



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης  
Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Ηλεκτρονικής και Υπολογιστών

Project στην προσωμοίωση και μοντελοποίηση  
δυναμικών συστημάτων

Εργασία του  
Φώτη Αλεξανδρίδη, ΑΕΜ: 9953  
faalexandr@ece.auth.gr

22 Ιουνίου 2023

# Περιεχόμενα

<b>1</b>	<b>Μέρος Α</b>	<b>2</b>
1.1	Θεωρητική Σκέψη . . . . .	2
1.2	Πειραματικός Έλεγχος . . . . .	2
<b>2</b>	<b>Μέρος Β</b>	<b>4</b>
2.1	Θεωρητική ανάλυση offline μεθόδου . . . . .	4
2.2	Αποτελέσματα offline μεθόδου . . . . .	5
2.3	Θεωρητική ανάλυση online μεθόδου . . . . .	7
2.4	Αποτελέσματα online μεθόδου . . . . .	8
<b>3</b>	<b>Μέρος Γ</b>	<b>10</b>
3.1	Σύγκριση αποτελεσμάτων offline και online μεθόδων . . . . .	10
	<b>Bibliography</b>	<b>13</b>

# Κεφάλαιο 1

## Μέρος Α

### 1.1 Θεωρητική Σκέψη

Από την θεωρία γνωρίζουμε ότι ένα σύστημα  $S$  είναι γραμμικό όταν για οποιεσδήποτε συναρτήσεις εισόδου  $u_1(t)$ ,  $u_2(t)$  και οποιεσδήποτε σταθερές  $a$ ,  $b$  ισχύει:

$$S\{au_1 + bu_2\} = aS\{u_1\} + bS\{u_2\} \quad (1.1)$$

Για να ελέγξουμε επομένως αν το άγνωστο σύστημα είναι γραμμικό, αρκεί να ελέγξουμε αν ισχύει η εξίσωση 1.1.

### 1.2 Πειραματικός Έλεγχος

Στον πειραματικό έλεγχο, πραγματοποιούνται δοκιμές για να αποφανθούμε αν το σύστημα είναι γραμμικό. Συγκεκριμένα σαν συναρτήσεις εισόδου και σταθερές έχουμε τις:

$$u_1(t) = \cos(t) \quad a_1 = 2 \quad (1.2)$$

$$u_2(t) = \cos(2t) \quad a_2 = 6 \quad (1.3)$$

$$u_3(t) = \cos(3t) \quad a_3 = 8 \quad (1.4)$$

$$u_4(t) = \cos(4t) \quad a_4 = -6 \quad (1.5)$$

$$u_5(t) = \cos(5t) \quad a_5 = -10 \quad (1.6)$$

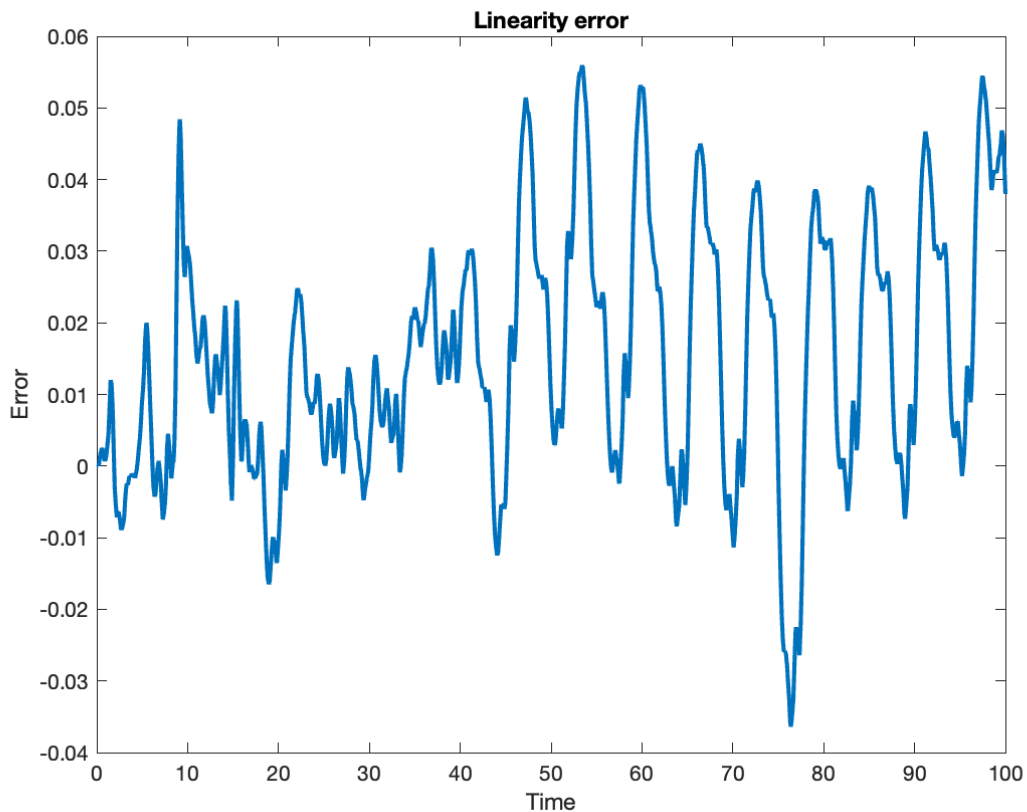
$$u_6(t) = \cos(6t) \quad a_6 = 12 \quad (1.7)$$

