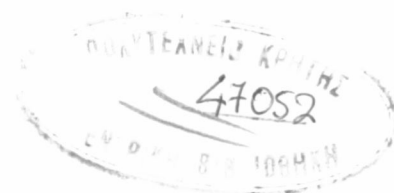


## Διπλωματική Εργασία



Sequential Monte Carlo μέθοδος προς βέλτιστο  
υπολογισμό σημάτων με χρονικά μεταβαλλόμενο  
πλάτος και φάση

Παλαμίδης Αλέξανδρος

### Εξεταστική Επιτροπή

Καθηγητής Σιδηρόπουλος Νικόλαος (Επιβλέπων)  
Καθηγητής Μαράς Ανδρέας  
Καθηγητής Πατεράκης Μιχάλης

Χανιά , Σεπτέμβριος 2002

## Περίληψη

Σε αυτή την εργασία έχουμε ως σκοπό τον υπολογισμό ακολουθιακά στον χρόνο (on-line) σημάτων, τα οποία εκφράζονται σε κάθε χρονική στιγμή ως το άθροισμα συνημιτόνων με μεταβαλλόμενο πλάτος και φάση. Επίσης το σήμα υπόκειται σε παραμόρφωση λόγω λευκού Γκαουσιανού (Gaussian) θορύβου. Η λύση που προτείνουμε σε αυτό το πρόβλημα, το οποίο είναι στην ουσία ένα πρόβλημα φιλτραρίσματος, είναι η μέθοδος *Sequential Monte Carlo (SMC)*.

Για να καταλήξουμε στο επιθυμητό αποτέλεσμα δηλαδή στην χωρίς παραμορφώσεις, λόγω θορύβου, ανάκτηση του μεταδιδόμενου σήματος θα χρησιμοποιήσουμε κάποιες τεχνικές μείωσης της διασποράς και μια σειρά *Markov Chain Monte Carlo (MCMC)* βημάτων.

## Ευχαριστίες

Θα ήθελα να εκφράσω την εκτίμηση μου και τις θερμές μου ευχαριστίες στον καθηγητή Ανδρέα Μαρά, που μου ανέθεσε αυτή την πολύ ενδιαφέρουσα διπλωματική εργασία και με βοήθησε καθ' όλη την διάρκεια της προσπάθειας μου να εκπονηθεί αυτή. Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή Νικόλαο Σιδηρόπουλο, ο οποίος δέχθηκε να είναι ο επιβλέπων καθηγητής μου σε αυτήν την εργασία, καθώς και τον καθηγητή Μιχάλη Πατεράκη ,για τον χρόνο που αφιέρωσαν για την ανάγνωση της εργασίας και τις παρατηρήσεις τους.

Τέλος θα ήθελα να ευχαριστήσω τον φίλο και συμφοιτητή Γιάννη Γεωργίου για την άριστη συνεργασία που είχαμε σε αυτά τα πέντε χρόνια φοίτησης μου στο Πολυτεχνείο Κρήτης.

## Περιεχόμενα

|             |   |        |
|-------------|---|--------|
| Κεφάλαιο 1. | Εισαγωγή                                | σελ. 8 |
| Κεφάλαιο 2. | Περιγραφή συστήματος                    | 9      |
| Κεφάλαιο 3. | Rao –Blackwellized Particle Filtering   | 13     |
| 3.1.        | Επιμέρους Πιθανότητα                    | 13     |
| 3.2.        | Ο Αλγόριθμος                            | 15     |
| 3.2.1.      | Importance Sampling                     | 15     |
| 3.2.1.α.    | Επιλογή της importance κατανομής        | 16     |
| 3.2.2.      | Selection Sampling                      | 17     |
| 3.2.2.α.    | Stratified resampling                   | 17     |
| 3.2.3.      | MCMC Sampling                           | 18     |
| Κεφάλαιο 4. | Παρουσίαση των αποτελεσμάτων            | 20     |
| 4.1.        | Γραφικές παραστάσεις των αποτελεσμάτων  | 20     |
| 4.2.        | Σύγκριση RB και EKF φίτρων              | 23     |
| Κεφάλαιο 5. | Συμπεράσματα και μελλοντικές επεκτάσεις | 24     |
| 5.1.        | Συμπεράσματα                            | 25     |
| 5.2.        | Μελλοντικές επεκτάσεις                  | 26     |
| Παράρτημα Α | Παρουσίαση επιμέρους αλγορίθμων         | 27     |
| A.1         | Αλγόριθμος Metropolis-Hasting (M-H)     | 27     |
| A.2         | Φίλτρο Kalman                           | 28     |
| A.3         | Backward information φίλτρο             | 29     |
| A.4         | Συνάρτηση δειγματοληψίας M-H            | 30     |

|                     |   |    |
|---------------------|---|----|
| <b>Παράρτημα Β</b>  | <b>Κώδικας</b>                              | 31 |
| <b>B.1</b>          | <b>Σχέδιο Κώδικα</b>                        | 31 |
| <b>B.2</b>          | <b>Rao–Blackwellized Particle Filtering</b> | 33 |
| <b>B.3</b>          | <b>Μέσο Τετραγωνικό Σφάλμα</b>              | 42 |
| <b>Βιβλιογραφία</b> |   | 44 |

## Κατάλογος διαγραμμάτων

|  |         |
|--|---------|
| Σχήμα 1..Παρατηρήσεις $y_i$                                      | σελ. 11 |
| Σχήμα 2..Κανονική κατανομή.                                      | 12      |
| Σχήμα 3..Διάγραμμα των φάσεων του αρχικού σήματος.               | 21      |
| Σχήμα 4..Διάγραμμα των εκτιμήσεων των φάσεων του σήματος.        | 21      |
| Σχήμα 5..Διάγραμμα των τιμών πλάτους του αρχικού σήματος.        | 22      |
| Σχήμα 6..Διάγραμμα των εκτιμήσεων των τιμών πλάτους του σήματος. | 22      |

## Κατάλογος Κυριότερων Συμβόλων και Όρων

- $\alpha_t$  :  $(\alpha_t(1), \dots, \alpha_t(K))$
- $\theta_t$  :  $(\theta_t(1), \dots, \theta_t(K))$
- $\tilde{\theta}$  :  $\theta$  από Importance Sampling Step
- $\bar{\theta}$  :  $\theta$  από Selection Step
- $E(\mathbf{x})$  : Μέση τιμή του διανύσματος  $\mathbf{x}$
- **EKF**: Extended Kalman Filter
- **MCMC**: Markov Chain Monte Carlo
- **M-H**: Metropolis-Hasting
- $n_{\mathbf{x}}$  : διαστάσεις του διανύσματος  $\mathbf{x}$
- $N(\mathbf{m}, \sigma)$ : Κανονική κατανομή με μέση τιμή  $\mathbf{m}$  και διακύμανση  $\sigma$
- **RB** : Rao – Blackwellized Particle Filtering
- **SMC**: Sequential Monte Carlo
- $U([i,j])$ : Ομοιόμορφη κατανομή από  $i$  ως  $j$ , με  $i \leq j$
- $\mathbf{x}_{i:j}$  :  $(x_i, x_{i+1}, \dots, x_j)$ , με  $i \leq j$

# Κεφάλαιο 1

## 1.1.Εισαγωγή

Στην στατιστική επεξεργασία σήματος ένα εξαιρετικά ενδιαφέρον ζήτημα είναι ο υπολογισμός σε πραγματικό χρόνο της φάσης και του πλάτους χρονικά μεταβαλλόμενων σημάτων. Το ζήτημα αυτό απαντάται σε ποικίλες εφαρμογές, όπως στην επεξεργασία φωνής, στις τηλεπικοινωνίες για αποδιαμόρφωση σημάτων, σε συστήματα ραντάρ, κι αλλού. Ο υπολογισμός σε πραγματικό χρόνο της φάσης και του πλάτους χρονικά μεταβαλλόμενων σημάτων μπορεί κάλλιστα να θεωρηθεί και να αντιμετωπιστεί ως ένα σύνθετο πρόβλημα βέλτιστου φιλτραρίσματος. Τέλος, πρέπει να σημειωθεί ότι το πρόβλημα δεν επιδέχεται καμία λύση κλειστής μορφής λόγω της μη γραμμικότητας του.

Για την επίλυση του προβλήματος κατά τα παρελθόντα έτη έχουν προταθεί αρκετές μεθοδολογίες. Η βάση στην οποία στηρίζονται είναι οι κλασσικές μέθοδοι που επιλύουν προβλήματα φιλτραρίσματος όπως το Extended Kalman Filter (EKF), το Gaussian sum filter καθώς και διάφορες αριθμητικές τεχνικές ολοκλήρωσης.

Τα δύο παραπάνω φίλτρα έχουν το πλεονέκτημα ότι είναι σχετικά υπολογιστικά “φθηνά”, αλλά δυστυχώς είναι σε αρκετά μεγάλο βαθμό ανακριβή και τα λάθη που προξενούν είναι πολύ δύσκολο να διαγνωστούν σε πραγματικό χρόνο. Από την άλλη πλευρά οι αριθμητικές τεχνικές ολοκλήρωσης είναι δύσκολο να υλοποιηθούν και η σύγκλιση του σφάλματος (ή, λάθους) προσέγγισης στο μηδέν είναι ανάλογη των διαστάσεων του χώρου καταστάσεων, γεγονός που καθιστά αυτές τις τεχνικές εξαιρετικά δύσκολα εφαρμόσιμες σε πολυδιάστατα μοντέλα.

Η λύση που θα προτείνουμε για την επίλυση του προβλήματος βέλτιστου φιλτραρίσματος για μοντέλα μη γραμμικού, μη Gaussian χώρου καταστάσεων στηρίζεται στην μέθοδο Sequential Monte Carlo (SMC) [3]-[5].

Η SMC μέθοδος χρησιμοποιεί τυχαία δείγματα, που λαμβάνονται από κατανομές, τα οποία διαδίδει και τροποποιεί στο χρόνο χρησιμοποιώντας διαδοχικά ένα συνδυασμό sequential importance sampling, selection resampling και Markov Chain Monte Carlo (MCMC) βημάτων. Η συγκεκριμένη μέθοδος είναι εύκολα υλοποιήσιμη και η σύγκλιση του λάθους προσέγγισης στο μηδέν είναι ανεξάρτητη της διάστασης του χώρου καταστάσεων.



