ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

«ΤΑΥΤΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΕΣΩ ΦΙΛΤΡΟΥ KALMAN»

ΣΚΟΥΡΑΣ ΙΩΑΝΝΗΣ

Περιεχόμενα

Σχήματα

Κεφάλαιο

1

Φίλτρα Kalman

1.1 **Εισαγωγή**.

Το **φίλτρο Kalman [1,4]** είναι μια μαθηματική μέθοδος που ονομάστηκε από το Rudolf E. Kalman. Ο σκοπός του είναι να χρησιμοποιήσει τις μετρήσεις που παρατηρούνται κατά τη διάρκεια του χρόνου που περιέχει το θόρυβο (τυχαίες παραλλαγές) και άλλες ανακρίβειες, και των τιμών προϊόντων που τείνουν να είναι πιο κοντά στις αληθινές τιμές των μετρήσεων και τις σχετικές υπολογισμένες τιμές τους. Πρόκειται για έναν αναδρομικό αλγόριθμο επίλυσης του γραμμικού προβλήματος φιλτραρίσματος διακριτού χρόνου ο οποίο μέχρι και σήμερα αποτελεί αντικείμενο επιστημονικής έρευνας.

Το φίλτρο Kalman παράγει τις εκτιμήσεις των αληθινών τιμών των μετρήσεων και των σχετικών υπολογισμένων τιμών τους με την πρόβλεψη μιας αξίας, τον υπολογισμό της αβεβαιότητας της προβλεφθείσας αξίας, και τον υπολογισμό ενός σταθμισμένου μέσου όρου της προβλεφθείσας αξίας και της μετρημένης αξίας. Το περισσότερο βάρος δίνεται στην αξία με τη λιγότερη αβεβαιότητα. Οι εκτιμήσεις που παράγονται με τη μέθοδο τείνουν να είναι πιο στενές στις αληθινές τιμές από τις αρχικές μετρήσεις επειδή ο σταθμισμένος μέσος όρος έχει μια καλύτερη, κατ' εκτίμηση, αβεβαιότητα από καθεμία εκ των τιμών που πήγαν στο σταθμισμένο μέσο όρο.

Το φίλτρο Kalman είναι ένας επαναλαμβανόμενος εκτιμητής. Αυτό σημαίνει ότι μόνο η κατ' εκτίμηση κατάσταση από το προηγούμενο χρονικό βήμα και η τρέχουσα μέτρηση απαιτούνται για να υπολογίσουν την εκτίμηση για την επικρατούσα κατάσταση.

Σε αντίθεση με τις τεχνικές εκτίμησης batch, καμία ιστορία των παρατηρήσεων ή/και των εκτιμήσεων δεν απαιτείται. Σε αυτό που ακολουθεί, η σημείωση $\hat{\mathbf{x}}_{n|m}$ αντιπροσωπεύει την εκτίμηση $\mathbf{x}$ στο χρόνο n των παρατηρήσεων δεδομένων μέχρι, και συμπεριλαμβανομένου στο χρόνο m.

Η κατάσταση του φίλτρου αντιπροσωπεύεται από δύο μεταβλητές:

$\hat{\mathbf{x}}_{k|k}$, *η* εκτίμηση κατάστασης στο χρόνο *K* των παρατηρήσεων δεδομένων μέχρι και συμπεριλαμβανομένου στο χρόνο *K* .

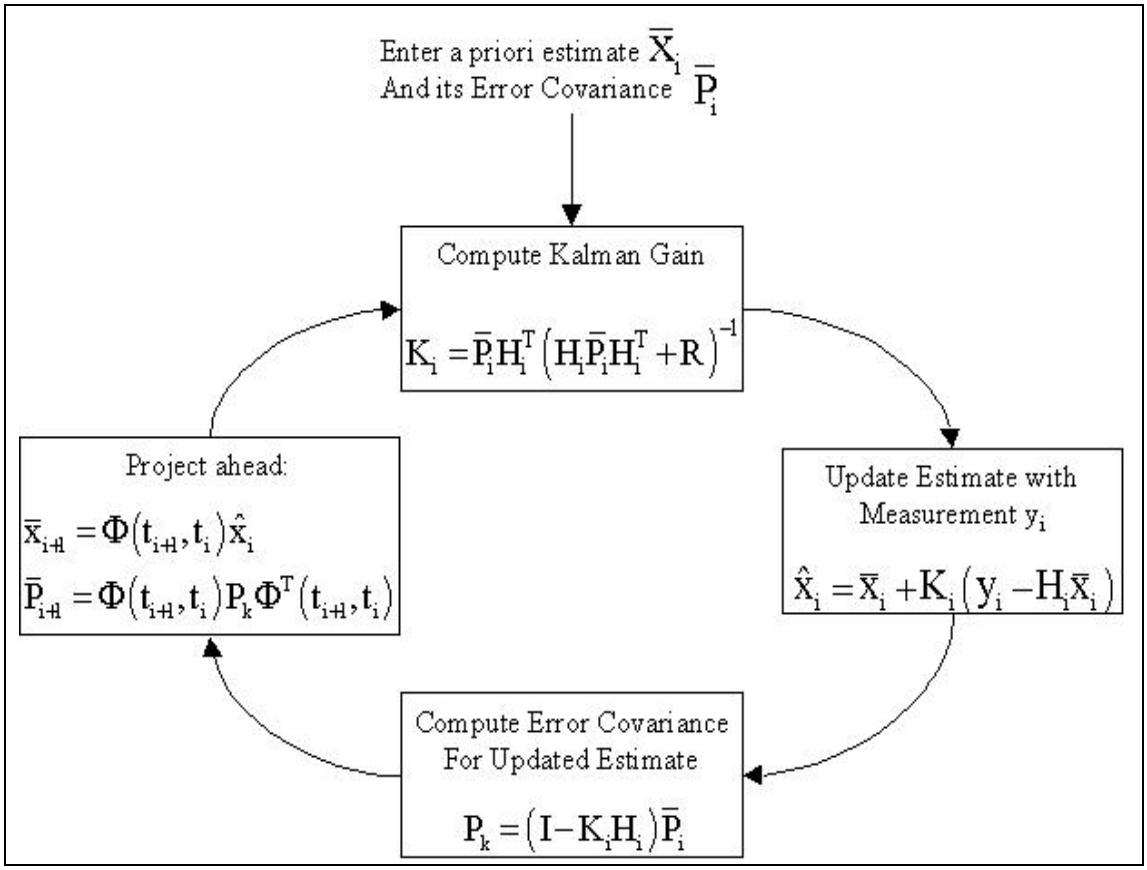
$\mathbf{P}_{k|k}$, η μήτρα συνδιακύμανσης λάθους (ένα μέτρο της κατ' εκτίμηση ακρίβειας της εκτιμώμενης κατάστασης).

Το φίλτρο Kalman είναι ένα αποδοτικό επαναλαμβανόμενο φίλτρο το οποίο εκτιμά την εσωτερική κατάσταση ενός γραμμικού δυναμικού συστήματος από μια σειρά θορυβώδους μετρήσεων.

Χρησιμοποιείται σε ένα ευρύ φάσμα της εφαρμοσμένης μηχανικής και οικονομετρικών εφαρμογών από το ραντάρ και την όραση υπολογιστών στην εκτίμηση των δομικών μακροοικονομικών προτύπων, και είναι ένα σημαντικό θέμα στη θεωρία ελέγχου και τα συστήματα ελέγχου την εφαρμοσμένη μηχανική.

Το φίλτρο Kalman έχει πολλές εφαρμογές στην τεχνολογία, και είναι ένα ουσιαστικό μέρος της ανάπτυξης της διαστημικής και στρατιωτικής τεχνολογίας. Ίσως ο συνηθέστερα χρησιμοποιημένος τύπος πολύ απλού φίλτρου Kalman είναι ο βρόχος με κλείδωμα φάσης, ο οποίος είναι πανταχού παρών στα ραδιόφωνα FM και στον περισσότερο ηλεκτρονικό εξοπλισμό επικοινωνιών.

Επίσης οι επεκτάσεις και οι γενικεύσεις στη μέθοδο έχουν αναπτυχθεί. Το φίλτρο αυτό έχει χρησιμοποιηθεί σε ένα ευρύτατο φάσμα εφαρμογών που περιλαμβάνει αεροναυπηγική, έλεγχο χημικών διεργασιών, επεξεργασία εικόνας και πολλά άλλα, μεταξύ των οποίων και προβλήματα σαν αυτό που θα μελετήσουμε παρακάτω.



Σχήμα 1. Κύκλος υπολογισμού Kalman.

Το φίλτρο Kalman περιλαμβάνει μια συλλογή εξισώσεων που φαίνεται παρακάτω:

Predicted (a priori) state $\hat{\mathbf{x}}_{k\mid k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1\mid k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1}$
Predicted (a priori) estimate covariance $\mathbf{P}_{k\mid k-1} = \mathbf{F}_k\mathbf{P}_{k-1\mid k-1}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_{k-1}$
Innovation or measurement residual $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k\mid k-1}$
Innovation (or residual) covariance $\mathbf{S}_k = \mathbf{H}_k\mathbf{P}_{k\mid k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k$
Optimal Kalman gain $\mathbf{K}_k = \mathbf{P}_{k\mid k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1}$
Updated (a posteriori) state estimate $\hat{\mathbf{x}}_{k\mid k} = \hat{\mathbf{x}}_{k\mid k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$
Updated (a posteriori) estimate covariance $\mathbf{P}_{k\mid k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k\mid k-1}$

Φίλτρο Kalman – Bucy.

Το φίλτρο Kalman–Bucy είναι μια έκδοση του κλασσικού φίλτρου Kalman, που αφορά εκτίμηση σε συνεχή χρόνο. Βασίζεται στο μοντέλο χώρου κατάστασης. Οι βασικές εξισώσεις που το περιγράφουν είναι αυτές που φαίνονται παρακάτω.

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{w}(t)$$

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t)$$

Σε αυτές τις εξισώσεις, η συνδιασπορά των παραγόντων – θορύβων $\mathbf{w}(t)$ και $\mathbf{v}(t)$, δίνονται από τα μητρώα $\mathbf{Q}(t)$ και $\mathbf{R}(t)$, αντίστοιχα.

Το φίλτρο αποτελείται από δύο διαφορικές εξισώσεις. Η μία αφορά την εκτίμησση της κατάστασης και η άλλη για την συνδιασπορά.

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \mathbf{F}(t)\hat{\mathbf{x}}(t) + \mathbf{K}(t)(\mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t))$$

$$\frac{d}{dt}\mathbf{P}(t) = \mathbf{F}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^T(t) + \mathbf{Q}(t) - \mathbf{K}(t)\mathbf{R}(t)\mathbf{K}^T(t)$$

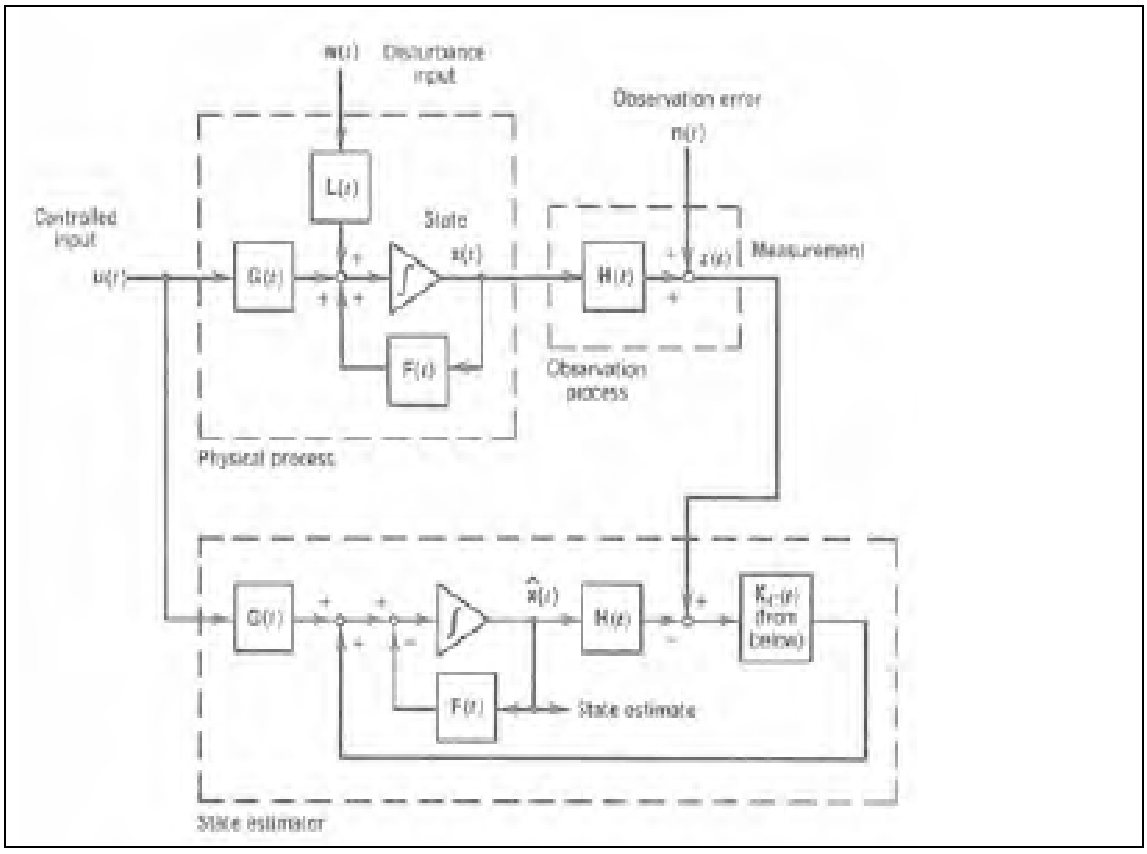
Το κέρδος του φίλτρου δίνεται από την εξίσωση:

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^T(t)\mathbf{R}^{-1}(t)$$

Σημειώνουμε εδώ, πως στην παραπάνω εξίσωση για το κέρδος $\mathbf{K}(t)$, η συνδιασπορά του θορύβου παρατήρησης $\mathbf{R}(t)$ αναπαριστά ταυτόχρονα την συνδιασπορά της πρόβλεψης, που αναπαριστάται από την παρακάτω εξίσωση.

