



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Σχεδιασμός και ανάπτυξη συστήματος μοντελοποίησης διαδικασιών
βιομηχανικής παραγωγής και υλοποίηση μεθόδων αξιολόγησης της
απόδοσης**

Διπλωματική Εργασία

ΤΟΥ

Γεωργόπουλος – Νίνος Νικόλαος
A.M. 1054385

Επιβλέπων

Νικολετσέας Σωτήρης, Καθηγητής

Μέλη Επιτροπής Αξιολόγησης
Βλάχος Κυριάκος, Καθηγητής
Μακρής Χρήστος, Αναπληρωτής Καθηγητής

Πάτρα, 2022

© Copyright συγγραφής Γεωργόπουλος – Νίνος Νικόλαος, 2022

© Copyright θέματος Νικολετσέας Σωτήρης

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής του Πανεπιστημίου Πατρών δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Ευχαριστίες

Θέλω να ευχαριστήσω το καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής και επιβλέποντά μου κ. Σωτήριο Νικολετσέα, για την εμπιστοσύνη που μου έδειξε και την υποστήριξη που μου παρείχε στην εκπόνηση της παρούσας διπλωματικής εργασίας.

Θέλω να ευχαριστήσω ιδιαίτερα τον Γαβρίλη Φίλιο για την καθοδήγηση και την άψογη συνεργασία που είχαμε όλο αυτό το διάστημα. Η βοήθεια και οι οδηγίες του ήταν πολύτιμες στο πλαίσιο της διπλωματικής μου εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τον κ. Χρήστο Μακρή και τον κ. Κυριάκο Βλάχο που μου έκαναν την τιμή να είναι μέλη της τριμελούς επιτροπής για την αξιολόγηση της διπλωματικής μου εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για την στήριξή τους, καθώς και τους φίλους και συμφοιτητές για την αμέριστη συμπαράσταση και υπομονή που έδειξαν καθ' όλη τη διάρκεια των σπουδών μου.

Στους δύο παππούδες μου

Περίληψη

Το **Production Line Simulator** είναι ένα σύστημα προσομοίωσης γραμμών παραγωγής με στόχο τον εντοπισμό προβλημάτων και την βελτιστοποίηση των δεικτών απόδοσης της παραγωγής. Πιο συγκεκριμένα, ο χρήστης έχει τη δυνατότητα να παρακολουθεί την παραγωγική διαδικασία και την συμπεριφορά της γραμμής από τις συνθήκες που την επηρεάζουν.

Για την λειτουργία της προσομοίωσης παρακολουθούνται όλα τα σταματήματα των μηχανημάτων λόγω προβλημάτων που εμφανίζονται τυχαία κατά την αυτόματη λειτουργία ή τα σταματήματα που προκαλεί χειροκίνητα ο χρήστης ώστε το σύστημα να εξάγει συγκεκριμένα συμπεράσματα.

Ο χρήστης μέσω της εφαρμογής βλέπει ποιοι παράγοντες και σε ποιο βαθμό επηρέασαν την λειτουργία των μηχανημάτων της γραμμής, την αλληλεπίδραση που υπάρχει μεταξύ των μηχανημάτων και την επίπτωση στην απόδοση της γραμμής παραγωγής.

Κατά τη λειτουργία της εφαρμογής παρακολουθούνται οι τρεις βασικές καταστάσεις (RUN – STANDBY – STOP) του κάθε μηχανήματος και πως αυτές επηρεάζονται από τις συνθήκες λειτουργίας του ιδίου μηχανήματος αλλά και από τις συνθήκες λειτουργίας του προηγούμενου και του επόμενου μηχανήματος στη γραμμή.

Ο χρήστης μέσω του προσομοιωτή μπορεί να εκτελέσει διαφορετικά σενάρια λειτουργίας για μία γραμμή παραγωγής και να αξιολογήσει τα αποτελέσματα προκειμένου να επιλέξει ποιες είναι οι βέλτιστες ρυθμίσεις που πρέπει να εφαρμόσει στα μηχανήματα της γραμμής. Έτσι, ο χρήστης εντοπίζει τις βέλτιστες ρυθμίσεις ή τις αλλαγές στις παραμετροποιήσεις των μηχανημάτων που πρέπει να κάνει, με αποτέλεσμα να εξασφαλίζεται η συνεχής ροή λειτουργίας της γραμμής, χωρίς αδικαιολόγητα σταματήματα, πετυχαίνοντας ταυτόχρονα την αύξηση των δεικτών απόδοσης και την μείωση του παραγωγικού κόστους.

Abstract

Production Line Simulator is a simulation of a production line system aimed at identifying problems in order to optimize production performance indicators. More specifically, the user has the ability to monitor the production process and the behavior of the line from the conditions that affect it.

For the operation of the simulation, all the stops of the automatic operation of the machines due to a problem or the stops caused manually by the user are monitored so the system can draw specific conclusions.

The user through the application sees which factors and to what extent influenced the operation of the line, sees the effects and what interaction there is in combination with the other machines in the production line.

For the operation of the application, monitor in each machine has three statuses (RUN - STANDBY - STOP) and each machine is affected by the operating conditions of the same machine but also by the operating conditions of the downstream and upstream machine. The user through the simulator can execute different operating scenarios with the results to make optimal settings or change in the configurations of the machines. Thus ensuring the continuous flow of the line operation, without unjustified stops, while achieving the increase of the performance indicators and the reduction of production costs.

Περιεχόμενα

1. Εισαγωγή.....	10
1.1 Σημασία του προβλήματος	10
1.2 Στόχοι της διπλωματικής εργασίας.....	11
1.3 Διάρθρωση της διπλωματικής εργασίας.....	12
2. Σχετική βιβλιογραφία	13
2.1 Λογισμικό προσομοίωσης.....	13
2.2 Γραμμές παραγωγής	14
2.3 Ανάλυση απόδοσης γραμμών παραγωγής	15
2.3.1 Βασικοί δείκτες απόδοσης	16
2.3.2 Προσδιορισμός απωλειών	18
2.3.3 Root Cause Analysis.....	20
2.4 Μοντελοποίηση – προσομοίωση γραμμών παραγωγής	22
2.4.1 Η διαδικασία της μοντελοποίησης.....	23
2.4.2 Process based models	24
2.4.3 Data driven models.....	26
2.4.4 Software Process Simulator.....	27
2.4.5 Digital twin vs process simulations.....	28
2.5 Καινοτομία παρούσας εργασίας	29
3. Σχεδιασμός του συστήματος.....	30
3.1 Παράμετροι συστήματος.....	30
3.1.1 Simulation level	30
3.1.2 Process level	31
3.1.3 UI – RCA level.....	35
3.2 Εργαλεία ανάπτυξης του συστήματος	36
3.2.1 Python.....	36
3.2.2 SimPy	37
3.2.3 Threading.....	42
3.2.4 Numpy	42
3.2.5 Άλλες βιβλιοθήκες.....	42
4. Αρχιτεκτονική του συστήματος.....	43
4.1 Αρχιτεκτονική συστήματος.....	43
4.1.1 Αρχιτεκτονική συστήματος προσομοίωσης	43
4.1.2 Αρχιτεκτονική UI οντότητας	44

4.1.3	Αρχιτεκτονική UI προσομοίωσης	46
4.2	Ανάλυση αρχιτεκτονικής	47
4.2.1	Διαχωρισμός υποσυστημάτων	47
4.2.2	Περιγραφή υποσυστημάτων	47
4.3	Βασικά στάδια προσομοίωσης.....	53
4.4	Συνεχής βελτίωση προσομοιωτή	54
5.	Υλοποίηση του συστήματος.....	58
5.1	Δομή υλοποίησης συστήματος	58
5.2	Υποσύστημα εισαγωγής βασικών παραμέτρων	60
5.3	Υποσύστημα καταχώρησης δεδομένων.....	61
5.4	Υποσύστημα προσομοίωσης.....	62
5.4.1	Υποσύστημα φάρου	65
5.4.2	Υποσύστημα μεταφορικής γραμμής	66
5.4.3	Υποσύστημα buffer	68
5.4.4	Υποσύστημα διεργασίας μηχανήματος	69
5.4.5	Υποσύστημα live χρόνου προσομοίωσης	72
5.4.6	Υποσύστημα μετρικών	72
5.5	Υποσύστημα βλαβών	73
5.6	Υποσύστημα καταγραφής δεδομένων.....	75
5.7	Υποσύστημα επεξεργασίας δεδομένων.....	77
5.8	Υποσύστημα live monitoring.....	79
5.9	Υποσύστημα επισκόπησης προσομοίωσης	80
5.10	Υποσύστημα ανάλυσης απόδοσης μηχανήματος.....	81
6.	Μελέτη περίπτωσης.....	84
6.1	Η γραμμή παραγωγής	84
6.2	Προσομοίωση γραμμής.....	85
7.	Συμπεράσματα και μελλοντικές επεκτάσεις	94
7.1	Συμπεράσματα	94
7.2	Μελλοντικές επεκτάσεις	95
8.	Βιβλιογραφία	97

1. Εισαγωγή

1.1 Σημασία του προβλήματος

Οι σύγχρονες βιομηχανικές επιχειρήσεις λειτουργούν σε ένα κόσμο που αλλάζει ταχύτατα σε ένα πλαίσιο υψηλού ανταγωνισμού όπου τα παραγωγικά τους σχέδια πρέπει να στοχεύουν στην παροχή πολύ καλύτερων προϊόντων και στην υψηλή ποιότητα εξυπηρέτησης των πελατών. Περισσότερο απ' ό,τι στο παρελθόν, οι εταιρείες οι οποίες δεν είναι δυνατό να αναθεωρούν συνεχώς και δυναμικά τις στρατηγικές τους και δεν τροποποιούν τις οργανωτικές διαδικασίες, βρίσκονται σε σοβαρό κίνδυνο να βγουν έξω από το ανταγωνιστικό όριο.

Την δεκαετία του 1990, οι εταιρίες έκαναν σημαντικές προσπάθειες για να βελτιώσουν τις εσωτερικές επιχειρησιακές διαδικασίες, με επίκεντρο την εύρεση και βελτίωση των δραστηριοτήτων που είναι σημαντικές στην αλυσίδα της αξίας του προϊόντος, και επένδυναν μαζί σε καινούργιες πληροφοριακές πλατφόρμες (ERP συστήματα) για την αποθήκευση των δεδομένων και την ενοποίηση της πληροφορίας στην επιχείρηση. Τα τελευταία χρόνια, όλες οι επιχειρήσεις, όπως και οι ΜΜΕ, συνειδητοποιούν ότι η αποδοτικότητά τους είναι σημαντικά εξαρτώμενη από τη στενή συνεργασία τόσο με τους προμηθευτές όσο και με τους πελάτες τους. Το να υιοθετείται η στρατηγική της διοίκησης στην εφοδιαστική αλυσίδα σημαίνει ότι εφαρμόζεται στην πράξη μία φιλοσοφία όπου οι περισσότεροι βιομηχανικοί κόμβοι ενεργούν σε ένα συνεργατικό περιβάλλον. Το όραμα μιας επιχείρησης εφαρμόζεται σε διαφορετικές βιομηχανικές μεθόδους (π.χ. προμήθεια, υπολογιστικές και διοικητικές μεθόδους) και ενεργούν με διαφορετικές πολιτικές. Έρχεται ένα νέο κύμα λύσεων με τέχνασμα να καταβάλλει όλα τα φυσικά, οργανωτικά και πληροφοριακά εμπόδια τα οποία μπορούν να διακινδυνέψουν σοβαρά οποιαδήποτε γραμμή παραγωγής. Η αναπτυγμένη σχεδίαση και δρομολόγηση των συστημάτων προσομοίωσης που σκοπεύουν να επιλύσουν σημαντικά προβλήματα στις τρέχουσες βιομηχανικές παραγωγές, παρέχουν μια κοινή δια-οργανωτική πλατφόρμα η οποία προσομοιώνει την παραγωγική αλυσίδα κατά μήκος ολόκληρου του κύκλου ζωής του προϊόντος, από τον αρχικό εφοδιασμό, στην επεξεργασία και δρομολόγηση και τελικά στην συσκευασία και την αποθήκευση των προϊόντων. Παρά το γεγονός ότι τρέχουν συγχρόνως πολλές λύσεις στην αγορά, τα κοινά χαρακτηριστικά γνωρίσματα των λογισμικών προσομοίωσης, με την εντατική χρήση σύγχρονων μεθόδων παρέχουν στους χρήστες την καλύτερη δυνατή λύση.

Μεταξύ αυτών των ποσοτικών μεθόδων, η προσομοίωση είναι αναμφίβολα μία από τις πιο ισχυρές τεχνικές που εφαρμόζονται. Αποτελεί μια τεχνική που υποστηρίζει το σύστημα, μέσα σε ένα περιβάλλον της γραμμής παραγωγής. Στην βιομηχανία, η προσομοίωση είχε κυρίως χρησιμοποιηθεί για δεκαετίες σαν μια σημαντική υποστήριξη για τις παραγωγικές μηχανές σε τεκμηρίωση νέων ταξιθετημένων επιλογών και σωστού μεγέθους του σχεδίου παραγωγής. Στις μέρες μας, η γνώση της προσομοίωσης θεωρείται μία από τις πιο σημαντικές ικανότητες για την απόκτηση της συνολικής εικόνας και συμπεριφοράς της γραμμής μέσα σε μία επιχείρηση με πολλές και διαφορετικές διαδικασίες. Μέσα στα οράματα για το **Industry 4.0**, η προσομοίωση θεωρείται το πιο αξιόπιστο κλειδί επιτυχίας των εργοστασίων για την επιβίωση εταιριών, χάρη στα προβλεπόμενα χαρακτηριστικά της. Η προσομοίωση θεωρείται ως μία ουσιώδης απόφαση για την κατανόηση του συστήματος.

Στην βιομηχανία, οι γραμμές παραγωγής παρακολουθούνται και καταγράφονται καθημερινά οι δείκτες απόδοσης τους (**KPIs**). Στην λειτουργία των μηχανημάτων τα παραγωγικά ποσοστά (%) είναι αρκετά χαμηλότερα από την ικανότητα τους. Τα τμήματα παραγωγής και υποστήριξης, παρότι καταβάλουν προσπάθειες βελτίωσης, δεν έχουν τα συγκεκριμένα εργαλεία που θα τους βοηθούσαν να εντοπίσουν δυσλειτουργίες και προβλήματα της παραγωγής. Αδυνατούν να αναδείξουν την σημαντικότητα των σημείων

βελτίωσης, να αναδείξουν τις αλλαγές που χρειάζονται στις ρυθμίσεις και τις παραμετροποιήσεις για να υπάρξει καλύτερη απόδοση σε κάθε μηχανήμα αλλά και στο σύνολο της παραγωγικής διαδικασίας. Λόγο της μεγάλης έκτασης της αλυσίδας των πολλών και διαφορετικών μηχανημάτων σε μια γραμμή παραγωγής δεν υπάρχει η λεπτομερής εικόνα, δεν υπάρχει η αξία της μοναδικότητας και εξειδίκευσης κάθε μηχανήματος. Δεν αποτιμώνται όλες οι επιπτώσεις που μπορεί να έχει μια δυσλειτουργία και τελικά πόσο επηρεάζει τα άλλα μηχανήματα, αλλά και στο σύνολο τη παραγωγική διαδικασία.

Οι γραμμές παραγωγής είναι ένα τυπικό περιβάλλον όπου η προσομοίωση μπορεί να θεωρηθεί ένα χρήσιμο τέχνασμα. Στην πραγματικότητα είναι απαραίτητο να βρεθεί πώς χρησιμοποιώντας την τεχνική της προσομοίωσης είναι πιθανό να αναπαράγεις και να εξετάσεις μια διαφορετική εναλλακτική λήψη αποφάσεων πάνω σε πιθανότερα προβλέψιμα σενάρια.

1.2 Στόχοι της διπλωματικής εργασίας

Σκοπός της διπλωματικής είναι με την λειτουργία της προσομοίωσης να δίνεται η δυνατότητα στους ανθρώπους της παραγωγής, στα τεχνικά τμήματα υποστήριξης αλλά και στην διοικητική ομάδα της βιομηχανίας, να έχει ένα εργαλείο που θα της παρέχει μία εικονική εποπτεία της λειτουργίας στην γραμμή παραγωγής τους, οι εργαζόμενοι θα μπορούν να γνωρίζουν τις συγκεκριμένες αιτίες και την πηγή που προκαλεί το κάθε πρόβλημα. Η τεχνική της προσομοίωσης μπορεί να αναδείξει σε διαφορετικούς τομείς πολλά προβλήματα που προέρχονται ή από την τροφοδοσία της πρώτη ύλη παραγωγής προϊόντων, ή από δυσλειτουργίες των μηχανημάτων λόγω κακών ρυθμίσεων, ή λανθασμένων παραμετροποιήσεων, ή μηχανικών ρυθμίσεων με αποτέλεσμα να υπολειτουργεί. Επίσης ο προσομοιωτής μπορεί να αναδείξει **κρυφές αιτίες** που δεν γίνονταν αντιληπτές από τον ανθρώπινο παράγοντα, προβλήματα που έχουν αντίκτυπο στην ποιότητα του παραγόμενου προϊόντος, στην απόδοση, στα κόστη συντήρησης, τα ενεργειακά κόστη και τα κόστη από τα απορριπτόμενα ή μη παραγόμενα υλικά. Παρατίθεται μία λεπτομερής περιγραφή του προσομοιωτή που αναλύει τον σκοπό της χρήσης του και τον τρόπο που εφαρμόστηκε σε πραγματικά παραδείγματα σε βιομηχανικά δεδομένα. Τα αποτελέσματα από την εφαρμογή των τεχνικών της προσομοίωσης ήταν άμεσα και σημαντικά, καθώς ανιχνεύτηκαν τα σημεία που μπορούσαν να γίνουν οι βελτιώσεις στην γραμμή και μειώθηκαν τα σταματήματα των μηχανημάτων, βελτιώθηκαν οι δείκτες απόδοσης, αυξήθηκε η ταχύτητα της παραγωγής, μειώθηκε το κόστος παραγωγής με αποτέλεσμα να υπάρξουν οικονομικά οφέλη.