## Κεφάλαιο 2

### 2.1 Περιγραφή συστήματος

Το μοντέλο με το οποίο θα διαπραγματευτούμε αποτελείται από ένα μεταδιδόμενο σήμα ημιτονικής μορφής με χρονικά μεταβαλλόμενο πλάτος και φάση, το οποίο κατά την μετάδοση του παραμορφώνεται λόγω λευκού Gaussian θόρυβου. Έτσι το σήμα  $y(t)$  που έχουμε ως παρατήρηση στο δέκτη μπορεί να εκφραστεί ως εξής :

$$y_t = \sum_{k=1:K} a_t(k) \cos(\theta_t(k)) + \sigma_w w_t, \quad K \geq 1 \quad (1)$$

Στο σχήμα 1 φαίνεται η γραφική παράσταση του  $y_t$  που λάβαμε από την προσομοίωση για  $t = 1000$  και  $K = 3$ . Ο θόρυβος  $w_t$  είναι λευκός Gaussian και επομένως μπορεί να θεωρηθεί ως ανεξάρτητη τυχαία μεταβλητή που ακολουθεί την τυποποιημένη κανονική κατανομή  $w_t \sim N(0,1)$ .

Στο σχήμα 2 παρουσιάζεται η γραφική παράσταση της κανονικής κατανομής για  $\mu = 0$ .

Οι συναρτήσεις πλάτους και φάσης είναι μεταξύ τους ανεξάρτητες τυχαίες μεταβλητές και θεωρούμε ότι η μεταβολή της φάσης στο χρόνο αποτελεί μια διαδικασία Markov με συνάρτηση μετάβασης πρώτου βαθμού που εκφράζεται ως εξής:

$$\theta_t / \theta_{t-1} \sim p(d\theta_t / \theta_{t-1}) \equiv p(\theta_t / \theta_{t-1}) d\theta_t \quad (2)$$

Να σημειωθεί ότι μεγαλύτερου βαθμού εξαρτήσεις είναι εφικτές και δεν τροποποιούν τον αλγόριθμο που θα παρουσιαστεί παρακάτω. Ο περιορισμός σε διαδικασίες Markov πρώτου βαθμού γίνεται όχι μόνο για λόγους απλοποίησης της παρουσίασης, αλλά και διότι τα περισσότερα σήματα στις τηλεπικοινωνίες ακολουθούν διαδικασίες Markov πρώτου βαθμού.

Όσον αφορά την μεταβολή του πλάτους στο χρόνο θεωρούμε ότι το πλάτος ακολουθεί ένα Gaussian μοντέλο του τύπου :

$$a_t = A a_{t-1} + B u_t \quad (3)$$

όπου  $u_t \sim N(\mathbf{0}, \mathbf{I})$ , με  $\mathbf{0}$  και  $\mathbf{I}$  να είναι αντίστοιχα ο μηδενικός και μοναδιαίος τετραγωνικοί πίνακες τάξης  $K$ .

Να αναφέρουμε επίσης ότι οι αρχικές τιμές (για  $t = 0$ ) του πλάτους και της φάσης είναι :

$$\theta_0 \sim U([0, \pi]^k) \quad \text{και} \quad a_0 \sim N(a_0, P_0).$$

Προκειμένου να μετασχηματίσουμε την (1) σε μια πιο εύχρηστη μορφή θέτουμε:

$$C(\theta_t) \equiv (\cos(\theta_t(1)), \dots, \cos(\theta_t(K))) \quad \text{και} \quad D = \sigma_w.$$

Ετσι η (1) γίνεται :

$$y_t = C(\theta_t) a_t + D w_t \quad (4)$$

Τέλος, πρέπει να σημειωθεί ότι οι ακολουθίες θορύβου  $u_t$  και  $w_t$  θεωρούνται μεταξύ τους ανεξάρτητες και ανεξάρτητες από την αρχική κατάσταση του συστήματος  $\{a_0, \theta_0\}$ .

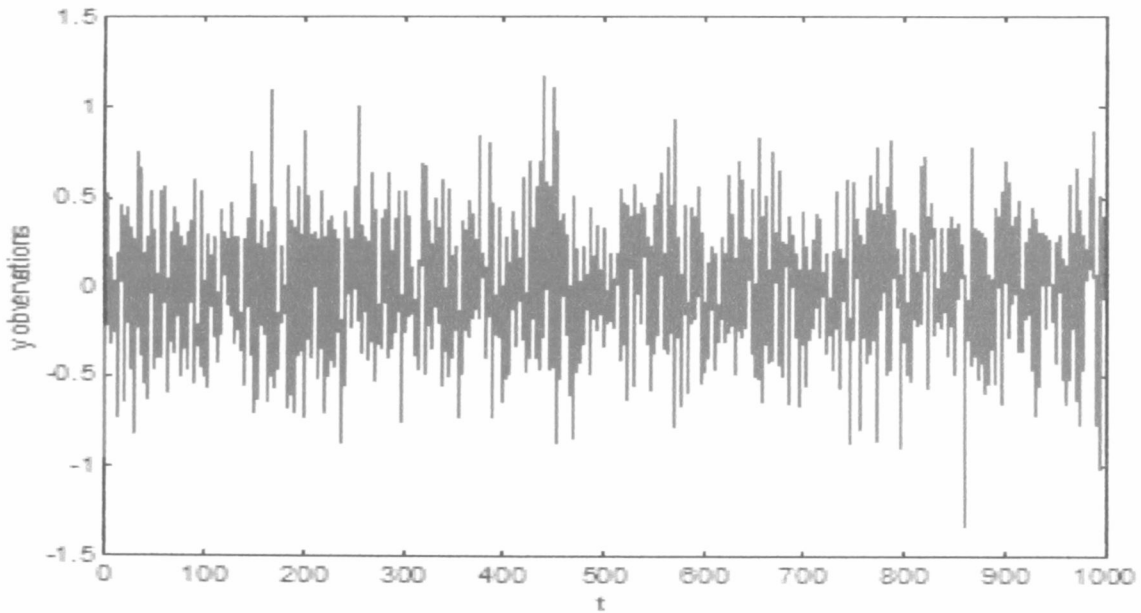
Το ζητούμενο σε αυτό το μοντέλο που έχει καθοριστεί από τις εξισώσεις (2), (3) και (4) είναι να υπολογίσουμε ακολουθιακά στο χρόνο κάποια χαρακτηριστικά της επιμέρους κατανομής με πυκνότητα πιθανότητας :

$$p(a_{0:t}, \theta_{0:t} / y_{1:t}) = p(y_{1:t} / a_{0:t}, \theta_{0:t}) p(a_{0:t}, \theta_{0:t}) \left( \int p(y_{1:t} / a_{0:t}, \theta_{0:t}) p(a_{0:t}, \theta_{0:t}) da_{0:t} d\theta_{0:t} \right)^{-1}$$

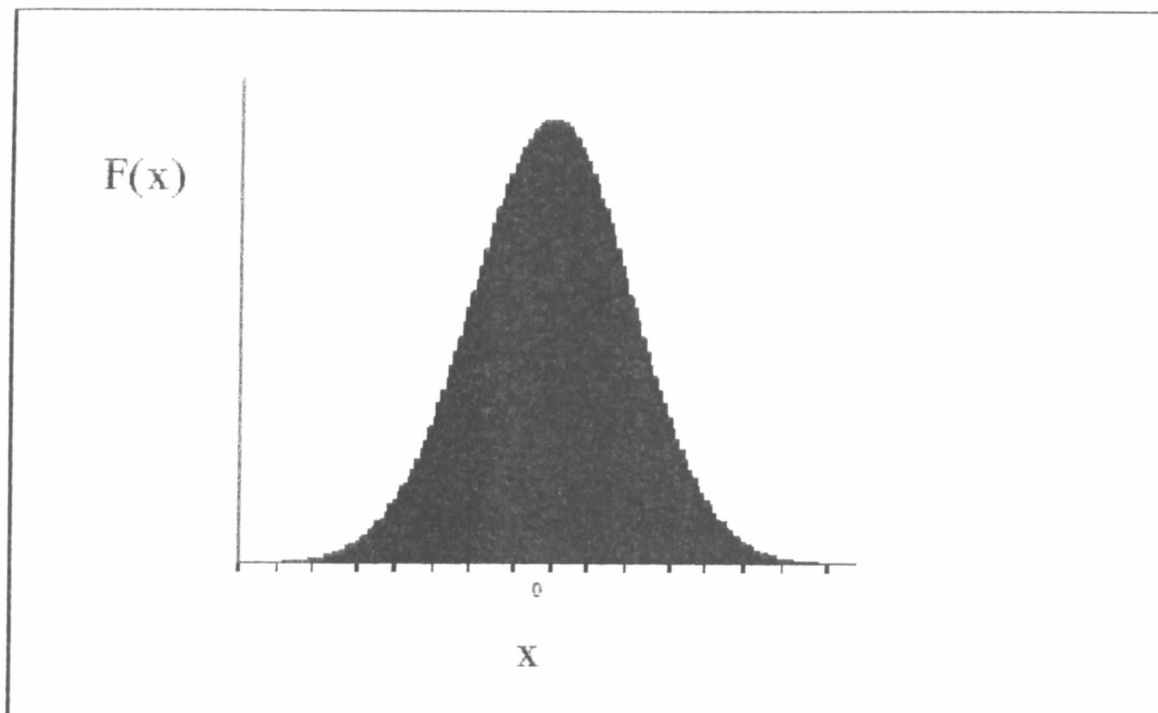
Τα χαρακτηριστικά που πρόκειται να υπολογίσουμε είναι οι εκτιμήσεις για τις φάσεις  $E(\theta_t / y_{1:t})$  και τα πλάτη  $E(a_t / y_{1:t})$  του αρχικού σήματος.

Η στρατηγική που πρόκειται να ακολουθήσουμε θα είναι η υλοποίηση μιας μεθόδου παρόμοιας με τις κλασικές SMC μεθόδους ,(π.χ., bootstrap filter) , η οποία όμως θα χρησιμοποιεί αποτελεσματικά την δομή και τα χαρακτηριστικά του μοντέλου που έχουμε ορίσει.

Έτσι θα χρησιμοποιήσουμε ένα συνδυασμό importance sampling, selection resampling και Markov Chain Monte Carlo (MCMC) βημάτων προκειμένου να επεξεργαστούμε ακολουθιακά στο χρόνο έναν αριθμό δειγμάτων που ακολουθούν την κατανομή  $p(a_{0:t}, \theta_{0:t} / y_{1:t})$  ,  $t \in (1, T)$ . Η καινοτομία που έχουμε να επιδείξουμε σε αυτή την τεχνική είναι η εστίαση στον υπολογισμό της  $p(d\theta_{0:t} / y_{1:t})$  και, αντίθετα με τις κλασικές SMC μεθόδους, όχι της  $p(a_{0:t}, \theta_{0:t} / y_{1:t})$ . Στην συνέχεια θα είναι εύκολο με την βοήθεια ενός απλού *Kalman* φίλτρου να υπολογίσουμε τις ζητούμενες εκτιμήσεις για το πλάτος  $a_t$  . Η τεχνική αυτή είναι γνωστή σαν RAO- BLACKWELLIZED particle filtering [5].