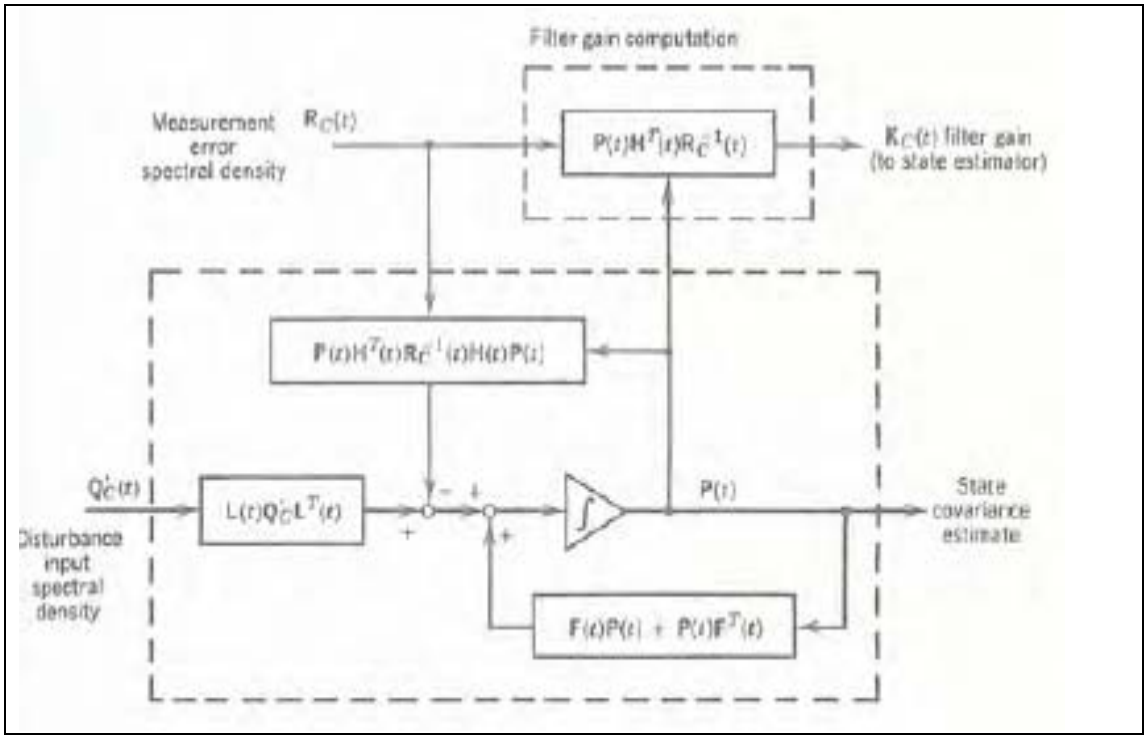
$$\tilde{\mathbf{y}}(t) = \mathbf{z}(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)$$

Αυτό συμβαίνει γιατί κάνουμε τους υπολογισμούς σε συνεχές πεδίο χρόνου. Στο παρακάτω σχήμα μπορούμε να δούμε το block diagram του φίλτρου.



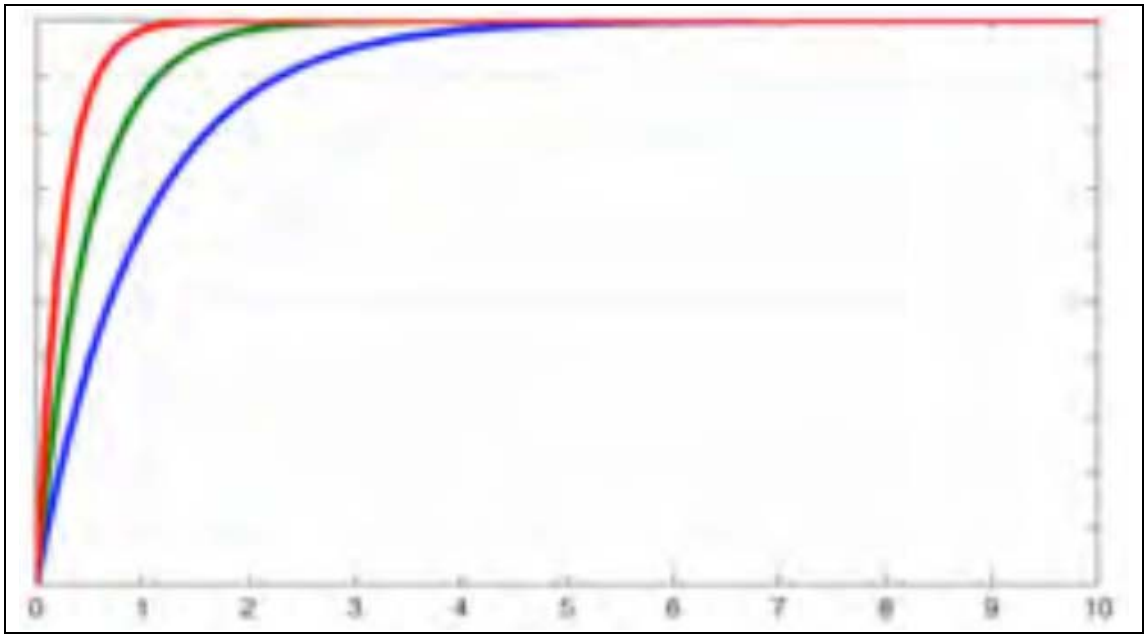
Σχήμα 2. Block diagram.

Στο παρακάτω σχήμα μπορούμε να δούμε το διάγραμμα block του φίλτρου, συμπεριλαμβάνομένων των υπολογισμών του κέρδους και της συνδιασποράς εκτίμησης.



Σχήμα 3. Block diagram.

Τέλος, μπορούμε να δούμε στο παρακάτω διάγραμμα, πώς εξελίσσεται η συνδιασπορά ώς προς το χρόνο.



Σχήμα 4. Συνδιασπορά.

1.2 **Φίλτρο** Kalman **για μη – γραμμικά προβλήματα.**

Η βασική μορφή του φίλτρου Kalman filter περιορίζει την χρήση του σε γραμμικά συστήματα. Πιο σύνθετα συστήματα, ωστόσο μπορεί να είναι μη – γραμμικά. Η μη – γραμμικότητα μπορεί να αφορά είτε το process model είτε το observation model είτε και τα δύο.

Εκτεταμένο φίλτρο Kalman (EKF).

Πρόκειται για την μη – γραμμική έκδοση του κλασσικού φίλτρου Kalman filter και διαφέρει ώς προς το ότι κάνει linearization ώς προς τον μέσο όρο και την συνδιασπορά των δεδομένων που επεξεργάζεται. Κάποτε, το εκτεταμένο φίλτρο θεωρούνταν το πιο καθιερωμένο πρότυπο στην θεωρία των μη – γραμμικών εκτιμήσεων.

Συνεπώς ήταν ιδιαίτερα εκτεταμένη και η χρήση του σε πληθώρα εφαρμογών, όπως τα GPS και τα συστήματα πλοήγησης. Ωστόσο, αυτήν την θέση την κατέχει το Unscented Kalman filter (UKF), το οποίο θα περιγράψουμε παρακάτω.

Στο EKF, η μετάβαση καταστάσεων και τα μοντέλα παρατήρησης δεν χρειάζεται να είναι γραμμικές συναρτήσεις ώς προς την κατάσταση, αλλά μπορεί αντί αυτού να είναι μια διαφορίσιμη συνάρτηση.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k$$

Στις παραπάνω σχέσεις, όπου $\mathbf{w}_k$ and $\mathbf{v}_k$ συμβολίζουμε τον θόρυβο της διαδικασίας και των παρατηρήσεων. Αυτά τα δύο μεγέθη υποθέτουμε ότι είναι Gaussian θόρυβοι με μηδενική μέση τιμή και συνδιασπορά $\mathbf{Q}_k$ και $\mathbf{R}_k$, αντίστοιχα.

Η συνάρτηση *f* χρησιμοποιείται για να υπολογιστεί η κατάσταση με την γνώση προηγούμενων εκτιμήσεων. Ομοίως, η συνάρτηση *h* χρησιμοποιείται για να υπολογιστεί η μέτρηση από την εκτιμισθείσα κατάσταση. Παρ'όλα αυτά ,οι *f* και *h* δεν μπορούν να χρησιμοποιηθούν για την εκτίση της συνδιασπορά, όπως έχουν. Χρειάζεται να υπολογίσουμε πρώτα ένα Ιακωβιανό μητρώο (Jacobian).

Σε κάθε χρονική στιγμή, το μητρώο αυτό υπολογίζεται με τις τρέχουσες εκτιμημένες καταστάσεις. Αυτά τα μητρώα χρησιμοποιεί το φίλτρο.αυτή η διαδικασία ουσιαστικά γραμμικοποιεί την μη – γραμμική συνάρτηση ώς προς τις

τρέχουσες εκτιμήσεις. Παρακάτω μπορούμε να δούμε τις εξισώσεις που εμπλέκονται στον υπολογισμό του εκτεταμένου φίλτρου.

Εκτίμηση κατάστασης	$\hat{\mathbf{x}}_{k\mid k-1} = f(\hat{\mathbf{x}}_{k-1\mid k-1}, \mathbf{u}_{k-1})$
Συνδιασπορά εκτιμημένης κατάστασης.	$\mathbf{P}_{k\mid k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1\mid k-1}\mathbf{F}_{k-1}^{\top} + \mathbf{Q}_{k-1}$

Βέλτιστο κέρδος φίλτρου	$\mathbf{K}_k = \mathbf{P}_{k\mid k-1}\mathbf{H}_k^{\top}\mathbf{S}_k^{-1}$
Ανανέωση εκτίμησης κατάστασης.	$\hat{\mathbf{x}}_{k\mid k} = \hat{\mathbf{x}}_{k\mid k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$
Ανανέωση συνδιασποράς εκτιμημένης κατάστασης.	$\mathbf{P}_{k\mid k} = (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k\mid k-1}$

Τα Ιακωβιανά μητρώα που προαναφέραμε, είναι τα παρακάτω.

$$\mathbf{F}_{k-1} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1\mid k-1}, \mathbf{u}_{k-1}}$$

$$\mathbf{H}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k\mid k-1}}$$

Unscented Kalman.

Όταν η μετάβαση καταστάσεων και το μοντέλο παρατήρησης είναι μη - γραμμικά το εκτεταμένο φίλτρο Kalman υστερεί σε απόδοση. Το **unscented Kalman filter (UKF)** χρησιμοποιεί μια τεχνική ντετερμινιστικής δειγματοληψίας, γνωστή ώς «unscented transform» για να συγκεντρώσει ένα σύνολο σημείων (σημεία sigma).

Αυτά τα σημεία χρησιμοποιούνται για να υπολογιστεί ο μέσος όρος και η συνδιασπορά των εκτιμήσεων. Έτσι, το φίλτρο είναι πιο ακριβές και γρήγορο. Επιπλέον, εκλίπει πια η αναγκη του υπολογισμού των Ιακωβιανών μητρώων. Παρακάτω μπορούμε να δούμε τις βασικές εξισώσεις του φίλτρου.

$$\mathbf{x}^a_{k-1|k-1} = [\hat{\mathbf{x}}^T_{k-1|k-1} \quad E[\mathbf{w}^T_k] \ ]^T$$

$$\mathbf{P}^a_{k-1|k-1} = \begin{bmatrix} \mathbf{P}_{k-1|k-1} & 0 \\ 0 & \mathbf{Q}_k \end{bmatrix}$$

Με τις παρακάτω εξισώσεις μπορούμε να υπολογίσουμε τα σημεία σίγμα.

