματοποιήθηκαν σε διαφορετικές χρονικές περιόδους της ημέρας και σε συνολικά 50 διαφορετικά σημεία της περιοχής δοκιμών. Έτσι μπορέσαμε να πάρουμε ένα ικανοποιητικό δείγμα (415 μετρήσεις) για την σωστή αξιολόγηση της μεθόδου εντοπισμού.

Πριν προχωρήσουμε στην παρουσίαση και ανάλυση των μετρήσεων, κρίνεται σκόπιμο να τονίσουμε ότι υπήρξαν διάφοροι αρνητικοί παράγοντες οι οποίοι συνέβαλλαν στον περιορισμό της ακρίβειας του αλγόριθμου εντοπισμού, οι οποίοι οφείλονται κυρίως στη μη επαρκή περιγραφή του περιβάλλοντος δοκιμών. Αυτοί οι παράγοντες είναι οι ακόλουθοι:

- ◆ Δεν υπήρχε η πλήρης Βάση Δεδομένων για την περιοχή της Λυκόβρυσης, παρά μόνο ένας περιορισμένος αριθμός Σταθμών Βάσης με την Λίστα Γειτόνων. Έτσι δεν μπορούσε παρά να γίνει η δοκιμή του αλγόριθμου σε μια περιορισμένη περιοχή δοκιμών.
- ◆ Μερικές από τις Γειτονικές Κυψέλες δεν ήταν καταχωρημένες στη Βάση (πίνακας Cell_Info) με αποτέλεσμα κατά τη διάρκεια των μετρήσεων ο αλγόριθμος να έχει λιγότερα διαθέσιμα στοιχεία προς εντοπισμό, απ' ό,τι θα έπρεπε να έχει στην πραγματικότητα (πλήρης βάση δεδομένων).
- ◆ Λόγω έλλειψης της πληροφορίας για το εύρος μισής ισχύος των κεραιών των Σ/Β, θεωρήθηκε ότι όλες ανεξαιρέτως οι κεραιές είχαν το ίδιο εύρος, ίσο με 120 μίρες. Κάτι τέτοιο είναι προφανώς λάθος, αλλά λόγω αδυναμίας εύρεσης των πραγματικών γωνιών καταφύγαμε τελικά σ' αυτή τη λύση.
- ◆ Για όλους τους Σταθμούς Βάσης τέθηκε ως περιβάλλον ραδιοκάλυψης αναφοράς το sub-urban (Hata = 2) αφού δεν υπήρχε καμία άλλη σχετική πληροφορία.

Λόγω των παραπάνω παραγόντων μπορούμε να ισχυριστούμε ότι η εν δυνάμει ακρίβεια του αλγόριθμου εντοπισμού θα πρέπει να είναι ελαφρώς καλύτερη από αυτή που θα παρουσιάσουμε στη στις επόμενες σελίδες του Κεφαλαίου.

5.2. ΠΕΡΙΠΤΩΣΕΙΣ ΜΕΤΡΗΣΕΩΝ

Σ' αυτή τη παράγραφο θα παρουσιάσουμε διάφορες περιπτώσεις από τις μετρήσεις που ελήφθησαν και παρουσιάζουν σχετικό ενδιαφέρον, η κάθε μία για διαφορετικό λόγο. Σε κάθε μία από αυτές τις περιπτώσεις δίνεται η σχετική γραφική αναπαράσταση της μέτρησης καθώς και ο ανάλογος σχολιασμός.

5.2.1. ΜΕΤΡΗΣΗ ΜΕ ΤΗ ΜΕΓΑΛΥΤΕΡΗ ΑΚΡΙΒΕΙΑ

Serv. Cell Hata = 745m

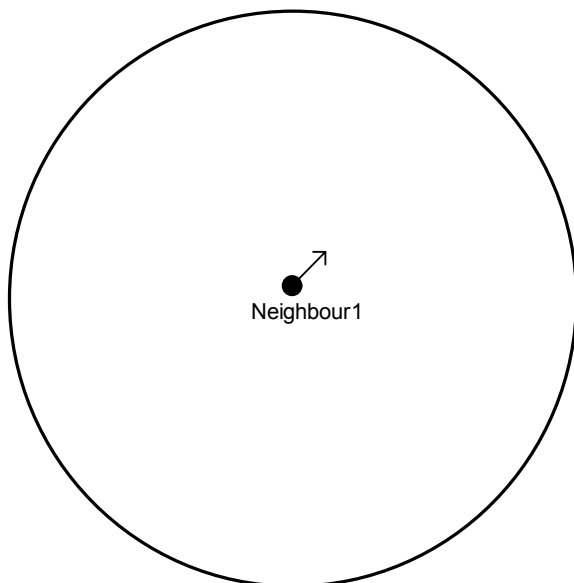
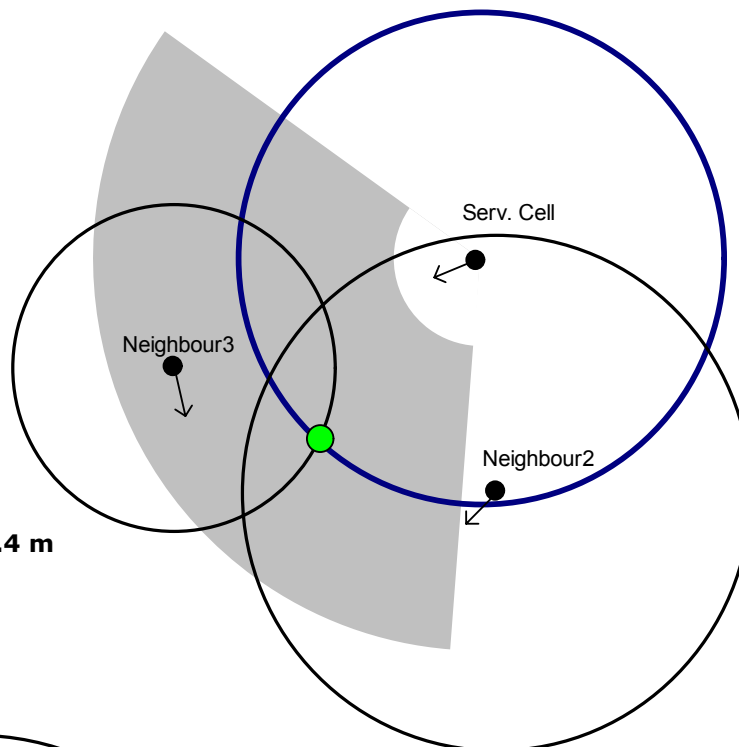
Neighb.1 Hata = 867m

Neighb.2 Hata = 790m

Neighb.3 Hata = 511m

Σφάλμα Αλγόριθμου = **13.4 m**

Σφάλμα CID = **754.5 m**



Σ' αυτή εδώ την περίπτωση βλέπουμε ότι το σφάλμα είναι ελάχιστο, μόνο 13.4m (Η υπολογιζόμενη θέση του Κ/Τ σχεδόν συμπίπτει με τη θέση από το GPS), σε αντίθεση με το σφάλμα της μεθόδου CID που είναι πολύ μεγάλο, 754.5m. Επίσης παρατηρούμε ότι η TRUST AREA που υπολόγισε ο αλγόριθμος εμπεριέχει την πραγματική θέση του Κ/Τ.

Η πολύ καλή απόδοση στη συγκεκριμένη περίπτωση οφείλεται στην μεγάλη ακρίβεια της ακτίνας HATA Radius τόσο της Κυψέλης Εξυπηρέτησης (Serving Cell), όσο και της 3^{ης} Γειτονικής Κυψέλης (Neighbour 3).

5.2.2. ΜΕΤΡΗΣΗ ΜΕ ΤΗ ΜΙΚΡΟΤΕΡΗ ΑΚΡΙΒΕΙΑ

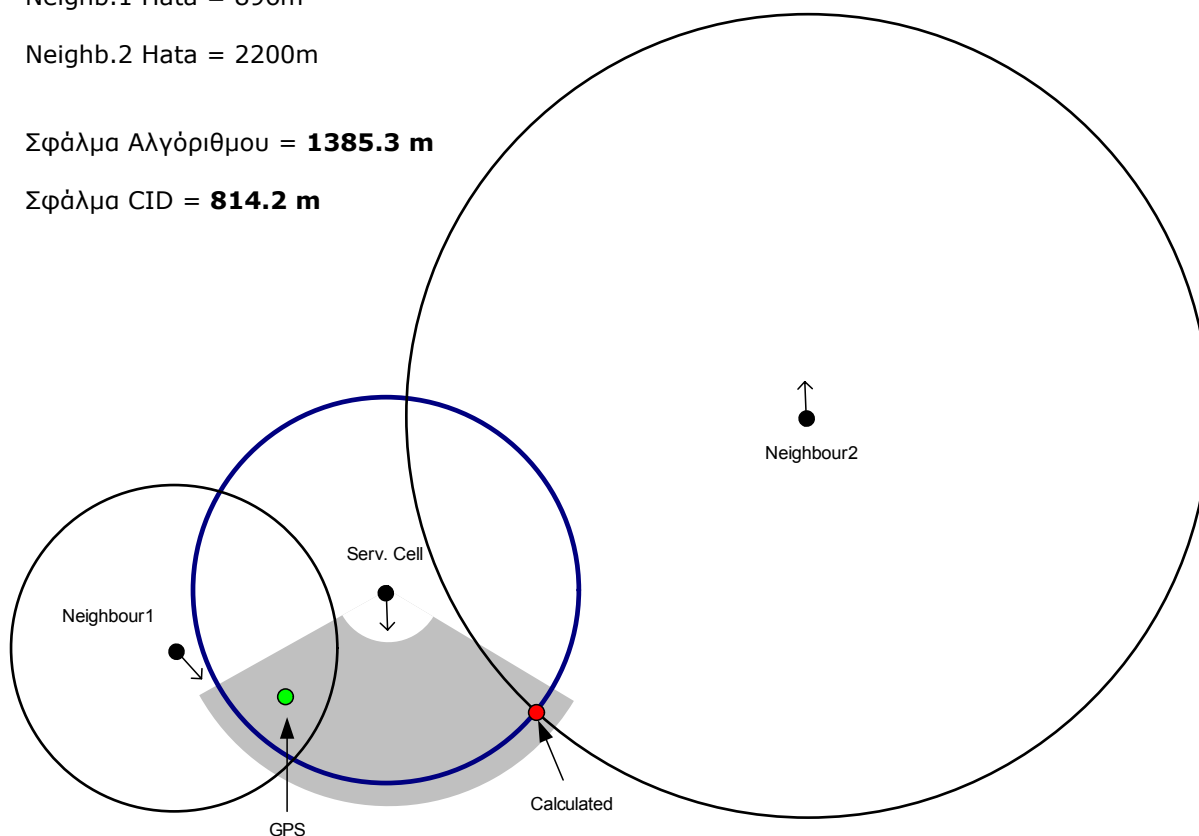
Serv. Cell Hata = 1065m

Neighb.1 Hata = 896m

Neighb.2 Hata = 2200m

Σφάλμα Αλγόριθμου = **1385.3 m**

Σφάλμα CID = **814.2 m**



Από τη γραφική αναπαράσταση παρατηρούμε ότι το πολύ μεγάλο σφάλμα στον υπολογισμό οφείλεται κυρίως στην λανθασμένη ακτίνα HATA Radius του Neighbour2, η οποία θα έπρεπε να είναι μεγαλύτερη. Πρέπει να τονίσουμε ότι το συγκεκριμένο γειτονικό BTS, εμφάνιζε συνεχώς προβλήματα κατά τη διάρκεια των μετρήσεων. Όπως μας έγινε γνωστό αργότερα, αυτός ο Σταθμός Βάσης είναι ένα umbrella-BTS τοποθετημένος στο βουνό της Πάρνηθας με μια ομοιοκατευθυντική κεραία, με σκοπό να καλύπτει τις προβληματικές περιοχές οι οποίες δεν καλύπτονται από το υπόλοιπο δίκτυο.

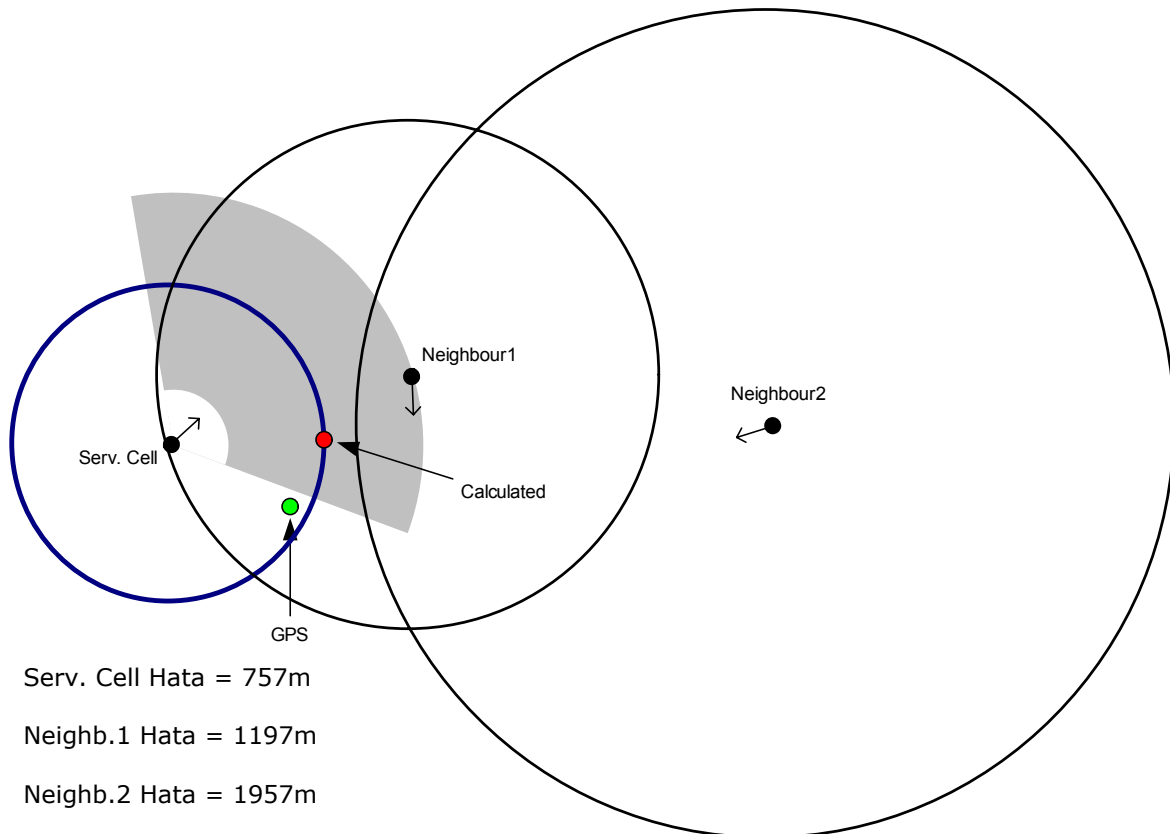
Αν αγνοηθεί η παρουσία του Neighbour2 και πραγματοποιηθεί εντοπισμός χρησιμοποιώντας μόνο το Serving Cell και τον Neighbour1 τότε η ακρίβεια του αλγόριθμου αυξάνεται δραματικά, με σφάλμα στα 336.4m (πρόκειται για το σημείο τομής των δύο κύκλων που βρίσκεται μέσα στην περιοχή εμπιστοσύνης TRUST AREA).

Ένα άλλο σημείο στο οποίο πρέπει να σταθούμε είναι ότι παρά το πολύ μεγάλο σφάλμα τον υπολογισμό της θέσης του Κ/Τ, και σ' αυτήν την περίπτωση η περιοχή εμπιστοσύνης εμπεριέχει την πραγματική θέση του Κ/Τ.

5.2.3. ΣΦΑΛΜΑ ΣΤΗΝ ΠΕΡΙΟΧΗ ΕΜΠΙΣΤΟΣΥΝΗΣ

Σφάλμα στην περιοχή εμπιστοσύνης μπορεί να υπάρξει για δύο λόγους. Πρώτον, λόγω σφάλματος στις γωνίες έναρξης (START_SECTOR) και λήξης (STOP_SECTOR) της περιοχής, και δεύτερον λόγω σφάλματος στις ακτίνες TRUST_MIN και TRUST_MAX.

- Σφάλμα λόγω των γωνιών λήξης και έναρξης.



Σφάλμα Αλγόριθμου = **370.5 m**, Σφάλμα CID = **643.6 m**

Εδώ, παρότι πετυχαίνουμε καλύτερη ακρίβεια με τη μέθοδό μας σε σχέση με τη μέθοδο CID, παρατηρούμε ότι η περιοχή εμπιστοσύνης δεν περιέχει την πραγματική θέση του Κ/Τ (GPS). Ένα τέτοιο πρόβλημα αντιμετωπίζεται μόνο αν αλλάξουμε τα στοιχεία για την περιοχή κάλυψης της Κυπέλης Εξυπηρέτησης (Serving Cell). Αυτό συνεπάγεται ότι θα υπάρχουν και περιοχές αλληλοκάλυψης μεταξύ cosited κυψελών, γεγονός που δεν προβληματίζει αφού κάτι τέτοιο συμβαίνει και στην πραγματικότητα. Πρέπει να υπενθυμίσουμε ότι για όλους τους Σταθμούς Βάσης ανεξαιρέτως έχει ληφθεί ως γωνία μισής ισχύος (3dB) 120 μοίρες, λόγω έλλειψης πληροφοριών.

➤ **Σφάλμα λόγω των ακτινών TRUST_MIN & TRUST_MAX**

Serv. Cell Hata = 306m

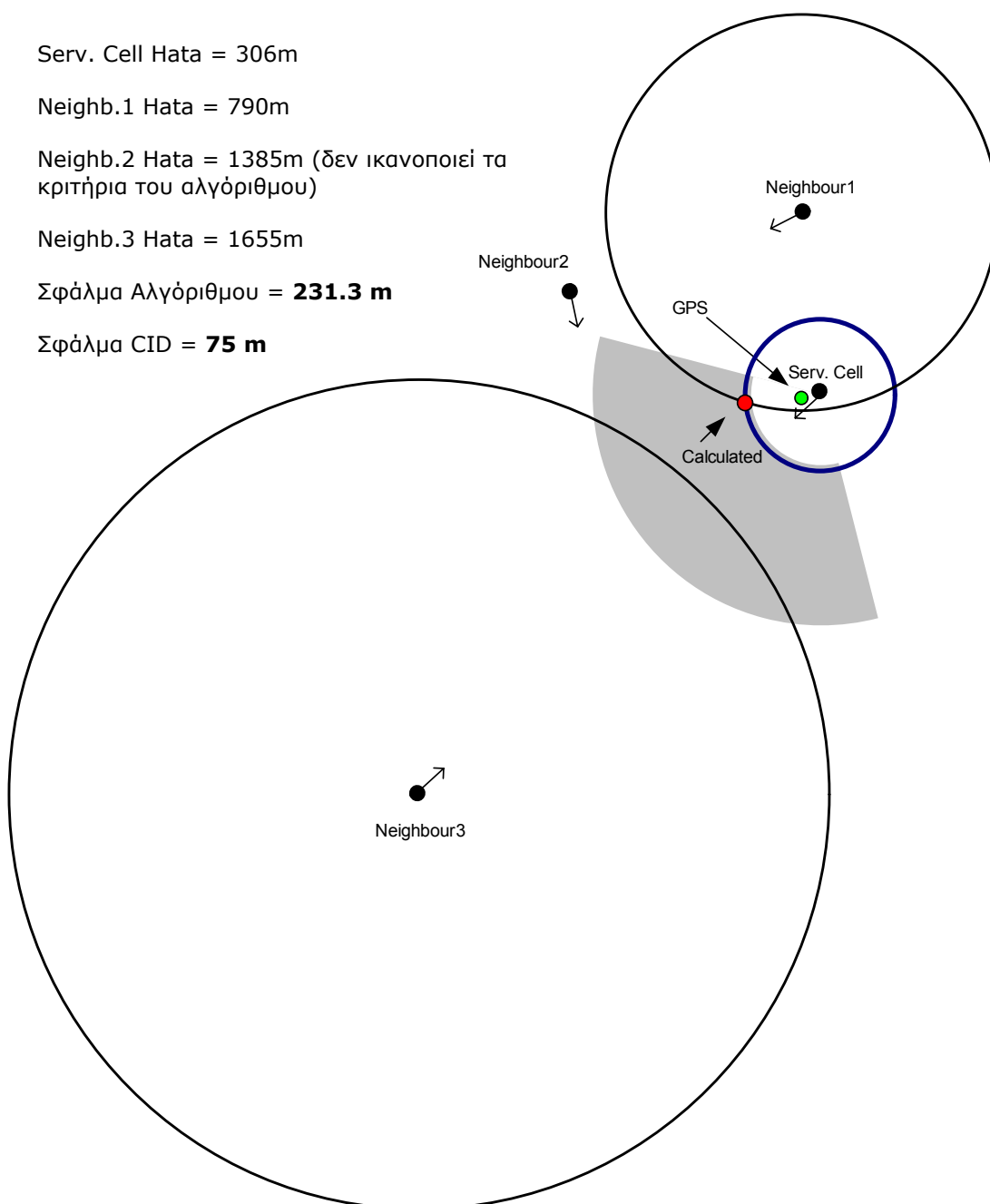
Neighb.1 Hata = 790m

Neighb.2 Hata = 1385m (δεν ικανοποιεί τα κριτήρια του αλγόριθμου)

Neighb.3 Hata = 1655m

Σφάλμα Αλγόριθμου = **231.3 m**

Σφάλμα CID = **75 m**



Σ' αυτή την περίπτωση έχουμε να κάνουμε με ένα σημαντικό πρόβλημα του αλγόριθμου. Η συγκεκριμένη μέτρηση μας δίνει TA = 1 ενώ κανονικά θα έπρεπε να είχαμε TA = 0. Μπορεί επίσης να συμβεί και το αντίστροφο (να έχουμε TA = 0 ενώ θα έπρεπε να είχαμε TA = 1).

Τέτοιες περιπτώσεις είναι αδύνατον να τις αποφύγουμε με οποιοδήποτε τρόπο, αλλά, όπως θα φανεί και από τα συγκεντρωτικά αποτελέσματα, ευτυχώς αποτελούν ένα πολύ μικρό κομμάτι του συνόλου (η περιοχή εμπιστοσύνης έχει ποσοστό επιτυχίας γύρω στο 90%).

5.2.4.ΤΟ SERVING CELL ΔΕΝ ΕΙΝΑΙ Ο ΚΟΤΙΝΟΤΕΡΟΣ Σ/Β

Serv. Cell Hata = 1150m

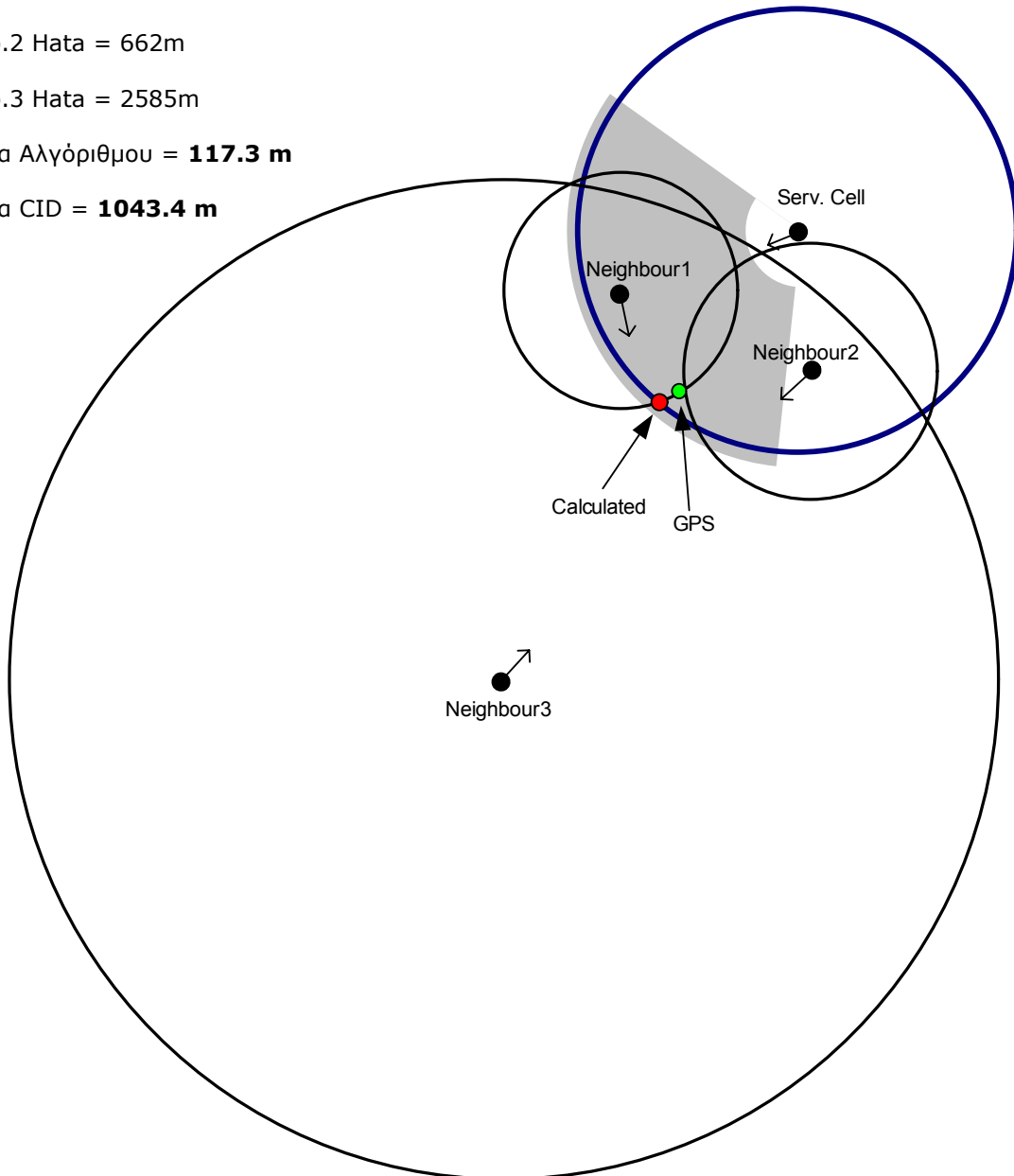
Neighb.1 Hata = 615m

Neighb.2 Hata = 662m

Neighb.3 Hata = 2585m

Σφάλμα Αλγόριθμου = **117.3 m**

Σφάλμα CID = **1043.4 m**



Όπως είναι εμφανές, τόσο ο Neighbour1 όσο και ο Neighbour2 απέχουν πολύ μικρότερη απόσταση από την πραγματική θέση του Κ/Τ σε σχέση με το Serving Cell. Επίσης η ισχύς λήψης και στους δύο αυτούς Γείτονες (-80dBm) είναι σημαντικά μεγαλύτερη απ' ότι στην Κυψέλη Εξυπηρέτησης (-84dBm). Παρόλα αυτά, ο αλγόριθμος παρουσιάζει εκπληκτική ακρίβεια (117m), γεγονός που μας καθησυχάζει για τέτοιου είδους παρόμοιες περιπτώσεις. Επισημαίνουμε ότι η ακρίβεια και των τριών ακτινών HATA Radius (Serving Cell, Neighbour1, Neighbour2) είναι πάρα πολύ καλή.

5.2.5. ΠΟΛΥ ΜΕΓΑΛΗ ΑΚΡΙΒΕΙΑ ΤΗΣ HATA RADIUS ΤΟΥ SERV. CELL

Serv. Cell Hata = 655m

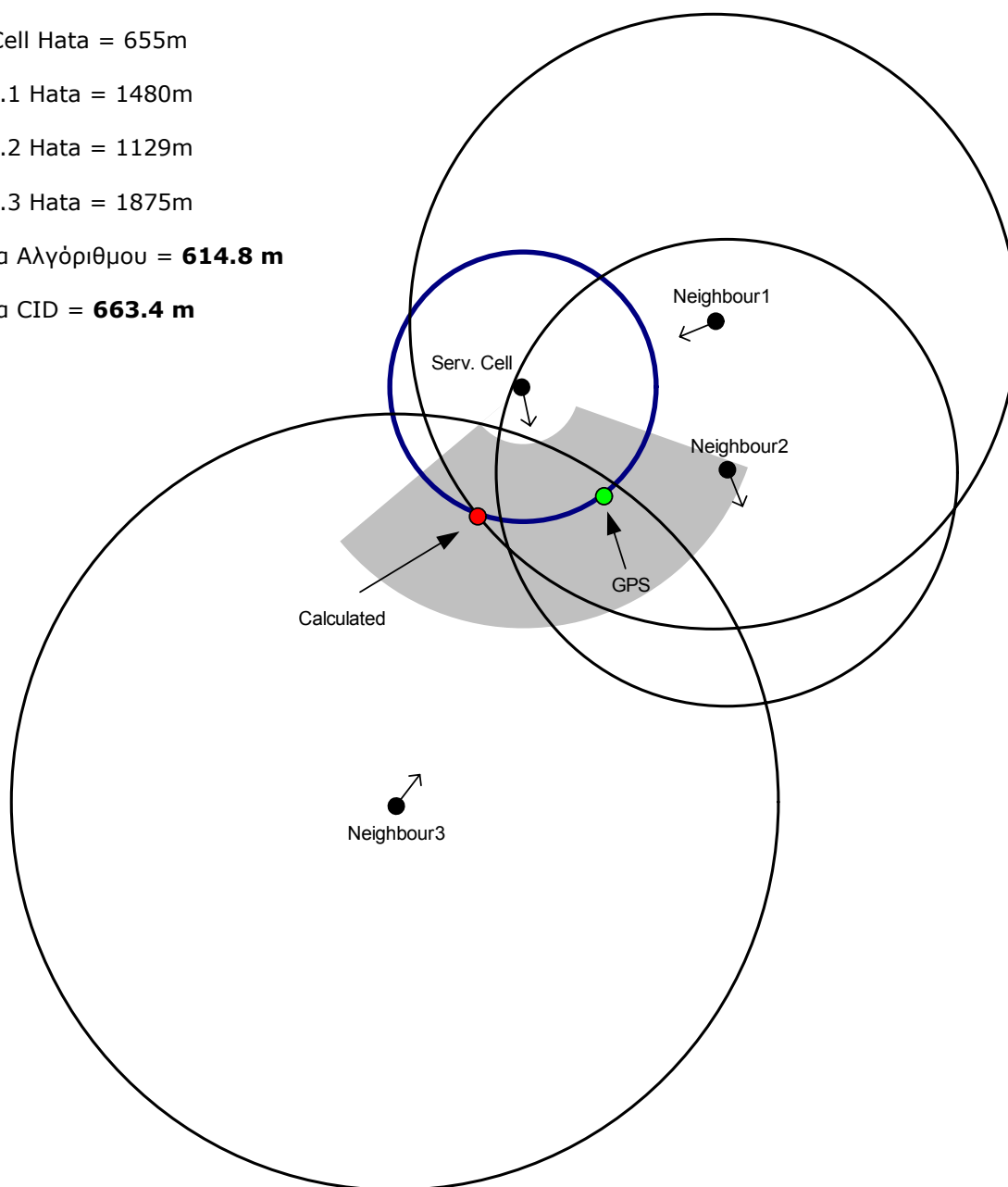
Neighb.1 Hata = 1480m

Neighb.2 Hata = 1129m

Neighb.3 Hata = 1875m

Σφάλμα Αλγόριθμου = **614.8 m**

Σφάλμα CID = **663.4 m**



Ενώ η ακρίβεια της ακτίνας HATA Radius της Κυψέλης Εξυπηρέτησης είναι πάρα πολύ μεγάλη (9m), ο αλγόριθμος δεν βελτιώνει σημαντικά την ακρίβεια της μεθόδου CID. Όπως φαίνεται, ως θέση του Κ/Τ λαμβάνεται το σημείο τομής του Neighbour1 με το Serving Cell. Αντίθετα, αν παίρναμε το σημείο τομής Neighbour3 με το Serving Cell, η ακρίβεια θα ήταν εντυπωσιακή (69m).

Γενικότερα παρατηρήθηκε πολύ καλή ακρίβεια της ακτίνας HATA Radius του Serving Cell εκτός μερικών εξαιρέσεων. Αυτό επιβεβαιώνει την ορθότητα της επιλογής μας σχετικά με τον ρόλο του Serving Cell στον υπολογισμό της εκτιμώμενης θέσης του Κ/Τ.

5.2.6. ΤΟ ΠΡΟΒΛΗΜΑ ΣΤΙΣ ΚΟΝΤΙΝΕΣ ΑΠΟΣΤΑΣΕΙΣ

Serv. Cell Hata = 272m

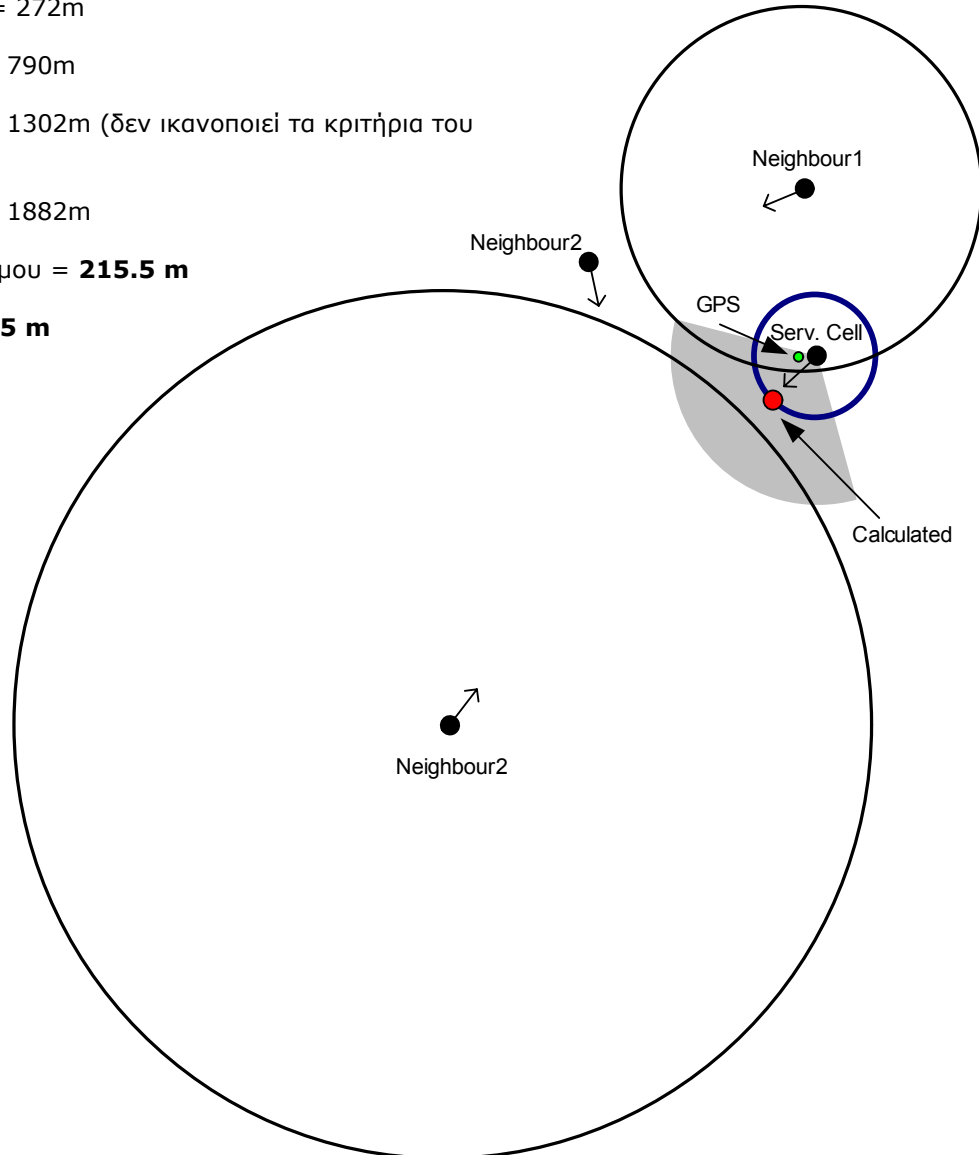
Neighb.1 Hata = 790m

Neighb.2 Hata = 1302m (δεν ικανοποιεί τα κριτήρια του αλγόριθμου)

Neighb.3 Hata = 1882m

Σφάλμα Αλγόριθμου = **215.5 m**

Σφάλμα CID = **75 m**



Όπως μπορούμε να διαπιστώσουμε από τη συγκεκριμένη περίπτωση, αλλά και όπως θα δούμε αργότερα στο κεφάλαιο, στις πολύ κοντινές αποστάσεις ο αλγόριθμος εντοπισμού θέσης εμφανίζει σημαντικό πρόβλημα. Είναι αξιοσημείωτο το γεγονός ότι ο αλγόριθμος σε καμία περίπτωση (σχετικά με τις κοντινές αποστάσεις) δεν κατάφερε να δώσει καλύτερη ακρίβεια από αυτή που παίρνουμε από τη μέθοδο του CID. Αυτό οφείλεται σχεδόν αποκλειστικά στην ακτίνα HATA Radius της Κυψέλης Εξυπηρέτησης. Είναι φανερό δηλαδή, ότι το μοντέλο ραδιοκάλυψης του Okumura-Hata δεν λειτουργεί επιτυχώς σε μικρές αποστάσεις.

