



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Θετικών Επιστημών
Τμήμα Μαθηματικών
Θεωρητική Πληροφορική και Θεωρία Συστημάτων και Ελέγχου

Εργασία για το μάθημα

Εύρωστος Έλεγχος

Νίκος Β. Βασίλας

Θεσσαλονίκη, Μάιος 2017

Περιεχόμενα

1	Εισαγωγικά στο MATLAB	4
1.1	Συναρτήσεις μεταφοράς	4
1.2	Απόκριση συστήματος	7
1.3	Επίλυση συστημάτων βέλτιστου ελέγχου	9
2	Ευστάθεια	12
2.1	Ευστάθεια κατά Lyapunov	12
2.2	Ευστάθεια αυτόνομου γραμμικού αναλλοίωτου συστήματος	15
2.3	Ευστάθεια γραμμικού αναλλοίωτου συστήματος ελέγχου	16
3	Αβεβαιότητες	18
3.1	Παράδειγμα αβέβαιου συστήματος	20
4	Μέθοδοι LQG εγγυημένου κόστους	25
4.1	Συστήματα με αβεβαιότητα στον πίνακα κατάστασης	27
4.1.1	Φράγμα με βάση τις ιδιοτιμές	27
4.1.2	Γραμμικό φράγμα	27
4.2	Συστήματα με αβεβαιότητα στον πίνακα κατάστασης και στον πίνακα εισόδου	34
4.2.1	Αβεβαιότητες που ικανοποιούν τις συνθήκες προσαρμογής	34
4.2.2	Γενική περίπτωση	39
4.3	Ελαχιστοποίηση του εγγυημένου κόστους	45
5	Εύρωστη τοποθέτηση πόλων	53
5.1	Αλγόριθμος εύρωστης τοποθέτησης πόλων	57
5.1.1	Βήμα 1	59
5.1.2	Βήμα 2	61
5.1.3	Βήμα 3	62
	Βιβλιογραφία	71

Κατάλογος Σχημάτων

1.1	Διάγραμμα πόλων-μηδενικών, (ο μηδενικά, x πόλοι)	6
1.2	Μοναδιαία βηματική απόκριση	8
1.3	Κρουστική απόκριση	8
1.4	Απόκριση σε τετραγωνικό σήμα εισόδου	10
1.5	Μοναδιαία βηματική απόκριση βέλτιστου τετραγωνικού ελεγκτή	11
2.1	Ευσταθές σημείο ισορροπίας	13
2.2	Ασυμπτωτικά ευσταθές σημείο ισορροπίας	13
2.3	Ασταθές σημείο ισορροπίας	13
3.1	Έλεγχος ευστάθειας συναρτήσει της μεταβολής των αβεβαιοτήτων r_1, r_2	22
3.2	Μοναδιαία βηματική απόκριση ονομαστικού και αβέβαιου συστήματος	23
4.1	Αρχικό σύστημα χωρίς αβεβαιότητες	30
4.2	Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1$	31
4.3	Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1$	32
4.4	Αρχικό σύστημα χωρίς αβεβαιότητες	37
4.5	Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, p_1 = 1$	38
4.6	Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, p_1 = -0.5$	39
4.7	Αρχικό σύστημα χωρίς αβεβαιότητες	42
4.8	Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, p_1 = 1$	43
4.9	Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, p_1 = -1$	44
4.10	Αρχικό σύστημα χωρίς αβεβαιότητες	48
4.11	Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, p_1 = 1$	49
4.12	Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, p_1 = -1$	50
5.1	Ονομαστικό σύστημα, $r_1 = 0, r_2 = 0, a = 1$	66
5.2	Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, a = 1$	67
5.3	Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, a = 1$	68

Κατάλογος Πινάκων

5.1	Πραγματικά μέρη ιδιοτιμών κλειστού συστήματος για διάφορες αβεβαιότητες και $a = 1$	67
5.2	Πραγματικά μέρη ιδιοτιμών κλειστού συστήματος για διάφορες αβεβαιότητες και $a = 3$	68
5.3	Πραγματικά μέρη ιδιοτιμών κλειστού συστήματος για διάφορες αβεβαιότητες και $a = 10$	69

Κεφάλαιο 1

Εισαγωγικά στο MATLAB

Για να κατανοήσουμε και να ελέγξουμε διάφορα πολύπλοκα συστήματα πρέπει να καταφύγουμε σε κάποιο μαθηματικό μοντέλο των συστημάτων αυτών. Η σύνθεση του μαθηματικού μοντέλου προκύπτει από τις διαφορικές εξισώσεις που περιγράφουν τη δυναμική συμπεριφορά του συστήματος. Τα περισσότερα συστήματα παρουσιάζουν μη-γραμμική συμπεριφορά. Η προσέγγιση που ακολουθούμε στα συστήματα αυτομάτου ελέγχου περιλαμβάνει τη γραμμικοποίηση των μη-γραμμικών διαφορικών εξισώσεων γύρω από ένα σημείο ισορροπίας και έτσι λαμβάνεται το αντίστοιχο προσεγγιστικό γραμμικό μοντέλο. Στη συνέχεια χρησιμοποιείται ο μετασχηματισμός Laplace όπου μετατρέπουμε τις διαφορικές εξισώσεις, από το πεδίο του χρόνου, σε αλγεβρικές, στο πεδίο της συχνότητας, που λύνονται πολύ πιο εύκολα. Με τη βοήθεια του μετασχηματισμού καταλήγουμε σε μία σχέση εισόδου-εξόδου του συστήματος, που καλείται συνάρτηση μεταφοράς. Η συνάρτηση μεταφοράς αποτελεί ένα μέσο για τη μελέτη της συμπεριφοράς του συστήματος για διάφορα σήματα εισόδου που επιδρούν στο σύστημα. Για περισσότερες πληροφορίες για τα θέματα του αυτομάτου ελέγχου προτείνονται τα βιβλία [4] και [9].

1.1 Συναρτήσεις μεταφοράς

Η συνάρτηση μεταφοράς είναι της μορφής

$$G(s) = \frac{Y(s)}{U(s)},$$

όπου $Y(s)$ και $U(s)$ είναι οι μετασχηματισμοί Laplace των χρονικών συναρτήσεων εξόδου και εισόδου αντίστοιχα. Γενικά τα συστήματα ελέγχου είναι ιδιαίτερα πολύπλοκα, για και η μελέτη τους γίνεται με εξειδικευμένα λογισμικά, όπως το MATLAB. Στη συνέχεια θα παρουσιάσουμε βασικές εντολές του MATLAB για τη δημιουργία, την ανάλυση και την παρουσίαση συστημάτων ελέγχου. Αυτό θα γίνει με τη μορφή παραδειγμάτων και την παράθεση του αντίστοιχου κώδικα και την επεξήγηση αυτού.

Η δημιουργία συνάρτησης μεταφοράς στο MATLAB γίνεται με την εντολή `tf` και συντάσσεται

$$\text{sys} = \text{tf}(\text{num}, \text{den})$$

όπου οι είσοδοι της συνάρτησης είναι οι πίνακες `num` και `den`, που δηλώνουν τα πολυώνυμα του αριθμητή και παρανομαστή αντίστοιχα και η έξοδος `sys` είναι η συνάρτηση μεταφοράς. Οι εντολές για τη δήλωση των παραπάνω στο MATLAB φαίνονται παρακάτω.

```
1 num = [1, 2];
2 den = [1, 3, 5];
3 sys = tf(num, den);
```

Μία άλλη χρήσιμη δυνατότητα είναι η γρήγορη εύρεση των μηδενικών και των πόλων του συστήματος. Μηδενικά ονομάζονται οι όροι που μηδενίζουν τον αριθμητή της συνάρτησης μεταφοράς και πόλοι οι όροι που μηδενίζουν τον παρανομαστή. Είναι γνωστό ότι αν πόλοι βρίσκονται στο αριστερό ημιεπίπεδο τότε το σύστημα είναι ευσταθές. Μιας και ασχολούμαστε με γραμμικά συστήματα, η ευστάθεια είναι του συστήματος είναι ανεξάρτητη της διέγερσης. Με την εντολή `pzmap(sys)` θα δημιουργηθεί διάγραμμα με τον άξονα x να είναι ο άξονας των πραγματικών αριθμών και τον άξονα y να είναι ο άξονας των φανταστικών αριθμών. Ένα τέτοιο παράδειγμα είναι το εξής.

```
1 sys = tf([2, 5, 1], [1, 3, 5]);
2 pzmap(sys);
3 grid on;
```

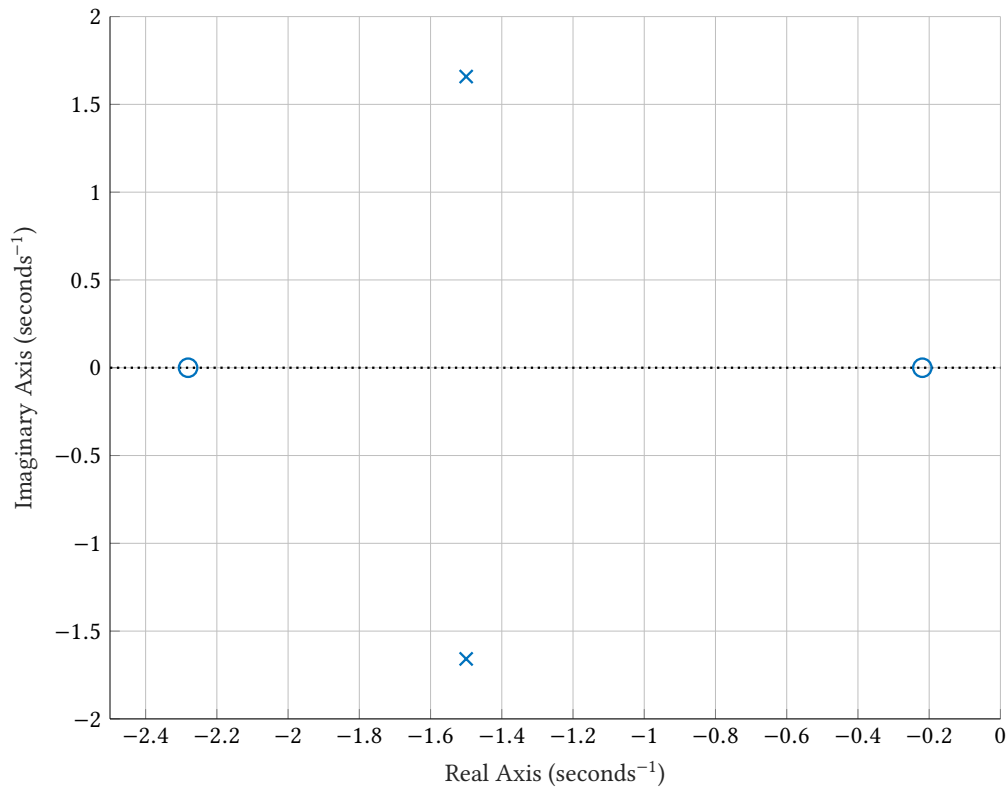
Το διάγραμμα που παίρνουμε φαίνεται στο σχήμα 1.1. Οι πόλοι συμβολίζονται με το σύμβολο x και τα μηδενικά με το σύμβολο o . Βλέπουμε πως οι πόλοι βρίσκονται στο αριστερό ημιεπίπεδο και άρα το σύστημα είναι ευσταθές.

Με το MATLAB μπορούμε να δηλώσουμε ένα σύστημα κλειστού βρόχου με την εντολή `feedback`. Με την ανάδραση σε ένα σύστημα κλειστού βρόχου μπορούμε να συγκρίνουμε την έξοδο του συστήματος με την είσοδο του συστήματος, έτσι ώστε η επιθυμητή δράση ελέγχου να αποτελεί συνάρτηση της εξόδου και της εισόδου του συστήματος. Η σύνταξη στο MATLAB είναι

$$\text{sys} = \text{feedback}(\text{sys1}, \text{sys2})$$

όπου `sys1` είναι η συνάρτηση μεταφοράς του ανοιχτού βρόχου και `sys2` είναι συνάρτηση μεταφοράς του κλάδου της ανάδρασης. Αξίζει να σημειωθεί ότι από προεπιλογή η εντολή `feedback` θεωρεί αρνητική ανάδραση. Φυσικά έχουμε την επιλογή να αλλάξουμε τη συμπεριφορά αυτή αν το επιθυμούμε. Για περαιτέρω ο ενδιαφερόμενος παραπέμπεται στη βοήθεια του MATLAB. Στη συνέχεια παρουσιάζεται παράδειγμα με τις απαραίτητες εντολές, που έχει συνάρτηση μεταφοράς

$$G(s) = \frac{4}{s^2 + 2s + 3},$$



Σχήμα 1.1: Διάγραμμα πόλων-μηδενικών, (ο μηδενικά, x πόλοι)

και θέλουμε μοναδιαία ανάδραση.

```
1 sys1 = tf([4], [1, 2, 3]);
2 sys2 = 1;
3 sys = feedback(sys1, sys2);
```

Αν το σύστημα που πρόκειται να ελέγξουμε είναι μη-γραμμικό ή χρονικά μεταβαλλόμενο ή έχει πολλαπλές εισόδους και εξόδους τότε είναι δύσκολο, αν όχι αδύνατον, να εξαχθεί η συνάρτηση μεταφοράς του μοντέλου. Για αυτόν το λόγο χρησιμοποιούμε το μοντέλο μεταβλητών κατάστασης για να περιγράψουμε το σύστημα.

Σύμφωνα με το βιβλίο [7], το μοντέλο μεταβλητών κατάστασης ενός συστήματος ορίζεται ως ο ελάχιστος αριθμός μεταβλητών, όπου η γνώση των οποίων σε οποιοδήποτε χρόνο σε συνδυασμό με την πληροφορία της εισόδου που εφαρμόζεται στο χρόνο αυτό, είναι ικανή συνθήκη για να προσδιορίσουμε την κατάσταση του συστήματος για κάθε μελλοντικό χρόνο. Συνήθως χρησιμοποιούμε ένα διάνυσμα διάστασης n για να συμβολίσουμε τις μεταβλητές κατάστασης, $x \in \mathbb{R}^n$. Χρησιμοποιούμε το διάνυσμα $u \in \mathbb{R}^m$ για να συμβολίσουμε τις m μεταβλητές εισόδου και $y \in \mathbb{R}^p$ για να συμβολίσουμε τις p μεταβλητές εξόδου. Η γενική μορφή ενός μοντέλου στο χώρο των καταστάσεων είναι της μορφής

$$\dot{x} = f(x, u, t)$$

$$y = g(x, u, t).$$

Στην περίπτωση που οι συναρτήσεις f, g είναι ανεξάρτητες του χρόνου t αλλά είναι και γραμμικές, τότε το μοντέλο μεταβλητών κατάστασης γράφεται στη μορφή

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du.\end{aligned}\tag{1.1}$$

Η δημιουργία μοντέλου μεταβλητών κατάστασης στο MATLAB γίνεται με την εντολή `ss`.

$$\text{sys} = \text{ss}(A, B, C, D)$$

Οι είσοδοι της συνάρτησης είναι οι πίνακες A, B, C, D , όπως φαίνονται στις σχέσεις (1.1). Παρακάτω παρατίθεται ένα παράδειγμα με τις αντίστοιχες εντολές του MATLAB.

```
1 A = [0, 1; 2, 1];
2 B = [0; 1];
3 C = [1, 1];
4 sys = ss(A, B, C, [])
```

1.2 Απόκριση συστήματος

Μπορούμε να διερευνήσουμε την απόκριση του συστήματος στο πεδίο του χρόνου χρησιμοποιώντας στη συνάρτηση `step`. Η συνάρτηση αυτή είναι από τις πιο χρήσιμες συναρτήσεις του MATLAB για το σχεδιασμό συστημάτων ελέγχου. Δοθέντος ενός συστήματος, λαμβάνουμε την απόκριση σε μοναδιαία είσοδο, χωρίς την ανάγκη να λύσουμε ως προς το χρόνο το σύστημα. Η συνάρτηση `step` μπορεί να περιγράψει ως η αλλαγή στην είσοδο από το μηδέν στο ένα στο χρόνο ένα. Η σύνταξη είναι

$$\text{step}(\text{sys})$$

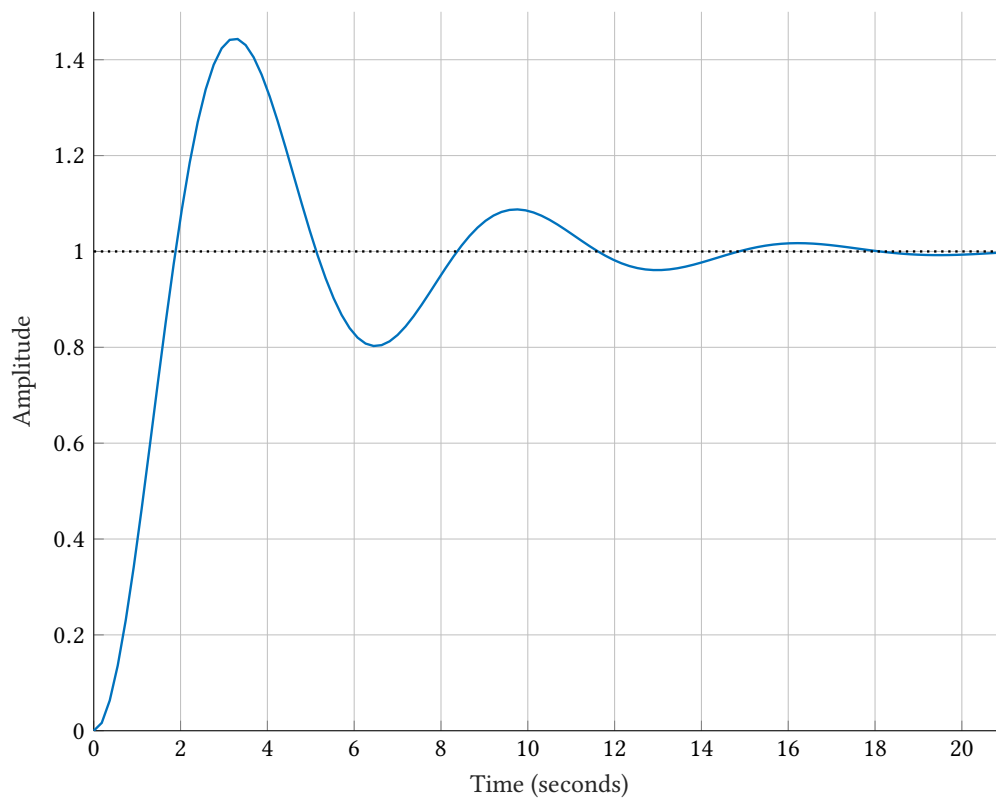
όπου `sys` είναι δυναμικό σύστημα, για παράδειγμα θα μπορούσε να είναι κάποιο από τα παραπάνω παραδείγματα. Αν πάρουμε για παράδειγμα ένα σύστημα δεύτερης τάξης

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2},$$

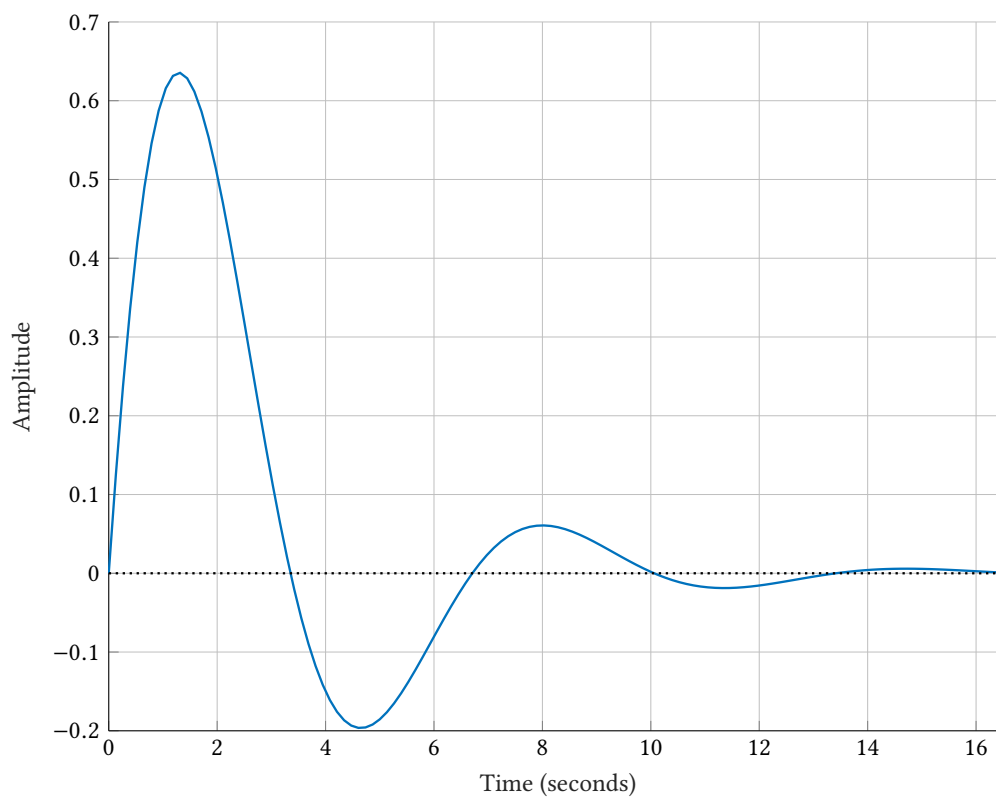
με $\omega_n = 1$ και $\zeta = 0.25$ και τρέξουμε τις ακόλουθες εντολές στο MATLAB θα λάβουμε το σχήμα 1.2.

```
1 sys = tf([1], [1, 0.5, 1]);
2 step(sys);
3 grid on;
```

Η απόκριση του συστήματος ανοιχτού βρόχου, είναι η αναμενόμενη διότι η απόσβεση είναι μικρή και έτσι έχουμε ταλαντώσεις αλλά και αργή σύγκλιση.



Σχήμα 1.2: Μοναδιαία βηματική απόκριση



Σχήμα 1.3: Κρουστική απόκριση

Ένα άλλο εργαλείο που μας παρέχει το MATLAB για τη διερεύνηση της απόκρισης του συστήματος είναι η συνάρτηση `impulse`. Η συνάρτηση αυτή υπολογίζει την απόκριση του συστήματος που υπόκειται σε κρουστική είσοδο, δηλαδή στη συνάρτηση $\delta(t)$ του Dirac. Η σύνταξη είναι όμοια με αυτή της `step` που παρουσιάσαμε παραπάνω. Παρακάτω παρουσιάζονται οι εντολές του MATLAB για την απόκριση του προηγούμενου παραδείγματος, αλλά με $\zeta = 0.7$ σε κρουστική είσοδο και η συμπεριφορά του συστήματος εμφανίζεται στο σχήμα 1.3. Παρατηρούμε πως η συγκεκριμένη τιμή του ζ μας δίνει καλύτερα δυναμικά χαρακτηριστικά, δηλαδή μικρότερες ταλαντώσεις και γρηγορότερη σύγκλιση στη μόνιμη κατάσταση.

```
1 sys = tf([1], [1, 0.7, 1]);
2 impulse(sys);
3 grid on;
```

Εκτός από τις συναρτήσεις `step` και `impulse` έχουμε τη δυνατότητα να μελετήσουμε την απόκριση του συστήματος για οποιοδήποτε σήμα εισόδου επιθυμούμε. Αυτό γίνεται με τη συνάρτηση `lsim` που συντάσσεται

$$\text{lsim}(\text{sys}, u, t)$$

όπου `sys` είναι το σύστημα, `u` είναι συνάρτηση της εισόδου που επιθυμούμε και `t` είναι ο χρονικός ορίζοντας. Σαν παράδειγμα μπορούμε να δούμε πως ανταποκρίνεται ένα σύστημα δεύτερης τάξης σε “τετραγωνικό” σήμα.

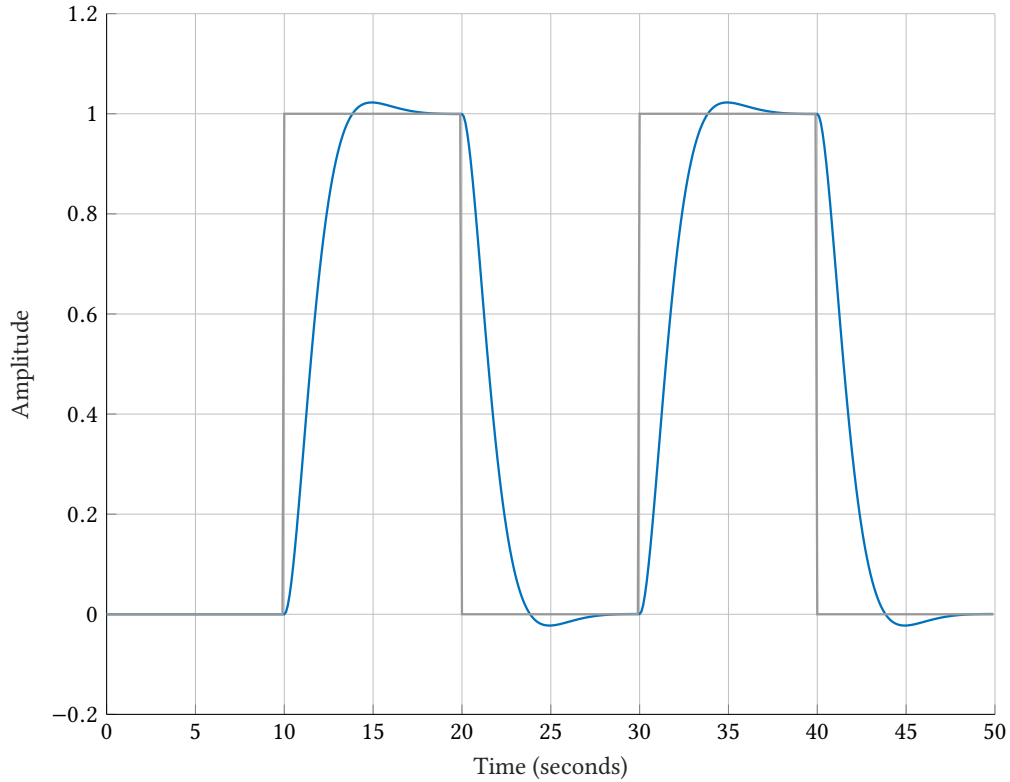
```
1 sys = tf([1], [1, 1.54, 1]);
2 [u, t] = gensig('square', 20, 50, 0.1);
3 lsim(sys, u, t);
4 grid on;
```

Στο παράδειγμα έχουμε φυσική συχνότητα $\omega_n = 1$, συντελεστή απόσβεσης $\zeta = 0.77$ και τετραγωνικό σήμα εισόδου. Στο σχήμα 1.4, με τη μπλε γραμμή εμφανίζεται η απόκριση του συστήματος και με τη γκρι το σήμα εισόδου, δηλαδή η επιθυμητή τροχιά. Με τη συνάρτηση `gensig` μπορούμε να δημιουργήσουμε διάφορα σήματα εισόδου. Φυσικά όμως έχουμε τη δυνατότητα να ορίσουμε οποιοδήποτε συνάρτηση εισόδου `u` επιθυμούμε και να τη χρησιμοποιήσουμε με την `lsim`.

