



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής
στίγματος εξ αναμετρήσεως**

Βασίλειος Δ. Ξούρης

Επιβλέπων: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΜΑΪΟΣ 2009

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

Βασίλειος Δ. Ξούρης

A.M.: 1115200200193

ΕΠΙΒΛΕΠΩΝ:

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία γίνεται μια επισκόπηση των κυριότερων και πιο εξελιγμένων τεχνικών που χρησιμοποιούνται στις μέρες μας στον τομέα της πλοήγησης εσωτερικών χώρων. Παρουσιάζονται και αναλύονται μεθόδοι που χρησιμοποιούνται για την ανάπτυξη συστημάτων προσδιορισμού θέσης με την χρήση της τεχνικής στίγματος εξ αναμετρήσεως και δίνεται ιδιαίτερη προσοχή στην περίπτωση που αφορά πεζό χρήστη. Στην συνέχεια εξετάζουμε διάφορους τρόπους ανίχνευσης βήματος του χρήστη καθώς και μεθόδους φιλτραρίσματος των δεδομένων που λαμβάνονται από τους αισθητήρες. Τέλος, αναπτύσσεται και περιγράφεται λεπτομερώς ένα αυτόνομο σύστημα προσδιορισμού θέσης με την τεχνική στίγματος εξ αναμετρήσεως.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Υπηρεσίες θέσης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: τεχνική στίγματος εξ αναμετρήσεως, πλοήγηση, ανίχνευση βήματος, προσδιορισμός θέσης, εσωτερικός χώρος

ABSTRACT

This thesis is an overview of the most advanced techniques that are used nowadays in the field of indoor space navigation. Methods used in the development of position estimation systems based on dead reckoning technique are presented and analyzed while giving further attention to pedestrian dead reckoning methods. Thereafter, we examine several techniques for step detection and sensor data filtering. Finally, an autonomous dead reckoning system for position estimation is developed and analyzed in detail.

SUBJECT AREA: Location-based services

KEYWORDS: pedestrian dead reckoning, navigation, step detection, position estimation, indoor space

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ	7
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	8
ΚΕΦΑΛΑΙΟ 2: ΑΝΙΧΝΕΥΣΗ ΒΗΜΑΤΟΣ.....	12
2.1 Μέθοδος ανίχνευσης κορυφών	14
2.2 Μέθοδος ανίχνευσης μηδενικών διασταυρώσεων (zero crossing method)	17
2.3 Μέθοδος Pan-Tompkins.....	18
2.4 Template-matching μέθοδος	21
2.5 Μέθοδος ανίχνευσης κορυφών βασισμένη σε συνδιασμένο σήμα 2 αξόνων	25
ΚΕΦΑΛΑΙΟ 3: ΦΙΛΤΡΑΡΙΣΜΑ ΔΕΔΟΜΕΝΩΝ ΕΠΙΤΑΧΥΝΣΗΣ	28
3.1 Φιλτράρισμα Μέσου όρου	28
3.2 Το φίλτρο Kalman	29
ΚΕΦΑΛΑΙΟ 4: ΥΠΟΛΟΓΙΣΜΟΣ ΚΑΤΕΥΘΥΝΣΗΣ	36
4.1 Φιλτράρισμα κατεύθυνσης με κατωπερατό φίλτρο	36
4.2 Δυναμική διόρθωση κατεύθυνσης.....	36
4.3 Στατική διόρθωση κατεύθυνσης	37
4.4 Απόκλιση από τον πραγματικό Βορρά	39
4.5 Πλεονεκτήματα διόρθωσης κατευθύνσεως	40
ΚΕΦΑΛΑΙΟ 5: ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΣΤΑΣΗΣ ΒΗΜΑΤΟΣ	42
5.1 Καθορισμένη απόσταση βήματος	42
5.2 Υπολογισμός απόστασης βήματος με βάση την κάθετη επιτάχυνση	43
5.3 Υπολογισμός απόστασης βήματος με βάση την διάρκεια	45
5.4 Προσαρμοστικός αλγόριθμος για κάθε χρήστη	45
5.5 Ένας πιο γενικός τύπος	47
ΚΕΦΑΛΑΙΟ 6: ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΥΠΟΛΟΓΙΣΜΟΥ ΘΕΣΗΣ ΜΕ ΤΗ ΜΕΘΟΔΟ DR	48

6.1 Επιλογή μεθόδων και συσκευών	48
6.2 Υλοποίηση κώδικα	51
6.2.1 Αρχικοποίηση μεταβλητών και ορισμός σταθερών.....	52
6.2.2 Μονάδα συλλογής και φιλτραρίσματος δεδομένων	53
6.2.3 Μονάδα ανίχνευσης βήματος.....	53
6.2.4 Κύρια μονάδα συστήματος για ανανέωση θέσης.....	54
6.3 Αξιολόγηση συστήματος	55
6.4 Συμπεράσματα και μελλοντική δουλειά	58
ΠΑΡΑΡΤΗΜΑ.....	60
Π1. Καθολικές μεταβλητές συστήματος.....	60
Π2. Κώδικας συνάρτησης αρχικοποίησης μεταβλητών	63
Π3. Κώδικας συνάρτησης φιλτραρίσματος μετρήσεων	65
Π4. Κώδικας συνάρτησης ανίχνευσης βήματος	70
Π5. Κώδικας κύριου νήματος συστήματος για ανανέωση θέσης.....	76
ΟΡΟΛΟΓΙΑ.....	79
ΑΚΡΩΝΥΜΙΑ – ΣΥΝΤΜΗΣΕΙΣ – ΑΡΤΙΚΟΛΕΞΑ	82
ΑΝΑΦΟΡΕΣ	83

ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία εκπονήθηκε στα πλαίσια του Προπτυχιακού Προγράμματος Σπουδών του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Το αντικείμενο μελέτης της είναι η ανάλυση και υλοποίηση των πιο εξελιγμένων τεχνικών που χρησιμοποιούνται αυτήν τη στιγμή για την υλοποίηση συστημάτων πλοήγησης σε εσωτερικούς χώρους με τη βοήθεια της τεχνικής στίγματος εξ αναμετρήσεως.

Θα ήθελα να ευχαριστήσω θερμά τον Οδυσσέα Σέκκα, υποψήφιο διδάκτορα του Τμήματος Πληροφορικής και Τηλεπικοινωνιών, για την πολύτιμη βοήθειά του καθ' όλη τη διάρκεια εκπόνησης της παρούσης πτυχιακής εργασίας. Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα της εργασίας, κ. Ευστάθιο Χατζηευθυμιάδη, επίκουρο καθηγητή του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ, για την καθοδήγησή του και τη συνεχή παρακολούθηση της προόδου της εργασίας. Τέλος, θα ήθελα να ευχαριστήσω και τον υποψήφιο διδάκτορα Βασίλειο Τσέτσο για τις συμβουλές του σε αρκετά σημεία της παρούσης πτυχιακής.

Αθήνα, Μάιος 2009

Βασίλειος Δ. Ξούρης

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Τα συστήματα πλοήγησης είναι πλέον αναπόσπαστο κομμάτι της καθημερινότητάς μας. Το πιο διαδεδομένο σύστημα πλοήγησης είναι το GPS (Global Positioning System) το οποίο με τη βοήθεια δορυφόρων μπορεί ανά πάσα στιγμή να δώσει την ακριβή θέση του χρήστη με μέγιστη απόκλιση λίγων μέτρων.

Πολλές φορές όμως το GPS δεν είναι σε θέση να μας ενημερώσει για την θέση μας καθώς παράγοντες όπως ψηλά κτήρια, βουνά και διάφορα άλλα εμπόδια μπορούν να καταστήσουν αδύνατη τη λειτουργία του. Επιπλέον υπάρχει πάντα η πιθανότητα παρεμβολής άλλων ραδιοσημάτων που καθιστούν αδύνατη την επικοινωνία με τους δορυφόρους. Όσο αναφορά την πλοήγηση σε εσωτερικούς χώρους, η λειτουργία του GPS είναι αδύνατη, γι αυτό το λόγο έχουν αναπτυχθεί διάφορα συστήματα βασισμένα σε διαφορετικές τεχνολογίες για να καλύψουν αυτό το κενό τα οποία παρουσιάζουμε συνοπτικά στις επόμενες παραγράφους[1].

Το Active Badge [2] είναι το πρώτο σύστημα εντοπισμού θέσης σε εσωτερικούς χώρους που αναπτύχθηκε στο Olivetti Research Laboratory. Αποτελείται από ένα σύστημα κυψελοειδούς εγγύτητας που χρησιμοποιεί τεχνολογία διάχυσης υπέρυθρης ακτινοβολίας. Όπως και με κάθε άλλο σύστημα που χρησιμοποιεί υπέρυθρη ακτινοβολία έτσι και το Active Badge έχει δυσκολία στο να λειτουργήσει σε μέρη όπου υπάρχουν λαμπτήρες φθορισμού ή απευθείας ηλιακή ακτινοβολία, καθώς αυτές οι πηγές φωτός παράγουν υπέρυθρη ακτινοβολία και δημιουργούνται παρεμβολές. Επίσης ένας άλλος περιορισμός είναι ότι η ακτινοβολία έχει ορισμένη ακτίνα διάδοσης και δεν διαδίδεται μέσα από τοίχους.

Η AT&T ανέπτυξε σχετικά πρόσφατα το σύστημα εντοπισμού θέσης Active Bat [3] που χρησιμοποιεί υπερήχους για να εντοπίσει τη θέση ατόμων ή αντικειμένων που φέρουν πάνω τους μία ειδική ετικέτα (tag). Η ετικέτα αυτή εκπέμπει ένα παλμό υπερήχων σε ένα πλέγμα από δέκτες (αισθητήρες) που βρίσκονται τοποθετημένοι στην οροφή. Χρησιμοποιώντας τους υπερήχους για τον εντοπισμό της θέσης απαιτείται ένας μεγάλος αριθμός από αισθητήρες (δέκτες) που βρίσκονται εγκατεστημένοι στην οροφή. Έτσι, η κλιμάκωση, η ευκολία εγκατάστασης αλλά και το κόστος αποτελούν τα σημαντικότερα μειονεκτήματα του συστήματος Active Bat.

Motion Star DC magnetic tracker [4]: Αυτό το σύστημα παράγει μαγνητικούς παλμούς μέσω μιας κεραίας που βρίσκεται σε σταθερή θέση και υπολογίζει τη θέση και τον

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

προσανατολισμό των κινητών κεραιών με διάφορες μετρήσεις των μαγνητικών παλμών. Οι μετρήσεις αυτές εκτελούνται πάντα σε συνδυασμό με τη σταθερή επίδραση του γήινου μαγνητικού πεδίου. Το σύστημα παρέχει μεγάλη ορθότητα (accuracy), της τάξης του 1 mm, και ακρίβεια (precision) για το 100% του χρόνου. Παρόλα αυτά όμως έχει και αρκετά μειονεκτήματα όπως το υψηλό κόστος καθώς και η μείωση της απόδοσης με την παρουσία μεταλλικών αντικειμένων στο χώρο. Επίσης, η κλιμάκωσή του δεν είναι ικανοποιητική.

Το σύστημα εντοπισμού θέσης Cricket [5] χρησιμοποιεί πομπούς υπερήχων στην υποδομή, ενώ οι δέκτες υπερήχων βρίσκονται στα αντικείμενα τα οποία εντοπίζονται. Αυτή η προσέγγιση αναγκάζει τους δέκτες να εκτελούν όλους τους υπολογισμούς για την εκτίμηση της θέσης τους. Τα μειονεκτήματα του είναι η έλλειψη κεντρικού ελέγχου και διαχείρισης καθώς και ο αριθμός των υπολογισμών που εκτελούνται στους κινητούς δέκτες κάνοντας τους έτσι πολυπλοκότερους και οδηγώντας τους σε μεγαλύτερη κατανάλωση ενέργειας.

Το RADAR [6] αναπτύχθηκε από τη Microsoft και βασίζεται στην τεχνολογία IEEE 802.11 WaveLAN. Το σύστημα μετράει στο σταθμό βάσης την ισχύ του σήματος που στέλνουν οι ασύρματες συσκευές και χρησιμοποιεί αυτήν την πληροφορία για να υπολογίσει τη θέση τους μέσα σε ένα κτίριο. Ο υπολογισμός της θέσης γίνεται στις δύο και όχι στις τρεις διαστάσεις.

Το σύστημα Easy Living [7] (επίσης της Microsoft) παρέχει πληροφορία θέσης χρησιμοποιώντας τρισδιάστατες στερεοσκοπικές κάμερες (Digiclops). Απαιτείται μεγάλη υπολογιστική ισχύς για να επεξεργαστούν την εικόνα που λαμβάνουν οι κάμερες. Η λειτουργία του Smart Floor [8] βασίζεται σε ειδικούς αισθητήρες πίεσης οι οποίοι είναι εγκατεστημένοι στο δάπεδο ενός δωματίου και αντιλαμβάνονται το βάδισμα των ατόμων. Τα δεδομένα από τους αισθητήρες χρησιμοποιούνται για τον εντοπισμό των ατόμων. Το πλεονέκτημα του συστήματος είναι ότι δεν απαιτείται από τα άτομα να φέρουν πάνω τους ειδική ετικέτα. Τα μειονεκτήματα του είναι η δυσκολία στην κλιμάκωση αλλά και το μεγάλο κόστος διότι το δάπεδο κάθε κτιρίου πρέπει να αλλαχθεί κατάλληλα ώστε να εγκατασταθούν οι αισθητήρες.

Στην παρούσα πτυχιακή όμως, η τεχνική που θα αναλυθεί σε βάθος είναι η τεχνική στίγματος εξ αναμετρήσεως (dead reckoning technique). Η τεχνική αυτή προέρχεται από πολύ παλιά, όταν χρησιμοποιούσαν από τους ναυτικούς για να υπολογίσουν την απόσταση που είχαν διανύσει χρησιμοποιώντας μια πυξίδα και υπολογίζοντας την ταχύτητα του πλοίου. Αυτή μάλιστα την τεχνική είχε χρησιμοποιήσει ο Χριστόφορος

Κολόμβος στα ταξίδια του για την ανακάλυψη του Νέου Κόσμου. Τα συστήματα «dead reckoning» λοιπόν, μπορούν είτε να χρησιμοποιηθούν σαν βοηθητικά στα συστήματα GPS (π.χ. όταν χαθεί το σήμα μπαίνει σε λειτουργία η τεχνική dead reckoning) είτε να είναι αυτόνομα και να λειτουργήσουν σαν ένα ολοκληρωμένο σύστημα πλοήγησης έχοντας σαν μόνη είσοδο την αρχική θέση του χρήστη.

Ο πιο σημαντικός κλάδος αυτής της τεχνικής είναι η τεχνική στίγματος εξ αναμετρήσεως για πεζούς (pedestrian dead reckoning), η οποία με την χρήση κατάλληλων εξελιγμένων αλγορίθμων μπορεί να ανιχνεύει τα βήματα ενός πεζοπόρου χρήστη, να υπολογίζει την απόσταση που διανύθηκε και να ανανεώνει την θέση του. Πιθανοί τομείς όπου μπορεί να χρησιμοποιηθεί αυτή η τεχνολογία είναι τομείς όπως :

- Στρατιωτικές επιχειρήσεις
- Υπηρεσίες άμεσης δράσης (αστυνομία, πυροσβεστική)
- Πλοήγηση και καθοδήγηση πεζοπόρων
- Τουρισμός
- Υπηρεσίες πλοήγησης για άτομα με διάφορες αναπηρίες

Τα αποτελέσματα που παίρνουμε από την τεχνική αυτή μπορούν να είναι αρκετά αξιόπιστα με ελάχιστη απόκλιση από την πραγματική μας θέση. Παρόλαυτα συνιστάται να υπάρχει ανά τακτά διαστήματα διόρθωση των συντεταγμένων του συστήματος με τις πραγματικές έτσι ώστε να μην βγαίνει εκτός πορείας. Αν και σε κάθε μέτρηση βήματος το λάθος σε μια καλή υλοποίηση μπορεί να είναι πολύ μικρό (της τάξεως των μερικών εκατοστών), εάν δεν υπάρχει ανατροφοδότηση με την πραγματική μας θέση τότε το λάθος συσσωρεύεται συνεχώς, κάτι το οποίο μας οδηγεί σε μεγάλες αποκλίσεις.

Η τεχνική αυτή βασίζεται στον τρόπο βάρδησης του ανθρώπου και χωρίζεται στις εξής λειτουργίες οι οποίες θα αναλυθούν παρακάτω:

- Συλλογή δεδομένων από τα όργανα μέτρησης
- Φιλτράρισμα δεδομένων για απομάκρυνση του θορύβου

- Επεξεργασία δεδομένων και ανίχνευση βήματος
- Υπολογισμός μήκους βήματος και νέας θέσης

Τα δεδομένα που χρειάζεται η συγκεκριμένη μέθοδος είναι η κατεύθυνση του χρήστη και οι επιτάχυνσεις του σώματός του στους άξονες X,Y,Z έτσι ώστε να μπορέσει να ανιχνεύσει τα διαστήματα κατά τα οποία πραγματοποιείται ένα βήμα.

Τελειώνοντας την εισαγωγή θα αναφερθούμε στον τρόπο με τον οποίο έχει οργανωθεί η παρούσα πτυχιακή. Στο δεύτερο κεφάλαιο παρουσιάζονται οι διάφοροι τρόποι τοποθέτησης των αισθητήρων στο σώμα του χρήστη. Στο τρίτο κεφάλαιο γίνεται εκτενής αναφορά στις πιο λειτουργικές και διαδεδομένες μεθόδους ανίχνευσης βήματος ενός πεζού χρήστη μέσω της επιτάχυνσης του σώματός του. Στο τέταρτο κεφάλαιο ακολουθεί μια ανάλυση των τρόπων με των οποίων μπορούν να φιλτραριστούν τα δεδομένα της επιτάχυνσης του χρήστη που συλλέγονται από τους αισθητήρες οι οποίοι είναι τοποθετημένοι πάνω στο σώμα του. Στο πέμπτο κεφάλαιο γίνεται αναφορά σε μεθόδους υπολογισμού της κατεύθυνσης του χρήστη ενώ στο έκτο κεφάλαιο παρουσιάζονται τεχνικές υπολογισμού της απόστασης που διανύεται σε κάθε ανιχνευμένο βήμα. Τέλος, στο έβδομο κεφάλαιο περιγράφεται ένα σύστημα πλοήγησης που αναπτύχθηκε για τις ανάγκες της παρούσης πτυχιακής και βασίζεται στην τεχνική στίγματος εξ αναμετρήσεως.

Στο τέλος της εργασίας περιέχονται οι διάφορες αναφορές (άρθρα, ιστοσελίδες, βιβλία), οι οποίες χρησιμοποιήθηκαν για την άντληση γνώσης ενώ ακολουθεί και ο κώδικας ο οποίος χρησιμοποιήθηκε για την υλοποίηση του συστήματος που περιγράφεται στο κεφάλαιο 7.

ΚΕΦΑΛΑΙΟ 2

ΑΝΙΧΝΕΥΣΗ ΒΗΜΑΤΟΣ

Πολύ σημαντικό ρόλο παίζει η τοποθέτηση των αισθητήρων πάνω στο σώμα του χρήστη. Ανάλογα με το μέρος του σώματος που θα επιλεγεί θα πρέπει να αναπτυχθεί και ο κατάλληλος τρόπος ανίχνευσης βήματος καθώς η γενική εικόνα των δεδομένων επιταχύνσεως που συλλέγονται αλλάζει κατά πολύ. Έτσι πριν αρχίσει η ανάπτυξη ενός συστήματος πλοήγησης dead reckoning πρέπει πρώτα να αποφασιστεί ο τρόπος τοποθέτησης που θα ακολουθηθεί.

Τα σημεία του σώματος που επιλέγονται συνήθως είναι η μέση, το πόδι(όσο πιο χαμηλά γίνεται), το κεφάλι. Παρακάτω παρατίθενται κάποιες ενδεικτικές τοποθετήσεις των επιταχυνσιομέτρων στα σημεία που αναφέρθηκαν.



Εικόνα 2.1: Παράδειγμα τοποθέτησης συστήματος στο πόδι του χρήστη



Εικόνα 2.2: Παράδειγμα τοποθέτησης συστήματος στο κεφάλι του χρήστη

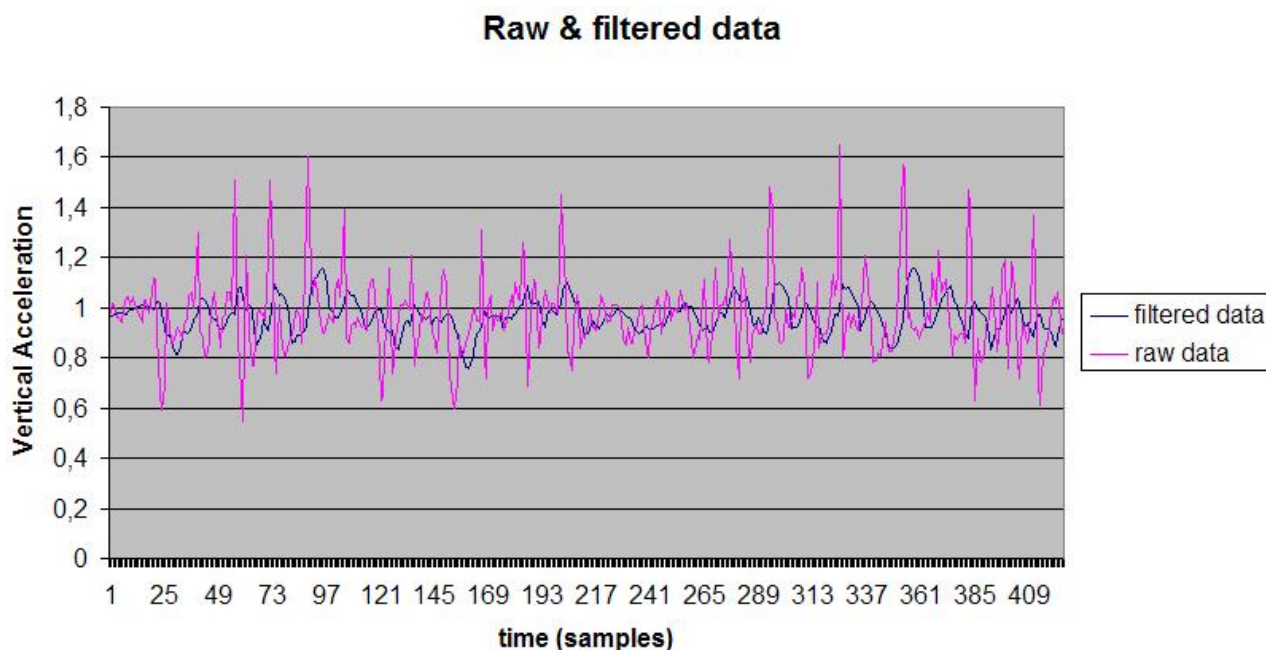


Εικόνα 2.3 : Παράδειγμα τοποθέτησης συστήματος στη μέση του χρήστη

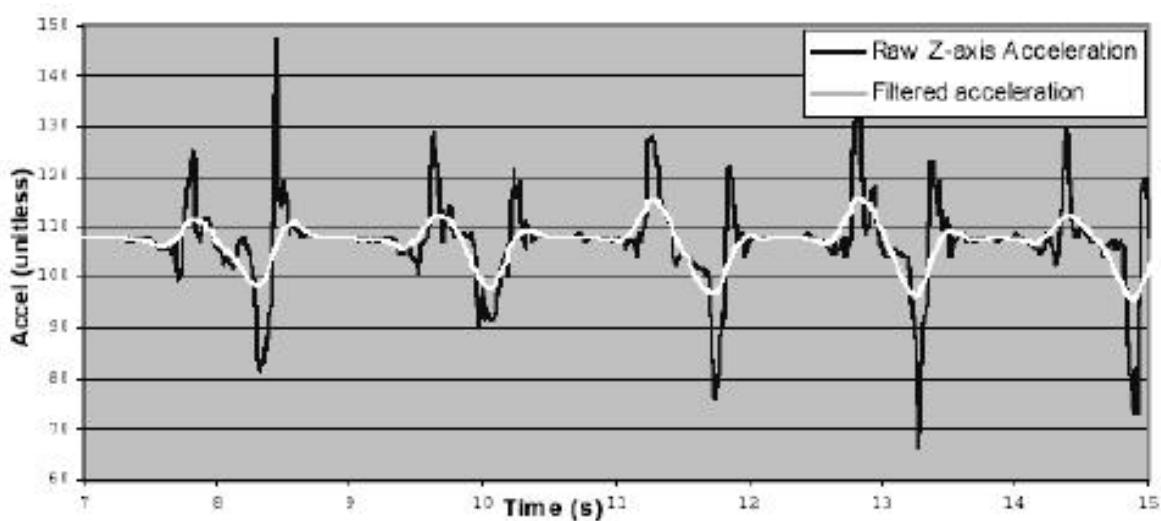
Όπως ήδη αναφέρθηκε, τα δεδομένα που χρειάζεται η μέθοδος dead reckoning είναι η κατεύθυνση του χρήστη και οι επιταχύνσεις στους άξονες X,Y,Z. Δεδομένου ότι η συχνότητα βηματισμού του ανθρώπου σε κανονικό περπάτημα δεν ξεπερνά τα 3 Hz (το πολύ 3 βήματα σε ένα δευτερόλεπτο) [9], μπορούμε να πούμε ότι αν η δειγματοληψία προσαρμοστεί σε μια συχνότητα μεταξύ 20 και 30 Hz τότε θα υπάρχουν αρκετά δείγματα για να τροφοδοτηθεί η λειτουργία φιλτραρίσματος των «raw data» και να δώσει αξιόπιστα αποτελέσματα.

