



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΚΕΝΤΡΟ ΕΠΙΜΟΡΦΩΣΗΣ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗΣ

Επιμορφωτικό Πρόγραμμα Επίλυση Προβλημάτων με Τεχνητή Νοημοσύνη και
Προηγμένοι Αλγόριθμοι Εκτίμησης (Problem Solving with Artificial Intelligence
and Advanced Estimation Algorithms)

ΕΡΓΑΣΙΕΣ

Εκπαιδευτική Ενότητα 3: Προηγμένοι Αλγόριθμοι Εκτίμησης

ΑΣΠΡΟΥΔΗΣ ΙΑΣΟΝΑΣ-ΧΡΙΣΤΟΦΟΡΟΣ

ΑΘΗΝΑ 2022

Contents

1. Φίλτρα Kalman	3
2. Υπολογιστικός φόρτος φίλτρων Kalman και Λαϊνιότητα	8
3. Εξίσωση Riccati και εξίσωση Lyapunov	9
4. Μόνιμη κατάσταση: Υπολογισμός παραμέτρων φίλτρου Kalman μόνιμης κατάστασης.....	13
5. Εκτίμηση θέσης κινούμενου σώματος με σταθερή ταχύτητα	14
6. Εκτίμηση θέσης κινούμενου σώματος με σταθερή επιτάχυνση	19
7. Τυχαίος περίπατος	23
8. Πρόβλεψη κρουσμάτων Covid-19 με φίλτρο Kalman	26
9-10. Εργασία – Εκτίμηση τυχαίας σταθεράς με χρονικά αμετάβλητο φίλτρο Kalman.....	28

1. Φίλτρα Kalman

A) Βιβλιογραφία

Να αναζητήσετε στο διαδίκτυο (Internet) και να εντοπίσετε το βιβλίο B.D.O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, 1979.

<http://users.cecs.anu.edu.au/~john/papers/BOOK/B02.PDF>

B.D.O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, 1979.

B) Τύποι του φίλτρου Kalman

Να αναζητήσετε στο διαδίκτυο (Internet) τύπους του φίλτρου Kalman.

Πόσους βρήκατε;

Να τους καταγράψετε.

https://en.wikipedia.org/wiki/Kalman_filter

- Square root form
- Parallel form
- Information filter
- Frequency-weighted Kalman filters
- Nonlinear filters
- Extended Kalman filter
- Unscented Kalman filter
- Discriminative Kalman filter
- Adaptive Kalman filter
- Kalman–Bucy filter

Εκπαιδευτικό υλικό

- Square Root Kalman Filter
- UDU Kalman Filter
- Information Kalman Filter
- UDU Information Kalman Filter
- Kalman Filter using General Chandrasekhar
- Information Kalman Filter using General Chandrasekhar Algorithm
- Sigma-Rho Kalman Filter
- Schmidt-Kalman Filter ή Consider Kalman Filter
- Steady State Kalman Filter
- FIR Steady State Kalman Filter
- Periodic Steady State Kalman Filter
- Implicit Kalman Filter

- Square Root Implicit Kalman Filter
- Complex Kalman Filter
- Interval Kalman Filter
- Fuzzy Kalman Filter
- Monte Carlo Kalman Filter
- Maximum Correntropy Kalman Filter

Γ) Εφαρμογές των φίλτρων Kalman

Να αναζητήσετε στο διαδίκτυο (Internet) εφαρμογές των φίλτρων Kalman.

https://en.wikipedia.org/wiki/Kalman_filter

- Attitude and heading reference systems
- Autopilot
- Electric battery state of charge (SoC) estimation
- Brain–computer interfaces
- Chaotic signals
- Tracking and vertex fitting of charged particles in particle detectors
- Tracking of objects in computer vision
- Dynamic positioning in shipping
- Economics, in particular macroeconomics, time series analysis, and econometrics
- Inertial guidance system
- Nuclear medicine – single photon emission computed tomography image restoration
- Orbit determination
- Power system state estimation
- Radar tracker
- Satellite navigation systems
- Seismology
- Sensorless control of AC motor variable-frequency drives
- Simultaneous localization and mapping
- Speech enhancement
- Visual odometry
- Weather forecasting
- Navigation system
- 3D modeling
- Structural health monitoring
- Human sensorimotor processing

Εκπαιδευτικό υλικό

- στην αεροναυπηγική
- στον έλεγχο χημικών διεργασιών

- στη σχεδίαση τηλεπικοινωνιακών συστημάτων
- στην πρόβλεψη ρύπανσης
- σε ενεργειακά συστήματα
- στην παρακολούθηση στόχου
- στην επεξεργασία εικόνας
- στην επεξεργασία ήχου
- σε GPS
- στην οικονομετρία
- στην παρακολούθηση δορυφόρων
- στην πλοήγηση πλοίων
- στην πρόβλεψη θερμοκρασίας
- στην εκτίμηση επιπέδου πληρότητας δεξαμενών
- στην εκτίμηση ηλεκτρικού φορτίου
- στην εκτίμηση παραγωγής ενέργειας

4) Βιογραφικά στοιχεία του R. Kalman

Να αναζητήσετε στο διαδίκτυο (Internet) βιογραφικά στοιχεία του R. Kalman.

https://www.researchgate.net/publication/317581277_Rudolf_E_Kalman_-_O_themeliotes_tes_Monternas_Theorias_Elenchou

Rudolf E. Kalman

Ο θεμελιωτής της Μοντέρνας Θεωρίας Ελέγχου

Ο Rudolf Emil Kalman (1930-2016) υπήρξε ένας από τους επιδραστικότερους επιστήμονες και μηχανικούς του περασμένου αιώνα, και ο θεμελιωτής της μοντέρνας θεωρίας συστημάτων ελέγχου. Η σύγχρονη θεωρία συστημάτων ελέγχου είναι ένας ευρύς κλάδος της μηχανικής και των μαθηματικών που ασχολείται με δυναμικά συστήματα που περιγράφονται από διαφορικές εξισώσεις ή εξισώσεις διαφορών, τη μελέτη της συμπεριφοράς τους (ανάλυση) αλλά και τη χειραγωγή των ελέγχιμων παραμέτρων ή εισόδων ενός συστήματος με σκοπό την επίτευξη της επιθυμητής συμπεριφοράς του (σύνθεση). Κατά τη διάρκεια της μακρόχρονης και δραστήριας επιστημονικής του καριέρας, ο Kalman κατάφερε όχι μόνο να θέσει τα θεμέλια αυτού του κλάδου, αλλά και να κατευθύνει με τις ιδέες του την ανάπτυξή του για πολλές δεκαετίες [1,2,7,16]. Ο R.E. Kalman γεννήθηκε στη Βουδαπέστη στις 19 Μαΐου το 1939 και μετακόμισε στην Αμερική στην ηλικία των δεκατριών χρόνων. Έλαβε το πτυχίο και το μεταπτυχιακό του από το Τεχνολογικό Ινστιτούτο της Μασαχουσέτης (MIT) το 1953 και 1954 και το διδακτορικό του δίπλωμα από το Πανεπιστήμιο της Κολούμπια το 1957. Οι επόμενες θέσεις που κατείχε ήταν αυτές του ερευνητή στο RIAS (Research Center for Advanced Study) στη Βαλτιμόρη (1958-1964), του καθηγητή στο Πανεπιστήμιο του Στάνφορντ (1964-1971), του ερευνητή καθηγητή στο Center for Mathematical System Theory στο Πανεπιστήμιο της Φλόριντα (1971-1992) και ταυτόχρονα του προέδρου της Mathematical System Theory στο Swiss Federal Institute of Technology στη Ζυρίχη (1973-1997).

Από τις πιο σημαντικές συνεισφορές του Kalman υπήρξε η ανάπτυξη του φίλτρου Kalman[11-13]. Το φίλτρο Kalman είναι ένας αλγόριθμος που εφαρμόζεται σε συστήματα που διέπονται από

εξωτερικές διαταραχές (θορύβους), με σκοπό τον «καθαρισμό» των μετρήσεων που γίνονται και την απόκτηση μιας νέας εκτίμησης της κατάστασης του συστήματος, απογυμνωμένη από διαταραχές. Το φίλτρο αρχικά αναπτύχθηκε σε διακριτό χρόνο και η επέκτασή του στον συνεχή έγινε αργότερα σε συνεργασία με τον R. Bucy[12]. Η ανάπτυξη αυτής της τεχνικής για την εκτίμηση των καταστάσεων ενός συστήματος που υπόκειται σε διαταραχές μπορεί να θεωρηθεί χωρίς υπερβολή μια από τις σημαντικότερες θεωρητικές και τεχνολογικές ανακαλύψεις του προηγούμενου αιώνα. Αυτό γίνεται κατανοητό αν αναλογιστεί κανείς πως στις περισσότερες μηχανικές και τεχνολογικές εφαρμογές οι μετρήσεις είναι θορυβώδεις και υπάρχει συνεχώς η ανάγκη για την απόκτηση καλύτερων εκτιμήσεων, ώστε να εξασφαλίζεται η ορθή λειτουργία τους. Το φίλτρο Kalman επομένως βρίσκει εφαρμογές στην οικονομετρία, στη σεισμολογία, στην μετεωρολογία, στη μηχανική όραση (computer vision), στον έλεγχο στατικής επάρκειας (structural health monitoring), στον έλεγχο κινητήρων, σε όργανα GPS, σε προγράμματα αυτόματης πλοήγησης, στην πλοήγηση δορυφόρων, σε ραντάρ, σε ατομικά ρολόγια (atomic clocks) και σε αναρίθμητες ακόμη εφαρμογές[2,4-6,15]. Ξεχωριστή σημασία ανάμεσα στο πλήθος των παραπάνω εφαρμογών, έχει η ιστορική πρώτη αποστολή ανθρώπων στο φεγγάρι το 1969 με το Apollo 11, στην επιτυχία της οποίας το φίλτρο Kalman έπαιξε καίριο ρόλο, συμβάλλοντας στον ακριβή υπολογισμό των τροχιών του επανδρωμένου αεροσκάφους. Η αρχή της συνεργασίας αυτής ξεκίνησε όταν S. F. Schmidt, ο οποίος ήταν διευθυντής στο Τμήμα Δυναμικής Ανάλυσης της NASA στο Ames Research Center στην Καλιφόρνια προσκάλεσε τον Kalman να τους παρουσιάσει τις ιδέες του, και πίστεψε πως η μεθοδολογία του Kalman θα μπορούσε να εφαρμοστεί στη βελτίωση του προγράμματος πλοήγησης του υπό κατασκευή διαστημικού σκάφους.