1.3 Επίλυση συστημάτων βέλτιστου ελέγχου

Τέλος, με τη βοήθεια του MATLAB είναι πολύ εύκολη η επίλυση συστημάτων βέλτιστου ελέγχου. Με την εντολή

$$[K, S, e] = \text{lqr}(A, B, Q, R, N)$$



Σχήμα 1.4: Απόκριση σε τετραγωνικό σήμα εισόδου

υπολογίζουμε το βέλτιστο νόμο ελέγχου K . Πιο συγκεκριμένα, για ένα σύστημα ελέγχου συνεχούς χρόνου, ο βέλτιστος νόμος ελέγχου $u = -Kx$ είναι αυτός που ελαχιστοποιεί την τετραγωνική αντικειμενική συνάρτηση

$$J = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt$$

όπου Q, R είναι τα μητρώα στάθμισης των μεταβλητών κατάστασης και των μεταβλητών ελέγχου και N είναι το μητρώο στάθμισης των μεταβλητών κατάστασης συναρτήσει των μεταβλητών ελέγχου. Η ελαχιστοποίηση της παραπάνω γίνεται δεδομένου ότι ικανοποιούνται οι εξισώσεις της δυναμικής του συστήματος

$$\dot{x} = Ax + Bu.$$

Τελικά, καταλήγουμε στην αλγεβρική εξίσωση Riccati

$$A^T + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0,$$

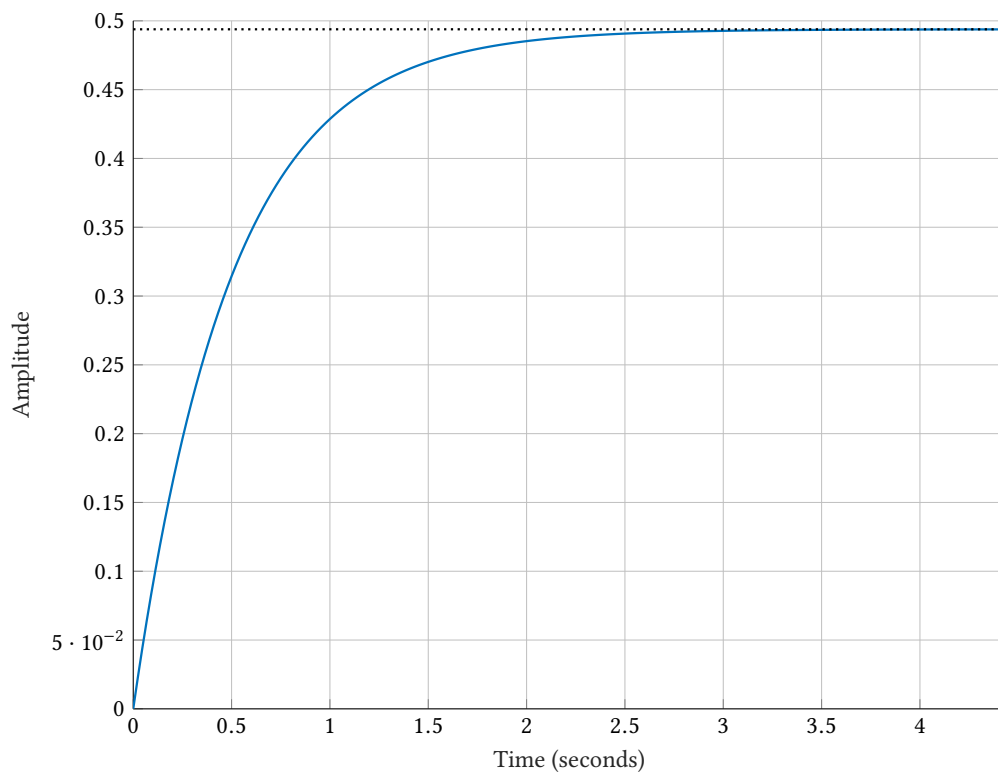
της οποίας τη λύση βρίσκουμε με την εντολή `lqr`. Σαν παράδειγμα θα δούμε το σύστημα μεταβλητών κατάστασης που είδαμε σε προηγούμενο παράδειγμα. Οι ιδιοτιμές του μητρώου A είναι $\lambda_1 = -1, \lambda_2 = 2$ και συνεπώς είναι ασταθές. Με τις παρακάτω εντολές κατασκευάζουμε βέλτιστο ελεγκτή με μοναδιαία στάθμιση στις μεταβλητές κατάστασης $Q = I_2$, στάθμιση στη μεταβλητή εισόδου $R = 10$ και $N = 0$.

```
1 A = [0, 1; 2, 1];  
2 B = [0; 1];  
3 C = [1, 1];  
4 [K, s, e] = lqr(A, B, eye(2), 10, []);  
5 step(ss(A - B*K, B, C, []));  
6 grid on;
```

Το κέρδος υπολογίζεται

$$K = \begin{bmatrix} 4.0248 & 4.0248 \end{bmatrix},$$

και η μοναδιαία βηματική απόκριση του βέλτιστου τετραγωνικού ελεγκτή παρουσιάζεται στο σχήμα 1.5.



Σχήμα 1.5: Μοναδιαία βηματική απόκριση βέλτιστου τετραγωνικού ελεγκτή

Κεφάλαιο 2

Ευστάθεια

Τα συστήματα σχεδιάζονται ώστε να εκτελούν κάποιες εργασίες. Αν ένα σύστημα δεν είναι ευσταθές τότε το σύστημα μπορεί να χαλάσει, μπορεί να αποτύχει να εκτελέσει την εργασία ή να εκτελεσθεί μερικώς. Επομένως στην πράξη ένα ασταθές σύστημα δεν είναι αξιοποιήσιμο και η ευστάθεια είναι μία βασική προδιαγραφή που πρέπει κάθε σύστημα να ικανοποιεί.

Η έννοια της ευστάθειας που συναντάται συνήθως στη μελέτη των δυναμικών συστημάτων και της θεωρίας ελέγχου είναι η ευστάθεια κατά Lyapunov και αυτή θα παρουσιάσουμε. Οι μαθηματικοί ορισμοί που θα ακολουθήσουν είναι από το βιβλίο [6].

Θεωρούμε ότι η δυναμική συμπεριφορά ενός συστήματος μοντελοποιείται μαθηματικά από τις λύσεις του δυναμικού συστήματος

$$\dot{x} = f(x), \quad f : U \rightarrow \mathbb{R}^n. \quad (2.1)$$

Ενδιαφερόμαστε για την μακροπρόθεσμη συμπεριφορά των τροχιών της παραπάνω. Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα σημεία ισορροπίας. Τέτοια σημεία $\bar{x} \in U$ είναι αυτά που δεν αλλάζουν με το χρόνο. Μαθηματικά, αυτό σημαίνει ότι η σταθερή απεικόνιση $t \rightarrow \bar{x}$ είναι λύση του δυναμικού συστήματος, ή ισοδύναμα, $f(\bar{x}) = 0$. Έτσι ορίζουμε *σημείο ισορροπίας* της (2.1) το σημείο $\bar{x} \in U$ τέτοιο ώστε $f(\bar{x}) = 0$.

2.1 Ευστάθεια κατά Lyapunov

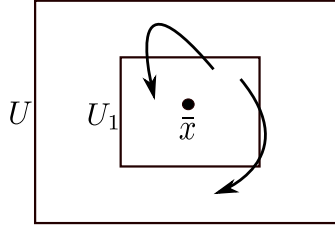
Θεωρούμε ότι ένα σημείο ισορροπίας είναι ευσταθές αν λύσεις που ξεκινούν “κοντά σε αυτό”, παραμένουν “κοντά σε αυτό” για κάθε μελλοντικό χρόνο. Ο μαθηματικοί ορισμοί της ευστάθειας κατά Lyapunov διατυπώνονται παρακάτω.

Ορισμός 1 (Ευστάθεια κατά Lyapunov, των Hirsch και Smale [6]). Έστω $\bar{x} \in W$ σημείο

ισορροπίας της διαφορικής εξίσωσης

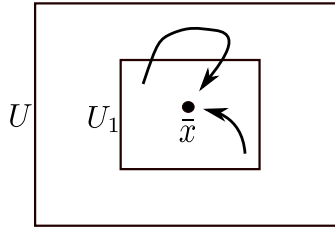
$$\dot{x} = f(x),$$

όπου $f : W \rightarrow E$ είναι C^1 συνάρτηση από το ανοιχτό σύνολο W του διανυσματικού χώρου E στο E . Τότε το \bar{x} είναι *ευσταθές* σημείο ισορροπίας αν για κάθε γειτονιά U του \bar{x} του W υπάρχει γειτονιά U_1 του \bar{x} στο U τέτοια ώστε κάθε λύση $x(t)$ με $x(0)$ στο U_1 να ορίζεται στο U για κάθε $t > 0$.



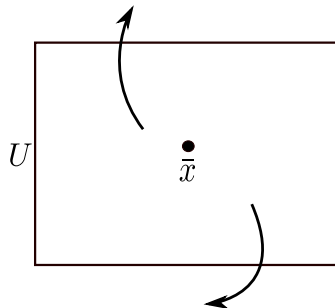
Σχήμα 2.1: Ευσταθές σημείο ισορροπίας

Ορισμός 2 (Ασυμπτωτική ευστάθεια κατά Lyapunov, των Hirsch και Smale [6]). Αν το U_1 μπορεί να επιλεγεί έτσι ώστε να ισχύουν τα όσα περιγράφηκαν στον παραπάνω ορισμό και $\lim_{t \rightarrow \infty} x(t) = \bar{x}$, τότε το \bar{x} είναι *ασυμπτωτικά ευσταθές*.



Σχήμα 2.2: Ασυμπτωτικά ευσταθές σημείο ισορροπίας

Ορισμός 3 (Αστάθεια κατά Lyapunov, των Hirsch και Smale [6]). Ένα σημείο ισορροπίας \bar{x} που δεν είναι ευσταθές είναι *ασταθές*. Αυτό σημαίνει ότι υπάρχει γειτονιά U του \bar{x} τέτοια ώστε για κάθε γειτονιά U_1 του \bar{x} στο U , υπάρχει τουλάχιστον μία λύση $x(t)$ που ξεκινάει από το $x(0) \in U_1$ και δε βρίσκεται στο U .



Σχήμα 2.3: Ασταθές σημείο ισορροπίας

Πρέπει να σημειωθεί ότι μόνο η σύγκλιση στο σημείο ισορροπίας στον δεύτερο ορισμό, δεν αρκεί για να χαρακτηριστεί το σημείο ασυμπτωτικά ευσταθές για γενική συνάρτηση f της

σχέσης (2.1). Παρόλο αυτά αποδεικνύεται ότι αν το υπό μελέτη σύστημα είναι γραμμικό και χρονικά αναλλοίωτο τότε η έννοια της σύγκλισης συμπίπτει με την έννοια της ασυμπτωτικής ευστάθειας. Με τους παραπάνω ορισμούς μπορούμε να χαρακτηρίσουμε ένα σημείο ισορροπίας ως προς την ευστάθεια του μέσω των ιδιοτιμών της γραμμικοποίησης. Παρόλο αυτά με την τεχνική αυτή, η ευστάθεια δε μπορεί να διαπιστωθεί για κάθε περίπτωση. Ο Ρώσος μαθηματικός και μηχανικός Lyapunov στη διδακτορική του διατριβή μας έδωσε ένα χρήσιμο κριτήριο περί ευστάθειας. Προτού διατυπώσουμε το θεώρημα θα δώσουμε κάποιους συμβολισμούς.

Έστω $V : V \rightarrow \mathbb{R}$ διαφορίσιμη συνάρτηση σε μία γειτονιά $U \subset W$ του \bar{x} . Συμβολίζουμε με $\dot{V} : U \rightarrow \mathbb{R}$ τη συνάρτηση που ορίζεται

$$\dot{V}(x) = DV(x)(f(x)).$$

Το δεξί μέλος της παραπάνω είναι απλώς ο τελεστής $DV(x)$ εφαρμοσμένος στο διάνυσμα $f(x)$. Τότε αν $\phi_t(x)$ είναι η λύση της (2.1) που περνάει από το x όταν $t = 0$, προκύπτει

$$\dot{V}(x) = \left. \frac{d}{dt} V(\phi_t(x)) \right|_{t=0},$$

από τον κανόνα της αλυσίδας. Συνεπώς, αν $\dot{V}(x)$ είναι αρνητικό, τότε η V μειώνεται συναρτήσει της λύσης της (2.1) ως προς το x . Ο ορισμός του θεωρήματος ευστάθειας του Lyapunov διατυπώνεται παρακάτω.

Θεώρημα 1 (Άμεση μέθοδος Lyapunov, των Hirsch και Smale [6]). Έστω $\bar{x} \in W$ σημείο ισορροπίας της (2.1). Έστω $V : U \rightarrow \mathbb{R}$ συνεχής συνάρτηση ορισμένη σε μία γειτονιά $U \subset W$ του \bar{x} , διαφορίσιμη στο $U - \bar{x}$, τέτοια ώστε

$$(\alpha) \quad V(\bar{x}) = 0 \text{ και } V(x) > 0 \text{ αν } x \neq \bar{x},$$

$$(\beta) \quad \dot{V} \leq 0 \text{ στο } U - \bar{x}.$$

Τότε το \bar{x} είναι ευσταθές. Επιπλέον, αν ακόμη

$$(\gamma) \quad \dot{V} < 0 \text{ στο } U - \bar{x},$$

τότε το \bar{x} είναι ασυμπτωτικά ευσταθές.

Συνάρτηση V που ικανοποιεί τα (α) και (β) ονομάζεται συνάρτηση Lyapunov για το \bar{x} . Αν ακόμη ικανοποιεί για το (γ) , τότε ονομάζεται αυστηρή συνάρτηση Lyapunov.

Το πλεονέκτημα του θεωρήματος του Lyapunov είναι ότι μπορεί να εφαρμοστεί χωρίς να λύσουμε τη διαφορική εξίσωση. Από την άλλη, δεν υπάρχει κάποια συγκεκριμένη μέθοδος για να βρούμε συναρτήσεις Lyapunov. Σε περιπτώσεις μηχανολογικών και ηλεκτρικών συστημάτων, η συνάρτηση της συνολική ενέργεια είναι καλή υποψήφια συνάρτηση Lyapunov. Για περαιτέρω παραπέμπουμε στις δημοσιεύσεις του Lyapunov [8] και στα βιβλία [6].

2.2 Ευστάθεια αυτόνομου γραμμικού αναλλοίωτου συστήματος

Το παραπάνω θεώρημα του Lyapunov απλοποιείται σημαντικά στην περίπτωση που έχουμε γραμμικό και χρονικά αναλλοίωτο σύστημα της μορφής

$$\dot{x}(t) = Ax(t), \quad x(0) = x_0. \quad (2.2)$$

Γνωρίζουμε ότι το σύστημα είναι ασυμπτωτικά ευσταθές αν όλες οι ιδιοτιμές του A έχουν αρνητικό πραγματικό μέρος. Ο πίνακας A τότε λέγεται Hurwitz πίνακας. Έτσι έχουμε την πρόταση:

Πρόταση 2 (Πρόταση 2.1 των Dullerud και Paganini [5]). Το αυτόνομο σύστημα (2.2) είναι εσωτερικά ευσταθές αν και μόνο αν ο πίνακας A είναι Hurwitz.

Το ισοδύναμο θεώρημα του Lyapunov για το παραπάνω σύστημα σύμφωνα με το βιβλίο [3], είναι το εξής:

Θεώρημα 3 (Θεώρημα 5.5 του Chen [3]). Όλες οι ιδιοτιμές του A έχουν αρνητικό πραγματικό μέρος αν και μόνο αν για κάποιον θετικά ορισμένο συμμετρικό πίνακα N , η συνάρτηση Lyapunov

$$A^T M + MA = -N,$$

έχει μοναδική συμμετρική λύση M και ο M είναι θετικά ορισμένος.

Επίσης, για τον υπολογισμό του πίνακα M προκύπτει το παρακάτω θεώρημα.

Θεώρημα 4 (Θεώρημα 5.6 του Chen [3]). Αν όλες οι ιδιοτιμές του A έχουν αρνητικό πραγματικό μέρος, τότε η συνάρτηση Lyapunov

$$A^T M + MA = -N,$$

έχει μοναδική λύση για κάθε N , και η λύση μπορεί να εκφραστεί ως

$$M = \int_0^\infty e^{A^T t} N e^{At} dt.$$

Όσα αναφέραμε παραπάνω αφορούν αυτόνομα συστήματα της μορφής (2.1). Στα συστήματα ελέγχου συνήθως, η είσοδος έχει μορφή ανάδρασης της κατάστασης, δηλαδή συνάρτηση του x . Έτσι με τους παραπάνω ορισμούς και θεωρήματα έχοντας την είσοδο ως $u(x)$ μπορούμε να αποφανθούμε για την ευστάθεια του συστήματος.

2.3 Ευστάθεια γραμμικού αναλλοίωτου συστήματος ελέγχου

Δοθέντος συστήματος της μορφής

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad (2.3)$$

με $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, τότε ονομάζουμε τον πίνακα

$$\begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$$

πίνακα ελεγχσιμότητας. Όταν ο πίνακας έχει πλήρη βαθμό n , τότε λέμε ότι το ζεύγος (A, B) είναι ελέγξιμο, ή αντίστοιχα ότι το σύστημα είναι ελέγξιμο. Αν βρούμε ένα νόμο ανάδρασης της μορφής $u(t) = Fx(t)$ με F σταθερό πίνακα στο $\mathbb{R}^{m \times n}$, τότε η παραπάνω γίνεται

$$\dot{x}(t) = (A + BF)x(t), \quad x(0) = x_0.$$

Τότε έχουμε το παρακάτω θεώρημα.

Θεώρημα 5 (Θεώρημα 2.19 των Dullerud και Paganini [5]). Οι ιδιοτιμές του $A + BF$ είναι ελεύθερα επιλέξιμες, επιλέγοντας τον πίνακα $F \in \mathbb{R}^{m \times n}$, αν και μόνο αν το ζεύγος (A, B) είναι ελέγξιμο.

Στην πράξη, πολύ συχνά μελετούμε γραμμικά αναλλοίωτα συστήματα της μορφής (2.3), όπου η ανάδραση είναι γραμμική συνάρτηση των καταστάσεων. Στις περιπτώσεις αυτές έχουμε μία ικανή και αναγκαία συνθήκη ώστε να μετατρέψουμε το πιθανόν ασταθές αυτόνομο σύστημα σε ευσταθές ή να επιλέξουμε σύμφωνα με κάποια κριτήρια τις ιδιοτιμές του κλειστού συστήματος με κατάλληλη επιλογή του πίνακα F .

Πολλές φορές όταν η είσοδος είναι μη μηδενική συναντούμε συχνά δύο τύπους ευστάθειας. Σύμφωνα με το βιβλίο [2] οι ορισμοί είναι:

Ορισμός 4 (Ορισμός 10.3 του Brogan [2]). Αν υπάρχει ένας σταθερός, πεπερασμένος και σταθερός K τέτοιος ώστε $\|u\| \leq K$ για κάθε χρόνο t , τότε η είσοδος λέμε ότι είναι φραγμένη. Αν για κάθε φραγμένη είσοδο, και για οποιαδήποτε αρχική συνθήκη $x(t_0)$, υπάρχει θετικό βαθμωτό $\delta(K, t_0, x(t_0))$ τέτοιο ώστε η επακόλουθη κατάσταση να ικανοποιεί $\|x\| \leq \delta$, τότε το σύστημα λέμε ότι είναι φραγμένης εισόδου, φραγμένης κατάστασης ευσταθές, γνωστό και ως BIBS.

Οι παραπάνω ορισμοί της ευστάθειας αντιμετωπίζουν τη συμπεριφορά του διανύσματος κατάστασης ως προς ένα σημείο ισορροπίας. Πολλές φορές, το βασικό ενδιαφέρον είναι η συμπεριφορά της εξόδου του συστήματος. Έτσι έχουμε τον ακόλουθο ορισμό περί ευστάθειας.

Ορισμός 5 (Ορισμός 10.4 του Brogan [2]). Έστω u φραγμένη είσοδος με K_m το ελάχιστο πάνω φράγμα. Αν υπάρχει βαθμωτό a τέτοιο ώστε για κάθε χρόνο t , η έξοδος ικανοποιεί $\|y\| \leq aK_m$, τότε το σύστημα λέμε ότι είναι φραγμένης εισόδου, φραγμένης εξόδου ευσταθές,

γνωστό και ως BIBO.

Η έννοια της ευστάθειας εισόδου, εξόδου γενικεύονται και ορίζεται μέσω των χώρων L_p . Για περαιτέρω παραπέμπουμε στο βιβλίο [10].

Κεφάλαιο 3

Αβεβαιότητες

Έστω ότι μας δίνεται ένα φυσικό σύστημα και επιθυμούμε να σχεδιάσουμε ένα σύστημα ελέγχου. Το πρώτο βήμα στο σχεδιασμό του συστήματος ελέγχου, είναι να περιγράψουμε μαθηματικά το σύστημα βασιζόμενοι στους φυσικούς νόμους που διέπουν τη συμπεριφορά του. Πολλές φορές, εμφανίζονται δυσκολίες που δυσχεραίνουν την ανάλυση του μοντέλου, πχ μη γραμμικό σύστημα. Για το λόγο αυτό προσεγγίζουμε το σύστημα με γραμμικό και σταθερών συντελεστών μοντέλο, που εκτιμούμε ότι προσεγγίζει ικανοποιητικά το πραγματικό σύστημα. Τα συστήματα ελέγχου δηλαδή, σχεδιάζονται χρησιμοποιώντας μαθηματικά μοντέλα τα οποία είναι μόνο προσεγγίσεις του πραγματικού συστήματος. Οι αποκλίσεις μεταξύ του συστήματος και της μαθηματικής του περιγραφής μπορεί να οδηγήσουν σε μη αποδεκτές επιδόσεις, ή ακόμα και σε αστάθεια του κλειστού συστήματος. Για να περιγραφούν οι αποκλίσεις αυτές χρησιμοποιείται ο όρος *αβεβαιότητα*. Έτσι είναι απαραίτητο στοιχείο του σχεδιασμού να συμπεριληφθούν οι αβεβαιότητες του συστήματος και η μοντελοποίηση ουσιαστικά, είναι ολοκληρωμένη μονάχα όταν έχουν ληφθεί υπόψιν οι αβεβαιότητες αυτές.

Υπάρχουν διάφοροι λόγοι που μπορεί να εμφανιστούν αβεβαιότητες, κάποιοι από αυτούς είναι:

- (α) Δεν γνωρίζουμε με ακρίβεια τις παραμέτρους του συστήματος που θέλουμε να ελέγξουμε.
- (β) Η δυναμική του συστήματος μπορεί να μεταβάλλεται με το χρόνο, και η μεταβολή αυτή να μην έχει ληφθεί στη μοντελοποίηση.
- (γ) Τα περισσότερα συστήματα ελέγχου κατά το σχεδιασμό θεωρούνται γραμμικά και χρονικά αναλλοίωτα. Αυτό απλοποιεί μεν την ανάλυση και τον έλεγχο του συστήματος άλλα, στην πραγματικότητα όμως τα πραγματικά συστήματα είναι μη γραμμικά. Έτσι το γραμμικό σύστημα είναι απλά μία προσέγγιση του πραγματικού.
- (δ) Πολλές φορές εμφανίζονται συστήματα που είναι πολύπλοκη η περιγραφή της συμπε-

ριφορά τους. Συνεπώς, δεν έχουμε άλλη επιλογή από το να χρησιμοποιήσουμε κάποια προσέγγιση αυτού.

(ε) Εξωτερικές διαταραχές που δε δύναται να προβλεφθούν.

Σύμφωνα με το βιβλίο [11], οι αβεβαιότητες που εμφανίζονται σε ένα σύστημα χωρίζονται σε δύο κατηγορίες, τις *δομημένες αβεβαιότητες* και τις *μη δομημένες αβεβαιότητες*. Η μαθηματική περιγραφή των δομημένων αβεβαιοτήτων είναι τέτοια ώστε γνωρίζουμε πως επηρεάζουν το σύστημα, και συνήθως περιγράφονται στο πεδίο του χρόνου. Όσον αφορά τις μη δομημένες αβεβαιότητες, καμία πληροφορία δεν είναι διαθέσιμη για το πως επιδρούν στο σύστημα. Συνήθως περιγράφονται στο πεδίο της συχνότητας και η απόκλιση από κάποια ονομαστική τιμή είναι φραγμένη από κάποιο άνω όριο. Ένα παράδειγμα δομημένης αβεβαιότητας είναι κάποια παραμετρική μεταβολή στη δυναμική του μοντέλου, που έχει ως αποτέλεσμα τη μεταβολή στα μηδενικά και στους πόλους της συνάρτησης μεταφοράς του μοντέλου. Παραδείγματα μη δομημένης αβεβαιότητας περιλαμβάνουν αβεβαιότητες σχετικά με τη συχνότητα, όπως υψηλές συχνότητες που συνήθως αγνοούμε κατά τη μοντελοποίηση ενός ταλαντωτή. Ένα ακόμα παράδειγμα μη δομημένης αβεβαιότητας είναι η γραμμικοποίηση του μη γραμμικού συστήματος για να λάβουμε ένα γραμμικό απλούστερο μοντέλο.

Θα κάνουμε μία απλή αναφορά στις επικρατέστερες περιγραφές και παραπέμπουμε στο βιβλίο [11] για περαιτέρω πληροφορίες.

