Προσομοίωση και Μοντελοποίηση Συστημάτων

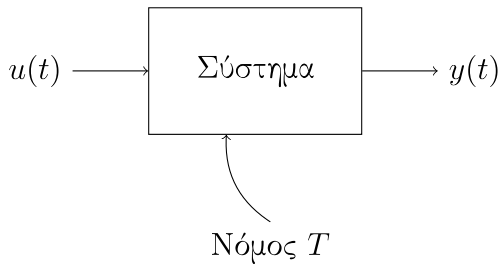
Ιωαννης-Παναγιωτης
Μπουντουριδης

ΑΕΜ: 8872

project

1) Γραμμικό σύστημα

Δοθέντος του παρακάτω συστήματος:



έστω οι έξοδοι

$$y_1(t) = T\left[u_1(t)\right]$$
$$y_2(t) = T\left[u_2(t)\right]$$

το σύστημα είναι γραμμικό ανν:

$$T\left[u_1(t) + u_2(t)\right] = T\left[u_1(t)\right] + T\left[u_2(t)\right]$$

επιλέγουμε τις εισόδους:

$$\boxed{u_1 = 4 \cdot sin(2 \cdot t)}$$

$$\boxed{u_2 = 3 \cdot cos(5 \cdot t)}$$

$$\boxed{u = 4 \cdot sin(2 \cdot t) + 3 \cdot cos(5 \cdot t) = u_1 + u_2}$$

παίρνουμε τις εξόδους απο το αρχείο $out.p$:

$$\boxed{y_1(t) = T\left[u_1(t)\right]}$$

$$\boxed{y_1(t) = T\left[u_2(t)\right]}$$

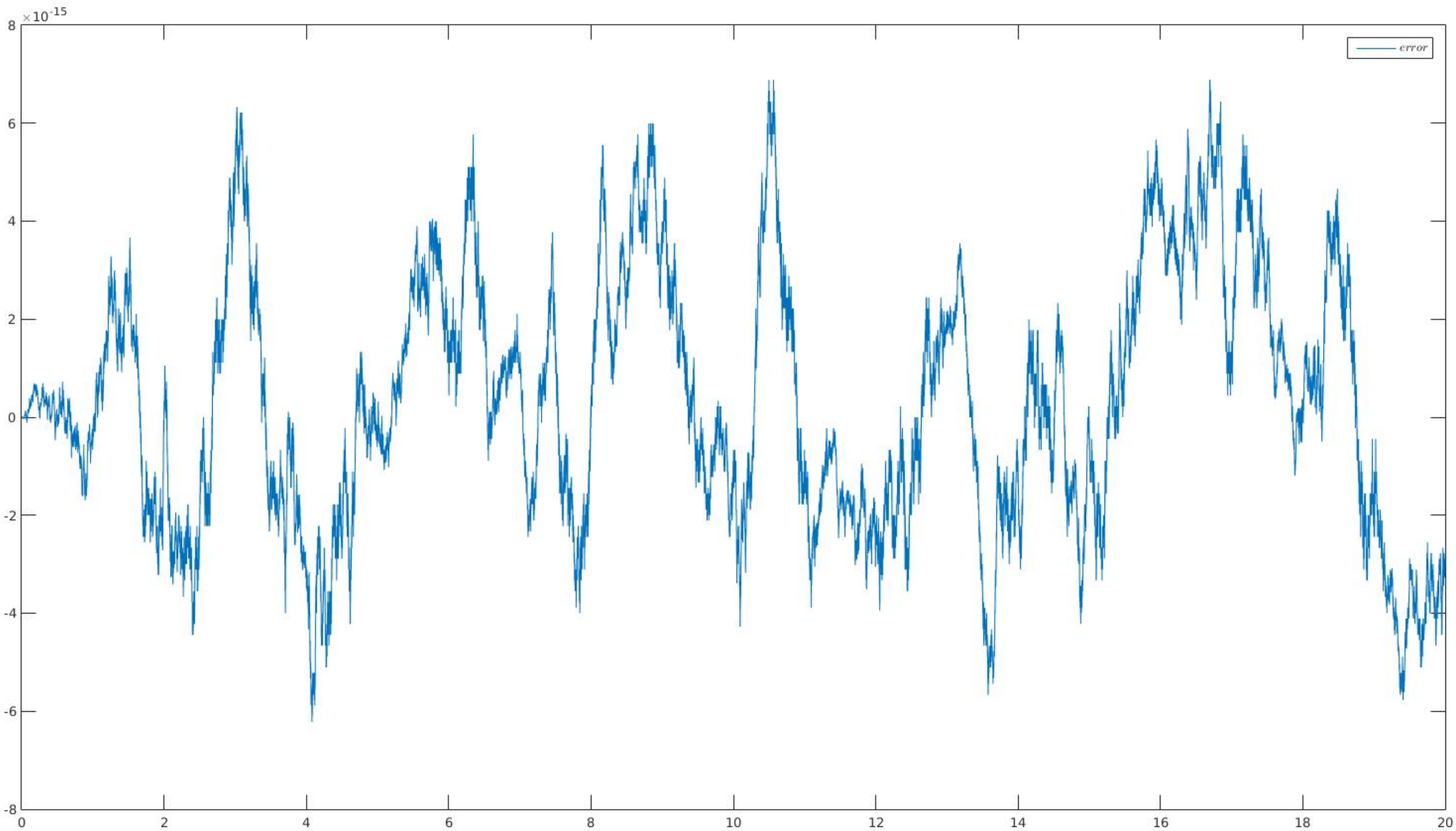
$$\boxed{y(t) = T\left[u(t)\right] = T\left[u_1(t) + u_2(t)\right]}$$

ορίζουμε ως σφάλμα την διαφορά εξόδου των σημάτων:

$$e(t) = y(t) - y_1(t) - y_2(t)$$

για χρόνο: $0 \leq t \leq 20$ με βήμα 0.001 sec παρουσιάζεται το σφάλμα τιμών συναρτήσει του χρόνου.

Γραφική παράσταση σφάλματος



Το σφάλμα της γραφικής παράστασης είναι της κλίμακας 10^{-15} και θεωρείται αμεληταίο. Είναι σαφές λοιπόν πως ικανοποιείται ορισμός του **γραμμικού συστήματος.**

$$\boxed{T\left[u_1(t)+u_2(t)\right]=T\left[u_1(t)\right]+T\left[u_2(t)\right]}$$

Ο κώδικας για την υλοποίηση του έλεγχου γραμμικότητας του συστήματος είναι διαθέσιμος στο αρχείο *check_linear.m* εντος του φάκελου *matlab_source_code*

Μέθοδος μη-πραγματικού χρόνου

Για την εκτίμηση των παραμέτρων με μια μέθοδο μη-πραγματικού χρόνου, **επιλέγουμε τον αλγόριθμο των ελαχίστων τετραγώνων.**

Επειδή οι τάξεις της εισόδου και της εξόδου του συστήματος μας είναι παντελώς άγνωστες, έστω n και m αντίστοιχα οι τάξεις της εισόδου και της εξόδου που επιθυμούμε να βρούμε.

Το σύστημα στο οποίο αναζητούμε τις λύσεις των παραμέτρων θα είναι της μορφής

$$y^{(n)}+\theta_0\cdot y^{(n-1)}+\theta_1\cdot y^{(n-2)}+...+\theta_{n-1}\cdot y = \theta_n\cdot u^{(m)}+\theta_{n+1}\cdot u^{(m-1)}+\theta_{n+2}\cdot u^{(m-2)}+...+\theta_{n+m}\cdot u$$

επιθυμούμε να φέρουμε το σύστημα στην μορφή

$$y = {\theta_\lambda}^T \zeta$$

οπου

$$\theta_\lambda = \left[{\theta_1}^{*T} - \lambda^T \quad {\theta_2}^{*T} \right]^T$$

$${\theta_1}^* = \left[\theta_0 \quad \theta_1 \quad \theta_2 \quad ... \quad \theta_{n-1} \right]^T$$

$${\theta_2}^* = \left[\theta_n \quad \theta_{n+1} \quad \theta_{n+2} \quad ... \quad \theta_{n+m} \right]^T$$

και για κάθε περίπτωση επιλέγουμε ενα φίλτρο n ταξής

$$\Lambda(s) = s^n + \lambda^T \Delta_{n-1}$$

$$\lambda = [\ \lambda_1 \ \ \lambda_2 \ \ ... \ \ \lambda_n \ ]^T$$

κατα τα γνωστά συμφωνα με την θεωρία

$$\zeta = \left[-\frac{{\Delta^T}_{n-1}(s)}{\Lambda(s)} y \quad \frac{{\Delta^T}_m(s)}{\Lambda(s)} u \right]$$

Συνεπώς αυτό που απαιτείται στην μοντελοποίηση ενος συστήματος κατα μαύρο κουτί είναι να υποθέσουμε τις άγνωστες τάξεις εισόδου και εξόδου.

Παρακάτω παρουσιάζεται μια σειρά δοκιμών που έγινε για την επιλογή της τάξης εισόδου και εξόδου του συστήματος.

