ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

Μοντελοποίηση και Έλεγχος Βιομηχανικών Συστημάτων

με χρήση Γενικευμένων Αυτομάτων

Διδακτορική Διατριβή:

Βασίλειος Δεληγιάννης

Εργαστήριο Γενικής Ηλεκτροτεχνίας

*Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών
Πολυτεχνική Σχολή Πανεπιστημίου Πατρών*

Αριθμός Διατριβής: 224
2009

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η παρούσα διδακτορική διατριβή με θέμα

«ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΛΕΓΧΟΣ ΒΙΟΜΗΧΑΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ»

του κ. Βασιλείου Δεληγιάννη, Διπλωματούχου Ηλεκτρολόγου Μηχανικού και Μηχανικού Τεχνολογίας Υπολογιστών, παρουσιάσθηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών την 19^η Μαΐου 2009, εξετάσθηκε και εγκρίθηκε από την ακόλουθη επιτροπή:

1. Ν. Κούσουλας, Καθηγητής Πανεπιστημίου Πατρών, Μέλος της τριμελούς Συμβουλευτικής Επιτροπής
2. Α.Τζέζ, Καθηγητής Πανεπιστημίου Πατρών, Μέλος της τριμελούς Συμβουλευτικής Επιτροπής
3. Ν. Ασπράγκαθος, Καθηγητής Πανεπιστημίου Πατρών
4. Γ. Μπιτσώρης, Καθηγητής Πανεπιστημίου Πατρών
5. Κ. Θραμπουλίδης, Αναπ. Καθηγητής Πανεπιστημίου Πατρών
6. Δ. Καζάκος, Επίκ. Καθηγητής Πανεπιστημίου Πατρών
7. Σ. Μάνεσης, Αναπ. Καθηγητής Πανεπιστημίου Πατρών, Επιβλέπων

Πάτρα 19 Μαΐου 2009

Ο Επιβλέπων

Αναπ. Καθηγητής Σταμάτης Μάνεσης

Ο Πρόεδρος του Τμήματος

Καθηγητής Αθανάσιος Σαφάκας

Περιεχόμενα

	Σελίδα
Πρόλογος	5
Περίληψη	7
Λεξιλόγιο	11
Λίστα σχημάτων	13
1. Εισαγωγή	17
1.1 Γενικά στοιχεία διατριβής	17
1.2 Γενική περιγραφή διατριβής	17
1.3 Νέα φορμαλιστική μέθοδος μοντελοποίησης	18
1.4 Εφαρμογές των Global Automata	19
1.5 Δομή κεφαλαίων διατριβής	21
2. Βιομηχανικά Συστήματα	23
2.1 Ολοκληρωμένη Παραγωγή με Υπολογιστές	24
2.2 Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLCs).	28
2.3 Δίκτυα Βιομηχανικού Αυτοματισμού	32
2.4 Συστήματα εποπτικού ελέγχου και συλλογής πληροφοριών SCADA	36
3. Μοντελοποίηση και έλεγχος βιομηχανικών συστημάτων	39
3.1 Η μοντελοποίηση βιομηχανικών συστημάτων στην βιβλιογραφία	43
3.2 Αυτόματη παραγωγή κώδικα για Προγραμματιζόμενους Λογικούς Ελεγκτές στην βιβλιογραφία	44
4. Θεωρία Αυτομάτων	49
4.1 Πεπερασμένα Αυτόματα (Finite Automata)	49
4.2 Υβριδικά Αυτόματα (Hybrid Automata)	51
4.3 Χρονισμένα Αυτόματα (Timed Automata)	52
4.4 PLC Αυτόματα	53
4.5 Λογισμικά πακέτα μοντελοποίησης Συστημάτων Διακριτών Γεγονότων	54
4.5.1 JGrafchart	54
4.5.2 UPPAAL	55
4.5.3 DiaGen	56
4.5.4 KRONOS	58
4.5.5 HyTech	58
4.5.6 Moby/PLC	60
4.5.7 Shift	60
4.5.8 Supremica	60
5. Γενικευμένα Αυτόματα (<i>Global Automata</i>)	63
5.1 Γενικευμένα Αυτόματα	63
5.2 Ιδιότητες Γενικευμένων Αυτομάτων.	66
5.3 Επαλήθευση μέσω σύγκρισης με υπάρχουσες μεθόδους	73
5.3.1 Σύγκριση με Χρονισμένα Αυτόματα	74
5.3.2 Σύγκριση με Υβριδικά Αυτόματα	75
5.3.2 Σύγκριση με CHARON	76
5.3.2 Σύγκριση με Ptolemy	76
5.3.2 Σύγκριση με Shift	77

6. Εργαλεία μοντελοποίησης στα γενικευμένα αυτόματα	79
6.1 Συγχώνευση καταστάσεων	79
6.1.1 Σύστημα τριών μηχανών με παύση αυτών υπό ειδικές συνθήκες.	81
6.2 Σύνθεση αυτομάτων	88
6.2.1 Σύστημα τριών δεξαμενών.	89
6.3 Ιεραρχία αυτομάτων	95
7. Εργαλεία υλοποίησης γενικευμένων αυτομάτων σε βιομηχανικά συστήματα	97
7.1 Οδηγός γραφής κώδικα για την εξομοίωση Γενικευμένων Αυτομάτων	98
7.1.1 Περιγραφή οδηγού γραφής κώδικα	99
7.1.2 Η εξομοίωση γενικευμένων αυτομάτων σε πλατφόρμα SCADA	102
7.1.3 Παραδείγματα χρήσης του οδηγού	104
7.1.4 Ένας οικιακός θερμοστάτης	104
7.1.5 Τοπικό δίκτυο υπολογιστών με πέρασμα κουπονιού	105
7.1.6 Εξομοίωση λειτουργίας τριών μηχανών με παύση αυτών υπό ειδικές συνθήκες	107
7.2 Οδηγός γραφής κώδικα για Προγραμματιζόμενους Λογικούς Ελεγκτές.	109
7.2.1 Οδηγός για προγραμματισμό σε IL, LAD και FBD.	112
7.2.2 Οδηγός για προγραμματισμό σε Structured Text.	116
7.2.3 Οδηγός για προγραμματισμό σε Sequential Function Chart.	120
7.2.4 Συστήματα με μοντέλο πολλαπλών αυτομάτων.	121
7.2.5 Παράδειγμα χρήσης οδηγού.	121
7.3 Εξομοίωση και έλεγχος με χρήση PLC και SCADA.	123
8. Εφαρμογές χρήσης γενικευμένων αυτομάτων	125
8.1 Μηχανή εσωτερικής καύσης	125
8.2 Μονάδα παραγωγής έτοιμου σκυροδέματος	131
8.2.1 Μοντελοποίηση και έλεγχος με χρήση Γενικευμένων Αυτομάτων	133
8.2.2 Σταθμός εποπτικού ελέγχου και συλλογής δεδομένων	137
8.3 Εργαστηριακή διάταξη τριών δεξαμενών.	138
9. Επίλογος – Εργασίες που προέκυψαν από το έργο	143
9.1 Επίλογος – Μελλοντική εργασία.	143
9.2 Εργασίες που προέκυψαν από το έργο.	144
10. Βιβλιογραφία	147

Πρόλογος

Η παρούσα διατριβή έλαβε χώρα την περίοδο 2002-2009 στο εργαστήριο Γενικής Ηλεκτροτεχνίας του τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών.

Το αντικείμενο μιας διδακτορικής διατριβής αποτελεί την έρευνα, κυρίως θεωρητική, σε ένα συγκεκριμένο, είτε σε στενά πλαίσια είτε όχι, επιστημονικό πεδίο. Το αντικείμενο της παρούσας διατριβής αποτέλεσε η μοντελοποίηση και ο έλεγχος των βιομηχανικών συστημάτων. Η αλήθεια είναι ότι η ελληνική βιομηχανία δεν επενδύει σημαντικά σε έρευνα, για τη δημιουργία πρωτοπόρων τεχνικών παραγωγής ή βελτίωσης των υπαρχόντων. Συνεπώς, η ερευνητική αυτή προσπάθεια έτυχε οικονομικής υποστήριξης μόνο από την Επιτροπή Ερευνών του Πανεπιστημίου Πατρών στα πλαίσια του προγράμματος Κ. Καραθεοδωρή.

Θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη συμπαράσταση όλα αυτά τα χρόνια που μου έδινε κουράγιο και δύναμη για να μπορέσω να ολοκληρώσω με επιτυχία το έργο αυτό. Καταλυτικό ρόλο στην εξέλιξη και ολοκλήρωση της διατριβής είχε πάντα ο κ. Σταμάτης Μάνεσης, επιβλέπων καθηγητής, και θα ήθελα να του εκφράσω τις βαθύτατες ευχαριστίες μου. Τέλος, θέλω επίσης να ευχαριστήσω όλους τους φίλους, συναδέλφους, και γενικά όλους τους ανθρώπους που μου στάθηκαν και με βοήθησαν, λίγο ως πολύ, στην προσπάθειά μου.

Περίληψη

Κύριος στόχος της παρούσας διατριβής ήταν να συμβάλλει στην συστηματοποίηση του έργου του Ηλεκτρολόγου Μηχανικού όταν αναλύει, μοντελοποιεί και σχεδιάζει τον έλεγχο ενός βιομηχανικού συστήματος.

Τα βιομηχανικά συστήματα περιγράφονται κυρίως ως Συστήματα Διακριτών Γεγονότων, ή ως Υβριδικά Συστήματα. Για την μοντελοποίηση τους έχουν προταθεί διάφορες μέθοδοι, όπως τα δίκτυα Petri και όλες οι μορφές αυτομάτων ελέγχου. Αντίθετα με τις προσδοκίες των επιστημόνων τα δίκτυα Petri δεν παρουσιάστηκαν ιδιαίτερα αποδοτικά στον έλεγχο κλειστού βρόχου, κυρίως λόγω του ασαφούς διαχωρισμού μεταξύ ελεγκτή και συστήματος. Εν αντιθέσει, τα αυτόματα παρουσιάζονται ως καταλληλότερο εργαλείο, και λόγω τούτου, αποτέλεσαν την βάση για την δημιουργία μιας νέας μεθόδου μοντελοποίησης στα πλαίσια της παρούσας διατριβής.

Τα Γενικευμένα Αυτόματα (Global Automata), όπως ονομάστηκε η νέα μέθοδος, δανείζονται χαρακτηριστικά από τις διάφορες προϋπάρχουσες μορφές αυτομάτων, ενώ εισαγάγουν νέες παραμέτρους μοντελοποίησης ώστε να καλύψουν τις όποιες ανάγκες των σύγχρονων βιομηχανικών συστημάτων. Τα βιομηχανικά συστήματα γίνονται ολοένα και πολυπλοκότερα ενώ οι ανάγκες για ευέλικτη και ποικιλόμορφη παραγωγή απαιτούν μια μέθοδο μοντελοποίησης ικανή να ανταπεξέλθει στις συνθήκες παρουσιάζοντας προσαρμοστικότητα και ευκολία επίτευξης αλλαγών. Τέλος είναι απαραίτητη η δυνατότητα υλοποίησης του μοντέλου σε ένα βιομηχανικού τύπου ελεγκτή, που στην πλειονότητα των περιπτώσεων είναι ένας Προγραμματιζόμενος Λογικός Ελεγκτής (PLC).

Τα γενικευμένα αυτόματα παρουσιάζουν ικανοποιητικά στοιχεία ως παραμέτρους μοντελοποίησης για την κάλυψη μεγάλου φάσματος βιομηχανικών διεργασιών, αλλά η θεωρητική μέθοδος από μόνη της δεν είναι αρκετή. Στη προσπάθεια ενίσχυσης της μεθόδου αναπτύχθηκαν διάφορα εργαλεία με στόχο την αύξηση της δύναμης

μοντελοποίησης της, δηλαδή της διεύρυνσης του πεδίου εφαρμογής της. Το πρώτο εργαλείο περιλαμβάνει μια συστηματοποιημένη τεχνική συγχώνευσης καταστάσεων ενός μοντέλου και αποσκοπεί στον περιορισμό της υπέρμετρης αύξησης του γράφου καταστάσεων, ο οποίος σε περιπτώσεις μεγάλων συστημάτων μπορεί να φτάσει τις δεκάδες χιλιάδες καταστάσεις. Είναι λοιπόν σαφές, ότι η συγχώνευση καταστάσεων για την δημιουργία πιο εύχρηστων μοντέλων είναι ένα πολύ χρήσιμο εργαλείο στα χέρια του μηχανικού ελέγχου. Το δεύτερο εργαλείο που αναπτύχθηκε, αποτελεί την σύνθεση αυτομάτων με στόχο την δημιουργία ενός νέου αυτομάτου που έχει την ίδια ακριβώς συμπεριφορά με τα αρχικά. Η σύνθεση επιτρέπει την δημιουργία μοντέλου ακόμα και ενός συστήματος υψηλής πολυπλοκότητας, μέσω σύνθεσης των μοντέλων όλων των υποσυστημάτων του, τα οποία προφανώς μπορούν να αναπτυχθούν ευκολότερα. Τέλος, τα γενικευμένα αυτόματα επιτρέπουν την δημιουργία ιεραρχικών μοντέλων που αποτελούνται από πλήθος αυτομάτων ιεραρχικά δομημένων. Σε αυτή την περίπτωση το αυτόματο του ανωτέρου επιπέδου έχει τον ρόλο του επόπτη-συντονιστή και τροφοδοτεί τα αυτόματα του κατώτερου επιπέδου με δεδομένα, τιμές αναφοράς κτλ.

Όπως προαναφέρθηκε, η μοντελοποίηση ενός συστήματος δεν επαρκεί, αναγκαίος είναι και ο τελικός προγραμματισμός των βιομηχανικών ελεγκτών που θα επωμιστούν τον έλεγχο της διεργασίας. Στην κατεύθυνση αυτή οδηγούν δύο εργαλεία παραγωγής κώδικα από ένα γενικευμένο αυτόματο. Το πρώτο εργαλείο παρουσιάζει συστηματικά τα βήματα μετατροπής ενός αυτομάτου σε εκτελέσιμο κώδικα μιας γλώσσας προγραμματισμού υψηλού επιπέδου. Η χρησιμότητα είναι σημαντική κατά την διάρκεια εξομοίωσης ενός μοντέλου και επαλήθευσης της ορθής λειτουργίας του ελεγκτή, αλλά όχι μόνο. Οι σύγχρονες βιομηχανίες είναι εξοπλισμένες με Σταθμούς Εποπτικού Έλεγχου και Συλλογής Πληροφοριών (SCADA) οι οποίοι στην πλειονότητά τους φέρουν ενσωματωμένη δυνατότητα εκτέλεσης κώδικα σε μια γλώσσα υψηλού επιπέδου. Με εκτέλεση του παραγόμενου, από το εργαλείο, κώδικα σε λογισμικό SCADA, μπορεί να επιτευχθεί και έλεγχος του συστήματος. Βέβαια, ο επικρατέστερος τύπος ελεγκτή παραμένει το Προγραμματιζόμενος Λογικός Ελεγκτής, λόγω των ιδιαίτερων χαρακτηριστικών που διαθέτει και υπερσχύει συγκριτικά με τις εναλλακτικές προσεγγίσεις. Αυτός ήταν πάντα και ο σκοπός της διατριβής, η υλοποίηση των γενικευμένων αυτομάτων σε PLC και επιτεύχθηκε μέσω του δεύτερου παρουσιαζόμενου εργαλείου. Το εργαλείο παραγωγής κώδικα για PLC δημιουργήθηκε με βάση το διεθνές πρότυπο IEC 61131-3 και περιγράφει όλα τα συστηματικά βήματα

που πρέπει να ακολουθήσει ο μηχανικός για να υλοποιήσει την στρατηγική ελέγχου, που εμπεριέχει ένα γενικευμένο αυτόματα, σε ένα PLC. Το πρότυπο IEC 61131-3 περιλαμβάνει ένα σύνολο πέντε γλώσσών για τον προγραμματισμό PLCs. Οι πλέον διαδεδομένες γλώσσες είναι οι *Instruction List (IL)*, *Ladder Diagram (LAD)* και *Function Block Diagram (FBD)*, οι οποίες υποστηρίζονται από το σύνολο των κατασκευαστών. Το πρότυπο επίσης περιλαμβάνει τις γλώσσες *Structured Text* και *Sequential Function Charts (SFC)* οι οποίες συγκαταλέγονται στις γλώσσες υψηλού επιπέδου αλλά δεν τυγχάνουν της υποστήριξης του συνόλου των κατασκευαστών. Το παρουσιαζόμενο εργαλείο σύνθεσης περιλαμβάνει όλες τις γλώσσες του προτύπου δίνοντας κατάλληλες κατευθύνσεις υλοποίησης με βάση τα ιδιαίτερα χαρακτηριστικά της καθεμιάς.

Η παρουσίαση των γενικευμένων αυτομάτων και των εργαλείων που αναπτύχθηκαν ολοκληρώνεται μέσω παραδειγμάτων χρήσης. Οι παρουσιαζόμενες εφαρμογές αποτελούν είτε θεωρητικά συστήματα για επεξήγηση, είτε πραγματικές εφαρμογές υλοποιημένες ως εργαστηριακές διατάξεις ή ως πραγματικά συστήματα παραγωγής. Μια εργαστηριακή διάταξη η οποία αποτελείται από τρεις δεξαμενές έχει μοντελοποιηθεί με χρήση γενικευμένων αυτομάτων και έχει ελεγχθεί με PLC που προγραμματίστηκε με βάση το εργαλείο παραγωγής κώδικα. Τέλος, μια μονάδα παραγωγής έτοιμου σκυροδέματος έχει μοντελοποιηθεί και ελεγχθεί με σταθμό SCADA ο οποίος προγραμματίστηκε με χρήση του αντίστοιχου εργαλείου.

Τα δύο εργαλεία παραγωγής κώδικα αποτελούν την βάση για την υιοθέτηση των γενικευμένων αυτομάτων στην βιομηχανία. Στην μελλοντική εργασία κυρίαρχο ρόλο κατέχει η δημιουργία ενός λογισμικού πακέτου, το οποίο να επιτρέπει στον χρήστη την εισαγωγή του αυτομάτου στον υπολογιστή με γραφικό τρόπο και την εξομίωση αυτού για την επαλήθευση της ορθής λειτουργίας του. Η ολοκλήρωση ενός τέτοιου λογισμικού θα ήταν η δυνατότητα αυτόματης παραγωγής εκτελέσιμου κώδικα για SCADA ή για PLC. Με αυτό τον τρόπο θα ελαχιστοποιηθεί ο χρόνος υλοποίησης ενός ελεγκτή, καθώς ο χειροκίνητος προγραμματισμός ενός PLC αποτελεί επίπονη και χρονοβόρα διαδικασία η οποία αναπόφευκτα δεν είναι πάντα αλάνθαστη. Όμως, στην περίπτωση μοντελοποίησης ενός συστήματος, ικανοποιητικής εξομίωσης αυτού και τέλος αυτόματης παραγωγής κώδικα, η πιθανότητα λάθους ελαχιστοποιείται.

Λεξιλόγιο

DCS: *Distributed Control System.* Κατανεμημένο Σύστημα Ελέγχου

DES: *Discrete Event System.* Σύστημα Διακριτών Γεγονότων

ERP: *Enterprise Resource Planning.*

FBD: *Function Block Diagram.*

IEC: *International Electrotechnical Commission.*

IL: *Instruction List.*

ISO: *International Organization for Standardization.*

LAD: *Ladder Diagram.*

MCC: *Motor Control Center.*

MES: *Manufacturing Execution System.*

MIS: *Management Information System.*

MTBF: *Mean Time Between Failure.*

MTTF: *Mean Time To Failure.*

MTTR: *Mean Time To Repair.*

OSI: *Open Systems Interconnection.*

PFD: *Probability of Failure on Demand.*

PLC: *Programmable Logic Controller.* Προγραμματιζόμενος Λογικός Ελεγκτής.

RTU: *Remote Terminal Unit.* Απομακρυσμένη Τερματική Μονάδα.

SCADA: *Supervisory Control And Data Acquisition.* Σταθμός Εποπτικού Ελέγχου και Συλλογής Δεδομένων.

SIL: *Safety Integrity Level.*

SFC: *Sequential Function Charts.*

ST: *Structured Text.*

TL: *Temporal Logic.*

UML: Unified Modeling Language

Λίστα Σχημάτων

Κεφάλαιο 2.

- Σχήμα 2.1. Ιεραρχική δομή CIM.
- Σχήμα 2.2. Ιεραρχική δομή CIM και αντίστοιχες συσκευές.
- Σχήμα 2.3. Ολοκληρωμένο Βιομηχανικό Δίκτυο επικοινωνίας.
- Σχήμα 2.4. Ολοκληρωμένο Βιομηχανικό Πληροφοριακό Σύστημα.
- Σχήμα 2.5 Η δομή ενός PLC.

Κεφάλαιο 4.

- Σχήμα 4.1. Πεπερασμένο αυτόματο.
- Σχήμα 4.2. Υβριδικό αυτόματο θερμοστάτη.
- Σχήμα 4.3. Ένα απλό PLC αυτόματο.
- Σχήμα 4.4. Screenshot από το λογισμικό πακέτο JGrafchart.
- Σχήμα 4.5. Screenshot από το λογισμικό πακέτο UPPAAL.
- Σχήμα 4.6. Screenshot από το λογισμικό πακέτο DiaGen.
- Σχήμα 4.7. Screenshots από το λογισμικό πακέτο DiaGen.
- Σχήμα 4.8. Screenshots από το λογισμικό πακέτο KRONOS.
- Σχήμα 4.9. Σχηματικά η δομή του λογισμικού πακέτου HyTech.
- Σχήμα 4.10. Screenshot από το λογισμικό πακέτο Moby/PLC.
- Σχήμα 4.11. Screenshot από το λογισμικό πακέτο Shift.
- Σχήμα 4.12. Screenshot από το λογισμικό πακέτο Supremica.

Κεφάλαιο 5.

- Σχήμα 5.1. Γενικευμένο αυτόματο.
- Σχήμα 5.2. Παράδειγμα γραμμικού αυτομάτου.
- Σχήμα 5.3. Παράδειγμα χρονικά αμετάβλητου αυτομάτου.
- Σχήμα 5.4. Παράδειγμα απεικόνισης συστήματος μεταβάσεων.
- Σχήμα 5.5. Σύστημα της πρέσας αλουμινίου.
- Σχήμα 5.6. Έλεγχος θερμοκρασίας στο σύστημα της πρέσας αλουμινίου.

Κεφάλαιο 6.

- Σχήμα 6.1. Αυτόματο με δύο καταστάσεις.
- Σχήμα 6.2. Αυτόματο μίας κατάστασης.
- Σχήμα 6.3. Διάγραμμα καταστάσεων του συστήματος των τριών μηχανών.
- Σχήμα 6.4. Μειωμένο διάγραμμα καταστάσεων με πέντε καταστάσεις.
- Σχήμα 6.5. Μειωμένο διάγραμμα καταστάσεων με τρεις καταστάσεις.
- Σχήμα 6.6. Αυτόματο μιας μόνο κατάστασης
- Σχήμα 6.7. Διάγραμμα τριών καταστάσεων για n μηχανές.
- Σχήμα 6.8. Διάγραμμα μιας κατάστασης για n μηχανές.
- Σχήμα 6.9. Σύστημα τριών δεξαμενών.
- Σχήμα 6.10. Το αυτόματο της πρώτης δεξαμενής.
- Σχήμα 6.11. Το αυτόματο της δεύτερης δεξαμενής.
- Σχήμα 6.12. Το αυτόματο της τρίτης δεξαμενής.
- Σχήμα 6.13. Το *αυτόματο*₁₂.
- Σχήμα 6.14. Το *αυτόματο*₁₂₃.
- Σχήμα 6.15. Αυτόματα μιας κατάστασης για τις δυο πρώτες δεξαμενές.
- Σχήμα 6.16. Το νέο *αυτόματο*₁₂.
- Σχήμα 6.17. Το τελικό αυτόματο πέντε καταστάσεων

Κεφάλαιο 7.

- Σχήμα 7.1. Οδηγός γραφής κώδικα για σταθμό SCADA.
- Σχήμα 7.2. Ο ελεγκτής στο SCADA.
- Σχήμα 7.3. Αυτόματο θερμοστάτη.
- Σχήμα 7.4. Εξομοίωση αυτομάτου.
- Σχήμα 7.5. Δίκτυο περάσματος κουπονιού με τέσσερις κόμβους.
- Σχήμα 7.6. Αυτόματο τριών μηχανών.
- Σχήμα 7.7. Έργο μηχανικού ελέγχου.
- Σχήμα 7.8. Σύνοψη εργαλείου.
- Σχήμα 7.9. Αρχικοποίηση μεταβλητών σε IL.
- Σχήμα 7.10. Επιλογή ενεργής κατάστασης με χρήση της δομής Jump to List.
- Σχήμα 7.11. Παράδειγμα συναρτήσεων σε IL και LAD.
- Σχήμα 7.12. Παράδειγμα μεταβάσεων FBD και LAD.
- Σχήμα 7.13. Δήλωση μεταβλητών και αρχικοποίηση σε ST.
- Σχήμα 7.14. Συναρτήσεις σε ST.
- Σχήμα 7.15. Μεταβάσεις σε ST.

- Σχήμα 7.16. Το μπλοκ OB1 σε SFC.
- Σχήμα 7.17. Δομή μετάβασης σε SFC.
- Σχήμα 7.18. Γενικευμένο αυτόματο ηλεκτρονικής κλειδαριάς.
- Σχήμα 7.19. Υλοποίηση ηλεκτρονικής κλειδαριάς σε IL.
- Σχήμα 7.20. Ο ελεγκτής στο PLC.

Κεφάλαιο 8.

- Σχήμα 8.1. Κύλινδρος μηχανής εσωτερικής καύσης.
- Σχήμα 8.2. Διάγραμμα πίεσης-όγκου για τους δύο αεροδυναμικούς κύκλους.
- Σχήμα 8.3. Διάγραμμα πίεσης-όγκου στο μοντέλο προσομοίωσης.
- Σχήμα 8.4. Γενικευμένο αυτόματο μηχανής.
- Σχήμα 8.5. Φυσικά μεγέθη κυλίνδρου
- Σχήμα 8.6. Αρχική οθόνη SCADA.
- Σχήμα 8.7. Οθόνη διαγραμμάτων.
- Σχήμα 8.8. Σχηματικό διάγραμμα του συστήματος
- Σχήμα 8.9. Διάγραμμα ροής του συστήματος.
- Σχήμα 8.10. Αυτόματο-Συνταγής
- Σχήμα 8.11. Αυτόματο ελέγχου αδρανών υλικών.
- Σχήμα 8.12. Αυτόματο ελέγχου τσιμέντων.
- Σχήμα 8.13. Αυτόματο ελέγχου πρόσθετων υλικών.
- Σχήμα 8.14. Αυτόματο ελέγχου αναμίκτη.
- Σχήμα 8.15. Η αρχική οθόνη του SCADA.
- Σχήμα 8.16. Εκτέλεση του αυτομάτου των πρόσθετων.
- Σχήμα 8.17. Σχηματική απεικόνιση διάταξης.
- Σχήμα 8.18. Φωτογραφία εργαστηριακής διάταξης.
- Σχήμα 8.19. Μοντέλο αυτομάτου της δεξαμενής 1.
- Σχήμα 8.20. Μοντέλο αυτομάτου της δεξαμενής 2.
- Σχήμα 8.21. Μοντέλο αυτομάτου της δεξαμενής 3.
- Σχήμα 8.22. STL κώδικας για την κατάσταση Q_{37} .

Κεφάλαιο 1^ο

Εισαγωγή

1.1 Γενικά στοιχεία διατριβής

Η παρούσα διδακτορική διατριβή εκπονήθηκε στο Εργαστήριο Γενικής Ηλεκτροτεχνίας του τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών από τον υποψήφιο διδάκτορα Βασίλειο Δεληγιάννη (Α.Μ. 432) από τον Σεπτέμβριο του 2002. Μέρος του ερευνητικού έργου υποστηρίχθηκε από το πρόγραμμα Κ. Καραθεοδωρή (2003-2006). Η τριμελής συμβουλευτική επιτροπή αποτελείται από τον κ. Σταμάτη Μάνεση, Αναπληρωτή Καθηγητή (επιβλέπων), τον κ. Νικόλαο Κούσουλα, Καθηγητή και τον κ. Αντώνιο Τζε, Καθηγητή, ενώ την επταμελή εξεταστική επιτροπή συμπληρώνουν ο κ. Νικόλαος Ασπράγκαθος, Καθηγητής, ο κ. Γεώργιος Μπισώρης, Καθηγητής, ο κ. Κλεάνθης Θραμπουλίδης, Αναπληρωτής Καθηγητής και ο κ. Δημοσθένης Καζάκος, Επίκουρος Καθηγητής.

1.2 Γενική περιγραφή διατριβής

Γενικά, μια σύγχρονη εργοστασιακή γραμμή παραγωγής αποτελείται από διάφορες συσκευές (αισθητήρες, ενεργοποιητές, μηχανές, ρομποτικούς βραχίονες κτλ.). Ο έλεγχος αυτών των συσκευών και μηχανών γίνεται είτε με χρήση αποκεντρωμένων τοπικών ελεγκτών, είτε με κεντρικό εποπτικό ελεγκτή. Τα βιομηχανικά συστήματα αντιμετωπίζονται συνήθως ως Συστήματα Διακριτών Γεγονότων (ΣΔΓ), καθώς η εξέλιξη ενός τέτοιου συστήματος οφείλεται σε στιγμιαίες αλλαγές των μεταβλητών της διεργασίας. Μια πιο ακριβής προσέγγιση είναι ο ορισμός τους ως Υβριδικά Συστήματα, καθώς κατά μήκος μιας γραμμής παραγωγής υπάρχουν τόσο διακριτές όσο και συνεχείς μεταβλητές. Η όλο και αυξανόμενη πολυπλοκότητα των σύγχρονων βιομηχανικών συστημάτων σε συνδυασμό με την απαίτηση για γρήγορες αποκρίσεις, έχουν δημιουργήσει την ανάγκη δημιουργίας μεθόδων μοντελοποίησης, με κύριο στόχο την ανάλυση και τελικώς τον επιτυχή έλεγχο τέτοιων συστημάτων.

Οι πρώτες μέθοδοι μοντελοποίησης περιλάμβαναν τα δίκτυα Petri καθώς και όλες τις μορφές των «αυτομάτων» ελέγχου. Οι διάφορες μορφές αυτομάτων παρουσιάζονται ως καταλληλότερα εργαλεία μοντελοποίησης καθώς δεν εξαρτώνται από ειδικά χαρακτηριστικά της εκάστοτε διεργασίας. Με βάση το δημιουργηθέν μοντέλο μπορεί να σχεδιαστεί η στρατηγική ελέγχου εύκολα μέσω κάποιου λογισμικού εργαλείου σε έναν ηλεκτρονικό υπολογιστή. Τέλος, ένας Προγραμματιζόμενος Λογικός Ελεγκτής (PLC) αναλαμβάνει τον έλεγχο της διεργασίας σε επίπεδο εφαρμογής. Συγκεκριμένα, οι κύριες μορφές αυτομάτων που απαντώνται στην βιβλιογραφία είναι τα πεπερασμένα αυτόματα, τα υβριδικά αυτόματα, τα χρονισμένα αυτόματα και τέλος τα PLC αυτόματα.

Οι υπάρχουσες μέθοδοι μοντελοποίησης δεν καλύπτουν πλήρως τις ανάγκες των σύγχρονων βιομηχανικών συστημάτων. Η αυξημένη πολυπλοκότητα των γραμμών παραγωγής σε συνδυασμό με την αδυναμία κάλυψης από κάποια μέθοδο μεγάλου φάσματος συστημάτων, οδήγησε στην αναζήτηση νέων μεθόδων μοντελοποίησης που δεν θα εξαρτώνται από τα ιδιαίτερα χαρακτηριστικά του εκάστοτε συστήματος. Η προσπάθεια αυτή, στα πλαίσια της παρούσας διατριβής, απέδωσε καρπούς και η νέα προτεινόμενη μέθοδος παρουσιάστηκε σε μια πρόδρομη μορφή στο παγκόσμιο συνέδριο ελέγχου *IFAC 2005* στην Πράγα [1], ενώ στην τελική της μορφή δημοσιεύτηκε στο διεθνές περιοδικό *International Journal of Industrial and Systems Engineering* το 2008 [2].

1.3 Νέα φορμαλιστική μέθοδος μοντελοποίησης

Η νέα αυτή μέθοδος ονομάστηκε *Global Automata*, ενώ η απόδοση του όρου στα ελληνικά είναι *Γενικευμένα ή Καθολικά Αυτόματα*. Όπως καταδεικνύει και το όνομα, τα global automata βασίζονται στην θεωρία αυτομάτων και ιδίως στα χρονισμένα και υβριδικά αυτόματα. Μερικές έννοιες από αυτές τις δύο μορφές αυτομάτων έχουν διατηρηθεί, ενώ έχουν προστεθεί νέες ώστε να καλύψουν τις όποιες αδυναμίες και να δώσουν στην νέα μέθοδο την απαραίτητη «δύναμη μοντελοποίησης». Με τον όρο δύναμη μοντελοποίησης αναφερόμαστε στην ικανότητα μοντελοποίησης μεγάλου εύρους συστημάτων, χωρίς την απαίτηση το εκάστοτε σύστημα να πληροί ορισμένες προδιαγραφές και να έχει συγκεκριμένα χαρακτηριστικά.

1.4 Εφαρμογές των Global Automata

Οι εφαρμογές των global automata, δεν περιορίζονται στα στενά πλαίσια του βιομηχανικού κόσμου αν και προς αυτόν προσανατολίζονται. Εκτός του πρώτου ορισμού της νέας μεθόδου, συνεχίστηκε η προσπάθεια για την παρουσίαση των δυνατοτήτων της μεθόδου μέσω των εργαλείων μοντελοποίησης που αυτή παρέχει. Συγκεκριμένα, τα δύο εργαλεία που μπορούν να χρησιμοποιηθούν κατά την μοντελοποίηση ενός συστήματος είναι η **συγχώνευση καταστάσεων** καθώς και η **σύνθεση αυτομάτων**. Η σύνθεση αυτομάτων παρουσιάστηκε στην εργασία [3] και αποτελεί τον ορισμό σύνθεσης δύο ή περισσότερων αυτομάτων που αλληλεπιδρούν με στόχο την δημιουργία ενός νέου υπέρ-αυτομάτου. Το προκύπτων αυτόματο έχει τόσες καταστάσεις όσο το γινόμενο των καταστάσεων των υπό σύνθεση αυτομάτων. Η συγχώνευση καταστάσεων αφετέρου είναι η ελάττωση του διαγράμματος καταστάσεων ενός αυτομάτου μέσω συγχώνευσης δύο ή περισσότερων καταστάσεων και της δημιουργίας μιας νέας υπέρ-κατάστασης η οποία έχει ακριβώς την ίδια συμπεριφορά με τις αρχικές καταστάσεις. Μια ολοκληρωμένη παρουσίαση με αναλυτική παράθεση των δυνατοτήτων και των εργαλείων της μεθόδου δίνεται στην εργασία [2]. Πλήρης ανάλυση των ιδιοτήτων των γενικευμένων αυτομάτων περιέχεται στην εργασία [4], όπου ορίζονται έννοιες όπως γραμμικότητα, προσεγγισιμότητα, συστήματα μεταβάσεων κ.α. και έχει υποβληθεί προς δημοσίευση στο διεθνές επιστημονικό περιοδικό *Simulation Modeling Practice and Theory*.

Παραδείγματα εφαρμογών των γενικευμένων αυτομάτων έχουν αποτελέσει εργασίες σε διεθνή συνέδρια [5, 6, 7]. Παράλληλα αναπτύχθηκε ένα συστηματικός οδηγός (εργαλείο) γραφής κώδικα για την εξομοίωση συστημάτων που έχουν μοντελοποιηθεί με την προτεινόμενη μέθοδο. Ο οδηγός αυτός αποτέλεσε το περιεχόμενο της εργασίας [8], η οποία παρουσιάστηκε στο διεθνές συνέδριο 7th IFAC Symposium on Advances in Control Education. Ο οδηγός περιέχει κανόνες που έχουν ως στόχο την μετατροπή ενός αυτομάτου (διάγραμμα καταστάσεων, μεταβάσεις, συναρτήσεις κτλ.) σε εκτελέσιμο κώδικα σε κάποιο υπολογιστικό περιβάλλον εξομοίωσης. Τα παραδείγματα που παρατίθενται υλοποιούν παραδείγματα βιομηχανικών εφαρμογών σε περιβάλλον Matlab αλλά και σε περιβάλλον CX-Supervisor, ένα λογισμικό πακέτο Supervisory Control and Data Acquisition (SCADA) της OMRON. Η επιλογή των δύο αυτών λογισμικών πακέτων έγινε, αφενός για το Matlab με βάση την δημοτικότητα του στον χώρο ελέγχου, και αφετέρου για το CX-Supervisor λόγω της οικειότητας

των μηχανικών ελέγχου της βιομηχανίας με λογισμικά πακέτα SCADA. Ο πλήρης οδηγός με παραδείγματα κρίθηκε ως καταλληλότερος για την εκπαίδευση νέων μηχανικών στο αντικείμενο του ελέγχου βιομηχανικών συστημάτων, καθώς τα παραγόμενα μοντέλα μπορούν να εξομοιωθούν και να αποτελέσουν διδακτικό εργαλείο για την εξοικείωση φοιτητών με καθαρά βιομηχανικού τύπου λογισμικά. Για τον λόγο αυτό ο ολοκληρωμένος οδηγός [9] έχει υποβληθεί προς δημοσίευση στο διεθνές επιστημονικό περιοδικό *International Journal of Engineering Education*.

Κύριο μέλημα της όλης διατριβής εξ αρχής ήταν η δυνατότητα υλοποίησης της νέας μεθόδου μοντελοποίησης σε πραγματικά βιομηχανικά συστήματα. Μια ολοκληρωμένη εφαρμογή σε πραγματικό βιομηχανικού τύπου σύστημα παρουσιάζεται στο διεθνές επιστημονικό περιοδικό *International Journal of Automation in Construction* [10], όπου μια μονάδα έτοιμου σκυροδέματος έχει μοντελοποιηθεί και ελεγχθεί εξ ολοκλήρου με χρήση των γενικευμένων αυτομάτων. Στην συγκεκριμένη εφαρμογή η υλοποίηση έγινε σε ένα σταθμό εποπτικού ελέγχου SCADA. Το υπό μελέτη σύστημα μοντελοποιήθηκε με χρήση πέντε διαφορετικών αυτομάτων που εκτελούνται ταυτόχρονα. Σε αυτή την περίπτωση το ένα αυτόματο έχει τον ρόλο του επόπτη-συντονιστή συγχρονίζοντας τα υπόλοιπα και καθορίζοντας την ιεραρχία εκτέλεσης των αυτομάτων.

Όμως, ο κυρίαρχος τύπος ελεγκτή στις βιομηχανικές διεργασίες είναι ο Προγραμματιζόμενος Λογικός Ελεγκτής (PLC) και συνεπώς η υλοποίηση των γενικευμένων αυτομάτων στην βιομηχανία θα έπρεπε να έχει σαν πλατφόρμα υλικού το PLC. Στην προσπάθεια προτυποποίησης των γλωσσών προγραμματισμού για PLCs η διεθνής επιτροπή *International Electrotechnical Commission* δημοσίευσε το πρότυπο IEC 61131-3 σε πρόδρομη μορφή το 1993 και σε επανέκδοση το 2003, το οποίο έχει υιοθετηθεί από τους κυριότερους κατασκευαστές PLC. Το διεθνές πρότυπο ορίζει τα κοινά στοιχεία των προγραμμάτων και καθορίζει την σύνταξη των γλωσσών προγραμματισμού. Βεβαίως η συμβατότητα προγραμμάτων μεταξύ διαφορετικών κατασκευαστών δεν υπάρχει καθώς το πρότυπο είναι αρκετά γενικό.