Η περιγραφή στο πεδίο της συχνότητας, όπως αναφέραμε, είναι συνήθως μη δομημένες και κάποιες περιγραφές είναι η περιγραφή με αθροιστική αβεβαιότητα, η περιγραφή με πολλαπλασιαστική αβεβαιότητα στην είσοδο, η περιγραφή με πολλαπλασιαστική αβεβαιότητα στην έξοδο, η περιγραφή με αντίστροφη πολλαπλασιαστική αβεβαιότητα στην είσοδο, η περιγραφή με αντίστροφη πολλαπλασιαστική αβεβαιότητα στην έξοδο. Επίσης, είναι δυνατή και μία δομημένη περιγραφή της αβεβαιότητας.

Η περιγραφή ενός αβέβαιου συστήματος στο χώρο καταστάσεων είναι

$$\begin{aligned}\dot{x}(t) &= (A + \Delta A)x(t) + (B + \Delta B)u(t) \\ y(t) &= (C + \Delta C)x(t) + (D + \Delta D)u(t),\end{aligned}$$

όπου οι πίνακες ΔA , ΔB , ΔC και ΔD περιγράφουν τις αβεβαιότητες, που είναι μεταβολές των παραμέτρων του μοντέλου. Θα θεωρήσουμε την περίπτωση όπου μόνο ο πίνακας κατάστασης είναι αβέβαιος.

Για την περίπτωση των μη δομημένων αβεβαιοτήτων έχουμε δύο περιγραφές. Τη *μη δομημένη παραμετρική αβεβαιότητα*. Ο πίνακας ΔA περιέχει όλες τις μεταβολές των παραμέτρων, χωρίς όμως να είναι γνωστή η δομή του, αλλά είναι γνωστό ένα άνω φράγμα $\|\Delta A\| \leq a$, όπου a κάποιο θετικό και βαθμωτό μέγεθος. Την *παραμετρική στοχαστική αβεβαιότητα*, όπου η αβεβαιότητα περιγράφεται από κάποια στατιστικό μοντέλο.

Για την περίπτωση της δομημένης αβεβαιότητας, η *δομημένη παραμετρική αβεβαιότητα* είναι η συνηθέστερη. Οι επικρατέστερες περιγραφές αυτής είναι οι ακόλουθες. Η *γραμμική παραμετρική αβεβαιότητα*, όπου οι αβεβαιότητες είναι της μορφής

$$\Delta A = \sum_{i=1}^k A_i p_i,$$

όπου οι πίνακες A_i προσδιορίζουν τον τρόπο που επηρεάζει η αβεβαιότητα στον πίνακα κατάστασης και p_i είναι μεταξύ ενός κάτω και ενός άνω ορίου. Η *πολυτοπική αβεβαιότητα*, όπου ο πίνακας κατάστασης ανήκει σε ένα πολύτοπο που ορίζεται

$$A \in D_A = \{A : A = \sum_{i=1}^k A_i a_i, a_i \geq 0, \sum_{i=1}^k a_i = 1\},$$

όπου k είναι ο αριθμός των κορυφών του πολυέδρου D_A . Άλλες περιγραφές αβεβαιότητας είναι η *αβεβαιότητα με φραγμένη νόρμα*, *πραγματική θετική αβεβαιότητα* και η *πραγματική φραγμένη αβεβαιότητα*.

3.1 Παράδειγμα αβέβαιου συστήματος

Έστω το σύστημα

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx,\end{aligned}$$

όπου οι πίνακες A, B, C είναι

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

Το σύστημα είναι ασταθές, καθώς οι ιδιοτιμές του πίνακα κατάστασης A είναι $\lambda_{1,2,3} = -0.2757, 1.1378 + 1.5273i, 1.1378 - 1.5273i$ και έχουμε θετικό πραγματικό μέρος.

Για να σταθεροποιήσουμε το σύστημα θα κατασκευάσουμε ένα βέλτιστο τετραγωνικό ελεγκτή απείρου χρόνου που ελαχιστοποιεί το δείκτη επίδοσης

$$J = \int_0^\infty (x^T Q x + u^T R u), dt,$$

με $Q = I_3$ και $R = 1$. Τρέχοντας την παρακάτω εντολή στο MATLAB με τους πίνακες όπως έχουν οριστεί παραπάνω

$$[K, S, e] = \text{lqr}(A, B, Q, R, N)$$

υπολογίζουμε το βέλτιστο κέρδος

$$K = \begin{bmatrix} 0.4142 & 1.7127 & 4.9027 \end{bmatrix}.$$

Ο βέλτιστος νόμος ανάδρασης είναι $u = -Kx$ που οδηγεί στο κλειστό σύστημα με πίνακα κατάστασης

$$A_c = A - BK,$$

που είναι ευσταθές, καθώς οι ιδιοτιμές του είναι $\lambda_{1,2,3} = -0.3758, -1.2634 + 1.4720i, -1.2634 - 1.4720i$ και όλες έχουν αρνητικό πραγματικό μέρος.

Τώρα θεωρούμε ότι το σύστημα υπόκειται σε αβεβαιότητες που επηρεάζουν τον πίνακα κατάστασης. Έστω ακόμα ότι η δομή της αβεβαιότητας είναι τέτοια που ο πίνακας A γίνεται

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 + r_1 & 2 + r_2 \end{bmatrix},$$

όπου r_1, r_2 εκφράζουν το εύρος της αβεβαιότητας. Σύμφωνα με τα όσα είπαμε παραπάνω, βρισκόμαστε στην κατηγορία της γραμμικής παραμετρικής αβεβαιότητας. Ο νέος πίνακας κατάστασης με την είσοδο των αβεβαιοτήτων μπορεί να γραφτεί

$$A = A_0 + \Delta A_1 + \Delta A_2,$$

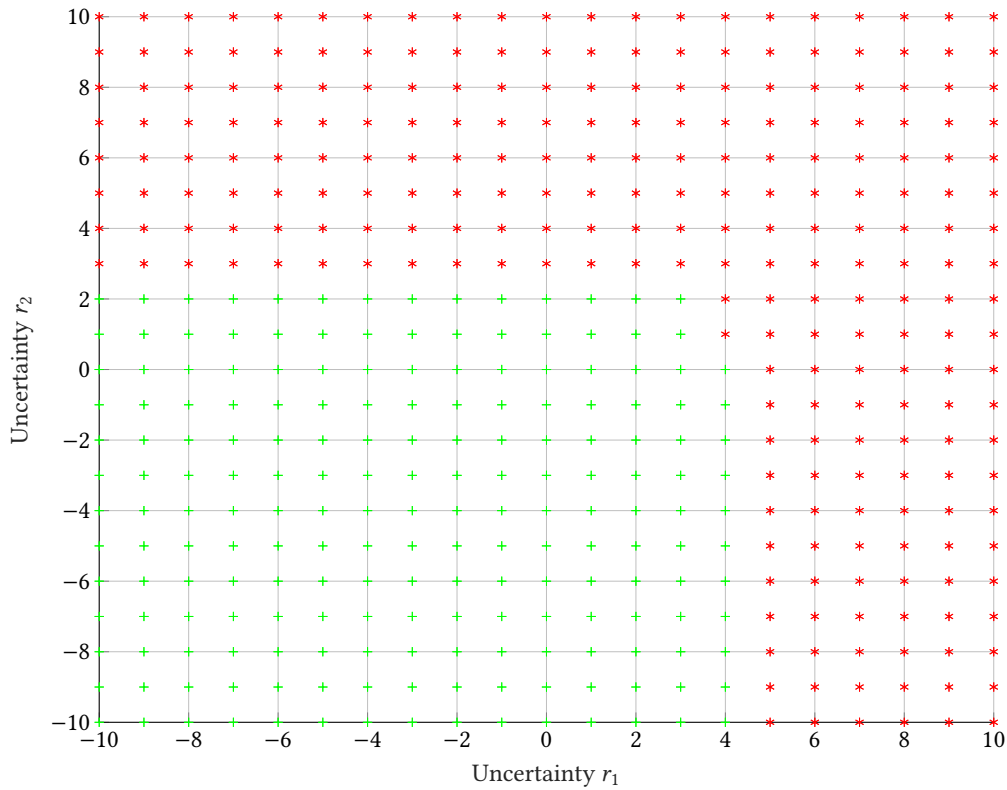
όπου ο πίνακας A_0 είναι ο ονομαστικός πίνακας κατάστασης, δηλαδή ο πίνακας του συστήματος χωρίς αβεβαιότητες και $\Delta A_1, \Delta A_2$ είναι οι αβεβαιότητες. Έτσι αντικαθιστώντας τους πίνακες στην παραπάνω σχέση προκύπτει

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -3 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} r_1 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} r_2.$$

Μεταβάλλοντας τις αβεβαιότητες r_1, r_2 εντός του εύρους τιμών $[-10, 10]$ μπορούμε να διαπιστώσουμε την επίδραση του κάθε ζεύγους (r_1, r_2) στον πίνακα A , όπως φαίνεται παραπάνω, και ως εκ τούτου στο σύστημα.

Εφαρμόζοντας τη μεταβολή αυτή στο MATLAB, προκύπτει το σχήμα 3.1. Όπως παρατηρούμε τα ζεύγη τιμών κατά τα οποία μας δίνουν το πράσινο σταυρό, είναι αυτά τα ζεύγη r_1, r_2 που δεν μετατρέπουν το ευσταθές σύστημα σε ασταθές. Παρόλο αυτά με τη μεταβολή των αβεβαιοτήτων το σύστημα επηρεάζεται. Όσο αλλάζουν κατά απόλυτη τιμή τα r_1, r_2 τόσο αποκλίνει το σύστημα από τους στόχους, είτε αυτοί είναι γρήγορη σύγκλιση, μικρή υπερύψωση κτλ, μέχρι που καταλήγει να γίνει ασταθές.

Για να δούμε την επιρροή των αβεβαιοτήτων στο σύστημα παρουσιάζεται το σχήμα 3.2. Το



Σχήμα 3.1: Έλεγχος ευστάθειας συναρτήσει της μεταβολής των αβεβαιοτήτων r_1, r_2

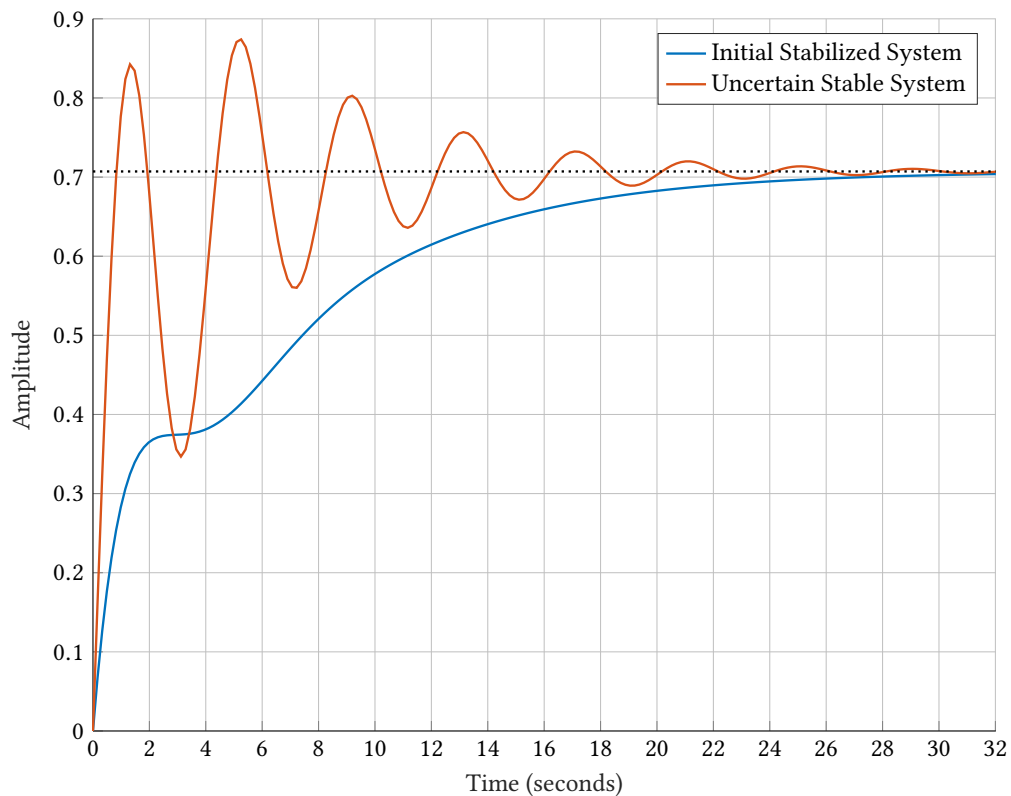
ζεύγος τιμών $(r_1, r_2) = (2, 2)$ δε μετατρέπει το αβέβαιο σύστημα σε ασταθές, όπως φαίνεται στο σχήμα 3.1. Παρόλο αυτά η επίδραση του στο σύστημα σε σχέση με το ονομαστικό είναι πολύ μεγάλη. Οι ταλαντώσεις που δημιουργούνται είναι πολύ ισχυρές και ο χρόνος αποκατάστασης μεγαλώνει αισθητά. Καταλαβαίνουμε δηλαδή ότι βρισκόμαστε κοντά σε ασταθή κατάσταση.

Παρακάτω παρατίθεται ο κώδικας στο MATLAB που χρησιμοποιήθηκε για το παράδειγμα.

```

1 %% Define the matrix A
2 A = [0 1 0; 0 0 1; -1 -3 2];
3 eig(A);
4 %Since the system is in canonical controllability form B=[0 0 1]'
5 B = [0 0 1]';
6 C = [1 1 1];
7 D = 0;
8 %% Apply LQR with Q=eye(3) and R=1
9 Q = eye(3);
10 R = 1;
11
12 K = lqr(A, B, Q, R);
13 %The state matrix for the stabilized (Ac) system is A-BK
14 Ac = A - B*K;
15 eig(Ac);
16

```



Σχήμα 3.2: Μοναδιαία βηματική απόκριση ονομαστικού και αβέβαιου συστήματος

```

17 %% Define the uncertainties
18 %Suppose uncertainties affect the elements in the positions
19 %(3,2) and (3,3) of the initial matrix A
20 %Thus the matrices that state the positions of each
21 %uncertainties in A are
22
23 A1 = [0 0 0;0 0 0; 0 1 0];
24 A2 = [0 0 0;0 0 0; 0 0 1];
25
26 %Also define the range of each uncertainty, namely r1 and r2.
27 r1 = 1;
28 r2 = 1;
29 %NOTE:
30 %Here the range does not affect the matrices A1 and A2, but
31 %this will now always be the case
32
33 %% Stability through uncertainties check
34 %Now check if K that was calculated for the matrix A,
35 %stabilizes all the uncertain systems that arise
36 figure(1)
37 title('Stability Check on the plane of uncertainties');
38 hold all;
39 for i = -10:1:10

```



```
40     for j = -10:1:10
41         A_uncertain = A + A1*i + A2*j - B*K;
42         eigvals = eig(A_uncertain)';
43         %using max of eigenvalue vector > 0 means that at
44         %least one of the eigenvalues are positive and
45         %thus the system is unstable
46         if max(real(eigvals)) >= 0
47             plot(i, j, '*r')
48         else
49             plot(i, j, '+g')
50         end
51     end
52 end
53 grid on;
54 xlabel('Uncertainty r1');
55 ylabel('Uncertainty r2');
56
57 %% Plot the step responses for one of the stable systems
58 %that arise from certain values of uncertainties.
59 % Also plot the initial stabilized system and
60 %compare it with the uncertain one.
61 figure(2)
62 title('Step Responses');
63 hold all;
64 sys1 = ss(Ac, B, C, D);
65 sys2 = ss(A + A1*2 + A2*2 - B*K, B, C, D);
66 step(sys1, sys2);
67 grid on;
68 legend('Initial Stabilized System','Uncertain Stable System');
```

Κεφάλαιο 4

Μέθοδοι LQG εγγυημένου κόστους

Στο σημείο αυτό θα παρουσιάσουμε τον εύρωστο έλεγχο γραμμικών αβέβαιων συστημάτων με τετραγωνικό δείκτη συμπεριφοράς στο πλαίσιο του ελέγχου εγγυημένου κόστους. Η ανάπτυξη που θα ακολουθήσει βασίστηκε στο βιβλίο [11]. Ο ενδιαφερόμενος παραπέμπεται στο συγκεκριμένο καθώς θα παρουσιάσουμε μονάχα κάποια βασικά αποτελέσματα, χωρίς τις αντίστοιχες αποδείξεις, αλλά θα επικεντρωθούμε στην υλοποίηση των μεθόδων μέσω παραδειγμάτων με τη βοήθεια του MATLAB.

Ένα τυπικό γραμμικό αβέβαιο σύστημα είναι της μορφής

$$\dot{x}(t) = (A + \Delta A)x(t) + (B + \Delta B)u(t), \quad x(0) = x_0, \quad (4.1)$$

για $t \geq 0$, όπου $x(t) \in \mathbb{R}^n$ το διάνυσμα κατάστασης, $u(t) \in \mathbb{R}^m$ το διάνυσμα ελέγχου, A και B οι πίνακες κατάστασης και ελέγχου αντίστοιχα, κατάλληλων διαστάσεων. Οι αβεβαιότητες περιγράφονται με τη μορφή γραμμικών συναρτήσεων

$$\Delta A = \sum_{i=1}^k A_i r_i, \quad \Delta B = \sum_{i=1}^l B_i p_i,$$

όπου A_i και B_i είναι σταθεροί πίνακες και r_i, p_i βαθμωτά μεγέθη, που μπορεί να είναι και χρονικά μεταβαλλόμενα στη γενική περίπτωση. Με την παραπάνω περιγραφή ορίζουμε τα σύνολο

$$\mathcal{R} := \{r \in \mathbb{R}^k : |r_i| \leq \bar{r}, \bar{r} > 0, i = 1, \dots, k\}$$

$$\mathcal{J} := \{p \in \mathbb{R}^l : |p_i| \leq \bar{p}, \bar{p} > 0, i = 1, \dots, l\}.$$

Χωρίς να μειωθεί η γενικότητα τα όρια μπορούν να κανονικοποιηθούν, δηλαδή $\bar{r} = \bar{p} = 1$. Έτσι ονομάζουμε *αποδεκτή αβεβαιότητα*, τις αβεβαιότητες που ικανοποιούν τις γραμμικές μορφές, όπως ορίστηκαν παραπάνω, και βρίσκονται εντός των φραγμένων συνόλων.

Στόχος είναι να βρεθεί γραμμικός νόμος ελέγχου με ανάδραση κατάστασης σταθερού κέρ-

δους, έτσι ώστε το κλειστό σύστημα να διατηρεί την ασυμπτωτική ευστάθεια και ένα επιθυμητό επίπεδο επιδόσεων, για κάθε αποδεκτή αβεβαιότητα.

Θα θεωρήσουμε τις επιδόσεις του συστήματος ότι περιγράφονται από τετραγωνική συνάρτηση κόστους

$$J(x_0, \Delta A, \Delta B, t) = \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)] dt, \quad (4.2)$$

με $Q \geq 0$ και $R > 0$ όπου θέλουμε να ελαχιστοποιήσουμε. Για να το επιτύχουμε αυτό υποθέτουμε ότι το ζεύγος (A, B) είναι ελέγξιμο και το ζεύγος $(Q^{1/2}, A)$ είναι παρατηρήσιμο. Ακόμη θεωρούμε το διάνυσμα κατάστασης είναι διαθέσιμο προς μέτρηση. Στην περίπτωση που δεν υπάρχουν αβεβαιότητες το πρόβλημα συμπίπτει με το πρόβλημα του γραμμικού τετραγωνικού ρυθμιστή.

Για το σύστημα (4.1) με την τετραγωνική συνάρτηση κόστους (4.2) η συνάρτηση $\tilde{u} : \mathbb{R} \rightarrow \mathbb{R}^m$ ονομάζεται *νόμος ελέγχου εγγυημένου κόστους*, αν υπάρχει ένας θετικός αριθμός \tilde{J} , που λέγεται *εγγυημένο κόστος*, τέτοιος ώστε

$$J(x_0, \tilde{u}, \Delta A, \Delta B, t) \leq \tilde{J}$$

για όλες τις αποδεκτές αβεβαιότητες. Αποδεικνύεται ότι ο γραμμικός νόμος ελέγχου ανάδρασης κατάστασης

$$\tilde{u}(t) = Kx(t) = -R^{-1}B^T Px(t), \quad (4.3)$$

είναι ένας νόμος ελέγχου εγγυημένου κόστους και

$$\tilde{J}(x_0) = x_0^T Px_0,$$

είναι ένα εγγυημένο κόστος ή με κάποιες τεχνικές παραδοχές στην αντίστοιχη μορφή

$$\tilde{J} = \text{tr } P, \quad (4.4)$$

αν υπάρχει ένας θετικά ορισμένος συμμετρικός πίνακας P που ικανοποιεί τη γενικευμένη αλγεβρική εξίσωση Riccati, για συντομία GARE,

$$PA + A^T P - PBR^{-1}B^T P + Q + \mathcal{U}(P) = 0. \quad (4.5)$$

Η συνάρτηση $\mathcal{U}(P)$ είναι μία συνάρτηση άνω φράγματος που εξαρτάται από τις αβεβαιότητες και ισχύει

$$2x^T(t)P\Delta Ax(t) + 2x^T(t)P\Delta B Kx(t) \leq x^T(t)\mathcal{U}(P)x(t), \quad (4.6)$$

για κάθε x και για όλες τις αποδεκτές αβεβαιότητες.

4.1 Συστήματα με αβεβαιότητα στον πίνακα κατάστασης

Στην περίπτωση αυτή έχουμε αβεβαιότητα μόνο στον πίνακα κατάστασης, δηλαδή έχουμε $\Delta B = 0$. Έτσι για το άνω φράγμα $\mathcal{U}(P)$, η σχέση (4.6) θα γίνει

$$P\Delta A + \Delta A^T P \leq \mathcal{U}. \quad (4.7)$$

Τότε διακρίνουμε δύο περιπτώσεις φράγματος,

- (α) φράγμα με βάση τις ιδιοτιμές,
- (β) γραμμικό φράγμα.

4.1.1 Φράγμα με βάση τις ιδιοτιμές

Στην περίπτωση αυτή, εφόσον ο πίνακας στο αριστερό μέλος της σχέσης (4.7) είναι συμμετρικός, υπάρχουν ορθογώνιοι πίνακες μετασχηματισμού M_i τέτοιοι ώστε

$$M_i^T (PA_i + A_i^T P) M_i = \Lambda_i,$$

όπου Λ_i διαγώνιοι πίνακες. Αν συμβολίσουμε με $|\Lambda_i|$ τους αντίστοιχους πίνακες όπου τα διαγώνια στοιχεία έχουν αντικατασταθεί από τις απόλυτες τιμές, τότε μία συνάρτηση άνω φράγματος που ικανοποιεί την (4.7) είναι

$$\mathcal{U}(P) = \sum_{i=1}^k M_i |\Lambda_i| M_i^T.$$

Με αντικατάσταση της παραπάνω $\mathcal{U}(P)$ στη γενικευμένη αλγεβρική εξίσωση Riccati (4.5), μπορούμε να υπολογίσουμε το πίνακα P . Στη συνέχεια από τη σχέση (4.3) βρίσκουμε το νόμο ελέγχου εγγυημένου κόστους και το αντίστοιχο εγγυημένο κόστος από τη σχέση (4.4). Παρόλο που η τεχνική είναι εφικτή από θεωρητική σκοπιά, πρακτικά είναι δύσκολα εφαρμόσιμη. Ακόμη δεν υπάρχει απόδειξη σύγκλισης ούτε και συνθήκη ύπαρξης θετικά ορισμένης λύσης P .

4.1.2 Γραμμικό φράγμα

Στην περίπτωση αυτή αποδεικνύεται ότι, για το αριστερό μέλος της σχέσης (4.7) ισχύει η ανισότητα

$$P\Delta A + \Delta A^T P \leq P\Gamma P + \sum_{i=1}^k A_i^T \Gamma^{-1} A_i.$$

Αν επιλέξουμε $\Gamma = \varepsilon P^{-1}$ τότε μπορούμε να επιλέξουμε σαν συνάρτηση άνω φράγματος

$$\mathcal{U}(P) = k\varepsilon P + \varepsilon^{-1} \sum_{i=1}^k A_i^T P A_i. \quad (4.8)$$

Έτσι λαμβάνουμε μία συνάρτηση άνω φράγματος γραμμική ως προς τον πίνακα P . Ακόμα μέση της παραμέτρου ε μπορούμε να ικανοποιήσουμε περαιτέρω προδιαγραφές σχεδίασης, με μία πιθανή επιλογή της παραμέτρου να είναι $\varepsilon = \sum \|A_i\|$ ώστε να εισαχθεί ένα μέτρο του μεγέθους της αβεβαιότητας. Τέλος η παράμετρος k που πολλαπλασιάζει τον πρώτο όρο της παραπάνω σχέσης, επιλέγεται έτσι ώστε να μηδενίζει τη συνάρτηση φράγματος όταν δεν υπάρχουν αβεβαιότητες.

Αντικαθιστώντας τη συνάρτηση άνω φράγματος (4.8) στη γενικευμένη αλγεβρική εξίσωση Riccati (4.5), τότε προκύπτει μία μη γραμμική αλγεβρική εξίσωση της μορφής $f(x) = 0$. Μία διαδεδομένη μέθοδος για την επίλυση τέτοιων εκφράσεων είναι με τη μέθοδο Newton (ή Newton-Raphson). Η σχέση (4.5), είναι μία σχέση πινάκων, διάστασης $n \times n$, όπου n η διάσταση του διανύσματος κατάστασης. Επομένως για να εφαρμόσουμε τη μέθοδο Newton πρέπει να μετατρέψουμε την (4.5) σε διανυσματική. Ακόμα με τη συνάρτηση `fsolve` του MATLAB, που υλοποιεί τη μέθοδο Newton-Raphson υπολογίζουμε τελικά τον θετικά ορισμένο πίνακα P . Στη συνέχεια από τη σχέση (4.3) βρίσκουμε το νόμο ελέγχου εγγυημένου κόστους και το αντίστοιχο εγγυημένο κόστος από τη σχέση (4.4).

Ένα γνωστό πρόβλημα της μεθόδου Newton-Raphson είναι ότι η σύγκλιση και η επιτυχία της μεθόδου εξαρτάται από την αρχική εκτίμηση. Μία λογική αρχική εκτίμηση για την εκκίνηση της μεθόδου είναι ο πίνακας P που υπολογίζεται λύνοντας την αλγεβρική εξίσωση Riccati. Δηλαδή η λύση αν το σύστημα δεν ήταν αβέβαιο. Όπως παρατηρήσαμε στην πράξη, αυτή η αρχική εκτίμηση είναι αρκετά καλή καθώς παρουσιάστηκε σύγκλιση σε οποιοδήποτε παράδειγμα επιλύσαμε. Φυσικά στη συγκεκριμένη εργασία θα παρουσιάσουμε ένα μονάχα παράδειγμα για τη συγκεκριμένη τεχνική.