Η ανίχνευση βήματος είναι από τα πιο σημαντικά στάδια της μεθόδου «dead reckoning». Για την ανίχνευση ενός βήματος χρησιμοποιείται κυρίως η κάθετη επιτάχυνση του σώματος του χρήστη, η εικόνα της οποίας -όταν περπατάει ο χρήστης- παρουσιάζει μια συγκεκριμένη μορφή για κάθε του βήμα, ανάλογα πάντα βέβαια με το πού έχει επιλεγεί να τοποθετηθούν τα επιταχυνσιόμετρα.

Στις εικόνες 2.3 και 2.4 παρουσιάζεται το σήμα της κάθετης επιτάχυνσης όταν η τοποθέτηση έχει γίνει στη μέση και στο πόδι του χρήστη αντίστοιχα.



Εικόνα 2.3 : Τοποθέτηση επιταχυνσιόμετρου στη μέση του χρήστη



Εικόνα 2.4 : Τοποθέτηση επιταχυνσιόμετρου στο πόδι του χρήστη [10]

2.1 Μέθοδος ανίχνευσης κορυφών

Σε αυτό το κομμάτι θα αναλύσουμε πως γίνεται η ανίχνευση βήματος με χρήση της μεθόδου για ανίχνευση κορυφών (peak detection) στην περίπτωση που η μέτρηση της κάθετης επιτάχυνσης γίνεται **με βάση τη μέση** του χρήστη.

Κάθε άνθρωπος έχει διαφορετικό τρόπο βηματισμού. Τα βασικά χαρακτηριστικά του ανθρώπινου βηματισμού (βλέπε εικόνα 2.1.1) όμως παραμένουν τα ίδια [11], που σημαίνει ότι :

- ένα βήμα αρχίζει ακριβώς μετά από ένα άλλο.
- Όσο το ένα πόδι παραμένει πίσω από το άλλο, το σώμα σηκώνεται προς τα πάνω.
- Όταν το ένα πόδι περνάει το άλλο, τότε το σώμα αρχίζει να πέφτει προς τα κάτω.
- Τη στιγμή που το πόδι ακουμπάει το έδαφος, εκείνη τη στιγμή το βήμα τελειώνει και αρχίζει το επόμενο.



Εικόνα 2.1.1 : Ακολουθία ανθρώπινου βηματισμού

Τα επιταχυνσιόμετρα λοιπόν που είναι τοποθετημένα στη μέση του χρήστη μπορούν να εντοπίσουν την μέγιστη και ελάχιστη κάθετη επιτάχυνση ενός βήματος. Ακόμα και όταν ο χρήστης κάνει πλάγια βήματα.

Ας υποθεσουμε ότι έχουμε το σήμα της κάθετης επιτάχυνσης που φαίνεται στην εικόνα 2.4. Η ανίχνευση βήματος γίνεται με βάση τα φιλτραρισμένα δεδομένα az τα οποία έχουν φιλτραριστεί με ένα φίλτρο «μέσου όρου» (moving average) των 8 τελευταίων μετρήσεων. Ο αλγόριθμος ανίχνευσης προσπαθεί να βρεί τις ακραίες τιμές του az και θεωρεί ότι ανιχνεύθηκε ένα πιθανό βήμα κάθε φορά που εντοπίζει ένα τοπικό μέγιστο ακολουθούμενο από ένα τοπικό ελάχιστο. Οι ακραίες τιμές βρίσκονται ελέγχοντας το πρόσημο της διαφοράς μεταξύ 2 διαδοχικών φιλτραρισμένων μετρήσεων, με πιο απλά λόγια ελέγχοντας το $sgn\{az(t) - az(t-1)\}$. Ο κανόνας ανίχνευσης είναι ότι όταν το πρόσημο αυτό αλλάζει από θετικό σε αρνητικό, τότε έχουμε τοπικό μέγιστο της κάθετης επιτάχυνσης. Όταν αλλάζει από αρνητικό σε θετικό τότε έχουμε τοπικό ελάχιστο. Αυτός ο έλεγχος βοηθάει στην ανίχνευση ακρότατων τιμών της κάθετης επιτάχυνσης σε πραγματικό χρόνο.

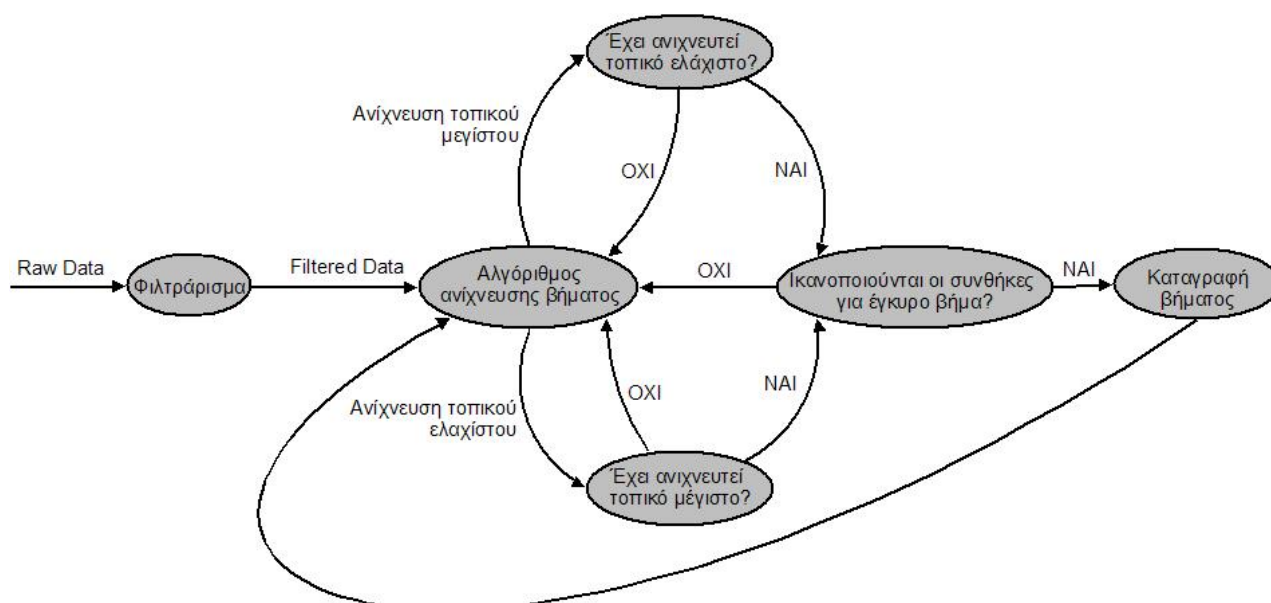
Έτσι αν καταγράψουμε μόνο τις ακρότατες τιμές που ανιχνεύονται σε πραγματικό χρόνο, έχουμε μια σημαντική μείωση αποθηκευμένων δεδομένων ανά πάσα στιγμή. Αν δηλαδή υποθέσουμε ότι η συχνότητα δειγματοληψίας είναι 30 Hz, με την τεχνική αυτή αποθηκεύουμε μόνο 4 τιμές κάθετης επιτάχυνσης (που αντιστοιχούν σε 4 ακραία σημεία, 2 τοπικά μέγιστα και 2 τοπικά ελάχιστα) κάθε δευτερόλεπτο θεωρώντας μια συχνότητα βηματισμού της τάξης των 2 Hz [9].

Με αυτήν την τεχνική παρατηρούνται πολλές φορές λανθασμένες ανιχνεύσεις βημάτων οι οποίες όμως μπορούν να εξαλειφθούν κάνοντας κάποιους επιπλέον ελέγχους μετά από κάθε πιθανή ανίχνευση βήματος όπως :

- Αν η μέγιστη και η ελάχιστη επιτάχυνση είναι μέσα σε κάποια όρια (ή αν η διαφορά τους είναι πάνω από κάποιο όριο).
- Αν μεταξύ της ανίχνευσης ενός τοπικού ελαχίστου και τοπικού μεγίστου της κάθετης επιτάχυνσης έχει περάσει ένα ελάχιστο χρονικό όριο.
- Αν έχουμε εναλλαγή της ανίχνευσης τοπικού μεγίστου και ελαχίστου.

Οι έλεγχοι αυτοί βοηθούν στην αποφυγή λανθασμένης ανίχνευσης βημάτων σε φάση στάσης του χρήστη, σε περιπτώσεις πιθανού θορύβου του σήματος επιτάχυνσης λόγω ανώμαλου εδάφους κλπ.

Εφόσον λοιπόν γίνει εντοπισμός τοπικού μεγίστου ακολουθούμενου από εντοπισμό τοπικού ελαχίστου, και οι παραπάνω 3 έλεγχοι επιστρέψουν θετικά αποτελέσματα τότε το βήμα καταγράφεται σαν έγκυρο και αποθηκεύονται η μέγιστη και η ελάχιστη κάθετη επιτάχυνση οι οποίες θα χρησιμοποιηθούν για τον υπολογισμό της απόστασης που διάνυσε ο χρήστης κατά τη διάρκεια του συγκεκριμένου βήματος (βλέπε παράγραφο 6). Στην εικόνα 2.1.2 φαίνεται μια γραφική αναπαράσταση του αλγορίθμου ανίχνευσης βήματος που μόλις περιγράφηκε.

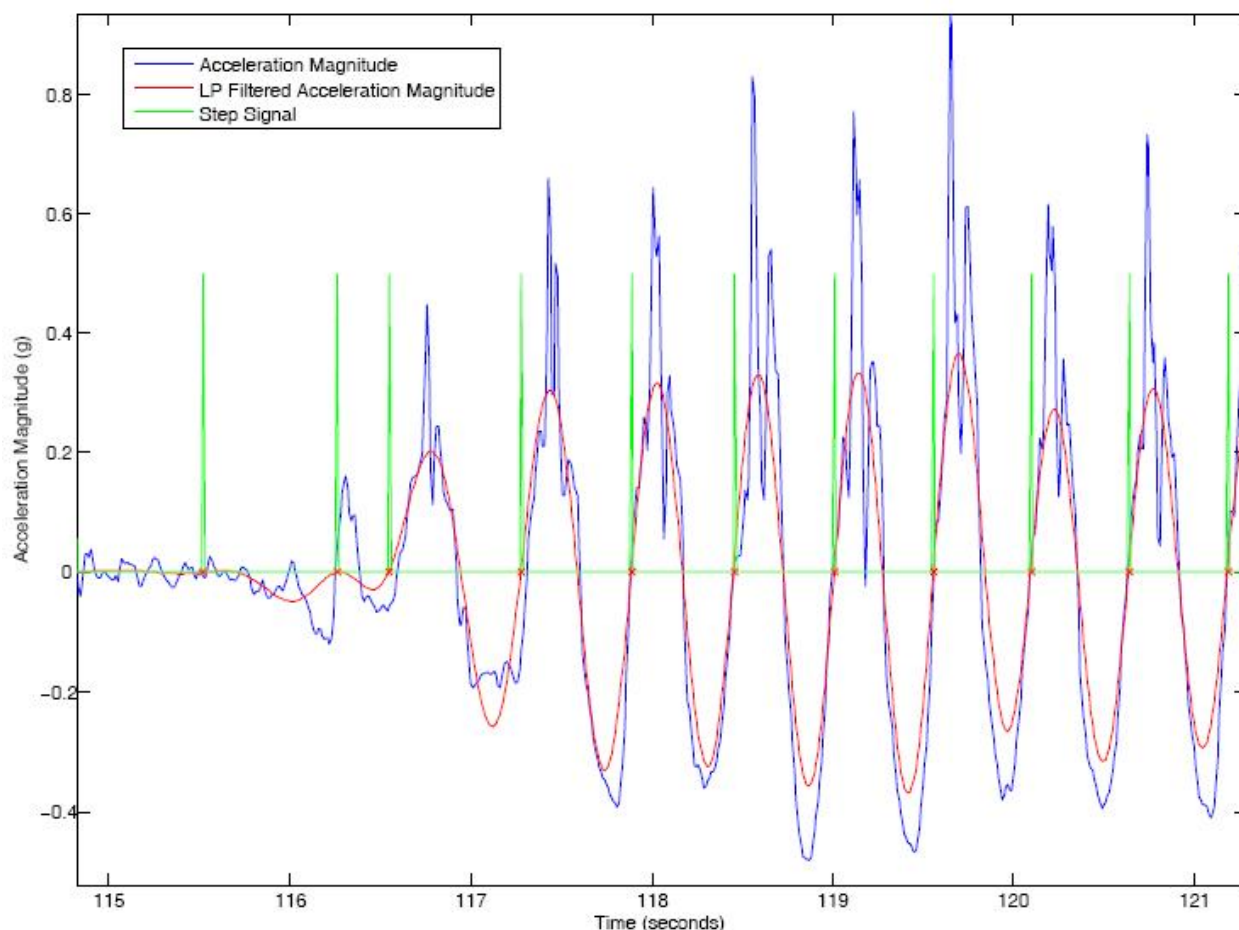


Εικόνα 2.1.2 : Αναπαράσταση αλγορίθμου ανίχνευσης βήματος

2.2 Μέθοδος ανίχνευσης μηδενικών διασταυρώσεων (zero crossing method)

Ένας άλλος παρόμοιος τρόπος ανίχνευσης βήματος με τον παραπάνω (στην περίπτωση που τα επιταχυνσιόμετρα έχουν τοποθετηθεί στη μέση του χρήστη) είναι η μέθοδος κατά την οποία ελέγχεται αν το φιλτραρισμένο σήμα της κάθετης επιτάχυνσης γίνεται απο αρνητικό, θετικό. Δηλαδή αν το σήμα διασταυρώνεται με τον άξονα χ.

Κάθε φορά που συμβαίνει αυτό και παράλληλα ικανοποιούνται κάποιοι επιπλέον ελέγχοι, παρόμοιοι με αυτούς που θέσαμε στην παράγραφο 2.1 για αποφυγή ανίχνευσης λανθασμένων βημάτων, τότε το βήμα καταγράφεται σαν έγκυρο. Ένα παράδειγμα ανίχνευσης μηδενικών διασταυρώσεων φαίνεται στην εικόνα 2.2.1, παρατηρούμε ότι ανιχνεύεται βήμα (αναπαριστάται με μια κάθετη πράσινη γραμμή) κάθε φορά που έχουμε «zero crossing» της κάθετης επιτάχυνσης. [12]

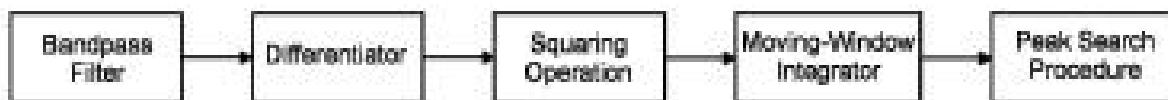


Εικόνα 2.2.1 : Ανίχνευση βήματος με τη μέθοδο zero crossing

2.3 Μέθοδος Pan-Tompkins

Αυτή η μέθοδος προτάθηκε από τους Pan και Tompkins για την ανίχνευση R-peak σε ECG (electrocardiogram) σημάτα. Ο συγκεκριμένος όμως αλγόριθμος μπορεί να χρησιμοποιηθεί και για την περίπτωση ανίχνευσης βημάτων πάνω στο σήμα της επιτάχυνσης.

Περιλαμβάνει μια σειρά απο φίλτρα και μεθόδους που έχουν ως σκοπό να ομαλοποιήσουν το σήμα της κάθετης επιτάχυνσης και να κάνουν πιο εύκολη την ανίχνευση κορυφών, δηλαδή πιθανών βημάτων. Στο σχήμα 2.3.1 παρουσιάζονται τα βήματα που ακολουθεί ο αλγόριθμος Pan-Tompkins.



Εικόνα 2.3.1 : Τα βήματα του αλγορίθμου Pan-Tompkins

- «bandpass» φίλτρο : το ζωνοπερατό φίλτρο μειώνει την επιρροή των εξωγενών παραγόντων στο σήμα. Στην συγκεκριμένη μέθοδο χρησιμοποιείται ένα «low-pass» (χαμηλοπερατό) φίλτρο με μικρούς ακέραιους συντελεστές κάτι που οδηγεί σε γρηγορότερη εκτέλεση του αλγορίθμου. Για παράδειγμα, μπορεί να χρησιμοποιηθεί το χαμηλοπερατό φίλτρο που παρουσιάζεται απο κάτω με μια συχνότητα αποκοπής της τάξεως των 20 Hz :

$$H(z) = \frac{1 (1 - z^{-4})^2}{16 (1 - z^{-1})^2}$$

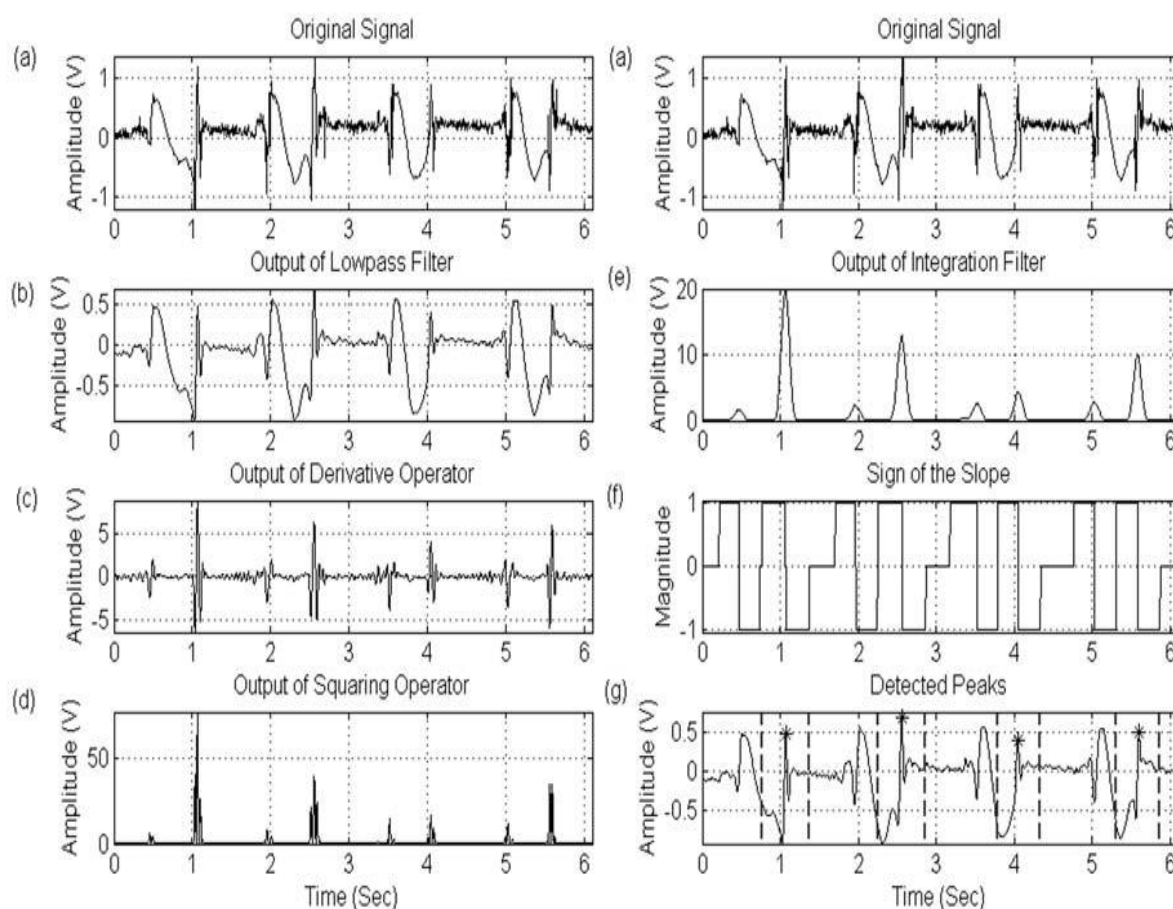
- Τελεστής παραγώγου: Στο σχήμα 2.3.2(b) φαίνεται η λειτουργία του βήματος αυτού, η οποία είναι να καταστείλλει τα στοιχεία χαμηλών συχνοτήτων και να μεγαλώσει εκείνα των υψηλών συχνοτήτων που βρίσκονται στις περιοχές μεγάλων κλίσεων. Ο τελεστής παραγώγου ορίζεται από την σχέση :

$$y(n) = \frac{1}{8} [2x(n) + x(n-1) - x(n-3) - 2x(n-4)]$$

- Διαδικασία τετραγωνισμού («squaring»): Η διαδικασία τετραγωνισμού οδηγεί παντα σε θετικό αποτέλεσμα και ουσιαστικά ενισχύει τις μεγάλες τιμές περισσότερο από τις μικρές. Το αποτέλεσμα της διαδικασίας αυτής φαίνεται στο σχήμα 2.3.2 (c).
- Ολοκληρωτής : Το αποτέλεσμα της προηγούμενης διαδικασίας (τετραγωνισμός) έχει σαν αποτέλεσμα την εμφάνιση πολλαπλών κορυφών στο σήμα μας και αυτό που χρειαζόμαστε τώρα είναι να το ομαλοποιήσουμε όσο μπορούμε περισσότερο. Σε αυτό το στάδιο, αυτό επιτυγχάνεται με τη βοήθεια ενός ολοκληρωτή κυλιόμενου παραθύρου (moving-window) της μορφής:

$$y(n) = \frac{1}{N} [x(n - (N - 1)) + \dots + x(n - (N - 2)) + \dots + x(n)]$$

Όπου ο αριθμός N έχει επιλεγθεί εμπειρικά να είναι ίσος με 20. Το αποτέλεσμα της διαδικασίας ολοκλήρωσης φαίνεται στο σχήμα 2.3.2 (e).



Εικόνα 2.3.2 : Αποτελέσματα των βημάτων του αλγορίθμου Pan-Tompkins

Καθοδηγητικό σημάδι (Fiducial point): Το αποκαλούμενο καθοδηγητικό σημείο, το οποίο καθορίζεται σαν η θέση μιας κορυφής, ανιχνεύεται χρησιμοποιώντας έναν προσαρμοστικό κατώτατο όριο στην μέθοδο Pan-Tompkins. Δυστυχώς, το μέγεθος των κορυφών στο σήμα επιταχυνσιομέτρων ποικίλλει ανάλογα με το χρόνο, τη θέση του σώματος, τον τρόπο βαδίσματος, την ένταση με την οποία το πόδι χτυπάει στο έδαφος και διάφορους άλλους παράγοντες. Επιπλέον, οι μαθηματικές μετατροπές ενισχύουν τη

διαφορά αυτή εντυπωσιακά. Επομένως, ο αλγόριθμος ανίχνευσης κορυφών που χρησιμοποιεί κάποια κατώτατα όρια δεν είναι επαρκής.

Γι αυτόν το λόγο, σε αυτήν την περίπτωση χρησιμοποιούνται τα διαδοχικά ανηφορίσματα και κατηφορίσματα του τροποποιημένου σήματος για να φέρουν σε πέρας αυτήν τη λειτουργία. Στο σχήμα 2.3.2 (f) φαίνεται πως μετασχηματίζεται το σήμα μας όταν ερμηνεύσουμε τις θετικές κλίσεις (ανηφορίσματα) σαν 1 και τις αρνητικές κλίσεις (κατηφορίσματα) σαν -1. Με αυτόν τον τρόπο, ένας ολοκληρωμένος κύκλος βηματισμού αναπαριστάται σαν ένα διαδοχικό ζευγάρι (1,-1) ,δηλαδή [(1,-1) , (1,-1)].

Το διάστημα του δεύτερου ζευγαριού (1,-1) ορίζεται σαν «διάστημα ανίχνευσης κορυφών» και μπορεί να φανεί καθαρά στο σχήμα 2.3.2 (g) όπου είναι η περιοχή ανάμεσα από τις διακεκομμένες γραμμές. Επίσης, το τοπικό μέγιστο που ανιχνεύεται μέσα στο «διάστημα ανίχνευσης κορυφών» ορίζεται σαν ανιχνευμένο βήμα, δηλαδή η στιγμή κατά την οποία το πόδι του χρήστη χτυπάει στο έδαφος (σημειώνεται με έναν αστερίσκο στο σχήμα 2.3.2 (g)).

2.4 Template-matching μέθοδος

Η κύρια ιδέα αυτής της μεθόδου είναι να παράγει ένα πρότυπο το οποίο αναπαριστά ένα τυπικό κύκλο ενός ανθρωπίνου βήματος. Έτσι ο αλγόριθμος συγκρίνει το σήμα της επιτάχυνσης που δέχεται με το πρότυπο που έχει αποθηκευμένο και σε περίπτωση που αυτά τα δύο ταιριάζουν σε κάποιο συγκεκριμένο ποσοστό τότε το σύστημα ειδοποιείται ότι ανιχνεύθηκε βήμα.