Με βάση το πρότυπο IEC 61131-3 δημιουργήθηκε ένα εργαλείο σύνθεσης για την υλοποίηση των γενικευμένων αυτομάτων σε PLCs. Το εργαλείο, που έχει γίνει δεκτό προς δημοσίευση στο διεθνές επιστημονικό περιοδικό *International Journal of Industrial and Systems Engineering* [11], περιγράφει όλα τα συστηματικά βήματα

που πρέπει να ακολουθήσει ο μηχανικός ελέγχου για να υλοποιήσει την στρατηγική ελέγχου, που εμπεριέχει ένα γενικευμένο αυτόματα, σε ένα PLC. Ο γράφος ελέγχου καθώς και όλες οι παράμετροι του αυτομάτου μετατρέπονται σε εκτελέσιμο κώδικα βάσει του οποίου το PLC ελέγχει το υπό μελέτη σύστημα. Εδώ πρέπει να σημειωθεί ότι το πρότυπο IEC 61131-3 περιλαμβάνει ένα σύνολο πέντε γλωσσών για τον προγραμματισμό PLCs. Οι πλέον διαδεδομένες γλώσσες είναι οι *Instruction List (IL)*, *Ladder Diagram (LAD)* και *Function Block Diagram (FBD)*, οι οποίες υποστηρίζονται από το σύνολο των κατασκευαστών. Επιπροσθέτως οι γλώσσες *Structured Text* και *Sequential Function Charts (SFC)* συγκαταλέγονται στις γλώσσες υψηλού επιπέδου αλλά δεν τυγχάνουν της υποστήριξης του συνόλου των κατασκευαστών. Το εργαλείο σύνθεσης για την υλοποίηση γενικευμένων αυτομάτων περιγράφει όλες τις γλώσσες του προτύπου δίνοντας τα ειδικά χαρακτηριστικά της καθεμιάς και αναλυτικά παραδείγματα χρήσης τους.

1.5 Δομή κεφαλαίων διατριβής

Το Κεφάλαιο 2 αποτελεί εισαγωγικό κεφάλαιο στα βιομηχανικά συστήματα παρουσιάζοντας τα ιδιαίτερα χαρακτηριστικά και τις απαιτήσεις μοντελοποίησης των βιομηχανικών συστημάτων.

Το Κεφάλαιο 3 επιχειρεί μια σύντομη βιβλιογραφική ανασκόπηση στον χώρο της μοντελοποίησης βιομηχανικών συστημάτων με χρήση της θεωρίας αυτομάτων και όχι μόνο. Οι παρουσιαζόμενες αναφορές περιλαμβάνουν και ορισμένες προσπάθειες αυτοματοποίησης του έργου του μηχανικού και συγκεκριμένα της αυτοματοποιημένης παραγωγής κώδικα για προγραμματισμό PLCs.

Το Κεφάλαιο 4 παρουσιάζει τις υπάρχουσες μεθόδους μοντελοποίησης βιομηχανικών συστημάτων από την περιοχή της Θεωρίας Αυτομάτων, τα Πεπερασμένα, Υβριδικά, Χρονισμένα και PLC Αυτόματα. Επίσης παρατίθενται τα προσφερόμενα λογισμικά πακέτα μοντελοποίησης συστημάτων διακριτών γεγονότων, που έχουν αναπτυχθεί κυρίως από ερευνητικά ινστιτούτα για μη εμπορική χρήση.

Το Κεφάλαιο 5 εισάγει τα Γενικευμένα Αυτόματα (Global Automata), ως μια νέα μέθοδο μοντελοποίησης βιομηχανικών συστημάτων με αυξημένες δυνατότητες συγκριτικά με τις προαναφερθείσες μεθόδους. Επίσης παρουσιάζονται οι βασικές

ιδιότητες των γενικευμένων αυτομάτων, ενώ γίνεται μία πρώτη σύγκριση με άλλες μεθόδους μοντελοποίησης, τόσο από το χώρο των αυτομάτων, όσο και από άλλες προσεγγίσεις με βιομηχανικό προσανατολισμό.

Το Κεφάλαιο 6 παρουσιάζει τα εργαλεία μοντελοποίησης που αναπτύχθηκαν για τα γενικευμένα αυτόματα και ειδικά τα εργαλεία Συγχώνευσης Καταστάσεων και Σύνθεσης Αυτομάτων, τα οποία παρέχουν πρόσθετες δυνατότητες μοντελοποίησης.

Το Κεφάλαιο 7 προβάλλει τα εργαλεία υλοποίησης των γενικευμένων αυτομάτων, είτε σε γλώσσες προγραμματισμού υψηλού επιπέδου, είτε στις γλώσσες των Προγραμματιζόμενων Λογικών Ελεγκτών, όπως αυτές ορίζονται από το διεθνές πρότυπο IEC 61131-3.

Το Κεφάλαιο 8 παρουσιάζει χαρακτηριστικά παραδείγματα εφαρμογών των γενικευμένων αυτομάτων ως μεθόδου μοντελοποίησης. Μέσω των παραδειγμάτων υπάρχει αναλυτική εξήγηση χρήσης όλων των εργαλείων που αναφέρθηκαν στα δύο προηγούμενα κεφάλαια, τόσο στα εργαλεία μοντελοποίησης όσο και στα εργαλεία υλοποίησης.

Το Κεφάλαιο 9 κλείνει την παρούσα διατριβή συνοψίζοντας τα αποτελέσματα της, θεωρητικά και πρακτικά όπως αυτά παρουσιάστηκαν στις δημοσιευμένες εργασίες που προέκυψαν κατά την διάρκεια εκπόνησης της συγκεκριμένης διατριβής. Τέλος παρατίθενται προτάσεις για περαιτέρω έρευνα.

Κεφάλαιο 2^ο

Βιομηχανικά Συστήματα

Τα τελευταία είκοσι χρόνια, η κύρια πρόοδος στον τομέα των βιομηχανικών συστημάτων βασίστηκε στην τεχνολογία των προγραμματιζόμενων λογικών ελεγκτών PLCs, των κατανεμημένων συστημάτων ελέγχου (DCS) και των απλών προσωπικών υπολογιστών (PCs). Έως σήμερα δεν υπάρχει κοινά αποδεκτή μεθοδολογία για την επιλογή και υλοποίηση του καταλληλότερου για κάθε εφαρμογή συστήματος ελέγχου, και η συνήθης στρατηγική που ακολουθείται διέπεται από την φιλοσοφία, «υλοποίηση εξαρτώμενη από την εφαρμογή». Αυτό οφείλεται κυρίως στο μεγάλο κενό μεταξύ θεωρίας και εφαρμογών. Ενώ η θεωρία δυναμικών συστημάτων αποτελεί μια κοινή θεωρητική βάση, στα συστήματα διακριτών γεγονότων δεν υπάρχει κάποια αντίστοιχη, και επιπλέον εικάζεται ότι θα αργήσει πολύ η εμφάνιση μιας τέτοιας κοινής πλατφόρμας [12]. Το σημαντικό επιστημονικό ερώτημα που υπογραμμίζει την τρέχουσα κατάσταση στα Συστήματα Διακριτών Γεγονότων (ΣΔΓ) είναι το «πώς θα κατανοήσουμε και συνεπώς θα περιγράψουμε με μαθηματικό τρόπο τα μεγάλα και σύνθετα βιομηχανικά συστήματα». Η διαχείριση της εκθετικής «έκρηξης» του χώρου κατάστασης των σύνθετων συστημάτων απαιτεί βελτίωση της αρχιτεκτονικής με την οποία δομούνται τα μοντέλα διακριτών γεγονότων.

Η κύρια πρακτική στην περιγραφή των Βιομηχανικών διαδικασιών και διατάξεων παραγωγής είναι η αντιμετώπισή τους ως Συστήματα Διακριτών Γεγονότων και στον έλεγχό τους η χρήση Προγραμματιζόμενων Λογικών Ελεγκτών (ΠΛΕ). Ένα ολοκληρωμένο βιομηχανικό σύστημα ελέγχου συνήθως αποτελείται, εκτός από τους ελεγκτές τύπου PLC, τόσο από το αντίστοιχο σύστημα δικτύωσης όσο και από ένα σύστημα Εποπτικού Ελέγχου και Συλλογής Πληροφοριών (Supervisory Control and Data Acquisition SCADA). Στην συνέχεια θα ολοκληρωθεί η εισαγωγή στα Βιομηχανικά Συστήματα παραγωγής μέσω παρουσίασης των χαρακτηριστικών της σύγχρονης δομής των παραγωγικών μονάδων. Επίσης παρουσιάζονται εκτενώς τα χαρακτηριστικά των Προγραμματιζόμενων Λογικών Ελεγκτών, των Βιομηχανικών

Δικτύων Αυτοματισμού καθώς και των συστημάτων Εποπτικού Ελέγχου και Συλλογής Πληροφοριών.

2.1 Ολοκληρωμένη Παραγωγή με Υπολογιστές

Η Ολοκληρωμένη Παραγωγή με Υπολογιστές (Computer Integrated Manufacturing CIM), αποτελεί την σύγχρονη ιεραρχική δομή ελέγχου των παραγωγικών μονάδων [13]. Με βάση την δομή CIM, οι διάφορες εργασίες επιμερίζονται στα αντίστοιχα υπολογιστικά συστήματα, τα οποία έχουν συγκεκριμένους βαθμούς ευθύνης όπου οι αποφάσεις των υψηλότερων βαθμίδων μεταδίδονται προς εκτέλεση στα υπολογιστικά συστήματα των κατώτερων βαθμίδων. Η τεχνολογία CIM έχει επιφέρει επαναστατικές αλλαγές στην παραγωγική διαδικασία, κυρίως των μεγάλων Βιομηχανιών, αλλάζοντας εργασίες όπως η συλλογή πληροφοριών, η λειτουργία συστημάτων ψηφιακού αυτομάτου ελέγχου, η επιτήρηση και ο προγραμματισμός της παραγωγής.

Σχήμα 2.1. Ιεραρχική δομή CIM.

Η υλοποίηση μιας στρατηγικής CIM απαιτεί αυστηρή διαίρεση στα ιεραρχικά επίπεδα οργάνωσης, εποπτείας και ελέγχου, όπως αυτή απεικονίζεται στο σχήμα 2.1. Το επίπεδο της οργάνωσης παράγει αποφάσεις που σχετίζονται με την διαχείριση της παραγωγής. Το επίπεδο της εποπτείας είναι δέκτης των αποφάσεων αυτών και

δρομολογεί την στρατηγική ελέγχου βασισμένο κυρίως στις ποιοτικές απαιτήσεις της παραγωγής. Το επίπεδο του ελέγχου υλοποιεί τις προηγούμενες αποφάσεις έχοντας στενή φυσική σύνδεση με τις διεργασίες και εξασφαλίζει την λειτουργική ευστάθεια των διεργασιών. Το ιεραρχικό αυτό σύστημα έχει την ιδιότητα ότι η άνοδος από την βάση προς την κορυφή της ιεραρχίας συνεπάγεται συνεχή αύξηση της σοβαρότητας των λαμβανόμενων αποφάσεων, αλλά ταυτόχρονα μείωση του ρυθμού των αποφάσεων. Παράλληλα, η άνοδος προς την κορυφή της ιεραρχίας αυξάνει συνεχώς τον όγκο της πληροφορίας προς επεξεργασία.

Τα τέσσερα επίπεδα της δομής CIM στα σύγχρονα βιομηχανικά συστήματα υλοποιούνται μέσω διαφορετικών συσκευών ή λογισμικών πακέτων. Στο σχήμα 2.2 απεικονίζονται τα κύρια είδη σε κάθε επίπεδο.

Σχήμα 2.2. Ιεραρχική δομή CIM και αντίστοιχες συσκευές.

Επίπεδο 1

Οι συσκευές πεδίου (ενεργοποιητές, αισθητήρες) παραδοσιακά είναι συνδεδεμένες ενσύρματα, απευθείας στην μονάδα ελέγχου, σε ψηφιακές εισόδους-εξόδους ή σε αντίστοιχες αναλογικές για έλεγχο συνεχών μεταβλητών. Στα σύγχρονα βιομηχανικά συστήματα ευρύνεται ολοένα και περισσότερο η χρήση «έξυπνων» συσκευών

εισόδων/εξόδων με δικτύωση μέσω διαύλου επικοινωνίας ή ακόμα και ασύρματα. Οι νέες συσκευές σε συνδυασμό με τα δίκτυα επικοινωνίας έχουν την δυνατότητα να παρέχουν επιπρόσθετες πληροφορίες, όπως δεδομένα για διάγνωση λαθών.

Επίπεδο 2

Το δεύτερο επίπεδο, επίπεδο ελέγχου, περιλαμβάνει όλους τους ελεγκτές πεδίου, μεταξύ των οποίων κυρίαρχη θέση κατέχει ο Προγραμματιζόμενος Λογικός Ελεγκτής (PLC). Η χρήση PLC έχει ως στόχο τον έλεγχο όλων των συσκευών του πρώτου επιπέδου, τόσο των ψηφιακών όσο και των αναλογικών. Εκτός των PLCs περίοπτη θέση κατέχουν και τα καταναμεμημένα συστήματα ελέγχου (DCS). Στα PLCs και στα DCSs συνδέονται οι αισθητήρες και οι ενεργοποιητές, δίνοντας διπλή υπόσταση σε αυτές τις συσκευές που λειτουργούν ως μονάδες εισόδων/εξόδων αλλά και ως ελεγκτές πεδίου για την εκτέλεση της στρατηγικής ελέγχου. Όσο οι συσκευές του πρώτου πεδίου θα γίνονται πιο «έξυπνες» με ενσωματωμένες δυνατότητες δικτύωσης, τόσο οι ελεγκτές πεδίου θα παρουσιάζονται λιγότερο ως μονάδες εισόδων/εξόδων και θα κυριαρχεί το έργο του ελέγχου.

Επίπεδο 3

Το τρίτο επίπεδο αποτελεί το επίπεδο εποπτείας της όλης παραγωγικής διαδικασίας μιας σύγχρονης βιομηχανίας. Τυπικά αποτελεί την διεπαφή μεταξύ της μονάδας παραγωγής και του επιχειρηματικού πλάνου. Χρησιμοποιείται κυρίως για να εξασφαλίσει ότι τα παραγόμενα προϊόντα συνάδουν με τις ανάγκες του πελάτη τόσο ποιοτικά, όσο και ποσοτικά και χρονικά. Τα λογισμικά πακέτα που υλοποιούν το επίπεδο εποπτείας είναι είτε Manufacturing Execution System (MES), είτε Supervisory Control And Data Acquisition (SCADA), δηλαδή Συστήματα Εποπτικού Ελέγχου και Συλλογής Πληροφοριών.

Επίπεδο 4

Το τέταρτο επίπεδο αποτελεί το επιχειρηματικό σύστημα, όπως ένα Enterprise Resource Planning. Ένα σύστημα ERP αποτελεί το πληροφοριακό σύστημα διαχείρισης, το οποίο είναι υπεύθυνο για την οργάνωση προμηθειών και πληροφοριών, διαχείριση αποθηκών, και γενικότερα για όλες τις απαραίτητες διεργασίες για την ορθή επίτευξη επιχειρηματικών στόχων.

Στο σχήμα 2.3 παρουσιάζεται μια γραμμή παραγωγής που διαιρείται σε τρία επίπεδα. Στο κατώτατο επίπεδο λαμβάνει χώρα η παραγωγική διαδικασία η οποία περιλαμβάνει διάφορες υποδιαδικασίες. Όλα τα δεδομένα μεταφέρονται στο επόμενο επίπεδο μέσω ενός δικτύου επικοινωνίας. Στο δεύτερο επίπεδο βρίσκονται οι κεντρικοί σταθμοί ελέγχου μέσω των οποίων συντονίζονται όλες οι υποδιαδικασίες του προηγούμενου επιπέδου και επιβλέπεται η λειτουργία της όλης διαδικασίας. Το δεύτερο επίπεδο επικοινωνεί με το πρώτο, επίσης μέσω δικτύου επικοινωνίας, έτσι ώστε να διασφαλίζεται η ομαλή παραγγελία και προμήθεια πρώτων υλών. Όπως γίνεται εύκολα κατανοητό, τα τοπικά δίκτυα επικοινωνίας αποτελούν σημαντικό κομμάτι για την πλήρη επίτευξη του CIM μοντέλου, καθώς η δικτυακή διασύνδεση των διαφόρων επιπέδων είναι πρωταρχική ανάγκη.

Σχήμα 2.3. Ολοκληρωμένο Βιομηχανικό Δίκτυο επικοινωνίας.

Οι απαιτήσεις που πρέπει να ικανοποιεί ένα ολοκληρωμένο Βιομηχανικό Πληροφοριακό Σύστημα είναι οι εξής:

- Ολοκληρωμένη αρχιτεκτονική επικοινωνίας

- Κεντρική οργάνωση και επίβλεψη
- Κατανεμημένο και ιεραρχικό έλεγχο
- Συνύπαρξη συσκευών-μηχανών κάθε είδους
- Ανοικτές και διαφανείς επικοινωνίες
- Διάλογο μεταξύ ανθρώπου και συστήματος

Ένα τέτοιο σύστημα απεικονίζεται στο σχήμα 2.4, όπου παρουσιάζεται γραφικά η απαιτούμενη δικτύωση ανάμεσα στις διάφορες μονάδες-επίπεδα μιας Βιομηχανίας.

Σχήμα 2.4. Ολοκληρωμένο Βιομηχανικό Πληροφοριακό Σύστημα.

2.2 Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLCs).

Τα πρώτα δειλά βήματα χρήσης υπολογιστικών μονάδων στην βιομηχανία ξεκίνησαν την δεκαετία του 1960 με την χρήση υπολογιστών. Όμως λόγω του μεγάλου όγκου των πρώτων υπολογιστών και της μικρής ταχύτητας επεξεργασίας η χρήση τους παρέμεινε περιορισμένη. Η επανάσταση στο χώρο προκλήθηκε από την ανάπτυξη των μικροϋπολογιστών τη δεκαετία του 1970. Τα χαρακτηριστικά του νέου τύπου υπολογιστή, όπως μικρό μέγεθος, χαμηλό κόστος και υψηλή ταχύτητα, άνοιξαν την πύλη για την ευρεία χρησιμοποίησή τους στη βιομηχανία, σε οποιοδήποτε είδος εφαρμογής [14, 15].

Με βάση τους μικροϋπολογιστές κάνουν την εμφάνισή τους οι πρώτοι Προγραμματιζόμενοι Λογικοί Ελεγκτές. Οι νέες αυτές μονάδες αντικατέστησαν την απαραίτητη μέχρι τότε καλωδίωση με μια σειρά διαδοχικών λογικών εντολών. Η εξέλιξη των PLCs ακολούθησε μια ανοδική πορεία μέχρι τη σημερινή τους μορφή. Τα τελευταία χρόνια οι δυνατότητες των PLCs αυξήθηκαν με την εισαγωγή μικροελεγκτών. Κάποιες από αυτές τις δυνατότητες είναι η ικανότητα υλοποίησης αντισταθμιστών PID, ελεγκτών σερβομηχανισμών, αλλά και διασύνδεσης μεταξύ τους μέσω δικτύου πεδίου.

Η σημερινή χρησιμότητα των Προγραμματιζόμενων Λογικών Ελεγκτών έγκειται στις ακόλουθες δυνατότητες που έχουν τα PLCs.

- Ακολουθιακός έλεγχος της βιομηχανικής διαδικασίας (εκκίνηση, στάση, παρακολούθηση).
- Απόκτηση ψηφιακών και αναλογικών σημάτων από τους αισθητήρες για την παρακολούθηση της υπό έλεγχο διαδικασίας.
- Μετάδοση των επιθυμητών σημάτων στους ενεργοποιητές της διεργασίας και ενεργοποίηση ή απενεργοποίηση διακοπών και ηλεκτρονόμων.
- Αποκατάσταση επικοινωνίας με άλλα PLCs ή με άλλους κόμβους του δικτύου.

Στα PLCs το κύκλωμα αυτοματισμού υλοποιείται με την μορφή προγράμματος. Εκτός όμως από τις βασικές λειτουργίες σχεδόν όλα τα PLCs, διαθέτουν στον χρήστη και ορισμένες βοηθητικές συσκευές όπως χρονοδιακόπτες, απαριθμητές κ.α. Αν θέλαμε να δώσουμε έναν ακριβή ορισμό για τον Προγραμματιζόμενο Λογικό Ελεγκτή θα λέγαμε ότι:

"Προγραμματιζόμενος Λογικός Ελεγκτής είναι μια ψηφιακή συσκευή με μικροεπεξεργαστή και προγραμματιζόμενη μνήμη, η οποία μπορεί να αποθηκεύσει και να εκτελέσει εντολές του χρήστη για υλοποίηση συναρτήσεων λογικής Boole, ακολουθιακής λογικής, χρονισμού, απαρίθμησης και αριθμητικών πράξεων προκειμένου μέσω ψηφιακών ή αναλογικών εισόδων / εξόδων, να ελέγχει τη λειτουργία μιας σύνθετης μηχανής ή μιας ολόκληρης διεργασίας".

- **Βαθμίδα ή κάρτα εξόδων.** Μέσω των βαθμίδων εξόδου τα αποτελέσματα της εκτέλεσης του προγράμματος μετατρέπονται σε σήματα εντολών προς τις συσκευές εξόδου.
- **Προγραμματιστής.** Είναι η συσκευή προγραμματισμού του PLC και χρησιμοποιείται μόνο κατά τον προγραμματισμό. Σε μικρά PLCs μπορεί να είναι ενσωματωμένη. Πολλές φορές, ιδιαίτερα στα σύγχρονα PLCs, χρησιμοποιείται Η/Υ αντί για προγραμματιστή.

Στους Προγραμματιζόμενους Λογικούς Ελεγκτές η εκτέλεση των διάφορων ενεργειών γίνεται με συγκεκριμένη σειρά. Η όλη ακολουθία είναι κυκλική και περιλαμβάνει τρεις φάσεις που επαναλαμβάνονται κυκλικά και ατέρμονα κατά την διάρκεια λειτουργίας.

- Σάρωση εισόδων.
- Εκτέλεση προγράμματος.
- Ενημέρωση εξόδων.

Σε κάθε κύκλο η βαθμίδα εισόδων σαρώνει τις εισόδους της και μετατρέπει τα λαμβανόμενα σήματα σε ψηφιακή πληροφορία, που αποθηκεύεται σε τμήμα της μνήμης αφιερωμένο ειδικά στις βαθμίδες εισόδου. Στη συνέχεια ο μικροεπεξεργαστής διαβάζει την πληροφορία αυτή από την μνήμη και εκτελεί το πρόγραμμα. Ανάλογα με το πρόγραμμα και την κατάσταση των εισόδων ενεργοποιούνται από τη CPU κάποιες θέσεις μνήμης, που είναι αφιερωμένες στις βαθμίδες εξόδου. Τέλος οι βαθμίδες εξόδου διαβάζουν τη μνήμη και δημιουργούν τα ανάλογα σήματα ενεργοποίησης των συσκευών εξόδου.

Ολοκληρώνοντας την αναφορά στους Προγραμματιζόμενους Λογικούς Ελεγκτές, παρατίθενται τα πλεονεκτήματα και οι ευκολίες από την χρήση τους σε σχέση με τα παραδοσιακά κυκλώματα αυτοματισμού.

- **Εύκολη τροποποίηση προγράμματος.** Αν κάποιοι κανόνες ελέγχου αλλάξουν μετά την εγκατάσταση του συστήματος, η υλοποίηση των αλλαγών περιορίζεται σε μερική τροποποίηση του προγράμματος του ελεγκτή, διαδικασία ιδιαίτερα εύκολη.
- **Εύκολος εντοπισμός βλαβών.** Με χρήση του προγραμματιστή δίνεται η δυνατότητα παρακολούθησης της κατάστασης κάθε θέσεως μνήμης κατά την

διάρκεια λειτουργίας, επιβεβαιώνοντας την ορθή ή μη λειτουργία του συστήματος.

- **Οικονομία χώρου.** Οι σύγχρονοι λογικοί ελεγκτές ως ψηφιακές συσκευές καταλαμβάνουν περιορισμένο χώρο και επιπρόσθετα περιέχουν όλα τα βοηθητικά στοιχεία όπως απαριθμητές, χρονοδιακόπτες, βοηθητικά ρελέ.
- **Έλλειψη εκτεταμένης καλωδίωσης.** Η μόνη καλωδίωση που υπάρχει, είναι για την διασύνδεση όλων των συσκευών με το PLC και όχι για την υλοποίηση ενός πολύπλοκου συμβατικού κυκλώματος αυτοματισμού.
- **Επάρκεια επαφών.** Για την υλοποίηση της λογικής ελέγχου στα κυκλώματα αυτοματισμού είναι απαραίτητη η χρήση βοηθητικών επαφών της κάθε συσκευής, κυρίως των ηλεκτρονόμων, που δεν είναι πάντα διαθέσιμες. Στην περίπτωση των PLCs αναπαρίστανται ως θέσεις μνήμης και μπορούν να χρησιμοποιηθούν θεωρητικά άπειρες φορές στο πρόγραμμα αυτοματισμού.
- **Αυξημένη αξιοπιστία** σε σχέση με τα συμβατικά συστήματα που παρουσιάζουν μηχανικές βλάβες στα κινούμενα μέρη τους.
- Ύπαρξη αναλογικών και ψηφιακών βαθμίδων εισόδου / εξόδου.
- Ύπαρξη καρτών ειδικού σκοπού όπως, υλοποίηση αντισταθμιστών PID, έλεγχος βηματικών κινητήρων κ.α.
- Δυνατότητα επικοινωνίας με άλλα PLCs ή με άλλα συστήματα για λόγους ελέγχου, εποπτείας ή αναφοράς.

Για την αξιοποίηση της τελευταίας δυνατότητας των σύγχρονων Προγραμματιζόμενων Λογικών Ελεγκτών είναι απαραίτητη η διασύνδεση του ελεγκτή σε ένα βιομηχανικό δίκτυο. Στην επόμενη παράγραφο αναφέρονται οι βασικές αρχές των βιομηχανικών δικτύων αυτοματισμού.

2.3 Δίκτυα Βιομηχανικού Αυτοματισμού

Τα Βιομηχανικά Δίκτυα Υπολογιστών αποτελούν μία υποκατηγορία των Τοπικών Δικτύων Υπολογιστών με συγκεκριμένες απαιτήσεις και προδιαγραφές. Ο σκοπός των δικτύων αυτών είναι να αποτελέσουν τη βάση για σύγχρονα, ευέλικτα και αποδοτικά συστήματα αυτοματοποίησης. Στα συστήματα αυτά απαιτείται η αξιόπιστη διασύνδεση ετερογενών μονάδων, όπως υπολογιστικά συστήματα γενικού ή / και ειδικού σκοπού, προγραμματιζόμενοι λογικοί ελεγκτές, αισθητήρες, ενεργοποιητές κ.τ.λ.

Η κύρια λειτουργική απαίτηση που πρέπει να ικανοποιούν τα προηγμένα συστήματα αυτοματοποίησης είναι η εξασφάλιση της δυνατότητας απόκρισής τους σε πραγματικό χρόνο. Συνεπώς, τα βιομηχανικά δίκτυα, που παρέχουν την υποδομή για ανάπτυξη κατανεμημένων συστημάτων αυτοματοποίησης, θα πρέπει να εξασφαλίζουν δικτυακή απόκριση σε πραγματικό χρόνο.

Η πιο πρόσφατη εφαρμογή των βιομηχανικών δικτύου έχει ως στόχο την ενσωμάτωση σε ένα ενιαίο δίκτυο όλων των διαδικασιών ελέγχου και επεξεργασίας στο εργοστασιακό περιβάλλον. Η επίτευξη αυτού του στόχου επιβάλλει την διασύνδεση όλων των επιπέδων του εργοστασιακού περιβάλλοντος, από το κατώτερο επίπεδο παρακολούθησης και ελέγχου, μέχρι το ανώτερο επίπεδο της διαχείρισης υλικών, παραγγελιών, προϊόντων κ.τ.λ. Η πληροφορία που μεταδίδεται σε κάθε επίπεδο έχει εντελώς διαφορετικά χαρακτηριστικά. Στο κατώτερο επίπεδο τα δεδομένα είναι μικρά σε μέγεθος, αλλά είναι επιτακτική η ανάγκη για ταχύτατη μετάδοσή τους. Αντίθετα, όσο πάμε από τη βάση προς την κορυφή της ιεραρχίας, ο όγκος των δεδομένων αυξάνεται δραματικά αλλά μειώνονται οι απαιτήσεις ως προς την ταχύτητα.

Σημαντική παράμετρος είναι η χρησιμοποιούμενη μεθοδολογία διασύνδεσης των σταθμών που απαρτίζουν ένα δίκτυο, η οποία αποτελεί την τοπολογία του δικτύου. Στον χώρο του βιομηχανικού αυτοματισμού, συναντώνται σχεδόν όλα τα είδη τοπολογιών, όπως σύνδεση αστέρα, δακτυλίου, ιεραρχική ή τοπολογία δέντρου, διαύλου, καθώς και οι υβριδικές τοπολογίες. Τα δίκτυα υβριδικής τοπολογίας αποτελούνται από συνδυασμό υποδικτύων με κάποιες από τις παραπάνω τοπολογίες. Δύο παραδείγματα τέτοιων τοπολογιών είναι ο συνδυασμός δακτυλίου και αστέρα, όπως και η δενδρική τοπολογία διαύλων.

Τα μέσα μετάδοσης δεδομένων που χρησιμοποιούνται στα δίκτυα βιομηχανικού αυτοματισμού είναι τέσσερα και χρησιμοποιούνται αναλόγως την εκάστοτε εφαρμογή. Το συνεστραμμένο ζεύγος καλωδίων αποτελεί το πιο απλό μέσο μετάδοσης και αποτελείται από δύο συνεστραμμένους μονωμένους χάλκινους αγωγούς. Η συστροφή γίνεται για την αποφυγή ηλεκτρικών αλληλεπιδράσεων μεταξύ των αγωγών. Χρησιμοποιείται ως μια φτηνή λύση με ικανοποιητική απόδοση για αποστάσεις μερικών χιλιομέτρων. Τα επόμενα δύο μέσα είναι τα ομοαξονικά καλώδια

βασικής και ευρείας ζώνης. Το καλώδιο βασικής ζώνης των 50Ωm χρησιμοποιείται για ψηφιακή μετάδοση. Έχει ικανοποιητική ανοχή σε θόρυβο και για αποστάσεις μέχρι ενός χιλιομέτρου και ο ρυθμός μετάδοσης δεδομένων μπορεί να φτάσει τα 10Mbps. Το ομοαξονικό καλώδιο ευρείας ζώνης των 75Ωm χρησιμοποιείται για αναλογική μετάδοση. Χαρακτηρίζεται από υψηλό εύρος ζώνης και μεγάλες αποστάσεις μετάδοσης. Χρησιμοποιείται κυρίως για μετάδοση εικόνας, φωνής και δεδομένων. Τέλος, οι οπτικές ίνες κερδίζουν ολοένα μεγαλύτερο έδαφος στον βιομηχανικό κόσμο καθώς παρέχουν σημαντικά μεγάλο εύρος ζώνης, μικρές απώλειες ισχύος και μετάδοση μεγάλων αποστάσεων. Δεν επηρεάζονται από ηλεκτρομαγνητικές παρεμβολές, μεταβολές της τάσεως του δικτύου και χημικά διαβρωτικά. Μειονεκτήματα είναι το υψηλό κόστος και η δύσκολη διασύνδεση οπτικών ινών μεταξύ τους.

Βέβαια, τα τελευταία χρόνια υπάρχει μια διαρκής προσπάθεια υιοθέτησης, από τον βιομηχανικό κόσμο, του ασύρματου τρόπου μετάδοσης. Ασύρματοι αισθητήρες μετρούν τα φυσικά μεγέθη της διεργασίας και ενημερώνουν τον ελεγκτή, ενώ και οι επιμέρους ελεγκτές (PLC) έχουν την δυνατότητα μεταξύ τους ασύρματης επικοινωνίας. Η ασύρματη μετάδοση δεδομένων αποτελεί την σύγχρονη «επανάσταση», καθώς θα εκλείψει η ανάγκη ενσύρματης δικτύωσης των συσκευών μιας βιομηχανικής διεργασίας.

Μοντέλο ανοικτών επικοινωνιών ISO/OSI.

Στα πρώτα βήματα στο χώρο των δικτύων επικοινωνίας, κάθε κατασκευαστής χρησιμοποιούσε ξεχωριστό πρωτόκολλο επικοινωνίας, με αποτέλεσμα η σύνδεση συσκευών διαφορετικών κατασκευαστών να καθίσταται αδύνατη. Για την αποφυγή ασυμβατοτήτων, ο Διεθνής Οργανισμός Τυποποίησης (ISO) ανέπτυξε το Ανοικτό Μοντέλο Διασύνδεσης (OSI) δίνοντας έτσι ένα πρότυπο μοντέλο επικοινωνίας που ακολουθούν όλοι οι κατασκευαστές. Βασικό χαρακτηριστικό του μοντέλου είναι η ιεραρχική διαστρωμάτωση σε επτά επίπεδα. Κάθε επίπεδο είναι άμεσα συνδεδεμένο με το προηγούμενο και διαθέτει ένα συγκεκριμένο τρόπο διασύνδεσής του με τα υπόλοιπα. Παρακάτω παραθέτονται τα επτά επίπεδα του μοντέλου, με απλή αναφορά των κύριων χαρακτηριστικών τους.

- Φυσικό Επίπεδο 1.
 - Μέσα μετάδοσης.

- Αναλογική και ψηφιακή μετάδοση.
- Μεταγωγή.
- Επίπεδο Γραμμής 2.
 - Πλαισίωση.
 - Έλεγχος σφαλμάτων.
 - Ανίχνευση και διόρθωση σφαλμάτων.
 - Έλεγχος ροής
- Επίπεδο Δικτύου 3.
 - Αλγόριθμοι δρομολόγησης.
 - Έλεγχος συμφόρησης.
 - Διασύνδεση δικτύων.
- Επίπεδο Μεταφοράς 4.
 - Διαχείριση συνδέσεων.
 - Διευθυνσιοδότηση.
 - Εγκατάσταση σύνδεσης.
 - Τερματισμός σύνδεσης.
- Επίπεδο Συνόδου 5.
 - Διαχείριση διαλόγου.
 - Συγχρονισμός.
 - Κλήση απομακρυσμένων διαδικασιών.
- Επίπεδο Παρουσίασης 6.
 - Αναπαράσταση δεδομένων.
 - Συμπίεση δεδομένων.
 - Ασφάλεια δικτύου.
 - Κρυπτογραφία.
- Επίπεδο Εφαρμογής 7.
 - Μεταφορά, προσπέλαση και διαχείριση αρχείων.
 - Ηλεκτρονικό ταχυδρομείο.
 - Νοητά τερματικά.

Ολοκληρώνοντας την σύντομη περιγραφή των βασικών χαρακτηριστικών των δικτύων, παρουσιάζονται συνοπτικά τα πλεονεκτήματα που προκύπτουν από τη χρήση τους στο βιομηχανικό περιβάλλον.

- ✓ Αισθητή μείωση του κόστους καλωδίωσης.

- ✓ Μείωση του όγκου των ταμπλό χειρισμού λόγω αποκέντρωσης και χρήσης οθονών.
- ✓ Μείωση των επιπρόσθετων εξαρτημάτων διασύνδεσης.
- ✓ Μείωση του κόστους εγκατάστασης.
- ✓ Μείωση του κόστους συντήρησης.
- ✓ Κεντρική επεξεργασία και συγκέντρωση δεδομένων.
- ✓ Μεγαλύτερη λειτουργική αξιοπιστία και παραγωγικότητα.
- ✓ Δυνατότητα εγκατάστασης Συστημάτων Εποπτικού Ελέγχου και Συλλογής Πληροφοριών.

2.4 Συστήματα εποπτικού ελέγχου και συλλογής πληροφοριών SCADA

Η βιομηχανική πληροφορική αποτελεί έναν από τους σημαντικότερους και ραγδαία εξελισσόμενους κλάδους της Πληροφορικής. Η απαίτηση για μια αποτελεσματικότερη διαχείριση όλων των πληροφοριών των διαφόρων τμημάτων μιας επιχείρησης οδήγησε στην ανάπτυξη εξειδικευμένων εφαρμογών βιομηχανικής πληροφορικής. Η ανταλλαγή των πληροφοριών μεταξύ όλων των τμημάτων προϋποθέτει τη σύνδεση όλων των επιμέρους εφαρμογών, σε ένα κεντρικό σύστημα Πληροφόρησης και Διαχείρισης (Management Information System).

Τα συστήματα SCADA (Supervisory Control And Data Acquisition) αποτελούν εφαρμογή της βιομηχανικής πληροφορικής για την εποπτεία της παραγωγής. Κάθε διεργασία παραγωγής χαρακτηρίζεται από κάποιες κρίσιμες παραμέτρους, οι οποίες παίζουν καθοριστικό ρόλο στην παραγωγική διαδικασία και στην ποιότητα των παραγόμενων προϊόντων (π.χ. πίεση, θερμοκρασία, χρόνος, πυκνότητα κ.α.). Τα συστήματα SCADA παρακολουθούν on-line μέσω μονάδων Προγραμματιζόμενων Λογικών Ελεγκτών και καταγράφουν συνεχώς σε ηλεκτρονικούς υπολογιστές όλες τις κρίσιμες παραμέτρους της παραγωγικής διαδικασίας, για την επίτευξη εποπτείας σε πραγματικό χρόνο.

Ένα σύστημα SCADA περιλαμβάνει απεικόνιση σε μιμικά διαγράμματα όλων των διεργασιών παραγωγής, ενδείξεις των τιμών των μετρούμενων μεγεθών, διαρκή συλλογή και αποθήκευση δεδομένων σε Η/Υ, γνωστοποίηση σφαλμάτων κ.α. Επίσης σημαντική είναι η δυνατότητα παρακολούθησης του συστήματος μέσω κατάλληλα

διαμορφωμένων σελίδων του διαδικτύου. Οι κύριες λειτουργίες ενός συστήματος SCADA είναι οι ακόλουθες.

- Συλλογή δεδομένων από τα PLCs και τις RTU. Όλα τα επιθυμητά σήματα μεταδίδονται προς το σύστημα SCADA μέσω του δικτύου βιομηχανικού αυτοματισμού.
- Αποθήκευση των πληροφοριών στη βάση δεδομένων και αναπαράστασή τους μέσω γραφημάτων. Οι επιλεγμένες πληροφορίες αναπαρίστανται είτε αυτούσια είτε έπειτα από κατάλληλη επεξεργασία.
- Ανάλυση δεδομένων και ειδοποίηση του προσωπικού σε περιπτώσεις σφάλματος. Όταν τα δεδομένα πάρουν τιμές μη κανονικές το σύστημα SCADA ειδοποιεί με οπτική ή ακουστική σήμανση τους χειριστές, ώστε να αποφευχθούν δυσάρεστες επιπτώσεις.
- Έλεγχος κλειστού βρόχου διεργασιών. Υπάρχει η δυνατότητα εφαρμογής τεχνικών ελέγχου, αυτόματες ή χειροκίνητες.
- Γραφική απεικόνιση των τμημάτων της διεργασίας σε μιμικά διαγράμματα και παρουσιάσεις των δεδομένων σε ενεργά πεδία. Τα μιμικά διαγράμματα απεικονίζουν ρεαλιστικά τμήματα της διεργασίας με στόχο την ευκολότερη εποπτεία και την κατανόηση των δεδομένων από τους χειριστές του συστήματος.
- Καταγραφή όλων των συμβάντων κανονικών και μη για την δημιουργία ιστορικού αρχείου. Σε κάθε βιομηχανία υπάρχει καταγραφή όλων των κρίσιμων παραμέτρων. Παλιότερα γινόταν με χειρόγραφη καταγραφή, ενώ σήμερα την ευθύνη αυτή έχει αναλάβει η βάση δεδομένων του συστήματος SCADA.
- Υποστήριξη διπλού υπολογιστικού συστήματος με αυτόματη εναλλαγή, αν αυτό κρίνεται σκόπιμο βάση της υπό έλεγχο διεργασίας. Σε διεργασίες υψηλής επικινδυνότητας πρέπει να ελαχιστοποιηθεί όσο το δυνατόν περισσότερο η εμφάνιση σφάλματος λόγω βλάβης του εξοπλισμού. Για τον λόγο αυτό τα συστήματα SCADA υποστηρίζουν δεύτερο υπολογιστικό σύστημα που αναλαμβάνει σε περίπτωση σφάλματος.
- Μεταφορά δεδομένων σε άλλα τμήματα του κεντρικού συστήματος Πληροφόρησης και Διαχείρισης.
- Έλεγχος της πρόσβασης χειριστών στα διάφορα υποσυστήματα του συστήματος SCADA.

- Ειδικές εφαρμογές λογισμικού όπως εκτέλεση κώδικα C++ ή ανάπτυξη ευφυών συστημάτων.