$\chi^0_{k-1\|k-1}$	$= \mathbf{x}^a_{k-1\|k-1}$	
$\chi^i_{k-1\|k-1}$	$= \mathbf{x}^a_{k-1\|k-1} + \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1\|k-1}}\right)_i$	$i = 1..L$
$\chi^i_{k-1\|k-1}$	$= \mathbf{x}^a_{k-1\|k-1} - \left(\sqrt{(L+\lambda)\mathbf{P}^a_{k-1\|k-1}}\right)_{i-L}$	$i = L+1, \ldots 2L$

Τα σημεία αυτά παράγουν την πρόβλεψη και την συνδιασπορά.

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2L} W^i_s \chi^i_{k|k-1}$$
$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2L} W^i_c \ [\chi^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}][\chi^i_{k|k-1} - \hat{\mathbf{x}}_{k|k-1}]^T$$

Το κέρδος του φίλτρου δίνεται από την εξίσωση:

$$K_k = \mathbf{P}_{x_k z_k} \mathbf{P}^{-1}_{z_k z_k}$$

από την παρακάτω εξίσωση, παίρνουμε την ανανεωμένη κατάσταση.

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

Τέλος, από την παρακάτω εξίσωση παίρνουμε την ανανεωμένη συνδιασπορά.

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - K_k \mathbf{P}_{z_k z_k} K^T_k$$

1.3 **Αντίστροφα προβλήματα.**

Το αντίστροφο πρόβλημα, εμφανίζεται τόσο στις θετικές επιστήμες, όσο και στις τεχνικές εφαρμογές. Τι είναι όμως το αντίστροφο πρόβλημα: Σύμφωνα με τα [12,13], αντιστροφή, είναι το σύνολο των μεθόδων που εφαρμόζεται σε ένα σύνολο παρατηρήσεων, και χρησιμοποιείται για την εξαγωγή χρήσιμων συμπερασμάτων.

Σκοπός δεν είναι η απλή εφαρμογή των μεθόδων αλλά η κατάλληλη οργάνωσή τους, έτσι ώστε να αποφέρουν το μέγιστο των πληροφοριών από ένα δοσμένο σύνολο παρατηρήσεων.

Η επιλογή των παραμέτρων που θα περιγράψουν το υπό μελέτη σύστημα, δεν θα είναι η μοναδική. Υπάρχουν πολλοί συνδυασμοί παραμέτρων μοντέλου που θα οδηγήσουν στο ίδιο αποτέλεσμα. Κοινή επιδίωξη πάντα είναι η επιλογή μοντέλου που να παρέχει το μέγιστο των πληροφοριών για τον πραγματικό χώρο μελέτης.

Το σύνολο των πιθανών μοντέλων που περιγράφουν επαρκώς το υπό μελέτη φυσικό σύστημα, αποτελεί τον χώρο του μοντέλου. Αντίστοιχα, το σύνολο των δεδομένων τα οποία χαρακτηρίζουν το μοντέλο, αποτελούν τον χώρο των δεδομένων.

Πως γίνεται όμως η επίλυση του αντιστρόφου προβλήματος; Γενικά, ακολουθείται η εξής διαδικασία: πρόβλεψη, αριθμητικός υπολογισμός, σύγκριση. Προστέθηκε ακόμα ένα στοιχείο κατά την επίλυση του αντιστρόφου προβλήματος που είναι η αναπροσαρμογή του μοντέλου με βάση τα αποτελέσματα (feedback).

Όταν ερευνάται ένας φυσικός νόμος, γίνεται αρχικά μια πρόβλεψη, υπολογίζεται το αποτέλεσμα και συγκρίνεται με το αποτέλεσμα του πειράματος. Αν η σύγκριση είναι μη ικανοποιητική, αυτό σημαίνει ότι και η επιλογή του αρχικού μοντέλου (αρχική πρόβλεψη) ήταν κακή.

Η ύπαρξη σφαλμάτων στα δεδομένα υποχρεώνει τη λεπτομερή μελέτη του αντιστρόφου προβλήματος, σχετικά με τα ακόλουθα θέματα :

Ευστάθεια (stability) της λύσης. Σε ορισμένα αντίστροφα προβλήματα, συνήθως σε μια περιοχή του χώρου του μοντέλου (model space), οι λύσεις είναι ασταθείς. Σε αυτή τη περίπτωση, μικρά σφάλματα στα δεδομένα οδηγούν σε μεγάλη αβεβαιότητα στις λύσεις, δηλαδή σε ένα μεγάλο αριθμό πιθανών λύσεων. Είναι εύκολο να αντιληφθούμε ότι σε αυτό το θέμα η επιθυμητή κατάσταση είναι εντελώς αντίθετη με αυτή του ευθέως προβλήματος. Στο ευθύ πρόβλημα θα θέλαμε μεγάλες διαταραχές στις παραμέτρους να μην οδηγούν σε σημαντικές αλλαγές στα δεδομένα παρατήρησης. Όμως, σε μια τέτοια περίπτωση, θα είχαμε ένα φοβερά ασταθές αντίστροφο πρόβλημα. Η ιδανική λύση θα ήταν η περίπτωση όπου διαταραχές στις παραμέτρους του μοντέλου, να προκαλούν ίδιου μεγέθους διαταραχές στα δεδομένα μας, έτσι ώστε τόσο το ευθύ όσο και το αντίστροφο πρόβλημα να παρουσιάζουν μια σχετική ευστάθεια.

Ευρωστία (robustness) της λύσης. Με τον όρο αυτό εννοούμε την ανθεκτικότητα της λύσης σε δεδομένα τα οποία ξεφεύγουν σημαντικά λόγω μεγάλων τυχαίων σφαλμάτων από τη στατιστική σφάλματος των δεδομένων.

Γνωστό παράδειγμα, η ευρύτατα χρησιμοποιούμενη μέθοδος των ελαχίστων τετραγώνων, η οποία δεν είναι εύρωστη μια και το κάθε σημείο επιδρά στη λύση με βάρος, το οποίο αυξάνει όσο περισσότερο απέχει το σημείο από τη μέση λύση, με αποτέλεσμα σημεία με μεγάλα τυχαία σφάλματα να επηρεάζουν πολύ τη λύση.

Βάρος των δεδομένων (data importance). Σε κάθε περίπτωση επίλυσης ενός αντιστρόφου προβλήματος, πρέπει να εξετάζεται το βάρος της επίδρασης των δεδομένων στην τελική λύση. Η μελέτη αυτή έχει ιδιαίτερη σημασία ως προς τον απαραίτητο αριθμό και την κατάλληλη επιλογή των δεδομένων, τα οποία θα επιτρέψουν την καλύτερη ανακατασκευή του μοντέλου.

1.4 **Φίλτρο** Kalman **για αντίστροφα προβλήματα.**

Ο υπολογιστικός φόρτος του υπολογισμού του αντίστροφου προβλήματος ενός υπολογισμού Kalman είναι ανάλογος του κύβου του αριθμού των μετρήσεων που επεξεργάζονται.

Ακόμα και όταν επεξεργαζόμαστε δεδομένα ταυτόχρονα, πράγμα που βελτιώνει την απόδοση του υπολογισμού, δεν είναι απίθανο να έχουμε να κάνουμε με αραιά μητρώα.

Για να είναι το φιλτράρισμα με χρήση Kalman αξιπόπιστο, πρέπει να το τροφοδοτούμε συνεχώς με δεδομένα πραγματικού χρόνου. Το πόσο καλό είναι το φιλτράρισμα, εξαρτάται από την χρήση της διασποράς και της συνδιασποράς του σφάλματος μεταξύ όλων των μετρήσεων και της εκτιμημένης κατάστασης. Τα μεγάλα αραιά μητρώα, μπορούμε να τα αντιστρέψουμε χωρίζοντας τα σε blocks, όπως στην παρακάτω σχέση.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

Κεφάλαιο

2

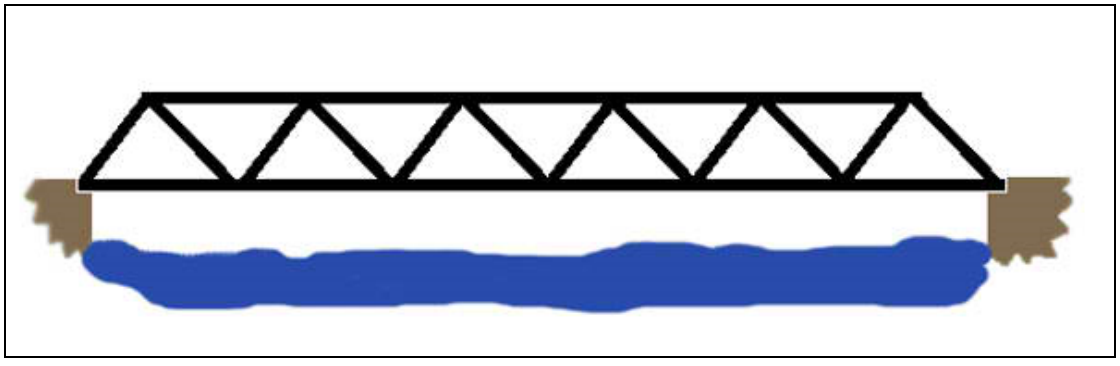
Το πρόβλημα της γέφυρας.

2.1 Εισαγωγή.

Στην εργασία αυτή, ο αντικειμενικό μας σκοπός είναι να μελετήσουμε την άσκηση δυνάμεων σε ένα δικτύωμα που αναπαριστά μια γέφυρα και να εκτιμήσουμε τιμές δυνάμεων με χρήση φιλτραρίσματος Kalman. Αναλύουμε το πρόβλημα με περισσότερη λεπτομέρεια παρακάτω.

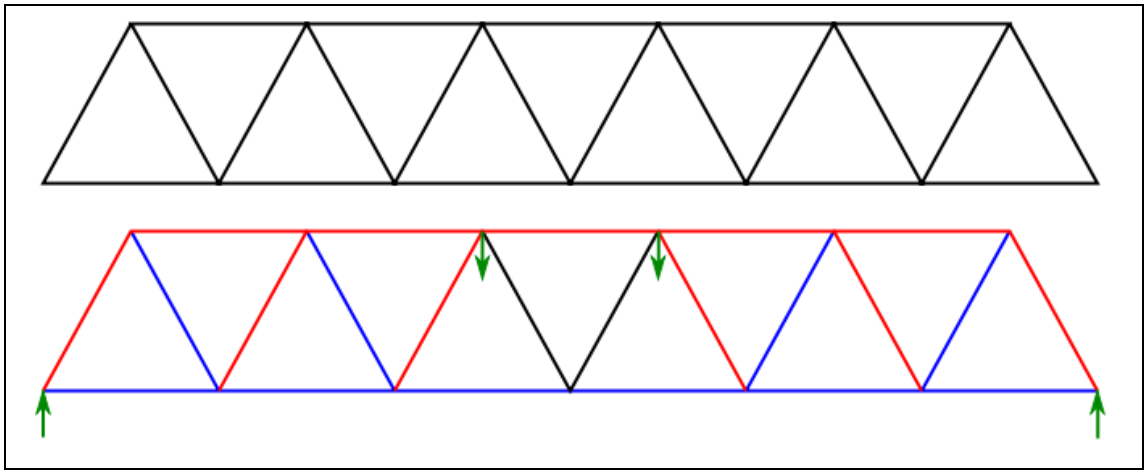
2.2 Το πρόβλημα της γέφυρας.

Στην παρούσα εργασία, θεωρούμε μια γέφυρα όμοια με το σχήμα που φαίνεται στην παρακάτω εικόνα.