Αμέσως μετά την απεικόνιση των διαγραμμάτων που θα ακολουθήσουν κρίνουμε κατα πόσο εγκυρα ηταν αποτελέσματα κάθε μοντελου.

Δοκιμές	n	m
1^{η} δοκιμή	1	0
2^{η} δοκιμή	2	0
3^{η} δοκιμή	2	1
4^{η} δοκιμή	3	0
5^{η} δοκιμή	3	1
6^{η} δοκιμή	3	2
7^{η} δοκιμή	4	0
8^{η} δοκιμή	4	1
9^{η} δοκιμή	4	2
10^{η} δοκιμή	4	3

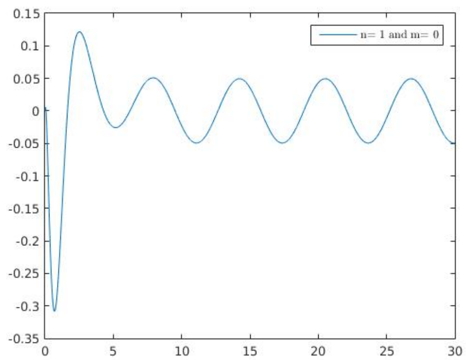
Τα διαγράμματα που ακολουθούν είναι γραφικές παράστασείς του σφάλματος εκτίμησης εξόδου.

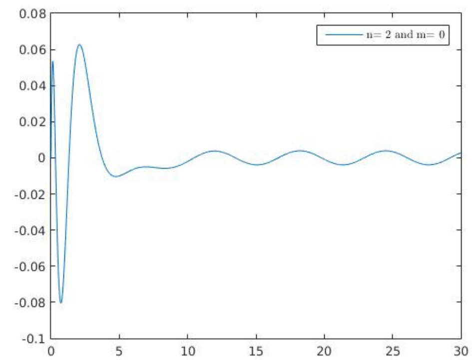
$$error = Y_{real} - Y_{least\_square}$$

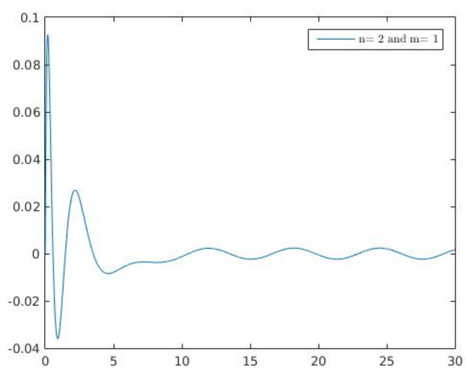
όπου:

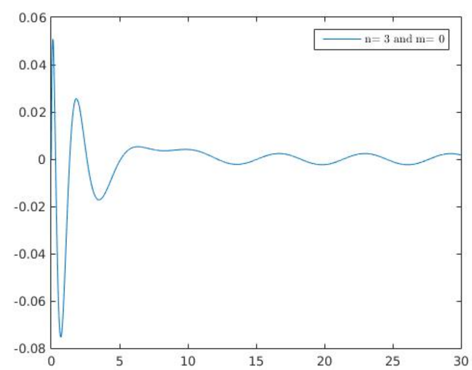
- Y_{real} είναι οι τιμές εξόδου του πραγματικου σύστημος που μας επιστρέφει το αρχείο out.p
- Y_{least_square} είναι οι τιμές που προέκυψαν απο το γινόμενο του πίνακα παραμέτρων (αποτέλεσμα του αλγόριθμου ελαχίστων τετραγώνων) με τον πίνακα οπισθοδρομητών φ.

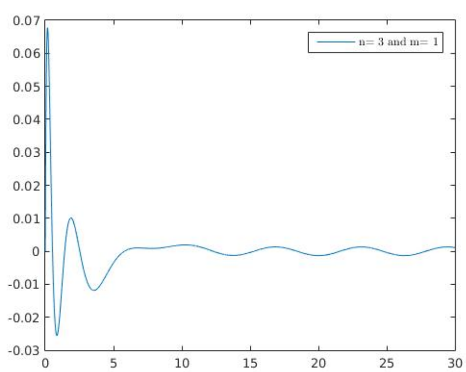
Γραφηκές παραστάσεις δοκιμών

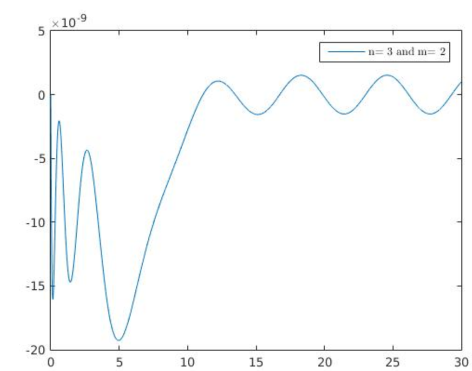


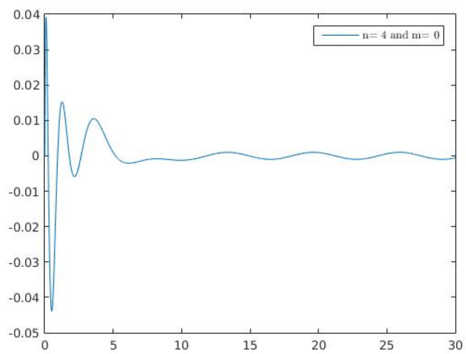


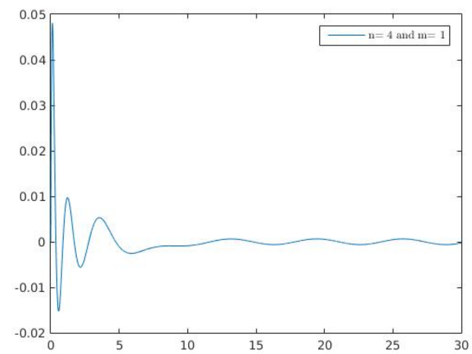


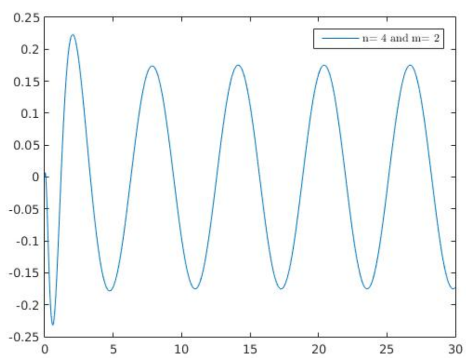


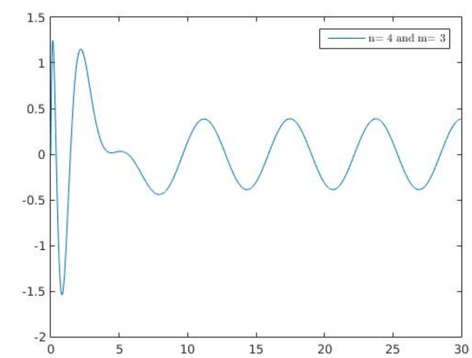












Σκοπός των δοκιμών αυτών είναι να παρατηρήσουμε ποιος συνδυασμος τάξεων n και m μας οδηγεί στο μικρότερο σφάλμα και κατ επέκταση ποιο μοντελοποιημένο σύστημα προσεγγίζει καλύτερα τις τιμές εξόδου του πραγματικού. Εφόσον η επιλογή των τάξεων n και m γίνει είμαστε σε θέση να βρούμε τις τιμές των παραμέτρων του συστήματος. Ο κώδικας *matlab* για την δημιουργία των παραπάνω γραφικών παραστάσεων αλλα και για τον υπολογισμό των παραμέτρων του συστήματος, υπάρχει στο αρχείο *offline.m* και βρίσκεται στον φάκελο *matlab_source_code/offline*. Στον φάκελο αυτό υπαρχουν επίσης καποια βοηθητικά αρχεία όπως:

- *filtermaster.m*: το οποίο επιλέγει φίλτρο για το σύστημα ανάλογα με την τάξη εξόδου του συστήματος.
- *calcmaster.m*: το οποίο υπολογίζει τον πίνακα οπισθοδρομιτών φ του συστήματος.