Ας δούμε την μέθοδο αυτή όμως πιο αναλυτικά:

- Αρχικά, το σήμα της επιτάχυνσης χωρίζεται σε αρκετά ξεχωριστά ίσα κομμάτια (π.χ. των 10 δευτερολέπτων έκαστο).
- Στη συνέχεια το σήμα φιλτράρεται με ένα lowpass φίλτρο με μια συχνότητα αποκοπής των 20 Hz.
- Γίνεται έλεγχος για το αν υπάρχει ήδη κάποιο πρότυπο αποθηκευμένο στη μνήμη. Στην περίπτωση που δεν υπάρχει, το πρώτο κομμάτι (το 1^ο δείγμα 10 δευτερολέπτων) εξάγεται και χρησιμοποιείται σαν προσωρινό πρότυπο. Εφόσον αυτή είναι η πρώτη εκτέλεση του αλγορίθμου και εφόσον έχουμε ένα προσωρινό πρότυπο, όλες οι

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

παράμετροι αρχικά είναι προσεγγιστικές λόγω του ότι ακόμα αγνοούνται τα πραγματικά χαρακτηριστικά του συγκεκριμένου τύπου βήματος.

- Στο επόμενο βήμα, το πρότυπο που δημιουργήθηκε «κυλάει» κατά μήκος ολόκληρου του μπλοκ δεδομένων του σήματος επιτάχυνσης και υπολογίζεται ένα ομαλοποιημένο σήμα συσχέτισης μεταξύ των μπλοκ και του αποθηκευμένου προτύπου (normalized cross-correlation signal) το οποίο δείχνει κατα πόσον τα δυο διανύσματα μοιάζουν μεταξύ τους σύμφωνα με τον παρακάτω τύπο:

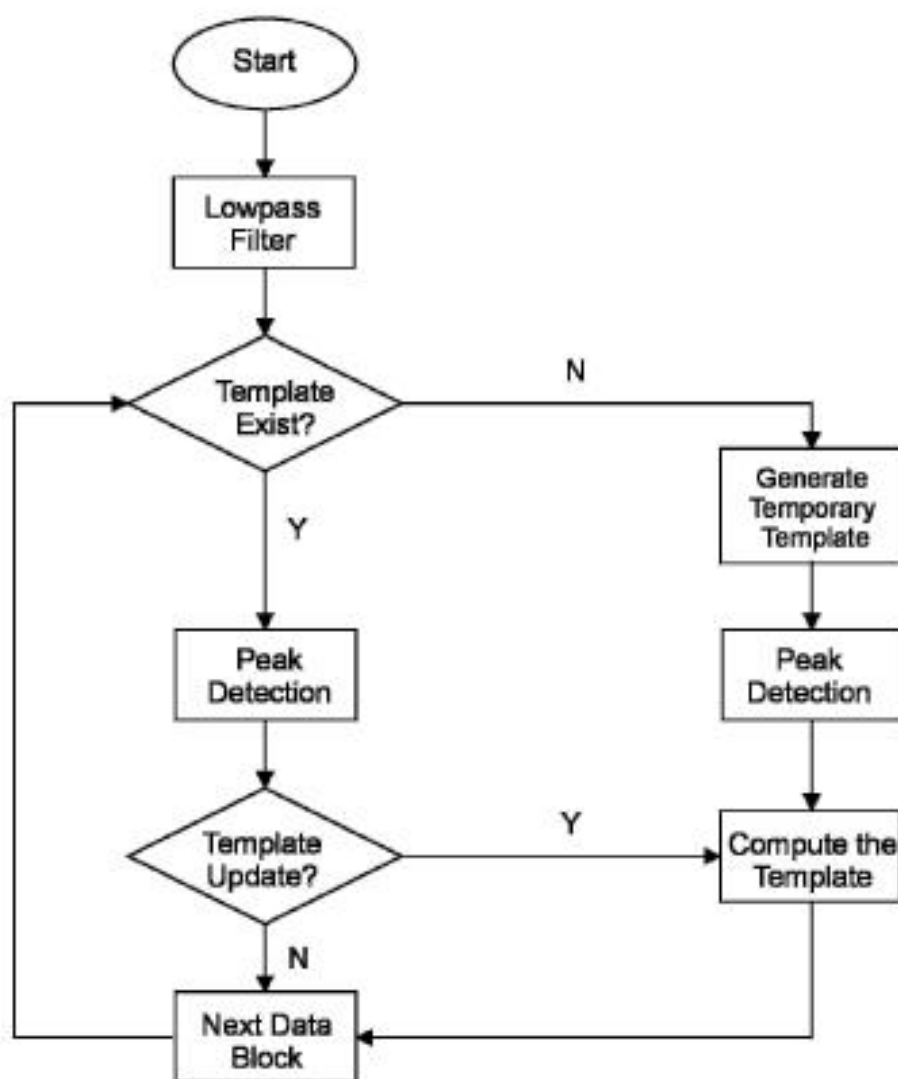
$$R_N[k] = \frac{\langle X, Y \rangle}{\sqrt{\|X\| \cdot \|Y\|}} = \frac{R_{XY}(k)}{\sqrt{R_{XX}(0) \cdot R_{YY}(0)}}$$

Όπου $\langle X, Y \rangle$ το εσωτερικό γινόμενο των διανυσμάτων X και Y ,

$\|X\|$ η νόρμα του διανύσματος X ,

$R_{xy}(k)$ η συσχέτιση (cross-correlation) των X και Y , και

$R_{xx}(0)$ η αυτοσυσχέτιση (autocorrelation) του X στο σημείο 0.



Εικόνα 2.4.1 : Αναπαράσταση αλγορίθμου «template matching»

Το οριακό σημείο για το μέγιστο είναι 1 το οποίο επιτρέπει να ορίσουμε ένα ενιαίο κατώφλι (threshold) για όλα τα δεδομένα παρά τα διαφορετικά μεγέθη του σήματος επιτάχυνσης.

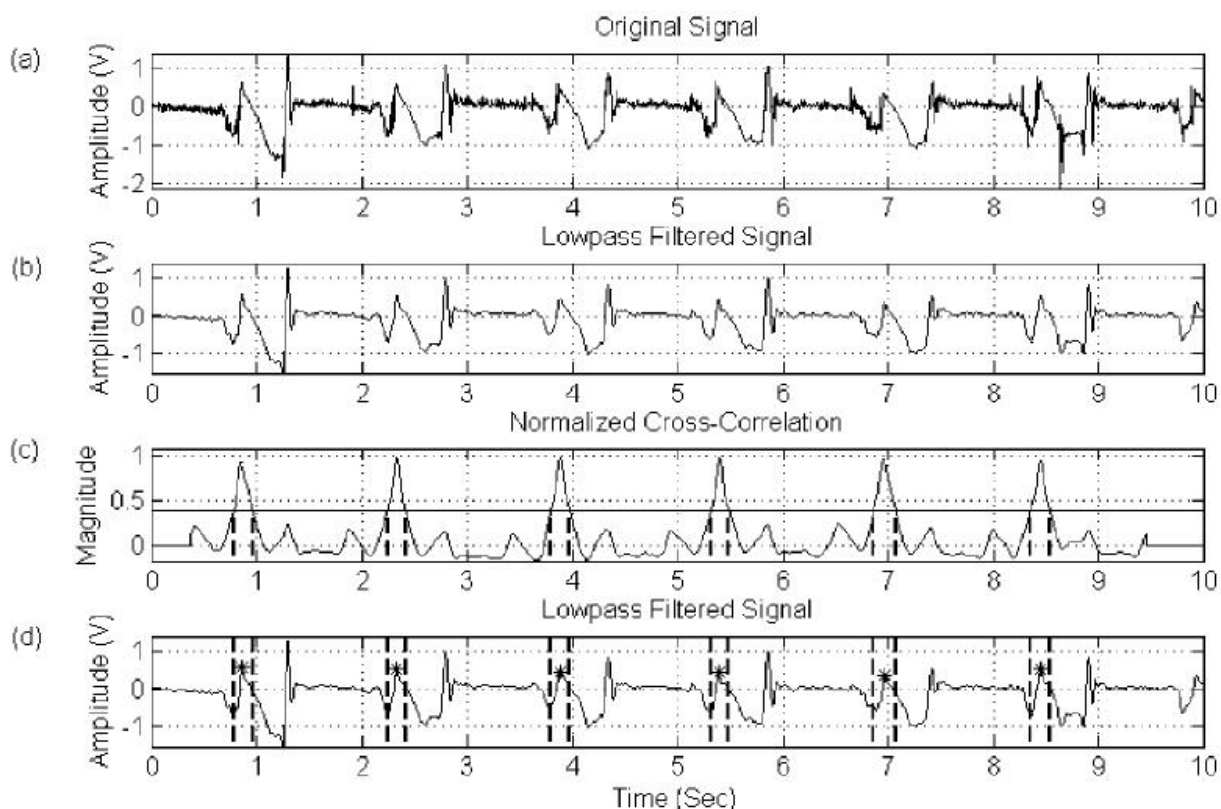
Το διάστημα κατά το οποίο η συσχέτιση ξεπερνάει το κατώφλι που έχουμε ορίσει καλείται «διάστημα ανίχνευσης κορυφών» και φαίνεται στο σχήμα 2.4.2 (c) σαν μια οριζόντια γραμμή. Τα τοπικά μέγιστα που πέφτουν εντός αυτού του διαστήματος καλούνται αξιόπιστα σημεία βημάτων και είναι αυτά που εντοπίζονται όταν ανιχνεύεται ένα νέο βήμα από τον αλγόριθμο της μεθόδου «template matching».

Αρχικά η μέθοδος αυτή ανιχνεύει τα πρώτα βήματα όταν το πόδι ανυψώνεται από το έδαφος. Στη συνέχεια, με την εξέταση του πρώτου μπλοκ δεδομένων από τον αλγόριθμο παράγεται ένα πιο αντιπροσωπευτικό πρότυπο αντί του προσωρινού

προτύπου που χρησιμοποιήσαμε στην αρχή, ευθυγραμμίζοντας τα βήματα που ανιχνεύθηκαν σε αυτό το μπλοκ και υπολογίζοντας το μέσο όρο τους. Έτσι κατασκευάζουμε ένα νέο πιο αντιπροσωπευτικό πρότυπο το οποίο ίσως χρησιμοποιηθεί στη συνέχεια για τα επόμενα δεδομένα. Πριν όμως προχωρήσουμε σε επεξεργασία του επόμενου μπλοκ δεδομένων πρέπει ο αλγόριθμος να αποφασίσει εάν το νέο πρότυπο που παράχθηκε είναι πιο αντιπροσωπευτικό των δεδομένων. Το σήμα επιτάχυνσης του βήματος ενδέχεται να αλλάζει με την πάροδο του χρόνου, ενδέχεται όμως να παραμένει και το ίδιο. Κάτι τέτοιο σημαίνει ότι μετά από κάθε εξέταση ενός νέου μπλοκ δεδομένων, πρέπει να παίρνεται μια απόφαση για το αν πρέπει να αντικατασταθεί το αποθηκευμένο πρότυπο στη μνήμη.

Αυτό γίνεται ελέγχοντας αν οι μέγιστες κορυφές που ανιχνεύθηκαν στο ομαλοποιημένο σήμα συσχέτισης είναι μικρότερες από μια σταθερά που δίνεται σαν παράμετρος (εμπειρικά μπορεί να χρησιμοποιηθεί η σταθερά 0.55). Στην περίπτωση που ισχύει αυτή η σύγκριση, τότε το νέο πρότυπο παράγεται από τα βήματα που ανιχνεύθηκαν στο τρέχον μπλοκ δεδομένων, διαφορετικά ο αλγόριθμος συνεχίζει στο επόμενο μπλοκ δεδομένων με την ίδια ακριβώς μέθοδο. Στο σχήμα 2.4.2 (c) οι ανιχνεύσεις μέγιστων κορυφών συμβολίζονται με έναν αστερίσκο.

Ο αλγόριθμος συνεχίζει αυτήν την διαδικασία μέχρι να μην υπάρχουν άλλα μπλοκ δεδομένων οπότε και τερματίζει τη λειτουργία του **[13]**.

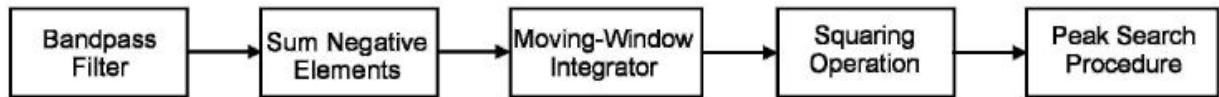


Εικόνα 2.4.2 : Αποτελέσματα πάνω στο σήμα της επιτάχυνσης του x-άξονα

2.5 Μέθοδος ανίχνευσης κορυφών βασισμένη σε συνδιασμένο σήμα 2 αξόνων

Η μέθοδος αυτή βασίζεται στον συνδιασμό 2 σημάτων επιτάχυνσης από διαφορετικούς άξονες και προϋποθέτει ότι η τοποθέτηση των επιταχυνσιομέτρων έχει γίνει στο πόδι του χρήστη. Στο παράδειγμα που θα δούμε θα χρησιμοποιήσουμε τα σήματα της επιτάχυνσης στους άξονες x και z όπως φαίνονται στο σχήμα 2.5.2.

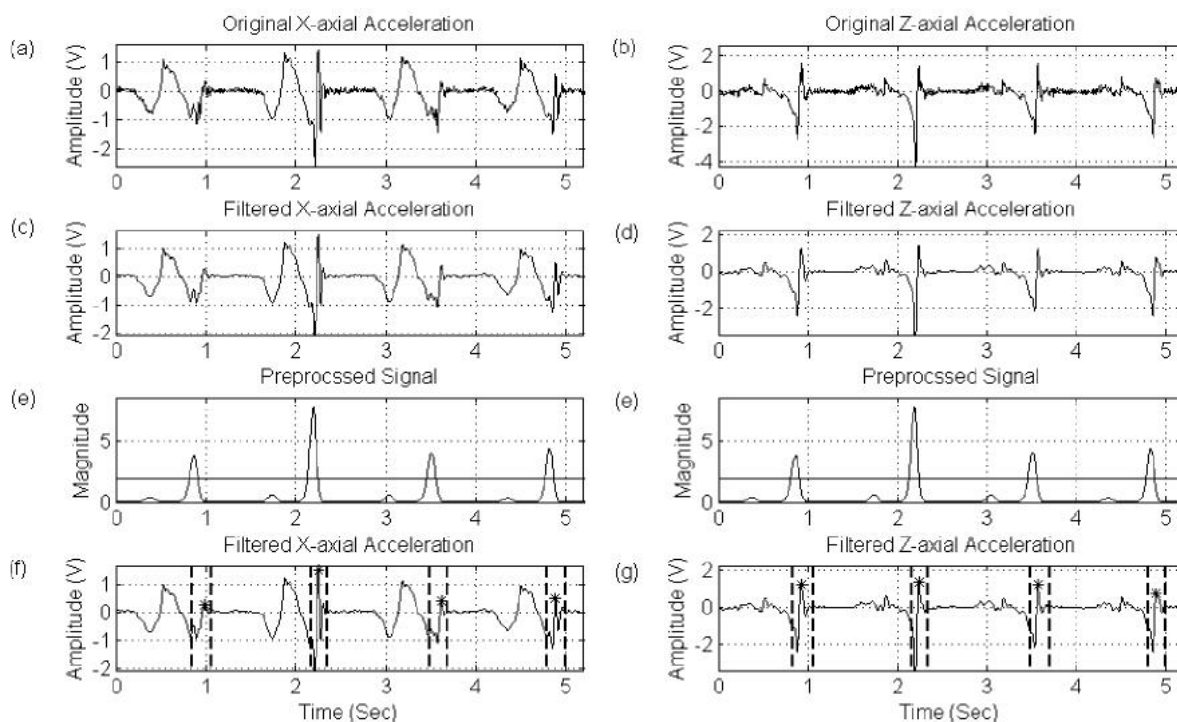
Βλέποντας το σχήμα 2.5.2 (a), παρατηρούμε ότι το αρνητικό μέρος του σήματος επιτάχυνσης στον x-άξονα συμπίπτει χρονικά με το αρνητικό μέρος του σήματος στον y-άξονα. Αυτήν ακριβώς την ιδιότητα εκμεταλεύεται η μέθοδος αυτή. Μερικές φορές, η αρνητική κορυφή του σήματος επιτάχυνσης είναι πιο φανερή από την θετική λόγω της απότομης επιβράδυνσης του ποδιού όταν αυτό χτυπάει στο έδαφος μετά την ολοκλήρωση ενός βήματος κάτι που κάνει την ανίχνευσή του πιο εύκολη.



Εικόνα 2.5.1 : Αλγόριθμος ανίχνευσης κορυφών σε συνδυασμένο σήμα 2 αξόνων

Για την σωστή εκτέλεση της μεθόδου αυτής απαιτείται μια προεπεξεργασία του αρχικού σήματος η οποία φαίνεται στο σχήμα 2.5.1 και αποτελείται από :

- ένα ζωνοπερατό (bandpass) φίλτρο, το οποίο εφαρμόζεται στα 2 σήματα ξεχωριστά και μειώνει την επίδραση των ακραίων μετρήσεων πάνω τους
- τα θετικά στοιχεία των 2 σημάτων μηδενίζονται και παραμένουν μόνο τα αρνητικά. Στη συνέχεια γίνεται πρόσθεση των 2 σημάτων επιταχύνσεως στοιχείο ανά στοιχείο. Στην ουσία προστίθονται όλες οι αρνητικές τιμές εφόσον οι θετικές έχουν εξαλειφθεί.
- έναν ολοκληρωτή «κυλιόμενου παραθύρου» (moving-window), ο οποίος εκτελείται πάνω στο νέο σήμα που παράχθηκε και βοηθάει στην ομαλοποίησή του.
- μια διαδικασία τετραγωνισμού (squaring), η οποία αυξάνει κατά πολύ τις μεγάλες τιμές του συνδυασμένου σήματος ενώ κατα λίγο τις μικρές. Αυτό βοηθάει στο να ξεχωρίσουμε πιο εύκολα τις μέγιστες αρνητικές τιμές οι οποίες και αντιστοιχούν στην στιγμή που το πόδι του χρήστη πέφτει με δύναμη στο έδαφος προκαλώντας μεγάλη επιβράδυνση σε σχέση με τις υπόλοιπες στιγμές του βήματος.



Εικόνα 2.5.2 : Αποτελέσματα προεπεξεργασίας της μεθόδου συνδιασμένου σήματος

Κατά τη διάρκεια της προεπεξεργασίας παράγεται ένα συνδιασμένο ομαλοποιημένο σήμα μεγάλου πλάτους όπως φαίνεται στο σχήμα 2.5.2 (e). Για την ανίχνευση των κορυφών στο προεπεξεργασμένο σήμα χρησιμοποιούμε ένα κατώφλι («threshold») το οποίο ισούται με το ένα τέταρτο ($\frac{1}{4}$) της μέγιστης τιμής της συνδιασμένης επιτάχυνσης που βρέθηκε. Το κατώφλι αυτό συμβολίζεται με μια ευθεία γραμμή στο σχήμα 2.5.2 (e).

Στη συνέχεια για την ανίχνευση των πραγματικών κορυφών ελέγχουμε το αρχικό σήμα και εντοπίζουμε τη μέγιστη τιμή επιτάχυνσης που βρίσκεται εντός του «διαστήματος ανίχνευσης κορυφών» το οποίο στην ουσία αντιστοιχεί στις χρονικές στιγμές κατά τις οποίες το προεπεξεργασμένο σήμα παίρνει τιμές μεγαλύτερες από το 0 (σχήμα 2.5.2 (f) - 2.5.2 (g)). **[13]**

ΚΕΦΑΛΑΙΟ 3

ΦΙΛΤΡΑΡΙΣΜΑ ΔΕΔΟΜΕΝΩΝ ΕΠΙΤΑΧΥΝΣΗΣ

Το φιλτράρισμα των ανεπεξέργαστων δεδομένων (raw data) επιτάχυνσης είναι μια απαραίτητη διαδικασία καθώς οι μετρήσεις των επιταχυνσιομέτρων περιέχουν πολλές φορές θόρυβο. Αυτό οφείλεται στον τρόπο βάρδησης του ανθρώπου και στην μορφολογία του εδάφους όπου περπατάει ο χρήστης.

Για το φιλτράρισμα των δεδομένων επιτάχυνσης μπορούμε να χρησιμοποιήσουμε διάφορες μεθόδους. Σε αυτό το κομμάτι θα αναλύσουμε τις πιο σημαντικές.

3.1 Φιλτράρισμα Μέσου όρου

Ένας από τους πιο απλούς αλλά συγχρόνως και αρκετά αποδοτικούς αλγορίθμους για το φιλτράρισμα δεδομένων επιτάχυνσης σε ανθρώπινο βηματισμό είναι το φιλτράρισμα Μέσου Όρου. Παρά την απλότητά του, το φίλτρο αυτό είναι βέλτιστο για μια πολύ κοινή δουλειά : την μείωση του τυχαίου θορύβου ενός σήματος.

Εφόσον ο ανθρώπινος βηματισμός έχει μια φυσική συχνότητα όχι περισσότερο από 3 Hz, μπορούμε να ισχυριστούμε ότι ένα φίλτρο Μέσου όρου μπορεί να ομαλύνει το σήμα της κάθετης επιτάχυνσης. Αν λοιπόν αναπαραστήσουμε το αρχικό σήμα με a_z και το φιλτραρισμένο με \bar{a}_z , ο τύπος του φίλτρου μπορεί να γραφεί κάπως έτσι :

$$\bar{a}_z = \frac{1}{n} \sum_{i=t-n}^t a_z(i)$$

Δηλαδή αυτό σημαίνει ότι το φιλτραρισμένο στοιχείο μιας χρονικής στιγμής t ισούται με το άθροισμα των n προηγούμενων στοιχείων που συλλέχθηκαν πριν την χρονική στιγμή t διαιρούμενο με n . Στην ουσία ισούται με τον μέσο όρο των τελευταίων n συλλεγένων στοιχείων, εξού και το όνομα «Φίλτρο μέσου όρου».

Η υλοποίηση του συγκεκριμένου φίλτρου θα μπορούσε να γίνει όπως φαίνεται στην εικόνα 3.1.1.

```
int z[1000];           //στοιχεία κάθετης επιτάχυνσης που συλλέχθηκαν (δεδομένα)  
int filtered[1000];   //φιλτραρισμένα στοιχεία (ζητούμενα)  
int n = 8;           //πόσα προηγούμενα στοιχεία θα λάβουμε υπόψη  
  
for (int i=0;i<=1000;i++) {  
    filtered[i] = 0;  
    for (int j=j-n;j<=i;j++)  
        if (j>=0)  
            filtered[i] += z[j];  
}
```

Εικόνα 3.1.1 : Παράδειγμα υλοποίησης φίλτρου μέσου όρου στην γλώσσα C

3.2 Το φίλτρο Kalman

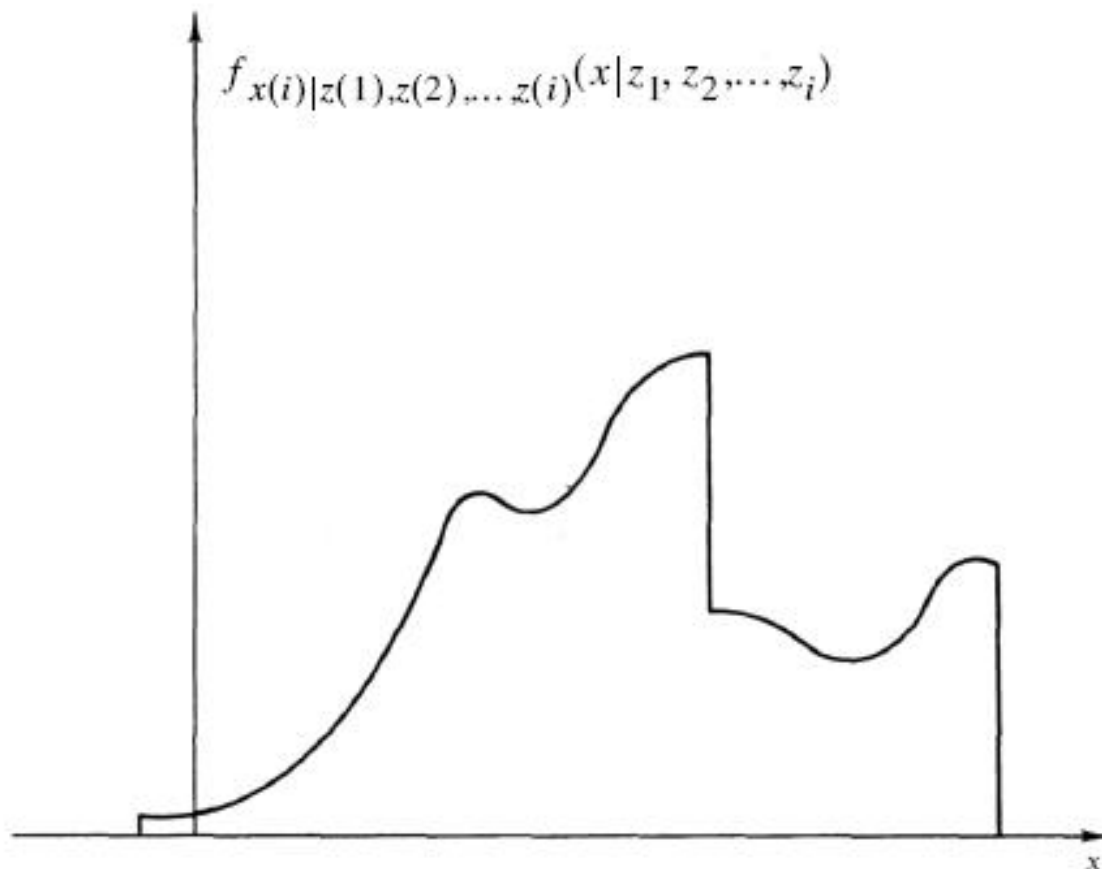
Το φίλτρο Kalman είναι ένας βέλτιστος αναδρομικός αλγόριθμος επεξεργασίας δεδομένων. Ένας παράγοντας που το κάνει βέλτιστο είναι ότι έχει τη δυνατότητα να συγχωνεύει οποιαδήποτε πληροφορία του δίνεται. Επεξεργάζεται κάθε διαθέσιμη μέτρηση από τους αισθητήρες, ανεξαρτήτως της ακρίβειάς τους, για να υπολογίσει την παρούσα τιμή των μεταβλητών που μας ενδιαφέρουν περισσότερο έχοντας όμως πάντα υπόψην εξωτερικούς ή εσωτερικούς παράγοντες που μπορεί να επιρρεάσουν το αποτέλεσμα καθώς και οτιδήποτε μπορεί να φανεί χρήσιμο πριν την εξαγωγή ενός αποτελέσματος όπως για παράδειγμα την ακριβή λειτουργία του συστήματος, την ευπάθεια των αισθητήρων συλλογής δεδομένων, μια στατιστική περιγραφή του θορύβου που εισέρχεται στο σύστημα κλπ.