Η διπλωματική εργασία έχει στόχο την συλλογή ψηφιακών δεδομένων και την δημιουργία προσομοίωσης που θα είναι το μοντέλο αναπαράστασης των μηχανημάτων μιας γραμμής παραγωγής. Το σύστημα έχει ως σκοπό να αναπαριστά και να «μιμείται» τη συμπεριφορά της γραμμής και συνεπώς να επιτρέπει την κατανόηση και την εξοικείωση με τα χαρακτηριστικά της λειτουργίας της. Όλα αυτά έχουν σκοπό οι χρήστες να χρησιμοποιούν το εργαλείο προσομοίωσης που θα τους προσφέρει ξεκάθαρη εικόνα για την παραγωγή καθώς εντοπίζονται τα προβλήματα με την μεγαλύτερη βαρύτητα στον παραγωγικό χρόνο και έτσι μπορούν να δώσουν συγκεκριμένες λύσεις για να βελτιωθούν οι δείκτες απόδοσης.

Αυτή η τεχνολογική καινοτομία με την ανάπτυξη λογισμικού, με τον υπολογισμό επιδόσεων του συστήματος, την ανάλυση της ικανότητας διεκπεραίωσης, την αξιολόγηση στρατηγικών ελέγχου του εξοπλισμού, τον υπολογισμό του μέσου χρόνου αναμονής μέρους του προϊόντος στην ουρά των μηχανών και μεταφορικών γραμμών, αλλάζει τον τρόπο που οι χρήστες μπορούν να εντοπίσουν και να βελτιώσουν τα προβλήματα σε σχέση με τον χρόνο.

Η λειτουργία της εφαρμογής θα καταφέρει να μειώσει τους χρόνους και την συχνότητα σταματήματος της λειτουργίας των μηχανημάτων ή και το σύνολο της γραμμής παραγωγής,

Θα μπορεί να αυξήσει τους δείκτες παραγωγικότητας (KPIs), να μειώσει τα κόστη συντήρησης, τα παραγωγικά και τα ενεργειακά κόστη ώστε να αυξηθούν τα οικονομικά οφέλη.

Συνεισφορά της διπλωματικής εργασίας

Η αξία των δεδομένων και η βοήθεια που μπορεί να παρέχει το σύστημα προσομοίωσης όπως αυτό περιγράφεται στην Ενότητα 1.2, μαζί με την έρευνα που διεξήχθη και οδήγησε στο συμπέρασμα της έλλειψης αντίστοιχου συστήματος στην Ελλάδα, ήρθε να καλύψει η παρούσα Διπλωματική Εργασία. Με άλλα λόγια έχοντας αναφερθεί όχι μόνο στα συχνά αλλά και τα εξειδικευμένα ή κρυφά προβλήματα που παρουσιάζουν οι γραμμές παραγωγής και πως αυτά τα προβλήματα επηρεάζουν συνολικά την απόδοσή και κάτ' επέκταση τη βιωσιμότητα της ίδιας της βιομηχανίας, με αυτό ως γνώμονα θα αναπτυχθεί το λογισμικό με βάση τους στόχους που έχουν τεθεί.

Η υλοποίηση του προσομοιωτή βασίζεται σε δύο λειτουργίες, στη διεπαφή με το χρήστη και στην επικοινωνία του με το πρόγραμμα προσομοίωσης. Η παρούσα διπλωματική υλοποιεί και τις δύο αυτές λειτουργίες. Μέσω της διεπαφής ο χρήστης έχει την ικανότητα να θέσει τις παραμέτρους του συστήματος με τις οποίες ορίζει τη ταχύτητα της παραγωγής, τα buffer των μηχανημάτων και το χρόνο προσομοίωσης σε σχέση με το πραγματικό χρόνο. Επίσης έχει την δυνατότητα να επιλέξει την συχνότητα εμφάνισης των αυτόματων βλαβών είτε την χειροκίνητη ενεργοποίηση βλαβών, οι οποίες θα προκύπτουν στο προσομοιωτή.

Παρέχεται πρόσβαση στην πλήρη εποπτεία της γραμμής παραγωγής που προσομοιώνει το πρόγραμμα. Μέσω της διεπαφής ο χρήστης εξαγεί συμπεράσματα για τη συμπεριφορά της γραμμής σύμφωνα με το συγκεκριμένο σενάριο που εκτελείται κάθε φορά σε πραγματικό χρόνο. Στη πραγματικότητα η εκτέλεση πολλαπλών σεναρίων σε μια πραγματική γραμμή παραγωγής απαιτεί ανθρώπινους πόρους, πρώτες ύλες, ενεργειακά κόστη και δέσμευση παραγωγικού χρόνου της γραμμής. Αναλυτικότερη περιγραφή για το σύστημα γίνεται στο Κεφάλαιο 4 & 5.

1.3 Διάρθρωση της διπλωματικής εργασίας

Στη συνέχεια, στο Κεφάλαιο 2 θα αναφερθεί η σχετική βιβλιογραφία και παρόμοια συστήματα ή εργαλεία που εντοπίστηκαν κατά την σχετική έρευνα. Στο κεφάλαιο 3 θα αναφερθούν λεπτομερώς η αρχιτεκτονική και οι απαιτήσεις του προσομοιωτή, ενώ στο Κεφάλαιο 4 θα αναλυθούν τα εργαλεία και οι τεχνικές που χρησιμοποιήθηκαν για την υλοποίηση του προγράμματος, η μεθοδολογία που ακολουθήθηκε, καθώς και οι συνθήκες με τις οποίες λειτουργεί ο προσομοιωτής. Τέλος, στο Κεφάλαιο 5 θα γίνει η παρουσίαση των σεναρίων χρήσης του προγράμματος και στο Κεφάλαιο 6 συνοψίζονται τα συμπεράσματα καθώς επίσης αναφέρονται και οι πιθανές μελλοντικές προεκτάσεις του συστήματος.

2. Σχετική βιβλιογραφία

Στο παρόν κεφάλαιο γίνεται αναφορά σε σχετικές δημοσιεύσεις που έχουν προσφέρει πολύτιμες παρατηρήσεις, υλοποιήσεις και τεχνικές, που κυρίως εστιάζουν στο θέμα της προσομοίωσης παραγωγικών διαδικασιών. Οι εργασίες και οι πληροφορίες που παρέχονται, χρησιμοποιούνται ως θεμέλια στην παρούσα διπλωματική εργασία. Αποτελούν έμπνευση για την δημιουργία ενός νέου λογισμικού προσομοίωσης το οποίο μπορεί να εφαρμοστεί στο μεγαλύτερο ποσοστό των παραγωγικών διαδικασιών.

2.1 Λογισμικό προσομοίωσης

Για την προσομοίωση των συστημάτων παραγωγής χρησιμοποιούνται συνήθως εμπορικά προϊόντα λογισμικού προσομοίωσης και λιγότερο συστήματα που έχουν αναπτυχθεί σε γλώσσες γενικού σκοπού (π.χ. C++, Java). Ορισμένα από τα σημαντικότερα κριτήρια για την επιλογή ενός λογισμικού προσομοίωσης είναι:

- α) η ευελιξία μοντελοποίησης (η δυνατότητα να μοντελοποιούμε οποιοδήποτε σύστημα ανεξάρτητα από την πολυπλοκότητα ή την μοναδικότητα του) και
- β) η ευκολία χρήσης

Υπάρχουν δύο βασικές κατηγορίες λογισμικών για προσομοίωση συστημάτων παραγωγής. Η πρώτη κατηγορία περιλαμβάνει τις **γλώσσες προσομοίωσης** (Simulation Languages) [1]. Η γλώσσα προσομοίωσης είναι ένα λογισμικό σύστημα αρκετά γενικό (όσο αφορά στις δυνατότητες και στις περιπτώσεις στις οποίες μπορεί να χρησιμοποιηθεί) και στο οποίο η κατασκευή του μοντέλου πραγματοποιείται με "προγραμματισμό". Το βασικό πλεονέκτημα μιας καλής γλώσσας προσομοίωσης αποτελεί η ευελιξία μοντελοποίησης, ενώ το βασικό μειονέκτημα είναι ότι απαιτείται εμπειρία και χρόνος σε "προγραμματισμό". Βέβαια στις γλώσσες προγραμματισμού οι οποίες είναι εξειδικευμένες για την προσομοίωση συστημάτων παραγωγής υποστηρίζονται ειδικές δομές για την μοντελοποίηση των συστημάτων (όπως μηχανές, ιμάντες μεταφοράς κλπ.), οι οποίες μειώνουν αρκετά τον απαιτούμενο χρόνο για τον "προγραμματισμό".

Η δεύτερη κατηγορία περιλαμβάνει τους **προσομοιωτές συστημάτων παραγωγής** (Simulators) [2], οι οποίοι χρησιμοποιούν εικονίδια (icon-based). Χρησιμοποιώντας λογισμικά συστήματα αυτού του τύπου, το μοντέλο δομείται καθορίζοντας τις ιδιότητες των εικονιδίων, τα οποία αναπαριστούν εξαρτήματα, μηχανές ή άλλου είδους εξοπλισμό. Η γραφική αναπαράσταση απλοποιεί σε μεγάλο βαθμό την κωδικοποίηση και τον έλεγχο για λάθη του μοντέλου. Αρκετά συστήματα αυτής της κατηγορίας, παρέχουν την δυνατότητα γραφικής αναπαράστασης της κίνησης (Animation). Κάθε φορά, κατά την οποία η κατάσταση της προσομοίωσης αλλάζει, μια αντίστοιχη αλλαγή εμφανίζεται στην γραφική αναπαράσταση. Η γραφική προσομοίωση της κίνησης έχει γίνει ευρέως αποδεκτή διαδικασία στην προσομοίωση των συστημάτων παραγωγής εξ αιτίας της καλής επικοινωνίας του χρήστη με την δυναμική συμπεριφορά του μοντέλου προσομοίωσης, γεγονός το οποίο αυξάνει σημαντικά την αξιοπιστία του συστήματος.

Εκτός του γεγονότος, ότι η γραφική αναπαράσταση βοηθά την επικοινωνία του χρήστη με το μοντέλο, είναι επίσης χρήσιμη στον έλεγχο των λαθών, την επιβεβαίωση και την βελτίωση του μοντέλου. Παραδείγματα λογισμικού προσομοίωσης είναι τα [Extend](#), [ProModel](#), [Enterprise Dynamics](#), [Flexsim](#) και [WITNESS](#). Σε ένα λογισμικό αυτού του τύπου, το μοντέλο της προσομοίωσης δημιουργείται με την χρήση γραφικού περιβάλλοντος (π.χ. με

την χρήση drag & drop), επιλέγοντας αντικείμενα από μενού με το ποντίκι και συμπληρώνοντας τις παραμέτρους τους σε παράθυρα διαλόγου.

Το μεγαλύτερο **πλεονέκτημα** ενός προγράμματος προσομοίωσης αποτελεί η σημαντική μείωση του χρόνου που απαιτείται για τη δημιουργία του μοντέλου προσομοίωσης. Βασικό **μειονέκτημα** αποτελεί η μειωμένη ευελιξία τους, σε σχέση με τις γλώσσες προσομοίωσης, στην μοντελοποίηση πολύπλοκων διαδικασιών.

Για τον λόγο αυτό, επιλέχθηκε να γίνει η κατασκευή του λογισμικού προσομοίωσης με προγραμματισμό.

2.2 Γραμμές παραγωγής

Η βιομηχανική παραγωγή προϊόντων βασίζεται στην απρόσκοπτη λειτουργία όλων των μηχανημάτων που επεξεργάζονται διαδοχικά τα προϊόντα στη γραμμή παραγωγής. Η γραμμή περιλαμβάνει διαφορετικά μηχανήματα με διαφορετικές λειτουργίες, τα οποία είναι δομημένα και οργανωμένα σε μία παραγωγική σειρά ανάλογα με την επεξεργασία που πρέπει να κάνουν στο συγκεκριμένο προϊόν. Πολλές γραμμές παραγωγής έχουν την δυνατότητα τα μηχανήματα τους με την αλλαγή εξαρτημάτων και την αλλαγή των ρυθμίσεων να μπορούν να παράξουν διαφορετικά προϊόντα. Οι αλλαγές των διαφορετικών προϊόντων παραγωγής στην ίδια γραμμή μηχανημάτων είναι μία βασική αιτία που έχουν μειωμένη απόδοση και αυτό οφείλεται στην λάθος προσαρμογή των σωστών ρυθμίσεων για το προϊόν που θα παράξουν.

Στην αυτόματη λειτουργία της γραμμής παραγωγής συνήθως υπάρχουν πολλές καθυστερήσεις που οφείλονται σε πρόβλημα που παρουσιάζει συχνά είτε ένα συγκεκριμένο μηχανήμα είτε κάθε φορά ένα διαφορετικό μηχανήμα. Οι κατασκευαστές για να βοηθήσουν την αδιάκοπη λειτουργία παραγωγής, ανάμεσα στα μηχανήματα προσθέτουν μεταφορικές γραμμές με αυξημένη χωρητικότητα προϊόντων που ονομάζονται BUFFER, έτσι ώστε όταν σταματήσει κάποιο μηχανήμα να υπάρχουν διαθέσιμα προϊόντα για να συνεχίσουν να παράγουν τα άλλα μηχανήματα μέχρι να ξεκινήσει πάλι να παράγει το μηχανήμα που ήταν σταματημένο. Έτσι δίνεται η δυνατότητα και στον χειριστή να έχει χρόνο να αποκαταστήσει το πρόβλημα σε κάποιο μηχανήμα χωρίς να σταματήσει ολόκληρη η γραμμή. Όμως είναι συχνό φαινόμενο να μην φτάνει ο χρόνος που παρέχει ο BUFFER με αποτέλεσμα να σταματάνε και τα υπόλοιπα μηχανήματα και άρα να σταματάει το σύνολο της παραγωγής με σημαντικές επιπτώσεις στους δείκτες παραγωγικότητας της γραμμής.

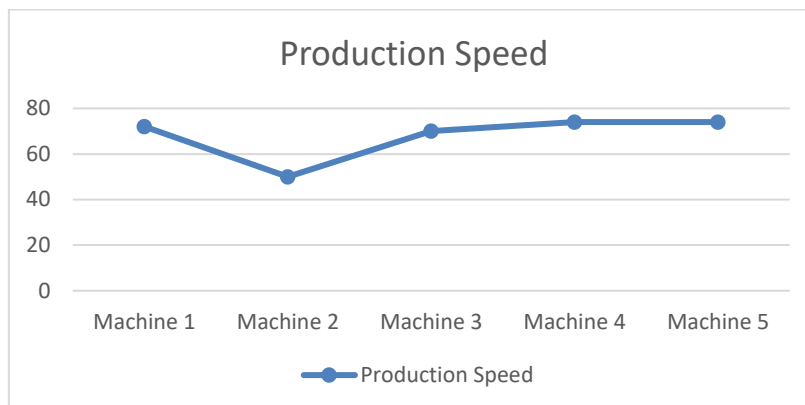
Σήμερα η αυτοματοποιημένες γραμμές παραγωγής καθώς και η εφαρμογή των ρομποτικών κυψελών, δίνουν τη δυνατότητα να υπάρχει μείωση των αδικαιολόγητων σταματημάτων στα μηχανήματα, και να υπάρχει περαιτέρω βελτίωση της ταχύτητας παραγωγής και του ελέγχου ποιότητας των παραγόμενων προϊόντων.

Bottleneck γραμμές

Ο Schroeder ορίζει το **bottleneck** ως, «ένα κέντρο εργασίας του οποίου η χωρητικότητα είναι μικρότερη από τη ζήτηση και μικρότερη από τις δυνατότητες όλων των άλλων πόρων. Ένα μηχανήμα bottleneck θα περιορίσει τη χωρητικότητα ολόκληρης της γραμμής και μια ώρα καθυστέρησης, βλάβης ή σταματήματος που προστίθεται στο bottleneck, προτίθεται μια ώρα συνολικά σε ολόκληρη τη Παραγωγή. Μια ώρα που προστίθεται σε οποιοδήποτε άλλο μηχανήμα της γραμμής δε θα επηρεάσει καθόλου το χρονοδιάγραμμα της συνολικής γραμμής». [3]

Το bottleneck καθορίζει την χωρητικότητα και την εξέλιξη ολόκληρης της γραμμής παραγωγής. Αποτελεί το μηχανήμα το οποίο είναι υπεύθυνο για την ομαλή λειτουργία της

παραγωγής. Ιδανικά δεν πρέπει να επηρεάζεται από κανένα εξωτερικό παράγοντα και πρέπει να λειτουργεί αδιάκοπα. Για παράδειγμα σε μια γραμμή συναρμολόγησης εξαρτημάτων, αν προστεθούν 5 λεπτά εργασίας στις εργασίες ενός σταθμού bottleneck, όλοι οι υπόλοιποι σταθμοί θα πρέπει να περιμένουν για 5 λεπτά μέχρι να παραλάβουν τα συναρμολογημένα προϊόντα. Αυτό δείχνει ότι, στην πραγματικότητα, κάθε χειριστής στη γραμμή είναι ένα σημείο συμφόρησης για τη γραμμή. Είναι ζωτικής σημασίας να εξασφαλιστεί η ισορροπία στη γραμμή με σκοπό να μηδενιστούν οι απώλειες και έτσι να έχει ως αποτέλεσμα τη μεγιστοποίηση της απόδοσης της γραμμής. [4]



Σχήμα 2.1: Bottleneck

Στο σχήμα 2.1 αποτυπώνεται η ταχύτητα των μηχανημάτων με την οποία παράγουν προϊόντα. Συγκεκριμένα η δυναμική ταχύτητα του μηχανήματος 1 είναι: 72 προϊόντα/λεπτό, κάθε ένα λεπτό το μηχάνημα βγάζει στην έξοδό του 72 προϊόντα. Για το μηχάνημα 3 η ταχύτητά του είναι 70 προϊόντα το λεπτό. Παρατηρώντας το σχεδιάγραμμα, το μηχάνημα 2 έχει την μικρότερη ταχύτητα στη γραμμή παραγωγής. Αυτό το μηχάνημα ορίζεται ως **bottleneck**. Σκοπός του είναι να λειτουργεί στη μέγιστη δυνατή ταχύτητα που του έχει οριστεί σύμφωνα με τη συνολική ταχύτητα της παραγωγής αλλά και των επιμέρους μηχανημάτων. Έτσι οποιαδήποτε περίπτωση και να αντιμετωπίσει στην οποία δεν ευθύνεται το ίδιο, όπως σταμάτημα του μηχανήματος 4 εξαιτίας βλάβης. Το μηχάνημα 3 θα πρέπει να μειώσει τη ταχύτητα παραγωγής του χωρίς να υποβεί την κατώτατη ταχύτητα της γραμμής παραγωγής (την οποία έχει το bottleneck). Επομένως, το μηχάνημα 2 θα συνεχίσει να λειτουργεί με σταθερό ρυθμό μέχρι ένα συγκεκριμένο χρονικό διάστημα. Αν στο συγκεκριμένο χρονικό διάστημα, στο μηχάνημα 4 αποκατασταθεί η βλάβη τότε η γραμμή παραγωγής επανέρχεται στα φυσιολογικά επίπεδα παραγωγής. Διαφορετικά, αν δεν αποκατασταθεί έγκαιρα η βλάβη στο μηχάνημα 4 τότε μετά από το συγκεκριμένο χρονικό διάστημα θα σταματήσει το **bottleneck** και κατ' επέκταση ολόκληρη η παραγωγή. [5]

2.3 Ανάλυση απόδοσης γραμμών παραγωγής

Ο καθορισμός των δεικτών απόδοσης μιας γραμμής παραγωγής είναι ιδιαίτερα σημαντικός, καθώς μέσω αυτών καταγράφεται και παρακολουθείται στατιστικά η απόδοση της γραμμής. Παρακάτω παρουσιάζεται πως ορίζονται οι τρεις Βασικοί Δείκτες απόδοσης σε μια γραμμή παραγωγής μέσω των οποίων τίθενται και οι παραγωγικοί στόχοι. Σε κάθε περίπτωση, η επίτευξη του στόχου Βελτίωσης ενός δείκτη, προϋποθέτει γνώση τόσο του τρόπου υπολογισμού του, όσο και της ουσιαστικής σημασίας του. Για τον λόγο αυτό, στις υποενότητες που ακολουθούν, παρουσιάζεται αναλυτικά ο τρόπος υπολογισμού και η ουσιαστική συνεισφορά των δεικτών αυτών.