5.2.7. ΔΙΑΦΟΡΕΣ ΠΕΡΙΠΤΩΣΕΙΣ

Σ' αυτή την παράγραφο θα παρουσιάσουμε δύο τελευταίες περιπτώσεις οι οποίες δεν μας βοηθούν στο να εξάγουμε συμπεράσματα για την συμπεριφορά/απόδοση του αλγόριθμου, αλλά παρουσιάζουν κάποιο ενδιαφέρον.

- **ΠΕΡΙΠΤΩΣΗ 1**

Serv. Cell Hata = 697m

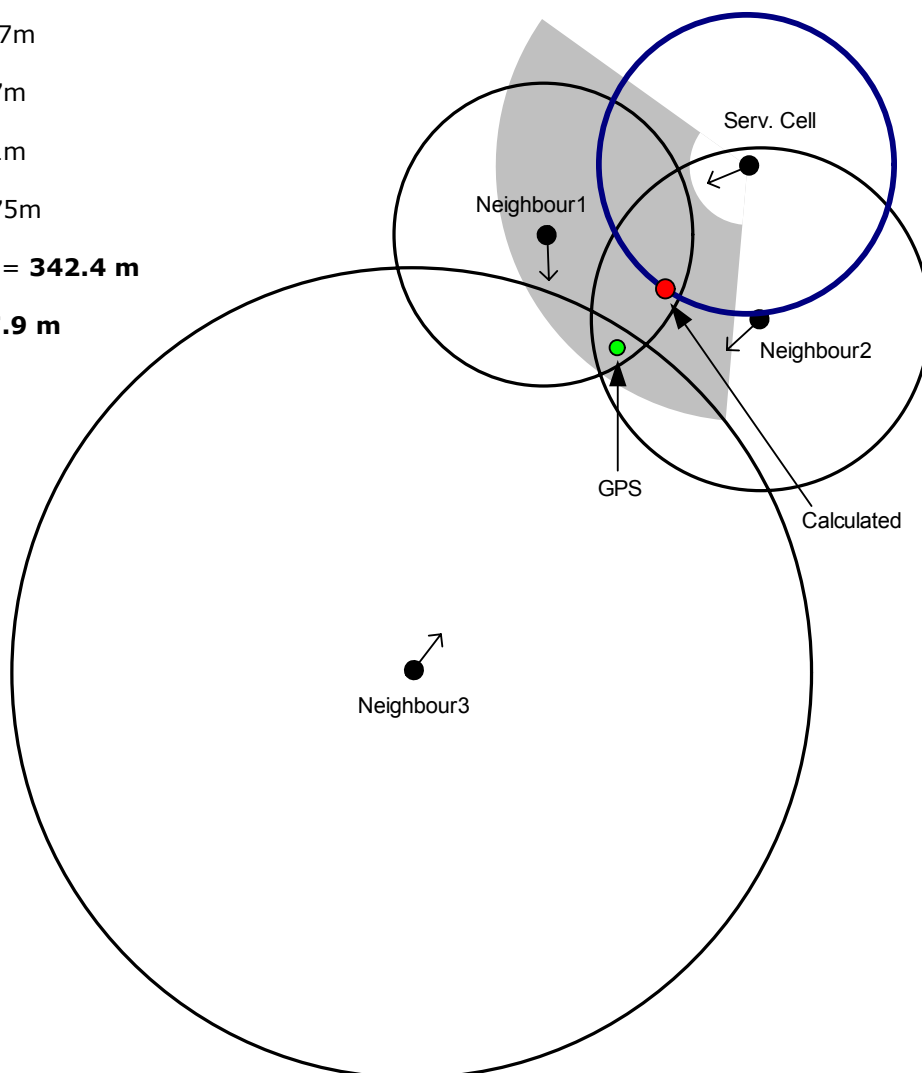
Neighb.1 Hata = 697m

Neighb.2 Hata = 791m

Neighb.3 Hata = 1875m

Σφάλμα Αλγόριθμου = **342.4 m**

Σφάλμα CID = **1037.9 m**



Πέρα από την πολύ καλή ακρίβεια, συγκριτικά με τη μέθοδο CID, μπορούμε να παρατηρήσουμε ότι οι τρεις ακτίνες HATA Radius των Γειτονικών Κυψελών ορίζουν μια πολύ μικρή περιοχή μέσα στην οποία βρίσκεται η αληθής θέση του Κ/Τ (Η HATA Radius του Serving Cell είναι εκτός αυτής της περιοχής). Κάτι τέτοιο συμβαίνει πολύ σπάνια και όταν συμβαίνει, η περιοχή μπορεί να μην συμπεριλαμβάνει την ακριβή θέση του Κ/Τ, οπότε δεν μπορούμε να το εκμεταλλευτούμε ώστε να ελαχιστοποιήσουμε την περιοχή εμπιστοσύνης.

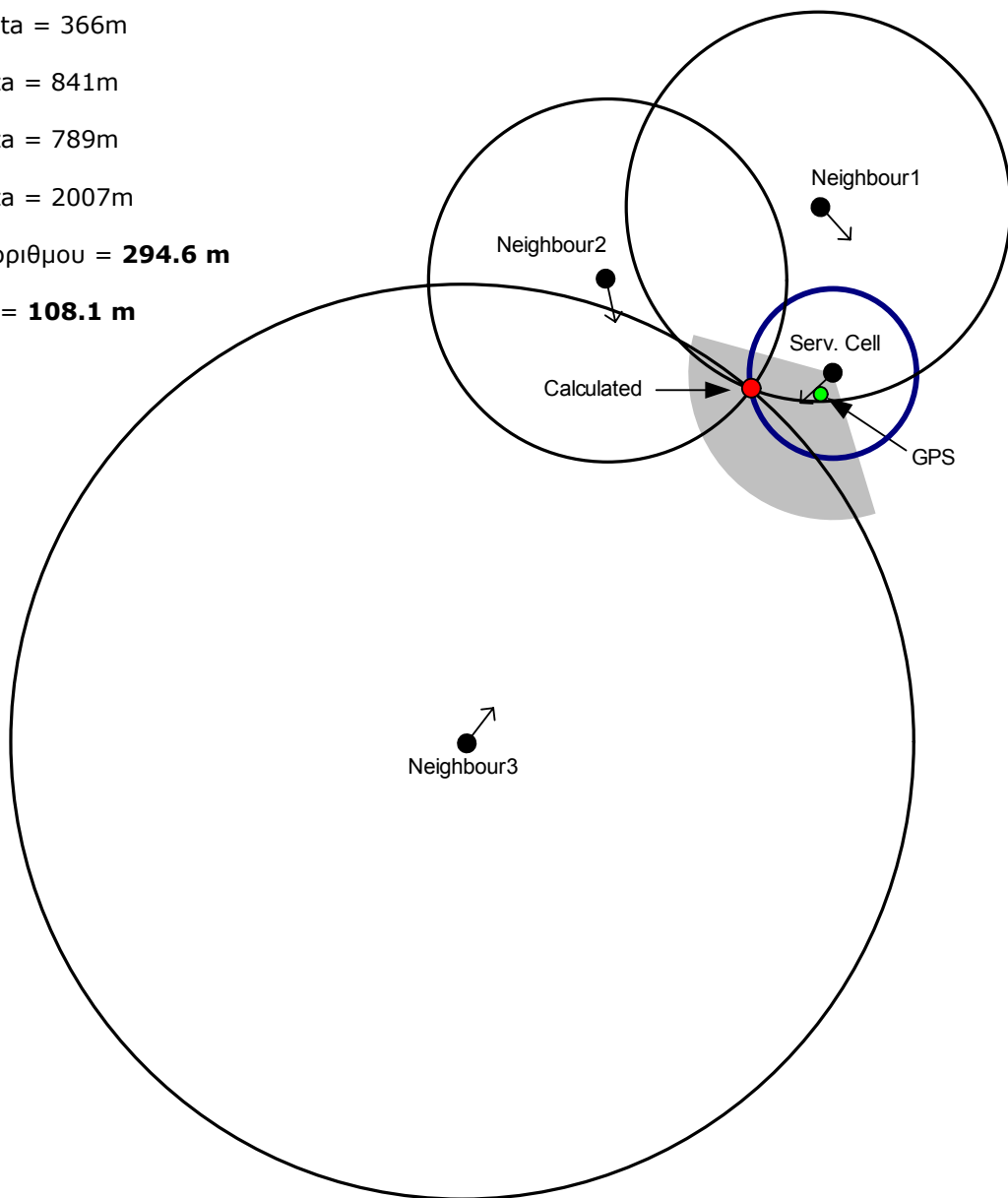
• ΠΕΡΙΠΤΩΣΗ 2

Serv. Cell Hata = 366m

Neighb.1 Hata = 841m

Neighb.2 Hata = 789m

Neighb.3 Hata = 2007m

Σφάλμα Αλγόριθμου = **294.6 m**Σφάλμα CID = **108.1 m**

Όπως μπορούμε εύκολα να δούμε, όλες οι ακτίνες HATA Radius συναντιούνται σε ένα κοινό σημείο το οποίο είναι ασφαλώς η υπολογιζόμενη θέση του Κ/Τ. Ενώ θα περίμενε κανένας ότι η ακρίβεια σ' αυτή τη περίπτωση θα ήταν πάρα πολύ καλή έως άριστη, αντίθετα έχουμε μεγάλο σφάλμα συγκριτικά με τη μέθοδο CID!

Αυτό το παράδειγμα λοιπόν, είναι ενδεικτικό της δυσκολίας, της πολυπλοκότητας και της αβεβαιότητας που εμπεριέχει το πρόβλημα του εντοπισμού της θέσης του Κ/Τ, μέσω της ισχύος λήψης σήματος.

5.3. ΣΥΓΚΕΝΤΡΩΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ - ΑΝΑΛΥΣΗ

Στις παρακάτω σελίδες ο αλγόριθμος εντοπισμού θέσης που υλοποιήσαμε θα αναφέρεται χάριν συντομογραφίας ως **M.P.A** (**M**obile **P**ositioning **A**lgorithm).

5.3.1. ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Συνολικά πραγματοποιήθηκαν **415** μετρήσεις και τα αποτελέσματα από την επεξεργασία των μετρήσεων είναι τα ακόλουθα:

ΑΚΡΙΒΕΙΑ ΤΟΥ "M.P.A" ΣΕ ΣΥΓΚΡΙΣΗ ΜΕ ΤΗ ΜΕΘΟΔΟ "CID":

- Στο σύνολο των μετρήσεων:

Ακρίβεια M.P.A: **319m**

Ακρίβεια CID: **527m**

*Συγκριτική Βελτίωση: **208m** ή **39.4%***

Ποσοστό επί των μετρήσεων όπου ο M.P.A βελτίωσε την ακρίβεια έναντι της Μεθόδου CID: **79%**

- Στο 95% των μετρήσεων:

Βέλτιστη ακρίβεια M.P.A: **290m**

Βέλτιστη ακρίβεια CID: **496m**

*Συγκριτική Βελτίωση: **206m** ή **41.5%***

- Στο 67% των μετρήσεων:

Βέλτιστη ακρίβεια M.P.A: **142m**

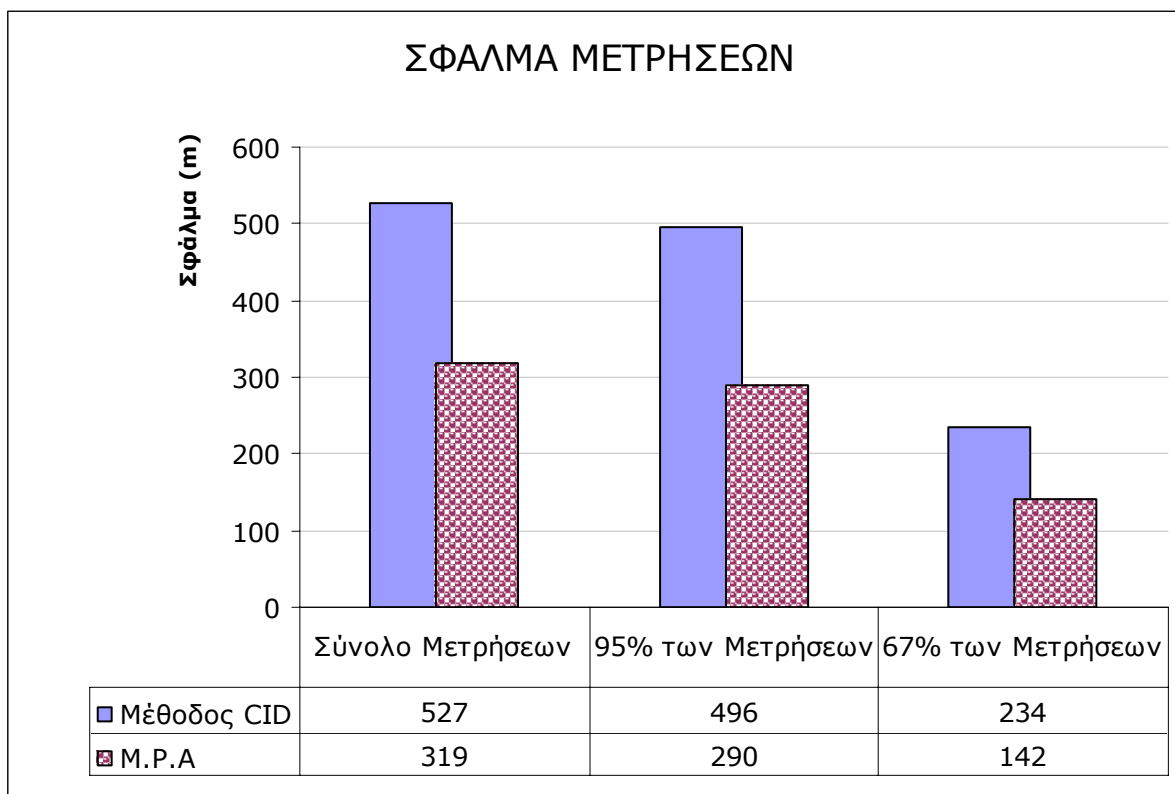
Βέλτιστη ακρίβεια CID: **234m**

*Συγκριτική Βελτίωση: **92m** ή **39.3%***

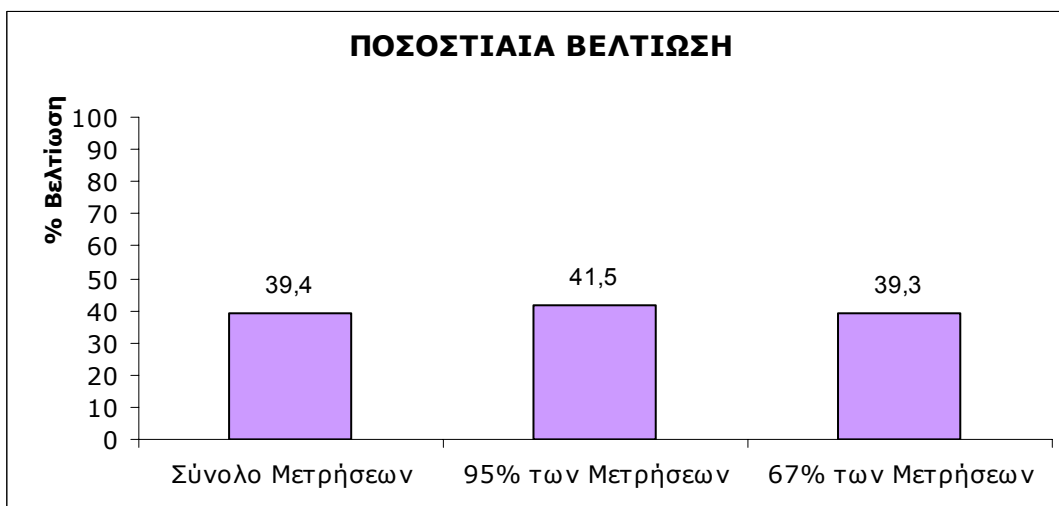
Εκτός από τη συνολική ακρίβεια του M.P.A παραθέτουμε και την βέλτιστη ακρίβεια στο 95% και 67% των μετρήσεων για να συγκρίνουμε τα αποτελέσματα με τις απαιτήσεις της FCC(99-245), που υπενθυμίζουμε ότι είναι 300m για το 95% και 100m για το 67%.

Συγκρίνοντας, βλέπουμε ότι καλύπτεται μόνο η πρώτη απαίτηση για το 95%, ενώ στο 67% ο M.P.A υπολείπεται των απαιτήσεων κατά 42m.

Ένα άλλο σημείο το οποίο παρουσιάζει ενδιαφέρον είναι ότι η ποσοστιαία τιμή της συγκριτικής βελτίωσης παραμένει πρακτικά σταθερή και ίση με 40%.



ΣΧΗΜΑ 5.1

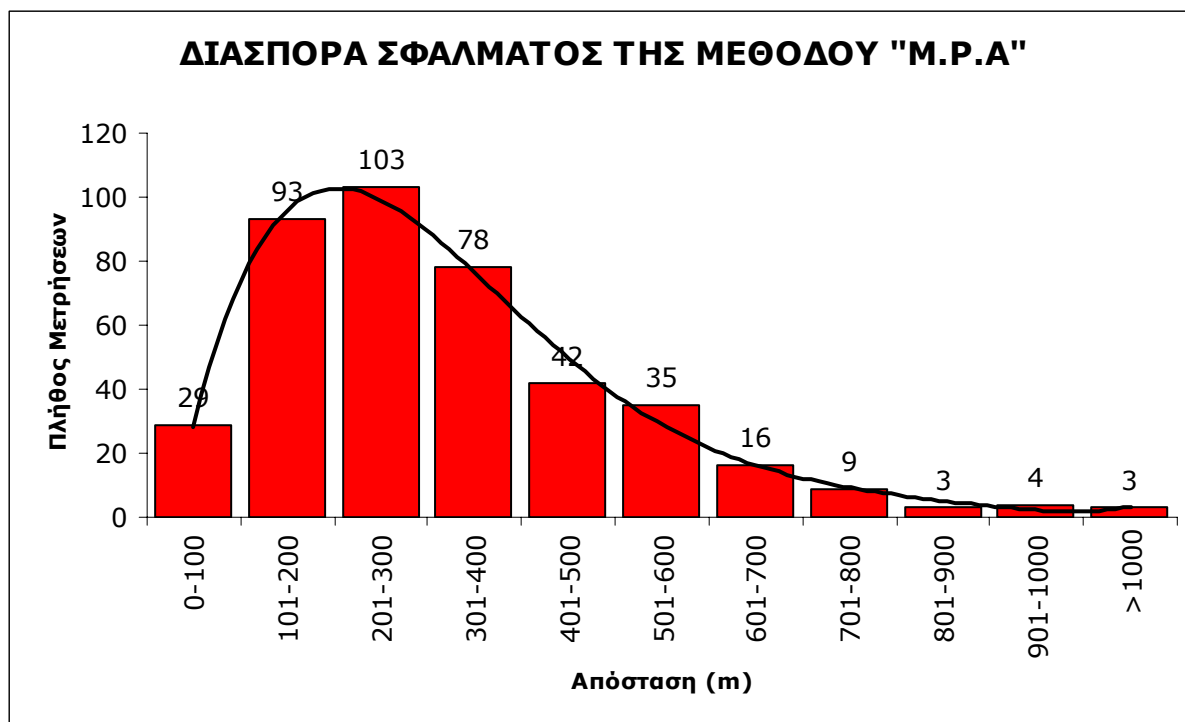


ΣΧΗΜΑ 5.2

ΔΙΑΣΠΟΡΑ ΑΚΡΙΒΕΙΑΣ (ΣΦΑΛΜΑΤΟΣ) ΤΗΣ ΜΕΘΟΔΟΥ "Μ.Ρ.Α":

Ουσιαστικά με τον όρο ακρίβεια και σφάλμα εννοούμε την ίδια αριθμητική ποσότητα (η απόσταση του υπολογιζόμενου σημείου από την πραγματική θέση του Κ/Τ), με την διαφορά ότι αύξηση του αριθμού αυξάνει το σφάλμα και μειώνει την ακρίβεια, ενώ μείωση του αριθμού μειώνει το σφάλμα και αυξάνει την ακρίβεια.

Στο Σχήμα 5.3 δίνεται η διασπορά του Σφάλματος της Μεθόδου Μ.Ρ.Α στο σύνολο των μετρήσεων.

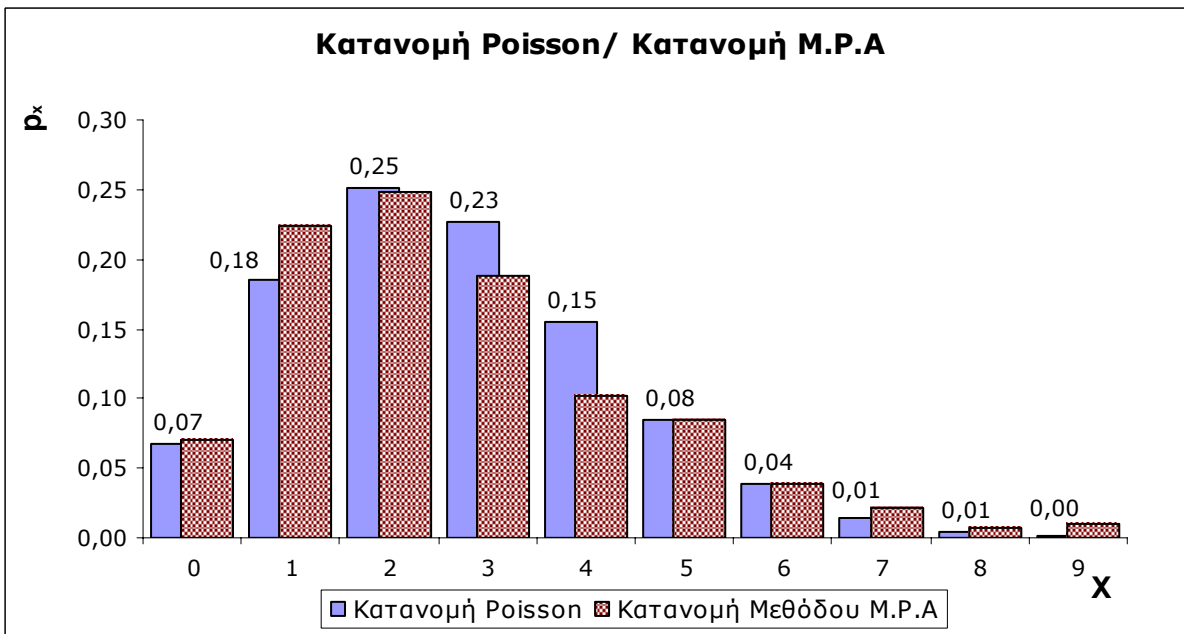


ΣΧΗΜΑ 5.3

Μελετώντας την γραμμή τάσης, παρατηρούμε ότι ακολουθεί κατανομή Poisson. Πράγματι αν θεωρήσουμε κατανομή Poisson με $\lambda = 4.1$:

$$p_x = e^{-\lambda} \frac{\lambda^x}{x!}$$

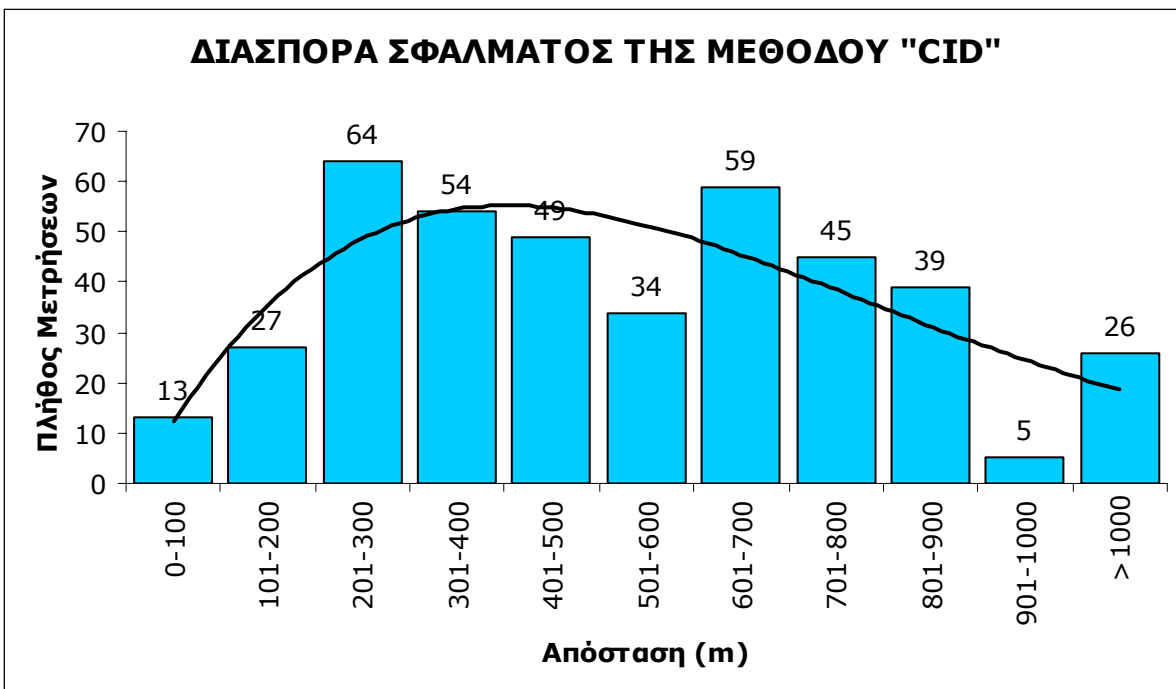
και αν σχεδιάσουμε τις δύο κατανομές σε αντιπαράθεση, παίρνουμε την γραφική παράσταση του Σχήματος 5.4:



ΣΧΗΜΑ 5.4

Η ομοιότητα και η σύγκλιση των τιμών στις δύο κατανομές είναι εντυπωσιακή, υπάρχουν μόνο μικρές διαφορές.

ΔΙΑΣΠΟΡΑ ΑΚΡΙΒΕΙΑΣ (ΣΦΑΛΜΑΤΟΣ) ΤΗΣ ΜΕΘΟΔΟΥ "CID":



ΣΧΗΜΑ 5.5

Πρόκειται ουσιαστικά για τις αποστάσεις από την Κυψέλη Εξυπηρέτησης στις οποίες ελήφθησαν οι μετρήσεις. Η ιδανική περίπτωση θα ήταν να είχαμε ισόποση κατανομή των μετρήσεων σε όλες τις αποστάσεις, έτσι ώστε να μπορούμε να κάνουμε βέλτιστη αξιολόγηση του αλγόριθμου εντοπισμού θέσης. Κάτι τέτοιο βέβαια δεν συμβαίνει (Σχήμα 5.5) αφού έχουμε έλλειψη μετρήσεων κυρίως στις πολύ κοντινές και πολύ μακρινές αποστάσεις, αλλά το συνολικό πλήθος των μετρήσεων είναι αρκετό για να εξάγουμε ασφαλή συμπεράσματα.

ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΜΕΘΟΔΟΥ Μ.Ρ.Α ΣΤΙΣ ΜΙΚΡΕΣ ΑΠΟΣΤΑΣΕΙΣ

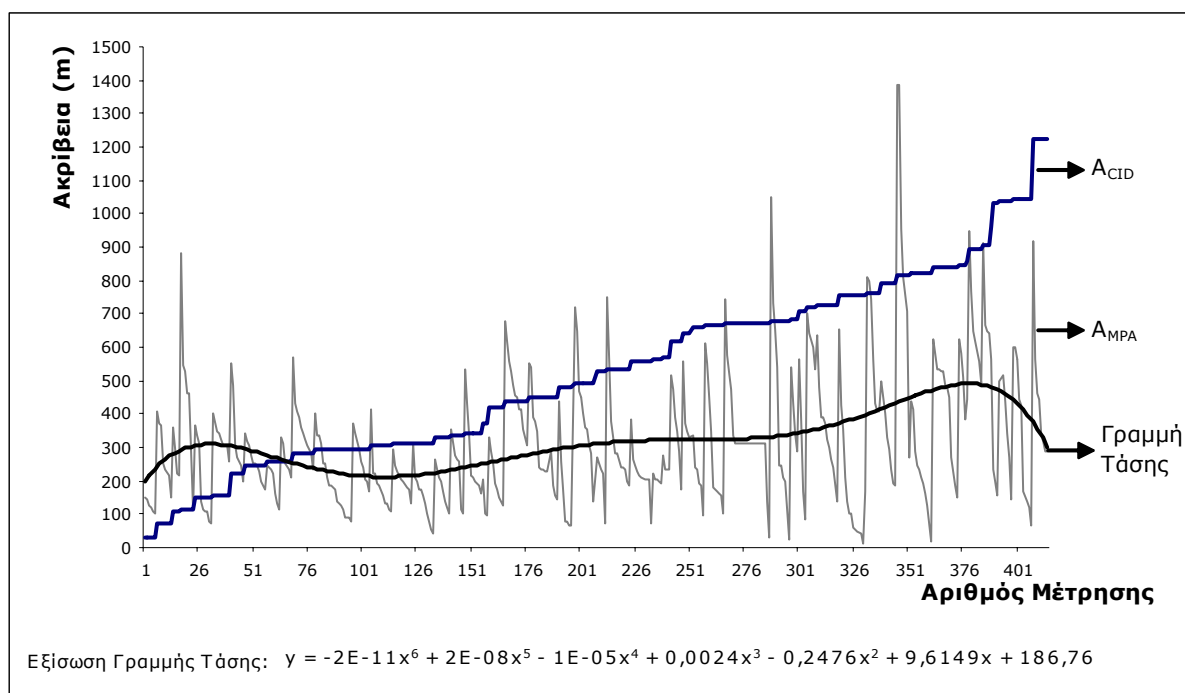
Τρία είναι τα μεγέθη που μας ενδιαφέρουν στη συγκεκριμένη παράγραφο:

- 1) Η ακρίβεια της μεθόδου CID (A_{CID})
- 2) Η ακρίβεια της μεθόδου Μ.Ρ.Α (A_{MPA})
- 3) Η συγκριτική βελτίωση της μεθόδου Μ.Ρ.Α ($B = A_{CID} - A_{MPA}$)

Όταν το B είναι αρνητικό τότε δεν έχουμε βελτίωση αλλά χειρότερη απόδοση της μεθόδου Μ.Ρ.Α συγκριτικά με τη μέθοδο CID.

Στο Σχήμα 5.6 παρουσιάζονται γραφικά τα ακόλουθα μεγέθη:

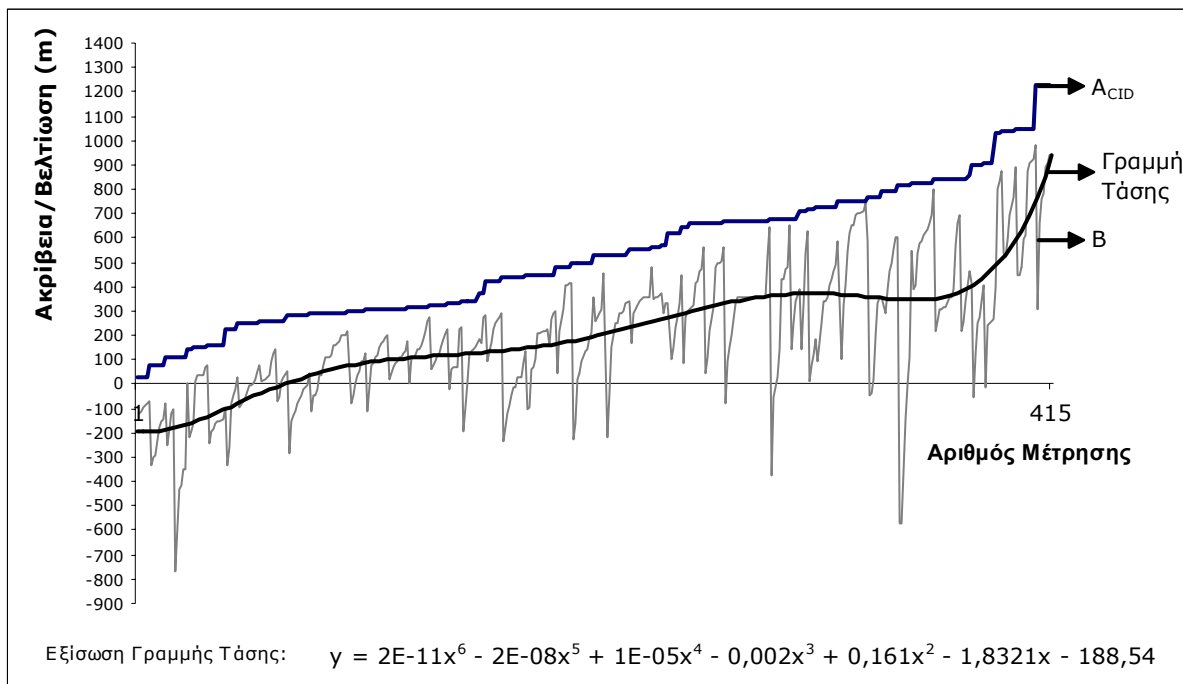
- a) A_{CID} με αύξουσα σειρά εμφάνισης
- b) Η αντίστοιχη τιμή της ακρίβειας A_{MPA} όπου έχει σχεδιαστεί και η γραμμή τάσης.



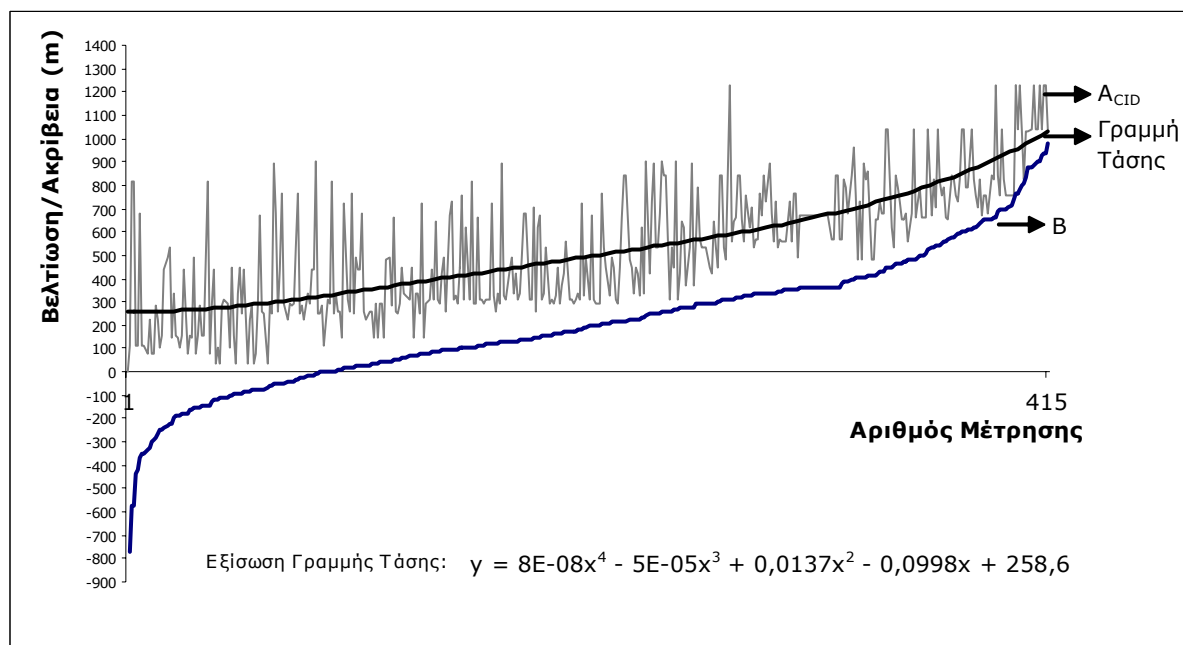
ΣΧΗΜΑ 5.6

Στο Σχήμα 5.7 παρουσιάζονται γραφικά τα ακόλουθα μεγέθη:

- a) **A_{CTID}** με αύξουσα σειρά εμφάνισης
- b) Η αντίστοιχη τιμή της συγκριτικής βελτίωσης **B** της οποίας έχει σχεδιαστεί και η γραμμή τάσης.



ΣΧΗΜΑ 5.7



ΣΧΗΜΑ 5.8

Τέλος, στο Σχήμα 5.8 παρουσιάζονται γραφικά τα ακόλουθα μεγέθη:

- Η τιμή της συγκριτικής βελτίωσης **B** με αύξουσα σειρά εμφάνισης
- Η αντίστοιχη τιμή της ακρίβειας **A_{CID}** όπου έχει σχεδιαστεί και η γραμμή τάσης.