Το γεγονός ότι το φίλτρο Kalman είναι αναδρομικό είναι πολύ σημαντικό γιατί στην ουσία αυτό σημαίνει ότι δεν απαιτείται να κρατούνται αποθηκευμένες στη μνήμη όλες οι προηγούμενες μετρήσεις και να ξαναεπεξεργάζονται από το σύστημα κάθε φορά που συλλέγεται μια νέα μέτρηση. Αυτό το στοιχείο του φίλτρου Kalman είναι επίσης πολύ σημαντικό. Σε γενικές γραμμές η λειτουργία του συγκεκριμένου φίλτρου είναι να συγχωνεύει όλες τις μετρήσεις για τις μεταβλητές που μας ενδιαφέρουν, συν οποιαδήποτε γνώση για τη ακριβή λειτουργία του συστήματος και των αισθητήρων μετρήσεων για να εξάγει μια προσέγγιση των μεταβλητών αυτών με τέτοιο τρόπο έτσι ώστε το εύρος του λάθους να ελαχιστοποιηθεί στατιστικά.

Για να εξετάσουμε την λειτουργία του φίλτρου Kalman μπορούμε να υιοθετήσουμε μια προσέγγιση της τεχνικής Bayes, δηλαδή να υπολογίσουμε την υπό συνθήκη εξαρτώμενη πιθανότητα πυκνότητας των μεταβλητών που μας ενδιαφέρουν ώντας βασισμένοι σε πραγματικά δεδομένα που συλλέγονται από τους αισθητήρες του συστήματος. Στο σχήμα 3.1 παρουσιάζεται η υπό συνθήκη εξαρτώμενη πιθανότητα πυκνότητας μιας ποσότητας x για μια τυχαία χρονική στιγμή i . Με $x(i)$ συμβολίζεται η τιμή της μεταβλητής x την χρονική στιγμή i ενώ η υπό συνθήκη εξαρτώμενη πιθανότητα πυκνότητας βασίζεται στην γνώση ότι ο αισθητήρας z την χρονική στιγμή 1 είχε πάρει την τιμή Z_1 , την χρονική στιγμή 2 είχε πάρει την τιμή Z_2 κλπ φτάνοντας μέχρι την παρούσα στιγμή όπου ανιχνεύθηκε ότι έχει την τιμή Z_i . Αυτή η πιθανότητα που μόλις περιγράψαμε συμβολίζεται με

$$f_{x(i)|z(1),z(2),\dots,z(i)}(x|z_1, z_2, \dots, z_i)$$

Για παράδειγμα, μπορούμε να θεωρήσουμε ότι το $x(i)$ συμβολίζει την θέση ενός οχήματος την χρονική στιγμή 1 και ότι το $z(j)$ είναι ένα διάνυσμα 2 διαστάσεων που περιέχει τις μετρήσεις από 2 διαφορετικούς αισθητήρες (πχ από 2 διαφορετικά ραντάρ) την χρονική στιγμή i . Σε αυτήν την περίπτωση, ένα σχήμα σαν αυτό που παρουσιάζεται στην εικόνα 3.1 είναι επαρκές για να περιγράψει κάθε διαθέσιμη πληροφορία για την τιμή της μεταβλητής x την χρονική στιγμή i , δηλαδή ποια είναι η πιθανότητα του $x(i)$.



Εικόνα 3.1 : Εξαρτώμενη πιθανότητα πυκνότητας φίλτρου Kalman

Το σχήμα που προκύπτει από την επεξεργασία του φίλτρου Kalman μας δίνει κάποιες επιπλέον πληροφορίες σχετικά με την βεβαιότητα που μπορούμε να έχουμε για ορισμένες τιμές του x την χρονική στιγμή i . Αν το σχήμα αυτό σχηματίζει μια στενή κορυφή σε κάποιο σημείο του, τότε το περισσότερο βάρος της πιθανότητας συγκεντρώνεται σε ένα μικρό εύρος τιμών του $x(i)$, ενώ αν σχηματίζεται ένα κλιμακωτό σχήμα με προοδευτική αύξηση και μείωση της τιμής του $x(i)$ σημαίνει ότι η πιθανότητα μοιράζεται σε ένα μεγαλύτερο εύρος τιμών του x κάτι που δείχνει ότι είμαστε λιγότερο σίγουροι για το ποια είναι η πραγματική τιμή της μεταβλητής που μας ενδιαφέρει.

Έχοντας πλέον υπολογίσει και αναπαραστήσει την υπό συνθήκη εξαρτημένη πιθανότητα πυκνότητας, το φίλτρο Kalman είναι σε θέση να υπολογίσει και την αντίστοιχη βέλτιστη τιμή που αντιστοιχεί στην πιθανότητα αυτή. Σε αυτό το σημείο υπάρχουν 3 πιθανές επιλογές για τον υπολογισμό της βέλτιστης τιμής της μεταβλητής μας :

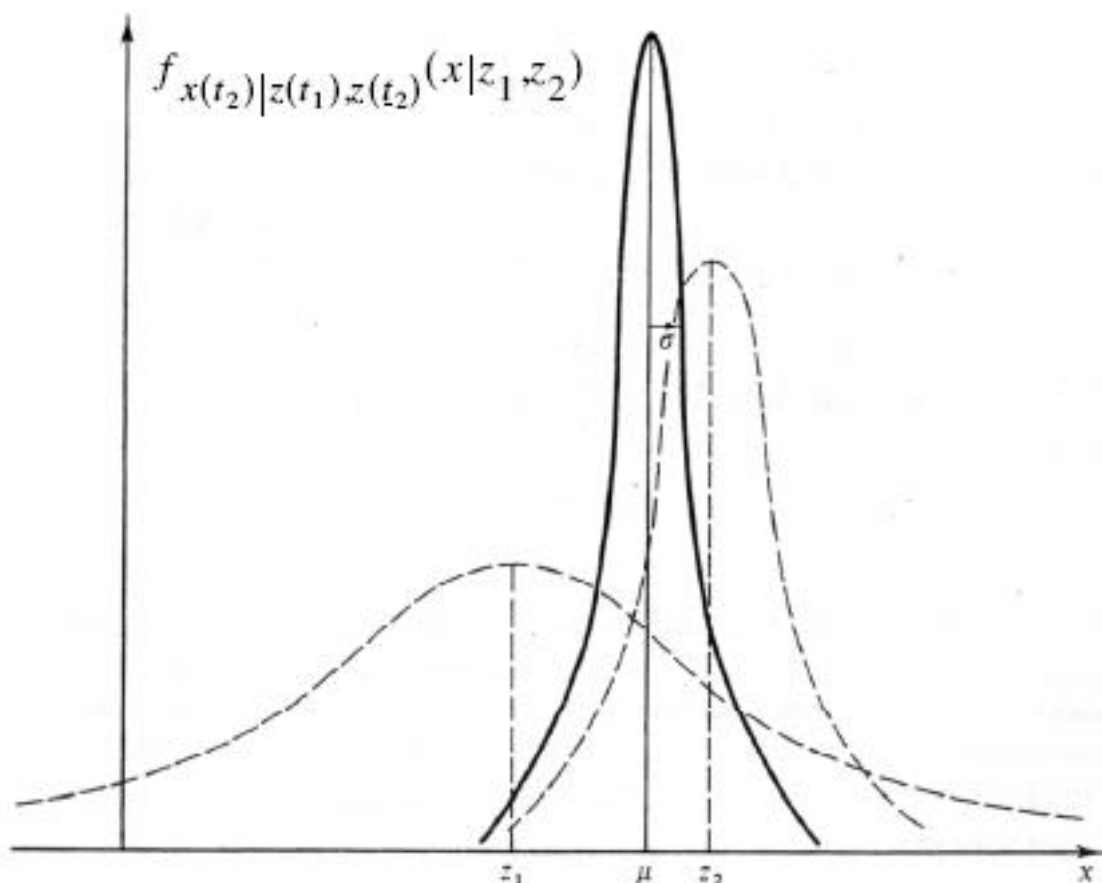
Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

- το μέσο, δηλαδή το κέντρο του πεδίου που καλύπτει η συνολική μάζα της πιθανότητας
- η μέγιστη κορυφή, δηλαδή η τιμή του x για την οποία έχουμε την μέγιστη πυκνότητα
- η διάμεσος, δηλαδή η τιμή του x για την οποία το $\frac{1}{2}$ του βάρους πιθανότητας μοιράζεται αριστερά της και το υπόλοιπο $\frac{1}{2}$ μοιράζεται δεξιά της.

Όταν το σύστημα πάνω στο οποίο γίνεται το φιλτράρισμα ακολουθεί ένα γραμμικό μοντέλο και οι θόρυβοι που εισέρχονται σε αυτό είναι λευκοί ή Gaussian θόρυβοι τότε οι 3 επιλογές που περιγράψαμε από πάνω συμπίπτουν, δίνοντάς μας στην ουσία μια μοναδική βέλτιστη τιμή για την μεταβλητή x .

Το φίλτρο Kalman μπορεί να αποδειχθεί ότι είναι το καλύτερο φίλτρο που μπορεί να χρησιμοποιηθεί υπό αυτές τις συνθήκες. Ακόμα κι αν κάποια από αυτές τις συνθήκες δεν ισχύει για κάποια συγκεκριμένο σύστημα, ακόμα και τότε το φίλτρο Kalman μπορεί να δώσει αρκετά ικανοποιητικά αποτελέσματα σε σχέση με άλλες τεχνικές φιλτραρίσματος.

Ας δούμε τώρα ένα παράδειγμα, για το πως συνδιάζονται μετρήσεις από διαφορετικές πηγές και πως φιλτράρονται από το φίλτρο Kalman αλλά και τι αποτέλεσμα μας δίνουν μετά από την επεξεργασία.



Σχήμα 3.2 : Υπό συνθήκη πυκνότητα βασισμένη σε μετρήσεις από 2 διαφορετικές πηγές (z_1, z_2)

Στο σχήμα 3.2 φαίνεται καθαρά το αποτέλεσμα του συνδιασμού δύο διαφορετικών μετρήσεων από δύο διαφορετικούς αισθητήρες για την μεταβλητή x τις χρονικές στιγμές t_1 και t_2 όπου $t_1 \cong t_2$. Για την μέτρηση από τον πρώτο αισθητήρα παρατηρούμε ότι το μέσο, η μέγιστη κορυφή και η διάμεσος που περιγράψαμε παραπάνω συμπίπτουν και ότι η βέλτιστη τιμή του x την χρονική στιγμή t_1 ισούται με z_1 έχοντας μια διακύμανση λάθους στον υπολογισμό αυτό που ισούται με $\sigma_x^2(t_1) = \sigma_{z_1}^2$. Ομοίως και για την μέτρηση από τον δεύτερο αισθητήρα, η οποία λαμβάνει χώρα ακριβώς μετά την πρώτη, θα έχουμε βέλτιστη τιμή την τιμή z_2 και διακύμανση λάθους $\sigma_x^2(t_2) = \sigma_{z_2}^2$. Έχοντας πλέον αυτά τα στοιχεία μπορούμε να προχωρήσουμε στον υπολογισμό της βέλτιστης τιμής της μεταβλητής x που μας ενδιαφέρει ως εξής:

$$\mu = \frac{\sigma_{z_2}^2}{(\sigma_{z_1}^2 + \sigma_{z_2}^2)} z_1 + \frac{\sigma_{z_1}^2}{(\sigma_{z_1}^2 + \sigma_{z_2}^2)} z_2$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2}$$

Όπου μ είναι το μέσο της εξαρτημένης πιθανότητας από τους δύο αισθητήρες και σ^2 η διακύμανση αυτής. Όπως παρατηρούμε και στο σχήμα 3.2, το σ είναι μικρότερο από το σ_{z_1} αλλά και από το σ_{z_2} . Αυτό σημαίνει ότι συνδιάζονται τις δύο μετρήσεις από τους δύο σιαφορετικούς αισθητήρες καταφέραμε μέσω του φίλτρου Kalman να βγάλουμε μια ακόμα πιο ακριβή μέτρηση με αρκετά πιο στενό εύρος λάθους.

Έτσι καταλήγουμε ότι την χρονική στιγμή t_2 η βέλτιστη προσέγγιση στην τιμή του x θα ισούται με $\hat{x}(t_2) = \mu$. Η εξίσωση όμως του $\hat{x}(t_2)$ μπορεί να ξαναγραφεί ως εξής:

$$\hat{x}(t_2) = z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} (z_2 - z_1)$$

Για να έρθουμε σιγά σιγά στην μορφή με την οποία χρησιμοποιείται σε υλοποιήσεις λαμβάνοντας υπόψη ότι :

$$\hat{x}(t_1) = z_1$$

$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)]$$

$$\text{Όπου } K(t_2) = \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2}$$

Στην ουσία με αυτούς τους 2 τελευταίους μαθηματικούς τύπους καταλήξαμε στο συμπέρασμα ότι η βέλτιστη προσέγγιση την χρονική στιγμή t_2 της τιμής του x ισούται

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

με την βέλτιστη προσέγγιση που υπήρχε πριν από αυτήν τη χρονική στιγμή, δηλαδή την $\hat{x}(t_1)$, στην οποία έχει προστεθεί μια διόρθωση που αντιστοιχεί σε μια βέλτιστα υπολογισμένη τιμή βάρους και πολλαπλασιάζεται με την διαφορά της δεύτερης μέτρησης z_2 από το $\hat{x}(t_1)$ [17].

ΚΕΦΑΛΑΙΟ 4

ΥΠΟΛΟΓΙΣΜΟΣ ΚΑΤΕΥΘΥΝΣΗΣ

Ο υπολογισμός της σωστής κατεύθυνσης σε ένα σύστημα «dead reckoning» είναι καθοριστικής σημασίας. Ακόμα και ένα φαινομενικά μικρό λάθος στον υπολογισμό της κατεύθυνσης μπορεί να αποκλίνει το χρήστη από την πραγματική του πορεία κατά ένα μεγάλο ποσοστό. Δεν είναι τυχαίο ότι η κύρια πηγή λαθών σε μια πλοήγηση βασισμένη στην τεχνική DR προέρχεται από την λειτουργία υπολογισμού της κατεύθυνσης του χρήστη.

Σε αυτό το κεφάλαιο θα αναλυθούν μερικές από τις πιο σημαντικές μεθόδους με τις οποίες επιτυγχάνεται μεγαλύτερη ακρίβεια σε αυτόν το τομέα.

4.1 Φιλτράρισμα κατεύθυνσης με κατωπερατό φίλτρο

Σε πρώτη φάση μπορεί να χρησιμοποιηθεί ένα χαμηλοπερατό φίλτρο, για παράδειγμα ένα φίλτρο Μέσου Όρου έτσι ώστε να απομακρυνθούν οι όποιες αναταραχές έχουν επιρρεάσει τις μετρήσεις της κατεύθυνσης του χρήστη και παράλληλα να τις ομαλοποιήσουν.

Αυτό όμως το φιλτράρισμα από μόνο του δεν είναι αρκετό να μας δώσει μια αξιόπιστη μέτρηση κατεύθυνσης.

4.2 Δυναμική διόρθωση κατεύθυνσης

Η δυναμική διόρθωση των μετρήσεων της κατεύθυνσης του χρήστη είναι απαραίτητη για να αντισταθμίζει τις λανθασμένες μετρήσεις κατά την διάρκεια του βαδίσματος. Γι αυτήν τη λειτουργία μπορούμε να χρησιμοποιήσουμε τα φιλτραρισμένα δεδομένα της επιτάχυνσης στους 3 άξονες (X,Y,Z) που έχουμε συλλέξει για να υπολογίσουμε την κλίση της επιφάνειας (pitch) .

Τα βήματα που πρέπει να ακολουθήσουμε για τον υπολογισμό μετρήσεων διορθωμένων από τυχόν κλίσεις (H_x, H_y, H_z) είναι :

- Αρχικά φιλτράρουμε τις μετρήσεις (X, Y, Z) που παίρνουμε κατευθείαν από την πυξίδα με ένα φίλτρο Μέσου Όρου έτσι ώστε να πάρουμε τις φιλτραρισμένες μετρήσεις τις οποίες θα συμβολίσουμε με $(\bar{X}, \bar{Y}, \bar{Z})$.
- Στη συνέχεια, αν δεν το έχουμε κάνει ήδη, φιλτράρουμε τις μετρήσεις από τα επιταχυνσιόμετρα και στους 3 άξονες (a_x, a_y, a_z) με ένα φίλτρο Μέσου Όρου και καταλήγουμε στις φιλτραρισμένες πλέον μετρήσεις τις οποίες θα συμβολίσουμε με $(\bar{a}_x, \bar{a}_y, \bar{a}_z)$.
- Υπολογίζουμε τις μεταβλητές W , οι οποίες θα παίξουν το ρόλο των βαρών σε κάθε άξονα, δηλαδή τι επίδραση θα έχει η μέτρηση σε κάθε άξονα στο τελικό αποτέλεσμα. Οι μεταβλητές αυτές υπολογίζονται ως εξής :

$$w_x = \bar{a}_x / |\bar{a}|$$

$$w_y = \bar{a}_y / |\bar{a}|$$

$$w_z = \bar{a}_z / |\bar{a}|$$

- Τέλος, υπολογίζουμε τις τελικές διορθωμένες τιμές της κατεύθυνσης με τους ακόλουθους τύπους :

$$H_z = w_x \cdot \bar{X} + w_y \cdot \bar{Y} + w_z \cdot \bar{Z}$$

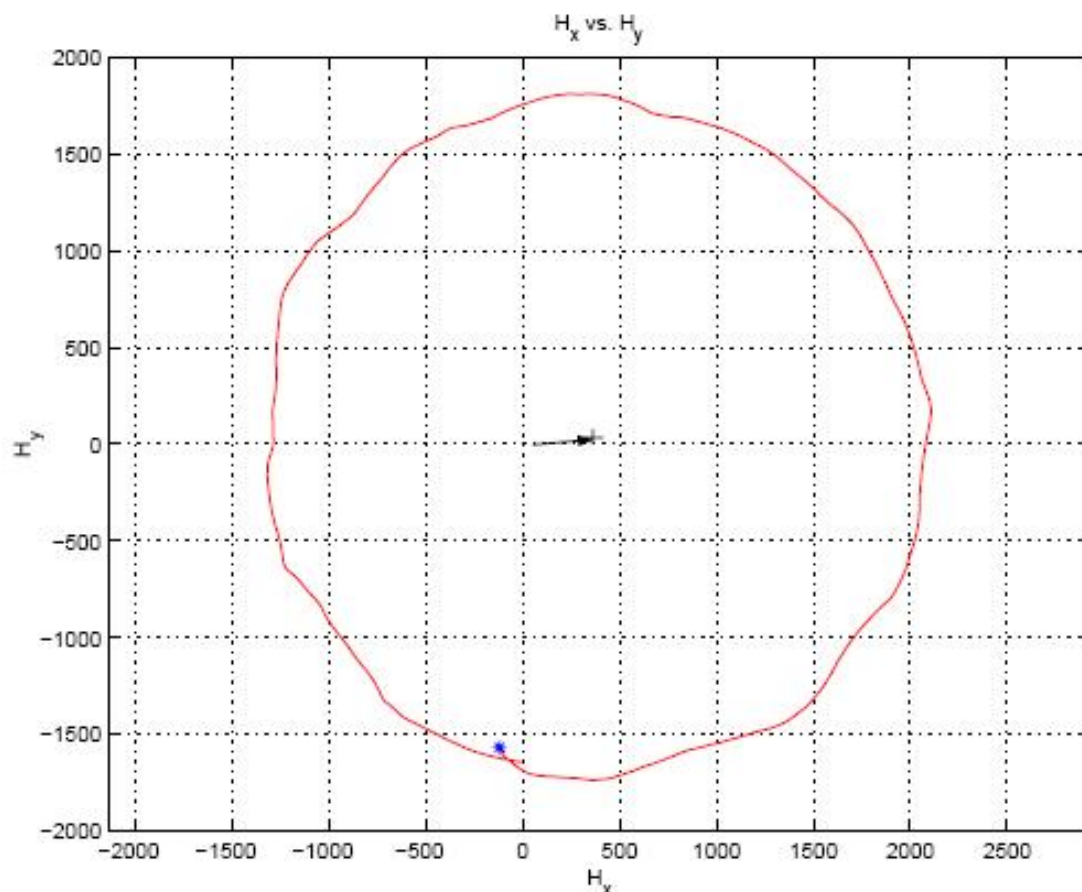
$$H_y = w_y \cdot \bar{Z} - w_z \cdot \bar{Y}$$

$$H_x = \bar{X} - w_x \cdot H_z$$

4.3 Στατική διόρθωση κατεύθυνσης

Η στατική διόρθωση της κατεύθυνσης του χρήστη είναι μια λειτουργία που μπορεί να βελτιώσει σημαντικά την ακρίβεια των μετρήσεων. Όταν ο αισθητήρας μιας μαγνητικής πυξίδας 2 αξόνων περιστραφεί στο οριζόντιο επίπεδο XY, δηλαδή αν ο χρήστης περπατήσει σε έναν μικρό κύκλο, χωρίς καμία αναταραχή τότε αν σχεδιάσουμε ένα

διάγραμμα των συνιστωσών H_x και H_y θα σχηματιστεί ένας κύκλος με κέντρο το σημείο (0,0). Όπως παρατηρούμε όμως και στο σχήμα 4.3.1, ακόμη και το ανθρώπινο σώμα μπορεί να διαταράξει τις μετρήσεις μιας μαγνητικής πυξίδας δίνοντας σαν αποτέλεσμα έναν παραμορφωμένο κύκλο, που μοιάζει περισσότερο με ένα ελλειψοειδές σχήμα, κάτι που σημαίνει ότι το ανθρώπινο σώμα αλλιώνει κατά κάποιο ποσοστό τις μετρήσεις της πυξίδας.



Εικόνα 4.3.1 : Η επίδραση του ανθρώπινου σώματος στις μετρήσεις μαγνητικής πυξίδας

Για να αντιμετωπιστεί αυτό το πρόβλημα είναι αναγκαίο να υπολογιστούν οι ακόλουθες τιμές

$$X_{sf} = \max \left\{ 1, \left(H_{y.95} - H_{y.05} \right) / \left(H_{x.95} - H_{x.05} \right) \right\}$$

$$Y_{sf} = \max \left\{ 1, \left(H_{x.95} - H_{x.05} \right) / \left(H_{y.95} - H_{y.05} \right) \right\}$$

$$X_{off} = 0.8 \times \left[\left(H_{x.95} - H_{x.05} \right) / 2 - H_{x.05} \right] \times X_{sf}$$

$$Y_{off} = 0.8 \times \left[\left(H_{y.95} - H_{y.05} \right) / 2 - H_{y.05} \right] \times Y_{sf}$$

Όπου οι τιμές $H_{x.95}$ και $H_{x.05}$ αντιστοιχούν στο 95% και 5% της τιμής της συνιστώσας H_x (αντιστοίχως και τα $H_{y.95}$, $H_{y.05}$ για την συνιστώσα H_y), οι τιμές X_{sf}, Y_{sf} είναι παράγοντες διαβάθμισης που χρησιμεύουν στο να μετατρέψουν το ελλειψοειδές σχήμα που προκύπτει σε κύκλο και οι τιμές X_{off}, Y_{off} είναι παράγοντες αντιστάθμισης για να θέσουν το κύκλο που σχηματίζεται γύρω από τη σημείο (0,0),

Τέλος, χρησιμοποιούμε τις τιμές που μόλις υπολογίσαμε για να βρούμε την βελτιωμένη και πιο ακριβή τιμή της κατεύθυνσης του χρήστη η οποία προκύπτει από τις σχέσεις :

$$\bar{H}_x = X_{sf} \times H_x + X_{off}$$

$$\bar{H}_y = Y_{sf} \times H_y + Y_{off}$$

$$Αζιμούθιο = \text{atan} 2 \left(\bar{H}_y, \bar{H}_x \right)$$

Όπου η πράξη $\text{atan}2()$ υπολογίζει την τιμή της εφαπτομένης καμπύλης του τόξου της τιμής \bar{H}_y / \bar{H}_x χρησιμοποιώντας τα πρόσημα και των δύο παραμέτρων για να καθορίσει το τεταρτημόριο στο οποίο βρίσκεται η ζητούμενη τιμή του αζιμούθιου.