Πέρα από το ομώνυμο φίλτρο, ο R. Kalman συνέβαλε με την έρευνα του σε πάρα πολλά πεδία των μαθηματικών, παρουσιάζοντας συνεχώς νέες ιδέες, αλλά και θέτοντας νέα ερωτήματα των οποίων η απάντηση οδηγούσε σε νέες καινοτομίες[3,7,8,10]. Η ανάπτυξη της θεωρίας των συστημάτων του χώρου των καταστάσεων (state space systems) και των εννοιών της ελεγχιμότητας (controllability) και παρατηρησιμότητας (observability)[14] άλλαξε ριζικά τη μελέτη των δυναμικών συστημάτων και έθεσε τις βάσεις για τη μελέτη νέων μεθόδων σύνθεσης ελεγκτών (LQR), την απόρριψη διαταραχών (disturbance rejection) και τη δημιουργία παρατηρητών κατάστασης (state observers). Σημαντική ήταν επίσης η συμβολή του στο πρόβλημα της πραγμάτωσης (realization) και της αναγνώρισης συστημάτων (system identification), που μελετάει την κατασκευή της κατάστασης ενός συστήματος χρησιμοποιώντας μετρήσεις της εισόδου και της εξόδου[3,9,15].

Για τη συνολική του συμβολή στην επιστήμη, ο R. Kalman τιμήθηκε κατά τη διάρκεια της ζωής του με τα IEEE Medal of Honor (1974), το IEEE Centennial Medal (1984), το Rufus Oldenburger Medal (1976), το Kyoto Prize in Advanced Technology (1985), το Steele Prize της American Mathematical Society (1987), το Richard Bellman Control Heritage Award (1997), το Charles Stark Draper Prize for Engineering (2008) και το National Medal of Science (2008) των Η.Π.Α.

Ο R. Kalman έφυγε από τη ζωή στις 2 Ιουλίου 2016, σε ηλικία 86 χρονών.

Αναφορές:

- [1] Antoulas, A., Georgiou, T. T., Khargonekar, P. P., Ozguler, A. B., Sontag, E. D., & Yamamoto, Y. (2017). A Tribute to Rudolf Kalman: His Research, Life, and Influence [Historical Perspectives]. *IEEE Control Systems*, 37(2), 153-153.
- [2] Antoulas, A., Georgiou, T. T., Khargonekar, P. P., Ozguler, A. B., Sontag, E. D., & Yamamoto, Y. (2017). Prof. Rudolf Emil Kalman [Obituary]. *IEEE Control Systems*, 37(1), 151-152.
- [3] Antoulas, A. (Ed.). (1991). *Mathematical system theory: the influence of RE Kalman*. Springer Science & Business Media.
- [4] Galleani, L., & Tavella, P. (2010). Time and the Kalman filter. *IEEE Control Systems*, 30(2), 44-65.
- [5] Gelb, A. (1974). *Applied optimal estimation*. MIT press.
- [6] Grewal, M. S., & Andrews, A. P. (2010). Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems*, 30(3), 69-78.
- [7] Isidori, A. (2017). Kalman: The Scientist Who Defined Our Field [Historical Perspectives]. *IEEE Control Systems*, 37(2), 166-168.
- [8] Kalman, R. (2010). Old and new directions of research in system theory. In *Perspectives in Mathematical System Theory, Control, and Signal Processing* (pp. 3-13). Springer Berlin Heidelberg.
- [9] Kalman, R. E. (1982). Identification from real data. In *Current developments in the interface: Economics, Econometrics, Mathematics* (pp. 161-196). Springer Netherlands.
- [10] Kalman, R. (2003). Discovery and invention: The newtonian revolution in systems technology. *Journal of guidance, control, and dynamics*, 26(6), 833-837.
- [11] Kalman, R. E. (1963). Mathematical description of linear dynamical systems. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 1(2), 152-192.
- [12] Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(3), 95-108.
- [13] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1), 35-45. 7
- [14] Kalman, R. (1959). On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3), 110-110.
- [15] Pincus, S., & Kalman, R. E. (2004). Irregularity, volatility, risk, and financial market time series. *Proceedings of the National Academy of Sciences of the United States of America*, 101(38), 13709-13714.
- [16] Sontag, E. (2010). Rudolf E. Kalman and His Students [Historical Perspectives]. *IEEE Control Systems*, 30(2), 87-88.

2. Υπολογιστικός φόρτος φίλτρων Kalman και Λαϊνιώτη

Να μελετήσετε την εργασία:

Assimakis N., Adam M., “Discrete Time Kalman and Lainiotis Filters Comparison”, Int. Journal of Mathematical Analysis, vol. 1, no. 13, pp. 635-659, 2007.

A) Να υπολογίσετε τον υπολογιστικό φόρτο του φίλτρου Kalman μόνιμης κατάστασης, συμπληρώνοντας τον πίνακα:

Φίλτρο Kalman μόνιμης κατάστασης χωρίς είσοδο Υπολογιστικός φόρτος ανά επανάληψη		
Πράξη Πινάκων	Διαστάσεις Πινάκων	Υπολογιστικός Φόρτος
$Ax(k k)$	$(n \times n) \cdot (n \times 1)$	$2n^2 - n$
$Kz(k+1)$	$(n \times m) \cdot (m \times 1)$	$2nm - n$
$x(k+1 k+1) = Ax(k k) + Kz(k+1)$	$(n \times 1) + (n \times 1)$	n
Σύνολο		

B) Να υπολογίσετε ποιο φίλτρο είναι ταχύτερο: το φίλτρο Kalman ή το φίλτρο Λαϊνιώτη;
Να συμπληρώσετε τον πίνακα:

μοντέλο	πλήθος καταστάσεων n	πλήθος μετρήσεων m	ταχύτερο φίλτρο Kalman / Λαϊνιώτη
χρονικά μεταβαλλόμενο	4	4	Φίλτρο Kalman
	10	4	Φίλτρο Kalman
	4	10	Φίλτρο Kalman
	10	10	Φίλτρο Kalman
χρονικά αμετάβλητο	4	4	Φίλτρο Λαϊνιώτη
	10	4	Φίλτρο Kalman
	4	10	Φίλτρο Λαϊνιώτη
	10	10	Φίλτρο Λαϊνιώτη
χρονικά αμετάβλητο μόνιμη κατάσταση	4	4	Φίλτρο Kalman Φίλτρο Λαϊνιώτη
	10	4	Φίλτρο Kalman Φίλτρο Λαϊνιώτη
	4	10	Φίλτρο Kalman Φίλτρο Λαϊνιώτη
	10	10	Φίλτρο Kalman Φίλτρο Λαϊνιώτη

3. Εξίσωση Riccati και εξίσωση Lyapunov

Α) Επαναληπτικοί αλγόριθμοι για την επίλυση της εξίσωσης *Riccati*

Ανά βήμα Αλγόριθμος – Per step Algorithm

$$P(k+1|k) = Q + FP(k|k-1)F^T - FP(k|k-1)H^T[HP(k|k-1)H^T + R]^{-1}HP(k|k-1)F^T$$

για $k=0,1,\dots$

με αρχική συνθήκη $P(0| -1)$

μέχρι $||P(k+1|k) - P(k|k-1)|| < \varepsilon$

$$P_p = \lim_{k \rightarrow \infty} P(k+1|k)$$

Αλγόριθμος Διπλασιασμού – Doubling Algorithm

$$a(k+1) = a(k)[I + b(k)c(k)]^{-1}a(k)$$

$$b(k+1) = b(k) + a(k)[I + b(k)c(k)]^{-1}b(k)a^T(k)$$

για $k=0,1,\dots$

$$c(k+1) = c(k) + a^T(k)c(k)[I + b(k)c(k)]^{-1}a(k)$$

με αρχικές συνθήκες

$$a(0) = F^T$$

$$b(0) = H^T R^{-1} H$$

$$c(0) = Q$$

μέχρι $||c(k+1) - c(k)|| < \varepsilon$

$$P_p = \lim_{k \rightarrow \infty} c(k)$$

Να γράψετε πρόγραμμα για τον υπολογισμό της λύσης της εξίσωσης Riccati με παραμέτρους