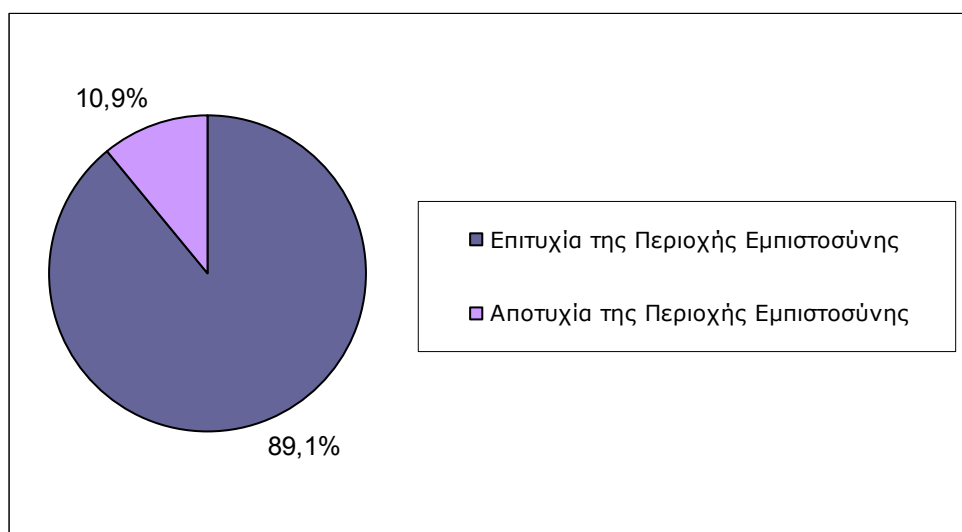
Εξετάζοντας τα τρία προηγούμενα Σχήματα, παρατηρούμε τα εξής:

- Στο Σχήμα 5.6 η γραμμή τάσης τέμνει την γραμμή του CID σε ένα μόνο σημείο (259m).
- Στο Σχήμα 5.7 η γραμμή τάσης τέμνει τον άξονα x σε ένα σημείο (x_1). Η ακρίβεια CID που αντιστοιχεί στο x_1 είναι 259m.
- Στο Σχήμα 5.8 η γραμμή **B** τέμνει τον άξονα x σε ένα σημείο (x_2). Η τιμή για την γραμμή τάσης του CID στο σημείο x_2 είναι 300m (μέχρι το σημείο x_2 , η συγκριτική βελτίωση **B**, έπαιρνε αρνητικές τιμές).

Το συμπέρασμα που μπορούμε να εξαγάγουμε είναι ότι μέχρι τα 250-300 μέτρα, ο αλγόριθμος που υλοποιήσαμε (M.P.A) λειτουργεί προβληματικά. Μετά τα 300 μέτρα όμως, εμφανίζει πολύ καλή συγκριτική βελτίωση, η οποία μπορούμε να πούμε ότι στις μεγάλες αποστάσεις αυξάνεται εκθετικά (βλ. Σχήμα 5.8).

Η ΑΚΡΙΒΕΙΑ ΤΗΣ ΠΕΡΙΟΧΗΣ ΕΜΠΙΣΤΟΣΥΝΗΣ (TRUST AREA):

Στο σύνολο των μετρήσεων που πραγματοποιήθηκαν, το ποσοστό επιτυχίας της Περιοχής Εμπιστοσύνης (TRUST AREA) ήταν 89.1%



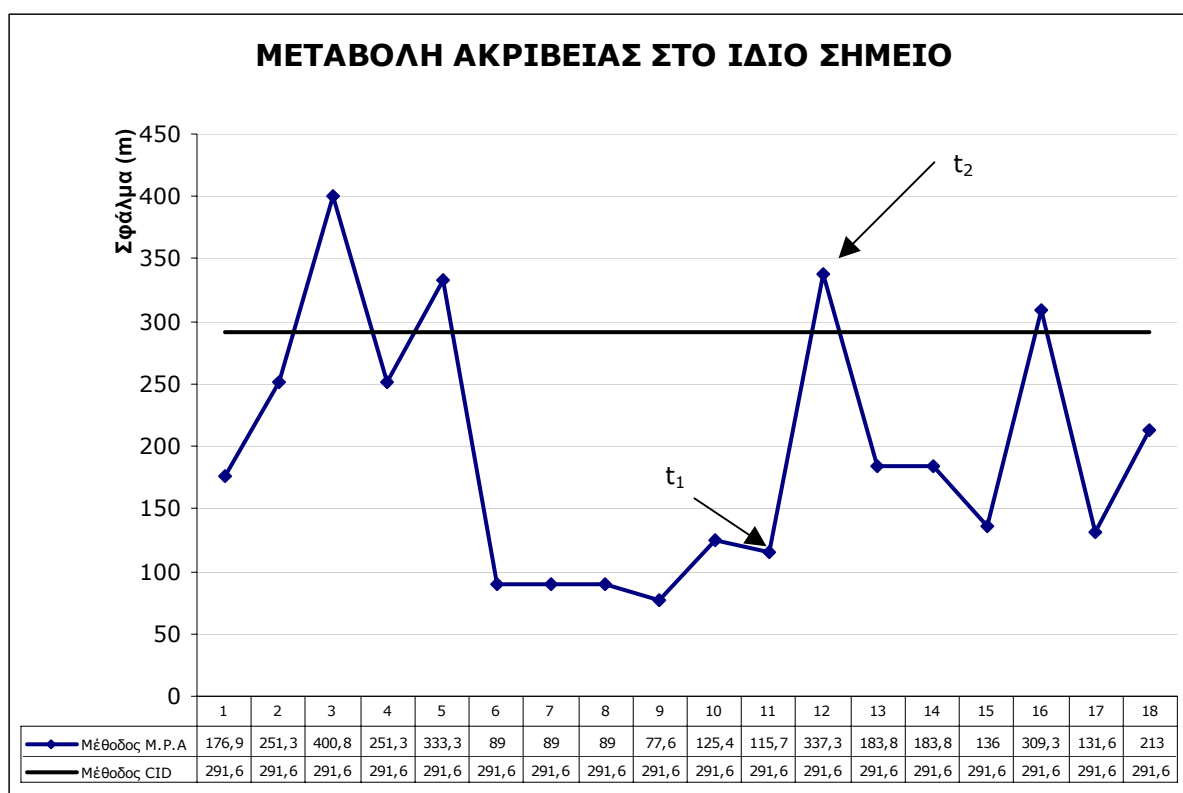
ΣΧΗΜΑ 5.9

Αυτό το ποσοστό είναι πολύ κοντά στο επιδιωκόμενο ποσοστό (90%) που είχαμε θέσει ως όριο, οπότε μπορούμε να θεωρήσουμε ότι η περιοχή εμπιστοσύνης αποδίδει επιτυχώς και στα αναμενόμενα επίπεδα.

Παρόλ' αυτά, υπάρχει η ανάγκη για βελτίωση της Περιοχής Εμπιστοσύνης, κυρίως στον τομέα της ελαχιστοποίησης του χώρου που καλύπτει. Αυτό βέβαια μπορεί να επιτευχθεί μάλλον δύσκολα και για το σκοπό αυτό χρειάζονται περαιτέρω δοκιμές και έρευνα, κάτι που δεν μπορεί να πραγματοποιηθεί στα πλαίσια της συγκεκριμένης Διπλωματικής Εργασίας.

ΜΕΤΑΒΟΛΗ ΤΗΣ ΑΚΡΙΒΕΙΑΣ ΤΗΣ ΜΕΘΟΔΟΥ Μ.Ρ.Α ΣΤΟ ΙΔΙΟ ΣΗΜΕΙΟ

Είναι προφανές από το Σχήμα 5.10 ότι η ακρίβεια σε μια δεδομένη και σταθερή θέση του Κ/Τ μεταβάλλεται με τον χρόνο. Οι μεταβολή μπορεί να είναι τόσο μεγάλη, ώστε ενώ κάποια χρονική στιγμή t_1 ο αλγόριθμος δρα βελτιωτικά της μεθόδου CID, μια άλλη χρονική στιγμή t_2 ο αλγόριθμος δίνει χειρότερο αποτέλεσμα από τη μέθοδο CID.



ΣΧΗΜΑ 5.10

Αυτό οφείλεται κυρίως στην μεταβολή της ισχύος λήψης του σήματος από την Κυψέλη Εξυπηρέτησης και από τους Γειτονικούς Σταθμούς Βάσης, λόγω φαινομένων εξασθένησης, σκίασης, πολυδιαδρομικής διάδοσης και παρεμβολών.

Σε αντίθεση με τη μέθοδο Μ.Ρ.Α, η μέθοδος CID δεν εμφανίζει τέτοια προβλήματα σε καμία περίπτωση, αφού βασίζεται αποκλειστικά στην γεωγραφική θέση της Κυψέλης Εξυπηρέτησης και τίποτ' άλλο.

5.3.2. ΤΕΛΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ

Κλείνοντας αυτό το κεφάλαιο, μπορούμε να αναφέρουμε τα 4 κυριότερα σημεία του Αλγόριθμου Εντοπισμού Θέσης που παρουσιάζουν το μεγαλύτερο ενδιαφέρον:

- 1) Η μέθοδος M.P.A βελτιώνει κατά 40% την ακρίβεια της μεθόδου CID.
- 2) Σε αποστάσεις μικρότερες από τα 300m η μέθοδος δεν αποδίδει στα επιθυμητά πλαίσια (εμφανίζει χειρότερη ακρίβεια από τη μέθοδο CID).
- 3) Υπάρχει χρονική διακύμανση της ακρίβειας της μεθόδου σε μια δεδομένη γεωγραφική θέση του Κ/Τ.
- 4) Η περιοχή εμπιστοσύνης (TRUST AREA) εμφανίζει ποσοστό επιτυχίας 89.1%, αλλά το εμβαδόν που καλύπτει είναι αρκετά μεγάλο (μεγαλύτερο από το επιθυμητό).

Αρνητικοί Παράγοντες που επηρέασαν τη μέθοδο M.P.A:

- ✓ Στο μοντέλο Ραδιοκάλυψης των Okumura/Hata εισάγουμε σταθερό ύψος κεραίας Δέκτη (2m), γεγονός που δεν είναι πάντοτε αληθές. Επίσης, μας δίνεται το ύψος των κεραιών των Σταθμών Βάσης από την επιφάνεια του εδάφους, επομένως δεν συνυπολογίζονται οι υψομετρικές διαφορές λόγω της γεωλογικής διαμόρφωσης του εδάφους. Για παράδειγμα αν το Κ/Τ λαμβάνει σήμα από έναν Σταθμό Βάσης που βρίσκεται σε κάποιο βουνό, τότε η ακτίνα HATA Radius που υπολογίζουμε θα έχει σημαντικό σφάλμα (συνέβη κατά τη διάρκεια των μετρήσεων).
- ✓ Εκτός του μοντέλου Okumura/Hata, υπάρχουν και άλλα μοντέλα ραδιοκάλυψης, πιο εξειδικευμένα στο ελληνικό περιβάλλον (με περισσότερους τύπους περιοχών, κλπ.), τα οποία όμως δεν ήταν δυνατόν να βρεθούν για να τα χρησιμοποιήσουμε. Ενδεχομένως χρησιμοποιώντας τέτοια μοντέλα, η ακρίβεια της μεθόδου να βελτιωνόταν σημαντικά.
- ✓ Η Βάση Δεδομένων που χρησιμοποιήθηκε ήταν ελλειπής, με αποτέλεσμα να δυσχεραίνει τη διαδικασία του εντοπισμού.

Παρόλ' αυτά, η απόδοση του Αλγόριθμου Εντοπισμού Θέσης κρίνεται ικανοποιητική και σε γενικές γραμμές εντός των αναμενόμενων προσδοκιών. Άλλωστε το αντικείμενο του Γεωγραφικού Προσδιορισμού Θέσης του Κ/Τ μέσω του Δικτύου είναι κάτι καινούργιο στον χώρο των Τηλεπικοινωνιών και χρίζει μεγάλης μελέτης, ανάλυσης και πολλών πειραμάτων και δοκιμών για να βρεθεί μια βέλτιστη μέθοδος που να ικανοποιεί όλες τις απαιτήσεις, τόσο των Παροχών Υπηρεσιών όσο και των Συνδρομητών.



ΠΑΡΑΡΤΗΜΑ Α

ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΚΩΔΙΚΑ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΠΑΡΑΡΤΗΜΑ Α

| | |
|------------------------------|------------|
| A.1. GLOBAL VARIABLES | 141 |
| A.1.1. "modVariables" Module | 141 |
| A.2. FORMS | 143 |
| A.2.1. frmStart | 143 |
| A.2.2. frmModeNormal | 144 |
| A.2.3. frmModeUser | 152 |
| A.2.4. frmModeFull | 155 |
| A.2.5. frmFinal | 167 |
| A.2.6. frmBrowse | 171 |
| A.2.7. frmSavePicture | 173 |
| A.2.8. frmGPS | 175 |
| A.2.9. frmSplash2 | 178 |
| A.3. MODULES | 179 |
| A.3.1. ModPropagation | 180 |
| A.3.2. UTMWGS84 | 180 |
| A.3.3. ModSub | 187 |
| A.4. INDEX | 223 |

A.1. GLOBAL VARIABLES

A.1.1. "modVariables" Module

Option Explicit

```
Global ComPrt As String           'The variable that controls the com port
                                  'which is used by the M/T

Global RepTimes As Integer        'A counter used for recording Data from M/T

Global Metriseis() As Long        'The Matrix used for storing data

Global cid() As Long              'The Matrix Used for the Cell ID Measurement

Global TA() As Long               'The Matrix Used for the
                                  'Timing Advance Measurement

Global cntr As Integer            'A Counter that is used when recording Data

Global Final(1 To 7, 1 To 2) As Long 'The Matrix where Mean values are Stored

Global FinalOk() As Variant        'The Final Matrix which has as many rows as
                                  'the available data

Global FinalFull() As Variant

Global FinalTA As Long            'The Final value for the Timing Advance

Global FinalCID As Long           'The Final value for the Cell ID

Global GraphMatr() As Variant     'Matrix used in M/T Location and in drawing
                                  'the results.

Global metraw As Integer          'A counter

Global stoptry As Integer          'A Counter which helps the program to know
                                  'when to stop the measurements

Global DataMax As Integer         'It counts the number of Cells Available
                                  '(including the Serving Cell)

Global FullDataMax As Integer

Global arouklas As Variant

Global DataBaseError As Boolean   'Indicates Whether there is a problem with the
                                  'DatabaseFind Procedure.

Global CalcX As Double            'The Calculated Final Position in UTM

Global CalcY As Double            'The Calculated Final Position in UTM

Global CalcLon As Double          'The Calculated Final Position (Longitude)

Global CalcLat As Double          'The Calculated Final Position (Latitude)

Global Starttime As Variant       'Start time: Calculation Begins

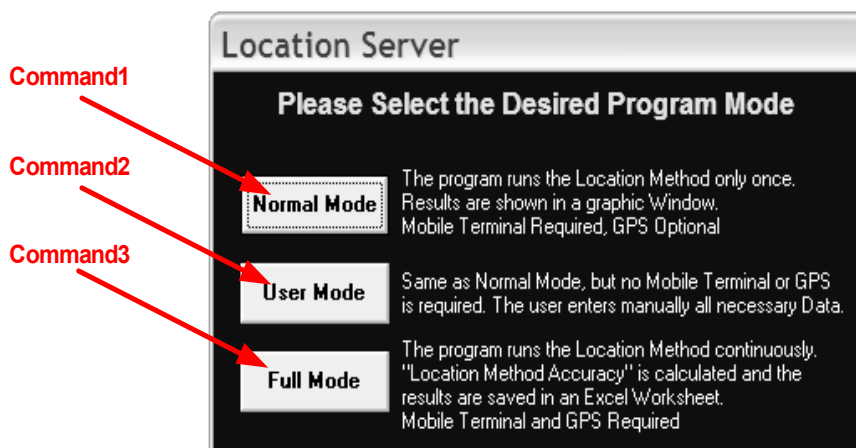
Global Stoptime As Variant        'Stop time: Calculation Ends

Global Port As Integer            'COM port of GPS
```

| | |
|-------------------------------------|---|
| Global NMEA As Integer | 'Type of NMEA format |
| Global Error As Double | 'Calculation Error |
| Global MicroBTS As Integer | 'Number of MicroBTS (used in NotLocate SUB) |
| Global ShowText As Variant | |
| Global DataLinkFile As String | 'The Address where the Data Link File is 'located |
| Global OpenWhat As String | 'Variable used when switching between 'different program modes |
| Global LocError As Double | 'Used in Full Mode |
| Global GPSDistance As Double | 'Used in Full Mode |
| Global Sector As Boolean | 'If True, Trust Area is a Sector |
| Global StartSector As Double | 'Variable used in CheckNeighbourHata SUB |
| Global StopSector As Double | 'Variable used in CheckNeighbourHata SUB |
| Global OutRad As Double | 'Variable used to define the outer 'Radius of the TRUST AREA |
| Global TRUST_TYPE As String | 'TRUST AREA Parameter |
| Global TRUST_LON As Double | 'TRUST AREA Parameter |
| Global TRUST_LAT As Double | 'TRUST AREA Parameter |
| Global TRUST_CIRCLE_BIG As Double | 'TRUST AREA Parameter |
| Global TRUST_CIRCLE_SMALL As Double | 'TRUST AREA Parameter |
| Global TRUST_ANGLE_START As Double | 'TRUST AREA Parameter |
| Global TRUST_ANGLE_STOP As Double | 'TRUST AREA Parameter |
| Global heavy() As Double | 'The Matrix used in the Location Algorithm 'Filled in CheckNeighbourHata SUB |
| Global NeighbourCross() As Double | 'In this Matrix the program saves the 'Cross Section points of the two circles |

A.2. FORMS

A.2.1. frmStart

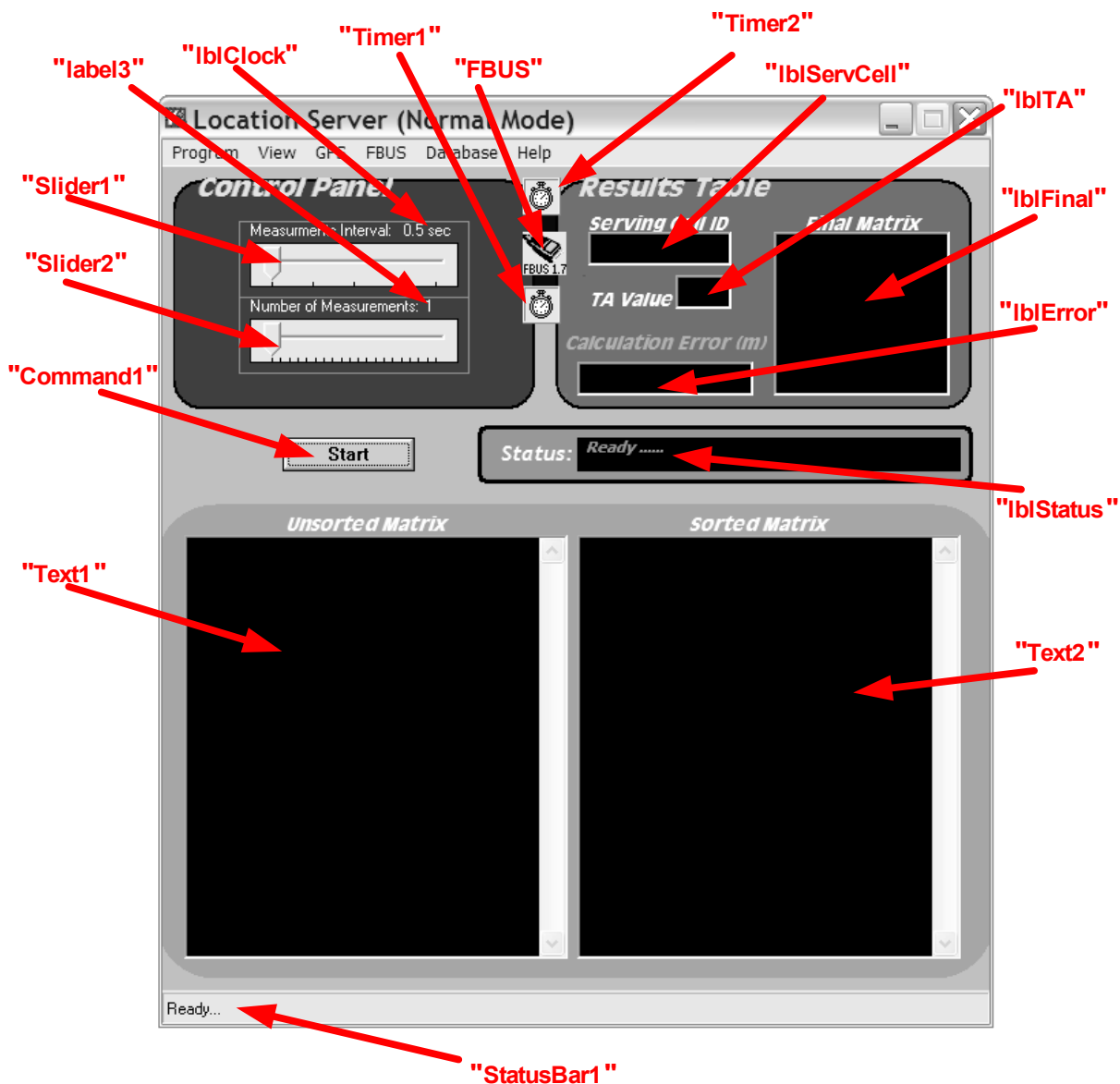


```
Private Sub Command1_Click()  
  
    Load frmModeNormal  
    frmModeNormal.Visible = True  
    frmModeNormal.Enabled = True  
    Unload Me  
  
End Sub
```

```
Private Sub Command2_Click()  
  
    Load frmModeUser  
    frmModeUser.Visible = True  
    frmModeUser.Enabled = True  
    Unload Me  
  
End Sub
```

```
Private Sub Command3_Click()  
  
    Load frmModeFull  
    frmModeFull.Visible = True  
    frmModeFull.Enabled = True  
    Unload Me  
  
End Sub
```

A.2.2. frmModeNormal



Option Explicit

```
Private Sub Form_Load()

    NMEA = 2
    Port = 1
    frmModeNormal.Height = 10335
    StatusBar1.SimpleText = "Ready..."
    ComPrt = "COM1F" 'The default Com Port
    lblClock.Caption = "0.5 sec" 'The Default Interval
    Label3.Caption = CStr(Slider2.Value)
    RepTimes = CInt(Slider2.Value)
End Sub
```



```
Private Sub mnuDataLink_Click()  
    OpenWhat = "Normal"  
    frmBrowse.Visible = True  
    frmModeNormal.Enabled = False  
End Sub
```

```
Private Sub mnuFBUScom1_Click()  
    If mnuFBUScom1.Checked = False Then  
        mnuFBUScom1.Checked = True  
        mnuFBUScom2.Checked = False  
        ComPrt = "COM1F" 'COM1 Is Selected by the User  
    End If  
End Sub
```

```
Private Sub mnuFBUScom2_Click()  
    If mnuFBUScom2.Checked = False Then  
        mnuFBUScom2.Checked = True  
        mnuFBUScom1.Checked = False  
        ComPrt = "COM2F" 'COM2 Is Selected by the User  
    End If  
End Sub
```

```
Private Sub mnuGPSNMEA21_Click()  
    If mnuGPSNMEA21.Checked = False Then  
        mnuGPSNMEA21.Checked = True  
        NMEA = 2  
        mnuGPSNMEA15.Checked = False  
    End If  
End Sub
```

```
Private Sub mnuGPSNMEA15_Click()  
    If mnuGPSNMEA15.Checked = False Then  
        mnuGPSNMEA15.Checked = True  
        NMEA = 1  
        mnuGPSNMEA21.Checked = False  
    End If  
End Sub
```

```
Private Sub mnuGPSport1_Click()  
    If mnuGPSport1.Checked = False Then  
        mnuGPSport1.Checked = True  
        Port = 1  
        mnuGPSport2.Checked = False  
    End If  
End Sub
```

```
Private Sub mnuGPSport2_Click()  
    If mnuGPSport2.Checked = False Then  
        mnuGPSport2.Checked = True  
        Port = 2  
        mnuGPSport1.Checked = False  
    End If  
End Sub
```

```
Private Function CheckPorts() As Boolean
```

```
Dim Response As Integer
```

```
    If (mnuFBUScom1.Checked = True And mnuGPSport1.Checked = True) Or _  
        (mnuFBUScom2.Checked = True And mnuGPSport2.Checked = True) Then  
  
        Response = _  
            MsgBox("You have selected the same Port (COM" & Port _  
                & ") for both the Mobile Device and the GPS." & _  
                "Press OK to continue, or Cancel to change your settings and Retry.", _  
                vbOKCancel + vbExclamation + vbDefaultButton2, "ATTENTION")  
  
        If Response = vbOK Then  
            CheckPorts = True  
        Else  
            CheckPorts = False  
        End If  
  
    Else: CheckPorts = True  
  
    End If  
End Function
```

```
Private Sub mnuGPSEnabled_Click()
```

```
    If mnuGPSEnabled.Caption = "Enable" Then  
        If CheckPorts = True Then  
            mnuGPSEnabled.Caption = "Disable"  
            Call frmGPS.Initialize  
            frmGPS.Enabled = True  
            frmGPS.Visible = True  
            mnuGPSport.Enabled = False  
            mnuGPSNMEA.Enabled = False  
        End If  
    Else  
        mnuGPSport.Enabled = True  
        mnuGPSNMEA.Enabled = True  
        mnuGPSEnabled.Caption = "Enable"  
        Unload frmGPS  
    End If  
End Sub
```

```
Private Sub mnuHelpAbout_Click(Index As Integer)
```

```
    frmAbout.Visible = True  
    Load frmAbout  
End Sub
```

```
Private Sub mnuPrgExit_Click()  
  
'Make sure user really wants to exit  
Dim Response As Integer  
  
Response = _  
    MsgBox("Are you sure you want to exit the Program?", vbYesNo + _  
        vbCritical + vbDefaultButton2, "Exit Program")  
  
If Response = vbNo Then  
    Exit Sub  
Else  
    End  
End If  
  
End Sub
```

```
Private Sub mnuPrgMode_Click()  
  
Load frmStart  
frmStart.Visible = True  
frmStart.Enabled = True  
Unload Me  
  
End Sub
```

```
Private Sub mnuViewDebug_Click()  
  
If mnuViewDebug.Checked = False Then  
    mnuViewDebug.Checked = True  
    Picture1.Visible = False  
    frmModeNormal.Height = 10335  
Else  
    mnuViewDebug.Checked = False  
    Picture1.Visible = True  
    frmModeNormal.Height = 4950  
End If  
  
End Sub
```

```
Private Sub Slider1_Click()  
'The values that are stored for the Measurement Interval  
  
If Slider1.Value = 0 Then  
    Timer1.Interval = 1  
    lblClock.Caption = "0.5 sec"  
ElseIf Slider1.Value = 1 Then  
    Timer1.Interval = 500  
    lblClock.Caption = "1 sec"  
ElseIf Slider1.Value = 2 Then  
    Timer1.Interval = 1500  
    lblClock.Caption = "2 sec"  
ElseIf Slider1.Value = 3 Then  
    Timer1.Interval = 2500  
    lblClock.Caption = "3 sec"  
ElseIf Slider1.Value = 4 Then  
    Timer1.Interval = 3500
```

```
        lblClock.Caption = "4 sec"  
    End If
```

```
End Sub
```

```
Private Sub Slider2_Click()
```

```
    'Slider2 sets the Number of Samples, (variable= RepTimes)  
    RepTimes = Slider2.Value  
    Label3.Caption = CStr(Slider2.Value)
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
Dim Response As Integer  
Dim Answer As Variant
```

```
    Unload frmFinal
```

```
    'Avoid showing results buttons in case the location process has been invoked before
```

```
    'Initialize DataBaseError Variable
```

```
    DataBaseError = False  
    lblTA.Caption = ""  
    lblServCell.Caption = ""  
    lblFinal.Caption = ""  
    lblError.Caption = ""
```

```
    If mnuGPSEnabled.Caption = "Disable" Then
```

```
        GPSlat = CDb1(frmGPS.lblLat)  
        GPSlon = CDb1(frmGPS.lblLon)
```

```
        If GPSlat = 0 Then
```

```
            Response = MsgBox("The GPS doesn't generate a Position, " & _  
                "do you want to enter position manually?", vbYesNo + vbQuestion, _  
                "Caution")
```

```
            If Response = vbYes Then
```

```
                Answer = InputBox("Type the Latitude(format: xx.xxxxx) " & _  
                    vbCrLf & "eg. 38.55555", "Manual Latitude")
```

```
                If (Answer <> "") Then
```

```
                    GPSlat = CDb1(Answer)
```

```
                    Answer = InputBox("Type the Longitude(format: xxx.xxxxx, _  
                        eg." & " 023.55555", "Manual Longitude")
```

```
                    If (Answer <> "") Then
```

```
                        GPSlon = CDb1(Answer)
```

```
                        GPS = True
```

```
                    Else
```

```
                        GPS = False
```

```
                    End If
```

```
                Else
```

```
                    GPS = False
```

```
                End If
```

```
            Else
```

```
                GPS = False
```

```
            End If
```

```
        Else
```

```
            GPS = True
```

```
        End If
```

```

    If (mnuFBUScom1.Checked = True And mnuGPSport1.Checked = True) Or _
        (mnuFBUScom2.Checked = True And mnuGPSport2.Checked = True) Then
        Unload frmGPS
        MsgBox "Please Disconnect now the GPS and Connect the Mobile" & _
            Device. Press OK when Done.", vbOKOnly, "Device Change"

    End If
Else
    GPS = False
End If

cntr = 1 'Set the counter for the Matrix Metriseis()
        '(used in Timer1.Enabled)

'Connect with Mobile Terminal
lblStatus.Caption = " Connecting..."
lblStatus.Refresh
StatusBar1.SimpleText = "Establishing Connection..."
FBUS.SetLicense "Mobile FBUS 1.7", "xxxxxxxxxxxxxxxxxxxx"
FBUS.RaiseError = False
FBUS.Connect ComPrt 'ComPrt is the variable for the FBUS COM Port Number

If FBUS.IsAlive = True Then 'Check if Connection has been established
    StatusBar1.SimpleText = "Connected"
    lblStatus.Caption = " Getting Data From Mobile Equipment..."
    FBUS.RaiseError = True
    Command1.Enabled = False

    'Declaring the size of Matrixes
    ReDim Metriseis(1 To (RepTimes * 2), 1 To 7) As Long
    ReDim cid(1 To RepTimes) As Long
    ReDim TA(1 To RepTimes) As Long

    'Start Counters
    metraw = 0
    stoptry = 0

    'Start Recording Measurements (Timer1 Control)
    Timer1.Enabled = True
    StatusBar1.SimpleText = "Getting Data from Mobile Phone..."

Else
    Beep
    MsgBox "Check your Cable Connection & COM Port Settings!", vbOKOnly, _
        "Failed to Connect"

    StatusBar1.SimpleText = "Failed to Connect..."
    lblStatus.Caption = "Failed to Connect..."

End If

```

End Sub

```

Private Sub Timer1_Timer()

Dim j As Integer
Dim aris As Variant
Dim BTS(1 To 3) As Variant
Dim i As Integer
Dim TimingQual As Variant
Dim CellID As Variant

```

```

metraw = metraw + 1 'Shows how many times the Procedure has been Executed

'Get Data from Netmonitor

For i = 1 To 3
    BTS(i) = FBUS.NetmonitorScreen("0" & CStr(i + 2))
Next i

TimingQual = FBUS.NetmonitorScreen("01")
CellID = FBUS.NetmonitorScreen("11")

'Store Data in Metriseis() Matrix

For i = 1 To 7
    Metriseis(cntr, i) = CLng(Val(Mid(BTS(1 + Int((i - 1) / 3)), _
        (1 + 13 * (i - (1 + Int((i - 1) / 3) * 3))), 3)))

    Metriseis(cntr + 1, i) = CLng(Val(Mid(BTS(1 + Int((i - 1) / 3)), _
        (7 + 13 * (i - (1 + Int((i - 1) / 3) * 3))), 3)))

    If Metriseis(cntr + 1, i) > 0 Then
        Metriseis(cntr + 1, i) = Metriseis(cntr + 1, i) * (-1)
    End If
Next i

cid(1 + (cntr - 1) / 2) = CLng(Val(Mid(CellID, 40, 6)))
TA(1 + (cntr - 1) / 2) = CLng(Val(Mid(TimingQual, 17, 2)))

'The case where there is no value for 2nd Neighbour is not to count
If Metriseis(cntr, 3) > 0 Then
    cntr = cntr + 2
Else
    stoptry = stoptry + 1
End If

StatusBar1.SimpleText = "Getting Data from Mobile Phone..." & CStr(metraw)

'Case where there isn't any useful value after five tries
If stoptry = 5 Then
    MsgBox "Can't get the signal data, too weak", vbOKOnly, _
        "Failed to Collect Data"
    Timer1.Enabled = False
End If

'Check if Timer1 will stop or not. When the timer Stops, the Unsorted Matrix is
'being Print on Screen and the SortMatrix Procedure is being called

If RepTimes = ((cntr + 1) / 2 - 1) And (stoptry < 5) Then
    FBUS.Disconnect
    lblStatus.Caption = "  Sorting/Calculating Mean Values....."
    lblStatus.Refresh
    StatusBar1.SimpleText = "Sorting Matrix..."

    'Print Data to Screen

    For i = 1 To RepTimes * 2
        For j = 1 To 7
            aris = aris & Format(CStr(Metriseis(i, j)), "#####")
        Next j
        aris = aris & vbCrLf
        If i Mod 2 = 0 Then
            aris = aris & vbCrLf
        End If
    Next i

```

```
End If

Next i
Text1.Text = aris
Starttime = Now
Call ModSub.SortMatrix 'Sorting of the unsorted Data
Text2.Text = ShowText

Call ModSub.ProcessData 'Calculate Mean Values
lblFinal.Caption = ShowText

lblTA.Caption = CStr(FinalTA)
lblServCell.Caption = CStr(FinalCID)

Call ModSub.StoreValues 'Store results as variables for the Database Search

lblStatus.Caption = "Searching Database for Neighbours..."
lblStatus.Refresh
Call ModSub.Databasefind 'Search for given Cells in Database to get
                        'the required Information
Call ModSub.ProcessDist

If DataBaseError = False Then
    If ModSub.MicroCheck Then
        Zoom = 4
        Call ModSub.NotLocate
        lblError.Caption = Format(Error, "###0.0")
        Call ModSub.DrawMicro
    Else

        lblStatus.Caption = " Locating Mobile Terminal...."
        Zoom = 4 'Variable Declared in Module UTMWGS84
        Call ModSub.TAHATAMatch

        Call ModSub.CalcDistances
        lblError.Caption = Format(Error, "###0.0")
        Call ModSub.DrawGraph

        lblStatus.Caption = " Finished...."
        lblStatus.Refresh
    End If
Else
    lblStatus.Caption = " Finished...."
End If

Command1.Enabled = True
Timer1.Enabled = False

End If

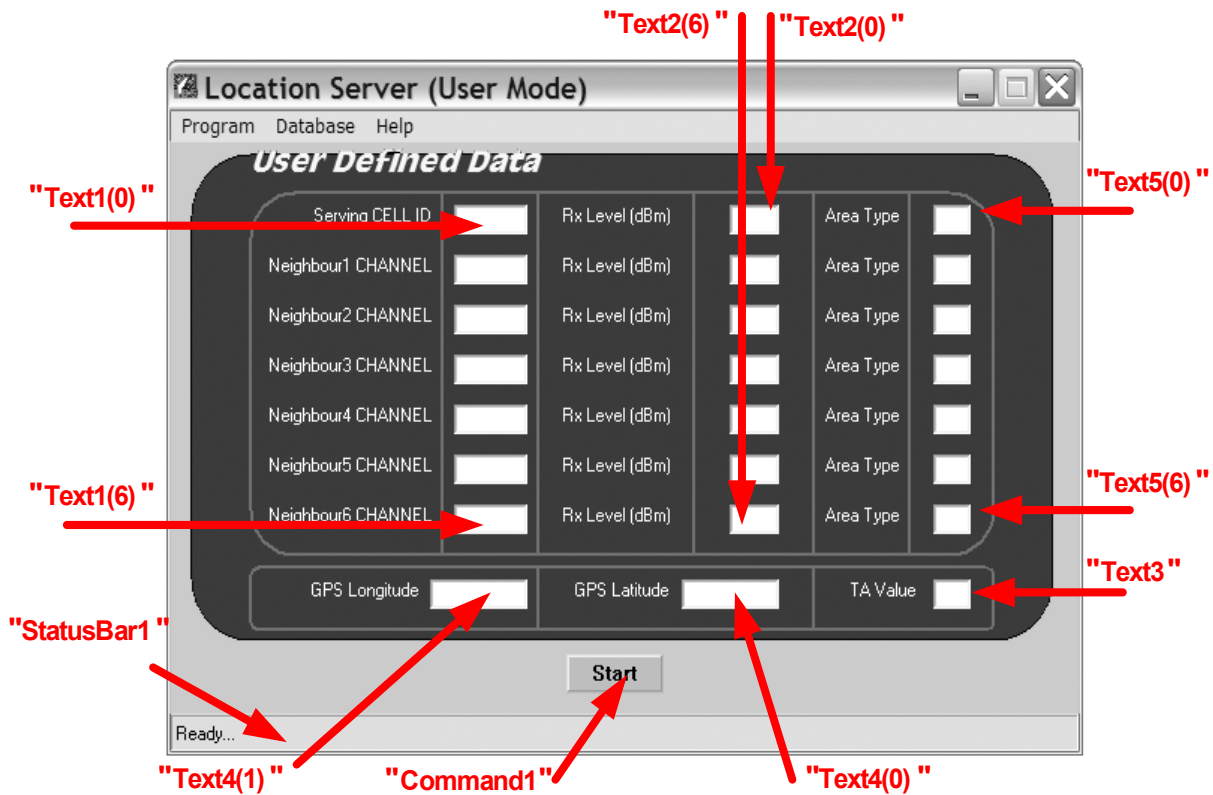
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)

    StatusBar1.SimpleText = "Exiting..."
    Call frmSplash2.Clos
    Call frmFinal.Clos
    Unload frmGPS
```

```
End Sub
```