Συμπέρασματα

Δεδομένου των σφαλμάτων που παρουσιάζονται στα παραπάνω διαγράμματα, παρατηρούμε πως το μικρότερο σφάλμα της διαφοράς:

$$πραγματικής\ εξόδου\ με\ την\ εκτιμώμενη\ έξοδο$$

το συναντάμε για τις τιμές

$$\boxed{n = 3 \text{ και } m = 2}$$

η επιλογή του φίλτρου που κάναμε για αυτή την τάξη εξόδου ήταν

$$\Lambda(s) = s^3 + 4s^2 + 5s + 2$$

συνεπώς λαμβανοντας υπ όψιν το παραπάνω φίλτρο υπολογίζονται οι τιμές των παραμέτρων:

$$\theta^T{}_\lambda = \begin{bmatrix} 6 & 11 & 6 & 6 & 1 & -3 & 2 \end{bmatrix}$$

και το σύστημα περιγράφεται απο την διαφορική:

$$y^{(3)} + 6y^{(2)} + 11\dot{y} + 6y = u^{(2)} - 3\dot{u} + 2u$$

Αξίζει να σημειωθεί πως για την επιλογή των τάξεων εισόδου και εξόδου του συστήματος δεν αρκούμαστε μόνο στην παρατήρηση του σφάλματος. Σε μια περίπτωση που δύο δοκιμές (τάξης εισόδου-εξόδου) μας οδηγούσαν σε παρόμοια σφάλματα, το να στηρίζαμε την επιλογή ενός μοντέλου απλά παρατηρόντας την γραφική παράσταση με το μάτι θα ήταν αβάσιμο και επικύνδυνο. Για αυτό τον λόγο πρόκειται παρακάτω να αξιολογίσουμε τις επιλογές των τάξεων αυτων και κατ επέκταση να τεκμιριώσουμε οτι η επιλογή μας δίνει ελάχιστο σφάλμα μοντέλου.

Μέθοδος πραγματικού χρόνου

Για την εκτίμηση των παραμέτρων με μια μέθοδο πραγματικού χρόνου επιλέγουμε τον αναδρομικό αλγόριθμο ελαχίστων τετραγώνων.

Θεωρούμε το γραμμικά παραμετροποιημένο σύστημα:

$$y = \theta^{*T}\varphi$$

οπου y η έξοδος, θ^* το διάνυσμα των άγνωστων αλλα σταθερών παραμέτρων και φ το διάνυσμα των οπισθιδρομητών. Τόσο η έξοδος όσο και το φ είναι μετρήσιμα.

Έστω το σύστημα αναγνώρισης:

$$\hat{y} = \hat{\theta^T}\varphi$$

όπου $\hat{y}$ η εκτιμώμενη έξοδος και $\hat{\theta}$ το διάνυσμα των εκτιμώμενων παραμέτρων.

Το σφάλμα αναγνώρισης ορίζεται ως:

$$\tilde{y} = y - \hat{\theta^T}\varphi$$

Θέλουμε να υπολογίσουμε την τιμή $\hat{\theta}$ που θα ελαχιστοποιεί το κριτήριο:

$$K(\hat{\theta}) = \frac{1}{2}e^{-\beta t}(\hat{\theta}(t)-\hat{\theta}(0))^T Q_0(\hat{\theta}(t)-\hat{\theta}(0))+\frac{1}{2}\int_0^t e^{-\beta(t-\tau)}[y(\tau)-\hat{\theta}(t)^T\varphi(\tau)]^2 d\tau$$

οπου $Q_0 = {Q_0}^T$ (θετικά ορισμένος πίνακας), β≥0 μια σταθερά. Η K($\hat{\theta}$) είναι κυρτή συνάρτηση του $\hat{\theta}$. Επομένως κάθε τοπικό ελάχιστο είναι και ολικό και ικανοποιεί:

$$\nabla K(\hat{\theta}) = 0$$

καταλήγουμε:

$$\hat{\theta}(t) = P(t)[e^{-\beta t}Q_0\hat{\theta}(0) + \int_0^t e^{-\beta(t-\tau)}y(\tau)\varphi(\tau)d\tau]$$

$$P(t) = [e^{-\beta t}Q_0 + \int_0^t e^{-\beta(t-\tau)}\varphi(\tau)\varphi^T(\tau)d\tau]^{-1}$$

γνωρίζουμε οτι ΡP^{-1} = Ι οπου Ι ο μοναδιαίος πίνακας συνεπώς:

$$\frac{d}{dt}[PP^{-1}] = 0$$

λύνοντας ως προς $\dot{P}$:

$$\dot{P} = -P(\frac{d}{dt}P^{-1})P$$

$$P^{-1}(t) = e^{-\beta t}Q_0 + \int_0^t e^{-\beta(t-\tau)}\varphi(\tau)\varphi^T(\tau)d\tau$$

$$\frac{d}{dt}P^{-1} = -\beta P^{-1} + \varphi\varphi^T$$

αντικαθιστώντας βγάζουμε:

$$\boxed{\dot{P} = \beta P - P\varphi\varphi^T P} \ \ \kappa\alpha\iota \ \ P(0) = {Q_0}^{-1}$$

ομοίως:

$$\hat{\theta}(t) = P(t)\Omega(t)$$

$$\dot{\Omega} = -\beta\Omega + y\varphi \ \ \Omega(0) = Q_0\hat{\theta}(0)$$

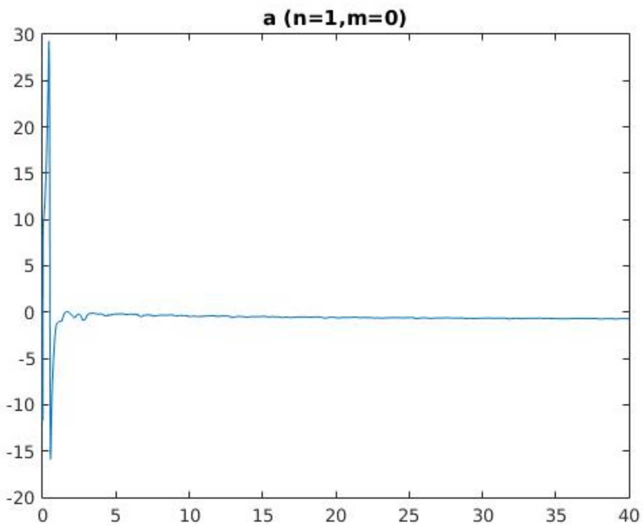
παραγωγίζοντας το $\hat{\theta}$ καταλήγουμε:

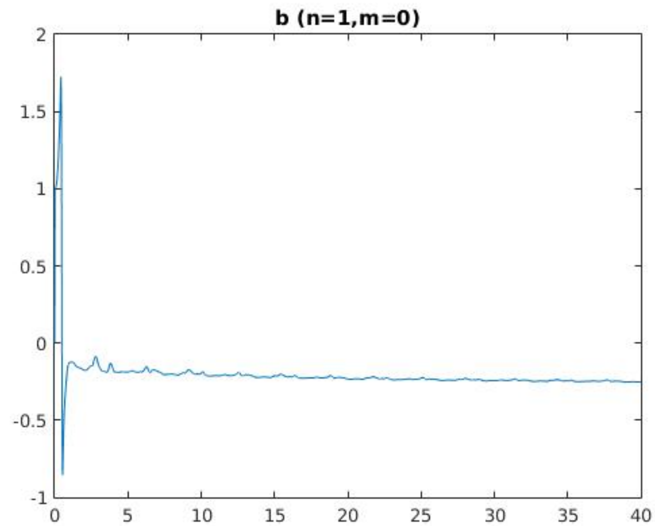
$$\boxed{\dot{\hat{\theta}} = P(t)\tilde{y}(t)\varphi(t)}$$

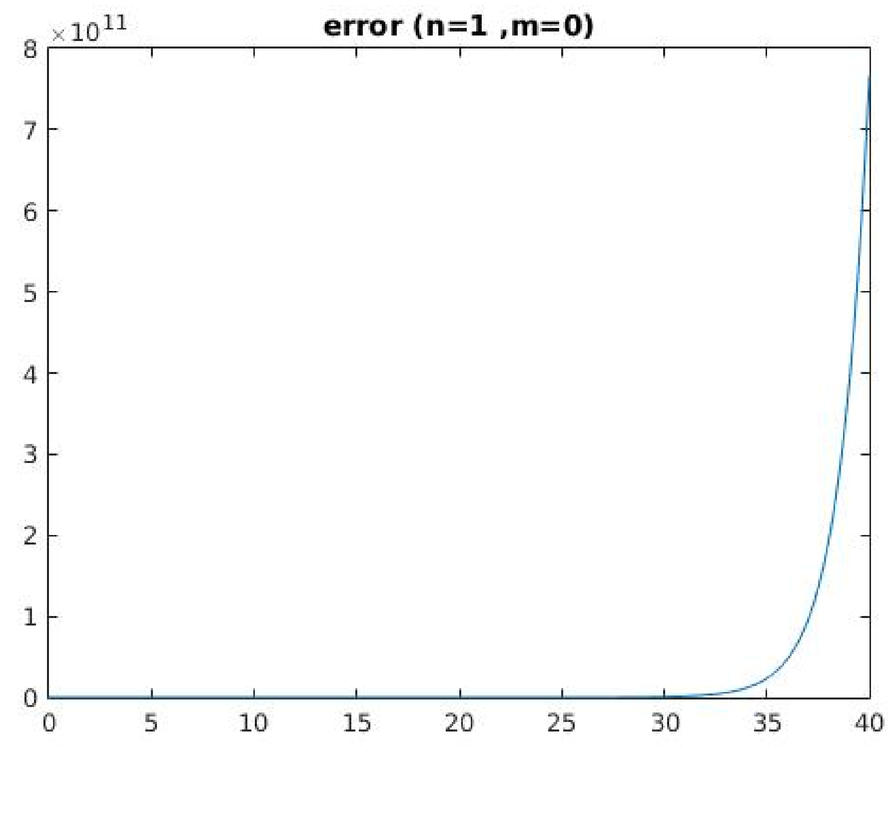
Η αναδρομική μέθοδος των ελαχίστων τετραγώνων εχει υλοποιηθεί στο αρχείο *online.m*