$F=0.9$, $H=1.2$, $Q=2$, $R=4$ χρησιμοποιώντας

- τον ανά βήμα αλγόριθμο
- τον αλγόριθμο διπλασιασμού

Ανά βήμα αλγόριθμος

```

1 % Riccati Equation Kalman Filter
2 % Per Step Algorithm
3
4 function [ppss, iterations] = repsa(f, h, q, r, epsilon)
5     k=0;
6     p=q;
7     k=k+1;
8     p1=q+f*p*f'-f*p*h'*inv(h*p*h'+r)*h*p*f';
9     d=p1-p;
10    while norm(d)>epsilon
11        k=k+1;
12        p1=q+f*p*f'-f*p*h'*inv(h*p*h'+r)*h*p*f';
13        d=p1-p; p=p1;
14    end
15    ppss=p;
16    iterations=k;
17 end

```

Αλγόριθμος διπλασιασμού

```

Editor - C:\Users\jasproudis\Documents\MATLAB\reda.m
repsa.m  ee3exercise3_1.m  eyemovement.m  reda.m  +
1  function [ppss, iterations]=reda(f,h,q,r,epsilon)
2  % Riccati Equation Kalman Filter
3  % Doubling Algorithm
4  n=size(q,1);
5  mon=eye(n);
6  k=1;
7  a=f';
8  b=h'*inv(r)*h;
9  c=q;
10     k=k+1;
11     a1=a*inv(mon+b*c)*a;
12     b1=b+a*inv(mon+b*c)*b*a';
13     c1=c+a'*c*inv(mon+b*c)*a;
14     d=c1-c;
15     a=a1;
16     b=b1;
17     c=c1;
18     while norm(d)>epsilon
19         k=k+1;
20         a1=a*inv(mon+b*c)*a;
21         b1=b+a*inv(mon+b*c)*b*a';
22         c1=c+a'*c*inv(mon+b*c)*a;
23         d=c1-c;
24         a=a1;
25         b=b1;
26         c=c1;
27     end
28     ppss=c;
29     iterations=k;

```

Να συμπληρώσετε τον πίνακα

Αλγόριθμος	Κριτήριο σύγκλισης	Επαναλήψεις	Λύση
Ανά βήμα αλγόριθμος	10^{-2}	5	3.203924457198510
	10^{-3}	7	3.205360410662081
	10^{-4}	8	3.205397685436089
	10^{-5}	9	3.205404193237079
	10^{-6}	11	3.205405527790788
Αλγόριθμος διπλασιασμού	10^{-2}	4	3.205397685436090
	10^{-3}	5	3.205405569741588
	10^{-4}	5	3.205405569741588
	10^{-5}	6	3.205405569748394
	10^{-6}	6	3.205405569748394

B) Επαναληπτικοί αλγόριθμοι για την επίλυση της εξίσωσης Lyapunov

Ανά βήμα Αλγόριθμος – Per step Algorithm

$$P(k+1|k) = Q + FP(k|k-1)F^T \quad \text{για } k = 0, 1, \dots$$

με αρχική συνθήκη $P(0|-1)$

μέχρι $||P(k+1|k) - P(k|k-1)|| < \varepsilon$

$$P_p = \lim_{k \rightarrow \infty} P(k+1|k)$$

Αλγόριθμος Διπλασιασμού – Doubling Algorithm

$$a(k+1) = a(k)a(k)$$

$$c(k+1) = c(k) + a^T(k) c(k)a(k) \quad \text{για } k = 0, 1, \dots$$

Με αρχικές συνθήκες

$$a(0) = F^T$$

$$c(0) = Q$$

Μέχρι $||c(k+1) - c(k)|| < \varepsilon$

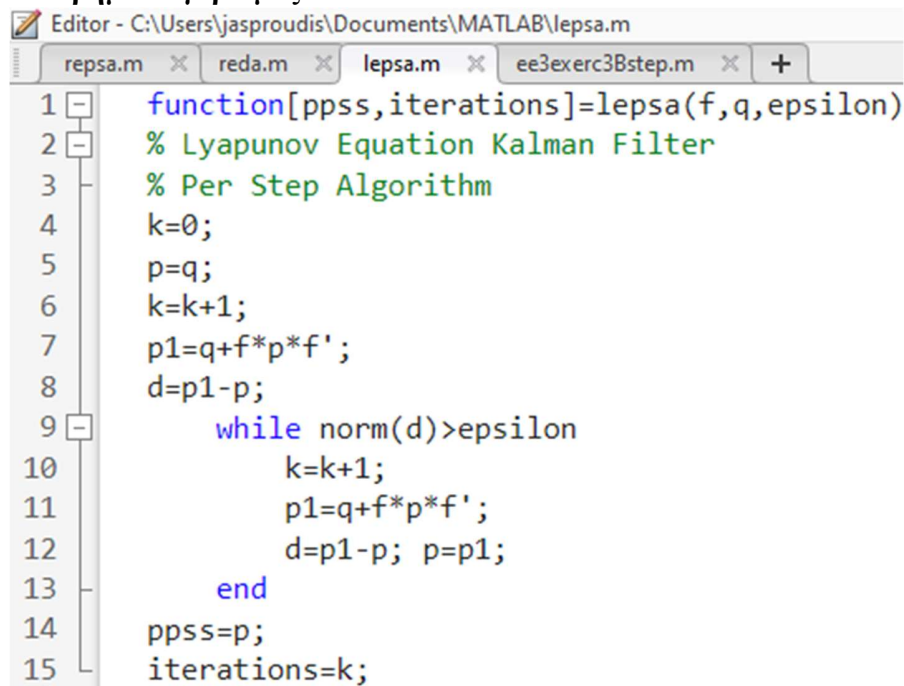
$$P_p = \lim_{k \rightarrow \infty} c(k)$$

Να γράψετε πρόγραμμα για τον υπολογισμό της λύσης της εξίσωσης Lyapunov με παραμέτρους $F=0.8$ και $Q=2.5$

χρησιμοποιώντας

- τον ανά βήμα αλγόριθμο
- τον αλγόριθμο διπλασιασμού

Ανά βήμα αλγόριθμος



```

Editor - C:\Users\jasproudis\Documents\MATLAB\lepsa.m
repsa.m x reda.m x lepsa.m x ee3exerc3Bstep.m x +
1 function [ppss, iterations] = lepsa(f, q, epsilon)
2 % Lyapunov Equation Kalman Filter
3 % Per Step Algorithm
4 k=0;
5 p=q;
6 k=k+1;
7 p1=q+f*p*f';
8 d=p1-p;
9 while norm(d)>epsilon
10     k=k+1;
11     p1=q+f*p*f';
12     d=p1-p; p=p1;
13 end
14 ppss=p;
15 iterations=k;
  
```

Αλγόριθμος διπλασιασμού

```

Editor - C:\Users\jasproudis\Documents\MATLAB\leda.m
repas.m x reda.m x lepsa.m x ee3exerc3Bstep.m x leda.m x
1 function [ppss, iterations]=leda(f,q,epsilon)
2 % Lyapunov Equation Kalman Filter
3 % Doubling Algorithm
4 k=1;
5 a=f';
6 c=q;
7 k=k+1;
8 a1=a*a;
9 c1=c+a'*c*a;
10 d=c1-c;
11 a=a1;
12 c=c1;
13 while norm(d)>epsilon
14     k=k+1;
15     a1=a*a;
16     c1=c+a'*c*a;
17     d=c1-c;
18     a=a1;
19     c=c1;
20 end
21 ppss=c;
22 iterations=k;

```

Να συμπληρώσετε τον πίνακα

Αλγόριθμος	Κριτήριο σύγκλισης	Επαναλήψεις	Λύση
Ανά βήμα αλγόριθμος	10^{-2}	14	6.931011935337617
	10^{-3}	19	6.943002139761520
	10^{-4}	24	6.944289578158345
	10^{-5}	29	6.944427815803595
	10^{-6}	35	6.944443301732948
Αλγόριθμος διπλασιασμού	10^{-2}	6	6.944440085346019
	10^{-3}	7	6.944444444441711
	10^{-4}	7	6.944444444441711
	10^{-5}	7	6.944444444441711
	10^{-6}	8	6.944444444444447

4. Μόνιμη κατάσταση: Υπολογισμός παραμέτρων φίλτρου Kalman μόνιμης κατάστασης

A) Βαθμωτό μοντέλο

Δίνεται το βαθμωτό μοντέλο με παραμέτρους $F=1, H=1, Q=1, R=1$

Να επιλύσετε την εξίσωση Riccati.

Να υπολογίσετε τις παραμέτρους του φίλτρου Kalman μόνιμης κατάστασης.

Ποιο είναι το κέρδος στη μόνιμη κατάσταση;

Ποια είναι η σχέση του κέρδος στη μόνιμη κατάσταση με την χρυσή τομή;

```
Command Window

>> p = dare(1,1,1,1)

p =

    1.618033988749895

>> goldensection=(sqrt(5)-1)/2

goldensection =

    0.618033988749895

fx >>
```

B) Πολυδιάστατο μοντέλο

Δίνεται το πολυδιάστατο μοντέλο με παραμέτρους

$$F = \begin{bmatrix} -0.9 & 0.7 \\ -0.3 & 0.1 \end{bmatrix}, H = [1 \quad 1], Q = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}, R = 1$$

Να γράψετε πρόγραμμα για τον υπολογισμό της λύσης της εξίσωσης Riccati, χρησιμοποιώντας τον ανά βήμα αλγόριθμο, τον αλγόριθμο διπλασιασμού, την συνάρτηση dare.