A.2.3. frmModeUser



Option Explicit

```
Private Sub Form_Load()
    StatusBar1.SimpleText = "Ready..."
End Sub
```

```
Private Sub mnuDataLink_Click()
    OpenWhat = "User"
    frmBrowse.Visible = True
    frmModeUser.Enabled = False
End Sub
```

```
Private Sub mnuHelpAbout_Click(Index As Integer)
    frmAbout.Visible = True
    Load frmAbout
End Sub
```



```
Private Sub mnuPrgExit_Click()  
  
'Make sure user really wants to exit  
Dim Response As Integer  
  
    Response = MsgBox("Are you sure you want to exit the Program?", vbYesNo + _  
        vbCritical + vbDefaultButton2, "Exit Program")  
  
    If Response = vbNo Then  
        Exit Sub  
    Else  
        End  
    End If  
  
End Sub
```

```
Private Sub Command1_Click()  
  
Dim Response As Integer  
Dim Answer As Variant  
Dim i As Integer  
  
    Unload frmFinal  
    Unload frmSplash2  
'Avoid showing results buttons in case the location process has been invoked  
before  
  
'Initialize DataBaseError Variable  
    DataBaseError = False  
  
    Starttime = Now  
    GPS = True  
  
'Store the TA Value the User Has Entered  
    FinalTA = CInt(Text3.Text)  
  
'Store the GPS Fix point Entered by the User  
    GPSlat = CDb1(Text4(0).Text)  
    GPSlon = CDb1(Text4(1).Text)  
  
'Calculate DataMax for User Defined Input  
    DataMax = 7  
    For i = 1 To 6  
        If CLng(Text1(i).Text) = 0 Then  
            DataMax = i  
            i = 6  
        End If  
    Next i  
  
    ReDim FinalOk(1 To DataMax, 1 To 16) As Variant  
    FinalCID = CLng(Text1(0).Text)  
    For i = 0 To (DataMax - 1)  
        FinalOk(i + 1, 1) = CLng(Text1(i).Text)  
        FinalOk(i + 1, 2) = CLng(Text2(i).Text)  
  
    Next i  
  
    Call ModSub.Databasefind
```

```
'Call Databasefind 'Search for given Cells in Database to get
'the required Information
Call ModSub.ProcessDist
If DataBaseError = False Then

    If ModSub.MicroCheck Then
        Zoom = 4
        Call ModSub.NotLocate
        Call ModSub.DrawMicro
    Else

        Zoom = 4 'Variable Declared in Module UTMWGS84
        Call ModSub.TAHATAMatch
        Call ModSub.CalcDistances
        Call ModSub.DrawGraph
    End If

End If
```

End Sub

```
Private Sub Form_Unload(Cancel As Integer)

    StatusBar1.SimpleText = "Exiting..."
    Call frmSplash2.Clos
    Call frmFinal.Clos
```

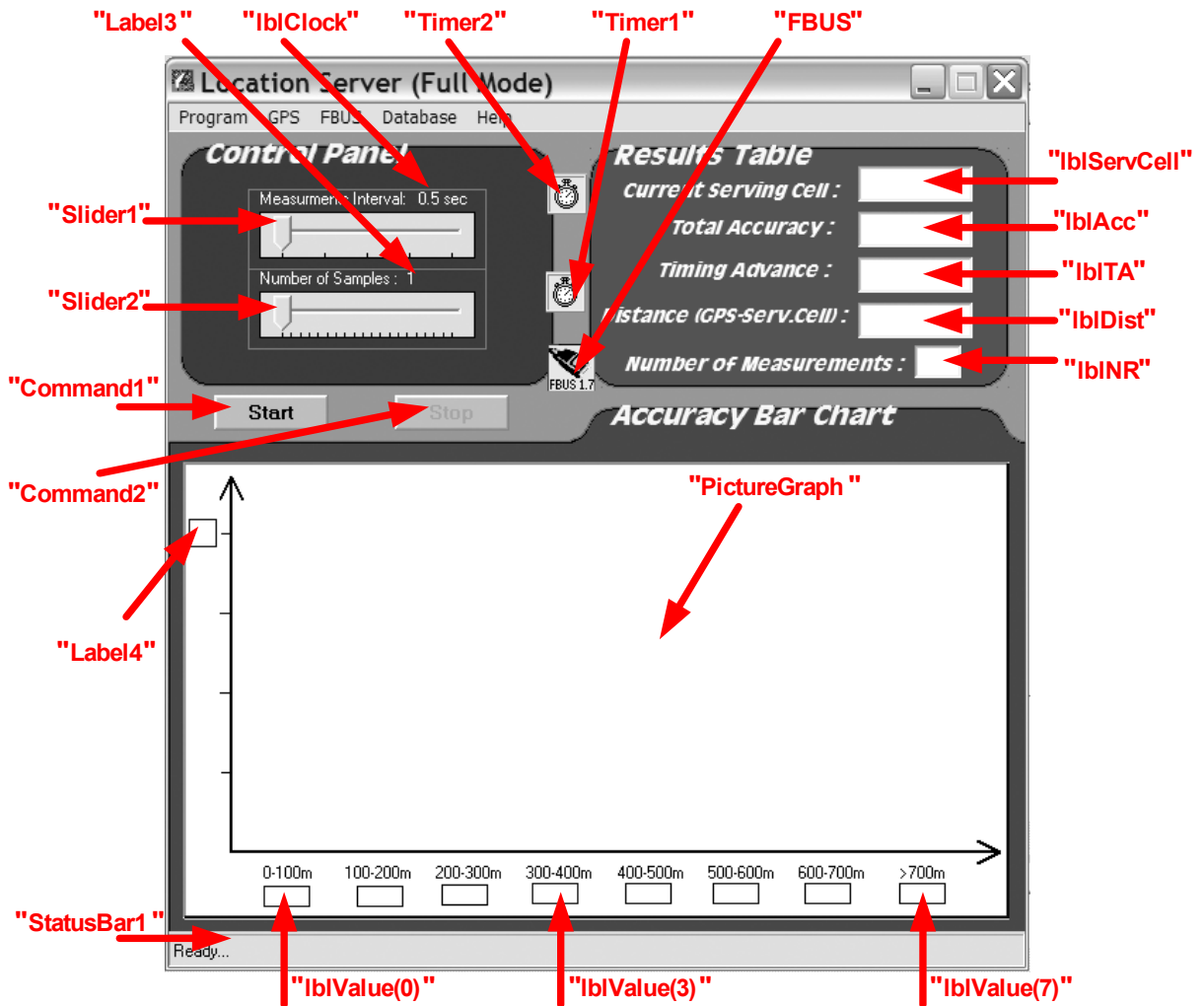
End Sub

```
Private Sub mnuPrgMode_Click()

    Load frmStart
    frmStart.Visible = True
    frmStart.Enabled = True
    Unload Me
```

End Sub

A.2.4. frmModeFull



Option Explicit

```

Dim xlApp As Excel.Application
Dim xlBook As Excel.Workbook
Dim xlSheet As Excel.Worksheet
Dim ExcelCntr As Long
Dim StopNow As Boolean
Dim ErrorMax As Long
Dim Error0_100 As Long
Dim Error100_200 As Long
Dim Error200_300 As Long
Dim Error300_400 As Long
Dim Error400_500 As Long
Dim Error500_600 As Long
Dim Error600_700 As Long
Dim Error700 As Long
    
```

```
Private Sub Command2_Click()
```

```
    StopNow = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    NMEA = 2
```

```
    Port = 1
```

```
    StatusBar1.SimpleText = "Ready..."
```

```
    ComPrt = "COM1F" 'The default Com Port
```

```
    lblClock.Caption = "0.5 sec" 'The Default Interval
```

```
    Label3.Caption = CStr(Slider2.Value)
```

```
    RepTimes = CInt(Slider2.Value)
```

```
End Sub
```

```
Private Sub mnuDataLink_Click()
```

```
    OpenWhat = "Full"
```

```
    frmBrowse.Visible = True
```

```
    frmModeFull.Enabled = False
```

```
End Sub
```

```
Private Sub mnuFBUScom1_Click()
```

```
    If mnuFBUScom1.Checked = False Then
```

```
        mnuFBUScom1.Checked = True
```

```
        mnuFBUScom2.Checked = False
```

```
        ComPrt = "COM1F" 'COM1 Is Selected by the User
```

```
    End If
```

```
End Sub
```

```
Private Sub mnuFBUScom2_Click()
```

```
    If mnuFBUScom2.Checked = False Then
```

```
        mnuFBUScom2.Checked = True
```

```
        mnuFBUScom1.Checked = False
```

```
        ComPrt = "COM2F" 'COM2 Is Selected by the User
```

```
    End If
```

```
End Sub
```

```
Private Sub mnuGPSNMEA21_Click()
```

```
    If mnuGPSNMEA21.Checked = False Then
```

```
        mnuGPSNMEA21.Checked = True
```

```
        NMEA = 2
```

```
        mnuGPSNMEA15.Checked = False
```

```
    End If
```

```
End Sub
```

```
Private Sub mnuGPSNMEA15_Click()  
  
    If mnuGPSNMEA15.Checked = False Then  
        mnuGPSNMEA15.Checked = True  
        NMEA = 1  
        mnuGPSNMEA21.Checked = False  
    End If  
  
End Sub
```

```
Private Sub mnuGPSport1_Click()  
  
    If mnuGPSport1.Checked = False Then  
        mnuGPSport1.Checked = True  
        Port = 1  
        mnuGPSport2.Checked = False  
    End If  
  
End Sub
```

```
Private Sub mnuGPSport2_Click()  
  
    If mnuGPSport2.Checked = False Then  
        mnuGPSport2.Checked = True  
        Port = 2  
        mnuGPSport1.Checked = False  
    End If  
  
End Sub
```

```
Private Function CheckPorts() As Boolean
```

```
Dim Response As Integer
```

```
    If (mnuFBUScom1.Checked = True And mnuGPSport1.Checked = True) Or _  
        (mnuFBUScom2.Checked = True And mnuGPSport2.Checked = True) Then  
  
        Response = MsgBox("You have selected the same Port (COM" & Port & _  
            ") for both the Mobile Device and the GPS. Press OK to continue, " & _  
            "or Cancel to change your settings and Retry.", _  
            vbOKCancel + vbExclamation + vbDefaultButton2, "ATTENTION")  
  
        If Response = vbOK Then  
            CheckPorts = True  
        Else  
            CheckPorts = False  
        End If  
    Else: CheckPorts = True  
    End If
```

```
End Function
```

```
Private Sub mnuGPSEnabled_Click()  
  
    If mnuGPSEnabled.Caption = "Enable" Then  
        If CheckPorts = True Then
```

```
        mnuGPSEnabled.Caption = "Disable"  
        Call frmGPS.Initialize  
        frmGPS.Enabled = True  
        frmGPS.Visible = True  
        mnuGPSport.Enabled = False  
        mnuGPSNMEA.Enabled = False  
    End If  
Else  
    mnuGPSport.Enabled = True  
    mnuGPSNMEA.Enabled = True  
    mnuGPSEnabled.Caption = "Enable"  
    Unload frmGPS  
End If
```

End Sub

```
Private Sub mnuHelpAbout_Click(Index As Integer)  
  
    frmAbout.Visible = True  
    Load frmAbout
```

End Sub

```
Private Sub mnuPrgExit_Click()  
  
'Make sure user really wants to exit  
Dim Response As Integer  
  
    Response = MsgBox("Are you sure you want to exit the Program?", vbYesNo + _  
        vbCritical + vbDefaultButton2, "Exit Program")  
  
    If Response = vbNo Then  
        Exit Sub  
    Else  
        End  
    End If
```

End Sub

```
Private Sub mnuPrgMode_Click()  
  
    Load frmStart  
    frmStart.Visible = True  
    frmStart.Enabled = True  
    Unload Me
```

End Sub

```
Private Sub Slider1_Click()  
  
'The values which are stored for the Measurement Interval  
  
    If Slider1.Value = 0 Then  
        Timer1.Interval = 1  
        lblClock.Caption = "0.5 sec"
```

```
ElseIf Slider1.Value = 1 Then
    Timer1.Interval = 500
    lblClock.Caption = "1 sec"

ElseIf Slider1.Value = 2 Then
    Timer1.Interval = 1500
    lblClock.Caption = "2 sec"

ElseIf Slider1.Value = 3 Then
    Timer1.Interval = 2500
    lblClock.Caption = "3 sec"

ElseIf Slider1.Value = 4 Then
    Timer1.Interval = 3500
    lblClock.Caption = "4 sec"
End If
```

End Sub

```
Private Sub Slider2_Click()

    'Slider2 sets the Number of Measurements, variable: RepTimes
    RepTimes = Slider2.Value
    Label3.Caption = CStr(Slider2.Value)
```

End Sub

```
Private Sub Command1_Click()
Dim i As Integer
Dim Response As Integer
Dim Answer As Variant

    PictureGraph.Cls
    Label4(0).Caption = ""
    lblAcc.Caption = ""
    lblNR.Caption = ""
    lblServCell.Caption = ""
    lblTA.Caption = ""
    lblDist.Caption = ""
    For i = 0 To 7
        lblValue(i).Caption = ""
    Next i
    'Initialize DataBaseError Variable
    DataBaseError = False

    If mnuGPSEnabled.Caption = "Disable" Then
        GPSlat = CDBl(frmGPS.lblLat)
        GPSlon = CDBl(frmGPS.lblLon)
        If GPSlat = 0 Then
            Response = MsgBox("The GPS doesn't generate a Position, " & _
                "do you want to enter position manually?", _
                vbYesNo + vbQuestion, "Caution")

            If Response = vbYes Then
                Answer = _
                    InputBox("Type the Latitude (format: xx.xxxxxx)" & vbCrLf & _
                        "eg. 38.55555", "Manual Latitude")
                If (Answer <> "") Then
                    GPSlat = CDBl(Answer)
```

```

        Answer = _
        InputBox("Type the Longitude (Format: xxx.xxxxx, " & _
            "eg. 023.55555", "Manual Longitude")

        If (Answer <> "") Then
            GPSLon = CDBl(Answer)
            GPS = True
        Else
            GPS = False
        End If
    Else
        GPS = False
    End If
Else
    GPS = False
End If
Else
    GPS = True
End If
If (mnuFBUScom1.Checked = True And mnuGPSport1.Checked = True) Or _
    (mnuFBUScom2.Checked = True And mnuGPSport2.Checked = True) Then

    Unload frmGPS
    MsgBox _
        "Please Disconnect now the GPS and Connect the Mobile Device." & _
        " Press OK when Done.", vbOKOnly, "Device Change"

End If
Else
    GPS = False
End If

cntr = 1 'Set the counter for the Matrix Metriseis()
        '(used in Timer1.Enabled)

'Connect to Mobile Terminal
StatusBar1.SimpleText = "Establishing Connection..."
FBUS.SetLicense "Mobile FBUS 1.7", "xxxxxxxxxxxxxxxxxxxx"
FBUS.RaiseError = False
FBUS.Connect ComPrt

If FBUS.IsAlive = True Then 'Check if Connection is true

    StatusBar1.SimpleText = "Connected"
    FBUS.RaiseError = True
    Command1.Enabled = False
    Command2.Enabled = True

'Declaring the size of Matrixes
ReDim Metriseis(1 To (RepTimes * 2), 1 To 7) As Long
ReDim cid(1 To RepTimes) As Long
ReDim TA(1 To RepTimes) As Long

'Start Counters
ErrorMax = 1
Error0_100 = 0
Error100_200 = 0
Error200_300 = 0
Error300_400 = 0
Error400_500 = 0
Error500_600 = 0
Error600_700 = 0
Error700 = 0

```



```
StopNow = False
LocError = 0
metraw = 0
stoptry = 0
ExcelCntr = 1
Set xlApp = New Excel.Application
Set xlBook = xlApp.Workbooks.Add
Set xlSheet = xlApp.ActiveSheet
xlSheet.Cells(1, 1).Value = "Serving_Cell"
xlSheet.Cells(1, 2).Value = "RxLev"
xlSheet.Cells(1, 3).Value = "Neighbour_1"
xlSheet.Cells(1, 4).Value = "RxLev"
xlSheet.Cells(1, 5).Value = "Neighbour_2"
xlSheet.Cells(1, 6).Value = "RxLev"
xlSheet.Cells(1, 7).Value = "Neighbour_3"
xlSheet.Cells(1, 8).Value = "RxLev"
xlSheet.Cells(1, 9).Value = "Neighbour_4"
xlSheet.Cells(1, 10).Value = "RxLev"
xlSheet.Cells(1, 11).Value = "Neighbour_5"
xlSheet.Cells(1, 12).Value = "RxLev"
xlSheet.Cells(1, 13).Value = "Neighbour_6"
xlSheet.Cells(1, 14).Value = "RxLev"
xlSheet.Cells(1, 15).Value = "Cell_ID"
xlSheet.Cells(1, 16).Value = "TA"
xlSheet.Cells(1, 17).Value = "Calc_Lon"
xlSheet.Cells(1, 18).Value = "Calc_Lat"
xlSheet.Cells(1, 19).Value = "GPS_Lon"
xlSheet.Cells(1, 20).Value = "GPS_Lat"
xlSheet.Cells(1, 21).Value = "ERROR"
xlSheet.Cells(1, 22).Value = "GPS_Distance_From_ServCell"
xlSheet.Cells(1, 23).Value = "TRUST_TYPE"
xlSheet.Cells(1, 24).Value = "TRUST_LON"
xlSheet.Cells(1, 25).Value = "TRUST_LAT"
xlSheet.Cells(1, 26).Value = "TRUST_CIRCLE_BIG"
xlSheet.Cells(1, 27).Value = "TRUST_CIRCLE_SMALL"
xlSheet.Cells(1, 28).Value = "TRUST_ANGLE_START"
xlSheet.Cells(1, 29).Value = "TRUST_ANGLE_STOP"

'Start Recording Measurements (by Procedure Timer1)
Timer1.Enabled = True
StatusBar1.SimpleText = "Getting Data from Mobile Phone..."

Else
    Beep
    MsgBox "Check your Cable Connection and COM Port Settings!", _
        vbOKOnly, "Failed to Connect"

    StatusBar1.SimpleText = "Failed to Connect..."
End If

End Sub
```

```
Private Sub Timer1_Timer()

Dim j As Integer
Dim aris As Variant
Dim BTS(1 To 3) As Variant
Dim i As Integer
Dim TimingQual As Variant
Dim CellID As Variant
```

```
metraw = metraw + 1 'Shows how many times the Procedure has been Executed

'Get Data From Netmonitor

For i = 1 To 3
    BTS(i) = FBUS.NetmonitorScreen("0" & CStr(i + 2))
Next i

TimingQual = FBUS.NetmonitorScreen("01")
CellID = FBUS.NetmonitorScreen("11")

'Store Data in Metriseis() Matrix

For i = 1 To 7
    Metriseis(cntr, i) = CLng(Val(Mid(BTS(1 + Int((i - 1) / 3)), _
        (1 + 13 * (i - (1 + Int((i - 1) / 3) * 3))), 3)))

    Metriseis(cntr + 1, i) = CLng(Val(Mid(BTS(1 + Int((i - 1) / 3)), _
        (7 + 13 * (i - (1 + Int((i - 1) / 3) * 3))), 3)))

    If Metriseis(cntr + 1, i) > 0 Then
        Metriseis(cntr + 1, i) = Metriseis(cntr + 1, i) * (-1)
    End If
Next i

cid(1 + (cntr - 1) / 2) = CLng(Val(Mid(CellID, 40, 6)))
TA(1 + (cntr - 1) / 2) = CLng(Val(Mid(TimingQual, 17, 2)))

'The case where there is no value for 2nd Neighbour is not to count
If Metriseis(cntr, 3) > 0 Then
    cntr = cntr + 2
Else
    stoptry = stoptry + 1
End If

StatusBar1.SimpleText = "Getting Data from Mobile Phone..." & CStr(metraw)

'Case where there isn't any useful value for five consecutive tries
If stoptry = 5 Then
    MsgBox "Can't get the signal data, too weak", vbOKOnly, _
        "Failed to Collect Data"

    metraw = 0
    stoptry = 0
    cntr = 1

End If

'Check if Timer1 will stop or not. When the timer Stops, the Unsorted Matrix is
'being Print on Screen and the SortMatrix Procedure is being called
If RepTimes = ((cntr + 1) / 2 - 1) And (stoptry < 5) Then
    Timer1.Enabled = False
    Call Dothings

End If

End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    StatusBar1.SimpleText = "Exiting..."
    Call frmSplash2.Clos
    Call frmFinal.Clos
    Unload frmGPS
```

```
End Sub
```

```
Private Sub DrawBar()
```

```
    PictureGraph.Cls

    If Error < 100 Then
        Error0_100 = Error0_100 + 1
        If Error0_100 > ErrorMax Then ErrorMax = Error0_100
    ElseIf Error < 200 Then
        Error100_200 = Error100_200 + 1
        If Error100_200 > ErrorMax Then ErrorMax = Error100_200
    ElseIf Error < 300 Then
        Error200_300 = Error200_300 + 1
        If Error200_300 > ErrorMax Then ErrorMax = Error200_300
    ElseIf Error < 400 Then
        Error300_400 = Error300_400 + 1
        If Error300_400 > ErrorMax Then ErrorMax = Error300_400
    ElseIf Error < 500 Then
        Error400_500 = Error400_500 + 1
        If Error400_500 > ErrorMax Then ErrorMax = Error400_500
    ElseIf Error < 600 Then
        Error500_600 = Error500_600 + 1
        If Error500_600 > ErrorMax Then ErrorMax = Error500_600
    ElseIf Error < 700 Then
        Error600_700 = Error600_700 + 1
        If Error600_700 > ErrorMax Then ErrorMax = Error600_700
    Else
        Error700 = Error700 + 1
        If Error700 > ErrorMax Then ErrorMax = Error700
    End If

    PictureGraph.ScaleHeight = ErrorMax
    Label4(0).Caption = ErrorMax

    PictureGraph.PSet (1, ErrorMax)
    PictureGraph.Line -(3, ErrorMax - Error0_100), vbYellow, BF
    PictureGraph.PSet (5, ErrorMax)
    PictureGraph.Line -(7, ErrorMax - Error100_200), vbYellow, BF
    PictureGraph.PSet (9, ErrorMax)
    PictureGraph.Line -(11, ErrorMax - Error200_300), vbYellow, BF
    PictureGraph.PSet (13, ErrorMax)
    PictureGraph.Line -(15, ErrorMax - Error300_400), vbYellow, BF
    PictureGraph.PSet (17, ErrorMax)
    PictureGraph.Line -(19, ErrorMax - Error400_500), vbYellow, BF
    PictureGraph.PSet (21, ErrorMax)
    PictureGraph.Line -(23, ErrorMax - Error500_600), vbYellow, BF
    PictureGraph.PSet (25, ErrorMax)
    PictureGraph.Line -(27, ErrorMax - Error600_700), vbYellow, BF
    PictureGraph.PSet (29, ErrorMax)
    PictureGraph.Line -(31, ErrorMax - Error700), vbYellow, BF

    If Error0_100 <> 0 Then
        lblValue(0).Caption = Error0_100
    Else: lblValue(0) = ""
```

```
End If
If Error100_200 <> 0 Then
    lblValue(1).Caption = Error100_200
Else: lblValue(1) = ""
End If
If Error200_300 <> 0 Then
    lblValue(2).Caption = Error200_300
Else: lblValue(2) = ""
End If
If Error300_400 <> 0 Then
    lblValue(3).Caption = Error300_400
Else: lblValue(3) = ""
End If
If Error400_500 <> 0 Then
    lblValue(4).Caption = Error400_500
Else: lblValue(4) = ""
End If
If Error500_600 <> 0 Then
    lblValue(5).Caption = Error500_600
Else: lblValue(5) = ""
End If
If Error600_700 <> 0 Then
    lblValue(6).Caption = Error600_700
Else: lblValue(6) = ""
End If
If Error700 <> 0 Then
    lblValue(7).Caption = Error700
Else: lblValue(7) = ""
End If
```

End Sub

```
Private Sub StoreResults()
Dim i As Integer
Dim j As Long

For i = 1 To FullDataMax

    xlSheet.Cells(ExcelCntr + 1, 2 * i - 1).Value = FinalFull(i, 1)
    xlSheet.Cells(ExcelCntr + 1, 2 * i).Value = FinalFull(i, 2)
Next i

xlSheet.Cells(ExcelCntr + 1, 15).Value = FinalCID
xlSheet.Cells(ExcelCntr + 1, 16).Value = FinalTA
xlSheet.Cells(ExcelCntr + 1, 17).Value = CalcLon
xlSheet.Cells(ExcelCntr + 1, 18).Value = CalcLat
xlSheet.Cells(ExcelCntr + 1, 19).Value = GPSlon
xlSheet.Cells(ExcelCntr + 1, 20).Value = GPSlat
xlSheet.Cells(ExcelCntr + 1, 21).Value = Error
xlSheet.Cells(ExcelCntr + 1, 22).Value = GPSDistance
xlSheet.Cells(ExcelCntr + 1, 23).Value = TRUST_TYPE
xlSheet.Cells(ExcelCntr + 1, 24).Value = TRUST_LON
xlSheet.Cells(ExcelCntr + 1, 25).Value = TRUST_LAT
xlSheet.Cells(ExcelCntr + 1, 26).Value = TRUST_CIRCLE_BIG
xlSheet.Cells(ExcelCntr + 1, 27).Value = TRUST_CIRCLE_SMALL
xlSheet.Cells(ExcelCntr + 1, 28).Value = TRUST_ANGLE_START
xlSheet.Cells(ExcelCntr + 1, 29).Value = TRUST_ANGLE_STOP

lblNR.Caption = CStr(ExcelCntr)
LocError = LocError + Error
lblAcc.Caption = Format((LocError / ExcelCntr), "###0.0")
```

```
lblTA = FinalTA
lblDist = GPSDistance
ExcelCntr = ExcelCntr + 1
```

End Sub

```
Private Sub Dothings()
```

```
Dim excelsave As String
Dim SaveTime As Variant
```

```
StatusBar1.SimpleText = "Sorting Matrix..."
```

```
Starttime = Now
```

```
Call ModSub.SortMatrix 'Sorting of the unsorted Data
```

```
Call ModSub.ProcessData 'Calculate Mean Values
```

```
lblServCell.Caption = CStr(FinalCID)
```

```
Call ModSub.StoreValues 'Store The results as variables for the Database Search
```

```
Call ModSub.Databasefind 'Search for given Cells in Database to get
                          'the required Information
```

```
Call ModSub.ProcessDist
```

```
If DataBaseError = False Then
```

```
    If ModSub.MicroCheck Then
```

```
        Zoom = 4
```

```
        Call ModSub.NotLocate
```

```
    Else
```

```
        Zoom = 4 'Variable Declared in Module UTMWGS84
```

```
        Call ModSub.TAHATAMatch
```

```
        Call ModSub.CalcDistances
```

```
    End If
```

```
    Call DrawBar
```

```
    Call StoreResults
```

```
End If
```

```
'Declaring the size of Matrixes
```

```
ReDim Metriseis(1 To (RepTimes * 2), 1 To 7) As Long
```

```
ReDim cid(1 To RepTimes) As Long
```

```
ReDim TA(1 To RepTimes) As Long
```

```
'Start Counters
```

```
DataBaseError = False
```

```
metraw = 0
```

```
stoptry = 0
```

```
cntr = 1
```

```
If StopNow = False Then
```

```
    Timer1.Enabled = True
```

```
Else
```

```
    Command1.Enabled = True
```

```
    Command2.Enabled = False
```

```
    FBUS.Disconnect
```

```
    SaveTime = Format(Now, "dd_mm_yyyy__hh_mm_ss")
```

```
    excelsave = App.Path & "\" & SaveTime & ".xls"
```

```
    xlSheet.SaveAs (excelsave)
```

```
    xlApp.Quit
```

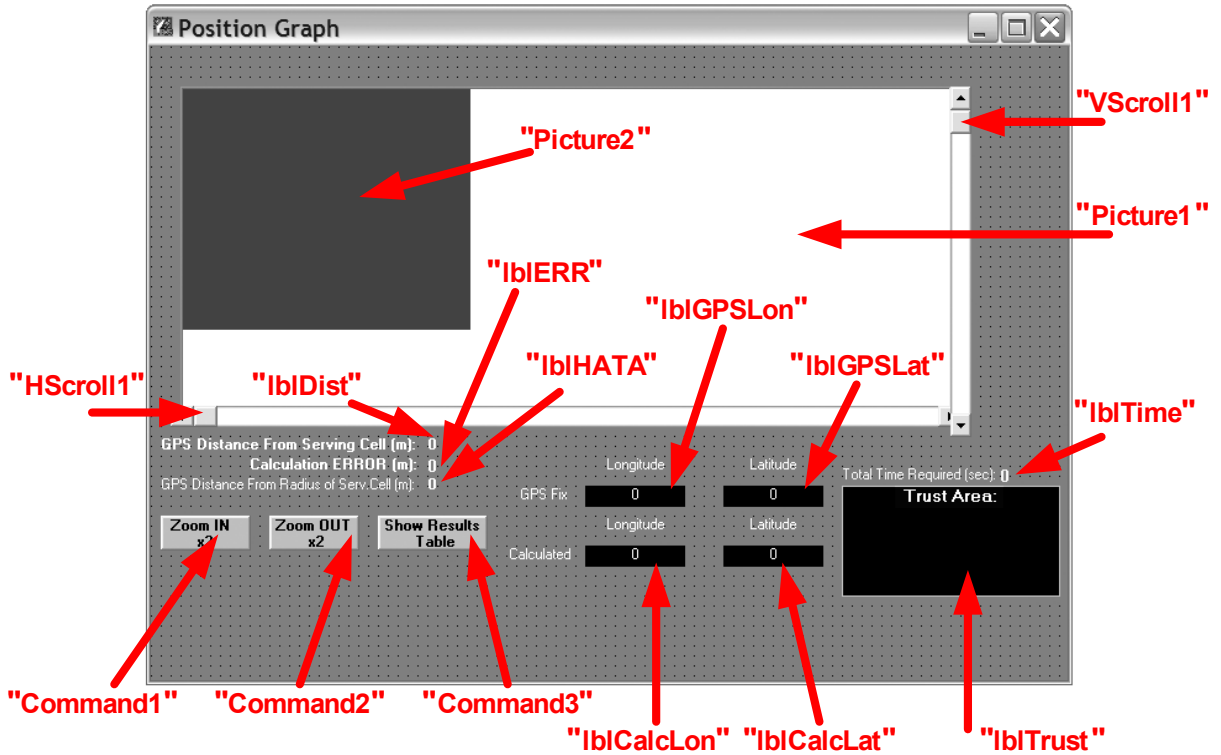
```
    Set xlApp = Nothing
```

```
Set xlBook = Nothing  
Set xlSheet = Nothing  
MsgBox "Results have been saved in " & excelsave & " (Excel Worksheet)", _  
vbOKOnly, "Finished"
```

```
End If
```

```
End Sub
```

A.2.5. frmFinal



Option Explicit

```
Sub Clos()
    Unload Me
End Sub
```

```
Private Sub Command1_Click()
    Zoom = Zoom / 2
    If ModSub.MicroCheck Then
        Call ModSub.DrawMicro
    Else
        Call ModSub.DrawGraph
    End If
    Call FixPositions
    HScroll1.Value = 0
    VScroll1.Value = 0
    If Zoom = 2 Then Command1.Enabled = False
    If Command2.Enabled = False And Zoom = 8 Then Command2.Enabled = True
End Sub
```

```
Private Sub Command2_Click()  
  
    Zoom = Zoom * 2  
    If ModSub.MicroCheck Then  
        Call ModSub.DrawMicro  
    Else  
        Call ModSub.DrawGraph  
    End If  
    Call FixPositions  
    HScroll1.Value = 0  
    VScroll1.Value = 0  
    If Command1.Enabled = False And Zoom = 4 Then Command1.Enabled = True  
    If Zoom = 16 Then Command2.Enabled = False  
  
End Sub
```

```
Private Sub Command3_Click()  
  
    Unload frmSplash2  
    frmSplash2.Visible = True  
    Call ModSub.FillMatrData
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
  
    Unload frmSplash2
```

```
End Sub
```

```
Private Sub HScroll1_Change()  
' Picture2.Left is set to the negative of the value because  
' as you scroll the scroll bar to the right, the display  
' should move to the Left, showing more of the right  
' of the display, and vice-versa when scrolling to the  
' left.  
  
Dim temp As Double  
  
    temp = (Picture2.Width - Picture1.Width) / HScroll1.Max  
    Picture2.Left = -Round(temp * HScroll1.Value)
```

```
End Sub
```

```
Private Sub Picture1_DblClick()  
  
    frmSavePicture.Enabled = True  
    frmSavePicture.Visible = True
```

```
End Sub
```

```
Private Sub Picture2_Click()  
  
    frmSavePicture.Enabled = True
```



```
frmSavePicture.Visible = True
```

```
End Sub
```

```
Private Sub VScroll1_Change()  
' Picture2.Top is set to the negative of the value because  
' as you scroll the scroll bar down, the display  
' should move up, showing more of the bottom  
' of the display, and vice-versa when scrolling up.
```

```
Dim temp As Double
```

```
temp = (Picture2.Height - Picture1.Height) / VScroll1.Max  
Picture2.Top = -Round(temp * VScroll1.Value)
```

```
End Sub
```

```
Private Sub Form_Resize()
```

```
If frmFinal.WindowState <> 1 Then  
' The Picture1 picture box dimensions are changed when the form  
' size is changed.  
With Picture1  
    .Height = frmFinal.Height  
    .Width = frmFinal.Width  
End With  
  
' Re-Initializes picture positions & scroll bars.  
Picture1.Move 0, 0, ScaleWidth - VScroll1.Width, _  
    ScaleHeight - HScroll1.Height - 2000  
HScroll1.Value = 0  
VScroll1.Value = 0  
Call FixPositions  
Unload frmSplash2  
Else  
    Unload frmSplash2  
End If
```

```
End Sub
```

```
Sub Initialize()
```

```
Call FixPositions  
Picture1.Move 0, 0, ScaleWidth - VScroll1.Width, _  
    ScaleHeight - HScroll1.Height - 2000  
HScroll1.Max = 20  
VScroll1.Max = 20  
HScroll1.LargeChange = 5  
VScroll1.LargeChange = 5
```

```
End Sub
```

```
Private Sub FixPositions()
```

```
Command1.Top = frmFinal.Height - 1200
```

```
Command2.Top = frmFinal.Height - 1200
Command3.Top = frmFinal.Height - 1200
lblCalc.Top = frmFinal.Height - 1200
lblCalcLon.Top = frmFinal.Height - 1200
lblCalcLat.Top = frmFinal.Height - 1200
lbl1.Top = frmFinal.Height - 1500
lbl2.Top = frmFinal.Height - 1500
lblGPS.Top = frmFinal.Height - 2000
lblGPSLon.Top = frmFinal.Height - 2000
lblGPSLat.Top = frmFinal.Height - 2000
lbl3.Top = frmFinal.Height - 2300
lbl4.Top = frmFinal.Height - 2300
lbl5.Top = frmFinal.Height - 2300
lblERR.Top = frmFinal.Height - 2300
lbl6.Top = frmFinal.Height - 2000
lblDist.Top = frmFinal.Height - 2000
lbl7.Top = frmFinal.Height - 1700
lblHATA.Top = frmFinal.Height - 1700
lbl8.Top = frmFinal.Height - 2300
lblTime.Top = frmFinal.Height - 2300
lbl9.Top = frmFinal.Height - 1950
Shape1.Top = frmFinal.Height - 2000
lblTrust.Top = frmFinal.Height - 1700

With Picture2
' Initialize position of both pictures.
.Move 0, 0
End With

With HScroll1
.Top = Picture1.Height
.Left = 0
.Width = Picture1.Width
End With

With VScroll1
.Top = 0
.Left = Picture1.Width
.Height = Picture1.Height
End With

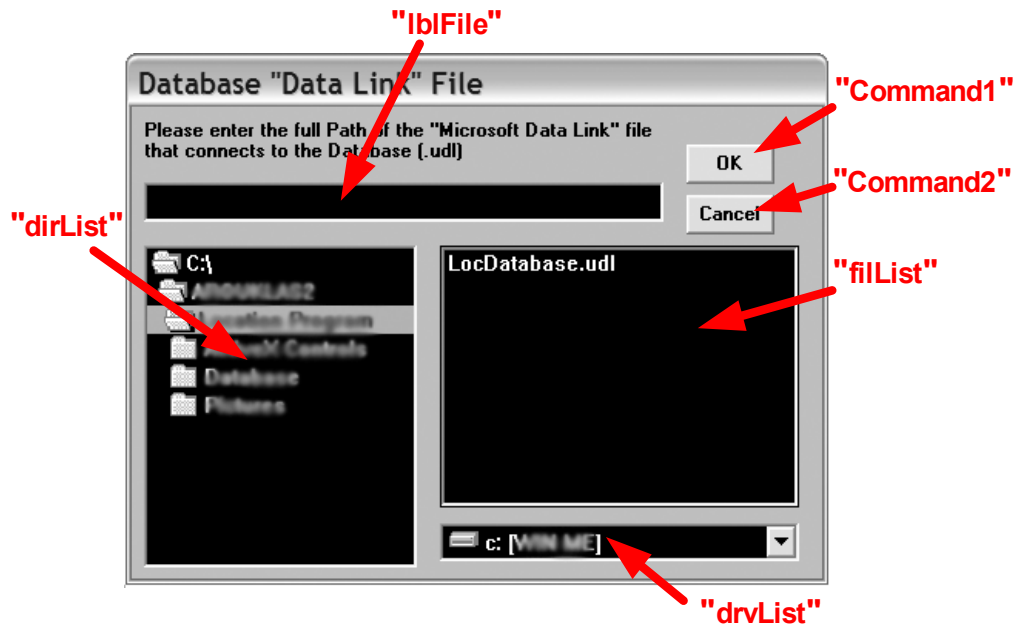
' Determine if child picture will fill up screen.
' If so, then there is no need to use scroll bars.

VScroll1.Visible = (Picture1.Height < Picture2.Height)
If VScroll1.Visible = False Then Picture2.Move Picture2.Left, _
Round((Picture1.Height - HScroll1.Height - Picture2.Height) / 2)

HScroll1.Visible = (Picture1.Width < Picture2.Width)
If HScroll1.Visible = False Then Picture2.Move Round((Picture1.Width - _
VScroll1.Width - Picture2.Width) / 2), Picture2.Top
```