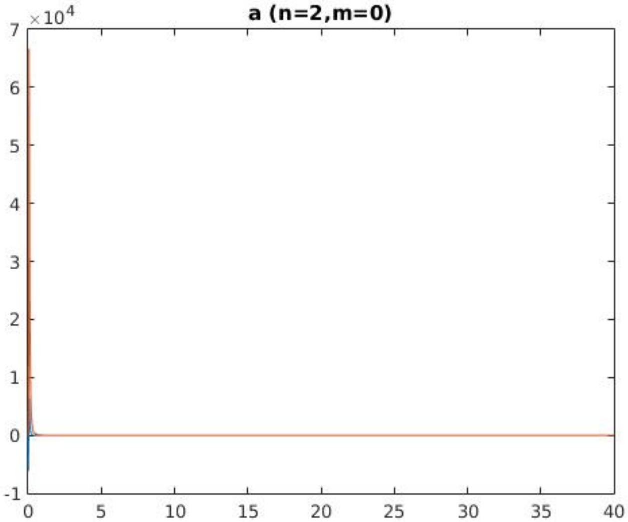
Αντίστοιχα με την offline μέθοδο πρόκειται να λύσουμε το σύστημα μέσα απο μια σειρά δοκιμών για διάφορες τιμές των n και m, παρακάτω παρουσιάζουμε τις τιμές που δοκιμάσαμε συνοπτικά στον πίνακα που ακολουθεί

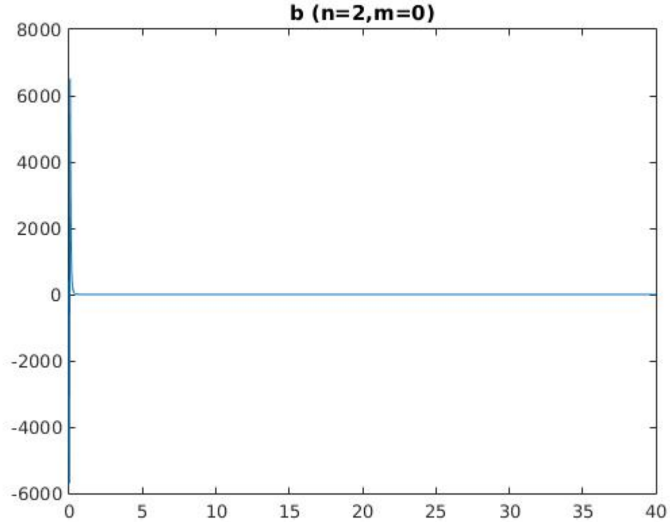
Δοκιμές	n	m
1^η δοκιμή	1	0
2^η δοκιμή	2	0
3^η δοκιμή	2	1
4^η δοκιμή	3	0
5^η δοκιμή	3	1
6^η δοκιμή	3	2