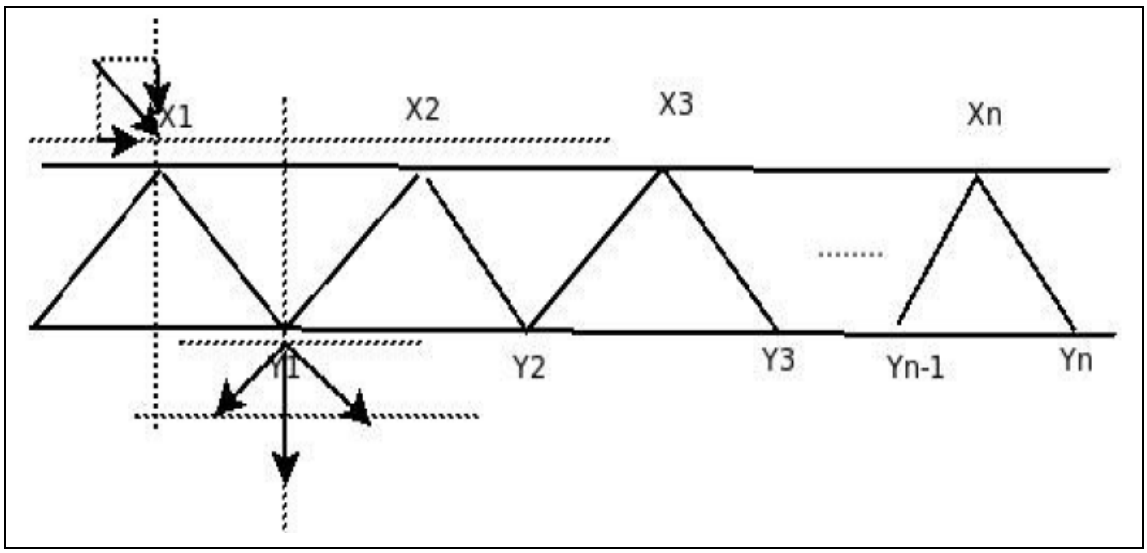
Σχήμα 5. Η γέφυρα του προβλήματος (πηγή: Encyclopedia Britannica, 1999).

Όπως είναι προφανές, στα σημεία – κόμβους της γέφυρας ασκούνται δυνάμεις. Όπως φαίνεται και στο παρακάτω σχήμα, σε κάποια σημεία ασκούνται τάσεις και σε κάποια, συμπιέσεις.



Σχήμα 6. Τάσεις και συμπιέσεις στην γέφυρα (Πηγή: wikimedia, 2010).

Έστω ότι έχουμε σημεία X, πλήθους n. Δηλαδή |X| = n. Τότε θα έχουμε n+1 σημεία Y. Στα σημεία Xi με i = {1,2,….,n} ασκούνται δυνάμεις με τιμές που μεταβάλλονται με τον χρόνο και γωνίες εφαρμογής που είναι τυχαίες. Αυτό έχει σαν αποτέλεσμα να ασκούνται δυνάμεις στα σημεία Yi με i = {1,2,….,n, n+1}. Δείτε το παρακάτω σχήμα.

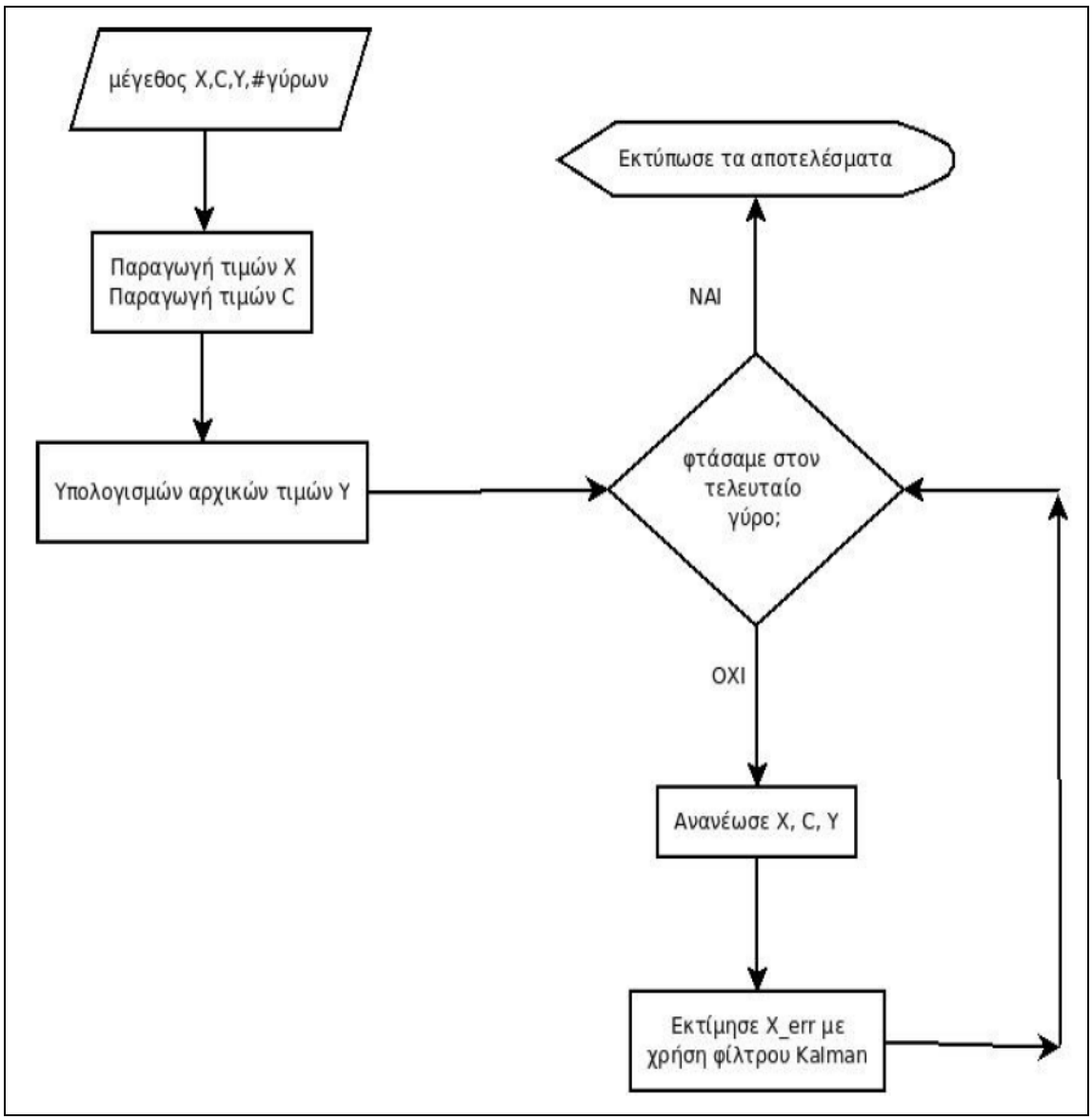


Σχήμα 7. Ανάλυση δυνάμεων.

Με ανάλυση δυνάμεων, μπορούμε να υπολογίσουμε τις δυνάμεις στα σημεία Yi, εφ'όσον γνωρίζουμε πώς μεταβάλλονται οι τιμές στα Xi.

Τί γίνεται όμως αν για κάποιο λόγο δεν μπορούμε να γνωρίζουμε τα Xi; Σε αυτήν την περίπτωση πρέπει να εκτιμήσουμε τις τιμές στα σημεία Xi.

Για μια τέτοια εκτίμηση χρησιμοποιούμε το λεγόμενο φίλτρο Kalman. Στην εργασία αυτή, εμείς θα πρέπει γνωρίζοντας τις δυνάμεις στα σημεία Y, να εφαρμόσουμε φίλτρο Kalman για να δούμε πόσο σωστές ήταν οι εκτιμήσεις. Επίσης μας ενδιαφέρει πόσο γρήγορα φτάνει αυτό το φίλτρο σε «καλές» εκτιμήσεις. Χρήσιμο είναι και το παρακάτω διάγραμμα ροής.

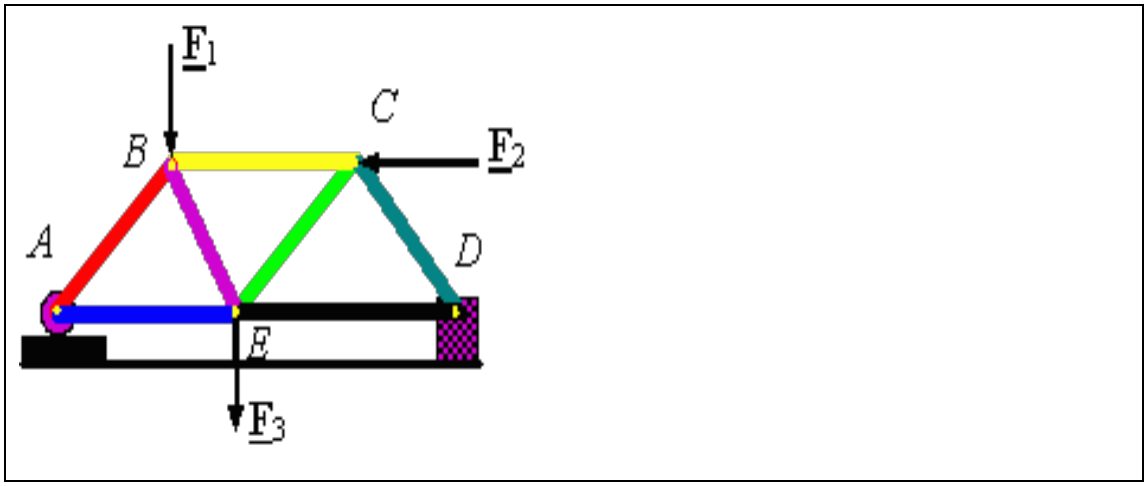


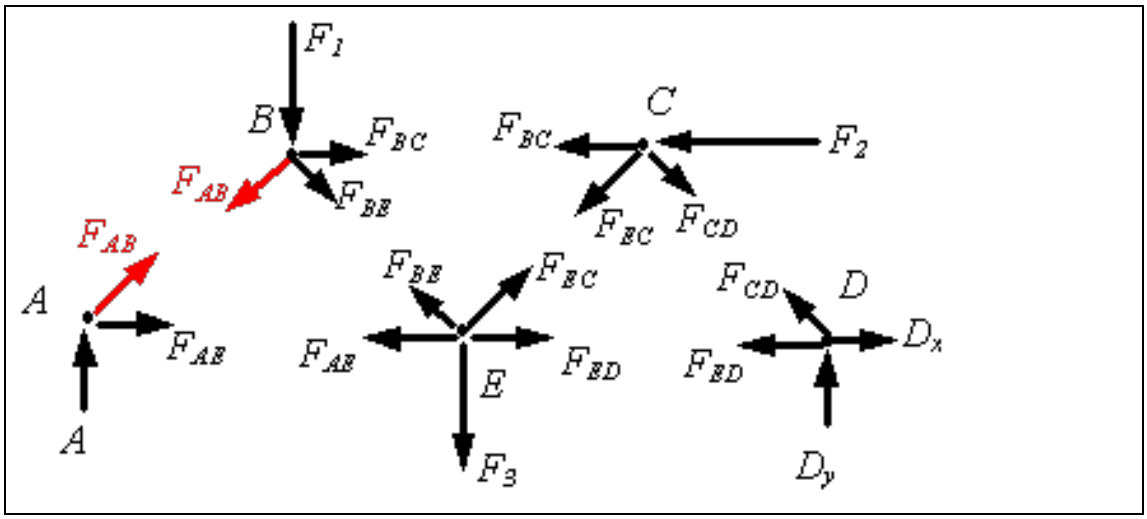
Σχήμα 8. Διάγραμμα Ροής.

2.3 **Λύση.**

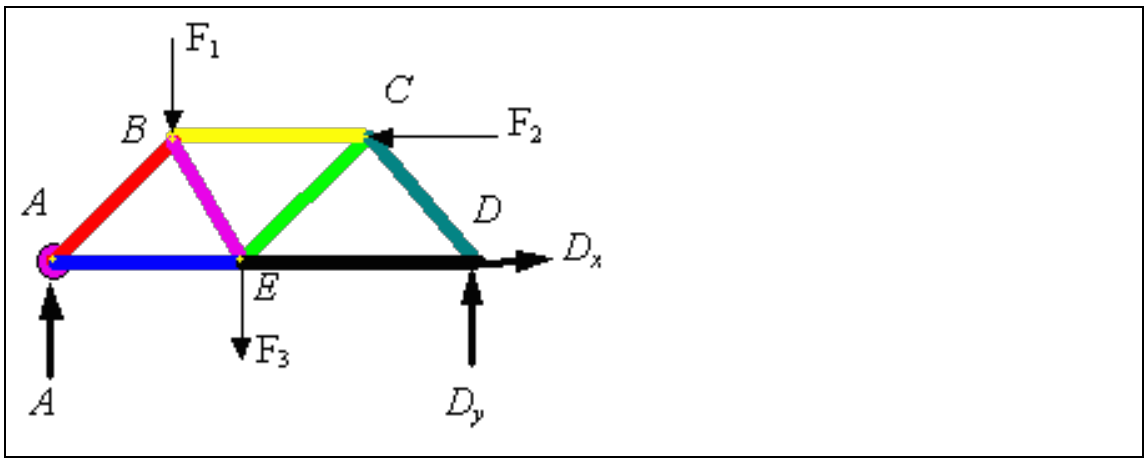
Γνωρίζοντας το μέγεθος του προβλήματος, δηλαδή το πλήθος των στοιχείων Χ, μπορούμε αλγεβρικά να υπολογίσουμε τις δυνάμεις στα σημεία Υ. Αυτό γίνεται με ανάλυση δυνάμεων, όπως την παρουσιάζουμε παρακάτω.