End Sub

A.2.6. frmBrowse



[Option Explicit](#)

```
Private Sub Command1_Click()

    If lblFile.Caption = "" Then
        MsgBox "You have to Select a valid 'Data Link' File", vbOKOnly, "Error"
    Else
        DataLinkFile = lblFile.Caption
        If OpenWhat = "User" Then frmModeUser.Enabled = True
        If OpenWhat = "Full" Then frmModeFull.Enabled = True
        If OpenWhat = "Normal" Then frmModeNormal.Enabled = True
        Unload Me
    End If
End Sub
```

[End Sub](#)

```
Private Sub Command2_Click()

    If OpenWhat = "User" Then frmModeUser.Enabled = True
    If OpenWhat = "Full" Then frmModeFull.Enabled = True
    If OpenWhat = "Normal" Then frmModeNormal.Enabled = True
    Unload Me
End Sub
```

[End Sub](#)

```
Private Sub drvList_Change()

    On Error GoTo DriveHandler
End Sub
```

```
' If new drive was selected, the Dir1 box
' updates its display.
dirList.Path = drvList.Drive
Exit Sub
' If there is an error, reset drvList.Drive with the
' drive from dirList.Path.

DriveHandler:
    drvList.Drive = dirList.Path
    Exit Sub
```

End Sub

```
Private Sub dirList_Change()

    ' Update file list box to synchronize with the
    ' directory list box.
    fileList.Path = dirList.Path
```

End Sub

```
Private Sub fileList_DblClick()

    lblFile.Caption = fileList.Path & "\" & fileList.FileName
```

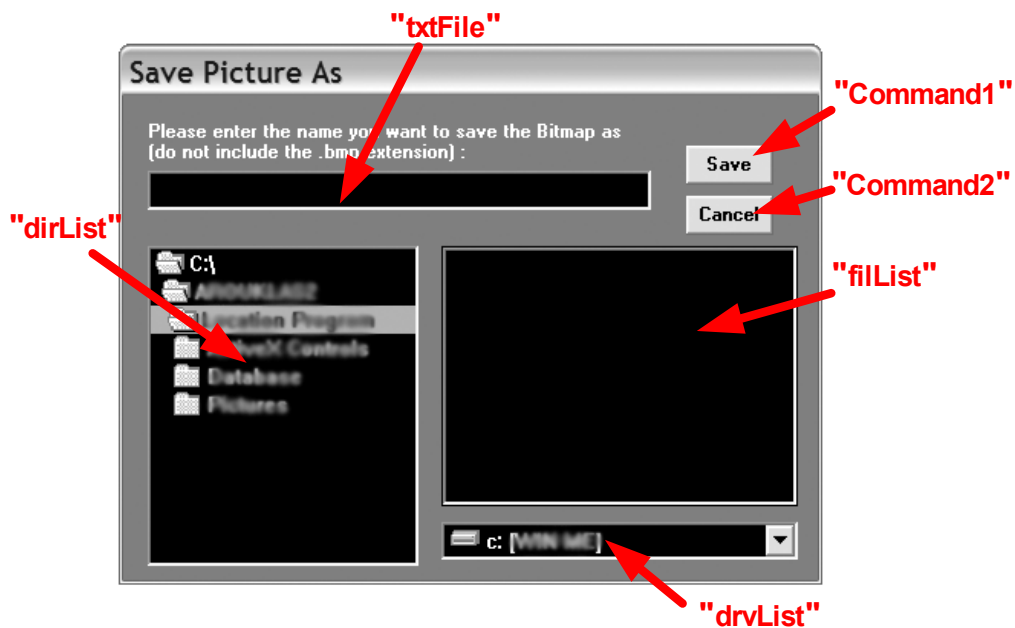
End Sub

```
Private Sub Form_Load()

    drvList.Drive = App.Path
    dirList.Path = App.Path
```

End Sub

A.2.7. frmSavePicture



Option Explicit

```
Private Sub Command1_Click()

    If txtFile.Text = filList.Path & "\" Then
        MsgBox "You have to Select a valid File Name", vbOKOnly, "Error"
    Else
        SavePicture frmFinal.Picture2.Image, txtFile.Text & ".bmp"
        Unload Me
    End If
End Sub
```

End Sub

```
Private Sub Command2_Click()
```

```
    Unload Me
```

End Sub

```
Private Sub drvList_Change()
```

```
    On Error GoTo DriveHandler
    ' If new drive was selected, the Dir1 box
    ' updates its display.
    dirList.Path = drvList.Drive
    Exit Sub
    ' If there is an error, reset drvList.Drive with the
    ' drive from dirList.Path.
```

```
DriveHandler:  
    drvList.Drive = dirList.Path  
    Exit Sub
```

```
End Sub
```

```
Private Sub dirList_Change()  
  
    ' Update file list box to synchronize with the  
    ' directory list box.  
    fillList.Path = dirList.Path  
    txtFile.Text = fillList.Path & "\"
```

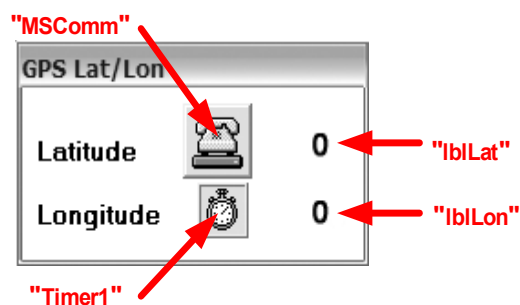
```
End Sub
```

```
Private Sub fillList_DblClick()  
  
    txtFile.Text = fillList.Path & "\" & fillList.FileName
```

```
End Sub
```

```
Private Sub Form_Load()  
  
    drvList.Drive = App.Path  
    dirList.Path = App.Path  
    fillList.Path = App.Path  
    txtFile.Text = fillList.Path & "\"
```

```
End Sub
```

A.2.8. frmGPS

```
Option Explicit
```

```
Dim Sentence As Variant
Dim lat As Double
Dim lon As Double
Dim LatDeg, LonDeg As String
Dim LatMin, LonMin As String
Dim LatDec, LonDec As Double
Dim i As Long
Dim j As Integer
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    MSComm.PortOpen = False
```

```
End Sub
```

```
Private Sub MSComm_OnComm()
```

```
    i = i + 1
    If MSComm.InputLen = 1 Then
        If MSComm.Input = "$" Then
            MSComm.RThreshold = 5
            MSComm.InputLen = 5
        End If
    ElseIf MSComm.InputLen = 5 Then
        If MSComm.Input = "GPGLL" Then
            If NMEA = 1 Then
                MSComm.RThreshold = 21
                MSComm.InputLen = 21
            Else
                MSComm.RThreshold = 25
                MSComm.InputLen = 25
            End If
        Else
            MSComm.RThreshold = 1
            MSComm.InputLen = 1
        End If
    ElseIf (MSComm.InputLen = 21) Or (MSComm.InputLen = 25) Then
        Sentence = MSComm.Input
        Call ProcessSentence
        MSComm.RThreshold = 1
```

```
MSComm.InputLen = 1  
End If
```

```
End Sub
```

```
Private Sub ProcessSentence()
```

```
    If NMEA = 1 Then  
        LatDeg = Mid(CStr(Sentence), 2, 2)  
        LatMin = Mid(CStr(Sentence), 4, 5)  
        LatDec = Round(CDbl(LatMin) / 60, 5)  
        LatDec = LatDec + CDbl(LatDeg) 'Latitude in Decimal Format  
        LonDeg = Mid(CStr(Sentence), 12, 3)  
        LonMin = Mid(CStr(Sentence), 15, 5)  
        LonDec = Round(CDbl(LonMin) / 60, 5)  
        LonDec = LonDec + CDbl(LonDeg) 'Latitude in Decimal Format  
        lblLat = LatDec  
        lblLon = LonDec  
    Else  
        LatDeg = Mid(CStr(Sentence), 2, 2)  
        LatMin = Mid(CStr(Sentence), 4, 7)  
        LatDec = Round(CDbl(LatMin) / 60, 5)  
        LatDec = LatDec + CDbl(LatDeg) 'Latitude in Decimal Format  
        LonDeg = Mid(CStr(Sentence), 14, 3)  
        LonMin = Mid(CStr(Sentence), 17, 7)  
        LonDec = Round(CDbl(LonMin) / 60, 5)  
        LonDec = LonDec + CDbl(LonDeg) 'Latitude in Decimal Format  
        lblLat = LatDec  
        lblLon = LonDec  
    End If
```

```
End Sub
```

```
Sub Initialize()
```

```
    i = 0  
    j = 0  
    MSComm.CommPort = Port  
    MSComm.InputLen = 1  
    MSComm.InputMode = comInputModeText  
    MSComm.RThreshold = 1  
    MSComm.Settings = "4800,n,8,1"  
    MSComm.PortOpen = True  
    Timer1.Enabled = True
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
Dim temp As Long
```

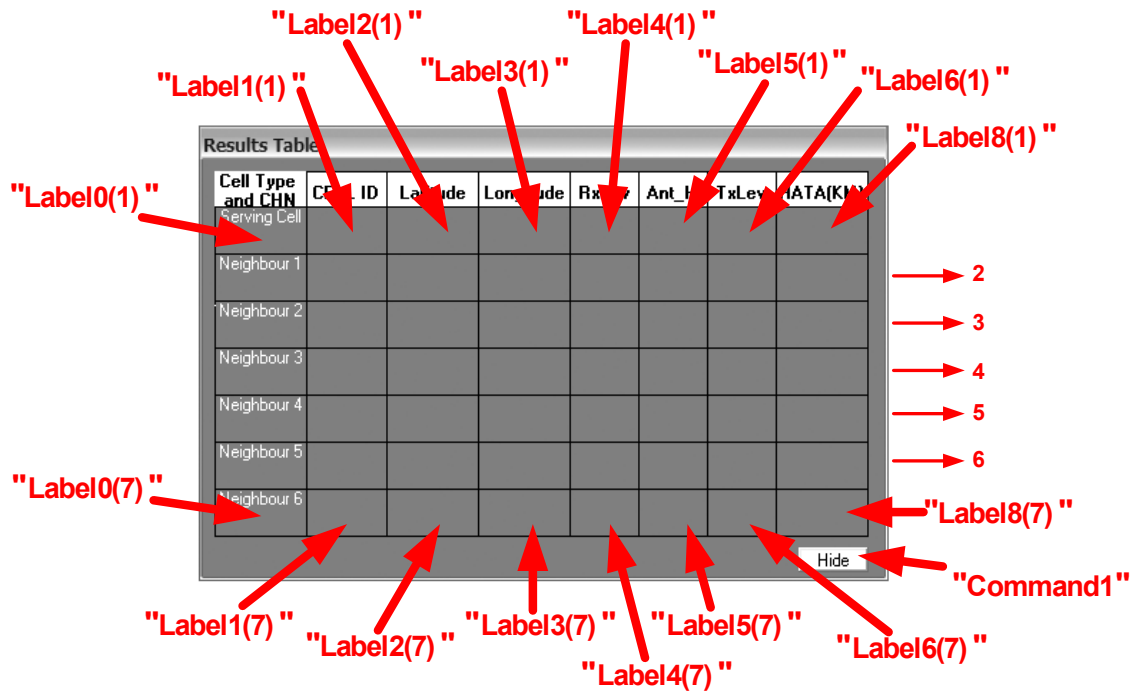
```
    If j = 0 Then temp = i  
    j = j + 1  
    If j = 4 Then
```



```
    If i = temp Then  
        lblLat.Caption = "0"  
        lblLon.Caption = "0"  
    End If  
    j = 0  
End If
```

[End Sub](#)

A.2.9. frmSplash2



Option Explicit

```
Private Sub Command1_Click()
    Unload Me
End Sub
```

```
Sub Clos()
    Unload Me
End Sub
```

A.3. MODULES

A.3.1. ModPropagation

Option Explicit

```

Function HataDist(ByVal Ht As Double, _
                 ByVal Hr As Double, _
                 ByVal fMHz As Double, _
                 ByVal Tx As Double, _
                 ByVal Rx As Double, _
                 ByVal PropType As Integer) As Double

'Ht: The Base Station Height
'Hr: The Mobile antenna Height
'fMHz: The Center Frequency in Megahertz
'Tx: The Output Power in dBm
'Rx: The Received Power in dBm
'HataDist: The estimated distance in Kilometers
'PropType: Type of Area-----> 1 for Open Area
                                   2 for Suburban Area
                                   3 for Medium-Small City
                                   4 for Large City

Dim temp As Double
Dim factorA As Double
Dim factorK As Double

    Select Case PropType

        Case 1 'Open Area
            factorA = (1.1 * Log10(fMHz) - 0.7) * Hr - (1.56 * Log10(fMHz) - 0.8)
            factorK = 4.78 * (Log10(fMHz)) ^ 2 - 18.33 * Log10(fMHz) + 40.94
        Case 2 'Suburban Area
            factorA = (1.1 * Log10(fMHz) - 0.7) * Hr - (1.56 * Log10(fMHz) - 0.8)
            factorK = 2 * ((Log10(fMHz / 28)) ^ 2) + 5.4
        Case 3 'Medium-Small City
            factorA = (1.1 * Log10(fMHz) - 0.7) * Hr - (1.56 * Log10(fMHz) - 0.8)
            factorK = 0
        Case 4 'Large City
            factorA = 3.2 * ((Log10(11.75 * Hr)) ^ 2) - 4.97
            factorK = 0

    End Select

    temp = 10 ^ (((Tx - Rx) - (69.55 + 26.16 * Log10(fMHz) - _
        13.82 * Log10(Ht) - factorA - factorK)) / (44.9 - 6.55 * Log10(Ht)))

    HataDist = temp
End Function

```

Function Log10(X)

Log10 = Log(X) / Log(10)

End Function

A.3.2. UTMWGS84

Option Explicit

```
Global GPSx As Double
Global GPSy As Double
Global GPSlon As Double
Global GPSlat As Double
Global GPS As Boolean
Global Zoom As Integer
```

```
Const pi = 3.14159265358979
Const sm_a = 6378137
Const sm_b = 6356752.314
Const sm_EccSquared = 0.00669437999013
Const UTMScaleFactor = 0.9996
Dim zone As Double
Dim xy(0 To 1) As Double
Dim latlon(0 To 1) As Double
Dim philambda(0 To 1) As Double
```

'This Function Converts UTM Easting/Northing (x,y) to WGS84 Datum Longitude and Latitude

'Once you call this function, you will get the longitude
'and Latitude conversion vaules in Global variables GPSlon and GPSlat

```
Function GetLatLon(X As Double, Y As Double, zone As Double)
Dim southhemi As Boolean
```

```
    southhemi = False
    Call UTMXYToLatLon(X, Y, zone, southhemi)
    GPSlat = latlon(0)
    GPSlon = latlon(1)
```

End Function

'This Function Converts WGS84 Datum Longitude and Latitude to UTM Easting/Northing (x,y)

'Once you call this function, you will get the easting
'and northing conversion vaules in Global variables GPSx and GPSy

```
Function GetUTM(lat As Double, lon As Double) As Double
```

```
    zone = (((lon + 180) / 6) + 1)
    If zone < Round((lon + 180) / 6) + 1 Then
        zone = Round(zone) - 1
    Else: zone = Round(zone)
    End If
    zone = LatLonToUTMXY(DegToRad(lat), DegToRad(lon), zone)
    GPSx = Round(xy(0))
    GPSy = Round(xy(1))
```

End Function

```
Private Function DegToRad(deg As Double) As Double
```

```
    DegToRad = deg / 180 * pi
```

```
End Function
```

```
Private Function RadToDeg(rad As Double) As Double
```

```
    RadToDeg = rad / pi * 180
```

```
End Function
```

```
' ArcLengthOfMeridian
```

```
' Computes the ellipsoidal distance from the equator to a point at a
' given latitude.
'
' Reference: Hoffmann-Wellenhof, B., Lichtenegger, H., and Collins, J.,
' GPS: Theory and Practice, 3rd ed. New York: Springer-Verlag Wien, 1994.
'
' Inputs:
'     phi - Latitude of the point, in radians.
'
' Globals:
'     sm_a - Ellipsoid model major axis.
'     sm_b - Ellipsoid model minor axis.
'
' Returns:
'     The ellipsoidal distance of the point from the equator, in meters.
```

```
Private Function ArcLengthOfMeridian(phi As Double) As Double
```

```
Dim alpha, beta, gamma, delta, epsilon, n As Double
```

```
'Precalculate n
```

```
n = (sm_a - sm_b) / (sm_a + sm_b)
```

```
'Precalculate alpha
```

```
alpha = ((sm_a + sm_b) / 2) * (1 + n ^ 2 / 4 + n ^ 4 / 64)
```

```
'Precalculate beta
```

```
beta = (-3 * n / 2) + (9 * n ^ 3 / 16) + (-3 * n ^ 5 / 32)
```

```
'Precalculate gamma
```

```
gamma = (15 * (n ^ 2) / 16) + (-15 * (n ^ 4) / 32)
```

```
'Precalculate delta
```

```
delta = (-35 * (n ^ 3) / 48) + (105 * (n ^ 5) / 256)
```

```
'Precalculate epsilon
```

```
epsilon = (315 * (n ^ 4) / 512)
```

```
'Now calculate the sum of the series and return
```

```
ArcLengthOfMeridian = alpha * (phi + (beta * Sin(2 * phi)) + _
    (gamma * Sin(4 * phi)) + (delta * Sin(6 * phi)) + (epsilon * Sin(8 * phi)))
```

```
End Function
```

```

' UTMCentralMeridian
'
' Determines the central meridian for the given UTM zone.
'
' Inputs:
'   zone - An integer value designating the UTM zone, range [1,60].
'
' Returns:
'   The central meridian for the given UTM zone, in radians, or zero
'   if the UTM zone parameter is outside the range [1,60].
'   Range of the central meridian is the radian equivalent of [-177,+177].

```

```
Private Function UTMCentralMeridian(zone As Double) As Double
```

```
    UTMCentralMeridian = DegToRad(-183 + (zone * 6))
```

```
End Function
```

```

' FootpointLatitude
'
' Computes the footpoint latitude for use in converting transverse
' Mercator coordinates to ellipsoidal coordinates.
'
' Reference: Hoffmann-Wellenhof, B., Lichtenegger, H., and Collins, J.,
'   GPS: Theory and Practice, 3rd ed. New York: Springer-Verlag Wien, 1994.
'
' Inputs:
'   y - The UTM northing coordinate, in meters.
'
' Returns:
'   The footpoint latitude, in radians.

```

```
Private Function FootpointLatitude(Y As Double) As Double
```

```
Dim y_, alpha_, beta_, gamma_, delta_, epsilon_, n As Double
```

```
    'Precalculate n (Eq. 10.18)
```

```
    n = (sm_a - sm_b) / (sm_a + sm_b)
```

```
    'Precalculate alpha_ (Eq. 10.22)
```

```
    alpha_ = ((sm_a + sm_b) / 2) * (1 + (n ^ 2) / 4 + (n ^ 4) / 64)
```

```
    'Precalculate y_ (Eq. 10.23)
```

```
    y_ = Y / alpha_
```

```
    'Precalculate beta_ (Eq. 10.22)
```

```
    beta_ = (3 * n / 2) + (-27 * (n ^ 3) / 32) + (269 * (n ^ 5) / 512)
```

```
    'Precalculate gamma_ (Eq. 10.22)
```

```
    gamma_ = (21 * (n ^ 2) / 16) + (-55 * (n ^ 4) / 32)
```

```
    'Precalculate delta_ (Eq. 10.22)
```

```
    delta_ = (151 * (n ^ 3) / 96) + (-417 * (n ^ 5) / 128)
```

```
    'Precalculate epsilon_ (Eq. 10.22)
```

```
    epsilon_ = (1097 * (n ^ 4) / 512)
```

```
    'Now calculate the sum of the series (Eq. 10.21)
```

```
    FootpointLatitude = y_ + (beta_ * Sin(2 * y_)) + (gamma_ * Sin(4 * y_)) + _
        (delta_ * Sin(6 * y_)) + (epsilon_ * Sin(8 * y_))
```

```
End Function
```

```

' MapLatLonToXY
'
' Converts a latitude/longitude pair to x and y coordinates in the
' Transverse Mercator projection. Note that Transverse Mercator is not
' the same as UTM; a scale factor is required to convert between them.
'
' Reference: Hoffmann-Wellenhof, B., Lichtenegger, H., and Collins, J.,
' GPS: Theory and Practice, 3rd ed. New York: Springer-Verlag Wien, 1994.
'
' Inputs:
'   phi - Latitude of the point, in radians.
'   lambda - Longitude of the point, in radians.
'   lambda0 - Longitude of the central meridian to be used, in radians.
'
' Outputs:
'   xy - A 2-element array containing the x and y coordinates
'        of the computed point.
'
' Returns:
'   The function does not return a value.

Private Function MapLatLonToXY(phi As Double, lambda As Double, lambda0 As Double)
Dim n, nu2, ep2, t, t2, l As Double
Dim l3coef, l4coef, l5coef, l6coef, l7coef, l8coef As Double
Dim tmp As Double

'Precalculate ep2
ep2 = (sm_a ^ 2 - sm_b ^ 2) / sm_b ^ 2

'Precalculate nu2
nu2 = ep2 * (Cos(phi) ^ 2)

'Precalculate N
n = (sm_a ^ 2) / (sm_b * (1 + nu2) ^ (1 / 2))

'Precalculate t
t = Tan(phi)
t2 = t * t
tmp = (t2 * t2 * t2) - t ^ 6

'Precalculate l
l = lambda - lambda0

'Precalculate coefficients for l**n in the equations below so
'a normal human being can read the expressions for easting and northing
' l**1 and l**2 have coefficients of 1.0

l3coef = 1 - t2 + nu2
l4coef = 5 - t2 + 9 * nu2 + 4 * (nu2 * nu2)
l5coef = 5 - 18 * t2 + (t2 * t2) + 14 * nu2 - 58 * t2 * nu2
l6coef = 61 - 58 * t2 + (t2 * t2) + 270 * nu2 - 330 * t2 * nu2
l7coef = 61 - 479 * t2 + 179 * (t2 * t2) - (t2 * t2 * t2)
l8coef = 1385 - 3111 * t2 + 543 * (t2 * t2) - (t2 * t2 * t2)

'Calculate easting(x)

xy(0) = n * Cos(phi) * l + (n / 6 * (Cos(phi) ^ 3) * l3coef * (l ^ 3)) + _
        (n / 120 * (Cos(phi) ^ 5) * l5coef * (l ^ 5)) + _
        (n / 5040 * (Cos(phi) ^ 7) * l7coef * (l ^ 7))

'Calculate northing(y)
xy(1) = ArcLengthOfMeridian(phi) + (t / 2 * n * (Cos(phi) ^ 2) * (l ^ 2)) + _
        (t / 24 * n * Cos(phi) ^ 4 * l4coef * (l ^ 4)) + _
        (t / 720 * n * (Cos(phi) ^ 6) * _

```

$$16coef * (1 ^ 6)) + (t / 40320 * n * (\text{Cos}(\text{phi}) ^ 8) * 18coef * (1 ^ 8))$$

End Function

```
' MapXYToLatLon
'
' Converts x and y coordinates in the Transverse Mercator projection to
' a latitude/longitude pair. Note that Transverse Mercator is not
' the same as UTM; a scale factor is required to convert between them.
'
' Reference: Hoffmann-Wellenhof, B., Lichtenegger, H., and Collins, J.,
' GPS: Theory and Practice, 3rd ed. New York: Springer-Verlag Wien, 1994.
'
' Inputs:
'   x - The easting of the point, in meters.
'   y - The northing of the point, in meters.
'   lambda0 - Longitude of the central meridian to be used, in radians.
'
' Outputs:
'   philambda - A 2-element containing the latitude and longitude
'               in radians.
'
' Returns:
'   The function does not return a value.
'
' Remarks:
'   The local variables Nf, nuf2, tf, and tf2 serve the same purpose as
'   N, nu2, t, and t2 in MapLatLonToXY, but they are computed with respect
'   to the footpoint latitude phif.
'
'   x1frac, x2frac, x2poly, x3poly, etc. are to enhance readability and
'   to optimize computations.

Private Function MapXYToLatLon(X As Double, Y As Double, lambda0 As Double)
Dim phif, Nf, Nfpow, nuf2, ep2, tf, tf2, tf4, cf, x1frac, x2frac, x3frac, x4frac, _
    x5frac, x6frac, x7frac, x8frac As Double
Dim x2poly, x3poly, x4poly, x5poly, x6poly, x7poly, x8poly As Double

'Get the value of phif, the footpoint latitude
phif = FootpointLatitude(Y)
'Precalculate ep2
ep2 = ((sm_a ^ 2) - (sm_b ^ 2)) / (sm_b ^ 2)

'Precalculate cos(phif)
cf = Cos(phif)

'Precalculate nuf2
nuf2 = ep2 * (cf ^ 2)

'Precalculate Nf and initialize Nfpow
Nf = (sm_a ^ 2) / (sm_b * (1 + nuf2) ^ (1 / 2))
Nfpow = Nf

'Precalculate tf
tf = Tan(phif)
tf2 = tf * tf
tf4 = tf2 * tf2

'Precalculate fractional coefficients for x**n in the equations below to
'simplify the expressions for latitude and longitude

x1frac = 1 / (Nfpow * cf)
```



```

Nfpow = Nfpow * Nf
x2frac = tf / (2 * Nfpow)
Nfpow = Nfpow * Nf
x3frac = 1 / (6 * Nfpow * cf)
Nfpow = Nfpow * Nf
x4frac = tf / (24 * Nfpow)
Nfpow = Nfpow * Nf
x5frac = 1 / (120 * Nfpow * cf)
Nfpow = Nfpow * Nf
x6frac = tf / (720 * Nfpow)
Nfpow = Nfpow * Nf
x7frac = 1 / (5040 * Nfpow * cf)
Nfpow = Nfpow * Nf
x8frac = tf / (40320 * Nfpow)

'Precalculate polynomial coefficient
x2poly = -1 - nuf2
x3poly = -1 - 2 * tf2 - nuf2
x4poly = 5 + 3 * tf2 + 6 * nuf2 - 6 * tf2 * nuf2 - 3 * (nuf2 * nuf2) - _
    9 * tf2 * (nuf2 * nuf2)

x5poly = 5 + 28 * tf2 + 24 * tf4 + 6 * nuf2 + 8 * tf2 * nuf2
x6poly = -61 - 90 * tf2 - 45 * tf4 - 107 * nuf2 + 162 * tf2 * nuf2
x7poly = -61 - 662 * tf2 - 1320 * tf4 - 720 * (tf4 * tf2)
x8poly = 1385 + 3633 * tf2 + 4095 * tf4 + 1575 * (tf4 * tf2)

'Calculate Latitude
philambda(0) = phif + x2frac * x2poly * (X * X) + x4frac * x4poly * (X ^ 4) + _
    x6frac * x6poly * (X ^ 6) + x8frac * x8poly * (X ^ 8)

'Calculate Longitude
philambda(1) = lambda0 + x1frac * X + x3frac * x3poly * (X ^ 3) + _
    x5frac * x5poly * (X ^ 5) + x7frac * x7poly * (X ^ 7)

```

End Function

```

' LatLonToUTMXY
'
' Converts a latitude/longitude pair to x and y coordinates in the
' Universal Transverse Mercator projection.
'
' Inputs:
'   lat - Latitude of the point, in radians.
'   lon - Longitude of the point, in radians.
'   zone - UTM zone to be used for calculating values for x and y.
'         If zone is less than 1 or greater than 60, the routine
'         will determine the appropriate zone from the value of lon.
'
' Outputs:
'   xy - A 2-element array where the UTM x and y values will be stored.
'
' Returns:
'   The UTM zone used for calculating the values of x and y.

```

```

Private Function LatLonToUTMXY(lat As Double, lon As Double, zone As Double) _
    As Double

```

```

    Call MapLatLonToXY(lat, lon, UTMCentralMeridian(zone))

```

```

'Adjust easting and northing for UTM system
xy(0) = xy(0) * UTMScaleFactor + 500000
xy(1) = xy(1) * UTMScaleFactor

```

```
If (xy(1) < 0) Then
    xy(1) = xy(1) + 10000000
End If
LatLonToUTMXY = zone
```

End Function

```
' UTMXYToLatLon
'
' Converts x and y coordinates in the Universal Transverse Mercator
' projection to a latitude/longitude pair.
'
' Inputs:
' x - The easting of the point, in meters.
' y - The northing of the point, in meters.
' zone - The UTM zone in which the point lies.
' southhemi - True if the point is in the southern hemisphere;
'             false otherwise.
'
' Outputs:
' latlon - A 2-element array containing the latitude and
'          longitude of the point, in radians.
'
' Returns:
' The function does not return a value.
```

```
Private Function UTMXYToLatLon(X As Double, Y As Double, zone As Double, _
    southhemi As Boolean)
```

```
Dim cmeridian As Double
```

```
X = X - 500000
X = X / UTMScaleFactor
'If in southern hemisphere, adjust y accordingly
If (southhemi) Then
    Y = Y - 10000000

End If
Y = Y / UTMScaleFactor
cmeridian = UTMCentralMeridian(zone)
Call MapXYToLatLon(X, Y, cmeridian)
latlon(0) = RadToDeg(philambda(0))
latlon(1) = RadToDeg(philambda(1))
```

End Function

A.3.3. ModSub

```

Option Explicit
Const pi = 3.14159265358979

Sub SortMatrix()
' This SUB-Procedure performs the sorting of Metriseis() Matrix

Dim X As Integer
Dim k As Integer
Dim i As Integer
Dim aris As Variant
Dim j As Integer

'-----
'***** THE SORTING ALGORITHM *****
'-----
    For X = 2 To 7
        k = 3
        Do While RepTimes > ((k + 1) / 2 - 1)
            If (CLng(Metriseis(k, X)) <> CLng(Metriseis(1, X))) Then
                Call Swap(k, X, FindWhere(Metriseis(1, X), k, X))
            End If
            k = k + 2
        Loop
    Next X
'#####

' The Sorted Matrix is being Print (ONLY in Normal Mode with Debug Matrix Visible)
    For i = 1 To RepTimes * 2
        For j = 1 To 7
            aris = aris & Format(CStr(Metriseis(i, j)), "@@@@")
        Next j
        aris = aris & vbCrLf
        If i Mod 2 = 0 Then
            aris = aris & vbCrLf
        End If
    Next i
    ShowText = aris

End Sub

```

```

Private Function FindWhere(a As Long, b As Integer, c As Integer) As Integer

```

```

' This Function uses Metriseis() Matrix.
' It finds where in (b) row the channel number (a) is located
' If not found it returns (c) which is the current position of the cursor
Dim i As Integer
Dim j As Integer

    j = c
    i = 1
    Do While (i <= 7)
        If Metriseis(b, i) = a Then
            j = i
            i = 8
        Else

```

```

        i = i + 1
    End If
Loop
FindWhere = j

```

End Function

```

Private Sub Swap(a As Integer, b As Integer, c As Integer)
'This SUB-Procedure Swaps two numbers in positions (b) and (c)
'of the (a)th and (a+1)th row of the Matrix Metriseis()

```

```

Dim temp As Long
Dim i As Integer

```

```

For i = 1 To 2
    temp = Metriseis((a + i - 1), b)
    Metriseis((a + i - 1), b) = Metriseis((a + i - 1), c)
    Metriseis((a + i - 1), c) = temp
Next i

```

End Sub

```

Sub ProcessData()

```

```

'This SUB-Procedure checks if the Serving Cell has changed during measurements.
'Afterwards, the algorithm calculates mean values for RXLev of Serving Cell
'and Neighbours and at Last it sorts the Final() Matrix by Order of RXLEV

```

```

Dim i As Integer
Dim j As Integer
Dim SafeMeasure As Integer
Dim divider As Long
Dim Sum As Long
Dim k As Integer
Dim aris As Variant
Dim TAsum As Long
Dim passNum As Integer
Dim temp As Long

```

```

'-----
'***** CHECK IF SERVING CELL HAS CHANGED DURING MEASUREMENTS *****
'-----

```

```

    i = 1
    SafeMeasure = RepTimes

    Do While i < RepTimes

        If cid(i) = cid(i + 1) Then
            i = i + 1
        Else
            SafeMeasure = i
            i = RepTimes
            MsgBox "During Measurements the Serving cell has Changed", vbOKOnly, _
                "Caution!!"
        End If

    Loop

```

```

'#####

```

```

'-----
'***** CALCULATE MEAN VALUES *****
'-----

```

```

k = 0

For i = 1 To 7
  If Metriseis(1, i) = 0 Then
    k = k + 1
  Else
    j = 3
    Sum = Metriseis(2, i)
    divider = 1

    Do While SafeMeasure > ((j + 1) / 2 - 1)
      If Metriseis(j, i) = Metriseis((j - 2), i) Then
        Sum = Sum + Metriseis((j + 1), i)
        divider = divider + 1
        j = j + 2
      Else
        j = 2 * (RepTimes + 1) - 1
      End If
    Loop

    Final(i - k, 1) = Metriseis(1, i)
    Final(i - k, 2) = Sum / divider
  End If
Next i

FinalTA = 0
FinalCID = cid(1)

For i = 1 To SafeMeasure
  FinalTA = FinalTA + TA(i)
Next i
FinalTA = FinalTA / SafeMeasure

'#####

'-----
'***** BUBBLESORT OF THE FINAL RESULTS BY ORDER OF RXLEV *****
'      (Serving Cell's RXLEV is not taken into consideration)
'-----

For passNum = 1 To 5
  For i = 1 To (6 - passNum)
    If Final((i + 1), 2) < Final((i + 2), 2) Then
      For j = 1 To 2
        temp = Final((i + 1), j)
        Final((i + 1), j) = Final((i + 2), j)
        Final((i + 2), j) = temp
      Next j
    End If
  Next i
Next passNum
'#####

'The results are being Print (ONLY in Normal Mode)

For i = 1 To 7
  For j = 1 To 2
    aris = aris & Format(CStr(Final(i, j)), "@@@") & " "
  Next j
  aris = aris & vbCrLf
Next i

```

```
ShowText = aris
```

```
End Sub
```

```
Sub StoreValues()
```

```
'This very simple SUB-Procedure calculates DataMax value (Number of  
'available Cells including serving cell) and stores Final() Matrix  
'into FinalOk() and FinalFull() Matrixes
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
DataMax = 7
```

```
i = 4
```

```
Do While i < 8
```

```
  If Final(i, 2) < 0 Then
```

```
    i = i + 1
```

```
  Else
```

```
    i = 8
```

```
    DataMax = i - 1
```

```
  End If
```

```
Loop
```

```
ReDim FinalOk(1 To DataMax, 1 To 16) As Variant
```

```
ReDim FinalFull(1 To DataMax, 1 To 2) As Variant
```

```
For i = 1 To DataMax
```

```
  For j = 1 To 2
```

```
    FinalOk(i, j) = Final(i, j)
```

```
    FinalFull(i, j) = Final(i, j)
```

```
  Next j
```

```
Next i
```

```
FullDataMax = DataMax
```

```
End Sub
```

```
Sub Databasefind()
```

```
' This SubProcedure performs the Database Search and correlation for the Serving  
'Cell and Neighbours. If a fatal Error occurs DatabaseError variable is set to  
' "TRUE"
```

```
Dim ChnrecMax As Integer
```

```
Dim RecordMax As Integer
```

```
Dim CHNSearch As Long
```

```
Dim SQLString As String
```

```
Dim SQLFind As String
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
Dim k As Integer
```

```
Dim connectString As String
```

```
Dim adoConnection As ADODB.Connection
```

```
Dim adoRecordset As ADODB.Recordset
```

```
Dim chnRecordset As ADODB.Recordset
```

```
Dim srvRecordset As ADODB.Recordset
```

```
Dim temp(1 To 7, 1 To 2) As Long
```

```
Dim findwhat As String
```

```
Dim SameNeighbour As Integer
```

```
Dim CheckSame() As Variant
```

```
arouklas = CStr(FinalCID)
```

```

'Get from Database the Neighbour List for the Serving Cell
SQLString = _
  "SELECT * FROM Neighbours WHERE CELLID = 'G" & arouklas & _
  "' OR CELLID = 'D" & arouklas & "'

'The Character (G) indicates (GSM900)
'The Character (D) indicates (DCS1800)

'Initialize ADODB Connection
Set adoConnection = New ADODB.Connection
connectString = "FILE NAME=C:\AROUKLAS\Location Program\LocDatabase.udl"
adoConnection.Open connectString

Set adoRecordset = adoConnection.Execute(SQLString)

'Change Cursor Type So that Cell count is possible
adoRecordset.Close
adoRecordset.CursorType = adOpenStatic
adoRecordset.LockType = adLockOptimistic
adoRecordset.Open

  If (adoRecordset.EOF) Then 'The Serving Cell has no Neighbour information
    'in the Database
    MsgBox "There is an Error in Database Search (ServingCell Not in DB)", _
      vbOKOnly, "Failed to Collect Data"
    DataBaseError = True
  Else

    'Store required information for Serving Cell in FinalOK Matrix
    adoRecordset.MoveFirst
    arouklas = adoRecordset.Fields("CELLID").Value

    SQLString = "SELECT * FROM Cell_Info WHERE CELLID = '" & arouklas & "'"
    Set srvRecordset = adoConnection.Execute(SQLString)
    If Not srvRecordset.EOF Then

      FinalOk(1, 3) = srvRecordset.Fields("CELLID").Value
      FinalOk(1, 4) = srvRecordset.Fields("Latitude").Value
      FinalOk(1, 5) = srvRecordset.Fields("Longitude").Value
      FinalOk(1, 6) = srvRecordset.Fields("Height").Value
      FinalOk(1, 7) = srvRecordset.Fields("Output_Power").Value
      FinalOk(1, 9) = srvRecordset.Fields("Micro").Value
      FinalOk(1, 10) = srvRecordset.Fields("North").Value
      FinalOk(1, 11) = srvRecordset.Fields("East").Value
      FinalOk(1, 12) = srvRecordset.Fields("Azimuth").Value
      FinalOk(1, 13) = srvRecordset.Fields("Hata").Value
      FinalOk(1, 14) = srvRecordset.Fields("Frequency").Value
      FinalOk(1, 15) = srvRecordset.Fields("Start_Sector").Value
      FinalOk(1, 16) = srvRecordset.Fields("Stop_Sector").Value
    End If
    srvRecordset.Close
    Set srvRecordset = Nothing

    'Try to find the correspondent CELLID for each Neighbour Channel
    For i = 2 To DataMax
      ReDim CheckSame(1 To 5, 1 To 17)
      SameNeighbour = 0
      CHNSearch = FinalOk(i, 1) ' i = 2 ----> Neighbour1 Channel
                              ' i = 3 ----> Neighbour2 Channel
                              ' etc...

      'Find all Cells that have Existing BCCH Channel = CHNSearch
      SQLString = _