Στο παρακάτω γράφημα φαίνεται το σφάλμα γραμμικότητας, που έχουμε ορίσει σύμφωνα με τις παρακάτω εξισώσεις:

$$error = constructed - combined \quad (1.8)$$

$$constructed = a_1 S\{u_1\} + a_2 S\{u_2\} + a_3 S\{u_3\} + a_4 S\{u_4\} + a_5 S\{u_5\} + a_6 S\{u_6\} \quad (1.9)$$

$$combined = S\{a_1 u_1 + a_2 u_2 + a_3 u_3 + a_4 u_4 + a_5 u_5 + a_6 u_6\} \quad (1.10)$$



Από το γράφημα βλέπουμε ότι υπάρχει σφάλμα, οπότε το σύστημά μας δεν είναι αυστηρά γραμμικό. Παρ' όλα αυτά, βλέπουμε ότι το σφάλμα είναι μικρό (της τάξης του 0.05) σε σχέση με την είσοδό μας και φραγμένο, οπότε μπορούμε να πούμε ότι το σύστημα μας είναι ψευτογραμμικό και, στα πλαίσια των επόμενων ερωτημάτων, να το αντιμετωπίσουμε ως γραμμικό.

Αυτό το σφάλμα μπορεί να προέρχεται από ατελής αριθμητικές μεθόδους (π.χ. σφάλμα διακριτής ολοκλήρωσης).

# Κεφάλαιο 2

## Μέρος Β

### 2.1 Θεωρητική ανάλυση offline μεθόδου

Για την μέθοδο μη πραγματικού χρόνου (offline) αποφασίστηκε να χρησιμοποιηθεί ένας offline εκτιμητής ελαχίστων τετραγώνων.

Εφόσον έχουμε αποφανθεί από το προηγούμενο μέρος της εργασίας ότι το σύστημα είναι ψευτογραμμικό, θα κάνουμε γραμμική μοντελοποίηση. Επομένως, πέραν από τις παραμέτρους του συστήματος, είναι υπο διερεύνηση τόσο η τάξη των παραγώγων της εξόδου, όσο και της εισόδου. Επίσης, λόγω της μεθοδολογίας που ακολουθείται (φιλτράρισμα παραγώγων εξόδου), παράμετροι προς εύρεση είναι και οι παράμετροι του φίλτρου. Για λόγους απλότητας και ελαχιστοποίησης των μεταβλητών, επιλέγεται φίλτρο με πολλαπλή πραγματική αρνητική ρίζα (γνωρίζουμε πως ήδη είναι ευσταθές).