Σχημα 1. Οι παρατηρήσεις  $y_t$  που πήραμε προσομοιώνοντας το σύστημα για  $K=3$ ,  $t = 1000$ ,  $\sigma_w = 0.3$ .



Σχήμα 2. Η κανονική κατανομή με μέση τιμή  $m = 0$ . Η πιθανότητα απεικονίζεται με το γραμμοσκιασμένο εμβαδόν. Ισχύει ότι  $\int F(x) = 1$ .

## Κεφάλαιο 3

### RAO – BLACKWELLIZED PARTICLE FILTERING

Η παρουσίαση του RAO-BLACKWELLIZED particle filtering θα γίνει σε δύο στάδια. Στο πρώτο στάδιο θα δοθεί μια απόδειξη του γιατί είναι εφικτό και επιθυμητό να περιοριστούμε στον υπολογισμό της  $p(d\theta_{0:t}/y_{1:t}) \equiv p(\theta_{0:t}/y_{1:t})d\theta_{0:t}$  αντί της  $p(a_{0:t}, \theta_{0:t}/y_{1:t})$  και πως αυτό οδηγεί στην βελτίωση της απόδοσης του αλγορίθμου και στο δεύτερο και κυριότερο στάδιο, θα δοθεί μια αναλυτική περιγραφή του αλγορίθμου που χρησιμοποιείται για τον υπολογισμό αυτής.

#### 3.1 Επιμέρους Πιθανότητα

Επαναλαμβάνουμε τις εξισώσεις (2), (3) και (4) που περιγράφουν το μοντέλο προς χάριν ευκολίας αναφοράς :

$$\theta_t/\theta_{t-1} \sim p(d\theta_t/\theta_{t-1}) \equiv p(\theta_t/\theta_{t-1})d\theta_t \quad (2)$$

$$a_t = A a_{t-1} + B u_t \quad \text{με } u_t \sim N(0, I_{nu}) \quad (3)$$

$$y_t = C(\theta_t)a_t + D w_t \quad \text{με } w_t \sim N(0, 1) \quad (4)$$

Όσον αφορά την  $p(a_{0:t}/y_{1:t})$  έχουμε :

$$p(a_{0:t}/y_{1:t}) = \int p(a_{0:t}/y_{1:t}, \theta_{0:t}) p(d\theta_{0:t}/y_{1:t}) \quad (5)$$

Η  $p(d\theta_{0:t}/y_{1:t})$  μέσω μιας SMC μεθόδου επιδέχεται μια προσέγγιση της μορφής

$$\hat{p}_N(d\theta_{0:t}/y_{1:t}) = \sum_{i=1:N} w_t^{(i)} \delta_{\theta_{0:t}}^{(i)}(d\theta_{0:t}), \quad (6\alpha)$$

με

$$w_t^{(i)} > 0 \quad \text{και} \quad \sum_{i=1:N} w_t^{(i)} = 1. \quad (6\beta)$$

Για να μπορέσουμε να πραγματοποιήσουμε την ανωτέρω προσέγγιση πρέπει να έχουμε μερική γνώση της  $p(\theta_k/\theta_{k-1})$ . Έτσι έχουμε:

$$p(d\theta_{0:t}/y_{1:t}) \propto p(\theta_0) \prod_{k=1:K} p(y_k/y_{1:k-1}, \theta_{0:k}) p(\theta_k/\theta_{k-1})$$

όπου η  $p(\theta_k/\theta_{k-1})$  γνωστή και η  $p(y_k/y_{1:k-1}, \theta_{0:k})$  δίδεται απευθείας από το *Kalman* φίλτρο.

Έτσι η (5), λόγω της (6), γίνεται :

$$\hat{p}_N(a_{0:t}/y_{1:t}) = \int p(a_{0:t}/y_{1:t}, \theta_{0:t}) \sum_{i=1:N} w_t^{(i)} \delta_{\theta_{0:t}}^{(i)}(d\theta_{0:t})$$

και συνεπώς

$$\hat{p}_N(a_{0:t}/y_{1:t}) = \sum_{i=1:N} w_t^{(i)} p(a_{0:t}/y_{1:t}, \theta_{0:t}^{(i)}) \quad (7)$$

Έχοντας αυτή την προσέγγιση, μια εκτίμηση του  $E(a_t/y_{1:t})$  υπολογίζεται εύκολα ως εξής :

$$E^{\wedge}_N(a_t/y_{1:t}) = \int a_t \hat{p}_N(a_{0:t}/y_{1:t}) da_{0:t}$$

και συνεπώς

$$E^{\wedge}_N(a_t/y_{1:t}) = \sum_{i=1:N} w_t^{(i)} E(a_t/y_{1:t}, \theta_{0:t}^{(i)}) \quad (8)$$

με την εκτίμηση  $E(a_t/y_{1:t}, \theta_{0:t}^{(i)})$  να δίδεται από το *Kalman* φίλτρο.

Το γεγονός ότι υπολογίζουμε μόνο τις  $p(da_{0:t}/y_{1:t})$ ,  $p(d\theta_{0:t}/y_{1:t})$  και όχι την  $p(da_{0:t}, d\theta_{0:t}/y_{1:t})$  προσδίδει μεγαλύτερη αποτελεσματικότητα στην μέθοδο αφού έτσι μειώνεται η διασπορά. Πράγματι όπως αναφέρεται στο [1] για οποιαδήποτε συνάρτηση  $g$  ισχύει ότι :

$$\text{var} [g(a_{0:t}/y_{1:t})] \geq \text{var} [E\{g(a_{0:t}/y_{1:t}, \theta_{0:t})\}/y_{1:t}]$$

### 3.2 Ο Αλγόριθμος

Ο αλγόριθμος του συγκεκριμένου φίλτρου μπορεί να συνοψισθεί στην εξής έκφραση : Αν την χρονική στιγμή  $t-1$  έχουμε  $N$  δείγματα (particles)  $\{\theta_{0:t-1}^{(i)}\}_{i=1:N}$  κατανεμημένα σύμφωνα με την  $p(d\theta_{0:t-1}/y_{1:t-1})$ , τότε την χρονική στιγμή  $t$  αποκτούμε  $N$  καινούρια δείγματα  $\theta_t^{(i)}$  κατανεμημένα σύμφωνα με την  $p(d\theta_t/y_{1:t},\theta_{0:t})$  ακολουθώντας ένα συνδυασμό importance sampling, selection resampling και Markov Chain Monte Carlo (MCMC) βημάτων. Αναλυτικότερα έχουμε:

#### 3.2.1 Sequential Importance Sampling Step

α) Για  $i=1:N$  δειγματοληπτούμε  $\theta_t^{-(i)} \sim q(d\theta_t/y_{1:t},\theta_{0:t-1})$  και θέτουμε  $\theta_{0:t}^{-(i)} \equiv \{\theta_{0:t-1}^{-(i)}, \theta_t^{-(i)}\}$ , όπου  $q(d\theta_t/y_{1:t},\theta_{0:t-1})$  η importance κατανομή του φίλτρου.

β) Για  $i=1:N$  υπολογίζουμε τα importance βάρη  $w_t^{(i)}$  τα οποία δίνονται από

$$w_t^{(i)} \propto w(\theta_{0:t}^{-(i)}) \propto p(y_t/y_{1:t-1}, \theta_{0:t}^-) p(\theta_t^-/\theta_{t-1}^-) (q(\theta_t^-/y_{1:t}, \theta_{0:t-1}))^{-1}$$

και κατόπιν κανονικοποιούμε τα βάρη έτσι ώστε

$$\sum_{i=1:N} w_t^{(i)} = 1.$$

### 3.2.1.α) Επιλογή της Importance κατανομής

Το κύριο ζήτημα όσον αφορά το Sequential Importance Sampling Step είναι ο καθορισμός της importance κατανομής  $q(d\theta_t/y_{1:t}, \theta_{0:t-1})$  σύμφωνα με την οποία θα γίνεται η δειγματοληψία. Η μόνη αναγκαία συνθήκη για την  $q(d\theta_t/y_{1:t}, \theta_{0:t-1})$  είναι να περιέχει ως όρο την  $p(\theta_t/\theta_{t-1})$ . Η βέλτιστη  $q(d\theta_t/y_{1:t}, \theta_{0:t-1})$  ορίζεται ως εξής:

$$q(d\theta_t/y_{1:t}, \theta_{0:t-1}) \propto p(y_t/y_{1:t-1}, \theta_{0:t}) p(\theta_t/\theta_{t-1}) .$$

Τότε τα βάρη θα έπρεπε να υπολογίζονται σύμφωνα με

$$w(\theta_{0:t}) = p(y_t/y_{1:t-1}, \theta_{0:t-1}) = \int p(y_t/y_{1:t-1}, \theta_{0:t-1}, \tilde{\theta}_t) p(\tilde{\theta}_t/\tilde{\theta}_{t-1})$$

Ο υπολογισμός αυτός όμως δεν μπορεί να γίνει αναλυτικά. Έτσι έχουμε δύο εναλλακτικές προτάσεις για την επιλογή της importance κατανομής .

Η πρώτη πρόταση είναι η απλούστερη και πιο εύκολα υλοποιήσιμη. Χρησιμοποιούμε ως importance κατανομή την  $q(d\theta_t/y_{1:t}, \theta_{0:t-1}) = p(\theta_t/\theta_{t-1})$ . Τότε τα importance βάρη δίνονται απευθείας από το *Kalman* φίλτρο και πιο συγκεκριμένα

$$w(\theta_{0:t}) \propto p(y_t/y_{1:t-1}, \theta_{0:t}).$$

Η δεύτερη πρόταση διορθώνει ένα μειονέκτημα της πρώτης. Πιο συγκεκριμένα, χρησιμοποιεί τις παρατηρήσεις  $y_t$  και με την χρήση κλασσικών μεθόδων φιλτραρίσματος (π.χ., EKF) παράγει πολύ ικανοποιητικές importance κατανομές.



### 3.2.2 Selection Step

Για  $i=1:N$  πολλαπλασιάζουμε ή απομακρύνουμε τα δείγματα  $\{\tilde{\theta}_{0:t}^{(i)}\}$  ανάλογα με τα κανονικοποιημένα βάρη που αντιστοιχούν στο καθένα ώστε να δημιουργήσουμε ξανά  $N$  δείγματα  $\{\tilde{\theta}_{0:t}^{(i)}\}$ .