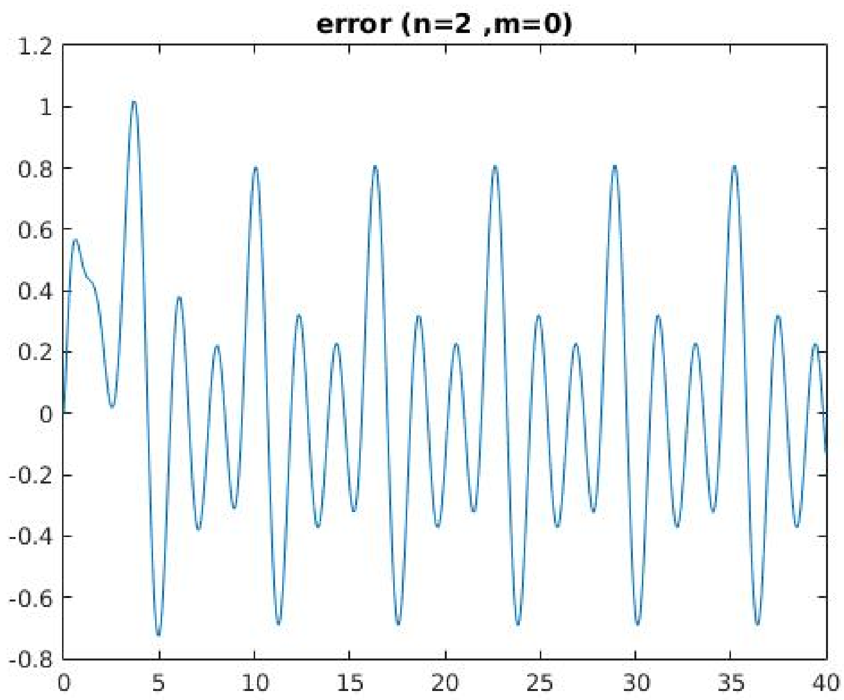

a (n=1,m=0)
30
25
20
15
10
5
0
-5
-10
-15
-20
0
5
10
15
20
25
30
35
40

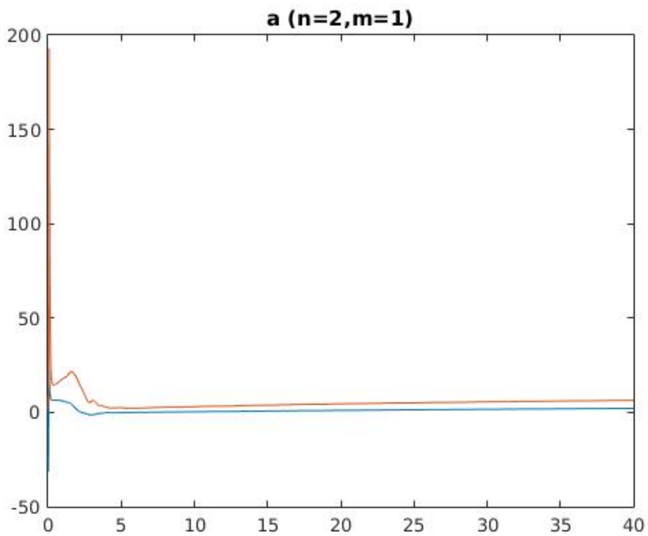

b (n=1,m=0)
2
1.5
1
0.5
0
-0.5
-1
0
5
10
15
20
25
30
35
40

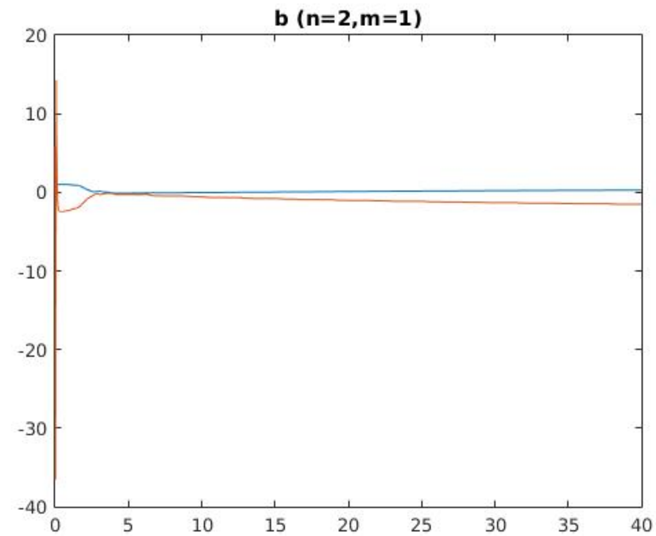

error (n=1 ,m=0)
×10^11
8
7
6
5
4
3
2
1
0
0
5
10
15
20
25
30
35
40

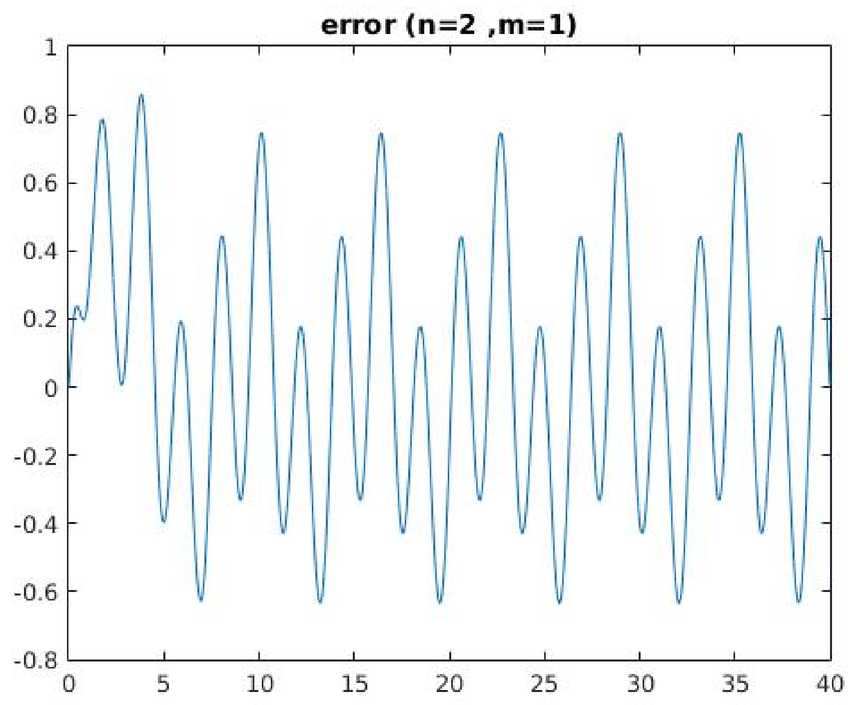

a (n=2,m=0)
×10^4

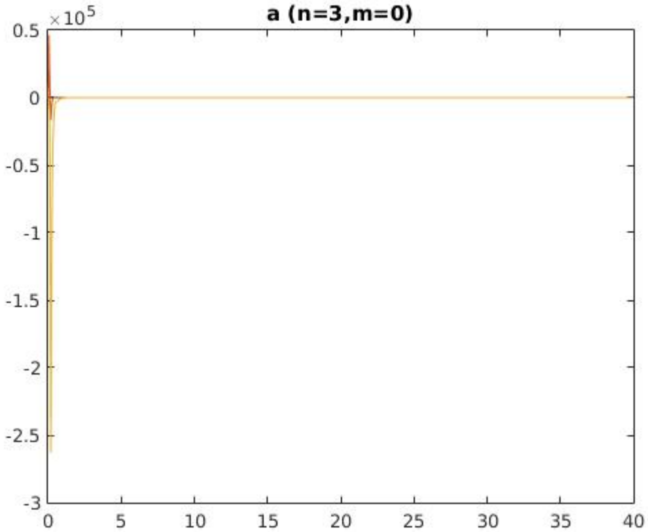

b (n=2,m=0)

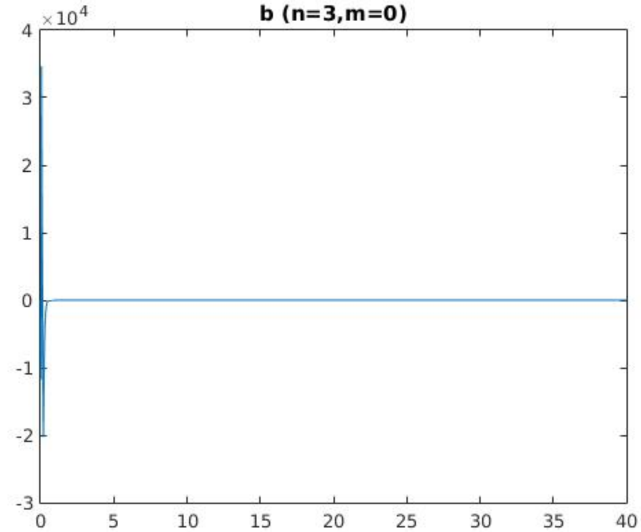

error (n=2 ,m=0)

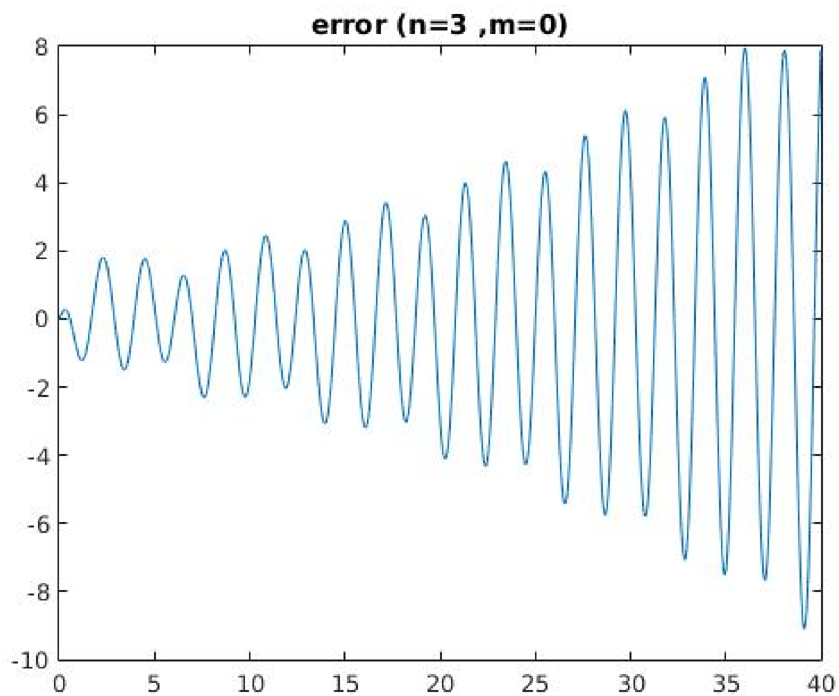

a (n=2,m=1)
200
150
100
50
0
-50
0 5 10 15 20 25 30 35 40

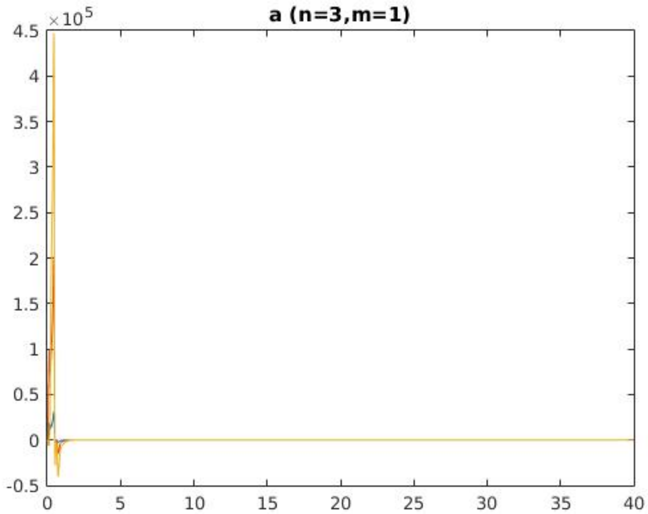
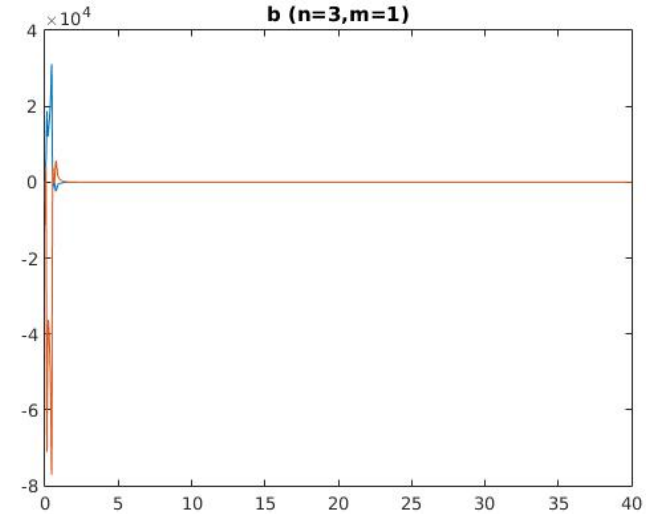
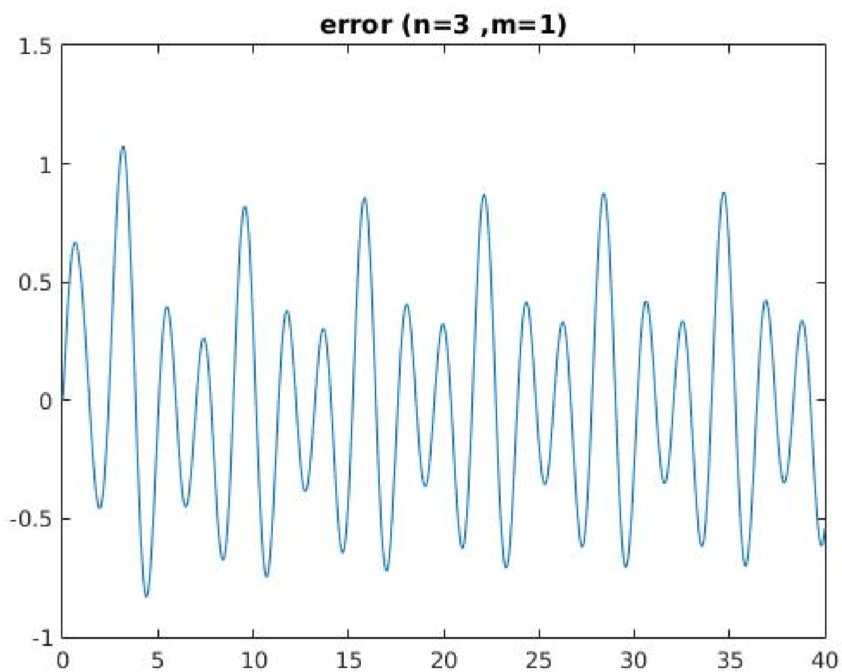