Να γράψετε πρόγραμμα για τον υπολογισμό των παραμέτρων του φίλτρου Kalman μόνιμης κατάστασης.

```

Command Window

>> F=[-0.9 0.7; -0.3 0.1];
H=[1 1];
Q=[1 0; 0 5];
R=1;
p=dare(F',H',Q,R)

p =

    6.884628729717408    1.483759330214887
    1.483759330214887    5.379368821468075

fx >>

Command Window

>> gain=p*(H'*H*p+R)^-1*H'*p

gain =

    0.515564162388224
    0.422827298584202

>> a=(eye(2)-gain*H)*F

a =

   -0.281323005134131    0.287548670089421
    0.207392758301043   -0.238261838867362

```

5. Εκτίμηση θέσης κινούμενου σώματος με σταθερή ταχύτητα

εκτίμηση της θέσης και της ταχύτητας
ενός αντικειμένου που κινείται στο επίπεδο με σταθερή ταχύτητα

Η θέση προσδιορίζεται από την απόσταση που διανύει το αντικείμενο (p_x, p_y)

Η ταχύτητα προσδιορίζεται από τις συνιστώσες της στους δύο άξονες (v_x, v_y)

Κατάσταση:

$$x(k) = \begin{bmatrix} p_x(k) \\ p_y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix}$$

Μετρήσεις:

$$z(k) = x(k) + v(k)$$

παράμετροι μοντέλου

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = 0.3^2 \begin{bmatrix} \frac{1}{4}(\Delta t)^4 & 0 & 0 & 0 \\ 0 & \frac{1}{4}(\Delta t)^4 & 0 & 0 \\ 0 & 0 & \frac{1}{4}(\Delta t)^4 & 0 \\ 0 & 0 & 0 & \frac{1}{4}(\Delta t)^4 \end{bmatrix}$$

$$R = \begin{bmatrix} 3^2 & 0 & 0 & 0 \\ 0 & 3^2 & 0 & 0 \\ 0 & 0 & 0.03^2 & 0 \\ 0 & 0 & 0 & 0.03^2 \end{bmatrix}$$

αρχικές συνθήκες:

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 4^2 & 0 & 0 & 0 \\ 0 & 4^2 & 0 & 0 \\ 0 & 0 & 0.4^2 & 0 \\ 0 & 0 & 0 & 0.4^2 \end{bmatrix}$$

Παράδειγμα προσομοίωσης (simulation)

κίνηση με σταθερή ταχύτητα στο επίπεδο: $v_x=5m/s$, $v_y=6m/s$

Οι μετρήσεις των θέσεων είναι οι θέσεις με θόρυβο με τυπική απόκλιση 3

Οι μετρήσεις των ταχυτήτων είναι οι ταχύτητες με θόρυβο με τυπική απόκλιση 0.03

Η αρχική θέση και η αρχική ταχύτητα θεωρούνται μηδέν.

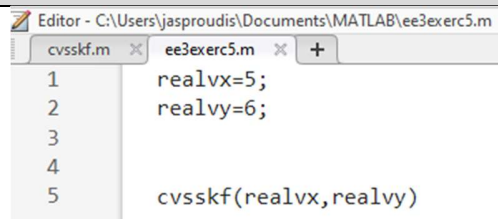
Ο κώδικας που θα χρησιμοποιηθεί με βάση αυτό το μοντέλο είναι ο παρακάτω:

```
function cvsskf(realvx,realvy)
% constant velocity
% Steady State Kalman Filter
% 2 dimensions
% model
n=4;
dt=0.1;
```

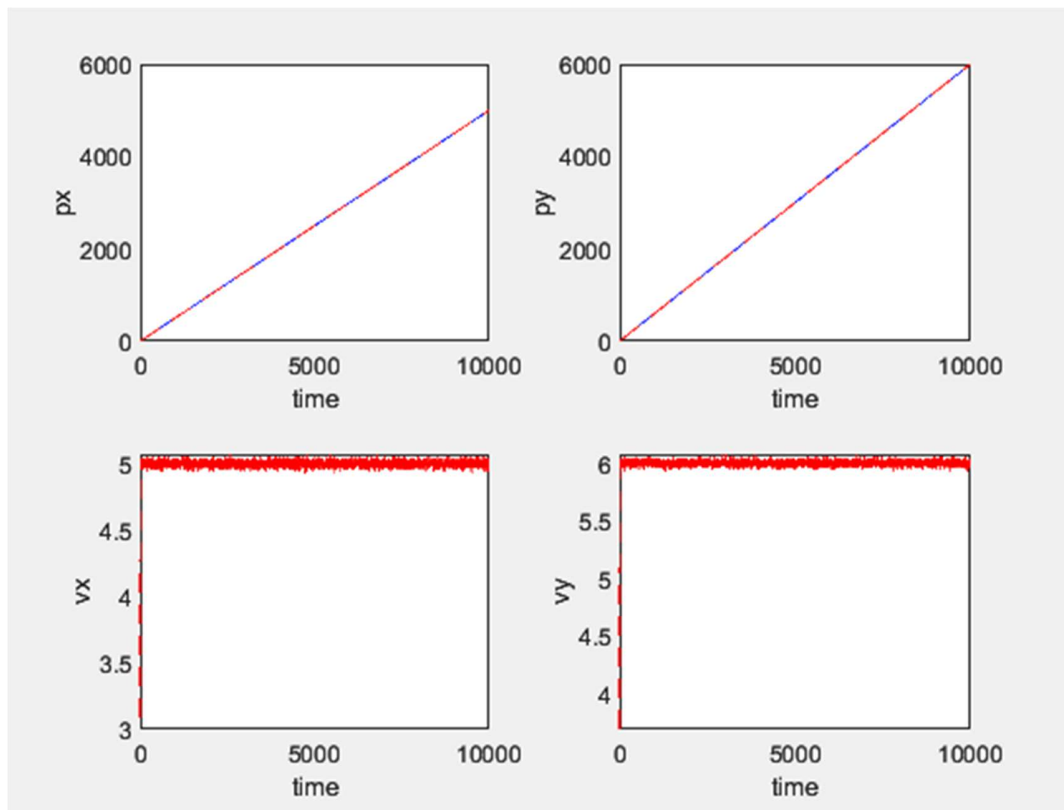
```

F=[1 0 dt 0;0 1 0 dt;0 0 1 0;0 0 0 1];
H=eye(n);
sigma1=0.3;
Q=sigma1^2*[0.25*(dt)^4 0 0 0;0 0.25*(dt)^4 0 0;0 0 (dt)^2 0 ;0 0 0 (dt)^2];
sigma2=3; sigma3=0.03;
R=[sigma2^2 0 0 0;0 sigma2^2 0 0;0 0 sigma3^2 0;0 0 0 sigma3^2];
% initial conditions
x0=[0 0 0 0]';
sigma4=4; sigma5=0.4;
pe0=[sigma4^2 0 0 0;0 sigma4^2 0 0;0 0 sigma5^2 0;0 0 0 sigma5^2];
% observability
om=obsv(H,F);
display(om);
rom=rank(om);
display(rom);
% Riccati Equation
PPss=dare(F',H',Q,R);
% Steady State Kalman Filter parameters
Gss=PPss*H'*inv(H*PPss*H'+R);
Ass=(eye(n)-Gss*H)*F;
display(Ass);
display(Gss);
% number of iterations
kmax=1000/dt;
% measurements
sigmav=0.03; sigmap=3;
% rng default;
for k=0:kmax
    vx(k+1)=realvx; zvx(k+1)=vx(k+1)+sigmav*randn;
    vy(k+1)=realvy; zvy(k+1)=vy(k+1)+sigmav*randn;
    px(k+1)=realvx*k*dt; zpx(k+1)=px(k+1)+sigmap*randn;
    py(k+1)=realvy*k*dt; zpy(k+1)=py(k+1)+sigmap*randn;
end;
X=[px; py; vx; vy];
Z=[zpx; zpy; zvx; zvy];
% Steady State Kalman Filter
xe=(eye(n)-Gss*H)*x0+Gss*Z(:,1:1);
XEESKF=xe;
for k=0:kmax-1
    xe=Ass*xe+Gss*Z(:,k+1:k+1);
    XEESKF=[XEESKF xe];
end;
% plots
figure(1);
timet=[0:kmax];
subplot(2,2,1); plot(timet,X(1:1,:), 'b',timet,XEESKF(1:1,:), 'r--'); xlabel('time'); ylabel('px');
subplot(2,2,2); plot(timet,X(2:2,:), 'b',timet,XEESKF(2:2,:), 'r--'); xlabel('time'); ylabel('py');
subplot(2,2,3); plot(timet,X(3:3,:), 'b',timet,XEESKF(3:3,:), 'r--'); xlabel('time'); ylabel('vx');
subplot(2,2,4); plot(timet,X(4:4,:), 'b',timet,XEESKF(4:4,:), 'r--'); xlabel('time'); ylabel('vy');
% percent Mean Absolute Error
X=X(:,2:kmax+1);
XEESKF=XEESKF(:,2:kmax+1);
pcMAE1=sum(abs(X(1:1,:)-XEESKF(1:1,:))./X(1:1,:))*100/kmax;
pcMAE2=sum(abs(X(2:2,:)-XEESKF(2:2,:))./X(2:2,:))*100/kmax;
pcMAE3=sum(abs(X(3:3,:)-XEESKF(3:3,:))./X(3:3,:))*100/kmax;
pcMAE4=sum(abs(X(4:4,:)-XEESKF(4:4,:))./X(4:4,:))*100/kmax;
pcMAE=[pcMAE1 pcMAE2 pcMAE3 pcMAE4];
display(pcMAE)