Με το importance sampling step έχουμε δημιουργήσει μια κατανομή με βάρη που εκφράζεται ως :

$$p^{-N}(d\theta_{0:t}/y_{1:t}) = \sum_{i=1:N} w_t^{(i)} \delta_{\tilde{\theta}_{0:t}}^{(i)}(d\theta_{0:t})$$

Σκοπός του selection step είναι η παραγωγή μιας χωρίς βάρη προσεγγιστικής κατανομής  $p^{-N}(d\theta_{0:t}/y_{1:t})$  της  $p^{-N}(d\theta_{0:t}/y_{1:t})$ . Δηλαδή

$$p^{-N}(d\theta_{0:t}/y_{1:t}) = N^{-1} \sum_{i=1:N} \delta_{\tilde{\theta}_{0:t}}^{(i)}(d\theta_{0:t})$$

Η βάση στην οποία στηρίζεται το selection step είναι η απομάκρυνση των “αδύναμων” δειγμάτων (αυτά με μικρά importance βάρη) και η ενίσχυση/πολλαπλασιασμός των “ισχυρών” δειγμάτων (αυτά με μεγάλα importance βάρη). Για την υλοποίηση του selection step έχουν προταθεί αρκετά σχήματα και τεχνικές, όπως το multinomial sampling ή το stratified resampling. Εδώ θα παρουσιάσουμε το stratified sampling scheme.

#### 3.2.2.a) Stratified Sampling Scheme

Στο stratified sampling scheme η συνάρτηση επιλογής αντιστοιχεί σε κάθε δείγμα  $\tilde{\theta}_{0:t}^{(i)}$  ένα συντελεστή  $N_i$  που ανήκει στους φυσικούς αριθμούς τέτοιο ώστε

$$\sum_{i=1:N} N_i^{(i)} = 1$$

και ούτως ώστε να προκύψουν και πάλι  $N$  δείγματα  $\tilde{\theta}_{0:t}^{(i)}$  με κατανομή  $p^{-N}(d\theta_{0:t}/y_{1:t})$ .

Για την  $p^{-N}(d\theta_{0:t}/y_{1:t})$  ισχύει :

$$p^{-N}(d\theta_{0:t}/y_{1:t}) = N^{-1} \sum_{i=1:N} N_t^{(i)} \delta_{\theta_{0:t}}^{-(i)}(d\theta_{0:t})$$

Άρα

$$p^{-N}(d\theta_{0:t}/y_{1:t}) = N^{-1} \sum_{i=1:N} \delta_{\theta_{0:t}}^{-(i)}(d\theta_{0:t}).$$

Στα βάρη των δειγμάτων που δεν απομακρύνονται δίνεται η τιμή  $(1/N)$  και τα δείγματα αυτά είναι προσεγγιστικά κατανεμημένα σύμφωνα με την  $p(d\theta_{0:t}/y_{1:t})$ .

### 3.2.3 MCMC Step

Για  $i=1:N$  εφαρμόζουμε στα δείγματα  $\theta_{0:t}^{-(i)}$  έναν Markov πυρήνα μετάβασης  $K(d\theta_{0:t}/\theta_{0:t}^{-(i)})$  με κατανομή  $p(d\theta_{0:t}/y_{1:t})$  ώστε να δημιουργήσουμε ξανά  $N$  δείγματα  $\{\theta_{0:t}^{(i)}\}, i=1:N$ .

Το πρόβλημα που μπορεί να δημιουργηθεί με το selection step είναι ότι αν η κατανομή των importance βαρών είναι σχετικά ασύμμετρη και με πολλές αυξομειώσεις, τότε τα δείγματα που έχουν μεγάλα βάρη επίλεγονται αρκετές φορές. Έτσι πολλά δείγματα  $\theta_{0:t}^{-(i)}$  είναι όμοια και παρατηρείται μια έλλειψη δειγμάτων. Ο σκοπός αυτού εδώ του βήματος θα είναι να επαναπαράξουμε τα δείγματα  $\theta_{0:t}$ .

Η λογική επάνω στην οποία στηρίζεται αυτό το βήμα είναι η ακόλουθη: Τα δείγματα  $\theta_{0:t}$  είναι κατανεμημένα σύμφωνα με την  $p(d\theta_{0:t}/y_{1:t})$ . Εάν εφαρμόσουμε στα δείγματα  $\theta_{0:t}^{-(i)}$  έναν Markov πυρήνα μετάβασης  $K(d\theta_{0:t}/\theta_{0:t}^{-(i)})$  με κατανομή  $p(d\theta_{0:t}/y_{1:t})$  τα νέα δείγματα  $\theta_{0:t}$  που θα παραχθούν θα είναι και πάλι κατανεμημένα σύμφωνα με την  $p(d\theta_{0:t}/y_{1:t})$ . Η δυσκολία σε αυτό το βήμα έγκειται στην κατάλληλη επιλογή του MCMC πυρήνα. Αν και στην βιβλιογραφία έχουν προταθεί αρκετοί τρόποι υλοποίησης του MCMC πυρήνα, η επιλογή μας για την κατασκευή του στηρίχθηκε στο *backward information filter*.

Αφού υλοποιήσουμε τον πυρήνα θέτουμε  $\theta_{0:t-L}^{(i)} = \theta_{0:t-L}^{(i)}$  και για  $\kappa = t-L+1$  μέχρι  $\kappa = t$  θα δειγματολογήσουμε τα  $\theta_{\kappa}^{(i)}$  χρησιμοποιώντας τον αλγόριθμο *Metropolis–Hastings* με συνάρτηση κατανομής  $p(d\theta_{\kappa}/y_{1:t}, \theta_{-\kappa}^{(i)})$  όπου

$$p(d\theta_{\kappa}/y_{1:t}, \theta_{-\kappa}^{(i)}) = p(\theta_{\kappa}/y_{1:t}, \theta_{-\kappa}^{(i)})d\theta_{\kappa}$$

και

$$\theta_{-\kappa}^{(i)} = (\theta_{0:t-L}^{(i)}, \theta_{t-L+1}^{(i)}, \dots, \theta_{k-1}^{(i)}, \theta_{k+1}^{(i)}, \theta_t^{(i)}).$$

Ο λόγος που θέτουμε  $\theta_{0:t-L}^{(i)} = \theta_{0:t-L}^{(i)}$  και δεν παίρνουμε δείγματα απο την χρονική στιγμή  $t=0$  είναι ότι ο *Metropolis–Hastings* αλγόριθμος απαιτεί μια χρονική περίοδο  $L$  προκειμένου να ομαλοποιηθεί η διαδικασία (*burn-in period*) και να προκύψουν τα επιθυμητά αποτελέσματα.

Στο παράρτημα (A) θα παραθέσουμε τον *Metropolis–Hastings* αλγόριθμο, το *backward information filter* καθώς και τον τρόπο με τον οποίο υπολογίσαμε την συνάρτηση κατανομής του *Metropolis–Hastings* αλγορίθμου.