2.3.1 Βασικοί δείκτες απόδοσης

Line Utilization (LU)

Ένα από τα βασικότερα στοιχεία για την αξιολόγηση μιας γραμμής παραγωγής είναι το ποσοστό χρησιμοποίησής της, η αλλιώς Line Utilization. Πολύ απλά, αν ένα εργοστάσιο δύναται να χρησιμοποιεί μια γραμμή παραγωγής για 3 βάρδιες την ημέρα και στην πράξη την λειτουργεί μόνο δύο, τότε η χρήση της γραμμής αυτής θα είναι 66,66%. Κατ' αυτό τον τρόπο μπορούμε να υπολογίσουμε και την απόδοση μιας γραμμής βάσει της χρήσης της. Αν για παράδειγμα μπορούμε να παράξουμε σε μια βάρδια 10.000 τεμάχια ενός προϊόντος αλλά στην πράξη παράγουμε 8.000 τεμάχια (γιατί δούλεψε λιγότερο χρόνο), τότε η λέμε ότι η εν λόγω γραμμή λειτουργεί με λόγο χρήσης 80%. Μαθηματικά, το ποσοστό χρήσης μπορεί να εκφραστεί ως τα πραγματικά εξερχόμενα μείον τα αναμενόμενα εξερχόμενα προς τα αναμενόμενα εξερχόμενα. Οπότε, το ποσοστό χρήσης της γραμμής προκύπτει από το πηλίκο της διαίρεσης των ωρών χρήσης προς τις διαθέσιμες ώρες.

$$LU = \frac{\text{Διαθέσιμος χρόνος για παραγωγή (συμπαιριλαμβανομένων των απωλειών)}}{\text{Συνολικό χρόνος προσομοίωσης}}$$

Στην πράξη, σπάνια μια βιομηχανία λειτουργεί με Capacity Utilization 100% εκτός από τις περιπτώσεις που αφορά ημερήσια ή έστω εβδομαδιαία αναφορά. Ο δείκτης αυτός αποκτά αξία, αν παρατηρείται και αξιολογείται σε μηνιαία ή ετήσια βάση, ώστε να συμπεριλαμβάνει τυχόν χαρακτηριστικά εποχικότητας του παραγόμενου προϊόντος. Ένα καλό ποσοστό χρήσης μιας παραγωγικής γραμμής θεωρείται το 85%, ειδικά αν εμφανίζει και σταθερότητα καθ' όλη τη διάρκεια του χρόνου [6].

Overall Equipment Effectiveness (OEE)

Ο δείκτης συνολικής απόδοσης του εξοπλισμού παραγωγής μπορεί να εκφραστεί ως ο λόγος της πραγματικής απόδοσης του εξοπλισμού προς την μέγιστη απόδοσή του. Η συνολική απόδοση που αφορά τον εξοπλισμό δίνεται από το πηλίκο της διαίρεσης του παραγωγικού χρόνου μέγιστης ταχύτητας και μηδενικών απωλειών, προς τον πραγματικό διαθέσιμο χρόνο για παραγωγή.

$$OEE = \frac{\text{Παραγωγικό χρόνο μέγιστης ταχύτητας και μηδενικών απωλειών}}{\text{Συνολικό χρόνος προσομοίωσης}}$$

Η συνολική απόδοση του εξοπλισμού, σαν Βασικός Δείκτης Απόδοσης προέρχεται από την πρακτική της Ολικής Παραγωγικής Συντήρησης (Total Productive Maintenance), που αναπτύχθηκε από τον S. Nakajima, του ιαπωνικού ινστιτούτου βιομηχανικής συντήρησης. Στόχος της είναι η επίτευξη της απόλυτης απόδοσης και των μηδενικών απωλειών, που στην πράξη μεταφράζεται με μηδενικά ελαττωματικά προϊόντα, **μηδενικά σταματήματα παραγωγής**, μηδενικές απορρίψεις ελαττωματικών προϊόντων κατά την διάρκεια παραγωγής ή αλλαγής κωδικού και μηδενικές απώλειες ταχύτητας εξοπλισμού. Τα αποτελέσματα του δείκτη αυτού, αποτελούν εργαλείο για τους διοικούντες της παραγωγής και της συντήρησης, καθώς τους παρέχει μια σφαιρική εικόνα της απόδοσης του εργοστασίου και ταυτόχρονα τους κατευθύνει στην Βελτίωση του σημαντικότερου παράγοντα απωλειών [7].

Μια άλλη προσέγγιση της συνολικής απόδοσης του εξοπλισμού δίνεται από την ακόλουθη σχέση, στην οποία εφαρμόζονται οι τρεις παράγοντες απόδοσης:

$$OEE = Availability \times Performance \times Quality$$

1. Availability: Στην διαθεσιμότητα, συγκαταλέγονται προβλήματα που σχετίζονται με καθυστερήσεις της παραγωγικής διαδικασίας που οφείλονται σε δυσλειτουργίες του εξοπλισμού.
2. Performance: Στην απόδοση, συγκαταλέγονται κυρίως απώλειες ταχύτητας αλλά και ελλείψεις υλικών και προσωπικού.
3. Quality: Στην ποιότητα, συγκαταλέγονται απώλειες που οφείλονται στην παραγωγή ελαττωματικών προϊόντων, είτε αυτά πρέπει να επεξεργαστούν πάλι στην γραμμή παραγωγής είτε απορρίπτονται.

Η τιμή του καθένα από τους τρεις βασικούς αυτούς παράγοντες απόδοσης, κυμαίνεται μεταξύ του 0 και του 1. Αν τους εξετάσουμε μεμονωμένα, μια ικανοποιητική τιμή του καθενός θα ήταν το 0.9 ήτοι 90%. Σε αυτή τη περίπτωση λοιπόν η συνολική απόδοση του εξοπλισμού θα ήταν: $OEE = 0,9 \times 0,9 \times 0,9 = 0,73$ ή 73%. Συνεπώς, οι απώλειες της παραγωγής λόγω του εξοπλισμού θα έφταναν το 27%, γεγονός που θα οφειλόταν στο γινόμενο των τριών παραγόντων. Η χρήση των τριών παραγόντων απόδοσης βοηθά στην ποσοτικοποίηση του OEE, ωστόσο μειονεκτεί στο γεγονός ότι τους θεωρεί ίσης βαρύτητας, ενώ στην πράξη η κάθε βιομηχανία μπορεί να τους αξιολογεί οικονομικά αλλά και στρατηγικά με διαφορετική βαρύτητα [8].

Global Efficiency (GE)

Για τον προσδιορισμό την συνολικής απόδοσης μια γραμμής παραγωγής, εξάγεται ο δείκτης GE (Global Efficiency), ο οποίος χρησιμοποιεί όλα τα δεδομένα που επηρεάζουν την παραγωγική διαδικασία, είτε αυτά προέρχονται από προγραμματισμένες εργασίες είτε όχι. Η μαθηματική προσέγγιση της συνολικής απόδοσης (GE) μιας γραμμής παραγωγής, δίνεται από το πηλίκο της διαίρεσης του συνολικού χρόνου παραγωγής με μέγιστη ταχύτητα και μηδενικά ελαττωματικά προϊόντα προς το χρόνο χρήσης της γραμμής.

$$GE = \frac{\text{Παραγωγικός χρόνος μέγιστης ταχύτητας και μηδενικών απωλειών}}{\text{Διαθέσιμος χρόνος για παραγωγή (συμπ/μενων απωλειών απόδοσης)}}$$

Στο σημείο αυτό, πρέπει να αποσαφηνιστεί η μέθοδος προσδιορισμού της μέγιστης ταχύτητας μιας παραγωγικής γραμμής. Αρχικά, ως γραμμή παραγωγής ορίζεται το σύνολο του εξοπλισμού, των μηχανών που συμμετέχουν στην παραγωγή τελικών ή ημιτέτοιμων προϊόντων. Η μέγιστη ταχύτητα παραγωγής λοιπόν μιας παραγωγικής γραμμής εξαρτάται από την μέγιστη ταχύτητα του πιο «αργού» εξοπλισμού. Ουσιαστικά, ο εξοπλισμός που καθυστερεί την υπόλοιπη γραμμή (**bottle-neck**) καθορίζει την μέγιστη ταχύτητα παραγωγής. Σε βαθύτερη ανάλυση όμως, ως μέγιστη ταχύτητα ορίζεται αυτή στην οποία ο εξοπλισμός παράγει απρόσκοπτα, χωρίς απώλειες, προϊόν εντός προδιαγραφών. Γίνεται λοιπόν σαφές, ότι η μέγιστη ταχύτητα παραγωγής ενδέχεται να διαφέρει από την μέγιστη ταχύτητα που έχει καθορίσει ο κατασκευαστής.

Ένα ακόμη κρίσιμο στοιχείο στον υπολογισμό της συνολικής απόδοσης της γραμμής, είναι η σωστή και αμερόληπτη εφαρμογή των μετρήσεων και των δεδομένων. Εάν για παράδειγμα υποτιμήσουμε για οποιοδήποτε λόγο την μέγιστη ταχύτητα παραγωγής μιας

γραμμής, ενδέχεται η συνολική της απόδοση (GE) για κάποια περίοδο, να ξεπεράσει το 100%. Αν αυτό συμβεί, τότε αυτομάτως αποδεικνύεται λανθασμένη η εκτίμηση της μέγιστης ταχύτητας και ως εκ τούτου η συνολική απόδοση της γραμμής έχει αξιολογηθεί λανθασμένα. Ο Βασικός Δείκτης Απόδοσης μιας γραμμής, αποτελεί μέτρο προσδιορισμού στόχων αλλά και έργων Συνεχούς Βελτίωσης. Όπως θα δούμε στη συνέχεια, η εφαρμογή συγκεκριμένων τεχνικών και μεθοδολογιών στοχεύει στον περιορισμό των παραμέτρων που επηρεάζουν τόσο την απόδοση των μηχανών όσο και την συνολική απόδοση της γραμμής παραγωγής. Στα πλαίσια της βιομηχανικής παραγωγής, καλό ποσοστό απόδοσης για μια γραμμή με υψηλή χρήση, θεωρείται >80%. Αυτό φυσικά δεν σημαίνει πως υπάρχουν περιπτώσεις που η συνολική απόδοση πλησιάζει ή και ξεπερνά το 90%.

2.3.2 Προσδιορισμός απωλειών

Η εξαγωγή των δεικτών GE και OEE είναι μια διαδικασία που αφενός απεικονίζει την απόδοση μια γραμμής παραγωγής ή και ολόκληρου του εργοστασίου, δίνοντας στους διοικούντες την δυνατότητα να λάβουν άμεσες αποφάσεις και αφ' ετέρου εστιάζει στις διαδικασίες που επιδέχονται Βελτίωση. Ο προσδιορισμός των «ευκαιριών» Βελτίωσης που εμπεριέχονται στους δείκτες, απαιτεί αρχικά την κατανόηση των έξι βασικών παραγόντων που επηρεάζουν την παραγωγή. Αυτοί είναι οι ακόλουθοι [9].

1. Αλλαγές κωδικών προϊόντων παραγωγής
2. Ηλεκτρομηχανολογικές βλάβες
3. Μικρά σταματήματα
4. Απώλειες ταχύτητας
5. Παραγωγή ελαττωματικών προϊόντων
6. Παραγωγή προϊόντων εκτός προδιαγραφών

Κάθε ένας από τους παραπάνω παράγοντες δύναται να μειώσει την παραγόμενη ποσότητα προϊόντων και ως εκ τούτου θα πρέπει να υπολογίζονται σε κοινούς όρους, ώστε η επιρροή τους να είναι συγκρίσιμη. Οι απώλειες των παραγόντων αυτών, εκφράζονται σε μονάδες χρόνου. Είναι πολύ σημαντικό να έχει καθοριστεί συγκεκριμένος τρόπος καταγραφής ή μετατροπής των απωλειών. Δεν είναι λίγες οι περιπτώσεις εργοστασίων – βιομηχανιών, που τα διαθέσιμα δεδομένα τους καταγράφονται σε διαφορετικές, μη συγκρίσιμες μονάδες, ή αφορούν διαφορετικές χρονικές περιόδους. Οι διεργασίες που επηρεάζουν την απόδοση, μπορούν να διαιρεθούν σε δυο κατηγορίες: α) Προγραμματισμένες και β) Μη προγραμματισμένες

Προγραμματισμένες απώλειες

Οι απώλειες της κατηγορίας αυτής, είναι προγραμματισμένες και οφείλονται κυρίως σε εργασίες προληπτικής συντήρησης εξοπλισμού, βάρδιες χωρίς ζήτηση για παραγωγή ή χωρίς εργατικό δυναμικό, ημέρες διακοπών, αργιών κ.ο.κ. Οι απώλειες αυτές, συνδέονται άμεσα με την χρήση – ζήτηση της γραμμής παραγωγής και τον δείκτη GE (Global Efficiency). Είναι σαφές πως η εντατικοποίηση των εργασιών δεν μπορεί να μεταβάλει την χρήση των γραμμών, αλλά ούτε μπορεί να θυσιαστεί η συντήρηση του εξοπλισμού, προκειμένου να κερδίσουμε σε απόδοση. Στην πραγματικότητα η συντήρηση είναι αυτή που εξασφαλίζει την υψηλή απόδοση λειτουργίας των μηχανών. Για τον λόγο αυτό, απαιτείται μια εξισορρόπηση μεταξύ της χρήσης των μηχανών και της λειτουργίας του σε υψηλά επίπεδα απόδοσης. Γενικά, είναι προτιμότερο να λειτουργεί μια γραμμή παραγωγής για μικρά χρονικά διαστήματα, με υψηλή απόδοση των μηχανών, παρά μεγάλα διαστήματα με χαμηλή απόδοση.

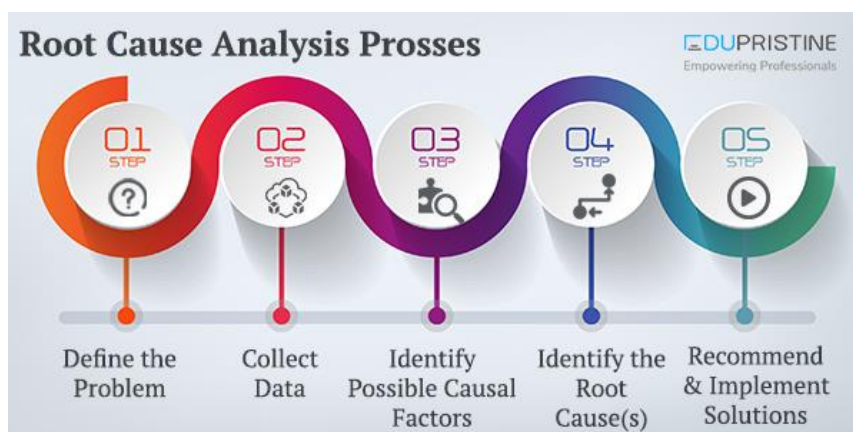
Μη προγραμματισμένες απώλειες

Οι απώλειες που συγκαταλέγονται σε αυτή τη κατηγορία, θεωρούνται ανεπιθύμητες και οι διαδικασίες Συνεχούς Βελτίωσης οφείλουν να εστιάσουν σε αυτές, προκειμένου να επιτύχουν αύξηση της απόδοσης. Αυτό που συμβαίνει συνήθως σε πραγματικές συνθήκες παραγωγής, είναι ότι οι χειριστές των γραμμών, καταγράφουν στα δελτία τους, τα **σταματήματα** που βλέπουν ή θυμούνται στο τέλος της βάρδιας τους. Επιπλέον, πολλές φορές, τα σταματήματα αυτά, θεωρούνται ηλεκτρομηχανολογικές βλάβες και κατά συνέπεια οι χειριστές καταλογίζουν ευθύνες στο τμήμα συντήρησης. Το τμήμα συντήρησης δικαιολογεί τις βλάβες ως αστοχία εξοπλισμού και εν τέλει τα σταματήματα αυτά, φαντάζουν αναπόφευκτα. Για τον λόγο αυτό, οι απώλειες που οφείλονται σε **μικροσταματήματα**, θεωρούνται ιδιαίτερα σημαντικά στοιχεία κατά την εφαρμογή μεθόδων Συνεχούς Βελτίωσης. Ωστόσο, σπάνια αποτελούν σημαντικό μέρος των συνολικών απωλειών. Σε κάθε περίπτωση, οι μη προγραμματισμένες απώλειες πρέπει να καταγράφονται συστηματικά και με λεπτομέρεια για κάθε έναν από τους έξι βασικούς παράγοντες που προαναφέρθηκε.