Κεφάλαιο 3^ο

Μοντελοποίηση και Έλεγχος Βιομηχανικών Συστημάτων

Η σπουδαιότητα της μοντελοποίησης ενός βιομηχανικού συστήματος για ανάγκες εξομοίωσης είναι αναγνωρισμένη ευρέως, τόσο στους ακαδημαϊκούς κόλπους, όσο και στον χώρο της αγοράς. Σημαντικότερο πλεονέκτημα, η σχεδίαση συστημάτων υψηλών προδιαγραφών ποιότητας και ασφάλειας με ταυτόχρονη μείωση του χρόνου και του κόστους υλοποίησης. Το ερώτημα που προκύπτει είναι: *Ποιος είναι ο ορθότερος ορισμός ενός βιομηχανικού συστήματος;* Μια βιομηχανική γραμμή παραγωγής περιλαμβάνει εξοπλισμό διαφόρων ειδών (αισθητήρες, ενεργοποιητές, ρομποτικούς μηχανισμούς, βαλβίδες κ.α.) υπό τον έλεγχο είτε ενός κεντρικού ελεγκτή, είτε πολλαπλών αποκεντρωμένων ελεγκτών. Συνεπώς ένα βιομηχανικό σύστημα μπορεί να θεωρηθεί ως ένα δυναμικό σύστημα το οποίο εξελίσσεται στο χρόνο σύμφωνα με τις στιγμιαίες αλλαγές φυσικών μεγεθών, δηλαδή ως ένα Σύστημα Διακριτών Γεγονότων [16]. Μια διαφορετική προσέγγιση, θα υποστήριζε ότι τα βιομηχανικά συστήματα διέπονται από ελεγκτές διακριτών καταστάσεων, που οι μεταβάσεις μεταξύ των καταστάσεων πυροδοτούνται από συνθήκες επί μετρούμενων φυσικών μεγεθών (π.χ. θερμοκρασία, ταχύτητα, ροή υγρού). Η κατηγορία συστημάτων που περιγράφονται συγχρόνως και από διακριτές και από συνεχείς μεταβλητές είναι τα Υβριδικά Συστήματα [17, 18].

Η μελέτη ενός βιομηχανικού συστήματος μέσω εξομοίωσης επιτρέπει την μελέτη της δυναμικής συμπεριφοράς του και τον καθορισμό των ιδιαίτερων λειτουργικών χαρακτηριστικών του [19]. Ειδικότερα σήμερα, που ο χρόνος ζωής των βιομηχανικών προϊόντων μειώνεται, ενώ παράλληλα αυξάνεται ο παγκόσμιος ανταγωνισμός, η ανάγκη παραγωγής ορθών μοντέλων σε μικρότερο χρονικό διάστημα γίνεται ολοένα και εντονότερη. Αναδεικνύεται λοιπόν η αναγκαιότητα ύπαρξης φορμαλιστικών μεθόδων μοντελοποίησης με ευρύ πεδίο εφαρμογών [20, 21], καθώς επιτρέπουν την εις βάθος ανάλυση της συμπεριφοράς ενός συστήματος. Ιδιαίτερη αναφορά πρέπει να

γίνει στην διαδικασία εξακρίβωσης και εξάλειψης πιθανών δυσλειτουργιών ενός συστήματος με πιθανούς κινδύνους τόσο για υλικές ζημιές, όσο και για ανθρώπινες απώλειες. Η κατηγοριοποίηση των επιμέρους επισφαλών υποσυστημάτων (π.χ. που δύνανται να προκαλέσουν εκρήξεις ή διαρροές) γίνεται με βάση το πρότυπο Safety Integrity Level (SIL-1 έως SIL-4 ανάλογα με τον βαθμό επικινδυνότητας IEC 61508 [22] – IEC 61511 [23]) λαμβάνοντας υπόψη κινδύνους για υλικές ζημιές, ανθρώπινες απώλειες αλλά και περιβαλλοντικές καταστροφές. Ο καθορισμός της κατηγορίας ασφάλειας του συστήματος γίνεται με βάση τον χρησιμοποιούμενο εξοπλισμό και τους πιστοποιημένους, σε αυτόν, χρόνους λανθάνουσας κατάστασης. Οι χρόνοι αυτοί περιλαμβάνουν τον μέσο χρόνο παρουσίασης μιας βλάβης/δυσλειτουργίας του εξοπλισμού MTTF (Mean Time To Failure), τον απαιτούμενο χρόνο επισκευής MTTR (Mean Time To Repair) και το άθροισμα αυτών που αποτελεί τον μέσο χρόνο μεταξύ βλαβών MTBF=MTTF+MTTR (Mean Time Between Failure). Στη συνέχεια η μελέτη ολοκληρώνεται με τον υπολογισμό της πιθανότητας αστοχίας του εξοπλισμού σε πιθανή απαίτηση λειτουργίας PFD (Probability of Failure on Demand) και αναλόγως της επικινδυνότητας της συγκεκριμένης εφαρμογής καθορίζεται η κατηγορία SIL.

Τις περασμένες δεκαετίες πολλές μέθοδοι έχουν προταθεί και εξεταστεί από ερευνητές για την μοντελοποίηση βιομηχανικών συστημάτων. Κυρίαρχη θέση μεταξύ αυτών κατέχουν τα Δίκτυα Petri και οι διάφορες μορφές αυτομάτων. Τα Δίκτυα Petri [24] πρωτοπαρουσιάστηκαν στις αρχές της δεκαετίας του 1960 και έκτοτε αποτελούν ένα ισχυρότατο εργαλείο μοντελοποίησης και ανάλυσης δυναμικών συστημάτων διακριτών γεγονότων. Σε αντίθεση με τις προσδοκίες των επιστημόνων, τα Δίκτυα Petri δεν είχαν αντίστοιχη επιτυχία στον έλεγχο κλειστού βρόχου, κυρίως γιατί δεν υπήρχε ξεκάθαρος διαχωρισμός του φυσικού συστήματος και του ελεγκτή. Στον αντίποδα, τα αυτόματα [25, 26, 27, 28, 29] παρουσιάζονται ως καταλληλότερο εργαλείο μοντελοποίησης. Γενικά, ένα αυτόματο, πέραν της φορμαλιστικής του διατύπωσης, αποτελεί μια γραφική αναπαράσταση της φυσικής λειτουργίας ενός συστήματος διακριτών γεγονότων. Τα Πεπερασμένα Αυτόματα (Finite automata) [28], αποτελούν την βασική μορφή αυτομάτων και ίσως αποτελούν την απλούστερη μαθηματική μέθοδο μοντελοποίησης συστημάτων διακριτών γεγονότων. Τα πεπερασμένα αυτόματα αναπαριστούν ένα σύστημα με ένα πεπερασμένο πλήθος καταστάσεων και μεταβάσεις μεταξύ αυτών. Μια επέκταση της βασικής αυτής δομής που αναπαριστά την αλληλεπίδραση μεταξύ διακριτών γεγονότων και δυναμικών συστημάτων συνεχούς χρόνου, αποτελούν τα Υβριδικά Αυτόματα (Hybrid automata).

Τα υβριδικά αυτόματα [27, 29, 30, 31, 32] έχουν εισαχθεί ως μια φορμαλιστική μέθοδος μοντελοποίησης υβριδικών συστημάτων, κατέχοντας περίοπτη θέση μεταξύ των αυτομάτων ελέγχου. Ένα άλλο υπερσύνολο των πεπερασμένων αυτομάτων είναι τα Χρονισμένα Αυτόματα (Timed Automata) [26, 33], στα οποία κάθε μετάβαση περιορίζεται από ένα πρόσθετο σύνολο ρολογιών. Τέλος, ένα υποσύνολο των χρονισμένων αυτομάτων αποτελούν τα PLC Αυτόματα [25, 34], που πρωτοεμφανίστηκαν το 1997 από τον H. Dierks, ως ένα ισχυρό εργαλείο περιγραφής καταμετρημένων συστημάτων πραγματικού χρόνου. Τα PLC αυτόματα διαφέρουν από τα χρονισμένα, καθώς οι μεταβάσεις περιορίζονται μόνο από ένα ρολόι, το ρολόι της κεντρικής μονάδας επεξεργασίας του PLC. Όλες αυτές οι μέθοδοι, παρά τις ακαδημαϊκές προσπάθειες, δεν έτυχαν ευρείας αποδοχής από τον βιομηχανικό κόσμο, κυρίως γιατί δεν έχουν ευρύ πεδίο εφαρμογών. Εκτός των διαφόρων τύπων αυτομάτων, στην βιβλιογραφία συναντώνται και άλλα εργαλεία μοντελοποίησης, όπως το CHARON, το Ptolemy και το Shift. Οι μέθοδοι αυτοί παρουσιάστηκαν κυρίως για τον σχεδιασμό, την μοντελοποίηση και την εξομοίωση υβριδικών συστημάτων, χωρίς όμως ιδιαίτερο βιομηχανικό προσανατολισμό.

Στην διατριβή αυτή αναπτύχθηκαν και παρουσιάστηκαν τα Γενικευμένα Αυτόματα (Global Automata) [2], ως μια νέα μέθοδος μοντελοποίησης μεγάλης γκάμας συστημάτων χωρίς περιορισμούς σε τυχόν ιδιομορφίες ή ιδιαίτερα χαρακτηριστικά. Η δομή των γενικευμένων αυτομάτων δανείζεται στοιχεία από άλλους τύπους αυτομάτων, όπως το γράφο ελέγχου με ένα πεπερασμένο σύνολο καταστάσεων και μεταβάσεις μεταξύ αυτών. Τα γενικευμένα αυτόματα μπορούν να μοντελοποιήσουν υβριδικά συστήματα συνδυάζοντας εξισώσεις, όρια και περιορισμούς από τα υβριδικά αυτόματα και εισόδους που υπόκεινται σε χρονικούς περιορισμούς από τα χρονισμένα και PLC αυτόματα. Επιπρόσθετα, εισάγονται νέες παράμετροι μοντελοποίησης, όπως πίνακας αρχικοποίησης σε κάθε μετάβαση, καθώς και ιεράρχηση των ενεργών γεγονότων σε κάθε κατάσταση. Το γεγονός ότι αποτελούν υπερσύνολο κάθε άλλου τύπου αυτομάτων αποδεικνύει ότι έχουν ευρύτερο φάσμα εφαρμογών σε σύγκριση με τους άλλους τύπους.

Ένας άλλος σημαντικός ανασχετικός παράγοντας στην υλοποίηση ακαδημαϊκών μεθόδων σε πραγματικά συστήματα αποτελεί το μέγεθος του χώρου κατάστασης. Τυπικά, οι ακαδημαϊκές εργασίες περιέχουν μερικές δεκάδες καταστάσεων και γεγονότων, ενώ τα βιομηχανικά συστήματα, όπως παρουσιάζεται στην [35],

περιέχουν δεκάδες χιλιάδες καταστάσεις. Συνεπώς, είναι εύκολα κατανοητό ότι η δημιουργία του μοντέλου ενός πραγματικού συστήματος από το μηδέν είναι πρακτικά ανέφικτη. Η μόνη εφικτή λύση προκύπτει από την σύνθεση των μοντέλων διαφόρων υποσυστημάτων με στόχο την δημιουργία του ολοκληρωτικού μοντέλου. Επιπρόσθετα, είναι συνήθης η ανάγκη σχεδίασης ενός νέου τμήματος, το οποίο θα πρέπει, συνδυαζόμενο με μια υπάρχουσα εγκατάσταση, να δίνει το επιθυμητό τελικό αποτέλεσμα [36]. Η λύση και σε αυτή την περίπτωση παραμένει η ίδια, δηλαδή η σύνθεση του μοντέλου του νέου τμήματος με το υπάρχον μοντέλο του συστήματος, ώστε το προκύπτον μοντέλο να εξεταστεί μέσω εξομοίωσης για την ικανοποίηση των αναγκαίων προδιαγραφών. Επομένως, η σύνθεση δύο ή περισσότερων ανεξάρτητων μοντέλων είναι ιδιαίτερος χρήσιμη στα βιομηχανικά συστήματα. Τα γενικευμένα αυτόματα προσφέρουν αυτή την δυνατότητα, όπως και οι άλλες παραδοσιακές μορφές αυτομάτων [28, 37, 38, 39, 40]. Όμως, η σύνθεση δύο αυτομάτων με m και n πλήθος καταστάσεων δημιουργεί ένα νέο αυτόματο με $m \times n$ καταστάσεις [3, 28], προκαλώντας έτσι υπέρμετρη αύξηση του χώρου κατάστασης. Τα γενικευμένα αυτόματα προσφέρουν ένα ισχυρό εργαλείο για την αποφυγή αυτής της αύξησης, το οποίο είναι η συγχώνευση καταστάσεων. Στην συγχώνευση καταστάσεων πραγματοποιείται συνένωση δύο ή περισσότερων καταστάσεων με στόχο την δημιουργία μιας νέας υπερ-κατάστασης η οποία θα έχει την ίδια ακριβώς συμπεριφορά με τις αρχικές καταστάσεις. Τα εργαλεία της σύνθεσης αυτομάτων και της συγχώνευσης καταστάσεων παρουσιάζονται αναλυτικά στο κεφάλαιο 6, ενώ στο κεφάλαιο 7 παρουσιάζονται τα εργαλεία υλοποίησης των γενικευμένων αυτομάτων σε βιομηχανικού τύπου λογισμικά πακέτα, καθώς και στον κυρίαρχο βιομηχανικό ελεγκτή, τον Προγραμματιζόμενο Λογικό Ελεγκτή.

Οι επόμενες δύο παράγραφοι περιέχουν μια συνοπτική βιβλιογραφική αναφορά. Η πρώτη αναφέρεται στις ανάγκες μοντελοποίησης των βιομηχανικών συστημάτων και τις διάφορες προσεγγίσεις αναφορικά με την θεωρία αυτομάτων και όχι μόνο. Τέλος, η δεύτερη παράγραφος παρουσιάζει, δημοσιευμένες στην βιβλιογραφία, προσεγγίσεις αυτόματης παραγωγής κώδικα για Προγραμματιζόμενους Λογικούς Ελεγκτές από το μοντέλο περιγραφής του συστήματος.

3.1 Η μοντελοποίηση βιομηχανικών συστημάτων στην βιβλιογραφία

Η μοντελοποίηση και η εξομοίωση είναι δύο βασικότατα εργαλεία στην προσπάθεια σχεδίασης υψηλής ποιότητας βιομηχανικών συστημάτων σε βέλτιστο χρόνο. Σε αυτή την κατεύθυνση κινείται και η εργασία [41], όπου παρουσιάζονται σκέψεις περί της ισοδυναμίας μοντέλων συστημάτων διακριτών γεγονότων. Γίνεται χρήση τεχνικών από τον χώρο του ταυτοχρονισμού (concurrency) και της Άλγεβρας Milner από τον χώρο των τηλεπικοινωνιών, και φαίνεται να πλεονεκτεί σε σύγκριση με προηγούμενες δουλειές. Αναφορικά με την θεωρία αυτομάτων, πολλές εργασίες έχουν δημοσιευτεί στην μοντελοποίηση βιομηχανικών συστημάτων, αλλά δεν περιορίζονται σε αυτά. Τα αυτόματα μπορούν επίσης να χρησιμοποιηθούν ως εργαλείο μοντελοποίησης στην πεδίο των ενσωματωμένων συστημάτων (embedded systems). Ειδικότερα, τα χρονισμένα αυτόματα έχουν χρησιμοποιηθεί ως μοντέλο ανάλυσης ιδιοτήτων συντονισμού σε αρχιτεκτονικές ενσωματωμένων συστημάτων [42], όπου η παρουσιαζόμενη μελέτη, εμπνευσμένη από πρακτικές της βιομηχανίας, παρουσιάζει ένα σύστημα αυτόματης πλοήγησης οχημάτων. Τα πεπερασμένα αυτόματα, η θεμελιώδη μορφή αυτομάτων, χρησιμοποιούνται στην [43] ως η βάση μιας αντικειμενοστραφούς προσέγγισης της κλασσικής θεωρίας εποπτικού ελέγχου. Το πεδίο εφαρμογής του παρουσιαζόμενου πλαισίου εξομοίωσης είναι τα ευέλικτα συστήματα παραγωγής (flexible manufacturing systems). Τα χρονισμένα αυτόματα επίσης χρησιμοποιούνται ως μέσο μοντελοποίησης των χρόνων λανθάνουσας κατάστασης ενός συστήματος, όπως αυτοί αναλύθηκαν στην προηγούμενη παράγραφο. Ο Supavatanakul κ.α. [44] αντιμετωπίζουν επίσης το πρόβλημα της διάγνωσης λαθών σε συστήματα διακριτών γεγονότων που αναπαρίστανται με χρονισμένα αυτόματα. Ο προτεινόμενος αλγόριθμος διάγνωσης αναγνωρίζει τυχόν δυσλειτουργίες συγκρίνοντας εισόδους και εξόδους με το μοντέλο του αυτομάτου. Συγκρινόμενη με άλλες, η δοθείσα μέθοδος δεν απαιτεί γνώση της αρχικής κατάστασης και συνεπώς μπορεί να εφαρμοστεί περιοδικά και αυτοβούλως. Ο αλγόριθμος έχει χρησιμοποιηθεί για διάγνωση λαθών και απομόνωσης αυτών σε ενεργοποιητές DAMADICS και όλα τα πειραματικά αποτελέσματα παρουσιάζονται αναλυτικά. Στην εργασία [45] παρουσιάζεται ένας αποτελεσματικός τρόπος μοντελοποίησης σύνθετων συστημάτων μέσω προκαθορισμένων καταστάσεων λειτουργίας. Η όλη προσέγγιση παρουσιάζεται με βιομηχανικό προσανατολισμό και τα αποτελέσματα των εξομοιώσεων κρίνονται ικανοποιητικά. Επιπρόσθετα, τα χρονισμένα αυτόματα έχουν χρησιμοποιηθεί για την μοντελοποίηση σύνθετων

προβλημάτων προγραμματισμού της παραγωγικής διαδικασίας, όπως για παράδειγμα τα γνωστά προβλήματα προγραμματισμού job-shop και open-shop. Μια άλλη προσέγγιση στην επίλυση παρόμοιων προβλημάτων παρουσιάζεται στην [46], όπου γίνεται χρήση προηγμένων εργαλείων εξακρίβωσης, όπως το UPAAL. Αναφορικά με το πρόβλημα προγραμματισμού παραγωγής job-shop, η εργασία [47] παρουσιάζει μια διαφορετική μεθόδευση μέσω ανάλυσης προσεγγισημότητας στα χρονισμένα αυτόματα. Στην εργασία βελτιώνεται η αποτελεσματικότητα του αλγορίθμου με αποκλεισμό μη βέλτιστων και επιπρόσθετων επιλύσεων από τον χώρο αναζήτησης. Τα πειραματικά αποτελέσματα αποδεικνύουν ότι, για διάφορα χαρακτηριστικά προβλήματα, η προτεινόμενη μέθοδος προδιαγράφει καταλληλότερη αλληλουχία ενεργειών. Οι Balduzzi και Kumar [32] μοντελοποιούν και ελέγχουν αυτοματοποιημένα συστήματα παραγωγής με χρήση υβριδικών αυτομάτων. Συγκεκριμένα, κάθε βασική λειτουργία του συστήματος αναπαρίσταται από ένα στοιχειώδες υβριδικό αυτόματο και κατόπιν σύνθεσης όλων των αυτομάτων προκύπτει το μοντέλο του όλου συστήματος. Οι Guo κ.α. [48] παρουσίασαν ένα πρωτότυπο σύστημα προσανατολισμένο στα συστήματα παραγωγής, το SimOpt. Το SimOpt αποτελείται από ένα υποσύστημα εικονικής εξομίωσης μοντέλου βασισμένο σε αντικειμενοστραφείς τεχνικές και από ένα υποσύστημα βελτιστοποίησης που χρησιμοποιεί έναν υβριδικό αλγόριθμο βελτιστοποίησης. Οι προθέσεις των συγγραφέων είναι να βελτιώσουν περαιτέρω το όλο σύστημα ώστε να αντιμετωπίζει το ίδιο αποδοτικά και πιο σύνθετα συστήματα. Μια διαφορετική προσέγγιση μοντελοποίησης και ανάλυσης σύνθετων βιομηχανικών συστημάτων δίνεται στην [49]. Η αποτελεσματικότητα της μεθόδου κρίνεται στην πράξη, με εφαρμογή της σε ένα εργοστάσιο της Γαλλίας. Σε αυτή την εργασία, περιγράφεται ένα γενικό πλαίσιο για φυσικά καταμεμημένα και σύνθετα συστήματα βασισμένο στο υπόδειγμα Holonic Multi-Agent. Τέλος, μια λεπτομερής βιβλιογραφική αναζήτηση σχετικά με τα βιομηχανικά συστήματα παραγωγής παρουσιάζεται στην [50] αναφορικά με μέτρηση και βελτίωση παραγωγικότητας. Επίσης περιλαμβάνει μια επισκόπηση των διαθέσιμων εμπορικών εργαλείων επαλήθευσης της λειτουργικής αρτιότητας και παραγωγής.

3.2 Αυτόματη παραγωγή κώδικα για Προγραμματιζόμενους Λογικούς Ελεγκτές στην βιβλιογραφία

Όπως ειπώθηκε και παραπάνω ο Προγραμματιζόμενος Λογικός Ελεγκτής είναι ο κυρίαρχος ελεγκτής στον βιομηχανικό χώρο και εφόσον η δημιουργία προγραμμάτων

για ΠΛΕ είναι μια επίπονη, χρονοβόρα και δαπανηρή δραστηριότητα [51, 52], έχουν παρουσιαστεί διάφορες τεχνικές αυτόματης παραγωγής εκτελέσιμου προγράμματος από το μοντέλο του συστήματος. Στην βιβλιογραφία υπάρχουν διάφορες προσεγγίσεις που χρησιμοποιούν διαφορετικές μεθόδους μοντελοποίησης ως αρχικό μοντέλο και διαφορετικές γλώσσες προγραμματισμού για την παραγωγή κώδικα. Στην εργασία [53] παρουσιάζονται δύο τρόποι επίλυσης του προβλήματος υλοποίησης: α) η «από πάνω προς τα κάτω» προσέγγιση, με αυτόματη παραγωγή κώδικα από το υπάρχον μοντέλο ενός συστήματος, και β) η «από κάτω προς τα πάνω» προσέγγιση, που ξεκινά με χειροκίνητο λογικό προγραμματισμό και μετέπειτα μετατροπή αυτού σε αυτόματο ώστε να εξεταστεί μέσω λογισμικών αλγορίθμων εξομοίωσης και επαλήθευσης.

Το εργαλείο Supremica περιέχει αλγορίθμους διακρίβωσης και σύνθεσης [54], ενώ επιπρόσθετα, έχει την δυνατότητα αυτόματης παραγωγής κώδικα για ένα πλήθος γλωσσών προγραμματισμού, όπως ANSI C και ST και IL από το πρότυπο IEC 61131-3. Η εργασία [55] επικεντρώνεται στην φάση της υλοποίησης, όπου τα παραχθέντα μοντέλα αυτομάτων μετατρέπονται σε εκτελέσιμο κώδικα κατάλληλο για εκτέλεση σε συνήθη βιομηχανικά PLC. Με αυτό τον τρόπο, οι χρόνοι ανάπτυξης και υλοποίησης έργων βιομηχανικού αυτοματισμού μειώνονται σημαντικά. Μια αντίστοιχη προσέγγιση παρουσιάζεται και στην εργασία [56] δίνοντας ένα μηχανισμό μετασχηματισμού του δοθέντος μοντέλου αυτομάτου σε εκτελέσιμο PLC κώδικα.

Μια απόπειρα υλοποίησης της θεωρίας εποπτικού ελέγχου σε PLC πραγματοποιείται στην εργασία [57]. Το παραγόμενο PLC πρόγραμμα είναι κατάλληλα δομημένο ώστε να επιτρέπει την εύκολη χειροκίνητη τροποποίηση ή τον εντοπισμό τυχόν λαθών. Μια άλλη εργασία παρουσιάζει την διαδικασία παραγωγής κώδικα PLC για παρακολούθηση μιας διεργασίας από το μοντέλο διακριτών γεγονότων, που αναπαρίσταται είτε ως ένα πεπερασμένο αυτόματο, είτε από άλλο πρόγραμμα PLC [58]. Με χρήση των γλωσσών προγραμματισμού του προτύπου IEC 61131-3 [59] για την μοντελοποίηση διεργασιών, μικραίνει το χάσμα μεταξύ πραγματικής βιομηχανίας και ακαδημαϊκού χώρου. Οι προγραμματιζόμενοι λογικοί ελεγκτές αποτελούν πλατφόρμα υλοποίησης εποπτών και στην εργασία [60], όπου παρουσιάζεται ένας νέος αλγόριθμος μετατροπής που μετατρέπει απευθείας έναν επόπτη που αναπαρίσταται από ένα πεπερασμένο αυτόματο σε διάγραμμα Ladder Logic. Τέλος, στην [61] παρουσιάζεται μια προσπάθεια μείωσης του χρόνου ανάπτυξης μέσω αυτοματοποίησης του έργου

σχεδίασης του λογικού ελεγκτή και της υλοποίησης αυτού σε PLC. Σαν μέθοδο μοντελοποίησης χρησιμοποιείται το t-MPSG, που αποτελεί ένα εκτεταμένο πεπερασμένο αυτόματο. Το t-MPSG εποπτεύει τις μεταβάσεις από μια κατάσταση σε μια άλλη σε συνδυασμό με τις απαραίτητες προϋποθέσεις του συστήματος και χρησιμοποιείται για μοντελοποίηση του λογικού ελεγκτή. Έπειτα, ο λογικός ελεγκτής μπορεί να μετατραπεί σε γλωσσικό κώδικα και τέλος σε εκτελέσιμο κώδικα βάσει του IEC προτύπου.

Μια προσπάθεια υλοποίησης, σε Προγραμματιζόμενο Λογικό Ελεγκτή, μερών λογισμικού πραγματικού χρόνου βάσει του διεθνούς προτύπου IEC 61499 δίνεται στην [62]. Το πρότυπο IEC 61499 [63], που παρουσιάστηκε από την διεθνή επιτροπή, υιοθετεί την έννοια του function block και προτείνει μια ανοιχτή αρχιτεκτονική για καταναμημένα βιομηχανικά συστήματα ελέγχου. Στην εργασία [64] παρουσιάζεται ένας οδηγός μετατροπής υπαρχόντων function block βάσει του προτύπου IEC 61131 σε function block του νέου προτύπου. Ένα λογισμικό εργαλείο για την υλοποίηση εφαρμογών βάσει του προτύπου IEC 61499 σε ενσωματωμένους ελεγκτές παρουσιάζεται στην [65].

Τα παραπάνω αποτελούν βιβλιογραφικά παραδείγματα και υπάρχει έντονη ακαδημαϊκή δραστηριότητα στο συγκεκριμένο ερευνητικό πεδίο. Όμως, η εργασία [66] εξετάζει τις προκλήσεις του νέου προτύπου ώστε να γίνει αποδεκτό τόσο από κατασκευαστές όσο και από τελικούς χρήστες. Ένας αντίστοιχος προβληματισμός για τα ανοιχτά ζητήματα του προτύπου και ανάλυση πιθανών εναλλακτικών σχεδιάσεων παρουσιάζεται στην [67]. Συμπερασματικά, το πρότυπο IEC 61499 ίσως αποτελεί το αύριο του βιομηχανικού αυτοματισμού στην σχεδίαση καταναμημένων ελεγκτών, μένει μόνο να αποδειχτεί και στην πράξη στην βιομηχανία. Στην προσπάθεια γεφύρωσης της απόστασης μεταξύ θεωρίας και εφαρμογών, μια διαφοροποιημένη τεχνική υλοποίησης με χρήση δικτύων function blocks παρουσιάζεται στην [68].

Η εργασία [69], προτείνει την γλώσσα UML (Unified Modeling Language) ως εργαλείο μοντελοποίησης καταναμημένων συστημάτων ελέγχου βάσει του προτύπου IEC 61499, καθώς η κοινή αντικειμενοστραφής αρχιτεκτονική δομή επιτρέπει την αυτόματη μετατροπή μοντέλων UML σε μοντέλα του προτύπου. Αντίστοιχα στην [70], παρουσιάζεται μια μεθοδολογία μοντελοποίησης, επαλήθευσης και αυτόματης παραγωγής κώδικα για βιομηχανικούς ελεγκτές. Η μέθοδος περιλαμβάνει ένα σύνολο

απαιτήσεων βάσει της μηχανής καταστάσεων UML, και μια τεχνική παραγωγής εκτελέσιμου προγράμματος. Η μηχανή καταστάσεων UML επιλέχθηκε καθώς αναγνωρίζεται ευρέως ως κατάλληλη μέθοδος γραφής προδιαγραφών. Οι παραγόμενες προδιαγραφές λειτουργίας μπορούν να επικυρωθούν από τον χρήστη, να επαληθευτούν μέσω ενός κατάλληλου εργαλείου και τέλος να μετατραπούν αυτόματα σε κώδικα μιας εκ των γλωσσών προγραμματισμού του προτύπου IEC 61131-3.

Όλες οι παραπάνω προσεγγίσεις στο θέμα της αυτοματοποιημένης παραγωγής κώδικα για τον αυτοματισμό μιας διεργασίας χρησιμοποιούν κάποια από τις γλώσσες του προτύπου IEC 61131-3 ή το πρότυπο IEC 61499. Ολοκληρώνοντας την αναφορά στα διάφορα εργαλεία παραγωγής κώδικα για έλεγχο βιομηχανικών συστημάτων, σημειώνεται ότι στο κεφάλαιο 7 παρουσιάζεται το αντίστοιχο εργαλείο παραγωγής κώδικα για PLC από μοντέλο γενικευμένων αυτομάτων. Το εργαλείο έχει την μορφή ενός οδηγού βημάτων για την αυτοματοποιημένη παραγωγή κώδικα προς όλες τις γλώσσες προγραμματισμού του προτύπου IEC 61131-3, οι οποίες αποτελούν τις γλώσσες προγραμματισμού της πλειονότητας των εμπορικών βιομηχανικών ελεγκτών τύπου PLC.

Κεφάλαιο 4^ο

Θεωρία Αυτομάτων

Τα αυτόματα συναντώνται στην επιστημονική βιβλιογραφία σε διάφορες μορφές, όπως πεπερασμένα αυτόματα, υβριδικά αυτόματα, χρονισμένα αυτόματα και PLC αυτόματα. Ο ευρύτερα χρησιμοποιούμενος τύπος είναι τα πεπερασμένα αυτόματα. Παρακάτω παρατίθενται οι ορισμοί όλων των αναφερόμενων μορφών αυτομάτων καθώς και τα κυριότερα χαρακτηριστικά τους.

4.1 Πεπερασμένα Αυτόματα (Finite Automata)

Τα πεπερασμένα αυτόματα είναι η πιο απλή μαθηματική μέθοδος για την αναπαράσταση Συστημάτων Διακριτών Γεγονότων και αποτελούνται από ένα σύνολο καταστάσεων και μεταβάσεων μεταξύ αυτών. Κάθε χρονική στιγμή το αυτόματο βρίσκεται σε μία από τις καταστάσεις. Μια μετάβαση μεταξύ δύο καταστάσεων προκαλείται από ένα γεγονός. Κάθε γεγονός θεωρείται στιγμιαίο και συνεπώς δεν μπορεί να διακοπεί. Επιπροσθέτως, δύο γεγονότα δεν μπορούν να συμβούν ταυτόχρονα και μόνο τα γεγονότα της τρέχουσας κατάστασης είναι εκτελέσιμα.

Ορισμός 1. Πεπερασμένα Αυτόματα.

Ένα ντετερμινιστικό αυτόματο πεπερασμένων καταστάσεων περιγράφεται από την τετράδα παραμέτρων: $A = \langle Q_A, \Sigma_A, i_A, \delta_A \rangle$ όπου:

- Q_A είναι ένα πεπερασμένο μη κενό σύνολο που καλείται σύνολο καταστάσεων,
- Σ_A είναι ένα πεπερασμένο μη κενό σύνολο γεγονότων που καλείται αλφάβητο,
- $i_A \in Q_A$ είναι η αρχική κατάσταση και προσδιορίζεται με ένα βέλος και τέλος,
- $\delta_A: Q_A \times \Sigma_A \rightarrow Q_A$ είναι η συνάρτηση μεταβάσεων, που περιγράφει τη μετάβαση μεταξύ δύο καταστάσεων που προκαλείται από ένα γεγονός.

Το σχήμα 4.1 παρουσιάζει ένα απλό πεπερασμένο αυτόματο με δύο καταστάσεις (q_1 και q_2) και δύο γεγονότα (a και b). Η κατάσταση q_1 είναι η αρχική κατάσταση του αυτομάτου.

Σχήμα 4.1. Πεπερασμένο αυτόματο.

Στην βιβλιογραφία συναντάται επίσης ένας παρόμοιος ορισμός για τα πεπερασμένα αυτόματα που έχει μια επιπλέον παράμετρο, το σύνολο των μαρκκαρισμένων καταστάσεων $M_A \subseteq Q_A$. Οι μαρκκαρισμένες καταστάσεις είναι συνήθως οι επιθυμητές και χρησιμοποιούνται για να υποδηλώσουν πότε μια διεργασία έχει ολοκληρωθεί. Το σύνολο αυτό παραμέτρων μπορεί να συναντηθεί επίσης ως σύνολο τελικών καταστάσεων.

Συχνά δύο αυτόματα αλληλεπιδρούν μεταξύ τους σε ένα ευρύτερο σύστημα. Τα κοινά γεγονότα σηματοδοτούν την αλληλεπίδραση και πρέπει να έχουν κοινές ιδιότητες στα δύο αυτόματα. Η αλληλεπίδραση μεταξύ δύο αυτομάτων A και B μοντελοποιείται ως η σύγχρονη σύνθεση των δύο, $A \parallel B$ όπου:

$$A \parallel B = \langle Q_A \times Q_B, \Sigma_A \cup \Sigma_B, \langle i_A, i_B \rangle, \delta_{A \parallel B} \rangle,$$

$$\text{με } \delta_{A \parallel B}(\langle q_A, q_B \rangle, \sigma) = \begin{cases} \delta_A(q_A, \sigma) \times \delta_B(q_B, \sigma), & \sigma \in \Sigma_A \cup \Sigma_B \\ \delta_A(q_A, \sigma) \times q_B, & \sigma \in \Sigma_A - \Sigma_B \\ q_A \times \delta_B(q_B, \sigma), & \sigma \in \Sigma_B - \Sigma_A \end{cases}$$

και : $q_A \in Q_A, q_B \in Q_B$.

Ορισμός 2. Εκτέλεση – Υπολογισμός.

Η εκτέλεση ή ισοδύναμα ο υπολογισμός ενός αυτομάτου A σε μία ακολουθία εισόδου $u = \sigma_1, \sigma_2, \dots, \sigma_n$ είναι μια μοναδική ακολουθία καταστάσεων $y = s_1, s_2, \dots, s_{n+1}$ τέτοια ώστε $s_1 = i_A$ και για κάθε $j \leq n, \delta(\sigma_j) = s_{j+1}$.

Ο ορισμός αυτός περιγράφει τη συμπεριφορά του αυτομάτου σε μία ακολουθία εισόδου και την παραγωγή εξόδου. Αξίζει να σημειωθεί ότι η πρώτη κατάσταση της εξόδου είναι η αρχική κατάσταση του αυτομάτου και συνεπώς η ακολουθία εξόδου είναι μεγαλύτερη από την ακολουθία εισόδου κατά ένα στοιχείο.

4.2 Υβριδικά Αυτόματα (Hybrid Automata)

Τα υβριδικά αυτόματα προτάθηκαν σαν μια φορμαλιστική μέθοδος μοντελοποίησης υβριδικών συστημάτων και αποτελούν ένα υπέρ-σύνολο των πεπερασμένων αυτομάτων. Τα υβριδικά αυτόματα έχουν επιπλέον συνεχή δυναμική συμπεριφορά που αντιστοιχείται σε κάθε κατάσταση του αυτομάτου. Η δυναμική συμπεριφορά εκφράζεται μέσω διαφορικών εξισώσεων. Το σχήμα 4.2 απεικονίζει ένα τυπικό παράδειγμα ενός υβριδικού αυτομάτου που μοντελοποιεί έναν οικιακό θερμοστάτη.

Σχήμα 4.2. Υβριδικό αυτόματο θερμοστάτη.

Η πραγματική μεταβλητή x αναπαριστά τη θερμοκρασία. Οι δύο καταστάσεις ονομάζονται *On* και *Off* και κάθε μία έχει μια συνάρτηση βάσει της οποίας η θερμοκρασία αυξάνεται ή μειώνεται αντίστοιχα. Αρχικά η θερμοκρασία είναι 20° βαθμοί και ο θερμαντήρας είναι κλειστός. Σύμφωνα με την συνθήκη μετάβασης, η θερμοκρασία πέφτει σύμφωνα με την διαφορική εξίσωση $\dot{x} = -0.1 \cdot x$, μέχρι το κάτω όριο των 18° βαθμών. Τότε, ο θερμαντήρας ανάβει και η θερμοκρασία αυξάνεται, σύμφωνα με την διαφορική εξίσωση $\dot{x} = 10 - 0.1 \cdot x$, μέχρι το άνω όριο των 23° βαθμών όπου ο θερμαντήρας ξανασβήνει. Το απλό αυτό παράδειγμα δηλώνει την αλληλεπίδραση των διακριτών μεταβλητών (καταστάσεις *On* και *Off*) με τις συνεχείς συνιστώσες (διαφορετική διαφορική εξίσωση σε κάθε κατάσταση). Ακολουθεί ένας απλός ορισμός των υβριδικών αυτομάτων.

Ορισμός 3. Υβριδικά Αυτόματα.

Ένα υβριδικό αυτόματο ορίζεται ως $H = \langle Q, X, Init, f, D, E, G, R \rangle$ όπου:

- Q είναι ένα πεπερασμένο μη κενό σύνολο διακριτών μεταβλητών που καλείται σύνολο καταστάσεων,
- X είναι ένα πεπερασμένο μη κενό σύνολο συνεχών μεταβλητών,
- $Init \subseteq Q \times X$ είναι το σύνολο των αρχικών συνθηκών,
- $f: Q \times X \rightarrow X$ είναι μια διανυσματική αντιστοίχιση,
- $D: Q \rightarrow P(X)$ είναι το πεδίο ορισμού,
- $E \subset Q \times Q$ είναι το σύνολο των μεταβάσεων,
- $G: E \rightarrow P(X)$ είναι οι οριακές συνθήκες και
- $R: E \times X \rightarrow P(X)$ είναι ο χάρτης αρχικοποίησης.

Στην βιβλιογραφία υπάρχουν επίσης μελέτες σε μια υποκατηγορία των υβριδικών αυτομάτων, τα Ζήνων υβριδικά αυτόματα. Η μόνη διαφοροποίηση είναι ότι τα Ζήνων υβριδικά αυτόματα δέχονται εκτελέσεις με μη πεπερασμένο αριθμό μεταβάσεων σε πεπερασμένο χρόνο. Επιστημονικό ενδιαφέρον προκύπτει κατά την εξομοίωση τέτοιων συστημάτων, καθώς ένας μεγάλος αριθμός μεταβάσεων εμφανίζεται σε ένα μικρό χρονικό διάστημα.

4.3 Χρονισμένα Αυτόματα (Timed Automata)

Τα χρονισμένα αυτόματα πρωτοπαρουσιάστηκαν από τους Allur και Drill το 1994 και έκτοτε αποτελούν ένα από τα πιο ευρέως χρησιμοποιούμενα εργαλεία μοντελοποίησης συστημάτων πραγματικού χρόνου. Είναι μια προέκταση των πεπερασμένων αυτομάτων, καθώς κάθε μετάβαση περιορίζεται από ένα πρόσθετο σύνολο ρολογιών.

Ορισμός 4. Χρονισμένα Αυτόματα.

Ένα χρονισμένο αυτόματο ορίζεται από την πεντάδα $A = \langle Q, \Sigma, i, C, E \rangle$ όπου:

- Q είναι ένα πεπερασμένο μη κενό σύνολο μεταβλητών που καλείται σύνολο καταστάσεων,
- Σ είναι ένα πεπερασμένο μη κενό σύνολο γεγονότων που καλείται αλφάβητο,
- $i_A \in Q_A$ είναι η αρχική κατάσταση,

- C είναι ένα πεπερασμένο σύνολο ρολογιών και
- $E \subseteq Q \times Q \times \Sigma \times 2^C \times \Phi(C)$ είναι το σύνολο των μεταβάσεων όπου το $\Phi(C)$ ορίζεται ως το σύνολο των χρονικών περιορισμών (συνθήκες επί των τιμών των ρολογιών).