Έτσι η μοντελοποίησή μας είναι η εξής (αντλημένη από τις σημειώσεις [1]):

$$z = \partial^{*T} \zeta \quad (2.1)$$

$$z = \frac{1}{\Lambda(s)} y^{(n)} = \frac{s^n}{\Lambda(s)} y \quad (2.2)$$

$$\partial^* = [a_1 \ a_2 \ \dots \ a_{n-1} \ a_n \ b_0 \ b_1 \ \dots \ b_{m-1} \ b_m]^T \quad (2.3)$$

$$\zeta = \left[ -\frac{\Delta_{n-1}^T(s)}{\Lambda(s)} y \ \frac{\Delta_m^T(s)}{\Lambda(s)} u \right] \quad (2.4)$$

$$\Lambda(s) = s^n + \hat{\lambda}_1 s^{n-1} + \dots + \hat{\lambda}_{n-1} s + \hat{\lambda}_n = (s + p)^n \quad (2.5)$$

$$\Delta_i(s) = [s^i \ s^{i-1} \ \dots \ 1]^T \quad (2.6)$$

$$0 < m < n \quad (2.7)$$

όπου ο περιορισμός  $m < n$  προσδιορίζει ότι έχουμε ένα πραγματικό σύστημα που δεν γίνεται να δέχεται παράγωγο εισόδου μεγαλύτερης τάξης από την παράγωγο εξόδου (πηγάει από τον ορισμό της αιτιατότητας).

και για κάθε συνδυασμό των  $n, m, p$  εξάγουμε το σφάλμα μοντελοποίησης που ορίζεται ως εξής:

$$modelling\_error = y_{true} - y_{simulated} \quad (2.8)$$

Όπου  $y_{true}$  είναι η έξοδος του συστήματος που μας δίνεται και  $y_{simulated}$  είναι η έξοδος του προσωμοιωμένου γραμμικού συστήματος.

Κριτήριο για την επιλογή μοντέλου είναι το σφάλμα μοντελοποίησης να είναι το δυνατόν μικρότερο.

## 2.2 Αποτελέσματα offline μεθόδου

Για τις δοκιμές, ορίστηκε :

$$2 \leq n \leq 7 \quad (2.9)$$

$$1 \leq m \leq n \quad (2.10)$$

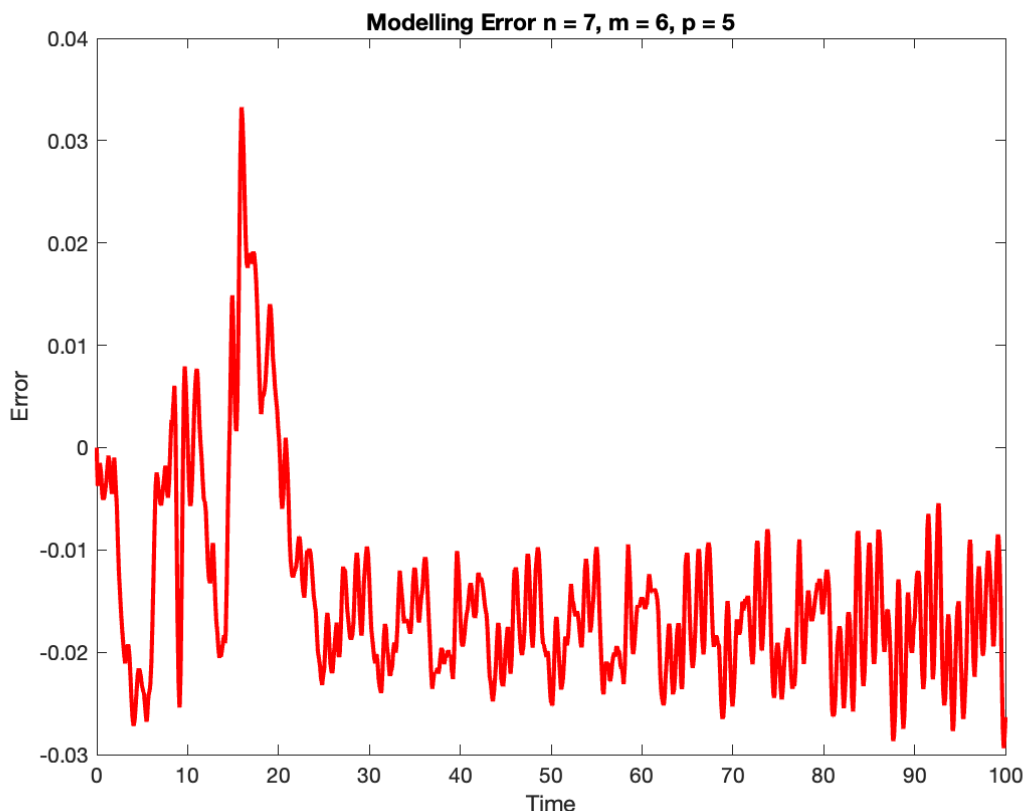
$$p = 5 \quad (2.11)$$

σύνολο δηλαδή δοκιμάστηκαν 21 συστήματα. Ως είσοδος χρησιμοποιήθηκε η ίδια συνδυασμένη είσοδος του μέρους Α, δηλαδή :

$$u(t) = 2\cos(t) + 6\cos(2t) + 8\cos(3t) - 6\cos(4t) - 10\cos(5t) + 12\cos(6t) \quad (2.12)$$