Προσδιορισμός απωλειών απόδοσης

Συνολική Θεωρητική Παραγωγή 24 ώρες x Θεωρητική Μέγιστη Ταχύτητα			
Προγραμματισμένος Χρόνος Παραγωγής Χρόνος Χρήσης Γραμμής Παραγωγής - LU			Προγραμματισμένο Χρόνος χωρίς Παραγωγή
Διαθεσιμότητα Γραμμής Παραγωγής Availability		Χρόνος Αλλαγής	
Απόδοση Γραμμής Performance		Απώλειες Ταχύτητας	
Ποιότητα	Παραγωγή ελαττωματικών		

2.3.3 Root Cause Analysis



Εικόνα 2.1: Root Cause Analysis

Το **Root Cause Analysis (RCA)** είναι η διαδικασία ανακάλυψης των βασικών αιτιών των προβλημάτων προκειμένου να εντοπιστούν οι κατάλληλες λύσεις. Το RCA υποθέτει ότι είναι πολύ πιο αποτελεσματική η συστηματική πρόληψη και επίλυση υποκείμενων ζητημάτων παρά η απλή θεραπεία ad hoc συμπτωμάτων. Η ανάλυση της βασικής αιτίας μπορεί να πραγματοποιηθεί με μια συλλογή αρχών, τεχνικών και μεθοδολογιών οι οποίες μπορούν όλες να αξιοποιηθούν για τον εντοπισμό των βασικών αιτιών ενός γεγονότος ή μιας τάσης. Εξετάζοντας πέρα από την επιφανειακή αιτία, το RCA μπορεί να δείξει πού απέτυχαν οι διεργασίες ή τα συστήματα ή που προκάλεσαν πρόβλημα εξαρχής. Ο **πρώτος στόχος** της ανάλυσης είναι να ανακαλύψει τη βασική αιτία ενός προβλήματος ή ενός συμβάντος. Ο **δεύτερος στόχος** είναι να κατανοηθούν πλήρως πώς να διορθώνεται ή να μαθαίνεται από τυχόν υποκείμενα ζητήματα στη βασική αιτία. Ο **τρίτος στόχος** είναι να εφαρμόζονται όσα διδάχθηκαν από αυτήν την ανάλυση για να αποτρέψουμε συστηματικά μελλοντικά προβλήματα. Ο **τέταρτος στόχος** του RCA είναι σημαντικός. Το RCA χρησιμοποιείται για να τροποποιηθούν επίσης ζητήματα βασικής διαδικασίας και συστήματος με τρόπο που να αποτρέπει μελλοντικά προβλήματα. Αντί να αντιμετωπίζονται απλώς τα προβλήματα μιας παραγωγικής διαδικασίας, για παράδειγμα, η ανάλυση της βασικής αιτίας μπορεί να προτείνει τη χρήση μεθόδων για τη μείωση του κινδύνου μελλοντικών εμφανίσεων βλάβης. Η αντιμετώπιση των μεμονωμένων συμπτωμάτων μπορεί να φαίνεται παραγωγική. Η επίλυση μεγάλου αριθμού προβλημάτων μοιάζει να είναι αποτελεσματική. Αλλά αν στην πραγματικότητα δεν διαγνωστεί η πραγματική βασική αιτία ενός προβλήματος, πιθανότατα θα εμφανιστεί το ίδιο ακριβώς πρόβλημα ξανά και ξανά. [10] [11]

Βασικές αρχές

Υπάρχουν μερικές βασικές αρχές που καθοδηγούν την αποτελεσματική ανάλυση της βασικής αιτίας, μερικές από τις οποίες θα πρέπει να είναι ήδη εμφανείς. Όχι μόνο βοηθούν στην ποιότητα της ανάλυσης, αλλά επίσης και τον αναλυτή να κερδίσει την εμπιστοσύνη και την ανταπόκριση από ενδιαφερόμενους φορείς, πελάτες. Συνειδητοποιήστε ότι μπορεί να υπάρχουν, και συχνά υπάρχουν, πολλαπλές βασικές αιτίες. Το RCA εστιάζει στο ΠΩΣ και ΓΙΑΤΙ συνέβη κάτι, όχι στο ΠΟΙΟΣ ήταν υπεύθυνος. Χρησιμοποιείται με μια γενική έννοια, υπάρχει μια συνέπεια ότι χρησιμοποιείται μια μεθοδολογία στην ανάλυση. Οι περισσότεροι ειδικοί του RCA πιστεύουν ότι η επίτευξη συνολικής πρόληψης με μία μόνο παρέμβαση δεν είναι πάντα δυνατή και βλέπουν το RCA ως μια συνεχή διαδικασία που προσπαθεί για συνεχή βελτίωση.

Στόχος

Ο πρωταρχικός **στόχος** της χρήσης RCA είναι η ανάλυση προβλημάτων ή συμβάντων για τον εντοπισμό:

- Τι συνέβη
- Πως έγινε
- Γιατί έγινε

Οφέλη

Το RCA εντοπίζει τα εμπόδια και τις αιτίες των προβλημάτων, με σκοπό να βρίσκουν μόνιμες λύσεις. Αναπτύσσει μια λογική προσέγγιση για την επίλυση προβλημάτων, χρησιμοποιώντας δεδομένα που υπάρχουν ήδη. Προσδιορίζει τις τρέχουσες και μελλοντικές ανάγκες για οργανωτική βελτίωση. Καθιερώνει επαναλαμβανόμενες, βήμα προς βήμα διαδικασίες, στις οποίες μια διαδικασία μπορεί να επιβεβαιώσει τα αποτελέσματα μιας άλλης.

Αρχές

Η εστίαση σε διορθωτικά μέτρα των βασικών αιτιών είναι πιο αποτελεσματική από την απλή αντιμετώπιση των συμπτωμάτων ενός προβλήματος ή ενός συμβάντος. Το RCA εκτελείται πιο αποτελεσματικά όταν επιτυγχάνεται μέσω μιας συστηματικής διαδικασίας με συμπεράσματα που υποστηρίζονται από στοιχεία. Συνήθως υπάρχουν περισσότερες από μία βασικές αιτίες για ένα πρόβλημα ή ένα συμβάν. Το επίκεντρο της έρευνας και της ανάλυσης μέσω της αναγνώρισης του προβλήματος είναι ΓΙΑΤΙ συνέβη το συμβάν και όχι ποιος έκανε το λάθος.

Ρίζες

Η ανάλυση της βασικής αιτίας γεννήθηκε από την ανάγκη να αναλυθούν διάφορες επιχειρηματικές δραστηριότητες της:

- Ανάλυση ατυχημάτων και επαγγελματική ασφάλεια και υγεία
- Έλεγχος ποιότητας
- Αποτελεσματική επιχειρηματική διαδικασία
- Ανάλυση αστοχίας μηχανής και συντήρησης
- Διάφορες διαδικασίες που βασίζονται σε συστήματα, συμπεριλαμβανομένης της διαχείρισης αλλαγών και της διαχείρισης κινδύνων

Εφαρμογές RCA

Παραδείγματα συμβάντων όπου το RCA χρησιμοποιείται για την επίλυση προβλημάτων και την παροχή προληπτικών ενεργειών περιλαμβάνουν:

- Μεγάλα ατυχήματα
- Καθημερινά περιστατικά
- Μικρά σταματήματα
- Ανθρώπινα λάθη
- Προβλήματα συντήρησης
- Θέματα παραγωγικότητας
- Λάθη κατασκευής
- Ανάλυση κινδύνου, χαρτογράφηση κινδύνου
- Βασική μέθοδος χρήσης

Μέθοδοι RCA

Η φύση του RCA είναι να αναγνωρίζει όλους τους παράγοντες που συμβάλλουν σε ένα πρόβλημα ή ένα συμβάν. Αυτό επιτυγχάνεται πιο αποτελεσματικά μέσω μιας μεθόδου ανάλυσης. Ορισμένες μέθοδοι που χρησιμοποιούνται στο RCA περιλαμβάνουν:

Ανάλυση «5-Γιατί» — Μια απλή τεχνική επίλυσης προβλημάτων που βοηθά τους χρήστες να βρουν γρήγορα τη ρίζα του προβλήματος. Αυτή η στρατηγική περιλαμβάνει την εξέταση ενός προβλήματος και την ερώτηση «γιατί» και «τι προκάλεσε αυτό το πρόβλημα». Συχνά η απάντηση στο πρώτο «γιατί» προκαλεί ένα δεύτερο «γιατί» και ούτω καθεξής—παρέχοντας τη βάση για την ανάλυση «5-γιατί».

Ανάλυση φραγμού — Μέθοδος σχεδιασμού που περιλαμβάνει τον εντοπισμό μονοπατιών μέσω των οποίων ένας στόχος επηρεάζεται αρνητικά από έναν κίνδυνο, συμπεριλαμβανομένης της αναγνώρισης τυχόν αποτυχημένων ή ελλειπόντων αντιμέτρων που θα μπορούσαν ή θα έπρεπε να είχαν αποτρέψει την ανεπιθύμητη επίδραση.

Ανάλυση Αλλαγών — Αναζητά συστηματικά πιθανές επιπτώσεις κινδύνου και κατάλληλες στρατηγικές διαχείρισης κινδύνου σε καταστάσεις όπου συμβαίνει αλλαγή. Αυτό περιλαμβάνει καταστάσεις στις οποίες αλλάζουν οι διαμορφώσεις του συστήματος, αναθεωρούνται οι πρακτικές ή οι πολιτικές λειτουργίας, θα εκτελεστούν νέες ή διαφορετικές δραστηριότητες κ.λπ.

Ανάλυση λειτουργίας και επιπτώσεων αστοχίας — Μια διαδικασία «μηχανικής συστήματος» που εξετάζει αστοχίες σε προϊόντα, διαδικασίες και μηχανήματα. [12]

2.4 Μοντελοποίηση – προσομοίωση γραμμών παραγωγής

Η **μοντελοποίηση** (η διαδικασία κατασκευής ενός μοντέλου) είναι μια πιο πλατιά έννοια από την προσομοίωση, στο μέτρο που κάθε προσομοίωση απαιτεί ένα μοντέλο, με γενικευμένη, έστω, ερμηνεία. Το μοντέλο που προσομοιώνεται μπορεί να είναι διαφορετικό από ένα αναλυτικό ή αριθμητικό ή και συμβολικό μοντέλο του συστήματος. Ο δόκιμος όρος που περιγράφει ακριβώς αυτή τη δυαδικότητα είναι "μοντελοποίηση και προσομοίωση" (modeling and simulation).

Υπάρχει επίσης και ο χαρακτηρισμός **hardware-in-the-loop** που μπορεί να μεταφρασθεί ελεύθερα "υλικό στον βρόχο". Με τον όρο αυτό υπονοούμε ότι σε ένα τμήμα της προσομοίωσης διαδικασίας χρησιμοποιούμε το πραγματικό υλικό ή διάταξη που θα εμφανιστεί και στο τελικό σύστημα. Με αυτόν τον τρόπο μπορούμε να παρακάμψουμε ορισμένα προβλήματα μοντελοποίησης υποσυστημάτων που διαφορετικά θα ήταν υπερβολικά χρονοβόρα ή δύσκολα.

Τελευταία εμφανίστηκε και ο όρος **simulation-in-the-loop**, δηλαδή "προσομοίωση στον βρόχο" ο οποίο αντιστοιχεί σε μια κατάσταση όπου σε ένα ή περισσότερα τμήματα ενός μεγάλου συστήματος χρησιμοποιείται η προσομοίωση κάποιου υποσυστήματος για τη λήψη αποφάσεων ή για καθοδήγηση. Χαρακτηριστικό παράδειγμα αποτελεί η ρομποτική όπου κάποιος προσομοιωτής κινηματικής χρησιμοποιείται για να σχεδιασθεί ή/και να ελεγχθεί η διαδρομή που πρόκειται να ακολουθήσει το ρομπότ για την εκτέλεση ενός δύσκολου συνήθως έργου. Αν η διαδρομή είναι ικανοποιητική (πιθανόν μετά από αλλαγές) μεταφράζεται η τροχιά που προσομοιώνεται στη γλώσσα του ρομπότ και εκτελείται στην πραγματικότητα.

Σημαντικό πλεονέκτημα που προσφέρει η κατασκευή ενός **μοντέλου** αποτελεί η δυνατότητα για μελέτη και πρόβλεψη της συμπεριφοράς κι αποδοτικότητας του συστήματος κάτω από ακραίες συνθήκες λειτουργίας που για διάφορους λόγους δεν μπορούν να εφαρμοστούν σε αυτό (λόγω της ύπαρξης κινδύνων, μεγάλου κόστους ή ακόμα και της πρόκλησης αντιδράσεων) ή σε περιπτώσεις που το φυσικό σύστημα δεν είναι άμεσα διαθέσιμο. [13] [14]

2.4.1 Η διαδικασία της μοντελοποίησης

Η διαδικασία της μοντελοποίησης αποτελείται από 5 βήματα:

1. Αναγνώριση των συστατικών μερών του μοντέλου
2. Συγκέντρωση πληροφοριών για τις μεταξύ τους σχέσεις, προτεραιότητες, τρόπους σύνδεσης και λειτουργίας, χαρακτηριστικά, χρήση κοινών πόρων
3. Κατασκευή μοντέλου
4. Δοκιμή του μοντέλου κάνοντας χρήση το ίδιο
5. Τροποποίηση του μοντέλου αν αυτό θεωρηθεί αναγκαίο κι ορισμός της αρχικής του κατάστασης

Τα πέντε βήματα δεν αποτελούν γραμμική διαδικασία, αφού κάθε ένα τροφοδοτεί τα επόμενα κι ανατροφοδοτεί τα προηγούμενα, οδηγώντας σε αρκετές επαναλήψεις τον αλγόριθμο μέχρι το μοντέλο να λάβει την οριστική του μορφή.

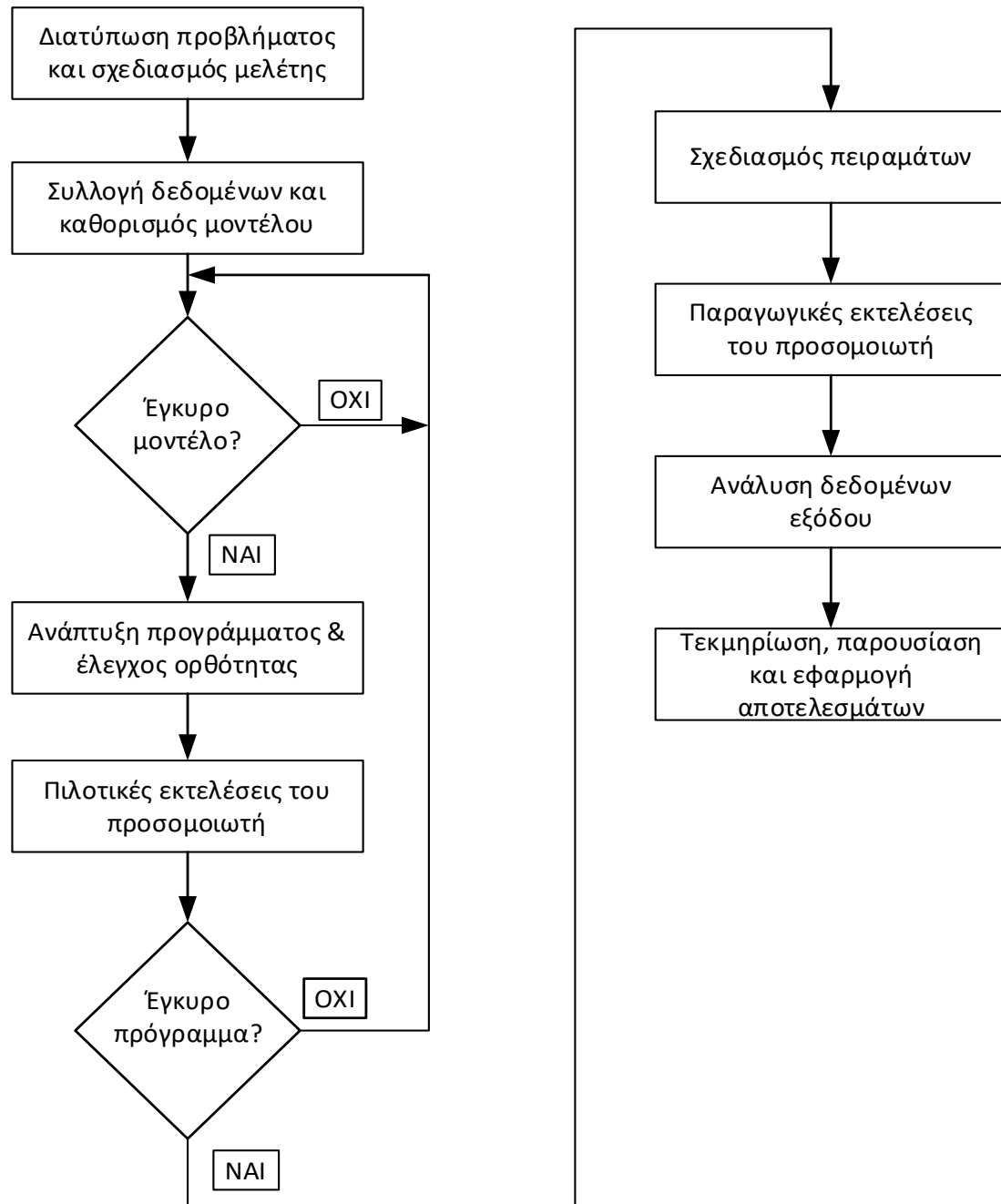
Επιπλέον, η τελική μορφή αναθεωρείται:

- οποτεδήποτε προκύψουν νέα δεδομένα
- όταν το μοντέλο αποδειχθεί ανεπαρκές για την περιγραφή καταστάσεων
- όταν επεκταθεί ή τροποποιηθεί η χρήση του συστήματος

Η διαδικασία της μοντελοποίησης χωρίζεται σε δυο στο στάδια: το στάδιο κατασκευής και το στάδιο προσομοίωσης του μοντέλου. Στο πρώτο, δημιουργείται και σχεδιάζεται η βασική δομή κι ορίζονται οι κανόνες του μοντέλου για να του δώσουν την μορφή που μπορεί να περιγράψει τις καταστάσεις που συναντώνται στο σύστημα. Στη συνέχεια το μοντέλο πρέπει να αναλυθεί με χρήση κατάλληλων εργαλείων και να προσδιοριστούν οι ιδιότητες του, ώστε να αποφευχθούν λάθη που δυσχεραίνουν τη χρήση του και μειώνουν την αποτελεσματικότητά του. Κατόπιν, το μοντέλο διερευνάται κι εξετάζεται κατά πόσον η συμπεριφορά του αντιστοιχεί σε αυτή του πραγματικού συστήματος. Κάποια επιμέρους χαρακτηριστικά του μπορούν να τροποποιηθούν, αλλά η βασική του δομή παραμένει αναλλοίωτη.

