

Σχεδιασμός βέλτιστων δοσολογικών σχημάτων
φαρμάκων με τεχνολογία προβλεπτικού ελέγχου
σε φαρμακοκινητικά μοντέλα
Εφαρμογή στη χορήγηση ινσουλίνης



Νικόλαος Μάριος Κατρίτσης

Επιβλέπων καθηγητής: Χαράλαμπος Σαρίμβεης

Σχολή Χημικών Μηχανικών
Εθνικό Μετσόβιο Πολυτεχνείο

Διπλωματική εργασία

Αθήνα, Σεπτέμβριος 2018

Ευχαριστίες

Με την ολοκλήρωση αυτής της εργασίας θα ήθελα να ευχαριστήσω θερμά τον υπεύθυνο καθηγητή μου, Χαράλαμπο Σαρίμβη, για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, καθώς και για την εμπιστοσύνη που μου έδειξε και τη βοήθεια που μου προσέφερε καθόλη τη διάρκεια εκπόνησης της εργασίας.

Θα ήθελα ακόμα να ευχαριστήσω τον Σωτήρη Ντούσκα για την πολύτιμη συμβολή του στον σχεδιασμό του συστήματος αυτομάτου ελέγχου, χωρίς την οποία είναι πολύ πιθανό η εργασία αυτή να μην είχε την παρούσα της μορφή.

Το πιο θερμό ευχαριστώ ανήκει όμως στους γονείς και στους φίλους μου, *Γιάννη, Γιώργο, Δημήτρη, Ελίζ, Ιωάννα, Κυριακή, Λευτέρη, Μιχάλη, και Χριστίνα*, που βρίσκονται δίπλα μου και με στηρίζουν σε όλη την πορεία ως εδώ. *Δε θα μπορούσα σε καμία περίπτωση να εκφράσω με λόγια το πόσο ευγνώμων αισθάνομαι που είστε μέρος της ζωής μου.*

Abstract

Type I diabetes mellitus sets on after an autoimmune response against pancreatic β -cells takes place. Those are responsible for the production and release of insulin into the blood flow and them being no longer present in sufficient amount results in a dampened insulin response, ultimately resulting in improper blood glucose regulation. In present time, those suffering from the disease reside in manual insulin injections, while in recent years devices capable of infusing insulin subcutaneously, in a continuous fashion, have also become available, allowing for a more hassle-free approach.

The concept of artificial pancreas lies in the implementation of automatic control procedures so that proper blood glucose regulation is regained. Such an approach comes with great potential of improving the quality of life of patients with diabetes, setting them free of the burden of manual insulin control and quite possibly reducing both short-term and long-term complications of the disease.

The work presented here focuses on the development of optimal drug dosage control schemes which are then applied to the case of an in-silico patient with type I diabetes mellitus. Simulations are made using the physiologically-based pharmacokinetic model describing glucose-insulin metabolism which has been developed by Schaller et al. ^[1] using Bayer's MoBi software package. It is to be noted though that all algorithms utilised throughout this thesis have been developed in MATLAB.

The in-silico patient is simulated so that properly formulated experimental data is gathered. The process consists of subcutaneously infusing insulin while blood glucose levels are measured continuously. System identification is then carried out, leading to the estimation of autoregressive models with exogenous inputs. It is later proven that such models cannot provide an adequate description of system dynamics. The non-linear behaviour of the real system is finally captured by utilising a Takagi-Sugeno fuzzy model structure. This assumes the role of choosing among a set of locally linear models, the one that most probably performs the best. The process effectively transforms the previously static models to linear parameter-varying or pseudolinear ones.

The model predictive controller makes predictions over a finite horizon in the future. Optimal behaviour is achieved by utilising a terminal state cost weight, which effectively

extends the cost horizon to infinity. The evolution of disturbances affecting the system is taken into consideration by the utilisation of a Kalman filter observer. The state-space model is also augmented with a constant state disturbance model, leading to a controller formulation capable of rejecting unmeasured constant disturbances entering the system. Safety concerns are integrated into the controller design by imposing constraints. A hard constraint limits the quantity of insulin that can be infused per sampling interval. Additional soft constraints specify upper and lower bounds for blood glucose. Allowing for the violation of the latter is of key importance, ensuring that a feasible solution always exists.

The model predictive controller utilising the pseudolinear model performs well, proving capable of keeping postprandial blood glucose levels within normal range in most cases. The controller is also tested against meals of varying glycemic load, retaining its good behaviour. Under all circumstances zero tracking control is achieved, while also staying far off hypoglycemic conditions. The controller formulation presented here is novel in that no meal or any other information needs to be provided by the patient. There also exists great potential for future work, which is a matter discussed thoroughly later on.

Περίληψη

Ο σακχαρώδης διαβήτης τύπου I οφείλεται στην αυτοάνοση αντίδραση που καταστρέφει τα β-κύτταρα του παγκρέατος, που είναι υπεύθυνα για την έκκριση της ινσουλίνης, μίας εκ των ορμονών που ευθύνονται για τη ρύθμιση της γλυκόζης στο αίμα. Η αντιμετώπιση της νόσου σήμερα βασίζεται στη χειροκίνητη χορήγηση ινσουλίνης από τον ασθενή, με τη μορφή πολλαπλών ενέσεων στη διάρκεια της ημέρας ή με τη χρήση συσκευών για συνεχή έγχυση ινσουλίνης.

Η ιδέα του τεχνητού παγκρέατος έγκειται στον αυτόματο έλεγχο του επιπέδου της γλυκόζης στο αίμα. Τα δυνητικά πλεονεκτήματα είναι πολλαπλά και αφορούν τόσο στην αποδέσμευση του ασθενή από το βάρος της χειροκίνητης ρύθμισης της γλυκόζης, όσο και στον περιορισμό των βραχυπρόθεσμων και μακροπρόθεσμων επιπλοκών της νόσου, οδηγώντας σε γενικότερη βελτίωση της ποιότητας ζωής του.

Στην παρούσα εργασία μελετάται ο σχεδιασμός βέλτιστων δοσολογικών σχημάτων φαρμάκων, με εφαρμογή στη χορήγηση ινσουλίνης σε *in-silico* ασθενή με σακχαρώδη διαβήτη τύπου I. Η προσομοίωση του ασθενή βασίζεται στην περιγραφή του μεταβολισμού γλυκόζης-ινσουλίνης με τη χρήση του φαρμακοκινητικού μοντέλου με βάση τη φυσιολογία που έχει αναπτυχθεί από τους Schaller et al.^[1] στο υπολογιστικό πακέτο MoBi της Bayer. Η προσομοίωση του ασθενή και οι υπόλοιποι αλγόριθμοι της εργασίας αναπτύσσονται σε υπολογιστικό περιβάλλον MATLAB.

Πειραματικά δεδομένα συλλέγονται από τον *in-silico* ασθενή, εγχέοντας ινσουλίνη υποδόρια και μετρώντας τη μεταβολή της γλυκόζης στο αίμα. Στη συνέχεια επιχειρείται αναγνώριση της δυναμικής συμπεριφοράς του συστήματος με χρήση αυτοπαλινδρομικών μοντέλων με εξωγενή είσοδο. Όπως διαπιστώνεται, αυτά αδυνατούν να αποτυπώσουν τη συμπεριφορά του συστήματος επαρκώς. Οι μη γραμμικότητες λαμβάνονται υπόψη με την επέκταση των μοντέλων σε γραμμικά με χρονικά μεταβαλλόμενες παραμέτρους ή ψευδογραμμικά. Για τον σκοπό αυτό εξάγονται πολλά τοπικά γραμμικά μοντέλα και αναπτύσσεται σύστημα ασαφούς συλλογιστικής τύπου Takagi-Sugeno που αναλαμβάνει την επιλογή του κατάλληλου μοντέλου σε κάθε περίπτωση.

Ο προβλεπτικός ελεγκτής παίρνει αποφάσεις σε πεπερασμένο χρονικό ορίζοντα μπροστά στο μέλλον. Η βέλτιστη συμπεριφορά του ελεγκτή εξασφαλίζεται με την προσθήκη

κόστους τελικής κατάστασης, γεγονός που επεκτείνει τον ορίζοντα κόστους στο άπειρο. Για την παρακολούθηση των διαταραχών χρησιμοποιείται βέλτιστο φίλτρο Kalman. Το σύστημα επίσης επεκτείνεται με το μοντέλο σταθερής διαταραχής κατάστασης προκειμένου να εξασφαλιστεί έλεγχος μηδενικού σφάλματος. Οι περιορισμοί ασφαλείας λαμβάνονται υπόψη με την επιβολή φραγμών στο σύστημα. Στην ποσότητα της εγχυόμενης ινσουλίνης σε κάθε χρονική στιγμή επιβάλλεται σκληρός (απαραβίαστος) φραγμός. Τα άνω και κάτω φυσιολογικά όρια για τη γλυκόζη στο αίμα τίθενται ως μαλακοί φραγμοί, καθώς σε πολλές περιπτώσεις η παραβίασή τους είναι αναπόφευκτη για ύπαρξη λύσης.

Ο ελεγκτής προβλεπτικού μοντέλου που αναπτύσσεται με βάση το ψευδογραμμικό μοντέλο επιδεικνύει πολύ καλή συμπεριφορά, διατηρώντας τη γλυκόζη εντός φυσιολογικών ορίων στην πλειονότητα των περιπτώσεων. Η καλή συμπεριφορά του ελεγκτή επικυρώνεται σε γεύματα με διαφορετικό γλυκαιμικό φορτίο. Σε κάθε περίπτωση ο ελεγκτής καταφέρνει να ακολουθήσει την τιμή-στόχο για τη γλυκόζη με μηδενικό σφάλμα και διατηρώντας ασφαλή απόσταση από συνθήκες υπογλυκαιμίας. Η καινοτομία της υλοποίησης έγκειται στη λειτουργία του ελεγκτή χωρίς την ανάγκη για οποιαδήποτε είσοδο από την πλευρά του ασθενή. Ταυτόχρονα προσφέρει σημαντικά περιθώρια μελλοντικής εξέλιξης, τα οποία αναπτύσσονται λεπτομερώς.

Περιεχόμενα

Κατάλογος σχημάτων	xi
Κατάλογος πινάκων	xiii
Κατάλογος παραθέσεων	xv
1 Εισαγωγικά στοιχεία	1
2 Φαρμακοκινητικά μοντέλα βασισμένα στη φυσιολογία	3
2.1 Εισαγωγικές έννοιες	3
2.2 Το μοντέλο Γλυκόζης- Ινσουλίνης	6
2.2.1 Σακχαρώδης διαβήτης τύπου I	7
2.2.2 Μοντελοποίηση σε MoBi	8
3 Αναγνώριση δυναμικών συστημάτων	11
3.1 Εισαγωγικές έννοιες	11
3.2 Πολυωνυμικά μοντέλα	13
3.2.1 Το αυτοπαλινδρομικό μοντέλο με εξωγενή είσοδο	13
3.2.2 Μία ευρύτερη προσέγγιση	15
3.2.3 Η μέθοδος των ελαχίστων τετραγώνων	18
3.2.4 Κριτήρια επικύρωσης	19
3.3 Μετατροπή σε μοντέλο χώρου καταστάσεων	21
3.4 Χρήση για προβλεπτικό σκοπό	23
4 Ρύθμιση προβλεπτικού μοντέλου	27
4.1 Εισαγωγικές έννοιες	27
4.2 Διαμόρφωση του προβλήματος	30
4.2.1 Έλεγχος μηδενικού σφάλματος	30
4.2.2 Επέκταση σε άπειρο ορίζοντα	32

4.2.3	Επιβολή φραγμών στο σύστημα	34
4.3	Μετατροπή σε πρόβλημα τετραγωνικού προγραμματισμού	36
5	Συστήματα ασαφούς λογικής	39
5.1	Εισαγωγικές έννοιες	39
5.2	Μοντέλο Takagi- Sugeno	41
5.3	Μοντελοποίηση μη γραμμικών συστημάτων	43
5.3.1	Ψευδογραμμικό ARX μοντέλο	43
5.3.2	Επέκταση της μεθόδου ελαχίστων τετραγώνων	44
5.3.3	Αλγόριθμοι ασαφούς συσταδοποίησης	45
6	Εφαρμογή τεχνητού παγκρέατος σε in-silico ασθενή διαβήτη τύπου I	49
6.1	Σχεδιασμός MPC ελεγκτή με γραμμικό μοντέλο	49
6.1.1	Κατασκευή συνόλων πειραματικών δεδομένων	50
6.1.2	Προσδιορισμός ARX μοντέλου	51
6.1.3	Υλοποίηση του MPC ελεγκτή	52
6.2	Επέκταση ελεγκτή με ψευδογραμμικό μοντέλο	56
6.2.1	Σχεδιασμός μοντέλου Takagi-Sugeno	56
6.2.2	Υλοποίηση βελτιωμένου MPC ελεγκτή	59
7	Συζήτηση και μελλοντική δουλειά	61
	Βιβλιογραφία	65
	Παράρτημα Α' Κώδικες MATLAB	69
	Παράρτημα Β' Παράμετροι MoBi και ARX μοντέλων	85

Κατάλογος σχημάτων

2.1	Όργανα και ιστοί υπό μορφή διαμερισμάτων στο PK-Sim· διαγραμματική αναπαράσταση του ολικού ισοζυγίου μάζας που επιλύεται	4
3.1	Διαγραμματική απεικόνιση δυναμικού συστήματος με έξοδο y , είσοδο u , μετρήσιμη διαταραχή w , και μη μετρήσιμη διαταραχή v	12
3.2	Η πορεία της αναγνώρισης δυναμικών συστημάτων	13
3.3	Διαγραμματική απεικόνιση ARX μοντέλου	14
3.4	Δύο χαρακτηριστικές περιπτώσεις πολυωνυμικών μοντέλων	17
4.1	Διαγραμματική απεικόνιση ελέγχου κλειστού βρόγχου	28
4.2	Απεικόνιση του ολισθαίνοντα ορίζοντα στη ρύθμιση προβλεπτικού μοντέλου	29
4.3	Λογικό διάγραμμα ελέγχου ανατροφοδότησης μεταβλητών κατάστασης .	33
5.1	Τραπεζοειδής συνάρτηση συμμετοχής	40
6.1	Πειραματικά δεδομένα εισόδου-εξόδου για τον in-silico ασθενή	51
6.2	Εφαρμογή του ARX μοντέλου για πρόβλεψη 12 βημάτων	52
6.3	Ρυθμιστική δράση MPC ελεγκτή με γραμμικό ARX μοντέλο	55
6.4	Συναρτήσεις συμμετοχής με είσοδο $\bar{y}(t)$ · η 'ss' αντιστοιχεί στη μόνιμη κατάσταση και οι 'sm', 'mm' και 'bm' περιγράφουν μικρό, μεσαίο και μεγάλο γεύμα αντίστοιχα	57
6.5	Παράδειγμα εφαρμογής της προτεινόμενης συνάρτησης εισόδου για το τμήμα απόφασης του συστήματος Takagi-Sugeno σε πειραματικά δεδομένα	58
6.6	Συναρτήσεις συμμετοχής με είσοδο $y(t)$ · η 'nm' αντιστοιχεί στην απουσία και η 'mp' στην παρουσία επίδρασης από γεύμα	59
6.7	Ρυθμιστική δράση MPC ελεγκτή με ψευδογραμμικό ARX μοντέλο	60
7.1	Διαγραμματική απεικόνιση της πορείας που ακολουθείται στην εργασία για την ανάπτυξη του προβλεπτικού συστήματος	62

7.2	Διαγραμματική απεικόνιση της προτεινόμενης αλλαγής στην πορεία ανάπτυξης του προβλεπτικού συστήματος	63
-----	--	----

Κατάλογος πινάκων

6.1	Παραμετροποίηση γευμάτων στο MoBi	54
B.1	Ατομικές παράμετροι ασθενή με διαβήτη τύπου Ι στο MoBi	85

Κατάλογος παραθέσεων

A'.1	GlucoseInsulinModel.m	69
A'.2	ModelPredictiveController.m	76
A'.3	FuzzyPatientModel.m	80
A'.4	simulateMPC_FUZZY.m	82

Κεφάλαιο 1

Εισαγωγικά στοιχεία

Υπολογίζεται ότι ο σακχαρώδης διαβήτης τύπου I^[2] επηρεάζει περίπου του 3% όσων πάσχουν από διαβήτη. Η ασθένεια οφείλεται στην αυτοάνοση αντίδραση που οδηγεί στην καταστροφή των β-κυττάρων του ήπατος, τα οποία είναι υπεύθυνα για την έκκριση ινσουλίνης. Υπό συνθήκες μειωμένης παραγωγής ινσουλίνης, ο οργανισμός αδυνατεί να ρυθμίσει σωστά το επίπεδο της γλυκόζης στο αίμα και τελικά επικρατούν συνθήκες υπεργλυκαιμίας. Η ασθένεια συνίσταται στην αδυναμία του οργανισμού να αξιοποιήσει τη γλυκόζη για να καλύψει τις ενεργειακές του ανάγκες και χωρίς θεραπεία είναι θανατηφόρα. Σήμερα αντιμετωπίζεται με τη χειροκίνητη χορήγηση ινσουλίνης από τον ασθενή, ο οποίος καλείται να παρακολουθεί την εξέλιξη της γλυκόζης και να προχωρά στις απαραίτητες διορθωτικές ενέσεις πολλές φορές στη διάρκεια της μέρας.

Η παρούσα εργασία ξεκινά από την παρατήρηση πως ο ασθενής επιτελεί τον ρόλο ενός στοιχειώδους ελεγκτή, αποφασίζοντας να εγχύσει ινσουλίνη βάσει της μέτρησης της γλυκόζης στο αίμα του. Γίνεται λοιπόν σαφές πως τα δυναμικά πλεονεκτήματα που μπορούν να προκύψουν από την εφαρμογή κάποιου κατάλληλα διαμορφωμένου συστήματος αυτομάτου ελέγχου είναι πολλά. Ένα τέτοιο σύστημα ονομάζεται τεχνητό πάγκρεας και έχει αποτελέσει αντικείμενο μελέτης για περισσότερα από 50 χρόνια^[3], χωρίς όμως να έχει προκύψει ακόμα κάποια λύση που να είναι ικανή να επιτύχει αξιόπιστο γλυκαιμικό έλεγχο σε καθημερινή βάση. Αυτό εν μέρει οφείλεται στην πολυπλοκότητα του συστήματος προς εξέταση.

Στη συνέχεια θα μελετηθεί η εφαρμογή μεθόδων αυτομάτου ελέγχου για τον σχηματισμό βέλτιστων δοσολογικών σχημάτων φαρμάκων. Στην πραγματικότητα το ζήτημα είναι ευρύ και ξεκινά από την περιγραφή της φαρμακοκινητικής και φαρμακοδυναμικής συμπεριφοράς του μεταβολισμού γλυκόζης-ινσουλίνης στο Κεφάλαιο 2. Στη συνέχεια, στο Κεφάλαιο 3, παρουσιάζεται μία μεθοδολογία αναγνώρισης του πολύπλοκου δυναμικού συστήματος της προηγούμενης ενότητας. Το αποτέλεσμα είναι ο προσδιορισμός

απλών γραμμικών μοντέλων που διευκολύνουν το σχεδιασμό του αυτόματου ελεγκτή προβλεπτικού μοντέλου στο Κεφάλαιο 4. Αμέσως μετά, στο Κεφάλαιο 5, τα γραμμικά μοντέλα του Κεφαλαίου 3 επεκτείνονται σε ψευδογραμμικά με βελτιωμένη προβλεπτική ικανότητα. Αυτό επιτυγχάνεται με τη χρήση ενός κατάλληλα διατυπωμένου συστήματος ασαφούς συλλογιστικής τύπου Takagi-Sugeno.

Τα παραπάνω συνθέτουν το πρώτο μέρος της εργασίας, στο οποίο παρουσιάζεται λεπτομερώς το θεωρητικό υπόβαθρο των μεθόδων που εφαρμόζονται. Η ανάπτυξη των αλγορίθμων λαμβάνει χώρα στο υπολογιστικό περιβάλλον MATLAB και οι κώδικες παρατίθενται στο Παράρτημα Α'. Στο Κεφάλαιο 6 παρατίθενται τα αποτελέσματα που αφορούν στον σχεδιασμό τεχνητού παγιρέατος και τέλος, στο Κεφάλαιο 7, διατυπώνονται τα συμπεράσματα και γίνονται προτάσεις για μελλοντική δουλειά.

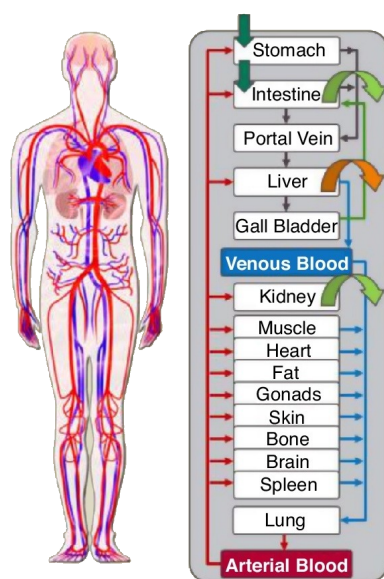
Κεφάλαιο 2

Φαρμακοκινητικά μοντέλα βασισμένα στη φυσιολογία

Το κεφάλαιο που ακολουθεί επιχειρεί να φέρει τον αναγνώστη σε επαφή με την έννοια των φαρμακοκινητικών μοντέλων που βασίζονται στη φυσιολογία. Μία εισαγωγή στο θέμα δίνεται στην Ενότητα 2.1 και αμέσως μετά, στην Ενότητα 2.2, περιγράφεται η ασθένεια του σακχαρώδους διαβήτη τύπου I και η μοντελοποίησή της με το μοντέλο Γλυκόζης- Ινσουλίνης στο υπολογιστικό πακέτο MoBi της Bayer. Το τελευταίο παρέχει το υπόβαθρο για την επικοινωνία με το υπολογιστικό περιβάλλον MATLAB, το οποίο αξιοποιείται παρακάτω στην εργασία.

2.1 Εισαγωγικές έννοιες

Τα υπολογιστικά φαρμακοκινητικά μοντέλα με βάση τη φυσιολογία^[4] ή μοντέλα PBPK (Physiologically Based Pharmacokinetic Model), αποτελούν υλοποιήσεις που συνδυάζουν πληροφορία σχετικά με τον τρόπο δράσης της φαρμακευτικής ουσίας με πρότερη γνώση της ανατομίας και της φυσιολογίας του οργανισμού. Η τελευταία λαμβάνεται υπόψη ανεξάρτητα με την πρώτη και παρέχει μία μηχανιστική περιγραφή των υποκείμενων μηχανισμών σε επίπεδο οργάνων ή και ακόμα πιο λεπτομερώς. Τα πλεονεκτήματα της προσέγγισης είναι πολλά. Καθώς τα επιμέρους όργανα και ιστοί λαμβάνονται υπόψη με ακρίβεια, υπό τη μορφή ξεχωριστών διαμερισμάτων, είναι δυνατό να γίνει ποσοτικός προσδιορισμός των επιθυμητών προφίλ συγκέντρωσης-χρόνου σε καθένα από αυτά. Επιπλέον, η μηχανιστική αναπαράσταση που ακολουθείται, επιτρέπει την ανάπτυξη ενός μοντέλου που στη συνέχεια μπορεί να προεκταθεί με σχετική ευκολία για πρόβλεψη σε διαφορετικές πληθυσμιακές ομάδες^[5]. Στη συνέχεια θα εστιάσουμε αναλυτικότερα στη δομή των μοντέλων αυτών. Μάλιστα ειδικότερη αναφορά θα γίνει στο Computational Systems Biology Software Suite^[5,6] που αναπτύσσεται από την Bayer. Το υπολογιστικό



Σχήμα 2.1 Όργανα και ιστοί υπό μορφή διαμερισμάτων στο PK-Sim· διαγραμματική αναπαράσταση του ολικού ισοζυγίου μάζας που επιλύεται^[4]

πακέτο είναι ανοιχτού κώδικα και περιλαμβάνει τα επιμέρους εργαλεία PK-Sim και MoBi. Το πρώτο επιτρέπει την παραμετροποίηση ενός PBPΚ μοντέλου σε επίπεδο ολόκληρου οργανισμού. Το μοντέλο δίνεται σε γενικευμένη μορφή και παραμετροποιείται με βάση την εκτενή βάση δεδομένων φυσιολογίας και ανατομίας που είναι διαθέσιμη με το λογισμικό. Το PK-Sim επιτρέπει μεν την εύκολη ανάπτυξη μοντέλων, όμως οι δυνατότητές του είναι περιορισμένες. Το MoBi συνεργάζεται με το PK-Sim και επιτρέπει την επεξεργασία και επέκταση μοντέλων του τελευταίου. Το λογισμικό επιτρέπει τη δομική επεξεργασία των μοντέλων, με την αφαίρεση υπαρχόντων ή την προσθήκη νέων διαμερισμάτων, τη μοντελοποίηση της αλληλεπίδρασης φαρμάκου-μεταβολίτη ή φαρμάκου-φαρμάκου, όπως και την προσομοίωση μεταβολικών δικτύων, λαμβάνοντας δηλαδή επιπλέον υπόψη φαρμακοδυναμικές¹ αλληλεπιδράσεις που λαμβάνουν χώρα.

Στα μοντέλα PBPΚ, ο οργανισμός περιγράφεται στη βάση των επιμέρους οργάνων, καθένα από τα οποία μοντελοποιείται ως ξεχωριστό διαμέρισμα. Στην περίπτωση του γενικευμένου μοντέλου στο PK-Sim, 17 όργανα και ιστοί λαμβάνονται υπόψη. Τα όργανα συνδέονται μεταξύ τους με ροή από τη δεξαμενή αρτηριακού αίματος προς αυτή του φλεβικού. Το ισοζύγιο μάζας ικανοποιείται θεωρώντας αντίθετη φορά αίματος διαμέσου των πνευμόνων^[5]. Τα παραπάνω δίνονται επίσης στο Σχήμα 2.1. Το κάθε όργανο δια-

¹ Αν η φαρμακοκινητική περιγράφει τον τρόπο που ο οργανισμός επιδρά επάνω στην φαρμακευτική ουσία, τότε η έννοια της φαρμακοδυναμικής έγκειται στο αντίστροφο, δηλαδή στη βιοχημική και φυσιολογική επίδραση που επέρχεται από την παρουσία του φαρμάκου στον οργανισμό.

ρείται περαιτέρω στα διαμερίσματα πλάσματος, ερυθρών αιμοσφαιρίων, μεσοκυττάριου και ενδοκυττάριου χώρου, οπότε επιτυγχάνεται ακόμα πιο λεπτομερής περιγραφή των διεργασιών που λαμβάνουν χώρα. Για παράδειγμα, καθίσταται δυνατή η προσομοίωση της ενζυμικής δραστηριότητας αποκλειστικά στον ενδοκυττάριο χώρο, που με τη σειρά του αλληλεπιδρά με τον εξωκυττάριο μέσω παθητικής (διάχυσης) ή πιθανά και ενεργούς (καταλυόμενης από ένζυμα) μεταφοράς^[4]. Αξίζει να σημειωθεί πως στην περίπτωση μεγάλων, θεραπευτικών πρωτεϊνών, όσα παρουσιάστηκαν παραπάνω επεκτείνονται προκειμένου να ληφθούν υπόψη μηχανισμοί που πλέον παίζουν σημαντικό ρόλο. Το μοντέλο των δύο πόρων θεωρεί ότι μεταξύ του πλάσματος και του μεσοκυττάριου χώρου βρίσκεται μία μεμβράνη που φέρει λίγους μεγάλους και πολλούς μικρούς πόρους. Με τον τρόπο αυτό τα μακρομόρια θεωρείται πως εξαγγειώνονται επιπλέον μέσω συναγωγής, εκτός από διάχυση. Επίσης προστίθεται ένα επιπλέον διαμέρισμα για τον ενδοσωμικό χώρο. Με τον τρόπο αυτό λαμβάνεται επίσης υπόψη η αποδόμηση της πρωτεΐνης που συμβαίνει στα λυσοσώματα. Από την άποψη αυτή, ο ρόλος του νέου διαμερίσματος δεν είναι άλλος από την ανακύκλωση στο πλάσμα, του μέρους εκείνου του φαρμάκου που είναι προσδεμένο στον υποδοχέα FcRn, ενώ το υπόλοιπο καταβολίζεται^[4,6].

Όπως πάντως αναφέρθηκε στην αρχή, η καινοτομία των μοντέλων έγκειται στην ιεραρχική διάρθρωσή τους. Το κάθε όργανο αναπαρίσταται από τις ανατομικές και φυσιολογικές παραμέτρους του, για παράδειγμα τον όγκο, τη σύσταση και την άρδευση αίματος σε αυτό. Αυτές υπολογίζονται με βάση την εσωτερική βιβλιοθήκη του PK-Sim και αποτελούν συνάρτηση της εθνότητας και της ανθρωπομετρίας του ασθενή, δηλαδή της ηλικίας, του φύλου και του ύψους του^[5,7]. Το μοντέλο κατανομής του φαρμάκου στα επιμέρους διαμερίσματα διατυπώνεται ανεξάρτητα από τα παραπάνω και συνίσταται στις διεργασίες παθητικής και ενεργούς μεταφοράς. Οι υπολογισμοί αυτοί βασίζονται στις φυσικοχημικές ιδιότητες της φαρμακευτικής ουσίας, όπως είναι το μοριακό βάρος, η λιποφιλία και η διαλυτότητά της. Στην περίπτωση των παραμέτρων που αφορούν στις διεργασίες παθητικής μεταφοράς, όπως για τους συντελεστές κατανομής μεταξύ ιστών και αίματος και τις διαπερατότητες των οργάνων, υπάρχει επίσης εξάρτηση από την φυσιολογία του οργανισμού. Είναι σύνηθες ο προσδιορισμός των παραμέτρων αυτών να γίνεται *in-silico*. Το ίδιο δε συμβαίνει όμως στην περίπτωση των ενεργών διεργασιών. Αυτές γενικά αφορούν τόσο στην κατανομή, όσο επίσης στον μεταβολισμό και στην απέκκριση της ουσίας και οι παράμετροι προς προσδιορισμό τυπικά αποτελούν κινητικές σταθερές. Ο προσδιορισμός τους συνήθως έγκειται στην προεκβολή *in-vitro* πειραματικών δεδομένων για χρήση σε *in-vivo* περιβάλλον ή στη χρήση πρότερης γνώσης για τις παραμέτρους που είναι δύσκολο να μετρηθούν πειραματικά^[4]. Τα παραπάνω βοηθούν στην αποσαφήνιση της έννοιας της ιεραρχικής δομής. Ο υπολογισμός των φυσιολογι-

κών παραμέτρων από τη βάση δεδομένων, η διάρθρωση της ανατομίας, οι παθητικές και ενεργές διεργασίες λαμβάνονται υπόψη ανεξάρτητα. Αυτός είναι άλλωστε ο λόγος που η μεταφορά του μοντέλου σε διαφορετικές πληθυσμιακές ομάδες καθίσταται εύκολη, απαιτώντας απλώς την κατάλληλη επαναμετροποίηση των φυσιολογικών παραμέτρων^[5].

Τα προηγούμενα συνοψίζουν τον τρόπο με τον οποίο το μοντέλο λαμβάνει υπόψη την κατανομή, μεταβολισμό, και απέκκριση του φαρμάκου. Για την απορρόφησή του, τα PBPK μοντέλα ορίζουν διάφορα πρότυπα χορήγησης. Το PK-Sim μπορεί να προσομοιώσει την λήψη φαρμάκου διά στόματος και ενδοφλέβια με ένεση ή συνεχή έγχυση με ορό. Δίνεται επίσης η δυνατότητα στον χρήστη να πραγματοποιήσει έγχυση φαρμάκου σε οποιοδήποτε όργανο-διαμέρισμα επιθυμεί. Στην περίπτωση της λήψης φαρμάκου διά στόματος πάντως, θα πρέπει επιπλέον να καθοριστεί η σύνθεση του φαρμάκου, η οποία καθορίζει το τοπικό προφίλ διαλυτοποίησης και απορρόφησής του από τον οργανισμό. Υπάρχει τέλος πρόβλεψη για χρονικά εξαρτώμενα συμβάντα. Αυτά επιτρέπουν τη χορήγηση γευμάτων, προκειμένου να μελετηθεί η επίδραση αυτών στην φαρμακοκινητική του φαρμάκου υπό εξέταση^[4,5].