```

```

"SELECT * FROM Cell_Info WHERE Existing_BCCH = " & CHNSearch & _
" ORDER BY CELLID"
Set chnRecordset = adoConnection.Execute(SQLString)

'Change Cursor Type So that Cell count is possible
chnRecordset.Close
chnRecordset.CursorType = adOpenStatic
chnRecordset.Open
If Not chnRecordset.EOF Then
    chnRecordset.MoveFirst
    chnRecordset.MoveLast
    ChnrecMax = chnRecordset.RecordCount 'Count how many results we
                                        'have

    chnRecordset.MoveFirst
    j = 1

    Do While (j <= ChnrecMax)
        adoRecordset.MoveFirst

        'Find out which of these results exist in the Neighbour List
        '(it is rare but possible to have more than two Cells that
        'match with our criteria. In this Case we select the Cell which
        'is closer to the Serving Cell
        findwhat = CStr(chnRecordset.Fields("CELLID").Value)
        SQLFind = "NCELL = '" & findwhat & "'"
        adoRecordset.Find SQLFind
        If Not adoRecordset.EOF Then
            SameNeighbour = SameNeighbour + 1 'helpful variable in case
                                            'we have multiple results

            'Temporarily store required information for Neighbour(i-1)
            CheckSame(SameNeighbour, 3) = _
                chnRecordset.Fields("CELLID").Value

            CheckSame(SameNeighbour, 4) = _
                chnRecordset.Fields("Latitude").Value

            CheckSame(SameNeighbour, 5) = _
                chnRecordset.Fields("Longitude").Value

            CheckSame(SameNeighbour, 6) = _
                chnRecordset.Fields("Height").Value

            CheckSame(SameNeighbour, 7) = _
                chnRecordset.Fields("Output_Power").Value

            CheckSame(SameNeighbour, 9) = _
                chnRecordset.Fields("Micro").Value

            CheckSame(SameNeighbour, 10) = _
                chnRecordset.Fields("North").Value

            CheckSame(SameNeighbour, 11) = _
                chnRecordset.Fields("East").Value

            CheckSame(SameNeighbour, 12) = _
                chnRecordset.Fields("Azimuth").Value

            CheckSame(SameNeighbour, 14) = _
                chnRecordset.Fields("Hata").Value

            'In Column 13 we store the distance between The neighbour
            ' and The Serving Cell. This will be used as a criteria in
            ' case of multiple results

```



```

CheckSame(SameNeighbour, 13) = _
  ((CheckSame(SameNeighbour, 10) - FinalOk(1, 10)) ^ 2 + _
  (CheckSame(SameNeighbour, 11) - FinalOk(1, 11)) ^ 2) _
  ^ (1 / 2)

CheckSame(SameNeighbour, 15) = _
  chnRecordset.Fields("Frequency").Value

CheckSame(SameNeighbour, 16) = _
  chnRecordset.Fields("Start_Sector").Value

CheckSame(SameNeighbour, 17) = _
  chnRecordset.Fields("Stop_Sector").Value

If SameNeighbour > 1 Then 'More than 1 result
  'Determine which Cell we are going to correspond.
  'Criteria is the smallest distance from Serving Cell
  If CheckSame(SameNeighbour, 13) < _
    CheckSame(SameNeighbour - 1, 13) Then

    'Replace information for Neighbour(i-1)
    FinalOk(i, 3) = CheckSame(SameNeighbour, 3)
    FinalOk(i, 4) = CheckSame(SameNeighbour, 4)
    FinalOk(i, 5) = CheckSame(SameNeighbour, 5)
    FinalOk(i, 6) = CheckSame(SameNeighbour, 6)
    FinalOk(i, 7) = CheckSame(SameNeighbour, 7)
    FinalOk(i, 9) = CheckSame(SameNeighbour, 9)
    FinalOk(i, 10) = CheckSame(SameNeighbour, 10)
    FinalOk(i, 11) = CheckSame(SameNeighbour, 11)
    FinalOk(i, 12) = CheckSame(SameNeighbour, 12)
    FinalOk(i, 13) = CheckSame(SameNeighbour, 14)
    FinalOk(i, 14) = CheckSame(SameNeighbour, 15)
    FinalOk(i, 15) = CheckSame(SameNeighbour, 16)
    FinalOk(i, 16) = CheckSame(SameNeighbour, 17)
  End If
Else
  'This is the first result ( SameNeighbour =1 )
  'and by default the required information for
  'Neighbour(i-1) are being stored in
  'FinalOk() Matrix
  FinalOk(i, 3) = chnRecordset.Fields("CELLID").Value
  FinalOk(i, 4) = chnRecordset.Fields("Latitude").Value
  FinalOk(i, 5) = chnRecordset.Fields("Longitude").Value
  FinalOk(i, 6) = chnRecordset.Fields("Height").Value
  FinalOk(i, 7) = _
    chnRecordset.Fields("Output_Power").Value

  FinalOk(i, 9) = chnRecordset.Fields("Micro").Value
  FinalOk(i, 10) = chnRecordset.Fields("North").Value
  FinalOk(i, 11) = chnRecordset.Fields("East").Value
  FinalOk(i, 12) = chnRecordset.Fields("Azimuth").Value
  FinalOk(i, 13) = chnRecordset.Fields("Hata").Value
  FinalOk(i, 14) = chnRecordset.Fields("Frequency").Value
  FinalOk(i, 15) = _
    chnRecordset.Fields("Start_Sector").Value

  FinalOk(i, 16) = _
    chnRecordset.Fields("Stop_Sector").Value

End If
j = j + 1
chnRecordset.MoveNext
If j > ChnrecMax Then
  chnRecordset.Close

```

```

        Set chnRecordset = Nothing
    End If
Else
    j = j + 1
    chnRecordset.MoveNext
    If j > ChnrecMax Then 'The search could not match any Cell
                        'With the Neighbour List
        chnRecordset.Close
        Set chnRecordset = Nothing
        If SameNeighbour = 0 Then
            FinalOk(i, 1) = "ERROR"
            FinalOk(i, 3) = "*****"
            FinalOk(i, 4) = "Not a valid"
            FinalOk(i, 5) = "Neighbour"
            FinalOk(i, 2) = "but"
            FinalOk(i, 6) = "Existing"
            FinalOk(i, 7) = "*****"
        End If
    End If
End If

    End If
Loop
Else
    'The search could not find any Cell in the Database
    'with Existing BCCH Channel = (CHNSearch)
    FinalOk(i, 1) = "ERROR"
    FinalOk(i, 3) = "*****"
    FinalOk(i, 4) = "Not Existing"
    FinalOk(i, 5) = "in the"
    FinalOk(i, 2) = "Current"
    FinalOk(i, 6) = "Database"
    FinalOk(i, 7) = "*****"
End If
Next i
End If
adoRecordset.Close
Set adoRecordset = Nothing

```

End Sub

```

Sub ProcessDist()
' This SubProcedure Calculates the "Hata Distances" for all Cells (Serving and
' Neighbour). It is called after the "DatabaseFind" SubProcedure.
Dim i As Integer

If DataBaseError = False Then
    'Store Hata Radius for Serving Cell
    FinalOk(1, 8) = ModPropagation.HataDist(FinalOk(1, 6), 2, _
        CLng(FinalOk(1, 14)), FinalOk(1, 7), FinalOk(1, 2), _
        CInt(FinalOk(1, 13)))

    For i = 2 To DataMax

        If FinalOk(i, 1) <> "ERROR" Then
            'Store Hata Radius for Neighbour Cells ( Neighbour number = i-1)
            FinalOk(i, 8) = ModPropagation.HataDist(FinalOk(i, 6), 2, _
                CLng(FinalOk(i, 14)), FinalOk(i, 7), FinalOk(i, 2), _
                CInt(FinalOk(i, 13)))

        End If

    Next i

```

```
End If

Call GetFinalMatrix
```

End Sub

```
Sub GetFinalMatrix()
' This SubProcedure is called after the "ProcessDist" SubProcedure and calculates
' the new "DataMax" value
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim nr As Integer

i = 2
Do 'The SubProcedure discards all cases where the DatabaseFind Procedure has
'found an Error (Cosite Cells, Cell not found in Database)
If FinalOk(i, 1) = "ERROR" Or CheckCoSite(i) = True Then

    If i = DataMax Then
        DataMax = DataMax - 1
    Else
        For j = i To DataMax - 1
            For k = 1 To 16
                FinalOk(i + j - i, k) = FinalOk(j + 1, k)
            Next k
        Next j
        DataMax = DataMax - 1
        i = i - 1
    End If
End If
i = i + 1
Loop Until i > DataMax
```

End Sub

```
Function CheckCoSite(i As Integer) As Boolean
'This Function Checks for Co-Site Cells. Input i is the number of the cell that we
'Want to Check (eg. i = 3 ----> is the 2nd Neighbour)
Dim j As Integer
Dim check As Boolean

j = 1
check = False
Do
    If FinalOk(i, 10) = FinalOk(j, 10) And FinalOk(i, 11) = FinalOk(j, 11) _
        Then check = True

    j = j + 1
Loop Until j = i
CheckCoSite = check
```

End Function

```
Function MicroCheck() As Boolean
' This Function performs the following checks:
' 1: If there is no info for Neighbour Cells (DataMax = 1)
```

```

' 2: If the Serving Cell is a Microcell
' 3: (OPTIONAL) If TA = 0
' 4: (OPTIONAL) If RXLev of Serving Cell is < -55dBm
Dim i As Integer
Dim istrue As Boolean

    istrue = False

    If DataMax = 1 Then 'Or (FinalOk(1, 2) > (-55)) Or FinalTA = 0
        MicroBTS = 1
        istrue = True
    End If
    For i = 1 To DataMax
        If FinalOk(i, 9) > 0 Then
            MicroBTS = i
            istrue = True
        End If
    Next i
    MicroCheck = istrue

End Function

```

```

Sub NotLocate()
'This SubProcedure is being called only if MicroCheck = True
'In this subProcedure the program bypasses the location algorithm
'and sets the Serving Cells' Longitude/Latitude as the Calculated Position

Dim TestLimit As Double

    'The Maximum Value of FinalTA can be 1
    If FinalTA > 1 Then FinalTA = 1

    'Store Calculated Location
    CalcX = FinalOk(MicroBTS, 11) / 1000 'X Coordinate
    CalcY = FinalOk(MicroBTS, 10) / 1000 'Y Coordinate
    CalcLon = FinalOk(MicroBTS, 5) 'Longitude
    CalcLat = FinalOk(MicroBTS, 4) 'Latitude

    'Check if HATA(1) Radius of ServCell is smaller than TA Outer Radius.
    'If this is true, set outer Radius of Trust Area = HATA1 Radius
    TestLimit = ModPropagation.HataDist(FinalOk(MicroBTS, 6), 2, _
        CLng(FinalOk(MicroBTS, 14)), FinalOk(MicroBTS, 7), FinalOk(MicroBTS, 2), 1)

    If FinalOk(MicroBTS, 9) > 0 Then
        If TestLimit < Cdbl(FinalOk(MicroBTS, 9)) Then
            OutRad = TestLimit
        Else
            OutRad = Cdbl(FinalOk(MicroBTS, 9))
        End If
    Else
        If TestLimit < (0.55 * (FinalTA + 1) + 0.1) Then
            OutRad = TestLimit
        Else
            OutRad = (0.55 * (FinalTA + 1) + 0.1)
        End If
    End If

    '-----
    '***** DEFINE TRUST AREA *****
    '-----

```

```

'TRUST_TYPE ..... Type of Trust Area
'TRUST_ANGLE_START ..... The Angle where the Sector Starts (North Ref)
'TRUST_ANGLE_STOP ..... The Angle where the Sector Stops (North Ref)
'TRUST_LON ..... Longitude of the center of the Circular Sector
'TRUST_LAT ..... Latitude of the center of the Circular Sector
'TRUST_CIRCLE_BIG ..... The Outer Radius of the Sector
'TRUST_CIRCLE_SMALL .... The inner Radius of the Sector

TRUST_TYPE = "SECTOR"
TRUST_LON = Format(FinalOk(MicroBTS, 5), "###.000000")
TRUST_LAT = Format(FinalOk(MicroBTS, 4), "###.000000")
TRUST_CIRCLE_BIG = OutRad
TRUST_CIRCLE_SMALL = 0
TRUST_ANGLE_START = FinalOk(MicroBTS, 15)
TRUST_ANGLE_STOP = FinalOk(MicroBTS, 16)

#####

-----
***** FILL "frmFinal" LABELS WITH CALCULATED DATA *****
-----

frmFinal.lblTrust.Caption = TRUST_TYPE & vbCrLf & TRUST_LON & "°" & _
    TRUST_LAT & "°" & vbCrLf & Format(TRUST_CIRCLE_BIG, "##0.000") & _
    "(km)" & _
    TRUST_CIRCLE_SMALL & "(km)" & vbCrLf & TRUST_ANGLE_START & "°" & _
    TRUST_ANGLE_STOP & "°" 'The Trust Area

If (GPS = True) Then
    Call UTMWGS84.GetUTM(GPSlat, GPSlon)
    GPSx = GPSx / 1000
    GPSy = GPSy / 1000
    'Error: the distance between the Calculated and the GPS position
    Error = Format(1000 * ((CalcX - GPSx) ^ 2 + (CalcY - GPSy) ^ 2) ^ _
        (1 / 2), "##0.0")

    frmFinal.lblERR = Error
    'The Distance Between the GPS and the Serving Cell Location
    frmFinal.lblDist = Error
    GPSDistance = Error
    frmFinal.lblGPSLon = GPSlon 'The GPS Longitude
    frmFinal.lblGPSLat = GPSlat 'The GPS Latitude
    frmFinal.lblHATA = "Unknown"

Else
    frmFinal.lblERR = "Unknown"
    frmFinal.lblDist = "Unknown"
    frmFinal.lblGPSLon = "Unknown"
    frmFinal.lblGPSLat = "Unknown"
    frmFinal.lblHATA = "Unknown"

End If

Stoptime = Now

frmFinal.lblTime.Caption = Format(CDb1(Stoptime - Starttime) * 100000, "##.00")
frmFinal.lblCalcLon = Format(CalcLon, "##.000000") 'Calculated Position
    'Longitude

frmFinal.lblCalcLat = Format(CalcLat, "##.000000") 'Calculated Position
    'Latitude

```

```
'#####
```

```
End Sub
```

```
Sub DrawMicro()
' This SubProcedure is called after the NotLocate() subprocedure.
' It draws the results in the "frmFinal" form, in order to have a visual
' representation of the environment and the accuracy of the Algorithm.
Dim MaxY As Variant
Dim MaxX As Variant
Dim MinY As Variant
Dim MinX As Variant
Dim MeasureX As Double
Dim MeasureY As Double

'Convert Start and Stop Sector angles into Radians and format them properly
'so that they can be drawn in "frmFinal" PictureBox
StopSector = (5 * pi / 2 - ((FinalOk(1, 15)) * pi / 180)) * (-1)
If StopSector < -2 * pi Then StopSector = StopSector + 2 * pi
StartSector = (5 * pi / 2 - ((FinalOk(1, 16)) * pi / 180)) * (-1)
If StartSector < -2 * pi Then StartSector = StartSector + 2 * pi

frmFinal.PictureBox2.Cls 'Clear the PictureBox from previous drawings

'-----
' Calculate MaxX, MaxY, MinX and MinY in order to determine the picture's
' width/height and scale
'-----

MaxY = FinalOk(1, 10) / 1000 + TRUST_CIRCLE_BIG
MaxX = FinalOk(1, 11) / 1000 + TRUST_CIRCLE_BIG
MinY = FinalOk(1, 10) / 1000 - TRUST_CIRCLE_BIG
MinX = FinalOk(1, 11) / 1000 - TRUST_CIRCLE_BIG

If GPS = True Then
    If GPSy + 0.5 > MaxY Then MaxY = GPSy + 0.5
    If GPSx + 0.5 > MaxX Then MaxX = GPSx + 0.5
    If GPSy - 0.5 < MinY Then MinY = GPSy - 0.5
    If GPSx - 0.5 < MinX Then MinX = GPSx - 0.5
End If

If MicroBTS > 1 Then
    If CalcY + 0.5 > MaxY Then MaxY = CalcY + 0.5
    If CalcX + 0.5 > MaxX Then MaxX = CalcX + 0.5
    If CalcY - 0.5 < MinY Then MinY = CalcY - 0.5
    If CalcX - 0.5 < MinX Then MinX = CalcX - 0.5
End If

MeasureX = MaxX - MinX
MeasureY = MaxY - MinY
frmFinal.PictureBox2.Width = 10000 * MeasureX / Zoom 'Set PictureBox Width
frmFinal.PictureBox2.Height = 10000 * MeasureY / Zoom 'Set PictureBox Height
'4KM FOR EVERY 10000PIXELS if Zoom = 4

'Set the Scale of the PictureBox according to MinX, MinY, MaxX, MaxY
frmFinal.PictureBox2.Scale (MinX - 0.03, MaxY - 0.03)-(MaxX + 0.03, MinY + 0.03)

'#####
```

```

'-----
'*****DRAW THE TRUST AREA*****
'-----

frmFinal.Picture2.FillStyle = 0
frmFinal.Picture2.FillColor = &H606060
frmFinal.Picture2.Circle (FinalOk(1, 11) / 1000, FinalOk(1, 10) / 1000), _
    TRUST_CIRCLE_BIG, vbBlack, StartSector, StopSector

frmFinal.Picture2.FillStyle = 0
frmFinal.Picture2.FillColor = vbBlack
frmFinal.Picture2.Circle (FinalOk(1, 11) / 1000, FinalOk(1, 10) / 1000), _
    TRUST_CIRCLE_SMALL, vbBlack

frmFinal.Picture2.FillStyle = 1
frmFinal.Picture2.ForeColor = vbYellow

'#####

'-----
'***** DRAW SERVING CELL AND GPS *****
'-----

frmFinal.Picture2.FillStyle = 0
frmFinal.Picture2.FillColor = vbRed
frmFinal.Picture2.Circle (FinalOk(1, 11) / 1000, FinalOk(1, 10) / 1000), _
    0.03, vbRed

frmFinal.Picture2.FillStyle = 1

If MicroBTS > 1 Then
    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = vbRed
    frmFinal.Picture2.Circle (CalcX, CalcY), 0.03, vbRed
    frmFinal.Picture2.FillStyle = 1
    frmFinal.Picture2.PSet (CalcX, CalcY)
    frmFinal.Picture2.Print "MicroCell(N" & MicroBTS & ")"
    frmFinal.Picture2.PSet (FinalOk(1, 11) / 1000, FinalOk(1, 10) / 1000)
    frmFinal.Picture2.Print "Serving Cell"
Else
    frmFinal.Picture2.PSet (FinalOk(1, 11) / 1000, FinalOk(1, 10) / 1000)
    frmFinal.Picture2.Print "Serving Cell"

End If

If GPS = True Then
    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = vbGreen
    frmFinal.Picture2.Circle (GPSx, GPSy), 0.015, vbGreen
    frmFinal.Picture2.FillStyle = 1
    frmFinal.Picture2.Print "GPS"
    frmFinal.Picture2.FillColor = vbRed

End If

'#####

'Initialize "frmFinal" form and make it Visible
Call frmFinal.Initialize

```

```
frmFinal.Enabled = True
frmFinal.Visible = True
```

End Sub

```
Sub TAHATAMatch()
'This SubProcedure processes the HATA Radius of the serving Cell in conjunction
'With the TA Area. In few words, this subprocedure tries to fit Hata Radius into
'the TA Area (if it is outside by default)
Dim HataType As Integer

'The Maximum Value of FinalTA can be 1
If FinalTA > 1 Then FinalTA = 1

Do While (FinalOk(1, 8) > 0.55 * (FinalTA + 1) + 0.1) 'while HATA Radius is
'outside
'TA Area Outer Radius
'limit.

HataType = FinalOk(1, 13)

Select Case UCase(HataType)
'Here we take the the next type of Hata Propagation Environment, so that
'the HATA Radius becomes smaller and in this way it is possible to meet
'with our criteria.

    Case 1
        FinalOk(1, 13) = 2
        FinalOk(1, 8) = ModPropagation.HataDist(FinalOk(1, 6), 2, _
        CLng(FinalOk(1, 14)), FinalOk(1, 7), FinalOk(1, 2), CInt(FinalOk(1, 13)))

    Case 2
        FinalOk(1, 13) = 3
        FinalOk(1, 8) = ModPropagation.HataDist(FinalOk(1, 6), 2, _
        CLng(FinalOk(1, 14)), FinalOk(1, 7), FinalOk(1, 2), CInt(FinalOk(1, 13)))

    Case 3
        FinalOk(1, 13) = 4
        FinalOk(1, 8) = ModPropagation.HataDist(FinalOk(1, 6), 2, _
        CLng(FinalOk(1, 14)), FinalOk(1, 7), FinalOk(1, 2), CInt(FinalOk(1, 13)))

    Case 4
        Exit Do

End Select
Loop
'If no HATA Type could be found to bring the desirable result
'then, the HATA Radius is taken in the middle of the TA-Area
If (FinalOk(1, 8) > 0.55 * (FinalTA + 1) + 0.1) Then

    FinalOk(1, 8) = 0.55 * (FinalTA + 1) - 0.3625

End If

'If HATA Radius is Smaller than the inner radius of the TA Area (TA = 1),
'then we set TA = 0
If (FinalOk(1, 8) < 0.55 * (FinalTA) - 0.275) Then FinalTA = 0
```

End Sub

```

Sub CalcDistances()
' This SubProcedure performs "partly" the Location Algorithm
' It also uses the CheckNeighbourHata subProcedure
Dim tempX As Double
Dim tempY As Double
Dim x0 As Double
Dim y0 As Double
Dim R0 As Double
Dim FinalX0 As Double
Dim FinalY0 As Double
Dim thita As Long
Dim TotalDistance() As Double
Dim Distance() As Double
Dim Leastthita As Long
Dim i As Long
Dim j As Long
Dim ScanX As Double
Dim ScanY As Double
Dim k As Integer
Dim LeastDistance As Double
Dim Count1 As Long
Dim TestAzimuth As Long
Dim TestLimit As Double

If GPS = True Then 'Get the Coordinates of the GPS in Easting and Northing
                    ' (X,Y) If there is a GPS Position.

    Call UTMWGS84.GetUTM(GPSlat, GPSlon)
    GPSx = GPSx / 1000 'Convert from Meters to Kilometers
    GPSy = GPSy / 1000 'Convert from Meters to Kilometers
End If

ReDim GraphMatr(1 To DataMax, 1 To 10) 'Set the dimensions of GraphMatr()
                                     'Matrix
ReDim heavy(1 To DataMax, 1 To 3)    'Set the dimensions of heavy() Matrix

For i = 1 To DataMax
    'Store Data in GraphMatr() Matrix for more ease

    GraphMatr(i, 2) = FinalOk(i, 10) / 1000 'NorthY
    GraphMatr(i, 3) = FinalOk(i, 11) / 1000 'EastX
    GraphMatr(i, 4) = FinalOk(i, 8)         'Radius
    GraphMatr(i, 5) = FinalOk(i, 12)        'Azimuth Direction
    GraphMatr(i, 6) = FinalOk(i, 15)        'Sector Start
    GraphMatr(i, 7) = FinalOk(i, 16)        'Sector Stop

    'If StartSector Value is bigger than StopSector, then
    'StartSector = StartSector - 360 ( < 0 )
    If GraphMatr(i, 6) > GraphMatr(i, 7) Then GraphMatr(i, 6) = _
        GraphMatr(i, 6) - 360

Next i

ReDim TotalDistance(0 To 7200, 2 To DataMax + 1) As Double
'Last Column (Datamax+1) is used to store the Sum of all distances

x0 = GraphMatr(1, 3)
y0 = GraphMatr(1, 2)
R0 = GraphMatr(1, 4)

```

```

Call CheckNeighbourHata
'Process the Neighbours Hata Radius and find the proper "heavy" values

For thita = GraphMatr(1, 6) * 10 To GraphMatr(1, 7) * 10
'If you want to scan the whole 360degrees area and not only the sector:
'"0 To 3600"
'In this case Trust Area may be not a Sector but a Circular Disc
'thita : The angle between North and the Azimuth of the test point
'(the position where we want to calculate the distances). "thita" value
'depends on the serving cell's Start and Stop Sector.
'Since StartSector Can take negative values, so does
'"thita"

    'Set the Test Point (TP) Coordinates (X,Y)
    tempx = x0 + R0 * Sin(AsRadians(CDbl(thita / 10)))
    tempy = y0 + R0 * Cos(AsRadians(CDbl(thita / 10)))

    For i = 2 To DataMax 'Calculate the distance of the TP from every Neighbour
        ' i = 2 ---> Neighbour1
        ' i = 3 ---> Neighbour2
        ' etc.

        If heavy(i, 3) <> 0 Then 'distance is calculated only if there is a
            'valid heavy value ( <> 0 )

            'Calculate distance from Neighbour (i-1)
            TotalDistance(thita + 3600, i) = Abs((((tempx - GraphMatr(i, 3)) ^ 2 _
                + (tempy - GraphMatr(i, 2)) ^ 2) ^ (1 / 2) - GraphMatr(i, 4)) + _
                Abs((((tempx - heavy(i, 1)) ^ 2 + _
                    (tempy - heavy(i, 2)) ^ 2) ^ (1 / 2)))
            'Add all Distances (multiplied with "heavy" coefficient) and store the
            'sum in the (DataMax + 1) Column
            TotalDistance(thita + 3600, DataMax + 1) = _
                TotalDistance(thita + 3600, _
                    DataMax + 1) + heavy(i, 3) * TotalDistance(thita + 3600, i)

        End If

    Next i 'Repeat for every "thita"

    'Find Which sum is the smallest and store "thita" in "Leastthita"
    If thita = GraphMatr(1, 6) * 10 Then
        Leastthita = thita 'initialize Leastthita
    Else
        If TotalDistance(thita + 3600, DataMax + 1) < _
            TotalDistance(Leastthita + 3600, DataMax + 1) Then

            Leastthita = thita
        End If
    End If

    End If

Next thita

'Get the Coordinates (X,Y) for LeastThita which is the estimated
'location of the M/T

FinalX0 = x0 + R0 * Sin(AsRadians(Leastthita / 10)) 'X Coordinate
FinalY0 = y0 + R0 * Cos(AsRadians(Leastthita / 10)) 'Y Coordinate

CalcX = FinalX0
CalcY = FinalY0

For i = 2 To DataMax

```

```

    If heavy(i, 3) = 0 Then
        GraphMatr(i, 4) = 0
    Else
        FinalOk(i, 8) = GraphMatr(i, 4)
    End If
Next i

'-----
'***** PRE-CALCULATIONS FOR "TRUST AREA" *****
'-----

TestAzimuth = FindAzimuth(CalcX, CalcY, CDb1(GraphMatr(1, 3)), _
    CDb1(GraphMatr(1, 2)))

'This is the Azimuth direction of the (CalcX, CalcY) point

'Check if it is inside the Serving Cells' Sector.
'For the case where the location
'Algorithm performs the search only inside the sector:
'ALWAYS ----> Sector = True!
If TestAzimuth > GraphMatr(1, 6) And TestAzimuth < GraphMatr(1, 7) Then
    Sector = True
    'Convert Start and Stop Sector angles into Radians and format them properly
    'so that they can be drawn in "frmFinal" PictureBox
    StopSector = (5 * pi / 2 - ((GraphMatr(1, 6)) * pi / 180)) * (-1)
    If StopSector < -2 * pi Then StopSector = StopSector + 2 * pi
    StartSector = (5 * pi / 2 - ((GraphMatr(1, 7)) * pi / 180)) * (-1)
    If StartSector < -2 * pi Then StartSector = StartSector + 2 * pi

Else
    Sector = False

End If

'Check if HATA1 Radius of ServCell is smaller than TA Outer Radius.
'If this is true, set outer Radius of Trust Area = HATA1 Radius
TestLimit = ModPropagation.HataDist(FinalOk(1, 6), 2, CLng(FinalOk(1, 14)), _
    FinalOk(1, 7), FinalOk(1, 2), 1)

If TestLimit < (0.55 * (FinalTA + 1)) + 0.1 Then
    OutRad = TestLimit
Else
    OutRad = (0.55 * (FinalTA + 1)) + 0.1
End If
'#####

'-----
'***** DEFINE TRUST AREA *****
'-----
'TRUST_TYPE ..... Type of Trust Area
'TRUST_ANGLE_START ..... The Angle where the Sector Starts (North Ref)
'TRUST_ANGLE_STOP ..... The Angle where the Sector Stops (North Ref)
'TRUST_LON ..... Longitude of the center of the Circular Sector
'TRUST_LAT ..... Latitude of the center of the Circular Sector
'TRUST_CIRCLE_BIG ..... The Outer Radius of the Sector
'TRUST_CIRCLE_SMALL .... The inner Radius of the Sector

If Sector Then
    TRUST_TYPE = "SECTOR"
    TRUST_ANGLE_START = FinalOk(1, 15)
    TRUST_ANGLE_STOP = FinalOk(1, 16)

```

```

Else
    TRUST_TYPE = "CIRCLE"
    TRUST_ANGLE_START = 0
    TRUST_ANGLE_STOP = 360

End If

TRUST_LON = Format(FinalOk(1, 5), "###.000000")
TRUST_LAT = Format(FinalOk(1, 4), "###.000000")
TRUST_CIRCLE_BIG = OutRad

If FinalTA = 1 Then

    TRUST_CIRCLE_SMALL = 0.55 * (FinalTA) - 0.275

Else
    TRUST_CIRCLE_SMALL = 0

End If

'#####

'-----
'***** FILL "frmFinal" LABELS WITH CALCULATED DATA *****
'-----

frmFinal.lblTrust.Caption = TRUST_TYPE & vbCrLf & TRUST_LON & "°" & _
    TRUST_LAT _
    & "°" & vbCrLf & Format(TRUST_CIRCLE_BIG, "##0.000") & "(km)" & _
    Format(TRUST_CIRCLE_SMALL, "##0.000") & "(km)" & vbCrLf & _
    TRUST_ANGLE_START & _
    "°" & TRUST_ANGLE_STOP & "°" 'The Trust Area

If (GPS = True) Then

    'Error: the distance between the Calculated and the GPS Location
    Error = _
        Format(1000 * ((CalcX - GPSx) ^ 2 + (CalcY - GPSy) ^ 2) ^ (1 / 2), _
            "##0.0")
    frmFinal.lblERR = Error

    'The Distance Between the GPS and the Serving Cell Location
    frmFinal.lblDist = Format(1000 * ((GraphMatr(1, 3) - GPSx) ^ 2 + _
        (GraphMatr(1, 2) - GPSy) ^ 2) ^ (1 / 2), "##0.0")

    GPSDistance = CDb1(frmFinal.lblDist)

    frmFinal.lblGPSLon = GPSLon 'The GPS Longitude
    frmFinal.lblGPSLat = GPSLat 'The GPS Latitude

    'The distance between the Serving Cells' HATA Radius and the GPS Point
    frmFinal.lblHATA = Format(1000 * Abs(frmFinal.lblDist / 1000 - _
        GraphMatr(1, 4)), "##0.0")

    Call UTMWGS84.GetLatLon(CalcX * 1000, CalcY * 1000, 34)
    CalcLon = GPSLon
    CalcLat = GPSLat
    frmFinal.lblCalcLon = Format(GPSLon, "###.000000")
    CalcLon = GPSLon 'Calculated Position Longitude
    frmFinal.lblCalcLat = Format(GPSLat, "###.000000")
    CalcLat = GPSLat 'Calculated Position Latitude
    GPSLon = frmFinal.lblGPSLon'Re-assign true GPS longitude to GPSLon variable
    GPSLat = frmFinal.lblGPSLat'Re-assign true GPS latitude to GPSLat variable
    Stoptime = Now

```

```

>Show time required for Position Calculation
frmFinal.lblTime.Caption = _
    Format(CDbl(Stoptime - Starttime) * 100000, "##.00")

Else
frmFinal.lblERR = "Unknown"
frmFinal.lblDist = "Unknown"
frmFinal.lblGPSLon = "Unknown"
frmFinal.lblGPSLat = "Unknown"
frmFinal.lblHATA = "Unknown"

Call UTMWGS84.GetLatLon(CalcX * 1000, CalcY * 1000, 34)
frmFinal.lblCalcLon = Format(GPSlon, "##.000000")
frmFinal.lblCalcLat = Format(GPSlat, "##.000000")
GPSlon = 0
GPSlat = 0
Stoptime = Now
frmFinal.lblTime.Caption = _
    Format(CDbl(Stoptime - Starttime) * 100000, "##.00")

End If
#####

End Sub

```

```

Private Sub CheckNeighbourHata()
'This SubProcedure processes Neighbours' HATA Radius and calculates "heavy" matrix.
Dim i As Integer
Dim CheckNeighbour As Boolean
Dim TestAzimuth As Long
Dim TestAzimuth1 As Long
Dim CheckMe As Long

Dim Try As Integer

ReDim NeighbourCross(2 To DataMax, 1 To 4)

For i = 2 To DataMax 'Calculate distance between Neighbour(i-1) and Serv. Cell
GraphMatr(i, 8) = ((GraphMatr(1, 3) - GraphMatr(i, 3)) ^ 2 + _
    (GraphMatr(1, 2) - GraphMatr(i, 2)) ^ 2) ^ (1 / 2)

GraphMatr(i, 9) = FinalOk(i, 13) 'HATA type
Next i

For i = 2 To DataMax

CheckNeighbour = False
Try = 0

'distance = distance between Neighbour and Serving Cell
'Serving Cell: (x0,y0)
'Neighbour: (x1,y1)
'distance = [(x0-x1)^2 + (y0-y1)^2 ]^(1/2)
'R0 = HATA Radius of Serving Cell
'R1 = HATA Radius of Neighbour

Do

```

```

#####
'Case A (distance > R0 + R1) --> No Cross-Section Points
#####

If (GraphMatr(i, 8) > (GraphMatr(1, 4) + GraphMatr(i, 4))) Then
  'Azimuth Direction (Serving Cell --> Neighbour)
  TestAzimuth = CLng(FindAzimuth(GraphMatr(i, 3), GraphMatr(i, 2), _
    GraphMatr(1, 3), GraphMatr(1, 2)))

  If GraphMatr(1, 6) < 0 Then
    If TestAzimuth > GraphMatr(1, 6) + 360 Then TestAzimuth = _
      TestAzimuth - 360

  End If

  'Check if the TestAzimuth is inside Serving Cell's Sector
  If TestAzimuth > GraphMatr(1, 6) And TestAzimuth < GraphMatr(1, 7) Then
    CheckNeighbour = True
    'Store "heavy" data:
    'Column 1 = X-Position of the point onto HATA Radius of Serving Cell
    '          which is in the direction of "TestAzimuth"
    'Column 2 = Y-Position .....
    'Column 3 = [10^(-3)] / [ distance - R0 -R1]
    heavy(i, 1) = _
      GraphMatr(1, 4) * Sin(AsRadians(TestAzimuth)) + GraphMatr(1, 3)

    heavy(i, 2) = _
      GraphMatr(1, 4) * Cos(AsRadians(TestAzimuth)) + GraphMatr(1, 2)
    heavy(i, 3) = _
      0.001 / (GraphMatr(i, 8) - GraphMatr(i, 4) - GraphMatr(1, 4))
  Else
    'If possible: HATA type = HATA type - 1
    If GraphMatr(i, 9) = 1 Or Try = 3 Then
      CheckNeighbour = True
      heavy(i, 1) = 0
      heavy(i, 2) = 0
      heavy(i, 3) = 0

    Else
      Try = Try + 1 'Maximum number of tries = 3
      GraphMatr(i, 9) = GraphMatr(i, 9) - 1
      GraphMatr(i, 4) = ModPropagation.HataDist(FinalOk(i, 6), 2, _
        CLng(FinalOk(i, 14)), FinalOk(i, 7), FinalOk(i, 2), _
        CInt(GraphMatr(i, 9)))

    End If

  End If

End If

#####
' Case B (distance > R1 And distance < R0 + R1) --> 2 Cross-Section Points
'Cross-Section Point 1 : (Xp1,Yp1)
'Cross-Section Point 2 : (Xp2,Yp2)
#####

ElseIf (GraphMatr(i, 8) < (GraphMatr(1, 4) + GraphMatr(i, 4))) And _
  (GraphMatr(i, 8) > GraphMatr(i, 4)) Then

  Call FindCrossSection(GraphMatr(i, 3), GraphMatr(i, 2), GraphMatr(1, 3), _
    GraphMatr(1, 2), GraphMatr(i, 4), GraphMatr(1, 4), i)