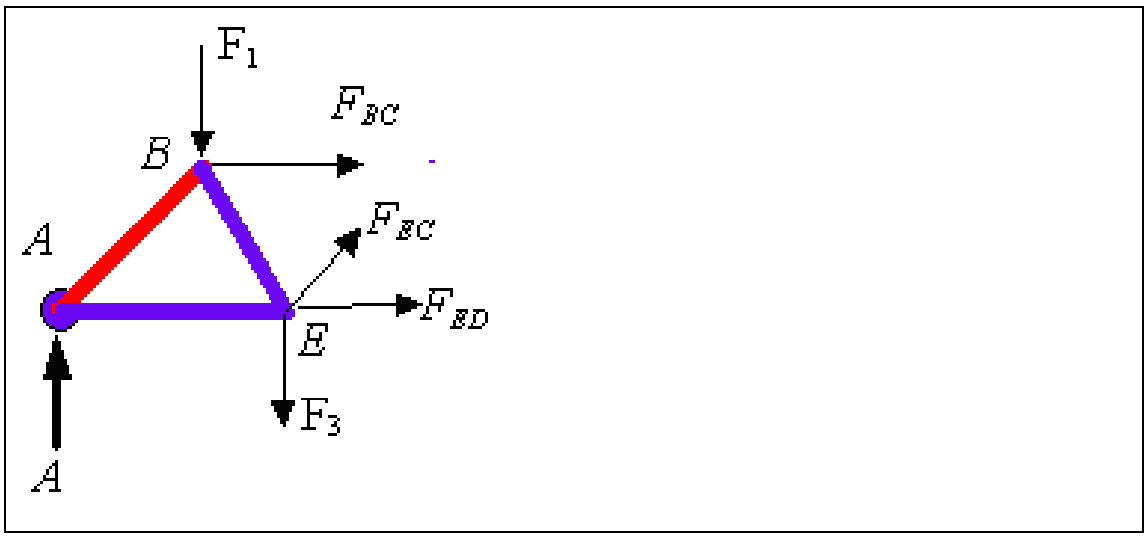
Το δικτύωμα που χρησιμοποιούμε στο πρόβλημα μας, ονομάζεται δικτύωμα warren (warren truss) και χρησιμοποιείται αρκετά συχνά στην κατασκευή γεφυρών [3,5,8,9,10]. Στα παρακάτω σχήματα μπορούμε να δούμε, πως αναλύονται οι δυνάμεις σε μια τυπική γέφυρα τέτοιου τύπου.





Σημειωτέον, μπορύμε να γράψουμε δύο εξισώσεις ανα κόμβο - σημείο, ($\sum F_x = 0, \sum F_y = 0$). Στο παρακάτω παράδειγμα, όπου έχουμε πέντε κόμβους, μπορούμε να δημιουργήσουμε δέκα εξισώσεις ισσοροπίας.





Εφ'όσον γνωρίζουμε και το μέγεθος του προβλήματος και τις δυνάμεις στα σημεία X και βρήκαμε τις δυνάμεις στα σημεία Y, μπορούμε να υποθέσουμε ότι δεν γνωρίζουμε τις δυνάμεις στα σημεία X και να εφαρμόσουμε φίλτρο Kalman για να τις εκτιμήσουμε. Έπειτα, μπορούμε να δούμε και πόσο καλή ήταν η εκτίμηση μας και πόσο γρήγορα συγκλίνουν οι εκτιμήσεις μας με τις πραγματικές τιμές.

2.4 **Επισκόπηση πηγών.**

Για την υλοποίηση της εργασίας μας χρησιμοποιήσαμε τρείς κατηγορίες πηγών. Η πρώτη αφορούσε την πλατφόρμα που χρησιμοποιήσαμε για να υλοποιήσουμε την εργασία. Αυτή είναι το προγραμματιστικό περιβάλλον Matlab. Μας βοήθησε αρκετά το επίσημο help section που συνοδεύει το προϊόν, το οποίο είναι αρκετά λεπτομερές και εισάγει τον χρήστη σε κάθε λεπτομέρεια που αφορά την δομή και τα πακέτα εργαλείων.

Επίσης, βοηθηθήκαμε από το επίσημο forum υποστήριξης της MathWorks, στο οποίο μπορεί κάποιος χρήστης του Matlab να βρει λύσεις για σχεδόν οποιοδήποτε πρόβλημα που μπορεί να έχει κατά τον σχεδιασμό και την ανάπτυξη κώδικα σε Matlab. Παράλληλα, χρήσιμο αποδείχθηκε το KalmTool του Magnus Nørgaard.

Η δεύτερη κατηγορία αφορούσε την μελέτη του προβλήματος που είχαμε να λύσουμε. Επειδή αυτό αφορούσε μια γέφυρα στην οποία επιδρούν διάφορες δυνάμεις υπό διάφορες γωνίες, όπως θα δούμε παρακάτω, χρησιμοποιήσαμε το σύγγραμμα του Young για να αναλύσουμε σωστά τις δυνάμεις που παίζουνε ρόλο στο πρόβλημα μας.

Η τρίτη κατηγορία πηγών αφορούσε την υλοποίηση και χρήση του φίλτρου στο συγκεκριμένο πρόβλημα. Σε αυτό μας βοήθησε το σύγγραμμα του Ασημάκη Νικόλαου «Φίλτρα Kalman και Λαϊνιώτη» το οποίο περιγράφει αναλυτικά το συγκεκριμένο φίλτρο, τις παραλλαγές του, την φυσική σημασία και τέλος δίνει χρήσιμα παραδείγματα. Επειδή το συγκεκριμένο φίλτρο συνεπάγεται και υπολογισμούς μητρώων θεωρήσαμε χρήσιμο να συμβουλευτούμε και το σύγγραμμα του Gilbert Strang, «Γραμμική άλγεβρα και εφαρμογές».

Κεφάλαιο

3

Η υλοποίηση

3.1 Εισαγωγή.

Στο κεφάλαιο αυτό, θα παρουσιάσουμε την υλοποίηση του προβλήματος που μελετήσαμε και την λύσης που δώσαμε. Θα παρουσιάσουμε την πλατφόρμα υλοποίησης, την λειτουργία και τον πηγαίο κώδικα που την αποτελεί.

3.2 Απαιτήσεις συστήματος.

Η υλοποίηση αναπτύχθηκε στο σύστημα του οποίου τα χαρακτηριστικά φαίνονται στον παρακάτω πίνακα.

CPU	RAM	HDD	OS	S/W
AMD ATHLON XP 2400+ 32bit	1,5GB DDR	WD 80GB IDE	OpenSuse 10.3, 32bit, MS Windows XP 32 bit	GNU C compiler ver. 4.2.4, make, doxygen, Matlab R2009a

3.3 Αρχεία.

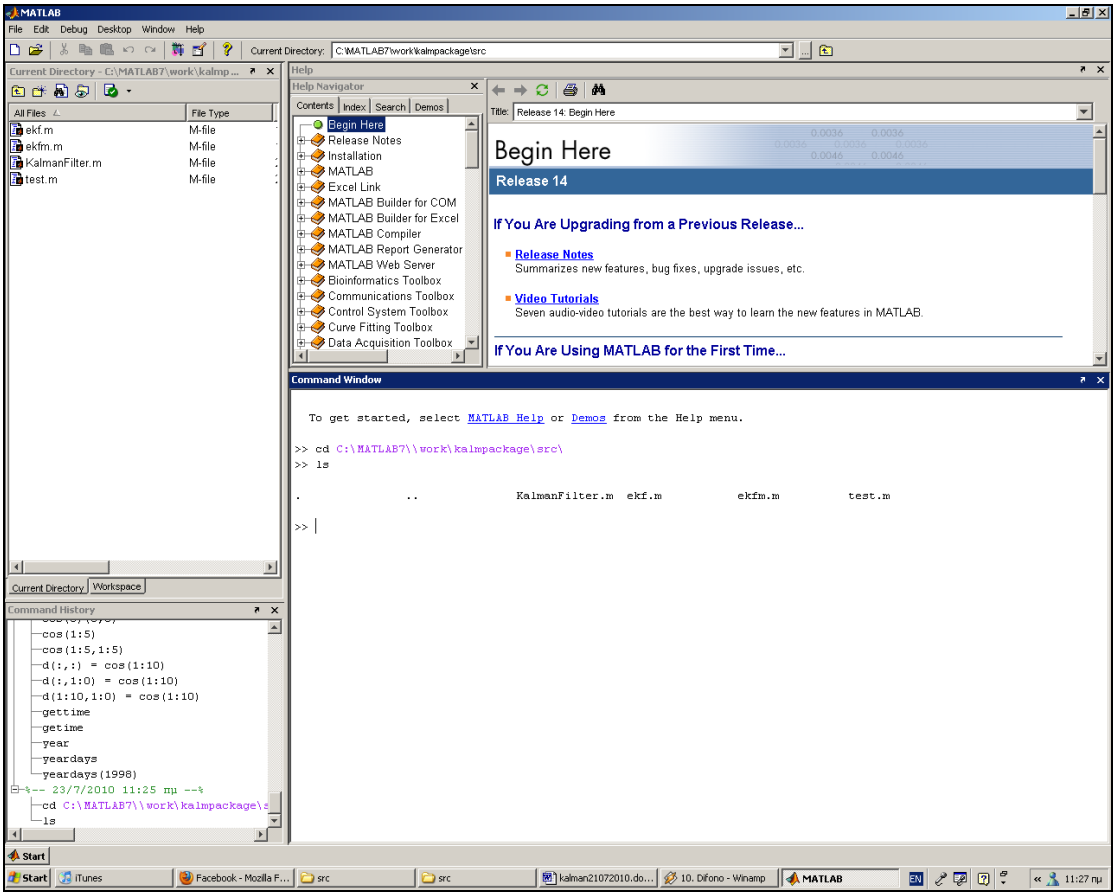
Η υλοποίηση αποτελείται από ένα σύνολων αρχείων – matlab scripts, των οποίων τις λεπτομέρειες λειτουργία και τον πηγαίο κώδικα παρουσιάζουμε στις επόμενες ενότητες. Χρήσιμες πληροφορίες είχαν τα [2,6,7,11].

Το πακέτο kalmpackage αποτελείται από τους φακέλους data και src. Ο φάκελος data περιέχει δειγματοληπτικά δεδομένα από δοκιμές που κάναμε τα οποία παρουσιάζουμε στο επόμενο κεφάλαιο.

Ο φάκελος src περιέχει τον πηγαίο κώδικα της εφαρμογής, που έχει υλοποιηθεί σε matlab. Περιλαμβάνει το kalmtool, στο οποίο βασίστηκε η υλοποίηση μας και τα αρχεία που εκτελούν πειράματα για το πρόβλημα της γέφυρας, που περιγράψαμε παραπάνω.

3.4 **Λειτουργία.**

Για να εκτελέσουμε την εφαρμογή, πρέπει να έχουμε έτοιμο ένα υπολογιστικό σύστημα που να ανταποκρίνεται στις απαιτήσεις που τέθηκαν στην παράγραφο 3.2. Όπως φαίνεται και στην εικόνα παρακάτω, εκκινούμε το περιβάλλον εργασίας της matlab, πλοηγούμαστε στον φάκελο Demo και τρέχουμε το σενάριο trussdemo.m.



Σχήμα 9. Το περιβάλλον εργασίας της Matlab.