Θα εφαρμόσουμε τη παραπάνω διαδικασία σε ένα παράδειγμα. Το παράδειγμα που θα παρουσιάσουμε είναι απλό και έχει εσκεμμένα επιλεγεί έτσι για ευκολότερη κατανόηση. Παρόλα αυτά η σχεδίαση του κώδικα έχει γραφτεί με το σκεπτικό να επιλύει οποιοδήποτε πρόβλημα εισάγουμε.

Έστω το σύστημα

$$\dot{x} = Ax + Bu,$$

όπου οι πίνακες A και B είναι

$$A = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Τα μητρώα στάθμισης του δείκτη απόδοσης, σχέση (4.2), επιλέχθηκαν $Q = I_3$ και $R = 10$.

Ακόμα οι αβεβαιότητες στον πίνακα κατάστασης επιλέχθηκαν

$$A_1(r_1) = r_1 \begin{bmatrix} 0 & 0 \\ 3 & 0 \end{bmatrix}, \quad A_2(r_2) = r_2 \begin{bmatrix} 0 & 0 \\ 0 & 1.5 \end{bmatrix}.$$

Η σχέση (4.8) μπορεί να γραφτεί στην ισοδύναμη μορφή

$$\mathcal{U}(P) = \varepsilon P + \varepsilon^{-1} \Delta A^T P \Delta A,$$

καθώς υπάρχουν αβεβαιότητες. Έτσι η γενικευμένη αλγεβρική εξίσωση Riccati θα γίνει

$$PA + A^T P - PBR^{-1}B^T P + Q + \varepsilon P + \varepsilon^{-1} \Delta A^T P \Delta A = 0. \quad (4.9)$$

Στο παράδειγμα επιλέξαμε την παράμετρο $\varepsilon = 1$ και δίχως μείωση της γενικότητας βρήκαμε το εγγυημένο κόστος για τα κανονικοποιημένα όρια των αποδεκτών αβεβαιοτήτων, δηλαδή για $|r_1| \leq 1$ και $|r_2| \leq 1$.

Τελικά, βρίσκουμε τον πίνακα P της γενικευμένης αλγεβρικής εξίσωσης Riccati

$$P = \begin{bmatrix} 16.5034 & 3.0037 \\ 3.0037 & 1.1554 \end{bmatrix},$$

που είναι συμμετρικός και θετικά ορισμένος με ιδιοτιμές $\lambda_{1,2} = 0.5884, 17.0703$. Από τη σχέση (4.3) ο νόμος ελέγχου εγγυημένου κόστους υπολογίζεται

$$\tilde{u}(t) = -R^{-1}B^T P x(t) = \begin{bmatrix} -0.3004 & -0.1155 \end{bmatrix} x(t).$$

Το εγγυημένο κόστος υπολογίζεται από τη σχέση (4.4)

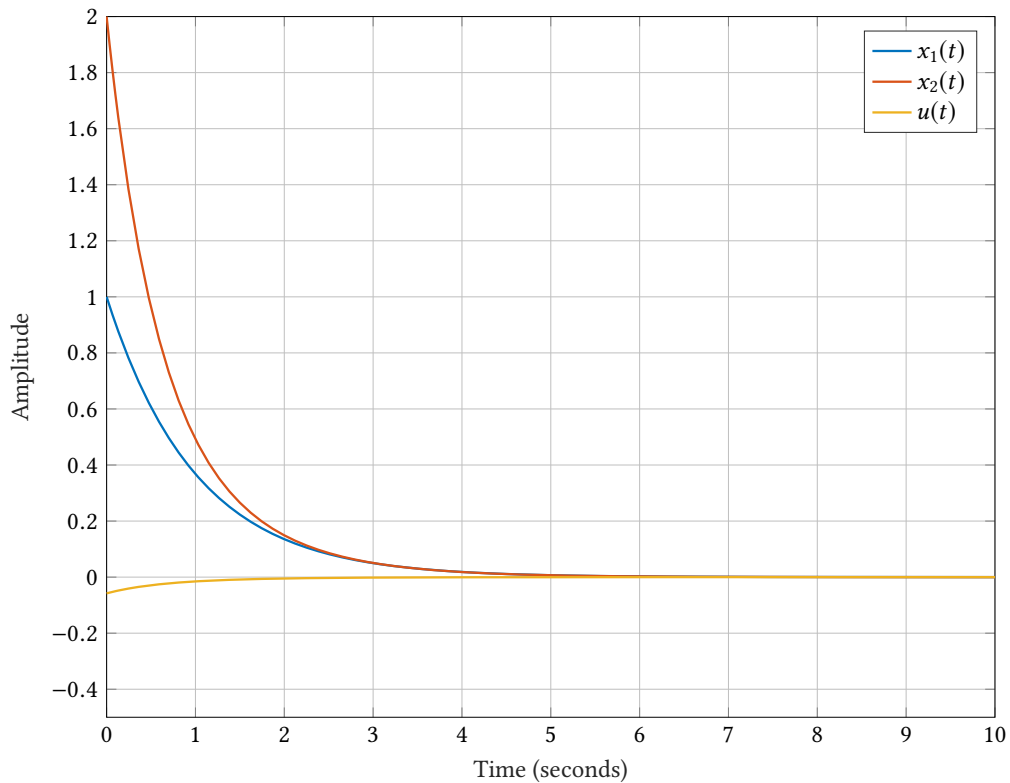
$$\tilde{J} = \text{tr } P = 17.6587,$$

ενώ από τη σχέση που εξαρτάται από τις αρχικές συνθήκες

$$\tilde{J}(x_0) = x_0^T P x_0 = \begin{bmatrix} 1 & 2 \end{bmatrix} P \begin{bmatrix} 1 & 2 \end{bmatrix}^T = 33.1396.$$

Στα σχήματα 4.1, 4.2 και 4.3 παρουσιάζονται οι καταστάσεις του συστήματος για τις διάφορες τιμές των αβεβαιοτήτων καθώς και ο νόμος ελέγχου εγγυημένου κόστους.

Στο ίδιο αρχείο κώδικα, γίνεται υλοποίηση του θεωρήματος 3.4 του βιβλίου [11]. Σύμφωνα με το θεώρημα αυτό, αν υπάρχει μία θετικά ορισμένη λύση της γενικευμένης αλγεβρικής εξίσωσης Riccati τότε αυτή μπορεί να βρεθεί επαναληπτικά, επιλύοντας μία εξίσωση Lyapunov. Στο παράδειγμα υλοποιήσαμε το θεώρημα με τη συνάρτηση `gcc_lyap`. Σαν αρχική λύση της εξίσωσης Lyapunov, όπως αυτή ορίζεται στο αντίστοιχο θεώρημα, δώσαμε τη λύση που υπολογίσαμε από την πρώτη `fsolve`. Παρατηρούμε ότι η εξίσωση Lyapunov ικανοποιείται για τον πίνακα P που υπολογίσαμε και έτσι επιβεβαιώνεται ότι η λύση που υπολογίσαμε



Σχήμα 4.1: Αρχικό σύστημα χωρίς αβεβαιότητες

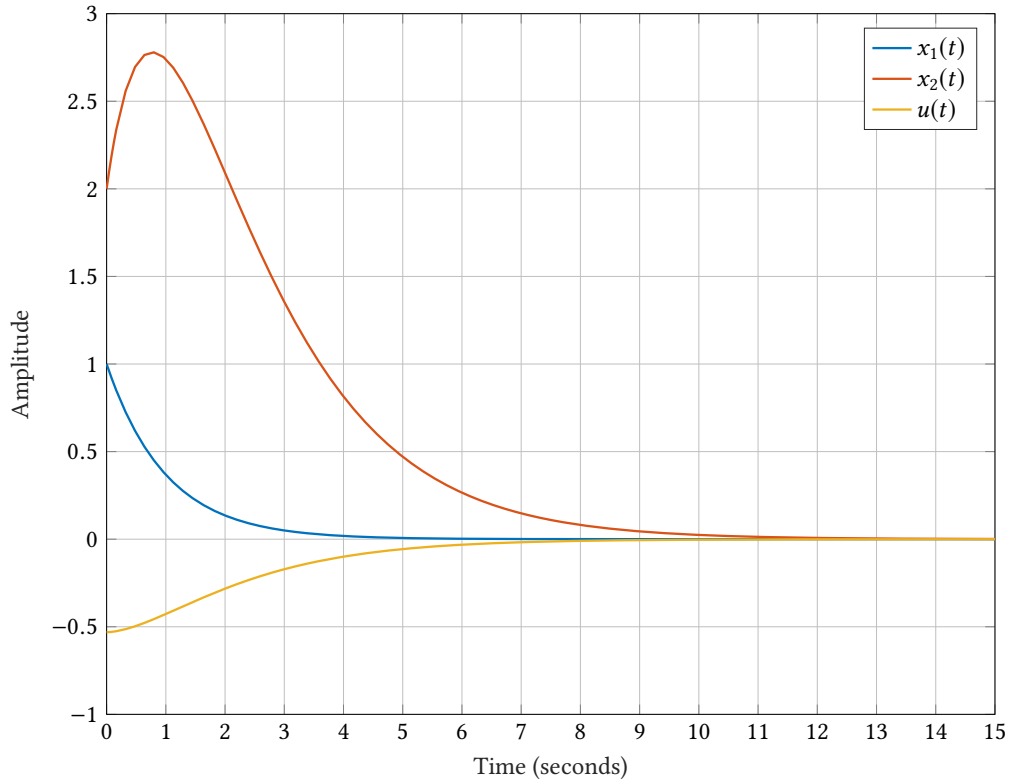
είναι όντως η ζητούμενη.

Ο κώδικας για το παράδειγμα της μεθόδου εγγυημένου κόστους με γραμμικό φράγμα και αβεβαιότητες στον πίνακα των καταστάσεων παρατίθεται παρακάτω.

```

1 function [A, B, R, P, Pnot, Gc] = gcc_linear_bound()
2     A = [-1, 0; 1, -2];
3     B = [0; 1];
4     Q = eye(length(A));
5     R = 10;
6
7     A_1 = @(r1) r1*[0, 0; 3.0, 0];
8     A_2 = @(r2) r2*[0, 0; 0, 1.5];
9     DA = @(r1, r2) A_1(r1) + A_2(r2);
10    U = @(P, r1, r2, e) e*P + ...
11        (1/e)*(DA(r1, r2)'*P*DA(r1, r2));
12
13    e = 1;
14    Pnot = care(A, B, Q, R);
15    [P, fval] = fsolve(@(p) gare(p, A, B, Q, R, ...
16        U(p, 1, 1, e)), Pnot);
17
18    [P_t, fval_t] = fsolve(@(p_t) gcc_lyap(p_t, A, B, Q, R, ...
19        U(p_t, 1, 1, e)), P);

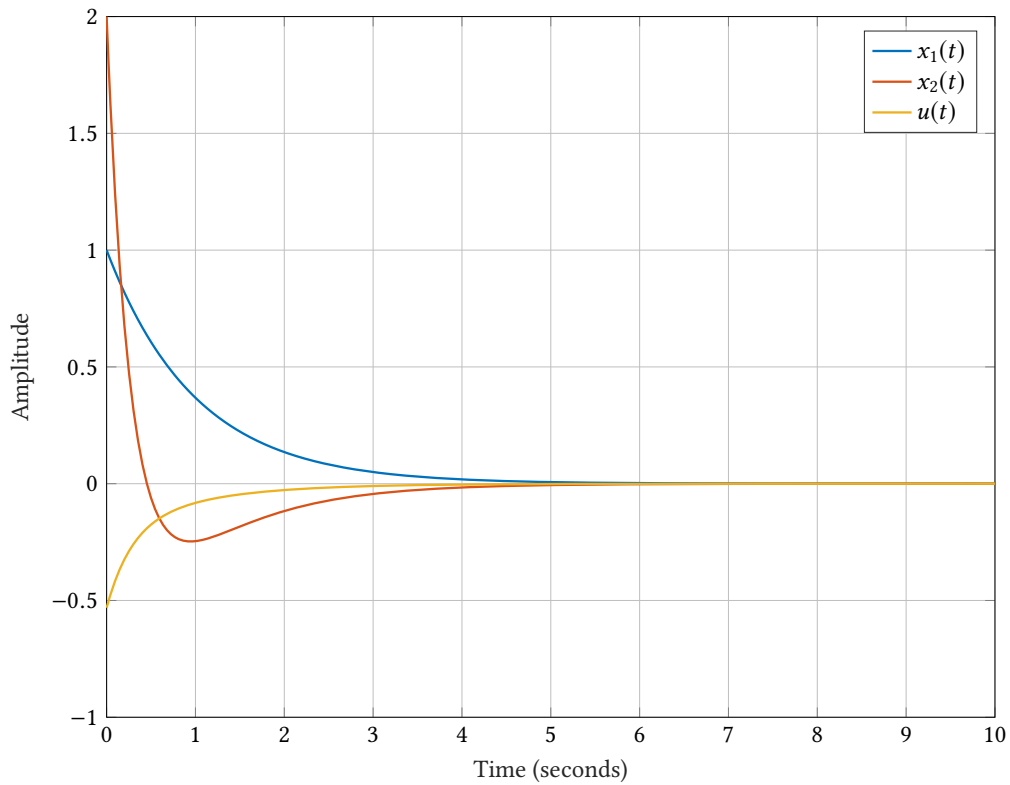
```

Σχήμα 4.2: Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1$

```

20
21     Gc = @(r1, r2) A + DA(r1, r2) - B*(R\ (B'*P));
22     makeplot(A, B, R, P, Pnot, Gc);
23 end
24
25 function f = gare(p, A, B, Q, R, U)
26     P = reshape(p(1:length(A)^2), size(A));
27     F = P*A + A'*P - P*B*(R\ (B'*P)) + Q + U;
28     f = F(:);
29 end
30
31 function f = gcc_lyap(p, A, B, Q, R, U)
32     P = reshape(p(1:length(A)^2), size(A));
33     K_t = -R\ (B'*P);
34     A_t = A + B*K_t;
35     Q_t = Q + U;
36     F = P*A_t + A_t'*P + Q_t + K_t'*R*K_t;
37     f = F(:);
38 end
39
40 function dxdt = state(~, x, Gc)
41     dxdt = Gc*x;
42 end
43

```


Σχήμα 4.3: Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1$

```

44 function makeplot(A, B, R, P, Pnot, Gc)
45     u = @(P) -R\ (B'*P);
46     close all;
47     figure(1)
48     [t, x] = ode45(@(t, x) ...
49         state(t, x, A - B*(R\ (B'*Pnot))), [0 10], [1; 2]);
50     plot(t, x, t, u(Pnot)*x', 'LineWidth', 1.0);
51     title('Initial System')
52     legend('x_1(t)', 'x_2(t)', 'u(t)');
53     grid on;
54
55     figure(2)
56     [t, x] = ode45(@(t, x) ...
57         state(t, x, Gc(1, 1)), [0 15], [1; 2]);
58     plot(t, x, t, u(P)*x', 'LineWidth', 1.0);
59     title('Uncertain System r1 = 1, r2 = 1, p1 = 1')
60     legend('x_1(t)', 'x_2(t)', 'u(t)');
61     grid on;
62
63     figure(3)
64     [t, x] = ode45(@(t, x) ...
65         state(t, x, Gc(-1, -1)), [0 10], [1; 2]);
66     plot(t, x, t, u(P)*x', 'LineWidth', 1.0);
67     title('Uncertain System r1 = -1, r2 = -1, p1 = -0.5')

```

```

68     legend('x_1(t)', 'x_2(t)', 'u(t)');
69     grid on;
70 end

```

Στο παραπάνω παράδειγμα που παρουσιάσαμε επιλέξαμε την παράμετρο $\varepsilon = 1$. Όπως αναφέραμε, μέσω της παραμέτρου αυτής μπορούμε να ικανοποιήσουμε περαιτέρω προδιαγραφές σχεδίασης. Όμως η σχέση του άνω φράγματος (4.8), δεν εξαρτάται γραμμικά από την παράμετρο ε και σίγουρα δεν είναι τετριμμένο το πως επηρεάζει τη λύση της GARE (4.5). Για το λόγο αυτό υλοποιήθηκε πρόγραμμα στο MATLAB όπου ελέγχει την ύπαρξη θετικά ορισμένου πίνακα P της GARE για το γραμμικό φράγμα, σχέση (4.9), για τις διάφορες τιμές του ε .

Τελικά, εκτελώντας τον αντίστοιχο κώδικα βρίσκουμε ότι για τις τιμές της παραμέτρου $\varepsilon = (0.7, 0.8, 0.9, 1.0)$ υπάρχει θετικά ορισμένος πίνακας P της GARE, σχέση (4.9), και επομένως για τις κανονικοποιημένες αποδεκτές αβεβαιότητες το κλειστό σύστημα του συγκεκριμένου παραδείγματος είναι ευσταθές.

Ο κώδικας για το παράδειγμα της μεθόδου εγγυημένου κόστους με γραμμικό φράγμα και αβεβαιότητες στον πίνακα των καταστάσεων για τις διάφορες τιμές της παραμέτρου ε παρατίθεται παρακάτω.

```

1 function [err, tr_p] = gcc_linear_bound_all()
2     A = [-1, 0; 1, -2];
3     B = [0; 1];
4     Q = eye(size(A, 1));
5     R = 10;
6
7     A_1 = @(r1) r1*[0, 0; 3.0, 0];
8     A_2 = @(r2) r2*[0, 0; 0, 1.5];
9     DA = @(r1, r2) A_1(r1) + A_2(r2);
10    U = @(P, r1, r2, e) e*P + ...
11        (1/e)*(DA(r1, r2)'*P*DA(r1, r2));
12
13    options = optimset('Display','off');
14    err = [];
15    Pnot = care(A, B, Q, R);
16    cnt = 0;
17    for e = 0.1:0.1:1
18        [P, fval] = fsolve(@(p) gare(p, A, B, Q, R, ...
19            U(p, 1, 1, e)), Pnot, options);
20        if all(eig(P) > eps)
21            eig(P)
22            cnt = cnt + 1;
23            err(cnt) = e;
24            tr_p(cnt) = trace(P);
25    end

```

```

26     end
27 end
28
29 function f = gare(p, A, B, Q, R, U)
30     P = reshape(p(1:length(A)^2), size(A));
31     F = P*A + A'*P - P*B*(R\ (B'*P)) + Q + U;
32     f = F(:);
33 end

```

Με τον κώδικα που παρουσιάσαμε παραπάνω, είναι πολύ εύκολο προσθέτοντας δύο βρόχους (for loops), κάποιος βρει τις θετικά ορισμένες λύσεις P της GARE συναρτήσει των μεταβολών των αβεβαιοτήτων $A_1(r_1), A_2(r_2)$. Η μετατροπή αυτή είναι τετριμμένη και για αυτό απλά την αναφέρουμε χωρίς να παραθέσουμε τον αντίστοιχο κώδικα.

4.2 Συστήματα με αβεβαιότητα στον πίνακα κατάστασης και στον πίνακα εισόδου

Θα μελετήσουμε την περίπτωση όπου έχουμε αβεβαιότητες στον πίνακα κατάστασης αλλά και στον πίνακα εισόδου, δηλαδή ισχύει $\Delta A \neq 0$ και $\Delta B \neq 0$. Τότε διακρίνουμε δύο περιπτώσεις φράγματος,

- (α) αβεβαιότητες που ικανοποιούν τις συνθήκες προσαρμογής,
- (β) γενική περίπτωση.

4.2.1 Αβεβαιότητες που ικανοποιούν τις συνθήκες προσαρμογής

Σύμφωνα με τη δημοσίευση [1], αν υποθέσουμε ότι οι αβεβαιότητες ικανοποιούν τις συνθήκες προσαρμογής, δηλαδή

$$\Delta A = \sum_{i=1}^k B \tilde{A}_i r_i, \quad \Delta B = \sum_{i=1}^l B \tilde{B}_i p_i,$$

όπου B είναι ο πίνακας εισόδου και \tilde{A}_i, \tilde{B}_i είναι σταθεροί πίνακες κατάλληλων διαστάσεων. Η υπόθεση αυτή δεν είναι ιδιαίτερα ισχυρή. Εύκολα αποδεικνύεται ότι εφόσον μπορεί οι πίνακες A, B να γραφτούν σε κανονική μορφή φάσης τότε οι αβεβαιότητες μπορούν σε γραφτούν σε μορφή που ικανοποιούν τις συνθήκες προσαρμογής. Στη συνέχεια ορίζουμε

$$\Theta_i = a_i^{-2} I_m, \quad \Xi_i = a_i^2 \tilde{A}_i^T \tilde{A}_i,$$

με $i = 1, \dots, k$ και

$$\Phi_i = b_i^{-2} I_m, \quad \Psi_i = b_i^2 R^{-1} \tilde{B}_i^T \tilde{B}_i R^{-1},$$

με $i = 1, \dots, l$, όπου a_i, b_i αυθαίρετες θετικές ποσότητες που ικανοποιούν περαιτέρω προδιαγραφές σχεδίασης. Τότε η

$$\mathcal{U}(P) = \sum_{i=1}^k \left(\Xi_i + PB\Theta_i B^T P \right) + \sum_{i=1}^l \left[PB(\Phi_i + \Psi_i) B^T P \right],$$

είναι μία συνάρτηση άνω φράγματος. Το πλεονέκτημα που μας δίνει η παραπάνω έκφραση είναι ότι η GARE, σχέση (4.5), μπορεί να γραφτεί στη μορφή μίας κανονικής αλγεβρικής εξίσωσης Riccati

$$PA + A^T P - PB\hat{R}^{-1}B^T P + \hat{Q} = 0, \quad (4.10)$$

όπου

$$\hat{Q} = Q + \sum_{i=1}^k \Xi_i, \quad \hat{R}^{-1} = R^{-1} - \sum_{i=1}^k \Theta_i + \sum_{i=1}^l (\Phi_i + \Psi_i). \quad (4.11)$$

Το πρόβλημα συνεπώς εκπίπτει σε πρόβλημα γραμμικού τετραγωνικού ρυθμιστή και έχει μοναδική θετικά ορισμένη λύση P , όταν το πίνακας \hat{R} είναι θετικά ορισμένος.

Θα εφαρμόσουμε τα παραπάνω σε ένα παράδειγμα. Θα θεωρήσουμε το σύστημα που είδαμε προηγουμένως, δηλαδή

$$\dot{x} = Ax + Bu,$$

όπου οι πίνακες A και B είναι

$$A = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Τα μητρώα στάθμισης του δείκτη απόδοσης, σχέση (4.2), επιλέχθηκαν $Q = I_3$ και $R = 10$. Σε αντίθεση με το προηγούμενο παράδειγμα θα επιλέξουμε πιο ακραίες αβεβαιότητες. Συνεπώς, επιλέξαμε για τον πίνακα κατάστασης

$$\tilde{A}_1 = \begin{bmatrix} 5.5 & 0 \end{bmatrix}, \quad \tilde{A}_2 = \begin{bmatrix} 0 & 2 \end{bmatrix},$$

όπου σύμφωνα με τους συμβολισμούς έχουμε $k = 2$ και για τον πίνακα ελέγχου

$$\tilde{B} = 2.5,$$

όπου έχουμε $l = 1$. Άρα οι αβεβαιότητες είναι

$$\Delta A(r_1, r_2) = B \left(\tilde{A}_1 r_1 + \tilde{A}_2 r_2 \right), \quad \Delta B(p_1) = \tilde{B} p_1.$$

Χωρίς μείωση της γενικότητας βρούμε τον εγγυημένο νόμο ελέγχου για τις κανονικοποιημένες αποδεκτές αβεβαιότητες, δηλαδή $|r_1| \leq 1, |r_2| \leq 1$ και $|p_1| \leq 1$. Σαν πρώτη θεώρηση θα επιλέξουμε τις βαθμωτές σταθερές ίσες με $a_1 = a_2 = 1$ και $b_1 = 1$, και στη συνέχεια κάποιος μπορεί μέσω των ποσοτήτων αυτών να βελτιστοποιήσει ως προς κάποια έννοια τα

αποτελέσματα μας. Τελικά, πραγματοποιώντας τους υπολογισμούς βρίσκουμε

$$\Theta = \sum_{i=1}^l \Theta_i = 2, \quad \Xi = \sum_{i=1}^l \Xi_i = \begin{bmatrix} 30.25 & 0 \\ 0 & 4 \end{bmatrix}$$

και

$$\Phi = \sum_{i=1}^l \Phi_i = 2, \quad \Psi = \sum_{i=1}^l \Psi_i = 0.0625.$$

Ακόμα από τις σχέσεις (4.11) προκύπτει

$$\hat{Q} = \begin{bmatrix} 31.25 & 0 \\ 0 & 5 \end{bmatrix}, \quad \hat{R}^{-1} = 0.1625.$$

Ο θετικά ορισμένος πίνακας P της γενικευμένης αλγεβρικής εξίσωσης Riccati μπορεί να βρεθεί με την εντολή `care` του MATLAB, με ορίσματα τους πίνακες (A, B, \hat{Q}, \hat{R}) . Έτσι βρίσκουμε ότι ο πίνακας P είναι

$$P = \begin{bmatrix} 15.987 & 0.3733 \\ 0.3733 & 1.1923 \end{bmatrix}.$$

Όπως παρατηρούμε ο πίνακας είναι συμμετρικός και θετικά ορισμένος, με ιδιοτιμές $\lambda_{1,2} = 1.1828, 15.9964$. Εφόσον η σχέση (4.10) είναι της μορφής του γραμμικού τετραγωνικού ρυθμιστή LQR, ως γνωστόν ο νόμος ελέγχου εγγυημένου κόστους υπολογίζεται

$$\tilde{u}(t) = -\hat{R}^{-1} B^T P x(t) = \begin{bmatrix} -0.0607 & -0.193 \end{bmatrix} x(t).$$

Το εγγυημένο κόστος υπολογίζεται από τη σχέση (4.4)

$$\tilde{J} = \text{tr } P = 17.1792$$

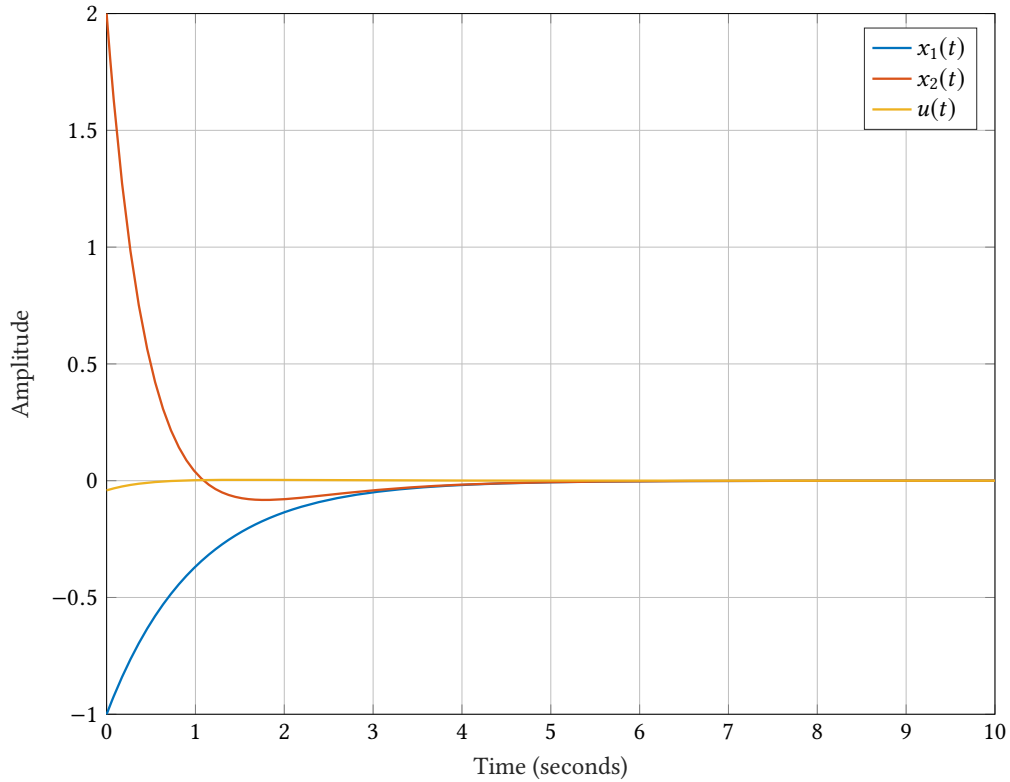
ενώ από τη σχέση που εξαρτάται από τις αρχικές συνθήκες

$$\tilde{J}(x_0) = x_0^T P x_0 = \begin{bmatrix} -1 & 2 \end{bmatrix} P \begin{bmatrix} -1 & 2 \end{bmatrix}^T = 19.2628.$$

Στα σχήματα 4.4, 4.5 και 4.6 παρουσιάζονται οι καταστάσεις του συστήματος για τις διάφορες τιμές των αβεβαιοτήτων καθώς και ο νόμος ελέγχου εγγυημένου κόστους.