2.2 Το μοντέλο Γλυκόζης- Ινσουλίνης

Το επίπεδο της γλυκόζης στο αίμα ελέγχεται μέσα από ένα πολύπλοκο φυσιολογικό σύστημα που περιλαμβάνει έναν μεγάλο αριθμό οργάνων. Στην ουσία αποτελεί αποτέλεσμα του ισοζυγίου μεταξύ της γλυκόζης που απελευθερώνεται στη ροή του αίματος από το πεπτικό σύστημα και από το ήπαρ και τους νεφρούς και εκείνης που καταναλώνεται για την κάλυψη των κυτταρικών ενεργειακών αναγκών, που αποθηκεύεται για μελλοντική χρήση ή απεκκρίνεται από τους νεφρούς^[8].

Το επίπεδο της γλυκόζης στο αίμα ρυθμίζεται από ορμόνες που παράγονται στο πάγκρεας και για έναν υγιή άνθρωπο κυμαίνεται μεταξύ 70–140 mg dl⁻¹^[9]. Σε κάθε περίπτωση, ινσουλίνη εκκρίνεται συνεχώς με χαμηλό σταθερό ρυθμό. Αυτό συνιστά τη βασική ινσουλιναίμια του οργανισμού. Μετά από ένα γεύμα, τα επίπεδα της γλυκόζης αυξάνονται. Υπό συνθήκες υπεργλυκαιμίας, ο πρωτεϊνικός γλυκομεταφορέας GLUT2 των β-κυττάρων του ήπατος επιτρέπει την υψηλότερη εισροή γλυκόζης προς το εσωτερικό τους, πράγμα που διεγείρει την έκκριση επιπλέον, γευματικής, ινσουλίνης στη ροή του αίματος^[10]. Αυτή επιδρά κυρίως στο ήπαρ, τον λιπώδη ιστό, τους σκελετικούς μύες και το κέντρο κορεσμού στον υποθάλαμο του εγκεφάλου. Τα μόρια της ινσουλίνης προσδένονται στους κατάλληλους υποδοχείς στην επιφάνεια των κυττάρων-στόχων, γεγονός που οδηγεί στη φωσφορυλίωση συγκεκριμένων πρωτεϊνών στην κυτταρική μεμβράνη. Στα λιποκύτταρα και τα μυϊκά κύτταρα αυτό συνεπάγεται τη μετακίνηση του γλυκομε-

ταφορέα GLUT4 από τις ενδοκυττάριας μεμβράνες προς την πλασματική. Η αυξημένη παρουσία γλυκομεταφορέων στην επιφάνεια των κυττάρων μεταφράζεται σε μεγαλύτερη εισροή γλυκόζης στο εσωτερικό τους. Στη συνέχεια ο υποδοχέας και η ινσουλίνη απορροφούνται με ενδοκυττάρωση, η πρώτη καταβολίζεται και ο υποδοχέας είναι και πάλι διαθέσιμος για πρόσδεση^[9,2].

Η παρουσία ινσουλίνης γενικά αυξάνει την ικανότητα των κυττάρων-στόχων να προσλάβουν γλυκόζη. Στην περίπτωση όμως που δεν υπάρχει ανάγκη για άμεση αξιοποίησή της προς παραγωγή ενέργειας, τότε λαμβάνει χώρα γλυκογένεση, δηλαδή μετατροπή της γλυκόζης σε γλυκογόνο και αποθήκευσή του για κάλυψη μελλοντικών κυτταρικών αναγκών. Αυτό συμβαίνει κυρίως στο ήπαρ και σε μικρότερο βαθμό στους νεφρούς, τους σκελετικούς μύες και άλλους ιστούς. Όταν οι αποθήκες γλυκογόνου κορεστούν, η περίσσεια γλυκόζη αποθηκεύεται με μορφή λιπαρών οξέων στα λιποκύτταρα^[9,2]. Στην περίπτωση υπογλυκαιμίας, η ορμόνη γλυκαγόνη εκκρίνεται στα α-κύτταρα του παγκρέατος. Αυτή επιδρά στα επίπεδα γλυκόζης του αίματος με τρόπο αντίθετο από ότι η ινσουλίνη, προκαλώντας τον καταβολισμό του γλυκογόνου και των λιπαρών οξέων και αυξάνοντας με τον τρόπο αυτό την παραγωγή γλυκόζης. Μάλιστα οι δύο ορμόνες δρουν ανταγωνιστικά, γεγονός που ενισχύει την επίδρασή τους^[8,9].

2.2.1 Σακχαρώδης διαβήτης τύπου I

Η παρουσία της ινσουλίνης είναι καθοριστικής σημασίας για τη φυσιολογική ρύθμιση της γλυκόζης στο αίμα. Ο σακχαρώδης διαβήτης τύπου I αναπτύσσεται μετά από την αυτοάνοση καταστροφή των β-κυττάρων του παγκρέατος, πράγμα που οδηγεί σε ανεπαρκή έκκριση ινσουλίνης από τον οργανισμό. Τα συμπτώματα εμφανίζονται μετά την καταστροφή του 90% των β-κυττάρων και τυπικά περιλαμβάνουν πολυουρία, έντονη αίσθηση δίψας και πείνας, και σε πολλές περιπτώσεις απώλεια βάρους.^[2,9]

Η αντιμετώπιση της νόσου σήμερα βασίζεται στη χειροκίνητη έγχυση ινσουλίνης από τον ασθενή, σε μία προσπάθεια μίμησης της φυσιολογικής έκκρισής της στον υγιή πληθυσμό. Συνήθως αυτό επιτυγχάνεται με μία ή δύο ενέσεις ινσουλίνης μακράς διάρκειας δράσης καθημερινά, οι οποίες αποσκοπούν στη διασφάλιση της βασικής ινσουλιναιμίας του οργανισμού και με επιπρόσθετες ενέσεις ινσουλίνης ταχείας δράσης πριν από κάθε γεύμα, λειτουργώντας διορθωτικά στη μειωμένη έκκριση γευματικής ινσουλίνης από το πάγκρεας. Μία εναλλακτική λύση αποτελούν οι αντλίες ινσουλίνης, οι οποίες επιτρέπουν την υποδόρια έγχυση ινσουλίνης ταχείας δράσης με συνεχή τρόπο. Η χρήση τους μπορεί μάλιστα να συνδυαστεί με συσκευές συνεχούς καταγραφής γλυκόζης, οπότε ο ασθενής αποκτά τη δυνατότητα εύκολου ελέγχου του επιπέδου της γλυκόζης στο αίμα του, που μπορεί δυνητικά να εξασφαλίσει την ασφαλέστερη ρύθμισή της^[8].

Η αδυναμία διασφάλισης συνθηκών γλυκαιμίας μπορεί να οδηγήσει σε σοβαρές βραχυπρόθεσμες επιπλοκές. Η έκκριση μεγάλης ποσότητας ινσουλίνης έχει ως αποτέλεσμα την έντονη πτώση της συγκέντρωσης της γλυκόζης στο αίμα. Όπως περιγράφηκε παραπάνω, αυτό οφείλεται στη διέγερση γλυκομεταφορέων όπως ο GLUT4, που είναι ευαίσθητοι στη συγκέντρωση της ινσουλίνης. Τα νευρωνικά κύτταρα από την άλλη δεν αποτελούν στόχο για την ινσουλίνη. Η λειτουργία του νευρικού συστήματος εξαρτάται κατά κύριο λόγο από την γλυκόζη του αίματος και για τον λόγο αυτό στην επιφάνεια των νευρωνικών κυττάρων συναντώνται οι γλυκομεταφορείς GLUT3. Αυτοί διασφαλίζουν πως οι ενεργειακές ανάγκες του νευρικού συστήματος καλύπτονται, διατηρώντας υψηλή και σταθερή εισροή γλυκόζης προς τα νευρωνικά κύτταρα. Σε συνθήκες ισχυρής υπογλυκαιμίας λοιπόν, το νευρικό σύστημα μοιραία υπολείπεται, οδηγώντας σε συμπτώματα όπως αποπροσανατολισμό, σύγχυση, και σπασμούς^[10,2]. Αντιθέτως, συνθήκες υπεργλυκαιμίας ελλείψει ινσουλίνης μπορεί να οδηγήσουν σε διαβητική κετοξέωση. Καθώς ο οργανισμός αδυνατεί να αξιοποιήσει την γλυκόζη του αίματος, λαμβάνει χώρα ταχεία λιπόλυση που οδηγεί στην παραγωγή κετονικών σωμάτων. Σε υψηλές συγκεντρώσεις, αυτό επιφέρει τελικά την ελάττωση του pH του αίματος. Πριν από την εύρεση θεραπείας, αυτή ήταν η κύρια αιτία θανάτου των διαβητικών^[8,2].

Οι μακροχρόνιες επιπλοκές της νόσου σχετίζονται με την παρατεταμένη κατάσταση υπεργλυκαιμίας. Αυτή επιφέρει υψηλή εισροή γλυκόζης στα κύτταρα, οδηγώντας στην υπερβολική κατανάλωσή της. Οι πιο συνήθεις χρόνιες επιπλοκές αφορούν στην αμφιβληστροειδοπάθεια, νευροπάθεια, νεφροπάθεια, και αθηροσκληρώση^[8,9,2].

2.2.2 Μοντελοποίηση σε MoBi

Στην παρούσα εργασία χρησιμοποιείται το μοντέλο που περιγράφει τον μεταβολισμό γλυκόζης- ινσουλίνης- γλυκαγόνης, το οποίο έχει αναπτυχθεί από τους Schaller et al.^[1] σε περιβάλλον MoBi. Πρόκειται συνεπώς για ένα PBPk/PD μοντέλο και ως τέτοιο χαίρει των πλεονεκτημάτων που αναφέρθηκαν προηγουμένως. Το μοντέλο είναι πάντως ιδιαίτερα πολύπλοκο και η προσπάθεια λεπτομερούς ανάπτυξής του εδώ θα ξέφευγε από το πνεύμα του κειμένου. Στη συνέχεια δίνεται λοιπόν μία σύντομη περιγραφή του. Για τον αναγνώστη που ενδιαφέρεται, εκτός από την πρωτότυπη δημοσίευση, μία αναλυτικότερη ανάπτυξη δίνεται στην διδακτορική διατριβή του Schaller^[6].

Το μοντέλο έχει να κάνει με μεγάλα πρωτεϊνικά μόρια και άρα χρησιμοποιείται ο κατάλληλος φορμαλισμός, όπως αυτός περιγράφηκε στην Ενότητα 2.1. Το PK-Sim περιλαμβάνει ένα λεπτομερές γαστρεντερικό μοντέλο για την περιγραφή της απορρόφησης φαρμάκων διά της στοματικής οδού. Αυτό επεκτείνεται με την προσθήκη των γλυκομεταφορέων GLUT2 και SGLT1 για τη μεταφορά της γλυκόζης από την εντερική κοιλότητα

στον μεσοκυττάριο χώρο του βλεννογόνου του λεπτού εντέρου. Το γεύμα λαμβάνεται υπόψη με τη μορφή χορήγησης γλυκόζης από το στόμα σε διαλυτοποιημένη μορφή. Γενικά το προφίλ εμφάνισης της γλυκόζης στο αίμα εξαρτάται από την ποσότητα των υδατανθράκων, ενώ διαφορετικοί τύποι υδατανθράκων, με άλλο γλυκαιμικό δείκτη ο καθένας, καθορίζουν το πόσο γρήγορα η γλυκόζη είναι διαθέσιμη για απορρόφηση, δηλαδή βρίσκεται σε διαλυτοποιημένη μορφή. Αυτό δε λαμβάνεται υπόψη από το μοντέλο στην παρούσα του μορφή. Σε ότι αφορά την επίδραση της σύστασης του γεύματος στη γαστρική κένωση και στον εντερικό ρυθμό διέλευσης για το λεπτό και το παχύ έντερο, αυτή λαμβάνεται υπόψη και υπολογίζεται βάσει του θερμιδικού περιεχομένου, του όγκου, και του στερεού κλάσματος του γεύματος. Ο βασικός μεταβολισμός της γλυκόζης λαμβάνεται κατά κύριο λόγο υπόψη στον εγκεφαλικό, λιπώδη και μυϊκό ιστό, στο ήπαρ και στα ερυθρά αιμοσφαίρια. Επίσης προστίθεται ο γλυκομεταφορέας GLUT2 στο ήπαρ, GLUT3 στον εγκέφαλο και GLUT4 στον λιπώδη και μυϊκό ιστό.

Για τη φαρμακοκινητική της ινσουλίνης προστίθεται ένα επιπλέον διαμέρισμα λίπους, μέσω του οποίου μοντελοποιείται η υποδόρια απορρόφηση ινσουλίνης ταχείας δράσης Lispro. Η ινσουλίνη Lispro που χορηγείται από την υποδόρια οδό παρουσιάζει βιοδιαθεσιμότητα 55 – 77%, πράγμα που μοντελοποιείται θεωρώντας μία διεργασία αποδόμησης. Στους ιστούς που είναι ευαίσθητοι στην ινσουλίνη, όπως ο μυϊκός, ο λιπώδης και το ήπαρ, η ενδοκυττάρωση της ινσουλίνης δεν πραγματοποιείται απλώς με παθητική μεταφορά, αλλά κατά κύριο λόγο με διαμεσολάβηση υποδοχέων ινσουλίνης. Στο μοντέλο λαμβάνεται επίσης υπόψη και η φαρμακοκινητική της γλυκαγόνης.

Οι φαρμακοδυναμικές διεργασίες περιγράφουν την αλληλεπίδραση μεταξύ γλυκόζης, ινσουλίνης, και γλυκαγόνης. Στο ήπαρ, η επίδραση της ινσουλίνης, της γλυκαγόνης και της γλυκόζης στην έκκριση και την αποθήκευση της γλυκόζης περιγράφονται με χρήση εμπειρικών εξισώσεων που αποτυπώνουν άμεσα την επίδραση της εισόδου στο παρατηρούμενο αποτέλεσμα. Η παραγωγή, ο μεταβολισμός, και η απέκκριση της ινσουλίνης δίνονται συναρτήσει των φωσφορυλιωμένων υποδοχέων ινσουλίνης στους αντίστοιχους ευαίσθητους ιστούς. Τέλος προστίθεται ένα μοντέλο για την προσομοίωση του φαινομένου ινιρετίνης, δηλαδή της έμμεσης έκκρισης ινσουλίνης πριν την εμφάνιση γλυκόζης στο αίμα, μετά από τη χορήγηση υδατανθράκων διά στόματος.

Το μοντέλο που αναπτύσσεται συνοψίζεται σε περίπου 100 εξισώσεις. Αυτές προστίθενται στις ήδη υπάρχουσες, οι οποίες περιγράφουν τις διεργασίες απορρόφησης, κατανομής, μεταβολισμού και απέκκρισης. Το γενικό μοντέλο του PK-Sim αποτελείται από περισσότερες από 100 διαφορικές εξισώσεις, πράγμα που καταδεικνύει την πολυπλοκότητα του τελικού μοντέλου.

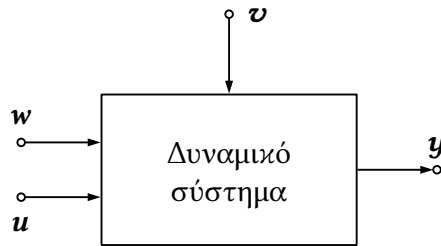
Κεφάλαιο 3

Αναγνώριση δυναμικών συστημάτων

Το παρόν κεφάλαιο επιχειρεί να εγκλιματίσει τον αναγνώστη στην έννοια της αναγνώρισης δυναμικών συστημάτων. Μετά την εισαγωγική συζήτηση στην Ενότητα 3.1, η δομή, ο τρόπος εξαγωγής και οι μέθοδοι επικύρωσης των μοντέλων που χρησιμοποιούνται στο υπόλοιπο της διπλωματικής εργασίας αναπτύσσονται στην Ενότητα 3.2. Στην Ενότητα 3.3 παρουσιάζεται η διαδικασία μετατροπής των μοντέλων αυτών σε μορφή πρόσφορη ώστε να εφαρμοστούν για προβλεπτικό σκοπό. Κλείνοντας, η διαδικασία που ακολουθείται προκειμένου το τελευταίο να καταστεί δυνατό περιγράφεται στην Ενότητα 3.4.

3.1 Εισαγωγικές έννοιες

Πριν εμβαθύνουμε στην έννοια της αναγνώρισης δυναμικών συστημάτων, πρέπει πρώτα να διευκρινιστεί τι εννοούμε αναφερόμενοι σε ένα δυναμικό σύστημα. Σε συμφωνία με τον τρόπο που συνήθως προσεγγίζεται το ζήτημα από την πλευρά των Συστημάτων Αυτόματου Ελέγχου (ΣΑΕ), ο Ljung^[11] περιγράφει μία διεργασία κατά την οποία μεταβλητές διαφόρων ειδών αλληλεπιδρούν προς σύνθεση παρατηρήσιμων σημάτων. Από τα τελευταία, εκείνα που μας ενδιαφέρουν (και συνεπώς μετρούμε) καλούνται σήματα εξόδου. Στη συνήθη περίπτωση που το σύστημα επηρεάζεται επίσης από εξωγενείς παράγοντες, αυτοί κατατάσσονται περαιτέρω σε δύο κατηγορίες. Όσοι παράγοντες αποτελούν μεταβλητές που είναι δυνατό να χειριστούμε ονομάζονται σήματα εισόδου. Οι υπόλοιποι θεωρούνται διαταραχές, καθώς επηρεάζουν τη συμπεριφορά του συστήματος χωρίς όμως να είναι εφικτό να παρέμβουμε και να μεταβάλλουμε τη μελλοντική εξέλιξή τους. Οι διαταραχές μπορεί να είναι μετρήσιμες ή όχι, με την παρουσία των δεύτερων να γίνεται αντιληπτή μόνο μέσω της επίδρασης που έχουν στα σήματα εξόδου. Τα προηγούμενα συνοψίζονται επίσης διαγραμματικά στο Σχήμα 3.1. Πέρα όμως από τη δομή, η έννοια του δυναμικού συστήματος έγκειται κυρίως στο γεγονός πως αυτό εξελίσσεται με την πάροδο του



Σχήμα 3.1 Διαγραμματική απεικόνιση δυναμικού συστήματος με έξοδο y , είσοδο u , μετρήσιμη διαταραχή w , και μη μετρήσιμη διαταραχή v

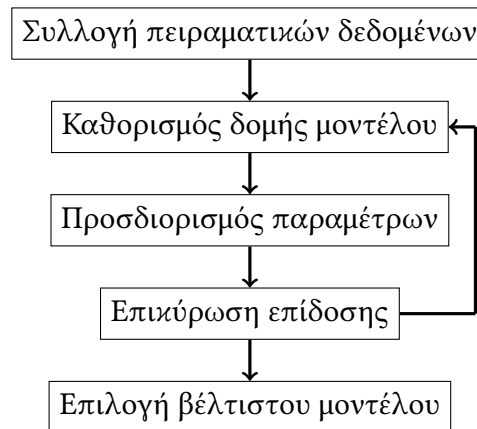
χρόνου. Στη συνέχεια μάλλον θα μας απασχολήσουν συστήματα τέτοια, που η έξοδος τους δεν εξαρτάται μόνο από τους εξωγενείς παράγοντες που επιδρούν επάνω τους στο παρόν, αλλά επίσης από τις παλαιότερες τιμές τους.

Η αναγνώριση δυναμικών συστημάτων^[11,8] αποσκοπεί στην παραμετρική μοντελοποίηση ενός πραγματικού συστήματος που γενικά βρίσκεται στη μορφή που περιγράφεται πιο πάνω. Η διαδικασία μπορεί να θεωρηθεί ότι αποτελείται από τρία επιμέρους βήματα:

Συλλογή πειραματικών δεδομένων. Αυτά αποτυπώνουν τη μεταβολή στο χρόνο των σημάτων εξόδου συναρτήσει της εισόδου που δίνεται στο σύστημα. Τα δεδομένα συλλέγονται είτε κατά τη διάρκεια της φυσιολογικής λειτουργίας του συστήματος, είτε, εφόσον αυτό είναι επιτρεπτό, μέσα από ειδικά σχεδιασμένα πειράματα που αποσκοπούν στη μεγιστοποίηση της πληροφορίας που παρέχεται για τη δυναμική του συστήματος.

Καθορισμός της δομής των υποψήφιων μοντέλων. Στη συνήθη περίπτωση επιλέγεται ένα σύνολο υποψήφιων μοντέλων, με διαφορετική δομή το καθένα, μεταξύ των οποίων επιθυμούμε να επιλέξουμε εκείνο που είναι ικανό να περιγράψει πιο πιστά το σύστημα. Αν υπάρχει πρότερη γνώση για τους μηχανισμούς που λαμβάνουν χώρα, τότε αυτό μπορεί να αποτελεί μία μαθηματική απεικόνιση με φυσικό νόημα. Σε αντίθετη περίπτωση, ή όταν το σύστημα είναι πολύ πολύπλοκο, επιλέγονται μοντέλα με δεδομένη δομή, που όμως δεν αποτυπώνουν καμία πληροφορία σχετικά με τους υποκείμενους μηχανισμούς. Για τον λόγο αυτό, τέτοιου είδους μοντέλα χαρακτηρίζονται ως 'μαύρα κουτιά'. Στο υπόλοιπο της εργασίας τα μοντέλα που θα μας απασχολήσουν ανήκουν σε αυτή την κατηγορία.

Προσδιορισμός και επικύρωση των μοντέλων. Για κάθε μοντέλο, ο προσδιορισμός των βέλτιστων παραμέτρων βασίζεται στην εφαρμογή κάποιας μεθόδου αναγνώρισης. Κοινό παρονομαστή μεταξύ των μεθόδων αυτών αποτελεί η προσπάθεια ελαχιστοποίησης μίας συνάρτησης κόστους που ποσοτικοποιεί τη διαφορά μεταξύ πραγματικών δεδομένων και πρόβλεψης. Ενώ η εφαρμογή μίας μεθόδου αναγνώρισης οδηγεί στον



Σχήμα 3.2 Η πορεία της αναγνώρισης δυναμικών συστημάτων

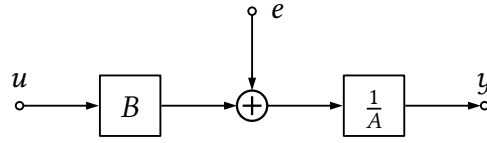
προσδιορισμό ενός μοντέλου, εντούτοις δεν εξασφαλίζει πως η πρόβλεψη που αυτό παρέχει θα είναι ικανοποιητική. Για τον σκοπό αυτό εφαρμόζονται μέθοδοι επικύρωσης της επίδοσης του κάθε μοντέλου, στη βάση των οποίων επιλέγεται τελικά και το βέλτιστο.

Η διαδικασία που περιγράφεται παραπάνω δίνεται επίσης στο Σχήμα 3.2. Πριν συνεχίσουμε στο υπόλοιπο του κεφαλαίου, είναι χρήσιμο να εισάγουμε μία σύμβαση που από εδώ και στο εξής θα εφαρμόζεται. Στο υπόλοιπο κείμενο, *οι μεταβλητές που παριστάνουν διανύσματα ή πίνακες θα σημειώνονται με έντονη γραμματοσειρά*. Για παράδειγμα, όπως ήδη αναφέρθηκε, στη γενική περίπτωση ένα δυναμικό σύστημα μπορεί να έχει περισσότερες από μία εισόδους, εξόδους, ή διαταραχές. Έτσι οι μεταβλητές του Σχήματος 3.1 είναι διανύσματα και ως τέτοια, σημειώνονται με έντονα γράμματα.

3.2 Πολυωνυμικά μοντέλα

3.2.1 Το αυτοπαλινδρομικό μοντέλο με εξωγενή είσοδο

Η γραμμική παλινδρόμηση επιχειρεί να προσδιορίσει ένα γραμμικό μοντέλο που συσχετίζει μία βαθμωτή απόκριση (ή εξαρτημένη μεταβλητή) με μία ή περισσότερες επεξηγηματικές (ή ανεξάρτητες) μεταβλητές. Στην περίπτωση που τη θέση των επεξηγηματικών μεταβλητών παίρνουν παλαιότερες τιμές της απόκρισης, τότε το μοντέλο που προκύπτει ονομάζεται αυτοπαλινδρομικό. Αν επιπρόσθετα λαμβάνονται υπόψη εξωγενείς επεξηγηματικές μεταβλητές, τότε πρόκειται για αυτοπαλινδρομικό μοντέλο με εξωγενή είσοδο ή



Σχήμα 3.3 Διαγραμματική απεικόνιση ARX μοντέλου

ARX (AutoRegressive model with eXogenous input)^[12] και σε διακριτό χρόνο είναι

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-n_k) + \dots + b_{n_b} u(t-n_k-n_b+1) + \epsilon(t) \quad (3.1)$$

Ένα τέτοιο μοντέλο συσχετίζει την τιμή της μεταβλητής εξόδου y σε χρόνο t με:

- τις n_a τελευταίες τιμές της εξόδου και
- n_b παλαιότερες τιμές εισόδου u , λαμβάνοντας υπόψη νεκρό χρόνο n_k

Στην τιμή της εξόδου συνυπολογίζεται επίσης ο όρος της διαταραχής ϵ . Απουσία παραπάνω πληροφορίας, το σύστημα θεωρείται πως επηρεάζεται από λευκό θόρυβο, ο οποίος υπολογίζεται σαν σειρά ανεξάρτητων και κανονικά κατανομημένων τυχαίων μεταβλητών με μηδενική μέση τιμή και τυπική απόκλιση σ , δηλαδή

$$\epsilon \sim \mathcal{N}_{iid}(0, \sigma^2) \quad (3.2)$$

Η Εξίσωση 3.1 συχνά δίνεται σε πιο συμπαγή μορφή κάνοντας χρήση του τελεστή υστέρησης q^{-k} , ο οποίος ορίζεται σύμφωνα με τον τύπο

$$q^{-k} x(t) = x(t-k) \quad (3.3)$$

με αποτέλεσμα ένα μοντέλο ARX να διατυπώνεται ισοδύναμα ως

$$A(q^{-1}) y(t) = B(q^{-1}) u(t) + \epsilon(t) \quad (3.4)$$

όπου για τα πολυώνυμα $A(q)$ και $B(q)$ είναι

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{n_a} q^{-n_a} \quad (3.5)$$

$$B(q) = b_1 q^{-n_k} + b_2 q^{-n_k-1} + \dots + b_{n_b} q^{-n_k-n_b+1} \quad (3.6)$$

Μία ακόμα πιο συμπαγής παρουσίαση επιτυγχάνεται αν το μοντέλο παρασταθεί με τη χρήση πινάκων. Ξεινώνοντας από την Εξίσωση 3.1 και ορίζοντας τα διανύσματα

$$\boldsymbol{\phi}(t) = \left[\begin{array}{ccc|ccc} -y(t-1) & \cdots & -y(t-n_a) & u(t-1) & \cdots & u(t-n_b) \end{array} \right]^T \quad (3.7)$$

$$\boldsymbol{\theta} = \left[\begin{array}{ccc|ccc} a_1 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} \end{array} \right]^T \quad (3.8)$$

Τελικά το μοντέλο μπορεί να δοθεί στη μορφή

$$\hat{y}(t|\boldsymbol{\theta}) = \boldsymbol{\phi}^T(t)\boldsymbol{\theta} \quad (3.9)$$

Παραπάνω παραλείπεται σκόπιμα ο όρος της διαταραχής ϵ . Η τιμή που υπολογίζεται αποτελεί συνεπώς πρόβλεψη και συμβολίζεται ως \hat{y} για να διακριθεί από την πραγματική τιμή εξόδου του συστήματος. Επειδή θα βρεθούμε προ του ίδιου ζητήματος αρκετά συχνά στη συνέχεια, αυτή η σύμβαση διατηρείται και στο υπόλοιπο κείμενο.

Στο Σχήμα 3.3 δίνεται το λογικό διάγραμμα του ARX μοντέλου. Από εδώ και στο εξής όταν αναφερόμαστε σε αυτό, θα μεταπηδούμε μεταξύ των αναπαραστάσεων που δίνονται στις Εξισώσεις 3.1, 3.4, και 3.9 ανάλογα με εκείνη που κρίνεται καταλληλότερη ανά περίπτωση. Στην πράξη βέβαια, όπως θα δειχθεί στη συνέχεια, η μορφή της Εξίσωσης 3.9 αποδεικνύεται ως η πιο χρήσιμη ανάμεσα στις τρεις.

3.2.2 Μία ευρύτερη προσέγγιση

Στην πραγματικότητα τα ARX μοντέλα ανήκουν σε μία ευρύτερη οικογένεια γραμμικών μοντέλων μαύρου κουτιού^[11], που συνοψίζονται στον τύπο

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}\epsilon(t) \quad (3.10)$$

όπου τα πολυώνυμα $A(q)$ και $B(q)$ ορίζονται σύμφωνα με τις Εξισώσεις 3.5 και 3.6 αντίστοιχα, ενώ τα υπόλοιπα με τρόπο ανάλογο του $A(q)$, δηλαδή

$$C(q) = 1 + c_1q^{-1} + c_2q^{-2} + \cdots + c_{n_c}q^{-n_c} \quad (3.11)$$

$$D(q) = 1 + d_1q^{-1} + d_2q^{-2} + \cdots + d_{n_d}q^{-n_d} \quad (3.12)$$

$$F(q) = 1 + f_1q^{-1} + f_2q^{-2} + \cdots + f_{n_f}q^{-n_f} \quad (3.13)$$

Τα μοντέλα αυτά ονομάζονται πολυωνυμικά και γεννιούνται από την ιδέα ότι το πραγματικό σύστημα μπορεί να περιγραφεί πλήρως από μία εξίσωση στη μορφή

$$y(t) = G(q, \boldsymbol{\theta})u(t) + H(q, \boldsymbol{\theta})\epsilon(t) \quad (3.14)$$

Οι συναρτήσεις μεταφοράς G και H που εμφανίζονται πιο πάνω (καθώς και η αντίστροφη συνάρτηση H^{-1}) είναι ευσταθείς¹ και στη γενικότερη περίπτωση ορίζονται σαν

$$G(q, \mathbf{g}) = \sum_{k=0}^{\infty} g(k)q^{-k} \quad H(q, \mathbf{h}) = 1 + \sum_{k=1}^{\infty} h(k)q^{-k} \quad (3.15)$$

Ο τελεστής υστέρησης q^{-k} δίνεται από την Εξίσωση 3.3. Η επιλογή πεπερασμένου αριθμού συντελεστών στις συναρτήσεις μεταφοράς οδηγεί στην μετάβαση από τη γενική μορφή τους, στην Εξίσωση 3.15, σε εκείνη της Εξίσωσης 3.14. Η ποσότητα $\boldsymbol{\theta}$ αποτελεί διάνυσμα των παραμέτρων \mathbf{g} και \mathbf{h} μαζί και ως τέτοιο περιορίζεται σε ένα υποσύνολο του \mathbb{R}^d ανά περίπτωση, όπου d είναι η διάσταση του διανύσματος $\boldsymbol{\theta}$, δηλαδή $\boldsymbol{\theta} \subset \mathbb{R}^d$.

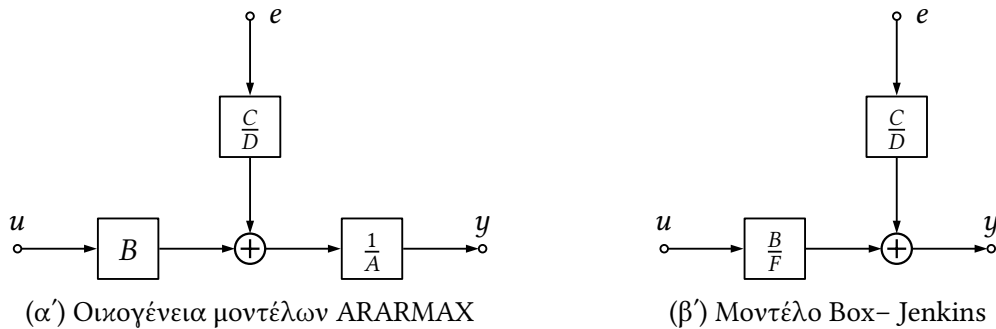
Έχοντας μία πληρέστερη εικόνα της Εξίσωσης 3.14 μπορούμε πλέον να στρέψουμε την προσοχή μας πάλι πίσω στο μοντέλο που ορίζεται από την Εξίσωση 3.10. Το πέρασμα από την πρώτη διατύπωση στη δεύτερη αποτελεί λογικό και επόμενο βήμα, καθώς προσφέρει σημαντικά μεγαλύτερη ευελιξία σε ότι αφορά την παραμετροποίηση του μοντέλου (που είναι άλλωστε και το ζητούμενο). Η μετάβαση αυτή έγκειται στην αναπαράσταση των συναρτήσεων μεταφοράς υπό τη μορφή των ρητών συναρτήσεων

$$G(q) = \frac{B(q)}{F(q)} \quad H(q) = \frac{C(q)}{D(q)} \quad (3.16)$$

Αν και παραλείπεται παραπάνω, όλα τα πολυώνυμα είναι συναρτήσεις του παραμετρικού διανύσματος $\boldsymbol{\theta}$. Στη συνήθη περίπτωση που επιθυμούμε να μοντελοποιήσουμε ένα πραγματικό δυναμικό σύστημα, ακόμα και αν υπάρχει γνώση των υποκείμενων μηχανισμών, αυτή δεν επαρκεί για τον καθορισμό των παραμέτρων του μοντέλου. Το γεγονός αυτό καθιστά αναγκαία τη χρήση κάποιου μεθόδου εκτίμησης του διανύσματος $\boldsymbol{\theta}$, πράγμα που αποτελεί ακρογωνιαίο λίθο της αναγνώρισης δυναμικών συστημάτων.