Κάθε διάνυσμα $(q, q', a, \lambda, \varphi)$ αναπαριστά τη μετάβαση από την κατάσταση q στην κατάσταση q' που προκαλείται από το σήμα εισόδου a . Το σύνολο $\lambda \subseteq C$ δίνει τα ρολόγια που θα αρχικοποιηθούν κατά την μετάβαση αυτή και φ είναι ο χρονικός περιορισμός.

4.4 PLC Αυτόματα

Τα PLC αυτόματα είναι ένα πολύ χρήσιμο εργαλείο για την περιγραφή καταναεμημένων συστημάτων πραγματικού χρόνου που ελέγχονται μέσω ενός Προγραμματιζόμενου Λογικού Ελεγκτή (Programmable Logic Controller). Τα PLC αυτόματα εισήχθησαν το 1997 από τον Dierks και αποτελούν μια υποκατηγορία των χρονισμένων αυτομάτων. Η κύρια διαφοροποίηση είναι ότι τα PLC αυτόματα υπόκεινται σε περιορισμούς πάνω σε ένα μόνο ρολόι, το ρολόι του επεξεργαστή του PLC.

Ορισμός 5. PLC Αυτόματα.

Ένα PLC αυτόματο περιγράφεται από ένα σύνολο εννέα παραμέτρων $A = \langle Q, \Sigma, q_0, \delta, \varepsilon, S_e, S_t, \Omega, \omega \rangle$ όπου:

- Q είναι ένα πεπερασμένο μη κενό σύνολο μεταβλητών που καλείται σύνολο καταστάσεων,
- Σ είναι ένα πεπερασμένο μη κενό σύνολο εισόδων,
- q_0 είναι η αρχική κατάσταση,
- $\delta: Q \times \Sigma \rightarrow Q$ είναι η συνάρτηση μετάβασης,
- $\varepsilon > 0$ είναι το άνω όριο για έναν κύκλο,
- $S_e: Q \rightarrow P(\Sigma) \setminus \{0\}$ είναι μία συνάρτηση που αντιστοιχεί σε κάθε κατάσταση ένα σύνολο εισόδων που θα καθυστερηθούν,
- $S_t: Q \rightarrow \mathbb{R}_{\geq 0}$ είναι μια συνάρτηση που αντιστοιχεί σε κάθε κατάσταση τον χρόνο της καθυστέρησης,
- Ω είναι ένα πεπερασμένο μη κενό σύνολο εξόδων και

- $\omega: Q \rightarrow \Omega$ είναι η συνάρτηση εξόδου.

Σχήμα 4.3. Ένα απλό PLC αυτόματο.

Το σχήμα 4.3 παρουσιάζει ένα παράδειγμα ενός PLC αυτομάτου. Το αυτόματο έχει τρεις καταστάσεις $\{q_0, q_1, q_2\}$, δύο εισόδους $\{i_1, i_2\}$ και την q_0 ως αρχική κατάσταση. Η σημειολογία σε κάθε κατάσταση αναπαριστά το χρόνο που θα καθυστερήσει η κάθε είσοδος. Για παράδειγμα, στην κατάσταση q_2 η είσοδος i_2 θα καθυστερηθεί για πέντε μονάδες χρόνου ενώ η i_1 δεν θα καθυστερηθεί καθόλου. Αντίστοιχα στην κατάσταση q_0 και οι δύο εισοδοί θα καθυστερηθούν για τρεις μονάδες χρόνου.

4.5 Λογισμικά πακέτα μοντελοποίησης Συστημάτων Διακριτών Γεγονότων

Στο κεφάλαιο αυτό παρουσιάζονται τα σημαντικότερα λογισμικά εργαλεία που έχουν δημιουργηθεί για την μοντελοποίηση συστημάτων διακριτών γεγονότων. Όλα έχουν αναπτυχθεί από ερευνητικά ινστιτούτα και Πανεπιστήμια και είναι ελεύθερα στη διάθεση του κοινού για μη κερδοσκοπική χρήση.

4.5.1 JGrafchart

Η JGrafchart είναι μια υλοποίηση της Grafchart [71], μιας γραφικής γλώσσας προγραμματισμού για εφαρμογές ακολουθιακού ελέγχου. Η όλη υλοποίηση έχει γίνει σε Java και είναι βασισμένη στη Grafcet ή αλλιώς Sequential Function Charts (SFC),

στα δίκτυα Petri, στα διαγράμματα καταστάσεων και σε γλώσσες προγραμματισμού υψηλού επιπέδου. Η JGrafchart απαρτίζεται από δύο μέρη:

- το εργαλείο γραφικής σχεδίασης και
- το σύστημα εκτέλεσης.

Οι δυνατότητες για εξομοίωση και επαλήθευση είναι τα κύρια προτερήματα της JGrafchart, ιδίως για Grafcet-SFC που αποτελούν πρότυπο της IEC 31131-3 ως γλώσσα προγραμματισμού για Προγραμματιζόμενους Λογικούς Ελεγκτές. Στο σχήμα 4.4 απεικονίζεται μια εικόνα του λογισμικού εργαλείου.

Σχήμα 4.4. Screenshot¹ από το λογισμικό πακέτο JGrafchart.

4.5.2 UPPAAL

Το UPPAAL [72] είναι ένα λογισμικό εργαλείο για την μοντελοποίηση, εξομοίωση και επαλήθευση συστημάτων πραγματικού χρόνου. Το εργαλείο αυτό είναι κατάλληλο για συστήματα που μπορούν να μοντελοποιηθούν σαν συλλογές διεργασιών με πεπερασμένη λογική ελέγχου και ρολόγια πραγματικού χρόνου. Οι επιμέρους

¹ «Οι υπολογιστές είναι η μόνη επιστήμη που οι όροι της δεν προέρχονται από τα Αρχαία Ελληνικά», οπότε γίνεται χρήση του αντίστοιχου αγγλικού όρου που έχει επικρατήσει.

διεργασίες επικοινωνούν μεταξύ τους είτε μέσω διαύλων επικοινωνίας, είτε μέσω κοινών μεταβλητών. Το UPPAAL αποτελείται από τρία επιμέρους εργαλεία:

- το εργαλείο περιγραφής
- το εργαλείο εξομοίωσης
- και το εργαλείο επαλήθευσης μοντέλου.

Το εργαλείο περιγραφής απεικονίζει τη συμπεριφορά του συστήματος σαν δίκτυα αυτομάτων με ρολόγια και μεταβλητές. Το εργαλείο εξομοίωσης χρησιμοποιείται για επαλήθευση, δίνοντας την δυνατότητα για υπολογισμό πιθανών εκτελέσεων ενός συστήματος. Τέλος, το εργαλείο επαλήθευσης μοντέλου χρησιμοποιείται για αυτόματη πιστοποίηση κρίσιμων παραμέτρων ενός συστήματος. Η όλη πιστοποίηση γίνεται με ανάλυση του χώρου κατάστασης ενός συστήματος και της ελεγκτότητας αυτού. Το UPPAAL χρησιμοποιείται κυρίως σε εφαρμογές ελεγκτών πραγματικού χρόνου και σε πρωτόκολλα επικοινωνίας. Στο σχήμα 4.5 παρουσιάζεται ένα στιγμιότυπο οθόνης του προγράμματος UPPAAL.

Σχήμα 4.5. Screenshot από το λογισμικό πακέτο UPPAAL.

4.5.3 DiaGen

Το DiaGen [73] είναι ένα σύστημα για την εύκολη δημιουργία εργαλείων σχεδίασης διαγραμμάτων ισχυρών δυνατοτήτων. Επιγραμματικά αναφέρεται η δημιουργία εργαλείων σχεδίασης για:

- διαγράμματα κλιμακωτής λογικής (Ladder logic diagrams),
- διαγράμματα για ιεραρχικά δίκτυα Petri,
- διαγράμματα καταστάσεων και
- διαγράμματα ροής (flow charts).

Σχήμα 4.6. Screenshot από το λογισμικό πακέτο DiaGen.

Σχήμα 4.7. Screenshots από το λογισμικό πακέτο DiaGen.

Τα παραγόμενα εργαλεία μπορούν να χρησιμοποιηθούν και ως εργαλεία εξομοίωσης και επαλήθευσης συστημάτων. Τα σχήματα 4.6 και 4.7 απεικονίζουν παραδείγματα χρήσης για διάφορα διαγράμματα.

4.5.4 KRONOS

Το KRONOS [74] είναι ένα λογισμικό εργαλείο επαλήθευσης, για συστήματα πραγματικού χρόνου. Είναι βασισμένο στη θεωρία των χρονισμένων αυτομάτων και της χρονικής λογικής (temporal logic). Συγκεκριμένα, τα μέρη ενός συστήματος πραγματικού χρόνου μοντελοποιούνται ως χρονισμένα αυτόματα, ενώ οι απαιτήσεις επαλήθευσης εκφράζονται μέσω της χρονικής λογικής TCTL. Μερικές από τις κύριες λειτουργίες του KRONOS είναι: εμπρόσθια και οπίσθια ανάλυση, βελτιστοποίηση χρήσης των ρολογιών του μοντέλου, δημιουργία παραδειγμάτων εκτέλεσης και δημιουργία διαγραμμάτων αποφάσεων.

Σχήμα 4.8. Screenshots από το λογισμικό πακέτο KRONOS.

4.5.5 HyTech

Το HyTech [75] είναι ένα λογισμικό εργαλείο για την ανάλυση ενσωματωμένων συστημάτων. Τα διάφορα μέρη του συστήματος μοντελοποιούνται ως γραμμικά υβριδικά αυτόματα, δηλαδή υβριδικά αυτόματα των οποίων οι συνεχείς συνιστώσες διέπονται από γραμμικές συναρτήσεις των μεταβλητών και των παραγώγων τους. Το HyTech χρησιμοποιείται κυρίως για τον υπολογισμό των συνθηκών κάτω από τις οποίες το σύστημα ικανοποιεί τις δεδομένες απαιτήσεις. Η ικανοποίηση των

απαιτήσεων επαληθεύεται με χρήση συμβολικών μοντέλων ελέγχου. Σε περίπτωση αποτυχίας επαλήθευσης, το HyTech παράγει κώδικα για την διάγνωση του λάθους. Η δομή του λογισμικού πακέτου HyTech φαίνεται σχηματικά στο σχήμα 4.9.

Σχήμα 4.9. Σχηματικά η δομή του λογισμικού πακέτου HyTech.

Σχήμα 4.10. Screenshot από το λογισμικό πακέτο Moby/PLC.

4.5.6 Moby/PLC

Το Moby/PLC [76] υποστηρίζει την ανάπτυξη κατανεμημένων προγραμμάτων ελέγχου πραγματικού χρόνου για Προγραμματιζόμενους Λογικούς Ελεγκτές. Αρχικά, το σύστημα μοντελοποιείται ως ένα δίκτυο από PLC-αυτόματα με χρήση του βασικού εργαλείου του Moby/PLC, ενός γραφικού σχεδιαστή. Στη συνέχεια το σύστημα μπορεί να επεξεργαστεί από τα υπόλοιπα εργαλεία του λογισμικού πακέτου Moby/PLC. Ενδεικτικά αναφέρουμε τις λειτουργίες εξομοίωσης και επαλήθευσης του όλου συστήματος. Τέλος, ένας κατάλληλος μεταγλωττιστής παράγει κώδικα για PLCs σε μορφή δομημένου κειμένου (structured text). Στιγμιότυπα του παραθυρικού περιβάλλοντος του Moby/PLC απεικονίζονται στο σχήμα 4.10.

4.5.7 Shift

Το Shift [77, 78] είναι μια γλώσσα προγραμματισμού για την περιγραφή δυναμικών δικτύων αποτελούμενων από υβριδικά αυτόματα. Τέτοια συστήματα αποτελούνται από διάφορα μέρη, τα οποία μπορεί να δημιουργούνται, να συνδέονται μεταξύ τους ή και να καταστρέφονται καθώς το σύστημα εξελίσσεται. Τα διάφορα μέρη παρουσιάζουν υβριδική συμπεριφορά, αποτελούμενη από φάσεις συνεχούς μεταβολής που διαχωρίζονται από διακριτές μεταβάσεις. Αλληλεπίδραση μεταξύ των μερών μπορεί να γίνει μέσω των εισόδων και των εξόδων κάθε μέρους, χωρίς όμως να αποκλείεται η ανεξαρτησία κάποιου τμήματος. Το Shift από την παρουσίαση του είχε βρει κύριο χώρο εφαρμογής τα συστήματα διαχείρισης κυκλοφορίας σε αυτοκινητοδρόμους. Μια οπτική απεικόνιση του αντίστοιχου λογισμικού πακέτου παρουσιάζεται στο σχήμα 4.11.

4.5.8 Supremica

Στο σχήμα 4.12 εικονίζεται ένα παράδειγμα από το λογισμικό πακέτο Supremica [79]. Το Supremica είναι ένα λογισμικό εργαλείο για μεθοδική σύνθεση και επαλήθευση συστημάτων ελέγχου. Συνήθως χρησιμοποιείται δημιουργώντας μοντέλο του υπό έλεγχο συστήματος και των επιθυμητών αποκρίσεων κλειστού βρόχου. Ακολούθως το Supremica παράγει έναν κατάλληλο επόπτη που να ικανοποιεί τις δοθείσες προδιαγραφές. Τέλος, το Supremica έχει την δυνατότητα παραγωγής εκτελέσιμου κώδικα για κάθε συμβατό, κατά IEC 61131, PLC.

Σχήμα 4.11. Screenshot από το λογισμικό πακέτο Shift.

Σχήμα 4.12. Screenshot από το λογισμικό πακέτο Supremica.

Κεφάλαιο 5^ο

Γενικευμένα Αυτόματα (*Global Automata*)

5.1 Γενικευμένα Αυτόματα

Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, οι υπάρχουσες μέθοδοι μοντελοποίησης Συστημάτων Διακριτών Γεγονότων (Σ.Δ.Γ.) δεν καλύπτουν πλήρως τις ανάγκες των σύγχρονων βιομηχανικών συστημάτων. Η αυξημένη πολυπλοκότητα των γραμμών παραγωγής σε συνδυασμό με την αδυναμία κάλυψης από κάποια μέθοδο μεγάλου φάσματος συστημάτων, οδήγησε στην αναζήτηση νέων μεθόδων μοντελοποίησης που δεν θα εξαρτώνται από τα ιδιαίτερα χαρακτηριστικά του εκάστοτε συστήματος. Η προσπάθεια αυτή απέδωσε καρπούς και η νέα προτεινόμενη μέθοδος, υπό τον τίτλο Γενικευμένα Αυτόματα, παρουσιάστηκε αρχικά στο παγκόσμιο συνέδριο ελέγχου IFAC 2005 ενώ δημοσιεύτηκε και σε διεθνές επιστημονικό περιοδικό [2].

Η νέα αυτή μέθοδος βασίζεται στα αυτόματα και ιδίως στα χρονισμένα και υβριδικά αυτόματα. Μερικές έννοιες από αυτές τις δύο μορφές αυτομάτων έχουν διατηρηθεί, ενώ έχουν προστεθεί νέες ώστε να καλύψουν τις όποιες αδυναμίες και να δώσουν στην νέα μέθοδο την απαραίτητη «δύναμη μοντελοποίησης». Με τον όρο δύναμη μοντελοποίησης αναφερόμαστε στην ικανότητα μοντελοποίησης μεγάλου εύρους συστημάτων, χωρίς την απαίτηση το εκάστοτε σύστημα να πληροί ορισμένες προδιαγραφές και να έχει συγκεκριμένα χαρακτηριστικά. Ο ορισμός της νέας μεθόδου όπως παρουσιάστηκε στο συνέδριο IFAC '05 είναι:

Ορισμός Γενικευμένων Αυτομάτων (ΓΑ):

Τα γενικευμένα αυτόματα ορίζονται ως $A = \langle X, Z, Q, \Sigma, Init, Flow, L, S, W, E, R_X, R_Z \rangle$ όπου:

- Οι μεταβλητές του συστήματος:
 - Πραγματικές μεταβλητές: $X = \{x_1, x_2, x_3, \dots, x_m\}$
 - Διακριτές μεταβλητές: $Z = \{z_1, z_2, z_3, \dots, z_k\}$

- Σύνολο καταστάσεων (states): $Q = \{q_1, q_2, q_3, \dots, q_n\}$
- Σύνολο γεγονότων (events): $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_\lambda\}$, που μπορεί να είναι:
 - ο είτε διακριτές μεταβλητές,
 - ο είτε συνθήκες επί των συνεχών μεταβλητών,
 - ο είτε λογικός συνδυασμός των παραπάνω.

Σχήμα 5.1. Γενικευμένο αυτόματο.

- Αρχικές συνθήκες (Init):
 - ο $X = X_0$
 - ο $Z = Z_0$
 - ο q_0
- Συναρτήσεις (flow conditions):
 - ο $F(X, \dot{X}) = 0$
 - ο $Z_{i+1} = G(Z_i)$
- Όρια μεταβλητών (invariant conditions): $L = \{l_1, l_2, l_3, \dots, l_n\}$
- Περιορισμοί ή ασφαλείς τιμές παραμέτρων: $S = \{s_1, s_2, s_3, \dots, s_n\}$
- Γεγονότα που αγνοούνται μέχρι την ικανοποίηση των περιορισμών στην δεδομένη κατάσταση: $W = \{w_1, w_2, w_3, \dots, w_n\}$ με $w_i \subseteq \Sigma$.
- Σύνολο μεταβάσεων (set of transitions): $E \subseteq Q \times Q \times \Sigma \times R_X \times R_Z$.
- Σύνολα «αρχικοποίησης» μεταβλητών κατά τη διάρκεια μιας μετάβασης:
 - ο Αρχικοποίηση πραγματικών μεταβλητών: $X = R_X$
 - ο Αρχικοποίηση διακριτών μεταβλητών: $Z = R_Z$
- Κάθε μετάβαση $(q, q', \sigma, r_X, r_Z)$ αναπαριστά τη μετάβαση από την κατάσταση q στην κατάσταση q' η οποία προκαλείται από το γεγονός $\sigma \in \Sigma$. Το σύνολο

r_X δίνει τις πραγματικές μεταβλητές που θα αρχικοποιηθούν κατά τη μετάβαση ενώ το σύνολο r_Z τις διακριτές.

- Κάθε γεγονός θεωρείται στιγμιαίο και συνεπώς δεν μπορεί να διακοπεί. Εκτελέσιμα είναι μόνο τα ενεργά γεγονότα της παρούσας κατάστασης. Δύο γεγονότα μπορούν να συμβούν ταυτόχρονα. Σε αυτή την περίπτωση εκτελείτε το γεγονός με υψηλότερη προτεραιότητα σύμφωνα με το σύνολο $\Sigma_i \subseteq \Sigma$.
- Σε κάθε κατάσταση q_i αντιστοιχεί ένα σύνολο παραμέτρων:
 - Συναρτήσεις (flow conditions):
 - $F_i(X, \dot{X}) = 0$
 - $Z_{j+1} = G_i(Z_j)$
 - Ενεργά γεγονότα στην παρούσα κατάσταση: $\Sigma_i \subseteq \Sigma$. Το σύνολο Σ_i έχει εξ ορισμού ζ στοιχεία που το καθένα εξ' αυτών ανήκει στο σύνολο Σ . $\Sigma_i = \{\sigma_j^{i,k}\}$ όπου: i η κατάσταση, $k = 1, 2, \dots, \zeta$ και $j \in [1, \lambda]$. Ο δείκτης k δείχνει επίσης και την προτεραιότητα εκτέλεσης μεταβάσεων που προκαλούνται από διαφορετικά γεγονότα. Αν δύο γεγονότα συμβούν ταυτόχρονα και πυροδοτούν διαφορετικές μεταβάσεις, η μετάβαση που θα πραγματοποιηθεί θα είναι αυτή που προκαλείται από το γεγονός με μικρότερο δείκτη k .
 - Όρια μεταβλητών (invariant conditions): l_i
 - Περιορισμοί ή ασφαλείς τιμές παραμέτρων: S_i
 - Γεγονότα που αγνοούνται μέχρι την ικανοποίηση των περιορισμών στην δεδομένη κατάσταση: $w_i \subseteq \Sigma$.

Ο παραπάνω ορισμός δίνει όλα τα απαραίτητα χαρακτηριστικά για την δημιουργία ενός γενικευμένου αυτομάτου. Θα ήταν όμως σκόπιμο να δοθούν ορισμένες επεξηγήσεις για την αποφυγή δυσερμηνιών του. Λόγω του γεγονότος ότι βασίζονται στην θεωρία αυτομάτων, τα γενικευμένα αυτόματα χαρακτηρίζονται από τον γράφο ελέγχου με έναν αριθμό καταστάσεων και πλήθος μεταβάσεων μεταξύ αυτών. Ο πρωταρχικός ορισμός της θεωρίας αυτομάτων ορίζει ως κατάσταση ενός αυτομάτου το σύνολο των διακριτών και συνεχών μεταβλητών του, το οποίο είναι όλη η απαραίτητη γνώση για τον υπολογισμό της συμπεριφοράς του συστήματος σε μια δεδομένη ακολουθία εισόδου. Αντιθέτως στα γενικευμένα, ο όρος κατάσταση παίρνει μια πιο ευρεία έννοια, οριζόμενη περισσότερο ως κατάσταση λειτουργίας ή τοποθεσία, αφήνοντας στην άκρη τον αρχικό ορισμό. Συνεπώς στην συνέχεια του

κειμένου η γενικευμένη αυτή μορφή της κατάστασης θα χρησιμοποιείται υπό τον τίτλο «κατάσταση».

Επιπροσθέτως, θα πρέπει να επισημανθούν και ορισμένες πτυχές των δράσεων ελέγχου. Πρωτίστως, η δυνατότητα συνεχούς ελέγχου μπορεί να περιληφθεί στις συναρτήσεις επί των συνεχών μεταβλητών του συστήματος. Για παράδειγμα, ο πλέον συνήθης τύπος συνεχούς ελεγκτή σε χρήση στις σύγχρονες βιομηχανίες, ο PID ελεγκτής, μπορεί να πάρει την συνήθη μορφή του ως μια συνάρτηση.

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

5.2 Ιδιότητες Γενικευμένων Αυτομάτων.

Στην παράγραφο αυτή εξετάζονται οι βασικές θεωρητικές ιδιότητες των γενικευμένων αυτομάτων. Κυριότερη ιδιότητα αποτελεί το γεγονός ότι τα γενικευμένα αυτόματα είναι εξ ορισμού ντετερμινιστικά, λόγω της ιεραρχίας εκτέλεσης γεγονότων. Ο ντετερμινιστικός χαρακτήρας διαφυλάχθηκε κατά τον ορισμό, καθώς ο βιομηχανικός προσανατολισμός της μεθόδου δεν μπορούσε να επιτρέψει ασάφειες στην εκτέλεση του αυτομάτου. Οι υπόλοιπες ιδιότητες παρατίθενται στη συνέχεια με τους αντίστοιχους ορισμούς και παραδείγματα.

Ορισμός 1. Ένα γενικευμένο αυτόματο ${}^G\mathcal{A}$ είναι γραμμικό αν:

1. Όλες οι συνεχείς μεταβλητές του ${}^G\mathcal{A}$ είναι γραμμικές,
2. για κάθε κατάσταση $q_i \in Q$, οι συναρτήσεις $Flow(\sigma_i)$ είναι όλες κυρτές συναρτήσεις,
3. για κάθε μετάβαση $T_i \in E$, οι συνθήκες μετάβασης $T_i(\sigma_i)$ είναι όλες κυρτές συναρτήσεις.

Σχήμα 5.2. Παράδειγμα γραμμικού αυτομάτου.

Γραμμικότητα στην θεωρία συστημάτων. Σύμφωνα με τον τυπικό ορισμό των γενικευμένων αυτομάτων, οι συναρτήσεις κάθε κατάστασης θεωρούνται ως διαφορικές – αλγεβρικές εξισώσεις. Ειδικότερα, οι συναρτήσεις μπορούν να αναπαρασταθούν από πρώτης τάξεως διαφορικές εξισώσεις n διάστασης

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

Όπου x είναι το διάνυσμα κατάστασης υπό την έννοια του χώρου κατάστασης – δεν πρέπει να συγχέεται με τις καταστάσεις-τοποθεσίες ενός ${}^G\mathcal{A}$ – παίρνοντας τιμές από τον Ευκλείδειο χώρο και $f(t, x, u)$ είναι ένας διανυσματικός χώρος ελέγχου (control vector field). Εφόσον στην θεωρία ελέγχου, τα γραμμικά δυναμικά συστήματα μοντελοποιούνται ως $\dot{x}(t) = A(t)x(t) + B(t)u(t)$, οι γραμμικές συναρτήσεις ενός αυτομάτου μπορούν να μοντελοποιηθούν ως

$$f(t, x, u) = A(t)x(t) + B(t)u(t)$$

Όπου το διάνυσμα κατάστασης $x \in X \subseteq \mathbb{R}^m$, και το διάνυσμα εισόδων ελέγχου $u \in U \subseteq \mathbb{R}^p$, $A(t)$ και $B(t)$ είναι μήτρες στο $\mathbb{R}^m \times \mathbb{R}^m$ και $\mathbb{R}^p \times \mathbb{R}^p$ αντιστοίχως. Η διάσταση m αποτελεί την διάσταση του αυτομάτου και p είναι το πλήθος των εισόδων ελέγχου.

Ορισμός 2. Ένα γενικευμένο αυτόματο ${}^G\mathcal{A}$ είναι χρονικά αμετάβλητο αν

1. για όλες τις καταστάσεις-τοποθεσίες $q_i \in Q$, οι συναρτήσεις $Flow(q_i)$ είναι χρονικά αμετάβλητες, και
2. για όλες τις μεταβάσεις $T_i \in E$, οι συνθήκες μετάβασης $T_i(\sigma_i)$ είναι χρονικά αμετάβλητες.

Σχήμα 5.3. Παράδειγμα χρονικά αμετάβλητου αυτομάτου.

Στο παράδειγμα του σχήματος 5.3, η εξίσωση $Y(t) = 4k + c$ δηλώνει ότι η κατάσταση q_1 έχει χρονικά αμετάβλητες συναρτήσεις, όπου k, c μπορεί να είναι σταθερές ή χρονικά αμετάβλητες αριθμητικές τιμές μεταβλητών, π.χ. μια μεταβλητή που υποδηλώνει την στάθμη μιας δεξαμενής. Ομοίως, όλες οι μεταβάσεις του αυτομάτου πυροδοτούνται από χρονικά αμετάβλητα γεγονότα, όπως $Y=12$.

Ορισμός 3. Ένα σύστημα μεταβάσεων $T = (Q, \Sigma, \rightarrow, Q_S, Q_F, R)$ αποτελείται από τα σύνολα Q και R , όπου $R = R_X \cup R_Z$, το αλφάβητο Σ , την αρχική κατάσταση $\text{Init}(Q)$ που δηλώνεται ως Q_S , το σύνολο $Q_F \subseteq Q$ των τελικών καταστάσεων και μια σχέση μετάβασης $\rightarrow$ μεταξύ τους, όπου $\rightarrow \subseteq Q \times Q \times \Sigma \times R_X \times R_Z$.

Παρόλο που περιέχεται στην τυπική σύνταξη ενός GA , είναι συνήθης η δήλωση μιας μετάβασης $(q, q', \sigma, r_x, r_z) \in \rightarrow$ ως $q \xrightarrow[r_x, r_z]{\sigma} q'$, όπου $q, q' \in Q$. Ένα σύστημα μεταβάσεων είναι πεπερασμένο, αν η cardinality των Q , $|Q|$ είναι πεπερασμένη, διαφορετικά είναι μη πεπερασμένο.

Σχήμα 5.4. Παράδειγμα απεικόνισης συστήματος μεταβάσεων.

Ορισμός 4. Ένα σύστημα μεταβάσεων $T = (Q, \Sigma, \rightarrow, Q_S, Q_F, R)$ είναι ελεύθερο κλειδώματος αν για κάθε κατάσταση $q \in Q$, υπάρχει μια κατάσταση $q' \in Q$ και ένα γεγονός $\sigma \in \Sigma$ τέτοιο ώστε $q \xrightarrow[r_x, r_z]{\sigma} q'$.

Ορισμός 5. Μια τροχιά ενός συστήματος μεταβάσεων είναι μια πεπερασμένη ή μη ακολουθία καταστάσεων $\{q_i\}_{i=0}^N$ τέτοια ώστε,

1. $q_0 \in Q_S$ και

$$2. \quad q_i \xrightarrow[r_x, r_z]{\sigma} q_{i+1} \text{ για κάθε } i = 0, 1, \dots, N - 1.$$

Στο αυτόματο του παραδείγματος του σχήματος 4.12, μια πιθανή τροχιά του συστήματος θα ήταν η ακόλουθη:

$$Motor\ off \xrightarrow{Start} Confirm\ Start \xrightarrow{Start} Motor\ On \xrightarrow{Stop} Motor\ Off.$$

Ορισμός 6. Με δεδομένο ένα σύστημα μεταβάσεων $T = (Q, \Sigma, \rightarrow, Q_S, Q_F, R)$ μια κατάσταση $q' \in Q_F$ είναι προσεγγίσιμη από μια κατάσταση $q \in Q_S$ και δηλώνετε ως $q \xrightarrow[\{r\}]{\{\sigma\}} q'$, αν υπάρχει ακολουθία μεταβάσεων για την οποία η σχέση $q \xrightarrow[\{r\}]{\{\sigma\}} q'$ είναι αληθής.

Το πρόβλημα της προσεγγισιμότητας ουσιαστικά ελέγχει αν,

$$Post(Q_S) \cap Q_F \neq \emptyset \text{ ή } Q_S \cap Pre(Q_F) \neq \emptyset$$

Όπου για μια περιοχή P , η οποία είναι ένα υποσύνολο καταστάσεων $P \subseteq Q$,

$$Post(P) = \left\{ q \in Q \mid \exists \sigma \in \Sigma \quad \exists p \in P \quad p \xrightarrow[r_x, r_z]{\sigma} q \right\} \text{ και}$$

$$Pre(P) = \left\{ q \in Q \mid \exists \sigma \in \Sigma \quad \exists p \in P \quad q \xrightarrow[r_x, r_z]{\sigma} p \right\}$$

Γενικά, μια δομή της temporal logic μπορεί να χρησιμοποιηθεί για την διατύπωση ενός γενικευμένου προβλήματος προσεγγισιμότητας. Με δεδομένο ένα σύστημα μεταβάσεων T και μια δομή της temporal logic φ , το κύριο πρόβλημα επαλήθευσης είναι $T \models \varphi$, όπου το φ καθορίζεται με χρήση κατάλληλης temporal logic.

Το έργο επαλήθευσης ορίζεται επίσης ως μια διαδικασία ανεύρεσης για το αν ικανοποιείται η επιθυμητή ιδιότητα, η οποία περιγράφεται από την εξίσωση φ . Οι πλέον συνήθεις ιδιότητες, Ασφάλειας-Safety και Βιωσιμότητας-Liveness, αφορούν την συμπεριφορά προσεγγισιμότητας του αυτομάτου. Για παράδειγμα, η ιδιότητα

“Η κατάσταση q πάντα παραμένει σε ένα σύνολο καταστάσεων αποκαλούμενο περιοχή $P \subseteq Q$ ”

είναι μια τέτοια ιδιότητα, όπως και η διπλή ιδιότητα

“Η κατάσταση q τελικά προσεγγίζει ένα σύνολο καταστάσεων αποκαλούμενο περιοχή $P \subseteq Q$ ”

Η πρώτη ιδιότητα είναι γνωστή ως ιδιότητα ασφαλείας, καθώς η περιοχή P αποτελείται από τις «επιθυμητές» ή «ασφαλείς» καταστάσεις. Για παράδειγμα στο σύστημα της πρέσας του σχήματος 5.5 με το αντίστοιχο αυτόματο του σχήματος 5.6, όπου απαιτείται η διασφάλιση ότι η αύξηση της θερμοκρασίας, εξαιτίας της πίεσης στην μπιγιέτα αλουμινίου, δεν θα υπερβεί την ήδη προθερμασμένη θερμοκρασία t_h , η περιοχή P είναι το σύνολο των καταστάσεων για τις οποίες $t_c \leq t_h$. Επομένως, είναι επιθυμητή η διασφάλιση ότι η πυρακτωμένη μπιγιέτα αλουμινίου δεν θα ξεπεράσει το σημείο τήξης της (δηλαδή $t_c \leq t_h$).

Σχήμα 5.5. Σύστημα της πρέσας αλουμινίου.

Η ιδιότητα ασφαλείας με χρήση σημειογραφίας από την Temporal Logic μπορεί να γραφεί ως $\Box(q \in P)$, καθώς το σύμβολο $\Box$ σημαίνει «πάντα». Η δεύτερη ιδιότητα, «η κατάσταση q προσεγγίζει την κατάσταση p » είναι γνωστή ως ιδιότητα της βιωσιμότητας. Ουσιαστικά εκφράζει το γεγονός ότι κάτι «καλό» θα συμβεί στο σύστημα. Για παράδειγμα, στο σύστημα της πρέσας αλουμινίου, η πυρακτωμένη μπιγιέτα όχι μόνο δεν πρέπει να ξεπεράσει το σημείο τήξης της αλλά πρέπει να μετατραπεί σε ένα αποδεκτό προφίλ αλουμινίου. Με χρήση σημειογραφίας από την Temporal Logic, η ιδιότητα αυτή μπορεί να γραφεί ως $\Diamond(q \in P)$, όπου το σύμβολο $\Diamond$ σημαίνει «τελικά». Η Temporal Logic επιτρέπει και συνδυασμούς ιδιοτήτων με στόχο την δημιουργία συνθετότερων, όπως $\Box \Diamond(q \in P)$ που συμβολίζει το «πάντα, τελικά η κατάσταση q βρίσκεται στο P », ή με άλλα λόγια, η κατάσταση βρίσκεται στο σύνολο P «απείρως συχνά». Ομοίως το $\Diamond \Box(q \in P)$ συμβολίζει το «τελικά, πάντα η

κατάσταση q βρίσκεται στο P ή με άλλα λόγια, το t_c κάποια στιγμή ικανοποιεί την συνθήκη $t_c \leq t_h$ και έπειτα μένει εκεί για πάντα.

Σχήμα 5.6. Έλεγχος θερμοκρασίας στο σύστημα της πρέσας αλουμινίου.

Σύμφωνα με τον ορισμό των γενικευμένων αυτομάτων, μια κατάσταση $q \in Q$ του $\mathcal{G}\mathcal{A}$ περιέχει την συνεχή χρονική εξέλιξη τόσο των πραγματικών μεταβλητών $x \in X$, όσο και των διακριτών $z \in Z$, όπου η μεταβολή των πρώτων οφείλεται στις εισόδους ελέγχου $u \in U$. Αναφερόμαστε στο ζεύγος $(q, u) \in Q \times U$ ως τη δυναμική κατάσταση του αυτομάτου $\mathcal{G}\mathcal{A}$. Το σύνολο U των εισόδων ελέγχου είναι υποσύνολο των μεταβλητών του συστήματος.

Ορισμός 7. Προσεγγισιμότητα στην έννοια της λειτουργίας ελέγχου. Το πρόβλημα της προσεγγισιμότητας για ένα γενικευμένο αυτόματο διερευνά, με δοθέν ένα αυτόματο $\mathcal{G}\mathcal{A}$ και ένα διάνυσμα ελέγχου $u \in U$, εάν υπάρχει μια αρχική τροχιά $\{q_i\}_{i=0}^N$ ενός συστήματος μεταβάσεων T που να συνάδει με την δυναμική κατάσταση (q_N, u) .

Ένα τρέξιμο ή εκτέλεση του αυτομάτου είναι μια τροχιά όπου όλες οι ενδιάμεσες καταστάσεις είναι προσεγγίσιμες. Αναφορικά με το παράδειγμα της πρέσας

αλουμινίου, επιθυμητή εκτέλεση του αυτομάτου είναι κάθε τροχιά που δεν περιέχει τις καταστάσεις συναγερμού (alarm states).

Εξομοίωση και αμφιεξομοίωση. Η σύνθεση δύο αυτομάτων βοηθά στην μοντελοποίηση της συμπεριφοράς ενός ολοκληρωμένου συστήματος – που είναι ένα υβριδικό σύστημα – με χρήση εποπτικού ελεγκτή και στόχο την αποφυγή δυσάρεστων για το σύστημα καταστάσεων. Η ισοδυναμία συμπεριφοράς μεταξύ των αρχικών αυτομάτων και του συντεθειμένου μπορεί να βασίζεται στις σχέσεις εξομοίωσης και αμφιεξομοίωσης.

Ορισμός 8. Με δεδομένα δύο συστήματα μεταβάσεων $T = (Q, \Sigma, \rightarrow, Q_S, Q_F, R)$ και $T' = (Q', \Sigma', \rightarrow', Q'_S, Q'_F, R')$ μια δυαδική σχέση $\tau \subseteq Q \times Q'$ αποκαλείται σχέση εξομοίωσης, αν

1. $(q_1 \tau q'_1) \wedge (q_1 \in Q_S) \Rightarrow (q'_1 \in Q'_S)$
2. $(q_1 \tau q'_1) \wedge (q_1 \in Q_F) \Rightarrow (q'_1 \in Q'_F)$
3. $(q_1 \tau q'_1) \wedge ((q_1, \sigma, q_2) \in \rightarrow) \Rightarrow \exists q'_2 \text{ τέτοιο ώστε}$
 $(q_2 \tau q'_2) \wedge ((q'_1, \sigma, q'_2) \in \rightarrow')$

Γράφουμε $(q \tau q')$ για να δηλώσουμε ότι υπάρχει μια σχέση εξομοίωσης τ , και διαβάζετε ως το q εξομοιώνεται από το q' . Εάν υπάρχει η σχέση εξομοίωσης υπάρχει λέγεται αντίστοιχα ότι το T εξομοιώνεται από το T' . Το ερώτημα που προκύπτει είναι κατά πόσο μπορεί να χρησιμοποιηθεί η ίδια δυαδική σχέση $\tau \subseteq Q \times Q'$ για ναδειχθεί ότι και το T' εξομοιώνεται από το T . Εάν ισχύουν και οι δύο ισχυρισμοί, το T εξομοιώνεται από το T' και το T' εξομοιώνεται από το T , για την ίδια σχέση εξομοίωσης, τότε τα συστήματα T και T' αμφιεξομοιώνονται. Η σχέση αμφιεξομοίωσης είναι μια συμμετρική σχέση εξομοίωσης που δίνεται στον παρακάτω ορισμό.

Ορισμός 9. Με δεδομένα δύο συστήματα μεταβάσεων $T = (Q, \Sigma, \rightarrow, Q_S, Q_F, R)$ και $T' = (Q', \Sigma', \rightarrow', Q'_S, Q'_F, R')$ μια δυαδική σχέση $\cong \subseteq Q \times Q'$ αποκαλείται σχέση αμφιεξομοίωσης, αν

1. $(q_1 \cong q'_1) \wedge (q_1 \in Q_S) \Rightarrow (q'_1 \in Q'_S)$
2. $(q_1 \cong q'_1) \wedge (q_1 \in Q_F) \Rightarrow (q'_1 \in Q'_F)$

3. $(q_1 \cong q'_1) \wedge ((q_1, \sigma, q_2) \in \rightarrow) \Rightarrow \exists q'_2$ τέτοιο ώστε
 $(q_2 \cong q'_2) \wedge ((q'_1, \sigma, q'_2) \in \rightarrow)$
4. $(q_2 \cong q'_2) \wedge ((q'_1, \sigma, q'_2) \in \rightarrow) \Rightarrow \exists q_1$ τέτοιο ώστε
 $(q_1 \cong q'_1) \wedge ((q_1, \sigma, q_2) \in \rightarrow)$

Εάν τα συστήματα μεταβάσεων T και T' αμφιεξομοιώνονται, τότε

$$T \models \varphi \Leftrightarrow T' \models \varphi$$

Η τελευταία σχέση παρουσιάζει το βασικό πλεονέκτημα της ιδιότητα της αμφιεξομοίωσης, καθώς έλεγχος ισχύος της σχέσης $T' \models \varphi$ ισοδυναμεί με έλεγχο της σχέσης $T \models \varphi$, και συνεπώς η πολυπλοκότητα του προβλήματος μειώνεται αισθητά. Η έκφραση των ιδιοτήτων της εξομοίωσης και αμφιεξομοίωσης στα γενικευμένα αυτόματα $\mathcal{GA}$ και όχι στα συστήματα μεταβάσεων είναι μια εύκολη επέκταση των παραπάνω ορισμών. Η σχέση $\cong$ μεταξύ δύο αυτομάτων $\mathcal{GA}_1$ και $\mathcal{GA}_2$ είναι σχέση αμφιεξομοίωσης, αν και μόνο αν, είναι σχέση εξομοίωσης του $\mathcal{GA}_1$ στο $\mathcal{GA}_2$, και την ίδια στιγμή είναι και σχέση εξομοίωσης του $\mathcal{GA}_2$ στο $\mathcal{GA}_1$. Αυτό σημαίνει όχι μόνο ότι κάθε εκτέλεση του $\mathcal{GA}_1$ μπορεί να ισοδυναμεί με αντίστοιχη εκτέλεση του $\mathcal{GA}_2$ και αντιστρόφως, αλλά επίσης ότι αυτή η ισοδυναμία μπορεί να επιτευχθεί από μια και μόνο σχέση. Συμπερασματικά, μια σχέση αμφιεξομοίωσης είναι ευρύτερα από ένα ζεύγος σχέσεων εξομοίωσης.