για το χρονικό διάστημα  $0 \leq t \leq 100$  με βήμα 0.001. Το (όχι ευδιάκριτο) διάγραμμα με τα σφάλματα μοντελοποίησης φαίνεται παρακάτω :





και μέσο τετραγωνικό σφάλμα 0.000296.

## 2.3 Θεωρητική ανάλυση online μεθόδου

Στην online μέθοδο χρησιμοποιήθηκε αναδρομικός εκτιμητής ελαχίστων τετραγώνων βασισμένος στις σημειώσεις του μαθήματος [1].

Πάλι κάνουμε γραμμική μοντελοποίηση. Αυτή την φορά, υπό διερεύνηση είναι η τάξη των παραγώγων της εξόδου και η τάξη των παραγώγων της εισόδου. Ακόμη, υπάρχει μια ακόμη παράμετρος (αρχική τιμή ενός πίνακα) η οποία όμως δεν μας απασχολεί πολύ).

Έτσι, έχουμε την ακόλουθη μοντελοποίηση [1]:

$$y = \partial^{*T} \phi \quad (2.17)$$

$$\hat{y} = \hat{\partial}^{*T} \phi \quad (2.18)$$

$$\tilde{y} = y - \hat{y} \quad (2.19)$$

$$\dot{\hat{\partial}} = P \tilde{y} \phi \quad (2.20)$$

$$\dot{P} = -P \phi \phi^T P, \quad P(0) = I \quad (2.21)$$

$$\phi = \begin{bmatrix} y(t-1) & y(t-2) & \dots & y(t-n) & u(t) & u(t-1) & \dots & u(t-m) \end{bmatrix}^T \quad (2.22)$$