```

```

'Check First For X1,Y1 Cross Section Point
TestAzimuth = CLng(FindAzimuth(NeighbourCross(i, 1), _
    NeighbourCross(i, 2), _
    GraphMatr(1, 3), GraphMatr(1, 2)))

If GraphMatr(1, 6) < 0 Then
    If TestAzimuth > GraphMatr(1, 6) + 360 Then TestAzimuth = _
        TestAzimuth - 360

End If
'Check if the TestAzimuth is inside Serving Cell's Sector
If TestAzimuth > GraphMatr(1, 6) And TestAzimuth < GraphMatr(1, 7) Then
    'Check now for X2,Y2 Cross Section Point
    TestAzimuth1 = CLng(FindAzimuth(NeighbourCross(i, 3), _
        NeighbourCross(i, 4), GraphMatr(1, 3), GraphMatr(1, 2)))

    If GraphMatr(1, 6) < 0 Then
        If TestAzimuth1 > GraphMatr(1, 6) + 360 Then TestAzimuth1 = _
            TestAzimuth1 - 360
    End If
    'Check if the TestAzimuth1 is inside Serving Cell's Sector
    If TestAzimuth1 > GraphMatr(1, 6) And _
        TestAzimuth1 < GraphMatr(1, 7) Then

        'decide which Cross-Section Point to use
        If Abs(TestAzimuth1 - GraphMatr(1, 5)) < Abs(TestAzimuth - _
            GraphMatr(1, 5)) Then
            'Two points within Sector, with the second closer
            'to the Azimuth
            'Store "heavy" data:
            'Column 1 = Second cross section point X-Position (Xp2)
            'Column 2 = Second cross section point Y-Position (Yp2)
            'Column 3 = [10^(-3)] / (|distance - R0 - R1|)
            heavy(i, 1) = NeighbourCross(i, 3)
            heavy(i, 2) = NeighbourCross(i, 4)
            heavy(i, 3) = 0.001 / Abs(GraphMatr(i, 8) - GraphMatr(1, 4) - _
                GraphMatr(i, 4))

            CheckNeighbour = True

        Else
            'Two points within Sector, with the first closer to the Azimuth
            'Store "heavy" data:
            'Column 1 = First cross section point X-Position (Xp1)
            'Column 2 = First cross section point Y-Position (Yp1)
            'Column 3 = [10^(-3)] / (|distance - R0 - R1|)
            heavy(i, 1) = NeighbourCross(i, 1)
            heavy(i, 2) = NeighbourCross(i, 2)
            heavy(i, 3) = 0.001 / Abs(GraphMatr(i, 8) - GraphMatr(1, 4) - _
                GraphMatr(i, 4))

            CheckNeighbour = True

        End If

    Else
        'Only the first point is within the Sector
        'Store "heavy" data:
        'Column 1 = First cross section point X-Position (Xp1)
        'Column 2 = First cross section point Y-Position (Yp1)
        'Column 3 = [8*10^(-4)] / (|distance - R0 - R1|)

        heavy(i, 1) = NeighbourCross(i, 1)
        heavy(i, 2) = NeighbourCross(i, 2)

```

```

heavy(i, 3) = 0.0008 / Abs(GraphMatr(i, 8) - GraphMatr(1, 4) - _
    GraphMatr(i, 4))

CheckNeighbour = True

End If

Else
'Check now for X2,Y2 Cross Section Point
TestAzimuth1 = CLng(FindAzimuth(NeighbourCross(i, 3), _
    NeighbourCross(i, 4), GraphMatr(1, 3), GraphMatr(1, 2)))

If GraphMatr(1, 6) < 0 Then
    If TestAzimuth1 > GraphMatr(1, 6) + 360 Then TestAzimuth1 = _
        TestAzimuth1 - 360

End If
If TestAzimuth1 > GraphMatr(1, 6) And TestAzimuth1 < _
    GraphMatr(1, 7) Then

'Only the Second point within the Sector
'Store "heavy" data:
'Column 1 = Second cross section point X-Position (Xp1)
'Column 2 = Second cross section point Y-Position (Yp1)
'Column 3 = [8*10(-4)] / (|distance - R0 - R1|)
heavy(i, 1) = NeighbourCross(i, 3)
heavy(i, 2) = NeighbourCross(i, 4)
heavy(i, 3) = 0.0008 / Abs(GraphMatr(i, 8) - GraphMatr(1, 4) - _
    GraphMatr(i, 4))

CheckNeighbour = True

Else
'Both Cross-Section Points are outside the sector
'The Program has to decide whether to take a bigger HATA
'or a smaller one.
If TestAzimuth < 0 Then TestAzimuth = TestAzimuth + 360
If TestAzimuth1 < 0 Then TestAzimuth1 = TestAzimuth1 + 360

If TestAzimuth1 > TestAzimuth Then
    CheckMe = TestAzimuth1 - TestAzimuth
    If CheckMe < 180 Then
        If GraphMatr(1, 6) > TestAzimuth And GraphMatr(1, 6) < _
            TestAzimuth1 And GraphMatr(1, 7) > TestAzimuth And _
            GraphMatr(1, 7) < TestAzimuth1 Then

            If GraphMatr(i, 9) < 4 And Try < 3 Then
                'HATA Type = HATA Type + 1
                GraphMatr(i, 9) = GraphMatr(i, 9) + 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _
                        2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                        FinalOk(i, 2), CInt(GraphMatr(i, 9)))

                Try = Try + 1 'Maximum number of tries = 3
            Else
                CheckNeighbour = True
                heavy(i, 1) = 0
                heavy(i, 2) = 0
                heavy(i, 3) = 0
                Try = 0
            End If
        Else
            If GraphMatr(i, 9) > 1 And Try < 3 Then
                'HATA Type = HATA Type - 1

```



```

        GraphMatr(i, 9) = GraphMatr(i, 9) - 1
        GraphMatr(i, 4) = _
            ModPropagation.HataDist(FinalOk(i, 6), _
                2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                FinalOk(i, 2), CInt(GraphMatr(i, 9)))

        Try = Try + 1 'Maximum number of tries = 3
    Else
        CheckNeighbour = True
        heavy(i, 1) = 0
        heavy(i, 2) = 0
        heavy(i, 3) = 0
        Try = 0
    End If

End If

Else
    TestAzimuth1 = TestAzimuth1 - 360
    If GraphMatr(1, 6) > TestAzimuth1 And GraphMatr(1, 6) < _
        TestAzimuth Then

        If GraphMatr(i, 9) < 4 And Try < 3 Then
            'HATA Type = HATA Type + 1
            GraphMatr(i, 9) = GraphMatr(i, 9) + 1
            GraphMatr(i, 4) = _
                ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                    FinalOk(i, 2), CInt(GraphMatr(i, 9)))

            Try = Try + 1 'Maximum number of tries = 3
        Else
            CheckNeighbour = True
            heavy(i, 1) = 0
            heavy(i, 2) = 0
            heavy(i, 3) = 0
            Try = 0
        End If
    Else
        If GraphMatr(i, 9) < 4 And Try < 3 Then
            'HATA Type = HATA Type - 1
            GraphMatr(i, 9) = GraphMatr(i, 9) - 1
            GraphMatr(i, 4) = _
                ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                    FinalOk(i, 2), CInt(GraphMatr(i, 9)))

            Try = Try + 1 'Maximum number of tries = 3
        Else
            CheckNeighbour = True
            heavy(i, 1) = 0
            heavy(i, 2) = 0
            heavy(i, 3) = 0
            Try = 0
        End If
    End If

End If

Else
    CheckMe = TestAzimuth - TestAzimuth1
    If CheckMe < 180 Then
        If GraphMatr(1, 6) > TestAzimuth1 And GraphMatr(1, 6) < _

```

```

TestAzimuth And GraphMatr(1, 7) > TestAzimuth1 And _
GraphMatr(1, 7) < TestAzimuth Then

If GraphMatr(i, 9) < 4 And Try < 3 Then
'HATA Type = HATA Type + 1
  GraphMatr(i, 9) = GraphMatr(i, 9) + 1
  GraphMatr(i, 4) = _
    ModPropagation.HataDist(FinalOk(i, 6), _
      2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
      FinalOk(i, 2), CInt(GraphMatr(i, 9)))

  Try = Try + 1 'Maximum number of tries = 3
Else
  CheckNeighbour = True
  heavy(i, 1) = 0
  heavy(i, 2) = 0
  heavy(i, 3) = 0
  Try = 0
End If
Else
If GraphMatr(i, 9) > 1 And Try < 3 Then
'HATA Type = HATA Type - 1
  GraphMatr(i, 9) = GraphMatr(i, 9) - 1
  GraphMatr(i, 4) = _
    ModPropagation.HataDist(FinalOk(i, 6), _
      2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
      FinalOk(i, 2), CInt(GraphMatr(i, 9)))

  Try = Try + 1 'Maximum number of tries = 3
Else
  CheckNeighbour = True
  heavy(i, 1) = 0
  heavy(i, 2) = 0
  heavy(i, 3) = 0
  Try = 0
End If
End If
Else
TestAzimuth = TestAzimuth - 360
If GraphMatr(1, 6) > TestAzimuth And GraphMatr(1, 6) < _
TestAzimuth1 Then

If GraphMatr(i, 9) < 4 And Try < 3 Then
'HATA Type = HATA Type + 1
  GraphMatr(i, 9) = GraphMatr(i, 9) + 1
  GraphMatr(i, 4) = _
    ModPropagation.HataDist(FinalOk(i, 6), _
      2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
      FinalOk(i, 2), CInt(GraphMatr(i, 9)))

  Try = Try + 1 'Maximum number of tries = 3
Else
  CheckNeighbour = True
  heavy(i, 1) = 0
  heavy(i, 2) = 0
  heavy(i, 3) = 0
  Try = 0
End If

Else
If GraphMatr(i, 9) < 4 And Try < 3 Then
'HATA Type = HATA Type - 1
  GraphMatr(i, 9) = GraphMatr(i, 9) - 1
  GraphMatr(i, 4) = _

```

```

ModPropagation.HataDist(FinalOk(i, 6), _
2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
FinalOk(i, 2), CInt(GraphMatr(i, 9)))

Try = Try + 1 'Maximum number of tries = 3
Else
CheckNeighbour = True
heavy(i, 1) = 0
heavy(i, 2) = 0
heavy(i, 3) = 0
Try = 0
End If

End If
End If
End If
End If

#####
' Case C(distance < R1 And distance + R0 > R1) --> 2 Cross-Section Points
'Cross-Section Point 1 : (Xp1,Yp1)
'Cross-Section Point 2 : (Xp2,Yp2)
#####

ElseIf (GraphMatr(i, 8) < GraphMatr(i, 4)) And _
((GraphMatr(i, 8) + GraphMatr(1, 4)) > GraphMatr(i, 4)) Then

Call FindCrossSection(GraphMatr(i, 3), GraphMatr(i, 2), GraphMatr(1, 3), _
GraphMatr(1, 2), GraphMatr(i, 4), GraphMatr(1, 4), i)

'Check first for X1,Y1 Cross Section Point
TestAzimuth = CLng(FindAzimuth(NeighbourCross(i, 1), _
NeighbourCross(i, 2), _
GraphMatr(1, 3), GraphMatr(1, 2)))

If GraphMatr(1, 6) < 0 Then
If TestAzimuth > GraphMatr(1, 6) + 360 Then TestAzimuth = _
TestAzimuth - 360

End If
'Check if the TestAzimuth is inside Serving Cell's Sector
If TestAzimuth > GraphMatr(1, 6) And TestAzimuth < GraphMatr(1, 7) Then
'Check now for X2,Y2 Cross Section Point
TestAzimuth1 = CLng(FindAzimuth(NeighbourCross(i, 3), _
NeighbourCross(i, 4), _
GraphMatr(1, 3), GraphMatr(1, 2)))

If GraphMatr(1, 6) < 0 Then
If TestAzimuth1 > GraphMatr(1, 6) + 360 Then TestAzimuth1 = _
TestAzimuth1 - 360

End If
'Check if the TestAzimuth1 is inside Serving Cell's Sector
If TestAzimuth1 > GraphMatr(1, 6) And TestAzimuth1 < _
GraphMatr(1, 7) Then

If Abs(TestAzimuth1 - GraphMatr(1, 5)) < Abs(TestAzimuth - _
GraphMatr(1, 5)) Then
'Two points within Sector, with the second
'closer to the Azimuth
'Store "heavy" data:

```

```

'Column 1 = Second cross section point X-Position (Xp2)
'Column 2 = Second cross section point Y-Position (Yp2)
'Column 3 = [10(-3)] / (distance + R0 - R1)
heavy(i, 1) = NeighbourCross(i, 3)
heavy(i, 2) = NeighbourCross(i, 4)
heavy(i, 3) = 0.001 / (GraphMatr(i, 8) + GraphMatr(1, 4) - _
    GraphMatr(i, 4))

CheckNeighbour = True
Else
'Two points within Sector, with the first closer to the Azimuth
'Store "heavy" data:
'Column 1 = First cross section point X-Position (Xp1)
'Column 2 = First cross section point Y-Position (Yp1)
'Column 3 = [10(-3)] / (distance + R0 - R1)
heavy(i, 1) = NeighbourCross(i, 1)
heavy(i, 2) = NeighbourCross(i, 2)
heavy(i, 3) = 0.001 / (GraphMatr(i, 8) + GraphMatr(1, 4) - _
    GraphMatr(i, 4))

CheckNeighbour = True
End If

Else
'Only the first point is within the Sector
'Store "heavy" data:
'Column 1 = First cross section point X-Position (Xp1)
'Column 2 = First cross section point Y-Position (Yp1)
'Column 3 = [8*10(-4)] / (distance + R0 - R1)
heavy(i, 1) = NeighbourCross(i, 1)
heavy(i, 2) = NeighbourCross(i, 2)
heavy(i, 3) = 0.0008 / (GraphMatr(i, 8) + GraphMatr(1, 4) - _
    GraphMatr(i, 4))

CheckNeighbour = True
End If

Else
'Check now for X2,Y2 Cross Section Point
TestAzimuth1 = CLng(FindAzimuth(NeighbourCross(i, 3), _
    NeighbourCross(i, 4), _
    GraphMatr(1, 3), GraphMatr(1, 2)))

If GraphMatr(1, 6) < 0 Then
    If TestAzimuth1 > GraphMatr(1, 6) + 360 Then TestAzimuth1 = _
        TestAzimuth1 - 360

End If
If TestAzimuth1 > GraphMatr(1, 6) And TestAzimuth1 < _
    GraphMatr(1, 7) Then

'Only the Second point within the Sector
'Store "heavy" data:
'Column 1 = Second cross section point X-Position (Xp1)
'Column 2 = Second cross section point Y-Position (Yp1)
'Column 3 = [8*10(-4)] / (distance + R0 - R1)

heavy(i, 1) = NeighbourCross(i, 3)
heavy(i, 2) = NeighbourCross(i, 4)
heavy(i, 3) = 0.0008 / (GraphMatr(i, 8) + GraphMatr(1, 4) - _
    GraphMatr(i, 4))

CheckNeighbour = True
Else

```

```

'Both Cross-Section Points are outside the sector
'The Program has to decide whether to take a bigger HATA
'or a smaller one.

If TestAzimuth < 0 Then TestAzimuth = TestAzimuth + 360
If TestAzimuth1 < 0 Then TestAzimuth1 = TestAzimuth1 + 360
CheckMe = CLng(FindAzimuth(GraphMatr(1, 3), GraphMatr(1, 2), _
    GraphMatr(i, 3), GraphMatr(i, 2)))

If TestAzimuth1 > TestAzimuth Then
    If CheckMe - 10 < (TestAzimuth + TestAzimuth1) / 2 < _
        CheckMe + 10 Then

        If GraphMatr(1, 6) > TestAzimuth And GraphMatr(1, 6) < _
            TestAzimuth1 And GraphMatr(1, 7) > TestAzimuth _
            And GraphMatr(1, 7) < TestAzimuth1 Then

            If GraphMatr(i, 9) < 4 And Try < 3 Then
                'HATA Type = HATA Type + 1
                GraphMatr(i, 9) = GraphMatr(i, 9) + 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _
                        2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                        FinalOk(i, 2), CInt(GraphMatr(i, 9)))

                Try = Try + 1 'Maximum number of tries = 3
            Else
                CheckNeighbour = True
                heavy(i, 1) = 0
                heavy(i, 2) = 0
                heavy(i, 3) = 0
                Try = 0
            End If
        Else
            If GraphMatr(i, 9) > 1 And Try < 3 Then
                'HATA Type = HATA Type - 1
                GraphMatr(i, 9) = GraphMatr(i, 9) - 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _
                        2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                        FinalOk(i, 2), CInt(GraphMatr(i, 9)))

                Try = Try + 1 'Maximum number of tries = 3
            Else
                CheckNeighbour = True
                heavy(i, 1) = 0
                heavy(i, 2) = 0
                heavy(i, 3) = 0
                Try = 0
            End If
        End If
    Else
        TestAzimuth1 = TestAzimuth1 - 360
        If GraphMatr(1, 6) > TestAzimuth1 And _
            GraphMatr(1, 6) < TestAzimuth Then

            If GraphMatr(i, 9) < 4 And Try < 3 Then
                'HATA Type = HATA Type + 1
                GraphMatr(i, 9) = GraphMatr(i, 9) + 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _

```

```

                2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                FinalOk(i, 2), CInt(GraphMatr(i, 9)))

                Try = Try + 1 'Maximum number of tries = 3
            Else
                CheckNeighbour = True
                heavy(i, 1) = 0
                heavy(i, 2) = 0
                heavy(i, 3) = 0
                Try = 0
            End If
        Else
            If GraphMatr(i, 9) < 4 And Try < 3 Then
                'HATA Type = HATA Type - 1
                GraphMatr(i, 9) = GraphMatr(i, 9) - 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                    FinalOk(i, 2), CInt(GraphMatr(i, 9)))

                Try = Try + 1 'Maximum number of tries = 3
            Else
                CheckNeighbour = True
                heavy(i, 1) = 0
                heavy(i, 2) = 0
                heavy(i, 3) = 0
                Try = 0
            End If

        End If

    End If

Else
    If CheckMe - 10 < (TestAzimuth + TestAzimuth1) / 2 < _
        CheckMe + 10 Then

        If GraphMatr(1, 6) > TestAzimuth1 And GraphMatr(1, 6) < _
            TestAzimuth And GraphMatr(1, 7) > TestAzimuth1 And _
            GraphMatr(1, 7) < TestAzimuth Then

            If GraphMatr(i, 9) < 4 And Try < 3 Then
                'HATA Type = HATA Type + 1
                GraphMatr(i, 9) = GraphMatr(i, 9) + 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                    FinalOk(i, 2), CInt(GraphMatr(i, 9)))

                Try = Try + 1 'Maximum number of tries = 3
            Else
                CheckNeighbour = True
                heavy(i, 1) = 0
                heavy(i, 2) = 0
                heavy(i, 3) = 0
                Try = 0
            End If
        Else
            If GraphMatr(i, 9) > 1 And Try < 3 Then
                'HATA Type = HATA Type - 1
                GraphMatr(i, 9) = GraphMatr(i, 9) - 1
                GraphMatr(i, 4) = _
                    ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _

```

```

        FinalOk(i, 2), CInt(GraphMatr(i, 9)))

        Try = Try + 1 'Maximum number of tries = 3
    Else
        CheckNeighbour = True
        heavy(i, 1) = 0
        heavy(i, 2) = 0
        heavy(i, 3) = 0
        Try = 0
    End If
End If
Else
    TestAzimuth = TestAzimuth - 360
    If GraphMatr(1, 6) > TestAzimuth And GraphMatr(1, 6) < _
        TestAzimuth1 Then

        If GraphMatr(i, 9) < 4 And Try < 3 Then
            'HATA Type = HATA Type + 1
            GraphMatr(i, 9) = GraphMatr(i, 9) + 1
            GraphMatr(i, 4) = _
                ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                    FinalOk(i, 2), CInt(GraphMatr(i, 9)))

            Try = Try + 1 'Maximum number of tries = 3
        Else
            CheckNeighbour = True
            heavy(i, 1) = 0
            heavy(i, 2) = 0
            heavy(i, 3) = 0
            Try = 0
        End If

    Else
        If GraphMatr(i, 9) < 4 And Try < 3 Then
            'HATA Type = HATA Type - 1
            GraphMatr(i, 9) = GraphMatr(i, 9) - 1
            GraphMatr(i, 4) = _
                ModPropagation.HataDist(FinalOk(i, 6), _
                    2, CLng(FinalOk(i, 14)), FinalOk(i, 7), _
                    FinalOk(i, 2), CInt(GraphMatr(i, 9)))

            Try = Try + 1 'Maximum number of tries = 3
        Else
            CheckNeighbour = True
            heavy(i, 1) = 0
            heavy(i, 2) = 0
            heavy(i, 3) = 0
            Try = 0
        End If

    End If
End If
End If
End If

#####
' Case D (distance + R0 < R1) --> no Cross-Section Points
#####

ElseIf (GraphMatr(i, 4) > (GraphMatr(1, 4) + GraphMatr(i, 8))) Then

```

```

'Azimuth Direction (Neighbour --> Serving Cell )
TestAzimuth = CLng(FindAzimuth(GraphMatr(1, 3), GraphMatr(1, 2), _
    GraphMatr(i, 3), GraphMatr(i, 2)))

If GraphMatr(1, 6) < 0 Then
    If TestAzimuth > GraphMatr(1, 6) + 360 Then TestAzimuth = _
        TestAzimuth - 360

End If
'Check if the TestAzimuth is inside Serving Cell's Sector
If TestAzimuth > GraphMatr(1, 6) And TestAzimuth < GraphMatr(1, 7) Then
    CheckNeighbour = True
    'Store "heavy" data:
    'Column 1 = X-Position of the point onto HATA Radius of Serving Cell
    '           which is in the direction of "TestAzimuth"
    'Column 2 = Y-Position .....
    'Column 3 = [10^(-3)] / [ R1 - distance - R0]
    heavy(i, 1) = _
        GraphMatr(1, 4) * Sin(AsRadians(TestAzimuth)) + GraphMatr(1, 3)

    heavy(i, 2) = _
        GraphMatr(1, 4) * Cos(AsRadians(TestAzimuth)) + GraphMatr(1, 2)

    heavy(i, 3) = _
        0.001 / (GraphMatr(i, 4) - GraphMatr(1, 4) - GraphMatr(i, 8))

Else
    'If possible: HATA type = HATA type + 1
    If GraphMatr(i, 9) = 4 Or Try = 3 Then
        CheckNeighbour = True
        heavy(i, 1) = 0
        heavy(i, 2) = 0
        heavy(i, 3) = 0
    Else
        Try = Try + 1 'Maximum number of tries = 3
        GraphMatr(i, 9) = GraphMatr(i, 9) + 1
        GraphMatr(i, 4) = ModPropagation.HataDist(FinalOk(i, 6), 2, _
            CLng(FinalOk(i, 14)), FinalOk(i, 7), FinalOk(i, 2), _
            CInt(GraphMatr(i, 9)))

    End If

End If

End If
End If
Loop Until (CheckNeighbour = True)
Next i

```

End Sub

```

Sub DrawGraph()
' This SubProcedure draws the results in the "frmFinal" form, in order to have
' a visual representation of the environment and the accuracy of the Algorithm.

```

```

Dim i As Integer
Dim MaxY As Variant
Dim MaxX As Variant
Dim MinY As Variant
Dim MinX As Variant
Dim MeasureX As Double
Dim MeasureY As Double
Dim Test As Double

```



```
frmFinal.PictureBox2.Cls 'Clear the PictureBox from previous drawings

'-----
' Calculate MaxX, MaxY, MinX and MinY in order to determine the picture's
' width/height and scale
'-----
MaxY = GraphMatr(1, 2) + TRUST_CIRCLE_BIG
MaxX = GraphMatr(1, 3) + TRUST_CIRCLE_BIG
MinY = GraphMatr(1, 2) - TRUST_CIRCLE_BIG
MinX = GraphMatr(1, 3) - TRUST_CIRCLE_BIG

For i = 1 To DataMax
    If GraphMatr(i, 2) + GraphMatr(i, 4) > MaxY Then
        MaxY = GraphMatr(i, 2) + GraphMatr(i, 4)
    End If
Next i

If GPS = True Then
    If GPSy > MaxY Then
        MaxY = GPSy
    End If
End If

For i = 1 To DataMax
    If GraphMatr(i, 3) + GraphMatr(i, 4) > MaxX Then
        MaxX = GraphMatr(i, 3) + GraphMatr(i, 4)
    End If
Next i

If GPS = True Then
    If GPSx > MaxX Then
        MaxX = GPSx
    End If
End If

For i = 1 To DataMax
    If GraphMatr(i, 2) - GraphMatr(i, 4) < MinY Then
        MinY = GraphMatr(i, 2) - GraphMatr(i, 4)
    End If
Next i

If GPS = True Then
    If GPSy < MinY Then
        MinY = GPSy
    End If
End If

For i = 1 To DataMax
    If GraphMatr(i, 3) - GraphMatr(i, 4) < MinX Then
        MinX = GraphMatr(i, 3) - GraphMatr(i, 4)
    End If
Next i

If GPS = True Then
    If GPSx < MinX Then
        MinX = GPSx
    End If
End If

MeasureX = MaxX - MinX
MeasureY = MaxY - MinY
```

```

frmFinal.Picture2.Width = 10000 * MeasureX / Zoom 'Set PictureBox Width
frmFinal.Picture2.Height = 10000 * MeasureY / Zoom 'Set PictureBox Height

'4KM FOR EVERY 10000PIXELS if Zoom = 4

'Set the Scale of the PictureBox accoring to MinX, MinY, MaxX, MaxY
frmFinal.Picture2.Scale (MinX - 0.03, MaxY + 0.03)-(MaxX + 0.03, MinY - 0.03)

'#####

'-----
'*****DRAW THE TRUST AREA*****
'-----

If Sector = False Then
    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = &H606060
    frmFinal.Picture2.Circle (GraphMatr(1, 3), GraphMatr(1, 2)), _
        TRUST_CIRCLE_BIG, vbBlack

    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = vbBlack
    frmFinal.Picture2.Circle (GraphMatr(1, 3), GraphMatr(1, 2)), _
        TRUST_CIRCLE_SMALL, vbBlack
Else
    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = &H606060
    frmFinal.Picture2.Circle (GraphMatr(1, 3), GraphMatr(1, 2)), _
        TRUST_CIRCLE_BIG, vbBlack, StartSector, StopSector

    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = vbBlack
    frmFinal.Picture2.Circle (GraphMatr(1, 3), GraphMatr(1, 2)), _
        TRUST_CIRCLE_SMALL, vbBlack
End If

frmFinal.Picture2.FillStyle = 1
frmFinal.Picture2.ForeColor = vbYellow
'#####

'-----
'***** DRAW SERVING CELL, NEIGHBOURS AND GPS *****
'-----
frmFinal.Picture2.DrawWidth = 2

'Serving Cell HATA Radius
frmFinal.Picture2.Circle (GraphMatr(1, 3), GraphMatr(1, 2)), _
    GraphMatr(1, 4), vbBlue

frmFinal.Picture2.FillStyle = 0
frmFinal.Picture2.FillColor = vbRed

'Serving Cell Position
frmFinal.Picture2.Circle (GraphMatr(1, 3), GraphMatr(1, 2)), 0.03, vbRed
frmFinal.Picture2.FillStyle = 1
frmFinal.Picture2.PSet (GraphMatr(1, 3), GraphMatr(1, 2))
frmFinal.Picture2.Print "Serving Cell"
frmFinal.Picture2.DrawWidth = 3

'Serving Cell Azimuth Direction
Call DrawArrow(frmFinal.Picture2, GraphMatr(1, 3), GraphMatr(1, 2), _

```

```

GraphMatr(1, 3) _
+ 0.15 * Sin(AsRadians(GraphMatr(1, 5))), GraphMatr(1, 2) + _
0.15 * Cos(AsRadians(GraphMatr(1, 5))), 0.04, True, 0, 60)

frmFinal.Picture2.DrawWidth = 1

'Draw GPS Position
If GPS = True Then
    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.FillColor = vbGreen
    frmFinal.Picture2.Circle (GPSx, GPSy), 0.015, vbGreen
    frmFinal.Picture2.FillStyle = 1
    frmFinal.Picture2.Print "GPS"
    frmFinal.Picture2.FillColor = vbRed

End If

'Draw Neighbours
For i = 2 To DataMax

    'Neighbour(i-1) HATA Radius
    frmFinal.Picture2.Circle (GraphMatr(i, 3), GraphMatr(i, 2)), _
        GraphMatr(i, 4), vbRed
    frmFinal.Picture2.FillStyle = 0
    'Neighbour(i-1) Position
    frmFinal.Picture2.Circle (GraphMatr(i, 3), GraphMatr(i, 2)), 0.03, vbRed
    frmFinal.Picture2.FillStyle = 1
    frmFinal.Picture2.PSet (GraphMatr(i, 3), GraphMatr(i, 2))
    frmFinal.Picture2.Print "Neighbour " & (i - 1)
    frmFinal.Picture2.DrawWidth = 3

    'Neighbour(i-1) Azimuth Direction
    Call DrawArrow(frmFinal.Picture2, GraphMatr(i, 3), GraphMatr(i, 2), _
        GraphMatr(i, 3) + 0.15 * Sin(AsRadians(GraphMatr(i, 5))), _
        GraphMatr(i, 2) + _
        0.15 * Cos(AsRadians(GraphMatr(i, 5))), 0.04, True, 0, 60)

    frmFinal.Picture2.DrawWidth = 1

Next i
'#####

'Draw M/T Calculated Position
frmFinal.Picture2.FillStyle = 0
frmFinal.Picture2.Circle (CalcX, CalcY), 0.03, vbRed
frmFinal.Picture2.PSet (CalcX, CalcY)
frmFinal.Picture2.Print "LOC"

'Draw "heavy()" coefficients for debugging purposes ONLY!
For i = 2 To DataMax
    frmFinal.Picture2.FillStyle = 0
    frmFinal.Picture2.Circle (heavy(i, 1), heavy(i, 2)), 0.01, vbYellow
    frmFinal.Picture2.PSet (heavy(i, 1), heavy(i, 2))
Next i

'Initialize frmFinal form and make it Visible
Call frmFinal.Initialize

frmFinal.Enabled = True
frmFinal.Visible = True

```

End Sub

```

Function FindAzimuth(x1 As Variant, y1 As Variant, x0 As Variant, y0 As Variant) As Double
'This Function Calculates the Azimuth Direction of a vector
'x0,y0 = the start of the vector
'x1,y1 = the end of the vector
'The result is the Azimuth in degrees
Dim helpthita As Double
Const pi = 3.14159265358979

    If y1 <> y0 Then
        helpthita = Atn(Abs((x1 - x0) / (y1 - y0))) * 180 / pi

        If x1 > x0 And y1 < y0 Then helpthita = 180 - helpthita
        If x1 < x0 And y1 < y0 Then helpthita = 180 + helpthita
        If x1 < x0 And y1 > y0 Then helpthita = 360 - helpthita

    Else
        If x1 > x0 Then
            helpthita = 90
        Else
            helpthita = 270
        End If

    End If

    FindAzimuth = helpthita

End Function

```

```

Sub FindCrossSection(x1 As Variant, y1 As Variant, x0 As Variant, y0 As Variant, _
    R1 As Variant, R0 As Variant, j As Integer)

'This SubProcedure calculates the Cross-Section points of two Circles
'x1,y1 = the center of the first circle
'x0,y0 = the center of the second circle
'R1 = the radius of the first circle
'R0 = the radius of the second circle
'j = the number of the NeighbourCross() Matrix Row, where the results are stored
'The Results (FinalX1,FinalY1) and (FinalX2,FinalY2) are stored in NeighbourCross()
'Matrix.
Dim X As Double
Dim Y As Double
Dim alpha As Double
Dim Bita As Double
Dim delta As Double
Dim FinalX1 As Double
Dim FinalX2 As Double
Dim FinalY1 As Double
Dim FinalY2 As Double

    X = x1 - x0
    Y = y1 - y0
    alpha = (R0 ^ 2 - R1 ^ 2 + X ^ 2 + Y ^ 2) / (2 * Y)
    Bita = X / Y
    delta = (4 * (alpha ^ 2) * (Bita ^ 2)) - 4 * ((1 + Bita ^ 2) * _
        (alpha ^ 2 - R0 ^ 2))

    FinalX1 = (2 * alpha * Bita + delta ^ (1 / 2)) / (2 * (1 + Bita ^ 2))
    FinalY1 = alpha - FinalX1 * Bita
    FinalX2 = (2 * alpha * Bita - delta ^ (1 / 2)) / (2 * (1 + Bita ^ 2))

```

```

FinalY2 = alpha - FinalX2 * Bita

NeighbourCross(j, 1) = FinalX1 + x0
NeighbourCross(j, 2) = FinalY1 + y0
NeighbourCross(j, 3) = FinalX2 + x0
NeighbourCross(j, 4) = FinalY2 + y0

```

End Sub

```

Sub DrawArrow(ByVal pic As Object, ByVal x1 As _
Single, ByVal y1 As Single, ByVal x2 As Single, ByVal _
y2 As Single, ByVal length As Single, Optional ByVal _
draw_head As Boolean = True, Optional ByVal _
num_fletchings As Integer = 0, Optional ByVal _
fletching_gap As Single = 60)

' This SubProcedure draws an Arrow in the "frmFinal" PictureBox.
' It is used by the "DrawGraph" SubProcedure and it is used to draw
' the Azimuth direction of the BTS Antenna

Dim vx As Single
Dim vy As Single
Dim dist As Single
Dim ax As Single
Dim ay As Single
Dim barb_scale As Single
Dim i As Integer
Dim gap_dx As Single
Dim gap_dy As Single

' Draw the shaft.
pic.Line (x1, y1)-(x2, y2)

' Find the arrow shaft unit vector.
vx = x2 - x1
vy = y2 - y1
dist = Sqr(vx * vx + vy * vy)
vx = vx / dist
vy = vy / dist

' See if we need to draw multiple fletchings.
If num_fletchings > 1 Then
' Get the fletching spacing vector.
gap_dx = vx * fletching_gap
gap_dy = vy * fletching_gap
End If

' Draw the right barb.
ax = -vy - vx
ay = vx - vy

' Set the proper length.
ax = ax * length
ay = ay * length
If draw_head Then pic.Line (x2, y2)-Step(ax, ay)
For i = 0 To num_fletchings - 1
pic.Line (x1 + i * gap_dx, y1 + i * _
gap_dy)-Step(ax, ay)
Next i

' Find the left barb.
ax = (vy - vx) * length

```

```
ay = (-vx - vy) * length
If draw_head Then pic.Line (x2, y2)-Step(ax, ay)
For i = 0 To num_fletchings - 1
    pic.Line (x1 + i * gap_dx, y1 + i * _
        gap_dy)-Step(ax, ay)
Next i
```

End Sub

```
Sub FillMatrData()
' This Sub Procedure is Called when the user presses the "Show Results Table"
' button in the "frmFinal" form. "frmSplash2" form is activated.
Dim i As Integer
Dim j As Integer

frmSplash2.Enabled = True
frmSplash2.Visible = True

For i = 1 To DataMax
    frmSplash2.Label10(i).Caption = CStr(FinalOk(i, 1)) 'Cell ID/Channel Nr
    frmSplash2.Label4(i).Caption = CStr(FinalOk(i, 2)) 'RXLev
    frmSplash2.Label11(i).Caption = CStr(FinalOk(i, 3)) 'Cell ID
    frmSplash2.Label12(i).Caption = CStr(FinalOk(i, 4)) 'Latitude
    frmSplash2.Label13(i).Caption = CStr(FinalOk(i, 5)) 'Longitude
    frmSplash2.Label15(i).Caption = CStr(FinalOk(i, 6)) 'Antenna Height
    frmSplash2.Label16(i).Caption = CStr(FinalOk(i, 7)) 'BTS Tx Power
    frmSplash2.Label18(i).Caption = CStr(FinalOk(i, 8)) 'Hata Radius
Next i
```

End Sub

```
Function AsRadians(ByVal pDb_Degrees As Double) As Double

'To convert decimal degrees to radians, multiply
'the number of degrees by pi/180 = 0.017453293 radians/degree
Const vbPi = 3.14159265358979
AsRadians = pDb_Degrees * (vbPi / 180)
```