Στην πράξη βέβαια, το μοντέλο της Εξίσωσης 3.10 είναι πολύ γενικό για να είναι χρηστικό και έτσι είναι τυπικό να απλοποιείται καταλλήλως πριν τη χρήση του. Με τον τρόπο αυτό δημιουργούνται επιμέρους μοντέλα, καθένα από τα οποία πλεονεκτεί έναντι των άλλων ως προς την ικανότητα να περιγράφει συστήματα με δεδομένα χαρακτηρι-

¹Για τον λόγο που αυτό συμβαίνει και για τον αναγνώστη που επιθυμεί να εμβαθύνει, μία πλήρης ανάλυση επί του θέματος γίνεται από τον Ljung^[11].



Σχήμα 3.4 Δύο χαρακτηριστικές περιπτώσεις πολυωνυμικών μοντέλων

στικά. Αυτό μπορεί να γίνει κατανοητό κοιτώντας σε δύο χαρακτηριστικές οικογένειες μοντέλων, οι οποίες γεννιούνται μέσα από δύο διαφορετικές απλοποιητικές παραδοχές.

Στην πρώτη περίπτωση τίθεται

$$F(q) = 1$$

οπότε δημιουργείται η οικογένεια των ARARMAX μοντέλων, που περιγράφονται από το λογικό διάγραμμα του Σχήματος 3.4α'. Με την επιπλέον απλοποιητική παραδοχή

$$D(q) = 1$$

προκύπτει το αυτοπαλινδρομικό μοντέλο κινητού μέσου όρου με εξωγενή είσοδο ή ARMAX (AutoRegressive Moving Average model with eXogenous input). Αν επιπλέον τεθεί

$$C(q) = 1$$

τότε το γενικό μοντέλο που δίνεται από την Εξίσωση 3.10 απλοποιείται προς την Εξίσωση 3.4 και πλέον περιγράφει ένα ARX μοντέλο. Παρατηρούμε ότι αυτά διαφέρουν μόνο ως προς τον τρόπο που φιλτράρουν τη διαταραχή ϵ , με το μοντέλο ARMAX να προσφέρει μεγαλύτερο βαθμό ευελιξίας απλώς και μόνο λόγω της παρουσίας του πολυωνύμου $C(q)$ που αυξάνει τον αριθμό των παραμέτρων προς προσδιορισμό. Και στις δύο περιπτώσεις πάντως, ο θόρυβος εισάγεται στο ίδιο σημείο της διεργασίας και η αντίστοιχη συνάρτηση μεταφοράς μοιράζεται τους ίδιους πόλους (και άρα παρόμοια δυναμική) με την υπόλοιπη διεργασία. Αυτό φανερώνεται εύκολα από τη θέση του πολυωνύμου $A(q)$ στο Σχήμα 3.4α'.

Στη δεύτερη περίπτωση επιλέγεται

$$A(q) = 1$$

οπότε προκύπτει το μοντέλο Box–Jenkins του Σχήματος 3.4β'. Χωρίς περαιτέρω ανάλυση γίνεται άμεσα φανερό πως, σε αντίθεση με τα μοντέλα ARARMAX, η δυναμική του θορύβου είναι πλήρως ανεξάρτητη από τη δυναμική της διεργασίας. Βεβαίως και εδώ υπάρχει η δυνατότητα περαιτέρω απλοποιήσεων, οι οποίες όμως δεν μεταβάλλουν το σημείο εισόδου του θορύβου στη διεργασία, αλλά αλλάζουν τη δυναμική με την οποία αυτός λαμβάνεται υπόψη. Για παράδειγμα μπορεί να επιλεγεί

$$C(q) = D(q) = 1$$

οπότε προκύπτει το μοντέλο σφάλματος εξόδου ή ΟΕ (Output-Error model), στο οποίο ο θόρυβος προστίθεται αφιλτράριστος απευθείας στην έξοδο του μοντέλου.

Τα παραπάνω αναδεικνύουν αυτό που ειπώθηκε στην αρχή: το κάθε επιμέρους μοντέλο υπερτερεί ανάλογα με τα ιδιαίτερα χαρακτηριστικά του προς μελέτη συστήματος. Το ARX μοντέλο παρέχει μία ικανοποιητική περιγραφή ενώ παραμένει σχετικά απλό στον χειρισμό του. Αν απαιτείται καλύτερη περιγραφή της συνεισφοράς του θορύβου στη διεργασία, τότε μία καλύτερη επιλογή θα αποτελούσε το μοντέλο ARMAX. Στην ειδική περίπτωση που ο θόρυβος εισέρχεται αργά στη διεργασία, φυσική επιλογή αποτελεί το μοντέλο Box–Jenkins ή, όταν μία τόσο λεπτομερής περιγραφή δεν είναι θεμιτή, το μοντέλο σφάλματος εξόδου.

3.2.3 Η μέθοδος των ελαχίστων τετραγώνων

Όπως θα δειχθεί πιο κάτω, όταν αντιμετωπίζουμε το πρόβλημα της παραμετροποίησης ARX μοντέλων, τότε η μέθοδος των ελαχίστων τετραγώνων^[12] πλεονεκτεί καθώς οδηγεί σε αναλυτική λύση. Η εφαρμογή οποιασδήποτε μεθόδου αναγνώρισης απαιτεί την ύπαρξη πειραματικών δεδομένων υπό τη μορφή ζευγών σημάτων εισόδου \mathbf{u} –εξόδου \mathbf{y} . Αν αυτά είναι διαθέσιμα για τις χρονικές στιγμές $t \in [1, \dots, N]$, τότε περιέχονται στο διάνυσμα

$$\mathbf{Z}^N = \begin{bmatrix} y(1) & \cdots & y(N) & \vdots & u(1) & \cdots & u(N) \end{bmatrix} \quad (3.17)$$

Η μέθοδος στοχεύει στην ελαχιστοποίηση της αντικειμενικής συνάρτησης

$$\begin{aligned} V_N(\boldsymbol{\theta}, \mathbf{Z}^N) &= \frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t|\boldsymbol{\theta}))^2 = \\ &= \frac{1}{N} \sum_{t=1}^N (y(t) - \boldsymbol{\phi}^T(t)\boldsymbol{\theta})^2 \end{aligned} \quad (3.18)$$

όπου η πρόβλεψη δίνεται από ένα μοντέλο ARX στη μορφή της Εξίσωσης 3.9. Το παραμετρικό διάνυσμα που υπολογίζεται είναι εκείνο για το οποίο η παραπάνω συνάρτηση αποκτά την ελάχιστη τιμή της, δηλαδή

$$\hat{\theta}_N = \min_{\theta} V_N(\theta, Z^N) \quad (3.19)$$

Καθώς η συνάρτηση κόστους είναι τετραγωνική, για τον υπολογισμό του διανύσματος $\hat{\theta}$ που την ελαχιστοποιεί επαρκεί ο μηδενισμός της πρώτης παραγώγου

$$0 = \frac{d}{d\theta} V_N(\theta, Z^N) = \frac{2}{N} \sum_{t=1}^N \phi(t)(y(t) - \phi^T(t)\theta) \quad (3.20)$$

από τον οποίο προκύπτει ότι

$$\sum_{t=1}^N \phi(t)y(t) = \sum_{t=1}^N \phi(t)\phi^T(t)\theta \quad (3.21)$$

και τελικά το διάνυσμα $\hat{\theta}$ υπολογίζεται ως

$$\hat{\theta}_N = \left[\sum_{t=1}^N \phi(t)\phi^T(t) \right]^{-1} \sum_{t=1}^N \phi(t)y(t) \quad (3.22)$$

Αυτό είναι πολύ βοηθητικό, καθώς είναι δυνατός ο προσδιορισμός του βέλτιστου παραμετρικού διανύσματος $\hat{\theta}$ σε ένα βήμα, χωρίς την ανάγκη χρήσης κάποιας επαναληπτικής μεθόδου που θα κόστιζε σε υπολογιστική ισχύ αλλά και σε χρόνο. Στην παρούσα εργασία τα μοντέλα εξάγονται μέσω της διαδικασίας αυτής και στη συνέχεια μετασχηματίζονται στην κατάλληλη δομή που επιτρέπει τη χρήση τους για προβλεπτικούς σκοπούς.

3.2.4 Κριτήρια επικύρωσης

Η ποιότητα του μοντέλου που προκύπτει μπορεί να ελεγχθεί στη βάση κατάλληλων κριτηρίων επικύρωσης της επίδοσής του^[12,13]. Η αναλυτική παρουσίαση του τρόπου εξαγωγής αυτών^[11] είναι βεβαίως εκτός της φύσης της παρούσας εργασίας και για τον λόγο αυτό παραλείπεται. Σε κάθε περίπτωση πάντως, τα σφάλματα στα εκτιμώμενα μοντέλα μπορούν να γραφούν ως άθροισμα τριών επιμέρους όρων:

- Ο πρώτος αποτυπώνει το μέρος του ολικού σφάλματος που είναι αδύνατο να αποφευχθεί ακόμα και στην περίπτωση που υπάρχει πλήρης γνώση της δυναμικής του

συστήματος. Με άλλα λόγια, η γνώση της πρότερης κατάστασης του συστήματος δεν επαρκεί για την ακριβή περιγραφή της μελλοντικής εξέλιξης του, πράγμα λογικό αν λάβουμε υπόψη την επίδραση των μη μετρήσιμων διαταραχών.

- Ο δεύτερος όρος περιγράφει το σφάλμα μεροληψίας. Αυτό προκύπτει από την απόκλιση μεταξύ της δομής του εν χρήσει μοντέλου και του πραγματικού συστήματος. Στην τυπική περίπτωση αυτό το σφάλμα γίνεται λιγότερο σημαντικό καθώς αυξάνει ο αριθμός των παραμέτρων που χρησιμοποιούνται.
- Ο τρίτος και τελευταίος όρος είναι το σφάλμα διακύμανσης. Πρόκειται για ποσότητα ανάλογη του αριθμού των παραμέτρων του μοντέλου και αντιστρόφως ανάλογη του αριθμού των πειραματικών σημείων. Δεν εξαρτάται από τη δομή του μοντέλου, ούτε από τις πειραματικές συνθήκες που εφαρμόζονται.

Στην πράξη, καθώς ο αριθμός των παραμέτρων αυξάνει, το μοντέλο καταφέρνει να αποτυπώσει καλύτερα τη χαρακτηριστική συμπεριφορά των πειραματικών δεδομένων. Όταν όμως φτάσουμε και ξεπεράσουμε τον αριθμό των παραμέτρων που επιτρέπουν τη βέλτιστη αναπαράσταση του συστήματος, τότε οι επιπλέον παράμετροι περιγράφουν τον θόρυβο των πειραματικών δεδομένων στη βάση των οποίων γίνεται η εκτίμηση του μοντέλου. Αυτό ονομάζεται υπερεκτίμηση (overfit) και είναι φυσικά ανεπιθύμητη, εφόσον το μοντέλο πρόκειται να χρησιμοποιηθεί σε δεδομένα με διαφορετικό θόρυβο.

Από τα παραπάνω είναι φανερό πως ενώ με την προσθήκη επιπλέον παραμέτρων το σφάλμα μεροληψίας μειώνεται, ταυτόχρονα το σφάλμα διακύμανσης αυξάνει. Υπάρχουν λοιπόν αρκετά κριτήρια που προσπαθούν να ποσοτικοποιήσουν την αρνητική επίδραση που έχει η αύξηση της πολυπλοκότητας του μοντέλου επάνω στην προβλεπτική ικανότητά του. Εδώ χρησιμοποιούνται τα Akaike's (AIC) και Bayesian Information Criteria (BIC), που υπολογίζονται σύμφωνα με τους τύπους

$$\text{AIC} = N \log \left[\det \left(\frac{1}{N} \sum_{t=1}^N \left(y(t) - \hat{y}(t|\hat{\theta}) \right)^2 \right) \right] + N (\log(2\pi) + 1) + 2 \dim \hat{\theta} \quad (3.23)$$

$$\text{BIC} = N \log \left[\det \left(\frac{1}{N} \sum_{t=1}^N \left(y(t) - \hat{y}(t|\hat{\theta}) \right)^2 \right) \right] + N (\log(2\pi) + 1) + \dim \hat{\theta} \log N \quad (3.24)$$

Ένα ακόμα προφανές κριτήριο της προβλεπτικής ικανότητας ενός μοντέλου είναι το μέσο τετραγωνικό σφάλμα ή Mean Squared Error (MSE), που υπολογίζεται ως

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^N \left(y(t) - \hat{y}(t|\hat{\theta}) \right)^2 \quad (3.25)$$

Κλείνοντας θα πρέπει να τονιστεί η σημασία της αντεπικύρωσης (cross-validation). Η έννοια έγκειται στον διαχωρισμό των διαθέσιμων πειραματικών δεδομένων σε δύο σύνολα. Το πρώτο χρησιμοποιείται για την εκτίμηση του μοντέλου και το δεύτερο για την επικύρωσή του, χρησιμοποιώντας τα κριτήρια των Εξισώσεων 3.23–3.25, παραπάνω. Εδώ σημειώνεται πως η εφαρμογή της αντεπικύρωσης είναι καθοριστικής σημασίας, ειδάλλως η χρήση των κριτηρίων δεν οδηγεί σε σωστά αποτελέσματα. Στη συνήθη περίπτωση τα $\frac{2}{3}$ των δεδομένων χρησιμοποιούνται για εκτίμηση και το υπόλοιπο $\frac{1}{3}$ για επικύρωση.

3.3 Μετατροπή σε μοντέλο χώρου καταστάσεων

Ως τώρα έχουμε εκφράσει ένα μοντέλο ARX χρησιμοποιώντας τρεις ισοδύναμες αναπαράστασεις (βλ. Εξισώσεις 3.1, 3.4, και 3.9). Στα ΣΑΕ συνηθίζεται, όμως, ένα σύστημα γραμμικών διαφορικών εξισώσεων να δίνεται στη μορφή μοντέλου χώρου καταστάσεων. Η ανωτερότητα της αναπαράστασης αυτής όταν ένα μοντέλο πρόκειται να χρησιμοποιηθεί για προβλεπτικό σκοπό θα γίνει φανερή στην επόμενη ενότητα. Για την ώρα θα εστιάσουμε στην μετατροπή από τη μία μορφή στην άλλη^[8,14].

Η εξίσωση κατάστασης ενός συστήματος με μία μεταβλητή εισόδου u και μία έξοδο y σε διακριτό χρόνο δίνεται από το σύστημα

$$\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) \quad (3.26)$$

$$\hat{y}(t) = \mathbf{C}\hat{\mathbf{x}}(t) + \mathbf{D}u(t) \quad (3.27)$$

όπου αν για την περιγραφή χρησιμοποιούνται n_x μεταβλητές κατάστασης, τότε

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} x_1(t) & \cdots & x_{n_x}(t) \end{bmatrix}^T \quad (3.28)$$

Στη συνέχεια θα δείξουμε ότι ένα ARX μοντέλο που δίνεται από την Εξίσωση 3.1 μπορεί να γραφεί σε κανονική παρατηρήσιμη μορφή στον χώρο των καταστάσεων ως

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n_a-1} & 0 & 0 & \cdots & 1 \\ -a_{n_a} & 0 & 0 & \cdots & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ \vdots \\ b_{n_b} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \quad \mathbf{D} = 0 \quad (3.29)$$

Προκειμένου αυτό να γίνει εύκολα κατανοητό, θα στηριχθούμε σε ένα απλό παράδειγμα.

Παράδειγμα: Έχουμε στη διάθεσή μας ένα ARX μοντέλο για το οποίο είναι

$$n_a = 3 \qquad n_b = 2 \qquad n_k = 1$$

και συνεπώς περιγράφεται από την εξίσωση

$$\begin{aligned} \hat{y}(t) + a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) = \\ b_1 u(t-1) + b_2 u(t-2) \end{aligned} \quad (3.30)$$

Χωρίς περαιτέρω εξήγηση, θα ξεκινήσουμε αρχικοποιώντας το διάνυσμα $\hat{x}(t)$ ως εξής

$$\hat{x}(t) = \begin{bmatrix} -a_1 y(t-1) - a_2 y(t-2) - a_3 y(t-3) + b_1 u(t-1) + b_2 u(t-2) \\ -a_2 y(t-1) - a_3 y(t-2) + b_2 u(t-1) \\ -a_3 y(t-1) \end{bmatrix} \quad (3.31)$$

Από την Εξίσωση 3.30 παρατηρούμε πως η πρώτη γραμμή του πίνακα ισούται με $y(t)$ και άρα αυτός μπορεί να ξαναγραφτεί στη μορφή

$$\hat{x}(t) = \begin{bmatrix} \hat{y}(t) \\ -a_2 y(t-1) - a_3 y(t-2) + b_2 u(t-1) \\ -a_3 y(t-1) \end{bmatrix} \quad (3.32)$$

η οποία βρίσκεται σε συμφωνία με την Εξίσωση 3.27, δηλαδή σε χρόνο t η πρώτη γραμμή του πίνακα προβλέπει την έξοδο του συστήματος $\hat{y}(t)$.

Αρκεί τώρα να μπορέσουμε να υπολογίσουμε την έξοδο για την επόμενη χρονική στιγμή, $\hat{y}(t+1)$, ορίζοντας τους πίνακες στη μορφή της Εξίσωσης 3.29 και να καταλήξουμε σε ένα διάνυσμα $\hat{x}(t+1)$ μορφής όμοιας με εκείνη του $\hat{x}(t)$, αλλά μετατοπισμένο μία χρονική στιγμή μπροστά στον χρόνο. Έτσι βεβαιώνουμε ότι αν επαναλάβουμε τη διαδικασία που παρουσιάζεται εδώ, τότε μπορούμε να προβλέψουμε την έξοδο του συστήματος και για χρόνους $t+2, t+3$ κ.ο.κ. Εφαρμόζοντας την Εξίσωση 3.26 με πίνακες

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T \quad \mathbf{D} = 0 \quad (3.33)$$

προκύπτει το νέο διάνυσμα των μεταβλητών κατάστασης

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} -a_1x_1(t) + x_2(t) + b_1u(t) \\ -a_2x_1(t) + x_3(t) + b_2u(t) \\ -a_3x_1(t) \end{bmatrix} \quad (3.34)$$

και αντικαθιστώντας με τις τιμές της προηγούμενης χρονικής στιγμής από την Εξίσωση 3.32, τότε αυτό αποκτά τη μορφή

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} -a_1\hat{y}(t) - a_2y(t-1) - a_3y(t-2) + b_1u(t) + b_2u(t-1) \\ -a_2\hat{y}(t) - a_3y(t-1) + b_2(t) \\ -a_3\hat{y}(t) \end{bmatrix} \quad (3.35)$$

Συγκρίνοντας και πάλι με την Εξίσωση 3.30 γίνεται φανερό ότι είναι

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} \hat{y}(t+1) \\ -a_2\hat{y}(t) - a_3y(t-1) + b_2(t) \\ -a_3\hat{y}(t) \end{bmatrix} \quad (3.36)$$

οπότε το διάνυσμα έχει διατηρήσει τη μορφή που είχε προηγουμένως, στην Εξίσωση 3.32, αλλά οι όροι έχουν μετατοπισθεί μία χρονική στιγμή μπροστά στον χρόνο. Αξίζει να παρατηρήσουμε επίσης ότι πλέον υπεισέρχονται μεταβλητές που αποτελούν πρόβλεψη και όχι πραγματική τιμή εξόδου του συστήματος. Το ζήτημα αυτό είναι ιδιαίτερα σημαντικό, ειδικά από τη στιγμή που τα μοντέλα αυτά πρόκειται να χρησιμοποιηθούν για προβλεπτικό σκοπό και έτσι θα το αναλύσουμε αμέσως παρακάτω.

3.4 Χρήση για προβλεπτικό σκοπό

Είναι επιθυμητό ένα ARX μοντέλο να χρησιμοποιηθεί για πρόβλεψη σε ορίζοντα μεγαλύτερο της μίας στιγμής μπροστά στο μέλλον. Αν με τη χρονική στιγμή t δηλώσουμε το παρόν, τότε μία πρόβλεψη N βημάτων έγκειται στη χρήση του μοντέλου για τον υπολογισμό της τιμής εξόδου \hat{y} στις χρονικές στιγμές $t+1, \dots, t+N$. Για τους υπολογισμούς έχουμε στη διάθεσή μας τις τιμές εξόδου του συστήματος ως τη χρονική στιγμή t , ενώ οι τιμές εισόδου είναι επιπλέον γνωστές και καθόλη τη διάρκεια του ορίζοντα πρόβλεψης, ως τη χρονική στιγμή $t+N$. Εδώ πρέπει να θυμηθεί κανείς ότι η τιμή εισόδου υπόκειται σε δικό μας χειρισμό και άρα δε θα πρέπει να αποτελεί έκπληξη το γεγονός ότι γνωρίζουμε εκ των προτέρων την εξέλιξή της στο μέλλον.

Αν επιστρέψουμε στην αναπαράσταση της Εξίσωσης 3.1, διαπιστώνουμε πως ενώ η πρόβλεψη για μία στιγμή μπροστά στον χρόνο είναι εφικτή, οι υπόλοιπες απαιτούν τιμές εξόδου που δεν έχουμε ακόμα στη διάθεσή μας. Το πρόβλημα αυτό μπορεί να επιλυθεί μόνο με την αντικατάσταση όσων τιμών δε γνωρίζουμε ακόμα με τις αντίστοιχες που έχουν προβλεφθεί προηγουμένως. Αυτό μάλιστα συμβαίνει στο παράδειγμα της προηγούμενης ενότητας, όπου παρατηρούμε ότι στο διάνυσμα $\hat{\mathbf{x}}(t+1)$ της Εξίσωσης 3.36 εμφανίζεται η πρόβλεψη $\hat{y}(t)$ και όχι η πραγματική τιμή της εξόδου στην προηγούμενη χρονική στιγμή. Εδώ γεννιέται βέβαια το εξής ερώτημα: *αν η τιμή $y(t)$ είναι γνωστή όπως είπαμε προηγουμένως, τότε γιατί χρησιμοποιείται η πρόβλεψη $\hat{y}(t)$ στη θέση της;*

Την απάντηση σε αυτό δίνει η έλλειψη κάποιου όρου στο μοντέλο των Εξισώσεων 3.26 και 3.27 που να λαμβάνει υπόψη τον θόρυβο. Στη συνέχεια το ζήτημα αυτό επιλύεται, αλλά όχι με τη μορφή της στοχαστικής προσέγγισης της Εξίσωσης 3.2. Εναλλακτικά θα θεωρήσουμε ως θόρυβο το μέρος εκείνο της τιμής της εξόδου που το μοντέλο αδυνατεί να περιγράψει, δηλαδή τη διαφορά της πρόβλεψης από την πραγματική τιμή. Για τον σκοπό αυτό προστίθεται ένας παρατηρητής που παρακολουθεί την εξέλιξη της τιμής του σφάλματος. Ο βέλτιστος παρατηρητής ονομάζεται φίλτρο Kalman^[11,8] και η εφαρμογή του απαιτεί ένα μοντέλο διατυπωμένο σε μορφή χώρου καταστάσεων, γεγονός που δικαιολογεί και την επαναδιατύπωση του μοντέλου στην Ενότητα 3.3.

Με την προσθήκη του όρου του θορύβου, το μοντέλο γράφεται πλέον στη μορφή

$$\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{B}_d d(t) \quad (3.37)$$

$$y(t) = \mathbf{C}\hat{\mathbf{x}}(t) + \mathbf{D}u(t) + d(t) \quad (3.38)$$

όπου το σφάλμα υπολογίζεται από τον τύπο

$$\begin{aligned} d(t) &= y(t) - \hat{y}(t) = \\ &= y(t) - \mathbf{C}\hat{\mathbf{x}}(t) - \mathbf{D}u(t) \end{aligned} \quad (3.39)$$

Όπως γίνεται σαφές, για να εφαρμοστεί η παραπάνω σχέση απαιτείται η γνώση της πραγματικής εξόδου του συστήματος. Το φίλτρο Kalman εφαρμόζεται λοιπόν μόνο για την πρόβλεψη ενός βήματος μπροστά στον χρόνο. Οι επόμενες προβλέψεις βασίζονται στην επαναληπτική χρήση του αρχικά διατυπωμένου μοντέλου των Εξισώσεων 3.26 και 3.27, όπου η έλλειψη κάποιου όρου θορύβου είναι αναπόφευκτη και προμηνύει τη χρήση τιμών πρόβλεψης και όχι πραγματικών. Το διάνυσμα \mathbf{B}_d είναι ο πίνακας ενίσχυσης του φίλτρου που για την περίπτωση του ARX μοντέλου ορίζεται ως

$$\mathbf{B}_d = \begin{bmatrix} -a_1 & \cdots & -a_{n_a} \end{bmatrix}^T \quad (3.40)$$

Για να γίνει κατανοητή η εφαρμογή των Εξισώσεων 3.37 και 3.38 στην πρόβλεψη ενός βήματος μπροστά στον χρόνο, στη συνέχεια δίνεται ένα παράδειγμα που στην ουσία αποτελεί επέκταση εκείνου της προηγούμενης ενότητας.

Παράδειγμα Τη χρονική στιγμή t έχουμε στη διάθεσή μας ένα διάνυσμα $\hat{\mathbf{x}}(t)$ στη μορφή της Εξίσωσης 3.32, οπότε και γράφουμε

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} \hat{y}(t) \\ -a_2y(t-1) - a_3y(t-2) + b_2u(t-1) \\ -a_3y(t-1) \end{bmatrix} \quad (3.41)$$

Εφαρμόζοντας την Εξίσωση 3.37 έχουμε το νέο διάνυσμα των μεταβλητών κατάστασης

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} -a_1x_1(t) + x_2(t) + b_1u(t) - a_1d(t) \\ -a_2x_1(t) + x_3(t) + b_2u(t) - a_2d(t) \\ -a_3x_1(t) - a_3d(t) \end{bmatrix} \quad (3.42)$$

και αντικαθιστώντας στη συνέχεια τις τιμές της Εξίσωσης 3.41 προκύπτει

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} -a_1[\hat{y}(t) - d(t)] - a_2y(t-1) - a_3y(t-2) + b_1u(t) + b_2u(t-1) \\ -a_2[\hat{y}(t) - d(t)] - a_3y(t-1) + b_2(t) \\ -a_3[\hat{y}(t) - d(t)] \end{bmatrix} \quad (3.43)$$

Αν λάβουμε επίσης υπόψη τον ορισμό της διαταραχής από την Εξίσωση 3.39, τότε

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} -a_1y(t) - a_2y(t-1) - a_3y(t-2) + b_1u(t) + b_2u(t-1) \\ -a_2y(t) - a_3y(t-1) + b_2(t) \\ -a_3y(t) \end{bmatrix} \quad (3.44)$$

οπότε καταλήγουμε στο επιθυμητό αποτέλεσμα. Αν συγκρίνουμε το διάνυσμα $\hat{\mathbf{x}}(t+1)$ που υπολογίζεται παραπάνω με εκείνο της Εξίσωσης 3.35 στο τέλος του παραδείγματος της προηγούμενης ενότητας, τότε γίνεται φανερό πως πλέον ο υπολογισμός βασίζεται στην πραγματική έξοδο του συστήματος και όχι στην πρόβλεψη $\hat{y}(t)$. Τελικά

$$\hat{\mathbf{x}}(t+1) = \begin{bmatrix} \hat{y}(t+1) \\ -a_2y(t) - a_3y(t-1) + b_2(t) \\ -a_3y(t) \end{bmatrix} \quad (3.45)$$

όπου γίνεται χρήση της Εξίσωσης 3.30. Αν επιθυμούμε να προβλέψουμε την επόμενη έξοδο, $\hat{y}(t + 2)$, τότε εφαρμόζεται η Εξίσωση 3.26 και στη θέση της τιμής $y(t + 1)$ που δεν είναι ακόμα γνωστή υπεισέρχεται η αντίστοιχη πρόβλεψη $\hat{y}(t + 1)$. Η επαναληπτική εφαρμογή του τύπου επιτρέπει τελικά τον υπολογισμό προβλέψεων πολλαπλών βημάτων μπροστά στον χρόνο.

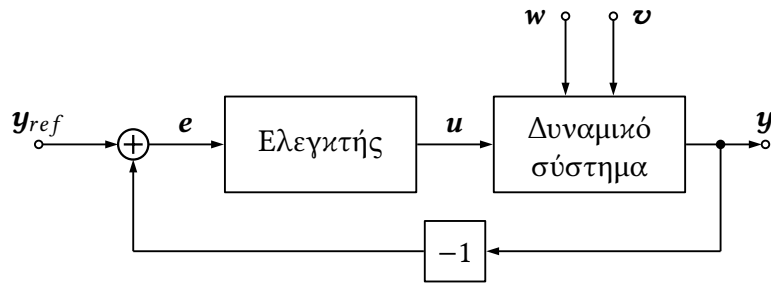
Κεφάλαιο 4

Ρύθμιση προβλεπτικού μοντέλου

Το κεφάλαιο εστιάζει στη μέθοδο της ρύθμισης προβλεπτικού μοντέλου. Η αρχή λειτουργίας της μεθόδου αναπτύσσεται στην Ενότητα 4.1, όπου τοποθετείται στο πλαίσιο μίας ευρύτερης εισαγωγικής συζήτησης με σημείο εκκίνησης την κλασική αντιμετώπιση ενός προβλήματος αυτομάτου ελέγχου. Η μαθηματική διατύπωση του προβλήματος βελτιστοποίησης που διαμορφώνεται δίνεται στην Ενότητα 4.2. Αμέσως μετά, στην Ενότητα 4.3, το τελευταίο μετατρέπεται σε πρόβλημα τετραγωνικού προγραμματισμού, όπως και επιλύεται τελικά.