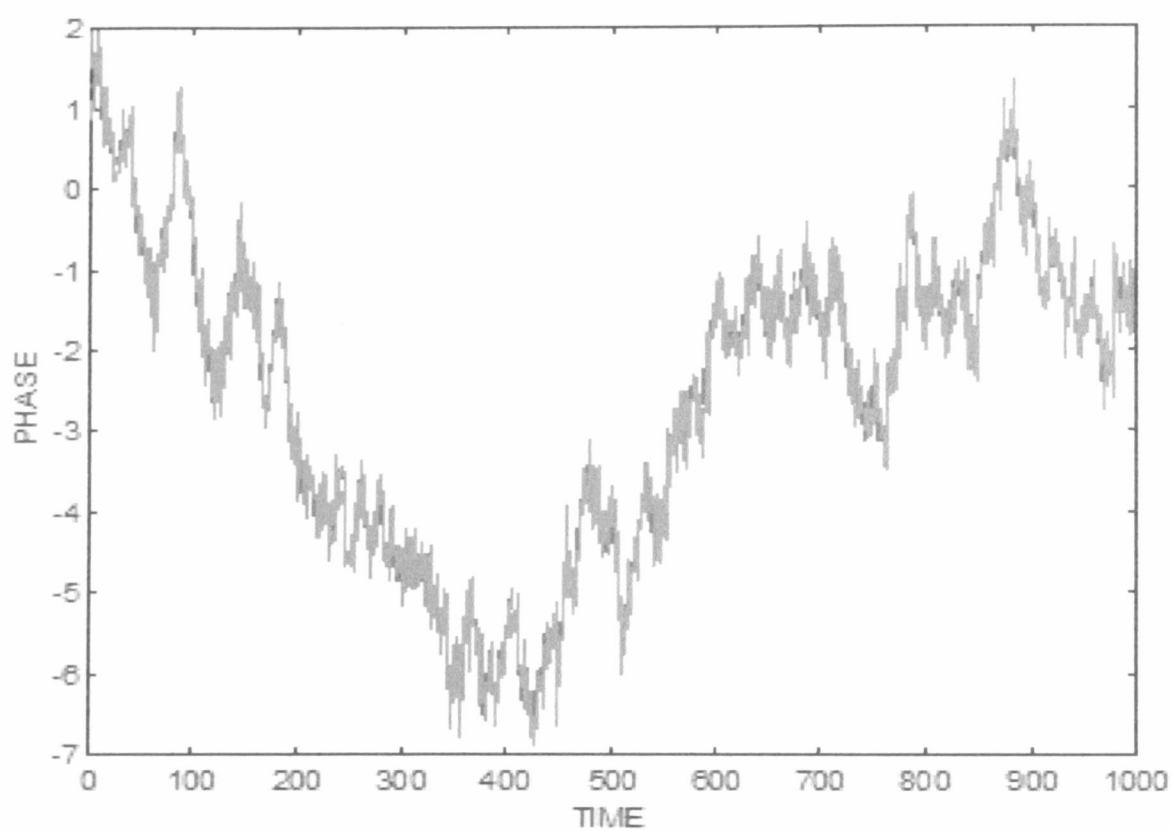
## Κεφάλαιο 4

### Παρουσίαση των αποτελεσμάτων

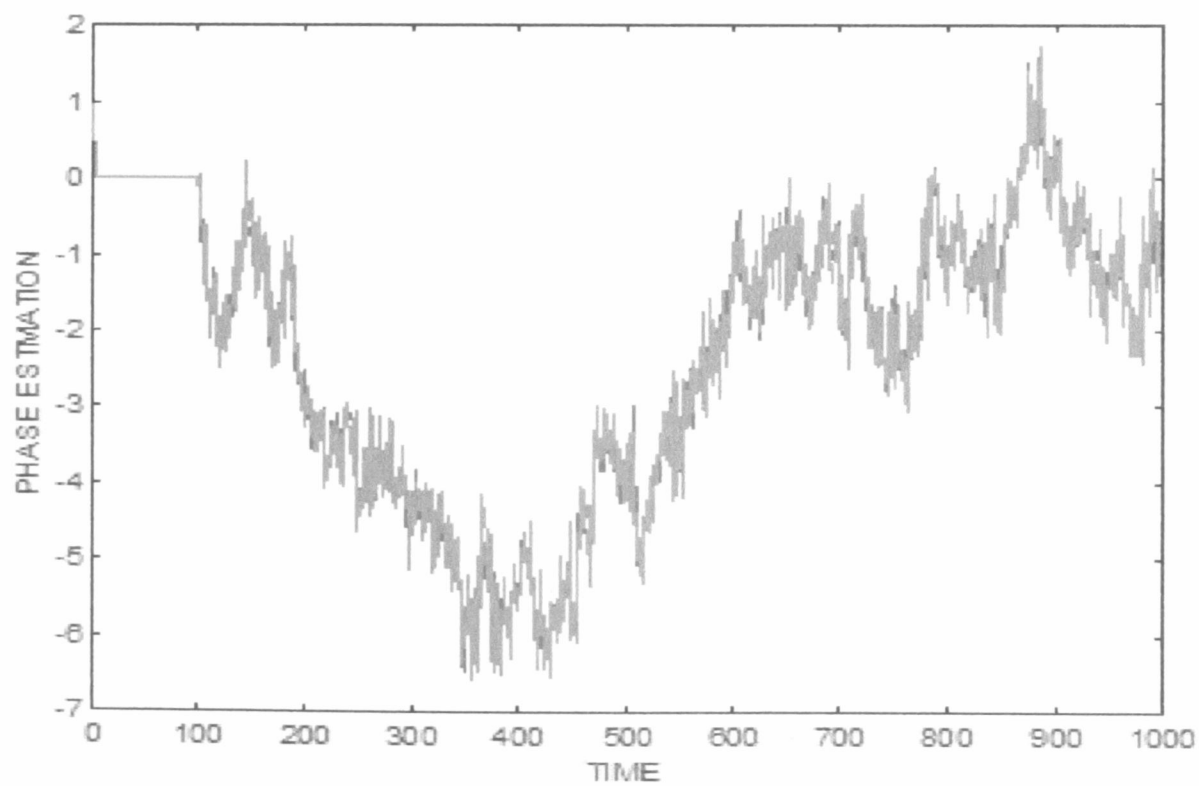
Στο κεφάλαιο αυτό πρόκειται να παρουσιάσουμε τα αποτελέσματα σε γραφική μορφή που πήραμε από την προσομοίωση του Rao–Blackwellized filter(RB). Ο κώδικας που χρησιμοποιήθηκε για την προσομοίωση καθώς και μια σύντομη αναφορά πάνω σ' αυτόν παρατίθενται στο παράρτημα (B). Ακόμη στο παρόν κεφάλαιο θα παρατεθεί και η σύγκριση μεταξύ του RB φίλτρου που υλοποιήσαμε και του Extended Kalman Filter (EKF).

#### 4.1 Γραφικές παραστάσεις

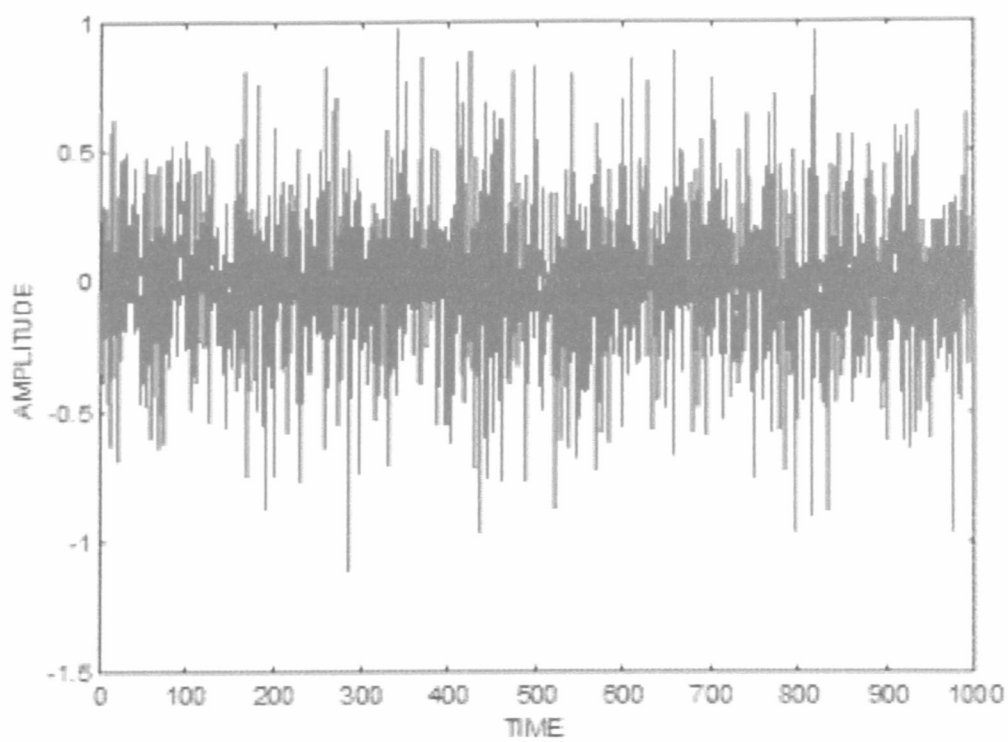
Οι γραφικές παραστάσεις που θα παρουσιαστούν ελήφθησαν με παραμέτρους  $t=1:T$  όπου  $T=1000$  ο χρόνος προσομείωσης,  $L = 900$  με  $T-L$  την χρονική στιγμή που ξεκινάμε να λαμβάνουμε αποτελέσματα από τον Metropolis-Hasting αλγόριθμο,  $N=3$  οι διαστάσεις των διανυσμάτων πλάτους και φάσης,  $\sigma_w = 0.3$  και  $\theta_t/\theta_{t-1} \sim N(\theta_{t-1}, 0.2)$



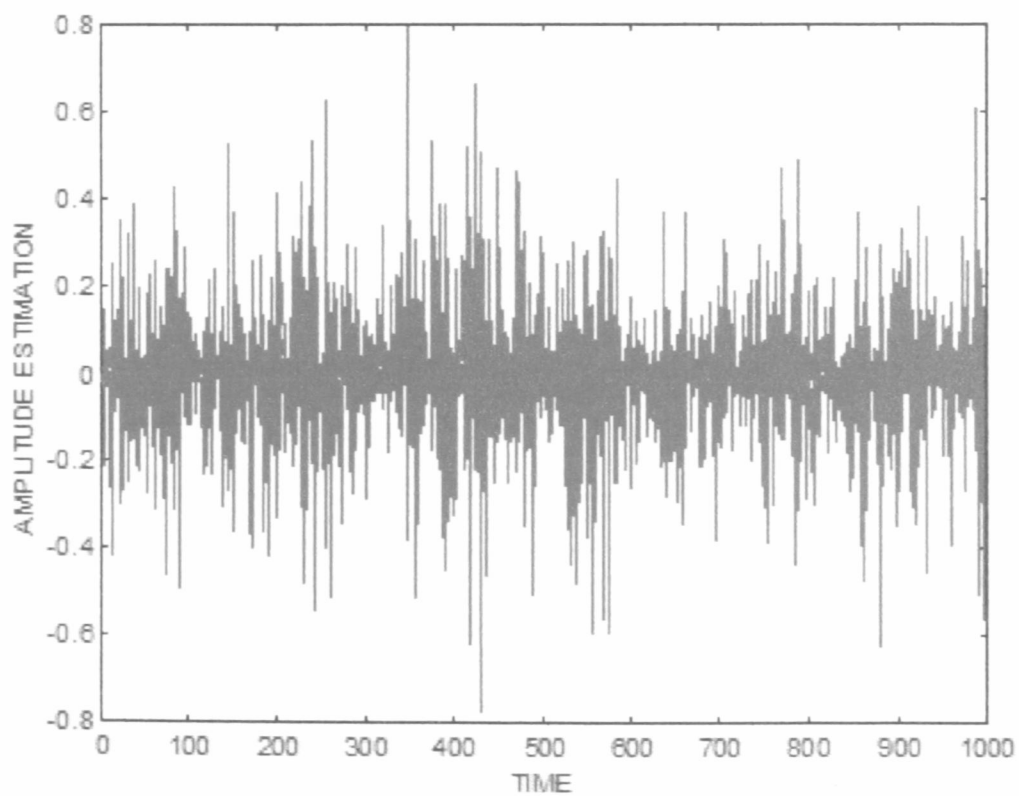
• Σχήμα 3 Γραφική παράσταση των αρχικών φάσεων



• Σχήμα 4 Γραφική παράσταση των εκτιμήσεων των φάσεων



- Σχήμα 5 Γραφική παράσταση των αρχικών τιμών πλάτους



- Σχήμα 6 Γραφική παράσταση των εκτιμήσεων των τιμών πλάτους

Όπως βλέπουμε οι κυματομορφές των εκτιμήσεων παρουσιάζουν μεγάλη ομοιότητα με αυτές των πραγματικών τιμών αφού ακολουθούν προσεγγιστικά την ίδια κατανομή.

#### 4.2 Σύγκριση απόδοσης RB filter και EKF

Η σύγκριση της απόδοσης των δύο μεθόδων έγινε ως εξής : Προσομοιώσαμε τις δυο μεθόδους  $P = 50$  φορές με τις εξής παραμέτρους :

$T = 200, N = 1, A = 1, B = 0.2, \hat{a}_0 = 0, P_0 = 1, \sigma_w = D = 0.3$  και  $\theta_t/\theta_{t-1} \sim N(\theta_{t-1}, 0.2)$ .