4.4 Απόκλιση από τον πραγματικό Βορρά

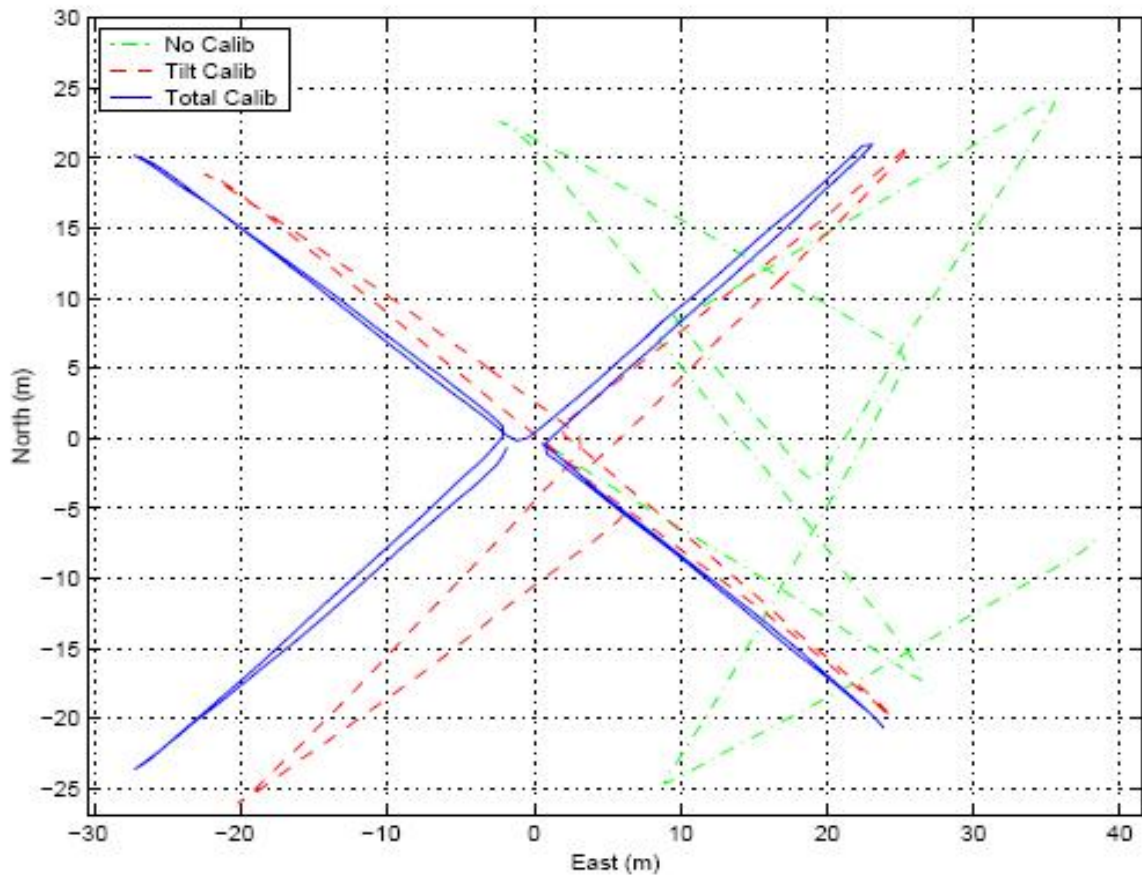
Ένα ακόμα πολύ σημαντικό βήμα που πρέπει να γίνει για την διόρθωση των μετρήσεων από μια μαγνητική πυξίδα είναι η προσθήκη της γωνίας απόκλισης από τον πραγματικό Βορρά [14].

Κάθε μαγνητική πυξίδα, επηρεάζεται από τα μαγνητικά πεδία της Γης ανάλογα με την τοποθεσία και τον χρόνο κατά τον οποίο γίνεται η μέτρηση. Ως εκ τούτου, ο μαγνητικός Βορράς διαφέρει από τον πραγματικό Βορρά σε διαφορετικά σημεία της Γης κατά κάποια τιμή η οποία δεν είναι σταθερή αλλά εξαρτάται από την τοποθεσία. Για να έχουμε ακόμα πιο αξιόπιστα αποτελέσματα, η τιμή αυτή θα πρέπει να λαμβάνεται υπόψη σε κάθε μέτρηση και να αφαιρείται ή να προστίθεται αναλόγως. Για να γίνει αυτό θα πρέπει να συμβουλευτούμε κάποια βάση δεδομένων που θα περιέχει την γωνία απόκλισης από τον πραγματικό Βορρά για το σημείο και τον χρόνο κατά τον οποίο λαμβάνει χώρα η μέτρηση.

Για παράδειγμα, χρησιμοποιώντας την βάση του Εθνικού Γεωφυσικού κέντρου των Ηνωμένων Πολιτειών της Αμερικής μπορούμε να πληροφορηθούμε ότι ο μαγνητικός από τον πραγματικό Βορρά στην Νέα Υόρκη τον Οκτώβριο του 2008 έχει μια γωνία απόκλισης της τάξεως $13^{\circ} 7'$ μοιρών προς την Δύση και μεταβάλλεται κατά $0^{\circ} 1'$ μοίρες προς την Ανατολή κάθε έναν χρόνο [15] [16]. Όπως καταλαβαίνουμε λοιπόν, η διαφορά αυτή είναι αρκετά σημαντική και πρέπει να ληφθεί υπόψη για να έχουμε αξιόπιστα αποτελέσματα.

4.5 Πλεονεκτήματα διόρθωσης κατευθύνσεως

Στο σχήμα 4.5.1 φαίνονται καθαρά τα αποτελέσματα μετά από την διαδικασία διόρθωσης των μετρήσεων μιας μαγνητικής πυξίδας με τους τρόπους που περιγραφηκαν σε αυτό το κεφάλαιο. Σύμφωνα με μετρήσεις, αν χρησιμοποιηθούν οι τεχνικές που περιγράφηκαν στα υποκεφάλαια 4.1 και 4.2, μπορεί να επιτευχθεί μια μέγιστη απόκλιση κατευθύνσεως του χρήστη της τάξεως της $1,1^{\circ}$ μοίρας την στιγμή που αν δεν υπήρχε κάποια διαδικασία διόρθωσης θα είχαμε απόκλιση της τάξεως των $6,8^{\circ}$ μοιρών [9].



Εικόνα 4.5.1 : Η επίδραση της διόρθωσης κατεύθυνσης

ΚΕΦΑΛΑΙΟ 5

ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΣΤΑΣΗΣ ΒΗΜΑΤΟΣ

Ας θεωρήσουμε ότι ένα σύστημα πλοήγησης στέλνει σήμα ότι ανιχνεύθηκε βήμα. Σε αυτήν την περίπτωση θα πρέπει να υπάρχει μια μέθοδος που θα υπολογίζει για το δεδομένο βήμα την απόσταση που διανύθηκε. Σε αυτό το κεφάλαιο θα αναλύσουμε τα υπέρ και τα κατά διαφόρων τέτοιων μεθόδων.

5.1 Καθορισμένη απόσταση βήματος

Η πιο απλή μέθοδος για τον υπολογισμό της απόστασης του βήματος ενός χρήστη είναι πολύ απλά να μην γίνεται υπολογισμός της απόστασης αυτής σε πραγματικό χρόνο αλλά αντί αυτού να χρησιμοποιείται μια σταθερά η οποία θα έχει οριστεί από πριν σαν προεπιλογή [22]. Για παράδειγμα θα μπορούσαμε να θεωρήσουμε δεδομένο ότι ένας χρήστης διανύει κατά μέσο όρο 0,5 μέτρα σε κάθε του βήμα. Έτσι ο αλγόριθμός μας θα μπορούσε να προσθέτει στην απόσταση που έχει διανυθεί από έναν χρήστη 0,5 μέτρα κάθε φορά που θα ανιχνεύεται ένα βήμα. Η μέθοδος αυτή, αν και είναι πολύ εύκολη στην υλοποίηση, δεν προτείνεται για λόγους όπως:

1. Αν και μπορεί να θεωρηθεί ένας μέσος όρος απόστασης βήματος, σε καμία περίπτωση δεν μπορούμε να πάρουμε ως δεδομένο ότι με κάθε βήμα του ο χρήστης διανύει την ίδια απόσταση.
2. Για κάθε χρήστη, η μέση απόσταση βήματος μπορεί να διαφέρει ανάλογα με το ύψος του αλλά και με τον τρόπο βαδίσεώς του. Για παράδειγμα κάποιος χρήστης μπορεί να διανύει μεγαλύτερη απόσταση με κάθε δεξιό βήμα απ' ότι με κάθε αριστερό. Αυτό το πρόβλημα βέβαια μπορεί να λυθεί αν πριν από την έναρξη του αλγορίθμου κάθε χρήστης εισάγει έναν ατομικό μέσο όρο βήματος.
3. Κάθε άνθρωπος βαδίζει διαφορετικά σε ευθεία απ'ότι σε κάποιο κεκλιμένο επίπεδο. Έτσι δεν μπορούμε να θεωρήσουμε ότι η απόσταση βήματος θα είναι ίδια πάνω σε οποιοδήποτε δάπεδο και κλίση δαπέδου.

Λαμβάνοντας υπόψην όλους αυτούς τους παράγοντες καταλήγουμε στο συμπέρασμα ότι ο τρόπος αυτός δεν μπορεί να προταθεί για σοβαρές υλοποιήσεις της τεχνικής «dead reckoning» καθώς με την πάροδο του χρόνου και με κάθε ανιχνεύσιμο βήμα συσσορεύεται λάθος στην τελική μας μέτρηση κάτι που τελικά οδηγεί σε μεγάλη απόκλιση από την πραγματική θέση του χρήστη.

5.2 Υπολογισμός απόστασης βήματος με βάση την κάθετη επιτάχυνση

Για πιο ακριβή υπολογισμό της απόστασης που διανύει ένας χρήστης θα πρέπει χρησιμοποιηθούν τεχνικές που συμβαίνουν σε πραγματικό χρόνο και μπορούν να προσαρμοστούν σε κάθε χρήστη ξεχωριστά. Γι αυτόν το λόγο είναι αναγκαίο να αναλυθεί η δυναμική του σώματος του χρήστη, και ειδικά της συμπεριφοράς των ποδιών όταν αυτός περπατάει.

Ένα πολύ σημαντικό στοιχείο όσον αφορά το περπάτημα του ανθρώπου είναι ότι όταν κάποιος θέλει να κάνει ένα μεγαλύτερο βήμα, εκτός του ότι τα πόδια του τεντώνονται περισσότερο, κινούνται και πιο γρήγορα για λόγους ισορροπίας του σώματος. Άρα κάτι τέτοιο συνεπάγεται ότι η επιτάχυνση του ποδιού του χρήστη είναι μεγαλύτερη όταν γίνεται ένα μεγαλύτερο βήμα, και γενικά παρατηρείται μεγαλύτερη διαφορά στον κάθετο άξονα της επιτάχυνσης απ'ότι στην κατεύθυνση της κίνησης κάτι που μας οδηγεί στον να προτιμούμε τα δεδομένα που συλλέγουμε από τον κάθετο άξονα για την λειτουργία αυτή [18]. Έτσι, έχοντας αυτήν την υπόθεση ως δεδομένη, μπορούμε να χρησιμοποιήσουμε τις ακραίες, maximum και minimum, τιμές της κάθετης επιτάχυνσης του ποδιού κατά την διάρκεια ενός βήματος για να διαβαθμίσουμε την σταθερά που αντιπροσωπεύει τον μέσο όρο που διανύει ένας άνθρωπος ανάλογα με το πόσο γρήγορα επιταχύνθηκε το πόδι του.

Για παράδειγμα αν θεωρήσουμε ότι ο μέσος όρος του βήματος ενός χρήστη είναι η σταθερά K , και ότι η μέγιστη και ελάχιστη επιτάχυνση που μετρήθηκε κατά την διάρκεια ενός βήματος είναι A_{max} και A_{min} αντίστοιχα τότε θα μπορούσαμε να πάρουμε μια αρκετά καλή προσέγγιση της απόστασης που διανύθηκε κατά αυτό το βήμα μέσω του τύπου [9] :

$$\text{απόσταση_βήματος} \approx \sqrt[4]{A_{max} - A_{min}} * K$$

Παρόλο που η σταθερά K εξαρτάται από τον κάθε χρήστη μελέτες έχουν δείξει ότι δεν διαφέρει πολύ από χρήστη σε χρήστη όπως φαίνεται και στον πίνακα 5.2.1. Παρατηρούμε ότι για διαφορετικά ύψη ποδιού η σταθερά K δεν αλλάζει πολύ. Έτσι εμπειρικά μπορεί να χρησιμοποιηθεί μια τιμή της τάξεως των 0,55 μέτρων σαν μέσος όρος ενός ανθρώπινου βήματος δίνοντάς μας αρκετά αξιόπιστα αποτελέσματα [19].

Πίνακας 5.2.1 : Η σταθερά K για διαφορετικούς χρήστες

Χρήστης	A	B	Γ
Μήκος ποδιού	0.94	1.06	1.08
K	0.50	0.55	0.57

Επίσης από πειραματικές δοκιμές έχει αποδειχτεί ότι η σταθερά K μπορεί να δώσει αξιόπιστα αποτελέσματα με την συγκεκριμένη μέθοδο για όλους τους τρόπους βαδίσματος, από πολύ αργά έως πολύ γρήγορα, χωρίς να υπεισέρχεται μεγάλο λάθος στην τελική μας μέτρηση [19]. Αυτό φαίνεται και από τον πίνακα 5.2.2 όπου παρουσιάζεται η απόκλιση που μετρήθηκε σε μια απόσταση 75 μέτρων που διανύθηκε από τρεις διαφορετικούς χρήστες.

Πίνακας 5.2.2 : Απόκλιση τριών χρηστών σε μια απόσταση 75 μέτρων

Χρήστης/Ταχύτητα	Κανονικά	Αργά	Γρήγορα
A (K=0.50)	75.3	74.2	73.5
B (K=0.55)	75.5	74.6	73.3
Γ (K=0.57)	74.9	76.0	72.8

5.3 Υπολογισμός απόστασης βήματος με βάση την διάρκεια

Ένα άλλο χαρακτηριστικό του ανθρώπινου βήματος είναι η διάρκειά του, δηλαδή από την στιγμή που το ένα μας πόδι, που βρίσκεται πίσω από το άλλο, φύγει από το έδαφος και μετά από κάποιο διάστημα διαγράψει μια πορεία χρησιμοποιώντας το άλλο πόδι σαν στήριγμα για να καταλήξει μπροστά του και να ακουμπήσει στο έδαφος, χρονικό διάστημα καλείται διάρκεια βήματος.

Από μελέτη προκύπτει ότι υπάρχει μια σχέση ανάμεσα σημαντικό χαρακτηριστικό της διάρκειας ενός βήματος είναι ότι μπορεί να χρησιμοποιηθεί για να διευκρινιστεί η ταχύτητα του βήματος που κάνει ο χρήστης. Όσο μικρότερη είναι η διάρκεια ενός βήματος τόσο μεγαλύτερη θεωρείται η ταχύτητά του. Έτσι στο σύστημά μας μπορούμε να εισάγουμε προ-υπολογισμένες τιμές μέγιστης και ελάχιστης ταχύτητας του μέσου ανθρώπινου βήματος οι οποίες μπορούν να αντιστοιχηθούν με συγκεκριμένες αποστάσεις βήματος και να χρησιμοποιηθούν για την βελτίωση του υπολογισμού της απόστασης.

Από μελέτη έχει αποδειχθεί ότι η ταχύτητα του βήματος ενός χρήστη είναι ανάλογη της απόστασης που διανύεται και του ρυθμού του[20]. Αυτό ουσιαστικά σημαίνει ότι όσο πιο γρήγορα περπατάει ένας άνθρωπος τόσο μεγαλύτερη είναι η απόσταση που διανύει και τόσο μεγαλύτερη είναι η συχνότητα ανίχνευσης βήματος. Η ταχύτητα ενός βήματος πρέπει να λαμβάνεται πάντα υπόψη από έναν αλγόριθμο που θέλει να θεωρείται αξιόπιστος καθώς το ανθρώπινο βήμα διαφέρει κατά πολύ όταν κάποιος περπατάει αργά απ'ότι όταν περπατάει γρήγορα. Έτσι είναι προφανές ότι αν δεν ληφθεί υπόψη αυτή η παράμετρος, το λάθος που θα εισέλθει στην μέτρηση είναι ικανό να «κρεμάσει» το σύστημα μας.

5.4 Προσαρμοστικός αλγόριθμος για κάθε χρήστη

Για να βελτιωθεί ακόμα περισσότερο, και μάλιστα κατά πολύ περισσότερο, η ακρίβεια του υπολογισμού της απόστασης ενός βήματος μπορεί να υλοποιηθεί ένα βοηθητικό υποσύστημα το οποίο θα εκτελείται πριν χρησιμοποιήσει ο χρήστης το κύριο σύστημα για την πλοήγησή του, και που θα έχει σαν στόχο να προετοιμάσει το κύριο σύστημα και να το προσαρμόσει στον εκάστοτε χρήστη. Αυτό το υποσύστημα θα καθοδηγεί τον κάθε χρήστη να προβεί σε ορισμένες ενέργειες έτσι ώστε να μπορέσει να ανιχνεύσει τα

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

χαρακτηριστικά της κίνησής του σε διαφορετικές περιστάσεις και συνθήκες και να δει τι αποστάσεις διανύει με τα βήματά του σε γνωστές καταστάσεις.

Για παράδειγμα ένα τέτοιο υποσύστημα θα μπορούσε να περιέχει τις εξής καθοδηγήσεις προς τον χρήστη:

1. Διάνυσε 20 μέτρα σε ευθεία με κανονικό βηματισμό.
2. Διάνυσε 20 μέτρα σε ευθεία με αργό βηματισμό.
3. Διάνυσε 20 μέτρα σε ευθεία με γρήγορο βηματισμό.
4. Επανάλαβε τα βήματα 1 έως 3 αλλά σε κεκλιμένο ανηφορικό επίπεδο 45 μοιρών.
5. Επανάλαβε τα βήματα 1 έως 3 αλλά σε κεκλιμένο κατηφορικό επίπεδο 45 μοιρών.

Αυτή η λίστα μπορεί να μεγαλώσει ακόμα περισσότερο για συλλογή περισσότερων πληροφοριών και για μεγαλύτερη βελτίωση της ποιότητας των αποτελεσμάτων.

Με την ολοκλήρωση των παραπάνω ενεργειών πλέον θα έχουμε στη διάθεσή μας πληροφορίες όπως:

1. πόσα βήματα χρειάστηκαν για να διανυθούν 20 μέτρα σε κανονικό βηματισμό;
2. ποιος είναι ο μέσος όρος της απόστασης ενός βήματος σε κανονικό βηματισμό;
3. ποια είναι η ταχύτητα του χρήστη σε κανονικό βηματισμό;
4. ποιες είναι η αντίστοιχες τιμές σε αργό και γρήγορο βηματισμό;
5. ποιος είναι οι αντίστοιχες τιμές σε κεκλιμένα επίπεδα;

Το πιο σημαντικό στοιχείο όμως αυτής της διαδικασίας είναι ότι οι πληροφορίες που συλλέγουμε αφορούν έναν συγκεκριμένο χρήστη, έτσι δεν υπάρχει κίνδυνος να εισέλθει λάθος στις μετρήσεις μας από γενικευμένες παραμέτρους καθώς τα πάντα είναι πλέον προσαρμοσμένα στον χρήστη που μας ενδιαφέρει. Έτσι σε αυτό το σημείο το κύριο σύστημά μας είναι σε πολύ πιο πλεονεκτική θέση απ'ότι θα ήταν ένα σύστημα που δεν θα είχε κάποια προετοιμασία προσαρμογής. Οι πληροφορίες δε που συλλέγονται είναι σε θέση να βοηθήσουν το σύστημα να ανιχνεύσει ανά πάσα στιγμή σε ποια κατάσταση

βρίσκεται ο χρήστης και να υπολογίσει προσαρμοστικά την απόσταση του βήματος του με σημαντική ακρίβεια.

5.5 Ένας πιο γενικός τύπος

Ένας πιο γενικός τύπος προτείνεται στο [21] όπου μετά από πειράματα παρατηρήθηκαν οι εξής ιδιαιτερότητες στο ανθρώπινο βάδισμα:

- Κάποιοι άνθρωποι παρουσίασαν έναν ιδανικό βηματισμό, δηλαδή σταθερό βηματισμό, ανεξαρτήτως της κλίσης και του τύπου του εδάφους όπου έγινε το πείραμα. Αυτή η ιδιαιτερότητα παρουσιάστηκε κυρίως σε αθλητικούς τύπους ανθρώπων.
- Δεν υπάρχει πάντα συμμετρία μεταξύ του αριστερού και του δεξιού βήματος ενός ανθρώπου και είναι σπάνιο να έχουν ακριβώς το ίδιο μήκος για τον ίδιο χρήστη. Για την ακρίβεια παρατηρήθηκαν συχνά διαφορές της τάξεως των 5 εκατοστών (cm).
- Παρατηρήθηκε ένας συσχετισμός μεταξύ της απόστασης που διανύεται σε ένα ανθρώπινο βήμα και της συχνότητας βηματισμού της τάξεως του 0.6 και πάνω (>0.6).

Έτσι λοιπόν, λαμβάνοντας υπόψη τις παρατηρήσεις αυτές, καταλήγουμε σε έναν τύπο για την απόσταση βήματος που έχει ως εξής:

$$\text{Step length} = A + B \cdot \text{Freq} + C \cdot \text{Var} + w$$

Όπου

- A,B,Γ : παράμετροι υπολογισμένες από την γραμμική απόκλιση,
- Freq : η πραγματική συχνότητα βήματος
- Var : η διασπορά (variance) του σήματος
- w : Ο Gaussian θόρυβος $N(0,\sigma)$

ΚΕΦΑΛΑΙΟ 6

ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΥΠΟΛΟΓΙΣΜΟΥ ΘΕΣΗΣ ΜΕ ΤΗ ΜΕΘΟΔΟ DR

Στα πλαίσια αυτής της πτυχιακής εργασίας υλοποιήθηκε εξ αρχής ένα σύστημα υπολογισμού θέσης βασισμένο σε μεθόδους που περιγράφονται παραπάνω.

Αρχικά θα περιγραφεί με ποιον τρόπο έγινε ο σχεδιασμός του συστήματος και στην συνέχεια θα ακολουθήσει μια λεπτομερής περιγραφή της υλοποίησης σε επίπεδο κώδικα όπου θα περιγραφεί κάθε μια λειτουργία ξεχωριστά και κατανοητά. Αμέσως μετά θα παρουσιαστούν ορισμένες μετρήσεις που συλλέχθηκαν από το σύστημα που υλοποιήθηκε καθώς και μερικά γραφήματα που θα δείχνουν την απόδοση του έτσι ώστε ο αναγνώστης να έχει μια πιο ολοκληρωμένη εικόνα γι αυτό. Τέλος θα ακολουθήσει μια παράγραφος με τον απολογισμό λειτουργίας του συστήματος και η παράγραφος αυτή θα ολοκληρωθεί με κάποιες προτάσεις βελτίωσης που θεωρήθηκε ότι θα ήταν χρήσιμο να ενσωματωθούν στο σύστημα.

6.1 Επιλογή μεθόδων και συσκευών

Το βασικότερο σημείο πριν την υλοποίηση ενός συστήματος υπολογισμού θέσης είναι να αποφασιστεί ποιες μέθοδοι θα χρησιμοποιηθούν. Αυτή η επιλογή πρέπει να γίνει ύστερα από προσεκτική μελέτη έτσι ώστε τα συστήματα που θα χρησιμοποιήσουμε να μπορούν να ανταπεξέλθουν στις απαιτήσεις των αλγορίθμων που θα υλοποιηθούν.