```



Εκτίμηση θέσης και ταχύτητας



% Μέσο Απόλυτο Σφάλμα (% Mean Absolute Error)

Εκτίμηση	%MAE
Απόσταση-x	0.030580946151857%
Απόσταση-y	0.033658713754736%
Ταχύτητα-x	0.320650634869846%
Ταχύτητα-y	0.262738197687769%

pcMAE =

0.030580946151857 0.033658713754736 0.320650634869846 0.262738197687769

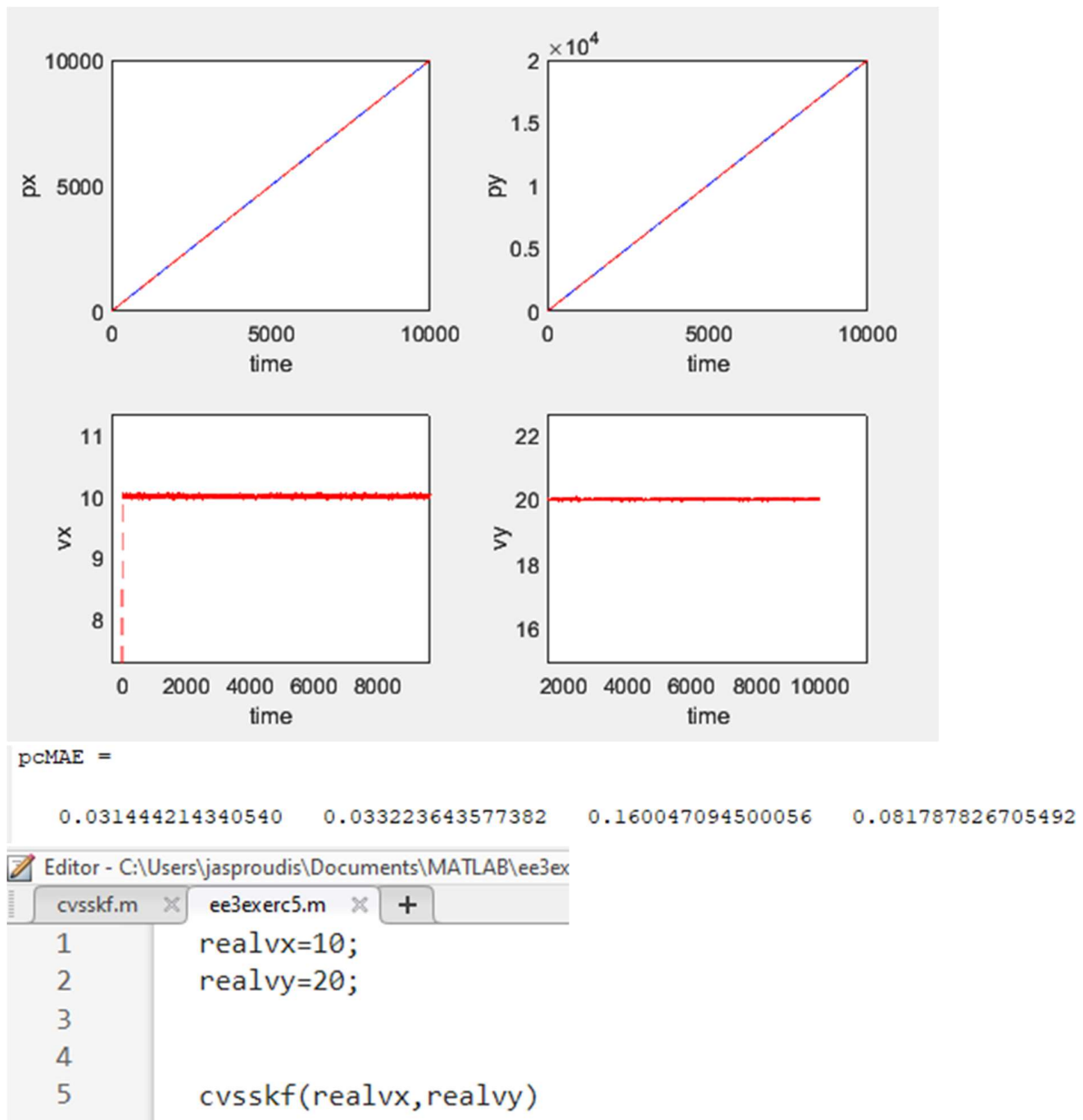
Προσομοίωση (simulation)

Να γράψετε πρόγραμμα για την προσομοίωση κίνησης με σταθερή ταχύτητα στο επίπεδο:

$v_x=10m/s, v_y=20m/s$

χρήση

- του φίλτρου Kalman μόνιμης κατάστασης
- του χρονικά αμετάβλητου φίλτρου Kalman



% Μέσο Απόλυτο Σφάλμα (% Mean Absolute Error)

Εκτίμηση	%MAE
Απόσταση-x	0.031444214340540%
Απόσταση-y	0.033223643577382%
Ταχύτητα-x	0.160047094500056%
Ταχύτητα-y	0.081787826705492%

6. Εκτίμηση θέσης κινούμενου σώματος με σταθερή επιτάχυνση

εκτίμηση της θέσης, της ταχύτητας και της επιτάχυνσης
ενός αντικειμένου που κινείται στο χώρο με σταθερή επιτάχυνση

Η θέση προσδιορίζεται από την απόσταση που διανύει το αντικείμενο (p_x, p_y, p_z), η ταχύτητα από τις συνιστώσες της στους τρεις άξονες (v_x, v_y, v_z) και η επιτάχυνση από τις συνιστώσες της στους τρεις άξονες (a_x, a_y, a_z).

Κατάσταση:

$$x(k) = \begin{bmatrix} p_x(k) \\ p_y(k) \\ p_z(k) \\ v_x(k) \\ v_y(k) \\ v_z(k) \end{bmatrix}$$

Είσοδος (επιτάχυνση):

$$u(k) = \begin{bmatrix} a_x(k) \\ a_y(k) \\ a_z(k) \end{bmatrix}$$

Μετρήσεις:

$$z(k) = x(k) + v(k)$$

Παράμετροι μοντέλου:

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{2}(\Delta t)^2 & 0 & 0 \\ 0 & \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & 0 & \frac{1}{2}(\Delta t)^2 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = 0.3^2 \begin{bmatrix} \frac{1}{4}(\Delta t)^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4}(\Delta t)^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4}(\Delta t)^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & (\Delta t)^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & (\Delta t)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (\Delta t)^2 \end{bmatrix}$$

$$R = \begin{bmatrix} 3^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.03^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.03^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.03^2 \end{bmatrix}$$

Αρχικές συνθήκες:

$$x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} 4^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.4^2 \end{bmatrix}$$

Προσομοίωση (simulation)

Να γράψετε πρόγραμμα για την προσομοίωση κίνησης με σταθερή επιτάχυνση στο χώρο:
κίνηση με σταθερή επιτάχυνση: $a_x=3m/s^2, a_y=2m/s^2, a_z=1m/s^2$

χρήση

- του φίλτρου Kalman μόνιμης κατάστασης
- του χρονικά αμετάβλητου φίλτρου Kalman

Οι μετρήσεις των θέσεων είναι οι θέσεις με θόρυβο με τυπική απόκλιση 3

Οι μετρήσεις των ταχυτήτων είναι οι ταχύτητες με θόρυβο με τυπική απόκλιση 0.03

Η είσοδος είναι η επιτάχυνση με θόρυβο με τυπική απόκλιση 0.3

Η αρχική θέση και η αρχική ταχύτητα θεωρούνται μηδέν.