Η σύγκριση των αποδόσεων έγινε με βάση την μέση τιμή και την διασπορά του μέσου τετραγωνικού λάθους (MSE) των φάσεων  $SE^P$ , όπου

$$m(SE^P) = (1/P) \sum_{p=1:P} SE^P$$

και

$$\sigma^2(SE^P) = (1/P) \sum_{p=1:P} [SE^P - m(SE^P)]^2$$

Έτσι για το EKF είχαμε  $m(SE^P_{(\theta)}) = 10^3, m(SE^P_{(\alpha)}) = 10^3$

και  $\sigma^2(SE^P_{(\theta)}) = 2.6 \times 10^6, \sigma^2(SE^P_{(\alpha)}) = 2.7 \times 10^6$

ενώ για το RB είχαμε  $m(SE^P_{(\theta)}) = 6.2, m(SE^P_{(\alpha)}) = 6.3$

και  $\sigma^2(SE^P_{(\theta)}) = 28.5, \sigma^2(SE^P_{(\alpha)}) = 38.2$

Η πολύ μεγάλη διαφορά οφείλεται στις συχνές αποκλίσεις του EKF. Και σε περιπτώσεις που το EKF δεν απέκλινε, η διαφορά ήταν και πάλι αρκετά μεγάλη, γεγονός που δείχνει την αποτελεσματικότητα του RAO – BLACKWELLIZED particle filtering αλγορίθμου. Ο κώδικας για το MSE παρατίθεται στο παράρτημα (B).



## Κεφάλαιο 5

### Συμπεράσματα και μελλοντικές επεκτάσεις

Στην εργασία αυτή ασχοληθήκαμε με ένα εξαιρετικά ενδιαφέρον πρόβλημα τον υπολογισμό σε πραγματικό χρόνο της φάσης και του πλάτους χρονικά μεταβαλλόμενων σημάτων. Αν και το πρόβλημα αυτό συναντάται σε ποικίλες εφαρμογές, όπως στην επεξεργασία φωνής ή σε συστήματα ραντάρ, εμείς εστιάσαμε το πρόβλημα όπως αυτό συναντάται στις τηλεπικοινωνίες και συγκεκριμένα, στην αποδιαμόρφωση σήματος.

Έτσι, αντιμετωπίζοντας τον υπολογισμό σε πραγματικό χρόνο της φάσης και του πλάτους χρονικά μεταβαλλόμενων σημάτων ως ένα σύνθετο πρόβλημα βέλτιστου φιλτραρίσματος και λαμβάνοντας υπόψιν τις ιδιαιτερότητες και τα χαρακτηριστικά του μοντέλου που περιγράψαμε στο κεφάλαιο 2 (μη γραμμικός, μη Gaussian χώρος καταστάσεων, περιβάλλον Gaussian θορύβου), υλοποιήσαμε μια SMC μέθοδο για τον σκοπό αυτό.

Η συγκεκριμένη μέθοδος που υλοποιήσαμε ονομάζεται RAO-BLACKWELLIZED particle filtering και λειτουργεί χρησιμοποιώντας διαδοχικά ένα συνδυασμό sequential importance sampling, selection resampling και Markov Chain Monte Carlo (MCMC) βημάτων.

Για την υλοποίηση του συστήματος προτιμήθηκε το εργαλείο προγραμματισμού Matlab διότι αφενός μεν υπερτερεί των κλασικών γλωσσών προγραμματισμού (C/C++) όσον αφορά τον χειρισμό πινάκων, αφετέρου δε παρέχει πολύ ικανοποιητικές συναρτήσεις ως προς την εύρεση ιδιοτιμών και ιδιοανυσμάτων ενός πίνακα (π.χ., eig και svd). Τέλος, το Matlab παρέχει εύκολα την δυνατότητα αναλυτικής γραφικής απεικόνισης, ιδιότητα που μας δίνει την δυνατότητα αποτελεσματικής και ξεκάθαρης επίδειξης των αποτελεσμάτων μας.



### 5.1 Συμπεράσματα

Ως συμπεράσματα από την εφαρμογή του Rao-Blackwellized particle filtering και την μελέτη των αποτελεσμάτων του μπορούμε να παραθέσουμε τα εξής:

- Η απόδοση της συγκεκριμένης μεθόδου είναι πολύ καλή αφού είναι εμφανώς ανώτερη της απόδοσης του EKF (η πιο ευρέως χρησιμοποιούμενη μέθοδος σε τέτοιου είδους προβλήματα).
- Η μέθοδος αποδίδει πολύ καλά και σε μοντέλα μεγαλύτερων διαστάσεων.
- Η μέθοδος είναι σχετικά απλή ως προς την υλοποίηση της και έχει μικρές απαιτήσεις σε υπολογιστικούς πόρους. Ως εκ τούτου μπορεί κάλλιστα να χρησιμοποιηθεί στην παραγωγή.
- Το μειονέκτημα της συγκεκριμένης μεθόδου πηγάζει από το ότι αποτελεί μια στατιστική μέθοδο. Έτσι ενώ έχουμε διόρθωση των σφαλμάτων που οφείλονται στο θόρυβο η μέθοδος εισαγάγει τα δικά της λάθη. Τα λάθη αυτά θεωρούμε ότι οφείλονται στο γεγονός ότι όταν η importance κατανομή δεν προσεγγίζει αρκετά καλά την πραγματική κατανομή του αρχικού σήματος η απόδοση που έχουμε είναι μη ικανοποιητική. Να σημειωθεί εδώ ότι η κύρια δυσκολία της μεθόδου που περιγράψαμε είναι ακριβώς αυτή, δηλαδή η εύρεση μίας καλής προσέγγισης της κατανομής του αρχικού σήματος.

## 5.2 Μελλοντικές επεκτάσεις

Ως αντικείμενο μελλοντικών επεκτάσεων, έχουμε να προτείνουμε τα εξής:

- Την εφαρμογή και ανάλυση της SMC μεθόδου σε διαφορετικό περιβάλλον θορύβου.
- Την εφαρμογή και ανάλυση της SMC μεθόδου σε μοντέλο που αποτελείται από δίαυλο επικοινωνίας που προκαλεί εξασθένιση στο αρχικό σήμα.
- Την χρήση του Gibbs αλγορίθμου αντί του Metropolis-Hasting που χρησιμοποιήθηκε στην δειγματοληψία του MCMC βήματος
- Την υλοποίηση της μεθόδου σε πραγματικό σύστημα ούτως ώστε να δοκιμασθεί η απόδοση της κάτω από πραγματικές συνθήκες

## Παράρτημα (Α)

Στο παρόν παράρτημα θα γίνει η παρουσίαση των Kalman και Backward Information φίλτρων που χρησιμοποιήθηκαν, καθώς και της backward forward procedure που χρησιμοποιήθηκε για τον καθορισμό της συνάρτησης  $p(d\theta_k/y_{1:t}, \theta^{(i)}_{-k})$  του Metropolis- Hastings αλγορίθμου. Το  $C(\theta_{0:t})$  θεωρείται γνωστό. Αρχικοποιούμε τα  $a_{0/0} = \hat{a}_0$  και  $P_{0/0} = P_0$ .

### A.1 Αλγόριθμος Metropolis-Hasting

Ο αλγόριθμος Metropolis-Hasting ανήκει στην κατηγορία των Monte Carlo μεθόδων. Έτσι όπως οι περισσότερες από αυτές τις μεθόδους (π.χ., importance sampling, rejection sampling) προκειμένου να προσεγγίσει μια κατανομή  $f'$  (target density) χρησιμοποιεί μια βοηθητική κατανομή  $g$  (proposal).

Η διαφοροποίηση του σε σχέση με τις προαναφερόμενες μεθόδους είναι ότι η βοηθητική κατανομή που χρησιμοποιεί εξαρτάται από την κατάσταση  $x(t)$  που βρίσκεται το σύστημα σε κάθε χρονική στιγμή. Έτσι σε κάθε χρονική στιγμή  $t$  ο αλγόριθμος εκτελείται ως εξής:

- Παράγουμε μία νέα κατάσταση  $x'$  από την proposal  $g(x'/x(t))$
- Υπολογίζουμε την ποσότητα
$$\alpha = f'(x')g(x(t)/x') [f'(x(t)) g(x'/x(t))]^{-1}.$$
- Αν  $\alpha \geq 1$  η  $x'$  γίνεται δεκτή και θέτουμε  $x(t+1)=x'$ .
- Αν  $\alpha < 1$  η  $x'$  απορρίπτεται με πιθανότητα  $\alpha$  και θέτουμε  $x(t+1) = x(t)$ .

Η συγκεκριμένη μέθοδος προτιμήθηκε γιατί είναι προσαρμοσμένη στην on-line λογική του SMC αλγορίθμου που υλοποιήσαμε. Επίσης ο Metropolis-Hasting αλγόριθμος επιλέχθηκε διότι, σε αντίθεση με τις κλασικές μεθόδους Monte Carlo, ενδείκνυται για εφαρμογές πολυμεταβλητών συστημάτων.

## A.2 Kalman filter

Πρωτού δείξουμε την λειτουργία του *Kalman* φίλτρου θα παρουσιάσουμε τι αναπαριστούν οι μεταβλητές που έχουμε σε αυτό. Έτσι:

$$a_{t/t-1}(\theta_{0:t}) = E\{a_t/y_{1:t-1}(\theta_{0:t})\}$$

$$a_{t/t-1}(\theta_{0:t}) = E\{a_t/y_{1:t}(\theta_{0:t})\}$$

$$y_{t/t-1}(\theta_{0:t}) = E\{y_t/y_{1:t}(\theta_{0:t})\}$$

$$\tilde{y}_{t/t-1}(\theta_{0:t}) = y_t - y_{t/t-1}(\theta_{0:t})$$

$$P_{t/t-1}(\theta_{0:t}) = \text{cov}\{a_t/y_{1:t-1}(\theta_{0:t})\}$$

$$P_{t/t}(\theta_{0:t}) = \text{cov}\{a_t/y_{1:t}(\theta_{0:t})\}$$

$$S_t(\theta_{0:t}) = \text{cov}\{y_t/y_{1:t}(\theta_{0:t})\}$$

Για χρόνο  $t=1:T$  οι εξισώσεις που διέπουν το φίλτρο είναι:

- $a_{t/t-1}(\theta_{0:t}) = A a_{t/t-1}(\theta_{1:t-1})$
- $P_{t/t-1}(\theta_{0:t}) = A P_{t/t-1}(\theta_{1:t-1})A^T + BB^T$
- $y_{t/t-1}(\theta_{0:t}) = C(\theta_t) a_{t/t-1}(\theta_{0:t})$
- $S_t(\theta_{0:t}) = C(\theta_t)P_{t/t-1}(\theta_{0:t})C^T(\theta_t) + DD^T$
- $a_{t/t}(\theta_{0:t}) = a_{t/t-1}(\theta_{0:t}) + P_{t/t-1}(\theta_{0:t}) C^T(\theta_t) [S_t(\theta_{0:t})]^{-1} \tilde{y}_{t/t-1}(\theta_{0:t})$
- $P_{t/t}(\theta_{0:t}) = P_{t/t-1}(\theta_{0:t}) - P_{t/t-1}(\theta_{0:t}) C^T(\theta_t) [S_t(\theta_{0:t})]^{-1} C(\theta_t) P_{t/t-1}(\theta_{0:t})$