Ο προσομοιωτής μπορεί να εφαρμοστεί σε πάρα πολλούς τομείς όπως στη βιομηχανία, το χώρο της υγείας αλλά και στον εμπορικό, περιβαλλοντικό και τον τομέα των κατασκευών. Για τα δεδομένα όμως της διπλωματικής η χρήση του μοντέλου περιορίζεται μόνο στο τομέα της βιομηχανίας και ειδικότερα στις γραμμές παραγωγής. Οι βιομηχανικές και κατασκευαστικές δραστηριότητες χαρακτηρίζονται από αυξημένο βαθμό πολυπλοκότητας και αβεβαιότητας και γι' αυτό η προσομοίωση θεωρείται ένα χρήσιμο εργαλείο ανάλυσης που προσφέρει αυξημένες δυνατότητες για τον χειρισμό σύνθετων δραστηριοτήτων. Άλλωστε η πολυπλοκότητα των τεχνικών και βιομηχανικών έργων είναι και ο λόγος που η προσομοίωση προτιμάται αντί αναλυτικών μαθηματικών μεθόδων, που αν και ευκολότερες στη χρήση και στην κατανόηση, δυσκολεύονται στο να απεικονίσουν το σύνολο των κατασκευαστικών δραστηριοτήτων. [15] [16]

Η διαδικασία μοντελοποίησης



2.4.2 Process based models

Ένα μοντέλο που βασίζεται σε διαδικασίες είναι η μαθηματική (και συνήθως βασισμένη σε υπολογιστή) αναπαράσταση μιας ή περισσότερων διαδικασιών που χαρακτηρίζουν τη λειτουργία καλά οριοθετημένων συστημάτων. Για παράδειγμα η αναπαράσταση της ροής μιας βιομηχανικής παραγωγικής διαδικασίας αποτελεί μοντέλο βασισμένο στη διεργασία. Συνήθως, τέτοια μοντέλα αποτελούνται από ένα σύνολο συνηθισμένων ή μερικών διαφορικών εξισώσεων που καθορίζουν την ουσία κάθε διεργασίας (χρονικά μοτίβα βασικών παραμέτρων), καθώς και τις εισόδους και τις εξόδους τους, ως συνάρτηση των πρώτων αρχών (generic PBM) ή αλλιώς εμπειρικές γνώση (descriptive PBM). Οι έξοδοι μιας διεργασίας μπορούν να χρησιμεύσουν ως είσοδος σε άλλες διεργασίες. Ένα τέτοιο

παράδειγμα μοντελοποίησης χρησιμοποιείται σε μεγάλο βαθμό στο πλαίσιο της διπλωματικής για τη μοντελοποίηση βιομηχανικής παραγωγής: Ένα μοντέλο παραγωγικής διαδικασίας που περιγράφει την ροή, την τροφοδοσία, τη συμπεριφορά και τη λειτουργία των μηχανημάτων της παραγωγής. [17] [18]

Ψηφιακό Δίδυμο

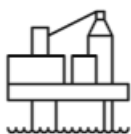
Το ψηφιακό δίδυμο λειτουργεί ως γέφυρα μεταξύ του φυσικού και του ψηφιακού κόσμου. Έξυπνα εξαρτήματα που χρησιμοποιούν αισθητήρες για τη συλλογή δεδομένων σχετικά με την κατάσταση των μηχανημάτων, τις συνθήκες εργασίας ή τη θέση σε πραγματικό χρόνο ενσωματώνονται με ένα φυσικό στοιχείο. Τα εξαρτήματα συνδέονται με ένα σύστημα που βασίζεται στο cloud. Το cloud αντίστοιχά λαμβάνει και επεξεργάζεται όλα τα δεδομένα που παρακολουθούν οι αισθητήρες. Αυτά τα δεδομένα αναλύονται με βάση την επιχείρηση. [19]

Εφαρμογές ψηφιακών διδύμων



Εξοπλισμό παραγωγής ηλεκτρικής ενέργειας

Οι μεγάλοι κινητήρες, συμπεριλαμβανομένων των κινητήρων αεροπλάνων, των κινητήρων ατμομηχανών και των τουρμπινών παραγωγής ενέργειας, επωφελούνται σε μεγάλο βαθμό από τη χρήση των ψηφιακών διδύμων. Ειδικότερα στον καθορισμό χρονοδιαγραμμάτων για την τακτική συντήρηση.



Δομές και τα συστήματά τους

Μεγάλες φυσικές κατασκευές, όπως μεγάλα κτίρια ή πλατφόρμες γεώτρησης ανοιχτής θαλάσσης, μπορούν να βελτιωθούν μέσω ψηφιακών διδύμων, ιδιαίτερα κατά τη διάρκεια του σχεδιασμού τους. Επίσης συμβάλει στο σχεδιασμό των συστημάτων που λειτουργούν εντός αυτών των δομών, όπως για παράδειγμα στα συστήματα HVAC.



Λειτουργίες βιομηχανίας

Δεδομένου ότι τα ψηφιακά δίδυμα έχουν σαν σκοπό να αντικατοπτρίζουν ολόκληρο τον κύκλο ζωής ενός προϊόντος, δεν προκαλεί έκπληξη το γεγονός ότι έχουν γίνει πανταχού παρόντα σε όλα τα στάδια της κατασκευής μιας μηχανής ή ενός μηχανήματος, καθοδηγώντας τα προϊόντα από το σχεδιασμό στο τελικό προϊόν και όλα τα ενδιάμεσα βήματα.



Αυτοκινητοβιομηχανία

Τα αυτοκίνητα αντιπροσωπεύουν πολλούς τύπους πολύπλοκων, πολλαπλών λειτουργιών συστημάτων και τα ψηφιακά δίδυμα χρησιμοποιούνται εκτενώς στον σχεδιασμό αυτοκινήτων, τόσο για τη βελτίωση της απόδοσης του οχήματος όσο και για την αύξηση της αποδοτικότητας που περιβάλλει την παραγωγή τους.



Πολεοδομικός σχεδιασμός

Οι πολιτικοί μηχανικοί και άλλοι αρμόδιοι που εμπλέκονται σε δραστηριότητες πολεοδομικού σχεδιασμού βοηθούνται σημαντικά από τη χρήση ψηφιακών διδύμων, τα οποία μπορούν να εμφανίσουν 3D και 4D χωρικά δεδομένα σε πραγματικό χρόνο και επίσης να ενσωματώσουν συστήματα επαυξημένης πραγματικότητας σε δομημένα περιβάλλοντα.

Το μέλλον του ψηφιακού δίδυμου

Είναι σαφές ότι διαδραματίζεται μια θεμελιώδης αλλαγή στα υπάρχοντα μοντέλα λειτουργίας. Ένα ψηφιακό δίδυμο λαμβάνει χώρα σε βιομηχανίες που αλλάζει τα μοντέλα λειτουργίας με αποδιοργανωτικό τρόπο, απαιτώντας έτσι μια ολοκληρωμένη φυσική και ψηφιακή προβολή των δεδομένων, του εξοπλισμού, των εγκαταστάσεων και των διαδικασιών. Τα ψηφιακά δίδυμα είναι ένα σημαντικό μέρος αυτής της ευθυγράμμισης.

Το μέλλον των ψηφιακών διδύμων είναι σχεδόν άπειρο, λόγω του γεγονότος ότι αυξανόμενα ποσά γνωστικής δύναμης αφιερώνονται συνεχώς στη χρήση τους. Έτσι, τα ψηφιακά δίδυμα μαθαίνουν συνεχώς νέες δεξιότητες και δυνατότητες, πράγμα που σημαίνει ότι μπορούν να συνεχίσουν να δημιουργούν τις απαραίτητες γνώσεις για να κάνουν τα προϊόντα καλύτερα και τις διαδικασίες πιο αποτελεσματικές. [20]

2.4.3 Data driven models

Τα μοντέλα που βασίζονται σε δεδομένα δημιουργούν σχέσεις μεταξύ δεδομένων εισόδου και εξόδου, χωρίς να δίνουν σημασία στις υποκείμενες διαδικασίες, χρησιμοποιώντας παράλληλα τεχνικές στατιστικής/μηχανικής μάθησης. Από την άλλη πλευρά, τα μοντέλα που βασίζονται στη διαδικασία βασίζονται σε καλά καθιερωμένους μαθηματικούς/φυσικούς νόμους.

Ένα μοντέλο γραμμικής παλινδρόμησης είναι ένα παράδειγμα μοντέλου βάσει δεδομένων που, για παράδειγμα, δημιουργεί μια σχέση μεταξύ μιας εξαρτημένης μεταβλητής και ενός συνόλου ανεξάρτητων μεταβλητών. Αυτό το μοντέλο επαναφοράς μπορεί στη συνέχεια να χρησιμοποιηθεί για την κατανόηση της σχέσης μεταξύ των μεταβλητών και σε ορισμένες περιπτώσεις μπορεί επίσης να χρησιμοποιηθεί για να κάνει προβλέψεις. [21]

Μοντέλα

Model	Description	Type	Method	Complexity
Digital Twin	Digital equivalent	Process-based	Digital equations	High
HTESSEL	ECMWF land surface scheme	Physically	Differential equations	High
SWBM	Simple Water Balance Model	Conceptual	Simplified equations	Medium
LSTM-Runoff	LSTM-based runoff model	Data-driven	Black-box concept	Low

Θετικά & αρνητικά

Τα μοντέλα που βασίζονται σε δεδομένα έχουν το πλεονέκτημα να ενσωματώνονται σε ακόμη και σε περιπτώσεις σφάλματος. Ένας μεγάλος όγκος δεδομένων χρησιμοποιείται για την εκτίμηση των παραμέτρων για την προσαρμογή του μοντέλου μεταξύ των δεδομένων εισόδου και εξόδου. Τα σφάλματα μπορούν να ποσοτικοποιηθούν και τα επίπεδα εμπιστοσύνης του μοντέλου μπορούν να εκτιμηθούν. Από την άλλη πλευρά, τα μοντέλα που βασίζονται στη διαδικασία έχουν κατασκευαστεί χρησιμοποιώντας υποκείμενη φυσική. Εδώ τα σφάλματα μπορεί να εισαχθούν μέσω αβέβαιων αρχικών/οριακών συνθηκών. Εδώ χρησιμοποιούνται οι πραγματικές παρατηρήσεις για την αξιολόγηση του μοντέλου.

2.4.4 Software Process Simulator

Μοντελοποίηση προσομοίωσης διεργασιών λογισμικού

Όπως κάθε προσομοίωση, η προσομοίωση διαδικασίας λογισμικού (Software Process Simulation - SPS) είναι η αριθμητική αξιολόγηση ενός μαθηματικού μοντέλου που μιμείται τη συμπεριφορά της διαδικασίας ανάπτυξης λογισμικού που μοντελοποιείται. Το SPS έχει τη δυνατότητα να μοντελοποιεί τη δυναμική φύση της ανάπτυξης λογισμικού και να χειρίζεται την αβεβαιότητα και την τυχαιότητα που ενυπάρχουν σε αυτό. [22]

Διαφορά λογισμικού προσομοίωσης διεργασιών και λογισμικό προσομοίωσης

Αυτοί οι όροι είναι συνώνυμοι. Τεχνικά, μια προσομοίωση εκτελείται μέσα σε έναν προσομοιωτή, αλλά δεδομένου ότι όλα τα πακέτα λογισμικού προσομοίωσης διεργασιών περιλαμβάνουν τη δυνατότητα τόσο για τη δημιουργία μιας διαδικασίας προσομοίωσης όσο και για την εκτέλεσή της, η διάκριση είναι άσχετη.

Είδος διαδικασιών προσομοίωσης

Μπορούν να μοντελοποιηθούν πολλά διαφορετικά είδη διαδικασιών. Οι χημικοί μηχανικοί μπορούν να προσομοιώσουν τη βιομηχανική παραγωγή χημικών. Οι βιολόγοι μπορούν να σχεδιάσουν βιοαντιδραστήρες για βιολογικά μόρια. τα διυλιστήρια πετρελαίου μπορούν να εκτελούν μοντελοποίηση διυλιστηρίων. Οι μηχανικοί πληροφορικής μπορούν να σχεδιάσουν συστήματα βιομηχανικών παραγωγών και διαδικασίες για μεγάλες βιομηχανίες. Ο κατάλογος των προσομοιώσεων διεργασιών που μπορούν να εκτελεστούν στο λογισμικό προσομοίωσης διεργασιών αυξάνεται μέρα με τη μέρα. [23]

Λογισμικά προσομοίωσης

Οι αποφάσεις σχετικά με την ανάπτυξη, τη βελτιστοποίηση ή την αναδιοργάνωση της παραγωγής καθορίζονται από πολλούς παράγοντες και είναι συχνά δαπανηρές, με τα οφέλη που είναι δύσκολο να δικαιολογηθούν πριν από την εφαρμογή. Παραδοσιακά, οι αποφάσεις λαμβάνονται με βάση τη διαίσθηση και την εμπειρία, μερικές φορές με την υποστήριξη εργαλείων υπολογιστικών φύλλων. Αυτές οι προσεγγίσεις μπορεί να είναι επικίνδυνες και περιττές στη λήψη αποφάσεων σήμερα. Η προσομοίωση είναι μια ισχυρή τεχνική για την ανάλυση των συστημάτων παραγωγής, την αξιολόγηση του αντίκτυπου των αλλαγών του συστήματος και για τη λήψη τεκμηριωμένων αποφάσεων.

Κορυφαία εταιρεία στη δημιουργία λογισμικού προσομοίωσης αποτελεί η **Siemens**. Κατά την Siemens το λογισμικό προσομοίωσης διεργασιών **Tecnomatix** [24] διευκολύνει το σχεδιασμό και την επικύρωση των διαδικασιών παραγωγής σε ένα τρισδιάστατο δυναμικό περιβάλλον. Είναι πλήρως ενσωματωμένο με τη κατασκευή, επιτρέποντας έτσι στους μηχανικούς να επαναχρησιμοποιήσουν, να συντάξουν και να επικυρώσουν τις διαδικασίες παραγωγής. Η προσομοίωση παρέχει ένα προηγμένο τρισδιάστατο περιβάλλον ικανό να μιμηθεί τη ρεαλιστική συμπεριφορά των διαδικασιών παραγωγής και να βελτιστοποιήσει τους χρόνους κύκλων και τη σειρά διεργασιών. Διευκολύνει την προσομοίωση διαδικασιών συναρμολόγησης, ανθρώπινων λειτουργιών και μηχανικών διαδικασιών εργαλείων, συσκευών και ρομπότ. Τέλος, είναι εξαιρετικά επεκτάσιμο, παρέχοντας σε διάφορους κλάδους μηχανικής δεδομένα και εργαλεία για την εξέταση λεπτομερών διαδικασιών και την επαλήθευσή τους σε διαφορετικές φάσεις και προοπτικές.

Επιπλέον σημαντικό πρωταγωνιστή στην δημιουργία λογισμικού προσομοίωσης αποτελεί η **Aspentech**. Το **ASPEN** [25] είναι ένα πακέτο λογισμικού προσομοίωσης διεργασιών που χρησιμοποιείται ευρέως στη βιομηχανία. Δεδομένου ενός σχεδιασμού διαδικασίας και μιας κατάλληλης επιλογής θερμοδυναμικών μοντέλων, το ASPEN χρησιμοποιεί μαθηματικά μοντέλα για να προβλέψει την απόδοση της διαδικασίας. Αυτές οι πληροφορίες μπορούν στη συνέχεια να χρησιμοποιηθούν με επαναληπτικό τρόπο για τη

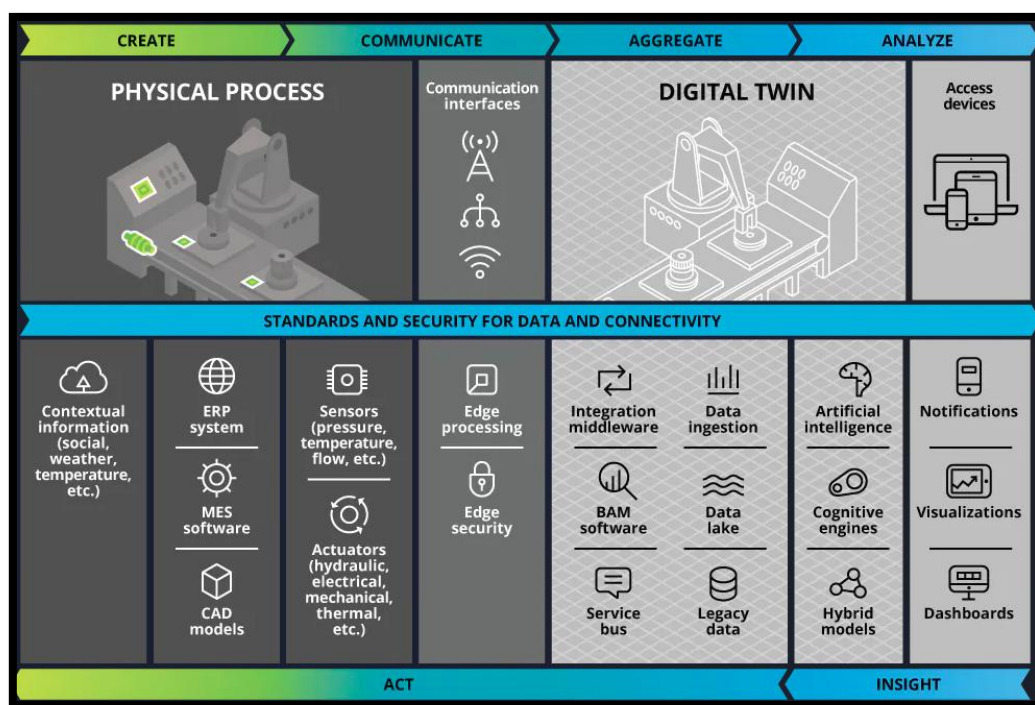
βελτιστοποίηση του σχεδιασμού. Αυτή η ακριβής μοντελοποίηση των θερμοδυναμικών ιδιοτήτων είναι ιδιαίτερα σημαντική για τον διαχωρισμό μη ιδανικών μιγμάτων και το ASPEN διαθέτει μεγάλες βάσεις δεδομένων με παλινδρομημένες παραμέτρους. Το ASPEN μπορεί να χειριστεί πολύ περίπλοκες διεργασίες, όπως συστήματα διαχωρισμού πολλαπλών στηλών, χημικούς αντιδραστήρες, απόσταξη χημικά αντιδραστικών ενώσεων, ακόμη και διαλύματα ηλεκτρολυτών όπως ορυκτά οξέα και διαλύματα υδροξειδίου του νατρίου. [26]

Τέλος, κατά την **AnyLogic** [27] συγκεκριμένες διαδικασίες και στρατηγικές, όπως το JIT ή το Lean, μπορούν να μοντελοποιηθούν και να προσομοιωθούν στο λογισμικό προσομοίωσης κατασκευής. Αυτό επιτρέπει την αποτελεσματική ανάλυση και παρέχει έναν αποτελεσματικό τρόπο πειραματισμού και μείωσης του κόστους των δοκιμών στον πραγματικό κόσμο. Το AnyLogic βοηθά στη βελτίωση των διαδικασιών παραγωγής. Είναι ένα ισχυρό λογισμικό προσομοίωσης διεργασιών με μια ειδικά σχεδιασμένη βιβλιοθήκη χειρισμού υλικού που απλοποιεί την προσομοίωση πολύπλοκων συστημάτων και λειτουργιών παραγωγής. Η βιβλιοθήκη βοηθά κατά το σχεδιασμό λεπτομερών μοντέλων εγκαταστάσεων παραγωγής και τη διαχείριση ροών εργασίας υλικού.