Ο κώδικας για το παράδειγμα της μεθόδου εγγυημένου κόστους με τετραγωνικό φράγμα και αβεβαιότητες στον πίνακα των καταστάσεων αλλά και ελέγχου παρατίθεται παρακάτω.

```
1 function [A, B, R, R_t_inv, P, Pnot, Gc] = gcc_quad_bound();
2     A = [-1, 0; 1, -2];
3     B = [0; 1];
4     Q = eye(length(A));
5     R = 10;
6
7     A_t1 = [5.5, 0];
```

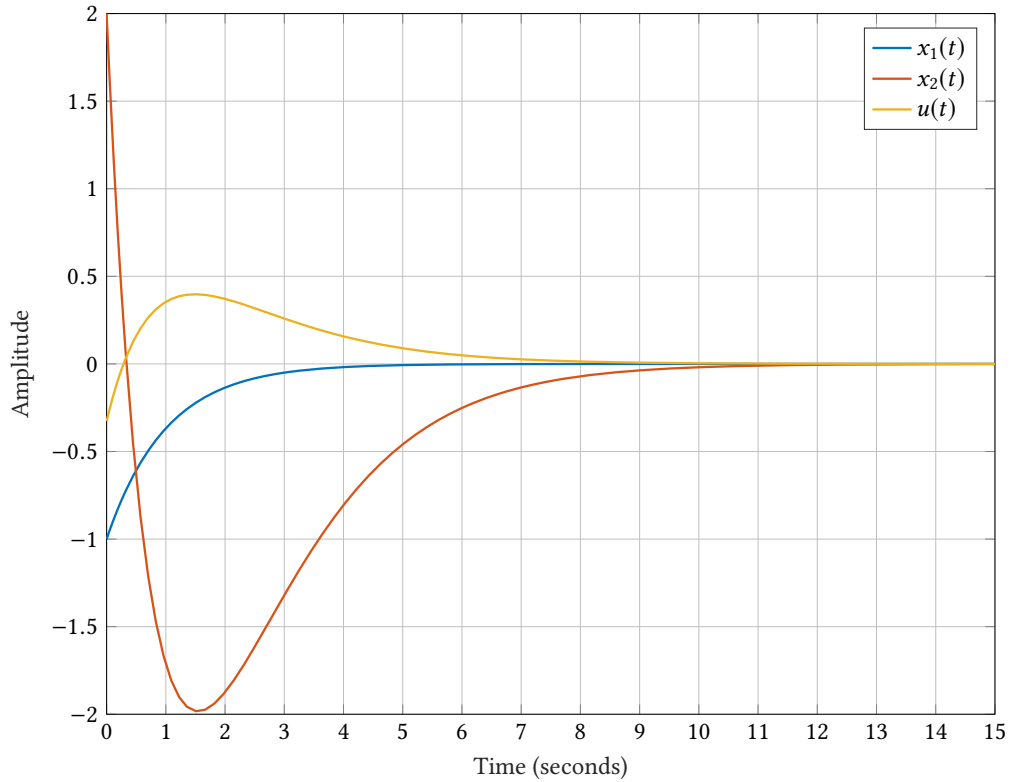


Σχήμα 4.4: Αρχικό σύστημα χωρίς αβεβαιότητες

```

8   A_t2 = [0, 2.0];
9   B_t1 = 2.5;
10  DA = @(r1, r2) B*(A_t1*r1 + A_t2*r2);
11  DB = @(p1) B*B_t1*p1;
12  Pnot = care(A, B, Q, R);
13
14  a = 1;
15  b = 1;
16
17  theta_ = (1/a^2 + 1/a^2)*eye(size(B, 2));
18  ksi_ = a^2*(A_t1')*A_t1 + a^2*(A_t2')*A_t2;
19  phi_ = (1/b^2 + 1/b^2)*eye(size(B, 2));
20  psi_ = b^2*R\ (B_t1'*(B_t1/R));
21
22  Q_t = Q + ksi_;
23  R_t_inv = inv(R) - theta_ + phi_ + psi_;
24  P = care(A, B, Q_t, inv(R_t_inv));
25  Gc = @(r1, r2, p1) ...
26      A + DA(r1, r2) - (B + DB(p1))*R_t_inv*B'*P;
27  makeplot(A, B, R, R_t_inv, P, Pnot, Gc);
28 end
29
30 function dxdt = state(~, x, Gc)
31     dxdt = Gc*x;

```

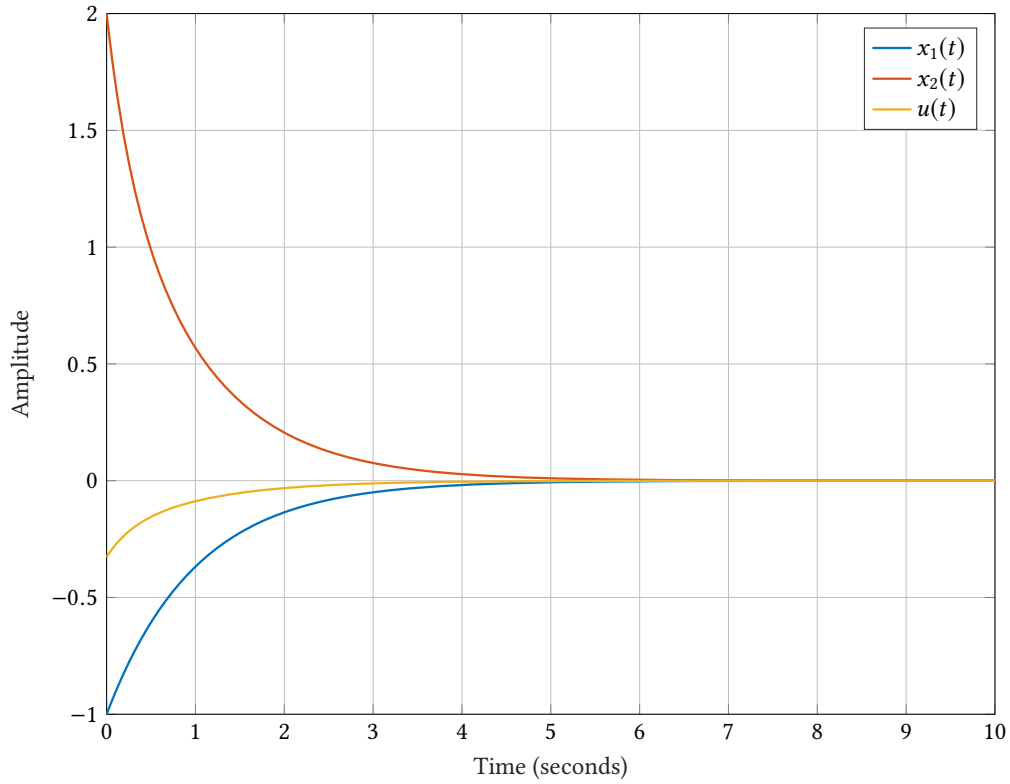


Σχήμα 4.5: Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, p_1 = 1$

```

32 end
33
34 function makeplot(A, B, R, R_t_inv, P, Pnot, Gc)
35     Knot = -R\ (B'*Pnot);
36     K = -R_t_inv*B'*P;
37     close all;
38     figure(1)
39     [t, x] = ode45(@(t, x) ...
40         state(t, x, A - B*(R\ (B'*Pnot))), [0 10], [-1; 2]);
41     plot(t, x, t, Knot*x', 'LineWidth', 1.0);
42     title('Initial System')
43     legend('x_1(t)', 'x_2(t)', 'u(t)');
44     grid on;
45
46     figure(2)
47     [t, x] = ode45(@(t, x) ...
48         state(t, x, Gc(1, 1, 1)), [0 15], [-1; 2]);
49     plot(t, x, t, K*x', 'LineWidth', 1.0);
50     title('Uncertain System r1 = 1, r2 = 1, p1 = 1')
51     legend('x_1(t)', 'x_2(t)', 'u(t)');
52     grid on;
53
54     figure(3)
55     [t, x] = ode45(@(t, x) ...

```



Σχήμα 4.6: Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, p_1 = -0.5$

```

56     state(t, x, Gc(-1, -1, -0.5)), [0 10], [-1; 2]);
57     plot(t, x, t, K*x', 'LineWidth', 1.0);
58     title('Uncertain System r1 = -1, r2 = -1, p1 = -0.5')
59     legend('x_1(t)', 'x_2(t)', 'u(t)');
60     grid on;
61 end

```

4.2.2 Γενική περίπτωση

Στην περίπτωση αυτή οι αβεβαιότητες μπορούν να γραφτούν στη μορφή

$$A_i = d_i e_i^T, \quad B_j = f_j g_j^T,$$

όπου $i = 1, \dots, k$ και $j = 1, \dots, l$ και d_i, e_i, f_j, g_j διανύσματα κατάλληλων διαστάσεων. Είναι προφανές ότι η επιλογή των διανυσμάτων δεν είναι μοναδική. Στη συνέχεια ορίζονται οι συμμετρικοί και θετικά ορισμένοι πίνακες

$$T = \sum_{i=1}^k d_i d_i^T, \quad U = \sum_{i=1}^k e_i e_i^T,$$

και

$$V = \sum_{i=1}^l g_i g_i^T, \quad W = \sum_{i=1}^l f_i f_i^T.$$

Αποδεικνύεται ότι η παρακάτω έκφραση

$$\mathcal{U}(P) = PTP + U + PWP + PBR^{-1}VR^{-1}B^T P$$

αποτελεί συνάρτηση άνω φράγματος της (4.6). Η γενικευμένη αλγεβρική εξίσωση Riccati γίνεται επομένως

$$PA + A^T P - P \left(\varepsilon^{-1} BR^{-1} B^T - \varepsilon^{-2} BR^{-1} VR^{-1} B - W - T \right) P + Q + \mathcal{U}(P) = 0. \quad (4.12)$$

Η παραπάνω είναι μη γραμμική αλγεβρική εξίσωση ως προς P . Δεν έχουμε γενική αναλυτική λύση αλλά ούτε και συνθήκες ύπαρξης λύσης. Στόχος είναι η εύρεση θετικά ορισμένου συμμετρικού πίνακα P . Η επίλυση της υλοποιήθηκε με τη μέθοδο Newton-Raphson, όπως και στην περίπτωση του γραμμικού φράγματος. Τα τεχνικά της μεθόδου αναφέρονται παραπάνω και δεν θα τα επαναλάβουμε, καθώς η διαδικασία είναι όμοια. Ακόμα προσθέσαμε την παράμετρο $\varepsilon \in (0, 1]$ ούτως ώστε να ικανοποιήσουμε περαιτέρω κριτήρια σχεδιασμού. Συνεπώς, ο νόμος ελέγχου εγγυημένου κόστους, όπως και προηγούμενος είναι

$$u(t) = -\varepsilon^{-1} R^{-1} B^T P x(t). \quad (4.13)$$

Αποδεικνύεται ότι η ικανή και αναγκαία συνθήκη ώστε η GARE να έχει λύση, δηλαδή να βρούμε θετικά ορισμένο συμμετρικό πίνακα P , είναι αν και μόνο αν ο πίνακας H ,

$$H = \begin{bmatrix} A & -M(\varepsilon) \\ -Q - U & -A^T \end{bmatrix},$$

όπου

$$M(\varepsilon) = \varepsilon^{-1} BR^{-1} B^T - \varepsilon^{-2} BR^{-1} VR^{-1} B - W - T,$$

δεν έχει καθαρά φανταστικές ιδιοτιμές. Παρόλο αυτά, αξίζει να σημειωθεί ότι η GARE, σχέση (4.12), λόγω της μη γραμμικότητας μπορεί να έχει λύσεις με P όχι συμμετρικό που σταθεροποιούν το κλειστό σύστημα.

Όπως και στις προηγούμενες μεθόδους, θα παρουσιάσουμε ένα παράδειγμα. Αυτή τη φορά θα διαφοροποιήσουμε το σύστημα. Θα θεωρήσουμε το σύστημα

$$\dot{x} = Ax + Bu,$$

όπου οι πίνακες A και B είναι

$$A = \begin{bmatrix} -1 & 2 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Τα μητρώα στάθμισης του δείκτη απόδοσης, σχέση (4.2), επιλέχθηκαν $Q = I_3$ και $R = 10$. Οι

αβεβαιότητες για τον πίνακα κατάστασης επιλέχθηκαν

$$A_1 = \begin{bmatrix} 0.54 & 0.18 \\ 0.18 & 0.06 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.04 & 0.09 \\ 0.12 & 0.27 \end{bmatrix},$$

που οδηγεί

$$\Delta A(r_1, r_2) = r_1 A_1 + r_2 A_2,$$

και για τον πίνακα ελέγχου έχουμε

$$B_1 = \begin{bmatrix} 0.18 \\ 0.12 \end{bmatrix}$$

και άρα

$$\Delta B(p_1) = p_1 B_1.$$

Οι αβεβαιότητες συνεπώς μπορούν να γραφτούν

$$d_1 = \begin{bmatrix} 0.9 \\ 0.3 \end{bmatrix}, \quad e_1 = \begin{bmatrix} 0.6 \\ 0.2 \end{bmatrix}, \quad d_2 = \begin{bmatrix} 0.1 \\ 0.3 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0.4 \\ 0.9 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}, \quad g_1 = 0.6.$$

Ακόμα σύμφωνα με τα παραπάνω ορίζονται οι συμμετρικοί και θετικά ορισμένοι πίνακες

$$T = \begin{bmatrix} 0.82 & 0.3 \\ 0.3 & 0.18 \end{bmatrix}, \quad U = \begin{bmatrix} 0.52 & 0.48 \\ 0.48 & 0.85 \end{bmatrix},$$

και

$$V = 0.36, \quad W = \begin{bmatrix} 0.09 & 0.06 \\ 0.06 & 0.04 \end{bmatrix}.$$

Στο παράδειγμα επιλέξαμε την παράμετρο $\varepsilon = 1$ και δίχως μείωση της γενικότητας βρήκαμε το εγγυημένο κόστος για τα κανονικοποιημένα όρια των αποδεκτών αβεβαιοτήτων, δηλαδή για $|r_1| \leq 1$, $|r_2| \leq 1$ και $|p_1| \leq 1$. Στη συνέχεια κάποιος μπορεί να λύσει την GARE (4.12), για τις διάφορες τιμές του ε , όπως κάναμε στην περίπτωση του γραμμικού φράγματος.

Ο θετικά ορισμένος πίνακας P της γενικευμένης αλγεβρικής εξίσωσης Riccati υπολογίσθηκε με την εντολή `fsolve` του MATLAB, υλοποιεί τη μέθοδο Newton-Raphson. Έτσι βρίσκουμε ότι ο πίνακας P είναι

$$P = \begin{bmatrix} 0.7806 & 0.3686 \\ 0.3686 & 2.3976 \end{bmatrix}.$$

Όπως παρατηρούμε ο πίνακας είναι συμμετρικός και θετικά ορισμένος, με ιδιοτιμές $\lambda_{1,2} = 0.7005, 2.4777$. Ακόμα από τη σχέση (4.13) ο νόμος ελέγχου εγγυημένου κόστους υπολογίζεται

$$\tilde{u}(t) = \begin{bmatrix} -0.0369 & -0.2398 \end{bmatrix} x(t).$$

Το εγγυημένο κόστος υπολογίζεται από τη σχέση (4.4)

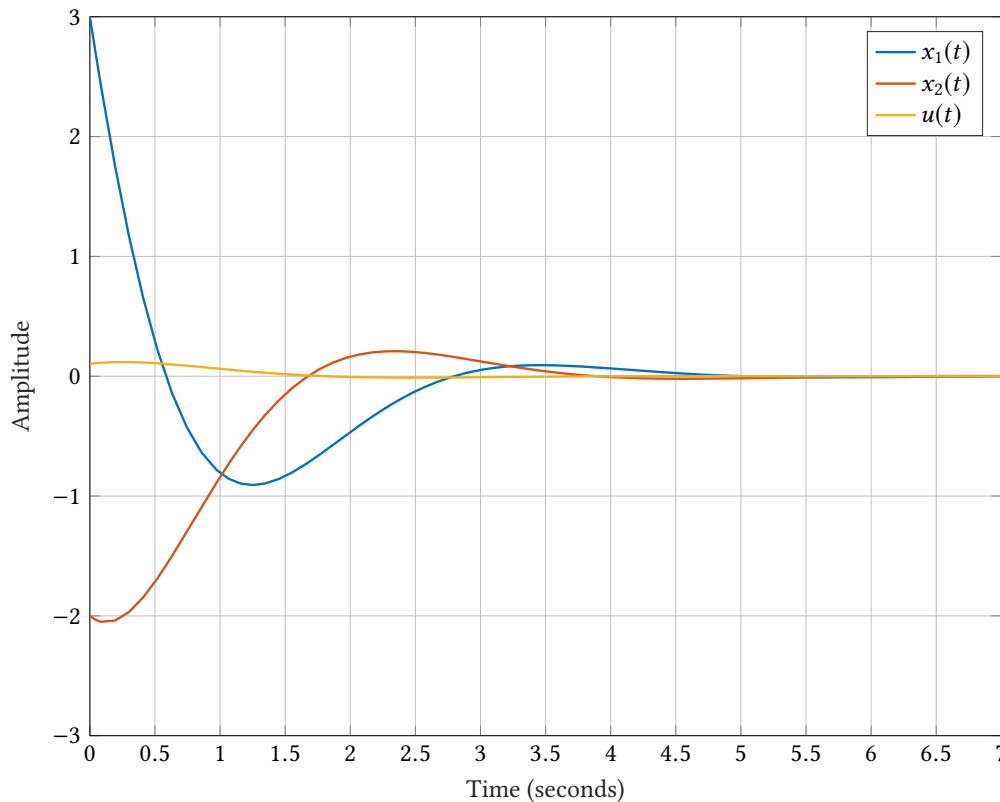
$$\tilde{J} = \text{tr } P = 3.1781$$

ενώ από τη σχέση που εξαρτάται από τις αρχικές συνθήκες

$$\tilde{J}(x_0) = x_0^T P x_0 = \begin{bmatrix} 3 & -2 \end{bmatrix} P \begin{bmatrix} 3 & -2 \end{bmatrix}^T = 12.1916.$$

Ακόμη οι ιδιοτιμές του πίνακα H δεν είναι καθαρά φανταστικές και επομένως ικανοποιείται η συνθήκη της ικανής και αναγκαίας συνθήκης για την ύπαρξη λύσης, κάτι που επιβεβαιώνει τους υπολογισμούς που πραγματοποιήθηκαν.

Στα σχήματα 4.7, 4.8 και 4.9 παρουσιάζονται οι καταστάσεις του συστήματος για τις διάφορες τιμές των αβεβαιοτήτων καθώς και ο νόμος ελέγχου εγγυημένου κόστους.



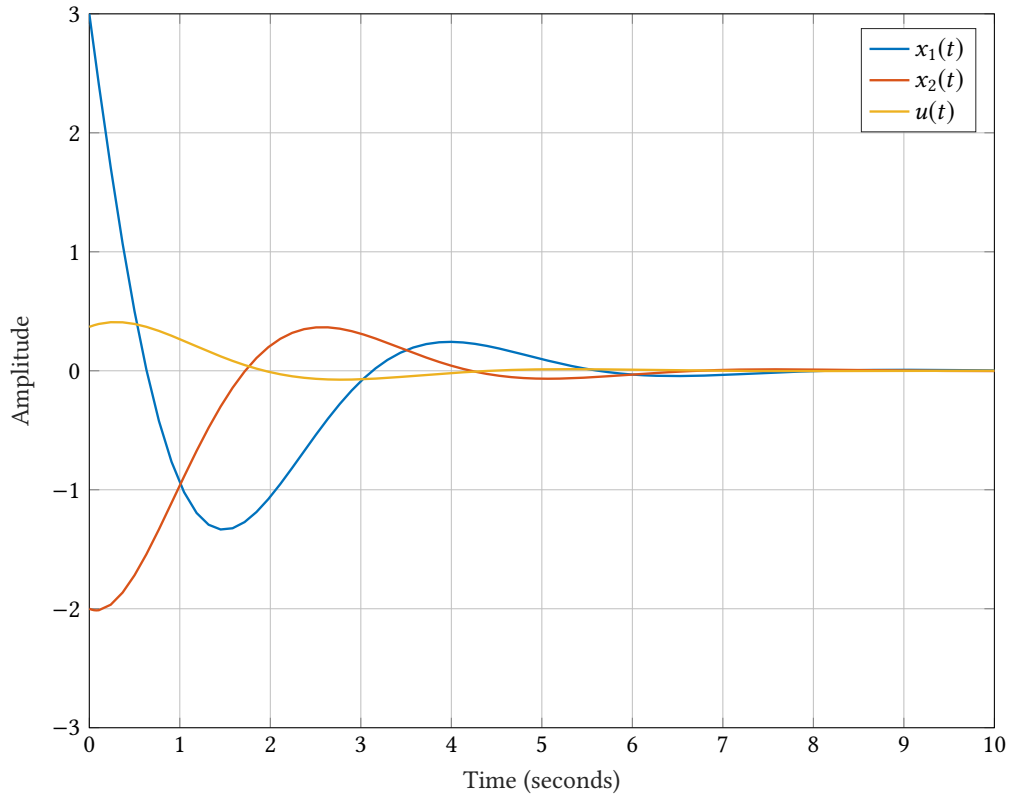
Σχήμα 4.7: Αρχικό σύστημα χωρίς αβεβαιότητες

Ο κώδικας για το παράδειγμα της μεθόδου εγγυημένου κόστους για τη γενική περίπτωση με αβεβαιότητες στον πίνακα των καταστάσεων και πίνακα ελέγχου παρατίθεται παρακάτω.

```

1 function [A, B, R, P, Pnot, Gc, e] = gcc_quad_bound_gen();
2     A = [-1, 2; -1, -1];
3     B = [0; 1];
4     Q = eye(length(A));
5     R = 10;
6
7     d_t1 = [0.9; 0.3];
8     e_t1 = [0.6; 0.2];
9     d_t2 = [0.1; 0.3];