Το σύστημα που αναπτύχθηκε για την συγκεκριμένη πτυχιακή σχεδιάστηκε στην γλώσσα C# του πακέτου εφαρμογών της Microsoft, .NET και έτρεξε πάνω σε ένα PDA με Windows Mobile 6.0 . Γι αυτόν το λόγο θεωρήθηκε ότι πρέπει να υλοποιηθούν αλγόριθμοι που δεν έχουν μεγάλες απαιτήσεις σε μνήμη και επεξεργαστική ισχύ καθώς αυτές οι δύο παράμετροι είναι περιορισμένες στις συσκευές PDA.

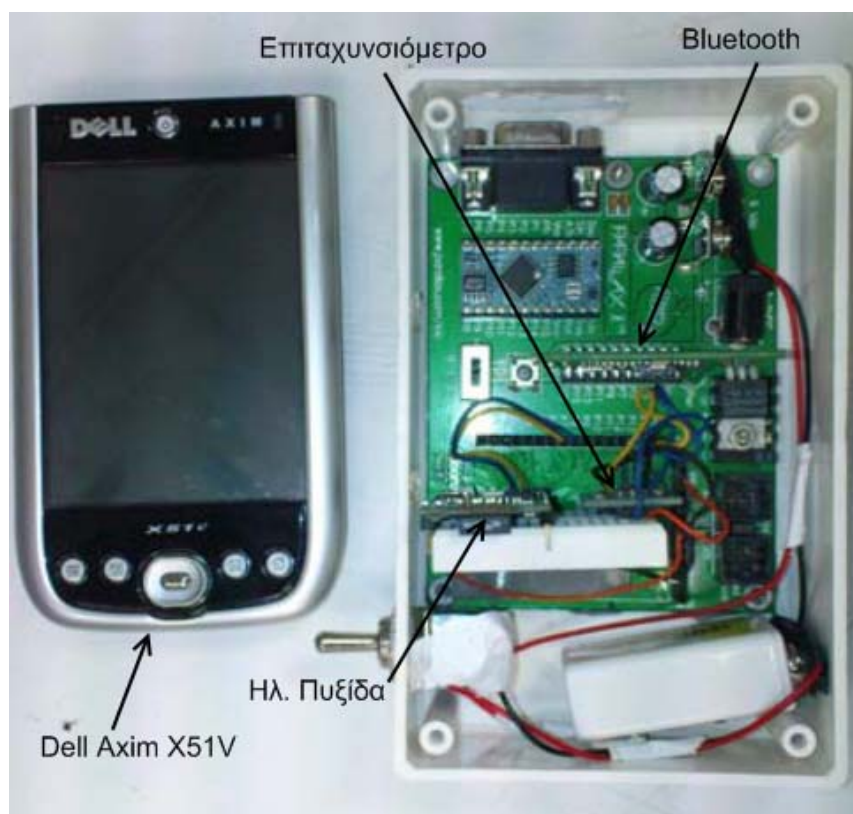
Οι συσκευές και αισθητήρες που χρησιμοποιήθηκαν ήταν οι εξής (εικόνα 6.1.1):

- Ηλεκτρονική πυξίδα 3 αξόνων Honeywell 3300 : Η διακριτική ικανότητα των $\pm 1.4\text{mg}$ των αισθητήρων επιτάχυνσης αντιστοιχεί σε αρκετά χαμηλά επίπεδα θορύβου για την

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

ανίχνευση βηματισμών. Η μέγιστη συχνότητα των 40 Hz για τους αισθητήρες επιτάχυνσης υπερκαλύπτει την απαιτούμενη συχνότητα δειγματοληψίας των 20 Hz.

- Επιταχυνσιόμετρο 3 αξόνων Hitachi H48C : Η ακρίβεια της διάταξης κατεύθυνσης κίνησης μέσω της μέτρησης με διάταξη τριών αξόνων του μαγνητικού πεδίου της γης ελέγχθηκε τόσο στατικά (ακίνητος χρήστης) όσο και δυναμικά (κινούμενος χρήστης) και βρέθηκε να ακολουθεί τις προδιαγραφές του κατασκευαστή (3 deg RMS με διακριτική ικανότητα 0.1 deg και για κλίσεις της όλης διάταξης από την κατακόρυφο $\pm 30^\circ$)
- Μονάδα Bluetooth
- PDA Dell Axim X51V : Microsoft Windows Mobile 5.0 software, Επεξεργαστής Intel XScale PXA270 Processor at 624MHz, Έγχρωμη οθόνη TFT VGA 3.7" με ανάλυση 640 x 480, Ενσωματωμένος Intel® 2700G multimedia accelerator με 16MB video memory, Ενσωματωμένες ασύρματες τεχνολογίες 802.11b και Bluetooth, 64MB SDRAM και 256MB Flash ROM, αποσπώμενη επαναφορτιζόμενη μπαταρία



Εικόνα 6.1.1 : Συσκευές και αισθητήρες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος πλοήγησης με την τεχνική DR

Στον τομέα της επιλογής μεθόδων τώρα χρησιμοποιήθηκαν οι εξής:

- Τα δεδομένα που συλλέγονταν από τους αισθητήρες ρυθμίστηκαν έτσι ώστε να γίνεται μια δειγματοληψία της τάξεως των 20 Hz, δηλαδή να συλλέγουμε 20 μετρήσεις το δευτερόλεπτο όπως περιγράφηκε στο κεφάλαιο 2.
- Για το φιλτράρισμα των δεδομένων επιτάχυνσης υλοποιήθηκε η μέθοδος που περιγράφεται στο κεφάλαιο 3.1, το φιλτράρισμα «Μέσου όρου» όπως επίσης και για το φιλτράρισμα της κατευθύνσεως.
- Η τοποθέτηση των αισθητήρων αποφασίστηκε να γίνει στην μέση του χρήστη (Εικόνα 6.1.2) υπό την μορφή ζώνης και ως εκ τούτου η μέθοδος ανίχνευσης κορυφών (η οποία περιγράφεται στο κεφάλαιο 2.1) θεωρήθηκε ως η καλύτερη λύση για την υλοποίηση της μονάδος ανίχνευσης βήματος.
- Τέλος, για τον υπολογισμό απόστασης βήματος του χρήστη υλοποιήθηκε η μέθοδος που βασίζεται στην κάθετη επιτάχυνση και που περιγράφεται στο κεφάλαιο 5.2.



Εικόνα 6.1.2 : Τοποθέτηση αισθητήρων στη μέση του χρήστη

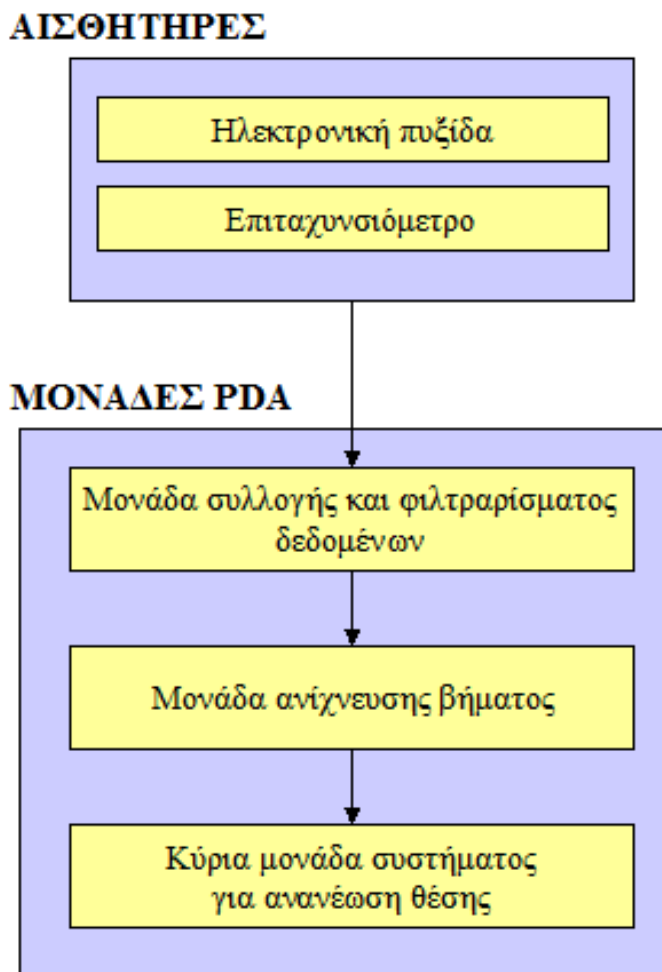
6.2 Υλοποίηση κώδικα

Το σύστημα, όπως αναφέρθηκε και πιο πριν, υλοποιήθηκε στην γλώσσα C# υπό την μορφή ξεχωριστών μονάδων οι οποίες εκτελούνται παράλληλα. Στην συγκεκριμένη περίπτωση της δικής μας υλοποίησης κάτι τέτοιο δεν ήταν εφικτό με βάση τον εξοπλισμό μας γι αυτό και υιοθετήθηκε η τεχνική των νημάτων (threads) όπου οι μονάδες εκτελούνται ψευδοπαράλληλα.

Οι μονάδες που υλοποιήθηκαν είναι οι εξής :

- Αρχικοποίηση μεταβλητών και ορισμός σταθερών
- Μονάδα συλλογής και φιλτραρίσματος δεδομένων
- Μονάδα ανίχνευσης βήματος
- Κύρια μονάδα συστήματος για ανανέωση θέσης

Ο πηγαίος κώδικας που χρησιμοποιήθηκε παρατίθεται αυτούσιος στο Παράρτημα 1 με λεπτομερή επεξήγηση για κάθε κομμάτι ξεχωριστά καθώς και μερικές επιπλέον πληροφορίες. Σε αυτήν την παράγραφο θα γίνει μια μικρή επεξήγηση κάθε μονάδας. Στην εικόνα 6.1.3 φαίνεται η αρχιτεκτονική του συστήματος.



Εικόνα 6.1.3 : Η αρχιτεκτονική του συστήματος DR που αναπτύχθηκε

6.2.1 Αρχικοποίηση μεταβλητών και ορισμός σταθερών

Το κομμάτι αυτό είναι πολύ σημαντικό καθώς ορίζονται οι τιμές των καθολικών μεταβλητών και σταθερών του συστήματος. Κάποια πολύ μικρή αλλαγή σε κάποια τιμή μπορεί να επιφέρει μεγάλες αλλαγές στη συμπεριφορά και στα αποτελέσματα του συστήματός μας. Πριν την ολοκλήρωση του συστήματος, χρειάστηκαν πολλές ώρες πειραμάτων και πολλές αλλαγές τιμών έτσι ώστε να φτάσουμε σε ένα ικανοποιητικό αποτέλεσμα. Αξίζει επίσης να σημειωθεί ότι οι τιμές κάποιων σταθερών εξαρτώνται από τις συσκευές που χρησιμοποιούμε. Για παράδειγμα, η συχνότητα λήψης μετρήσεων από το επιταχυνσιόμετρο (η οποία αντιστοιχεί σε έναν ακέραιο που δηλώνει ανά πόσα

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

milliseconds θα γίνεται λήψη μέτρησης) θα πρέπει να συμφωνεί με τις δυνατότητες του εκάστοτε επιταχυνσιόμετρου που χρησιμοποιείται. Δεν μπορούμε δηλαδή να ορίσουμε την συγκεκριμένη μεταβλητή σε 10 ms (100 Hz) την στιγμή που το επιταχυνσιόμετρό μας υποστηρίζει δειγματοληψία έως 50 Hz. Κάποιες άλλες σημαντικές σταθερές που πρέπει να οριστούν περιλαμβάνουν :

- την σταθερά K που περιγράφηκε στο κεφάλαιο 6.2
- την ελάχιστη επιτρεπτή διαφορά μεταξύ της μέγιστης και της ελάχιστης επιτάχυνσης στον κάθετο άξονα Z για να έχουμε έγγυρο βήμα
- την ελάχιστη επιτρεπτή διαφορά μεταξύ της μέγιστης και της ελάχιστης επιτάχυνσης στους άξονες X, Y για να έχουμε έγγυρο βήμα
- τον αριθμό των μετρήσεων, οι οποίες λαμβάνονται υπόψη, για το φιλτράρισμα των δεδομένων των επιταχύνσεων και της απόκλισης από το βορρά
- την συχνότητα εκτέλεσης του αλγορίθμου ανίχνευσης βήματος ορισμένη σε milliseconds

6.2.2 Μονάδα συλλογής και φιλτραρίσματος δεδομένων

Σε αυτήν τη μονάδα γίνεται η δειγματοληψία από το επιταχυνσιόμετρο (και στους 3 άξονες) και από την ηλεκτρονική πυξίδα σύμφωνα με την συχνότητα που ορίστηκε στο κομμάτι της Αρχικοποίησης μεταβλητών και ορισμού σταθερών. Μετά από κάθε λήψη δείγματος, γίνεται επιτόπου φιλτράρισμα της συγκεκριμένης τιμής με ένα φίλτρο «Μέσου όρου» και στη συνέχεια τα νέα δεδομένα αποθηκεύονται έτσι ώστε να εξεταστούν και να χρησιμοποιηθούν από τις επόμενες μονάδες.

6.2.3 Μονάδα ανίχνευσης βήματος

Αυτή είναι η πιο σημαντική μονάδα του συστήματος πλοήγησης καθώς βάση στα αποτελέσματα αυτής έχουμε ανανέωση θέσης του χρήστη. Στη ουσία όλες οι υπόλοιπες μονάδες δουλεύουν αδιάκοπα έτσι ώστε να τροφοδοτούν την μονάδα ανίχνευσης βήματος με ενημερωμένα δεδομένα. Η δουλειά της είναι να ανακτά τα δεδομένα που έχουν φιλτραριστεί, να τα εξετάζει και με βάση κάποιον αλγόριθμο ανίχνευσης βήματος

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

να βγάζει συμπέρασμα για το αν ο χρήστης έκανε κάποιο βήμα. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε η μέθοδος ανίχνευσης κορυφών.

Σε αυτό το σημείο θα πρέπει να ξανεπισημάνουμε ότι οι μονάδες ανίχνευσης βήματος και συλλογής και φιλτραρίσματος δεδομένων εκτελούνται αδιάκοπα και παράλληλα. Επίσης όπως καταλαβαίνουμε και οι δύο έχουν πρόσβαση στις ίδιες πληροφορίες δεδομένων : η μονάδα συλλογής δεδομένων συλλέγει, φιλτράρει και ανανεώνει τα δεδομένα επιτάχυνσης και κατεύθυνσης, και η μονάδα ανίχνευσης βήματος τα χρησιμοποιεί σε πραγματικό χρόνο. Ως εκ τούτου, λόγω της αδιάκοπης και παράλληλης λειτουργίας τους, είναι πολύ πιθανό κάποια στιγμή να υπάρξει κάποια διένεξη μεταξύ τους, την στιγμή δηλαδή που η μια μονάδα προσπαθεί να σώσει τα τελευταία δεδομένα που συλλέχθηκαν η άλλη να προσπαθεί να ανακτήσει τα δεδομένα αυτά που δεν υπάρχουν ακόμη. Για το λόγω αυτό χρειάζεται μεγάλη προσοχή όταν έχουμε να κάνουμε με πρόσβαση σε δεδομένα κοινής χρήσης. Στο σύστημα που αναπτύχθηκε το πρόβλημα αυτό αντιμετωπίστηκε ως εξής: κάθε φορά που κάποια μονάδα θέλει να χρησιμοποιήσει δεδομένα κοινής χρήσης ελέγχει αρχικά αν χρησιμοποιούνται από κάποια άλλη μονάδα. Αν χρησιμοποιούνται τότε περιμένει μέχρι να ελευθερωθούν, αν όχι τότε τα κλειδώνει και τα χρησιμοποιεί εκείνη. Όταν ολοκληρώσει την λειτουργία της τα ξεκλειδώνει και είναι πλέον ελεύθερα για οποιαδήποτε άλλη μονάδα τα χρειαστεί.

6.2.4 Κύρια μονάδα συστήματος για ανανέωση θέσης

Η μονάδα αυτή είναι ο συντονιστής του συστήματός μας. Οι λειτουργίες της χωρίζονται σε 3 μέρη :

- εκτέλεση μερικών τελικών ελέγχων πριν την αποδοχή ενός ανιχνευμένου βήματος από την αντίστοιχη μονάδα. Πχ έλεγχος μήπως το βήμα που ανιχνεύθηκε οφειλόταν απλά σε κάποιο μικρό τράνταγμα.
- ανανέωση θέσης όταν τα δεδομένα του βήματος περάσουν από τους τελικούς ελέγχους.
- εκκαθάριση μνήμης από στοιχεία που δεν χρειάζονται πλέον.

Η τρίτη λειτουργία είναι και η πιο σημαντική δεδομένου του ότι το σύστημά μας εκτελείται πάνω σε κάποιο PDA με περιορισμένη υπολογιστική ισχύ και μνήμη. Ως εκ

τούτου η εκκαθάριση της μνήμης ανά τακτά χρονικά διαστήματα από δεδομένα που δεν είναι πλέον χρήσιμα είναι ζωτικής σημασίας έτσι ώστε να μπορεί το σύστημα να εκτελείται όσο πιο ανενόχλητα γίνεται.

6.3 Αξιολόγηση συστήματος

Σε αυτήν την υποενότητα παρουσιάζουμε στοιχεία και μετρήσεις από τα πειράματα αυτά που θα βοηθήσουν στην αξιολόγηση του συστήματος πλοήγησης με τη τεχνική στίγματος εξ αναμετρήσεως. Για να προσδιοριστεί κατά πόσον το σύστημα που υλοποιήθηκε ήταν λειτουργικό έγιναν κάποια πειράματα στον χώρο του κτηρίου της Πληροφορικής του Πανεπιστημίου Αθηνών. Στη συνέχεια παρουσιάζονται μετρήσεις που έγιναν σε μια σχεδόν τετραγωνική διαδρομή συνολικής απόστασης περίπου 100 μέτρων (99,41 μέτρα). Για το πείραμα αυτό ορίστηκαν 5 κομβικά σημεία όπου σημειωνόταν η μέτρηση της μεθόδου Dead Reckoning για να συγκριθεί στη συνέχεια με τις πραγματικές συντεταγμένες των κομβικών σημείων. Να σημειωθεί ότι στα κομβικά σημεία δεν γινόταν διόρθωση συντεταγμένων με τις πραγματικές, αλλά οποιαδήποτε απόκλιση συσσωρευόταν στο τελικό αποτέλεσμα.

Πίνακας 6.3.1 : Πραγματικές συντεταγμένες κομβικών σημείων (σε μέτρα)

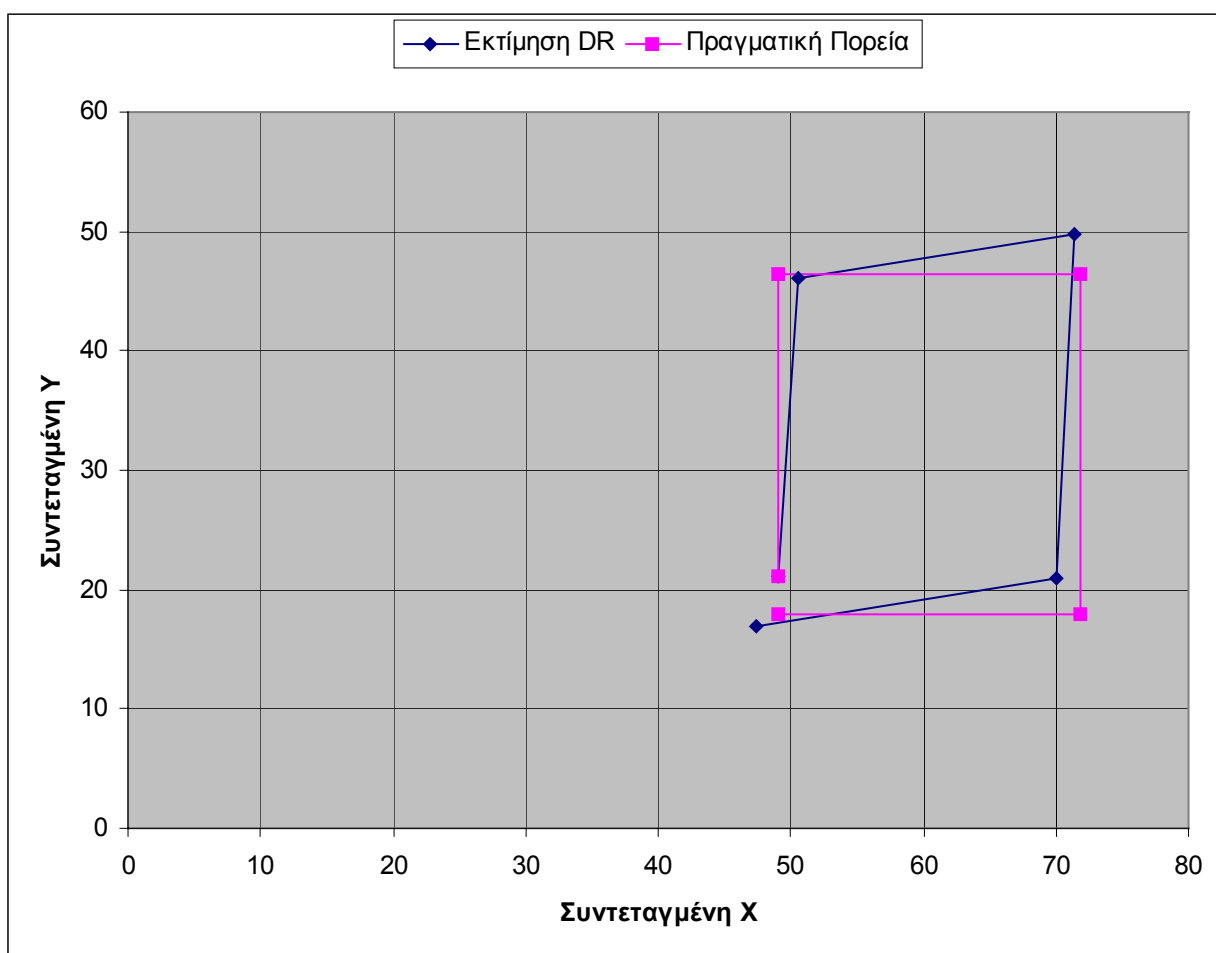
Σημεία	Συντεταγμένη X	Συντεταγμένη Y
Point 1	49,09	21,14
Point 2	49,09	46,51
Point 3	71,85	46,51
Point 4	71,85	17,98
Point 5	49,09	17,98

Πίνακας 6.3.2 : Εκτίμηση συντεταγμένων κομβικών σημείων με την μέθοδο DR (σε μέτρα)

Σημεία	Συντεταγμένη Χ	Συντεταγμένη Υ
Point 1	49,09	21,14
Point 2	50,51	46,05
Point 3	71,37	49,76
Point 4	70,01	20,95
Point 5	47,42	16,92

Πίνακας 6.3.3 : Απόκλιση εκτίμησης θέσης DR από την πραγματική θέση

Σημεία	Απόκλιση DR από πραγματική θέση
Από Point 1 μέχρι Point 2	1,49
Από Point 1 μέχρι Point 3	3,29
Από Point 1 μέχρι Point 4	3,49
Από Point 1 μέχρι Point 5	1,98



Εικόνα 6.3.4 : Πραγματική θέση και προσδιορισμός θέσης με την μέθοδο DR

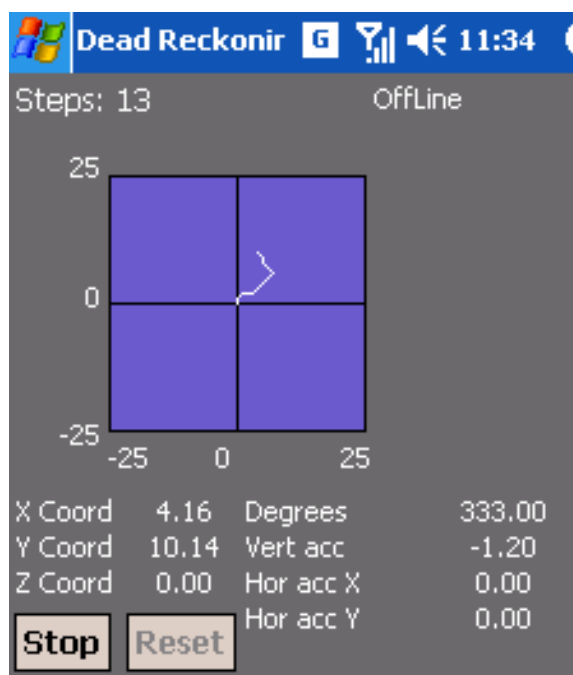
Σε γενικές γραμμές παρατηρούμε ότι η απόκλιση από την πραγματική θέση του χρήστη είναι αρκετά μικρή σε σχέση με την απόσταση που διανύθηκε. Μιλάμε δηλαδή για μια απόκλιση της τάξεως των 1,98 μέτρων σε μια συνολική απόσταση 100 μέτρων η οποία αξιολογείται ως αρκετά ικανοποιητική για τις ανάγκες της παρούσης πτυχιακής.

Ένα σημείο που θα πρέπει να δωθεί μεγάλη προσοχή είναι ότι ενώ οι αποστάσεις που διανύει ο χρήστης υπολογίζονται με αρκετά μεγάλη ακρίβεια, υπάρχει μια αρκετά σημαντική απόκλιση όσον αφορά την κατεύθυνση του χρήστη όπως φαίνεται και στην εικόνα 6.3.4.