b (n=2,m=1)
20
10
0
-10
-20
-30
-40
0 5 10 15 20 25 30 35 40

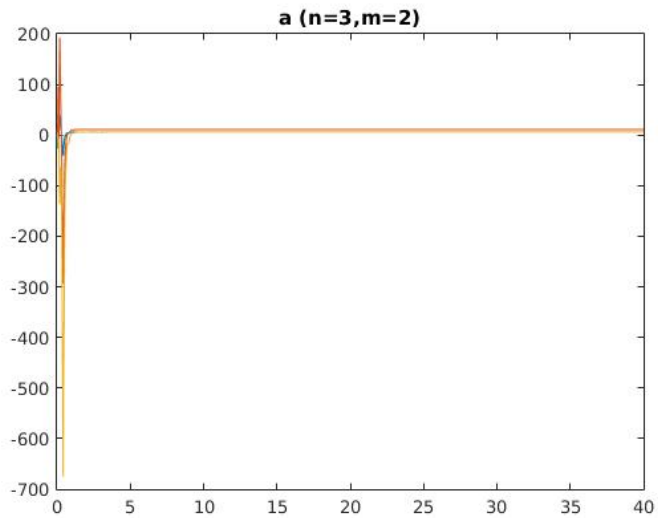

error (n=2 ,m=1)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0 5 10 15 20 25 30 35 40

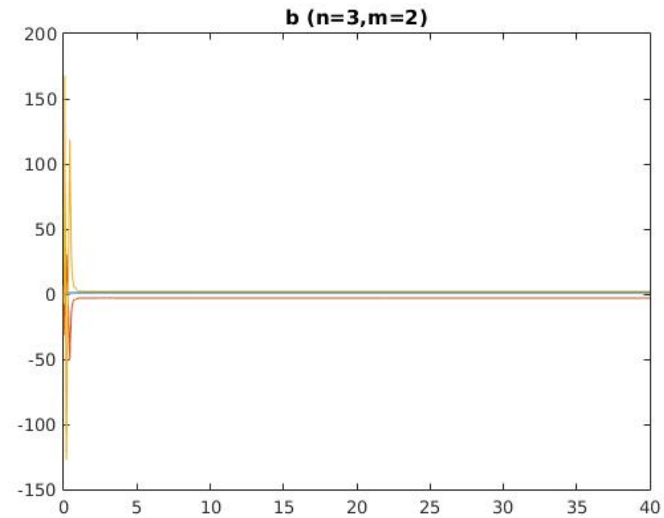

a (n=3,m=0)
×10^5

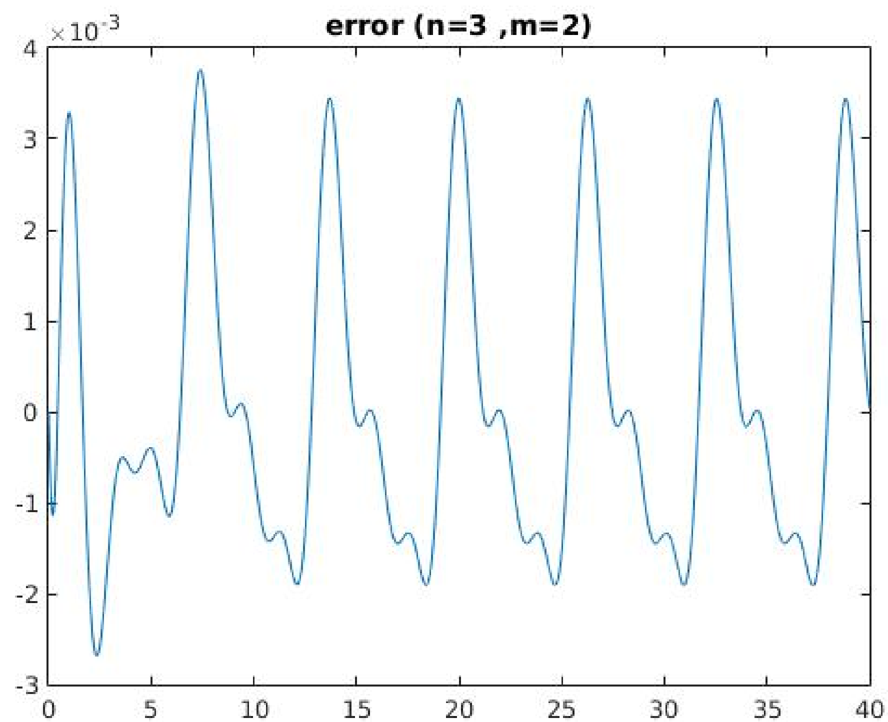

b (n=3,m=0)
×10^4

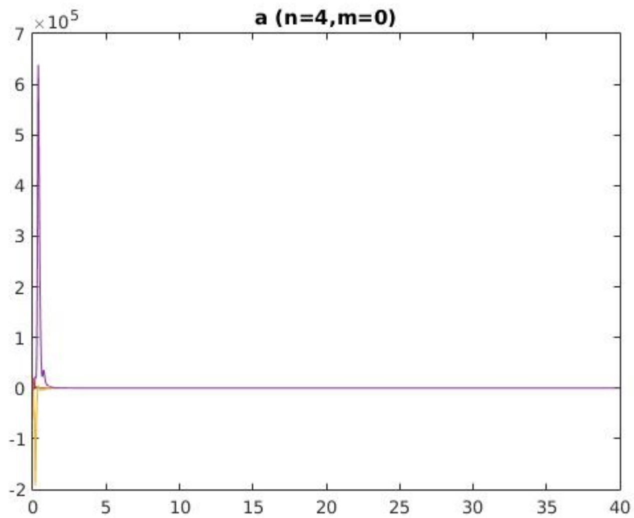

error (n=3 ,m=0)

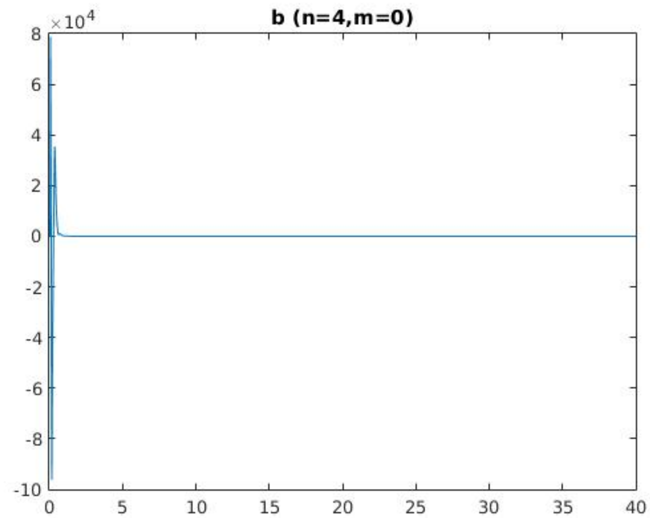

a (n=3,m=1)
×10^5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
5
10
15
20
25
30
35
40

b (n=3,m=1)
×10^4
4
2
0
-2
-4
-6
-8
0
5
10
15
20
25
30
35
40

error (n=3 ,m=1)
1.5
1
0.5
0
-0.5
-1
0
5
10
15
20
25
30
35
40

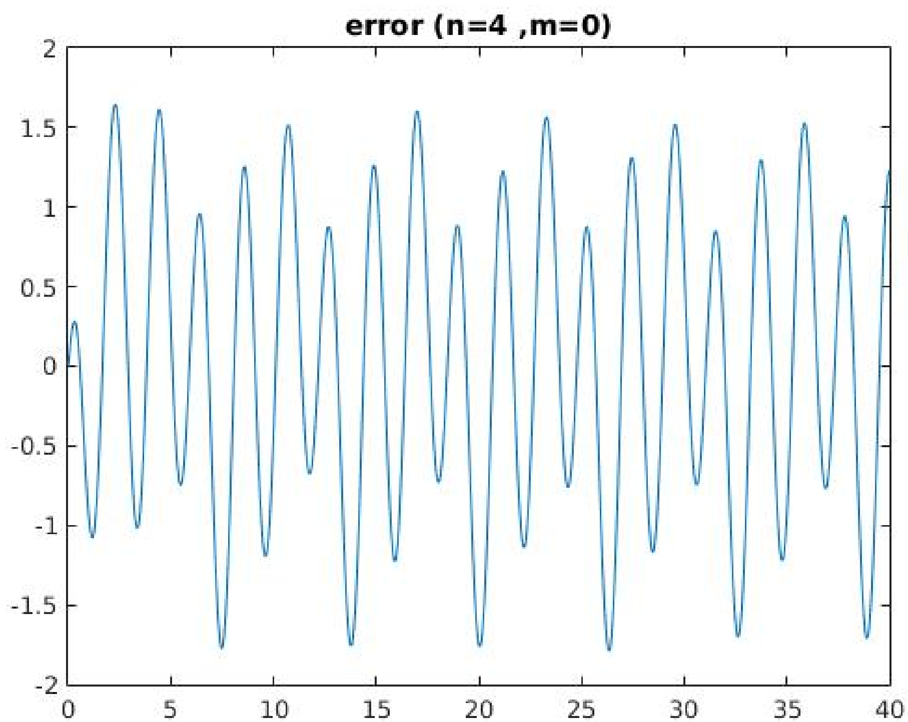

a (n=3,m=2)