Η χρήση λογισμικού προσομοίωσης διαδικασίας είναι το **κλειδί** για τη λεπτομερή ανάλυση και την αποτελεσματική λειτουργία μια βιομηχανικής γραμμής παραγωγής.

2.4.5 Digital twin vs process simulations

Η τεχνολογία **Digital Twin** χρησιμοποιεί live data από αισθητήρες για να τα ανεβάσει σε ένα live model του digital twin software, ενώ το λογισμικό προσομοίωσης διαδικασίας δεν παρακολουθεί απαραίτητα την πραγματική ροή των δεδομένων σε πραγματικό χρόνο. Επιπρόσθετα το Ψηφιακό Δίδυμο αναπαριστά το φυσικό αντικείμενο σε λογισμικό. Το λογισμικό προσομοίωσης διαδικασίας απλώς αναπαράγει την εξωτερική λειτουργία της οντότητας (αντικειμένου).



Εικόνα 2.2: Digital Twin vs Physical Process

Και τα ψηφιακά δίδυμα και οι προσομοιώσεις χρησιμοποιούν ψηφιακά μοντέλα για να αναπαράγουν τις διάφορες διαδικασίες ενός συστήματος. Ένα ψηφιακό δίδυμο στην

πραγματικότητα είναι ένα εικονικό περιβάλλον, γεγονός που το καθιστά πολύ πιο πλούσιο για μελέτη. Η διαφορά μεταξύ ψηφιακού δίδυμου και προσομοίωσης είναι σε μεγάλο βαθμό θέμα κλίμακας: ενώ μια προσομοίωση συνήθως μελετά μια συγκεκριμένη διαδικασία, ένα ψηφιακό δίδυμο μπορεί να εκτελέσει οποιονδήποτε αριθμό χρήσιμων προσομοιώσεων προκειμένου να μελετήσει πολλαπλές διεργασίες. [28]

Οι διαφορές δεν σταματούν εκεί. Για παράδειγμα, οι προσομοιώσεις συνήθως δεν ωφελούνται από την ύπαρξη δεδομένων σε πραγματικό χρόνο. Αλλά τα ψηφιακά δίδυμα σχεδιάζονται γύρω από μια αμφίδρομη ροή πληροφοριών κατά την οποία οι αισθητήρες παρέχουν σχετικά δεδομένα στον επεξεργαστή του συστήματος. Η ροή συνεχίζει όταν ξανά δημιουργούνται πληροφορίες οι οποίες κοινοποιούνται στον αρχικό επεξεργαστή.

Συνδυάζοντας τα καλύτερα και συνεχώς ενημερωμένα δεδομένα που σχετίζονται σε ένα ευρύ φάσμα περιοχών ενδιαφέροντος με την υπολογιστική ισχύ που συνοδεύει ένα εικονικό περιβάλλον, τα ψηφιακά δίδυμα είναι σε θέση να μελετούν περισσότερα ζητήματα από πολύ πιο σημαντικά σημεία από ό,τι μπορούν οι τυπικές προσομοιώσεις. Με αυτό το τρόπο συμβάλουν με σημαντικά αποτελέσματα στη βελτίωση της απόδοσης των προϊόντων και των διαδικασιών. [29]

2.5 Καινοτομία παρούσας εργασίας

Η αξία των ψηφιακών δεδομένων και η προσομοίωση είναι το μοντέλο αναπαράστασης των μηχανημάτων που «μιμείται» την συμπεριφορά της γραμμής παραγωγής. Αποτέλεσμα είναι να αναδεικνύεται το πραγματικό μέγεθος της **αξίας του χρόνου** σε σχέση με το πρόβλημα. Οι χειριστές συνειδητοποίησαν ότι το πλήθος των σταματημάτων σε σχέση με τη συνολική διάρκεια του παραγωγικού χρόνου έχει καθοριστική αξία.

Το σύστημα αλλάζει στους χρήστες τον τρόπο σκέψης και τον τρόπο που αντιμετωπίζουν την βαρύτητα του προβλήματος για να δώσουν βέλτιστες λύσεις με αποτέλεσμα να αυξήσουν τους δείκτες απόδοσης της παραγωγής.

Αυτή η **τεχνολογική καινοτομία**, με τον υπολογισμό επιδόσεων του συστήματος, την ανάλυση της ικανότητας διεκπεραίωσης, την αξιολόγηση στρατηγικών ελέγχου του εξοπλισμού, τον υπολογισμό του μέσου χρόνου αναμονής μέρος του προϊόντος στην ουρά των μηχανών σε σχέση με τις μεταφορικές γραμμές, έχει ως αποτέλεσμα να μειώνει τους χρόνους και την συχνότητα σταματήματος λειτουργίας των μηχανημάτων. Αυξάνει τους δείκτες παραγωγικότητας (**KPIs**), μειώνει τα κόστη συντήρησης, μειώνει τα παραγωγικά και τα ενεργειακά κόστη **με αποτέλεσμα να αυξάνονται τα οικονομικά οφέλη**.

3. Σχεδιασμός του συστήματος

Σε αυτό το Κεφάλαιο καταγράφονται οι βασικές παράμετροι του προσομοιωτή και τα εργαλεία ανάπτυξης του συστήματος. Πιο συγκεκριμένα, στο κεφάλαιο 2 έγινε μια πολύ σύντομη βιβλιογραφική ανασκόπηση των λογισμικών προσομοίωσης, των συστημάτων και των γραμμών παραγωγής. Για την υλοποίηση του αλγορίθμου προσομοίωσης χρησιμοποιήθηκε η βιβλιοθήκη SimPy [30]. Η **SimPy** αποτελεί μια από τις ολοκληρωμένες βιβλιοθήκες προσομοίωσης που μπορεί να εφαρμοστεί σε κάθε παραγωγική διαδικασία οποιασδήποτε κατηγορίας.

Στη συνέχεια, παρουσιάζονται όλα τα εργαλεία και οι τεχνικές που χρησιμοποιήθηκαν για την υλοποίηση και τη σχεδίαση του συστήματος προσομοίωσης.

3.1 Παράμετροι συστήματος

Σημαντικό για την ορθή λειτουργία του προσομοιωτή είναι ο καθορισμός των παραμέτρων. Οι παράμετροι έχουν τον καθοριστικό ρόλο στην εκτέλεση του προγράμματος και είναι απαραίτητο να καθορίζονται προτού εκτελεστεί. Για την σχεδίαση ενός προσομοιωτή οι **Descriptors** (παράμετροι περιγραφής μηχανημάτων) χωρίζονται σε τρεις κατηγορίες. Η πρώτη κατηγορία είναι το Simulation Level που αφορά κυρίως τις παραμέτρους και τις ρυθμίσεις αποκλειστικά και μόνο του προσομοιωτή, η δεύτερη κατηγορία είναι το Process Level που αφορά τα μηχανήματα και τις οντότητες μιας παραγωγικής διαδικασίας και τέλος η κατηγορία UI - RCA Level που αφορά την παραμετροποίηση του γραφικού περιβάλλοντος.

3.1.1 Simulation level

Στην κατηγορία αυτή ορίζονται οι μεταβλητές που είναι υπεύθυνες για την εκτέλεση του προσομοιωτή. Κάθε προσομοίωση έχει τέσσερις βασικές σταθερές οι οποίες αντικατοπτρίζουν την πραγματική παραγωγική διαδικασία. Παρακάτω περιγράφονται αναλυτικά:

- **SIM_TIME:** Αποτελεί την πιο σημαντική παράμετρο της προσομοίωσης. Είναι ο συνολικός θεωρητικός χρόνος που διαρκεί κάθε εκτέλεση προσομοίωσης, δηλαδή ο χρόνος που διαρκεί η παραγωγική διαδικασία σε πραγματικές συνθήκες. Η μονάδα μέτρησης ορίζεται σε δευτερόλεπτα. Ο χρήστης ορίζει τη τιμή της σύμφωνα με το χρόνο (HOURS) που επιθυμεί να εκτελέσει την παραγωγική διαδικασία και έπειτα η μεταβλητή SIM_TIME υπολογίζεται σε δευτερόλεπτα.
- **Hours:** Είναι η παράμετρος που ορίζει ο χρήστης πριν την εκτέλεση της κάθε προσομοίωσης. Η μεταβλητή καταχωρεί τον χρόνο προσομοίωσης σε ώρες. Ο χρήστης εισάγει τις ώρες επιθυμεί να τρέξει ο προσομοιωτής σύμφωνα με μια πραγματική γραμμή παραγωγής.
- **DT:** Είναι ο διακριτός χρόνος βήματος (Discrete Time) αποτύπωσης των μετρικών. Ορίζεται από το χρήστη και σηματοδοτεί την περίοδο με την οποία ο simulator

εμφανίζει τα αποτελέσματα στο γραφικό περιβάλλον. Η μονάδα μέτρησης είναι ίδια με τη μονάδα μέτρησης της μεταβλητής SIM_TIME (seconds). Αποτελεί τον ρυθμό ανανέωσης του γραφικού περιβάλλοντος. Όσο πιο μικρή η τιμή της τόσο πιο γρήγορα αλλάζουν οι τιμές του προσομοιωτή στο UI και η ακρίβεια αυξάνεται. Ενώ όσο αυξάνεται η τιμή της, τόσο πιο αργά ανανεώνεται το γραφικό περιβάλλον.

- **PROD_PER_HOUR:** Κάθε παραγωγή έχει μια προκαθορισμένη ταχύτητα παραγωγής σύμφωνα με την οποία τρέχει μια ιδανική γραμμή παραγωγής. Η μεταβλητή αποτελεί τον αριθμό των προϊόντων που παράγονται κάθε μια ώρα. Η τιμή της μεταβλητής ορίζει την τιμή της PRODUCTION_SPEED που είναι ο ρυθμός παραγωγής μιας ακολουθίας διαδικασιών. Δηλαδή η ταχύτητα της γραμμής παραγωγής.
- **EXPECTED_PRODS:** Είναι ο αριθμός των τελικών προϊόντων που θα παρήγαγε μια ιδανική γραμμή παραγωγής χωρίς την εμφάνιση δυσλειτουργιών ή προβλημάτων. Ο αριθμός αυτός καθορίζεται από τον χρόνο που έχει επιλέξει ο χρήστης να λειτουργήσει η γραμμή παραγωγής. Χρησιμοποιείται για την τελική εκτίμηση των προϊόντων που παράχθηκαν σε μία πραγματική γραμμή σε συνδυασμό με την ιδανική. Υπολογίζεται από την ταχύτητα της παραγωγής (πχ 60.000 cans/hour) επί την τις ώρες παραγωγής που εισάγει ο χρήστης.

3.1.2 Process level

Η υποκατηγορία Process Level είναι υπεύθυνη για την ορθή λειτουργία κάθε μηχανήματος – οντότητας. Στην κατηγορία των μηχανημάτων συγκαταλέγεται κάθε είδους μηχανήμα μιας παραγωγικής διαδικασίας που μπορεί να αναπαρασταθεί ψηφιακά καθώς και οι μεταφορικές γραμμές, τα υποσυστήματα και οι μικρο-ελεγκτές.

- **Initial_product:** είναι η πρώτη ύλη η οποία είναι απαραίτητη για την εκκίνηση της προσομοίωσης και γενικά μιας γραμμής παραγωγής. Στον αλγόριθμο η τιμή της μπορεί είτε να είναι άπειρη εκτιμώντας πως υπάρχει πληθώρα πρώτης ύλης, είτε μπορεί να είναι πεπερασμένη και η τιμή της να μεταβάλλεται από την κατανάλωση και το απόθεμα των υπόλοιπων μηχανημάτων της παραγωγής.

Στη συνέχεια, ορίζονται οι βασικές μεταβλητές – σταθερές που πρέπει να έχει κάθε οντότητα στον αλγόριθμο με σκοπό να προσομοιώνει τη πραγματική παραγωγική διαδικασία. Στην ονομασία **machine** ορίζεται και τοποθετείται από τον προγραμματιστή το όνομα του μηχανήματος που αναπαριστά. Η κατηγορία **process level** διαχωρίζεται σε 3 υποκατηγορίες για την κατανόηση της λειτουργίας και την ευκολία της αρχικοποίησης των μηχανημάτων. Η πρώτη υποκατηγορία είναι οι **standard vars**, αποτελούν τις σταθερές μεταβλητές που ορίζει ο κατασκευαστής του μηχανήματος και περνάνε στο πρόγραμμα απαθείας από το προγραμματιστή. Η δεύτερη υποκατηγορία είναι οι **dynamics vars** που αφορούν τις δυναμικές μεταβλητές των μηχανημάτων και προκαθορίζονται σύμφωνα με τις μεταβλητές του προσομοιωτή. Και τέλος η τρίτη υποκατηγορία ορίζει το **status** των μηχανημάτων το οποία θα αναλυθεί ενδελεχώς παρακάτω.

Υποκατηγορία 1: Standard Variables

- **machine_capacity:** Αποτελεί την χωρητικότητα του μηχανήματος σε προϊόν. Είναι δηλαδή ο αριθμός των προϊόντων που μπορούν να περιμένουν στον buffer κάθε μηχανήματος μέχρι να επεξεργαστούν. Είναι το όριο των προϊόντων που αναμένουν μέχρι να εισαχθούν στο ίδιο το μηχάνημα. Τα προϊόντα αυτά αποτελούν είτε την πρώτη ύλη είτε έχουν παραχθεί από κάποιο προηγούμενο μηχάνημα.
- **initial_machineProd:** Είναι το πλήθος των προϊόντων που βρίσκονται σε αναμονή, στην είσοδο του μηχανήματος έτοιμα προς επεξεργασία. Η τιμή της είναι συνήθως μηδέν. Διαφορετική τιμή από το μηδέν χρησιμοποιείται για την κατανόηση της συμπεριφοράς μιας παραγωγικής διαδικασίας σε μεταγενέστερο χρόνο από την έναρξή της.
- **current_machine_level:** Η τιμή της μεταβλητής ορίζει την ποσότητα των προϊόντων που το μηχάνημα ήδη παράγει. Δίνεται το δικαίωμα στο χρήστη να ξεκινήσει τη προσομοίωση μιας διαδικασίας από μια συγκεκριμένη κατάσταση της παραγωγής (μεταγενέστερη της έναρξης) με σκοπό να εξάγει συγκεκριμένα συμπεράσματα. Συνήθως η τιμή της μεταβλητής είναι μηδέν και σηματοδοτεί την έναρξη τη παραγωγής.
- **machine_batch:** Είναι ο αριθμός των προϊόντων που επιτρέπει ο κατασκευαστής να παράγει κάθε μηχάνημα. Είναι δηλαδή το πλήθος των προϊόντων που μπορεί να επεξεργάζεται ταυτόχρονα το μηχάνημα και έπειτα να τα εξάγει.
- **machine_critical_buffer:** Ένας buffer χρησιμοποιείται από ένα μηχάνημα για το προσαρμόζει στις αλλαγές και παρεμβολές της παραγωγικής διαδικασίας. Ο buffer είναι ένας τρόπος για να διασφαλιστεί ότι η γραμμή παραγωγής θα συνεχίσει να λειτουργεί ομαλά παρά τους απρόβλεπτους παράγοντες. Ένα παράδειγμα λειτουργίας του buffer είναι η διατήρηση αρκετών υλικών και προϊόντων στη διάθεση του μηχανήματος ως εγγύηση για τη διασφάλιση της ομαλής λειτουργίας των εργασιών. Οι κατασκευαστές κρατούν σε ετοιμότητα τις προμήθειες που απαιτούνται για παραγωγή και με τα αποθέματα για να βοηθήσουν στη σταθεροποίηση τυχόν διακυμάνσεων που αντιμετωπίζουν οι παραγωγικές διαδικασίες.

Υποκατηγορία 2: Dynamic Variables

- **MACHINE_SPEED:** Είναι η ταχύτητα με την οποία παράγει το μηχάνημα. Στα περισσότερα μηχανήματα η ταχύτητα αυτή ισούται με την ταχύτητα της παραγωγής (PRODUCTION_SPEED). Συνήθως η πλειοψηφία των μηχανημάτων σε μια παραγωγή

κινείται με ταχύτητα μεγαλύτερη των υπολοίπων έτσι ώστε να μην τεθεί το bottleneck εκτός λειτουργίας. Η ταχύτητα τροποποιείται αλγοριθμικά ανάλογα με την κατάσταση στην οποία βρίσκεται η γραμμή παραγωγής. Σε μηχανήματα τα οποία αποτελούν το **bottleneck** της παραγωγής, κινούνται πάντα με την μεγαλύτερη δυνατή ταχύτητα και είναι υπεύθυνα για την ομαλή λειτουργία της παραγωγής.