4.1 Εισαγωγικές έννοιες

Ο στόχος των ΣΑΕ μπορεί να οριστεί με πολλούς εναλλακτικούς τρόπους. Ένας ευρύς ορισμός θα έλεγε πως πρόκειται για διατάξεις που στοχεύουν στην κατά βούληση μεταβολή της δυναμικής συμπεριφοράς ενός συστήματος^[15]. Στην πράξη αυτό επιτυγχάνεται με τον σχεδιασμό ενός ελεγκτή, ο οποίος καθορίζει τα σήματα εισόδου του συστήματος με τρόπο τέτοιο ώστε τα ρυθμιζόμενα σήματα εξόδου να είναι τα αναμενόμενα. Στη συντριπτική πλειοψηφία των περιπτώσεων, οι διατάξεις που συναντώνται εφαρμόζουν έλεγχο κλειστού βρόγχου, όπου τα σήματα εξόδου του συστήματος τροφοδοτούνται ως σήματα εισόδου στον ελεγκτή. Αυτό ονομάζεται ανατροφοδότηση και δίνει τη δυνατότητα στον ελεγκτή να συγκρίνει τα τρέχοντα σήματα της διεργασίας με τα επιθυμητά και να πράξει αναλόγως. Αυτό βεβαίως είναι επιθυμητό επειδή η έξοδος ενός δυναμικού συστήματος δεν εξαρτάται αποκλειστικά από τα σήματα εισόδου που βρίσκονται υπό τον έλεγχό μας. Όπως αναφέρθηκε στην Ενότητα 3.1, σε ένα σύστημα επιδρούν διαταραχές, δηλαδή σήματα που είτε δεν μετρούμε, είτε είναι αδύνατο να μετρήσουμε. Αν αυτές επηρεάζουν τη ρυθμιζόμενη έξοδο, τότε η ύπαρξη ανατροφοδότησης θα παρέχει στον ρυθμιστή την επιπλέον αυτή γνώση και θα του δώσει την ευκαιρία να τη λάβει υπόψη του στον καθορισμό του σήματος εισόδου. Η εφαρμογή ελέγχου κλειστού βρόγχου στο σύστημα του



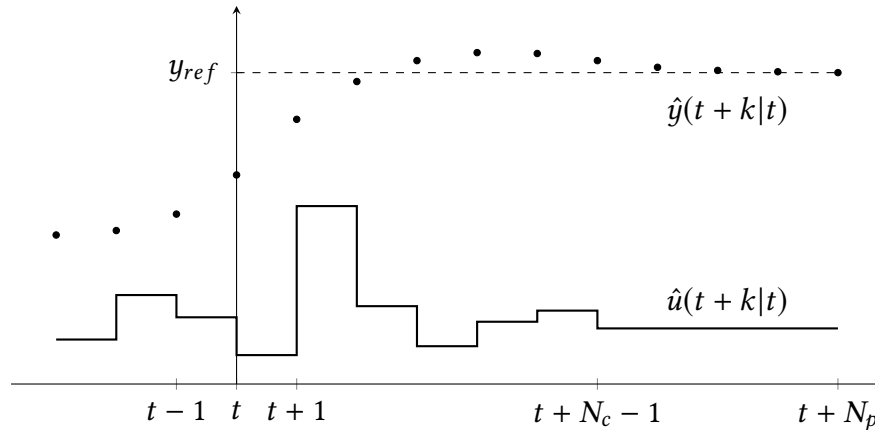
Σχήμα 4.1 Διαγραμματική απεικόνιση ελέγχου κλειστού βρόχου

Σχήματος 3.1 απεικονίζεται διαγραμματικά στο Σχήμα 4.1. Όπως βλέπουμε, η επιθυμητή τιμή y_{ref} των σημάτων εξόδου συγκρίνεται με την πραγματική και υπολογίζεται το ρυθμιστικό σφάλμα e που τελικά οδηγείται ως είσοδος στον ελεγκτή.

Μία νεότερη προσέγγιση στο πρόβλημα αποτελεί η μέθοδος της ρύθμισης προβλεπτικού μοντέλου ή MPC (Model Predictive Control)^[16]. Ενώ στην κλασική προσέγγιση που περιγράφηκε παραπάνω η απόφαση του ελεγκτή εξαρτάται αποκλειστικά από την τιμή του ρυθμιστικού σφάλματος την κάθε χρονική στιγμή, η ρύθμιση προβλεπτικού μοντέλου, όπως άλλωστε προδίδει το όνομα της, βασίζεται στην ύπαρξη ενός μοντέλου του συστήματος. Αυτό χρησιμοποιείται για την πρόβλεψη των τιμών εξόδου στη διάρκεια κάποιου πεπερασμένου χρονικού ορίζοντα μπροστά στο μέλλον. Αυτός ονομάζεται ορίζοντας πρόβλεψης. Σε κάθε χρονική στιγμή ο ελεγκτής καλείται να επιλύσει ένα πρόβλημα βελτιστοποίησης της ακολουθίας των μελλοντικών τιμών εισόδου που πρέπει να επιβάλει στο σύστημα προκειμένου να οδηγήσει την έξοδο στην επιθυμητή τιμή-στόχο, δηλαδή να μηδενίσει το ρυθμιστικό σφάλμα. Πολλές φορές οι αποφάσεις αυτές λαμβάνονται στη διάρκεια ενός διαστήματος που είναι μικρότερο του ορίζοντα πρόβλεψης. Αυτό το διάστημα αποτελεί τον ορίζοντα ελέγχου. Αφού καθοριστεί η βέλτιστη ακολουθία μελλοντικών εισόδων, ο ελεγκτής επιβάλει στο σύστημα μόνο την πρώτη σε σειρά τιμή και την επόμενη χρονική στιγμή η διαδικασία επαναλαμβάνεται από την αρχή. Για τον λόγο αυτό η μέθοδος αποκαλείται εναλλακτικά 'έλεγχος ολισθαίνοντα ορίζοντα'.

Τα παραπάνω μπορούν να γίνουν ευκολότερα κατανοητά με τη βοήθεια του Σχήματος 4.2. Αν τη χρονική στιγμή t επιλύεται το πρόβλημα βελτιστοποίησης με ορίζοντα ελέγχου N_c και πρόβλεψης N_p , τότε ο ελεγκτής καλείται να καθορίσει τη βέλτιστη μελλοντική αλληλουχία τιμών εισόδου, $\hat{u}(t, \dots, t + N_c - 1|t)$, προκειμένου η προβλεπόμενη εξέλιξη της τιμής εξόδου, $\hat{y}(t+1, \dots, t + N_p|t)$, να οδηγηθεί στην τιμή-στόχο y_{ref} . Στο χρονικό διάστημα πέρα από τον ορίζοντα ελέγχου και ως το τέλος του ορίζοντα πρόβλεψης συνήθως επιβάλλεται σταθερά η τελευταία πρόβλεψη για την τιμή εισόδου, δηλαδή

$$\hat{u}(t + N_c, \dots, t + N_p) = \hat{u}(t + N_c - 1|t) \quad (4.1)$$



Σχήμα 4.2 Απεικόνιση του ολισθαίνοντα ορίζοντα στη ρύθμιση προβλεπτικού μοντέλου

Σε κάθε περίπτωση πάντως, μόνο η πρώτη σε σειρά πρόβλεψη για την τιμή εισόδου, $\hat{u}(t|t)$, επιβάλλεται τελικά στο σύστημα. Την επόμενη χρονική στιγμή ο ορίζοντας εντός του οποίου επιλύεται το πρόβλημα βελτιστοποίησης μετατοπίζεται ένα βήμα προς τα δεξιά και η όλη διαδικασία επαναλαμβάνεται. Αυτό βέβαια υπονοεί πως σε κάθε χρονική στιγμή το μήκος του ορίζοντα πρόβλεψης και ελέγχου παραμένει σταθερό. Ο έλεγχος ολισθαίνοντα ορίζοντα έχει τις βάσεις του στο γεγονός πως κατά το πέρασμα του συστήματος από τη μία χρονική στιγμή στην επόμενη, έρχονται στην κατοχή μας νεότερα δεδομένα. Το πρόβλημα βελτιστοποίησης επιλύεται λοιπόν ξανά σε κάθε χρονική στιγμή, ώστε ο ελεγκτής να λάβει υπόψη του την πιο πρόσφατη διαθέσιμη πληροφορία.

Στην ουσία λοιπόν, ο έλεγχος προβλεπτικού μοντέλου βασίζεται στη σε πραγματικό χρόνο επίλυση ενός προβλήματος ελέγχου ανοιχτού βρόγχου κατά μήκος ενός πεπερασμένου ορίζοντα. Αυτό έρχεται σε αντίθεση με την κλασική προσέγγιση που περιγράφηκε στην αρχή και η οποία απαιτεί τον εκ των προτέρων προσδιορισμό ενός νόμου ελέγχου με ανατροφοδότηση. Από τη μία πλευρά, ο τρόπος λειτουργίας του ελέγχου προβλεπτικού μοντέλου δημιουργεί την επιπρόσθετη ανάγκη για δυνατότητα επίλυσης του προβλήματος βελτιστοποίησης σε λογικά χρονικά πλαίσια. Για τον λόγο αυτό άλλωστε η επίλυση λαμβάνει χώρα στη διάρκεια ενός πεπερασμένου ορίζοντα. Από την άλλη πλευρά όμως προσφέρει σημαντικά πλεονεκτήματα^[16]. Οι τιμές εισόδου και εξόδου ενός συστήματος συνήθως υπόκεινται σε περιορισμούς, είτε φυσικούς, είτε για λόγους ασφαλείας. Η μέθοδος δίνει τη δυνατότητα σχεδιασμού ελεγκτών που λαμβάνουν αυτούς τους περιορισμούς υπόψη κατά τη λειτουργία τους, ενώ υπερτερεί επίσης στην εφαρμογή σε μη γραμμικά συστήματα με δυναμική που μεταβάλλεται στον χρόνο. Η μέθοδος έχει μάλιστα αποδειχτεί ικανή να εξασφαλίζει ευστάθεια σε συστήματα που ο κλασικός έλεγχος με ανατροφοδότηση αδυνατεί να σταθεροποιήσει.

4.2 Διαμόρφωση του προβλήματος

4.2.1 Έλεγχος μηδενικού σφάλματος

Ο ελεγκτής που θα μας απασχολήσει βασίζεται στην αναπαράσταση του μοντέλου της διεργασίας στον χώρο των καταστάσεων και σε διακριτό χρόνο. Με το μοντέλο αυτό έχουμε ήδη καταπιαστεί πιο πάνω και γνωρίζουμε ότι η πρόβλεψη για ένα βήμα μπροστά στον χρόνο υπολογίζεται από το σύστημα των Εξισώσεων 3.37 και 3.38, δηλαδή

$$\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}u(t) + \mathbf{B}_d d(t) \quad (4.2)$$

$$y(t) = \mathbf{C}\hat{\mathbf{x}}(t) + d(t) \quad (4.3)$$

Εδώ υπενθυμίζεται ότι, όπως συζητήθηκε στην Ενότητα 3.4, για τις προβλέψεις δύο και περισσότερων βημάτων πρέπει να γίνει επαναληπτική εφαρμογή των Εξισώσεων 3.26 και 3.27 στη θέση αυτών που δίνονται παραπάνω. Η διαφορά των τελευταίων σε σχέση με τις πρώτες έγκειται στην απουσία των όρων διαταραχής $\mathbf{B}_d d(t)$ και $d(t)$, πράγμα που οφείλεται στην αδυναμία εφαρμογής φίλτρου Kalman στις περιπτώσεις αυτές.

Αν η τιμή-στόχος για την έξοδο είναι y_{ref} , τότε ο ελεγκτής προσπαθεί να οδηγήσει το σύστημα σε μόνιμη κατάσταση με μηδενικό ρυθμιστικό σφάλμα

$$0 = e = y_s - y_{ref} \quad (4.4)$$

Όμως το μοντέλο στη μορφή που περιγράφεται παραπάνω δεν επαρκεί ώστε ο ελεγκτής να είναι ικανός να επιτύχει μηδενικό ρυθμιστικό σφάλμα στην περίπτωση που το σύστημα επηρεάζεται από κάποια μη μετρήσιμη βηματική διαταραχή^[17]. Αυτή μπορεί βεβαίως να οφείλεται και στο ίδιο το μοντέλο της διεργασίας, αν αυτό συγκλίνει σε διαφορετικές μόνιμες συνθήκες από ότι το πραγματικό σύστημα. Στη βιβλιογραφία διατυπώνονται πολλές διαφορετικές προσεγγίσεις που εξασφαλίζουν ρύθμιση μηδενικού σφάλματος^[14,17,18]. Εδώ επιλέγεται να γίνει χρήση του μοντέλου σταθερής διαταραχής των μεταβλητών κατάστασης^[17,19]. Η σωστή λειτουργία της συγκεκριμένης υλοποίησης βασίζεται σε δύο σημεία, τα οποία αναπτύσσονται στη συνέχεια.

Το πρώτο αφορά στην προσαύξηση του συστήματος των Εξισώσεων 3.26 και 3.27 με την προσθήκη μίας επιπλέον μεταβλητής κατάστασης για τη διαταραχή. Με τον τρόπο αυτό οι προβλέψεις ενός και περισσότερων βημάτων υπολογίζονται πλέον από

την επαναληπτική χρήση του συστήματος

$$\begin{bmatrix} \hat{\mathbf{x}}(t+1) \\ \hat{d}(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_d \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}(t) \\ \hat{d}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(t) \quad (4.5)$$

$$y(t) = \begin{bmatrix} \mathbf{C} & \mathbf{C}_d \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}(t) \\ \hat{d}(t) \end{bmatrix} \quad (4.6)$$

που στην ουσία αποτυπώνει την υπόθεση ότι η διαταραχή που υπολογίζεται αρχικά είναι σταθερή και έτσι συνεχίζει να επηρεάζει όλες τις μελλοντικές μεταβλητές κατάστασης.

Το δεύτερο σημείο αφορά τον προσδιορισμό της επιθυμητής μόνιμης κατάστασης του συστήματος, κατά τον οποίο πρέπει να ληφθεί υπόψη η επίδραση της προαναφερθείσας διαταραχής. Αν υπό μόνιμες συνθήκες είναι

$$u(t+1) = u(t) = u_s \quad (4.7) \quad \hat{\mathbf{x}}(t+1) = \hat{\mathbf{x}}(t) = \hat{\mathbf{x}}_s \quad (4.8)$$

$$\hat{y}(t+1) = \hat{y}(t) = \hat{y}_s \quad (4.9) \quad d(t+1) = d(t) = d_s \quad (4.10)$$

τότε απαιτώντας την ισχύ της Εξίσωσης 4.4, από τις Εξισώσεις 4.5 και 4.6 προκύπτει

$$(\mathbf{A} - \mathbf{I})\hat{\mathbf{x}}_s + \mathbf{B}u_s = -\mathbf{B}_d d_s \quad (4.11)$$

$$\mathbf{C}\hat{\mathbf{x}}_s = y_{ref} - d_s \quad (4.12)$$

και από την επίλυση του παραπάνω συστήματος υπολογίζονται τελικά

$$\hat{\mathbf{x}}_s = (\mathbf{I} - \mathbf{A})^{-1} [\mathbf{B}u_s + \mathbf{B}_d d_s] \quad (4.13)$$

$$\hat{u}_s = [\mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}]^{-1} (y_{ref} - d_s) - \mathbf{B}^{-1}\mathbf{B}_d d_s \quad (4.14)$$

Το σύστημα των εξισώσεων 4.11 και 4.12 είναι απαραίτητο να επιλύεται ξανά σε κάθε χρονική στιγμή. Με τον τρόπο αυτό η μόνιμη κατάσταση του συστήματος επαναπροσδιορίζεται διαρκώς, καθώς το ύψος της διαταραχής d_s αποκτά την πιο πρόσφατη τιμή.

Στη συνέχεια θα εστιάσουμε στη διαμόρφωση του προβλήματος βελτιστοποίησης που πρόκειται να επιλυθεί. Στο εξής θα θεωρήσουμε επίσης ότι ο ελεγκτής λειτουργεί με ορίζοντα ελέγχου που ταυτίζεται με τον ορίζοντα πρόβλεψης και θα τους συμβολίσουμε από κοινού με N . Το πρόβλημα βελτιστοποίησης έγκειται στην ελαχιστοποίηση μίας αντικειμενικής συνάρτησης που στην πλειοψηφία των περιπτώσεων είναι τετραγωνική.

Το κριτήριο που επιλέγεται εδώ είναι

$$J_N(\hat{\mathbf{u}}, \hat{\mathbf{x}}(t)) = \sum_{k=0}^{N-1} [\hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}_s]^\top \mathbf{R} [\hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}_s] + \sum_{k=1}^N [\hat{\mathbf{x}}(t+k) - \hat{\mathbf{x}}_s]^\top \mathbf{Q} [\hat{\mathbf{x}}(t+k) - \hat{\mathbf{x}}_s] \quad (4.15)$$

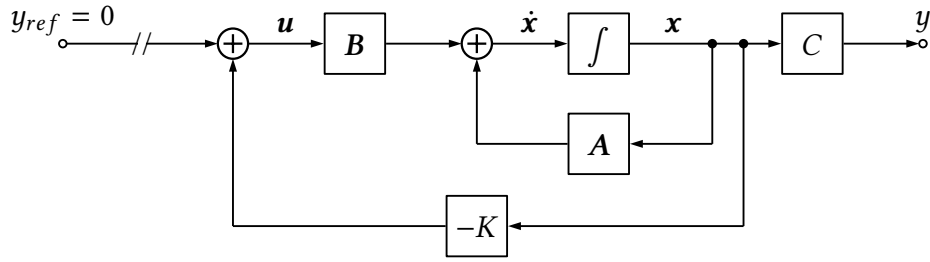
όπου το πρώτο άθροισμα τιμωρεί την απόκλιση της τιμής της εισόδου και το δεύτερο των μεταβλητών κατάστασης, σε κάθε χρονική στιγμή, από τις τιμές τους σε μόνιμες συνθήκες. Ο βαθμός συμμετοχής του κάθε αθροίσματος στην τελική τιμή της συνάρτησης κόστους καθορίζεται από τους πίνακες \mathbf{R} και \mathbf{Q} , η επιλογή των οποίων δίνει τη δυνατότητα στον σχεδιαστή να επηρεάσει τη συμπεριφορά του ελεγκτή^[15]. Γενικά όσο οι τιμές του πίνακα \mathbf{R} αυξάνουν σε σχέση με του \mathbf{Q} , ο ελεγκτής δίνει μεγαλύτερο βάρος στις τιμές εισόδου. Με τον τρόπο αυτό επιτυγχάνεται έλεγχος χαμηλού κόστους διότι ο ελεγκτής καταφεύγει σε πιο συντηρητικές κινήσεις. Στην αντίστροφη περίπτωση το βάρος δίνεται στις μεταβλητές κατάστασης. Τότε επιτυγχάνεται έλεγχος υψηλού κόστους και το σύστημα να χαρακτηρίζεται από ισχυρή απόσβεση. Πάντως είναι σύνηθες η παραπάνω επιλογή να απλοποιείται θέτοντας τον έναν πίνακα ίσο με τον μοναδιαίο και μεταβάλλοντας μόνο τις τιμές του δεύτερου

$$\mathbf{Q} = \mathbf{I}_{n_x} \quad \mathbf{R} = \rho \quad (4.16)$$

όπου n_x είναι ο αριθμός των μεταβλητών κατάστασης. Στη συνέχεια γίνεται η θεώρηση πως ο πίνακας \mathbf{Q} είναι θετικά ημι-ορισμένος και ο \mathbf{R} είναι θετικά ορισμένος. Αυτό βέβαια συνεπάγεται ότι η συνάρτηση κόστους μπορεί να λάβει μόνο θετικές τιμές ή να είναι το πολύ μηδέν (και δηλαδή ότι υπάρχει ολικό ελάχιστο).

4.2.2 Επέκταση σε άπειρο ορίζοντα

Η επέκταση του προβλήματος σε άπειρο ορίζοντα έχει τις ρίζες στην επιθυμία διασφάλισης ενός ευσταθούς ρυθμιστικού συστήματος. Καταρχάς θα στρέψουμε την προσοχή μας στον γραμμικό τετραγωνικό ρυθμιστή ή LQR (Linear Quadratic Controller) σε διακριτό χρόνο και από αυτόν θα μεταβούμε στην τελική υλοποίηση που μας αφορά. Αυτό αποτελεί ένα πρόβλημα βελτιστοποίησης με ανατροφοδότηση των μεταβλητών κατάστασης, υπό την έννοια πως η τιμή αυτών πρώτα ενισχύεται μέσα από έναν αναλογικό ενισχυτή



Σχήμα 4.3 Λογικό διάγραμμα ελέγχου ανατροφοδότησης μεταβλητών κατάστασης

και έπειτα οδηγείται ως είσοδος στο σύστημα. Συνεπώς αναζητούμε λύση στη μορφή

$$u(t) = -Kx(t) \quad (4.17)$$

Το λογικό διάγραμμα του τελικού συστήματος δίνεται στο Σχήμα 4.3. Το πρόβλημα επιλύεται σε άπειρο ορίζοντα και θεωρείται πως η τιμή της εισόδου και των μεταβλητών κατάστασης μηδενίζονται σε μόνιμες συνθήκες. Σε αναλογία με την Εξίσωση 4.15, η αντικειμενική συνάρτηση εδώ διατυπώνεται ως

$$J_{\infty}(\hat{x}(t)) = \sum_{k=0}^{\infty} \hat{u}(t+k)^{\top} R \hat{u}(t+k) + \hat{x}(t+k+1)^{\top} Q \hat{x}(t+k+1) \quad (4.18)$$

Το πρόβλημα επιδέχεται τη μοναδική λύση^[20]

$$K = (R + B^{\top} P B)^{-1} B^{\top} P A \quad (4.19)$$

όπου ο πίνακας P είναι θετικά ορισμένος και υπολογίζεται επιλύοντας την αλγεβρική εξίσωση Riccati σε διακριτό χρόνο

$$P = (A - BK)^{\top} P (A - BK) + K^{\top} R K + Q \quad (4.20)$$

οπότε η συνάρτηση κόστους της Εξίσωσης 4.18 αποκτά πλέον την τιμή

$$J_{\infty}(\hat{x}(t)) = \hat{x}(t)^{\top} P \hat{x}(t) \quad (4.21)$$

Το σύστημα που προκύπτει με τον τρόπο αυτό αποδεικνύεται ότι είναι ασυμπτωτικά ευσταθές υπό την προϋπόθεση ότι είναι σταθεροποιήσιμο^{1[20]}. Στη βιβλιογραφία έχει δειχθεί εκτενώς^[16] ότι το πρόβλημα πεπερασμένου ορίζοντα με συνάρτηση κόστους την

¹Εν συντομία, η έννοια της σταθεροποιησιμότητας έγκειται στη δυνατότητα ελέγχου όλων των ασταθών πόλων ενός συστήματος, πράγμα που υπονοεί ότι τυχόν μη ελέγξιμοι πόλοι είναι ευσταθείς.

Εξίσωση 4.15 μπορεί να διατυπωθεί με τη μορφή

$$\begin{aligned}
 J_N(\hat{\mathbf{u}}, \hat{\mathbf{x}}(t)) &= \sum_{k=0}^{N-1} [\hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}_s]^\top \mathbf{R} [\hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}_s] \\
 &+ \sum_{k=1}^{N-1} [\hat{\mathbf{x}}(t+k) - \hat{\mathbf{x}}_s]^\top \mathbf{Q} [\hat{\mathbf{x}}(t+k) - \hat{\mathbf{x}}_s] \\
 &+ [\hat{\mathbf{x}}(t+N) - \hat{\mathbf{x}}_s]^\top \mathbf{P} [\hat{\mathbf{x}}(t+N) - \hat{\mathbf{x}}_s]
 \end{aligned} \tag{4.22}$$

οπότε και διασφαλίζεται η ευστάθεια του. Συγκρίνοντας με την αρχική συνάρτηση είναι φανερό πως η αλλαγή αφορά μόνο στην χρήση διαφορετικής συνάρτησης κόστους για την τελική κατάσταση του συστήματος. Στην προκειμένη περίπτωση αυτή ταυτίζεται με τη συνάρτηση που δίνει την ελάχιστη τιμή κόστους για τον LQR ρυθμιστή και έτσι έχει τη μορφή της Εξίσωσης 4.21. Η προσθήκη τελικού κόστους έχει ουσιαστικά ως αποτέλεσμα την επέκταση του ορίζοντα επίλυσης του προβλήματος στο άπειρο.

4.2.3 Επιβολή φραγμών στο σύστημα

Στη συνήθη περίπτωση, ένα δυναμικό σύστημα περιγράφεται από ένα σύνολο φραγμών στις μεταβλητές εισόδου, εξόδου, ή και στις δύο. Αυτό μπορεί να οφείλεται στην ύπαρξη φυσικών περιορισμών στο σύστημα, αλλά και στην ανάγκη επιβολής ορίων ασφαλείας. Για ένα σύστημα διατυπωμένο στον χώρο των καταστάσεων, οι φραγμοί στις μεταβλητές εξόδου και εισόδου μπορούν να διατυπωθούν ως

$$\mathbf{y}_{min} \leq \mathbf{C}\hat{\mathbf{x}}(t) \leq \mathbf{y}_{max} \tag{4.23}$$

$$\mathbf{u}_{min} \leq \hat{\mathbf{u}}(t) \leq \mathbf{u}_{max} \tag{4.24}$$

οπότε ονομάζονται ισχυροί, υπό την έννοια πως είναι απαραβίαστοι. Παρότι κάτι τέτοιο δεν είναι προβληματικό για την μεταβλητή εισόδου, η οποία άλλωστε καθορίζεται από τον ελεγκτή και μόνο, εντούτοις το ίδιο δε μπορεί να βεβαιωθεί για την τιμή της εξόδου. Σε πολλές περιπτώσεις είναι αναπόφευκτη η παραβίαση των φραγμών που τίθενται στην έξοδο και αν αυτοί είναι ισχυροί, τότε το σύστημα προς επίλυση είναι αδύνατο.

Το ζήτημα αντιμετωπίζεται με τη χαλάρωσή τους, δηλαδή την επιβολή μαλακών φραγμών που επιτρέπεται να παραβιάζονται όταν κάτι τέτοιο δε μπορεί να αποφευχθεί. Η χαλάρωση των φραγμών μπορεί να επιτευχθεί με τη χρήση βοηθητικών μεταβλητών^[21],

οπότε η Εξίσωση 4.23 επαναδιατυπώνεται ως

$$\mathbf{y}_{min} - \delta \mathbf{y} \leq C \hat{\mathbf{x}}(t) \leq \mathbf{y}_{max} + \delta \mathbf{y} \quad (4.25)$$

$$\delta \mathbf{y} \geq \mathbf{0} \quad (4.26)$$

Παράλληλα η αντικειμενική συνάρτηση της Εξίσωσης 4.22 επεκτείνεται προκειμένου να τιμωρεί την παραβίαση των φραγμών

$$\begin{aligned} J_N(\hat{\mathbf{u}}, \hat{\mathbf{x}}(t)) = & \sum_{k=0}^{N-1} \left([\hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}_s]^T \mathbf{R} [\hat{\mathbf{u}}(t+k) - \hat{\mathbf{u}}_s] + \right. \\ & \left. [\delta \mathbf{y}(t+k+1)]^T \mathbf{S} [\delta \mathbf{y}(t+k+1)] \right) \\ & + \sum_{k=1}^{N-1} [\hat{\mathbf{x}}(t+k) - \hat{\mathbf{x}}_s]^T \mathbf{Q} [\hat{\mathbf{x}}(t+k) - \hat{\mathbf{x}}_s] \\ & + [\hat{\mathbf{x}}(t+N) - \hat{\mathbf{x}}_s]^T \mathbf{P} [\hat{\mathbf{x}}(t+N) - \hat{\mathbf{x}}_s] \end{aligned} \quad (4.27)$$

Η παρουσία του πίνακα \mathbf{S} προσθέτει φυσικά επιπλέον βαθμούς ελευθερίας στο πρόβλημα. Εν γένει, καθώς οι τιμές του πίνακα \mathbf{S} αυξάνουν, τόσο πιο ισχυροί γίνονται οι φραγμοί. Μικρές τιμές τυπικά οδηγούν στην παραβίασή τους ακόμα και σε περιπτώσεις που κάτι τέτοιο δεν είναι απαραίτητο. Καθώς οι τιμές αυξάνουν, τελικά προσεγγίζουν ένα όριο βέλτιστης συμπεριφοράς. Εξαιρετικά υψηλές τιμές μπορεί να επηρεάσουν πάντως αρνητικά την υπολογιστική επίλυση του προβλήματος.

Κλείνοντας, πρέπει να τονιστεί πως κανείς πρέπει να είναι ιδιαίτερα προσεκτικός όταν επιβάλει φραγμούς ενώ παράλληλα κάνει χρήση τελικού κόστους στην αντικειμενική συνάρτηση^[17,20]. Όσα παρουσιάστηκαν στην Υποενότητα 4.2.2 για τον LQR ρυθμιστή αφορούν σύστημα χωρίς φραγμούς. Η ευστάθεια του συστήματος ελέγχου που προκύπτει από την προσθήκη τελικού κόστους διασφαλίζεται συνεπώς μόνο υπό την ισχύ της προϋπόθεσης αυτής. Το ζήτημα μπορεί να αντιμετωπιστεί βεβαιώνοντας ότι στο όριο του πεπερασμένου ορίζοντα κανένας φραγμός δεν είναι ενεργός. Αυτό μπορεί να εξασφαλιστεί επιλέγοντας ορίζοντα επαρκώς μεγάλου μήκους.

4.3 Μετατροπή σε πρόβλημα τετραγωνικού προγραμματισμού

Το πρόβλημα βελτιστοποίησης που περιγράφεται παραπάνω μπορεί εύκολα να επαναδιατυπωθεί σε μορφή τετραγωνικού προγραμματισμού

$$\min_{\bar{\mathbf{x}}} \frac{1}{2} \bar{\mathbf{x}}^\top \mathbf{H} \bar{\mathbf{x}} + \mathbf{f}^\top \bar{\mathbf{x}} \quad (4.28)$$

$$\text{υπόκειται σε: } \mathbf{A}_{eq} \cdot \bar{\mathbf{x}} = \mathbf{b}_{eq} \quad (4.29)$$

$$\mathbf{A} \cdot \bar{\mathbf{x}} \leq \mathbf{b} \quad (4.30)$$

προκειμένου να διευκολυνθεί η υπολογιστική επίλυσή του. Το διάνυσμα $\bar{\mathbf{x}}$ ορίζεται έτσι, ώστε να περιλαμβάνει τις μεταβλητές κατάστασης, εισόδου και τις βοηθητικές

$$\bar{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}^\top(t+1) & \cdots & \hat{\mathbf{x}}^\top(t+N) \\ \mathbf{u}^\top(t) & \cdots & \mathbf{u}^\top(t+N-1) \\ \delta \mathbf{y}^\top(t+1) & \cdots & \delta \mathbf{y}^\top(t+N) \end{bmatrix}^\top \quad (4.31)$$

Πριν προχωρήσουμε παρακάτω, είναι χρήσιμο να οριστεί επίσης το γινόμενο Kronecker

$$\mathbf{A}_{m \times n} \otimes \mathbf{B} = \begin{bmatrix} a_{11} \mathbf{B} & \cdots & a_{1n} \mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1} \mathbf{B} & \cdots & a_{mn} \mathbf{B} \end{bmatrix} \quad (4.32)$$

καθώς και ο $m \times n$ πίνακας $\mathbf{1}_{m,n}$, ο οποίος συμβολίζεται σε αντιστοιχία με τον μηδενικό και έχει όλα τα στοιχεία του ίσα με τη μονάδα. Τα παραπάνω θα μας επιτρέψουν στη συνέχεια να διατυπώσουμε τους πίνακες σε μία πιο συμπαγή μορφή.

Πλέον μπορούμε να προχωρήσουμε, ξεκινώντας από τους πίνακες \mathbf{H} και \mathbf{f} της Εξίσωσης 4.28. Αυτοί αντιστοιχούν στην τελική συνάρτηση κόστους που περιγράφεται από την Εξίσωση 4.27 και ορίζονται ως

$$\mathbf{H} = 2 \begin{bmatrix} \mathbf{I}_{N-1} \otimes \mathbf{Q} & & & \\ & \mathbf{P} & & \\ & & \mathbf{I}_N \otimes \mathbf{R} & \\ & & & \mathbf{I}_N \otimes \mathbf{S} \end{bmatrix} \quad \mathbf{f} = -2 \begin{bmatrix} \mathbf{1}_{N,1} \otimes \mathbf{x}_s \\ \mathbf{1}_{N,1} \otimes \mathbf{u}_s \\ \mathbf{0}_{N \cdot n_x, 1} \end{bmatrix} \quad (4.33)$$

Εδώ θα πρέπει κανείς να παρατηρήσει ότι οι Εξισώσεις 4.27 και 4.28 διαφέρουν κατά τον σταθερό όρο $Nru_s^2 + Nx_s^T Qx_s$. Φυσικά αυτό μεταβάλλει μεν την απόλυτη τιμή της συνάρτησης, όμως δεν επηρεάζει το σχήμα της ή τη θέση του ολικού ελαχίστου και άρα αφήνει ανέπαφη τη διαδικασία ελαχιστοποίησης που εν προκειμένω είναι το ζητούμενο.