Τέλος σε κάθε χρονική στιγμή η Gaussian κατανομή πρόβλεψης είναι :

$$p(y_t/y_{1:t-1}(\theta_{0:t})) = N(\tilde{y}_{t/t-1}(\theta_{0:t}), S_t(\theta_{0:t})).$$

### A.3 Backward Information Filter

Το Backward Information Filter είναι το φίλτρο στο οποίο στηριχθήκαμε προκειμένου να υλοποιήσουμε τον MCMC πυρήνα. Το Backward Information Filter προχωράει ανάποδα στο χρόνο υπολογίζοντας χρήσιμες ποσότητες όπως τα

$$\begin{aligned} [P'_{t|t}(\theta_t)]^{-1} &= C^T(\theta_t) (DD^T)^{-1} C(\theta_t) \text{ και} \\ [P'_{t|t}(\theta_t)]^{-1} a'_{t|t}(\theta_t) &= C^T(\theta_t) (DD^T)^{-1} y_t \end{aligned}$$

Έτσι για  $k = t, \dots, t-L+1$  και  $L \geq 0$  έχουμε:

- $\Delta_{k+1}(\theta_{k+1:t}) = [I_{nu} + B^T [P'_{k+1/k+1}(\theta_{k+1:t})]^{-1} B]^{-1}$
- $[P'_{k/k+1}(\theta_{k+1:t})]^{-1} = A^T [P'_{k+1/k+1}(\theta_{k+1:t})]^{-1} \times (I_{na} - B \Delta_{k+1}(\theta_{k+1:t}) B^T [P'_{k+1/k+1}(\theta_{k+1:t})]^{-1}) A$
- $[P'_{k/k+1}(\theta_{k+1:t})]^{-1} a'_{k/k+1}(\theta_{k+1:t}) = A^T (I_{na} - [P'_{k+1/k+1}(\theta_{k+1:t})]^{-1} B \Delta_{k+1}(\theta_{k+1:t}) B^T) \times [P'_{k+1/k+1}(\theta_{k+1:t})]^{-1} a'_{k+1/k+1}(\theta_{k+1:t})$
- $[P'_{k/k}(\theta_{k:t})]^{-1} = [P'_{k/k+1}(\theta_{k+1:t})]^{-1} + C^T(\theta_k) (DD^T)^{-1} C(\theta_k)$
- $[P'_{k/k}(\theta_{k:t})]^{-1} a'_{k/k}(\theta_{k:t}) = [P'_{k/k+1}(\theta_{k+1:t})]^{-1} a'_{k/k+1}(\theta_{k+1:t}) + C^T(\theta_k) (DD^T)^{-1} y_k$

#### A.4 Συνάρτηση δειγματοληψίας Metropolis-Hasting

Οι ποσότητες  $[P'_{k/k+1}(\theta'_{k+1:t})]^{-1} a'_{k/k+1}(\theta'_{k+1:t})$  και  $[P'_{k/k+1}(\theta'_{k+1:t})]^{-1}$  δίνονται από το Backward Information Filter και τα  $y_{t-1}(\theta_{0:t})$ ,  $S_t(\theta_{0:t})$ ,  $P_{k/k}(\theta_{0:k})$  δίνονται από το Kalman Filter.

Έτσι για  $k = t-L+1$  μέχρι  $k = t$  έχουμε :

$$\begin{aligned} p(d\theta_k/y_{1:t}, \theta_{-k}) &= p(\theta_k/\theta_{k-1})p(\theta_{k+1}/\theta_k)N(y_{t-1}(\theta_{0:t}), S_t(\theta_{0:t})) \\ &\times \left| \Pi_{k/k}(\theta_{0:k})Q_{k/k}^T(\theta_{0:k})[P'_{k/k+1}(\theta'_{k+1:t})]^{-1}Q_{k/k}(\theta_{0:k}) + I_{nk} \right|^{-1/2} \\ &\times \left( \exp\{(-1/2) [a_{k/k}^T(\theta_{0:k}) [P'_{k/k+1}(\theta'_{k+1:t})]^{-1} a_{k/k}(\theta_{0:k}) - \right. \\ &- 2a_{k/k}^T(\theta_{0:k}) [P'_{k/k+1}(\theta'_{k+1:t})]^{-1} a'_{k/k+1}(\theta'_{k+1:t}) - \\ &- (a'_{k/k+1}(\theta'_{k+1:t}) - a_{k/k}(\theta_{0:k}))^T [P'_{k/k+1}(\theta'_{k+1:t})]^{-1} \\ &\left. \times R_{k/k}(\theta_{0:t})[P'_{k/k+1}(\theta'_{k+1:t})]^{-1} (a'_{k/k+1}(\theta'_{k+1:t}) - a_{k/k}(\theta_{0:k}))\} \right) \end{aligned}$$

Ο πίνακας  $Q_{k/k}(\theta_{0:k})$  είναι ο πίνακας ιδιοανυσμάτων του  $P_{k/k}(\theta_{0:k})$  και ο  $\Pi_{k/k}(\theta_{0:k})$  είναι ένας  $n_k \times n_k$  διαγώνιος πίνακας πού περιέχει τις μη μηδενικές ιδιοτιμές του  $P_{k/k}(\theta_{0:k})$ . Δηλαδή

$$P_{k/k}(\theta_{0:k}) = Q_{k/k}(\theta_{0:k}) \Pi_{k/k}(\theta_{0:k}) Q_{k/k}^T(\theta_{0:k})$$

Τέλος ο  $R_{k/k}(\theta_{0:t})$  ισούται με

$$R_{k/k}(\theta_{0:t}) = Q_{k/k}(\theta_{0:k}) \left( [\Pi_{k/k}(\theta_{0:k})]^{-1} + Q_{k/k}^T(\theta_{0:k}) [P'_{k/k+1}(\theta'_{k+1:t})]^{-1} Q_{k/k}(\theta_{0:k}) \right)^{-1} Q_{k/k}^T(\theta_{0:k})$$

## Παράρτημα (B)

### Ο Κώδικας

#### *B.1 Σχέδιο κώδικα*

Για  $t = 1 : T$

- Υλοποίηση των παραμέτρων του συστήματος
- Υλοποίηση του αρχικού σήματος (φάσεις – πλάτη)
- Υπολογισμός των παρατηρήσεων

Για  $t = 1 : T$

- Importance step
  - Παραγωγή των δειγμάτων
  - Kalman για τον υπολογισμό παραμέτρων για τα βάρη
  - Υπολογισμός βαρών
  - Κανονικοποίηση των βαρών
- Selection step
  - Απομάκρυνση των δειγμάτων με μικρά βάρη
  - Πολλαπλασιασμός των δειγμάτων με μεγάλα βάρη

Αν  $t > L$

- MCMC step
  - Kalman για υπολογισμό των παραμέτρων για την συνάρτηση δειγματοληψίας
  - Υπολογισμός της συνάρτησης δειγματοληψίας
  - Εφαρμογή του Metropolis-Hasting αλγορίθμου
  - Παραγωγή των εκτιμήσεων για τις τιμές της φάσης
  - Kalman για τον υπολογισμό των εκτιμήσεων για τις τιμές των πλατών.

Αν  $t = T$

- Υπολογισμός του μέσου τετραγωνικού λάθους



## B.2 *Rao-Blackwellized Particle Filtering*

```
N = input('N:');

T = input('T:');

L = input('L:');

ao=0;
Po=1;
state=0;
A =0.4*randn(N);
B =0.4*randn(N,1);
D= 0.3;

thita=zeros(T,N);
sthita=zeros(T,N);
resthita = zeros(T,N);
rthita = zeros(T,N+1);
mthita = zeros(T,N);

pro=zeros(T,N);
prom =zeros(T,N);
a=zeros (N,T);
am = zeros(N,T);
ap =zeros(N,T);
Pm = zeros(N,N,T);
Pp = zeros (N,N,T);

y_ob = zeros(1,T);
y = zeros(T,1);
S = zeros(T,1);
w = zeros(T,N);
maxi = zeros(T,1);

p = zeros(N,N,T);
q = zeros(N,N,T);

thita(1,:)= pi* rand(1,N);
a(:,1)=random('Normal',ao,Po,N,1);
Pm(:, :,1)=Po;
```

```

Pp(:,1)=1;
am(:,1)=ao;
ap(:,1)=1;

```

```

randn('state',stat);

```

```

for t=2:T
    for i=1:N
        thita(t,i)= random('Normal', thita(t-1),0.2,1,1) ;
    end
end

```

```

for t=1:T
    for i=1:N
        C(t,i) = cos ( thita(t,i) );
    end
end

```

```

for t =2:T
    a(:,t)= A*a(:,t-1) + B *randn(1,1);
end

```

```

for t =1:T
    y_ob(t)=C(t,:)*a(:,t) + D *randn(1,1);
end

```

```

Ina=eye(N);
gb=zeros(N,T);
qb=zeros(N,T);
zb=zeros(N,N,T);
vb=zeros (N,N,T);
zb(:, :, T)=C(T,:)' *inv(D*D')*C(T,:);
gb(:, T)= C(T,:)' *inv(D*D') *y_ob(T);

```

```

i = T-1;

```

```

while i >= T-L+1

DE(i+1)= inv( 1+ B'* zb(:,i+1)*B );

vb(:,i)= A'*zb(:,i+1)* ( Ina- B*DE(i+1)*B'* zb(:,i+1) ) * A;

qb(:,i)= A'* (Ina - zb(:,i+1) *B *DE(i+1) *B') *gb(:,i+1);

zb(:,i) = vb(:,i) + C(i,:)' *inv(D*D')*C(i,:);

gb(:,i) = qb(:,i) + C(i,:)' *inv(D*D') *y_ob(i);

i=i-1;

end

sthita(1,:)= pi* rand(1,N);
C(1,:)=cos(sthita(1,:));
y(1,:)= C(1,:)*ap(:,1);
S(1,:)=C(1,:)*Pp(:,1) * C(1,:)' + D * D';

for i=1:N
    wth(1,i) = random('Normal',(y_ob(1)-y(1)),S(1),1,1) ;
end

sumwth(1)=0;
for i= 1:N
    sumwth(1) = sumwth(1) + wth(1,i);
end

for i= 1:N
    w(1,i) = wth(1,i)/sumwth(1) ;
end

maxi(1) = -100;
for i=1:N
    if w(1,i) > maxi(1)
        maxi(1) = w(1,i);
    end
end

for i=1:N
    if w(1,i) < 0.1*maxi(1)
        w(1,i)=0 ;
    end
end