- **machine_input:** Είναι ο αριθμός των προϊόντων που εισάγει κάθε μηχανήμα για να επεξεργαστεί. Το πλήθος του μπορεί να πάρει τιμές από 1 έως και πολύ μεγαλύτερες. Ορίζεται από τον κατασκευαστή του μηχανήματος. Υπεύθυνος για την τροφοδότηση του μηχανήματος είναι ο buffer ο οποίος σύμφωνα με το απόθεμα που διαθέτει σερβίρει το input του μηχανήματος στο οποίο είναι αρμόδιο.
- **machine_output:** Ορίζει το πλήθος των επεξεργασμένων προϊόντων που εξάγει ένα μηχανήμα. Συνήθως, στο μεγαλύτερο αριθμό των μηχανημάτων παραγωγής το πλήθος αυτό ισοδυναμεί με την τιμή του machine_input. Δηλαδή με την ίδια ταχύτητα που εισάγει και επεξεργάζεται τα προϊόντα ένα μηχανήμα, με την ίδια τα εξάγει. Σε κάποιες περιπτώσεις μηχανημάτων, η έξοδος ορίζεται από το συνολικό πλήθος που υπάρχει στον buffer. Πιο συγκεκριμένα, όταν τα μηχανήματα εισάγουν προκαθορισμένο αριθμό προϊόντων σύμφωνα με το ρυθμό της παραγωγής, τότε εξάγουν συνολικά ένα άθροισμα αυτών των προϊόντων.
- **machine_ptime:** Είναι ο χρόνος επεξεργασίας του μηχανήματος. Είναι, δηλαδή ο χρόνος που χρειάζεται ένα μηχανήμα να εισάγει το προϊόν να το **επεξεργαστεί** και μετέπειτα να το εξάγει. Η τιμή του ορίζεται από τον κατασκευαστή του μηχανήματος. Αποτελεί μέρος τις υποκατηγορίας «δυναμικές μεταβλητές» διότι η τιμή της μεταβάλλεται ανάλογα με την συμπεριφορά της γραμμής, της συνθήκες και τις δυσλειτουργίες που παρουσιάζει. Η τιμή της ορίζεται από το πλήθος των προϊόντων που εξάγει προς την ταχύτητα παραγωγής.

$$machine_ptime = \frac{machine_output}{PRODUCTION_SPEED}$$

- **machine_batch_serve_time:** Η μεταβλητή αυτή ορίζεται σε συγκεκριμένα μηχανήματα τα οποία έχουν σύνθετη λειτουργία επεξεργασίας προϊόντων. Για παράδειγμα υπάρχουν μηχανήματα τα οποία παράγουν όσα προϊόντα βρουν διαθέσιμα προς επεξεργασία σε κάθε ορισμένη περίοδο. Άρα είναι ο χρόνος που απαιτείται για να δεχτεί το μηχανήμα προϊόντα σε μία περίοδο.
- **MTBF_machine:** Είναι ο χρόνος που έχει παρέλθει μεταξύ εγγενών αστοχιών – δυσλειτουργιών ενός μηχανικού συστήματος, κατά τη διάρκεια της κανονικής λειτουργίας του συστήματος. Το MTBF μπορεί να υπολογιστεί ως ο αριθμητικός μέσος των χρόνων μεταξύ διαδοχικών αστοχιών ενός συστήματος. Είναι σημαντικό να κατανοηθεί όταν ορίζουμε την "ώρα", μπορεί να μην σημαίνει πάντα ώρα ρολογιού. Είναι ο χρόνος κατά τον οποίο το σύστημα χρησιμοποιείται πραγματικά. Ο υπολογισμός του MTBF απαιτεί περισσότερα βήματα εκτέλεσης όταν υπάρχουν

μεγάλα χρονικά διαστήματα με αυξημένες εμφανίσεις δυσλειτουργιών (breakdowns).

$$MTBF_machine = \frac{total\ uptime}{\#\ breakdowns}$$

Υποκατηγορία 3: Machine Status

Η υποκατηγορία machine status αναπαριστά την κατάσταση των μηχανημάτων κάθε στιγμή στη διάρκεια της προσομοίωσης μιας παραγωγικής διαδικασίας. Σημαντικό είναι σε αυτό το σημείο να οριστούν οι καταστάσεις των μηχανημάτων.

3.1.2.1 Καταστάσεις μηχανημάτων

Οι καταστάσεις των μηχανημάτων είναι τρεις και ορίζουν αποκλειστικά την λειτουργία κάθε μηχανήματος με την μορφή του φάρου (αναλύεται στο κεφάλαιο 4).

- **RUN** (Green): Ένα μηχάνημα βρίσκεται σε κατάσταση **RUN** όταν είναι πλήρως λειτουργικό. Επεξεργάζεται δηλαδή τα προϊόντα τα οποία λαμβάνει. Η παραγωγή του είναι συνεχής χωρίς παρεμβολές.
- **STAND BY** (Yellow): Ένα μηχάνημα τίθεται σε κατάσταση **STAND BY** όταν ένα ή περισσότερα από τα επόμενα ή τα προηγούμενα μηχανήματα της γραμμής παραγωγής έχει τεθεί σε κατάσταση STOP με αποτέλεσμα είτε να μην υπάρχουν διαθέσιμα προϊόντα στην είσοδό του για να επεξεργαστεί το μηχάνημα, είτε να μην υπάρχει διαθέσιμος χώρος στην έξοδο του μηχανήματος για να χωρέσουν τα προϊόντα που παράγει.. Το μηχάνημα βρίσκεται σε κατάσταση αναμονής (stand by) χωρίς να έχει παρουσιαστεί κάποιο πρόβλημα στο ίδιο. Μόλις αποκατασταθεί η βλάβη του μεταγενέστερου μηχανήματος, είναι έτοιμο να δεχτεί το προϊόν από το προηγούμενο μηχάνημα θα εκκινήσει την παραγωγή του. Η κατάσταση αναμονής δηλώνει για το μηχάνημα ότι έχει σταματήσει να λειτουργεί χωρίς να είναι υπεύθυνο το ίδιο.
- **STOP** (Red): Το μηχάνημα τίθεται σε κατάσταση **STOP** όταν υπεύθυνο για την διακοπή της λειτουργίας του είναι μόνο αυτό και κανένα άλλο. Όταν κάποιο μηχάνημα παρουσιάσει δυσλειτουργία ή βλάβη κατά την διάρκεια της παραγωγής του, τίθεται εκτός λειτουργίας, δηλαδή σταματά να παράγει.

Οι μεταβλητές που ορίζονται στην υποκατηγορία 3: **Machine status** είναι:

- **machine_status**: Η μεταβλητή αυτή είναι υπεύθυνη για να ορίζει την κατάσταση στην οποία βρίσκεται κάθε μηχάνημα κατά τη διαδικασία της προσομοίωσης. Η τιμή της μεταβάλλεται σύμφωνα με την κατάσταση της γραμμής και εξαρτάται από τα υπόλοιπα μηχανήματα της παραγωγής αλλά και από τον εαυτό του. Συγκεκριμένα, αλλάζει τιμή όταν κάποιο γεγονός συμβεί κατά τη παραγωγική διαδικασία, ανεξαρτήτως σε ποιο μηχάνημα εμφανιστεί.

- **machine_description:** Είναι η περιγραφή που ορίζεται αυτόματα από το σύστημα κατά τη διάρκεια της παραγωγής. Εξαρτάται από το status το οποίο βρίσκεται κάθε μηχανήμα αλλά και από την συμπεριφορά της διαδικασίας παραγωγής. Η τιμή της καταγράφεται σε αρχείο του συστήματος ώστε να έχει πρόσβαση ο χρήστης κάθε στιγμή και να κατανοεί τον λόγο που βρέθηκε τη δεδομένη στιγμή το μηχανήμα στην εκάστοτε κατάσταση.

3.1.3 UI – RCA level

Στο επίπεδο αυτό ορίζονται οι μεταβλητές και οι παράμετροι του γραφικού περιβάλλοντος. Οι τιμές των μεταβλητών αφορούν αποκλειστικά και μόνο την οπτικοποίηση των μετρικών, των αποτελεσμάτων και συμπερασμάτων της προσομοίωσης.

- **MACHINE_IS_BROKEN:** Είναι μεταβλητή τύπου Boolean. Η τιμή της αλλάζει όταν το μηχανήμα εξέρχεται ή εισέρχεται από την κατάσταση stop στη κατάσταση run ή από run σε stop αντίστοιχα. Χρησιμοποιείται για να αποτρέψει την περίπτωση που ένα μηχανήμα είναι σε κατάσταση stop και προσπαθεί να ξανά εισέλθει στην ίδια κατάσταση. Γενικά, λειτουργεί ως παράγοντας σταθεροποίησης των καταστάσεων των μηχανημάτων κατά την προσομοίωση με σκοπό να ορίζει ότι κάποιο μηχανήμα βρίσκεται σε κατάσταση stop τη προκαθορισμένη στιγμή.
- **MACHINE_RUN_TIMES:** Είναι μια ακέραια τιμή η οποία αυξάνεται κατά μία μονάδα κάθε φορά που το μηχανήμα εισέρχεται στη κατάσταση run αφού προηγουμένως έχει βρεθεί στη κατάσταση stand by είτε stop.
- **MACHINE_RUN_DURATION:** Είναι μια λίστα τιμών η οποία επεκτείνεται όταν εκτελείται η διεργασία του συγκεκριμένου μηχανήματος (machine). Η τιμές που λαμβάνει είναι ο χρόνος σε δευτερόλεπτα που χρειάζεται για να επεξεργαστεί το προϊόν. Οι τιμές τις επεξεργάζονται από τη συνάρτηση τροποποίησης στην οποία: αθροίζονται, τροποποιούνται και αποτυπώνονται στο γραφικό περιβάλλον του προσομοιωτή.
- **MACHINE_STANDBY_TIMES:** Είναι το πλήθος των καταστάσεων stand by που τίθεται κάθε φορά ένα μηχανήμα. Δηλαδή όταν ένα μηχανήμα μεταβαίνει στη κατάσταση STAND BY, αυτόματα η τιμή της αυξάνεται κατά ένα.
- **MACHINE_STANDBY_DURATION:** Είναι μια λίστα τιμών που μετριέται σε δευτερόλεπτα. Κάθε τιμή της λίστας είναι ο χρόνος στον οποίο το μηχανήμα βρέθηκε στην κατάσταση αναμονής (stand by). Αποτελεί δηλαδή τα δευτερόλεπτα τα οποία το μηχανήμα χρειάστηκε να περιμένει ώστε να τεθεί πάλι σε κατάσταση run. Από τη συνάρτηση τροποποίησης οι τιμές της συμβάλουν στην οπτικοποίηση και την εξαγωγή συμπερασμάτων.

- **MACHINE_STANDBY_FLAG:** Είναι μεταβλητή τύπου Boolean. Η τιμή της μεταβάλλεται σύμφωνα με την κατάσταση την οποία βρίσκεται προτού εισέλθει στην κατάσταση stand by. Η σημαία αυτή χρησιμοποιείται έτσι ώστε το μηχάνημα όταν βρεθεί σε κατάσταση stand by, ο αλγόριθμος να το αποτρέψει να εισέλθει ξανά στην ίδια κατάσταση. Η λειτουργία της είναι παρόμοια με τη λειτουργία της μεταβλητής MACHINE_IS_BROKEN.
- **MACHNE_STOP_FLAG:** Είναι μια σημαία τύπου Boolean και έχει την ίδια λειτουργία με την μεταβλητή MACHINE_STANDBY_FLAG. Χρησιμοποιείται για να μην επιτρέπεται το μηχάνημα να εισέρχεται στην ίδια κατάσταση stop. Δηλαδή αποτρέπει το μηχάνημα να ξανά τεθεί στην κατάσταση stop όταν βρίσκεται ήδη σε αυτή.
- **MACHINE_STOP_TIMES:** Είναι το πλήθος των φορών που ένα μηχάνημα έχει τεθεί σε breakdown ή αλλιώς βλάβη. Η τιμή τους καταμετρείται για την οπτικοποίηση των αποτελεσμάτων του προσομοιωτή.
- **MACHINE_STOP_DURATION:** Είναι ένα λεξικό (python dictionary) το οποίο σαν κλειδί (key) έχει ποια φορά σε αύξουσα τιμή βρέθηκε σε stop και ως τιμή (value) έχει την διάρκεια που έμεινε στην κατάσταση stop της συγκεκριμένης φορές. Χρησιμοποιείται επίσης για την αποτύπωση των μετρικών και των συμπερασμάτων που εξάγει ο προσομοιωτής.

3.2 Εργαλεία ανάπτυξης του συστήματος

Όπως θα αποδειχθεί στη συνέχεια, η βιβλιοθήκη SimPy έχει ένα πλούσιο ρεπερτόριο εργαλείων προσομοίωσης διαδικασιών, το οποίο είναι πολύτιμο για την ανάπτυξη του αλγορίθμου προσομοίωσης. Στη πλειοψηφία η προσομοίωση συστημάτων γίνεται με γλώσσα προγραμματισμού την Python μιας και περιέχει μια πληθώρα εργαλείων που υποστηρίζουν την ανάπτυξη λογισμικού προσομοίωσης. Επομένως, αυτή η υποενότητα ασχολείται με την περιγραφή όλων των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη ενός πλήρους λογισμικού προσομοίωσης.

3.2.1 Python

Η Python είναι μια Γενικού Σκοπού και Υψηλού Επιπέδου γλώσσα προγραμματισμού η οποία δίνει έμφαση στην ανάπτυξη ευανάγνωστου κώδικα. Η φιλοσοφία αυτή διευκολύνει τη διαδικασία της ανάπτυξης έργων λογισμικού (πχ. Collaboration, Scaling κλπ.) και της συντήρησής τους (όπως Debugging). Τα Programming Paradigm που υποστηρίζει είναι ο Διαδικασιακός (Procedural), ο Οντοκεντρικός (Object-oriented) και ο Συναρτησιακός (Functional) προγραμματισμός. Παρέχει επίσης μηχανισμούς Garbage collection ο οποίος αυτοματοποιεί τη διαχείριση της μνήμης και μια πλούσια Standard Library που παρέχει Δομές Δεδομένων, διεπαφές Λειτουργικού Συστήματος και άλλα. Παρακάτω περιγράφονται συνοπτικά μερικά από τα modules της Standard Library που χρησιμοποιήθηκαν για τις ανάγκες του συστήματος προσομοίωσης, ενώ η ανάπτυξη του προγράμματος έγινε με χρήση Python 3.9.

- **Yield:** Η λειτουργία του σχετίζεται με την λειτουργία των generators και κατ' επέκταση με την λειτουργία των iterables. Κατά τη δημιουργία μιας λίστας τα αντικείμενα διαβάζονται ένα προς ένα, αυτή η λειτουργία ονομάζεται iteration. Οι generators είναι iterators, δηλαδή ένα είδος των iterable που μπορούν να διαβάσουν ένα προς ένα τα στοιχεία. Οι generators δεν αποθηκεύουν όλες τις τιμές στην μνήμη και αυτό δεν είναι πάντα επιθυμητό όταν υπάρχουν πολλές τιμές. Αντίθετα παράγουν τις τιμές τους on the fly, δηλαδή παράγουν μια τιμή, την ξεχνούν και υπολογίζουν την επόμενη. Το yield είναι μια λέξη – κλειδί που χρησιμοποιείται σαν την return, με την εξαίρεση ότι η συνάρτηση θα επιστρέψει ένα generator. Όταν καλείται η συνάρτηση, ο κώδικας που υπάρχει γραμμένος στο σώμα της συνάρτησης δεν τρέχει. Η συνάρτηση απλά επιστρέφει τον generator.
- **Timeout:** Αποτελεί τον χρόνο σύμφωνα με τον οποίο ο αλγόριθμος εκτελεί μια διεργασία. Η timeout έχει σαν όρισμα έναν αριθμό. Ο αριθμός αυτός μετριέται σύμφωνα με τη τιμή που έχει δοθεί στον προσομοιωτή. Συνήθως είναι σε δευτερόλεπτα. Άρα είναι ο χρόνος στον οποίο ο αλγόριθμος χρειάζεται για να εκτελέσει μια διεργασία.

3.2.2 SimPy

Το **SimPy** είναι ένα πλαίσιο προσομοίωσης (**framework**) διακριτών συμβάντων που βασίζεται σε διεργασίες με γλώσσα προγραμματισμού την Python. Οι διεργασίες στο SimPy ορίζονται από συναρτήσεις δημιουργίας Python και μπορούν, για παράδειγμα, να χρησιμοποιηθούν για τη μοντελοποίηση ενεργών στοιχείων όπως πελάτες, οχήματα ή πράκτορες, βιομηχανικές παραγωγικές διαδικασίες κλπ. Το SimPy παρέχει επίσης διάφορους τύπους κοινόχρηστων πόρων για τη μοντελοποίηση σημείων συμφόρησης περιορισμένης χωρητικότητας (όπως διακομιστές, μετρητές ταμείων και σήραγγες). Οι προσομοιώσεις μπορούν να εκτελεστούν «όσο το δυνατόν γρηγορότερα», σε πραγματικό χρόνο (ώρα ρολογιού) ή με μη αυτόματο βήμα στα συμβάντα. Η συγκεκριμένη βιβλιοθήκη αποτελεί την κύρια βιβλιοθήκη του προγράμματος αφού σύμφωνα με αυτή ορίζονται οι οντότητες που αποτελούν τα μηχανήματα, η διαδικασία της παραγωγής και η μοντελοποίηση του συστήματος.

Η βιβλιοθήκη **SimPy** παρέχει το **API** τελικού χρήστη του SimPy. Συγκεντρώνει τις πιο σημαντικές κλάσεις και μεθόδους του.

API τελικού χρήστη

- **Environment:** Είναι η κεντρική τάξη του SimPy. Περιέχει την κατάσταση της προσομοίωσης και επιτρέπει στα PEM να αλληλοεπιδρούν μαζί της (δηλαδή, να προγραμματίζουν συμβάντα - events).
- **Interrupt:** Αυτή η εξαίρεση εισάγεται σε μια διαδικασία εάν διακοπεί από μια άλλη.

- **Resource:** Μπορεί να χρησιμοποιηθεί από περιορισμένο αριθμό διεργασιών κάθε φορά (π.χ. πρατήριο καυσίμων με περιορισμένο αριθμό αντλιών καυσίμου).
- **PriorityResource:** Λειτουργεί όπως και ο πόρος, αλλά οι διαδικασίες αναμονής ταξινομούνται κατά προτεραιότητα.
- **PreemptiveResource:** Αποτελεί έκδοση του Resource με preemption.
- **Container:** Μοντελοποιεί την παραγωγή και την κατανάλωση ενός ομοιογενούς, αδιαφοροποιήτου όγκου. Μπορεί να είναι είτε συνεχές (όπως τα υγρά) είτε διακριτό (όπως τα μήλα).
- **Store:** Επιτρέπει την παραγωγή και κατανάλωση διακριτών αντικειμένων Python.
- **FilterStore:** Όπως και το Store, αλλά τα στοιχεία που αφαιρούνται από αυτό μπορούν να φιλτραριστούν με μια λειτουργία που ορίζει ο χρήστης.