Οι πίνακες A_{eq} και b_{eq} της Εξίσωσης 4.29 ορίζονται στη βάση της επαναληπτικής εφαρμογής της Εξίσωσης 4.5, από την οποία προκύπτουν οι εξισώσεις

$$\begin{aligned}\hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + B_d d(t) \\ \hat{x}(t+2) &= A\hat{x}(t+1) + Bu(t+1) + B_d d(t) \\ &\vdots \\ \hat{x}(t+N) &= A\hat{x}(t+N-1) + Bu(t+N-1) + B_d d(t)\end{aligned}\quad (4.34)$$

και συνεπώς το σύστημα προς επίλυση γίνεται

$$A_{eq} = \begin{bmatrix} I_{n_x} & & & & -B \\ -A & I_{n_x} & & & -B \\ & \ddots & \ddots & & \vdots \\ & & & -A & I_{n_x} \\ & & & & -B \end{bmatrix} \begin{bmatrix} \hat{x}(t)A \\ -B_d d_s \\ \vdots \\ -B_d d_s \end{bmatrix} \quad (4.35)$$

Τέλος οι φραγμοί των Εξισώσεων 4.25 και 4.26 επιβάλλονται από την Εξίσωση 4.30 και αποτυπώνονται στους πίνακες A και b που αποκτούν τη μορφή

$$A = \begin{bmatrix} -I_N \otimes C & & -I_N \\ I_N \otimes C & & -I_N \\ & -I_N & \\ & I_N & \\ & & -I_N \end{bmatrix} \quad b = \begin{bmatrix} -\mathbf{1}_{N,1} \cdot y_{min} \\ \mathbf{1}_{N,1} \cdot y_{max} \\ -\mathbf{1}_{N,1} \cdot u_{min} \\ \mathbf{1}_{N,1} \cdot u_{max} \\ \mathbf{0}_{N,1} \end{bmatrix} \quad (4.36)$$

Αφού το πρόβλημα διατυπωθεί σε μορφή τετραγωνικού προγραμματισμού, είναι δυνατό να επιλυθεί με τη χρήση ήδη υπάρχοντος λογισμικού. Στα πλαίσια της παρούσας εργασίας χρησιμοποιείται το υπολογιστικό πακέτο MATLAB με την εντολή quadprog.

Κεφάλαιο 5

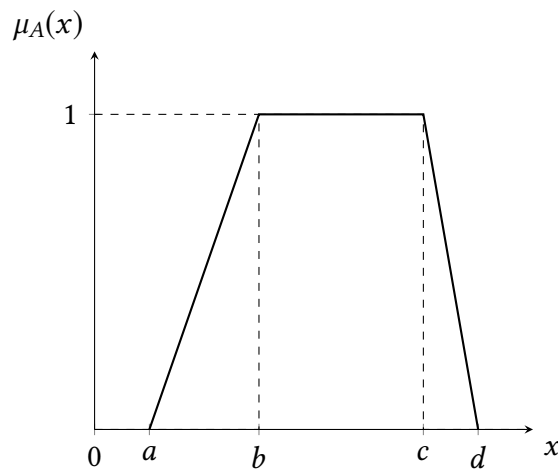
Συστήματα ασαφούς λογικής

Το κεφάλαιο αυτό αποσκοπεί στην εισαγωγή του αναγνώστη στην έννοια των συστημάτων ασαφούς λογικής. Οι βασικές έννοιες πίσω από τον τρόπο λειτουργίας των συστημάτων αυτών συνοψίζονται στην Ενότητα 5.1. Στην Ενότητα 5.2 διατυπώνεται ένα μοντέλο με δομή Takagi–Sugeno και αμέσως μετά, στην Ενότητα 5.3, αυτό εφαρμόζεται στη μοντελοποίηση μη γραμμικών συστημάτων. Η τελευταία ενότητα είναι ιδιαίτερα σημαντική υπό την άποψη ότι επεκτείνει το γραμμικό ARX μοντέλο του Κεφαλαίου 3, αλλά στη βάση πως αυτό πρόκειται να χρησιμοποιηθεί στον MPC ελεγκτή που περιγράφεται στο Κεφάλαιο 4.

5.1 Εισαγωγικές έννοιες

Ως άνθρωποι έχουμε την ικανότητα να συνεννοούμαστε στην καθημερινή μας ζωή χωρίς τη χρήση μετρήσεων και υπολογισμών. Για παράδειγμα κανείς μπορεί να περιγράψει τον καιρό ως ‘πολύ ζεστό’, ‘ζεστό’, ‘κρύο’, ή ‘πολύ κρύο’. Πρόκειται για αφηρημένους γλωσσικούς όρους, που όμως αρκούν για να αποκτήσουμε μία αίσθηση της θερμοκρασίας του περιβάλλοντος χωρίς να έχει προηγηθεί οποιαδήποτε αναφορά στην ακριβή τιμή της. Τα συστήματα ασαφούς λογικής επιχειρούν να γεφυρώσουν αυτό ακριβώς το χάσμα μεταξύ των ασαφών εννοιών που χαρακτηρίζουν τον ανθρώπινο τρόπο σκέψης και της σαφήνειας με την οποία είναι συνυφασμένη μία ακριβής μέτρηση. Θα μπορούσε κανείς να πει ότι αποτελούν μία επέκταση της δυαδικής λογικής, κατά την οποία κάτι αν δεν αληθεύει, είναι ψευδές. Με τον ασαφή λογισμό κάτι μπορεί να αληθεύει σε έναν βαθμό, οπότε τοποθετείται κάπου ενδιάμεσα στο φάσμα που έχει ως ακραίες έννοιες το πλήρως αληθές και πλήρως ψευδές^[22].

Για να αποσαφηνιστούν τα παραπάνω θα πρέπει καταρχάς να αναφερθούμε στην έννοια της συνάρτησης συμμετοχής και του ασαφούς συνόλου^[22,23]. Στην κλασική προσέγγιση, ένα σύνολο \mathbb{A} μπορεί να περιγραφεί από τη συνάρτηση συμμετοχής $\chi_{\mathbb{A}}$, που



Σχήμα 5.1 Τραπεζοειδής συνάρτηση συμμετοχής

για κάθε στοιχείο $x \in \mathbb{A}$ ορίζεται ως

$$\chi_{\mathbb{A}}(x) = \begin{cases} 1, & x \in \mathbb{A} \\ 0, & x \notin \mathbb{A} \end{cases} \quad (5.1)$$

Το γεγονός ότι η συνάρτηση μπορεί να πάρει μόνο δυαδικές τιμές υποδηλώνει ότι ένα στοιχείο είτε ανήκει στο σύνολο, είτε όχι. Αντιθέτως στην περίπτωση της ασαφούς λογικής, ο ορισμός της συνάρτησης συμμετοχής, που για ένα σύνολο \mathbb{A} θα συμβολίζουμε ως $\mu_{\mathbb{A}}$, επεκτείνεται ώστε αυτή να αποτελεί συνεχή συνάρτηση στο εύρος $[0, 1]$. Στη συνήθη περίπτωση η συνάρτηση αυτή ορίζεται σαν τριγωνική, τραπεζοειδής, ή γκαουσιανή. Στη συνέχεια θα μας απασχολήσει αποκλειστικά η περίπτωση της τραπεζοειδούς συνάρτησης, για την οποία είναι

$$\mu_{\mathbb{A}}(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & x > d \end{cases} \quad (5.2)$$

και διαγραμματικά απεικονίζεται στο Σχήμα 5.1. Ιδιαίτερη σημασία όμως έχει ο ρόλος που διαδραματίζουν οι συναρτήσεις συμμετοχής (και το αντίστοιχα ασαφή σύνολα που ορίζουν). Η κάθε συνάρτηση ορίζει τον βαθμό στον οποίο το κάθε στοιχείο ανήκει στο σύνολο που αυτή περιγράφει. Κατά τον ίδιο τρόπο στο παράδειγμα στην αρχή της ενότητας, ο άνθρωπος αποδίδει στην τιμή της θερμοκρασίας έναν βαθμό συμμετοχής

στην αφηρημένη έννοια του 'ζεστού' (όπου κάτι λίγο και καθόλου ζεστό μπορεί να θεωρηθεί 'κρύο' και 'πολύ κρύο' αντίστοιχα). Αυτή η ομοιότητα είναι κείριας σημασίας στον ασαφή λογισμό, καθώς επιτρέπει την περιγραφή ενός αφηρημένου γλωσσικού όρου αντιστοιχίζοντάς τον στην κατάλληλη συνάρτηση συμμετοχής. Τότε, φυσικά, ο βαθμός στον οποίο συνάδει μία μέτρηση ως προς την έννοια που περιγράφεται από μία οποιαδήποτε συνάρτηση συμμετοχής αποτυπώνεται με έναν αριθμό στο διάστημα $[0, 1]$.

Γενικά όλες οι πράξεις που ορίζονται για την κλασική θεώρηση των συνόλων επεκτείνονται και στην περίπτωση των ασαφών^[22]. Αυτό μεν ξεφεύγει από τη σκοπιά αυτής της εργασίας, όμως εδώ θα αναφερθούμε σε δύο από αυτές, οι οποίες βέβαια θα μας απασχολήσουν και παρακάτω. Συγκεκριμένα για τη λογική πράξη ΚΑΙ μεταξύ δύο ασαφών συνόλων \mathbb{A} και \mathbb{B} είναι

$$\begin{aligned} x \in \mathbb{A} \text{ ΚΑΙ } x \in \mathbb{B} = \\ \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \end{aligned} \quad (5.3)$$

ενώ η λογική πράξη Ή ορίζεται ως

$$\begin{aligned} x \in \mathbb{A} \text{ Ή } x \in \mathbb{B} = \\ \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)) \end{aligned} \quad (5.4)$$

Οι πράξεις αυτές θα φανούν χρήσιμες στη συνέχεια, όπου παρουσιάζεται η συλλογιστική μέθοδος Takagi- Sugeno. Αυτή μαζί με τη μέθοδο Mandani αποτελούν τους δύο τύπους ασαφών μοντέλων που συναντώνται κατά κόρον στη βιβλιογραφία. Εδώ θα εστιάσουμε αποκλειστικά στον πρώτο, πράγμα που οφείλεται στον τρόπο διατύπωσής του που προσφέρει σημαντικό πλεονέκτημα όταν πρόκειται για εφαρμογές αυτομάτου ελέγχου στη μορφή που παρουσιάστηκε στο Κεφάλαιο 4.

5.2 Μοντέλο Takagi- Sugeno

Γενικά τα μοντέλα ασαφούς λογικής αποτελούνται από μία σειρά κανόνων που συντίθενται από ένα τμήμα υπόθεσης και ένα τμήμα απόφασης^[22,23]

$$\underline{R^i} : \underline{\text{AN}} \quad z_1 \text{ είναι } \mathbb{F}_1^i \text{ ΚΑΙ } z_2 \text{ είναι } \mathbb{F}_2^i \dots \text{ ΚΑΙ } z_j \text{ είναι } \mathbb{F}_j^i \quad (5.5)$$

$$\underline{\text{TOTE}} \quad y^i = c_1^i x_1 + c_2^i x_2 + \dots + c_k^i x_k \quad (5.6)$$

τα οποία δίνονται από τις Εξισώσεις 5.5 και 5.6 αντίστοιχα. Ο κανόνας i συμβολίζεται ως R^i και έχει μεταβλητές εισόδου z_1, \dots, z_j και μεταβλητή εξόδου y^i . Στο μοντέλο

τύπου Takagi- Sugeno^[24] αυτή υπολογίζεται ως γραμμικός συνδυασμός των μεταβλητών x_1, \dots, x_k μέσα από τους συντελεστές c_1^i, \dots, c_k^i . Οι μεταβλητές που υπεισέρχονται στον υπολογισμό της εξόδου μπορεί να ταυτίζονται με τις μεταβλητές εισόδου, αλλά κάτι τέτοιο δεν είναι απαραίτητο. Εδώ είναι σημαντικό να προσέξει κανείς την εξάρτηση των συντελεστών c^i από τον εκάστοτε κανόνα. Αυτό σημαίνει πως κάθε κανόνας ορίζει ένα διαφορετικό γραμμικό μοντέλο για τον υπολογισμό της τιμής εξόδου. Αυτή είναι μία πολύ σημαντική παρατήρηση στην οποία θα επιστρέψουμε στην επόμενη ενότητα.

Από τα παραπάνω γίνεται φανερό πως έπειτα από την εκτέλεση των πράξεων σε ένα μοντέλο r κανόνων R^1, \dots, R^r , υπολογίζονται ισάριθμες μεταβλητές εξόδου y^1, \dots, y^r . Είναι λοιπόν σαφές ότι υπολείπεται ένα επιπλέον βήμα μέσα από το οποίο θα καθορίζεται μία τελική, ακριβής τιμή εξόδου. Αυτόν τον ρόλο επιτελεί η συνάρτηση αποσαφήνισης. Από την Εξίσωση 5.5 ορίζεται το ύψος πυροδότησης του κάθε κανόνα

$$f^i = \mu_{F_1^i}(z_1) \wedge \mu_{F_2^i}(z_2) \cdots \wedge \mu_{F_j^i}(z_j) \quad (5.7)$$

και υπολογίζεται το κανονικοποιημένο ύψος

$$\beta^i = \frac{f^i}{\sum_{j=1}^r f^j} \quad (5.8)$$

Τελικά η συνάρτηση αποσαφήνισης έγκειται στον υπολογισμό ενός σταθμισμένου μέσου όρου των τιμών εξόδου με βάρη τα κανονικοποιημένα αυτά ύψη πυροδότησης

$$y_c = \sum_{i=1}^r \beta^i y^i \quad (5.9)$$

Εδώ θα πρέπει να σημειωθεί πως η υπόθεση δε διατυπώνεται απαραίτητα στη μορφή της Εξίσωσης 5.5. Παρότι στη συνήθη περίπτωση οι μεταβλητές εισόδου συνδέονται με τη λογική πράξη της σύζευξης, κανείς δεν αποτρέπει τη χρήση διάζευξης ή οποιοδήποτε άλλου συνδυασμού λογικών πράξεων. Βεβαίως σε κάθε περίπτωση το ύψος πυροδότησης των κανόνων υπολογίζεται αναλόγως. Για παράδειγμα, για μία υπόθεση στη μορφή

$$z_1 \text{ είναι } F_1 \text{ ΚΑΙ } z_2 \text{ είναι } F_2 \text{ Ή } z_3 \text{ είναι } F_3 \quad (5.10)$$

το απόλυτο ύψος πυροδότησης υπολογίζεται ως

$$f^i = \mu_{F_1}(z_1) \wedge \mu_{F_2}(z_2) \vee \mu_{F_3}(z_3) \quad (5.11)$$

Κλείνοντας, είναι χρήσιμο να γίνει μία σύντομη αναφορά στη διαδικασία εξαγωγής ενός βέλτιστου μοντέλου Takagi- Sugeno^[22,24,25]. Πρόκειται για ένα σύνθετο πρόβλημα βελτιστοποίησης, γεγονός που αποδίδεται στους πολλούς βαθμούς ελευθερίας του μοντέλου. Πιο συγκεκριμένα, ο σχεδιαστής καλείται να καθορίσει τις μεταβλητές εισόδου, τις συναρτήσεις συμμετοχής, τον αριθμό των κανόνων και τη μορφή του τμήματος υπόθεσης ανά κανόνα, καθώς και τη δομή και τις παραμέτρους του τμήματος απόφασης. Λόγω της φιλοσοφίας της μεθόδου, συχνά ο καθορισμός των μοντέλων βασίζεται τουλάχιστον σε έναν βαθμό- σε πρότερη γνώση που υπάρχει για τη διεργασία, παρά σε κάποιο υπολογιστικό σχήμα βελτιστοποίησης. Σε κάθε περίπτωση, ο καθορισμός του τμήματος απόφασης μπορεί να βασιστεί στην επέκταση της μεθόδου των ελαχίστων τετραγώνων της Ενότητας 3.2.3. Η επιλογή των μεταβλητών εισόδου αποτελεί πρόβλημα συνδυαστικής βελτιστοποίησης και ως τέτοιο μπορεί να αντιμετωπιστεί με την εφαρμογή κάποιου ευρετικού αλγορίθμου. Για τον προσδιορισμό του αριθμού και της μορφής των υποθέσεων στους κανόνες χρησιμοποιούνται αλγόριθμοι ασαφούς συσταδοποίησης σε διαθέσιμα σύνολα πειραματικών δεδομένων. Πάντως τα προηγούμενα βήματα είναι αλληλένδετα και έτσι για τον προσδιορισμό του βέλτιστου μοντέλου απαιτείται σε κάθε περίπτωση χρήση κάποιου επαναληπτικού σχήματος.

5.3 Μοντελοποίηση μη γραμμικών συστημάτων

5.3.1 Ψευδογραμμικό ARX μοντέλο

Όπως είδαμε παραπάνω, το τμήμα απόφασης του μοντέλου Takagi- Sugeno ορίζει μία γραμμική εξίσωση για τον υπολογισμό της τιμής εξόδου. Αυτό σημαίνει πως μπορεί άμεσα να τροποποιηθεί σε γραμμικό ARX μοντέλο, διατυπωμένο στον χώρο των καταστάσεων σύμφωνα με τις Εξισώσεις 3.26 και 3.27. Ο κανόνας της Εξίσωσης 5.5 γίνεται έτσι

$$R^i : \underline{AN} \quad z_1 \text{ είναι } \mathbb{F}_1^i \text{ ΚΑΙ } z_2 \text{ είναι } \mathbb{F}_2^i \dots \text{ ΚΑΙ } z_j \text{ είναι } \mathbb{F}_j^i \quad (5.12)$$

$$\underline{TOTE} \quad \begin{cases} \hat{\mathbf{x}}(t+1) = \mathbf{A}^i \hat{\mathbf{x}}(t) + \mathbf{B}^i u(t) \\ \hat{\mathbf{y}}(t) = \mathbf{C}^i \hat{\mathbf{x}}(t) \end{cases} \quad (5.13)$$

όπου οι πίνακες \mathbf{A}^i , \mathbf{B}^i , και \mathbf{C}^i ορίζονται βάσει της Εξίσωσης 3.29. Η διατύπωση αυτή προσφέρει σημαντικά πλεονεκτήματα. Ένα μη γραμμικό σύστημα είναι δυνατό να προσεγγιστεί τοπικά από ένα γραμμικό μοντέλο. Η συλλογιστική Takagi- Sugeno στη μορφή που δίνεται παραπάνω αναλαμβάνει τον ρόλο του επόπτη, ο οποίος καλείται να συνδυάσει με κατάλληλο τρόπο τα επιμέρους ARX μοντέλα, προκειμένου να παρέχει μία

προσέγγιση της μη γραμμικής συμπεριφοράς του συστήματος σε όλο το φάσμα των συνθηκών λειτουργίας που μας αφορούν^[23,24]. Το σημαντικότερο σημείο της προσέγγισης αυτής, έγκειται στο γεγονός πως το τελικό μοντέλο συνεχίζει να διατυπώνεται στη μορφή του γραμμικού ARX μοντέλου, οπότε και καλείται ψευδογραμμικό. Αυτό μπορεί να γίνει εναλλακτικά αντιληπτό ως ένα γραμμικό μοντέλο με μεταβλητές που μεταβάλλονται στον χρόνο. Το πλεονέκτημα έγκειται τελικά στο γεγονός πως όσα διατυπώθηκαν για τον MPC ελεγκτή στο Κεφάλαιο 4 συνεχίζουν να ισχύουν ως έχουν, χωρίς την ανάγκη να υπάρξει οποιαδήποτε παρέμβαση.

5.3.2 Επέκταση της μεθόδου ελαχίστων τετραγώνων

Ο υπολογισμός των παραμέτρων των ARX μοντέλων στο τμήμα απόφασης ενός μοντέλου Takagi- Sugeno, μπορεί να βασιστεί σε μία επέκταση του αλγορίθμου ελαχίστων τετραγώνων που παρουσιάστηκε στην Ενότητα 3.2.3^[24]. Η τιμή εξόδου y^i του κάθε επιμέρους μοντέλου i συνεχίζει να υπολογίζεται από την Εξίσωση 3.9

$$\hat{y}^i(t|\theta) = \phi^T(t)\theta^i \quad (5.14)$$

αλλά πλέον συμβάλει στην τελική τιμή εξόδου y_c με βάση την Εξίσωση 5.9

$$\hat{y}_c = \sum_{i=1}^r \beta^i \hat{y}^i \quad (5.15)$$

Στη συνέχεια, το κανονικοποιημένο ύψος πυροδότησης β^i για κάθε κανόνα i θα θεωρηθεί γνωστό. Άλλωστε ο υπολογισμός του σε κάθε χρονική στιγμή είναι δυνατός με την άμεση εφαρμογή των Εξισώσεων 5.8 και 5.7 στο διαθέσιμο σύνολο πειραματικών δεδομένων Z^N της Εξίσωσης 3.17. Είναι τώρα επιθυμητό να οριστούν τα διανύσματα ϕ και θ με τρόπο κατάλληλο, ώστε η Εξίσωση 3.22

$$\hat{\theta}_N = \left[\sum_{t=1}^N \phi(t)\phi^T(t) \right]^{-1} \sum_{t=1}^N \phi(t)y(t) \quad (5.16)$$

να μπορεί να εφαρμοστεί αυτούσια για τον ταυτόχρονο υπολογισμό όλων των τοπικών μοντέλων. Στην περίπτωση που υπάρχουν r κανόνες, τότε απαιτείται ο προσδιορισμός ίσων σε αριθμό μοντέλων. Για τον λόγο αυτό το διάνυσμα θ επεκτείνεται ώστε να

περιλαμβάνει το σύνολο των παραμέτρων προς προσδιορισμό

$$\theta = \begin{bmatrix} a_1^1 & \cdots & a_{n_a}^1 & | & b_1^1 & \cdots & b_{n_b}^1 \\ & & & & a_1^2 & \cdots & a_{n_a}^2 & | & b_1^2 & \cdots & b_{n_b}^2 \\ & & & & & & & & & & & \dots \\ & & & & & & & & a_1^r & \cdots & a_{n_a}^r & | & b_1^r & \cdots & b_{n_b}^r \end{bmatrix}^T \quad (5.17)$$

όπου το κάθε τοπικό μοντέλο μπορεί να περιγράφεται από διαφορετικό αριθμό παραμέτρων. Τώρα αρκεί να επεκταθεί με παρόμοιο τρόπο και το διάνυσμα ϕ . Για τη σωστή εφαρμογή της μεθόδου, θα πρέπει τα πειραματικά δεδομένα που αντιστοιχούν στο μοντέλο i να ζυγιστούν με βάρος β^i . Τότε μπορεί να φανεί εύκολα πως σε κάθε στιγμή η τελική τιμή εξόδου πράγματι δίνεται από την Εξίσωση 5.15. Έτσι ορίζοντας

$$\theta = \begin{bmatrix} -\beta^1 y(t-1) & \cdots & -\beta^1 y(t-n_a^1) & | & \beta^1 u(t-1) & \cdots & \beta^1 u(t-n_b^1) \\ & & -\beta^2 y(t-1) & \cdots & -\beta^2 y(t-n_a^2) & | & \beta^2 u(t-1) & \cdots & \beta^2 u(t-n_b^2) \\ & & & & & & & & & & & \dots \\ & & & & & & & & -\beta^r y(t-1) & \cdots & -\beta^r y(t-n_a^r) & | & \beta^r u(t-1) & \cdots & \beta^r u(t-n_b^r) \end{bmatrix}^T \quad (5.18)$$

ολοκληρώνεται η επέκταση της μεθόδου. Φυσικά ως τώρα έχει σιωπηλά υποτεθεί πλήρης γνώση του αριθμού και του τμήματος υπόθεσης των κανόνων. Επίσης παραμένει το πρόβλημα της επιλογής του αριθμού των παραμέτρων των επιμέρους μοντέλων. Παρακάτω προτείνεται μία λύση για την αντιμετώπιση των ζητημάτων αυτών.

5.3.3 Αλγόριθμοι ασαφούς συσταδοποίησης

Ασαφείς c-μέσοι

Η συσταδοποίηση αποτελεί διαδικασία για την κατάταξη δεδομένων σε ομάδες, αποσκοπώντας στη μεγιστοποίηση της ομοιότητας μεταξύ των μελών της ίδιας ομάδας και στην ελαχιστοποίηση της ομοιότητας ανάμεσα στις διαφορετικές ομάδες^[26]. Οι αλγόριθμοι ασαφούς συσταδοποίησης, όπως κανείς θα περίμενε, επιτρέπουν την κατάταξη των δεδο-

μένων σε περισσότερες της μίας συστάδας. Υπό αυτή την έννοια κάθε συστάδα αποτελεί ένα ασαφές σύνολο και το αποτέλεσμα της διαδικασίας της ασαφούς συσταδοποίησης δεν είναι άλλο από τον προσδιορισμό της συνάρτησης συμμετοχής που την περιγράφει.

Αν έχουμε στη διάθεσή μας N διανύσματα

$$\mathbf{x}(k), k = 1, \dots, N \quad (5.19)$$

οι ασαφείς c -μέσοι μπορούν να τα κατατάξουν σε C συστάδες με κέντρα

$$\mathbf{c}(i), i = 1, \dots, C \quad (5.20)$$

ελαχιστοποιώντας την αντικειμενική συνάρτηση

$$V_m = \sum_{k=1}^N \sum_{i=1}^C \mu_{ik}^m \|\mathbf{x}(k) - \mathbf{c}(i)\|^2 \quad (5.21)$$

όπου μ_{ik} είναι ο βαθμός συμμετοχής του διανύσματος $\mathbf{x}(k)$ στη συστάδα i . Ο εκθέτης m καθορίζει τον βαθμό αλληλεπικάλυψης μεταξύ των συστάδων. Συνήθως επιλέγεται

$$m = 2 \quad (5.22)$$

κάτι το οποίο συμβαίνει και εδώ. Η κατάταξη των διανυσμάτων γίνεται στη βάση του μέτρου απόστασης $\|\cdot\|$. Στην τυπική περίπτωση αυτό δηλώνει ευκλείδεια απόσταση^[26].

Συσταδοποίηση διανυσματικών χρονοσειρών

Μία ενδιαφέρουσα προσέγγιση δίνεται από τον Liao^[27] για τη συσταδοποίηση χρονοσειρών. Το αποτέλεσμα είναι η κατάταξη της κάθε χρονοσειράς σε ένα ή περισσότερα ασαφή σύνολα. Η χρησιμότητα του αλγορίθμου στην περίπτωσή μας θα αναδειχθεί παρακάτω στην εργασία. Η μέθοδος λειτουργεί σε δύο στάδια.

Ξεκινώντας από N p -διάστατες χρονοσειρές

$$\mathbf{x}^i(1), \dots, \mathbf{x}^i(t-1), \mathbf{x}^i(t), \mathbf{x}^i(t+1), \dots, \mathbf{x}^i(n_i), \quad i = 1, \dots, N, \quad \mathbf{x} \in \mathbb{R}^p \quad (5.23)$$

Εφαρμόζεται καταρχάς ο αλγόριθμος των ασαφών c -μέσων που συζητήθηκε προηγουμένως. Αυτό συμβαίνει σε επίπεδο διανυσμάτων, οπότε τα δεδομένα λαμβάνονται υπόψη ανεξαρτήτως χρόνου. Στην ουσία ζητούμε την κατάταξη των $N \times \sum_{i=1}^N n_i$ διανυσμάτων που έχουμε στη διάθεσή μας σε $N_{c,p}$ συστάδες. Στη συνέχεια το κάθε διάνυσμα \mathbf{x} αντικαθίσταται από μία διακριτή τιμή. Αυτή αντιστοιχεί στον αύξοντα αριθμό της συστάδας

στην οποία το διάνυσμα ανήκει με τον υψηλότερο βαθμό συμμετοχής. Το κάθε διάνυσμα συνεπώς αντιστοιχίζεται σε μία από τις $N_{c,p}$ πιθανές διακριτές τιμές και οι χρονοσειρές της Εξίσωσης 5.23 γράφονται τελικά στη νέα μορφή

$$y^i(1), \dots, y^i(t-1), y^i(t), y^i(t+1), \dots, y^i(n_i), \quad i = 1, \dots, N, \quad y \in \mathbb{R} \quad (5.24)$$

που είναι σημαντικά απλοποιημένη. Τα δεδομένα όχι μόνο μετατρέπονται σε μονοδιάστατα, αλλά σε κάθε χρονική στιγμή το σύστημα μπορεί μόνο να βρίσκεται σε μία από τις $N_{c,p}$ καταστάσεις που προσδιορίζονται.

Στο δεύτερο στάδιο κάθε μία από τις σειρές μετατρέπεται σε πίνακα πιθανότητας μετάβασης. Ο πίνακας έχει διαστάσεις $N_{c,p} \times N_{c,p}$ και για το κάθε στοιχείο (k, l) είναι

$$p_{kl}^i = \frac{n(S_k \rightarrow S_l)}{n_i - 1} \quad (5.25)$$

όπου το n δηλώνει τον αριθμό των φορών που δύο γειτονικές τιμές στη σειρά αλλάζουν από κατάσταση S_i σε S_j . Πλέον μπορεί να εφαρμοστεί και πάλι ο αλγόριθμος των ασαφών c -μέσων, όμως με τροποποιημένο μέτρο απόστασης. Η απόσταση Kullback–Liebler ποσοτικοποιεί τον βαθμό στον οποίο δύο κατανομές διαφέρουν. Το μέτρο ορίζεται ως

$$d(p_k^1, p_k^2) = \sum_{l=1}^{N_{c,p}} p_{kl}^1 \log \left(\frac{p_{kl}^1}{p_{kl}^2} \right) \quad (5.26)$$

μεταξύ δύο σειρών που περιγράφονται από τους πίνακες πιθανότητας μετάβασης P^1 και P^2 . Το μέτρο δίνεται στη συμμετρική του μορφή

$$D(p_k^1, p_k^2) = \frac{d(p_k^1, p_k^2) + d(p_k^2, p_k^1)}{2} \quad (5.27)$$

και τελικά η μέση απόσταση μεταξύ των σειρών είναι

$$D(P^1, P^2) = \sum_{k=1}^{N_{c,p}} \frac{D(p_k^1, p_k^2)}{N_{c,p}} \quad (5.28)$$

Με τον τρόπο αυτό η συσταδοποίηση λαμβάνει χώρα στο επίπεδο των χρονοσειρών αυτούσιων και όχι των ανεξάρτητων πειραματικών δεδομένων. Το τελικό αποτέλεσμα είναι $N_{c,s}$ συστάδες, στις οποίες ομαδοποιούνται σειρές με υψηλή ομοιότητα μεταξύ τους.

Κεφάλαιο 6

Εφαρμογή τεχνητού παγκρέατος σε in-silico ασθενή διαβήτη τύπου I

Στο κεφάλαιο που ακολουθεί παρουσιάζονται τα αποτελέσματα της εφαρμογής όσων αναπτύχθηκαν προηγουμένως στη χορήγηση ινσουλίνης σε ασθενείς με διαβήτη τύπου I. Το κεφάλαιο διακρίνεται σε δύο ενότητες. Στην Ενότητα 6.1 σχεδιάζεται και αξιολογείται ένας τυπικός MPC ελεγκτής με γραμμικό ARX μοντέλο. Στη συνέχεια, στην Ενότητα 6.2, η διαδικασία επαναλαμβάνεται, αλλά αυτή τη φορά γίνεται χρήση μοντέλου Takagi-Sugeno για τη μετατροπή του απλού γραμμικού μοντέλου σε ψευδογραμμικό.

6.1 Σχεδιασμός MPC ελεγκτή με γραμμικό μοντέλο

Ο σχεδιασμός του MPC ελεγκτή μπορεί να διακριθεί σε τρία στάδια. Στο πρώτο κατασκευάζονται τα κατάλληλα σύνολα δεδομένων από τον in-silico ασθενή. Αυτά χρησιμοποιούνται σε δεύτερο στάδιο για την αναγνώριση της δυναμικής γλυκόζης-ινσουλίνης. Στο τέλος αυτού του σταδίου έχει προσδιοριστεί το γραμμικό μοντέλο που θα χρησιμοποιηθεί στον ελεγκτή. Το τελευταίο στάδιο αφορά στον σχεδιασμό του ελεγκτή αυτού καθαυτού και στον έλεγχο της συμπεριφοράς του. Στην πραγματικότητα τα παραπάνω βήματα εφαρμόζονται κυκλικά. Στην περίπτωση που η συμπεριφορά του ελεγκτή είναι η επιθυμητή, τότε η διαδικασία ολοκληρώνεται. Αν όμως αυτό δε συμβαίνει, τότε επιστρέφοντας στα προηγούμενα βήματα και κάνοντας κατάλληλες αλλαγές μπορεί τελικά να επιτευχθεί το επιθυμητό αποτέλεσμα.