Η αλληλουχία των εντολών είναι η παρακάτω:

```
cd path/to/src

trussdemo
```

3.5 **Κώδικας.**

Το πρώτο αρχείο που θα μελετήσουμε είναι το σενάριο προσομοίωσης, το trussdemo.m. Αυτό ξεκινάει την λειτουργία του, καθαρίζοντας την οθόνη εντολών της matlab και το workspace. Αυτό το κάνει ώς εξής:

```
clear;

clc;
```

Στην συνέχεια ορίζει παραμέτρους εκτέλεσης του πειράματος, όπως την χρονική διάρκεια (Time_steps) ή το μέγεθος του δικτυώματος – γέφυρας (Bridge_size). Αυτές τις μεταβλητές μπορεί και να αλλάζει ο χρήστης, προκειμένου να αλλάξουν τα δεδομένα της προσομοίωσης. Δείτε το παρακάτω τμήμα κώδικα.

```
x0 = [3e5;2e4;1e-3];       % Initial state vector
Q  = zeros(3);           % Covariance of process noise
r  = 1e4;                  % Covariance of measurement noise
P0 = diag([1e6 4e6 1e-4]); % Initial covariance on state estimate
gamma = 5e-5;              % Model parameter
M     = 1e5;               % Horizontal radar position (ft)
H     = 1e5;               % Vertical radar position (ft)
rksteps = 64;              % RK steps / sampling period
delta = 1/rksteps;         % Fast "Sampling period"
Bridge_size  = 5;
Tfinal = 30;               % Xronikes stigmes
rad(1:Bridge_size) = pi/randint(1,1,[1,1000]);

% ---- Generate test data ----
% randn('seed',0);           % Set seed for random noise - matlab
release
% 14

randn('state',0);           % Set seed for random noise
ysim   = zeros(Tfinal,1);  % Store true y sequence
xtrue  = [x0';zeros(Tfinal,3)];
xhatmat= zeros(Tfinal+1,3,Bridge_size);
v      = zeros(3,1);       % No process noise
w0     = 0;                % Mean of measurement noise
clear optpar
optpar.init = [delta M H gamma]; % Prepare initialization parameters
optpar.F = eye(3);
```

```
optpar.G=1;
X_points = zeros(Tfinal+1,Bridge_size);
Y_points = zeros(Tfinal,Bridge_size);
```

Ακολουθεί η αρχικοποίηση των δυνάμεων στα σημεία X και ο υπολογισμός των αντίστοιχων στα σημεία Y της γέφυρας. Για λόγους αναγνωσιμότητας του κώδικα, επιλέξαμε να κάνουμε αυτές τις διαδικασίες σε εμφωλευμένους βρόγχους. Δείτε το παρακάτω τμήμα κώδικα.

```
% Run the simulation
x      = x0;
truss1(optpar.init);              % Initialze state update
truss2(optpar.init);              % Initialize observation equation
for k=1:Tfinal,
  for kk=1:1/delta,
     x=truss1(x,[],v)*sin(pi/6);
  end
  xtrue(k+1,:) = x';
  X_points(k,:) = state_eq(X_points(k,:),rad,[]);
 ysim(k)=truss2(x,w0);
end
```

Στην συνέχεια, καλούμε το φίλτρο για να εκτιμήσουμε τις τιμές στα σημεία X που προκάλεσαν την παραμόρφωση που μετρήσαμε στα σημεία Y. Για όλα τα σημεία της γέφυρας και όλες τις χρονικές στιγμές, καλούμε το φίλτρο.

```
ytrue = repmat(ysim,1,Bridge_size) +
sqrt(r)*randn(Tfinal,Bridge_size);
x0hat = [x0(1:2);3e-5];        % Initial state estimate
idx   = [1:Tfinal]'*rksteps;   % Measurement time stamps (in rk-
periods)
[v,d] = eig(P0);           % Cholesky factor of initial state
covariance
Sx0 = real(v*sqrt(d));
[v,d] = eig(Q);            % Cholesky factor of process noise
covariance
Sv    = real(v*sqrt(d));
[v,d] = eig(r);            % Cholesky factor of measurement noise
covariance
Sw    = real(v*sqrt(d));

for k=1:Bridge_size,
fprintf('\nExperiment no. %d\n',k);

  %----- Estimate state trajectory -----
  switch Method
     case 1,

[xhat,Pmat]=ekf(xfunc,yfunc,linfunc,x0hat,P0,Q,r,[],ytrue(:,k),idx,op
tpar);
     case 2,

[xhat,Smat]=dd1(xfunc,yfunc,x0hat,P0,Q,r,[],ytrue(:,k),idx,optpar);
     case 3
```

```
        [xhat,Smat]=dd1fall(x0hat,Sx0,Sv,Sw,[],ytrue(:,k),idx,optpar);
      case 4,

[xhat,Smat]=dd2(xfunc,yfunc,x0hat,P0,Q,r,[],ytrue(:,k),idx,optpar);
      case 5,
        [xhat,Smat]=dd2fall(x0hat,Sx0,Sv,Sw,[],ytrue(:,k),idx,optpar);
      otherwise
    error('No valid filter method selected. Method=1...5')
end
```

Ακολουθεί η συγκέντρωση και η αποθήκευση των αποτελεσμάτων, στο αρχείο matlab.mat. Δείτε το παρακάτω τμήμα κώδικα.

```
  % ----- Store data -----
  xhatmat(:,:,k) = xhat([1;(idx+1)],:);
end
xhatmean = mean(xhatmat,3);    % Calculate mean values

for counter = 1:Bridge_size
    tmp = xhatmat(:,1,counter);
    X_points(:,counter) = tmp;
end;
Y_points = ytrue;

save;
```

Στο τέλος, με την εντολή save, αποθηκεύουμε τα δεδομένα εισόδου και εξόδου σε ένα .mat αρχείο, για την περίπτωση που θέλουμε να επεξεργαστούμε περαιτέρω τα αποτελέσματα. Στην συνέχεια παρουσιάζουμε την υλοποίηση του κώδικα του φίλτρου.

ekfm.m:

```
function [xhat_data,Pmat]=ekfm(kalmfilex,kalmfiley,linfile,xbar,...
         P0,q,r,u,y,timeidx,optpar)

...
nx           = size(P0,1); % # of states
nv           = size(q,1);  % # of process noise sources
if isempty(xbar),          % Set to x0=0 if not specified
  xbar = zeros(nx,1);
elseif length(xbar)~=nx,
  error('Dimension mismatch between x0 and P0');
end
streams    = length(y);
if ~(iscell(kalmfiley) & iscell(r) & iscell(timeidx) & iscell(y))
  error('"yfunc", "r", "tidx", and "y" must be cell array');
elseif (streams~=length(r) | streams~=length(timeidx) | ...
                                streams~=length(kalmfiley))
  error('"yfunc", "r", "tidx", and "y" must have same number of
cells');
end
ny         = 0;                % Total number of observations
```

```
lastsample = 0;                % Number of sample containing last
observation
idx1 = zeros(streams,1);       % Index to start of each stream in
ybar
idx2 = zeros(streams,1);       % Index to end of each stream in ybar
for n=1:streams,               % Wrap information about observation
stream
  obs(n).yfunc = kalmfiley{n}; % into data structure
  obs(n).y     = y{n};
  obs(n).tidx  = timeidx{n};
  obs(n).ny    = size(obs(n).y,2);
  obs(n).nobs  = size(obs(n).y,1);
  obs(n).r     = r{n};
  obs(n).nw    = size(obs(n).r,1);
  if (obs(n).nobs~=size(obs(n).tidx,1)),
    error('Dimension mismatch between y and tidx');
  end
  ny = ny + obs(n).ny;
  if obs(n).tidx(end)>lastsample,
     lastsample=obs(n).tidx(end);
  end
  idx1(n) = ny - obs(n).ny + 1;
  idx2(n) = ny;
end
if isempty(u),                 % No inputs
  nu = 0; samples = lastsample; uk1 = [];
else
  [samples,nu] = size(u);  % # of samples and inputs
end

Pxbar = P0;                    % A priori estimate = initial covariance
xhat_data = zeros(samples+1,nx); % Matrix for storing state estimates
Pmat      = zeros(samples+1,0.5*nx*(nx+1)); % Matrix for storing cov.
matrices
pidx      = find(tril(reshape(1:nx*nx,nx,nx))); % Index in P
ybar      = zeros(ny,1);
yidx  = ones(streams,1);   % Index into y-vectors


% ----- Initialize state+output equations and linearization -----
if nargin<11,                  % No optional parameters passed
   optpar = [];
end
if isfield(optpar,'init')  % Parameters for m-functions
   initpar = optpar.init;
else
   initpar = [];
end
vmean = zeros(nv,1);           % Mean of process noise
for n=1:streams,               % Mean of measurement noise
   obs(n).wmean = zeros(obs(n).nw,1);
end

feval(kalmfilex,initpar);      % Initialize state equation
for n=1:streams,
   feval(obs(n).yfunc,initpar);% Initialize output equations
end
feval(linfile,initpar);        % Initialize linearization

counter = 0;                   % Counts the progress of the filtering
session
```

```
waithandle=waitbar(0,'Filtering in progress'); % Initialize waitbar

% >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> FILTERING
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
for k=0:samples,

  % --- Measurement update (a posteriori update) ---
  for n=1:streams,
    ybar(idx1(n):idx2(n)) = feval(obs(n).yfunc,xbar,obs(n).wmean);
    if (k<=obs(n).tidx(end) & obs(n).tidx(yidx(n))==k),

      % Linearization
      [C,G] = feval(linfile,xbar,[],obs(n).wmean,n);

      % Kalman gain
      if isempty(G),
         K = Pxbar*C'/(C*Pxbar*C'+obs(n).r); % Noise enters directly
      else
         K = Pxbar*C'/(C*Pxbar*C'+G*obs(n).r*G');% General update
      end

      % A posteriori covariance
      Pxbar = Pxbar-K*C*Pxbar;

      % State estimate
      xbar = xbar + K*[obs(n).y(yidx(n),:)'-ybar(idx1(n):idx2(n))];
      yidx(n) = yidx(n) + 1;              % Update index in time
vector
    end
  end
  xhat = xbar;
  Px   = Pxbar;

  % --- Time update (a'priori update) of state and covariance ---
  if k<samples,
    if nu>0 uk1 = u(k+1,:)'; end
    xbar=feval(kalmfilex,xhat,uk1,vmean);    % State update
    [A,F] = feval(linfile,xhat,uk1,vmean,0); % Linearization
    if isempty(F),                    % Covariance update
       Pxbar = A*Px*A' + q;           % Noise enters directly
    else
       Pxbar = A*Px*A' + F*q*F';      % General update
    end
  end

  % --- Store results ---
  xhat_data(k+1,:) = xhat';
  Pmat(k+1,:)      = Px(pidx)';

  % --- How much longer? ---
  if (counter+0.01<= k/samples),
     counter = k/samples;
     waitbar(k/samples,waithandle);  % Update waitbar
  end
end
close(waithandle);              % Close waitbar window
```

Συνάρτηση για το linearization, trusslin.m.

```
function [M,N]=trusslin(x,u,vw,flag)

persistent A0 F0 C0 G0 gamma H Mp delta I % Make variables static

% Check if variables should be initialized
if nargin==1
   delta = x(1);
   gamma = x(4);
   Mp    = x(2);
   H     = x(3);

   A0 = eye(3);
   F0 = [];
   C0 = zeros(3,5); C0(1,1) = 1; C0(2,2) = 1; C0(3,3) = 1;
   G0 = [];
   return
end

% Linearize state equation
if flag==0,
   ak  = exp(-gamma*x(1))*delta;
   a12 = -ak*exp(gamma*x(1));
   a21 = ak*gamma*x(2)*x(2)*x(3);
   a22 = -ak*2*x(2)*x(3);
   a23 = -ak*x(2)*x(2);

   M = [1+0.5*a12*a21  a12+0.5*a12*a22  0.5*a12*a23;
        a21+0.5*a22*a21  1+a22+0.5*(a12*a21+a22*a22)
a23+0.5*a22*a23;
      0 0 1];
   N  = F0;

% Linearize output equation
elseif flag==1,
   tmp = x(1)-H;
   M = [tmp/sqrt(Mp*Mp+tmp*tmp) 0 0];
   N = G0;
end
```

Συνάρτηση για την ανανέωση των καταστάσεων, truss1.m, state_eq.m και trussdot.m

```
function x=truss1(x,u,v)

persistent delta
if nargin==1,
  delta = x(1);
  trussdot(x);
else
  k1 = delta*trussdot(x,u,v);
  k2 = delta*trussdot(x+0.5*k1,u,v);
  k3 = delta*trussdot(x+0.5*k2,u,v);
  k4 = delta*trussdot(x+k3,u,v);
```

```
  x  = x + (k1+2*k2+2*k3+k4)/6;
End

end
```

```
function xdot=trussdot(x,u,v)
% State equation.
persistent gamma
if nargin==1,
  gamma = x(4);
else
  xdot = [-x(2)+v(1);
          -exp(-gamma*x(1))*x(2)*x(2)*x(3)+v(2);
          v(3)];
End

end
```

```
function [X] = state_eq(X,RAD,void)

if nargin==1
    t = size(X);
for counter = 1:t(2)
    X(counter) = X(counter);
end;
else
    t = size(X);
for counter = 1:t(2)
    X(counter) = X(counter)*sin(RAD(counter)); % X einai i dynami,
RAD o thoryvos
end;
end;

end
```

Υπολογισμός εξόδων, truss2.m:

```
function y=truss2(x,w)

% Output equation.

persistent M H
if nargin==1,
  M = x(2);
  H = x(3);
else
  tmp = x(1)-H;
  y = sqrt(M*M+tmp*tmp)+w;
end

end

function [Y] = output_eq(X,RAD)

counter = 1;
t = size(X);
if nargin == 1
```

```
    for counter = 1:t(2)+1

if counter == 1
    Y(counter) = X(counter);
end;

if counter == t(2)+1
    Y(counter) = X(counter-1);
end;

if ( (counter >= 2 ) && (counter <= t(2)-1) )
    temp1 = X(counter);
    temp2 = sin( (pi/2)-(X(counter)) );
    temp3 = X(counter+1)*sin( X(counter+1) );
    temp4 = sin( (pi/2) - X(counter) );
    Y(counter)=(temp1*temp2)+(temp3*temp4);
end;

end;
else
for counter = 1:t(2)+1

if counter == 1
    Y(counter) = X(counter)*sin(RAD(counter));
end;

if counter == t(2)+1
    Y(counter) = X(counter-1)*sin(RAD(counter-1));
end;

if ( (counter >= 2 ) && (counter <= t(2)-1) )
    temp1 = X(counter)*sin(RAD(counter));
    temp2 = sin( (pi/2)-(RAD(counter)) );
    temp3 = X(counter+1)*sin( RAD(counter+1) );
    temp4 = sin( (pi/2) - RAD(counter) );
    Y(counter)=(temp1*temp2)+(temp3*temp4);
end;

end;
end;
end
```

Κεφάλαιο

4

Αποτελέσματα πειραμάτων.