Ο κώδικας που θα χρησιμοποιηθεί με βάση αυτό το μοντέλο είναι:

```
function caskf(realax, realay, realaz)
% constant acceleration
% Steady State Kalman Filter
% 3 dimensions
% model
n=6;
dt=0.1;
F=[1 0 0 dt 0 0; 0 1 0 0 dt 0; 0 0 1 0 0 dt; 0 0 0 1 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1];
B=[0.5*(dt)^2 0 0; 0 0.5*(dt)^2 0; 0 0 0.5*(dt)^2; dt 0 0; 0 dt 0; 0 0 dt];
```

```

H=eye(n);
sigma1=0.3;
Q=sigma1^2*[0.25*(dt)^4 0 0 0 0;0 0.25*(dt)^4 0 0 0;0 0 0.25*(dt)^4 0 0;0 0 0 (dt)^2 0;0 0 0 0 (dt)^2];
sigma2=3; sigma3=0.03;
R=[sigma2^2 0 0 0 0;0 sigma2^2 0 0 0;0 0 sigma2^2 0 0;0 0 0 sigma3^2 0;0 0 0 0 sigma3^2 0;0 0 0 0 0 sigma1^3];
% initial conditions
x0=[0 0 0 0 0]';
sigma4=4; sigma5=0.4;
pe0=[sigma4^2 0 0 0 0;0 sigma4^2 0 0 0;0 0 sigma4^2 0 0;0 0 0 sigma5^2 0;0 0 0 0 sigma5^2 0;0 0 0 0 0 sigma5^2];
% observability
om=obsv(H,F);
display(om);
rom=rank(om);
display(rom);
% controllability
cm=ctrb(F,B);
display(cm);
rcm=rank(cm);
display(rcm);
% Riccati Equation
PPss=dare(F',H',Q,R);
% Steady State Kalman Filter parameters
Gss=PPss*H'*inv(H*PPss*H'+R);
Ass=(eye(n)-Gss*H)*F;
Css=(eye(n)-Gss*H)*B;
display(Ass);
display(Css);
display(Gss);
% number of iterations
kmax=1000/dt;
% measurements
sigmaa=0.3; sigmav=0.03; sigmap=3;
% rng default;
for k=0:kmax
    ax(k+1)=realax; zax(k+1)=ax(k+1)+sigmaa*randn;
    ay(k+1)=realay; zay(k+1)=ay(k+1)+sigmaa*randn;
    az(k+1)=realaz; zaz(k+1)=az(k+1)+sigmaa*randn;
    vx(k+1)=realax*k*dt; zvx(k+1)=vx(k+1)+sigmav*randn;
    vy(k+1)=realay*k*dt; zvy(k+1)=vy(k+1)+sigmav*randn;
    vz(k+1)=realaz*k*dt; zvz(k+1)=vz(k+1)+sigmav*randn;
    px(k+1)=realax*0.5*(k*dt)^2;
    zpx(k+1)=px(k+1)+sigmap*randn;
    py(k+1)=realay*0.5*(k*dt)^2;
    zpy(k+1)=py(k+1)+sigmap*randn;
    pz(k+1)=realaz*0.5*(k*dt)^2;
    zpz(k+1)=pz(k+1)+sigmap*randn;
end;
A=[ax; ay; az];
X=[px; py; pz; vx; vy; vz];
Z=[zpx; zpy; zpz; zvx; zvy; zvz];
% Steady State Kalman Filter
xe=(eye(n)-Gss*H)*x0+Gss*Z(:,1:1);
XESSKF=xe;
U=[realax realay realaz]';
for k=0:kmax-1
    xe=Ass*xe+Gss*Z(:,k+1:k+1)+Css*U;
    XESSKF=[XESSKF xe];
end;
% plots
figure(1);
timet=[0:kmax];
subplot(3,3,1); plot(timet,A(1:1,:)); xlabel('time'); ylabel('ax');
subplot(3,3,2); plot(timet,A(2:2,:)); xlabel('time'); ylabel('ay');
subplot(3,3,3); plot(timet,A(3:3,:)); xlabel('time'); ylabel('az');
subplot(3,3,4); plot(timet,X(4:4,:), 'b', timet, XESSKF(4:4,:), 'r--'); xlabel('time'); ylabel('vx');

```

```

subplot(3,3,5); plot(timet,X(5:5,:), 'b',timet,XESSKF(5:5,:), 'r--'); xlabel('time'); ylabel('vy');
subplot(3,3,6); plot(timet,X(6:6,:), 'b',timet,XESSKF(6:6,:), 'r--'); xlabel('time'); ylabel('vz');
subplot(3,3,7); plot(timet,X(1:1,:), 'b',timet,XESSKF(1:1,:), 'r--'); xlabel('time'); ylabel('px');
subplot(3,3,8); plot(timet,X(2:2,:), 'b',timet,XESSKF(2:2,:), 'r--'); xlabel('time'); ylabel('py');
subplot(3,3,9); plot(timet,X(3:3,:), 'b',timet,XESSKF(3:3,:), 'r--'); xlabel('time'); ylabel('pz');
% percent Mean Absolute Error
X=X(:,2:kmax+1);
XESSKF=XESSKF(:,2:kmax+1);
pcMAE1=sum(abs(X(1:1,:)-XESSKF(1:1,:))./X(1:1,:))*100/kmax;
pcMAE2=sum(abs(X(2:2,:)-XESSKF(2:2,:))./X(2:2,:))*100/kmax;
pcMAE3=sum(abs(X(3:3,:)-XESSKF(3:3,:))./X(3:3,:))*100/kmax;
pcMAE4=sum(abs(X(4:4,:)-XESSKF(4:4,:))./X(4:4,:))*100/kmax;
pcMAE5=sum(abs(X(5:5,:)-XESSKF(5:5,:))./X(5:5,:))*100/kmax;
pcMAE6=sum(abs(X(6:6,:)-XESSKF(6:6,:))./X(6:6,:))*100/kmax;
pcMAE=[pcMAE1 pcMAE2 pcMAE3 pcMAE4 pcMAE5 pcMAE6];
display(pcMAE);

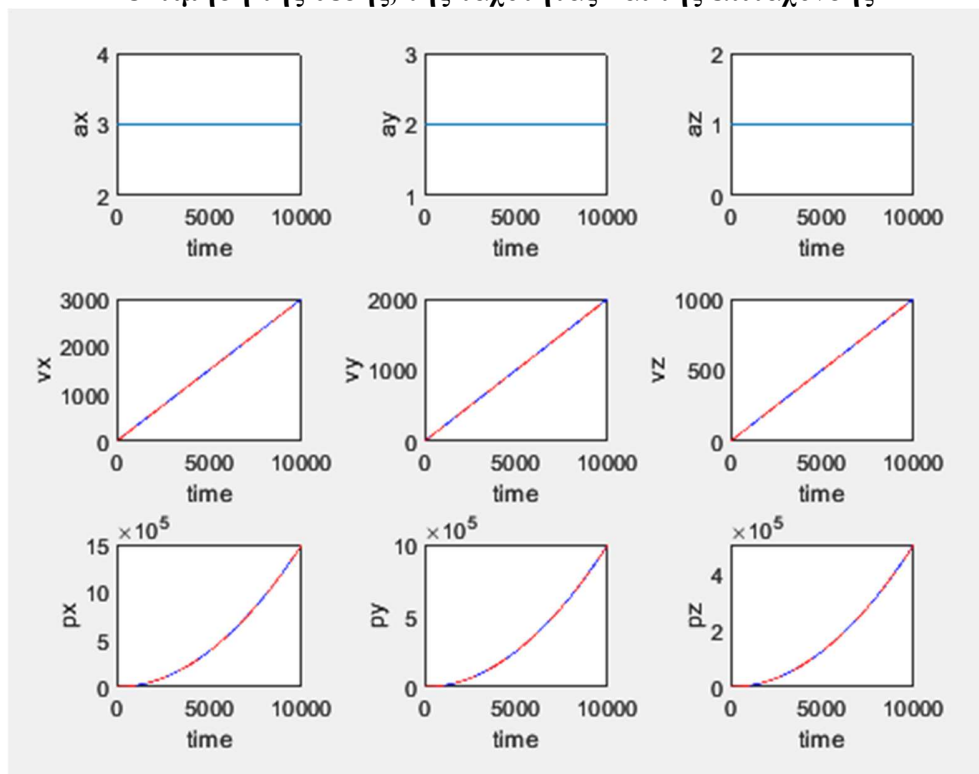
```

```

Editor - C:\Users\jasproudis\Documents\MATLAB\ee3exerc6.m
cvsskf.m x ee3exerc5.m x casskf.m x ee3exerc6.m
1      clc; clear all;
2
3      relax = 3;
4      relay = 2;
5      realaz = 1;
6
7      casskf(relax,relay,realaz);

```

εκτίμηση της θέσης, της ταχύτητας και της επιτάχυνσης



```
pcMAE =

Columns 1 through 5

    0.181482733756230    0.168359957404956    0.222009035288048    0.092488471127736    0.092834061202074

Column 6

    0.082336974928764
```

% Μέσο Απόλυτο Σφάλμα (% Mean Absolute Error)

εκτίμηση	%MAE
Απόσταση-x	0.181482733756230%
Απόσταση-y	0.168359957404956%
Απόσταση-z	0.222009035288048%
Ταχύτητα-x	0.092488471127736%
Ταχύτητα-y	0.092834061202074%
Ταχύτητα-z	0.082336974928764%

7. Τυχαίος περίπατος

Ένας άνθρωπος περπατάει με τυχαίο περίπατο (random walk) στο επίπεδο (x-axis, y-axis), για χρονικό διάστημα k χρονικών στιγμών, ξεκινώντας από την αρχή των αξόνων.