6.1.1 Κατασκευή συνόλων πειραματικών δεδομένων

Είναι επιθυμητό να κατασκευαστεί ελεγκτής που να ρυθμίζει την τιμή της γλυκόζης στο αίμα του ασθενή. Ως μεταβλητή εκ χειρισμού του συστήματος λαμβάνεται η ινσουλίνη που χορηγείται υποδόρια. Το μοντέλο γλυκόζης-ινσουλίνης στο MoBi χρησιμοποιείται στην αρχική του παραμετροποίηση. Οι καθολικές παράμετροι, όπως και εκείνες που είναι κοινές για όλους τους διαβητικούς ασθενείς, δίνονται από τους Schaller et al. [1]. Στο Παράρτημα Β', στο τέλος της εργασίας, σημειώνονται οι παράμετροι που αφορούν στον ασθενή που προσομοιώνεται παρακάτω. Πάντως αυτό είναι ένα ζήτημα που δε θα μας απασχολήσει στην παρούσα εργασία.

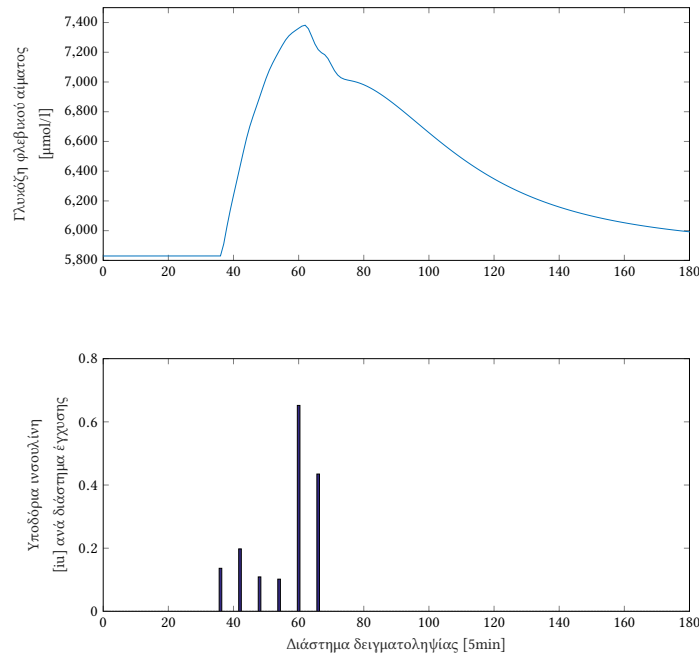
Η διαδικασία που ακολουθείται για την κατασκευή των συνόλων συνίσταται σε δύο σημεία: στο γεύμα που χορηγείται και στον τρόπο με τον οποίο πραγματοποιούνται οι εγχύσεις της ινσουλίνης. Πιο συγκεκριμένα:

Χορήγηση γεύματος. Το γεύμα προσομοιώνεται με τη χορήγηση γλυκόζης από το στόμα. Η επίδραση της σύστασής του στην γαστρική κένωση και στους εντερικούς ρυθμούς διέλευσης λαμβάνεται υπόψη βάσει του θερμιδικού περιεχομένου, του όγκου και του στερεού κλάσματός του. Στον ασθενή χορηγούνται 20 g γλυκόζης. Το θερμιδικό περιεχόμενο μεταβάλλεται τυχαία μεταξύ 0–1000 kcal, ο όγκος μεταξύ 0.15–0.8 l και το στερεό κλάσμα κυμαίνεται από 0 ως 1.

Έγχυση ινσουλίνης. Η βασική ινσουλιναιμία προσομοιώνεται με την προσθήκη μίας σταθερής, χαμηλής ροής ινσουλίνης ταχείας δράσης. Η ροή ορίζεται στα 0.16 iu/min, οπότε το ύψος της γλυκόζης ισορροπεί στα 5830 $\mu\text{mol l}^{-1}$ σε μόνιμες συνθήκες. Πριν την έναρξη οποιασδήποτε προσομοίωσης, το σύστημα αρχικοποιείται σε μόνιμη κατάσταση. Μετά από τη χορήγηση του γεύματος επιλέγεται να γίνουν συνολικά 6 εγχύσεις ινσουλίνης, μία κάθε 30 λεπτά, για διάστημα 3 ωρών. Η κάθε έγχυση διαρκεί 5 λεπτά και γίνεται με ρυθμό που επιλέγεται τυχαία στο διάστημα 0–0.15 iu/min.

Στο Σχήμα 6.1 δίνεται ένα από τα σύνολα πειραματικών δεδομένων που κατασκευάζονται. Συνολικά προσδιορίζονται 10 σύνολα τέτοιας μορφής.

Εδώ θα πρέπει να σημειωθούν δύο πράγματα. Καταρχάς, η χορήγηση σταθερής δόσης γλυκόζης μπορεί να φαντάζει παράλογη. Στην πραγματικότητα όμως, μεταβάλλοντας επιπλέον το ύψος της γλυκόζης, δεν προσδιορίζονται μοντέλα ικανά να παρέχουν ποιοτικότερη πρόβλεψη. Αυτό από τη μία οφείλεται στη χρήση μόνο της ινσουλίνης ως είσοδο στο μοντέλο. Οι Pedersen και Hansen [8] επίσης προσδιορίζουν ARX μοντέλα για τον υπολογισμό της γλυκόζης στο αίμα, όμως θεωρούν ως δεύτερη είσοδο την ποσότητα των υδατανθράκων του γεύματος. Τα μοντέλα που προκύπτουν παρουσιάζουν



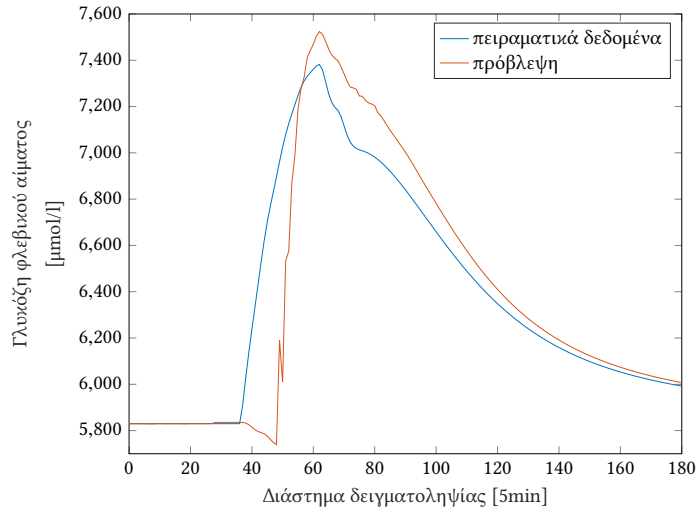
Σχήμα 6.1 Πειραματικά δεδομένα εισόδου-εξόδου για τον in-silico ασθενή

καλή προβλεπτική ικανότητα σε ορίζοντα πρόβλεψης 1 ώρας. Από την άλλη βέβαια, μεταβάλλοντας το ύψος του γεύματος αναδεικνύουμε εντονότερα τη μη γραμμικότητα του συστήματος προς αναγνώριση. Τα μοντέλα ARX είναι γραμμικά. Θα περίμενε λοιπόν κανείς να είναι ικανά να προβλέψουν μόνο την τοπική συμπεριφορά ενός μη γραμμικού συστήματος, όπως άλλωστε και συμβαίνει.

Το δεύτερο σημείο αναφοράς αφορά στον χρόνο δειγματοληψίας που χρησιμοποιείται. Προηγουμένως αναφέρθηκε πως η κάθε έγχυση διαρκεί 5 λεπτά. Από εδώ και στο εξής αυτό θα ισχύει πάντοτε. Ο ελεγκτής που σχεδιάζεται βασίζεται στη μέτρηση της τιμής της γλυκόζης ανά 5 λεπτά. Αυτό σημαίνει πως οι αποφάσεις που λαμβάνει για την εγχύσιμη ινσουλίνη παραμένουν ενεργές για όλη τη διάρκεια των 5 αυτών λεπτών, έως ότου γίνει διαθέσιμη η νεότερη μέτρηση και ενημερωθεί η απόφαση.

6.1.2 Προσδιορισμός ARX μοντέλου

Ο προσδιορισμός των ARX μοντέλων εδώ, βασίζεται στη μέθοδο των ελαχίστων τετραγώνων που περιγράφεται στην Υποενότητα 3.2.3. Η εφαρμογή της μεθόδου απλοποιείται χρησιμοποιώντας το System Identification Toolbox του MATLAB. Το εργαλείο αυτοματοποιεί σε σημαντικό βαθμό τον καθορισμό της τάξης του μοντέλου. Αυτό με τη βέλτιστη δομή επιλέγεται καταρχάς στη βάση της ελαχιστοποίησης των κριτηρίων AIC, BIC και MSE που δίνονται από τις Εξισώσεις 3.23–3.25. Στη συνέχεια πραγματοποιείται οπτικός



Σχήμα 6.2 Εφαρμογή του ARX μοντέλου για πρόβλεψη 12 βημάτων

έλεγχος της ποιότητας για την πρόβλεψη 12 βημάτων, δηλαδή μίας ώρας μπροστά στο μέλλον. Με τον τρόπο αυτό προσδιορίζεται τελικά το μοντέλο

$$\begin{aligned}
 y(t) = & 2.9329y(t-1) - 4.4927y(t-2) + 4.9346y(t-3) - 4.1337y(t-4) + 3.0330y(t-5) \\
 & - 2.0880y(t-6) + 1.4408y(t-7) - 0.8984y(t-8) + 0.3326y(t-9) + 0.0030y(t-10) \\
 & - 0.1015y(t-11) + 0.0374y(t-12) - 3.4792u(t-1) - 11.197u(t-2) + 26.717u(t-3) \\
 & - 14.962u(t-4) - 22.420u(t-5) - 12.893u(t-6) - 4.0676u(t-7) + 3.6880u(t-8) \\
 & + 9.7426u(t-9) + 14.958u(t-10) + 17.355u(t-11) + 14.542u(t-12) + 11.688u(t-13) \\
 & + 9.3928u(t-14) + 5.2111u(t-15) + 2.8811u(t-16) + 1.4288u(t-17)
 \end{aligned}
 \tag{6.1}$$

και στο Σχήμα 6.2 χρησιμοποιείται στην πρόβλεψη 12 βημάτων με τα πειραματικά δεδομένα του Σχήματος 6.1. Με βάση το μοντέλο αυτό σχεδιάζεται ο προβλεπτικός ελεγκτής αμέσως παρακάτω.

6.1.3 Υλοποίηση του MPC ελεγκτή

Η υλοποίηση του MPC ελεγκτή, όπως παρουσιάζεται στο Κεφάλαιο 4, είναι αρκετά περίπλοκη. Η συνάρτηση κόστους στην τελική μορφή της Εξίσωσης 4.27 απαιτεί τον καθορισμό των πινάκων βάρους Q , R και S . Έχει ήδη αναφερθεί πως συχνά επιλέγεται

$$Q = I_{n_x} \tag{6.2}$$

για την απλοποίηση της επιλογής που καλούμαστε να κάνουμε. Αυτό συμβαίνει και εδώ. Η μελέτη απλοποιείται περαιτέρω αν λάβουμε υπόψη πως το σύστημα έχει μόνο μία μεταβλητή εισόδου και μία εξόδου, οπότε είναι

$$R = \rho \qquad S = \sigma \qquad (6.3)$$

δηλαδή οι πίνακες εκφυλίζονται σε δύο βαθμωτές μεταβλητές. Η μεταβλητή σ κοστολογεί την παραβίαση των μαλακών φραγμών από τον ελεγκτή. Σε αυτή την πρώτη προσέγγιση, όμως, θα επιβάλλουμε αποκλειστικά σκληρούς φραγμούς. Βεβαίως αυτό επαναφέρει στο προσκήνιο το ζήτημα που πρώτο μας οδήγησε στην εφαρμογή μαλακών φραγμών: συχνά οι φραγμοί αναπόφευκτα παραβιάζονται και αν αυτοί είναι σκληροί, τότε το πρόβλημα βελτιστοποίησης δεν επιδέχεται λύσης. Αυτό παρακάμπτεται θέτοντας

$$y_{max} = \infty \qquad (6.4)$$

$$y_{min} \in \mathbb{R} \qquad (6.5)$$

$$\underline{AN} \exists t' \in [t, t + N_p] \text{ τέτοιο ώστε: } y(t') < y_{min} \quad \underline{TOPE} \quad u(t) = 0 \qquad (6.6)$$

γεγονός που φυσικά αποτελεί μη βέλτιστη διατύπωση. Το ζήτημα πλέον ανάγεται στον καθορισμό της μεταβλητής ρ , για την οποία επιλέγεται

$$\rho = 10^7 \qquad (6.7)$$

ως τιμή για την οποία ο ελεγκτής συμπεριφέρεται ικανοποιητικά. Ο ορίζοντας πρόβλεψης ορίζεται στις 6 ώρες. Απομένει πλέον να καθοριστούν οι τιμές για τους φραγμούς του συστήματος. Στην ινσουλίνη που εγχέεται υποδόρια επιβάλλονται οι σκληροί φραγμοί

$$0 \text{ iu} \leq u(t) \leq 0.75 \text{ iu} \qquad (6.8)$$

ενώ για την τιμή της γλυκόζης στο αίμα είναι

$$70 \text{ mg dl}^{-1} \leq C\hat{x}(t) \qquad (6.9)$$

Η απόδοση του ελεγκτή ελέγχεται στη βάση της χορήγησης τριών γευμάτων διαφορετικού γλυκαιμικού φορτίου σε διάρκεια 24 ωρών. Η παραμετροποίηση των γευμάτων στο MoBi δίνεται στον Πίνακα 6.1. Εδώ σημειώνεται ότι για περίοδο 3 ωρών μετά από το γεύμα είναι επιθυμητό η γλυκόζη να μην ξεπεράσει τα 180 mg dl^{-1} [1]. Η ρυθμιστική δράση του ελεγκτή παρουσιάζεται στο Σχήμα 6.3. Τα φυσιολογικά και τα διευρυμένα όρια της γλυκόζης σημειώνονται με πράσινο και κίτρινο χρώμα αντίστοιχα. Είναι αξιο-

Πίνακας 6.1 Παραμετροποίηση γευμάτων στο MoBi

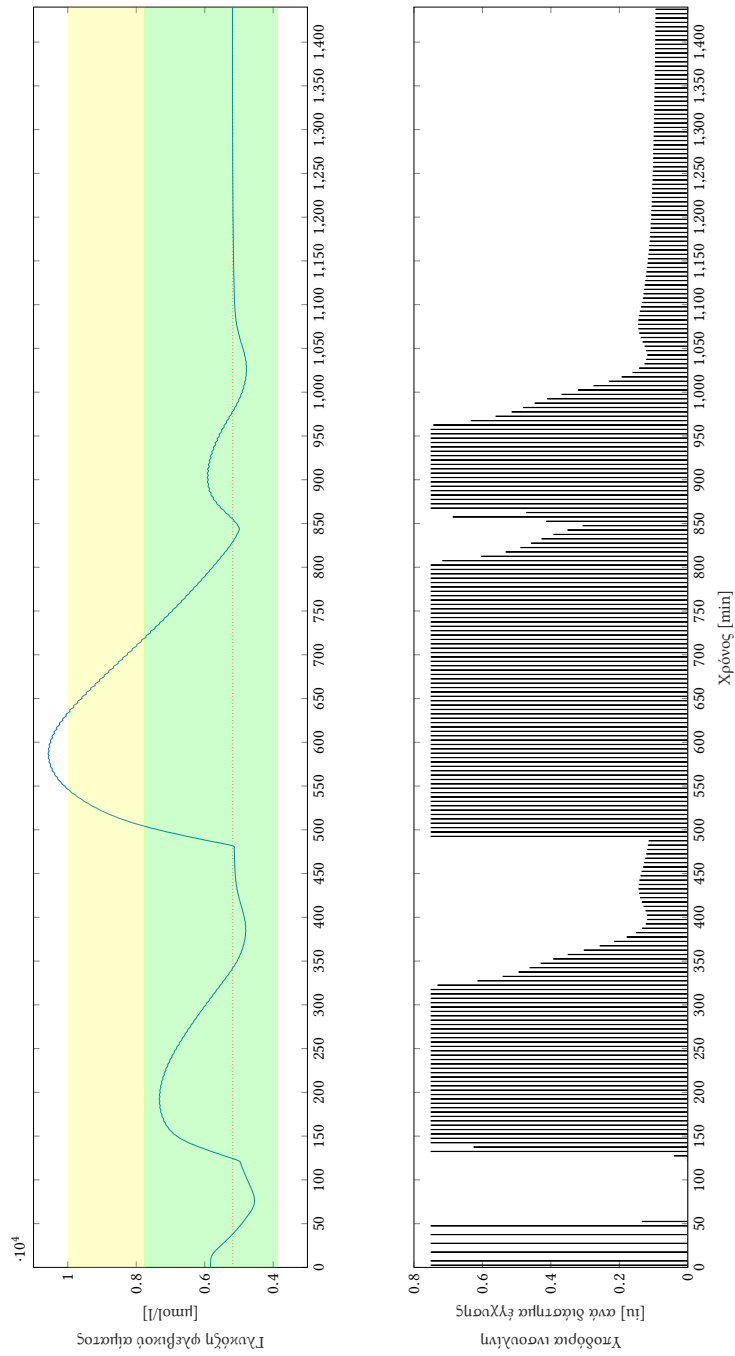
Γεύμα	Πρωινό	Μεσημεριανό	Βραδινό
Χρόνος χορήγησης ¹ (hr)	2 (7π.μ.)	8 (1μ.μ.)	14 (7μ.μ.)
Γλυκαιμικό φορτίο (g γλυκόζης)	50	100	20
Θερμιδικό περιεχόμενο (kcal)	690	1000	340
Όγκος (l)	0.50	0.59	0.15
Στερεό κλάσμα	0.8	0.6	1.0

¹ Αυτός είναι ο απόλυτος χρόνος στην προσομοίωση. Η ώρα χορήγησης στη διάρκεια της ημέρας δίνεται σε παρένθεση. Σε χρόνο προσομοίωσης $t = 0$ είναι δηλαδή 5π.μ.

σημείωτο πως η συμπεριφορά του ελεγκτή παραμένει ικανοποιητική, παρότι το μοντέλο που χρησιμοποιείται αδυνατεί να παρέχει ποιοτικές προβλέψεις. Η περιορισμένη αυτή προβλεπτική ικανότητα, όμως, καθιστά αδύνατη την εφαρμογή του μοντέλου σταθερής διαταραχής των μεταβλητών κατάστασης. Υπό αυτές τις συνθήκες δεν εξασφαλίζεται η ικανότητα του ελεγκτή να επιτύχει έλεγχο μηδενικού σφάλματος. Εδώ φυσικά γεννούνται τα εξής ερωτήματα: *πρώτον, γιατί συμβαίνει αυτό και δεύτερον, ποιος είναι τελικά ο λόγος σχεδιασμού ενός τέτοιου ελεγκτή;*

Το πρώτο ερώτημα μπορεί να απαντηθεί αν λάβουμε υπόψη τη φύση της διορθωτικής ενέργειας που λαμβάνει χώρα. Προκειμένου να εξασφαλιστεί απόρριψη τυχόν σταθερών διαταραχών που επηρεάζουν το σύστημα, γίνεται η θεώρηση πως η διαφορά μεταξύ πρόβλεψης και πραγματικής τιμής στο παρόν παραμένει σταθερή και στο μέλλον, επηρεάζοντας έτσι τις προβλέψεις σε όλο το μήκος του ορίζοντα πρόβλεψης. Εικάζεται πως η μη ικανοποιητική συμπεριφορά του μοντέλου οδηγεί τελικά στον υπολογισμό διαταραχών μεγάλου ύψους, με αποτέλεσμα η επέκτασή τους κατά μήκος του ορίζοντα πρόβλεψης να αποσταθεροποιεί τελικά το ρυθμιστικό σύστημα.

Το δεύτερο ερώτημα παρουσιάζει όμως μεγαλύτερο ενδιαφέρον: *ποιος ο λόγος σχεδιασμού ενός τέτοιου ελεγκτή τελικά;* Τα όσα παρουσιάζονται παραπάνω θα πρέπει να θεωρηθούν ως ενδιάμεσο βήμα που θα μας επιτρέψει να κατασκευάσουμε στη συνέχεια έναν ρυθμιστή υψηλότερης απόδοσης. Παρότι στην παρούσα του μορφή ο ελεγκτής δε θα μπορούσε να εφαρμοστεί για τη ρύθμιση της γλυκόζης σε έναν διαβητικό ασθενή επί καθημερινής βάσης, εντούτοις μπορεί να αποτελέσει χρήσιμο εργαλείο σε περιβάλλον κλινικής δοκιμής. Ο Schaller^[6] χρησιμοποιεί προβλεπτικό ελεγκτή για τον καθορισμό της επόμενης βέλτιστης δόσης. Το κλινικό προσωπικό αξιολογεί την ορθότητα της πρόβλεψης και έχει τον τελικό λόγο σε ότι αφορά την πραγματοποίηση ή μη της έγχυσης.



Σχήμα 6.3 Ρυθμιστική δράση MPC ελεγκτή με γραμμικό ARX μοντέλο

Στην περίπτωση που αυτή δεν κριθεί ασφαλής, τότε στον ασθενή χορηγείται κατάλληλα διορθωμένη δόση. Τα νέα δεδομένα εισάγονται στον ελεγκτή και η διαδικασία επαναλαμβάνεται για το επόμενο διάστημα δειγματοληψίας.

6.2 Επέκταση ελεγκτή με ψευδογραμμικό μοντέλο

Ο ελεγκτής που υλοποιείται παραπάνω, χρησιμοποιείται εδώ για την κατασκευή νέων συνόλων δεδομένων. Στον ασθενή χορηγούνται γεύματα με γλυκαιμικό φορτίο στο εύρος 0–100 g και η ρυθμιστική δράση του ελεγκτή καταγράφεται. Με τον τρόπο αυτό κατασκευάζονται σύνολα που αναμένεται να παρέχουν περισσότερη πληροφορία ως προς τη δυναμική του συστήματος, για τον τρόπο που εμείς σκοπεύουμε να το χρησιμοποιήσουμε.

6.2.1 Σχεδιασμός μοντέλου Takagi-Sugeno

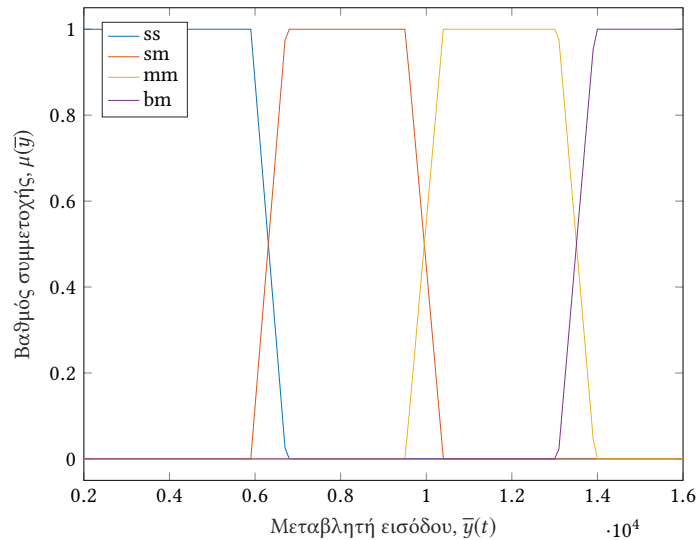
Στα νέα σύνολα δεδομένων εφαρμόζεται ο αλγόριθμος ασαφούς συσταδοποίησης που παρουσιάζεται στην Υποενότητα 5.3.3. Για την πρώτη συσταδοποίηση, που πραγματοποιείται σε επίπεδο πειραματικών σημείων, επιλέγεται

$$N_{c,p} = 4 \quad (6.10)$$

για το πλήθος των συστάδων. Η επιλογή του αριθμού των συστάδων προσεγγίζεται με δοκιμή και σφάλμα. Σε δεύτερο στάδιο, οι χρονοσειρές συσταδοποιούνται επιλέγοντας

$$N_{c,s} = 5 \quad (6.11)$$

οπότε προκύπτουν συναρτήσεις συμμετοχής τέτοιες, ώστε η κάθε χρονοσειρά να κατηγοριοποιείται αποκλειστικά σε μία συστάδα. Αυτό είναι βεβαίως επιθυμητό και δικαιολογεί την επιλογή των μεταβλητών $N_{c,p}$ και $N_{c,s}$ που γίνεται προηγουμένως. Οι συναρτήσεις στη μορφή αυτή, όμως, δεν είναι ακόμα χρηστικές. Για κάθε συστάδα υπολογίζονται οι μέγιστες τιμές της γλυκόζης ανά χρονοσειρά. Στη συνέχεια η χαμηλότερη και η υψηλότερη τιμή ορίζονται ως κάτω και άνω όριο της εκάστοτε συνάρτησης συμμετοχής, αντίστοιχα. Αυτές πλέον αποκτούν τη μορφή του Σχήματος 6.4. Σε κάθε συνάρτηση συμμετοχής αντιστοιχίζεται ένα τοπικό μοντέλο. Δεδομένου πως οι χρονοσειρές είτε ανήκουν, είτε δεν ανήκουν στη κάθε συστάδα, ο τροποποιημένος αλγόριθμος της Υποενότητας 5.3.2 παρακάμπτεται. Το κάθε τοπικό μοντέλο προσαρμόζεται με βάση τα πειραματικά σύνολα που ανήκουν στην αντίστοιχη συστάδα με την εφαρμογή του αλγορίθμου ελαχίστων τετραγώνων στην κλασική διατύπωσή του.



Σχήμα 6.4 Συναρτήσεις συμμετοχής με είσοδο $\bar{y}(t)$. η 'ss' αντιστοιχεί στη μόνιμη κατάσταση και οι 'sm', 'mm' και 'bm' περιγράφουν μικρό, μεσαίο και μεγάλο γεύμα αντίστοιχα

Ο ελεγκτής σχεδιάζεται με τέτοιο τρόπο, ώστε το μοντέλο που επιλέγεται κάθε χρονική στιγμή να χρησιμοποιείται για πρόβλεψη σε όλο το μήκος του ορίζοντα. Αυτό συνάδει με το γεγονός πως τα ARX μοντέλα που κατασκευάζονται παραπάνω είναι τοπικά, υπό την έννοια πως παρουσιάζουν καλή προβλεπτική ικανότητα για γεύματα σε δεδομένο εύρος γλυκαιμικού φορτίου. Είναι επιθυμητό, λοιπόν, όταν επιλεγεί το σωστό μοντέλο, αυτό να συνεχίσει να χρησιμοποιείται έως ότου το σύστημα επιστρέψει σε μόνιμες συνθήκες. Παρακολουθώντας απλώς το ύψος της γλυκόζης, κάτι τέτοιο δε θα ήταν εφικτό. Ενώ το ίδιο γεύμα συνεχίζει να επηρεάζει το σύστημα, η τιμή της γλυκόζης φθάνει σε μία μέγιστη τιμή και στη συνέχεια φθίνει. Αυτό σημαίνει πως κινούμαστε προς τα δεξιά, κατά μήκος των συναρτήσεων συμμετοχής του Σχήματος 6.4 και έπειτα πάλι προς τα αριστερά.

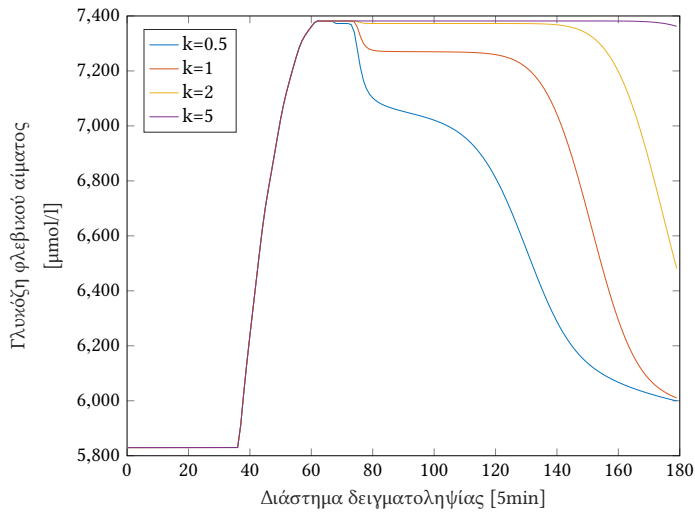
Το πρόβλημα αντιμετωπίζεται επιλέγοντας σαν είσοδο την τιμή που σε κάθε χρονική στιγμή υπολογίζεται από τη συνάρτηση

$$\bar{y}(t) = \left(1 - \frac{1}{1 + \exp(-k\dot{y}(t))}\right) \bar{y}(t-1) + \frac{1}{1 + \exp(-k\dot{y}(t))} y(t) \quad (6.12)$$

όπου k είναι μία αρκετά μεγάλη τιμή. Εδώ ορίζεται

$$k = 10 \quad (6.13)$$

Στο Σχήμα 6.5 δίνεται η εξέλιξη της τιμής της συνάρτησης για τα πειραματικά δεδομένα του Σχήματος 6.1, για διαφορετικές τιμές της παραμέτρου k . Για μεγάλα k , η συνάρτηση

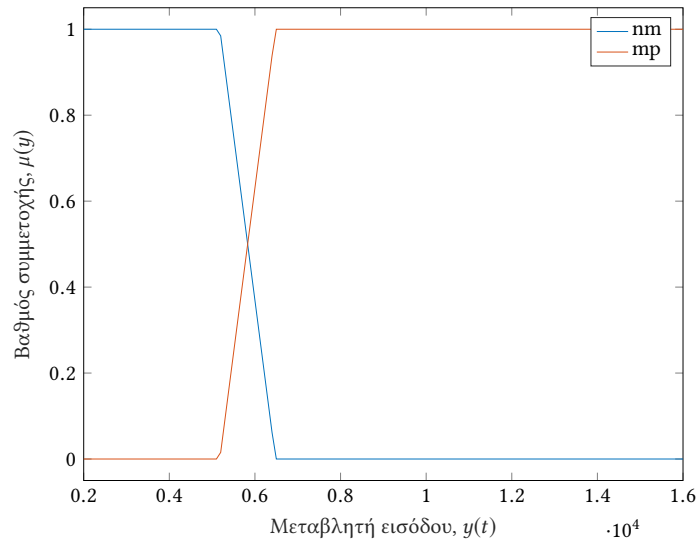


Σχήμα 6.5 Παράδειγμα εφαρμογής της προτεινόμενης συνάρτησης εισόδου για το τμήμα απόφασης του συστήματος Takagi-Sugeno σε πειραματικά δεδομένα

ακολουθεί την πραγματική τιμή της γλυκόζης όσο αυτή αυξάνει. Στη συνέχεια διατηρεί την τελευταία υψηλότερη τιμή, για όση διάρκεια τα επίπεδά της φθίνουν στο αίμα. Σε μόνιμες συνθήκες η συνάρτηση δίνει τον μέσο όρο των δύο πιο πρόσφατων μετρήσεων.