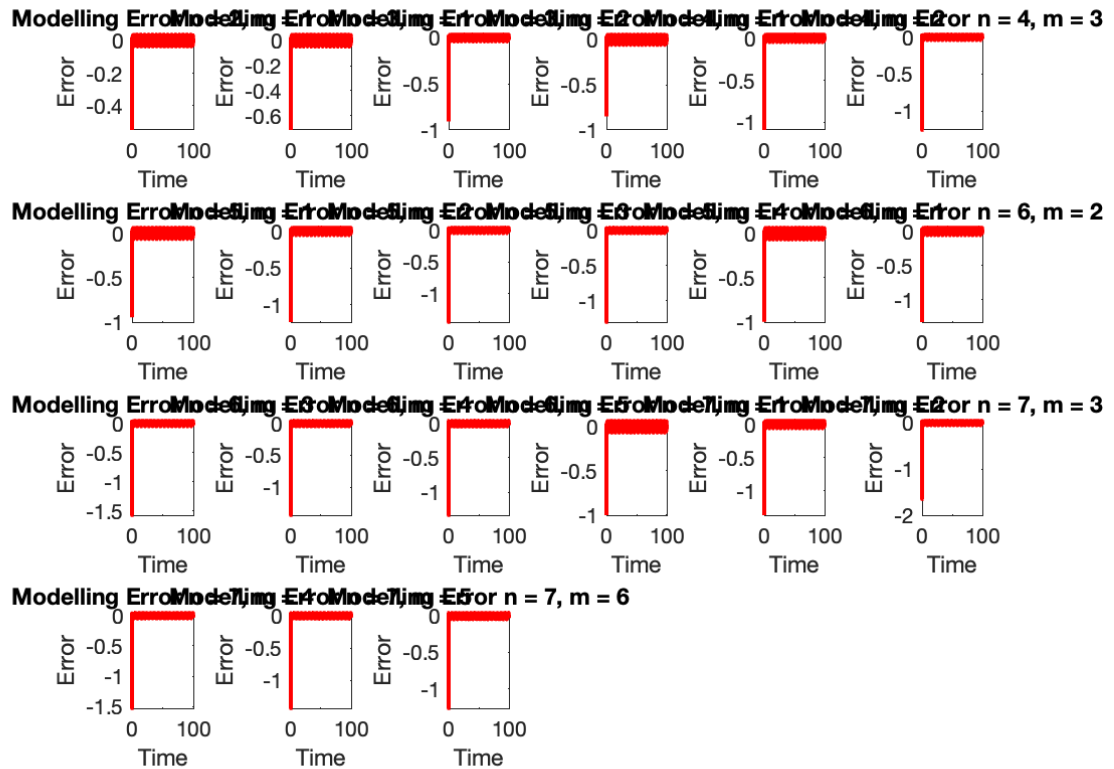


όπου επιλέξαμε ως αρχική τιμή για τον πίνακα  $P$  τον μοναδιαίο πίνακα (θετικά ορισμένος).

Το σφάλμα μοντελοποίησης ορίζεται όπως και στην offline μέθοδο. Το κριτήριο επιλογής είναι να έχουμε κατά το δυνατόν χαμηλότερο σφάλμα μοντελοποίησης (ανάλογα την εφαρμογή μπορεί να στοχεύσουμε μόνο στο τελικό σφάλμα, μετά δηλαδή το training των παραμέτρων).

## 2.4 Αποτελέσματα online μεθόδου

Για τις δοκιμές, ισχύουν οι ίδιες τιμές για τα  $m, n$ , καθώς και για την είσοδο και το χρονικό διάστημα. Το (επίσης όχι ευδιάκριτο) διάγραμμα σφαλμάτων μοντελοποίησης φαίνεται παρακάτω:



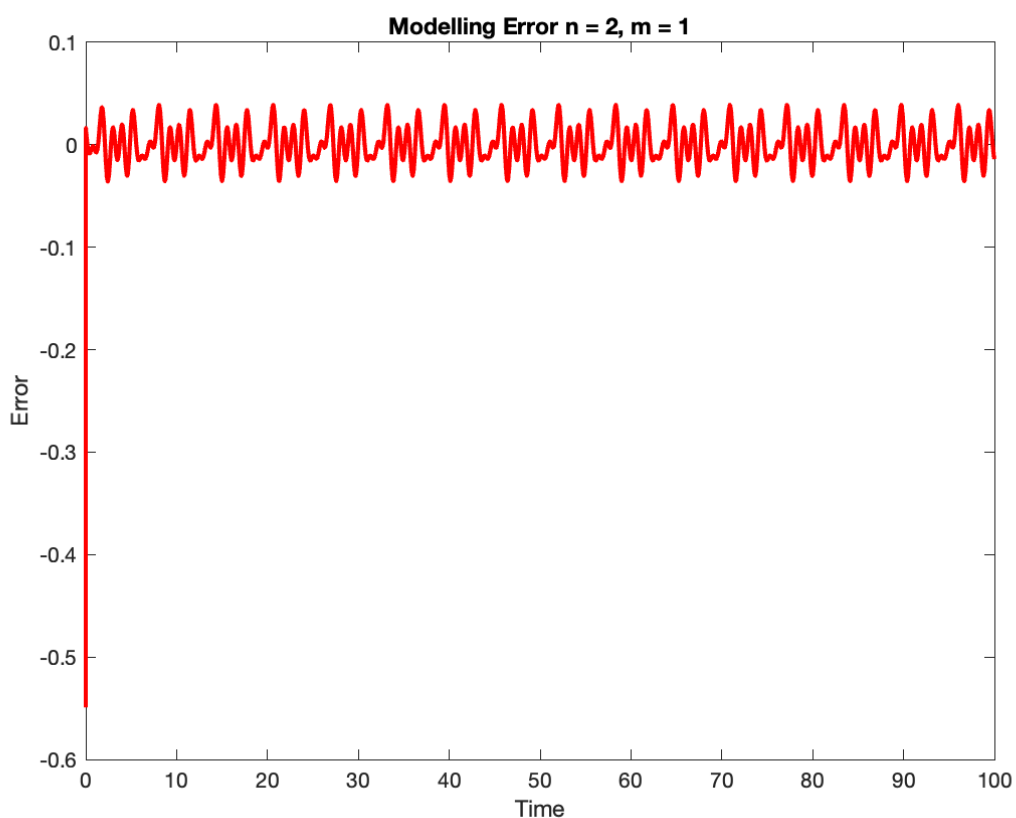
Παρατηρώντας τα διαγράμματα, και λαμβάνοντας ως κριτήριο επιλογής το μέσο τετραγωνικό σφάλμα (MSE) τόσο για όλη την διάρκεια όσο και για το τελευταίο 25% του χρονικού διαστήματος της προσομοίωσης, λαμβάνουμε ότι η καλύτερη μοντελοποίηση είναι η:

$$n = 2 \quad (2.23)$$

$$m = 1 \quad (2.24)$$

$$\partial^* = \begin{bmatrix} 0.5120 \\ 0.4873 \\ 0.0458 \\ -0.0457 \end{bmatrix} \quad (2.25)$$

με σφάλμα μοντελοποίησης:



και μέσο τετραγωνικό σφάλμα 0.000327 στο τελευταίο 25% της προσομοίωσης. Να σημειωθεί ότι και άλλες μοντελοποιήσεις είχαν αρκετά χαμηλό σφάλμα.

# Κεφάλαιο 3

## Μέρος Γ

### 3.1 Σύγκριση αποτελεσμάτων offline και online μεθόδων

Αρχικά, παρατηρούμε ότι τα βέλτιστα μοντέλα και με τους δύο τρόπους έχουν κοντινή αόδοση με πολύ χαμηλό τελικό μέσο τετραγωνικό σφάλμα. Το μη μηδενικό σφάλμα οφείλεται τόσο στην περιορισμένη ποσότητα δειγμάτων (θυμόμαστε από την θεωρία ότι η σύγκλιση είναι εκθετική) όσο και στην μη αυστηρή γραμμικότητα του συστήματος.

Σε γενικές γραμμές, η κάθε μέθοδος είχε τα δυνατά της σημεία. Για την offline μέθοδο, επιτεύχθηκε ελάχιστα χαμηλότερο μέσο τετραγωνικό σφάλμα (0.000296 έναντι 0.000327 που αν συμπεριλαμβάναμε και την αρχή που το σφάλμα ήταν μεγαλύτερο θα ήταν ακόμα μεγαλύτερο), και επίσης τόσο η ανάπτυξή της όσο και η λειτουργία της απαίτησε χαμηλότερη πολυπλοκότητα και υπολογιστικούς πόρους. Επίσης, στην αρχή λειτουργίας της online μεθόδου, λόγω κακής πρώτης “μαντεψιάς” των αρχικών τιμών των παραμέτρων, το σφάλμα δύναται (και συνήθως είναι) να είναι μεγαλύτερο από το αντίστοιχο αρχικό σφάλμα της offline μεθόδου.