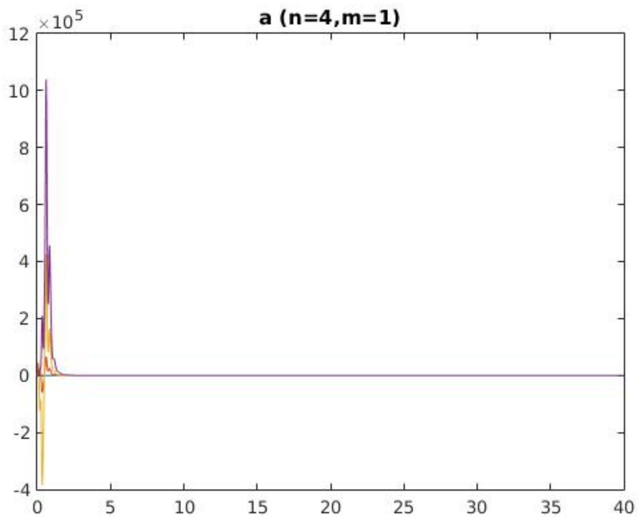

b (n=3,m=2)

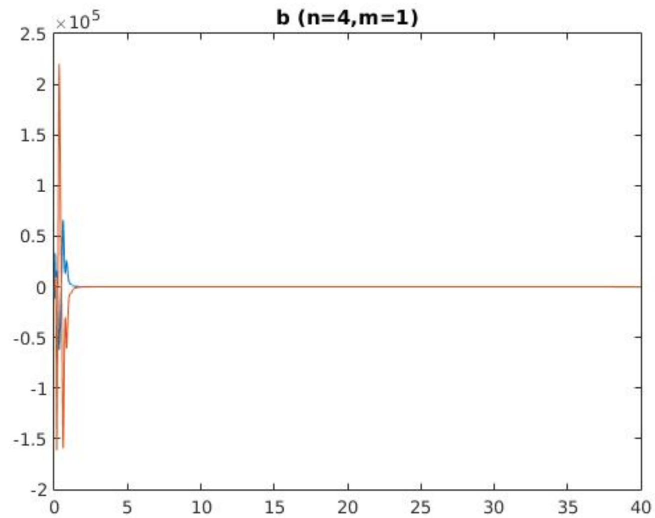

error (n=3 ,m=2)

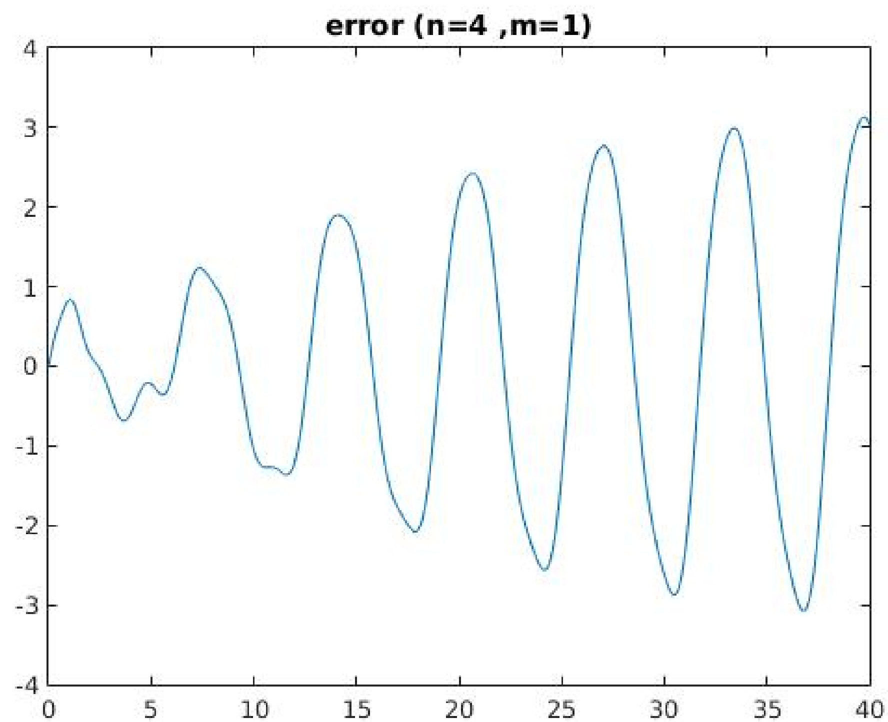

a (n=4,m=0)
×10^5
7
6
5
4
3
2
1
0
-1
-2
0
5
10
15
20
25
30
35
40

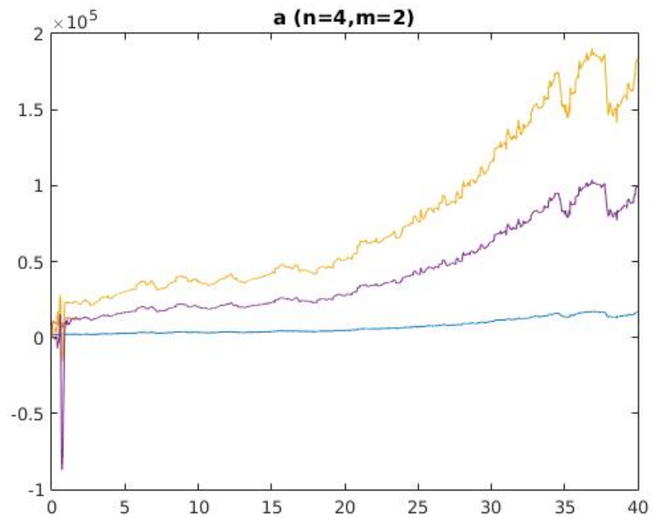

b (n=4,m=0)
×10^4
8
6
4
2
0
-2
-4
-6
-8
-10
0
5
10
15
20
25
30
35
40

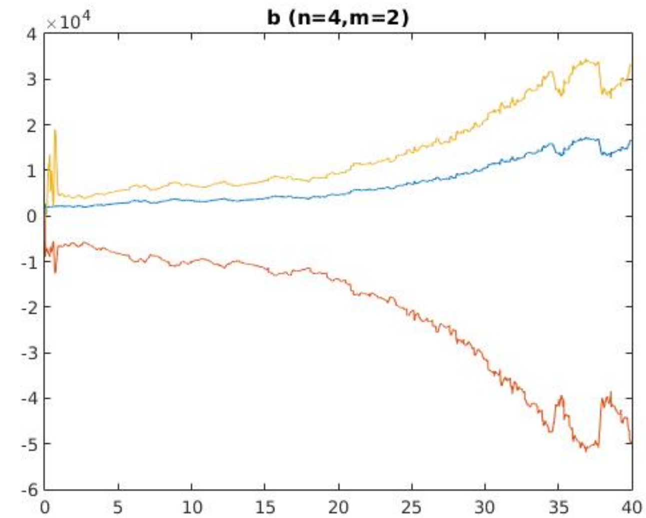

error (n=4 ,m=0)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5
10
15
20
25
30
35
40