Το σύστημα ολοκληρώνεται με την προσθήκη των συναρτήσεων συμμετοχής του Σχήματος 6.6. Ως τιμή εισόδου εδώ χρησιμοποιείται η πραγματική τιμή της γλυκόζης. Η λογική πίσω από αυτό είναι απλή. Με τη χρήση της εισόδου της Εξίσωσης 6.12, το σύστημα ακολουθεί την τελευταία υψηλότερη τιμή της γλυκόζης για όσο η πρώτη παράγωγός της είναι επαρκώς αρνητική. Για να γίνει φανερό το ενδεχόμενο πρόβλημα μιας τέτοιας διαμόρφωσης, ας υποθέσουμε ότι στον ασθενή χορηγείται γεύμα υψηλού γλυκαιμικού φορτίου. Η τιμή της γλυκόζης αυξάνει και έπειτα φθίνει, όμως το σύστημα ακολουθεί την τιμή τοπικού μεγίστου και συνεχίζει να χρησιμοποιεί το σωστό μοντέλο που αντιστοιχεί στο μεγάλο γεύμα. Έστω τώρα ότι ο ελεγκτής υπολειτουργεί και συνεχίζει να εγχέει ινσουλίνη, οδηγώντας έτσι τον ασθενή σε κατάσταση υπογλυκαιμίας. Το σύστημα όμως ποτέ δε μεταβαίνει στη χρήση του κατάλληλου μοντέλου. Αντιθέτως, όσο η πρώτη παράγωγος παραμένει αρνητική συνεχίζει να χρησιμοποιεί το μοντέλο μεγάλου γεύματος. Με την προσθήκη των επιπλέον συναρτήσεων συμμετοχής το πρόβλημα εξαλείφεται. Οι κανόνες του συστήματος διαμορφώνονται τελικά ως εξής

$$\begin{aligned}
 R^1 : & \underline{AN} \bar{y}(t) \text{ EINAI } \hat{Y}^{ss} \quad \underline{H} \quad y(t) \text{ EINAI } \Upsilon^{nm} \quad \underline{TOTE} \quad \hat{x}(t) = \mathbf{A}^{nm} \hat{x} + \mathbf{B}^{nm} u(t) \\
 R^2 : & \underline{AN} \bar{y}(t) \text{ EINAI } \hat{Y}^{sm} \quad \underline{KAI} \quad y(t) \text{ EINAI } \Upsilon^{mp} \quad \underline{TOTE} \quad \hat{x}(t) = \mathbf{A}^{sm} \hat{x} + \mathbf{B}^{sm} u(t) \\
 R^3 : & \underline{AN} \bar{y}(t) \text{ EINAI } \hat{Y}^{mm} \quad \underline{KAI} \quad y(t) \text{ EINAI } \Upsilon^{mp} \quad \underline{TOTE} \quad \hat{x}(t) = \mathbf{A}^{mm} \hat{x} + \mathbf{B}^{mm} u(t) \\
 R^4 : & \underline{AN} \bar{y}(t) \text{ EINAI } \hat{Y}^{bm} \quad \underline{KAI} \quad y(t) \text{ EINAI } \Upsilon^{mp} \quad \underline{TOTE} \quad \hat{x}(t) = \mathbf{A}^{bm} \hat{x} + \mathbf{B}^{bm} u(t)
 \end{aligned}$$



Σχήμα 6.6 Συναρτήσεις συμμετοχής με είσοδο $y(t)$. η 'nm' αντιστοιχεί στην απουσία και η 'mp' στην παρουσία επίδρασης από γεύμα

6.2.2 Υλοποίηση βελτιωμένου MPC ελεγκτή

Στην υλοποίηση του ελεγκτή πλέον επιβάλλονται μαλακοί φραγμοί για την τιμή της γλυκόζης. Γενικά είναι επιθυμητό οι φραγμοί να παραβιάζονται μόνο στην περίπτωση που το σύστημα δεν επιλύεται διαφορετικά. Αυτό μπορεί να εξασφαλιστεί επιλέγοντας μία αρκετά μεγάλη τιμή για την μεταβλητή σ . Εδώ ορίζεται

$$\sigma = 10^{-1} \quad (6.14)$$

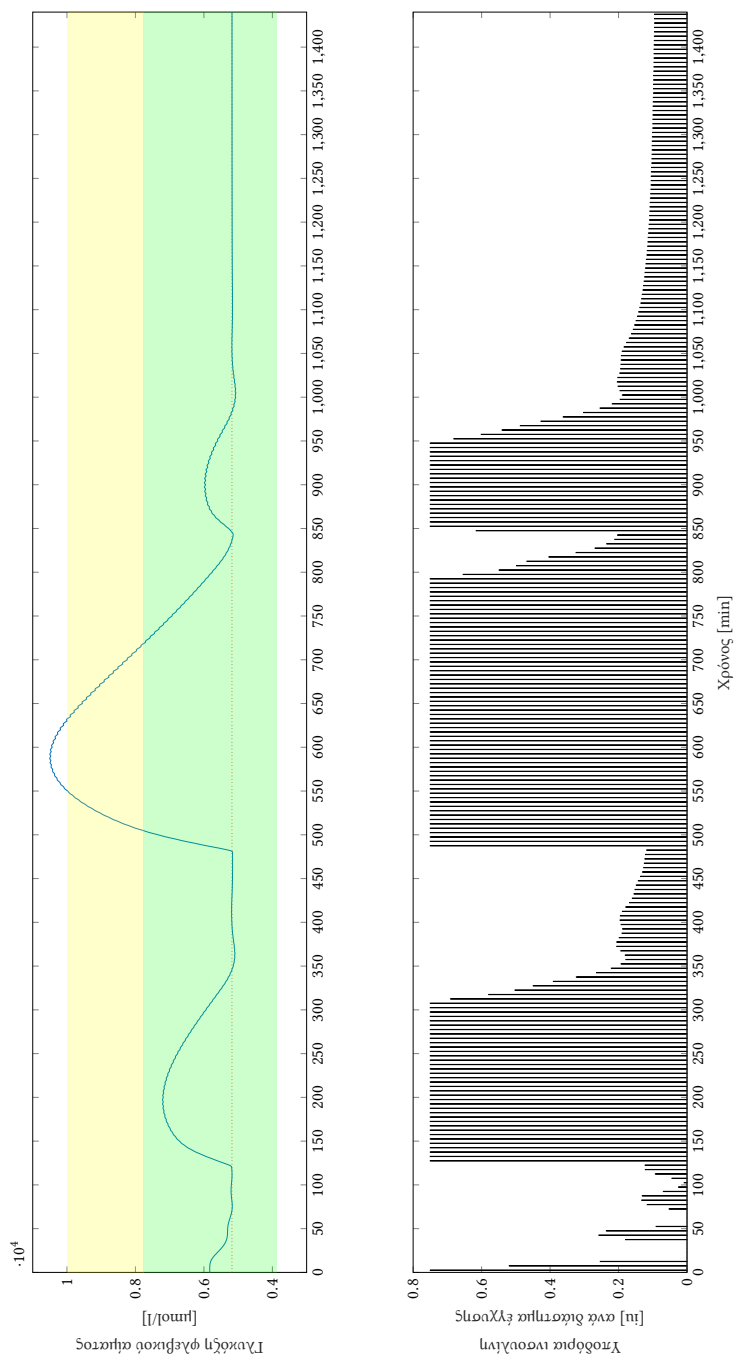
Σημειώνεται πως καθώς αυξάνουμε την τιμή της παραμέτρου, τελικά φθάνουμε σε ένα βέλτιστο όριο. Πολύ μεγάλες τιμές είναι βέβαια ανεπιθύμητες, καθώς είναι πιθανό να δυσχεράνουν την υπολογιστική επίλυση του προβλήματος. Γίνεται επίσης εκ νέου επιλογή της παραμέτρου ρ

$$\rho = 10^6 \quad (6.15)$$

έτσι ώστε να εξασφαλίζεται καλή συμπεριφορά. Οι φραγμοί του συστήματος διαφέρουν μόνο ως προς την τιμή της γλυκόζης, η οποία πλέον φράζεται μεταξύ των φυσιολογικών τιμών υπογλυκαιμίας και υπεργλυκαιμίας, δηλαδή

$$70 \text{ mg dl}^{-1} \leq C\hat{x}(t) \leq 140 \text{ mg dl}^{-1} \quad (6.16)$$

Τελικά η ρυθμιστική δράση του ελεγκτή δίνεται στο Σχήμα 6.7, όπου προσομοιώνεται η χορήγηση των γευμάτων του Πίνακα 6.1.



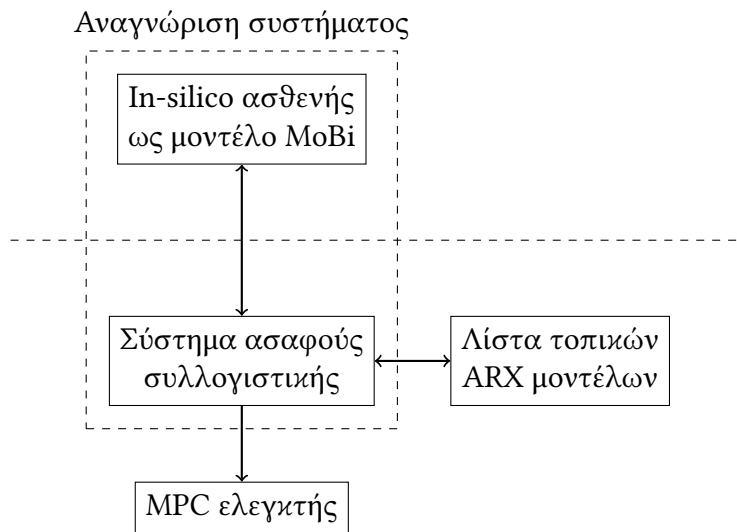
Σχήμα 6.7 Ρυθμιστική δράση MPC ελεγκτή με ψευδογραμμικό ARX μοντέλο

Κεφάλαιο 7

Συζήτηση και μελλοντική δουλειά

Στην εργασία δείχνεται πως το πρόβλημα της μη γραμμικότητας της δυναμικής του μεταβολισμού γλυκόζης-ινσουλίνης μπορεί να αντιμετωπιστεί αποτελεσματικά με την επέκταση των γραμμικών ARX σε ψευδογραμμικά. Με την εφαρμογή συσταδοποίησης σε επίπεδο χρονοσειρών και όχι πειραματικών σημείων, ο σχεδιασμός του μοντέλου Takagi-Sugeno απλουστεύεται σημαντικά. Ο τοπικός χαρακτήρας των μοντέλων που προσδιορίζονται με αυτόν τον τρόπο, έγκειται στην ικανότητα του καθενός να παρέχει ικανοποιητικές προβλέψεις για γεύματα σε δεδομένο εύρος γλυκαιμικού φορτίου. Η επιλογή του σωστού μοντέλου ανάγεται λοιπόν στον προσδιορισμό του ύψους του γεύματος που χορηγείται σε κάθε περίπτωση. Εφόσον το σωστό μοντέλο επιλεγεί, τότε αυτό χρησιμοποιείται έως ότου το σύστημα επανέλθει σε μόνιμες συνθήκες. Το αποτέλεσμα είναι ένας ελεγκτής MPC που λειτουργεί στη βάση των πολλαπλών μοντέλων, ενώ διατηρεί τις υπολογιστικές ανάγκες της κλασικής διατύπωσης με ένα μοντέλο. Ο ελεγκτής καινοτομεί από την άποψη πως δεν απαιτεί καμία είσοδο από την πλευρά του ασθενή, ενώ διατηρεί ικανοποιητική συμπεριφορά, καταφέρνοντας να διατηρήσει το ύψος της γλυκόζης σε φυσιολογικά επίπεδα. Εξαιρεση αποτελεί το μεγάλο γεύμα με γλυκαιμικό φορτίο 100 g. Στην περίπτωση αυτή ο ελεγκτής περιορίζεται από το άνω όριο που τίθεται για την ινσουλίνη, αλλά η συμπεριφορά που επιδεικνύει παραμένει η επιθυμητή. Πάντως εδώ το ύψος της γλυκόζης διατηρείται κάτω των 180 mg dl^{-1} για την πλειονότητα του χρόνου και επανέρχεται σε φυσιολογικά επίπεδα ($<140 \text{ mg dl}^{-1}$) περίπου 4 ώρες μετά το γεύμα.

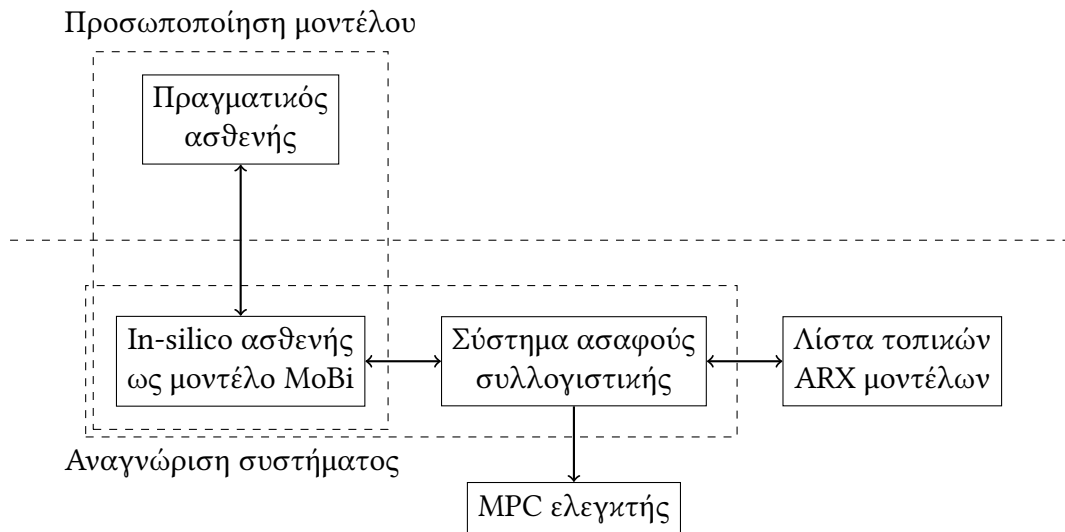
Στα πλεονεκτήματα της μεθόδου που παρουσιάζεται εδώ ανήκουν οι σημαντικές προοπτικές για μελλοντική δουλειά. Οι Schaller et al. ^[1] σχεδιάζουν προβλεπτικό ελεγκτή χρησιμοποιώντας απευθείας το μοντέλο της γλυκόζης-ινσουλίνης στο MoBi και αναφέρουν υποσχόμενα αποτελέσματα. Η προσέγγιση παρουσιάζει σημαντικά πλεονεκτήματα, αλλά ταυτόχρονα υπόκειται σε σοβαρούς περιορισμούς. Από τη μία, η απευθείας χρήση του PBRK μοντέλου για on-line ρύθμιση μπορεί δυνητικά να προσφέρει πολύ ποιοτικές



Σχήμα 7.1 Διαγραμματική απεικόνιση της πορείας που ακολουθείται στην εργασία για την ανάπτυξη του προβλεπτικού συστήματος

προβλέψεις. Ταυτόχρονα, τα νέα πειραματικά δεδομένα που γίνονται διαθέσιμα μπορούν να χρησιμοποιηθούν για την on-line παραμετροποίησή του, βελτιώνοντας έτσι περαιτέρω τις προβλεπτικές ικανότητές του στο μέλλον. Από την άλλη πλευρά, όμως, είναι απαραίτητη η είσοδος δεδομένων γεύματος από τον ασθενή. Μάλιστα αυτός είναι ένας περιορισμός που δεν μπορεί να αντιμετωπισθεί εύκολα στην περίπτωση αυτή. Η προσέγγιση κυρίως όμως υστερεί ως προς τις προοπτικές της για υλοποίηση στον πραγματικό κόσμο. Η υψηλή πολυπλοκότητα του PBPΚ μοντέλου αποτελεί μειονέκτημα όταν αυτό καλείται να επιλυθεί σε πραγματικό χρόνο, κάνοντας τη διαδικασία υπολογιστικά και χρονικά δαπανηρή. Αλλά και η χρήση του MoBi αυτού καθαυτού εμφανίζεται προβληματική, από την άποψη πως πρόκειται για λογισμικό που, παρότι ανοιχτού κώδικα, είναι διαθέσιμο μόνο σε λειτουργικό σύστημα Microsoft Windows. Είναι κοινώς αποδεκτό πως κάποια πραγματική υλοποίηση είναι αδύνατο να βασιστεί σε αυτό.

Στην παρούσα εργασία το MoBi έχει τον ρόλο του πραγματικού ασθενή. Η πορεία που ακολουθείται δίνεται στο Σχήμα 7.1. Σύνολα πειραματικών δεδομένων κατασκευάζονται από τον ασθενή και με βάση αυτά προσαρμόζεται το ψευδογραμμικό μοντέλο του MPC ελεγκτή. Με βάση όσα αναφέρθηκαν παραπάνω προτείνεται τροποποίηση της μεθόδου που παρουσιάζεται στην εργασία, όπως αποτυπώνεται στο Σχήμα 7.2. Εδώ δεδομένα από τον πραγματικό ασθενή χρησιμοποιούνται για την ανάπτυξη ενός προσωποποιημένου PBPΚ μοντέλου στο MoBi. Στη συνέχεια το προσωποποιημένο μοντέλο μπορεί να χρησιμοποιηθεί για την προσαρμογή κάποιου ψευδογραμμικού μοντέλου ακολουθώντας τα βήματα που παρουσιάζονται στην παρούσα εργασία. Μία τέτοια προσέγγιση διατηρεί



Σχήμα 7.2 Διαγραμματική απεικόνιση της προτεινόμενης αλλαγής στην πορεία ανάπτυξης του προβλεπτικού συστήματος

τα πλεονεκτήματα των δύο μεθόδων, εξαλείφοντας τα προβλήματα που αναφέρθηκαν προηγουμένως. Τα PBRK μοντέλα έχουν την πλέον κατάλληλη δομή για εύκολη παραμετροποίηση και κατασκευή προσωποποιημένων μοντέλων. Εδώ η παραμετροποίηση λαμβάνει χώρα off-line και άρα δεν τίθεται ζήτημα υπολογιστικού κόστους. Αυτό ανοίγει τον δρόμο για την εφαρμογή ακόμα πιο πολύπλοκων μεθόδων. Για παράδειγμα, οι Krauss et al. [7] κάνουν εφαρμογή μπεϋζιανών μεθόδων για τον προσδιορισμό των παραμέτρων που αποτελούν τις πιο σημαντικές πηγές αβεβαιότητας σε επίπεδο πληθυσμού. Στη συνέχεια το υπολογιστικό μοντέλο μπορεί να χρησιμοποιηθεί για την κατά βούληση κατασκευή συνόλων πειραματικών δεδομένων, χωρίς το κόστος, την ταλαιπωρία και τον κίνδυνο που επέρχονται από την ανάγκη εισαγωγής του ασθενή στο νοσοκομείο. Εδώ σημειώνεται το επιπλέον πλεονέκτημα της πραγματοποίησης πειραμάτων που σε άλλη περίπτωση θα ήταν αδύνατο να λάβουν χώρα στον ασθενή για λόγους ασφαλείας. Τελικά ο ελεγκτής MPC λειτουργεί στη βάση ενός συστήματος ασαφούς συλλογιστικής που σε κάθε στιγμή καλείται να επιλέξει το βέλτιστο τοπικό μοντέλο προς χρήση. Η υλοποίηση είναι υπολογιστικά επιτρεπτή, ενώ μπορεί εύκολα να επεκταθεί με τη μέθοδο των επαναληπτικών ελαχίστων τετραγώνων, ώστε το μοντέλο να βελτιώνεται διαρκώς, καθώς νεότερα πειραματικά δεδομένα γίνονται διαθέσιμα.

Βιβλιογραφία

- [1] S Schaller et al. “A Generic Integrated Physiologically based Whole-body Model of the Glucose-Insulin-Glucagon Regulatory System”. In: *CPT: Pharmacometrics & Systems Pharmacology* 2.8 (2013), p. 65. DOI: 10.1038/psp.2013.40. URL: <https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1038/psp.2013.40>.
- [2] Rod R Seeley, Philip Tate, and Trent D Stephens. *Anatomy & physiology*. English. 8th ed. Dubuque, IA : McGraw-Hill, 2007. ISBN: 9780072965575.
- [3] S. Schaller et al. “Robust PBPK/PD-Based Model Predictive Control of Blood Glucose”. In: *IEEE Transactions on Biomedical Engineering* 63.7 (July 2016), pp. 1492–1504. ISSN: 0018-9294. DOI: 10.1109/TBME.2015.2497273.
- [4] L Kuepfer et al. “Applied Concepts in PBPK Modeling: How to Build a PBPK/PD Model”. In: *CPT: Pharmacometrics & Systems Pharmacology* 5.10 (2016), pp. 516–531. DOI: 10.1002/psp4.12134. URL: <https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1002/psp4.12134>.
- [5] Thomas Eissing et al. “A Computational Systems Biology Software Platform for Multiscale Modeling and Simulation: Integrating Whole-Body Physiology, Disease Biology, and Molecular Reaction Networks”. In: *Frontiers in Physiology* 2 (2011), p. 4. ISSN: 1664-042X. DOI: 10.3389/fphys.2011.00004. URL: <https://www.frontiersin.org/article/10.3389/fphys.2011.00004>.
- [6] Stephan Schaller. “Automated optimal glycaemic control using a physiology based pharmacokinetic, pharmacodynamic model”. Aachen, Techn. Hochsch., Diss., 2014. Diss. Aachen: Aachen, Techn. Hochsch., 2015, 163 S. : Ill., graph. Darst. URL: <https://publications.rwth-aachen.de/record/464366>.
- [7] Markus Krauss et al. “Bayesian Population Physiologically-Based Pharmacokinetic (PBPK) Approach for a Physiologically Realistic Characterization of Interindividual Variability in Clinically Relevant Populations”. In: *PloS one*. 2015.
- [8] Søren Nygaard Pedersen and Lasse Bergenholz Hansen. “System Identification for Prediction of Glucose Levels in People with Type 1 Diabetes”. Kongens Lyngby, Denmark: Technical University of Denmark, Department of Informatics and Mathematical Modeling, 2010. URL: http://etd.dtu.dk/thesis/263827/bac10_13_net.pdf.
- [9] G. Pocock and C.D. Richards. *Human Physiology-The Basis of Medicine third edition*. Oxford: Oxford University Press, 2006.
- [10] Harvey F Lodish and James E Darnell. *Molecular cell biology*. 3rd ed. New York Scientific American Books, 1995. ISBN: 0716723808. URL: <http://openlibrary.org/books/OL1097706M>.

- [11] Lennart Ljung. *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986. ISBN: 0-138-81640-9.
- [12] Lennart Ljung. *System Identification*. Tech. rep. LiTH-ISY-R2809. Linköping University, The Institute of Technology, Department of Electrical Engineering, 2007, p. 65. URL: <http://www.diva-portal.org/smash/get/diva2:316967/FULLTEXT01.pdf>.
- [13] Tomas Schön. “Identification for Predictive Control : A Multiple Model Approach”. Linköping University, Department of Electrical Engineering, 2001, p. 87. URL: <http://www.diva-portal.org/smash/get/diva2:17245/FULLTEXT01.pdf>.
- [14] J. K. Huusom et al. “Tuning of methods for offset free MPC based on ARX model representations”. In: *Proceedings of the 2010 American Control Conference*. June 2010, pp. 2355–2360. DOI: 10.1109/ACC.2010.5530560.
- [15] Raymond T. Stefani. *Design of Feedback Control Systems*. 3rd. New York, NY, USA: Oxford University Press, Inc., 1993. ISBN: 0195143442.
- [16] D.Q. Mayne et al. “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36.6 (2000), pp. 789–814. ISSN: 0005-1098. DOI: [https://doi.org/10.1016/S0005-1098\(99\)00214-9](https://doi.org/10.1016/S0005-1098(99)00214-9). URL: <http://www.sciencedirect.com/science/article/pii/S0005109899002149>.
- [17] G. Pannocchia. “Offset-free tracking MPC: A tutorial review and comparison of different formulations”. In: *2015 European Control Conference (ECC)*. July 2015, pp. 527–532. DOI: 10.1109/ECC.2015.7330597.
- [18] F. Borrelli and M. Morari. “Offset free model predictive control”. In: *2007 46th IEEE Conference on Decision and Control*. Dec. 2007, pp. 1245–1250. DOI: 10.1109/CDC.2007.4434770.
- [19] Piotr Tatjewski. “Disturbance modeling and state estimation for offset-free predictive control with state-space process models”. In: *Applied Mathematics and Computer Science* 24 (2014), pp. 313–323.
- [20] D. Chmielewski and V. Manousiouthakis. “On constrained infinite-time linear quadratic optimal control”. In: *Systems & Control Letters* 29.3 (1996), pp. 121–129. ISSN: 0167-6911. DOI: [https://doi.org/10.1016/S0167-6911\(96\)00057-6](https://doi.org/10.1016/S0167-6911(96)00057-6). URL: <http://www.sciencedirect.com/science/article/pii/S0167691196000576>.
- [21] Pierre O. M. Sokaert and James B. Rawlings. “Feasibility issues in linear model predictive control”. In: *AIChE Journal* 45.8 (1999), pp. 1649–1659. DOI: 10.1002/aic.690450805. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aic.690450805>.
- [22] Huaguang Zhang and Derong Liu. *Fuzzy Modeling and Fuzzy Control*. Jan. 2006. ISBN: 978-0-8176-4491-8.
- [23] Rómulo Antão. *Type-2 Fuzzy Logic: Uncertain Systems’ Modeling and Control*. Jan. 2017. ISBN: 978-981-10-4632-2.
- [24] T. Takagi and M. Sugeno. “Fuzzy identification of systems and its applications to modeling and control”. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-15.1 (Jan. 1985), pp. 116–132. ISSN: 0018-9472. DOI: 10.1109/TSMC.1985.6313399.
- [25] M. Sugeno and T. Yasukawa. “A fuzzy-logic-based approach to qualitative modeling”. In: *IEEE Transactions on Fuzzy Systems* 1.1 (Feb. 1993), pp. 7–31. ISSN: 1063-6706. DOI: 10.1109/TFUZZ.1993.390281.

-
- [26] T. Warren Liao. “Clustering of time series data—a survey”. In: *Pattern Recognition* 38.11 (2005), pp. 1857–1874. ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2005.01.025>. URL: <http://www.sciencedirect.com/science/article/pii/S0031320305001305>.
- [27] T. Warren Liao. “A clustering procedure for exploratory mining of vector time series”. In: *Pattern Recognition* 40.9 (2007), pp. 2550–2562. ISSN: 0031-3203. DOI: <https://doi.org/10.1016/j.patcog.2007.01.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0031320307000155>.

Παράρτημα Α΄

Κώδικες MATLAB

Παράθεση Α΄.1 GlucoseInsulinModel.m

```
1 classdef GlucoseInsulinModel < handle
2
3     properties (Constant)
4         mg_per_iu = 0.0347;
5         joule_per_kcal = 4184;
6         kgdm2_per_min2_per_joule = 60^2 * 0.1^-2;
7
8         Ts_init = 7 * 24 * 60;
9     end
10
11    properties (SetAccess = protected)
12        mobi
13        results_archive
14    end
15
16    methods
17        %% Class constructor
18        % Create a new class instance, acting as a wrapper around MoBi's
19        % Glucose-Insulin Model. A preliminary initialization is carried
20        % out to ensure that the patient is at steady-state at time zero.
21        % The object can accept:
22        % (a) up to three predefined meals at various times, and
23        % (b) user-defined insulin doses given in real-time
24        function obj = GlucoseInsulinModel(Ts,varargin)
25            obj.mobi.simulation.Ts = Ts;
26            if nargin ~= 1
27                try
28                    assert(nargin <= 4)
29
30                    arg = [varargin{:}];
31                    assert(all(isfield(arg,{'id',...
32                        'Timp',...
33                        'energy_in_kcal',...
34                        'glu_in_g',...
35                        'vol_in_lt',...
```

```

36         'frac_solid'}}))
37
38         obj.mobi.mlevnts.nm = nargin - 1;
39         arg = struct2table(arg);
40         obj.mobi.mlevnts.tbl = sortrows(arg, 'Timp', 'ascend');
41     catch
42         obj.handleWarning('inc_def', '[0 meals]');
43         obj.mobi.mlevnts.nm = 0;
44     end
45 else
46     obj.mobi.mlevnts.nm = 0;
47 end
48
49 root_path = [fileparts(which('GlucoseInsulinModel.m')) filesep];
50 xml = [root_path 'GIM_T1DM_MPC.xml'];
51 initSimulation(xml, [], 'report', 'none');
52
53 init_struct = [];
54 [~, desc] = existsSpeciesInitialValue('*', 1, 'parameterType', 'readonly');
55 for i = 2 : length(desc)
56     is_formula = desc{i,6};
57     if ~is_formula
58         path = desc{i,2};
59         init_struct = initSpeciesInitialValue(init_struct, path, 'withWarning');
60     end
61 end
62
63 fileID = fopen([root_path 'GIM_T1DM_MPC.txt']);
64 out = textscan(fileID, '%s', 'Delimiter', '\n');
65 for i = 1 : length(out{1})
66     path = out{1}{i};
67     if ~isempty(path)
68         init_struct = initParameter(init_struct, path, 'withWarning');
69     end
70 end
71 fclose(fileID);
72 init_struct = initParameter(init_struct, 'GIM_T1DM|Organism|Weight', 'always');
73
74 initSimulation(xml, init_struct, 'report', 'none');
75 obj.mobi.simulation.is_properly_initialized = false;
76 obj.advance()
77
78 path = 'GIM_T1DM|SC Insulin Bolus|Application_1|ProtocolSchemaItem|
79         DosePerBodyWeight';
80 setRelativeParameter(1, path, 1);
81 end
82
83 %% Update insulin infusion values
84 % Set the amount of insulin in IU units to be infused next
85 function updateInsulinValues(obj, dose_in_iu)
86     dose_in_kg = dose_in_iu * obj.mg_per_iu * 1e-6;
87     dose_per_bw = dose_in_kg / obj.mobi.patient.bw;
88
89     obj.mobi.track_changes.ins.dose_per_bw = dose_per_bw;
90     obj.mobi.track_changes.ins.user_input_given = true;

```

```

90     end
91
92     %% Advance one time interval ahead
93     % Simulate one time step into the future and store the results
94     function advance(obj)
95         obj.trackChanges()
96         processSimulation(1);
97         obj.initializeNextTimeStep()
98
99         path_ins = 'GIM_T1DM|Organism|VenousBlood|Plasma|Insulin|Conc.';
100        path_glu = 'GIM_T1DM|Organism|VenousBlood|Plasma|Glucose|Conc.';
101
102        if obj.mobi.simulation.is_properly_initialized == false
103            obj.mobi.patient.bw = getParameter('GIM_T1DM|Organism|Weight',1);
104
105            obj.mobi.simulation.units.ins = getObserverFormula(path_ins,1,'property','unit'
106                );
107            obj.mobi.simulation.units.glu = getObserverFormula(path_glu,1,'property','unit'
108                );
109
110            obj.mobi.simulation.is_properly_initialized = true;
111            return
112        end
113
114        [~,blood_ins] = getSimulationResult(path_ins,1);
115        [Trel,blood_glu] = getSimulationResult(path_glu,1);
116
117        Tabs = Trel + obj.mobi.track_changes.Tabs;
118
119        ins_dose_per_bw = obj.mobi.track_changes.ins.vals(2);
120        ins_dose_in_kg = ins_dose_per_bw * obj.mobi.patient.bw;
121        ins_dose_in_iu = ins_dose_in_kg / obj.mg_per_iu / 1e-6;
122        infused_ins = repmat(ins_dose_in_iu,1,length(Trel));
123
124        tbl_res = table(Tabs',Trel',infused_ins',blood_ins',blood_glu');
125        tbl_res.Properties.VariableNames = {'Tabs','Trel','infused_ins','blood_ins','
126            blood_glu'};
127        obj.results_archive{end + 1} = tbl_res;
128    end
129
130    %% Return a subset of available results
131    % Output subcutaneously infused insulin and venous blood glucose
132    % values along a user-defined time horizon into the past, sorted
133    % from latest to earliest. More precisely, there can be output:
134    % (a) infused insulin values up to the previous time step:
135    %      $(t_{\text{now}} - T_{\text{s}})$ ,  $(t_{\text{now}} - 2T_{\text{s}})$ , \dots, and
136    % (b) venous blood glucose values up to the current timepoint:
137    %      $t_{\text{now}}$ ,  $(t_{\text{now}} - T_{\text{s}})$ ,  $(t_{\text{now}} - 2T_{\text{s}})$ , \dots
138    function out = returnResults(obj,ny,nu)
139        tbl_res = vertcat(obj.results_archive{:});
140        [~,ia] = unique(tbl_res(:, 'Tabs'), 'last');
141        tbl_res = tbl_res(ia,:);
142
143    try
144        out.y = table2array(tbl_res(:, 'blood_glu'));