5.3 Επαλήθευση μέσω σύγκρισης με υπάρχουσες μεθόδους

Η εισαγωγή μιας οποιασδήποτε νέας μεθόδου για την επίλυση ενός προβλήματος, πρέπει να συνοδεύεται από σύγκριση με τις υπάρχουσες μεθόδους ώστε να αποδείξει κατά πόσο υπερτερεί ή μη σε σχέση με αυτές. Με βάση αυτή τη θεώρηση, παρουσιάζεται σε αυτή την παράγραφο μια σύγκριση της προτεινόμενης νέας μεθόδου με τις πλέον χρησιμοποιούμενες μορφές αυτομάτων, δηλαδή τα χρονισμένα και τα υβριδικά αυτόματα. Οι ορισμοί των δύο αυτών μεθόδων μοντελοποίησης έχουν δοθεί σε προηγούμενο κεφάλαιο, οπότε εδώ θα δοθεί περισσότερο βάρος στην σύγκριση πλεονεκτημάτων και μειονεκτημάτων της κάθε μεθόδου. Αναφορικά με

άλλες μεθόδους μοντελοποίησης, γίνεται σύγκριση με τα εργαλεία CHARON, Ptolemy και Shift.

5.3.1 Σύγκριση με Χρονισμένα Αυτόματα

Μια πρώτη ματιά στη σύγκριση της νέας μεθόδου με τα χρονισμένα αυτόματα, οδηγεί στο συμπέρασμα ότι ένα χρονισμένο αυτόματο μπορεί απροβλημάτιστα να μεταφραστεί σε αυτόματο της νέας μεθόδου. Το συμπέρασμα αυτό πηγάζει από το γεγονός ότι οι περισσότερες παράμετροι είναι κοινές και στους δύο τύπους αυτομάτων, ενώ οι υπόλοιπες μπορούν να αντικατασταθούν από άλλες παρόμοιες.

Αν αναλογιστούμε τον ορισμό των χρονισμένων αυτομάτων, θα πρέπει να αντικαταστήσουμε την πεντάδα παραμέτρων $A = \langle Q, \Sigma, i, C, E \rangle$ με ισοδύναμες παραμέτρους της νέας μεθόδου. Οι δύο πρώτες, το σύνολο καταστάσεων και το σύνολο γεγονότων, είναι κοινές και στις δύο μεθόδους. Η τρίτη παράμετρος, η αρχική κατάσταση, περιλαμβάνεται στο σύνολο των αρχικών συνθηκών, όπως ορίζεται από το σύνολο *Init*, το οποίο περιλαμβάνει επιπροσθέτως αρχικές συνθήκες για το σύνολο των μεταβλητών του συστήματος. Η προτελευταία παράμετρος, το σύνολο των ρολογιών, μπορεί να αντικατασταθεί, στη νέα μέθοδο, με πραγματικές μεταβλητές που υπόκεινται σε διαφορικές εξισώσεις της μορφής $\dot{X} = 1$. Τέλος, το σύνολο των μεταβάσεων με τους χρονικούς περιορισμούς και τα ρολόγια που μηδενίζονται, μπορεί να αντικατασταθεί από κατάλληλο συνδυασμό του νέου συνόλου μεταβάσεων, του πίνακα αρχικοποίησης και των γεγονότων που αγνοούνται σε κάθε μετάβαση.

Μια ολοκληρωμένη σύγκριση μεταξύ των δύο μεθόδων θα πρέπει να περιλαμβάνει απαρίθμηση των πλεονεκτημάτων χρήσης της νέας μεθόδου. Καταρχήν, η νέα μέθοδος μπορεί να διαχειριστεί, επιπροσθέτως, και διακριτές μεταβλητές με διαφορετικές συναρτήσεις σε κάθε κατάσταση. Επιπλέον, όπως τονίστηκε και παραπάνω, οι αρχικές συνθήκες περιλαμβάνουν το σύνολο των μεταβλητών του συστήματος και όχι μόνο την αρχική κατάσταση. Εξάλλου, οι περιορισμοί στη νέα μέθοδο μπορούν να εφαρμοστούν σε κάθε μεταβλητή και όχι μόνο στις μεταβλητές «ρολογιών». Αυτό σε συνδυασμό με τα γεγονότα που αγνοούνται σε κάθε κατάσταση μέχρι την εκπλήρωση κάποιων συνθηκών, δίνουν την δυνατότητα στον μηχανικό να χειριστεί ακόμα και τα πιο πολύπλοκα συστήματα πραγματικού χρόνου.

5.3.2 Σύγκριση με Υβριδικά Αυτόματα

Τα υβριδικά αυτόματα έχουν εισαχθεί ως το κύριο εργαλείο μοντελοποίησης υβριδικών συστημάτων. Αποτελούν υπερσύνολο των πεπερασμένων αυτομάτων, καθώς έχουν επιπροσθέτως συνεχείς μεταβλητές, που υπόκεινται σε διαφορετικές συνθήκες μεταβολής σε κάθε κατάσταση του αυτομάτου. Ο ορισμός των υβριδικών αυτομάτων δόθηκε σε προηγούμενο κεφάλαιο, όπου παρουσιάστηκε και το τυπικό παράδειγμα ενός υβριδικού αυτομάτου που μοντελοποιεί την λειτουργία ενός θερμοστάτη. Αν ανατρέξουμε στο παράδειγμα αυτό θα ανακαλύψουμε ότι θα μπορούσε κάλλιστα να αποτελεί υλοποίηση με βάση την καινούργια μέθοδο μοντελοποίησης. Αυτό προκύπτει από το γεγονός ότι η νέα μέθοδος περιέχει όλες τις παραμέτρους των υβριδικών αυτομάτων, αποτελώντας ένα υπερσύνολο αυτών.

Για την επεξήγηση του παραπάνω συμπεράσματος ας παραθέσουμε μια σύγκριση μεταξύ των δύο τύπων αυτομάτων. Οι περισσότερες παράμετροι των υβριδικών αυτομάτων περιέχονται αυτούσιες στη νέα μορφή, όπως οι μεταβλητές, τα γεγονότα, οι αρχικές συνθήκες, τα όρια των μεταβλητών και οι συναρτήσεις των μεταβλητών. Ο γράφος ελέγχου, έχει αντικατασταθεί από το σύνολο των καταστάσεων και τις μεταβάσεις μεταξύ αυτών. Τέλος οι συνθήκες μεταπήδησης (jump conditions) περιέχονται στο σύνολο των γεγονότων, όπως έχει αυτό οριστεί στην παράγραφο 5.1.

Τελειώνοντας την σύγκριση των δύο μεθόδων, ας παραθέσουμε τις επιπρόσθετες παραμέτρους που περιέχουν τα αυτόματα της νέας προτεινόμενης μεθόδου. Καταρχήν υπάρχει η δυνατότητα χειρισμού διακριτών μεταβλητών, με καθορισμένες αρχικές συνθήκες και διαφορετικές συναρτήσεις σε κάθε κατάσταση. Επιπλέον, τα νέα αυτόματα έχουν την δυνατότητα καθυστέρησης μεταβάσεων μέχρι την εκπλήρωση συγκεκριμένων κριτηρίων. Τα κριτήρια αυτά μπορούν να δρουν ανασταλτικά σε ορισμένα μόνο γεγονότα, παρέχοντας μια ευελιξία στην μοντελοποίηση σύνθετων συστημάτων. Τέλος, η νέα μέθοδος επιτρέπει την αρχικοποίηση όλων των μεταβλητών του συστήματος σε κάθε μετάβαση, δυνατότητα που δεν παρέχεται από τα υβριδικά αυτόματα.

Συνοψίζοντας την σύγκριση με τα υβριδικά αυτόματα, το κύριο συμπέρασμα είναι ότι αποτελούν υποσύνολο των γενικευμένων αυτομάτων. Τα γενικευμένα αυτόματα με τα περισσότερα στοιχεία παρέχουν περισσότερες δυνατότητες στην μοντελοποίηση βιομηχανικών συστημάτων, καθώς κάθε επιπλέον παράμετρος δίνει στον μηχανικό ελέγχου την δυνατότητα να χειριστεί ακόμα περισσότερες κατηγορίες συστημάτων.

5.3.3 Σύγκριση με CHARON

Η CHARON [80] είναι μια αρθρωτή γλώσσα προδιαγραφών για αλληλεπιδρώντα υβριδικά συστήματα βασισμένα στις έννοιες του πράκτορα-agent και της κατάστασης-mode. Οι πράκτορες περιγράφουν την αρχιτεκτονική του συστήματος, ενώ οι καταστάσεις την συμπεριφορά του. Το ιδιαίτερο χαρακτηριστικό της CHARON είναι ότι υποστηρίζει τόσο αρχιτεκτονική περιγραφή, όσο και περιγραφή συμπεριφοράς και παρέχει συνεργασία και ιεραρχία σε μια αρθρωτή δομή. Επίσης διαχωρίζει τις μεταβλητές σε σύνολα αναλογικών-συνεχών και διακριτών. Η δυναμική συμπεριφορά του συστήματος αναπαρίσταται με χρήση διαφορικών εξισώσεων και οριακές συνθήκες περιορίζοντας τα πεδία ορισμού των μεταβλητών, η οποία μπορεί να αλλάζει στα διάφορα επίπεδα της ιεραρχίας. Επιγραμματικά, τα κύρια στοιχεία της CHARON είναι η αρχιτεκτονική και σχεσιακή ιεραρχία, καθώς και η συνεχής ενημέρωση και συγχρονισμός των διακριτών και συνεχών μεταβλητών.

Η γλώσσα CHARON αποτελεί ένα ισχυρότατο εργαλείο με πολύ σύνθετη σημειολογία, με την οποία ο βιομηχανικός κόσμος δεν παρουσιάζεται ιδιαίτερα οικείος. Βέβαια, η CHARON δεν παρουσιάστηκε με βιομηχανικό προσανατολισμό. Εκτενέστερη σύγκριση της σημειολογίας των γενικευμένων αυτομάτων με την CHARON δεν έχει πρακτικό ενδιαφέρον, καθώς οι δύο αυτές μέθοδοι έχουν διαφορετικό προσανατολισμό ή ακριβέστερα, διαφορετικό πεδίο εφαρμογής.

5.3.4 Σύγκριση με Ptolemy

Το Ptolemy (ή καλύτερα το Ptolemy II) [81] αποτελεί ένα λογισμικό πακέτο βασισμένο σε Java, το οποίο αναπτύχθηκε στα πλαίσια του ευρύτερου προγράμματος Ptolemy. Το πρόγραμμα Ptolemy ασχολείται με την μοντελοποίηση, εξομοίωση και σχεδιασμό ενσωματωμένων συστημάτων πραγματικού χρόνου που δρουν εν παραλλήλω, και το Ptolemy II περιέχει μια σουίτα προγραμμάτων-πεδίων εφαρμογής,

για την υλοποίηση τέτοιων συστημάτων. Η βιβλιοθήκη συστατικών ή υποσυστημάτων περιέχει πληθώρα έτοιμων παραδειγμάτων με πολυμορφικό χαρακτήρα, δηλαδή δυνατότητα χρήσης σε περισσότερα του ενός πεδία εφαρμογής. Από τα δημιουργηθέντα πεδία ξεχωρίζουν οι μηχανές πεπερασμένων καταστάσεων, τα ασύρματα συστήματα, μοντελοποίησης συνεχούς χρόνου κτλ., ενώ άλλα πεδία παραμένουν σε πειραματικό στάδιο. Συνοψίζοντας, το Ptolemy είναι περισσότερο μια γραφική γλώσσα προγραμματισμού και όχι μια φορμαλιστική μέθοδος. Βεβαίως παρουσιάζει ένα ευρύ πεδίο εφαρμογών αλλά οι απαιτήσεις του βιομηχανικού κόσμου μάλλον είναι ιδιαίτερες υψηλές και δύσκολες, ώστε το αντίστοιχο πεδίο να μην έχει αναπτυχθεί μέχρι σήμερα.

5.3.5 Σύγκριση με Shift

Το τελευταίο λογισμικό εργαλείο που μπορεί να συγκριθεί με τα γενικευμένα αυτόματα είναι το Shift, το οποίο παρουσιάστηκε και σε προηγούμενη παράγραφο, και αποτελεί μια παλαιότερη προσπάθεια για εξομοίωση υβριδικών συστημάτων. Το Shift [82, 83] είναι ένα λογισμικό εργαλείο για την περιγραφή δυναμικών δικτύων υβριδικών αυτομάτων. Το μοντέλο Shift αναπτύχθηκε με την πεποίθηση ότι θα αποτελέσει το κατάλληλο υπόβαθρο για την περιγραφή σύνθετων εφαρμογών, όπως αυτοματοποιημένα συστήματα συγκοινωνιών, συστήματα ελέγχου εναέριας κυκλοφορίας, σύνθετα ρομποτικά συστήματα, συστήματα πλοήγησης υποβρυχίων κ.α., δηλαδή συστήματα των οποίων η λειτουργία δεν μπορεί να εκφραστεί ικανοποιητικά με τις υπάρχουσες μεθόδους. Το Smart-AHS [84] είναι το μοναδικό ολοκληρωμένο έργο που έχει δημιουργηθεί με βάση το Shift, ως πλατφόρμα μοντελοποίησης και ελέγχου αυτοματοποιημένων συστημάτων αυτοκινητοδρόμων (Automated Highway Systems). Το Shift ουσιαστικά παραμένει ανενεργό από το 1997 που ολοκληρώθηκε το Smart-AHS. Το Shift, όπως αναφέρθηκε και παραπάνω, ήταν μια προσπάθεια υλοποίησης ενός εργαλείου για υβριδικά συστήματα, χωρίς όμως κανένα περιορισμό όσον αφορά το πεδίο εφαρμογής του και αυτό ήταν ουσιαστικά τροχοπέδη για το όλο έργο. Τα γενικευμένα αυτόματα έχουν καθαρά βιομηχανικό προσανατολισμό για να αποφύγουν το εμπόδιο της γενίκευσης.

Ολοκληρώνοντας την σύγκριση μεταξύ των γενικευμένων αυτομάτων και εναλλακτικών μεθόδων ας συνοψίσουμε τα συμπεράσματα. Εφόσον τα γενικευμένα αυτόματα αποτελούν υπερσύνολο των υβριδικών αυτομάτων, υπερκαλύπτουν και τα

πεπερασμένα, χρονισμένα και PLC αυτόματα. Συνεπώς, κάθε σύστημα που μοντελοποιείται σε κάποια από αυτές τις μορφές, μπορεί εύκολα να μετατραπεί σε ένα γενικευμένο αυτόματο. Επιπλέον, τα γενικευμένα αυτόματα είναι εφοδιασμένα με πρόσθετες παραμέτρους μοντελοποίησης οι οποίες μπορεί να φανούν ιδιαίτερα χρήσιμες στην αντιμετώπιση σύνθετων συστημάτων. Εν κατακλείδι, τα γενικευμένα αυτόματα παρουσιάζονται λιγότερο εξαρτώμενα από την εκάστοτε εφαρμογή σε σύγκριση με κάθε άλλη μορφή αυτομάτων. Οι υπόλοιπες τρεις μέθοδοι μοντελοποίησης CHARON, Ptolemy και Shift, δεν έχουν βιομηχανικό προσανατολισμό και συνεπώς δεν μπορούν να συγκριθούν ενδελεχώς με τα γενικευμένα αυτόματα.

Κεφάλαιο 6^ο

Εργαλεία μοντελοποίησης στα γενικευμένα αυτόματα

Οι επόμενες δύο παράγραφοι παρουσιάζουν αναλυτικά δύο χρήσιμα εργαλεία των γενικευμένων αυτομάτων για την μοντελοποίηση βιομηχανικών συστημάτων. Το πρώτο εργαλείο είναι η συγχώνευση καταστάσεων και χρησιμοποιείται για την αποφυγή της υπέρμετρης αύξησης του χώρου κατάστασης. Η συγχώνευση καταστάσεων πραγματοποιείται από την ένωση δύο ή περισσότερων καταστάσεων με στόχο την δημιουργία μίας υπέρ-κατάστασης. Το δεύτερο εργαλείο είναι η σύνθεση αυτομάτων που χρησιμοποιείται για την σύμπτυξη δύο ή περισσότερων αυτομάτων σε ένα. Πρέπει βεβαίως να υπογραμμιστεί ότι τα δύο αυτά εργαλεία σε καμία περίπτωση δεν αλλάζουν την συμπεριφορά του όλου συστήματος, καθώς το τελικό μοντέλο έχει την ίδια ακριβώς συμπεριφορά με τα μοντέλα από τα οποία προήλθε. Τέλος, στα εργαλεία μοντελοποίησης περιλαμβάνεται και η δυνατότητα ανάπτυξης μοντέλων με περισσότερα του ενός αυτόματα και ιεραρχική δόμηση αυτών.

6.1 Συγχώνευση καταστάσεων

Τα γενικευμένα αυτόματα αποτελούν μια νέα φορμαλιστική μέθοδο μοντελοποίησης με βιομηχανικό προσανατολισμό. Τα σύγχρονα βιομηχανικά συστήματα περιέχουν δεκάδες χιλιάδες καταστάσεις συνθέτοντας ένα τεράστιο συνολικό χώρο κατάστασης. Ένας υπερμεγέθης χώρος κατάστασης δεν προσφέρει τίποτα παραπάνω από το να μπερδεύει τον μηχανικό ελέγχου. Με βάση αυτό το υπαρκτό πρόβλημα, τα ΓΑ προσφέρουν την δυνατότητα της συγχώνευσης καταστάσεων, μια πολύτιμη ιδιότητα για την μείωση του αριθμού των καταστάσεων. Η συγχώνευση καταστάσεων ορίζεται ως η σύμπτυξη δύο ή περισσότερων καταστάσεων με στόχο την δημιουργία μίας υπέρ-κατάστασης. Η προκύπτουσα υπέρ-κατάσταση παρουσιάζει την ίδια συμπεριφορά με τις αρχικές καταστάσεις χωρίς την απώλεια κανενός είδους δεδομένων ή πληροφορίας, καθώς η συμπεριφορά του συστήματος και οι τιμές όλων των μεταβλητών παραμένουν αμετάβλητες. Η μόνη απώλεια, αν μπορεί να θεωρηθεί

ως απώλεια, είναι η οπτική πληροφορία που μπορεί να προσφέρει μία κατάσταση (π.χ. διαφορετικές καταστάσεις για τις καταστάσεις λειτουργίας μιας μηχανής). Αλλά ακόμα και αυτή η απώλεια μπορεί να ξεπερασθεί οποιαδήποτε χρονική στιγμή απλά με την αποσύνθεση της υπέρ-κατάστασης στις αρχικές καταστάσεις. Οι μεταβάσεις μεταξύ των συμπτυγμένων καταστάσεων μετατρέπονται σε μεταβάσεις βρόχου στην νέα υπέρ-κατάσταση, δηλαδή μεταβάσεις που η αρχική και τελική κατάσταση ταυτίζονται.

Η στρατηγική που χρησιμοποιείται για την συγχώνευση καταστάσεων διαφέρει από σύστημα σε σύστημα, αλλά υπάρχουν ορισμένοι γενικοί κανόνες που πρέπει να ακολουθήσει ένας μηχανικός ελέγχου για να επιτύχει την ορθή συγχώνευση καταστάσεων.

- Η συγχώνευση καταστάσεων οδηγεί σε μια νέα υπέρ-κατάσταση η οποία ορίζεται όμοια με τις υπό-καταστάσεις από τις οποίες προέρχεται.
- Η συγχώνευση δύο καταστάσεων σε μία προσθέτει μια επιπρόσθετη δυαδική μεταβλητή στο σύνολο των διακριτών μεταβλητών Z του συστήματος. Στην περίπτωση της συγχώνευσης περισσότερων καταστάσεων, η νέα μεταβλητή είναι ένας ακέραιος αριθμός που προστίθεται στο σύνολο X . Οι μεταβλητές αυτές σηματοδοτούν σε ποια από τις συγχωνευμένες υπό-καταστάσεις βρίσκεται το σύστημα.
- Η νέα υπέρ-κατάσταση έχει τα ίδια σύνολα μεταβλητών με την προσθήκη της διακριτής ή ακεραίας που επισημάνθηκε παραπάνω, ενώ όλα τα άλλα σύνολα παραμέτρων τροποποιούνται κατάλληλα ώστε να υπάρχει ισοδυναμία.
- Οι μεταβάσεις μεταξύ των αρχικών καταστάσεων συχνά μετατρέπονται σε μεταβάσεις βρόχου στην νέα κατάσταση.

Ο μηχανικός ελέγχου που επιχειρεί να συγχωνεύσει καταστάσεις ενός μεγάλου συστήματος πρέπει να είναι πολύ προσεκτικός κατά την ανάλυση όλων των μεταβάσεων μεταξύ των αρχικών καταστάσεων καθώς και με τις παραμέτρους της καθεμίας. Ειδική προσοχή απαιτείται για τα γεγονότα που αγνοούνται σε κάθε κατάσταση και για τα όρια της κάθε μεταβλητής. Εν κατακλείδι, η συγχώνευση καταστάσεων είναι μια διαισθητική διαδικασία όπου ο χρήστης παίζει τον καθοριστικό ρόλο και λόγω τούτου πρέπει να είναι ιδιαίτερα προσεκτικός. Ένα χαρακτηριστικό παράδειγμα συγχώνευσης καταστάσεων παρουσιάζεται στην παράγραφο 6.1.1 όπου

πραγματοποιείται συγχώνευση καταστάσεων σε διαδοχικά στάδια με στόχο τον όσο το δυνατόν μικρότερο χώρο κατάστασης. Σε ορισμένες περιπτώσεις, ένας έμπειρος μηχανικός μπορεί να οδηγηθεί σε αυτόματο με μόνο μία κατάσταση, ιδιαίτερα χρήσιμο σε περίπτωση σύνθεσης.

Ως ένα μικρό παράδειγμα συγχώνευσης καταστάσεων ας υποθέσουμε μια μηχανή που ξεκινά και σταματά χειροκίνητα από ένα αντίστοιχο ζεύγος μπουτόν. Ένα μοντέλο του συστήματος θα είχε δύο καταστάσεις που η καθεμιά θα αντιπροσώπευε την αντίστοιχη κατάσταση λειτουργίας, σχήμα 6.1. Η συγχώνευση αυτών των δύο καταστάσεων είναι ιδιαίτερα απλή και οδηγεί σε ένα αυτόματο με μόνο μία κατάσταση που απεικονίζεται στο σχήμα 6.2.

Σχήμα 6.1. Αυτόματο με δύο καταστάσεις.

Σχήμα 6.2. Αυτόματο μίας κατάστασης.

6.1.1 Σύστημα τριών μηχανών με παύση αυτών υπό ειδικές συνθήκες.

Ας υποθέσουμε ότι έχουμε μια σειρά τριών όμοιων μηχανών για τις οποίες θέλουμε να εκκινούν και να σταματούν χειροκίνητα με ισάριθμα ζεύγη μπουτόν start – stop. Για την εκκίνηση των μηχανών δεν υφίσταται κάποια ειδική απαίτηση, για την παύση όμως, λειτουργικοί λόγοι της διαδικασίας επιβάλλουν, να επενεργεί σε όλες τις μηχανές αμέσως πλην της τελευταίας εν λειτουργία μηχανής. Η μηχανή αυτή θα

πρέπει να σταματά μόνο αν το επιτρέψει ένας αισθητήρας ή κάποιος χρονοδιακόπτης π.χ. θερμοκρασία ενός αντιδραστηρίου ή παρέλευση κάποιου χρονικού διαστήματος. Όλες οι υπόλοιπες μηχανές σταματούν ανεξάρτητα από την τιμή του αισθητήρα με το πάτημα του αντίστοιχου μπουτόν σβέσης. Η δυσκολία του όλου προβλήματος έγκειται στο γεγονός ότι η τελευταία εν λειτουργία μηχανή δεν είναι προκαθορισμένη και συνεπώς μπορεί να είναι οποιαδήποτε από τις τρεις, πράγμα που αποτελεί εντελώς τυχαίο γεγονός. Προφανώς αν έχει ενεργοποιηθεί ο αισθητήρας αλλά δεν έχει πατηθεί το stop μπουτόν της τελευταίας μηχανής, αυτή συνεχίζει να λειτουργεί κανονικά. Συνοψίζοντας, οποιαδήποτε από τις τρεις μηχανές μπορεί να μείνει τελευταία σε λειτουργία οπότε θέλουμε να σταματά σε συνδυασμό με έναν αισθητήρα. Οι υπόλοιπες θα έχουν σταματήσει ήδη με κανονικό τρόπο.

Σχήμα 6.3. Διάγραμμα καταστάσεων του συστήματος των τριών μηχανών.

Στο σχήμα 6.3 παρουσιάζεται το διάγραμμα καταστάσεων με όλους τους λειτουργικούς συνδυασμούς των τριών μηχανών. Συνολικά υπάρχουν 9 καταστάσεις, από τις οποίες η 0 είναι η αρχική κατάσταση, στις καταστάσεις 1, 4 και 7 μόνο μία μηχανή είναι σε λειτουργία, στις 2, 5 και 8 λειτουργούν 2 μηχανές, ενώ στην κατάσταση 3 και οι τρεις μηχανές είναι σε λειτουργία. Τέλος, η κατάσταση 6 είναι η μεταβατική κατάσταση κατά την οποία έχει ζητηθεί η παύση της τελευταίας εν λειτουργία μηχανής και αναμένεται η παρέλευση του χρόνου των 30 δευτερολέπτων ώστε να σταματήσει.

Το διάγραμμα καταστάσεων του σχήματος μοιάζει κατά πολύ με τα κλασσικά διαγράμματα πεπερασμένων αυτομάτων, καθώς έχει τον ίδιο αριθμό καταστάσεων και μεταβάσεων με αυτά. Αν όμως δοκιμάσουμε να εκμεταλλευτούμε όλες τις δυνατότητες που μας παρέχει η νέα μέθοδος μπορούμε να επιτύχουμε συγχώνευση καταστάσεων και ελάττωση του συνολικού διαγράμματος. Συγκεκριμένα μια πρώτη μετατροπή θα μπορούσε να γίνει με συγχώνευση των καταστάσεων 1, 4 και 7 σε μια νέα κατάσταση όπου μόνο μία μηχανή είναι σε λειτουργία και των 2, 5 και 8 σε μια νέα όπου δύο μηχανές είναι σε λειτουργία. Το νέο διάγραμμα καταστάσεων, που πλέον έχει πέντε καταστάσεις παρουσιάζεται στο σχήμα 6.4.

Σχήμα 6.4. Μειωμένο διάγραμμα καταστάσεων με πέντε καταστάσεις.

Η σημειολογία του διαγράμματος πρέπει να αποσαφηνισθεί καθώς δεν είναι ευδιάκριτη στο σχήμα. Καταρχήν οι καταστάσεις 1 και 2 έχουν έξι ζεύγη από περιορισμούς και γεγονότα που αγνοούνται μέχρι την ικανοποίηση αυτών. Συγκεκριμένα τα πρώτα τρία ζεύγη είναι $[M_i=1]$, $\{s_i\}$, όπου $i=1,2,3$ και τα τρία επόμενα είναι $[M_i=0]$, $\{p_i\}$, όπου $i=1,2,3$. Η κατάσταση 3 όπου όλες οι μηχανές είναι σε λειτουργία, δεν έχει κανένα περιορισμό. Όλες οι μεταβάσεις με τίτλο Hs_i προκαλούνται από τα γεγονότα s_i και θέτουν τις διακριτές μεταβλητές M_i στην τιμή 1 αντίστοιχα. Ομοίως, οι μεταβάσεις Hp_i προκαλούνται από τα γεγονότα p_i και θέτουν τις μεταβλητές M_i στην τιμή 0. Τέλος, οι μεταβάσεις Hr_i που προκαλούνται από τα γεγονότα s_i , θέτουν τις μεταβλητές M_i στην τιμή 1, ενώ τις μεταβλητές M_j (όπου $j \neq i$) στην τιμή 0.

Μια άλλη προσέγγιση, με την προσθήκη μιας βοηθητικής μεταβλητής k , που αντιστοιχεί στο πλήθος των μηχανών που βρίσκονται σε λειτουργία, απεικονίζεται στο σχήμα 6.5. Το νέο διάγραμμα καταστάσεων έχει μόνο τρεις καταστάσεις, καθώς οι καταστάσεις 1, 2 και 4 συγχωνεύθηκαν σε μία νέα όπου τουλάχιστον μία μηχανή είναι σε λειτουργία.

Σχήμα 6.5. Μειωμένο διάγραμμα καταστάσεων με τρεις καταστάσεις.

Πλέον η κατάσταση 1 έχει επτά ζεύγη περιορισμών και γεγονότων που αγνοούνται. Τα πρώτα έξι είναι ακριβώς τα ίδια όπως και προηγουμένως, ενώ το έβδομο είναι $[k=1]$, $\{p_1, p_2, p_3\}$. Ομοίως με την προηγούμενη προσέγγιση, όλες οι μεταβάσεις Hs_i προκαλούνται από τα γεγονότα s_i και προκαλούν τη θέση των διακριτών μεταβλητών M_i στην τιμή 1 και την αύξηση του μετρητή k κατά μία μονάδα. Αντίστοιχα, οι μεταβάσεις Hr_i προκαλούνται από τα γεγονότα s_i , θέτουν τις μεταβλητές M_i στην τιμή 1, μηδενίζουν τις M_j (όπου $j \neq 1$) και αρχικοποιούν τον μετρητή k στην τιμή 1. Τέλος, το γεγονός α ορίζεται ως $\alpha = M_1 \cdot \overline{M_2} \cdot \overline{M_3} \cdot p_1 + \overline{M_1} \cdot M_2 \cdot \overline{M_3} \cdot p_2 + \overline{M_1} \cdot \overline{M_2} \cdot M_3 \cdot p_3$. Μία εκτέλεση του αυτομάτου στην ακολουθία εισόδου $u = s_2, s_1, p_1, \alpha, s_3, \alpha, T = 30sec$ ακολουθεί παρακάτω, όπου η σημειογραφία:

$$\begin{array}{c|c|c} q_0 & & q_1 \\ \hline \begin{array}{l} X = X_0 \\ Z = Z_0 \end{array} & \begin{array}{l} \dot{X} = F_0(X) \\ Z_{i+1} = G_0(Z_i) \end{array} & \xrightarrow{\sigma} \begin{array}{l} X = R_X \\ Z = R_Z \end{array} \\ & & \begin{array}{l} \dot{X} = F_1(X) \\ Z_{i+1} = G_1(Z_i) \end{array} \end{array}$$

αναπαριστά την μετάβαση από την κατάσταση q_0 στην κατάσταση q_1 που προκαλείται από το γεγονός σ . Οι δύο εξισώσεις κάτω από κάθε κατάσταση αναπαριστούν τον πίνακα αρχικοποίησης για την μετάβαση σ αρχικοποιώντας τις μεταβλητές του συστήματος. Οι άλλες δύο διαφορικές εξισώσεις εκφράζουν τις συναρτήσεις σύμφωνα με τις οποίες αλλάζουν τιμή οι μεταβλητές στην τρέχουσα κατάσταση.

Αν κάποιος προχωρούσε στην συγχώνευση αυτομάτων με στόχο την περαιτέρω μείωση του χώρου κατάστασης, θα μπορούσε να προσθέσει μια ακόμα δυαδική μεταβλητή. Το προκύπτον σύστημα έχει μόνο μία κατάσταση, καθώς οι τρεις συγχωνεύονται σε μία η οποία έχει 11 μεταβάσεις βρόχου. Η προσέγγιση αυτή παρουσιάζεται στο σχήμα 6.6.

Η κατάσταση q_0 έχει επτά ζεύγη περιορισμών και γεγονότων που αγνοούνται ακριβώς όπως και στην αρχική προσέγγιση, $[M_i=1]$, $\{s_i\}$, και $[M_i=0]$, $\{p_i\}$, με $i=1,2,3$, και $[k=1]$, $\{p_1, p_2, p_3\}$. Παρομοίως με το αρχικό μοντέλο, όλες οι μεταβάσεις hs_i προκαλούνται από τα γεγονότα s_i και προκαλούν την έναυση των μηχανών M_i , αυξάνουν τον περιεχόμενο του απαριθμητή k κατά μία μονάδα και μηδενίζουν την

μεταβλητή c . Επίσης, οι μεταβάσεις Hr_i προκαλούνται από τα γεγονότα p_i και σταματούν τις μηχανές M_i ελαττώνοντας τον μετρητή k κατά μία μονάδα. Οι μεταβάσεις Hr_i , που προκαλούνται από τα γεγονότα $s_i \cdot c$, θέτουν τις δυαδικές μεταβλητές M_i στην τιμή 1, ενώ τις μεταβλητές M_j (με $j \neq i$) στην τιμή 0, αρχικοποιούν τον μετρητή k στην τιμή 1, μηδενίζουν το χρονόμετρο και την μεταβλητή c . Τέλος, το γεγονός a ορίζεται ως $a = M_1 \cdot \overline{M_2} \cdot \overline{M_3} \cdot p_1 + \overline{M_1} \cdot M_2 \cdot \overline{M_3} \cdot p_2 + \overline{M_1} \cdot \overline{M_2} \cdot M_3 \cdot p_3$ και θέτει την μεταβλητή c στην τιμή 1. Η τελευταία μετάβαση, όπως απεικονίζεται στο σχήμα 6.6, προκαλείται από το γεγονός $T = 30sec$ και μηδενίζει όλες τις μεταβλητές.

Σχήμα 6.6. Αυτόματο μιας μόνο κατάστασης

Συνοψίζοντας την διαδικασία συγχώνευσης καταστάσεων, συμπεραίνουμε ότι ενώ το αρχικό σύστημα απαιτούσε 9 καταστάσεις, με την συγχώνευση οδηγηθήκαμε σε ένα αυτόματο με μία μόνο κατάσταση. Γενικά ένα σύστημα με n μηχανές θα απαιτούσε $2^n + 1$ καταστάσεις. Όμως με χρήση των γενικευμένων αυτομάτων, ένα σύστημα n -μηχανών μπορεί να μοντελοποιηθεί με τρεις ή ακόμα με μόλις μία κατάσταση. Η μόνη προϋπόθεση είναι η κατάλληλη τροποποίηση των δύο τελευταίων μοντέλων όπως δόθηκαν παραπάνω. Όλες οι αλλαγές που έγιναν αφορούν κυρίως τον αριθμό των μεταβάσεων μεταξύ των καταστάσεων ενώ διατηρούν τον ίδιο τύπο. Μια ακόμα αλλαγή αφορά το γεγονός a , το οποίο είναι:

$$a = M_1 \cdot \overline{M_2} \cdot \dots \cdot \overline{M_n} \cdot p_1 + \overline{M_1} \cdot M_2 \cdot \dots \cdot \overline{M_n} \cdot p_2 + \dots + \overline{M_1} \cdot \overline{M_2} \cdot \dots \cdot M_n \cdot p_n$$

Τα μοντέλα των n -μηχανών απεικονίζονται στα σχήματα 6.7 και 6.8.

Σχήμα 6.7. Διάγραμμα τριών καταστάσεων για n μηχανές.

Σχήμα 6.8. Διάγραμμα μιας κατάστασης για n μηχανές.

6.2 Σύνθεση αυτομάτων

Όλες οι ακαδημαϊκές μέθοδοι για την μοντελοποίηση ΣΔΓ έχουν μια βασική αδυναμία όταν δοκιμάζονται για βιομηχανική χρήση καθώς είναι μη πρακτικές για την μοντελοποίηση μεγάλων συστημάτων. Τα σύγχρονα βιομηχανικά συστήματα έχουν αυξημένη πολυπλοκότητα και η δημιουργία ενός μοντέλου είναι ιδιαίτερα επίπονη. Η εργασία των μηχανικών της βιομηχανίας δυσκολεύει ιδιαίτερα όταν το συνολικό σύστημα απαρτίζεται από πολλά υποσυστήματα με διαφορετικά χαρακτηριστικά και ιδιότητες. Σε αυτές τις περιπτώσεις που η απευθείας μοντελοποίηση δεν είναι εύκολη, η σύνθεση αυτομάτων μπορεί να αποτελέσει ένα πολύ ισχυρό εργαλείο στα χέρια του μηχανικού. Η σύνθεση επιτρέπει την δημιουργία μεγάλων και πολύπλοκων μοντέλων χρησιμοποιώντας τα απλά μοντέλα των διαφόρων υποσυστημάτων. Τα μοντέλα των υποσυστημάτων, λόγω του μεγέθους τους και συνήθως της απλής δομής τους, απαιτούν λιγότερο χρόνο για την υλοποίηση και με σύνθεση αυτών προκύπτει το συνολικό μοντέλο του συστήματος. Το τελικό μοντέλο έχει πολλαπλάσιες καταστάσεις συγκρινόμενο με τα αρχικά αυτόματα και το σύνολο των μεταβλητών όλων των υποσυστημάτων.

Η σύνθεση είναι εφαρμόσιμη μόνο όταν όλα τα υποσυστήματα έχουν διαφορετικά σύνολα μεταβλητών και γεγονότων, καθώς σε περίπτωση κοινών παραμέτρων μπορεί να δημιουργηθεί σύγχυση με αποτέλεσμα λανθασμένο μοντέλο συστήματος. Ο ορθός διαχωρισμός του όλου συστήματος σε υποσυστήματα υπαγορεύει διαφορετικά σύνολα μεταβλητών και παραμέτρων σε καθένα από αυτά.

Η σύνθεση A δύο γενικευμένων αυτομάτων A_1 και A_2 συμβολίζεται με $A = A_1 \parallel A_2$. Ο ορισμός της σύνθεσης γενικευμένων αυτομάτων δίνεται παρακάτω.

Ορισμός Σύνθεσης Αυτομάτων: Ας υποθέσουμε δύο αυτόματα $A_1 = \langle X_1, Z_1, Q_1, \Sigma_1, Init_1, Flow_1, L_1, S_1, W_1, E_1, R_{X_1}, R_{Z_1} \rangle$ και $A_2 = \langle X_2, Z_2, Q_2, \Sigma_2, Init_2, Flow_2, L_2, S_2, W_2, E_2, R_{X_2}, R_{Z_2} \rangle$. Η σύνθεση των δύο αυτομάτων είναι το αυτόματο $A = \langle X, Z, Q, \Sigma, Init, Flow, L, S, W, E, R_X, R_Z \rangle$ όπου:

- $X = X_1 \cup X_2$
- $Z = Z_1 \cup Z_2$
- $Q = Q_1 \times Q_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2$

- $Init = \left\{ \begin{array}{l} [(q_{01}, q_{02}), [(X_{01}, X_{02}), [(Z_{01}, Z_{02})]: \\ (q_{01}, X_{01}, Z_{01}) \in Init_1 \wedge (q_{02}, X_{02}, Z_{02}) \in Init_2 \end{array} \right\}$
- $Flow = (F, G)$, όπου:
 - $F(Q, X) = \begin{bmatrix} F_1(Q_1, X_1) \\ F_2(Q_2, X_2) \end{bmatrix}$
 - $G(Q, Z) = \begin{bmatrix} G_1(Q_1, Z_1) \\ G_2(Q_2, Z_2) \end{bmatrix}$
- $L = \{[(q_1, q_2), (X_1, X_2)]: (q_1, X_1) \in L_1 \wedge (q_2, X_2) \in L_2\}$
- $S = \{[(q_1, q_2), (X_1, X_2)]: (q_1, X_1) \in S_1 \wedge (q_2, X_2) \in S_2\}$
- $W = \{[(q_1, q_2), (X_1, X_2)]: (q_1, X_1) \in W_1 \wedge (q_2, X_2) \in W_2\}$
- $E \subseteq Q \times Q \times \Sigma \times R_X \times R_Z$
- $R_X = R_{X_1} \cup R_{X_2}$
- $R_Z = R_{Z_1} \cup R_{Z_2}$

Όπως αναφέρθηκε και παραπάνω η μόνη συνθήκη για την σύνθεση δύο αυτομάτων είναι η ανεξαρτησία τους.