Σημειώνεται ότι η χρησιμοποιούμενη ηλεκτρονική πυξίδα (HMR 3300) διαθέτει ενσωματωμένο μικροελεγκτή ο οποίος με κατάλληλα ψηφιακά φίλτρα υλοποιημένα στο λογισμικό του από την κατασκευάστρια εταιρεία επιτυγχάνει την διόρθωση της κλίσης σε στατικές και δυναμικές συνθήκες με τις προαναφερθείσες επιδόσεις. Η επίδραση μαγνητικών πεδίων μικρής έντασης και η οποία μπορεί να οφείλεται στον ίδιο τον χρήστη αξιολογήθηκε ότι δεν επηρεάζει σημαντικά τις μετρήσεις. Οι ισχυρές μαγνητικές

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

παρεμβολές που μπορούν να οφείλονται στην υποδομή του κτιρίου (σκελετός του κτιρίου, ανελκυστήρες, ηλεκτρικά καλώδια) μπορούν να αλλοιώσουν σημαντικά τις μετρήσεις και δεν μπορούν να αποφευχθούν παρά με διαδικασίες βαθμονόμησης της πυξίδας σε όλα τα προβληματικά σημεία του κτιρίου. Λόγω του συνθέτου των απαιτούμενων διαδικασιών (οι μετρήσεις μαγνητικού πεδίου πρέπει να καλύψουν όλο το κτίριο με μεγάλη χωρική πυκνότητα) στην τρέχουσα υλοποίηση δεν υποστηρίζεται σχετική διόρθωση. Στην εικόνα 6.3.5 παρουσιάζεται το γραφικό περιβάλλον του συστήματος.



Εικόνα 6.3.5 : Το γραφικό περιβάλλον του συστήματος πλοήγησης

6.4 Συμπεράσματα και μελλοντική δουλειά

Στην παρούσα πτυχιακή παρουσιάστηκαν πολλές διαφορετικές μεθόδους και τεχνικές που μπορούν να χρησιμοποιηθούν για την πλοήγηση ενός χρήστη σε εσωτερικούς χώρους. Αναπτύχθηκαν σε βάθος μέθοδοι τόσο για το φιλτράρισμα των δεδομένων που λαμβάνει ένα σύστημα από το επιταχυνσιόμετρο όσο και για το φιλτράρισμα της κατεύθυνσης του χρήστη. Και τα δύο βήματα κρίνονται ως απολύτως αναγκαία για την αξιοπιστία ενός συστήματος DR. Στη συνέχεια αναλύσαμε διάφορες από τις πιο διαδεδομένες τεχνικές που χρησιμοποιούνται σήμερα για τον υπολογισμό της απόστασης που διανύει κάποιος πεζός χρήστης, ο οποίος χρησιμοποιεί ένα σύστημα

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

πλοήγησης, με κάθε του βήμα. Τέλος αναπτύχθηκε ένα αυτόνομο σύστημα πλοήγησης για εσωτερικούς χώρους βασισμένο στην τεχνική στίγματος εξ αναμετρήσεως το οποίο βασίστηκε σε τεχνικές και μεθόδους που παρουσιάστηκαν παραπάνω ενώ παρουσιάστηκαν και τα αποτελέσματα αξιολόγησής του.

Σαν μελλοντική δουλειά θα μπορούσαν να υλοποιηθούν κάποιοι πιο σύνθετοι αλγόριθμοι τόσο για ανίχνευση βήματος όσο και για το φιλτράρισμα κατεύθυνσης οι οποίοι θα βελτίωναν αρκετά την εκτίμηση θέσης του πεζού χρήστη.

Επίσης για ακόμα μεγαλύτερη ακρίβεια και για να είναι το σύστημά μας ακόμα πιο ακριβές θα μπορούσε να συνδιαστεί με διάφορες άλλες τεχνολογίες όπως γίνεται για παράδειγμα στο υβριδικό σύστημα πλοήγησης σε εσωτερικούς χώρους «ΜΝΗΣΙΚΛΗΣ»[19] όπου εκτός από το σύστημα DR χρησιμοποιείται επίσης και η τεχνολογία RFID (Radio Frequency IDentification) για την ανίχνευση της εγγύτητας του χρήστη σε συγκεκριμένα σημεία του κτιρίου με την χρήση παθητικών UHF RFIDs τα οποία τοποθετούνται με μεγάλη πυκνότητα στην υποδομή του κτιρίου καθώς και η τεχνική συσχέτισης της θέσης στο κτίριο με την ένταση σήματος (RSSI) της κάρτας WLAN του φορητού υπολογιστή του χρήστη από ένα αριθμό σημείων πρόσβασης. Το σύστημα αυτό υποστηρίζεται από πολλαπλούς μηχανισμούς εντοπισμού θέσης και προσανατολισμού οι οποίοι συνδυάζονται μέσω κάποιου υποσυστήματος σύντηξης πληροφορίας θέσης (information fusion) μειώνοντας ως εκ τούτου το σφάλμα.

ΠΑΡΑΡΤΗΜΑ

ΚΩΔΙΚΑΣ

Π1. Καθολικές μεταβλητές συστήματος

Σε αυτόν την ενότητα θα γίνει μια επεξήγηση των μεταβλητών που θα χρειαστούν για την ομαλή λειτουργία τους συστήματος υπολογισμού θέσης. Θα χρειαστούμε διάφορους τύπους μεταβλητών για να αποθηκεύουμε την προσωρινή κατάσταση του συστήματος όπως την θέση μας (συντεταγμένες στους άξονες X,Y,Z) και την κατεύθυνσή μας σε μοίρες, διάφορες λίστες για την αποθήκευση των αφιτράριστων και φιλτραρισμένων δεδομένων από τα επιταχυνσιόμετρα και από την ηλεκτρονική πυξίδα κ.α. Αναλυτικότερη επεξήγηση παρατίθεται σε μορφή σχολίων μέσα στον κώδικα.

```
//εάν ο αλγόριθμος είναι σε λειτουργία η μεταβλητή αυτή
```

```
//είναι 'TRUE', αλλιώς 'FALSE'
```

```
bool running;
```

```
//οι μεταβλητές X,Y,Z κρατούν τις συντεταγμένες
```

```
//της θέσης μας ανά πάσα στιγμή
```

```
double X;
```

```
double Y;
```

```
double Z;
```

```
//περιέχουν τις επιταχύνσεις που συλλέγονται
```

```
//από τα επιταχυνσιόμετρα ανά πάσα στιγμή
```

```
double hor_acc_X;
```

```
double hor_acc_Y;
```

```
double ver_acc_Z;
```

```
//περιέχει την απόκλιση από τον βορρά ανά πάσα στιγμή
```

```
//όπως αυτή συλλέγεται από την ηλεκτρονική πυξίδα
```

```
double degrees;
```

```
//περιέχει την φιλτραρισμένη απόκλιση από το βορρά
```

```
//μετά από αλγόριθμο φιλτραρίσματος
```

```
double step_heading;
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
//arrays που περιέχουν τις τιμές των επιταχύνσεων
//και της απόκλισης της πυξίδας από το βορρά
//όπως αυτές συλλέγονται από τα επιταχυνσιόμετρα
//και την ηλεκτρονική πυξίδα
ArrayList x_acc_array = new ArrayList();
ArrayList y_acc_array = new ArrayList();
ArrayList z_acc_array = new ArrayList();
ArrayList degrees_array = new ArrayList();

//arrays που περιέχουν τις φιλτραρισμένες
//τιμές επιταχύνσεων και πυξίδας
ArrayList filtered_x_acc_array = new ArrayList();
ArrayList filtered_y_acc_array = new ArrayList();
ArrayList filtered_z_acc_array = new ArrayList();
ArrayList filtered_degrees_array = new ArrayList();

//arrays που περιέχουν τις απαραίτητες τελευταίες
//φιλτραρισμένες τιμές των επιταχύνσεων και της πυξίδας
//(χρησιμοποιούνται σε περιπτώσεις καθαρισμού των
//arrays που περιγράφονται ακριβώς από πάνω)
ArrayList x_acc_filtered_storage = new ArrayList();
ArrayList y_acc_filtered_storage = new ArrayList();
ArrayList z_acc_filtered_storage = new ArrayList();
ArrayList degrees_filtered_storage = new ArrayList();

//περιέχει το χρονικό διάστημα που πρέπει να περάσει
//για να πάρουμε νέες μετρήσεις από τα επιταχυνσιόμετρα
//και την ηλεκτρονική πυξίδα
//(εκφρασμένο σε milliseconds)
int collect_vert_acc_delay;

//περιέχει το χρονικό διάστημα που πρέπει να περάσει
//για να εκτελεστεί ο αλγόριθμος ανίχνευσης βήματος
//(εκφρασμένο σε milliseconds)
int stride_check_delay;

//αν έχει ανιχνευθεί ένα πιθανό βήμα τότε η μεταβλητή
//αυτή είναι TRUE , αλλιώς FALSE
bool STRIDE;

//περιέχει μια σταθερή τιμή η οποία χρησιμοποιείται
//για τον υπολογισμό της απόστασης του βήματος
```

```
double K;

//περιέχει τον αριθμό των βημάτων που έχουν ανιχνευθεί
//από την έναρξη της λειτουργίας του αλγορίθμου
int strides_counted;

//περιέχει ένα δείκτη στο τελευταίο στοιχείο των arrays
//το οποίο αντιστοιχεί στην ολοκλήρωση ενός πιθανού
//βήματος
int last_stride_element;

//περιέχει τον ελάχιστο χρόνο που θα πρέπει να μεσολαβεί
//κατά την ανίχνευση 2 ακροτάτων στο σήμα της κάθετης
//επιτάχυνσης (ο χρόνος αυτός εκφράζεται σε αριθμό
//αποθηκευμένων στοιχείων και χρησιμοποιείται για τον
//αποκλεισμό λανθασμένων ανιχνεύσεων βήματος)
int extrema_detection_minimum_time;

//περιέχει την ελάχιστη επιτρεπτή διαφορά μεταξύ
//της μέγιστης και της ελάχιστης επιτάχυνσης στον
//κάθετο άξονα Z για να έχουμε έγγυρο βήμα
double min_acc_diff;

//περιέχει την ελάχιστη επιτρεπτή διαφορά μεταξύ
//της μέγιστης και της ελάχιστης επιτάχυνσης στους
//οριζόντιους άξονες X,Y για να έχουμε έγγυρο βήμα
double min_hor_acc_diff;

//περιέχει τον αριθμό των μετρήσεων, οι οποίες
//λαμβάνονται υπόψη για το φιλτράρισμα των δεδομένων
//των επιταχύνσεων και της απόκλισης από το βορρά
int moving_average_elements;

//σε κάθε έλεγχο για πιθανό βήμα, οι μεταβλητές αυτές
//περιέχουν τις μέγιστες και ελάχιστες τιμές των
//επιταχύνσεων στους άξονες X,Y,Z
double stride_acc_max, stride_acc_min;
double stride_acc_X_max, stride_acc_X_min;
double stride_acc_Y_max, stride_acc_Y_min;

//μεταβλητή που περιέχει το νήμα από το οποίο
//χρησιμοποιούνται ανά πάσα στιγμή τα global δεδομένα
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
string DATA_LOCKED;  
  
//τα 3 νήματα που θα χρησιμοποιήσει η εφαρμογή  
//το νήμα για οριστικοποίηση βήματος και  
//υπολογισμό νέας θέσης  
//το νήμα για έλεγχο νέου πιθανού βήματος  
//το νήμα συλλογής νέων μετρήσεων  
Thread PDR_Thread;  
Thread PDR_check;  
Thread PDR_collect;
```

Π2. Κώδικας συνάρτησης αρχικοποίησης μεταβλητών

Σε αυτόν τον τομέα θα γίνει η ανάθεση αρχικών τιμών στις μεταβλητές του συστήματος. Οι τιμές κάποιων από αυτές τις μεταβλητές παίζουν μεγάλο ρόλο στην απόδοση του συστήματος και ο ορισμός τους έγινε έπειτα από διάφορα τεστ έτσι ώστε να βελτιστοποιηθεί η απόδοση της τεχνικής Dead Reckoning.

```
//συνάρτηση αρχικοποίησης μεταβλητών  
private void InitializeVariables()  
{  
    //η ελάχιστη διαφορά μεταξύ μέγιστης και ελάχιστης τιμής  
    //των μετρήσεων της επιτάχυνσης στον άξονα Z, κατά την  
    //διάρκεια ενός πιθανού βήματος ορίζεται σε 0.2 m/s^2  
    min_acc_diff = 0.20;  
  
    //η ελάχιστη διαφορά μεταξύ μέγιστης και ελάχιστης τιμής  
    //των μετρήσεων της επιτάχυνσης στον άξονα Z, κατά την  
    //διάρκεια ενός πιθανού βήματος ορίζεται σε 0.15 m/s^2  
    min_hor_acc_diff = 0.15;  
  
    //ο αριθμός των στοιχείων που χρησιμοποιούνται για το  
    //φιλτράρισμα των μετρήσεων της επιτάχυνσης στους άξονες  
    //X,Y,Z και την απόκλιση από τον βορρά ορίζεται σε 8
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
//(έχουμε λάβει υπόψην ότι η μέση συχνότητα βήματος ενός
//ανθρώπου είναι 2 με 3 Hz, ενώ η συχνότητα λήψης
//μετρήσεων είναι 20 Hz)
moving_average_elements = 8;

//ο ελάχιστος χρόνος που θα πρέπει να μεσολαβεί κατά την
//ανίχνευση 2 ακροτάτων στο σήμα της κάθετης
//επιτάχυνσης εκφρασμένος σε αριθμό αποθηκευμένων
//στοιχείων ορίζεται σε 4
extrema_detection_minimum_time = 4;

//η συχνότητα λήψης μετρήσεων εκφρασμένη σε milliseconds
//ορίζεται σε 50ms (20 Hz)
collect_vert_acc_delay = 50;

//το χρονικό διάστημα που πρέπει να περάσει για να
//εκτελεστεί ο αλγόριθμος ανίχνευσης βήματος ορίζεται
//σε 200 ms
stride_check_delay = 200;

//η σταθερά η οποία χρησιμοποιείται για τον υπολογισμό
//της απόστασης του βήματος ορίζεται σε 0.95
K = 0.95;

//αρχικά η πρόσβαση στα δεδομένα είναι ελεύθερη
//μέχρι να δεσμευτεί από κάποιο νήμα
DATA_LOCKED = "";

//η αρχικοποίηση των συντεταγμένων της θέσης μας
//ορίζεται στο σημείο 0,0,0
X = 0.0f;
Y = 0.0f;
Z = 0.0f;

//αρχικά ο αλγόριθμος δεν εκτελείται
running = false;

//αρχικοποίηση των τιμών των επιταχύνσεων στους άξονες
//X,Y,Z και την απόκλιση από το βορρά
hor_acc_X = 0.0f;
hor_acc_Y = 0.0f;
ver_acc_Z = 0.0f;
```


Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
degrees = 0.0f;

//αρχικοποίηση της μεταβλητής ανίχνευσης βήματος
//σε FALSE
STRIDE = false;

//αρχικοποίηση των βημάτων που έχουν ανιχνευθεί σε 0
strides_counted = 0;
}
```

Π3. Κώδικας συνάρτησης φιλτραρίσματος μετρήσεων

Σε αυτόν τον τομέα θα παρουσιαστεί η συνάρτηση φιλτραρίσματος των δεδομένων. Για να φιλτραριστούν τα δεδομένα και να απομακρυνθούν τυχόν λανθασμένες μετρήσεις χρησιμοποιείται ένα Moving average φίλτρο πάνω στις μετρήσεις των επιταχύνσεων στους άξονες X,Y,Z καθώς και στις μετρήσεις της πυξίδας. Κάθε φορά που συλλέγουμε μια καινούρια μέτρηση από τους αισθητήρες (επιταχυνσιομετρα, ηλεκτρονική πυξίδα) γίνεται επιτόπου φιλτράρισμα το οποίο βασίζεται σε προηγούμενες μετρήσεις που συλλέχθηκαν.

```
public void Collect_values()
{
    //γίνεται έλεγχος για το αν τα δεδομένα χρησιμοποιούνται
    //από άλλο νήμα. Αν χρησιμοποιούνται, τότε συνεχίζουμε
    //να ελέγχουμε μέχρι να ελευθερωθούν
    while (
        (DATA_LOCKED == "STRIDE_CHECK") ||
        (DATA_LOCKED == "LAST_CHECK"))
    {}
    //όταν ελευθερωθούν τα δεδομένα τότε κλειδώνονται από το
    //παρόν νήμα
    DATA_LOCKED = "COLLECTER";
}
```

```
//μεταβλητές που θα χρειαστούν για το φιλτράρισμα της
//μέτρησης.
//περιέχουν τις τιμές που λήφθηκαν από την νέα μέτρηση
double my_z_acc, my_x_acc, my_y_acc, my_degrees;

//ο συνολικός αριθμός προηγούμενων μετρήσεων που λήφθηκαν
//υπόψη για το φιλτράρισμα των τιμών της νέας μέτρησης
int total;

//περιέχουν το άθροισμα των τιμών των τελευταίων 'total'
//μετρήσεων
double tmp_sum, tmp_sum_X, tmp_sum_Y, tmp_sum_degrees;

//δείκτης που χρησιμοποιείται σε περίπτωση που τα arrays
//μετρήσεων έχουν διαγραφεί (για εξοικονόμηση μνήμης) και
//πρέπει να ληφθούν δεδομένα από τα arrays των πρόσφατων
//μετρήσεων
int pointer;

//ανάθεση των τιμών από τη νέα μέτρηση
my_z_acc = Z;
my_x_acc = X;
my_y_acc = Z;
my_degrees = new_degrees;

//εαν τα arrays μετρήσεων δεν περιέχουν κανένα στοιχείο,
//τοτε δε μπορεί να γίνει φιλτράρισμα
if (z_acc_array.Count == 0)
{
    //προσθέτουμε απλά τις τιμές από την νέα μέτρηση
    //στα arrays δεδομένων, και φιλτραρισμένων δεδομένων
    z_acc_array.Add(my_z_acc);
    filtered_z_acc_array.Add(my_z_acc);

    x_acc_array.Add(my_x_acc);
    filtered_x_acc_array.Add(my_x_acc);

    y_acc_array.Add(my_y_acc);
    filtered_y_acc_array.Add(my_y_acc);
}
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
degrees_array.Add(my_degrees);
filtered_degrees_array.Add(my_degrees);
}
//αλλιώς εάν τα arrays μετρήσεων περιέχουν περισσότερα από ένα στοιχεία
else
{
    //αρχικοποίηση μεταβλητών
    total = 0;
    tmp_sum = 0.0;
    tmp_sum_X = 0.0;
    tmp_sum_Y = 0.0;
    tmp_sum_degrees = 0.0;

    //περιέχει την τελευταία τιμή της ηλεκτρονικής πυξίδας που λήφθηκε υπόψη από
//τον αλγόριθμο φιλτραρίσματος.
    //αρχικοποίηση σε -1, που σημαίνει ότι δεν έχει αρχίσει η επεξεργασία
    double last_degrees = -1;

    //προσθήκη των τιμών της νέας μέτρησης στα arrays μετρήσεων
    z_acc_array.Add(my_z_acc);
    x_acc_array.Add(my_x_acc);
    y_acc_array.Add(my_y_acc);
    degrees_array.Add(my_degrees);

    //ο βρόγχος αυτός υπολογίζει τα αθροίσματα των τελευταίων 'N' μετρήσεων
//όπου 'N' ο αριθμός που ορίστηκε στην μεταβλητή 'moving_average_elements',
//σκανάρωντας τα arrays μετρήσεων
    for
        (int x = z_acc_array.Count - 2;
         x >= z_acc_array.Count - moving_average_elements
         &&
         x >= 0);
        x--)
    {
        //αύξηση της μεταβλητής που περιέχει τον αριθμό
//των στοιχείων που επεξεργάστηκαν
        total++;

        //ανανέωση του αθροίσματος των τιμών των επιταχύνσεων
        tmp_sum += (double)z_acc_array[x];
        tmp_sum_X += (double)x_acc_array[x];
    }
}
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
tmp_sum_Y += (double)y_acc_array[x];

//-----
//αλγόριθμος υπολογισμού κατάλληλης τιμής για την
//μέτρηση των μοιρών (απόκλιση από βορρά)
//-----
//αν αυτή είναι η πρώτη μας μέτρηση
if (last_degrees == -1)
{
    //απλά ανανέωσε το άθροισμα των τιμών της
    //πυξίδας, και το last_degrees
    tmp_sum_degrees += (double)degrees_array[x];
    last_degrees = (double)degrees_array[x];
}
//αλλιώς αν έχουμε πάρει ήδη μετρήσεις,
//προσάρμοσε την νέα μέτρηση στην προηγούμενη
else
{
    if ((double)degrees_array[x] - last_degrees > 180)
    {
        tmp_sum_degrees +=
            (double)degrees_array[x] - 360;
    }
    else if ((double)degrees_array[x] - last_degrees < -180)
    {
        tmp_sum_degrees +=
            (double)degrees_array[x] + 360;
    }
    else
    {
        tmp_sum_degrees +=
            (double)degrees_array[x];
    }
}
//-----
//Τέλος αλγορίθμου υπολογισμού κατάλληλης τιμής
//για την μέτρηση των μοιρών (απόκλιση από βορρά)
//-----

}

//σε περίπτωση που τα στοιχεία των arrays μετρήσεων
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
//έχουν διαγραφεί ή δεν επαρκούν για το φιλτράρισμα
//των τιμών της νέας μέτρησης,εφάρμοσε την ίδια
//μέθοδο που χρησιμοποιήθηκε στον προηγούμενο βρόγχο
//αλλά αυτήν τη φορά χρησιμοποίησε τα arrays των
//αποθηκευμένων προσφατων μετρήσεων
//(*_filtered_storage)
//(εφόσον υπάρχουν αποθηκευμένες μετρήσεις)
if ((total < moving_average_elements) &&
(z_acc_filtered_storage.Count != 0))
{
    pointer = -1;
    while (total < moving_average_elements)
    {
        //αν δεν υπάρχουν άλλες μετρήσεις
//αποθηκευμένες, βγές από το βρόγχο
        if (pointer + 1 >
z_acc_filtered_storage.Count - 1)                                break;

        pointer++;
        total++;
        tmp_sum +=
                                (double)z_acc_filtered_storage[pointer];
        tmp_sum_X +=
                                (double)x_acc_filtered_storage[pointer];
        tmp_sum_Y +=
                                (double)y_acc_filtered_storage[pointer];

        if (last_degrees == -1)
        {
            tmp_sum_degrees +=
(double)degrees_filtered_storage[pointer];
            last_degrees =
(double)degrees_filtered_storage[pointer];
        }
        else
        {
            if ((double)degrees_filtered_storage[pointer] - last_degrees > 180)
            {
                tmp_sum_degrees += (double)degrees_filtered_storage[pointer] - 360;
            }
            else if ((double)degrees_filtered_storage[pointer] - last_degrees < -180)
            {
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
        tmp_sum_degrees += (double)degrees_filtered_storage[pointer] + 360;
    }
    else
    {
        tmp_sum_degrees += (double)degrees_filtered_storage[pointer];
    }
}
}
```

```
//πλέον έχουν υπολογιστεί τα απαραίτητα στοιχεία
//για το φιλτράρισμα της νέας μέτρησης.
//προσθέτουμε τις φιλτραρισμένες τιμές στα
//arrays φιλτραρισμένων μετρήσεων
filtered_z_acc_array.Add(tmp_sum / total);
filtered_x_acc_array.Add(tmp_sum_X / total);
filtered_y_acc_array.Add(tmp_sum_Y / total);
filtered_degrees_array.Add(tmp_sum_degrees / total);
```

```
}
```

Π4. Κώδικας συνάρτησης ανίχνευσης βήματος

```
public void Stride_Check()
{
    //δήλωση τοπικών μεταβλητών συναρτήσεως

    //περιέχει την τελευταία τιμή κάθετης επιτάχυνσης που εξετάστηκε
    double last_acc_value;
    //περιέχει την τιμή του στοιχείου κάθετης επιτάχυνσης που εξετάζεται τώρα
    double current_acc_value;
    //περιέχει την τιμή του στοιχείου κατευθύνσεως που εξετάζεται τώρα
    double current_degrees;
    //περιέχει τον αύξοντα αριθμό του στοιχείου στο οποίο ανιχνεύθηκε κρίσιμο σημείο
    int last_extrema_detected_element;
    //περιέχει τον τύπο του τελευταίου κρίσιμου σημείου που ανιχνεύθηκε
    string last_extrema_detected_type;
    //περιέχει το πρόσημο της διαφοράς του τωρινού και του προηγούμενου στοιχείου κάθετης
    επιτάχυνσης
    char current_sign;
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