```

```

142         out.y = out.y(end : -obj.mobi.simulation.Ts : (end - obj.mobi.simulation.Ts * (
            ny - 1)));
143
144         out.v = table2array(tbl_res(:, 'blood_ins'));
145         out.v = out.v(end : -obj.mobi.simulation.Ts : (end - obj.mobi.simulation.Ts * (
            nu - 1)));
146
147         out.u = table2array(tbl_res(:, 'infused_ins'));
148         out.u = out.u(end : -obj.mobi.simulation.Ts : (end - obj.mobi.simulation.Ts * (
            nu - 1)));
149         out.u = out.u(2 : end);
150     catch
151         obj.handleWarning('inc');
152         out = [];
153     end
154 end
155
156 %% Print all results collected so far
157 % Stitch together the results stored on a per-time interval basis
158 % and then plot the evolution of insulin, glucose or both from the
159 % start of the simulation up to the current point in time. The user
160 % has the option to provide one of three input arguments:
161 % (a) 'ins', plot insulin concentration in bloodstream
162 % (b) 'glu', plot glucose concentration in bloodstream
163 % (c) 'mpc' (Default option), plot insulin amount infused per time
164 % step, along with insulin and glucose
165 % concentration in bloodstream
166 function printAllResults(obj,varargin)
167     tbl_res = vertcat(obj.results_archive{:});
168     [~,ia] = unique(tbl_res(:, 'Tabs'), 'last');
169     tbl_res = tbl_res(ia,:);
170
171     if nargin ~= 1
172         try
173             assert(nargin == 2)
174
175             arg = varargin{:};
176             assert(any(strcmp(arg,{'ins', 'glu', 'mpc'})))
177         catch
178             obj.handleWarning('inc_def', ['''mpc''']);
179             arg = 'mpc';
180         end
181     else
182         arg = 'mpc';
183     end
184
185     x = tbl_res.Tabs;
186     y1 = tbl_res.blood_ins;
187     y2 = tbl_res.blood_glu;
188     y3 = tbl_res.infused_ins;
189
190     figure(2)
191
192     switch arg
193         case 'ins'

```



```

194         plot(x,y1)
195
196         xlabel('Time [min]')
197         ylabel({'Venous blood insulin';strcat('[',obj.mobi.simulation.units.ins,']'
198         )})
198     case 'glu'
199         plot(x,y2)
200
201         xlabel('Time [min]')
202         ylabel({'Venous blood glucose';strcat('[',obj.mobi.simulation.units.glu,']'
203         )})
203     otherwise
204         ax1 = subplot(2,1,1);
205         p = get(ax1, 'pos');
206         delete(ax1)
207
208         sp = (70 + (140 - 70) / 3) / 180.16 * 10^4;
209         plot(x, repmat(sp, length(x), 1), 'r:');
210         hold on
211
212         [ax1,ln1,ln2] = plotyy(x,y2,x,y3, 'plot', 'stairs');
213         set(ax1, 'pos', p)
214
215         ax1(1).YAxis.Color = 'm';
216         ax1(2).YAxis.Color = 'r';
217         ylabel(ax1(1),{'Venous blood glucose';strcat('[',obj.mobi.simulation.units.
218         glu,']')})
219         ylabel(ax1(2),{'Subcutaneously infused insulin';'[IU per sampling interval]
220         '})
221
222         ln1.LineStyle = '-';
223         ln1.Color = 'm';
224         ln2.LineStyle = '--';
225         ln2.Color = 'r';
226
227         ax2 = subplot(2,1,2);
228         plot(ax2,x,y1, '-b')
229
230         ax2.YAxis.Color = 'b';
231         xlabel(ax2, 'Time [min]')
232         ylabel(ax2,{'Venous blood insulin';strcat('[',obj.mobi.simulation.units.ins
233         ,']')})
234
235     end
236 end
237 end
238
239 methods (Access = protected)
240
241     %% Set administered meal values
242     % Initialize the simulation with the proper meal event values
243     function setMealValues(obj)
244         path_ref = 'GIM_T1DM|OGTT|MealD|Application_1|ProtocolSchemaItem';
245         path_prm = {'Active',...
246         'DosePerBodyWeight',...
247         'Volume of water/body weight',...
248         'Start time'};

```

```

244
245     for i = 1 : 4
246         path = sprintf('%s%s',path_ref,path_prm{i});
247         val = obj.mobi.track_changes.mlevnts.glu.vals(i);
248         setParameter(val,path,1);
249     end
250
251     path_ref = 'GIM_T1DM|Events|Meal Effect_%d|';
252     path_prm = {'Active',...
253               'Meal energy content',...
254               'Meal fraction solid',...
255               'Meal volume',...
256               'Meal|Start time',...
257               'Relative_time'};
258
259     for i = 1 : 3
260         path_ref_tmp = sprintf(path_ref,i);
261         for j = 1 : 6
262             path = sprintf('%s%s',path_ref_tmp,path_prm{j});
263             val = obj.mobi.track_changes.mlevnts.effect.vals(i,j);
264             setParameter(val,path,1);
265         end
266     end
267 end
268
269 %% Set insulin infusion values
270 % Initialize the simulation with the proper insulin infusion values
271 function setInsulinValues(obj)
272     path_ref = 'GIM_T1DM|SC Insulin Bolus|Application_2|ProtocolSchemaItem|';
273     path_prm = {'Active',...
274               'DosePerBodyWeight',...
275               'Infusion time',...
276               'Start time'};
277
278     for i = 1 : 4
279         path = sprintf('%s%s',path_ref,path_prm{i});
280         val = obj.mobi.track_changes.ins.vals(i);
281         setParameter(val,path,1);
282     end
283 end
284
285 %% Track simulation changes
286 % Keep track of absolute simulation time and make sure that:
287 % (a) Insulin infusion values are set correctly:
288 %     (i) To the user-supplied value or,
289 %     (ii) To zero, when no user input has been given and,
290 % (b) Meal events are applied when they are supposed to:
291 %     (i) Glucose is administered orally once per meal while,
292 %     (ii) Meal effects are taken into account in a continuous,
293 %     uninterrupted fashion ever after a meal has been eaten
294 function trackChanges(obj)
295     if obj.mobi.simulation.is_properly_initialized == false
296         obj.mobi.track_changes.Tabs = -obj.mobi.simulation.Ts;
297         timepoints = 0 : obj.Ts_init;
298

```

```

299     obj.mobi.track_changes.ins.vals = zeros(1,4);
300     obj.mobi.track_changes.ins.user_input_given = false;
301
302     obj.mobi.track_changes.mlevnts.glu.vals = zeros(1,4);
303     obj.mobi.track_changes.mlevnts.effect.vals = zeros(3,6);
304     obj.mobi.track_changes.mlevnts.effect.pntr_row = 1;
305     else
306     obj.mobi.track_changes.Tabs = obj.mobi.track_changes.Tabs + obj.mobi.simulation
        .Ts;
307     timepoints = 0 : obj.mobi.simulation.Ts;
308
309     if obj.mobi.track_changes.ins.user_input_given == true
310         ins_dose_per_bw = obj.mobi.track_changes.ins.dose_per_bw;
311         obj.mobi.track_changes.ins.user_input_given = false;
312     else
313         ins_dose_per_bw = 0;
314     end
315     obj.mobi.track_changes.ins.vals = [1,...
316         ins_dose_per_bw,...
317         obj.mobi.simulation.Ts,...
318         0];
319
320     if obj.mobi.mlevnts.nm ~= 0
321         pntr_row = obj.mobi.track_changes.mlevnts.effect.pntr_row;
322         if pntr_row >= 2
323             for i = 1 : (pntr_row - 1)
324                 obj.mobi.track_changes.mlevnts.effect.vals(i,6) = obj.mobi.
                    track_changes.Tabs;
325             end
326         end
327
328         rows = obj.mobi.mlevnts.tbl.Timp >= obj.mobi.track_changes.Tabs &...
329             obj.mobi.mlevnts.tbl.Timp < (obj.mobi.track_changes.Tabs + obj.mobi.
                    simulation.Ts);
330         tbl = obj.mobi.mlevnts.tbl(rows,:);
331         if height(tbl) == 0
332             obj.mobi.track_changes.mlevnts.glu.vals = zeros(1,4);
333         else
334             glu_in_kg = tbl.glu_in_g * 1e-3;
335             glu_per_bw = glu_in_kg / obj.mobi.patient.bw;
336
337             h2o_vol_in_lt = tbl.vol_in_lt * (1 - tbl.frac_solid);
338             h2o_vol_per_bw = h2o_vol_in_lt / obj.mobi.patient.bw;
339
340             Timp_rel = tbl.Timp - obj.mobi.track_changes.Tabs;
341
342             obj.mobi.track_changes.mlevnts.glu.vals = [1,...
343                 glu_per_bw,...
344                 h2o_vol_per_bw,...
345                 Timp_rel];
346
347             energy_in_joule = obj.joule_per_kcal * tbl.energy_in_kcal;
348             energy_in_kgdm2_per_min2 = energy_in_joule * obj.
                    kgdm2_per_min2_per_joule;
349

```

```

350         obj.mobi.track_changes.mlevnts.effect.vals(pntr_row,:) = [1,...
351             energy_in_kgdm2_per_min2,...
352             tbl.frac_solid,...
353             tbl.vol_in_lt,...
354             tbl.Timp,...
355             obj.mobi.track_changes.Tabs];
356
357         obj.mobi.track_changes.mlevnts.effect.pntr_row = pntr_row + 1;
358     end
359 end
360 end
361
362     setSimulationTime(timepoints,1);
363     obj.setMealValues()
364     obj.setInsulinValues()
365 end
366 end
367
368 methods (Static, Access = protected)
369     %% Initialize the next time step to be simulated
370     % Sort through all parameters and only select those whose value
371     % does not depend on a function. Then copy the value they possess
372     % at the end of the previous time interval and set it as starting
373     % value before proceeding to the next time step
374     function initializeNextTimeStep()
375         [~,desc] = existsSpeciesInitialValue('*',1);
376         for i = 2 : length(desc)
377             is_formula = desc{i,6};
378             if ~is_formula
379                 path = desc{i,2};
380                 [~,vals] = getSimulationResult(path,1);
381                 val = vals(end);
382                 setSpeciesInitialValue(val,path,1);
383             end
384         end
385     end
386
387     %% Handle errors that occur
388     % Print an appropriately formatted warning when an error is caught
389     function handleWarning(arg,varargin)
390         switch arg
391             case 'inc_def'
392                 def = varargin{:};
393                 msg = sprintf('Assigning default value of: %s, instead.',def);
394             case 'inc'
395                 msg = 'Method execution terminated.';
396         end
397         warning('Incorrect input argument given. %s',msg)
398     end
399 end
400
401 end

```

```

1  classdef ModelPredictiveController < handle
2
3      properties (SetAccess = protected)
4          sys
5          mpc
6      end
7
8      methods
9
10         %% Class constructor
11         % Create a new class instance, enclosing both the system in
12         % state-space formulation and the Model Predictive Controller,
13         % which is capable of offset-free set-point tracking
14         function obj = ModelPredictiveController(sys,d_md1,q,r,p_hrz,sp,lb,ub)
15             obj.sys = sys;
16             obj.sys.nx = size(sys.A,1);
17             obj.sys.nu = size(sys.B,2);
18             obj.sys.ny = size(sys.C,1);
19             obj.sys.nz = obj.sys.nu + obj.sys.nx;
20
21             obj.mpc.Bd = d_md1.Bd;
22             obj.mpc.Cd = d_md1.Cd;
23
24             obj.mpc.R = r * eye(obj.sys.nu);
25             obj.mpc.Q = q * eye(obj.sys.nx);
26             [obj.mpc.P,~] = dare(obj.sys.A,obj.sys.B,obj.mpc.Q,obj.mpc.R);
27
28             obj.mpc.p_hrz = p_hrz;
29             obj.mpc.sp = sp;
30
31             tmplt = [1;zeros((obj.sys.nx - 1),1)];
32             obj.mpc.lb.x = lb.x * tmplt;
33             obj.mpc.ub.x = ub.x * tmplt;
34
35             obj.mpc.lb.u = lb.u * ones(obj.sys.nu,1);
36             obj.mpc.ub.u = ub.u * ones(obj.sys.nu,1);
37         end
38
39         %% Update past simulation values
40         % Recede the horizon one time interval ahead
41         function updatePastValues(obj,x0,dk)
42             obj.mpc.x0 = x0;
43             obj.mpc.dk = dk;
44         end
45
46         %% Change constraint values
47         % Set new constraints for system states and inputs
48         function changeConstraints(obj,lb,ub)
49             tmplt = [1;zeros((obj.sys.nx - 1),1)];
50             obj.mpc.lb.x = lb.x * tmplt;
51             obj.mpc.ub.x = ub.x * tmplt;
52
53             obj.mpc.lb.u = lb.u * ones(obj.sys.nu,1);
54             obj.mpc.ub.u = ub.u * ones(obj.sys.nu,1);
55         end

```

```

56     %% Change process model
57     % Update the model used for predicting system evolution
58     function changeModel(obj,sys,d_md1)
59         obj.sys = sys;
60         obj.sys.nx = size(sys.A,1);
61         obj.sys.nu = size(sys.B,2);
62         obj.sys.ny = size(sys.C,1);
63         obj.sys.nz = obj.sys.nu + obj.sys.nx;
64
65         obj.mpc.Bd = d_md1.Bd;
66         obj.mpc.Cd = d_md1.Cd;
67
68         [obj.mpc.P,~] = dare(obj.sys.A,obj.sys.B,obj.mpc.Q,obj.mpc.R);
69     end
70
71     %% Advance one time interval forward
72     % Return the next best move
73     function out = advance(obj)
74         obj.formulateQuadProgCostFunction()
75         obj.formulateQuadProgEqualityConstraints()
76         obj.formulateQuadProgInequalityConstraints()
77
78         blk_e = eye(obj.mpc.p_hrz * obj.sys.nx);
79         blk_e = blkdiag(blk_e,-blk_e);
80
81         obj.mpc.H = blkdiag(obj.mpc.H,1e-1 * eye(size(blk_e,1)));
82         obj.mpc.f = [obj.mpc.f;zeros(size(blk_e,1),1)];
83
84         obj.mpc.A = [[obj.mpc.A,[blk_e;zeros((2 * obj.mpc.p_hrz * obj.sys.nu),size(blk_e,1)
85             )]];...
86             [zeros(size(blk_e,1),size(obj.mpc.A,2)),-eye(size(blk_e,1))]];
87         obj.mpc.b = [obj.mpc.b;zeros(size(blk_e,1),1)];
88
89         obj.mpc.Aeq = [obj.mpc.Aeq,zeros(size(obj.mpc.Aeq,1),size(blk_e,1))];
90
91         out_xu = quadprog(obj.mpc.H,obj.mpc.f,obj.mpc.A,obj.mpc.b,obj.mpc.Aeq,obj.mpc.beq);
92         out_u = out_xu((obj.sys.nx * obj.mpc.p_hrz + 1) : (end - 1));
93         out = out_u(1:obj.sys.nu);
94     end
95
96     methods (Access = protected)
97         %% Formulate the quadratic cost function
98         % Define Hessian matrix $H$ and vector $f$ that are plugged into:
99         %  $\displaystyle \min_x \frac{1}{2} x^{\text{top}} H x + f^{\text{top}} x$ 
100         function formulateQuadProgCostFunction(obj)
101             tmplt = eye(obj.mpc.p_hrz - 1);
102             blk_x = kron(tmplt,obj.mpc.Q);
103             blk_xp = blkdiag(blk_x,obj.mpc.P);
104
105             tmplt = eye(obj.mpc.p_hrz);
106             blk_u = kron(tmplt,obj.mpc.R);
107
108             obj.mpc.H = 2 * blkdiag(blk_xp,blk_u);
109

```

```

110     As = [(obj.sys.A - eye(obj.sys.nx)), obj.sys.B; obj.sys.C, zeros(obj.sys.ny, obj.sys.nu
111           )];
112     Bs = [(-obj.mpc.Bd * obj.mpc.dk); (obj.mpc.sp - obj.mpc.Cd * obj.mpc.dk)];
113     X = linsolve(As, Bs);
114     xs = X(1:obj.sys.nx);
115     us = X((obj.sys.nx + 1):end);
116
117     f_x = obj.mpc.Q * xs;
118     f_p = obj.mpc.P * xs;
119     f_u = obj.mpc.R * us;
120
121     blk_x = repmat(f_x, (obj.mpc.p_hrz - 1), 1);
122     blk_xp = [blk_x; f_p];
123     blk_u = repmat(f_u, obj.mpc.p_hrz, 1);
124     obj.mpc.f = -2 * [blk_xp; blk_u];
125
126     end
127
128     %% Formulate equality constraints
129     % Define matrix $A_{eq}$ and vector $b_{eq}$ that are plugged into:
130     % $A_{eq} \cdot x = b_{eq}$
131     function formulateQuadProgEqualityConstraints(obj)
132         tmplt = eye(obj.mpc.p_hrz - 1);
133         blk_A = kron(tmplt, -obj.sys.A);
134         blk_A = padarray(blk_A, [obj.sys.nx, 0], 'pre');
135         blk_A = padarray(blk_A, [0, obj.sys.nx], 'post');
136
137         blk_eye = eye(obj.sys.nx * obj.mpc.p_hrz);
138         blk_x = blk_A + blk_eye;
139
140         tmplt = eye(obj.mpc.p_hrz);
141         blk_u = kron(tmplt, -obj.sys.B);
142
143         obj.mpc.Aeq = [blk_x, blk_u];
144
145         blk = repmat((obj.mpc.Bd * obj.mpc.dk), (obj.mpc.p_hrz - 1), 1);
146         obj.mpc.beq = [(obj.mpc.Bd * obj.mpc.dk + obj.sys.A * obj.mpc.x0); blk];
147     end
148
149     %% Formulate inequality constraints
150     % Define matrix $A$ and vector $b$ that are plugged into:
151     % $A \cdot x \leq b$
152     function formulateQuadProgInequalityConstraints(obj)
153         tmplt = eye(obj.mpc.p_hrz);
154         blk_x = [1; zeros((obj.sys.nx - 1), 1)];
155         blk_x = diag(blk_x);
156         blk_x = kron(tmplt, blk_x);
157         blk_u = eye(obj.sys.nu * obj.mpc.p_hrz);
158         obj.mpc.A = blkdiag([-blk_x; blk_x], [-blk_u; blk_u]);
159
160         blk_lb_x = repmat(obj.mpc.lb.x(1), obj.mpc.p_hrz, 1);
161         blk_ub_x = repmat(obj.mpc.ub.x(1), obj.mpc.p_hrz, 1);
162
163         blk_lb_u = repmat(obj.mpc.lb.u, obj.mpc.p_hrz, 1);
164         blk_ub_u = repmat(obj.mpc.ub.u, obj.mpc.p_hrz, 1);

```

```

164         obj.mpc.b = [[-blk_lb_x; blk_ub_x]; [-blk_lb_u; blk_ub_u]];
165     end
166 end
167
168 end

```

Παράθεση Α'3 FuzzyPatientModel.m

```

1  classdef FuzzyPatientModel < handle
2
3      properties (SetAccess = protected)
4          sys
5          mdl_array
6          data_io
7      end
8
9      methods
10         function obj = FuzzyPatientModel(mf, r1, mdl_arx_cell_array, ys, us)
11             obj.sys.mf = mf;
12             obj.sys.r1 = r1;
13             [obj.sys.nmf, obj.sys.nr1] = size(r1);
14             obj.sys.nfs = cellfun('length', mf);
15
16             obj.mdl_array.arx = mdl_arx_cell_array;
17             obj.convertARXtoStateSpace(obj.mdl_array.arx)
18
19             obj.data_io.y = repmat(ys, [obj.mdl_array.n_max, 1]);
20             obj.data_io.u = repmat(us, [obj.mdl_array.n_max, 1]);
21         end
22
23         function advanceIOVectors(obj, y_new, u_new)
24             obj.data_io.y = circshift(obj.data_io.y, 1);
25             obj.data_io.y(1) = y_new;
26             obj.data_io.u = circshift(obj.data_io.u, 1);
27             obj.data_io.u(1) = u_new;
28         end
29
30         function out = returnCrispModel(obj, x)
31             rule_fire_strength = obj.evaluateRuleFireStrength(x);
32             rule_fire_strength_cell = num2cell(rule_fire_strength);
33
34             mdl_local_weighted = cell(6, 1);
35             for i = 1 : 6
36                 mdl_local = obj.mdl_array.state_space_cell(i, :);
37                 mdl_local_weighted{i} = gmultiply(mdl_local, rule_fire_strength_cell);
38             end
39
40             out.mdl.A = sum(cat(3, mdl_local_weighted{1}{:}), 3);
41             out.mdl.B = sum(cat(3, mdl_local_weighted{2}{:}), 3);
42             out.mdl.C = sum(cat(3, mdl_local_weighted{3}{:}), 3);
43             out.mdl.D = sum(cat(3, mdl_local_weighted{4}{:}), 3);
44
45             out.mdl.dstrb.Bd = sum(cat(3, mdl_local_weighted{5}{:}), 3);
46             out.mdl.dstrb.Cd = sum(cat(3, mdl_local_weighted{6}{:}), 3);
47

```



```

48     y_part = zeros(obj.mdl_array.n_max);
49     u_part = y_part;
50     y_part(1,1) = 1;
51     for i = 2 : obj.mdl_array.n_max
52         y_part(i,(2 : (obj.mdl_array.n_max - i + 2))) = out.mdl.A((i : end),1);
53         u_part(i,(1 : (obj.mdl_array.n_max - i + 1))) = out.mdl.B(i : end);
54     end
55
56     out.x = y_part * obj.data_io.y + u_part * obj.data_io.u;
57 end
58 end
59
60 methods (Access = protected)
61     function convertARXtoStateSpace(obj,mdl_cell_array)
62         func = @(x) max(length(x.a),length(x.b)) - 1;
63         n_max = max(cellfun(func,mdl_cell_array));
64         obj.mdl_array.n_max = n_max;
65
66         out = struct('A',[],'B',[],'C',[],'D',[],'Bd',[],'Cd',[]);
67         for i = 1 : obj.sys.nrl
68             mdl_local = mdl_cell_array{i};
69             coef_a = mdl_local.a(2 : end);
70             coef_b = mdl_local.b(2 : end);
71
72             nx = length(coef_a);
73             nu = length(coef_b);
74
75             A_lft = -coef_a';
76             A_lft = padarray(A_lft,[(n_max - nx),0],'post');
77             A_rgt = [eye(n_max - 1);zeros(1,(n_max - 1))];
78             out(i).A = [A_lft,A_rgt];
79
80             B = coef_b';
81             out(i).B = padarray(B,[n_max - nu,0],'post');
82
83             C = zeros(1,n_max);
84             C(1) = 1;
85             out(i).C = C;
86             out(i).D = 0;
87
88             out(i).Bd = out(i).A(:,1);
89             out(i).Cd = 1;
90         end
91         obj.mdl_array.state_space = out;
92         obj.mdl_array.state_space_cell = squeeze(struct2cell(obj.mdl_array.state_space));
93     end
94
95     function out = evaluateRuleFireStrength(obj,x)
96         val_mf = zeros(obj.sys.nmf,obj.sys.nrl);
97         for i = 1 : obj.sys.nmf
98             val_x = x(i);
99             mf = obj.sys.mf{i};
100            for j = 1 : obj.sys.nrl
101                pntn_fs = obj.sys.rl(i,j);
102                if pntn_fs ~= -1

```

```

103         val_mf(i,j) = evalmf(val_x,mf(pntr_fs).params,mf(pntr_fs).type);
104     else
105         val_mf(i,j) = 1;
106     end
107 end
108 end
109 out_AND = prod(val_mf,1);
110 out_OR = sum(val_mf,1);
111 out = [out_OR(1) out_AND(2 : end)];
112 out = out / sum(out);
113 end
114 end
115
116 end

```

Παράθεση Α'4 simulateMPC_FUZZY.m

```

1 clear
2 clc
3
4 load('fuzzy_init_vars.mat')
5 ys = 5829.6;
6 us = 0;
7 fuzzy = FuzzyPatientModel(mf_cell_array,r1_array,mdl_arx_cell_array,ys,us);
8
9 kappa = 10;
10 y_prev = ys;
11 y_avg = ys;
12 fuzzy_out = fuzzy.returnCrispModel([ys,y_avg]);
13
14 q = 1;
15 r = 1e6;
16 p_hrz = 6 * 60 / 5;
17
18 sp = (70 + (140 - 70) / 3) / 180.16 * 10^4;
19
20 lb.x = 70 / 180.16 * 10^4;
21 ub.x = 180 / 180.16 * 10^4;
22
23 lb.u = 0;
24 ub.u = 0.15 * 5;
25
26 mpc = ModelPredictiveController(fuzzy_out.mdl,fuzzy_out.mdl_dstrb,q,r,p_hrz,sp,lb,ub);
27
28 Ts = 5;
29 meal1 = struct('id','glass of orange juice',...
30     'Timp',60,...
31     'energy_in_kcal',(2.5 * 50),...
32     'glu_in_g',(22.5 / 4),...
33     'vol_in_lt',0.25,...
34     'frac_solid',0);
35
36 meal2 = struct('id','low glycemic meal',...
37     'Timp',120,...
38     'energy_in_kcal',690,...

```

```
39     'glu_in_g',50,...
40     'vol_in_lt',0.5,...
41     'frac_solid',0.8);
42
43 meal3 = struct('id','high glycemc meal',...
44     'Timp',180,...
45     'energy_in_kcal',1000,...
46     'glu_in_g',100,...
47     'vol_in_lt',0.59,...
48     'frac_solid',0.6);
49
50 mdl_patient = GlucoseInsulinModel(Ts,meal1,meal2,meal3);
51
52 x_new = fuzzy_out.x;
53 dk = 0;
54
55 i_bk = 13 * 60 / 5;
56 for i = 1 : i_bk
57     mpc.updatePastValues(x_new,dk)
58
59     ins_dose_in_iu = mpc.advance();
60     if ins_dose_in_iu < 0
61         ins_dose_in_iu = 0;
62     end
63
64     mdl_patient.updateInsulinValues(ins_dose_in_iu)
65     mdl_patient.advance()
66     out = mdl_patient.returnResults(1,0);
67
68     dy = (out.y - y_prev) / 2;
69     y_prev = out.y;
70
71     alpha = 1 / (1 + exp(-kappa * dy));
72     y_avg = (1 - alpha) * y_avg + alpha * out.y;
73     fuzzy_out = fuzzy.returnCrispModel([out.y,y_avg]);
74
75     mpc.changeModel(fuzzy_out.mdl,fuzzy_out.mdl_dstrb)
76     x = fuzzy_out.x;
77
78     x_new = fuzzy_out.mdl.A * x + fuzzy_out.mdl.B * ins_dose_in_iu;
79     y_new = fuzzy_out.mdl.C * x_new + fuzzy_out.mdl.D * ins_dose_in_iu;
80     dk = out.y - y_new;
81
82     fuzzy.advanceIOVectors(out.y,ins_dose_in_iu)
83
84     mdl_patient.printAllResults('mpc')
85     pause(0.5)
86 end
```


Παράρτημα Β΄

Παράμετροι MoBi και ARX μοντέλων

Πίνακας Β.1 Ατομικές παράμετροι ασθενή με διαβήτη τύπου Ι στο MoBi

Διαδρομή στο MoBi	Παράμετρος	Τιμή
Organism	TypeDM	1
Organism	S_G	1
Organism	S_R_I	1
Organism	S_R_G	0.7
Organism	S_R_N	1
Organism	S_I	1
Organism	S_N	1
Organism Liver Interstitial	Glucagon_0	$2.0 \times 10^{-5} \mu\text{mol l}^{-1}$
Organism Liver Intracellular	Glucose_0	$5750 \mu\text{mol l}^{-1}$
Organism Pancreas Interstitial	Insulin_0	$1.4 \times 10^{-4} \mu\text{mol l}^{-1}$
Organism Liver Intracellular	End_IR_0_IUS	$0.2 \mu\text{mol l}^{-1}$
Organism Fat Endosome	End_IR_0	$0.1 \mu\text{mol l}^{-1}$
Organism Liver Intracellular	End_IR_0_liv	$18 \mu\text{mol l}^{-1}$
Organism Muscle Endosome	End_IR_0	$0.4 \mu\text{mol l}^{-1}$
Organism Liver Intracellular	End_IRp_0_IUS	$0.02 \mu\text{mol l}^{-1}$
Organism Fat Endosome	End_IRp_0	$0.01 \mu\text{mol l}^{-1}$
Organism Liver Intracellular	End_IRp_0_liv	$1.8 \mu\text{mol l}^{-1}$

Organism Muscle Endosome	End_IRp_0	0.04 $\mu\text{mol l}^{-1}$
Organism Fat Intracellular	IR_cell_0	$1.0 \times 10^{-4} \mu\text{mol l}^{-1}$
Organism Liver Intracellular	IR_cell_0	$1.2 \times 10^{-2} \mu\text{mol l}^{-1}$
Organism Muscle Intracellular	IR_cell_0	$5.0 \times 10^{-4} \mu\text{mol l}^{-1}$
Organism Fat Intracellular	IR_p_0	$4.4 \times 10^{-6} \mu\text{mol l}^{-1}$
Organism Liver Intracellular	IR_p_0	$1.9 \times 10^{-4} \mu\text{mol l}^{-1}$
Organism Muscle Intracellular	IR_p_0	$1.1 \times 10^{-5} \mu\text{mol l}^{-1}$
Organism Liver	InternalizationFactor-End	3
Organism Muscle	InternalizationFactor	3
Organism Fat	InternalizationFactor	3
Organism Liver	InternalizationFactor	5
Organism Liver	ReceptorRecyclingFactor-End	50
Organism Muscle	ReceptorRecyclingFactor	7
Organism Fat	ReceptorRecyclingFactor	7
Organism Liver	ReceptorRecyclingFactor	20
Organism Pancreas Interstitial	Vm_0_GPNR	1.3
Organism Pancreas Interstitial	Vm_GPNR	-0.6
Organism Pancreas Interstitial	Km_GPNR	1.07
Organism Pancreas Interstitial	n_GPNR	3
Insulin	GFR fraction IR	10

$y(t) =$

$$\begin{aligned}
& 1.5042y(t-1) - 0.5025y(t-2) - 8.6456u(t-1) - 4.6176u(t-2) - 30.014u(t-3) \\
& - 46.379u(t-4) - 35.789u(t-5) - 20.934u(t-6) - 8.7309u(t-7) - 2.9791u(t-8) \\
& + 2.3487u(t-9) + 5.3162u(t-10) + 6.7182u(t-11) + 1.6642u(t-12) + 3.4764u(t-13) \\
& + 3.6358u(t-14) + 3.7229u(t-15) + 5.0724u(t-16)
\end{aligned}$$

(B'.1)

Εξίσωση B.1: ARX μοντέλο 'ss' για το σύστημα fuzzy λογικής

$$\begin{aligned}
y(t) = & 2.0411y(t-1) - 2.2852y(t-2) + 1.3206y(t-3) - 0.6244y(t-4) + 0.2372y(t-5) \\
& - 0.0993y(t-6) + 0.0732y(t-7) - 0.0467y(t-8) + 0.0892y(t-9) - 0.1351y(t-10) \\
& + 0.1077y(t-11) - 0.0635y(t-12) + 0.0375y(t-13) - 0.0171y(t-14) + 0.0109y(t-15) \\
& - 0.0115y(t-16) + 0.0062y(t-17) - 4.6442u(t-1) - 1.9740u(t-2) - 26.136u(t-3) \\
& - 23.272u(t-4) - 14.306u(t-5)
\end{aligned}
\tag{B'.2}$$

Εξίσωση B.2: ARX μοντέλο 'sm' για το σύστημα fuzzy λογικής

$$\begin{aligned}
y(t) = & 2.1569y(t-1) - 1.4960y(t-2) + 0.4278y(t-3) - 0.1096y(t-4) + 0.0212y(t-5) \\
& - 0.0365y(t-6) + 0.0234y(t-7) - 0.0216y(t-8) - 0.0095y(t-9) + 0.0459y(t-10) \\
& + 0.0273y(t-11) + 0.0910y(t-12) - 0.2193y(t-13) + 0.1003y(t-14) - 17.051u(t-1) \\
& + 17.326u(t-2) - 5.2617u(t-3) - 40.730u(t-4) - 23.947u(t-5) + 8.8920u(t-6) \\
& - 3.2531u(t-7) - 13.875u(t-8)
\end{aligned}
\tag{B'.3}$$

Εξίσωση B.3: ARX μοντέλο 'mm' για το σύστημα fuzzy λογικής

$$\begin{aligned}
y(t) = & 2.0685y(t-1) - 1.3050y(t-2) + 0.2921y(t-3) - 0.1025y(t-4) + 0.0752y(t-5) \\
& - 0.0358y(t-6) + 0.0103y(t-7) + 0.0110y(t-8) - 0.0111y(t-9) - 0.0090y(t-10) \\
& - 0.0101y(t-11) - 0.0236y(t-12) + 0.0203y(t-13) + 0.0210y(t-14) + 41.657u(t-1) \\
& - 1.4529u(t-2) - 32.742u(t-3) - 20.327u(t-4) - 16.226u(t-5) - 17.175u(t-6) \\
& + 2.7859u(t-7) - 35.836u(t-8)
\end{aligned}
\tag{B'.4}$$

Εξίσωση B.4: ARX μοντέλο 'bm' για το σύστημα fuzzy λογικής