Για την online μέθοδο, το πλεονέκτημά της ήταν ότι για διαφορετικές μοντελοποιήσεις έδινε παρόμοια και πολύ ικανοποιητικά αποτελέσματα. Παρακάτω παρατίθεται τμήμα της εξόδου του προγράμματος `main.m`:

```
Offline estimator:  n = 2, m = 1, p = 5, MSE =  
0.465166  
Offline estimator:  n = 3, m = 1, p = 5, MSE =  
1460896609404957031631110...  
0996642448088910427862703866808107008.000000  
Offline estimator:  n = 3, m = 2, p = 5, MSE = 0.001081  
Offline estimator:  n = 4, m = 1, p = 5, MSE = 1.798846  
Offline estimator:  n = 4, m = 2, p = 5, MSE = 0.000684  
Offline estimator:  n = 4, m = 3, p = 5, MSE = 0.000343  
Offline estimator:  n = 5, m = 1, p = 5, MSE =  
16192.650386  
Offline estimator:  n = 5, m = 2, p = 5, MSE = NaN  
Offline estimator:  n = 5, m = 3, p = 5, MSE = Inf
```

```

Offline estimator:  n = 5, m = 4, p = 5, MSE =
0.000409
Offline estimator:  n = 6, m = 1, p = 5, MSE =
3425600587258959324...
1907294600429568.000000
Offline estimator:  n = 6, m = 2, p = 5, MSE = Inf
Offline estimator:  n = 6, m = 3, p = 5, MSE = NaN
Offline estimator:  n = 6, m = 4, p = 5, MSE =
118464387978696335235634893147209721765124226697...
28900739813243493281038728694687181827997696.000000
Offline estimator:  n = 6, m = 5, p = 5, MSE =
0.000306
Offline estimator:  n = 7, m = 1, p = 5, MSE =
570483254860062046000690863043213487658961...
8835918003080417035496759004575164344041472.000000
Offline estimator:  n = 7, m = 2, p = 5, MSE = NaN
Offline estimator:  n = 7, m = 3, p = 5, MSE = NaN
Offline estimator:  n = 7, m = 4, p = 5, MSE = NaN
Offline estimator:  n = 7, m = 5, p = 5, MSE =
5976075072179213722...
122084217498975830454304768.000000
Online estimator:  n = 2, m = 1, MSE = 0.000327
Online estimator:  n = 3, m = 1, MSE = 0.000549
Online estimator:  n = 3, m = 2, MSE = 0.000358
Online estimator:  n = 4, m = 1, MSE = 0.000779
Online estimator:  n = 4, m = 2, MSE = 0.000519
Online estimator:  n = 4, m = 3, MSE = 0.000383
Online estimator:  n = 5, m = 1, MSE = 0.000967
Online estimator:  n = 5, m = 2, MSE = 0.000664
Online estimator:  n = 5, m = 3, MSE = 0.000502
Online estimator:  n = 5, m = 4, MSE = 0.000436
Online estimator:  n = 6, m = 1, MSE = 0.001074
Online estimator:  n = 6, m = 2, MSE = 0.000766
Online estimator:  n = 6, m = 3, MSE = 0.000598
Online estimator:  n = 6, m = 4, MSE = 0.000530
Online estimator:  n = 6, m = 5, MSE = 0.000504
Online estimator:  n = 7, m = 1, MSE = 0.001088
Online estimator:  n = 7, m = 2, MSE = 0.000812
Online estimator:  n = 7, m = 3, MSE = 0.000657
Online estimator:  n = 7, m = 4, MSE = 0.000597
Online estimator:  n = 7, m = 5, MSE = 0.000577
Online estimator:  n = 7, m = 6, MSE = 0.000570

```

όπου παρατηρούμε ότι η online μέθοδος είναι πολύ πιο consistent και ανεκτική από την offline μέθοδο, και φαίνεται να επηρεάζεται ελάχιστα από τις επιλογές των  $m, n$ , ειδικά όταν στην offline μέθοδο λάθος επιλογή οδηγεί μέχρι και σε μη σύγκλιση. Επομένως, σε περιβάλλοντα όπου τα  $m, n$  θα είχαν περιορισμούς (π.χ. λόγω αισθητήρων ή υπολογιστικών διατάξεων) θα επιλέγαμε σίγουρα μια online μορφή, αν δεν μπορούσαμε

να υλοποιήσουμε την βέλτιστη offline.

Εν τέλει, η βέλτιστη επιλογή ανάγεται στις παρακάτω ερωτήσεις:

- Υπάρχουν περιορισμοί για την μέτρηση των  $n, m$ . Μας επιτρέπουν να έχουμε την βέλτιστη offline μορφή.
- Έχουμε προϋπάρχοντα δεδομένα εισόδου-εξόδου για να χρησιμοποιήσουμε έναν offline εκτιμητή.
- Είναι κρίσιμο το σφάλμα να παραμένει το δυνατόν χαμηλότερο ακόμα και στα πολύ αρχικά βήματα εκτέλεσης (δηλαδή καθ' όλη την διάρκεια εκτέλεσης).
- Χρειαζόμαστε προσαρμοστικότητα. Δηλαδή αλλάζουν οι παράμετροι κατά την διάρκεια λειτουργίας.

Να σημειωθεί πως στα πλαίσια όλης της εργασίας, δεν λήφθηκε πουθενά υπ' όψιν η παρουσία και η φύση του θορύβου, για τον οποίο θα έπρεπε να γίνουν εξαρχής πολύ πιο ενδελεχής αναλύσεις, εκτός της σκοπιάς της εργασίας.

# Bibliography

- [1] G. A. Rovithakis, *MODELLING AND SIMULATION OF SYSTEMS*, 2004.