Λήμμα. Δύο γενικευμένα αυτόματα $A_1 = \langle X_1, Z_1, Q_1, \Sigma_1, Init_1, Flow_1, L_1, S_1, W_1, E_1, R_{X_1}, R_{Z_1} \rangle$ και $A_2 = \langle X_2, Z_2, Q_2, \Sigma_2, Init_2, Flow_2, L_2, S_2, W_2, E_2, R_{X_2}, R_{Z_2} \rangle$, ονομάζονται ανεξάρτητα αν και μόνο αν, τα ζεύγη (X_1, X_2) , (Z_1, Z_2) και (Σ_1, Σ_2) είναι ανεξάρτητα δηλαδή δεν έχουν κοινά στοιχεία, $X_1 \cap X_2 = \emptyset$, $Z_1 \cap Z_2 = \emptyset$ και $\Sigma_1 \cap \Sigma_2 = \emptyset$.

6.2.1 Σύστημα τριών δεξαμενών.

Ας υποθέσουμε ένα παράδειγμα από μια χημική βιομηχανία που αποτελείται από τρεις διαφορετικές δεξαμενές, όπως απεικονίζεται στο σχήμα 6.9. Οι δύο πρώτες δεξαμενές περιέχουν διαφορετικά υλικά και τροφοδοτούν την τρίτη δεξαμενή. Η όλη διεργασία ξεκινά με ένα μπουτόν έναρξης. Αρχικά ανοίγει η βαλβίδα g και υγρό από την δεξαμενή ρέει προς την τρίτη. Όταν το συνολικό υγρό στην δεξαμενή 3 είναι 10 lt. Η βαλβίδα g κλείνει ενώ ταυτόχρονα ανοίγει η βαλβίδα h . Παράλληλα ξεκινά η διαδικασία ανάμιξης με την έναρξη του αναμίκτη p . Υγρό από την δεξαμενή 2 ρέει μέχρι ο συνολικός όγκος της δεξαμενής 3 ανέλθει στα 15 lt. και τότε η βαλβίδα h κλείνει και ξεκινά η φάση θέρμανσης ενεργοποιώντας την αντίσταση k , η οποία σταματά όταν η θερμοκρασία του μίγματος φτάσει τους 600° C. Τότε ο αναμίκτης και

η αντίσταση σβήνουν και ανοίγει η βαλβίδα j έως ότου αδειάσει η δεξαμενή 3. Η βαλβίδα e ελέγχεται αυτόματα ούτως ώστε να διατηρεί την στάθμη στην δεξαμενή 1 μεταξύ των ορίων a και b . Όταν η στάθμη πέφτει στο b , η βαλβίδα ανοίγει μέχρι η στάθμη να φτάσει το όριο a . Αντίστοιχα, η βαλβίδα f ελέγχεται ώστε να διατηρεί την στάθμη της δεξαμενής 2 μεταξύ των ορίων c και d .

Σχήμα 6.9. Σύστημα τριών δεξαμενών.

Το σύστημα που μόλις περιγράφηκε μπορεί να θεωρηθεί ως ένα ενιαίο σύστημα ή σαν σύνθεση τριών υποσυστημάτων. Κάθε υποσύστημα μοντελοποιεί καθεμία από τις τρεις δεξαμενές με ένα γενικευμένο αυτόματο. Τα τρία αυτόματα που δημιουργήθηκαν απεικονίζονται στα σχήματα 6.10, 6.11 και 6.12.

Σχήμα 6.10. Το αυτόματο της πρώτης δεξαμενής.

Σχήμα 6.11. Το αυτόματο της δεύτερης δεξαμενής.

Τα δύο πρώτα (αυτόματο 1 και αυτόματο 2) είναι ιδιαιτέρως απλά και ελέγχουν την εισροή των δεξαμενών συναρτήσει των σταθμών σε αυτές. Το τρίτο αυτόματο (αυτόματο 3) είναι ελαφρώς πιο σύνθετο, καθώς έχει ένα περιορισμό στην κατάσταση q_1 , όπου το γεγονός s_1 θα καθυστερήσει μέχρι την πλήρωση της δεξαμενής 1 ($a=1$). Επιπροσθέτως, η κατάσταση q_4 έχει μια συνάρτηση για την αύξηση της θερμοκρασίας μέχρι το άνω όριο των 600°C .

Σχήμα 6.12. Το αυτόματο της τρίτης δεξαμενής.

Τα τρία αυτόματα πρέπει να συντεθούν ώστε να προκύψει το μοντέλο του όλου συστήματος. Η σύνθεση των αυτομάτων όπως αυτή παρουσιάστηκε στην παράγραφο 6.1.2 πρέπει να πραγματοποιηθεί σε δύο στάδια. Αρχικά θα γίνει η σύνθεση των δύο πρώτων αυτομάτων και στην συνέχεια θα γίνει η σύνθεση του παραγώγου με το τρίτο αυτόματο. Συνεπώς τα αυτόματα των δεξαμενών 1 και 2 θα συντεθούν πρώτα

και θα παραχθεί ένα νέο αυτόματο (*αυτόματο 12*), το οποίο εξ ορισμού θα έχει 4 καταστάσεις και 8 μεταβάσεις μεταξύ των, όπως εμφανίζεται στο σχήμα 6.13. Η σύνθεση του *αυτόματου 3* με το *αυτόματο 12* δίνει ένα νέο αυτόματο (*αυτόματο 123*) το οποίο αναπαριστά το συνολικό σύστημα και αποτελείται από 20 καταστάσεις και 60 μεταβάσεις, οι οποίες προκαλούνται από 9 διαφορετικά γεγονότα. Το αυτόματο απεικονίζεται στο σχήμα 6.14 και οι παράμετροι στις καταστάσεις, οι οποίες σημειώνονται ως Q_{ij} με $i = 1,2,3,4$ και $j = 1,2,3,4,5$, είναι οι ακόλουθες:

Σχήμα 6.13. Το *αυτόματο 12*.

- Q_{1j} : $e = 0, f = 0$
- Q_{2j} : $e = 1, f = 0$
- Q_{3j} : $e = 1, f = 1$
- Q_{4j} : $e = 0, f = 1$
- Q_{i1} : $g = 0, h = 0, j = 0, k = 0, p = 0 / [s_1], \{\alpha\}$
- Q_{i2} : $g = 1, h = 0, j = 0, k = 0, p = 0$
- Q_{i3} : $g = 0, h = 1, j = 0, k = 0, p = 1$
- Q_{i4} : $g = 0, h = 0, j = 0, k = 1, p = 1, \dot{\theta} = a - b \cdot \theta$
- Q_{i5} : $g = 0, h = 0, j = 1, k = 0, p = 0$

Τα εννέα γεγονότα που προκαλούν τις μεταβάσεις μεταξύ των καταστάσεων είναι:

- a : $Q_{2j} \rightarrow Q_{1j}, Q_{3j} \rightarrow Q_{4j}$

- $b: Q_{1j} \rightarrow Q_{2j}, Q_{4j} \rightarrow Q_{3j}$
- $c: Q_{2j} \rightarrow Q_{3j}, Q_{4j} \rightarrow Q_{1j}$
- $d: Q_{1j} \rightarrow Q_{4j}, Q_{3j} \rightarrow Q_{2j}$
- $s_1: Q_{i1} \rightarrow Q_{i2}$
- $V_1: Q_{i2} \rightarrow Q_{i3}$
- $V_2: Q_{i3} \rightarrow Q_{i4}$
- $V_3: Q_{i5} \rightarrow Q_{i1}$
- $\theta = 600^\circ: Q_{i4} \rightarrow Q_{i5}$

Σχήμα 6.14. Το *αυτόματο123*.

Σε αυτό το σημείο πρέπει να σημειωθεί ότι το *αυτόματο123* θα μπορούσε να σχεδιασθεί εξ αρχής χωρίς την σύνθεση των τριών αυτομάτων. Βέβαια κάτι τέτοιο είναι σαφώς πιο δύσκολο συγκρινόμενο με την σύνθεση που περιγράφηκε παραπάνω. Γίνεται λοιπόν σαφές ότι ακόμα και σε ένα τόσο απλό σύστημα η σύνθεση είναι ένα πολύ ισχυρό εργαλείο που διευκολύνει την δουλειά του μηχανικού. Για ένα σύγχρονο βιομηχανικό σύστημα με μεγάλες διαστάσεις, η σύνθεση αποτελεί ίσως την μόνη ρεαλιστική λύση για μοντελοποίηση.

Στην παράγραφο 6.1.1 όπου περιγράφηκε η συγχώνευση καταστάσεων, είχε υπογραμμιστεί ιδιαίτερα η σημασία της συγχώνευσης κυρίως σε περιπτώσεις που οδηγούμαστε σε αυτόματο μιας μόνο κατάστασης. Το παράδειγμα των τριών δεξαμενών είναι ενδεικτικό για την παρουσίαση της χρησιμότητας της συγχώνευσης, καθώς τα δύο πρώτα αυτόματα μπορούν εύκολα να μετατραπούν σε αυτόματα μιας κατάστασης. Συνεπώς, το τελικό μοντέλο έχει μόνο πέντε καταστάσεις! Αυτό προκύπτει από το γεγονός ότι το μοντέλο της σύνθεσης έχει $m \times n$ καταστάσεις, όπου m και n οι καταστάσεις των δύο αρχικών αυτομάτων. Στο δοθέν παράδειγμα, τα τρία αυτόματα παράγουν ένα μοντέλο με $2 \times 2 \times 5 = 20$ καταστάσεις. Αν όμως γίνει χρήση της συγχώνευσης καταστάσεων, το τελικό μοντέλο θα έχει μόνο $1 \times 1 \times 5 = 5$ καταστάσεις. Τα αυτόματα μιας κατάστασης που μοντελοποιούν τις δεξαμενές 1 και 2 παρουσιάζονται στο σχήμα 6.15, ενώ το 6.16 παρουσιάζει το αποτέλεσμα της σύνθεσης αυτών. Το αυτόματο που προκύπτει μετά το δεύτερο στάδιο σύνθεσης απεικονίζεται στο σχήμα 6.17.

Σχήμα 6.15. Αυτόματα μιας κατάστασης για τις δυο πρώτες δεξαμενές.

Σχήμα 6.16. Το νέο αυτόματο¹².

Σχήμα 6.17. Το τελικό αυτόματο πέντε καταστάσεων

6.3 Ιεραρχία αυτομάτων

Οι απαιτήσεις των σύγχρονων βιομηχανικών συστημάτων ολοένα και περισσότερο υπαγορεύουν την χρήση μοντέλων με ιεραρχική δομή. Τα γενικευμένα αυτόματα προσφέρουν την δυνατότητα δημιουργίας τέτοιων μοντέλων με χρήση περισσότερων του ενός αυτομάτων. Αναλυτικότερα, κάθε αυτόματο μοντελοποιεί συγκεκριμένο μέρος του συνολικού συστήματος. Τέλος, ένα αυτόματο ανωτέρου ιεραρχικά επιπέδου αναλαμβάνει το ρόλο του επόπτη. Το αυτόματο-επόπτης τροφοδοτεί τα επιμέρους αυτόματα-ελεγκτές με όλα τα απαραίτητα δεδομένα για την επίτευξη της επιθυμητής παραγωγής.

Το ιεραρχικό μοντέλο που δύναται να κατασκευαστεί δεν περιορίζεται μόνο σε δύο επίπεδα ιεραρχίας, καθώς υπάρχει η δυνατότητα ιεράρχησης και σε περισσότερα επίπεδα. Σε αυτή την περίπτωση το κάθε αυτόματο δέχεται παραμέτρους από το αυτόματο του ανωτέρου επιπέδου ενώ τροφοδοτεί κατάλληλα τα αυτόματα του κατώτερου επιπέδου. Εκτεταμένο παράδειγμα ιεραρχικής δόμησης αυτομάτων για την μοντελοποίηση και τον έλεγχο ενός συστήματος παρουσιάζεται στην παράγραφο 8.2, όπου το υπό έλεγχο σύστημα είναι μια μονάδα παραγωγής έτοιμου σκυροδέματος.

Κεφάλαιο 7^ο

Εργαλεία υλοποίησης γενικευμένων αυτομάτων σε βιομηχανικά συστήματα

Έως αυτό το σημείο, έχουν περιγραφεί μόνο τα βήματα μοντελοποίησης ενός βιομηχανικού συστήματος αλλά το κύριο έργο ενός μηχανικού ελέγχου είναι ο επιτυχής έλεγχος ενός τέτοιου συστήματος. Στο κεφάλαιο αυτό παρουσιάζονται δύο εργαλεία υλοποίησης μοντέλων-ελεγκτών σε πραγματικά συστήματα του βιομηχανικού χώρου. Το πρώτο εργαλείο χρησιμοποιείται για την παραγωγή εκτελέσιμου κώδικα σε διάφορες γλώσσες προγραμματισμού υψηλού επιπέδου. Ο παραγόμενος κώδικας μπορεί να χρησιμοποιηθεί είτε για εξομοίωση συστημάτων, είτε για τον τελικό έλεγχο αυτών. Τα πλεονεκτήματα της εξομοίωσης τόσο για την ορθή σχεδίαση ελεγκτών, όσο και για την εκπαίδευση νέων μηχανικών, είναι ευρέως αναγνωρισμένα και παρατίθενται στην επόμενη παράγραφο. Σε περίπτωση χρήσης σταθμού εποπτικού ελέγχου, ο παραγόμενος κώδικας μπορεί να ενσωματωθεί στο SCADA εποπτείας και να αποτελέσει τον ελεγκτή του συστήματος. Αυτό επιτυγχάνεται από τις ενσωματωμένες δυνατότητες των σύγχρονων λογισμικών SCADA που δέχονται κώδικα σε υψηλού επιπέδου γλώσσες προγραμματισμού. Ο κώδικας περιλαμβάνει όλες τις εντολές ενεργοποίησης εξόδων και υπολογισμού παραμέτρων σύμφωνα με τις επιταγές του αρχικού μοντέλου και τέλος ένα ή περισσότερα PLC αναλαμβάνουν το τελικό έργο ελέγχου της διεργασίας.

Το δεύτερο εργαλείο που παρουσιάζεται σε αυτό το κεφάλαιο χρησιμοποιείται για παραγωγή εκτελέσιμου κώδικα για προγραμματιζόμενους λογικούς ελεγκτές. Το εργαλείο παρουσιάζει τα συστηματικά βήματα που πρέπει να ακολουθήσει ο μηχανικός ελέγχου για να μετατρέψει το μοντέλο ενός συστήματος σε κώδικα για PLC βάσει του διεθνούς προτύπου IEC 61131-3. Το πρότυπο IEC 61131-3 περιλαμβάνει πέντε διαφορετικές γλώσσες οι οποίες υποστηρίζονται από το σύνολο των κατασκευαστών PLC (έστω εν μέρει). Όλες οι γλώσσες υποστηρίζονται από το παρουσιαζόμενο εργαλείο με αναλυτικά βήματα και παραδείγματα χρήσης για μια πιο ολοκληρωμένη παρουσίαση.

7.1 Οδηγός γραφής κώδικα για την εξομοίωση Γενικευμένων Αυτομάτων

Η ραγδαία ανάπτυξη της τεχνολογίας οδήγησε αναπόφευκτα στην αύξηση της πολυπλοκότητας των συστημάτων. Η εξομοίωση συστημάτων παρέχει την δυνατότητα κατανόησης της δομής τέτοιων συστημάτων προσομοιώνοντας την συμπεριφορά τους με κατάλληλη χρήση υλικού και λογισμικού. Γενικά η εξομοίωση μπορεί να ορισθεί ως: *«η διαδικασία σχεδίασης του μοντέλου ενός πραγματικού συστήματος και η διεξαγωγή πειραμάτων στο μοντέλο, με στόχο την κατανόηση της συμπεριφοράς του συστήματος ή την δοκιμή διαφόρων στρατηγικών λειτουργίας του συστήματος»*. Μια ερμηνεία του δοθέντος ορισμού θα έλεγε ότι ένας εξομοιωτής είναι ένα σύνολο υλικού και λογισμικού το οποίο χρησιμοποιείται για την μίμηση της συμπεριφοράς κάποιας οντότητας ή κάποιου φαινομένου. Τυπικά, τα συστήματα που εξομοιώνονται ποικίλουν από ολοκληρωμένα κυκλώματα έως σύνθετα βιομηχανικά συστήματα. Οι εξομοιωτές μπορούν επίσης να χρησιμοποιηθούν για την ανάλυση και επαλήθευση θεωρητικών μοντέλων. Συνεπώς γίνεται σαφές ότι η έννοια της εξομοίωσης παίζει σημαντικό ρόλο τόσο σε βιομηχανικά όσο και σε ακαδημαϊκά θέματα.

Παρόλη την γενική αποδοχή των εξομοιωτών ως χρήσιμα και απαραίτητα εργαλεία, θα πρέπει να υπάρχει επιφυλακτική αντιμετώπιση καθώς κρύβουν και ορισμένα προβλήματα. Τα προβλήματα σχετίζονται κυρίως με τους υπολογιστικούς περιορισμούς του διαθέσιμου υλικού ή της αυξημένης πολυπλοκότητας της διεργασίας που εξομοιώνεται. Τα πλεονεκτήματα και μειονεκτήματα της εξομοίωσης παρατίθενται επιγραμματικά παρακάτω.

Το κύριο πλεονέκτημα των εξομοιωτών είναι η ικανότητα παροχής στον χρήστη ιδιαίτερα σημαντικής ανάδρασης στην φάση σχεδίασης πραγματικών συστημάτων. Αυτό επιτρέπει στον σχεδιαστή να καθορίσει την ορθότητα της σχεδίασης πριν ολοκληρωθεί η κατασκευή του συστήματος και να εξετάσει ενδεχομένως εναλλακτικές μεθόδους. Η δυνατότητα έρευνας των πλεονεκτημάτων κάθε στρατηγικής κατά την φάση της σχεδίασης πολύ πριν την υλοποίηση μειώνει κατά πολύ το συνολικό κόστος κατασκευής ενός συστήματος, πράγμα ιδιαίτερα σημαντικό. Επιπρόσθετα, οι εξομοιωτές μπορούν να χρησιμοποιηθούν ως εκπαιδευτικά μέσα για την παρουσίαση της λειτουργίας συστημάτων σε μαθητές ή φοιτητές. Τέτοια εργαλεία κάνουν χρήση υπολογιστών και μέσω κινούμενων εικόνων παρουσιάζουν

την συμπεριφορά ενός συστήματος για την καλύτερη κατανόηση της λειτουργίας του.

Παρόλα τα σημαντικά πλεονεκτήματα, οι εξομοιωτές μπορεί να κρύβουν και σοβαρά μειονεκτήματα. Μερικά από αυτά προκύπτουν από την υπολογιστική ισχύ που μπορεί να απαιτεί ένας εξομοιωτής. Τα αποτελέσματα μιας εξομοίωσης μπορεί να μην είναι άμεσα διαθέσιμα αλλά να απαιτείται μεγάλο χρονικό διάστημα για την εξαγωγή τους. Για παράδειγμα, ένα συμβάν που μπορεί στον πραγματικό κόσμο να συμβαίνει στιγμιαία, μπορεί να απαιτεί λεπτά ή και ώρες για να εξομοιωθεί. Τέτοιου είδους καθυστερήσεις οφείλονται κυρίως στις περίπλοκες αλληλεπιδράσεις μεταξύ των στοιχείων του φυσικού συστήματος. Εν τούτοις, όσο αυξάνει η υπολογιστική ισχύ των μηχανημάτων, τόσο το πρόβλημα εξαλείφεται. Μια άλλη μεθοδολογία για την μείωση των προβλημάτων λόγω πολυπλοκότητας είναι η εισαγωγή κάποιων παραδοχών ή απλοποιήσεων στον εξομοιωτή. Η τεχνική αυτή μπορεί να μειώσει θεαματικά τον απαιτούμενο χρόνο εξομοίωσης, αλλά μπορεί συγχρόνως να δίνει μια ψευδή ασφάλεια περί της ορθότητας των αποτελεσμάτων.

Παρά τις όποιες αδυναμίες η εξομοίωση συστημάτων είναι αναγνωρισμένη διεθνώς ως ένα πολύ σημαντικό εργαλείο, τόσο στον βιομηχανικό κόσμο, όσο και σε εκπαιδευτικά θέματα. Σε αυτό το κεφάλαιο παρουσιάζεται ένας οδηγός για την κατασκευή μοντέλων εξομοίωσης για βιομηχανικά συστήματα βασισμένα στα γενικευμένα αυτόματα. Ο οδηγός μπορεί να χρησιμοποιηθεί εξίσου για εκπαιδευτικούς λόγους και για πρακτικούς λόγους υλοποίησης μοντέλων ΓΑ.

Στην συνέχεια παρουσιάζεται ο αναλυτικός οδηγός καθώς και παραδείγματα χρήσης του σε διάφορα προγραμματιστικά περιβάλλοντα.

7.1.1 Περιγραφή οδηγού γραφής κώδικα

Ο οδηγός γραφής κώδικα έχει την μορφή ενός αλγορίθμου για την γραφή προγραμμάτων και την εκτέλεση αυτών σε διάφορα προγραμματιστικά περιβάλλοντα και γλώσσες. Τα βήματα για την δημιουργία του μοντέλου ενός συστήματος δίνονται παρακάτω.

Δήλωση Μεταβλητών. Αρχικά πρέπει να δηλωθούν όλες οι δυαδικές και συνεχείς

μεταβλητές του συστήματος. Επιπρόσθετα μια μεταβλητή ακεραίου Q η οποία δηλώνει την τρέχουσα κατάσταση του συστήματος είναι απαραίτητη.

Αρχικές Συνθήκες. Μετά την δήλωσή τους, όλες οι μεταβλητές του συστήματος πρέπει να αρχικοποιηθούν βάσει των αρχικών συνθηκών του αυτομάτου. Η μεταβλητή Q παίρνει την κατάλληλη τιμή ούτως ώστε να δεικνύει την αρχική κατάσταση q_0 .

Βρόχος Εκτέλεσης. Ο βρόχος εκτέλεσης αποτελεί την καρδιά του προγράμματος, καθώς εδώ λαμβάνει χώρα η εκτέλεση της εξομοίωσης και προκύπτουν όλα χρήσιμα συμπεράσματα για την λειτουργία του συστήματος.

Στην περίπτωση χρήσης του εργαλείου MATLAB ή κάποιου συναφούς, ο βρόχος εκτέλεσης είναι ένας βρόχος FOR-NEXT και πρωτίστως πρέπει να ορισθεί ο αριθμός επαναλήψεων χρησιμοποιώντας κάποια βοηθητική μεταβλητή (π.χ. FOR I=1:1000). Συνήθως, ένα κύριο στοιχείο κάθε εξομοίωσης είναι ο χρόνος, οπότε κάθε επανάληψη του βρόχου πρέπει να αντιστοιχεί σε μια χρονική στιγμή, πράγμα που υλοποιείται με εντολή του τύπου $t=t+0.001$ ώστε να έχουμε χρονική ακρίβεια χιλιοστού του δευτερολέπτου.

Κάθε βρόχος εκτελεί μια οντότητα που έχει την μορφή είτε SELECT – CASE είτε SWITCH – CASE. Κάθε CASE αναπαριστά μία κατάσταση του αυτομάτου και η επιλογή μεταξύ όλων γίνεται βάσει της μεταβλητής Q . Κάθε CASE επίσης περιέχει όλα τα στοιχεία της κατάστασης, όπως συναρτήσεις, γεγονότα, πίνακες αρχικοποίησης κτλ. Κάθε στοιχείο αναπαριστάται με μία εντολή εντός της οντότητας CASE. Η ιεραρχία έχει κυρίαρχο ρόλο και συνεπώς πρέπει να τηρείται ευλαβικά.

Εντολές Αναπαράστασης Συναρτήσεων. Οι συναρτήσεις των δυαδικών μεταβλητών είναι οι γνωστές συναρτήσεις της άλγεβρας Boole. Οι συναρτήσεις των συνεχών μεταβλητών μπορούν να είναι είτε χρονικές συναρτήσεις, είτε εξισώσεις διαφορών, αντίστοιχες των διαφορικών εξισώσεων.

Εντολές Αναπαράστασης Μεταβάσεων. Εξ ορισμού κάθε κατάσταση έχει ένα σύνολο ενεργών γεγονότων που μπορούν να προκαλέσουν μια μετάβαση μεταξύ καταστάσεων. Τα ενεργά γεγονότα πρέπει να περιλαμβάνονται στην CASE οντότητα

της κατάστασης και παίρνουν την μορφή IF – THEN. Κάθε μετάβαση προκαλείται από ένα γεγονός, αν αυτό δεν αγνοείται, και έχει ως αποτέλεσμα την αλλαγή της ενεργής κατάστασης και την αρχικοποίηση ενός συνόλου μεταβλητών. Αν υπάρχουν περισσότερα του ενός ενεργά γεγονότα σε μια κατάσταση, τότε αντίστοιχες δομές IF – THEN απαιτούνται σε πλήρη εναρμονισμό με το σύνολο Σ_i , το οποίο δίνει την ιεραρχία των γεγονότων προς εκτέλεση.

Ανακεφαλαιώνοντας την παρουσίαση του οδηγού, ένα παράδειγμα της μορφής του παρουσιάζεται παρακάτω, ενώ μια περιληπτική οπτική σύνοψη του εργαλείου δίνεται στο σχήμα 7.1. Ως πλατφόρμα υλοποίησης παρουσιάζεται ένα σύστημα εποπτικού ελέγχου SCADA, καθώς αποτελεί ένα λογισμικό εργαλείο με ευρεία αποδοχή και χρήση στον βιομηχανικό χώρο.

```
FOR  $I = 1 : 1000$ 
   $t = t + 0.001$ 
  SWITCH Q
    CASE 1
       $Z_1(i+1) = \bar{A} + Z_1(i) \cdot B$ 
       $X_1(i+1) = X_1(i) + \dots$ 
       $X_2(i+1) = f(t)$ 
      IF  $event_1$  occurs AND is not ignored THEN
         $Q = 4$ 
         $X = R_{X1}$ 
         $Z = R_{Z1}$ 
      ENDIF
      IF  $event_2$  occurs AND is not ignored THEN
         $Q = 3$ 
         $X = R_{X2}$ 
         $Z = R_{Z2}$ 
      ENDIF
    END – CASE
    CASE 2
      ...
    END – CASE
    ...
  END – SWITCH
END – FOR
```

Σχήμα 7.1. Οδηγός γραφής κώδικα για σταθμό SCADA.

7.1.2 Η εξομοίωση γενικευμένων αυτομάτων σε πλατφόρμα SCADA.

Ο οδηγός γραφής κώδικα που παρουσιάστηκε στην προηγούμενη παράγραφο μπορεί να έχει ως πλατφόρμα υλοποίησης διάφορες γλώσσες προγραμματισμού ή λογισμικά πακέτα. Στην επόμενη παράγραφο δίνονται παραδείγματα υλοποίησης γενικευμένων αυτομάτων σε διαφορετικά προγραμματιστικά περιβάλλοντα, δίνοντας ιδιαίτερη βαρύτητα στα συστήματα εποπτικού ελέγχου και συλλογής πληροφοριών (SCADA). Τα SCADA συστήματα είναι ευρέως γνωστά στους μηχανικούς της βιομηχανίας και

λόγω αυτού επιλέγονται ως το καταλληλότερο μέσο για την εξομίωση και υλοποίηση ελεγκτών βάσει του παρουσιαζόμενου οδηγού. Στο σχήμα 7.2 απεικονίζεται παραστατικά η υλοποίηση ενός ελεγκτή σε σταθμό εποπτικού ελέγχου σε συνδυασμό με την δομή ενός συστήματος SCADA. Ο ελεγκτής προσαρτάται στο σύστημα ως επισυναπτόμενη διεργασία.

Σχήμα 7.2. Ο ελεγκτής στο SCADA.

7.1.3 Παραδείγματα χρήσης του οδηγού

Στην παράγραφο αυτή παρουσιάζονται κάποια παραδείγματα που επεξηγούν την χρήση του οδηγού στην δημιουργία μοντέλων βιομηχανικών διεργασιών. Τα παραδείγματα υλοποιούνται είτε σε MATLAB, είτε σε CX-Supervisor, αλλά δεν υπάρχουν περιορισμοί όσον αφορά το λογισμικό περιβάλλον. Το MATLAB χρησιμοποιείται κυρίως λόγω του ότι είναι το πιο ευρέως διαδεδομένο λογισμικό πακέτο-εργαλείο στα Πανεπιστήμια όλου του κόσμου. Όσον αφορά το CX-Supervisor χρησιμοποιήθηκε λόγω της φύσης του ως ένα λογισμικό SCADA (Supervisory Control And Data Acquisition), δηλαδή ενός Σταθμού Εποπτικού Ελέγχου και Συλλογής Δεδομένων. Τα λογισμικά SCADA είναι πολύ γνώριμα με τον βιομηχανικό κόσμο καθώς όλοι οι μηχανικοί ελέγχου της βιομηχανίας τα χρησιμοποιούν. Επίσης το συγκεκριμένο λογισμικό παρέχει την δυνατότητα εξομοίωσης του εκάστοτε συστήματος που μοντελοποιείται.

7.1.4 Ένας οικιακός θερμοστάτης

Το γενικευμένο αυτόματο που μοντελοποιεί έναν οικιακό θερμοστάτη απεικονίζεται στο σχήμα 7.3. Η πραγματική μεταβλητή x αναπαριστά την θερμοκρασία. Οι δύο καταστάσεις τιτλοφορούνται ως *On* και *Off* και καθεμία έχει μια διαφορική εξίσωση βάση της οποίας η θερμοκρασία ανεβαίνει ή πέφτει. Αρχικά η θερμοκρασία είναι 20° C και η θερμάστρα είναι σβηστή. Σύμφωνα με την συνθήκη μετάβασης η θερμοκρασία πέφτει μέχρι τους 18° C. Τότε η θερμάστρα ανάβει και η θερμοκρασία ανεβαίνει μέχρι τους 23° C όπου η θερμάστρα ξανασβήνει.

Σχήμα 7.3. Αυτόματο θερμοστάτη.

Το απλό αυτόματο υλοποιήθηκε σε MATLAB. Ο κώδικας που γράφτηκε βάσει του οδηγού δίνεται παρακάτω. Τέλος το σχήμα 7.4 δίνει την εξομοίωση του αυτομάτου

δείχνοντας την διακύμανση της θερμοκρασίας.

```
x(1)=20;      %Initial temperature
k=0;          %Thermostat's initial status
q=1;          %Initial state
for i=1:100
    switch q
    case 1
        x(i+1)=0.9*x(i);    %Difference equation analogue to  $x'=-0.9x$ 
        k=0;
        if x(i+1)<18
            q=2;
        end
    case 2
        k=1;
        x(i+1)=10+0.9*x(i); %Equation analogue to  $x'=10-0.1x$ 
        if x(i+1)>23
            q=1;
        end
    end
end
end
plot(x),grid
```

Σχήμα 7.4. Εξομοίωση αυτομάτου.

7.1.5 Τοπικό δίκτυο υπολογιστών με πέρασμα κουπονιού

Το αυτόματο του σχήματος 7.5 μοντελοποιεί ένα τοπικό δίκτυο με τέσσερις κόμβους και χρήση του πρωτοκόλλου περάσματος κουπονιού. Οι τέσσερις καταστάσεις αναπαριστούν τον λογικό δακτύλιο μεταξύ των κόμβων. Όταν το αυτόματο βρίσκεται στην κατάσταση q_i , ο κόμβος i έχει το κουπόνι και την δυνατότητα χρήσης του

διαύλου. Η μεταβλητή t αναπαριστά το χρόνο, ενώ η τ είναι μια πρόσθετη μεταβλητή ρολογιού. Οι ακέραιες μεταβλητές M_1, M_2, M_3 και M_4 αντιπροσωπεύουν στον αριθμό πακέτων προς εξυπηρέτηση στην ουρά κάθε κόμβου. Οι διακριτές μεταβλητές s_1, s_2, s_3 και s_4 αναπαριστούν την άφιξη ενός νέου πακέτου στην ουρά του αντίστοιχου κόμβου. Η σταθερή παράμετρος T_s είναι ο χρόνος μετάδοσης του κουπονιού, ενώ P_s είναι ο χρόνος που απαιτείται για την μετάδοση ενός πακέτου δεδομένων.

Σχήμα 7.5. Δίκτυο περάσματος κουπονιού με τέσσερις κόμβους.

Όλες οι μεταβάσεις με τίτλο Hs_i προκαλούνται από τα γεγονότα s_i και αυξάνουν τους μετρητές M_i κατά μία μονάδα, για $i=1,2,3,4$. Οι μεταβάσεις Ht_i προκαλούνται από το γεγονός $\tau = T_s + P_s$ και επιφέρουν τον μηδενισμό της μεταβλητής τ και την μείωση των μετρητών M_i κατά μία μονάδα. Τέλος, οι μεταβάσεις HT προκαλούνται από το γεγονός $\tau = T_s$ και μηδενίζουν την μεταβλητή τ . Σε κάθε κατάσταση υπάρχουν δύο διαφορεικές εξισώσεις, μία για κάθε μεταβλητή χρόνου: $\dot{t} = 1$ και $\dot{\tau} = 1$. Υπάρχει επίσης ένα ζεύγος περιορισμού και γεγονότος που αγνοείται, το οποίο είναι: $[\tau = T_s], \{M_i > 0\}$. Η σημασία του είναι ότι όταν το αυτόματο είναι στην κατάσταση q_i και $M_i > 0$, τότε το γεγονός $\tau = T_s$ θα αγνοηθεί. Συνεπώς, η μετάβαση από την κατάσταση q_i στην κατάσταση q_{i+1} πραγματοποιείται μετά από T_s μονάδες χρόνου, όταν $M_i = 0$, ή διαφορετικά μετά από $T_s + P_s$ μονάδες χρόνου.

Αρχικά το αυτόματο είναι στην κατάσταση q_1 και παραμένει εκεί μέχρι $t=T_s$, αν $M_1=0$, ή μέχρι $t=T_s+P_s$, αν $M_1>0$. Η παραπάνω συνθήκη σημαίνει ότι αν ο κόμβος 1 δεν έχει πακέτο προς αποστολή, η διαχείριση του διαύλου θα περάσει στον κόμβο 2 αφού περάσει χρόνος ίσος με το χρόνο του κουπονιού (T_s). Σε κάθε άλλη περίπτωση ($M_1>0$), ο κόμβος 1 θα στείλει το πρώτο πακέτο που βρίσκεται στην ουρά του και η μετάβαση στον κόμβο 2 θα γίνει μετά από χρόνο T_s+P_s (χρόνος κουπονιού και χρόνος αποστολής πακέτου αντίστοιχα). Αυτοί είναι οι δύο διαφορετικοί τύποι γεγονότων που προκαλούν την μετάβαση από την τρέχουσα κατάσταση στην αμέσως επόμενη.

Μία εκτέλεση του αυτομάτου φαίνεται παρακάτω παρουσιάζοντας τις καταστάσεις και τις μεταβλητές του συστήματος σε συνάρτηση με το χρόνο. Για το συγκεκριμένο παράδειγμα υποθέσαμε ότι $T_s=1$ και $P_s=2$.

7.1.6 Εξομοίωση λειτουργίας τριών μηχανών με παύση αυτών υπό ειδικές συνθήκες

Το δεύτερο παράδειγμα σχετίζεται με την λειτουργία τριών μηχανών και την παύση αυτών υπό προϋποθέσεις και έχει παρουσιαστεί αναλυτικά στην παράγραφο 4.3.1.

Για την εξομοίωση του συστήματος χρησιμοποιήθηκε το μοντέλο με τις τρεις καταστάσεις που απεικονίζεται στο σχήμα 7.6.

Σχήμα 7.6. Αυτόματο τριών μηχανών.

Ο κώδικας που δημιουργήθηκε βάσει του οδηγού εκτελέστηκε σε MATLAB και δίνεται εδώ:

```
case 1
    if s1==TRUE & M1==FALSE
        M1=TRUE;
        k=k+1;
    end
    if s2==TRUE & M2==FALSE
        M2=TRUE;
        k=k+1;
    end
    if s3==TRUE & M3==FALSE
        M3=TRUE;
        k=k+1;
    end
    if p1==TRUE & M1==TRUE & k~=1
        M1=FALSE;
        k=k-1;
    end
    if p2==TRUE & M2==TRUE & k~=1
        M2=FALSE;
        k=k-1;
    end
    if p3==TRUE & M3==TRUE & k~=1
        M3=FALSE;
        k=k-1;
    end
    if a==TRUE
        q=2;
    end
end
```

7.2 Οδηγός γραφής κώδικα για Προγραμματιζόμενους Λογικούς Ελεγκτές.

Ο κυρίαρχος ελεγκτής στον βιομηχανικό κόσμο είναι ο Προγραμματιζόμενος Λογικός Ελεγκτής (PLC). Η καθιερωμένη πρακτική ελέγχου περιλαμβάνει τον προγραμματισμό των PLCs σε κάποια από τις χρησιμοποιούμενες γλώσσες προγραμματισμού. Στην προσπάθεια προτυποποίησης του προγραμματισμού PLC δημιουργήθηκε από την *International Electrotechnical Commission* το 1993 το διεθνές πρότυπο IEC 61131-3, ενώ η νεότερη δεύτερη έκδοση παρουσιάστηκε το 2003 και έχει γίνει δεκτό από το σύνολο των κατασκευαστών PLC. Το πρότυπο καθορίζει τα κοινά στοιχεία των προγραμμάτων, όπως δομή προγράμματος, δήλωση μεταβλητών κτλ. Καθώς και την σύνταξη διάφορων γλωσσών προγραμματισμού. Το πρότυπο περιλαμβάνει πέντε γλώσσες προγραμματισμού που είναι: Instruction List (IL), Ladder Diagram (LAD), Function Block Diagram (FBD), Structured Text και Sequential Function Charts (SFC). Οι τρεις πρώτες υποστηρίζονται σχεδόν από το σύνολο των κατασκευαστών, ενώ οι άλλες δύο συγκαταλέγονται στις γλώσσες υψηλού επιπέδου με μικρότερη όμως απήχηση στους κατασκευαστές. Αξίζει να σημειωθεί ότι καθώς το πρότυπο δεν είναι αρκετά περιοριστικό δεν υπάρχει μεταφερσιμότητα προγραμμάτων μεταξύ ελεγκτών διαφορετικών κατασκευαστών.

Σε αυτό το κεφάλαιο παρουσιάζεται ένα εργαλείο υλοποίησης γενικευμένων αυτομάτων σε Προγραμματιζόμενους Λογικούς Ελεγκτές βάσει του προτύπου IEC 61131-3. Το εργαλείο περιγράφει συστηματικά όλα τα απαραίτητα βήματα που πρέπει να ακολουθήσει ο μηχανικός ελέγχου για να υλοποιήσει την στρατηγική ελέγχου, όπως αυτή περιγράφεται σε ένα γενικευμένο αυτόματο, σε ένα PLC. Ο γράφος ελέγχου και το σύνολο των παραμέτρων του αυτομάτου μετατρέπονται σε κώδικα που φορτώνεται στο PLC για τον έλεγχο του συστήματος. Το έργο μηχανικού για τον έλεγχο ενός βιομηχανικού συστήματος απεικονίζεται σχηματικά στο σχήμα 7.7. Το πρώτο βήμα περιλαμβάνει την επιθεώρηση του συστήματος και την καταγραφή όλων των επιθυμητών προδιαγραφών. Ανάλογα με τις προδιαγραφές και τα ιδιαίτερα χαρακτηριστικά του συστήματος, ο μηχανικός δημιουργεί το γενικευμένο αυτόματο που ικανοποιεί όλα τα κριτήρια ελέγχου. Με την ολοκλήρωση του αυτομάτου και την επαλήθευσή του μέσω εξομοίωσης, πρέπει να μετατραπεί σε κώδικα για PLC με χρήση του παρουσιαζόμενου εργαλείου. Τέλος, ο παραγόμενος κώδικας φορτώνεται στο PLC και το πλήρες σύστημα τίθεται σε δοκιμαστική

λειτουργία. Πιθανές δυσλειτουργίες επιφέρουν επαναπροσδιορισμό των χαρακτηριστικών του συστήματος και την αναθεώρηση του αρχικού αυτομάτου.

Σχήμα 7.7. Έργο μηχανικού ελέγχου.