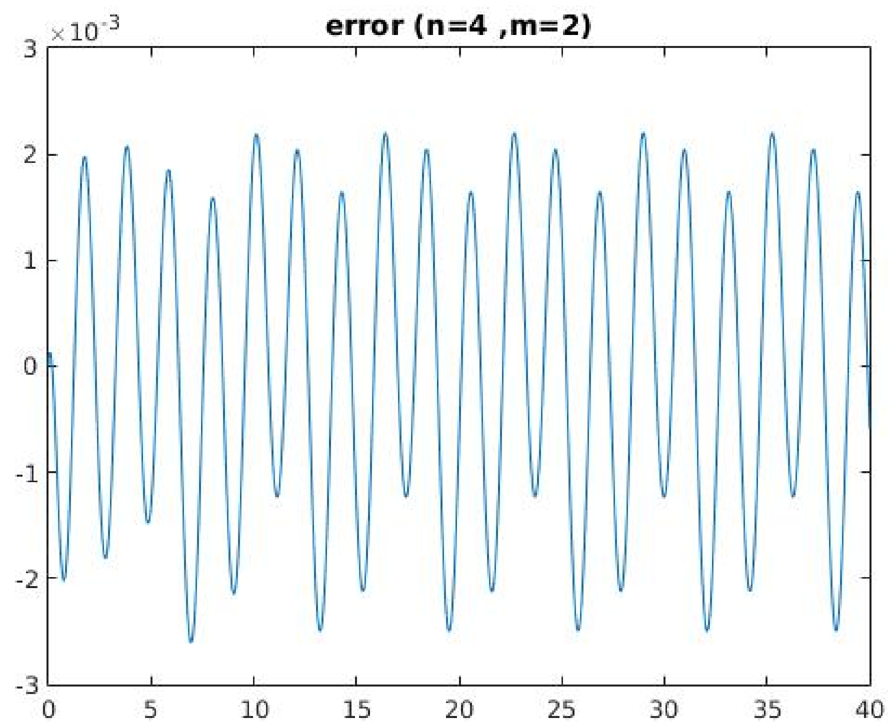

a (n=4,m=1)
×10^5

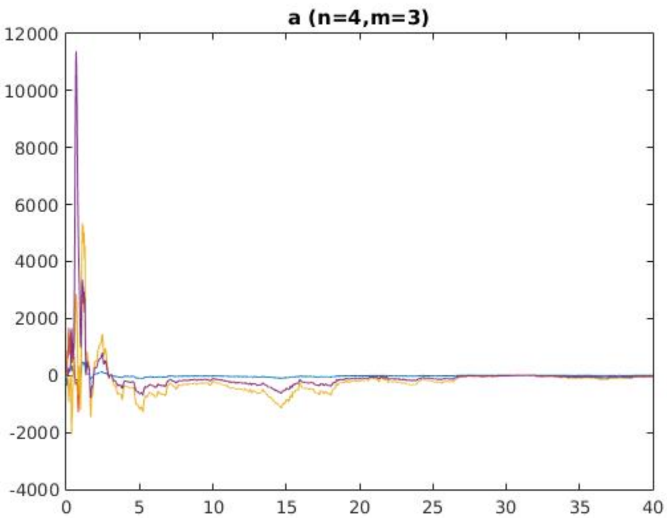

b (n=4,m=1)
×10^5

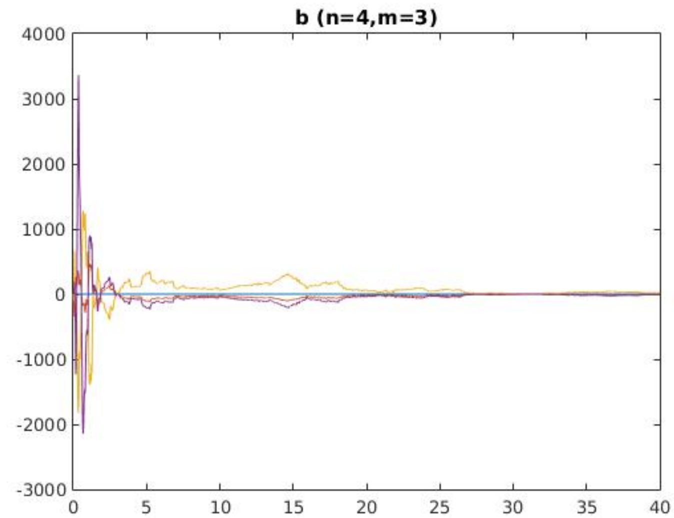

error (n=4 ,m=1)

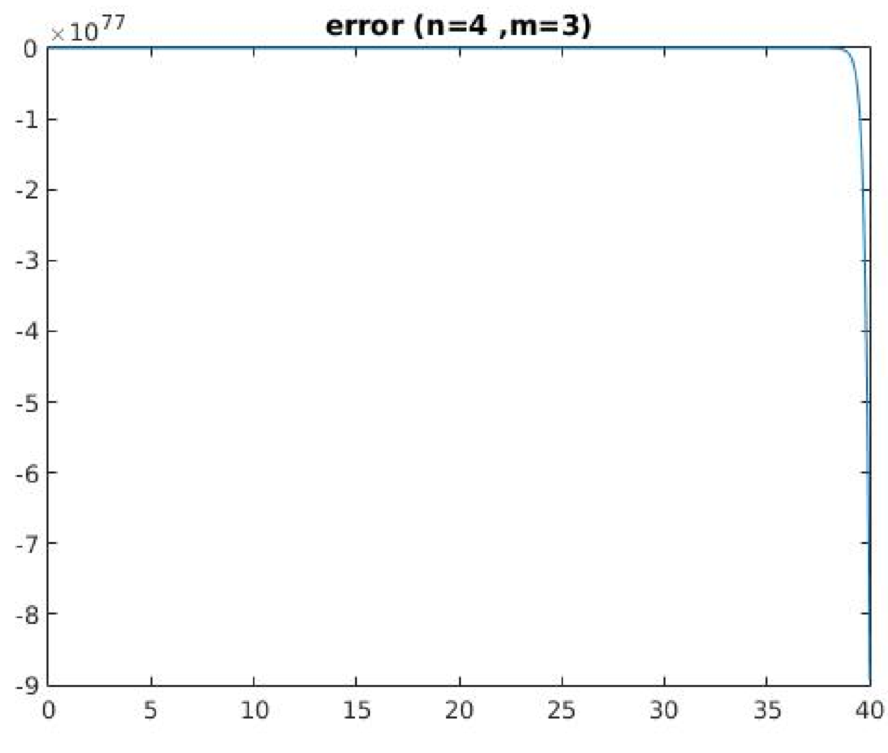

a (n=4,m=2)
×10^5
2
1.5
1
0.5
0
-0.5
-1
0
5
10
15
20
25
30
35
40


b (n=4,m=2)
×10^4
4
3
2
1
0
-1
-2
-3
-4
-5
-6
0
5
10
15
20
25
30
35
40


error (n=4 ,m=2)
×10^-3
3
2
1
0
-1
-2
-3
0
5
10
15
20
25
30
35
40


a (n=4,m=3)
12000
10000
8000
6000
4000
2000
0
-2000
-4000
0
5
10
15
20
25
30
35
40


b (n=4,m=3)
4000
3000
2000
1000
0
-1000
-2000
-3000
0
5
10
15
20
25
30
35
40


error (n=4 ,m=3)
×10^77
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
0
5
10
15
20
25
30
35
40

Συμπέρασματα

Δεδομένου των σφαλμάτων που παρουσιάζονται στα παραπάνω διαγράμματα, όμοια με την offline μέθοδο παρατηρούμε πως το μικρότερο σφάλμα της διαφοράς:

$$\textit{πραγματικής εξόδου με την εκτιμώμενη έξοδο}$$

το συναντάμε για τις τιμές

$$\boxed{n\ =\ 3\ \textit{και}\ m\ =\ 2}$$

η επιλογή του φίλτρου ήταν ίδια με αυτη που κάναμε στην offline

$$\Lambda(s) = s^3 + 4s^2 + 5s + 2$$

και προφανώς λαμβανοντας υπ όψιν το παραπάνω φίλτρο καταλήγουμε στις ίδιες τιμές παραμέτρων

$$\theta^T{}_\lambda = \begin{bmatrix} 6 & 11 & 6 & 6 & 1 & -3 & 2 \end{bmatrix}$$

αρα και στην ίδια διαφορική:

$$y^{(3)} + 6y^{(2)} + 11\dot{y} + 6y = u^{(2)} - 3\dot{u} + 2u$$

ακολουθεί η αξιολόγιση των μοντέλων...

Αξιολόγιση μοντέλου

Η αξιολόγηση ενός μοντέλου μπορεί να γίνει με διάφορους τρόπους και κριτήρια που υπάρχουν στις σημειώσεις του μαθήματος. Επιλέγουμε το κριτήριο του Akaine το οποίο περιγράφεται ως εξής:

$$\boxed{AIC = Nln\left(I(\theta)\right) + \rho\kappa}$$

Όπου N το πλήθος των δεδομένων ελέγχου, και ο αριθμός των παραμέτρων του συστήματος(n+m+1), ρ μια θετική σταθερα(επιλέχθκε ρ=2) και

$$I(\theta) = \frac{1}{N}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

περιπτώσεις	offline	online
μοντέλο (n=1,m=0)	$3.44 \cdot 10^6$	$0.201 \cdot 10^7$
μοντέλο (n=2,m=0)	$-0.07 \cdot 10^6$	$-0.0074 \cdot 10^7$
μοντέλο (n=2,m=1)	$-0.09 \cdot 10^6$	$-0.0076 \cdot 10^7$
μοντέλο (n=3,m=0)	$0.27 \cdot 10^6$	$0.0095 \cdot 10^7$
μοντέλο (n=3,m=1)	$-0.11 \cdot 10^6$	$-0.0065 \cdot 10^7$
μοντέλο (n=3,m=2)	$-0.92 \cdot 10^6$	$-0.0499 \cdot 10^7$
μοντέλο (n=4,m=0)	$-0.09 \cdot 10^6$	$-0.00058 \cdot 10^7$
μοντέλο (n=4,m=1)	$-0.12 \cdot 10^6$	$0.00412 \cdot 10^7$
μοντέλο (n=4,m=2)	$-0.79 \cdot 10^6$	$-0.0424 \cdot 10^7$
μοντέλο (n=4,m=3)	$0.017 \cdot 10^6$	$1.4122 \cdot 10^7$

Απο τον παραπάνω πίνακα επιβεβαιώνεται πως στην περίπτωση (n=3,m=2) εχουμε το ελάχιστο σφάλμα μοντέλου και για τις δύο μεθόδους καθώς και οτι η offline έχει μικρότερο σφάλμα μοντέλου απο την online.