End Function

A.4. INDEX

A

ArcLengthOfMeridian(function), UTMWGS84, **181**
AsRadians(function), ModSub, **222**

C

CalcDistances, ModSub, **201-205**
CheckCoSite(function), ModSub, **195**
CheckNeighbourHata, ModSub, **205-216**
CheckPorts(function),
 frmModeFull, **157**
 frmModeNormal, **146**

Clos,
 frmFinal, **167**
 frmSplash2, **178**

Command1_Click,
 frmBrowse, **171**
 frmFinal, **167**
 frmModeFull, **159-161**
 frmModeNormal, **148-149**
 frmModeUser, **153-154**
 frmSavePicture, **173**
 frmSplash2, **178**
 frmStart, **143**

Command2_Click,
 frmBrowse, **171**
 frmFinal, **168**
 frmModeFull, **156**
 frmSavePicture, **173**
 frmStart, **143**

Command3_Click,
 frmFinal, **168**
 frmStart, **143**

D

Databasefind, ModSub, **190-194**
DegToRad(function), UTMWGS84, **181**
dirList_Change,
 frmBrowse, **172**
 frmSavePicture, **174**
Dothings, frmModeFull, **165-166**
DrawArrow, ModSub, **221-222**
DrawBar, frmModeFull, **163-164**
DrawGraph, ModSub, **216-219**
DrawMicro, ModSub, **198-200**
drvList_Change,
 frmBrowse, **171-172**
 frmSavePicture, **173-174**

F

fillList_DbIclick,
 frmBrowse, **172**
 frmSavePicture, **174**
FillMatrData, ModSub, **222**
FindAzimuth(function), ModSub, **220**
FindCrossSection, ModSub, **220-221**
FindWhere(function), ModSub, **187-188**
FixPositions, frmFinal, **169-170**
FootpointLatitude(function), UTMWGS84, **182**
Form_Load,
 frmBrowse, **172**
 frmModeFull, **156**
 frmModeNormal, **144**
 frmModeUser, **152**
 frmSavePicture, **174**
Form_Resize, frmFinal, **169**
Form_Unload,
 frmFinal, **168**
 frmGPS, **175**
 frmModeFull, **163**
 frmModeNormal, **151**
 frmModeUser, **154**

G

GetFinalMatrix, ModSub, **195**
GetLatLon(function), UTMWGS84, **180**
GetUTM(function), UTMWGS84, **180**

H

HataDist(function), ModPropagation, **179**
HScroll1_Change, frmFinal, **168**

I

Initialize,
 frmFinal, **169**
 frmGPS, **176**

L

LatLonToUTMXY(function), UTMWGS84, **185-186**
Log10(function), ModPropagation, **179**

M

MapLatLonToXY(function), UTMWGS84, **183-184**
MapXYToLatLon(function), UTMWGS84, **184-185**

MicroCheck(function), ModSub, **195-196**

mnuDataLink_Click,
frmModeFull, **156**
frmModeNormal, **145**
frmModeUser, **152**

mnuFBUScom1_Click,
frmModeFull, **156**
frmModeNormal, **145**

mnuFBUScom2_Click,
frmModeFull, **156**
frmModeNormal, **145**

mnuGPSEnabled_Click,
frmModeFull, **157-158**
frmModeNormal, **146**

mnuGPSNMEA15_Click,
frmModeFull, **157**
frmModeNormal, **145**

mnuGPSNMEA21_Click,
frmModeFull, **156**
frmModeNormal, **145**

mnuGPSport1_Click,
frmModeFull, **157**
frmModeNormal, **145**

mnuGPSport2_Click,
frmModeFull, **157**
frmModeNormal, **145**

mnuHelpAbout_Click,
frmModeFull, **158**
frmModeNormal, **146**
frmModeUser, **152**

mnuPrgExit_Click,
frmModeFull, **158**
frmModeNormal, **147**
frmModeUser, **153**

mnuPrgMode_Click,
frmModeFull, **158**
frmModeNormal, **147**
frmModeUser, **154**

mnuViewDebug_Click, frmModeNormal, **147**

MSComm_OnComm, frmGPS, **175-176**

N

NotLocate, ModSub, **196-198**

P

Picture1_DbClick, frmFinal, **168**

Picture2_Click, frmFinal, **168-169**

ProcessData, ModSub, **188-190**

ProcessDist, ModSub, **194-195**

ProcessSentence, frmGPS, **176**

R

RadToDeg(function), UTMWGS84, **181**

S

Slider1_Click,
frmModeFull, **158-159**
frmModeNormal, **147**

Slider2_Click,
frmModeFull, **159**
frmModeNormal, **148**

SortMatrix, ModSub, **187**

StoreResults, frmModeFull, **164-165**

StoreValues, ModSub, **190**

Swap, ModSub, **188**

T

TAHATAMatch, ModSub, **200**

Timer1_Timer,
frmGPS, **176-177**
frmModeFull, **161 – 162**
frmModeNormal, **149 – 151**

U

UTMCentralMeridian(function), UTMWGS84, **182**

UTMXYToLatLon(function), UTMWGS84, **186**

V

VScroll1_Change, frmFinal, **169**



ΠΑΡΑΡΤΗΜΑ Β

ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ (USER'S GUIDE)



ΠΑΡΑΡΤΗΜΑ Β

| | |
|---|------------|
| INTRODUCTION | 229 |
| B.1. INSTALLATION | 231 |
| SYSTEM REQUIREMENTS | 231 |
| PROGRAM SETUP | 231 |
| B.2. USING "LOCATION SERVER" | 233 |
| B.2.1. WELCOME SCREEN | 233 |
| B.2.2. "NORMAL" MODE | 234 |
| B.2.2.1. MENU BAR | 235 |
| B.2.2.2. EXECUTING THE LOCATION ALGORITHM | 238 |
| B.2.3. "USER" MODE | 239 |
| B.2.4. "FULL" MODE | 241 |
| B.2.4.1. EXECUTING THE LOCATION ALGORITHM | 242 |
| B.2.4.2. RESULTS REVIEW | 243 |
| B.2.5. "POSITION GRAPH" WINDOW | 244 |
| B.3. INDEX | 247 |

This page is left blank on purpose

INTRODUCTION

This is an implementation of a Mobile Location Technique. It takes advantage of three different Location Technologies (CID, TA, RXLEV) and generates an estimated position of the Mobile Terminal in Longitude/Latitude format. In addition, it calculates a TRUST Area as supplementary aid for locating the Mobile Terminal.

The purpose of this program is to investigate the accuracy of "Network Based" Location Technologies and the possibility of applying Location Based Services (LBS) in Commercial Mobile Networks, using such methods. Numerous field tests have been conducted using the specific software with mostly successful results.

The program was created using Microsoft[®] Visual Studio 6. The code is written in Visual Basic 6 and writing took almost one year.

Commercial distribution and exploitation of this software is strictly prohibited. You may not reverse, engineer, decompile, or disassemble this software.

The present user's guide describes the main functions of the program, as well as S/W and H/W requirements. Since the program is only for test purposes, no Help files, Troubleshooting guides, FAQ is available. For any possible questions and suggestions please contact me (atzoumas@intraconnect.biz).

This page is left blank on purpose

B.1. INSTALLATION

SYSTEM REQUIREMENTS

In order to install and run the "Location Server" successfully, the computer system must meet with the following requirements:


Hardware Requirements:

- ◆ Intel® Pentium 4/III/II Celeron, AMD® K7/K6
- ◆ VGA/Super VGA monitor, supporting minimum 1024x768 resolution
- ◆ 32 MB of RAM or higher is suggested
- ◆ CD-ROM Drive
- ◆ Nokia 61xx Mobile Phone (Recommended: Nokia 6150)
- ◆ Nokia FBUS Serial Cable
- ◆ GPS unit, NMEA v1.5(APA/XTE)/v2.1(GSA) compatible
- ◆ Serial cable (for GPS connection)

Software Requirements:

- ◆ Microsoft® Windows 9x/ME/2000/XP
- ◆ Microsoft® Office 2000/XP
- ◆ Visual Basic Runtime Library 6

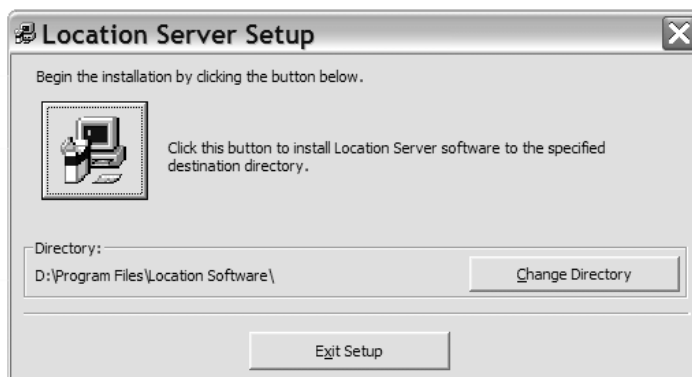
PROGRAM SETUP

1. Insert the "Location Server" CD into your CD-ROM drive.
2. Find the "Setup.exe" file and double-click.  Setup.exe
Setup Bootstrap for Visual Basic Set...
Microsoft Corporation
3. The Screen of Fig.1.1 will appear on your screen. Click on "OK" to continue.



–Figure 1.1 (setup first screen)--

4. The Main Setup Window will appear (Fig.1.2).



–Figure 1.2 (setup main screen)--

By default, the installation directory is:

"WINDRIVE\Program Files\Location Software"
(WINDRIVE is the drive where Windows Operating System is installed)

If you want to change the Installation directory click on "Change Directory" and select the desired directory name.

When done, click on  to continue.

5. In the next Screen(Fig.1.3) you can select the "Program Group" name:



–Figure 1.3 (Program Group screen)--

By default the program group name is "LBS". You can change it, or select an Existing Group from the list below. When finished, click on "Continue" to start copying the Program Files to your Computer.



–Figure 1.4 (Setup final screen)–

- When the setup program finishes copying all files, the Screen of Fig.1.4 will appear. Click on "OK" to complete the installation successfully!

B.2. USING "LOCATION SERVER"

Go to the Program Group, where you installed the program (Default: LBS) and click on the "Location Server" (Start → Programs → Program Group → Click -- Location Server).

Create Desktop Shortcut:

You may create a shortcut on your Desktop for quicker access of the program. Open Windows Explorer, go to the directory where you installed the program (eg. "C:\Program Files\Location Software\") and right-click on "Location Server.exe".

Click: Send To → Desktop(create shortcut)

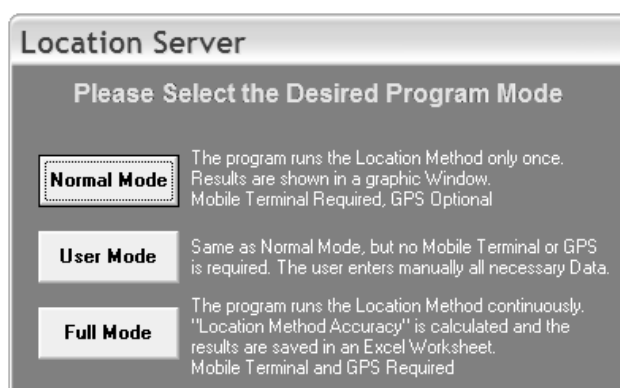
Now you will have the following icon on your Desktop:



Location Server

To run the program, just double-click on this Icon.

B.2.1. WELCOME SCREEN



–Figure 2.1 (Welcome Screen Window)–

This is the first screen you see when you execute the program. As shown above, the program can run in three different modes:

1. Normal Mode

The program runs the Location Algorithm only once and the result is displayed in a graphic window. In Normal Mode, the Mobile Terminal (M/T) is required in order generate a position successfully. GPS is optional.

2. User Mode

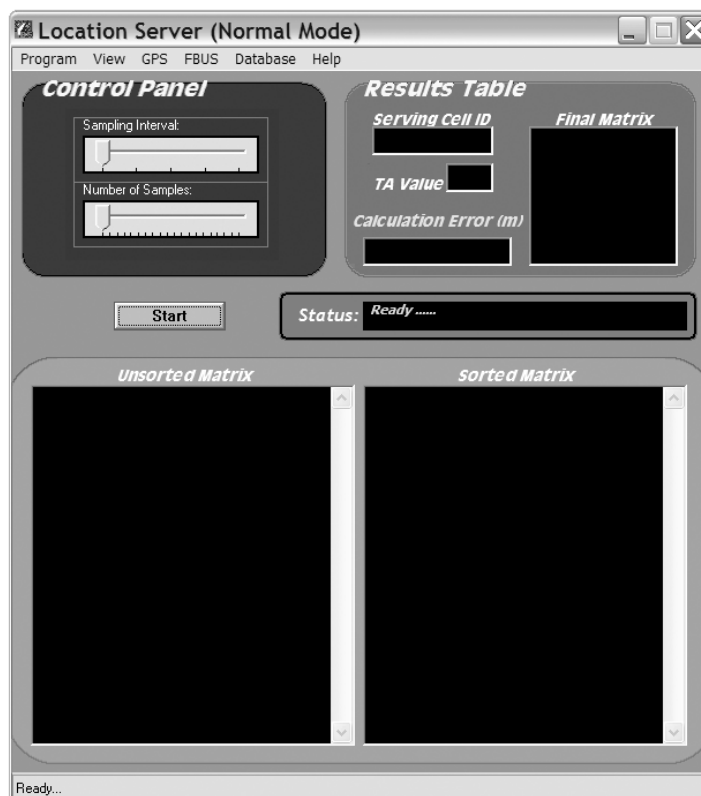
No Mobile Terminal (M/T) or GPS is required here. The user enters all necessary DATA manually and the program generates a location according to these DATA.

3. Full Mode

The program runs the Location Algorithm continuously, until the "Stop" button is pressed. The results are saved in an Excel Worksheet. In this mode, the Mobile Terminal (M/T) is required in order generate a position successfully. GPS is optional.

By clicking on one of these three buttons, the program enters in the respective mode.

B.2.2. "NORMAL" MODE



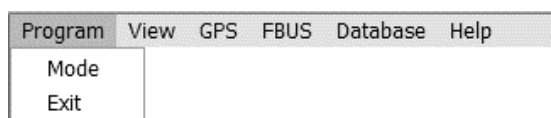
—Figure 2.2 (Normal Mode Window)—

B.2.2.1. MENU BAR



The menu bar is located at the top of the program window and it consists of the following options:

- **Program**

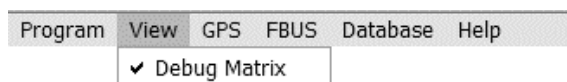


↻ **Mode:** change between program modes

If you click on Program → Mode, the Welcome Screen will appear, allowing you to enter in another program mode.

↻ **Exit:** exits the "Location Server" program.

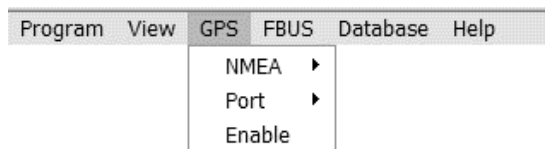
- **View**



↻ **Debug Matrix:** show/hide the Debug window

If the option is checked, the debug window is visible (default). If you want to hide the debug window, just click on the "Debug Matrix" option and the debug window will disappear.

- **GPS**



↻ **NMEA:** select the available NMEA compatibility.

Depending on the available GPS, two versions of NMEA are supported:

- Ver. 1.5 (APA/XTE)
- Ver. 2.1 (GSA) (default)

Click on the respective option to select the desired Version of NMEA.

↻ **Port:** select the COM port, where the GPS is connected.

Here you can select between two COM Ports:

- COM 1
- COM 2

Click on the respective field to select the desired Port.

The program needs one more COM port for connecting to the Mobile Terminal. If there is only one COM port available (eg. COM1) then select this port for both devices. When you start the Location Algorithm, the program will ask you to switch between devices when needed.

➤ **Enable:** enable GPS support.

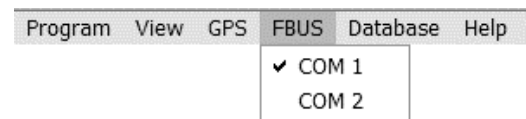
When you click on this option, GPS support is enabled (GPS Window becomes visible, Fig.2.3) and the program can read from the GPS the generated position (Lon/Lat in the GPS window).

➤ **Disable:** disable GPS support.

If GPS Window is visible and you want to disable GPS support, click on this option.



—Figure 2.3 (GPS Window)—

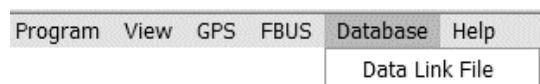


- **FBUS :** select the port where the Mobile Terminal is connected to.

➤ **COM 1:** click to this option if the Mobile Terminal is connected (via FBUS cable) to COM1 port (default).

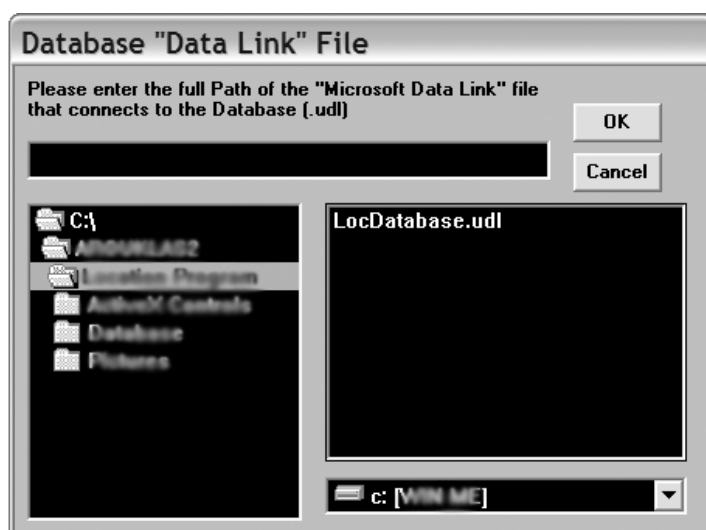
➤ **COM 2:** click to this option if the Mobile Terminal is connected (via FBUS cable) to COM2 port.

The program needs one more COM port for connecting to the GPS. If there is only one COM port available (eg. COM1) then select this port for both devices. When you start the Location Algorithm, the program will ask you to switch between devices when needed.



- **Database**

➤ **Data Link File:** click to select the "Data Link File" Location. Once you click this option, the window of Fig.2.4 appears.



—Figure 2.4 (Browse Window)—

Browse your hard disk to find the "Data Link File" and press "OK".

If no "Data Link File" is available, you can create your own, only under the condition that the Database containing the CELL INFORMATION exists.

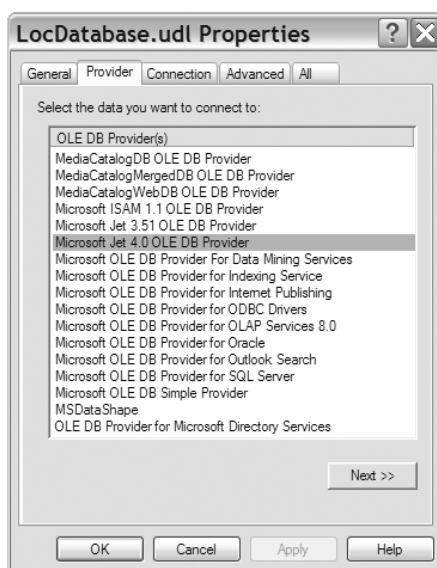
Creating a "Data Link File":

1. While you are on the Desktop, right-click → New → Text Document.
2. Name the text file "LocDatabase" and then change its extension to ".udl".

3. The following icon shall appear on your Desktop:



4. Right-Click on this icon and click on Properties. You will then see the following window:



Select the "Provider" control and click on:

Microsoft Jet 4.0 OLE DB Provider

Select the "Connection" control and enter:

- The "MS Access Database" address.
(eg. C:\Program Files\Location Software\LocDatabase.mdb)
- User name: Admin
- Password: blank

In "Advanced", check only the following boxes:

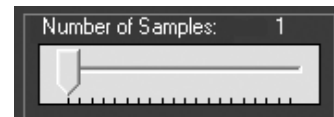
"ReadWrite" and "Share Deny None".

5. Move the "LocDatabase.udl" file, from your Desktop to the program directory.
6. Go to the "Location Server" program, click on the "Data Link File" option and enter the location of your newly created ".udl" file.

B.2.2.2. EXECUTING THE LOCATION ALGORITHM

- First of all, you have to adjust the following parameters:

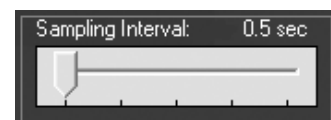
→ Number of Samples:



Select how many samples you want the program to take. If you select 1 sample (default value), the program, via the M/T, will take only one sample of the "Location Critical DATA" (TA, C-ID, RXLev) and then try to generate a position according to this single sample. If you select more, the program will take more samples, add them and generate a more trustworthy position.

Therefore, it is recommended to select more than one samples (3 ~ 10). The maximum number of samples you can select is 20.

→ Sampling Interval:



Select the interval between samples. The available values for this parameter are:

0.5sec (default) -- **1sec** -- **2sec** -- **3sec** -- **4sec**

It is strongly recommended to select a value between 0.5sec and 2sec. Greater values can result in a big "Location Algorithm" time-lag.

After adjusting the above Parameters, you are ready to execute the "Location Algorithm".

- Make sure that the Mobile Terminal and the GPS (Optional) are properly connected to the Computer and that you have assigned them to the correct COM ports.
- If everything seems ok, click on the **Start** button.
- If you have selected the same COM port for the GPS and the M/T, then the program will guide you which device to connect first. Follow the instructions on screen.
- The program will calculate the Mobile Terminal's estimated position and the "Position Graph" window will appear on screen.
*For more information on the "Position Graph" window see **PARAGRAPH 2.5**.*
- In the "Normal Mode" window the following information will appear:
 - ➔ **Serving Cell ID:** The Cell Identifier of the Serving Cell
 - ➔ **TA Value:** Measured value of Timing Advance
 - ➔ **Final Matrix:** The matrix containing Serving Cell and Neighbours information. In the first column you can see the Channel Numbers and in the second column, the Received Power Levels (in dBm). Each Row corresponds to a single BTS. (Row1→Serving Cell, Row2→Neighbour1, etc.)
 - ➔ **Calculation Error (m):** The distance of the estimated position from the GPS generated position.

B.2.3. "USER" MODE

| Serving CELL ID | Rx Level (dBm) | Area Type |
|--------------------|----------------|-----------|
| Neighbour1 CHANNEL | Rx Level (dBm) | Area Type |
| Neighbour2 CHANNEL | Rx Level (dBm) | Area Type |
| Neighbour3 CHANNEL | Rx Level (dBm) | Area Type |
| Neighbour4 CHANNEL | Rx Level (dBm) | Area Type |
| Neighbour5 CHANNEL | Rx Level (dBm) | Area Type |
| Neighbour6 CHANNEL | Rx Level (dBm) | Area Type |

GPS Longitude GPS Latitude TA Value

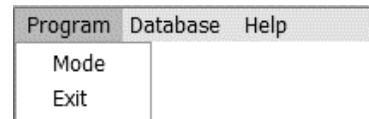
Start

Ready...

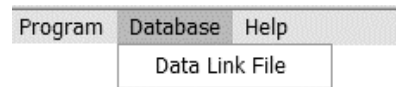
—Figure 2.5 (User Mode Window)—

- Menu Bar Options have already been described in "Normal Mode":

- Menu bar → Program (Mode/Exit) : see Page 10



- Menu bar → Database (Data Link File): see Page 12



- Before the Location Algorithm begins, you have to fill the text boxes with all the required DATA:

Serving Cell ID:

Enter the Cell identifier of the Serving Cell (only the numeral part, eg. if full Cell-ID is G100 → enter in text box: 100)

Neighbour(x) CHANNEL (x: 1, 2, 3, 4, 5, 6):

Enter the Neighbour's Channel number. If there is information for less than 6 neighbours, then in the first unavailable Neighbour's text box enter 0.

eg. If available Neighbours = 3, enter:

Neighbour4 CHANNEL = 0

Neighbour5 CHANNEL & Neighbour6 CHANNEL won't be included by the program even if they contain valid channel numbers.

Rx Level (dBm):

Enter the Received Power Level for each BTS (Serving Cell & Neighbours), in the respective text box.

Area Type:

Enter the default Hata type for each BTS.

Hata type = 1: Open Area

Hata type = 2: Suburban Area

Hata type = 3: Medium-Small City

Hata type = 4: Large City

TA Value:

Enter the Timing Advance value.

GPS Longitude:

Enter the real GPS position Longitude, in decimal format.

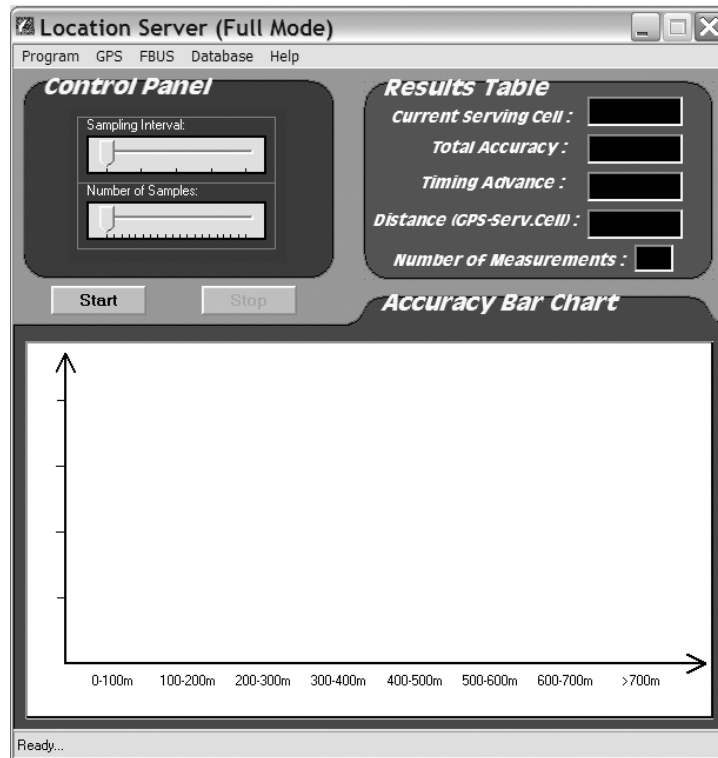
GPS Latitude:

Enter the real GPS position Latitude, in decimal format.

- If everything seems ok, click on the  button.

- The program will calculate the Mobile Terminal's estimated position and the "Position Graph" window will now appear on screen.
For more information on the "Position Graph" window see **PARAGRAPH 2.5**.

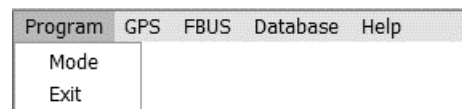
B.2.4. "FULL" MODE



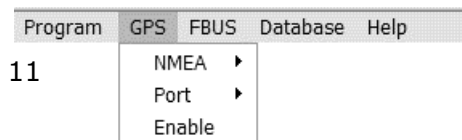
–Figure 2.5 (Full Mode Window)–

- Menu Bar Options have already been described in "Normal Mode":

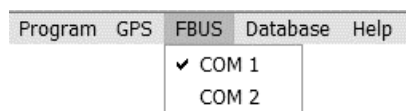
➤ Menu bar → Program (Mode/Exit) : see Page 10



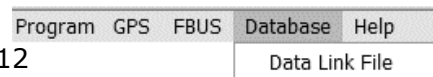
➤ Menu bar → GPS (NMEA/Port/Enable): see Page 11



- ➔ Menu bar → NMEA (COM 1/ COM 2) : see Page 12



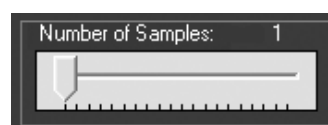
- ➔ Menu bar → Database (Data Link File): see Page 12



B.2.4.1. EXECUTING THE LOCATION ALGORITHM

- First of all, you have to adjust the following parameters:

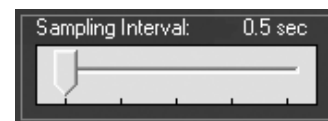
- ➔ Number of Samples:



Select how many samples you want the program to take. If you select 1 sample (default value), the program, via the M/T, will take only one sample of the "Location Critical DATA" (TA, C-ID, RXLev) and then try to generate a position according to this single sample. If you select more, the program will take more samples, add them and generate a more trustworthy position.

Therefore, it is recommended to select more than one samples (3 ~ 10). The maximum number of samples you can select is 20.

- ➔ Sampling Interval:



Select the interval between samples. The available values for this parameter are:


0.5sec (default) -- **1sec** -- **2sec** -- **3sec** -- **4sec**

It is strongly recommended to select a value between 0.5sec and 2sec. Greater values can result in a big "Location Algorithm" time-lag.

After adjusting the above Parameters, you are ready to execute the "Location Algorithm".

- Make sure that the Mobile Terminal and the GPS (Optional) are properly connected to the Computer and that you have assigned them to the correct COM ports.
- If everything seems ok, click on the **Start** button.
- The program will start calculating the Mobile Terminal's estimated position, continuously, until you click on the **Stop** button.

While the program is running you can see the following information on the "Full Mode" window (Fig.2.5):

- **Current Serving Cell:**
The Cell Identifier of the previous measurement (only the numeral part).
 - **Total Accuracy:**
It is referring to the accuracy of the M/T positioning method, adding all Measurements (mean value).
 - **Timing Advance:**
The Timing Advance value of the previous measurement
 - **Distance (GPS-Serv. Cell):**
This is the distance between the Serving Cell's location and the GPS's generated position.
 - **Number of Measurements:**
Shows how many measurements of the M/T's location have been performed.
- To stop the program click on the  button.

B.2.4.2. RESULTS REVIEW

When you stop the program, the measurement results are saved as an Excel Worksheet for later reviewing and algorithm debugging.

The Microsoft Excel file is saved into the application path and its filename contains the time and date of the previous measurement.

eg. C:\AppPath\15_04_2003__13_45_30.xls

Each measurement occupies one row of the Worksheet:

Measurement 1 → Row 2

Measurement 2 → Row 3

etc.

The contents of each Column are explained in the first row:

Columns A → P: Measured DATA

Columns A,B : Serving Cell Channel, RXLev

Columns C,D: Neighbour 1 Channel, RXLev

.....

Columns M,N: Neighbour 6 Channel, RXLev

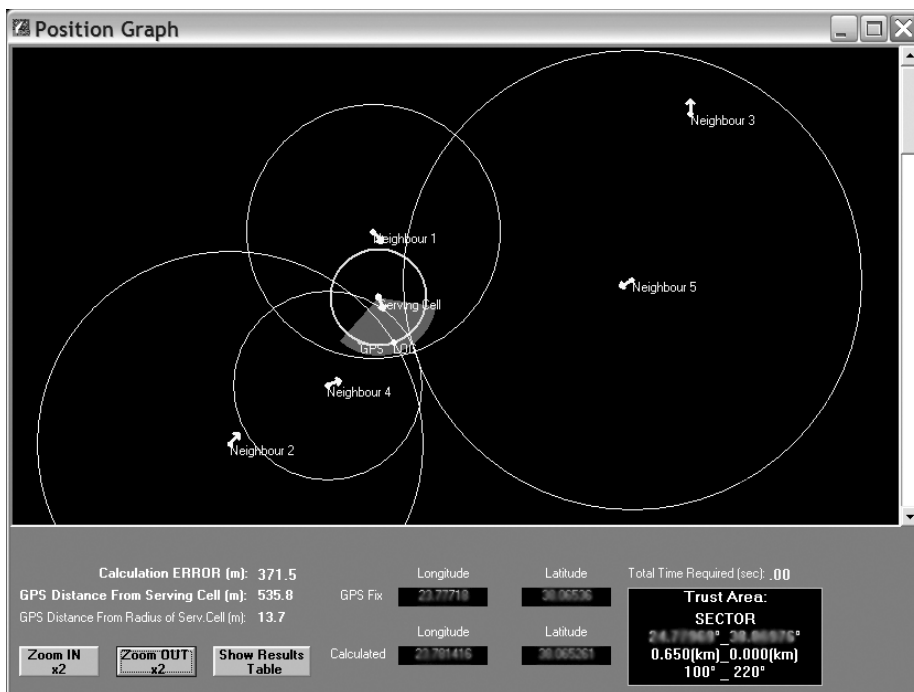
Column O: Serving Cell ID

Column P: Timing Advance

Columns Q → AC: Calculated DATA

- Columns Q,R : Estimated Position Longitude, Latitude
- Columns S,T: GPS generated position Longitude, Latitude
- Column U: Calculation ERROR
- Column V: Distance between GPS and Serving Cell
- Columns W, X ,Y ,Z ,AA ,AB ,AC: Trust Area info

B.2.5. "POSITION GRAPH" WINDOW



—Figure 2.6 (Position Graph Window)—

This window appears in "Normal" and "User" modes, after the location procedure has finished.

Information shown in this window:

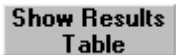
- ☞ **GPS Longitude / Latitude**
- ☞ **Calculated Position Longitude / Latitude**
- ☞ **Trust Area information**



- ☞ **Calculation Error (m)**
- ☞ **GPS Distance from Serving Cell (m)**
- ☞ **GPS Distance from HATA Radius of Serving Cell (m)**

Picture's Shapes and Colours:

- ☞ **Red points:** BTS positions (or the final estimated position of M/T).
Next to each point, the type of the BTS is written (Serving Cell / Neighbour (x))
The **LOC** point corresponds to the M/T calculated position.
- ☞ **Green point:** GPS position
- ☞ **Yellow arrows:** Antenna azimuth direction
- ☞ **Blue circle:** Serving Cell's HATA Radius
- ☞ **Red circles:** Neighbour's (x) HATA Radius

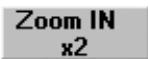
Click on  button if you want to view the Results Table window (Fig.2.7)

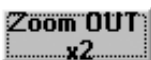
| Cell Type and CHN | CELL ID | Latitude | Longitude | RxLev | Ant_H | TxLev | HATA(KM) |
|-----------------------|---------|-----------|-----------|-------|-------|-------|------------|
| Serving Cell 39940 | 029942 | 38.69376 | 24.77969 | -76 | 8.5 | 35.8 | 0.52208569 |
| Neighbour 1 95 | 029952 | 38.676399 | 24.779262 | -87 | 18 | 36.5 | 1.39023672 |
| Neighbour 2 75 | 029947 | 38.69373 | 24.79864 | -90 | 24 | 38.14 | 2.10813293 |
| Neighbour 3 95 | 029950 | 38.69334 | 24.81932 | -91 | 15 | 38.14 | 1.83358673 |
| Neighbour 4 95 | 027411 | 38.69131 | 24.77962 | -94 | 13.5 | 36.5 | 1.03619148 |
| Neighbour 5 95 | 029623 | 38.67907 | 24.8114 | -95 | 18 | 37.92 | 2.51141629 |
| Neighbour 6 | | | | | | | |

—Figure 2.7 (Results Table Window)—

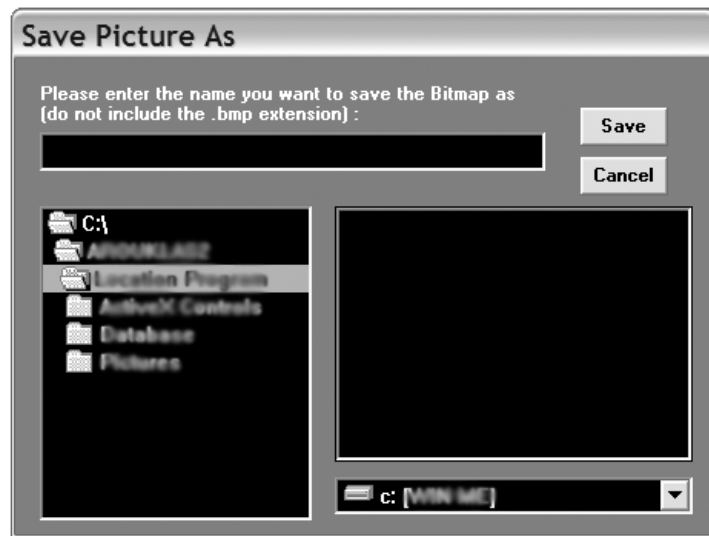
To close this window, click on "Hide" button.

Viewing / Saving the Picture

☞ Press the  button to get a closer look of the Picture.

☞ Press the  button to shrink the Picture.

- ➔ Double-Click anywhere on the Picture to save it as a bitmap image. Once you double-click, the following window will appear:



23

Select the picture's desired name and path and click on the "Save" button.

B.3. INDEX

C

create,
 Data Link File, 237
 desktop shortcut, 233
change Mode, 233
colours,
 Position Graph, 245
close Results table, 245

D

data Link File, 237
disable GPS, 236

E

enable GPS, 236

F

FBUS port, 236
Full mode, 241

G

GPS support,
 enable, 236
 disable, 236

I

information,
 Full mode, 242
 Normal mode, 238
 Position Graph, 244
install software, 231

M

menu,
 bar,
 full mode, 241
 normal mode, 235
 user mode, 240
 database, 236
 FBUS, 236

GPS, 235
 program, 235
 view, 235
mode,
 Full, 241
 Normal, 234
 User, 239

N

Normal mode, 234

P

picture,
 view, 245
 zoom,
 in, 245
 out, 245
 save, 246
Position Graph window, 244

R

requirements,
 hardware, 231
 software, 231
Results Table,
 show, 245
 close, 245

S

save Picture, 246
show Results Table, 245

T

Trust Area, 244

U

User mode, 239

Z

zoom picture, 245