```

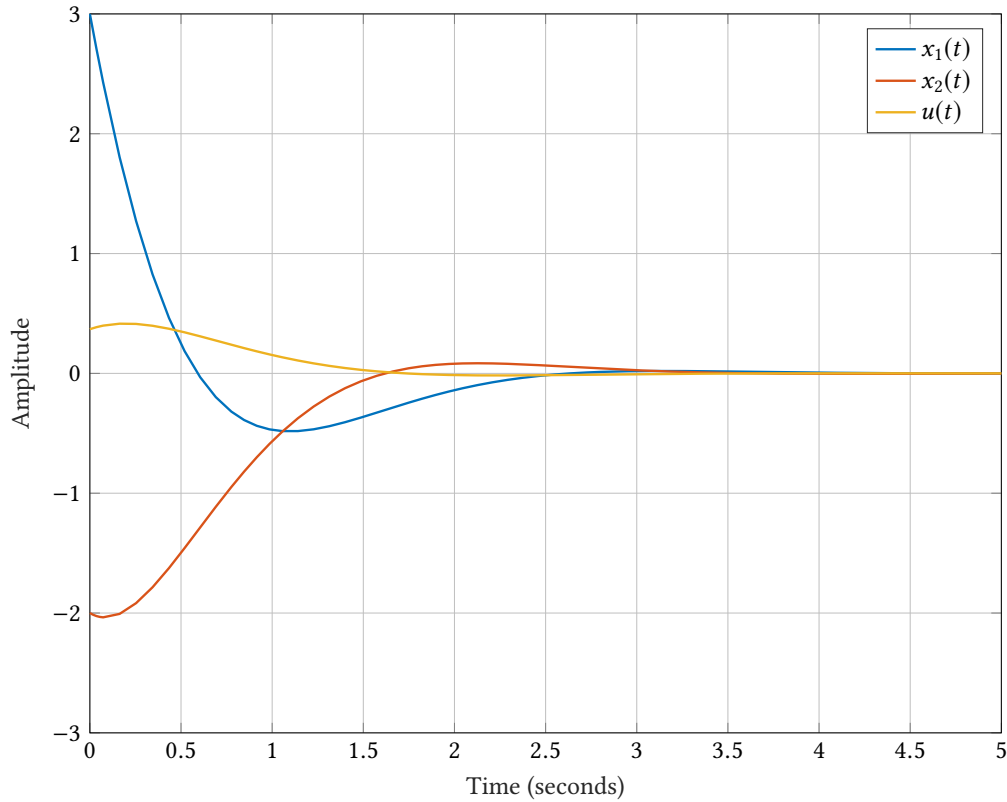


Σχήμα 4.8: Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, p_1 = 1$

```

10  e_t2 = [0.4; 0.9];
11  f_t1 = [0.3; 0.2];
12  g_t1 = 0.6;
13
14  DA = @(r1, r2) r1*d_t1*e_t1' + r2*d_t2*e_t2';
15  DB = @(p1) p1*f_t1*g_t1';
16
17  T = d_t1*d_t1' + d_t2*d_t2';
18  U = e_t1*e_t1' + e_t2*e_t2';
19  V = g_t1*g_t1';
20  W = f_t1*f_t1';
21
22  e = 1;
23  M = @(e) (1/e)*B*(R\B') ...
24      - (1/e^2)*B*(R\((V*(R\B')))) - W - T;
25  H = [A, -M(e); -Q - U, -A'];
26  lambdaH = eig(H);
27
28  Pnot = care(A, B, Q, R);
29  [P, fval] = fsolve(@(p) gare(p, A, M(e), Q, U), Pnot);
30  if all(eig(P) > eps)
31      Gc = @(r1, r2, p1) A + DA(r1, r2) ...
32          - (1/e)*(B + DB(p1))*(R\((B'*P)));

```



Σχήμα 4.9: Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, p_1 = -1$

```

33     makeplot(A, B, R, P, Pnot, Gc, e);
34     end
35 end
36
37 function f = gare(p, A, M, Q, U)
38     P = reshape(p(1:length(A)^2), size(A));
39     F = P*A + A'*P - P*M*P + Q + U;
40     f = F(:);
41 end
42
43 function dxdt = state(~, x, Gc)
44     dxdt = Gc*x;
45 end
46
47 function makeplot(A, B, R, P, Pnot, Gc, e)
48     u = @(P) -R\ (B'*P);
49     close all;
50     figure(1)
51     [t, x] = ode45(@(t, x) ...
52         state(t, x, A - B*(R\ (B'*Pnot))), [0 7], [3; -2]);
53     plot(t, x, t, u(Pnot)*x', 'LineWidth', 1.0);
54     title('Initial System')
55     legend('x_1(t)', 'x_2(t)', 'u(t)');

```

```

56     grid on;
57
58     figure(2)
59     [t, x] = ode45(@(t, x) ...
60         state(t, x, Gc(1, 1, 1)), [0 10], [3; -2]);
61     plot(t, x, t, (1/e)*u(P)*x', 'LineWidth', 1.0);
62     title('Uncertain System r1 = 1, r2 = 1, p1 = 1')
63     legend('x_1(t)', 'x_2(t)', 'u(t)');
64     grid on;
65
66     figure(3)
67     [t, x] = ode45(@(t, x) ...
68         state(t, x, Gc(-1, -1, -1)), [0 5], [3; -2]);
69     plot(t, x, t, (1/e)*u(P)*x', 'LineWidth', 1.0);
70     title('Uncertain System r1 = -1, r2 = -1, p1 = -1')
71     legend('x_1(t)', 'x_2(t)', 'u(t)');
72     grid on;
73 end

```

4.3 Ελαχιστοποίηση του εγγυημένου κόστους

Η προσέγγιση μέχρι τώρα της επίλυσης των γενικευμένων αλγεβρικών εξισώσεων Riccati οδηγεί σε υψηλές τιμές του εγγυημένου κόστους. Αυτό διότι το εγγυημένο κόστος είναι ανάλογο της λύσης P της GARE. Επομένως θα μπορούσαμε να εκφράσουμε το πρόβλημα ως ένα πρόβλημα ελαχιστοποίησης του εγγυημένου κόστους, ή ισοδύναμα του ίχνους του P , σχέση (4.4). Η κατάστρωση ενός τέτοιου προβλήματος ελαχιστοποίησης είναι μία δύσκολη διαδικασία και γίνεται ακόμα δυσκολότερη καθώς αυξάνονται οι διαστάσεις του προβλήματος. Όμως αν διατυπώσουμε το πρόβλημα ως ένα πρόβλημα κυρτής βελτιστοποίησης, μέσω των γραμμικών ανισοτήτων πινάκων LMI τότε μπορούμε με ευκολία να καταστρώσουμε και να επιλύσουμε το εν λόγω πρόβλημα.

Οι αβεβαιότητες μπορούν να γραφτούν στη μορφή

$$A_i = \sigma_i^{1/2} d_i \sigma_i^{-1/2} e_i^T, \quad B_j = \tau_j^{1/2} f_j \tau_j^{-1/2} g_j^T,$$

όπου $i = 1, \dots, k$ και $j = 1, \dots, l$ και d_i, e_i, f_j, g_j διανύσματα κατάλληλων διαστάσεων και σ_i, τ_i είναι θετικές βαθμωτές ποσότητες. Στην προηγούμενη μέθοδο είχαμε επισημάνει ότι η επιλογή των διανυσμάτων δεν είναι μοναδική. Έτσι μπορούμε να θεωρήσουμε ως μεταβλητές του προβλήματος ελαχιστοποίησης τις ποσότητες σ_i, τ_i . Στη συνέχεια ορίζονται οι πίνακες

$$D = \begin{bmatrix} d_1 & \dots & d_k \end{bmatrix}, \quad E = \begin{bmatrix} e_1 & \dots & e_k \end{bmatrix}^T,$$

και

$$F = \begin{bmatrix} f_1 & \dots & f_l \end{bmatrix}, \quad G = \begin{bmatrix} g_1 & \dots & g_l \end{bmatrix}.$$

Ακόμα ορίζονται οι διαγώνιοι πίνακες

$$\tilde{S} = \text{diag}(\sigma_1 \dots \sigma_k), \quad \tilde{T} = \text{diag}(\tau_1 \dots \tau_l).$$

Η GARE (4.12) έχει μία θετικά ορισμένη λύση τέτοια ώστε το αντίστοιχο εγγυημένο κόστος να γίνεται ελάχιστο, αν ισχύει το πρόβλημα ελαχιστοποίησης που διατυπώνεται στο παρακάτω θεώρημα.

Θεώρημα 6 (Θεώρημα 3.5 της Κοσμίδου [11]). Έστω το αβέβαιο σύστημα (4.1) και το αντίστοιχο τετραγωνικό κριτήριο (4.2). Αν υπάρχουν συμμετρικοί και θετικά ορισμένοι πίνακες \tilde{M}, \tilde{P} , διαγώνιοι και θετικά ορισμένοι πίνακες \tilde{S}, \tilde{T} και ένας θετικός αριθμός δ , τέτοιοι ώστε το ακόλουθο πρόβλημα ελαχιστοποίησης

$$\begin{aligned} & \underset{(\tilde{M}, \tilde{P}, \tilde{S}, \tilde{T}, \delta)}{\text{minimize}} && \text{tr } \tilde{M} \\ & \text{subject to} && \begin{bmatrix} \tilde{M} & I \\ I & \tilde{P} \end{bmatrix} > 0 \\ & && \begin{bmatrix} -\tilde{P}A^T - A\tilde{P} + \delta BR^{-1}B^T - D\tilde{S}D^T - F\tilde{T}F^T & \tilde{P}E^T & \delta BR^{-1}G^T & \tilde{P} \\ & E\tilde{P} & \tilde{S} & 0 & 0 \\ & \delta GR^{-1}B^T & 0 & \tilde{T} & 0 \\ & \tilde{P} & 0 & 0 & Q_0^{-1} \end{bmatrix} > 0, \end{aligned}$$

έχει ένα μη κενό σύνολο εφικτών λύσεων $(\tilde{M}, \tilde{P}, \tilde{S}, \tilde{T}, \delta)$, τότε ο νόμος ελέγχου

$$u^*(t) = -\delta R^{-1}B^T P x(t)$$

είναι ένας νόμος ελέγχου εγγυημένου κόστους και

$$J^* = \text{tr } P$$

είναι το εγγυημένο κόστος για το σύστημα (4.1) με τυχαίες αρχικές συνθήκες, όπου $P = \tilde{P}^{-1}$ και $\delta = 1/\varepsilon$.

Αποδεικνύεται ότι για οποιοδήποτε $Q_0 < Q$, η επίλυση του LMI που διατυπώνεται στο παραπάνω θεώρημα, ισοδυναμεί με την επίλυση της GARE (4.12). Η μεταβλητή \tilde{M} είναι μία εικονική μεταβλητή (dummy variable). Από την πρώτη γραμμική ανισότητα πινάκων, αν πάρουμε το συμπλήρωμα του Schur προκύπτει

$$\tilde{P} > 0 \quad \text{και} \quad \tilde{M} - \tilde{P}^{-1} > 0,$$

ή ισοδύναμα

$$\tilde{M} - P > 0.$$

Επομένως ελαχιστοποιώντας τον πίνακα \tilde{M} , ή αντίστοιχα το $\text{tr } \tilde{M}$, ελαχιστοποιούμε τον πίνακα \tilde{P} , ή αντίστοιχα το $\text{tr } \tilde{P}$ και άρα το εγγυημένο κόστος.

Θα παρουσιάσουμε το παράδειγμα που μελετήσαμε και προηγουμένως για σύγκριση των αποτελεσμάτων. Για ευκολία παραθέτουμε ξανά το σύστημα

$$\dot{x} = Ax + Bu,$$

όπου οι πίνακες A και B είναι

$$A = \begin{bmatrix} -1 & 2 \\ -1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Τα μητρώα στάθμισης του δείκτη απόδοσης, σχέση (4.2), επιλέχθηκαν $Q = I_3$ και $R = 10$. Για το πρόβλημα ελαχιστοποίησης οριακά θεωρούμε $Q_0 = Q$. Οι αβεβαιότητες επιλέχθηκαν και διατυπώθηκαν μέσω των πινάκων

$$D = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.3 \end{bmatrix}, \quad E = \begin{bmatrix} 0.6 & 0.2 \\ 0.4 & 0.9 \end{bmatrix}, \quad F = \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}, \quad G = 0.6.$$

Με τον κώδικα στο πρόγραμμα MATLAB που παραθέτουμε παρακάτω επιλύσαμε το πρόβλημα ελαχιστοποίησης με τις γραμμικές ανισότητες πινάκων. Τελικά, υπολογίζουμε τις μεταβλητές ελαχιστοποίησης στο βέλτιστο σημείο

$$\tilde{M} = \begin{bmatrix} 0.7394 & 0.2879 \\ 0.2879 & 1.8596 \end{bmatrix}, \quad \tilde{P} = \begin{bmatrix} 1.4391 & -0.2228 \\ -0.2228 & 0.5722 \end{bmatrix},$$

και

$$\tilde{S} = \text{diag} \begin{pmatrix} 0.6586 & 1.7735 \end{pmatrix}, \quad \tilde{T} = 2.0098e - 06, \quad \delta = 2.7145e - 09.$$

Από τη σχέση $P = \tilde{P}^{-1}$ βρίσκουμε

$$P = \begin{bmatrix} 0.7394 & 0.2879 \\ 0.2879 & 1.8596 \end{bmatrix},$$

που παρατηρούμε ότι είναι συμμετρικός και θετικά ορισμένος και έχει ιδιοτιμές $\lambda_{1,2} = 0.6698, 1.9293$. Ακόμα ο νόμος ελέγχου εγγυημένου κόστους υπολογίζεται

$$\begin{aligned} \tilde{u}^*(t) &= -\delta R^{-1} B^T P x(t) \\ &= 1.0e - 09 \begin{bmatrix} -0.0782 & -0.5048 \end{bmatrix} x(t). \end{aligned}$$

Το εγγυημένο κόστος υπολογίζεται

$$\tilde{J}^* = \text{tr } P = 2.5990,$$

ενώ από τη σχέση που εξαρτάται από τις αρχικές συνθήκες

$$\tilde{J}^*(x_0) = x_0^T P x_0 = \begin{bmatrix} 3 & -2 \end{bmatrix} P \begin{bmatrix} 3 & -2 \end{bmatrix}^T = 10.6383.$$

Αντίστοιχα το κόστος του αρχικού συστήματος υπολογίζεται

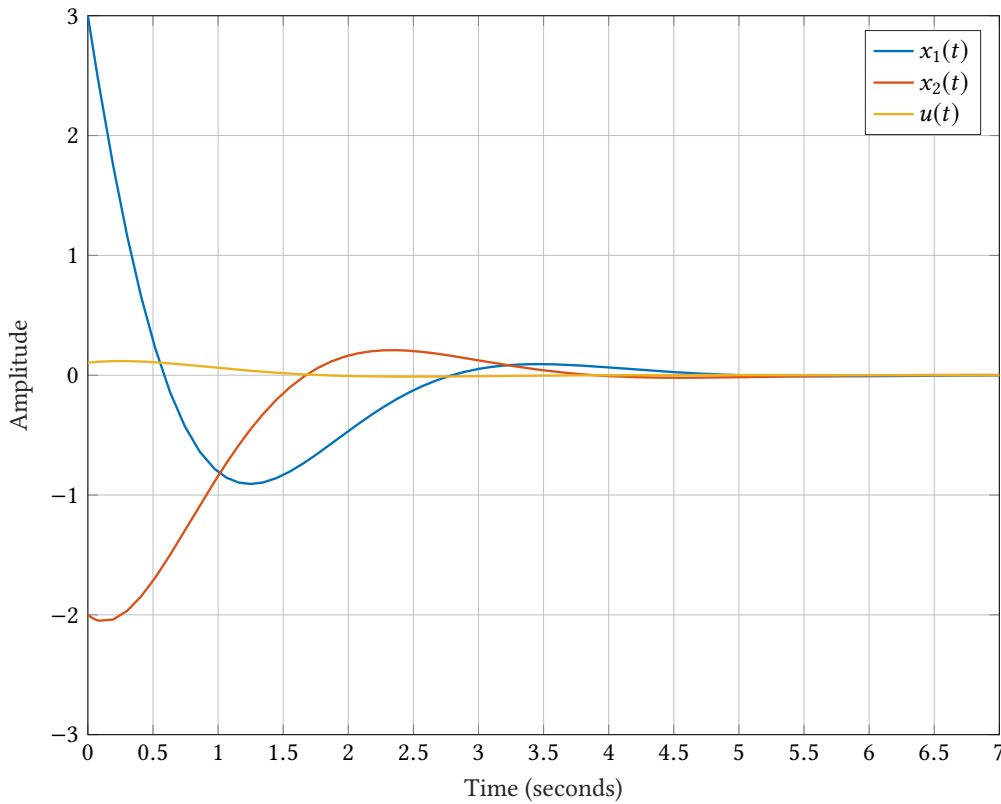
$$\tilde{J}_{not}^* = \text{tr } P_{not} = 1.0643,$$

ενώ από τη σχέση που εξαρτάται από τις αρχικές συνθήκες

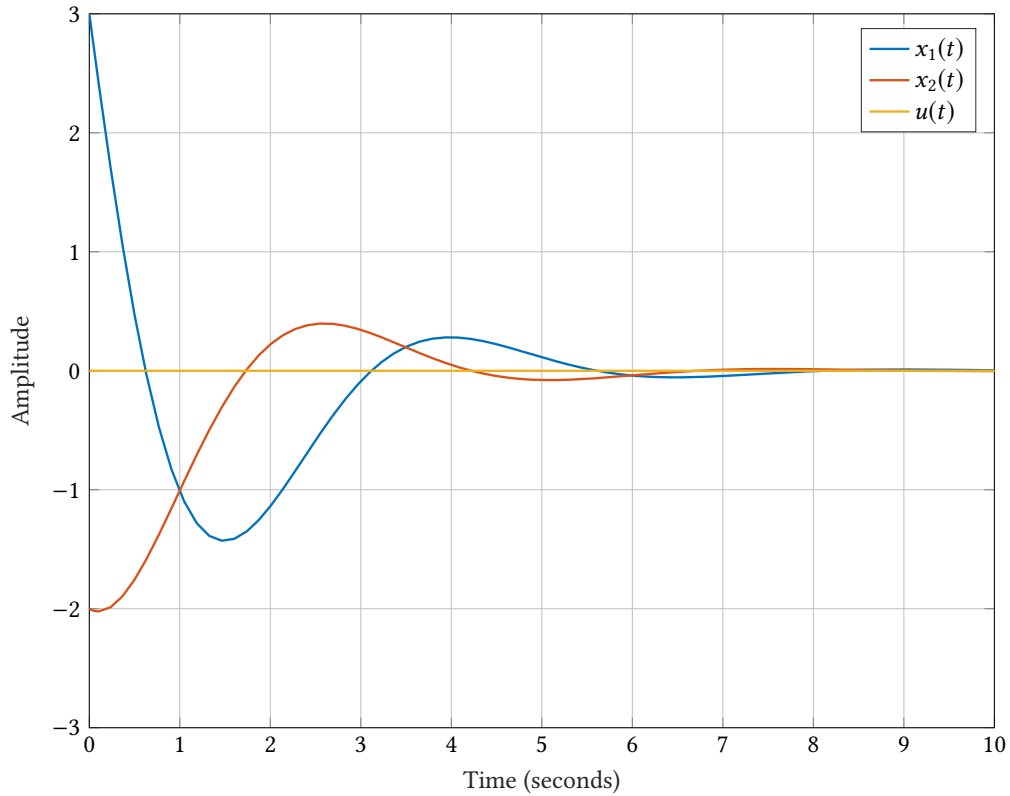
$$\tilde{J}_{not}^*(x_0) = x_0^T P_{not} x_0 = \begin{bmatrix} 3 & -2 \end{bmatrix} P_{not} \begin{bmatrix} 3 & -2 \end{bmatrix}^T = 5.2966.$$

Παρατηρούμε ότι ακριβώς το ίδιο πρόβλημα με το προηγούμενο μας δίνει βελτιωμένο, μικρότερο εγγυημένο κόστος πιο κοντά στο αρχικό κόστος που παρουσιάζει το σύστημα χωρίς αβεβαιότητες, που είναι φυσικά και το αναμενόμενο αποτέλεσμα. Ακόμα παρατηρούμε ότι ο νόμος ελέγχου εγγυημένου κόστους είναι ουσιαστικά μηδενικός. Αυτό είναι λογικό διότι ο συντελεστής R του $u(t)$ είναι υψηλός και εφόσον ο πίνακας A είναι Hurwitz, η μηδενική δράση οδηγεί σε ελαχιστοποίηση του κόστους.

Στα σχήματα 4.10, 4.11 και 4.12 παρουσιάζονται οι καταστάσεις του συστήματος για τις διάφορες τιμές των αβεβαιοτήτων καθώς και ο νόμος ελέγχου εγγυημένου κόστους.



Σχήμα 4.10: Αρχικό σύστημα χωρίς αβεβαιότητες



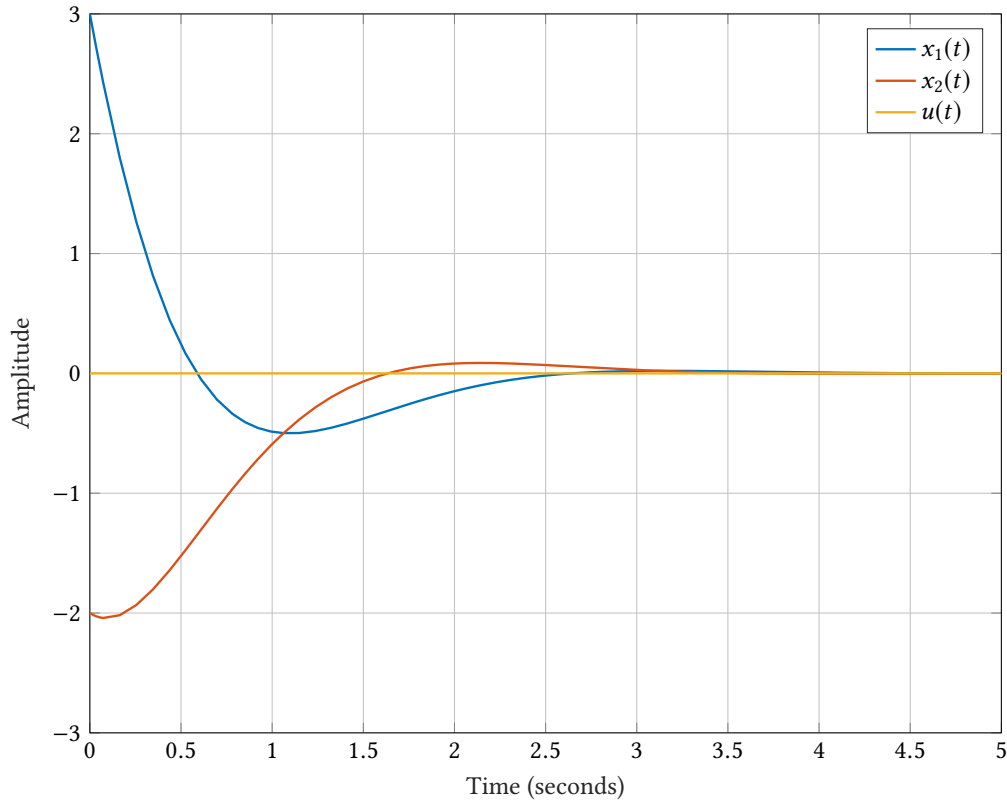
Σχήμα 4.11: Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, p_1 = 1$

Ο κώδικας για το παράδειγμα της μεθόδου ελαχιστοποίησης εγγυημένου κόστους με τις γραμμικές ανισότητες πινάκων, για την περίπτωση με αβεβαιότητες στον πίνακα των καταστάσεων και πίνακα ελέγχου παρατίθεται παρακάτω.

```

1 function [A, B, R, P, Pnot, Gc, d] = gcc_lmi();
2     A = [-1, 2; -1, -1];
3     B = [0; 1];
4     Q = eye(length(A));
5     Q_0 = inv(Q);
6     R = 10;
7
8     d_t1 = [0.9; 0.3];
9     e_t1 = [0.6; 0.2];
10    d_t2 = [0.1; 0.3];
11    e_t2 = [0.4; 0.9];
12    f_t1 = [0.3; 0.2];
13    g_t1 = 0.6;
14
15    n = length(A);
16    k = 2;
17    l = 1;
18
19    D = [d_t1, d_t2];
20    E = [e_t1, e_t2]';

```



Σχήμα 4.12: Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, p_1 = -1$

```

21 F = f_t1;
22 G = g_t1;
23
24 setlmis([]);
25 M_t = lmivar(1, [n 1]);
26 P_t = lmivar(1, [n 1]);
27 % k = 2
28 S_t = lmivar(1, [1 0; 1 0]);
29 % l = 1
30 T_t = lmivar(1, [1 0]);
31 d = lmivar(1, [1 0]);
32
33 % 1st lmi, -1 means LMI < 0
34 % (1,1) block, 's' means -A*P_t - P_t*A^T
35 lmiterm([-1 1 1 P_t], -A, 1, 's');
36 % (1,1) block, d*B*R^-1*B^T
37 lmiterm([-1 1 1 d], 1, B*(R\B'));
38 % (1,1) block, D*S_t*D^T
39 lmiterm([-1 1 1 S_t], -D, D');
40 % (1,1) block, F*T_t*F^T
41 lmiterm([-1 1 1 T_t], -F, F');
42 % (2,2) block, S_t
43 lmiterm([-1 2 2 S_t], 1, 1);

```

```

44 % (3,3) block, T_t
45 lmiterm([-1 3 3 T_t], 1, 1);
46 % (4,4) block, Q_0^-1
47 lmiterm([-1 4 4 0], Q_0);
48 % (2,1) block, E*P_t
49 lmiterm([-1 2 1 P_t], E, 1);
50 % (3,1) block, d*G*R^-1*B^T
51 lmiterm([-1 3 1 d], 1, G*(R*B'));
52 % (4,1) block, P_t
53 lmiterm([-1 4 1 P_t], 1, 1);
54
55 % 2nd lmi, -2 means LMI < 0
56 % (1,1) block, M_t
57 lmiterm([-2 1 1 M_t], 1, 1);
58 % (2,1) block, I_n
59 lmiterm([-2 2 1 0], eye(n));
60 % (2,2) block, P_t
61 lmiterm([-2 2 2 P_t], 1, 1);
62
63 % complete the LMI framework setting
64 lmis = getlmis;
65 % minimize the trace of M_t
66 c = mat2dec(lmis, eye(n), zeros(n), zeros(k), zeros(1), 0);
67 % set relative accuracy 1e-5
68 options = [1e-5,0,0,0,0];
69 % solve min problem
70 [copt xopt] = mincx(lmis, c, options);
71 % solution matrix M_t
72 M_t = dec2mat(lmis, xopt, M_t);
73 % solution matrix P_t
74 P_t = dec2mat(lmis, xopt, P_t);
75 % solution matrix S_t
76 S_t = dec2mat(lmis, xopt, S_t);
77 % solution matrix T_t
78 T_t = dec2mat(lmis, xopt, T_t);
79 % solution matrix d
80 d = dec2mat(lmis, xopt, d);
81 P = inv(P_t);
82
83 DA = @(r1, r2) ...
84     r1*S_t(1,1)^(0.5)*d_t1*S_t(1,1)^(-0.5)*e_t1' ...
85     + r2*S_t(2,2)^(0.5)*d_t2*S_t(2,2)^(-0.5)*e_t2';
86 DB = @(p1) ...
87     p1*T_t(1,1)^(0.5)*f_t1*T_t(1,1)^(-0.5)*g_t1';
88
89 Gc = @(r1, r2, p1) ...
90     A + DA(r1, r2) - (B + DB(p1))*(R\B'*P));

```

```

91     Pnot = care(A, B, Q, R);
92     makeplot(A, B, R, P, Pnot, Gc, d);
93 end
94
95 function dxdt = state(~, x, Gc)
96     dxdt = Gc*x;
97 end
98
99 function makeplot(A, B, R, P, Pnot, Gc, d)
100     u = @(P) -R\ (B'*P);
101     close all;
102     figure(1)
103     [t, x] = ode45(@(t, x) ...
104         state(t, x, A - B*(R\ (B'*Pnot))), [0 7], [3; -2]);
105     plot(t, x, t, u(Pnot)*x', 'LineWidth', 1.0);
106     title('Initial System')
107     legend('x_1(t)', 'x_2(t)', 'u(t)');
108     grid on;
109
110     figure(2)
111     [t, x] = ode45(@(t, x) ...
112         state(t, x, Gc(1, 1, 1)), [0 10], [3; -2]);
113     plot(t, x, t, d*u(P)*x', 'LineWidth', 1.0);
114     title('Uncertain System r1 = 1, r2 = 1, p1 = 1')
115     legend('x_1(t)', 'x_2(t)', 'u(t)');
116     grid on;
117
118     figure(3)
119     [t, x] = ode45(@(t, x) ...
120         state(t, x, Gc(-1, -1, -1)), [0 5], [3; -2]);
121     plot(t, x, t, d*u(P)*x', 'LineWidth', 1.0);
122     title('Uncertain System r1 = -1, r2 = -1, p1 = -0.5')
123     legend('x_1(t)', 'x_2(t)', 'u(t)');
124     grid on;
125 end

```

Κεφάλαιο 5

Εύρωστη τοποθέτηση πόλων

Στο παρόν κεφάλαιο θα εξετάσουμε το πρόβλημα της εύρωστης τοποθέτησης πόλων. Η τοποθέτηση πόλων είναι μία τεχνική ελέγχου κατά την οποία επιδιώκουμε να ικανοποιήσουμε συγκεκριμένα επιθυμητά χαρακτηριστικά, όπως χρόνος ανύψωσης, χρόνος αποκατάστασης, ευστάθεια κτλ. Στην περίπτωση αβέβαιων συστημάτων, η ανάγκη ικανοποίησης των επιθυμητών χαρακτηριστικών είναι επιπλέον επιτακτική, καθώς κάποια διαταραχή μπορεί να οδηγήσει σε έντονη απόκλιση από την επιθυμητή κατάσταση. Η ανάπτυξη του κεφαλαίου βασίζεται στο βιβλίο [11].