Το εργαλείο υλοποίησης παρουσιάζεται ως ένα εργαλείο σύνθεσης περιγράφοντας τα συστηματικά βήματα δημιουργίας κώδικα για τις γλώσσες του προτύπου IEC 61131-3. Μια περιληπτική σύνοψη του εργαλείου για τις διάφορες γλώσσες του προτύπου δίνεται στο σχήμα 7.8. Ακολουθεί η γενική παράθεση των βημάτων και στην συνέχεια η αναλυτική περιγραφή γραφής κώδικα βάσει του προτύπου.

- *Δήλωση Μεταβλητών.* Αρχικά, όλες οι μεταβλητές του συστήματος πρέπει να δηλωθούν, δηλαδή να ορισθεί ο τύπος κάθε μεταβλητής, πχ. δυαδική μεταβλητή, ακέραιος, πραγματικός κτλ. Επιπρόσθετα μια ακέραια μεταβλητή Q , η οποία θα δηλώνει την τρέχουσα κατάσταση του συστήματος, είναι απαραίτητη.
- *Αρχικές Συνθήκες.* Μετά την δήλωσή τους, όλες οι μεταβλητές πρέπει να αρχικοποιηθούν σύμφωνα με τις αρχικές συνθήκες του αυτομάτου. Επιπλέον, η μεταβλητή Q παίρνει κατάλληλη τιμή ώστε να υποδηλώνει την αρχική κατάσταση του αυτομάτου.
- *Κυρίως Πρόγραμμα.* Τα δύο πρώτα βήματα αποτελούν την θεμελίωση του όλου έργου, αλλά το σημαντικότερο βήμα είναι η δημιουργία του

εκτελέσιμου κώδικα. Το βήμα αυτό χωρίζεται σε τρία στάδια, όπου όλες οι παράμετροι του αυτομάτου μετατρέπονται σε κώδικα.

Synthesis Tool Overview

Σχήμα 7.8. Σύνοψη εργαλείου.

- ο *Επιλογή Ενεργής Κατάστασης*. Το συνολικό πρόγραμμα περιλαμβάνει εντολές για κάθε κατάσταση του αυτομάτου και συνεπώς είναι απαραίτητος ένας μηχανισμός επιλογής των εντολών της ενεργής κατάστασης και μόνο.
- ο *Εντολές Συναρτήσεων*. Σε κάθε κατάσταση δίνονται όλες οι συναρτήσεις που περιέχει η κατάσταση του αυτομάτου.
- ο *Εντολές Μεταβάσεων*. Τα ενεργά γεγονότα κάθε κατάστασης πρέπει να περιέχονται στον αντίστοιχο κώδικα. Γενικά, μια μετάβαση προκαλείται από ένα γεγονός υπό την προϋπόθεση ότι δεν είναι υπό περιορισμό και αποφέρει την αλλαγή της ενεργής κατάστασης και

την αρχικοποίηση ορισμένων μεταβλητών σύμφωνα με τον αντίστοιχο πίνακα αρχικοποίησης.

7.2.1 Οδηγός για προγραμματισμό σε IL, LAD και FBD.

Ο οδηγός για τις τρεις αυτές γλώσσες δίνεται από κοινού, καθώς όλες οι χρησιμοποιούμενες εντολές μπορούν να μεταφραστούν από την μία γλώσσα στις άλλες και αντιστρόφως. Τα περισσότερα λογισμικά προγραμματισμού (π.χ. STEP 7 για Siemens PLCs) που υποστηρίζουν τις τρεις γλώσσες έχουν ειδικές δυνατότητες για την μετάφραση προγραμμάτων μεταξύ αυτών. Τα συστηματικά βήματα περιγράφονται σε πλήρη αρμονία με την σύνοψη του εργαλείου όπως παρουσιάστηκε παραπάνω, ενώ ακολουθούν την σύνταξη του Siemens SIMATIC STEP7, ώστε τα παραγόμενα παραδείγματα να είναι πλήρως λειτουργικά.

- *Δήλωση Μεταβλητών.* Όλες οι μεταβλητές του συστήματος δηλώνονται στο αντίστοιχο εργαλείο του προγραμματιστικού περιβάλλοντος (π.χ. Variable Table στην STEP7).
- *Αρχικές Συνθήκες.* Όλες οι μεταβλητές αρχικοποιούνται με χρήση του λειτουργικού μπλοκ (operation block OB100), το οποίο στην STEP7 εκτελείται μία φορά σε κάθε επανεκκίνηση του συστήματος. Ένα παράδειγμα του απαιτούμενου κώδικα σε IL δίνεται στο σχήμα 7.9.

Σχήμα 7.9. Αρχικοποίηση μεταβλητών σε IL.

- *Κυρίως Πρόγραμμα.* Στο κυρίως πρόγραμμα χρησιμοποιείται ιεραρχική δόμηση προγράμματος για βέλτιστη εκτέλεση. Συγκεκριμένα γίνεται χρήση του operation block OB1 για την επιλογή της ενεργής κατάστασης η οποία εκτελείται ως μια συνάρτηση (Function Block FB). Συνεπώς απαιτείται μια συνάρτηση για κάθε κατάσταση του αυτομάτου. Τα τρία μέρη του κυρίως

προγράμματος δίνονται παρακάτω. Το πρώτο υλοποιείται στο OB1, ενώ τα δύο άλλα στις συναρτήσεις της κάθε κατάστασης (FB).

- ο *Επιλογή Ενεργής Κατάστασης*. Στο μπλοκ OB1 δημιουργούμε μια δομή Jump to List (JL), ελεγχόμενη από την τιμή της μεταβλητής q η οποία δείχνει την ενεργή κατάσταση. Η δομή JL αποτελεί έναν διανομέα λίστας και ακολουθείται από ένα σύνολο μεταπηδήσεων άνευ συνθήκης προς συγκεκριμένες θέσεις του κώδικα. Το σχήμα 7.10 δείχνει το σχετικό διάγραμμα ροής. Κάθε τμήμα καλεί το αντίστοιχο FB που αναπαριστά τις λειτουργίες κάθε κατάστασης.

Σχήμα 7.10. Επιλογή ενεργής κατάστασης με χρήση της δομής Jump to List.

- ο *Εντολές Συναρτήσεων*. Σε κάθε κατάσταση δίνονται όλες οι συναρτήσεις που περιέχει η κατάσταση του αυτομάτου. Αναφορικά με τις διακριτές μεταβλητές, οι συναρτήσεις είναι εξισώσεις της άλγεβρας Boole, οι οποίες μεταφράζονται εύκολα σε κώδικα για κάθε από τις τρεις γλώσσες. Αντιθέτως, οι συναρτήσεις των συνεχών μεταβλητών είναι συνήθως χρονικές συναρτήσεις ή

διαφορικές εξισώσεις. Τα PLCs δεν έχουν την δυνατότητα χειρισμού διαφορικών εξισώσεων, οπότε θα πρέπει να μετασχηματιστούν σε εξισώσεις διαφορών [85]. Για την αναπαράσταση των συναρτήσεων αυτών γίνεται χρήση των μαθηματικών τελεστών της κάθε γλώσσας (π.χ. +R για πρόσθεση πραγματικών αριθμών, /I για διαίρεση ακεραίων, SIN, EXP, LN, SQRT κτλ.). Στις γλώσσες FBD και LAD, οι μαθηματικές εξισώσεις δημιουργούνται με χρήση των ενσωματωμένων συναρτησιακών μπλοκ. Ένα παράδειγμα υπολογισμού της εξίσωσης $x = \sin(13 \cdot y) + k$ δίνεται στο σχήμα 7.11. Βεβαίως, ειδικές συναρτήσεις, όπως ελεγκτές τριών όρων PID, και ασαφείς ελεγκτές μπορούν να χρησιμοποιηθούν ως συναρτήσεις σε κάποια κατάσταση. Επίσης, πρέπει να σημειωθεί ότι στο FB κάθε κατάστασης οι εντολές των συναρτήσεων πρέπει να τοποθετούνται πάνω από τις εντολές μεταβάσεων.

Σχήμα 7.11. Παράδειγμα συναρτήσεων σε IL και LAD.

- ο *Εντολές Μεταβάσεων*. Τα ενεργά γεγονότα κάθε κατάστασης πρέπει να περιέχονται στον αντίστοιχο κώδικα. Γενικά, μια μετάβαση

προκαλείται από ένα γεγονός (π.χ. $I_1=TRUE$), υπό την προϋπόθεση ότι δεν είναι υπό περιορισμό (π.χ. $x>23$) και αποφέρει την αλλαγή της ενεργής κατάστασης (π.χ. $q=3$) και την αρχικοποίηση ορισμένων μεταβλητών σύμφωνα με τον αντίστοιχο πίνακα αρχικοποίησης (π.χ. $k=2$ και $f=TRUE$). Οι κλάδοι του πίνακα αρχικοποίησης και της αλλαγής ενεργής κατάστασης είναι δομές MOVE. Επιπρόσθετα, για την αποφυγή εκτέλεσης περισσότερων της μιας μεταβάσεων είναι απαραίτητη η άνευ συνθήκης μεταπήδηση στο τέλος της συνάρτησης της κατάστασης. Η ιεράρχηση των ενεργών γεγονότων δηλώνεται από την θέση των αντίστοιχων εντολών στον κώδικα. Το γεγονός με την ύψιστη προτεραιότητα τοποθετείται πρώτο, ενώ το αυτό με την ελάχιστη τελευταίο. Ένα παράδειγμα μεταβάσεων σε LAD και FBD δίνεται στο σχήμα 7.12.

Σχήμα 7.12. Παράδειγμα μεταβάσεων FBD και LAD.

Οι ενέργειες, που δόθηκαν παραπάνω παρουσιάζοντας τα βήματα του οδηγού προγραμματισμού, συνοψίζονται για την γλώσσα Instruction List στον πίνακα της σελίδας 117.

Οι αντίστοιχοι πίνακες για τις γλώσσες LAD και FBD είναι πανομοιότυποι με τον παραπάνω πίνακα, προσαρμοσμένοι κατάλληλα με μόνη διαφορά την αντικατάσταση των εντολών της IL από τις αντίστοιχες εντολές των LAD και FBD.

7.2.2 Οδηγός για προγραμματισμό σε Structured Text.

Η γλώσσα Structured Text (ST) είναι μια βασισμένη σε κείμενο γλώσσα υψηλού επιπέδου με σύνταξη παρόμοια με της γλώσσας PASCAL. Παρά τις όποιες ομοιότητες με την PASCAL η γλώσσα ST είναι μια ξεχωριστή γλώσσα σχεδιασμένη κατάλληλα για εφαρμογές βιομηχανικού ελέγχου. Τα βήματα για την παραγωγή κώδικα σε ST από ένα γενικευμένο αυτόματο, δίνονται παρακάτω ακολουθώντας την ίδια δομή όπως και για τις άλλες γλώσσες.

- *Δήλωση Μεταβλητών – Αρχικές Συνθήκες.* Στον προγραμματισμό με ST η δήλωση μεταβλητών και η αρχικοποίηση του συστήματος μπορεί να επιτευχθεί συγχρόνως. Η δήλωση γίνεται για άλλη μια φορά στο OB100 όπου απλές εντολές δίνουν τον τύπο και την αρχική τιμή κάθε μεταβλητής (σχήμα 7.13).

π.χ. q : INT :=1; (*Αρχική κατάσταση q1*)
X1 : REAL := 13; (*Πραγματική μεταβλητή X1=13*)
X4 : BOOL := 0; (*Δυαδική μεταβλητή X4=FALSE*)

Σχήμα 7.13. Δήλωση μεταβλητών και αρχικοποίηση σε ST.

PLC TOOL - ΟΔΗΓΟΣ ΓΡΑΦΗΣ ΚΩΔΙΚΑ ΓΙΑ PLC

- *Κυρίως Πρόγραμμα.* Σε κάθε βρόχο εκτέλεσης, το κυρίως πρόγραμμα εκτελείται σε δύο μέρη-ρουτίνες. Το πρώτο είναι το OB1 όπου επιλέγεται η ενεργή κατάσταση, και το δεύτερο είναι η συνάρτηση (function block) της ενεργής κατάστασης που περιλαμβάνει τις συναρτήσεις και τις μεταβάσεις. Οι εντολές των συναρτήσεων δίνονται πρώτα και στην συνέχεια οι εντολές που αναπαριστούν τις μεταβάσεις.

- ο *Επιλογή Ενεργής Κατάστασης.* Ο μηχανισμός επιλογής της ενεργής κατάστασης είναι μια δομή CASE με επιλογή βάσει τιμής ακεραίου. Σε αυτή την δομή, οι εντολές που έχουν σαν δείκτη την τιμή του ακεραίου επιλογής εκτελούνται. Ως ακέραιος επιλογής χρησιμοποιείται το *q* που υποδηλώνει την ενεργή κατάσταση και εκτελείται η εντολή που καλεί την συνάρτηση της αντίστοιχης κατάστασης.

π.χ.. CASE q OF (*Επιλογή βάσει της τιμής του q*)
 1 : FB1; (*If q=1, call FB1*)
 2 : FB2; (*If q=2, call FB2*)
 ...
 n : FBn; (*If q=n, call FBn*)
 END_CASE;

- ο *Εντολές Συναρτήσεων.* Σε κάθε κατάσταση δίνονται όλες οι συναρτήσεις που περιέχει η κατάσταση του αυτομάτου. Οι εξισώσεις της άλγεβρας Boole δίνονται στην ST με χρήση των λογικών τελεστών (π.χ. AND, &, OR, XOR, NOT, κτλ.). Οι συναρτήσεις των συνεχών μεταβλητών δίνονται ως εξισώσεις διαφορών με χρήση των αντίστοιχων μαθηματικών τελεστών (σχήμα 7.14).

π.χ. Area := PI * R * R;
 Delta := Delta + 0.2*R;
 V := K ** (-W*T);
 Start := Valve AND Steam AND Pump;
 Status := (Valve1 = Open) XOR (Valve2 = Shut);

Σχήμα 7.14. Συναρτήσεις σε ST.

- Εντολές Μεταβάσεων.* Τα ενεργά γεγονότα κάθε κατάστασης πρέπει να περιέχονται στον αντίστοιχο κώδικα με χρήση δομής IF – THEN. Κάθε μετάβαση προκαλείται από ένα γεγονός, εφόσον δεν υπόκεινται σε περιορισμό, και προκαλεί την αλλαγή της ενεργής κατάστασης και την αρχικοποίηση μεταβλητών. Σε περίπτωση περισσότερων του ενός μεταβάσεων, χρειάζονται αντίστοιχες δομές IF – THEN, οι οποίες δίνονται σε αρμονία με την ιεράρχηση των γεγονότων κάθε κατάστασης. Η γενική μορφή μιας μετάβασης δίνεται παρακάτω, ενώ το σχήμα 7.15 παρουσιάζει ένα χαρακτηριστικό παράδειγμα.

```

IF <event_occurs> AND <is_not_ignored> THEN
    <next_state>;
    <reset tables>;
END_IF

```

Σχήμα 7.15. Μεταβάσεις σε ST.

```
π.χ.    IF (pump = ON) AND (temp > 200) THEN
          q := 4;
          Valve1 := off;
          X2 := 100;
        END_IF
```

7.2.3 Οδηγός για προγραμματισμό σε Sequential Function Chart.

Η γλώσσα Sequential Function Chart (SFC) αποτελεί μια ισχυρή γραφική τεχνική περιγραφής της ακολουθιακής συμπεριφοράς ενός προγράμματος ελέγχου. Η SFC χρησιμοποιείται κυρίως σε ένα ανώτερο ιεραρχικά επίπεδο για να δείξει της διαδοχικές φάσεις μιας διεργασίας σε συνδυασμό με κάποια άλλη γλώσσα στα κατώτερα επίπεδα. Βέβαια, η SFC μπορεί να χρησιμοποιηθεί και ως γλώσσα κατώτερου επιπέδου, ώστε να μην είναι αναγκαία η χρήση εναλλακτικής γλώσσας. Σε αυτή την παράγραφο δίνονται μόνο δύο παραδείγματα υλοποίησης του μηχανισμού επιλογής ενεργής κατάστασης και των μεταβάσεων, καθώς όλες οι άλλες εργασίες στην SFC μπορούν να αναπαρασταθούν με χρήση κάποιας από τις γλώσσες FBD, LAD ή ST και τα αντίστοιχα παραδείγματα έχουν ήδη δοθεί. Το σχήμα 7.16 δίνει την δομή του OB1 για την επιλογή ενεργής κατάστασης και κλήσης της ανάλογης συνάρτησης προγράμματος. Το παράδειγμα δίνεται για περίπτωση αυτομάτου με τέσσερις μόνο καταστάσεις.

Σχήμα 7.16. Το μπλοκ OB1 σε SFC.

Αντίστοιχα, το σχήμα 7.17 παρουσιάζει την δομή μιας μετάβασης, με το γεγονός που την προκαλεί, τον περιορισμό του γεγονότος και την αλλαγή κατάστασης και αρχικοποίησης μεταβλητών.

Σχήμα 7.17. Δομή μετάβασης σε SFC.

7.2.4 Συστήματα με μοντέλο πολλαπλών αυτομάτων.

Σε ορισμένες περιπτώσεις, το συνολικό σύστημα μοντελοποιείται με δύο ή περισσότερα αυτόματα και υπάρχει η απαίτηση για παράλληλη εκτέλεση αυτών. Ένα τέτοιο παράδειγμα δίνεται στο κεφάλαιο 8, όπου το υπό μελέτη σύστημα είναι μια μονάδα παραγωγής έτοιμου σκυροδέματος. Το πλήρες σύστημα μοντελοποιείται με πέντε γενικευμένα αυτόματα που εκτελούνται παράλληλα. Σε σύγκριση με τα συστήματα ενός αυτομάτου, η μόνη διαφορά έγκειται στον μηχανισμό επιλογής ενεργής κατάστασης, που πρέπει να επιλέγει ενεργές καταστάσεις για κάθε ξεχωριστό αυτόματο και να καλεί τις αντίστοιχες συναρτήσεις προγράμματος. Η ιεραρχία επιλογής (ιεραρχία κλήσης των συναρτήσεων κάθε κατάστασης) υποδηλώνει την ιεραρχία εκτέλεσης των αυτομάτων.

7.2.5 Παράδειγμα χρήσης οδηγού.

Η παρουσίαση του οδηγού ολοκληρώνεται με ένα απλό παράδειγμα χρήσης του για την παραγωγή εκτελέσιμου κώδικα για PLC. Το σύστημα που εξετάζεται υλοποιεί μια ηλεκτρονική κλειδαριά με έλεγχο από δύο ψηφιακές μεταβλητές X και Y . Όταν το σύστημα τίθεται σε λειτουργία, βρίσκεται σε μια αρχική κατάσταση και η συμπεριφορά του εξαρτάται από τις δύο μεταβλητές. Η κλειδαριά πρέπει να ανοίξει εάν τα γεγονότα X και Y ενεργούν με την ακόλουθη σειρά:

1. Αρχικά το X ενεργοποιείται.
2. Στην συνέχεια το Y ενεργοποιείται και απενεργοποιείται.
3. Τότε, η πρώτη έξοδος ενεργοποιείται, ξεκλειδώνοντας τον πρώτο πείρο της κλειδαριάς K_1 .
4. Έπειτα το X απενεργοποιείται.
5. Τέλος, το Y ενεργοποιείται και απενεργοποιείται ξεκλειδώνοντας και τον δεύτερο πείρο K_2 και η πόρτα ανοίγει.

Εάν τα X και Y ενεργοποιηθούν με διαφορετική σειρά, το σύστημα επιστρέφει στην αρχική του κατάσταση όπου και οι δύο πείροι παραμένουν ασφαλισμένοι. Επίσης, επιστρέφει στην αρχική κατάσταση με το σήμα ενός αισθητήρα M ο οποίος δίνει σήμα την χρονική στιγμή που η πόρτα κλείνει.

Το γενικευμένο αυτόματο του περιγραφόμενου συστήματος αποτελείται από επτά καταστάσεις και απεικονίζεται στο σχήμα 7.18. Η κατάσταση q_0 αποτελεί την αρχική κατάσταση όπου βρίσκεται το σύστημα όταν τίθεται σε λειτουργία. Οι καταστάσεις από q_1 έως q_5 είναι ενδιάμεσες καταστάσεις πριν το σύστημα φτάσει στην κατάσταση q_6 όπου ανοίγει η πόρτα.

Σχήμα 7.18. Γενικευμένο αυτόματο ηλεκτρονικής κλειδαριάς.

Από το αυτόματο του παραπάνω σχήματος με χρήση των συστηματικών βημάτων του οδηγού υλοποίησης, παράγεται κώδικας για την γλώσσα Instruction List, ο οποίος παρουσιάζεται στο σχήμα 7.19. Η υλοποίηση περιλαμβάνει μια δομή Jump to List, ενώ το συνολικό πρόγραμμα είναι αρκετά απλό, καθώς το σύστημα έχει μόνο

διακριτές μεταβλητές και όχι συνεχείς που κατά κανόνα αυξάνουν την πολυπλοκότητα του συστήματος και του προγράμματος κατ' επέκταση.

L	"q"	ST2:	AN	"X"	ST5:	A	"X"
JL	LIST	L	L	"0"	L	L	"0"
JU	ST0	T	T	"q"	T	T	"q"
JU	ST1	JC	JC	COM	JC	JC	COM
JU	ST2	AN	AN	"Y"	AN	AN	"Y"
JU	ST3	L	L	"3"	L	L	"δ"
JU	ST4	T	T	"q"	T	T	"q"
JU	ST5	JU	JU	COM	JU	JU	COM
JU	ST6						
LIST:	JU	ST3:	S	"K1"	ST6:	S	"K2"
ST0:	R	A	A	"Y"	A	A	"M"
R	"K2"	L	L	"0"	L	L	"0"
A	"X"	T	T	"q"	T	T	"q"
L	"1"	JC	JC	COM	JC	JC	COM
T	"q"	AN	AN	"X"			
JU	COM	L	L	"4"	COM:	END	
		T	T	"q"			
ST1:	AN	JU	JU	COM			
L	"0"	ST4:	A	"X"			
T	"q"	L	L	"0"			
JC	COM	T	T	"q"			
A	"Y"	JC	JC	COM			
L	"2"	A	A	"Y"			
T	"q"	L	L	"5"			
JU	COM	T	T	"q"			
		JU	JU	COM			

Σχήμα 7.19. Υλοποίηση ηλεκτρονικής κλειδαριάς σε IL.

7.3 Εξομοίωση και έλεγχος με χρήση PLC και SCADA.

Στις παραγράφους 7.1 και 7.2 παρουσιάστηκαν δύο εργαλεία που μπορούν να φανούν χρήσιμοι οδηγοί του μηχανικού της βιομηχανίας για την υλοποίηση ελεγκτών που έχουν μοντελοποιηθεί με χρήση γενικευμένων αυτομάτων. Σε κάθε μια από τις περιπτώσεις, η πλατφόρμα υλοποίησης (SCADA και PLC) ήταν ιδιαίτερος γνώριμος στους ανθρώπους της βιομηχανίας. Βέβαια, η συνήθης πρακτική που ακολουθείται υπαγορεύει χρήση τόσο PLC για τον έλεγχο μιας διεργασίας, όσο και SCADA για την εποπτεία και παρακολούθηση του όλου συστήματος. Συνεπώς θα ήταν χρήσιμος ο συνδυασμός των δύο εργαλείων με ταυτόχρονη χρήση προγραμματιζόμενου λογικού ελεγκτή και σταθμού εποπτικού ελέγχου. Η νέα αυτή εκδοχή υλοποίησης ενός γενικευμένου αυτομάτου στην βιομηχανία παρουσιάζεται στο σχήμα 7.20. Το παρουσιαζόμενο σενάριο υπαγορεύει την υλοποίηση του ελεγκτή στο PLC με ταυτόχρονη εξομοίωση της λειτουργίας του αυτομάτου και παρακολούθησης της όλης διεργασίας στο SCADA. Η νέα αυτή τροποποιημένη εκδοχή συνδυάζει τα

πλεονεκτήματα χρήσης ενός PLC (χρόνος απόκρισης, διαθεσιμότητα, κτλ.) για τον έλεγχο μιας διεργασίας, ενώ παράλληλα υιοθετεί την γραφική απεικόνιση της χρονικής εξέλιξης του μοντέλου σε σύστημα SCADA που αποτελεί το σύγχρονο μέσο εποπτείας και οπτικής απεικόνισης ενός βιομηχανικού συστήματος.

Σχήμα 7.20. Ο ελεγκτής στο PLC.

Κεφάλαιο 8^ο

Εφαρμογές αυτομάτων χρήσης γενικευμένων

Με στόχο την ολοκληρωμένη παρουσίαση των Γενικευμένων Αυτομάτων, παραθέτονται τρία εκτενή παραδείγματα χρήσης τους. Το πρώτο παράδειγμα μοντελοποιεί μια μονοκύλινδρη μηχανή εσωτερικής καύσης η οποία έχει αποτελέσει αντικείμενο μελέτης από διάφορους ερευνητές του χώρου των υβριδικών συστημάτων. Στην συνέχεια παρουσιάζεται ένα ολοκληρωμένο σύστημα παραγωγής έτοιμου σκυροδέματος που αποτελεί ένα μικρού μεγέθους βιομηχανικό σύστημα. Τέλος, το τρίτο παράδειγμα είναι μια διάταξη τριών δεξαμενών που προσομοιώνει μια βιομηχανική διεργασία και έχει υλοποιηθεί στο εργαστήριο Γενικής Ηλεκτροτεχνίας του Πανεπιστημίου Πατρών.

8.1 Μηχανή εσωτερικής καύσης.

Η μηχανή εσωτερικής καύσης αποτελεί τον πλέον συνηθισμένο τύπο μηχανής ή γενικότερα πηγής κίνησης. Αναφορικά με το έργο ελέγχου, στόχος είναι η επίτευξη υψηλότερου ωφέλιμου έργου με την ελάχιστη κατανάλωση. Οι μηχανές εσωτερικής καύσης διακρίνονται σε δύο τύπους ανάλογα με τον τρόπο ανάφλεξης του μίγματος καυσίμου αέρα. Στον πρώτο τύπο η ανάφλεξη γίνεται με χρήση σπινθηριστή (μπουζί), ενώ στον δεύτερο τύπο πραγματοποιείται αυτανάφλεξη του μίγματος λόγω αύξησης της θερμοκρασίας κατά την συμπίεση. Ανάφλεξη με σπινθηριστή απαντάται στις μηχανές βενζίνης ή αερίου, οι οποίες ονομάζονται και μηχανές Otto από τον εφευρέτη τους. Αντίθετα, μηχανές με αυτανάφλεξη είναι οι μηχανές πετρελαίου ή μηχανές Diesel.

Σε κάθε περιστροφή του άξονα το έμβολο του κινητήρα κάνει δύο διαδρομές από το μέγιστο έως το ελάχιστο σημείο. Οι προαναφερθείσες μηχανές έχουν την δυνατότητα να λειτουργούν είτε με δύο διαδρομές-χρόνους του πιστονιού, είτε με τέσσερις. Στο συγκεκριμένο παράδειγμα θα αναφερθούμε σε μια μονοκύλινδρη τετράχρονη μηχανή εσωτερικής καύσης με σπινθηριστή. Μία τομή του κυλίνδρου της

υπό εξέτασης μηχανής φαίνεται στο σχήμα 8.1.

Σχήμα 8.1. Κύλινδρος μηχανής εσωτερικής καύσης.

Οι τέσσερις κύκλοι ή χρόνοι της λειτουργίας της μηχανής [86, 87] αναλύονται παρακάτω:

1) *Κύκλος εισαγωγής μίγματος:* Η βαλβίδα εισόδου ανοίγει και καθώς το πιστόνι διατρέχει τον κύλινδρο προς τα κάτω το μίγμα βενζίνης και αέρα εισέρχεται στον κύλινδρο.

2) *Κύκλος Συμπίεσης:* Οι δύο βαλβίδες παραμένουν κλειστές και το πιστόνι διατρέχει τον κύλινδρο προς τα πάνω συμπιέζοντας το μίγμα. Όταν το πιστόνι φτάσει στο ανώτερο σημείο πραγματοποιείται η ανάφλεξη του μίγματος.

3) *Κύκλος Έκτασης ή Λειτουργίας:* Η ανάφλεξη του μίγματος προκαλεί ακαριαία αύξηση της πίεσης και της θερμοκρασίας εντός του κυλίνδρου σπρώχνοντας το πιστόνι προς τα κάτω. Η κίνηση αυτή του πιστονιού αναγκάζει τον άξονα να περιστραφεί.

4) *Κύκλος Εξαγωγής:* Η βαλβίδα εξόδου ανοίγει και καθώς το πιστόνι οδεύει προς τα πάνω σπρώχνει τα υπολειπόμενα αέρια προς την έξοδο.

Μια μηχανή εσωτερικής καύσης ανεξαρτήτως τύπου καυσίμου και πλήθος χρόνων ακολουθεί τον μηχανικό κύκλο κίνησης του στροφαλοφόρου άξονα και όχι τον θερμοδυναμικό κύκλο καύσης του μίγματος. Όμως για την πληρέστερη ανάλυση του μοντέλου της μηχανής πρέπει να γίνει σύγκριση των διάφορων χρόνων με τον ιδανικό θερμοδυναμικό κύκλο (κύκλος Otto για την μηχανή του παραδείγματός). Οι

δύο αεροδυναμικοί κύκλοι που χαρακτηρίζουν την τετράχρονη μηχανή με σπινθηριστή [88, 89] φαίνονται στο σχήμα 8.2. Στον πρώτο κύκλο, το μίγμα αέρα-καυσίμου συμπιέζεται, αναφλέγεται και εκτονώνεται (compression-ignition-expansion), ενώ στον δεύτερο, τα καυσαέρια της καύσης οδηγούνται προς την εξάτμιση και ο κύλινδρος ξαναγεμίζει με νέα ποσότητα μίγματος (exhaust-induction).

Σχήμα 8.2. Διάγραμμα πίεσης-όγκου για τους δύο αεροδυναμικούς κύκλους.

Για την δημιουργία μοντέλου προσομοίωσης ήταν απαραίτητη η ύπαρξη κάποιων παραδοχών όσον αφορά τη φυσική σχέση των μεγεθών κατά τους δύο αεροδυναμικούς κύκλους. Οι παραδοχές που έγιναν δεν θα συζητηθούν περαιτέρω, καθώς κύριο μέλημα ήταν η δημιουργία ενός μοντέλου προσομοίωσης για την παρουσίαση στην πράξη του εργαλείου υλοποίησης μοντέλων εξομοίωσης και όχι η ανάλυση της λειτουργίας μιας μηχανής εσωτερικής καύσης. Το μαθηματικό μοντέλο που προέκυψε για κάθε χρόνο λειτουργίας απεικονίζεται στο σχήμα 8.3, όπου:

1. *Χρόνος Εισαγωγής:* τα περισσότερα μοντέλα [90] υποθέτουν ότι η εισαγωγή γίνεται υπό σταθερή πίεση, αλλά μια πιο ακριβής προσέγγιση θα ήταν $p = f(V^2)$.
2. *Χρόνος Συμπίεσης:* κατά τον χρόνο συμπίεσης [91], η σχέση πίεσης-όγκου κυλίνδρου δίνεται από την σχέση $p \cdot V^{1,3} = \text{σταθερό}$.
3. *Χρόνος Ισχύος:* θεωρούμε στιγμιαία [91] ανάφλεξη που προκαλεί απότομη αύξηση της πίεσης, ενώ η εκτόνωση δίνεται από την σχέση $p \cdot V^{1,48} = \text{σταθερό}$.
4. *Χρόνος Εξαγωγής:* ο χρόνος εξαγωγής ξεκινά με το άνοιγμα της βαλβίδας

εξόδου η οποία προκαλεί απότομη πτώση της πίεσης [88, 90], ενώ η εξαγωγή γίνεται υπό την σχέση $p = f(-V^2)$.

Σχήμα 8.3. Διάγραμμα πίεσης-όγκου στο μοντέλο προσομοίωσης.

Το μοντέλο μιας μηχανής βάσει των Γενικευμένων Αυτομάτων θα έχει τέσσερις καταστάσεις, μία για κάθε κύκλο λειτουργίας. Η προσέγγιση που χρησιμοποιήθηκε για την εξομοίωση παρουσιάζεται στο σχήμα 8.4. Η κάθε κατάσταση έχει ένα αντίστοιχο σύνολο παραμέτρων που δίνουν την κατάσταση των βαλβίδων, την έναυση του σπινθηριστή, και την διακύμανση της πίεσης και του όγκου του κυλίνδρου. Οι μεταβάσεις μεταξύ των καταστάσεων προκαλούνται από την θέση του πιστονιού στο μέγιστο και ελάχιστο σημείο της διαδρομής του μέσα στον κύλινδρο.

Σχήμα 8.4. Γενικευμένο αυτόματο μηχανής.

Το μοντέλο που περιγράφηκε χρησιμοποιήθηκε σε εξομοίωση στο λογισμικό πακέτο CX – Supervisor. Όλες οι παράμετροι των αυτομάτων αναπαραστάθηκαν με κώδικα που γράφηκε βάσει του οδηγού. Όλες οι μεταβλητές του συστήματος δίνονται ως συναρτήσεις της γωνίας περιστροφής του στροφαλοφόρου άξονα (σχήμα 8.5). Η μετατόπιση του πιστονιού από το μέγιστο σημείο της διαδρομής του είναι ανάλογη του όγκου του κυλίνδρου και δίνεται από τον τύπο:

$$s(a) = r \left(1 - \cos a + \frac{l}{r} \left(1 - \sqrt{1 - \frac{r^2}{l^2} \sin^2 a} \right) \right)$$

Όπου a είναι η γωνία του άξονα, r η ακτίνα του στροφαλοφόρου άξονα και l το μήκος διασύνδεσης. Στο μέγιστο σημείο του κυλίνδρου TDC έχουμε $a = 0$, $s(a) = 0$ και $V = \min$, ενώ στο ελάχιστο σημείο BDC, $a = \pi$, $s(a) = 2 \cdot r$ και $V = \max$ αντίστοιχα.

Ο κώδικας που δημιουργήθηκε για την κατάσταση συμπίεσης βάσει του οδηγού υλοποίησης, είναι:

```
CASE 2
  Vin=0
  Vout=0
  Spark=0
  V=K*r(1-cos(a)+(l/r)*(1-sqrt(1-r^2*(sin(a))^2/l^2))
  P=K2/V
  a=a+pi/180
  if V==Vmin then
    q=3
    P=P+30
  endif
```

Για την ολοκλήρωση του εξομοιωτή με εικονική αναπαράσταση του συστήματος, δημιουργήθηκαν δύο οθόνες, μία για την κινούμενη εικόνα της κίνησης του πιστονιού και η άλλη με γραφικές παραστάσεις όλων των μεταβλητών του συστήματος.

Η πρώτη οθόνη παρουσιάζεται στο σχήμα 8.6, ενώ στο σχήμα 8.7 παρουσιάζονται οι μεταβλητές του συστήματος, μετατόπιση, ταχύτητα και επιτάχυνση πιστονιού.

Σχήμα 8.5. Φυσικά μεγέθη κυλίνδρου

Σχήμα 8.6. Αρχική οθόνη SCADA.

Σχήμα 8.7. Οθόνη διαγραμμάτων.

8.2 Μονάδα παραγωγής έτοιμου σκυροδέματος.

Το σύστημα έτοιμου σκυροδέματος [92] που μελετήθηκε απεικονίζεται στο σχήμα 8.8. Η κύρια λειτουργία του είναι η ακριβής ζύγιση των συστατικών του μίγματος και η εισαγωγή αυτών στον αναμίκτη για μια πλήρη παραγωγή έτοιμου σκυροδέματος. Όλα τα προς ανάμιξη υλικά, αδρανή υλικά, τσιμέντο, χημικά πρόσθετα και νερό, ζυγίζονται ξεχωριστά στους αντίστοιχους συλλέκτες υλικού. Το σύστημα της εφαρμογής μας έχει πέντε διαφορετικούς τύπους αδρανών υλικών που ο καθένας είναι αποθηκευμένος σε διαφορετικό κάδο. Τα διαφορετικά είδη τσιμέντου είναι τρία, τοποθετημένα στα αντίστοιχα σιλό. Τα πρόσθετα υλικά, που είναι σε υγρή μορφή, υπάρχουν σε τρεις μορφές αποθηκευμένα σε τρεις δεξαμενές αντίστοιχα. Η ποσότητα του νερού που προστίθεται μετράται με ένα κατάλληλο μετρητικό σύστημα. Πρέπει να σημειωθεί ότι οι συλλέκτες υλικού έχουν μόνο έναν ζυγό ακριβείας και συνεπώς κάθε διαφορετικό υλικό προστίθεται ένα-ένα ώστε να ζυγίζεται ξεχωριστά. Για την ανάμιξη υπάρχει ένας αναμίκτης διπλού άξονα ο οποίος παρέχει την ασφάλεια ότι το μίγμα θα έχει ανακατευτεί πλήρως στο τέλος της παραγωγής. Στο τέλος της

παραγωγής το παραγόμενο σκυρόδεμα φορτώνεται σε ειδικά οχήματα-μπετονιέρες για την μεταφορά στον χώρο παράδοσης.

Ο πιο σημαντικός παράγοντας όσον αφορά τις μηχανικές και χημικές ιδιότητες του σκυροδέματος είναι η αναλογία των υλικών του μίγματος. Επιπρόσθετα, παράμετροι όπως η ιεραρχία εισαγωγής των υλικών στον αναμικτή και ο χρόνος ανάμιξης πρέπει να ληφθούν υπόψη για την διασφάλιση της ποιότητας του παραγόμενου υλικού. Όλες αυτές οι παράμετροι καθορίζονται επακριβώς στο αντίστοιχο φύλλο συνταγής που χρησιμοποιείται.

Σχήμα 8.8. Σχηματικό διάγραμμα του συστήματος

Ένα διάγραμμα ροής της όλης διεργασίας απεικονίζεται στο σχήμα 8.9, όπου όλες οι ενέργειες ζύγισης και ανάμιξης φαίνονται αναλυτικά. Κατά την έναρξη της παραγωγής, ο χειριστής πρέπει να επιλέξει την κατάλληλη συνταγή και την επιθυμητή ποσότητα παραγωγής, μέσω της γραφικής διεπαφής του συστήματος SCADA που ολοκληρώνει το σύστημα. Στην συνέχεια ξεκινά η ζύγιση όλων των υλικών και μετά την ολοκλήρωσή της αρχίζει η εισαγωγή των υλικών στον αναμίκτη και η ανάμιξη αυτών. Όταν ο χρόνος ανάμιξης ολοκληρωθεί ανοίγει η καταπακτή εξόδου του αναμίκτη και το σκυρόδεμα φορτώνεται στο όχημα μεταφοράς. Όταν ο αναμίκτης αδειάσει τελείως, το σύστημα είναι έτοιμο να δεχτεί την νέα παραγγελία για έναρξη καινούργιας παραγωγής.

Σχήμα 8.9. Διάγραμμα ροής του συστήματος.

8.2.1 Μοντελοποίηση και έλεγχος με χρήση Γενικευμένων Αυτομάτων

Στην βιβλιογραφία υπάρχει πλήθος διαφορετικών προσεγγίσεων στην μοντελοποίηση αντίστοιχων μονάδων [93, 94, 95]. Η μοντελοποίηση που χρησιμοποιήθηκε στο παρόν έργο περιλάμβανε την χρήση πολλαπλών αυτομάτων που το καθένα θα αναπαριστούσε συγκεκριμένο μέρος του συστήματος. Καταρχήν ένα αυτόματο που θα περιέγραφε τον μηχανισμό επιλογής συνταγών ήταν απαραίτητο. Το αυτόματο αυτό παίζει το ρόλο του επόπτη της όλης διεργασίας τροφοδοτώντας όλα τα υπόλοιπα με τις τιμές των μεταβλητών που απαιτούνται. Στο σχήμα 8.10 φαίνεται το αυτόματο με q_0 την αρχική του κατάσταση και μία κατάσταση για κάθε διαθέσιμη συνταγή. Κάθε συνταγή δίνει τιμές σε όλες τις μεταβλητές που είναι χρήσιμες για την όλη διεργασία. Αρχικά οι αναλογίες όλων των υλικών πρέπει να ορισθούν, $B_{agg} = \{b_{agg_1}, b_{agg_2}, b_{agg_3}, b_{agg_4}, b_{agg_5}\}$, $B_{cem} = \{b_{cem_1}, b_{cem_2}, b_{cem_3}\}$, $B_w = b_{water}$ και $B_{adm} = \{b_{adm_1}, b_{adm_2}, b_{adm_3}\}$. Επίσης η ιεραρχία εισαγωγής των υλικών στον αναμίκτη $T_{in} = \{t_{agg}, t_{adm}, t_{cem}, t_w\}$ και τέλος ο χρόνος ανάμιξης t_{mix} .