//περιέχει το πρόσημο της διαφοράς του προηγούμενου και του προ-προηγούμενου στοιχείου
κάθετης επιταχύνσης

```
char last_sign;
```

```
//γίνεται έλεγχος για το αν τα δεδομένα χρησιμοποιούνται
```

```
//από άλλο νήμα. Αν χρησιμοποιούνται, τότε συνεχίζουμε
```

```
//να ελέγχουμε μέχρι να ελευθερωθούν
```

```
while ((DATA_LOCKED == "COLLECTER") || (DATA_LOCKED == "LAST_CHECK"))
```

```
{ }
```

```
//όταν ελευθερωθούν τα δεδομένα τότε κλειδώνονται από το
```

```
//παρόν νήμα
```

```
DATA_LOCKED = "STRIDE_CHECK";
```

```
//αυτός ο βρόγχος εκτελείται μέχρι να διακοπή η λειτουργία του νήματος
```

```
//δηλαδή, ελέγχουμε διαρκώς ανά τακτά διαστήματα για πιθανά βήματα
```

```
while (true)
```

```
{
```

```
    //σε περίπτωση που έχουμε πολλά στοιχεία αποθηκευμένα στη μνήμη
```

```
    //κάνε μια εκκαθάριση για να μην βαρύνει το σύστημα
```

```
    if (z_acc_array.Count - 1 > 200)
```

```
    {
```

```
        //αφαίρεσε τα 50 πρώτα στοιχεία απο κάθε λίστα αποθηκευμένων δεδομένων
```

```
        z_acc_array.RemoveRange(0, 50);
```

```
        filtered_z_acc_array.RemoveRange(0, 50);
```

```
        x_acc_array.RemoveRange(0, 50);
```

```
        filtered_x_acc_array.RemoveRange(0, 50);
```

```
        y_acc_array.RemoveRange(0, 50);
```

```
        filtered_y_acc_array.RemoveRange(0, 50);
```

```
        degrees_array.RemoveRange(0, 50);
```

```
        filtered_degrees_array.RemoveRange(0, 50);
```

```
    }
```

```
    //ελευθέρωσε τα δεδομένα έτσι ώστε να μπορούν να χρησιμοποιηθούν
```

```
    //από άλλα νήματα
```

```
    DATA_LOCKED = "";
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
//περίμενε ένα μικρό διάστημα μέχρι να ξαναελέγξεις για βήμα
Thread.Sleep(stride_check_delay);

//γίνεται έλεγχος για το αν τα δεδομένα χρησιμοποιούνται
//από άλλο νήμα. Αν χρησιμοποιούνται, τότε συνεχίζουμε
//να ελέγχουμε μέχρι να ελευθερωθούν
while ((DATA_LOCKED == "COLLECTER") || (DATA_LOCKED == "LAST_CHECK"))
{
//όταν ελευθερωθούν τα δεδομένα τότε κλειδώνονται από το
//παρόν νήμα
DATA_LOCKED = "STRIDE_CHECK";

//αν υπάρχουν στοιχεία στα arrays των φιλτραρισμένων δεδομένων
//και δεν έχει ανιχνευθεί βήμα ακόμα, τότε ελέγχουμε για πιθανό βήμα
if ((z_acc_array.Count != 0) && (filtered_z_acc_array.Count != 0) &&
(filtered_degrees_array.Count != 0) && (!STRIDE))
{
//αρχικοποίησε τις τοπικές μεταβλητές

//τελευταίο στοιχείο που ανιχνεύθηκε αρχικοποιείται σε 0
last_extrema_detected_element = 0;
//τελευταίο κρίσιμο σημείο που ανιχνεύθηκε αρχικοποιείται σε 'ΕΛΑΧΙΣΤΟ'
last_extrema_detected_type = "minimum";
//τωρινό πρόσημο αρχικοποιείται σε 'ΘΕΤΙΚΟ'
current_sign = '+';
//προηγούμενο πρόσημο αρχικοποιείται σε 'ΑΡΝΗΤΙΚΟ'
last_sign = '-';
//τελευταία τιμή κάθετης επιτάχυνσης αρχικοποιείται
//στο 1ο αποθηκευμένο στοιχείο φιλτραρισμένης επιτάχυνσης στον άξονα Z
last_acc_value = (double)filtered_z_acc_array[0];
//ομοίως αρχικοποιείται και η τωρινή τιμή κάθετης επιτάχυνσης
current_acc_value = (double)filtered_z_acc_array[0];
//ομοίως αρχικοποιείται και η τωρινή τιμή κατευθύνσεως
current_degrees = (double)filtered_degrees_array[0];

//η αρχικοποίηση μεταβλητών μεγίστων και ελαχίστων σε όλους τους άξονες
//γίνεται χρησιμοποιώντας το 1ο στοιχείο των αφιτράριστων δεδομένων (raw data)
stride_acc_max = (double)z_acc_array[0];
stride_acc_min = (double)z_acc_array[0];
```


Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
stride_acc_X_max = (double)x_acc_array[0];
stride_acc_X_min = (double)x_acc_array[0];
stride_acc_Y_max = (double)y_acc_array[0];
stride_acc_Y_min = (double)y_acc_array[0];

//ξεκίνα από το 2ο στοιχείο των αποθηκευμένων δεδομένων επιτάχυνσης
//και συνέχισε για όσο υπάρχουν στοιχεία προς εξέταση
//ή μέχρι να ανιχνευθεί πιθανό βήμα
for (int x = 1; (x <= z_acc_array.Count - 1) && (x <= filtered_z_acc_array.Count - 1) && (x
<= filtered_degrees_array.Count - 1) ; x++)
{

    //ανανεώσε τις μεταβλητές με τα τωρινά στοιχεία κάθετης επιτάχυνσης και
κατευθύνσεως

    //με αυτά τα στοιχεία που εξετάζονται αυτήν τη στιγμή
current_acc_value = (double)filtered_z_acc_array[x];
current_degrees = (double)filtered_degrees_array[x];

    //ανανεώνουμε την μεταβλητή stride_acc_max αν εντοπίσουμε μεγαλύτερη τιμή
τοπικού μεγίστου στον άξονα Z
if ((double)z_acc_array[x] > stride_acc_max)
    stride_acc_max = (double)z_acc_array[x];
    //ανανεώνουμε την μεταβλητή stride_acc_min αν εντοπίσουμε μικρότερη τιμή τοπικού
ελαχίστου στον άξονα Z
if ((double)z_acc_array[x] < stride_acc_min)
    stride_acc_min = (double)z_acc_array[x];
    //ανανεώνουμε την μεταβλητή stride_acc_max αν εντοπίσουμε μεγαλύτερη τιμή
τοπικού μεγίστου στον άξονα X
if ((double)x_acc_array[x] > stride_acc_X_max)
    stride_acc_X_max = (double)x_acc_array[x];
    //ανανεώνουμε την μεταβλητή stride_acc_min αν εντοπίσουμε μικρότερη τιμή τοπικού
ελαχίστου στον άξονα X
if ((double)x_acc_array[x] < stride_acc_X_min)
    stride_acc_X_min = (double)x_acc_array[x];
    //ανανεώνουμε την μεταβλητή stride_acc_max αν εντοπίσουμε μεγαλύτερη τιμή
τοπικού μεγίστου στον άξονα Y
if ((double)y_acc_array[x] > stride_acc_Y_max)
    stride_acc_Y_max = (double)y_acc_array[x];
    //ανανεώνουμε την μεταβλητή stride_acc_min αν εντοπίσουμε μικρότερη τιμή τοπικού
ελαχίστου στον άξονα Y
if ((double)y_acc_array[x] < stride_acc_Y_min)
    stride_acc_Y_min = (double)y_acc_array[x];
```

```
//ελέγχουμε την διαφορά της τωρινής μέτρησης με την προηγούμενη
//και βγάζουμε συμπέρασμα για το πρόσημο, δηλαδή αν η επιτάχυνση
//στον Z άξονα αυξάνεται ή μειώνεται
if (current_acc_value - last_acc_value >= 0)
    current_sign = '+';
else
    current_sign = '-';

//αν εντοπίσουμε αλλαγή στο πρόσημο μεταξύ της τωρινής και της προηγούμενης
μέτρησης
//και δούμε ότι έχει περάσει μια ελάχιστη χρονική περίοδος από την τελευταία
ανίχνευση
//κρίσιμου σημείου (πρόληψη λανθασμένων ανιχνεύσεων από κραδασμούς σώματος)
//σημαίνει ότι έχουμε βρει κάποιο πιθανό κρίσιμο σημείο (μέγιστο ή ελάχιστο)
if ((current_sign != last_sign) && (x >= last_extrema_detected_element +
extrema_detection_minimum_time))
{
    //αν το πρόσημο αλλάζει από αρνητικό σε θετικό τότε εξετάζουμε για τοπικό μέγιστο
    if ((last_sign == '-') && (current_sign == '+'))
    {
        //εξετάζουμε αν η τελευταία ανίχνευση δεν ήταν επίσης τοπικό μέγιστο
        //(αν ήταν τότε θα ήταν λανθασμένη ανίχνευση)
        if (last_extrema_detected_type != "maximum")
        {
            //τότε έχουμε όντως τοπικό μέγιστο
            //ανανεώνουμε την μεταβλητή που θυμάται σε ποιο στοιχείο έγινε η ανίχνευση
            last_extrema_detected_element = x;
            //ανανεώνουμε τον τύπο της ανίχνευσης (τοπικό μέγιστο)
            last_extrema_detected_type = "maximum";
        }
    }
}
//αλλιώς (αν το πρόσημο αλλάζει από θετικό σε αρνητικό) τότε εξετάζουμε για τοπικό
ελάχιστο
else // alliw eksetazoume gia topiko elahisto
{
    //εξετάζουμε αν η τελευταία ανίχνευση δεν ήταν επίσης τοπικό ελάχιστο
    //(αν ήταν τότε θα ήταν λανθασμένη ανίχνευση)
    if (last_extrema_detected_type != "minimum")
    {
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
//τότε έχουμε όντως τοπικό ελάχιστο
//ανανεώνουμε τον τύπο της ανίχνευσης (τοπικό ελάχιστο)
last_extrema_detected_type = "minimum";

//τελική συνθήκη για να δεχτούμε το πιθανό βήμα είναι
//η διαφορά της μέγιστης επιτάχυνσης στον άξονα Z με την ελάχιστη που
ανιχνεύθηκε

//να είναι μεγαλύτερη από ένα κατώτατο κατώφλι που έχουμε ορίσει
//(πρόληψη λανθασμένων ανιχνεύσεων απο κραδασμούς σώματος χρήστη)
if (stride_acc_max - stride_acc_min >= min_acc_diff)
{
    //τότε όντως έχουμε ανιχνεύσει τοπικό ελάχιστο αλλά και πιθανό βήμα
    //ανανεώνουμε την μεταβλητή που θυμάται σε ποιο στοιχείο έγινε η
ανίχνευση του τελευταίου κρίσιμου σημείου
    last_extrema_detected_element = x;
    //ανανεώνουμε την μεταβλητή που θυμάται σε ποιο στοιχείο έγινε η
ανίχνευση του τελευταίου βήματος
    last_stride_element = x;
    //αποθηκεύουμε την μέτρηση της φιλτραρισμένης κατεύθυνσης την στιγμή
ανίχνευσης βήματος
    step_heading = current_degrees;
    //θέσε την μεταβλητή ανίχνευσης βήματος σε 'ΑΛΗΘΕΣ'
    STRIDE = true;
    //βγες από τον βρόχο (περιμένει μέχρι να επεξεργαστεί η μέτρηση από το
κύριο νήμα της μεθόδου)
    break;
}
}
}
}
//ανανέωσε το τελευταίο πρόσημο που ανιχνεύθηκε με το τωρινό
last_sign = current_sign;
//ανανέωσε την τελευταία μέτρηση φιλτραρισμένης επιτάχυνσης στον άξονα Z
last_acc_value = current_acc_value;
}
}
}
}
```

Π5. Κώδικας κύριου νήματος συστήματος για ανανέωση θέσης

```
private void PDR_Simulate()
{
    //δήλωση τοπικών μεταβλητών συναρτήσεως

    //περιέχει την κατεύθυνση του χρήστη κατά το βήμα που ανιχνεύθηκε υπολογισμένο σε rad
    double radians;
    //περιέχει την συνιστώσα X της απόστασης που διένυσε ο χρήστης με το βήμα του (σε μέτρα)
    double distance_x = 0;
    //περιέχει την συνιστώσα Y της απόστασης που διένυσε ο χρήστης με το βήμα του (σε μέτρα)
    double distance_y = 0;
    //περιέχει την συνολική απόσταση που διένυσε ο χρήστης με το βήμα του (σε μέτρα)
    double step_length = 0;

    //για όσο το πρόγραμμά μας τρέχει κάνει τα εξής
    while (running)
    {
        //περίμενε για ένα μικρό διάστημα
        Thread.Sleep(stride_check_delay);

        //εάν έχει ανιχνευθεί κάποιο πιθανό βήμα από την συνάρτηση ανίχνευσης βήματος
        if (STRIDE)
        {
            //γίνεται έλεγχος για το αν τα δεδομένα χρησιμοποιούνται
            //από άλλο νήμα. Αν χρησιμοποιούνται, τότε συνεχίζουμε
            //να ελέγχουμε μέχρι να ελευθερωθούν
            while ( (DATA_LOCKED == "COLLECTER") || (DATA_LOCKED == "STRIDE_CHECK") )
            {}
            //όταν ελευθερωθούν τα δεδομένα τότε κλειδώνονται από το
            //παρόν νήμα
            DATA_LOCKED = "LAST_CHECK";

            //Εκκαθάριση στοιχείων
            //σε αυτό το σημείο κάνουμε μια εκκαθάριση της μνήμης του προγράμματος
            //κρατώντας μόνο όσα στοιχεία είναι απαραίτητα για την ορθή λειτουργία του
            //με λίγα λόγια, κρατάμα όσα στοιχεία είναι απαραίτητα για το φιλτράρισμα δεδομένων
            z_acc_filtered_storage.Clear();
            x_acc_filtered_storage.Clear();
            y_acc_filtered_storage.Clear();
        }
    }
}
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
degrees_filtered_storage.Clear();
for (int x = filtered_z_acc_array.Count - 1; (x >= 0) && (x >= filtered_z_acc_array.Count -
moving_average_elements); x--)
{
    z_acc_filtered_storage.Add((double)z_acc_array[x]);
    x_acc_filtered_storage.Add((double)x_acc_array[x]);
    y_acc_filtered_storage.Add((double)y_acc_array[x]);
    degrees_filtered_storage.Add((double)degrees_array[x]);
}
//αφαιρούμε τα στοιχεία που βρίσκονται πριν το στοιχείο
//όπου ανιχνεύθηκε το πιθανό βήμα
z_acc_array.RemoveRange(0, last_stride_element + 1);
filtered_z_acc_array.RemoveRange(0, last_stride_element + 1);
x_acc_array.RemoveRange(0, last_stride_element + 1);
filtered_x_acc_array.RemoveRange(0, last_stride_element + 1);
y_acc_array.RemoveRange(0, last_stride_element + 1);
filtered_y_acc_array.RemoveRange(0, last_stride_element + 1);
degrees_array.RemoveRange(0, last_stride_element + 1);
filtered_degrees_array.RemoveRange(0, last_stride_element + 1);
//Τέλος εκκαθάρισης

//εδώ κάνουμε μερικούς επιπλέον ελέγχους για να σιγουρευτούμε
//ότι το βήμα είναι έγκυρο
//θα πρέπει διαφορά της μέγιστης επιτάχυνσης στον άξονα Z με την ελάχιστη που
ανιχνεύθηκε
//να είναι μεγαλύτερη από ένα κατώτατο κατώφλι που έχουμε ορίσει
//(πρόληψη λανθασμένων ανιχνεύσεων απο κραδασμούς σώματος χρήστη)
if (stride_acc_max - stride_acc_min >= min_acc_diff)
{
    //το ίδιο θα πρέπει να ισχύει και για τις διαφορές των μέγιστων και ελάχιστων
επιταχύνσεων
    //στους άξονες X και Y.
    if ((stride_acc_X_max - stride_acc_X_min >= min_hor_acc_diff) || (stride_acc_Y_max -
stride_acc_Y_min >= min_hor_acc_diff)) //ki an mas ikanopoiei i diafora tis orizontias epitahynsis
    {
        //πλέον έχουμε σιγουρευτεί για το βήμα του χρήστη
        //υπολογίζουμε την απόσταση βήματος
        step_length = (System.Math.Pow(stride_acc_max - stride_acc_min, 0.25)) * K;
        //υπολογίζουμε την κατεύθυνση σε rad
        radians = (step_heading * Math.PI) / 180;
        //υπολογίζουμε την συνιστώσα X της απόστασης που διανύθηκε
        distance_x = step_length * Math.Sin(radians);
```

Σύστημα προσδιορισμού θέσης με την χρήση τεχνικής στίγματος εξ αναμετρήσεως

```
//υπολογίζουμε την συνιστώσα Y της απόστασης που διανύθηκε
distance_y = step_length * Math.Cos(radians);
//ανανώνουμε την συνεταγμένη X προσθέτοντας την απόσταση που διανύθηκε στον
άξονα X
X = X + distance_x;
//ανανώνουμε την συνεταγμένη Y προσθέτοντας την απόσταση που διανύθηκε στον
άξονα Y
Y = Y + distance_y;

//αυξάνουμε κατά ένα τον αριθμό των βημάτων που ανιχνεύθηκαν
strides_counted++;
}
}

//θέτουμε την μεταβλητή ανίχνευσης βήματος σε 'ΨΕΥΔΕΣ'
//έτσι ώστε η συνάρτηση ανίχνευσης βήματος να αρχίσει και πάλι την λειτουργία της
STRIDE = false;

//ελευθέρωσε τα δεδομένα έτσι ώστε να μπορούν να χρησιμοποιηθούν
//από άλλα νήματα
DATA_LOCKED = "";
}
}
}
```

ΟΡΟΛΟΓΙΑ

Ξένος όρος	Ελληνικός Όρος	Ορισμός
Dead reckoning technique	Τεχνική στίγματος εξ αναμετρήσεως	Αλγόριθμοι για πλοήγηση σε εσωτερικούς χώρους βασισμένοι στην ανίχνευση των βημάτων του χρήστη
Lowpass filter	Κατωπερατό φίλτρο	Φίλτρο που χρησιμοποιείται για να αποκόψει τις τιμές ενός σήματος που βρίσκονται πάνω από κάποιο όριο.
Bandpass filter	Ζωνοπερατό φίλτρο	Φίλτρο που χρησιμοποιείται για να αποκόψει τις τιμές ενός σήματος που βρίσκονται πάνω και κάτω από κάποιο όριο.
Information fusion	Σύντηξη πληροφορίας	Ο συνδυασμός δεδομένων που προέρχονται από ετερογενείς πηγές έτσι ώστε η προκύπτουσα πληροφορία να είναι καλύτερη απ' ότι θα ήταν εάν αυτές οι πηγές χρησιμοποιούνταν ξεχωριστά.
Template matching algorithm	Αλγόριθμος ομοιότητας προτύπων	Αλγόριθμος ο οποίος βασίζεται στην σύγκριση δειγμάτων με κάποια αυτοαναπροσαρμοζόμενα πρότυπα.

Zero crossing method	Μέθοδος μηδενικών διασταυρώσεων	Μέθοδος που βασίζεται στην ανίχνευση διασταύρωσης του σήματος με τον άξονα Χ, δηλαδή όταν αλλάζει πρόσημο.
Precision	Ακρίβεια	Το ποσοστό του χρόνου που ένα σύστημα προσδιορισμού θέσης παρέχει συγκεκριμένη ορθότητα (accuracy).
Peak detection	Ανίχνευση κορυφών	Μέθοδος η οποία ανιχνεύει τα τοπικά ανώτατα και κατώτατα σημεία (κορυφές) ενός σήματος.
Moving average filter	Φίλτρο μέσου όρου	Φίλτρο που βασίζεται στην εξαγωγή του μέσου όρου των τελευταίων Ν στοιχείων.
Moving window integrator	Ολοκληρωτής κυλιόμενου παραθύρου	Διαδικασία ολοκλήρωσης για ομαλοποίηση ενός σήματος.
Squaring	Διαδικασία τετραγωνισμού	Διαδικασία που οδηγεί σε θετικές τιμές σήματος και ευνοεί τις μεγάλες τιμές περισσότερο από τις μικρές.
Fiducial point	Καθοδηγητικό σημάδι	Η θέση μιας κορυφής σε ένα σήμα.

Normalized cross-correlation signal	Ομαλοποιημένο σήμα συσχέτισης	Σήμα που δείχνει κατά πόσον δύο διανύσματα μοιάζουν μεταξύ τους.
Pitch	Κλίση επιφάνειας	Η κλίση της επιφάνειας ενός δαπέδου.
Variance	Διασπορά	Η διασπορά των τιμών ενός σήματος.

ΑΚΡΩΝΥΜΙΑ – ΣΥΝΤΜΗΣΕΙΣ – ΑΡΤΙΚΟΛΕΞΑ

DR	Dead Reckoning
PDR	Pedestrian Dead Reckoning
GPS	Global Positioning System
PDA	Personal Digital Assistant
RFID	Radio Frequency IDentification
RSSI	Receive Signal Strength Indicator
WLAN	Wireless Local Area Network
ECG	Electrocardiogram

ΑΝΑΦΟΡΕΣ

- [1] Ο. Σέκκας, “Προσδιορισμός Θέσης μέσω Σύντηξης Δεδομένων Αισθητήρων”, Διπλωματική Εργασία, Αθήνα, Απρίλιος 2006.
- [2] R. Want et al., “The Active Badge Location System, ”ACM Trans. Information Systems, Jan.1992, pp. 91-102.
- [3] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, “The anatomy of a context-aware application”, In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1999), Seattle, WA, August 1999, ACM Press, pp. 59-68.
- [4] Ascension Technology Corporation, PO Box 527, Burlington, VT 05402, Technical Description of DC Magnetic Trackers, 2001.
- [5] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support System”, Proc. 6th Ann.Int’l Conf. Mobile Computing and Networking (Mobicom 00), ACM Press, New York, 2000, pp. 32-43.
- [6] P. Bahl and V. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System”, Proc. IEEE Infocom 2000, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 775-784.
- [7] J. Krumm et al., “Multi-Camera Multi-Person Tracking for Easy Living”, Proc. 3rd IEEE Int’l Workshop Visual Surveillance, IEEE Press, Piscataway, N.J., 2000, pp. 3-10.
- [8] R. J. Orr and G. D. Abowd, “The Smart Floor: A Mechanism for Natural User Identification and Tracking”, Proc. 2000 Conf. Human Factors in Computing Systems (CHI 2000), ACM Press, New York, 2000.

- [9] L. Fang, P. J. Antsaklis, L. Montestruque, M. Brett McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie, "Design of a Wireless Assisted Pedestrian Dead Reckoning System – The NavMote Experience", IEEE Transactions on Instrumentation and Measurement, 2005, vol. 54, Issue 6, pp. 2342- 2358.

- [10] C. Randell, C. Djalllis, H. Muller, "Personal Position Measurement Using Dead Reckoning", iswc, 7th IEEE International Symposium on Wearable Computers (ISWC'03), 2003, p.166.

- [11] M. Dippold, "Personal Dead Reckoning with Accelerometers", 3rd International Forum on Applied Wearable Computing, IFAWC2006, Bremen, Germany, 2006.

- [12] S. Beauregard, M. Klepal, Widyawan, "Positioning in WearIT@Work", Liaison workshop, Athens, 2006.

- [13] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, "Automatic Step Detection in the Accelerometer Signal", 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), Volume 13, pp. 80-85.

- [14] M. J. Caruso, "Application of magnetoresistive sensors in navigation systems," Sensors and Actuators, SAE SP-1220, 1997, pp. 15-21.

- [15] National Geophysical Data Center, NOAA Satellite and Information Service, <http://www.ngdc.noaa.gov/geomagmodels/Declination.jsp>

- [16] Magnetic declination, Wikipedia, http://en.wikipedia.org/wiki/Magnetic_declination

- [17] P. S.Maybeck, "Stochastic Models, Estimation, and Control", vol. 1, ACADEMIC PRESS, 1979.
- [18] C. E. Bauby and A. D. Kuo, "Active control of lateral balance in human walking", in Journal of Biomechanics, 2000, pp. 1433–1440.
- [19] V. Papataxiarhis, V. Riga, V. Nomikos, O. Sekkas, K. Kolomvatsos, V. Tsetsos, P. Papageorgas, S. Vourakis, V. Xouris, S. Hadjiefthymiades and G. Kouroupetroglou, "MNISIKLIS: Indoor Location Based Services for All", in Proceedings of the 5th International Symposium on LBS and TeleCartography, Salzburg, Austria, November 26th-28th, 2008, pp. 263-282.
- [20] S.W. Lee and K. Mase, "Pi-Navi: Finding Location of a Pedestrian Using Pace Tracking", Joho Shori Gakkai Shinpojiumu Ronbunshu, vol. 2001, no. 5, pp. 231-232.
- [21] Q. Ladetto, "On Foot Navigation: Continuous Step Calibration using Both Complimentary Recursive Prediction and Adaptive Kalman Filtering", Proceedings of ION GPS 2000 (January 26-28, 2000, Anaheim, California), Alexandria, VA, USA, Institute of Navigation, pp. 1735-1740.
- [22] C. Tom Judd, "A Personal Dead Reckoning Module", Proceedings of the 10th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS 1997, Kansas City, MO, pp. 47-51.