Θα μελετήσουμε το τυπικό γραμμικό αβέβαιο σύστημα της μορφής

$$\dot{x}(t) = (A + \Delta A)x(t) + Bu(t), \quad x(t_0) = x_0, \quad (5.1)$$

για $t \geq t_0$, όπου $x(t) \in \mathbb{R}^n$ το διάνυσμα κατάστασης, $u(t) \in \mathbb{R}^m$ το διάνυσμα ελέγχου, A και B οι πίνακες κατάστασης και ελέγχου αντίστοιχα, κατάλληλων διαστάσεων. Η αβεβαιότητα περιγράφεται από τη μορφή

$$\Delta A = \sum_{i=1}^k A_i r_i$$

όπου A_i είναι σταθερός πίνακας και r_i βαθμωτό μέγεθος, που μπορεί να είναι και χρονικά μεταβαλλόμενα στη γενική περίπτωση. Με την παραπάνω περιγραφή, η παράμετρος r_i ανήκει στο σύνολο

$$\mathcal{R} := \{r \in \mathbb{R}^k : |r_i| \leq \bar{r}, \bar{r} > 0, i = 1, \dots, k\}.$$

Χωρίς να μειωθεί η γενικότητα τα όρια μπορούν να κανονικοποιηθούν, δηλαδή $\bar{r} = 1$. Έτσι ονομάζουμε *αποδεκτή αβεβαιότητα*, τις αβεβαιότητες που ικανοποιούν τις γραμμικές μορφές, όπως ορίστηκαν παραπάνω, και βρίσκονται εντός των φραγμένων συνόλων.

Όπως είδαμε στο κεφάλαιο του εγγυημένου κόστους, ο πίνακας που περιγράφει τις αβεβαι-

ότητες επιτρέπει, το μη μοναδικό, διαχωρισμό μοναδιαίας τάξης

$$A_i = d_i e_i^T,$$

όπου $i = 1, \dots, k$ και $d_i, e_i \in \mathbb{R}^n$. Με βάση τον παραπάνω διαχωρισμό ορίζονται οι συμμετρικοί και θετικά ορισμένοι πίνακες

$$T = \sum_{i=1}^k d_i d_i^T, \quad U = \sum_{i=1}^k e_i e_i^T.$$

Θα θεωρήσουμε τις επιδόσεις του συστήματος ότι περιγράφονται από τετραγωνική συνάρτηση κόστους

$$J(x_0, \Delta A, t) = \int_{t_0}^{\infty} [x^T(t) Q x(t) + u^T(t) R u(t)] dt. \quad (5.2)$$

Σε πλήρη αντιστοιχία με το κεφάλαιο του εγγυημένου κόστους, για το σύστημα (5.1) με την τετραγωνική συνάρτηση κόστους (5.2) η συνάρτηση $\tilde{u} : \mathbb{R} \rightarrow \mathbb{R}^m$ ονομάζεται νόμος ελέγχου εγγυημένου κόστους, αν υπάρχει ένας θετικός αριθμός \tilde{J} , που λέγεται εγγυημένο κόστος, τέτοιος ώστε

$$J(x_0, \tilde{u}, \Delta A, t) \leq \tilde{J}$$

για όλες τις αποδεκτές αβεβαιότητες. Αποδεικνύεται ότι ο γραμμικός νόμος ελέγχου ανάδρασης κατάστασης

$$\tilde{u}(t) = Kx(t) = -R^{-1}B^T P x(t),$$

είναι ένας νόμος ελέγχου εγγυημένου κόστους και

$$\tilde{J}(x_0) = x_0^T P x_0,$$

είναι ένα εγγυημένο κόστος ή με κάποιες τεχνικές παραδοχές στην αντίστοιχη μορφή

$$\tilde{J} = \text{tr } P, \quad (5.3)$$

αν υπάρχει ένας θετικά ορισμένος συμμετρικός πίνακας P που ικανοποιεί τη γενικευμένη αλγεβρική εξίσωση Riccati, για συντομία GARE,

$$PA + A^T P - P(BR^{-1}B^T - T)P + Q + U = 0. \quad (5.4)$$

Ακόμα το σύστημα που προκύπτει είναι ασυμπτωτικά ευσταθές για κάθε αποδεκτή αβεβαιότητα, δηλαδή για κάθε $r_i \in \mathcal{R}$.

Όπως αναφέραμε, επιθυμούμε το σύστημα να τείνει εκθετικά στο μηδέν αλλά επιθυμούμε να έχει το σύστημα κάποια συγκεκριμένα χαρακτηριστικά. Αυτό μεταφράζεται, οι ιδιοτιμές του κλειστού συστήματος θέλουμε να βρίσκονται αριστερά από κάποια επιλεγμένη τιμή, έστω a , ή αντίστοιχα το κλειστό σύστημα να έχει βαθμό σχετικής ευστάθειας a .

Για να επιτύχουμε τον προαναφερθέν βαθμό σχετικής ευστάθειας, θεωρούμε τους μετασχηματισμούς στο πεδίο του χρόνου, $\hat{x}(t) := e^{at}x(t)$ και $\hat{u}(t) := e^{at}u(t)$ και έτσι το σύστημα μετασχηματίζεται

$$\dot{\hat{x}}(t) = (A + \Delta A + aI)\hat{x}(t) + B\hat{u}(t), \quad \hat{x}(t_0) = \hat{x}_0 = e^{at_0}x_0, \quad (5.5)$$

με αντικειμενική συνάρτηση όπως της σχέσης (5.2) αλλά για τις νέες συντεταγμένες \hat{x} και \hat{u} .

Με την περιγραφή αυτή το πρόβλημα μετατρέπεται σε ένα πρόβλημα εγγυημένου κόστους που μπορεί να λυθεί με τις μεθόδους που αναφέραμε μέχρι τώρα.

Κατά το σχεδιασμό συστημάτων χωρίς αβεβαιότητες, η επιλογή των μητρώων στάθμισης επιτρέπει τη βελτίωση των χαρακτηριστικών κλειστού βρόχου, όπως περιθώρια ευρωστίας. Αντίστοιχα, για την περίπτωση αβέβαιων συστημάτων η κατάλληλη επιλογή του μητρώου στάθμισης μπορεί να μας οδηγήσει στο βαθμό σχετικής ευστάθειας που επιθυμούμε.

Έστω ότι ο συμμετρικός και θετικά ορισμένος πίνακας T μπορεί να γραφτεί στη μορφή $T = B\hat{R}B^T$, που ισχύει όταν οι αβεβαιότητες ικανοποιούν τις συνθήκες προσαρμογής. Τότε η αλγεβρική εξίσωση Riccati για το μη αβέβαιο σύστημα γίνεται

$$PA + A^TP - PB\hat{R}^{-1}B^T - TP + \hat{Q} = 0, \quad (5.6)$$

όπου $\hat{R}^{-1} = R^{-1} - \hat{T}$ και $\hat{Q} = Q + U$. Ο νόμος ελέγχου είναι

$$u(t) = -\hat{R}^{-1}B^TPx(t)$$

και το κλειστό σύστημα γίνεται

$$\dot{x}(t) = (A - B\hat{R}^{-1}B^TP)x(t) = Fx(t). \quad (5.7)$$

Αντίστοιχα, το κλειστό αβέβαιο σύστημα περιγράφεται από

$$\dot{x}(t) = Fx(t).$$

Η αλγεβρική γενικευμένη εξίσωση Riccati της σχέσης (5.4) συνδέεται με τη φασματική δομή του Χαμιλτονιανού πίνακα

$$H = \begin{bmatrix} A & -BR^{-1}B^T + T \\ -Q - U & -A^T \end{bmatrix}.$$

Αν πάρουμε τη μορφή Jordan του H προκύπτει

$$\begin{bmatrix} A & -BR^{-1}B^T + T \\ -Q - U & -A^T \end{bmatrix} \begin{bmatrix} X_0 & W_0 \\ Y_0 & Z_0 \end{bmatrix} = \begin{bmatrix} X_0 & W_0 \\ Y_0 & Z_0 \end{bmatrix} \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_u \end{bmatrix}, \quad (5.8)$$

όπου Λ_s και Λ_u είναι διαγώνιοι πίνακες των ευσταθών (αρνητικών) και ασταθών (θετικών)

ιδιοτιμών του H αντίστοιχα.

Πρέπει να αναφέρουμε ότι ο μετασχηματισμός του H σε διαγώνια μορφή ισχύει διότι από τις ιδιότητες των Χαμιλτονιανών πινάκων γνωρίζουμε ότι ένας Χαμιλτονιανός πίνακας έχει διακριτές ιδιοτιμές. Πιο συγκεκριμένα, γνωρίζουμε ότι αν λ είναι ιδιοτιμή του πίνακα H τότε και $\eta \lambda$ θα είναι ιδιοτιμή αυτού. Τότε οι στήλες του πίνακα που δημιουργείται από τους πίνακες X_0, W_0, Y_0, Z_0 , είναι τα γενικευμένα ιδιοδιανύσματα του H . Επίσης, κατά την υλοποίηση του αλγορίθμου είναι προτιμότερο ο μετασχηματισμός να γίνει μέσω των γενικευμένων ιδιοδιανυσμάτων του πίνακα H και όχι μέσω της μορφής Jordan. Αυτό διότι ο αλγόριθμος που υπολογίζει τη μορφή Jordan στο MATLAB αλλά και σε διάφορα άλλα υπολογιστικά πακέτα, δεν είναι ιδιαίτερα γρήγορος. Ένα ακόμα σημείο που πρέπει να προσέξουμε κατά την υλοποίηση του αλγορίθμου είναι ότι ζητάμε οι διαγώνιοι πίνακες Λ_s και Λ_u να είναι τοποθετημένοι όπως παρουσιάζονται στη σχέση (5.8), δηλαδή πρώτα οι αρνητικές και μετά οι θετικές ιδιοτιμές. Εφόσον κατά τον υπολογισμό δεν έχουμε την επιθυμητή μορφή, είναι γνωστό ότι με εναλλαγή στις στήλες γενικευμένων ιδιοδιανυσμάτων μπορούμε να λάβουμε την ισοδύναμη επιθυμητή μορφή.

Αποδεικνύεται ότι αν το ζεύγος (A, B) είναι σταθεροποιησιμο, τότε $P = Y_0 X_0^{-1}$ και $FX_0 = X_0 \Lambda_s$. Ακόμα, αν το ζεύγος (A, B) είναι πλήρως ελέγξιμο, τότε ο συμμετρικός και αρνητικά ορισμένος πίνακας $P_u = Z_0 W_0^{-1}$ υπάρχει και είναι λύση της (5.4). Επίσης, αποδεικνύεται ότι αντικαθιστώντας τον πίνακα στάθμισης των καταστάσεων Q με $Q + \Delta Q$, όπου $\Delta Q = -2aP_u$, στη συνάρτηση κόστους, οι ιδιοτιμές του κλειστού ονομαστικού συστήματος μετατοπίζονται κατά $-2a$. Η αντίστοιχη γενικευμένη εξίσωση Riccati γίνεται

$$\bar{P}(A + aI) + (A + aI)^T \bar{P} - \bar{P}(BR^{-1}B^T - T)\bar{P} + Q + U - 2aP_u = 0. \quad (5.9)$$

Σύμφωνα με τα παραπάνω προκύπτει το παρακάτω θεώρημα.

Θεώρημα 7 (Θεώρημα 5.7 της Κοσμίδου [11]). Έστω Λ_s το φάσμα του κλειστού συστήματος (5.7) για κάποιους δεδομένους πίνακες βάρους Q και R . Ακόμη, έστω οι εισάγεται βαθμός σχετικής ευστάθειας a και ο πίνακας βάρους κατάστασης Q τροποποιείται κατά

$$\Delta Q = -2aP_u,$$

όπου P_u είναι η αρνητικά ορισμένη λύση της (5.4). Τότε το φάσμα $\Lambda(\tilde{F}_s)$ του ονομαστικού κλειστού συστήματος με βαθμό σχετικής ευστάθειας a και τροποποιημένο δείκτη επιδόσεων είναι

$$\Lambda(\tilde{F}_s) = \Lambda_s - 2aI.$$

Σύμφωνα με το παραπάνω θεώρημα εφαρμόζοντας τον νόμο ελέγχου

$$u(t) = -\hat{R}^{-1}B^T \bar{P}_s x(t),$$

όπου \bar{P}_s είναι η θετικά ορισμένη λύση της σχέσης (5.9), οι ιδιοτιμές του ονομαστικού κλειστού συστήματος μετατοπίζονται κατά $-2a$ και αντίστοιχα οι ιδιοτιμές του αβέβαιου συστήματος (5.1) βρίσκονται αριστερά του $-a$ του μιγαδικού επιπέδου, για όλες τις αποδεκτές αβεβαιότητες.

Ο παραπάνω νόμος ελέγχου προϋποθέτει την ύπαρξη λύσεων των εξισώσεων (5.4) και (5.9). Ακόμα οι λύσεις οι οποίες προκύπτουν δεν είναι μοναδικές ή μπορεί ο αντίστοιχος πίνακας Riccati να μην είναι θετικά ορισμένος. Για το λόγο αυτό το παρόν κεφάλαιο συνοψίζεται σε έναν αλγόριθμο τριών βημάτων, όπου επιλύουμε προβλήματα ελαχιστοποίησης LMI ούτως ώστε το εγγυημένο κόστος να είναι το ελάχιστο και ακόμα να ικανοποιείται ο βαθμός σχετικής ευστάθειας.

5.1 Αλγόριθμος εύρωστης τοποθέτησης πόλων

Θα παρουσιάσουμε τα βήματα του αλγορίθμου εφαρμόζοντας ένα παράδειγμα και σε κάθε βήμα θα παρουσιάζουμε και τον αντίστοιχο κώδικα MATLAB που το υλοποιεί.

Θα μελετήσουμε το σύστημα της μορφής

$$\dot{x}(t) = (A + \Delta A)x(t) + Bu(t), \quad x(0) = 0,$$

όπου

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Σημειώνεται ότι το ονομαστικό σύστημα είναι ασταθές με ιδιοτιμές

$$\lambda_{1,2,3} = -1.8019, -0.445, 1.247.$$

Ακόμη οι αβεβαιότητες περιγράφονται από τους πίνακες

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Για συνέπεια με τους συμβολισμούς μέχρι τώρα έχουμε ότι η διάσταση του προβλήματος, δηλαδή η διάσταση του πίνακα A είναι $n = 3$ και το πλήθος των αβεβαιοτήτων $k = 2$. Ένας πιθανός διαχωρισμός των αβεβαιοτήτων είναι

$$A_1 = d_1 e_1^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad A_2 = d_2 e_2^T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

Ακόμη ορίζουμε τους πίνακες

$$D = \begin{bmatrix} d_1 & \dots & d_k \end{bmatrix}, \quad E = \begin{bmatrix} e_1 & \dots & e_k \end{bmatrix}^T,$$

ή αντικαθιστώντας

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Ακόμη επιλέγουμε τα μητρώα στάθμισης $Q = I_3$ και $R = 1$. Τέλος, επιλέγουμε το βαθμό σχετικής ευστάθειας $a = 1$. Τα δεδομένα του προβλήματος υλοποιούνται στη συνάρτηση `pp_data` στο MATLAB που παρατίθεται παρακάτω.

```

1 function [p] = pp_data()
2     A = [0 1 0; 0 0 1; 1 2 -1];
3     B = [0; 0; 1];
4     Q = eye(length(A));
5     Q_0 = inv(Q);
6     R = 1;
7
8     d_t1 = [0; 0; 1];
9     e_t1 = [0; 1; 0];
10    d_t2 = [0; 0; 1];
11    e_t2 = [0; 0; 1];
12
13    n = length(A);
14    k = 2; % # of uncertainties
15
16    D = [d_t1, d_t2];
17    E = [e_t1, e_t2]';
18
19    % relative stability degree
20    a = 1;
21
22    p = struct('A', A, 'B', B, 'Q', Q, ...
23              'Q_0', Q_0, 'R', R, 'n', n, 'k', k, ...
24              'D', D, 'E', E, 'a', a, 'sort_array', @sort_array);
25 end
26
27 function [V] = sort_array(V_, D, n)
28     % this function sorts the array
29     % of the generalized eigenvectors
30     % according to the form required by
31     % the step 2 of the algorithm
32     if n < 1
33         fprintf('Error n must >= 1\n');
34         return

```

```

35     end
36     [~, idx] = sort(diag(D));
37     V = V_(:, idx(1));
38     for i = 2 : 2*n
39         V = horzcat(V, V_(:, idx(i)));
40     end
41 end

```

5.1.1 Βήμα 1

Το πρώτο βήμα του αλγορίθμου περιλαμβάνει την επίλυση ενός προβλήματος ελαχιστοποίησης γραμμικών ανισοτήτων πινάκων

$$\begin{aligned}
 & \underset{(M, W, S)}{\text{minimize}} && \text{tr } M \\
 & \text{subject to} && \begin{bmatrix} M & I \\ I & P \end{bmatrix} > 0 \\
 & && \begin{bmatrix} -W(A + aI)^T - (A + aI)W + BR^{-1}B^T - DSD^T & WE^T & W \\ & EW & S & 0 \\ & W & 0 & Q_0^{-1} \end{bmatrix} > 0,
 \end{aligned}$$

όπου έχει ένα μη κενό σύνολο εφικτών λύσεων (M, W, S) με M και W συμμετρικοί και θετικά ορισμένοι πίνακες και $Q_0 < Q$. Ο πίνακας S επιλέγεται διαγώνιος

$$S = \text{diag}(\sigma_1 \quad \dots \quad \sigma_k)$$

και περιγράφει το διαχωρισμό των αβεβαιοτήτων μέσω της σχέσης

$$d_i e_i^T = \sigma_i d_i (1/\sigma_i) e_i^T, \quad i = 1, \dots, k. \quad (5.10)$$

Αν υπάρχει εφικτή λύση του προβλήματος τότε ο S περιγράφει το βέλτιστο διαχωρισμό υπό την έννοια ότι το ίχνος του πίνακα P της σχέσης (5.4) είναι ελάχιστο. Ακόμη ισχύει ότι $P = W^{-1}$.

Στο παράδειγμα η επίλυση του προβλήματος LMI δίνει

$$S = \begin{bmatrix} 0.1488 & 0 \\ 0 & 0.1578 \end{bmatrix}.$$

Παρακάτω παρατίθεται το πρώτο βήμα του αλγορίθμου όπως υλοποιήθηκε στο MATLAB.

```

1 function [S] = pp_step1(p)
2     A = p.A; B = p.B;
3     Q_0 = p.Q_0; R = p.R; n = p.n;
4     k = p.k; D = p.D; E = p.E;

```

```

5
6     setlmis([]);
7     M = lmivar(1, [n 1]);
8     W = lmivar(1, [n 1]);
9     % k = 2
10    S = lmivar(1, [1 0; 1 0]);
11
12    % 1st lmi, -1 means LMI < 0
13    % (1,1) block, 's' means -A*W - W*A^T
14    lmiterm([-1 1 1 W], -A, 1, 's');
15    % (1,1) block, B*R^-1*B^T
16    lmiterm([-1 1 1 0], B*(R\B'));
17    % (1,1) block, D*S*D^T
18    lmiterm([-1 1 1 S], -D, D');
19    % (2,2) block, S
20    lmiterm([-1 2 2 S], 1, 1);
21    % (3,3) block, Q_0^-1
22    lmiterm([-1 3 3 0], Q_0);
23    % (2,1) block, E*W
24    lmiterm([-1 2 1 W], E, 1);
25    % (3,1) block, W
26    lmiterm([-1 3 1 W], 1, 1);
27
28    % 2nd lmi, -2 means LMI < 0
29    % (1,1) block, M
30    lmiterm([-2 1 1 M], 1, 1);
31    % (2,1) block, I_n
32    lmiterm([-2 2 1 0], eye(n));
33    % (2,2) block, W
34    lmiterm([-2 2 2 W], 1, 1);
35
36    % complete the LMI framework setting
37    lmis = getlmis;
38    % minimize the trace of M
39    c = mat2dec(lmis, eye(n), zeros(n), zeros(k));
40    % set relative accuracy 1e-5
41    options = [1e-5,0,0,0,0];
42    % solve min problem
43    [copt, xopt] = mincx(lmis, c, options);
44    % solution matrix M
45    M = dec2mat(lmis, xopt, M);
46    % solution matrix W
47    W = dec2mat(lmis, xopt, W);
48    % solution matrix S
49    S = dec2mat(lmis, xopt, S);
50    P = inv(W);
51 end

```

5.1.2 Βήμα 2

Εφόσον υπάρχει λύση, και άρα ο βέλτιστος διαχωρισμός S είναι γνωστός από το πρώτο βήμα του αλγορίθμου, υπολογίζονται οι πίνακες T και U . Συγκεκριμένα, επιλέγουμε το βέλτιστο διαχωρισμό, δηλαδή από τη σχέση (5.10) προκύπτει

$$T = \sum_{i=1}^k \sigma_i^2 d_i d_i^T, \quad U = \sum_{i=1}^k (1/\sigma_i)^2 e_i e_i^T, \quad (5.11)$$

ή για το παράδειγμα υπολογίζουμε

$$T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.047 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 45.1914 & 0 \\ 0 & 0 & 40.138 \end{bmatrix}.$$

Δεδομένου του βαθμού σχετικής ευστάθειας $a = 1$ θέλουμε να υπολογίσουμε τη μεταβολή του πίνακα στάθμισης των καταστάσεων, όπως περιγράφεται στο προηγούμενο θεώρημα, από τη σχέση

$$\Delta Q = -2aP_u.$$

Για να υπολογίσουμε τον όρο ΔQ πρέπει πρώτα να βρούμε τον αρνητικά ορισμένο πίνακα P_u . Ο πίνακας αυτός συνδέεται με τον Χαμιλτονιανό πίνακα (5.8) και υπολογίζεται από

$$P_u = Z_0 W_0^{-1}, \quad (5.12)$$

όπως έχουμε αναφέρει προηγουμένως.

Συνεπώς υπολογίζουμε τη μορφή Jordan του Χαμιλτονιανού πίνακα (5.8) ή μέσω των γενικευμένων ιδιοδιανυσμάτων για μεγαλύτερη ταχύτητα υπολογισμού, και στη συνέχεια τους όρους P_u , ΔQ και τροποποιούμε τον πίνακα κατάστασης κατά ΔQ . Πραγματοποιώντας τις πράξεις βρίσκουμε

$$P_u = \begin{bmatrix} -14.5314 & 12.5177 & -0.4171 \\ 12.5177 & -67.0584 & 10.9952 \\ -0.4171 & 10.9952 & -9.2558 \end{bmatrix}, \quad \Delta Q = \begin{bmatrix} 29.0628 & -25.0355 & 0.8342 \\ -25.0355 & 134.1167 & -21.9904 \\ 0.8342 & -21.9904 & 18.5116 \end{bmatrix},$$

με τον πρώτο να είναι αρνητικά ορισμένος με ιδιοτιμές $\lambda_{1,2,3} = -71.7627, -12.4756, -6.6073$ και τον δεύτερο προφανώς θετικά ορισμένο.

Ο αντίστοιχος κώδικας MATLAB που υλοποιεί το δεύτερο βήμα του αλγορίθμου παρατίθεται παρακάτω.

```
1 function [P_u, error_flag] = pp_step2(p)
2     A = p.A; B = p.B; Q = p.Q;
3     R = p.R; n = p.n;
4     T = p.T; U = p.U;
```

```

5
6   H = [A, (-B*(R\B') + T); (-Q - U), -A'];
7   % [V, J] = jordan(H);
8   % it is far more efficient to make use
9   % of the generalized eigenvectors insted of
10  % finding the jordan form because of the
11  % slow implementation of MATLAB
12
13  [V_, D] = eig(H);
14  if 2*n ~= length(unique(diag(D)))
15      error_flag = true;
16  else
17      error_flag = false;
18  end
19  % need to swap the columns of the
20  % eigenvectors to get a diagonal
21  % matrix with the negative eigenvalues
22  % followed by the positive eigenvalues
23  V = p.sort_array(V_, D, n);
24
25  W_0 = V(1 : n, n+1: end);
26  Z_0 = V(n + 1 : end, n + 1 : end);
27  P_u = Z_0/W_0;
28  end

```

5.1.3 Βήμα 3

Στο τρίτο βήμα του αλγορίθμου επιλύουμε και πάλι ένα πρόβλημα ελαχιστοποίησης γραμμικών ανισοτήτων πινάκων LMI,

$$\begin{aligned}
 & \underset{(\bar{M}, \bar{W})}{\text{minimize}} && \text{tr } \bar{M} \\
 & \text{subject to} && \begin{bmatrix} \bar{M} & I \\ I & \bar{W} \end{bmatrix} > 0 \\
 & && \begin{bmatrix} -\bar{W}(A + aI)^T - (A + aI)\bar{W} + BR^{-1}B^T & \bar{W} \\ \bar{W} & \bar{Q}_0^{-1} \end{bmatrix} > 0,
 \end{aligned}$$

για κάποιο $\bar{Q}_0 < Q + U - 2aP_u$. Το U υπολογίζεται από τη σχέση (5.11) μετά το πρώτο βήμα του αλγορίθμου όπου έχει βρεθεί ο βέλτιστος διαχωρισμός και το P_u από τη σχέση (5.12) όπως βρέθηκε στο δεύτερο βήμα. Αν υπάρχει μία εφικτή λύση, τότε η θετικά ορισμένη λύση της σχέσης (5.9) είναι η

$$\bar{P} = \bar{W}^{-1},$$

και ελαχιστοποιεί το αντίστοιχο εγγυημένο κόστος.