```

```

    end
end

```

```

for i=1:N
    Nwt(1,i)=w(1,i)/maxi(1);
end

```

```

sumwt=0;
for i=1:N
    sumwt=sumwt+Nwt(1,i);
end

```

```

x(1)=N/sumwt;
for i=1:N
    Nt(1,i)=round(Nwt(1,i)*x(1));
end

```

```

j=1;
for i=1:N
    if Nt(1,i)~=0
        for count=1:Nt(1,i)
            rthita(1,j+count-1)=sthita(1,i);
        end
        j=j+count;
    end
end
end

```

```

for i=1:N
    resthita(1,i)=rthita(1,i);
end

```

```

for i=1:N
    mthita(1,i)=resthita(1,i);
end

```

```

randn('state',stat);
stae=stat;

```

```

for t = 2:T

```

```

    randn('state',stae);

```

```

for i=1:N
    sthita(t,i)= random('Normal',sthita(t-1),0.2,1,1) ;
    C(t,i)= cos(sthita(t,i));
end
stae=randn('state');

```

```

ap(:,t)= A * am(:,t-1);

```

```

Pp(:,t) = A*Pm(:,t-1)* A' +B * B';

```

```

y(t,:)= C(t,:)*ap(:,t);

```

```

S(t,:)=C(t,:)*Pp(:,t) * C(t,:)' + D * D';

```

```

am(:,t) = ap(:,t) +Pp(:,t)* C(t,:)' * inv (S(t)) * (y_ob(t) - y(t));

```

```

Pm(:,t) = Pp(:,t) - Pp(:,t)* C(t,:)' *inv(S(t)) * C(t,:) *Pp(:,t);

```

```

sumwth(t)=0;
for i= 1:N
    wth(t,i) = random('Normal',(y_ob(t)-y(t)) ,S(t),1,1) ;
end

```

```

for i= 1:N
    sumwth(t) = sumwth(t) + wth(t,i);
end

```

```

for i= 1:N
    w(t,i) = wth(t,i)/sumwth(t) ;
end

```

```

maxi(t) = -100;
for i=1:N
    if w(t,i) > maxi(t)
        maxi(t) = w(t,i);
    end
end

```

```

for i=1:N
    if w(t,i) < 0.1*maxi(t)
        w(t,i)=0;
    end
end

```

```

    end
end

```

```

for i=1:N
    Nwt(t,i)=w(t,i)/maxi(t);
end

```

```

sumwt=0;
for i=1:N
    sumwt=sumwt+Nwt(t,i);
end
x(t)=N/sumwt;

```

```

for i=1:N
    Nt(t,i)=round(Nwt(t,i)*x(t));
end

```

```

j=1;
for i=1:N
    if Nt(t,i)~=0
        for count=1:Nt(t,i)
            rthita(t,j+count-1)=sthita(t,i);
        end
        j=j+count;
    end
end

```

```

for i=1:N
    if rthita(t,i)==0
        resthita(t,i)=resthita(t-1,i);
    else
        resthita(t,i)=rthita(t,i);
    end
end

```

```

for i=1:N
    C(t,i) = cos ( resthita(t,i) );
end

```

```

if t >= T-L+1

```

```
ap(:,t)= A * am(:,t-1);
```

```
Pp(:,t) = A*Pm(:,t-1)* A' +B * B';
```

```
y(t,:)= C(t,:)*ap(:,t);
```

```
S(t,:)=C(t,:)*Pp(:,t) * C(t,:)' + D * D';
```

```
am(:,t) = ap(:,t) +Pp(:,t)* C(t,:)' * inv (S(t)) * (y_ob(t) - y(t));
```

```
Pm(:,t) = Pp(:,t) - Pp(:,t)* C(t,:)' *inv(S(t)) * C(t,:) *Pp(:,t);
```

```
[o l]=eigs(Pm(:,t),(rank(Pm(:,t))));
```

```
for j=1:N
```

```
    for i=1:rank(Pm(:,t))
```

```
        q(j,i,t)=o(j,i);
```

```
    end
```

```
end
```

```
for j=1:rank(Pm(:,t))
```

```
    for i=1:rank(Pm(:,t))
```

```
        p(j,i,t) =l(j,i);
```

```
    end
```

```
end
```

```
R(:,t) = q(:,t) *inv(inv( p(:,t)) + q(:,t)'* vb(:,t) *q(:,t) ) * q(:,t)';
```

```
Ink = eye(size(p(:,t)));
```

```
aa(:,t)= inv(vb(:,t))*qb(:,t);
```

```
ddeett(t)= (1/(sqrt(det(p(:,t)*q(:,t)'*vb(:,t)*q(:,t) +Ink))));
```

```
for i=1:N
```

```

    pro(t,i)= random('Normal',resthita(t-1),0.2,1,1) *
    random('Normal',resthita(t),0.2,1,1) *random('Normal',(y_ob(t)-y(t)) ,S(t),1,1)*
    ddeett(t) * exp( (-0.5) * ( am(:,t)'* vb(:,t)*am(:,t) - 2*am(:,t)'*vb(:,t)*aa(:,t) - (
    aa(:,t) - am(:,t))'*vb(:,t) *R(:,t)*vb(:,t)*(aa(:,t)- am(:,t))));
    prom(t,i)= random('Normal',mthita(t-1),0.2,1,1) *
    random('Normal',mthita(t),0.2,1,1) *random('Normal',(y_ob(t)-y(t)) ,S(t),1,1)*
    ddeett(t) * exp( (-0.5) * ( am(:,t)'* vb(:,t)*am(:,t) - 2*am(:,t)'*vb(:,t)*aa(:,t) - (
    aa(:,t) - am(:,t))'*vb(:,t) * R(:,t)*vb(:,t) * (aa(:,t) - am(:,t))));

```

end

```

for i=1:N
    ro(t,i)=(1/(prom(t,i)*random('Normal',resthita(t-1),0.2,1,1)))
    *(pro(t,i)*random('Normal',mthita(t-1),0.2,1,1));
    if ro(t,i)>=1
        mthita(t,i)=random('Normal',mthita(t-1),0.2,1,1);
    else
        mthita(t,i)=resthita(t-1,i);
    end
end
end

```

end

```

for i=1:N
    C(t,i)=cos(mthita(t,i));
end

```

ap(:,t)= A \* am(:,t-1);

Pp(:,t) = A\*Pm(:,t-1)\* A' +B \* B';

y(t,:)= C(t,:)\*ap(:,t);

S(t,:)=C(t,:)\*Pp(:,t) \* C(t,:)' + D \* D';

am(:,t) = ap(:,t) +Pp(:,t)\* C(t,:)' \* inv (S(t)) \* (y\_ob(t) - y(t));

Pm(:,t) = Pp(:,t) - Pp(:,t)\* C(t,:)' \*inv(S(t)) \* C(t,:) \*Pp(:,t);



end

```
figure(1);  
plot(thita); title('thita');
```

```
figure(2);  
plot(mthita); title('mthita');
```

```
figure(3);  
for i=1:N  
plot(a(i,:)); title('a (N) ');  
hold on;  
end  
hold off;
```

```
figure(4);  
for i=1:N  
plot(am(i,:)); title('am (N)');  
hold on;  
end  
hold off;
```

### B.3 Μέσο τετραγωνικό σφάλμα

```
P = input('P:');

global iq;
global tq;
global mini;

iq = 1;
tq = 1;
msethita = 0;
ssethita = 0;
sethita = zeros(P,1);
msea = 0;
ssea = 0;
sea = zeros(P,1);

for pe = 1:P

    stat = round(1000*rand);
    teliko;

    mini = 10000 ;
    for t = 1:T
        for i = 1:N
            if mini > (thita(t,i)-mthita(t,i))^2
                mini = ((thita(t,i)-mthita(t,i))^2) ;
                iq = i;
                tq = t;
            end
        end
    end

    for t = 1:T
        sethita(pe) = sethita(pe) + ( thita(t,iq)- mthita(tq,iq) )^2;
    end

    sethita(pe) = (1/T)*sethita(pe);

    mini = 10000;
    for t = 1:T
        for i = 1:N
```

```

        if mini > (a(i,t)-am(i,t))^2
            mini = ((a(i,t)-am(i,t))^2) ;
            iq=i;
            tq=t;
        end
    end
end

for t=1:T
    sea(pe)=sea(pe) + ( a(iq,t)- am(iq,tq) )^2;
end

sea(pe)=(1/T)*sea(pe);

```

end

```

for pe=1:P
    msethita= msethita +sethita(pe);
end
msethita=(1/P)*msethita;

```

```

for pe=1:P
    ssethita= ssethita +(sethita(pe)-msethita)^2 ;
end
ssethita =(1/P)*ssethita;

```

```

for pe=1:P
    msea= msea +sea(pe);
end
msea=(1/P)*msea;

```

```

for pe=1:P
    ssea= ssea +(sea(pe)-msea)^2 ;
end
ssea =(1/P)*ssea;

```

```

msea
ssea
msethita
ssethita

```

## Βιβλιογραφία

- [1]. Christophe Andrieu and Arnaud Doucet : *Optimal estimation of Amplitude and Phase Modulated Signals*. Cambrige University, Department of Engineering, Technical Report 395, (2001).
- [2]. Dan Crisan and Arnaud Doucet : *A Survey of Convergence Results on Particle Filtering Methods for Practitioners*. IEEE Transactions on Signal Proccesing, Vol. 50, No. 3, (2002).
- [3]. Julian Besag : *Marcov Chain Monte Carlo for Statistical Inference*. Center for statistics and Social Sciences. Working Paper No 9 (2001).
- [4]. D.J.C.Mackay : *Introduction to Monte Carlo Methods*. Learning in Graphical Models, M.I.T. Press, Mass. USA (1999).
- [5]. Arnaud Doucet, N. de Freitas and N.J. Gordon: *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Series Statistics for Engineering and Information Science, New York (2001).
- [6]. A.M.Law ,W.D.Kelton : *Simulation Modelling & Analysis*. 2<sup>nd</sup> ed., McGraw-Hill, New York (1995).
- [7]. Simon Haykin: *Communication Systems*. 2<sup>nd</sup> ed., John Wiley & Sons, New York (1994).
- [8]. Athanasios Papoulis: *Probability, random variables and stochastic processes*. 3<sup>rd</sup> ed., McGraw-Hill, New York (1991).
- [9]. Andrian Biran and Moshe Breiner : *Matlab 5 for Engineers*. 2<sup>nd</sup> ed., Tziolas Publications, Thessaloniki, Greece (1999).