4.1 Εισαγωγή.

Στα προηγούμενα κεφάλαια μελετήσαμε το φίλτρο kalman και την εφαρμογή του στο πρόβλημα της γέφυρας που περιγράψαμε. Σε αυτό το κεφάλαιο παρουσιάζουμε τα αποτελέσματα των πειραμάτων που διεξάγαμε με την εφαρμογή που υλοποιήσαμε.

Αφού με ανάλυση δυνάμεων ξέρουμε ακριβώς τι δεδομένα ψάχνουμε, άρα είναι εύκολο να εκτιμήσουμε το σφάλμα προσέγγισης σε κάθε εκτέλεση του αλγορίθμου και να αποφασίσουμε πότε θα σταματήσουμε. Τα αρχικά δεδομένα, που είναι το μέτρο των δυνάμεων που ασκούνται σε κάθε σημείο X.

Οι τιμές των δυνάμεων στα σημεία Y είναι αρχικά μηδέν και αλλάζουν κάθε φορά που αλλάζουν οι τιμές στα σημεία X. Κάθε φορά που αλλάζουν οι δυνάμεις στα σημεία Y, εφαρμόζεται το φίλτρο και προσπαθεί να εκτιμήσει τις δυνάμεις στα σημεία X που προκάλεσαν τις μετακινήσεις στα σημεία Y.

4.2 Αποτελέσματα πειραμάτων.

Διεξάγαμε μια σειρά πειραμάτων για να διαπιστώσουμε την λειτουργία του φίλτρου. Δοκιμάσαμε διαφορετικές τιμές στο μέγεθος του δικτυώματος καθώς και στο χρόνο διάρκειας του πειράματος.

Κάναμε επίσης μετρήσεις του σφάλματος για να δούμε πόσο γρήγορα το φίλτρο εκτιμά τιμές στα σημεία X που να είναι ίδιες με αυτές που πραγματικά υπήρχαν. Στο αρχείο data/matlab.mat υπάρχουν αποθηκευμένα τέτοια δεδομένα από ένα χαρακτηριστικο πείραμα. Αυτά τα δεδομένα «φορτώνονται» στην matlab για εξέταση και περαιτέρω επεξεργασία, με την παρακάτω εντολή:

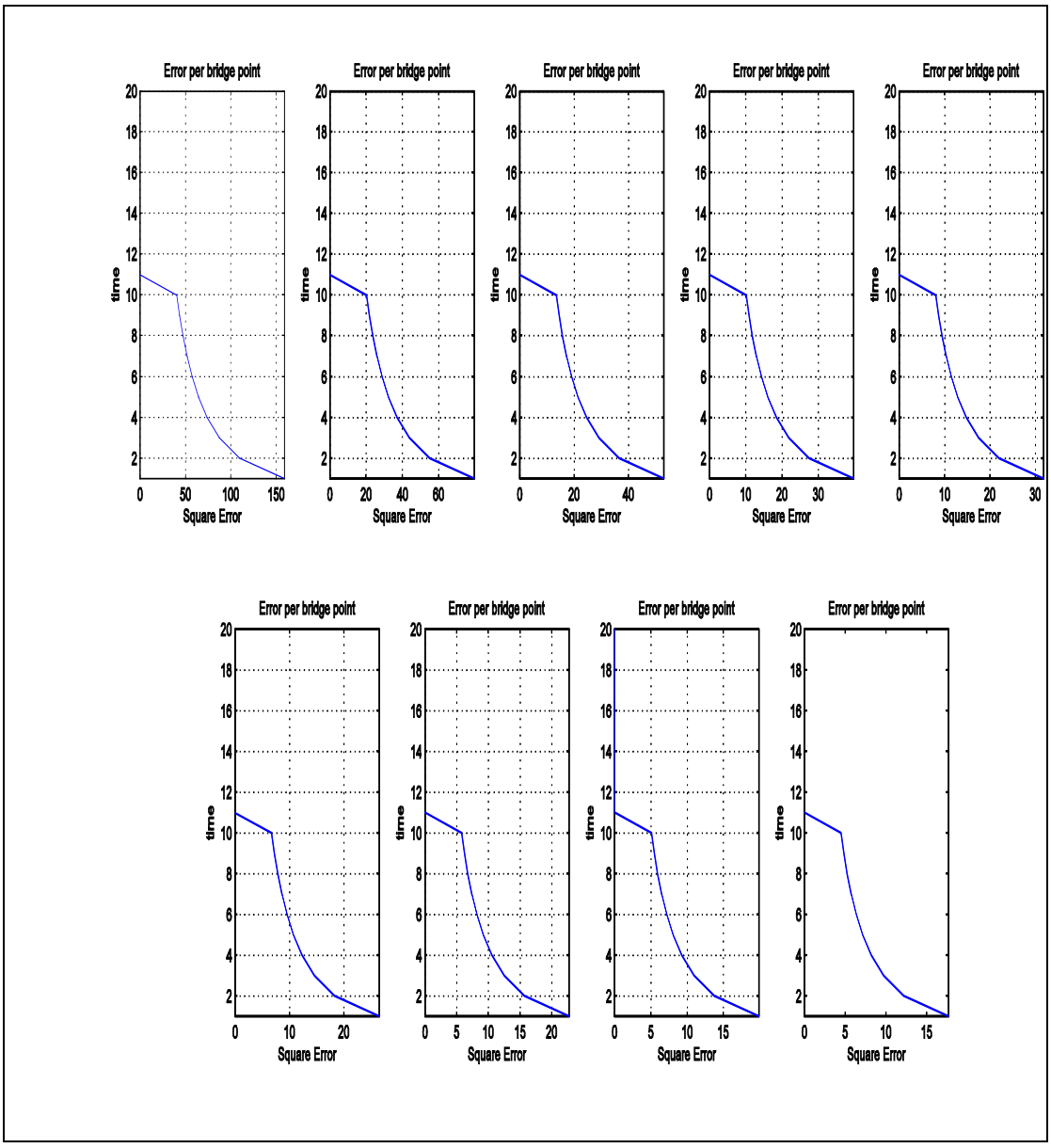
```
cd path/to/data
load matlab.mat
```

Παρακάτω μπορούμε να δούμε κάποια αποτελέσματα για τις δυνάμεις στα σημεία Χ και Υ και τις αντίστοιχες εκτιμήσεις για τα Χ.

	Σημεία της γέφυρας				
Χρόνος	Δυνάμεις στα σημεία Χ				
	3,00E+05	3,00E+05	3,00E+05	3,00E+05	3,00E+05
	2,06E+05	2,06E+05	2,06E+05	2,06E+05	2,06E+05
	2,74E+05	2,75E+05	2,74E+05	2,74E+05	2,75E+05
	2,33E+05	2,34E+05	2,33E+05	2,33E+05	2,33E+05
	2,05E+05	2,05E+05	2,05E+05	2,05E+05	2,05E+05
	1,93E+05	1,94E+05	1,93E+05	1,94E+05	1,93E+05
	1,88E+05	1,88E+05	1,88E+05	1,88E+05	1,88E+05
	1,85E+05	1,85E+05	1,85E+05	1,85E+05	1,85E+05
	1,85E+05	1,85E+05	1,85E+05	1,85E+05	1,85E+05
	1,85E+05	1,85E+05	1,85E+05	1,85E+05	1,85E+05
	1,86E+05	1,86E+05	1,86E+05	1,87E+05	1,86E+05
	1,87E+05	1,87E+05	1,87E+05	1,87E+05	1,87E+05
	1,88E+05	1,88E+05	1,88E+05	1,88E+05	1,88E+05
	1,89E+05	1,89E+05	1,89E+05	1,89E+05	1,89E+05
	1,89E+05	1,89E+05	1,89E+05	1,89E+05	1,89E+05
	1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05
	1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05
	1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05
	1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05
	1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05
	1,90E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05
	1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05

Σημεία της γέφυρας					
Y forces					Χρόνος
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,42E+05	1,42E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,42E+05	1,41E+05	1,41E+05	1,41E+05	
1,42E+05	1,41E+05	1,42E+05	1,41E+05	1,41E+05	
1,42E+05	1,42E+05	1,41E+05	1,42E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,42E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,42E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,42E+05	1,41E+05	
1,42E+05	1,42E+05	1,41E+05	1,41E+05	1,42E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,42E+05	
1,42E+05	1,41E+05	1,41E+05	1,41E+05	1,42E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,42E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,42E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,42E+05	1,42E+05	1,41E+05	
1,42E+05	1,42E+05	1,41E+05	1,41E+05	1,42E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,42E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,42E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,42E+05	1,41E+05	1,41E+05	
1,41E+05	1,41E+05	1,41E+05	1,42E+05	1,41E+05	

Σημεία της γέφυρας					
Εκτίμηση για τα Χ					Χρόνος
3,00E+05	3,00E+05	3,00E+05	3,00E+05	3,00E+05	
2,06E+05	2,06E+05	2,06E+05	2,06E+05	2,06E+05	
2,74E+05	2,75E+05	2,74E+05	2,74E+05	2,75E+05	
2,33E+05	2,34E+05	2,33E+05	2,33E+05	2,33E+05	
2,05E+05	2,05E+05	2,05E+05	2,05E+05	2,05E+05	
1,93E+05	1,94E+05	1,93E+05	1,94E+05	1,93E+05	
1,88E+05	1,88E+05	1,88E+05	1,88E+05	1,88E+05	
1,85E+05	1,85E+05	1,85E+05	1,85E+05	1,85E+05	
1,85E+05	1,85E+05	1,85E+05	1,85E+05	1,85E+05	
1,85E+05	1,85E+05	1,85E+05	1,85E+05	1,85E+05	
1,86E+05	1,86E+05	1,86E+05	1,87E+05	1,86E+05	
1,87E+05	1,87E+05	1,87E+05	1,87E+05	1,87E+05	
1,88E+05	1,88E+05	1,88E+05	1,88E+05	1,88E+05	
1,89E+05	1,89E+05	1,89E+05	1,89E+05	1,89E+05	
1,89E+05	1,89E+05	1,89E+05	1,89E+05	1,89E+05	
1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05	
1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05	
1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05	
1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05	
1,90E+05	1,90E+05	1,90E+05	1,90E+05	1,90E+05	
1,90E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	
1,91E+05	1,90E+05	1,90E+05	1,91E+05	1,91E+05	



Σχήμα 10. Σύγκλιση στο χρόνο ανά σημείο γέφυρας.