Εκτίμηση απόστασης τυχαίου περιπάτου

μοντέλο με παραμέτρους $F=1, H=1, Q=1, R=1$

χρονικά αμετάβλητο φίλτρο Kalman

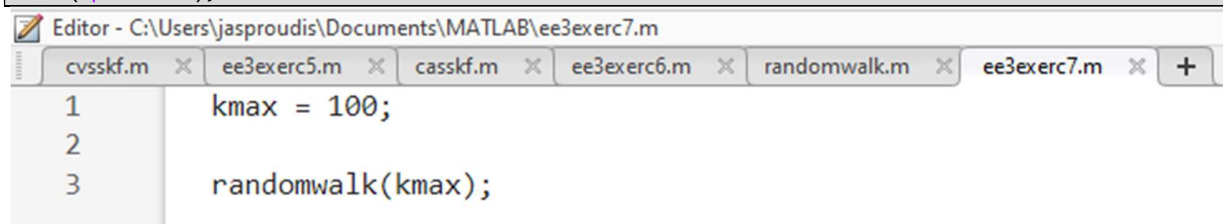
$$\begin{aligned}
 K(k) &= \frac{P(k|k-1)}{1 + P(k|k-1)} \\
 x(k|k) &= [1 - K(k)] x(k|k-1) + K(k)z(k) \\
 P(k|k) &= [1 - K(k)]P(k|k-1) \\
 x(k+1|k) &= x(k|k) \\
 P(k+1|k) &= 1 + P(k|k)
 \end{aligned}$$

Η εκτίμηση θα γίνει μέσω κώδικα Matlab ο οποίος:

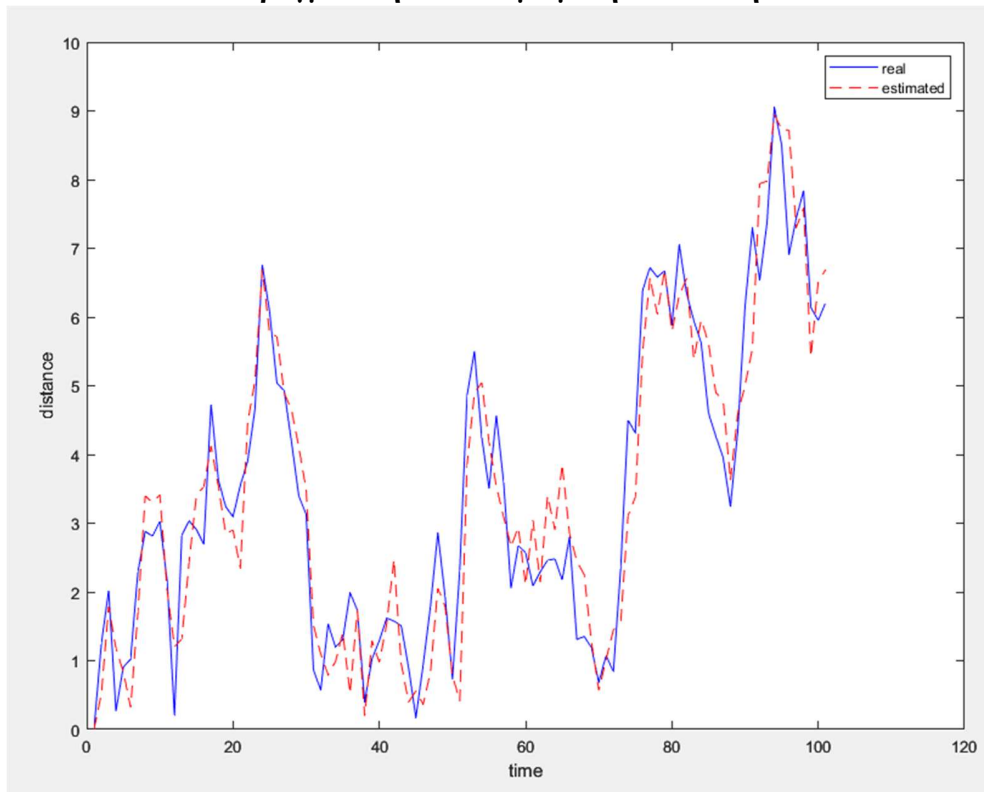
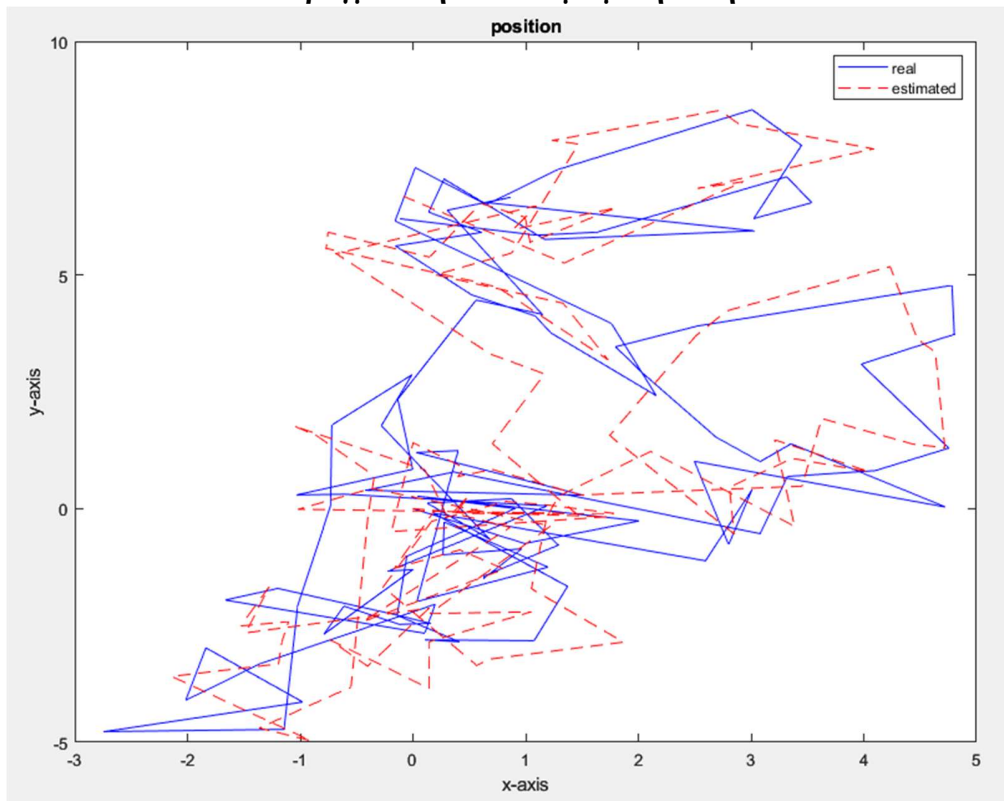
- Θα κάνει προσομοίωση του τυχαίου βηματισμού με μηδενικές αρχικές συνθήκες
- Θα υπολογίζει και σχεδιάζει την πραγματική απόσταση και την εκτιμώμενη απόσταση για 100 χρονικές στιγμές
- Θα υπολογίζει και σχεδιάζει την πραγματική θέση και την εκτιμώμενη θέση για 100 χρονικές στιγμές

Ο κώδικας που θα χρησιμοποιηθεί είναι:

```
function randomwalk(kmax)
% random walk
% rng default;
% model parameters
F=1;
H=1;
Q=1;
R=1;
% initial conditions
x0=0;
P0=0;
% time invariant Kalman filter
% x-axis movement
X=x0;
x=x0;
xe=x0; XE=xe;
pe=P0;
for i=1:kmax
    x=F*x+randn; X=[X x];
    z=H*x+randn;
    xp=F*xe;
    pp=Q+F*pe*F';
    g=pp*H'*inv(H*pp*H'+R);
    xe=(1-g*H)*xp+g*z; XE=[XE xe];
    pe=(1-g*H)*pp;
end;
Xreal=X;
Xest=XE;
% y-axis movement
X=x0;
x=x0;
xe=x0; XE=xe;
pe=P0;
for i=1:kmax
    x=F*x+randn; X=[X x];
    z=H*x+randn;
    xp=F*xe;
    pp=Q+F*pe*F';
    g=pp*H'*inv(H*pp*H'+R);
    xe=(1-g*H)*xp+g*z; XE=[XE xe];
    pe=(1-g*H)*pp;
end;
Yreal=X;
Yest=XE;
dreal=sqrt(Xreal.^2+Yreal.^2);
dest=sqrt(Xest.^2+Yest.^2);
% plots
figure(1);
tk=[1:kmax+1];
plot(tk,dreal,'b',tk,dest,'r--');
xlabel('time');
ylabel('distance');
legend('real','estimated');
figure(2);
plot(Xreal,Yreal,'b',Xest,Yest,'r--');
xlabel('x-axis');
ylabel('y-axis');
legend('real','estimated');
title('position');
```



```
Editor - C:\Users\jasproudis\Documents\MATLAB\ee3exerc7.m
cvsskf.m x ee3exerc5.m x casskf.m x ee3exerc6.m x randomwalk.m x ee3exerc7.m x +
1      kmax = 100;
2
3      randomwalk(kmax);
```


πραγματική και εκτιμώμενη απόσταση**πραγματική και εκτιμώμενη θέση**

8. Πρόβλεψη κρουσμάτων Covid-19 με φίλτρο Kalman

Να μελετήσετε την εργασία

Assimakis N., Adam M., Ktena A., Manasis C., “Steady state Kalman filter design for cases and deaths prediction of Covid-19 in Greece”, Results in Physics, vol. 26, doi:

<https://doi.org/10.1016/j.rinp.2021.104391>, 2021.

Να βρείτε τα δεδομένα της ελληνικής κυβέρνησης για τα κρούσματα.