Στο παράδειγμα η επίλυση του προβλήματος LMI δίνει

$$\bar{P} = P^* = \begin{bmatrix} 43.5181 & 43.3284 & 10.6657 \\ 43.3284 & 53.8495 & 16.5335 \\ 10.6657 & 16.5335 & 5.8922 \end{bmatrix},$$

όπου είναι θετικά ορισμένη λύση και η τιμή της αντικειμενικής συνάρτησης είναι

$$J^* = \text{tr } P^* = 103.2598.$$

Παρακάτω παρατίθεται το τρίτο βήμα του αλγορίθμου όπως υλοποιήθηκε στο MATLAB.

```

1 function [P] = pp_step3(p)
2     B = p.B; R = p.R;
3     n = p.n; T = p.T;
4     A_b = p.A + p.a*eye(n);
5     Q_b = p.Q_0 + p.U + p.DQ;
6
7     setlmis([]);
8     M_b = lmivar(1, [n 1]);
9     W_b = lmivar(1, [n 1]);
10
11     % 1st lmi, -1 means LMI < 0
12     % (1,1) block, 's' means -A_b*W_b - W_b*A_b^T
13     lmiterm([-1 1 1 W_b], -A_b, 1, 's');
14     % (1,1) block, B*R^-1*B^T
15     lmiterm([-1 1 1 0], B*(R\B'));
16     % (1,1) block, -T
17     lmiterm([-1 1 1 0], -T);
18     % (2,2) block, Q_b^-1
19     lmiterm([-1 2 2 0], Q_b);
20     % (2,1) block, W_b
21     lmiterm([-1 2 1 W_b], 1, 1);
22
23     % 2nd lmi, -2 means LMI < 0
24     % (1,1) block, M_b
25     lmiterm([-2 1 1 M_b], 1, 1);
26     % (2,1) block, I_n
27     lmiterm([-2 2 1 0], eye(n));
28     % (2,2) block, W_b
29     lmiterm([-2 2 2 W_b], 1, 1);
30
31     % complete the LMI framework setting
32     lmis = getlmis;
33     % minimize the trace of M
34     c = mat2dec(lmis, eye(n), zeros(n));

```



```

35 % set relative accuracy 1e-5
36 options = [1e-5,0,0,0,0];
37 % solve min problem
38 [copt, xopt] = mincx(lmis, c, options);
39 % solution matrix M_b
40 M_b = dec2mat(lmis, xopt, M_b);
41 % solution matrix W_b
42 W_b = dec2mat(lmis, xopt, W_b);
43 P = inv(W_b);
44 end

```

Ακόμη στην περίπτωση του ονομαστικού συστήματος, δηλαδή όταν δεν έχουμε αβεβαιότητες τότε κάποια βήματα του αλγορίθμου είναι περιττά και για αυτό η περίπτωση αυτή υλοποιήθηκε σε διαφορετική συνάρτηση, αλλά εφαρμόζοντας τον αλγόριθμο της εύρωστης τοποθέτησης πόλων. Έτσι όπως θα δούμε παρακάτω, οι ιδιοτιμές του κλειστού συστήματος στο ονομαστικό σύστημα θα είναι αριστερά του $-2a$ του μιγαδικού επιπέδου, όπου a ο βαθμός σχετικές ευστάθειας.

```

1 function [Gnom, Pnom, error_flag] = pp_nominal(p)
2     A = p.A; B = p.B;
3     Q = p.Q; R = p.R;
4     n = p.n; a = p.a;
5
6     H = [A, -B*(R\B'); -Q, -A'];
7     % [V, J] = jordan(H);
8     % it is far more efficient to make use
9     % of the generalized eigenvectors insted of
10    % finding the jordan form because of the
11    % slow implementation of MATLAB
12
13    [V_, D] = eig(H);
14    if 2*n ~= length(unique(diag(D)))
15        error_flag = true;
16    else
17        error_flag = false;
18    end
19    % need to swap the columns of the
20    % eigenvectors to get a diagonal
21    % matrix with the negative eigenvalues
22    % followed by the positive eigenvalues
23    V = p.sort_array(V_, D, n);
24
25    W_0 = V(1 : n, n+1: end);
26    Z_0 = V(n + 1 : end, n + 1 : end);
27    P_u = Z_0/W_0;
28    DQ = -2*a*P_u;
29

```

```

30     Pnom = care(A + a*eye(n), B, Q + DQ, R);
31     Gnom = A - B*(R\ (B'*Pnom));
32 end

```

Τέλος, παρατίθεται ο συνολικός αλγόριθμος στο MATLAB όπου ενώνει τα προηγούμενα βήματα για την εξαγωγή των αποτελεσμάτων και είναι η συνάρτηση που χρειάζεται κάποιος να τρέξει ώστε να εφαρμόσει τον αλγόριθμο του κεφαλαίου.

```

1  function pp_algorithm()
2      % get data
3      [p] = pp_data();
4
5      % check if uncertainties exists
6      if p.k == 0
7          fprintf('Solving the nominal system for a = %f\n', p.a);
8          fprintf('No uncertainties are present, k = 0\n');
9          [Gnom, Pnom, error_flag] = pp_nominal(p);
10         if error_flag
11             fprintf('Error H, no distinct eigenvalues\n');
12             return
13         end
14         return
15     end
16
17     % step 1 of algorithm
18     fprintf('Solving the LMI of step 1\n');
19     [S] = pp_step1(p);
20     % based on S set T, U for k = 2
21     T = p.D(:,1)*p.D(:,1)'\*S(1, 1)^2 ...
22         + p.D(:,2)*p.D(:,2)'\*S(2, 2)^2;
23     U = p.E(1,:)'\*p.E(1, :)*(1/S(1, 1)^2) ...
24         + p.E(2,:)'\*p.E(2, :)*(1/S(2, 2)^2)';
25     p.('T') = T;
26     p.('U') = U;
27
28     % step 2 of algorithm
29     [P_u, error_flag] = pp_step2(p);
30     if error_flag
31         fprintf('Error H, no distinct eigenvalues\n');
32         return
33     end
34     if any(eig(P_u) > eps)
35         fprintf('Error, P_u is not negative definite\n');
36         return
37     end
38     DQ = -2*p.a*P_u;
39     if all(abs(imag(DQ)) > eps)

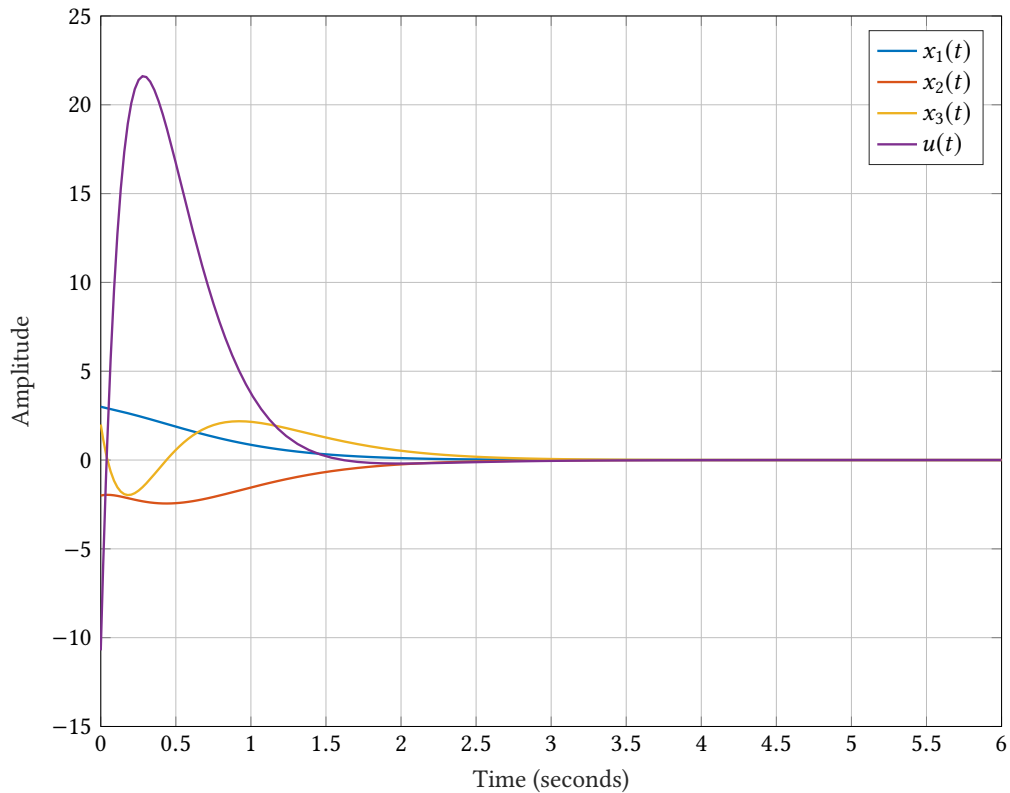
```

```

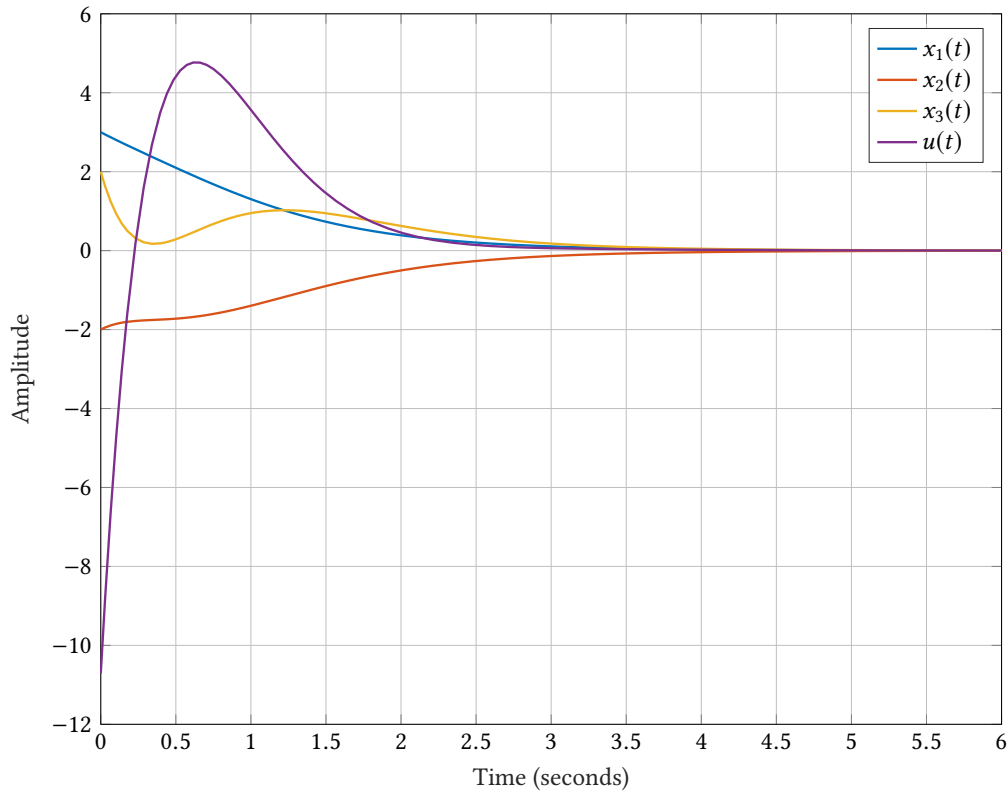
40     fprintf('Error DQ is complex, LMI does');
41     fprintf('not accept complex numbers\n');
42     return
43 else
44     % fixing numerical zeros
45     DQ = real(DQ);
46     p.('DQ') = DQ;
47 end
48
49 % step 3 of algorithm
50 fprintf('Solving the LMI of step 3\n');
51 [P_f] = pp_step3(p);
52 if all(eig(P_f) > eps)
53     fprintf('Optimal solution found\n');
54     p.('P_f') = P_f;
55     % plot data, use 'TEX' for latex plots
56     pp_makeplot(p, '');
57 end
58 end

```

Στα σχήματα 5.1, 5.2 και 5.3 παρουσιάζονται οι καταστάσεις για το ονομαστικό σύστημα και για διάφορες τιμές των αβεβαιοτήτων καθώς και ο νόμος ελέγχου εγγυημένου κόστους, για βαθμό σχετικής ευστάθειας $a = 1$, όπως προκύπτει για αρχικές συνθήκες $x = [3, -2, 2]^T$.



Σχήμα 5.1: Ονομαστικό σύστημα, $r_1 = 0, r_2 = 0, a = 1$



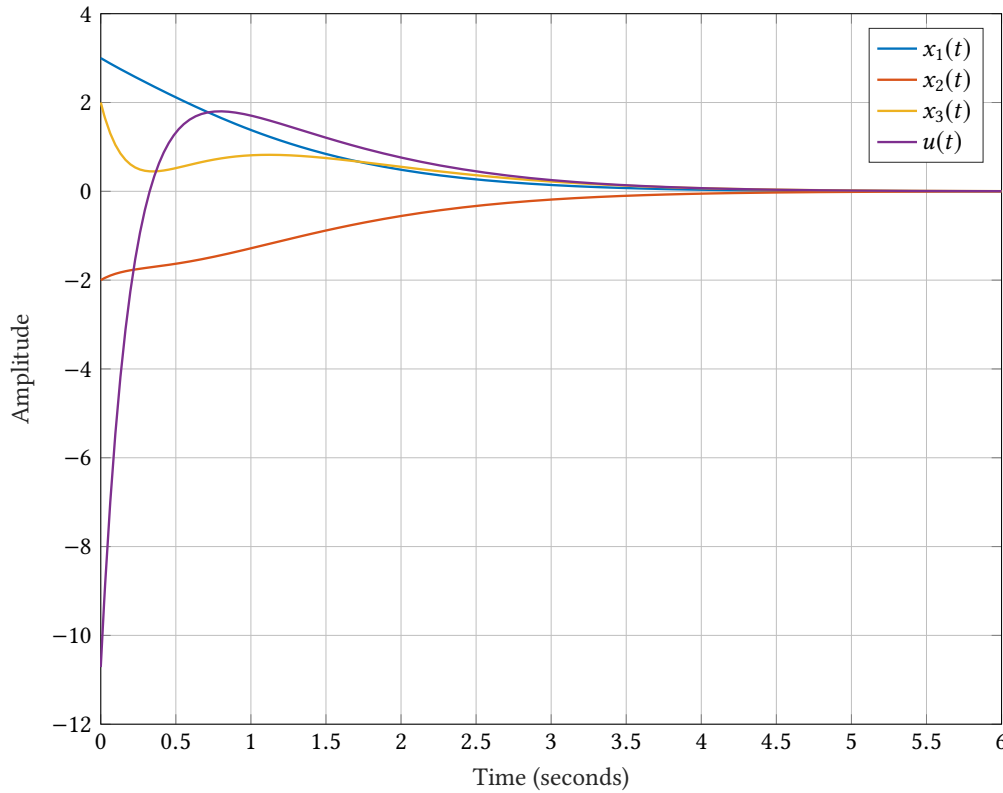
Σχήμα 5.2: Σύστημα με αβεβαιότητες, $r_1 = 1, r_2 = 1, a = 1$

Στον πίνακα 5.1 παρουσιάζονται οι ιδιοτιμές του κλειστού συστήματος και συγκεκριμένα το πραγματικό μέρος αυτών, για διάφορες τιμές των αβεβαιοτήτων. Ο βαθμός σχετικής ευστάθειας έχει επιλεχθεί $a = 1$, και παρατηρούμε ότι ο στόχος έχει επιτευχθεί διότι όλες οι ιδιοτιμές του κλειστού αβέβαιου συστήματος στο μιγαδικό επίπεδο βρίσκονται αριστερά του a . Συνεπώς, με τον αλγόριθμο του παρόντος κεφαλαίου, καταφέραμε όχι μόνο να σταθεροποιήσουμε το ονομαστικό σύστημα που είναι ασταθές, αλλά καταφέραμε να τοποθετήσουμε τους πόλους πέρα από το επιθυμητό όριο που επιλέξαμε. Ακόμα πρέπει να σημειωθεί ότι στην περίπτωση του ονομαστικού συστήματος, όταν δηλαδή δεν έχουμε αβεβαιότητες $r_1 = r_2 = 0$, τότε μέσω του αλγορίθμου οι ιδιοτιμές του κλειστού συστήματος μετατοπίζονται αριστερά του $-2a = -2$ όπως παρατηρείται και τον πίνακα 5.1.

	$r_1 = r_2 = 0$	$r_1 = r_2 = -1$	$r_1 = r_2 = 1$	$r_1 = 1, r_2 = -1$	$r_1 = 0, r_2 = -1$	$r_1 = 1, r_2 = 0$
λ_1	-2.6622	-5.3094	-1.2506	-5.866	-5.6079	-1.3281
λ_2	-3	-1.2914	-2.3208	-1.0131	-1.1421	-1.3281
λ_3	-4.1358	-1.2914	-2.3208	-1.0131	-1.1421	-4.236

Πίνακας 5.1: Πραγματικά μέρη ιδιοτιμών κλειστού συστήματος για διάφορες αβεβαιότητες και $a = 1$

Σαν έναν επιπλέον έλεγχο του αλγορίθμου τρέξαμε το πρόγραμμα για διαφορετικό βαθμό σχετικής ευστάθειας $a = 3$ και παραθέσαμε τον αντίστοιχο πίνακα των ιδιοτιμών του κλειστού αβέβαιου συστήματος. Έτσι προκύπτει ο πίνακας 5.2. Παρατηρούμε ότι ο αλγόριθμος



Σχήμα 5.3: Σύστημα με αβεβαιότητες, $r_1 = -1, r_2 = -1, a = 1$

τοποθέτησε τις ιδιοτιμές αριστερά του $-a = -3$ στην περίπτωση που υπάρχουν αβεβαιότητες και αριστερά του $-2a = -6$ για την περίπτωση του ονομαστικού συστήματος. Φυσικά πρέπει

	$r_1 = r_2 = 0$	$r_1 = r_2 = -1$	$r_1 = r_2 = 1$	$r_1 = 1, r_2 = -1$	$r_1 = 0, r_2 = -1$	$r_1 = 1, r_2 = 0$
λ_1	-6.6622	-3.6898	-3.3941	-3.5145	-3.5998	-4.1965
λ_2	-7	-3.6898	-6.6996	-3.5145	-3.5998	-4.1965
λ_3	-8.1358	-11.4138	-6.6996	-11.7642	-11.5938	-9.4002

Πίνακας 5.2: Πραγματικά μέρη ιδιοτιμών κλειστού συστήματος για διάφορες αβεβαιότητες και $a = 3$

να σημειωθεί ότι δε μπορούμε να επιλέξουμε αυθαίρετα το βαθμό σχετικής ευστάθειας a διότι αυτό έχει ως αποτέλεσμα την αύξηση του κόστους. Συγκεκριμένα, για την περίπτωση του παραδείγματος όπου επιλέξαμε $a = 3$ το βέλτιστο κόστος, υπό την έννοια του αλγορίθμου, έγινε $J^* = 5.916 \cdot 10^3$.

Ακόμα αν για παράδειγμα επιλέξουμε το βαθμό σχετικής ευστάθειας $a = 10$, προκύπτει ο πίνακας 5.3, τότε παρόλο που ικανοποιούνται τα κριτήρια που θέσαμε, δηλαδή οι ιδιοτιμές βρίσκονται αριστερά του -10 για το αβέβαιο σύστημα και αριστερά του -20 για το ονομαστικό, το βέλτιστο εγγυημένο κόστος φτάνει την τιμή $J^* = 3.0391 \cdot 10^6$. Το αν είναι αποδεκτή μία τέτοια τιμή έχει να κάνει με την υπό μελέτη εφαρμογή. Στην παρούσα φάση η πιθανόν ακραία αυτή τιμή χρησιμοποιήθηκε απλώς για τον έλεγχο της ορθότητας του αλγορίθμου.

	$r_1 = r_2 = 0$	$r_1 = r_2 = -1$	$r_1 = r_2 = 1$	$r_1 = 1, r_2 = -1$	$r_1 = 0, r_2 = -1$	$r_1 = 1, r_2 = 0$
λ_1	-20.6622	-14.497	-15.7631	-14.4131	-14.4548	-15.5262
λ_2	-21	-14.497	-20.2793	-14.4131	-14.4548	-15.5262
λ_3	-22.1358	-33.8695	-24.821	-34.0372	-33.9538	-30.8111

Πίνακας 5.3: Πραγματικά μέρη ιδιοτιμών κλειστού συστήματος για διάφορες αβεβαιότητες και $a = 10$

Τέλος, για πληρότητα στην εργασία παραθέτουμε τους αλγορίθμους που υλοποιήθηκαν στο MATLAB ώστε να εξαχθούν τα αντίστοιχα διαγράμματα και οι πίνακες.

```

1 function pp_makeplot(p, string)
2     K = -p.R\((p.B'*p.P_f);
3
4     if strcmp(string, 'TEX')
5         addpath('./pretty_fig');
6         TEX_pp(p, K, @Gc, @state);
7         return
8     end
9
10    close all;
11    figure(1)
12    [t, x] = ode45(@(t, x) ...
13        state(t, x, Gc(p, 0, 0)), [0 6], [3; -2; 2]);
14    plot(t, x, t, K*x', 'LineWidth', 1.0);
15    title('Nominal System r1 = 0, r2 = 0')
16    legend('x_1(t)', 'x_2(t)', 'x_3(t)', 'u(t)');
17    grid on;
18
19    figure(2)
20    [t, x] = ode45(@(t, x) ...
21        state(t, x, Gc(p, 1, 1)), [0 6], [3; -2; 2]);
22    plot(t, x, t, K*x', 'LineWidth', 1.0);
23    title('Uncertain System r1 = 1, r2 = 1')
24    legend('x_1(t)', 'x_2(t)', 'x_3(t)', 'u(t)');
25    grid on;
26
27    figure(3)
28    [t, x] = ode45(@(t, x) ...
29        state(t, x, Gc(p, -1, -1)), [0 6], [3; -2; 2]);
30    plot(t, x, t, K*x', 'LineWidth', 1.0);
31    title('Uncertain System r1 = -1, r2 = -1')
32    legend('x_1(t)', 'x_2(t)', 'x_3(t)', 'u(t)');
33    grid on;
34
35    maketable(p);

```

```

36 end
37
38 function closed = Gc(p, r1, r2)
39     if (r1 > -eps) && (r1 < eps) ...
40         && (r2 > -eps) && (r2 < eps)
41         [closed, ~, error_flag] = pp_nominal(p);
42         if error_flag
43             fprintf('Error H, no distinct eigenvalues\n');
44             return
45         end
46         return
47     end
48     DA = @(r1, r2) ...
49         r1*p.D(:,1)*p.E(1,:) ...
50         + r2*p.D(:,2)*p.E(2,:);
51     closed = p.A + DA(r1, r2) - p.B*(p.R\((p.B'*p.P_f)));
52 end
53
54 function dxdt = state(~, x, K)
55     dxdt = K*x;
56 end
57
58 function print_msg(lambda)
59     fprintf('lambda_1 = %s\n', num2str(lambda(1)));
60     fprintf('lambda_2 = %s\n', num2str(lambda(2)));
61     fprintf('lambda_3 = %s\n', num2str(lambda(3)));
62 end
63
64 function maketable(p)
65     fprintf('closed loop eigenvalues for r1 = 0, r2 = 0\n');
66     print_msg(eig(Gc(p, 0, 0)));
67
68     fprintf('\n closed loop eigenvalues for r1 = -1, r2 = -1\n');
69     print_msg(eig(Gc(p, -1, -1)));
70
71     fprintf('\n closed loop eigenvalues for r1 = 1, r2 = 1\n');
72     print_msg(eig(Gc(p, 1, 1)));
73
74     fprintf('\n closed loop eigenvalues for r1 = 1, r2 = -1\n');
75     print_msg(eig(Gc(p, 1, -1)));
76
77     fprintf('\n closed loop eigenvalues for r1 = 0, r2 = -1\n');
78     print_msg(eig(Gc(p, 0, -1)));
79
80     fprintf('\n closed loop eigenvalues for r1 = 1, r2 = 0\n');
81     print_msg(eig(Gc(p, 1, 0)));
82 end

```

Βιβλιογραφία

- [1] B. R. Barmish, M. Corless, and G. Leitmann. «A New Class of Stabilizing Controllers for Uncertain Dynamical Systems». In: *SIAM Journal on Control and Optimization* 21.2 (1983), pp. 246–255.
- [2] W.L. Brogan. *Modern Control Theory*. Prentice Hall, 1991.
- [3] C.T. Chen. *Linear System Theory and Design*. Linear System Theory and Design. Oxford University Press, 1999.
- [4] R.C. Dorf and R.H. Bishop. *Modern Control Systems*. Pearson Prentice Hall, 2011.
- [5] G.E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Texts in Applied Mathematics. Springer New York, 2005.
- [6] M.W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Pure and Applied Mathematics. Elsevier Science, 1974.
- [7] F. Lin. *Robust Control Design: An Optimal Control Approach*. Wiley, 2007.
- [8] A.M. Lyapunov. *General Problem of the Stability Of Motion*. Control Theory and Applications Series. Taylor & Francis, 1992.
- [9] K. Ogata. *Modern Control Engineering*. Pearson Education, 2011.
- [10] M. Vidyasagar. *Nonlinear Systems Analysis*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2002.
- [11] Κοσμίδου, Ο. *Εύρωστος Έλεγχος Δυναμικών Συστημάτων*. Γκιούρδας Β., 2009.