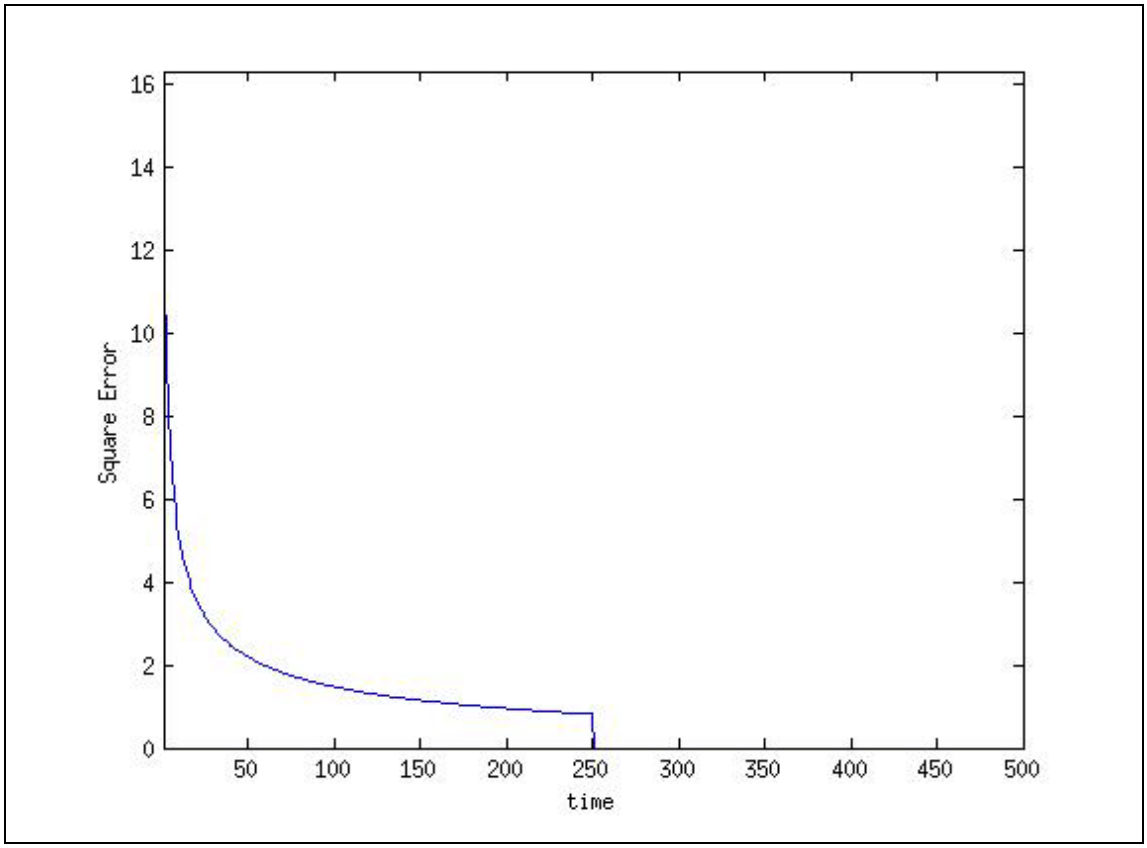
Όπως μπορούμε να δούμε, το σφάλμα μειώνεται εκθετικά, δηλαδή μη-γραμμικά, ώς προς το χρόνο το οποίο είναι μια πολύ ικανοποιητική συμπεριφορά κυρίως αν σκεφτούμε τις φυσικές προεκτάσεις αυτού του γεγονότος.

Φανταστείτε από αυτόν τον αλγόριθμο να εξαρτάται το σύστημα πλοήγησης ενός αεροπλάνου! Επίσης φαίνεται και η ικανότητα του φίλτρου να βελτιώνει τις εκτιμήσεις του όσο συγκεντρώνει παρατηρήσεις.

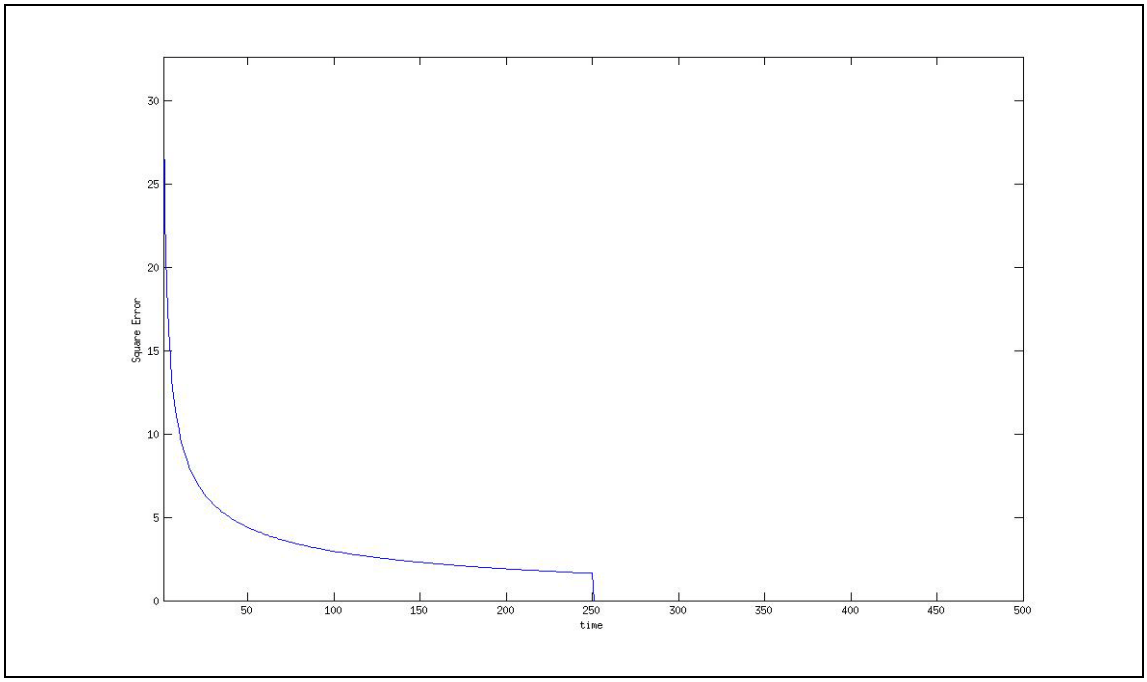
Παρατηρήσαμε ότι η σύγκλιση ήταν αρκετά γρήγορη όσο και αν αυξάνονταν το μέγεθος του προβλήματος. Επίσης, παρατηρήσαμε ότι η σύγκλιση γινόταν

απόλυτη, δηλαδή το σφάλμα μηδενίζονταν περίπου στα μέσα του χρονικού διαστήματος μελέτης.

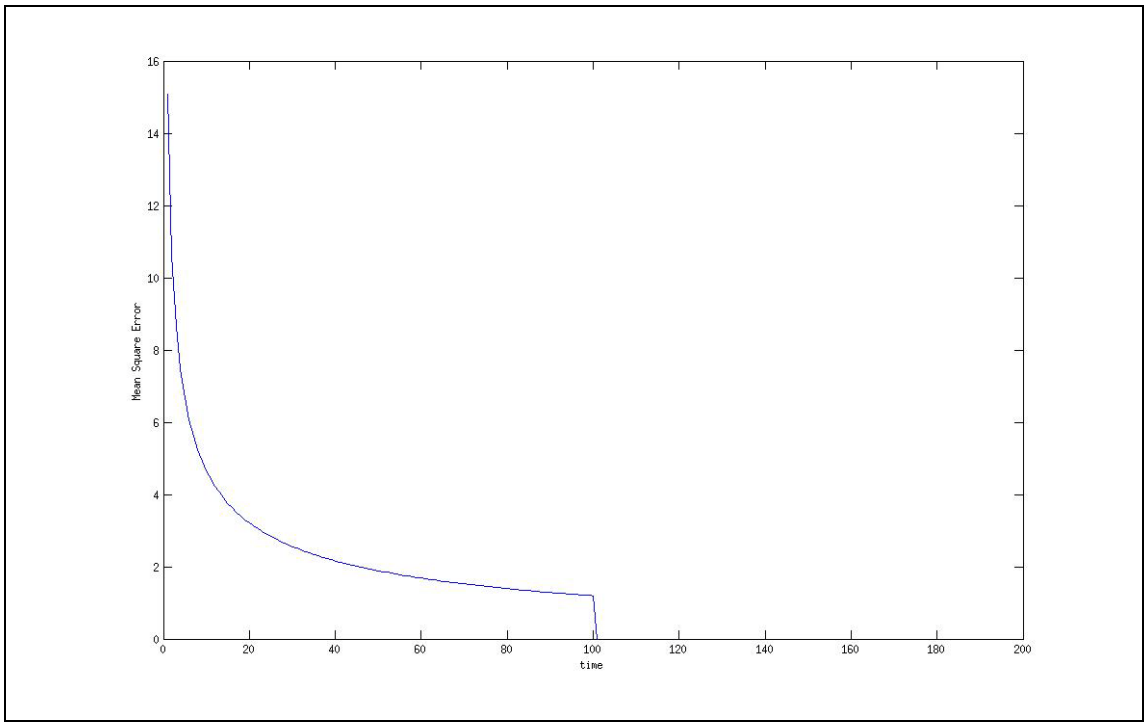
Στα παρακάτω σχήματα, μπορούμε να δούμε αντίστοιχα σχήματα από ενδεικτικούς κόμβους της γέφυρας καθώς και την σύγκλιση του μέσου σφάλματος.



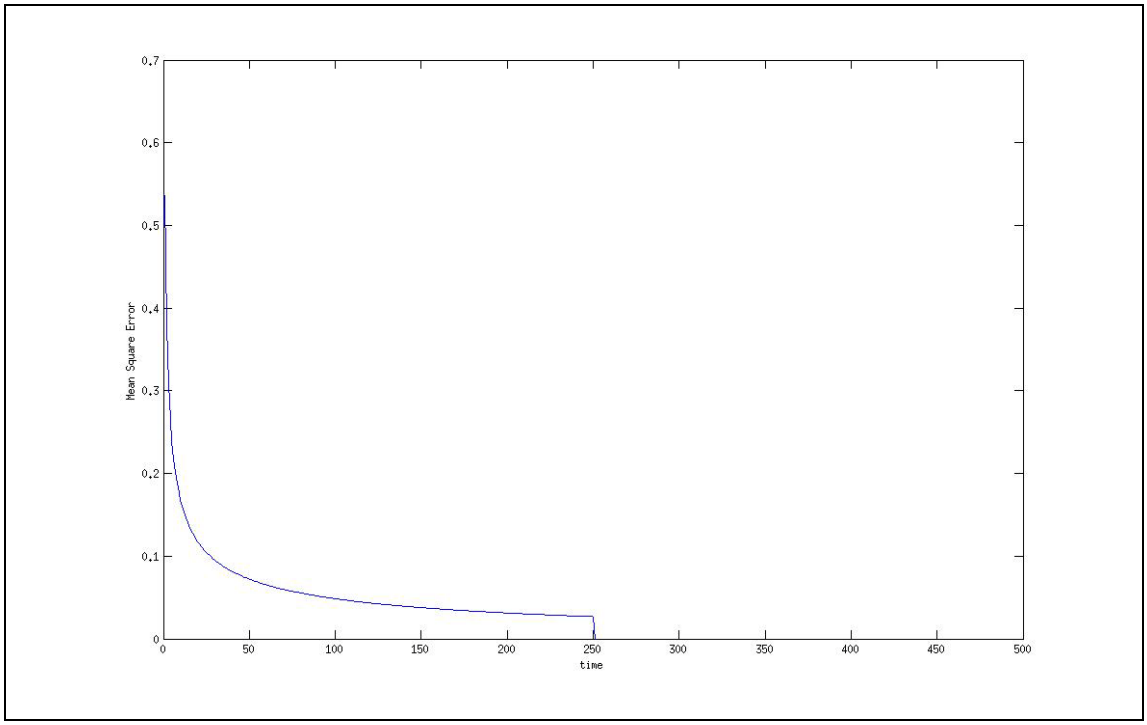
Σχήμα 11. Σύγκλιση στο χρόνο ενός σημείου της γέφυρας.



Σχήμα 12. Σύγκλιση στο χρόνο ενός σημείου της γέφυρας.



Σχήμα 13. Σύγκλιση στο χρόνο του μέσου σφάλματος.



Σχήμα 14. Σύγκλιση στο χρόνο του μέσου σφάλματος

Βιβλιογραφικές πηγές

[1] Wikipedia, επίσημη ιστοσελίδα: http://en.wikipedia.org/wiki/

[2] Mathworks support center, http://www.mathworks.com/support/

[3] Gilbert Strang, Γραμμική άλγεβρα και εφαρμογές, Πανεπιστημιακές Εκδόσεις Κρήτης, 2003.

[4] Ασημάκης Νικόλαος, Φίλτρα Kalman και Λαϊνιώτη, Πανεπιστημιακές εκδόσεις Αράκυνθος, 2009.

[5] Hugh Young, Πανεπιστημιακή Φυσική, 2002.

[6] JannN.Yang, Hongwei Huang, "Sequential non-linear, least-square, estimation for damage identification of structures with unknown inputs and unknown outputs", 2007.

[7] KalmTool, MagnusNørgaard, Department of Mathematical Modelling & Department of Automation, Technical University of Denmark, December17, 2002.

[8] G.Bolzon, R.Fedele, G.Maier, Parameter identification of a cohesive crack model b y Kalman filtering, Department of Structural Engineering, Technical University of Milan, January 2002.

[9] Mehrdad Negahban, The analysis of trusses, University of Nebraska, 2006, http://emweb.unl.edu/negahban/em223/note12/note12.htm .

[10] Matlab central, http://www.mathworks.com/matlabcentral/.

[11] Shahnam Navaee, Nirmal K. Das, Utilization of MATLAB in Structural Analysis, Proceedings of the 2002 ASEE/SEFI/TUB Colloquium, 2002.

[12] Stork, C., 1992a, Singular value decomposition of the velocity-reflector depth tradeoff, part1: Introduction using a two-parameter model, Geophysics, 57, 927-932.

[13] Lines, L. R., 1993, Ambiguity in analysis of velocity and depth: Geophysics, 58, 596-597.