Daily Overview, Information about Covid-19 in Greece, <https://covid19.gov.gr/covid19-live-analytics/>

A) Να υπολογίσετε τους συντελεστές του Golden Steady State Kalman Filter.

```
Command Window
>> a=(sqrt(5)-1)/2
a =
    0.618033988749895
>> a2=a^2
a2 =
    0.381966011250105
fx >>
```

$$\text{χρυσή τομή}$$

$$a = \frac{\sqrt{5} - 1}{2} = 0,618033989$$

$$\text{Golden Steady State Kalman Filter}$$

$$x(k+1/k) = a^2 x(k/k-1) + az(k)$$

B) Να υπολογίσετε την πρόβλεψη για τα νέα κρούσματα για το διάστημα 1/2/2022-10/2/2022.

Γράφουμε το παρακάτω script εντάσσοντας τα δεδομένα από το site:

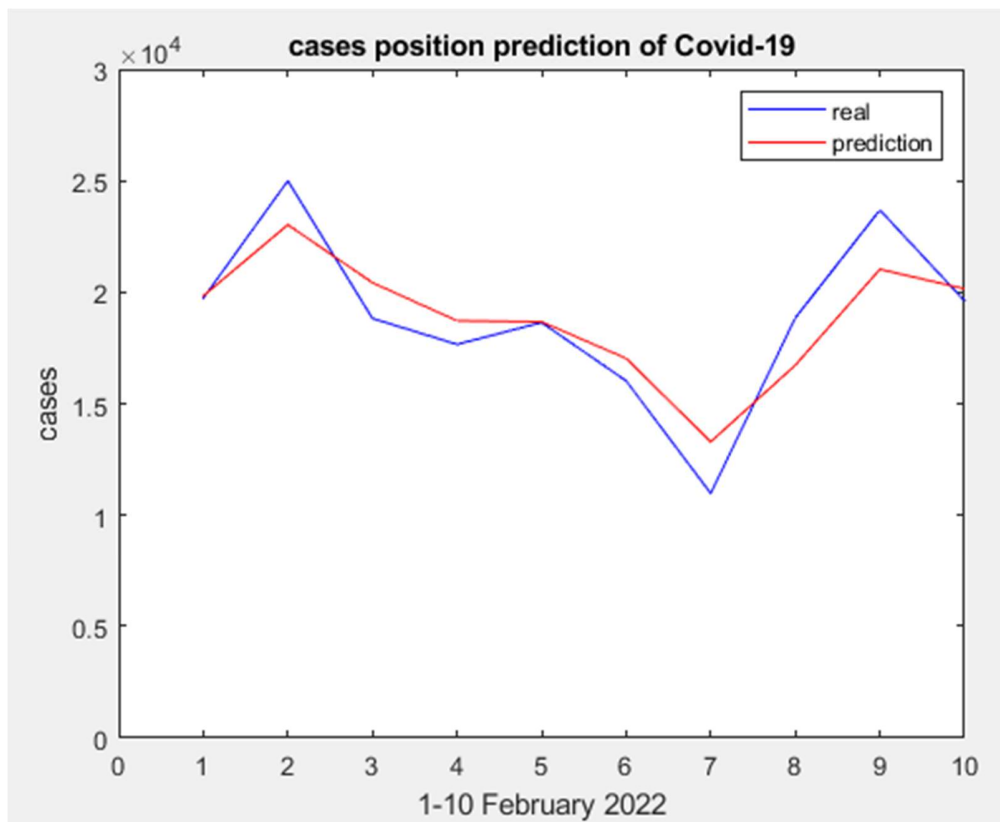
```
function kfcovid
% cases prediction of Covid-19
% steady state Kalman filter

% data cases
Z=[19731 24996 18825 17662 18640 16023 10980 18847 23687 19616];

% steady state Kalman Filter parameters
pp=dare(1,1,1,1);
b = pp/(pp+1);
a=1-b;

% steady state Kalman Filter
x0= 20000;
xe=x0;
```

```
XE=[];  
  
for k=1:10  
    xe=a*xe+b*Z(k);  
    XE=[XE xe];  
end;  
  
XESSKF=ceil(XE);  
  
% plots  
figure(1);  
clf;  
timek=[1:10];  
plot(timek,Z,'b',timek,XESSKF,'r');  
axis([0, 10, 0, 30000]);  
legend('real','prediction');  
xlabel ('1-10 February 2022');  
ylabel ('cases');  
title('cases position prediction of Covid-19');
```



9-10. Εργασία – Εκτίμηση τυχαίας σταθεράς με χρονικά αμετάβλητο φίλτρο Kalman

ΕΡΓΑΣΙΑ

Να μελετήσετε το παράδειγμα στις σελίδες 11-15 της εργασίας:
G. Welch and G. Bishop, “An Introduction to the Kalman Filter”, 2006.
http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

Να θεωρήσετε τη σταθερά $c=10$.

Να θεωρήσετε το μοντέλο με παραμέτρους: $F=1, H=1, Q=0.001, R=1$

Να παράγετε μετρήσεις της σταθεράς με θόρυβο.

Να γράψετε πρόγραμμα για να παράγετε την εκτίμηση της σταθεράς με χρήση του χρονικά αμετάβλητου φίλτρου Kalman, για $k=0,1,\dots,100$ με αρχικές συνθήκες $x_0=0, P_0=1$

Να σχεδιάσετε την πραγματική σταθερά, τις μετρήσεις και την εκτίμηση.

Να σχεδιάσετε τη διασπορά λάθους εκτίμησης.

Θα χρησιμοποιήσουμε τον παρακάτω κώδικα:

```
function tikfconstantest(c,kmax)
% constant estimation using Time Invariant Kalman Filter
rng default;

% constant
C=c.*ones(1,kmax+1);

% model parameters
F=1;
H=1;
Q=10^(-3);
R=1;

% measurements
Z=[];

for k=0:kmax
    z=c+R*randn; Z=[Z z];
end;

% TIKF
% initial conditions
xp0=0;
pp0=1;
xp=xp0;
pp=pp0;
XE=[];
PE=[];
```

```

for k=0:kmax

    g=(pp*H)/(H*pp*H+R);
    xe=(1-g*H)*xp+g*Z(k+1);
    pe=(1-g*H)*pp;
    xp=F*xe;
    pp=Q+F*pe*F';
    XE=[XE xe];
    PE=[PE pe];
end;
XEKF=XE;
PEKF=PE;

% plots
clf;
figure(1);
timek=[0:kmax];
plot(timek,C,'k',timek,Z,'b',timek,XEKF,'r');
legend('constant','measurements','KF estimation');
xlabel('time k');
ylabel('constant and estimation');
figure(2);
timek=[0:kmax];
plot(timek,PEKF,'r');
xlabel('time k');
ylabel('estimation error variance');

```

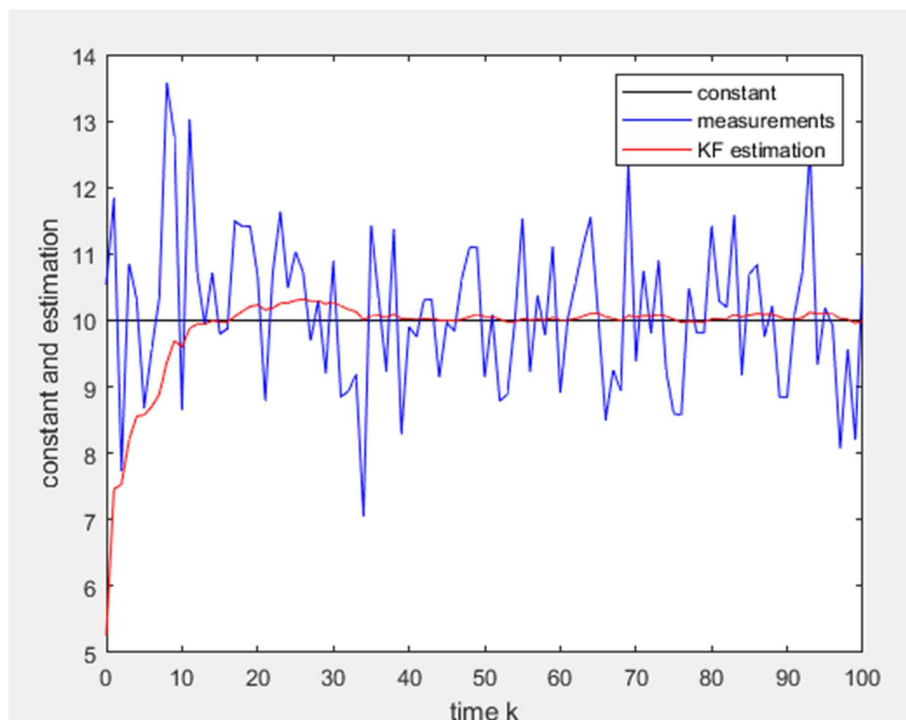
Χρησιμοποιώντας την συνάρτηση `tikfconstantest(c,kmax)` για $c=10$ και $K_{max}=100$ προκύπτουν τα εξής διαγράμματα:

```

Editor - C:\Users\jasproudis\Documents\MATLAB\ee3exerc9_10.m
tikfconstantest.m ee3exerc9_10.m untitled19 * kfcovid.m +
1      clc; clear all;
2
3      c = 10;
4      kmax = 100;
5
6
7      tikfconstantest(c,kmax)

```

Πραγματική σταθερά, μετρήσεις και εκτίμηση



Διασπορά λάθους εκτίμησης