Σχήμα 8.10. Αυτόματο-Συνταγής

Έπειτα, κατασκευάστηκε ένα αυτόματο για την ζύγιση κάθε υλικού και συνεπώς χρειάστηκαν τρία αυτόματα για την ζύγιση των αδρανών υλικών, των τσιμέντων και των πρόσθετων. Τα αυτόματα αυτά απεικονίζονται στα σχήματα 8.11, 8.12 και 8.13 αντίστοιχα.

Σχήμα 8.11. Αυτόματο ελέγχου αδρανών υλικών.

Σχήμα 8.12. Αυτόματο ελέγχου τιμών.

Σχήμα 8.13. Αυτόματο ελέγχου πρόσθετων υλικών.

Τα τρία αυτόματα ζύγισης είναι παρόμοια καθώς η κύρια λειτουργία τους είναι η τήρηση των δοθέντων ποσοτήτων για κάθε υλικό. Η σημειογραφία σε κάθε σχήμα πρέπει να ξεκαθαριστεί, αλλά λόγω ομοιότητας θα δοθεί μόνο για το σχήμα 8.11. Καταρχήν, η κατάσταση q_0 είναι η αρχική κατάσταση ηρεμίας όπου όλες οι πόρτες εξόδου είναι κλειστές (μεταβλητές *aggr_i*). Κάθε κατάσταση q_i , $i=1, \dots, 5$ συνεπάγεται το άνοιγμα της i^{th} πόρτας. Οι μεταβάσεις από την αρχική κατάσταση με τίτλο s_i

προκαλούνται από το γεγονός $start \cdot (b_{agg_1} > 0)$ και επιφέρουν μετάβαση στην κατάσταση q_i . Αντίστοιχα, οι μεταβάσεις ag_{ji} , $i=1, \dots, 4$ και $j=2, \dots, 5$ προκαλούνται από τα γεγονότα $(B_{agg_tot} = b_{agg_1} + \dots + b_{agg_i}) \cdot (b_{agg_{i+1}} = 0, \dots, b_{agg_{j-1}} = 0) \cdot (b_{agg_j} > 0)$ και επιφέρουν την μετάβαση από την q_i στην q_j κατάσταση. Παραδείγματος χάρη η μετάβαση ag_{25} προκαλείται από το $(B_{agg_tot} = b_{agg_1} + b_{agg_2}) \cdot (b_{agg_3} = 0, b_{agg_4} = 0) \cdot (b_{agg_5} > 0)$ γεγονός. Η μετάβαση αυτή πραγματοποιείται μόνο όταν η συνταγή ορίζει ως μηδενικές τις αναλογίες των αδρανών 3 και 4. Οι μεταβάσεις ag_{i0} προκαλούνται από τα γεγονότα $(B_{agg_tot} = b_{agg_1} + \dots + b_{agg_i}) \cdot (b_{agg_{i+1}} = 0, \dots, b_{agg_{j-1}} = 0)$. Αυτές έχουν επίσης ένα πίνακα αρχικοποίησης [$agg_ok=1$] ως επιβεβαίωση ότι η διαδικασία ζύγισης των αδρανών υλικών έχει ολοκληρωθεί επιτυχώς. Τέλος, όσον αφορά την ιεραρχία των γεγονότων σε κάθε κατάσταση, έχει ως εξής: (q_0 : s_1, s_2, s_3, s_4, s_5), (q_1 : $ag_{12}, ag_{13}, ag_{14}, ag_{15}, ag_{10}, v_{agg}$), (q_2 : $ag_{23}, ag_{24}, ag_{25}, ag_{20}, v_{agg}$), (q_3 : $ag_{34}, ag_{35}, ag_{30}, v_{agg}$), (q_4 : $ag_{45}, ag_{40}, v_{agg}$) και (q_5 : ag_{50}, v_{agg}). Επιπρόσθετα, ο ρυθμός ροής κάθε υλικού υπολογίζεται και αν είναι χαμηλός ένας δονητής ενεργοποιείται για μερικά δευτερόλεπτα (μεταβάσεις βρόχου v_{agg}).

Σχήμα 8.14. Αυτόματο ελέγχου αναμίκτη.

Τέλος ένα αυτόματο που θα ελέγχει την λειτουργία του αναμίκτη ήταν απαραίτητο. Το αυτόματο απεικονίζεται στο σχήμα 8.14 και η κύρια λειτουργία του είναι η

τροφοδότηση του αναμίκτη με τα υλικά και ο έλεγχος της ανάμιξης. Η κατάσταση q_0 είναι η αρχική κατάσταση και μετά το επιτυχές ζύγισμα όλων των υλικών (σήμα $agg_ok \cdot adm_ok \cdot cem_ok$) πραγματοποιείται μετάβαση στην κατάσταση q_1 . Όταν η διαδικασία ανάμιξης ολοκληρωθεί ξεκινά η εκκένωση του αναμίκτη και η φόρτωση του έτοιμου σκυροδέματος στο όχημα.

8.2.2 Σταθμός εποπτικού ελέγχου και συλλογής δεδομένων

Ο ελεγκτής του συστήματος όπως περιγράφηκε παραπάνω υλοποιήθηκε σε CX-Supervisor, ένα λογισμικό SCADA της εταιρείας OMRON Corp. Η κύρια οθόνη του σταθμού απεικονίζεται στο σχήμα 8.15 και δείχνει με κινούμενα γραφικά την όλη διεργασία. Έκτος όμως της κεντρικής σελίδας, κατασκευάστηκαν και άλλες που αναπαριστούν την εκτέλεση των αυτομάτων του συστήματος. Ένα παράδειγμα φαίνεται στο σχήμα 8.16 και δείχνει το αυτόματο περιγραφής των χημικών πρόσθετων. Η ενεργή κατάσταση σημειώνεται με ένα κόκκινο τετράγωνο, καθώς και η τελευταία μετάβαση που πραγματοποιήθηκε ώστε να υπάρχει καλύτερη εποπτεία της όλης λειτουργίας.

Σχήμα 8.15. Η αρχική οθόνη του SCADA.

Ο σταθμός ολοκληρώνεται με μια βάση δεδομένων για την αποθήκευση όλων των δεδομένων. Συγκεκριμένα για κάθε παραγωγή αποθηκεύονται οι ποσότητες του κάθε υλικού, θεωρητικές και μετρούμενες τιμές, όνομα πελάτη, ημερομηνία και ώρα, αριθμός πινακίδας οχήματος παράδοσης, συνολικός όγκος σκυροδέματος κ.α. Με χρήση των δυνατοτήτων της βάσης μπορούν να εξαχθούν τυπωμένες αναφορές με στατιστικά στοιχεία κίνησης συγκεκριμένης ημέρας ή περιόδου.

Σχήμα 8.16. Εκτέλεση του αυτομάτου των πρόσθετων.

8.3 Εργαστηριακή διάταξη τριών δεξαμενών.

Το τελευταίο παράδειγμα χρήσης των γενικευμένων αυτομάτων έχει υλοποιηθεί στο εργαστήριο Γενικής Ηλεκτροτεχνίας του Πανεπιστημίου Πατρών. Η εργαστηριακή διάταξη αποτελείται από τρεις ανεξάρτητες δεξαμενές, όπου το σενάριο λειτουργίας είναι η πλήρωση της κεντρικής-κύριας δεξαμενής από τις δύο άλλες, βάσει δεδομένων κριτηρίων αναλογίας. Οι δύο ακραίες δεξαμενές περιέχουν διαλύματα διαφορετικής χημικής σύνθεσης και τροφοδοτούν την κύρια δεξαμενή όπου πραγματοποιείται η ανάμιξη των δύο διαλυμάτων. Τον έλεγχο του συστήματος υλοποιεί ένας

προγραμματιζόμενος λογικός ελεγκτής S7-300 της εταιρίας Siemens, ο οποίος συλλέγει δεδομένα του συστήματος από τους αντίστοιχους αισθητήρες και εφαρμόζει την στρατηγική ελέγχου στις αντλίες και τις αναλογικά ρυθμιζόμενες βαλβίδες. Μια σχηματική απεικόνιση δίνεται στο σχήμα 8.17, ενώ στο σχήμα 8.18 φαίνεται μια φωτογραφία της όλης διάταξης (δεξαμενές, βαλβίδες, αισθητήρες, PLC).

Σχήμα 8.17. Σχηματική απεικόνιση διάταξης.

Σχήμα 8.18. Φωτογραφία εργαστηριακής διάταξης.

Η εργαστηριακή διάταξη προσομοιώνει την διεργασία παραγωγής χρωμάτων, όπου γίνεται ανάμιξη διαφορετικού χρώματος υγρών με στόχο την παραγωγή της επιθυμητής απόχρωσης. Το χρώμα κάθε υγρού μετράται από αντίστοιχους χρωματικούς αισθητήρες (cs_1 , cs_2 , cs_3) που είναι τοποθετημένοι στο κάτω μέρος κάθε δεξαμενής. Η όλη διεργασία ξεκινά με το πάτημα του πλήκτρου *Start*, όπου οι on/off βαλβίδες v_1 και v_2 ανοίγουν, οι αντλίες p_1 και p_2 τίθενται σε λειτουργία και οι αναλογικές βαλβίδες a_{v_1} και a_{v_2} ανοίγουν σύμφωνα με την στρατηγική ελέγχου και υγρά από τις δύο ακραίες δεξαμενές τροφοδοτούν την κεντρική. Παράλληλα ξεκινά και η διαδικασία ανάμιξης με έναυση του αναμίκτη, *mix*. Όταν το διάλυμα της κεντρικής δεξαμενής ανέλθει στο 70% της συνολικής χωρητικότητας, όλες οι βαλβίδες και οι αντλίες κλείνουν. Εάν δεν έχει επιτευχθεί η επιθυμητή απόχρωση προστίθεται υγρό από μια εκ των ακραίων δεξαμενών, ενώ αν το χρώμα είναι αποδεκτό, η διεργασία ανάμιξης ολοκληρώνεται, ο αναμίκτης σβήνει και η κεντρική δεξαμενή αδειάζει με άνοιγμα της βαλβίδας v_3 και έναυση της αντλίας p_3 .

Αυτό το σύστημα των τριών δεξαμενών μπορεί να εξεταστεί ως μια ολότητα, ή σε μια διαφορετική προσέγγιση, ως τρία ξεχωριστά υποσυστήματα. Το κάθε υποσύστημα μοντελοποιεί μια δεξαμενή με χρήση γενικευμένων αυτομάτων. Τα τρία αυτόματα που σχεδιάστηκαν για κάθε υποσύστημα παρουσιάζονται στα σχήματα 8.19, 8.20 και 8.21.

Σχήμα 8.19. Μοντέλο αυτομάτου της δεξαμενής 1.

Σχήμα 8.20. Μοντέλο αυτομάτου της δεξαμενής 2.

Σχήμα 8.21. Μοντέλο αυτομάτου της δεξαμενής 3.

Τα πρώτα δύο αυτόματα είναι σχετικά απλά ελέγχοντας την εισροή υγρού στις δύο ακραίες δεξαμενές in_1 και in_2 συναρτήσεως του ύψους του υγρού στις δεξαμενές. Το τρίτο αυτόματο είναι σχετικά περιπλοκότερο, καθώς εμπεριέχει την στρατηγική ελέγχου του όλου συστήματος. Αποτελείται από έξι λειτουργικές καταστάσεις, μια για κάθε φάση της διεργασίας. Επιπρόσθετα, υπάρχουν δύο καταστάσεις *ALARM* και *STOP*, όπου το σύστημα οδηγείται με το πάτημα των αντιστοίχων πλήκτρων. Στην κατάσταση *STOP*, σταματά η όλη διεργασία και κλείνουν όλες οι βαλβίδες και αντλίες. Εάν πατηθεί το πλήκτρο *start*, τότε το σύστημα επανέρχεται στην πρότερη κατάσταση και η διεργασία συνεχίζεται. Αντίθετα η κατάσταση *ALARM* σταματά την όλη διεργασία, αδειάζει την κεντρική δεξαμενή και επαναφέρει το σύστημα στην αρχική του κατάσταση Q_1 . Για λόγους ευκρίνειας, οι μεταβάσεις από και προς την κατάσταση *ALARM* δίνονται με σπικτικές γραμμές, ενώ οι μεταβάσεις από και προς την κατάσταση *STOP* δίνονται με διακεκομμένες γραμμές.

Η όλη διεργασία έχει ως εξής. Αρχικά το σύστημα βρίσκεται στην κατάσταση Q_1 και με το πάτημα του πλήκτρου *start* πυροδοτείται η μετάβαση στην κατάσταση Q_2 . Στην κατάσταση αυτή, ξεκινά η διαδικασία ανάμιξης καθώς τροφοδοτείται η κεντρική δεξαμενή από τις δύο ακραίες ως το ανώτερο όριο περιεκτικότητας, όπου το σύστημα μεταφέρεται στην λειτουργική κατάσταση Q_3 . Αυτή αποτελεί μια ενδιάμεση κατάσταση όπου μετράται η ακρίβεια του ελεγκτή ανάμιξης. Εάν η απόχρωση του διαλύματος είναι σύμφωνη με τις προδιαγραφές, πυροδοτείται η μετάβαση προς της κατάσταση Q_4 . Αν όμως το διάλυμα είναι εκτός προδιαγραφών, τότε πραγματοποιείται μετάβαση προς τις καταστάσεις Q_{31} ή Q_{32} , όπου προστίθεται υγρό από τις δεξαμενές 1 ή 2, αντιστοίχως. Έπειτα, το σύστημα οδηγείται στην τελική κατάσταση Q_4 , όπου ολοκληρώνεται η διεργασία ανάμιξης και το παραγόμενο διάλυμα οδεύει προς συσκευασία. Όταν η κεντρική δεξαμενή αδειάσει πλήρως, η όλη διεργασία λαμβάνει τέλος και το σύστημα επανέρχεται στην αρχική του κατάσταση.

Ο έλεγχος της όλης διεργασίας επιτυγχάνεται με ένα προγραμματιζόμενο λογικό ελεγκτή στον οποίο είναι συνδεδεμένοι όλοι οι χρωματικοί αισθητήρες και αισθητήρες στάθμης, καθώς και όλες οι αντλίες και βαλβίδες ελέγχου. Ο ελεγκτής της διάταξης προγραμματίστηκε σε γλώσσα STL βάσει του εργαλείου παραγωγής κώδικα για PLC από μοντέλο γενικευμένων αυτομάτων. Ένα τμήμα του κώδικα που δημιουργήθηκε παρουσιάζεται στο σχήμα 8.22 και αναπαριστά την κατάσταση Q_{31} . Ο έλεγχος στην συγκεκριμένη κατάσταση επιτυγχάνεται μέσω ενός PI ελεγκτή, ο οποίος υλοποιήθηκε με χρήση της ενσωματωμένης συνάρτησης του PLC.

<pre>// Function Q31 S "V2" S "P2" S "MIX" R "V1" R "V3" R "P1" R "P3" L 0.0 T "A_V1" CALL "CONT_C" , DB42 COM_RST :=FALSE MAN_ON :=M123.1 PVPER_ON:=TRUE P_SEL :=TRUE I_SEL :=TRUE INT_HOLD:=TRUE I_ITL_ON:=FALSE D_SEL :=FALSE CYCLE :=T#10MS SP_INT :=MD135 PV_IN :=MD141</pre>	<pre>PV_PER :=MW220 MAN :=MD240 GAIN :=MD260 TI :=T#20S TD :=T#10S TM_LAG :=T#2S DEADB_W :=0.000000e+000 LMN_HLM :=1.000000e+002 LMN_LLM :=0.000000e+000 PV_FAC :=1.000000e+000 PV_OFF :=0.000000e+000 LMN_FAC :=1.000000e+000 LMN_OFF :=0.000000e+000 I_ITLVAL :=0.000000e+000 DISV :=0.000000e+000 LMN :=MD300 LMN_PER :=MW320 QLMN_HLM:=M121.1 QLMN_LLM:=M121.2 LMN_P :=MD324 LMN_I :=MD328 LMN_D :=MD332 PV :=MD336 ER :=MD340</pre>	<pre>L MD400 //cs3 L MD440 //cs3_opt*0.95 >=R = M380.1 L MD400 //cs3 L MD460 //cs3_opt*1.05 <=R = M380.2 A M380.1 A M380.2 L "4" T "Q" A "STOP" L "5" T "Q" A "ALARM" L "6" T "Q" END</pre>
---	--	--

Σχήμα 8.22. STL κώδικας για την κατάσταση Q_{31} .

Κεφάλαιο 9^ο

Επίλογος – Εργασίες που προέκυψαν από το έργο

9.1 Επίλογος – Μελλοντική εργασία.

Στα μελλοντικά σχέδια περιλαμβάνεται η επέκταση του ορισμού των αυτομάτων και η εισαγωγή της έννοιας των σύγχρονων καταστάσεων, δηλαδή καταστάσεων στις οποίες το αυτόματο θα βρίσκεται συγχρόνως, ανατρέποντας έτσι την ισχύουσα θεωρία ότι κάθε αυτόματο βρίσκεται σε κάθε χρονική στιγμή σε μία και μόνο μία κατάσταση. Η ανάγκη αυτής της επέκτασης βασίζεται στο γεγονός ότι τα γενικευμένα αυτόματα έχουν καθαρά βιομηχανικό προσανατολισμό, όπου συχνή είναι η μοντελοποίηση καθενός τμήματος μιας παραγωγικής διαδικασίας με μία κατάσταση του αυτομάτου σχηματίζοντας το συνολικό αυτόματο. Όμως, η ορθή λειτουργία ενός βιομηχανικού συστήματος απαιτεί όλα τα τμήματα της γραμμής παραγωγής να δουλεύουν συγχρόνως, γεγονός που οδηγεί στην προαναφερόμενη επέκταση του ορισμού και χρήση των σύγχρονων καταστάσεων. Η ιδιότητα του ταυτοχρονισμού (concurrency) καλύπτεται στην παρούσα μορφή των γενικευμένων αυτομάτων με χρήση περισσοτέρων του ενός αυτομάτων για την μοντελοποίηση ενός συστήματος και ιεραρχική δομή του μοντέλου. Η υιοθέτηση σύγχρονων καταστάσεων θα μπορούσε να αποτρέψει την ανάγκη μοντέλου πολλαπλών αυτομάτων με στόχο την απλούστευση του τελικού μοντέλου.

Παράλληλα, λόγω του πρακτικού προβλήματος στην σχεδίαση αυτομάτων με μεγάλο αριθμό καταστάσεων, βρίσκεται σε εξέλιξη η προσπάθεια ανατροπής του κλασικού διαγράμματος καταστάσεων και η αντικατάστασή του με μία νέα μορφή διαγράμματος που όσο και αν μεγαλώνει ο αριθμός καταστάσεων να μην οδηγεί σε ένα δαιδαλώδες γράφημα.

Τέλος, θα ήταν ιδιαίτερα χρήσιμη η δημιουργία ενός εμπορικού λογισμικού πακέτου που ακολουθώντας τα περιγραφόμενα βήματα μετατροπής ενός αυτομάτου σε εκτελέσιμο κώδικα για PLC να τον παράγει αυτόματα για τις γλώσσες που περιλαμβάνει το πρότυπο IEC 61131-3. Συνεπώς το έργο του μηχανικού ελέγχου θα περιοριζόταν δραστικά, με κύριο αντικείμενο την ορθή δημιουργία του αυτομάτου ελέγχου της όλης διεργασίας.

9.2 Εργασίες που προέκυψαν από το έργο.

Κατά την εκτέλεση του έργου προέκυψε πλήθος ερευνητικών εργασιών, οι οποίες παρουσιάστηκαν σε διεθνή συνέδρια ή δημοσιεύτηκαν σε έγκριτα διεθνή επιστημονικά περιοδικά.

Παρακάτω δίνονται οι τίτλοι των εργασιών με τα πλήρη στοιχεία δημοσίευσης:

▪ Δημοσιεύσεις σε Διεθνή Επιστημονικά Περιοδικά.

1. **Deligiannis, V.**, S. Manesis and J. Lygeros, (2008). *“Global Automata: a new formal method for modeling industrial systems”*, International Journal of Industrial and Systems Engineering.
2. **Deligiannis, V.** and S. Manesis, (2008). *“Concrete batching and mixing plants: A new modelling and control approach based on global automata”*, International Journal of Automation in Construction.
3. **Deligiannis, V.** and S. Manesis, (2010). *“Implementing Global Automata in PLCs based on the IEC 61131-3 Programming Norm”*, International Journal of Industrial and Systems Engineering.
4. **Deligiannis, V.** and S. Manesis, *“Building Simulation Models Based on Global Automata for Undergraduate Courses on Industrial Systems Analysis”*, International Journal of Engineering Education, έχει υποβληθεί για δημοσίευση.
5. **Deligiannis, V.** and S. Manesis, *“Formal Analysis of Global Automata’s Structural Properties”*, Simulation Modeling Practice and Theory, έχει υποβληθεί για δημοσίευση.

▪ Δημοσιεύσεις σε Διεθνή Επιστημονικά Συνέδρια

6. **Deligiannis, V.** and S. Manesis, (2004). *"On Automata and Industrial Applications with Programmable Logic Controllers: theory and tools"*, IEEE 12th Mediterranean Conference on Control and Automation MED'04.
7. **Deligiannis, V.** and S. Manesis, (2005). *"A new formulation approach for modelling Discrete Event Systems"*, 1st IC-EpsMsO Conference.
8. **Deligiannis, V.** and S. Manesis, (2005). *"Introducing a New Mathematical abstraction for Modeling Real Time Systems"*, IFAC 2005, 16th World Congress of International Federation of Automatic Control.
9. **Deligiannis, V.** and S. Manesis, (2005). *"Automata Composition for Modeling Large Industrial Systems"*, CIMCA 2005, International Conference on Computational Intelligence for Modelling Control and Automation.
10. **Deligiannis, V.** and S. Manesis, (2006). *"Building Automata-Type Simulation Models for Undergraduate Teaching Industrial Systems Analysis"*, ACE '06, 7th IFAC Symposium on Advances in Control Education.
11. **Deligiannis, V.** and S. Manesis, (2006). *"Automata Based Modeling and Simulation: Application in an industrial software environment"*, ICINCO 2006.
12. **Deligiannis, V.** and S. Manesis, (2006). *"Modeling Internal Combustion Engines Using a Hyper-Class of Hybrid Automata: A Case Study"*, CACSD, IEEE Computer Aided Control Systems Design Conference.
13. **Deligiannis, V.** and S. Manesis, (2006). *"Automata Based Modelling for Batching and Mixing Plants for Concrete"*, Proceedings of the 11th IEEE ETFA 2006.
14. **Deligiannis, V.** and S. Manesis, (2007). *"Two-Mode Formulation of Automata Based Simulation and Control of Industrial Systems"*, ICMSAO '07.
15. **Deligiannis, V.** and S. Manesis, (2007). *"A Survey on Automata-based Methods for Modelling and Simulation of Industrial Systems"*, Proceedings of the 12th IEEE ETFA 2007.

Κεφάλαιο 10^ο

Βιβλιογραφία

- [1] Deligiannis, V. and S. Manesis, (2005). *"Introducing a New Mathematical abstraction for Modeling Real Time Systems"*, IFAC 2005, 16th World Congress of International Federation of Automatic Control
- [2] Deligiannis, V., S. Manesis and J. Lygeros, (2008). *"Global Automata: a new formal method for modeling industrial systems"*, International Journal of Industrial and Systems Engineering.
- [3] Deligiannis, V. and S. Manesis, (2005). *"Automata Composition for Modeling Large Industrial Systems"*, CIMCA 2005, International Conference on Computational Intelligence for Modelling Control and Automation.
- [4] Deligiannis, V. and S. Manesis, *"Formal Analysis of Global Automata's Structural Properties"*, Simulation Modeling Practice and Theory, έχει υποβληθεί για δημοσίευση.
- [5] Deligiannis, V. and S. Manesis, (2005). *"A new formulation approach for modelling Discrete Event Systems"*, 1st IC-EpsMsO Conference.
- [6] Deligiannis, V. and S. Manesis, (2006). *"Modeling Internal Combustion Engines Using a Hyper-Class of Hybrid Automata: A Case Study"*, CACSD, IEEE Computer Aided Control Systems Design Conference.
- [7] Deligiannis, V. and S. Manesis, (2006). *"Automata Based Modeling and Simulation: Application in an industrial software environment"*, ICINCO 2006.
- [8] Deligiannis, V. and S. Manesis, (2006). *"Building Automata-Type Simulation Models for Undergraduate Teaching Industrial Systems Analysis"*, ACE '06, 7th IFAC Symposium on Advances in Control Education.
- [9] Deligiannis, V. and S. Manesis, *"Building Simulation Models Based on Global Automata for Undergraduate Courses on Industrial Systems Analysis"*, International Journal of Engineering Education, έχει υποβληθεί για δημοσίευση.
- [10] Deligiannis, V. and S. Manesis, (2008). *"Concrete batching and mixing plants: A new modelling and control approach based on global automata"*, International Journal of Automation in Construction.

- [11] Deligiannis, V. and S. Manesis, (2010). *"Implementing Global Automata in PLCs based on the IEC 61131-3 Programming Norm"*, International Journal of Industrial and Systems Engineering.
- [12] Wonham, W.M., (2000) "Supervisory Control of Discrete-Event Systems: An Introduction", *Proc. IEEE Intl. Conf. on Industrial Technology 2000 (ICIT2000)*, Goa, India, 474-479, January 19-22, 2000.
- [13] King, R.E., (2004) *Βιομηχανική Πληροφορική*, εκδόσεις Τζιόλα.
- [14] Μάνεσης, Σ., (2009) *Συστήματα Προγραμματιζόμενων Βιομηχανικών Αυτοματισμών*.
- [15] Μαραντίδης, Ν., (2000) *Αυτοματισμός με SIMATIC S7*, Siemens.
- [16] Ramadge, P. and W.M. Wonham, (1989) "The Control of Discrete Event Systems", *Proceedings of the IEEE*, Vol.77, No.1, pp.81-98, 1989.
- [17] Antsaklis, P. J. and X. D. Koutsoukos, (2003) "Hybrid Systems: Review and Recent Progress", Chapter in *Software-Enabled Control: Information Technologies for Dynamical Systems*, IEEE Press
- [18] Lee, E. A., and H. Zheng, (2005) 'Operational Semantics of Hybrid Systems', Invited paper in *Hybrid Systems: Computation and Control: 8th International Workshop, HSCC*, LNCS 3414, Zurich, Switzerland, March, 9-11.
- [19] Kleijn, J. J. T., Reniers, M. A. and Rooda, J. E. (2003) 'Analysis of an Industrial System', *Formal Methods in System Design*, 22, 249-282.
- [20] Lauzon, S. C., Mills, J. K. and Benhabib, B. (1997) 'An Implementation Methodology for the Supervisory Control of Flexible Manufacturing Workcells', *SME J. of Manufacturing Systems*, Vol. 16, No.1, pp.91-101.
- [21] Cassandras, C.G. (1993) *Discrete Event Systems: Modelling and Performance Analysis*, Richard D. Irwin Inc., Boston (MA).
- [22] International Electrotechnical Commission (2005) "IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems".
- [23] International Electrotechnical Commission (2004) "IEC 61511 Functional safety - Safety instrumented systems for the process industry sector".
- [24] Peterson, J.L. (1981) *Petri Net Theory and the Modelling of Systems*, Prentice-Hall Inc.
- [25] Dierks, H. (1997) 'PLC-Automata: A New Class of Implementable Real-Time Automata', ARTS'97, LNCS, Springer Verlag.
- [26] Alur, R. and Dill, D.L. (1994) 'A Theory of Timed Automata', *Theoretical Computer Science*, 126:183-235.

- [27] Alur, R., Courcoubetis, C., Henzinger, T.A. and Ho, P-H. (1993) 'Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems', *Hybrid Systems*, Grossman, Nerode and Ravn, Lecture Notes in Computer Science, Vol. 736.
- [28] Khoussainov, B. and Nerode, A. (2001) *Automata theory and its applications*, Birkhauser, Boston.
- [29] Lygeros, J., Johansson, K. H., Simic, S. N., Zhang, J. and Sastry, S. S. (2003) 'Dynamical Properties of Hybrid Automata', *IEEE Transactions on Automatic Control*, vol. 48, No 1.
- [30] Antsaklis, P.J. (2000) 'A Brief Introduction to the Theory and Applications of Hybrid Systems', *Proceedings of the IEEE, Special Issue on Hybrid Systems: Theory and Applications*, Vol. 88, July 2000, pp. 879-887.
- [31] Henzinger, T.A. (1996) 'The theory of hybrid automata', In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278-292. IEEE Computer Society Press.
- [32] Balduzzi, F. and Kumar, R. (2003) 'Hybrid Automata Model of Manufacturing Systems and its Optimal Control Subject to Logical Constraints', *International Journal of Hybrid Systems*, pages 61-80, vol. 3, number 1.
- [33] Willems, H.X. (1999) 'Compact Timed Automata for PLC Programs', Technical Report CSR9925, University of Nijmegen.
- [34] Tapken, J. and Dierks, H. (1998) 'Graphical Development of PLC-Automata', *Proceedings of FTRTFT'98*, vol.1486, LNCS Springer.
- [35] Lucas, M.R. and Tilbury, D.M. (2004) 'The practice of industrial logic design', *Proceedings of the 2004 American Control Conference*, Boston.
- [36] Yevtushenko, N., Villa, T., Brayton, R. K. and Petrenko, A. (2001) 'Solution of parallel language equations for logic synthesis', *Proceedings of the International Conference on Computer Aided Design, ICCAD 2001*, San Jose.
- [37] Bengtsson, L. and Harju, L. (2002) 'Modelling and Verification of PLC Control in Buses', Chalmers University of Technology, Technical Report, Sweden.
- [38] Hobfaur, M. and Wotawa, F. (2005) 'A Diagnosis-Based Casual Analysis Method for Concurrent Hybrid Automata', *16th International Workshop on Principles of Diagnosis*, Monterey, California, USA, June 1-3.
- [39] Zimmerova, B. (2005) 'Component-Interaction Automata for Component-Based System Specification and Verification', *LASER'05*.

- [40]Pereira, F. C. N. and Riley, M. D. (1997) 'Speech Recognition by Composition of Weighted Finite Automata', In *Finite-State Language Processing* (Roche and Schabes, 1997b), chapter 15, pages 431–453.
- [41]Pooley, R. (2007). 'Behavioural equivalence in simulation modelling', *Simulation Modelling Practice and Theory*, 15 (1): 1-20.
- [42]Hendrics, M. and Verhoef, M. (2006), 'Timed automata based analysis of embedded system architectures', *20th International Parallel and Distributed Processing Symposium. IPDPS 2006*.
- [43]Yalcin, A. and Namballa, R. K. (2005), 'An object-oriented simulation framework for real-time control of automated flexible manufacturing systems', *Computers and Industrial Engineering*, 48 (1): 111-127
- [44]Supavatanakul, P., Lunze, J., Puig, V. and Quevedo, J. (2006). 'Diagnosis of timed automata: Theory and application to the DAMADICS actuator benchmark problem', *Control Engineering Practice*, 14 (6): 609-619.
- [45]Simeu-Abazi, Z. and Bouredji, Z. (2006), 'Monitoring and predictive maintenance: Modeling and analyse of fault latency', *Computers in Industry*, 57 (6): 504-515.
- [46]Panopoulos, D. and Metaxiotis, K. (2006). 'Solving production scheduling problems using advanced model checking tools', *International Journal of Computer Applications in Technology*, 26: 37-48.
- [47]Panek, S., Stursberg, O. and Engell, S. (2006). 'Efficient synthesis of production schedules by optimization of timed automata', *Control Engineering Practice*, 14 (10): 1183-1197.
- [48]Guo, Y., Liao, W., Cheng, X. and Liu, L. (2006). 'SimOpt: A new simulation optimization system based virtual simulation for manufacturing systems', *Simulation Modelling Practice and Theory*, 14 (5): 577-585.
- [49]Rodriguez, S., Hilaire, V. and Koukam, A. (2007). 'Towards a holonic multiple aspect analysis and modeling approach for complex systems: Application to the simulation of industrial plants', *Simulation Modelling Practice and Theory*, 15 (5): 521-543.
- [50]Muthiah, K. and Huang S. (2006), 'A review of literature on manufacturing systems productivity measurement and improvement', *International Journal of Industrial and Systems Engineering*, 1 (4): 461-484.

- [51]Younis, M. B. and Frey, G. (2006) 'UML-based Approach for the Re-Engineering of PLC Programs', *IEEE IECON 2006 - 32nd Annual Conference on Industrial Electronics*, Paris, pages 3691-3696.
- [52]Flordal, H., Fabian, M., Åkesson, K. and Spensieri, D. (2007) 'Automatic model generation and PLC-code implementation for interlocking policies in industrial robot cells', *Control Engineering Practice*, Volume 15, Issue 11, Pages 1416-1426.
- [53]Thapa, D., Park, C.M., Dangol, S. and Wang, G.N. (2006) 'III-Phase Verification and Validation of IEC Standard Programmable Logic Controller', *CIMCA Computational Intelligence for Modelling, Control and Automation*, Sydney, Australia.
- [54]Åkesson, K., Fabian, M., Flordal, H. and Malik, R. (2006) 'Supremica – An integrated environment for verification, synthesis and simulation of discrete event systems', *Proceedings of 2006 Workshop of Discrete Event Systems (WODES)*, Ann Arbor, Michigan.
- [55]Music, G., Gradisar, D. and Matko, D. (2005) 'IEC 61131-3 Compliant Control Code Generation from Discrete Event Models', *IEEE International Symposium on, Intelligent Control*, pages 346-351.
- [56]Giese, H., Glesner, S., Leitner, J., Schäfer, W. and Wagner, R. (2006) 'Towards Verified Model Transformations', *3rd International Workshop on Model Development, Validation and Verification (MoDeV²a)*, Genova, Italy.
- [57]Hasdemir, I.T., Kurtulan, S. and Gören, L. (2008) 'An implementation methodology for supervisory control theory', *International Journal of Advanced Manufacturing Technology*, Vol. 36, Issue 3-4, pages 373-385.
- [58]Alenljung, T., Sköldstam, M., Lennartson, B. and Åkesson, K. (2007) 'PLC-based implementation of process observation and fault detection for discrete event systems', *Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering*, pages 207-212.
- [59]International Electrotechnical Commission (2003) 'International Standard IEC 61131-3 Programmable Controllers – Part 3: Programming Languages', edition 2.0.
- [60]Moniruzzaman, M. and Gohari, P. (2007) 'Implementing supervisory control maps with PLC', *Proceedings of the American Control Conference 2007*, Article number 4282159, pages 3594-3599.

- [61]Thapa, D., Park, S.C., Park, C.M. and Wang, G.N. (2007) 'Controller logic design and Implementation using t-MPSG', *ICCAS 2007, International Conference on Control, Automation and Systems*, Article number 4406584, pages 1540-1544.
- [62]Khalgui, M. And Rebeuf, X. (2009) 'An approach to implement a Programmable Logic Controller from real-time software components', *International Journal of Industrial and Systems Engineering*, Vol. 4, No.1, pages 60 – 75.
- [63]International Electrotechnical Commission (2001) "IEC 61499 – Industrial Process Measurement and Control".
- [64]Gerber C., H.-M. Hanisch, and S. Ebbinghaus (2008) "From IEC 61131 to IEC 61499 for Distributed Systems: A Case Study", *EURASIP Journal on Embedded Systems*, Volume 2008, Article ID 231630.
- [65]Lastra, J.L.M. Godinho, L. Lobov, A. Tuokko, R (2005) "An IEC 61499 application generator for scan-based industrial controllers", *INDIN '05. 3rd IEEE International Conference on Industrial Informatics*.
- [66]Hall, K.H. Staron, R.J. Zoitl, A, (2007) "Challenges to Industry Adoption of the IEC 61499 Standard on Event-based Function Blocks", *5th IEEE International Conference on Industrial Informatics*.
- [67]Thramboulidis, K. (2006) "Design Alternatives in the IEC 61499 Function Block Model", *ETFA '06. IEEE Conference on Emerging Technologies and Factory Automation*.
- [68]Doukas, G. and K. Thramboulidis (2008) "Implementation Model Alternatives for IEC 61499 Function Block Networks", *INDIN '08, 6th IEEE International Conference on Industrial Informatics*.
- [69]Panjaitan, S. Frey, G (2006) "Combination of UML Modeling and the IEC 61499 Function Block Concept for the Development of Distributed Automation Systems", *ETFA '06. IEEE Conference on Emerging Technologies and Factory Automation*.
- [70]Sacha, K. (2008) 'Model-based implementation of real-time systems', *Lecture Notes in Computer Science*, Vol. 5219 LNCS, pages 332-345.
- [71]<http://www.control.lth.se/~karlerik/grafchart.html>
- [72]<http://www.uppaal.com>
- [73]<http://www2-data.informatik.unibw-muenchen.de/DiaGen>
- [74]<http://www-verimag.imag.fr/TEMPORISE/kronos/>
- [75]<http://www-cad.eecs.berkeley.edu/~tah/HyTech/>
- [76]<http://theoretica.informatik.uni-oldenburg.de/~moby/>

- [77]<http://www.path.berkeley.edu/shift/Default.htm>
- [78]Deshpande A., Göllü A. and Semenzato L. "The SHIFT programming language for dynamic networks of hybrid automata", *IEEE Transactions on Automatic Control*, 43(4):584{587, April 1998.
- [79]<http://www.supremica.org>
- [80]Alur, R., Grosu, R., Hur, Y., Kumar, V. and Lee, I. (2000)'Modular Specification of Hybrid Systems in CHARON', *Proceedings of the 3rd International Workshop on Hybrid Systems: Computation and Control*, Pittsburgh, PA, March 23-25.
- [81]Hylands, C., Lee, E. A., Liu, J., Liu, X., Neuendorffer, S., Xiong, Y. and Zheng, H. (2003) 'Ptolemy II - Heterogeneous Concurrent Modelling and Design in Java', vol. 2. Department of Electrical Engineering and Computer Sciences University of California at Berkeley. Memorandum UCB/ERL M03/29, version 3.0.
- [82]Deshpande, A., Gollu, A. and Varaiya, P. (1996) 'Shift: a formalism and a programming language for dynamic networks of hybrid automata', *In Hybrid Systems V, Lecture Notes in Computer Science 1567*. Springer-Verlag.
- [83]Deshpande, A. and Varaiya, P. (1997)'The Use of SHIFT in System Design', *Conference on Decision and Control*, San Diego.
- [84]Antoniotti, M. and Gollu, A. (1997) 'SHIFT and SMART-AHS: A Language for Hybrid Systems Engineering, Modelling, and Simulation', *Conference on Domain Specific Languages*, Santa Barbara, CA, U.S.A.
- [85]Hamming, R. W. (1973) *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York.
- [86]Kiencke, U., and L. Nielsen, (2000) *Automotive Control Systems for Engine, Driveline and Vehicle*, Springer, London, 2000.
- [87]Krutov, V. I. (1987) *Automatic Control of Internal Combustion Engines*. Moscow: Mir Publishers, 1987.
- [88]Stone, R. (1992) *Introduction to Internal Combustion Engines*. Hampshire: The Macmillan Press, 1992.
- [89]Taylor, C. F. (1985) *The Internal-Combustion Engine in Theory and Practice*. Massachusetts: The M.I.T. Press, 1985.
- [90]Heywood, J. B. (1988) *Internal Combustion Engine Fundamentals*. New York: McGraw-Hill, 1988.
- [91]Kuo, P. S. (1996) "Cylinder Pressure in a Spark-Ignition Engine: A Computational Model", *Engineering Sciences*, 1996.

- [92] Ferraris, C. (2001) "Concrete Mixing Methods and Concrete Mixers: State of the Art", *Journal of Research of the National Institute of Standards and Technology*, NIST, vol. 106, pp. 391-399, 2001.
- [93] Setien, J., I. Carrasal, J. Figueroa and J. Polanco, (2003) "Application of an artificial neural network to ready-mixed concretes mix design," *Journal of Materiales de Construccions*, Castineira Libreria Tecnica, vol. 53, issue 270, pp. 5-19, 2003.
- [94] Zayed, T. and D. Halpin, (2001) "Simulation of concrete batch plant production," *Journal of Construction Engineering and Management*, ASCE, vol. 127, pp. 132-141, 2001.
- [95] Sawhney, A., O. Abudayyeh and T. Chaitavatputtiporn, (1999) "Modeling and Analysis of Concrete Production Plant Using Petri Nets", *Journal of Computing in Civil Engineering*, ASCE, vol. 13, No. 3, pp. 178-186, 1999.