Βασικά δομικά στοιχεία SimPy

Η ενότητα αυτή περιέχει την υλοποίηση των βασικών κλάσεων του SimPy. Τα πιο σημαντικά είναι άμεσα εισαγόμενα μέσω του simpy.

```
class simpy.core.BaseEnvironment
```

Το BaseEnvironment είναι ο αφηρημένος ορισμός ενός περιβάλλοντος. Μια υλοποίηση πρέπει τουλάχιστον να παρέχει τα μέσα πρόσβασης στην τρέχουσα ώρα του περιβάλλοντος (now()) και προγραμματισμού (schedule()), καθώς και εκτέλεσης (step()) και (run()) συμβάντων.

Η κλάση προορίζεται ως υπό κλάση για διαφορετικά περιβάλλοντα εκτέλεσης. Για παράδειγμα, το SimPy ορίζει ένα Environment για προσομοιώσεις με εικονικό χρόνο και ένα RealtimeEnvironment που προγραμματίζει και εκτελεί συμβάντα σε πραγματικό χρόνο.

- **Now:** Η τρέχουσα ώρα του περιβάλλοντος. Ορίζει την κάθε χρονική στιγμή στην οποία βρίσκεται η προσομοίωση.
- **active_process:** Η τρέχουσα ενεργή διαδικασία του περιβάλλοντος.
- **schedule(event, priority=1, delay=0):** Προγραμματίζει ένα συμβάν με δεδομένη προτεραιότητα και καθυστέρηση. Υπάρχουν δύο προεπιλεγμένες τιμές προτεραιότητας, URGENT και NORMAL.
- **step():** Επεξεργάζεται το επόμενο συμβάν. Είναι το βήμα με το οποίο αυξάνεται ο χρόνος του περιβάλλοντος.

- **run(until=None):** Εκτελεί το `step()` μέχρι να ικανοποιηθεί το δεδομένο κριτήριο `until`. Ορίζει δηλαδή τον χρόνο για τον οποίο θα τρέξει ο αλγόριθμος προσομοίωσης.
 - Εάν είναι `None` (που είναι η προεπιλογή) αυτή η μέθοδος θα επιστρέψει εάν δεν υπάρχουν άλλα συμβάντα προς επεξεργασία.
 - Εάν πρόκειται για `event`, η μέθοδος θα συνεχίσει να εκτελεί τα βήματα μέχρι να ενεργοποιηθεί αυτό το συμβάν και θα επιστρέψει την τιμή του.
 - Εάν μπορεί να μετατραπεί σε αριθμό, η μέθοδος θα συνεχίσει να εκτελεί τα βήματα μέχρι να φτάσει ο χρόνος του περιβάλλοντος σε αυτή τη τιμή.

```
class simpy.core.Environment(initial_time=0)
```

Κληρονομεί τα στοιχεία του `BaseEnvironment` και εφαρμόζει ένα περιβάλλον προσομοίωσης που προσομοιώνει το πέρασμα του χρόνου περνώντας από γεγονόσ σε γεγονός. Η παράμετρος `initial_time` ορίζει τον χρόνο που θα εκτελεστεί ο αλγόριθμος προσομοίωσης. Από προεπιλογή, ξεκινά από το 0.

Αυτή η κλάση παρέχει επίσης τα ψευδώνυμα για κοινούς τύπους συμβάντων, όπως για παράδειγμα `process`, `timeout` και `event`. Περιέχει τις ίδιες παραμέτρους με την κλάση `BaseEnvironment` με επιπλέον τις παρακάτω:

- **process(generator):** Δημιουργεί ένα νέο `process` για τον `generator`.
- **timeout (delay, value=None):** Επιστρέφει ένα νέο συμβάν `Timeout` με καθυστέρηση και, προαιρετικά, μια τιμή (`value`).
- **event():** Επιστρέφει μια νέα παρουσία συμβάντος. Η απόδοση αυτού του συμβάντος αναστέλλει μια διαδικασία έως ότου μια άλλη διεργασία ενεργοποιήσει το συμβάν.
- **exit(value=None):** Σταματάει την τρέχουσα εκτέλεση της διεργασίας, παρέχοντας προαιρετικά μια τιμή.

Βασικοί τύποι συμβάντων

- **Timeout:** προγραμματίζεται με μια συγκεκριμένη καθυστέρηση και επιτρέπει στις διεργασίες να διατηρούν την κατάστασή τους για ορισμένο χρονικό διάστημα.

```
class simpy.events.Timeout(env, delay, value=None)
```

Η `timeout` προγραμματίζεται με κάποια καθυστέρηση μετά τη δημιουργία της. Ένα συμβάν μπορεί να χρησιμοποιηθεί από διεργασίες για την αναμονή (ή τη διατήρηση της κατάστασής τους) και για τα βήματα του χρόνου καθυστέρησης. Προγραμματίζεται έπειτα από χρόνο `env.now + delay` και επομένως (σε αντίθεση με το `Event`) δεν έχει μέθοδο `success()` ή `fail()`.

- **Initialize:** Εκκινεί μια νέα διαδικασία (`process`).

```
class simpy.events.Initialize(env, process)
```

Η initialize αρχικοποιεί μια διαδικασία (process) και μπορεί να χρησιμοποιηθεί εσωτερικά μόνο από το process.

- **Process:** Οι διεργασίες μοντελοποιούνται επίσης ως συμβάντα, ώστε άλλες διεργασίες να μπορούν να περιμένουν μέχρι να ολοκληρωθεί μια άλλη.

```
class simpy.events.Process(env, generator)
```

Μια διεργασία είναι ένα wrapper για τη δημιουργία διεργασιών (που επιστρέφεται από μια συνάρτηση διεργασίας) κατά την εκτέλεσή της. Περιέχει πληροφορίες εσωτερικής και εξωτερικής κατάστασης του μηχανήματος και χρησιμοποιείται για αλληλεπίδραση διεργασιών. Η διαδικασία κληρονομεί events. Έτσι, παρέχεται η δυνατότητα στην διεργασία να αναμένει για τον τερματισμό μιας άλλης διεργασίας απλώς αποσπώντας την από τη λειτουργία της εκτέλεσής της. Η παράμετρος **generator** ορίζει την συνάρτηση η οποία έχει σχεδιαστεί και προγραμματιστεί όπως ακριβώς επιτελείται η διαδικασία της στην πραγματική γραμμή παραγωγής. Ένα στιγμιότυπο αυτής της κλάσης επιστρέφεται από τη **simpy.core.Environment.process()**.

- **interrupt**(cause=None)

```
exception simpy.events.Interrupt(cause)
```

Μια διεργασία μπορεί να διακοπεί από τον αλγόριθμο για οποιοδήποτε λόγο. Η λειτουργία αυτή χρησιμεύει για να αναπαραστήσει την δημιουργία και εμφάνιση βλαβών σε ένα μηχάνημα. Μια διαδικασία δεν μπορεί να διακοπεί εάν έχει ήδη τερματιστεί. Μια διαδικασία δεν μπορεί επίσης να διακοπεί από μόνη της.

- **Condition:** Τα συμβάντα μπορούν να συνδυαστούν με | και & και περιμένει έως ότου ενεργοποιηθεί το ένα ή και τα δύο συμβάντα.

```
class simpy.events.Condition(env, evaluate, events)
```

Ένα **condition** event ομαδοποιεί πολλά συμβάντα και ενεργοποιείται εάν μια δεδομένη συνθήκη (που εφαρμόζεται από τη συνάρτηση αξιολόγησης) γίνει αληθής. Η τιμή της συνθήκης είναι ένα λεξικό (python dictionary) που αντιστοιχίζει τα συμβάντα εισόδου στις αντίστοιχες τιμές τους. Περιέχει καταχωρήσεις μόνο για εκείνα τα συμβάντα που συνέβησαν μέχρι να εκπληρωθεί η συνθήκη. Εάν ένα από τα συμβάντα αποτύχει, η συνθήκη επίσης αποτυγχάνει και προωθεί την εξαίρεση του συμβάντος που αποτυγχάνει. Η συνάρτηση **condition** λαμβάνει τη λίστα με τα συμβάντα-στόχους και το λεξικό με όλες τις διαθέσιμες τιμές αυτήν τη στιγμή. Εάν επιστρέψει True, η συνθήκη είναι προγραμματισμένη. Οι συναρτήσεις Condition.all_events() και Condition.any_events() χρησιμοποιούνται για την υλοποίηση και (&) και ή (|) για συμβάντα.

Πόροι τύπου container

Τα **container** μοντελοποιούν την παραγωγή και την κατανάλωση ενός ομοιογενούς, αδιαφοροποίητου όγκου. Μπορεί να είναι είτε συνεχές (όπως το νερό) είτε διακριτό (όπως τα μήλα, μπουκάλια κλπ.). Για παράδειγμα, ένα πρατήριο βενζίνης αποθηκεύει αέριο

(βενζίνη) σε μεγάλες δεξαμενές. Τα βυτιοφόρα αυξάνουν και τα αυτοκίνητα μειώνουν την ποσότητα βενζίνης στις δεξαμενές αποθήκευσης του σταθμού.

```
class simpy.resources.container.Container(env, capacity, init=0)
```

Η παράμετρος `env` είναι το περιβάλλον (**environment**) προσομοίωσης στο οποίο είναι δεσμευμένο το `container`. Η χωρητικότητα (**capacity**) καθορίζει το μέγεθος του `container`, δηλαδή το μέγιστο πλήθος προϊόντων που μπορεί να αποθηκεύσει μέχρι να επεξεργαστεί και πρέπει να είναι θετικός αριθμός (> 0). Από προεπιλογή, ένα κοντέινερ έχει απεριόριστο μέγεθος. Μπορεί να καθορίσετε το αρχικό επίπεδο του κοντέινερ μέσω του `init`. Το `init` ορίζει αν ένα μηχάνημα διαθέτει προϊόν στο εσωτερικό του πριν την έναρξη της παραγωγής. Πρέπει να είναι ≥ 0 και είναι 0 από προεπιλογή. Η κλάση `container` απαρτίζεται από τις παρακάτω παραμέτρους:

- **capacity**: Είναι ο μέγιστος χώρος που μπορεί να έχει το `container`.
- **level**: Είναι το επίπεδο των προϊόντων που διαθέτει το μηχάνημα κατά την αρχικοποίησή του. Η τιμή του μπορεί να είναι από 0 έως `capacity`.
- **put**: Δημιουργεί ένα νέο `ContainerPut` event.
- **get**: Δημιουργεί ένα νέο `ContainerGer` event.

Για την εισαγωγή και την εξαγωγή των προϊόντων από ένα `container` είναι υπεύθυνες οι εντολές **ContainerPut** και **ContainerGet** αντίστοιχα.

```
class simpy.resources.container.ContainerPut(container, amount)
```

Η **put** είναι ένα γεγονός που βάζει ποσότητα στο δοχείο. Το συμβάν ενεργοποιείται μόλις υπάρχει αρκετός χώρος στο κοντέινερ. Εάν, το `amount` είναι ≤ 0 δημιουργείται σφάλμα στο σύστημα. Το **amount** δηλώνει την ποσότητα που εισάγεται στο `container` αυτό. Η ποσότητα αυτή πρέπει να εισάγεται στο `container`.

```
class simpy.resources.container.ContainerGet(resource, amount)
```

Η **get** είναι ένα συμβάν που παίρνει ποσότητα από το `container`. Το συμβάν ενεργοποιείται μόλις υπάρχει αρκετό διαθέσιμο περιεχόμενο στο κοντέινερ. Εάν το `amount` είναι ≤ 0 δημιουργείται σφάλμα στο σύστημα. Το `amount` δηλώνει την ποσότητα που αφαιρείται από το `container` δηλαδή έχει επεξεργαστεί και είναι διαθέσιμη να εισαχθεί στο επόμενο μηχάνημα.

Προσομοίωση πραγματικού χρόνου

Τέλος, η `SimPy` μπορεί να εκτελέσει μια προσομοίωση σε **πραγματικό χρόνο**. Ο χρόνος δηλαδή της προσομοίωσης περνάει σύμφωνα με το κλιμακούμενο πραγματικό χρόνο.

```
class simpy.rt.RealtimeEnvironment(initial_time=0, factor=1.0, strict=True)
```

Ένα χρονικό βήμα παίρνει συντελεστές δευτερολέπτων σε πραγματικό χρόνο (ένα δευτερόλεπτο από προεπιλογή). π.χ., αν δημιουργηθεί βήμα από το 0 έως το 3 με `factor=0,5`, η κλήση `simpy.core.BaseEnvironment.run()` θα διαρκέσει τουλάχιστον 1,5 δευτερόλεπτο. Εάν η επεξεργασία των συμβάντων για ένα χρονικό βήμα διαρκεί πολύ, εμφανίζεται ένα `RuntimeError` στο `step()`. Μπορεί να απενεργοποιηθεί αυτή η συμπεριφορά ορίζοντας το `strict` σε `False`. Η παράμετρος `initial_time` ορίζει τον χρόνο για τον οποίο θέλουμε να τρέξει η προσομοίωση. Η τιμή του είναι σε δευτερόλεπτα. Ο `factor` ορίζει την κλίμακα του πραγματικού χρόνου. Και το `step()` περιμένει μέχρι να περάσει αρκετός πραγματικός χρόνος για να συμβεί το επόμενο συμβάν. Η καθυστέρηση κλιμακώνεται σύμφωνα με τον `factor` του πραγματικού χρόνου. Εάν τα συμβάντα ενός χρονικού βήματος υποβάλλονται σε επεξεργασία πολύ αργά για τον δεδομένο `factor` και εάν είναι ενεργοποιημένο το `strict`, εμφανίζεται ένα σφάλμα χρόνου εκτέλεσης.

3.2.3 Threading

Με την βιβλιοθήκη `threading` επιτρέπεται η παράλληλη επεξεργασία διεργασιών του προγράμματος. Κάθε νήμα τρέχει σε διαφορετικό πυρήνα του επεξεργαστή εκτελώντας την ίδια χρονική στιγμή διαφορετικά `tasks`. Κάθε επεξεργαστής μπορεί να έχει ένα ή δύο νήματα.

3.2.4 Numpy

Μέσω του `numpy` γίνεται χρήση της διωνυμικής κατανομής. Η κατανομή επιλέγει την πιθανότητα με την οποία ένα μηχάνημα θα τεθεί σε σταμάτημα. Επιστρέφει την τιμή της διωνυμικής κατανομής μεταξύ της μέγιστης και της ελάχιστης τιμής που έχει ως παραμέτρους.

3.2.5 Άλλες βιβλιοθήκες

Enum: Η `enum` είναι μια κλάση που δημιουργεί `enumerated` αριθμούς. Ένα `enumeration` είναι ένα σύνολο από συμβολικά ονόματα το οποία αντιστοιχούν σε μοναδικές σταθερές τιμές.

Random: Η `random` είναι μια βιβλιοθήκη που χρησιμοποιείται για να παράγει τυχαίους αριθμούς με σκοπό να χρησιμοποιούνται κάθε φορά διαφορετικοί. Για παράδειγμα η διάρκεια σταματήματος ενός μηχανήματος στην αυτόματη διαδικασία διαφέρει κάθε φορά σύμφωνα με το προκαθορισμένο εύρος. Επίσης χρησιμοποιείται η εντολή `seed` η οποία σε κάθε νέα εκτέλεση του προγράμματος αναπαράγει διαφορετικά αποτελέσματα από τα προηγούμενα.

Colorama: Κάνει τις ακολουθίες χαρακτήρων διαφυγής ANSI να λειτουργούν στα MS Windows. Παράγει επιλεγμένα χρώματα στο τερματικό για την διάκριση των πολλών και διαφορετικών συνεχόμενων αποτελεσμάτων.

FuncTools: Χρησιμοποιείται σε συναρτήσεις οι οποίες επιδρούν σε ή επιστρέφουν άλλες συναρτήσεις. Γενικά, οποιοδήποτε αντικείμενο που μπορεί να κληθεί αντιμετωπίζεται ως συνάρτηση για τους σκοπούς αυτής της ενότητας. Η συνάρτηση `partial` μπορεί να χρησιμοποιηθεί για την εξαγωγή εξειδικευμένων συναρτήσεων από γενικές συναρτήσεις και επομένως βοηθούν στο να τις χρησιμοποιήσουμε ξανά τον κώδικά.

Matplotlib: Δημιουργεί στατικές, `animated`, και διαδραστικές οπτικοποιήσεις στην `python`. Χρησιμοποιείται για την παραγωγή γραφημάτων.

4. Αρχιτεκτονική του συστήματος

Στο κεφάλαιο 4 δίνεται μια αφαιρετική αναπαράσταση του συστήματος που αναπτύχθηκε, με σκοπό την κατανόηση της λειτουργίας του. Είναι χρήσιμο αρχικά να σχεδιαστεί η αρχιτεκτονική του συστήματος, η οποία αποτελεί τον τρόπο προσέγγισης της υλοποίησης του προγράμματος προσομοίωσης. Στη συνέχεια, δίνεται η σχηματική αναπαράσταση της οντότητας του μηχανήματος και τέλος περιγράφονται όλα τα υποσυστήματά του.

4.1 Αρχιτεκτονική συστήματος

Η αρχιτεκτονική περιγράφει τη γενική δομή του συστήματος, τα βασικά δομικά του στοιχεία, πως αυτά τα στοιχεία επικοινωνούν και ποια είναι τα βασικά χαρακτηριστικά του κάθε στοιχείου. Επομένως, η αρχιτεκτονική του συστήματος προσομοίωσης, επί της ουσίας παρέχει ένα τρόπο ανάλυσης και περιγραφής του συστήματος σε υψηλό αφαιρετικό επίπεδο και εστιάζει στις βασικές σχεδιαστικές αποφάσεις που έχουν ληφθεί για το σύστημα. Σκοπός της αρχιτεκτονικής είναι το σύστημα προσομοίωσης να προσαρμόζεται στο μεγαλύτερο ποσοστό των γραμμών παραγωγής και των παραγωγικών διαδικασιών που υπάρχουν σήμερα. Παρακάτω γίνεται η περιγραφή των στοιχείων (components) από τα οποία αποτελείται το σύστημα της διπλωματικής εργασίας και οι αλληλεπιδράσεις ανάμεσα σε αυτά τα στοιχεία. Ο διαχωρισμός γίνεται με βάση τα τμήματα που απαρτίζουν την συνολική υλοποίηση του προσομοιωτή.

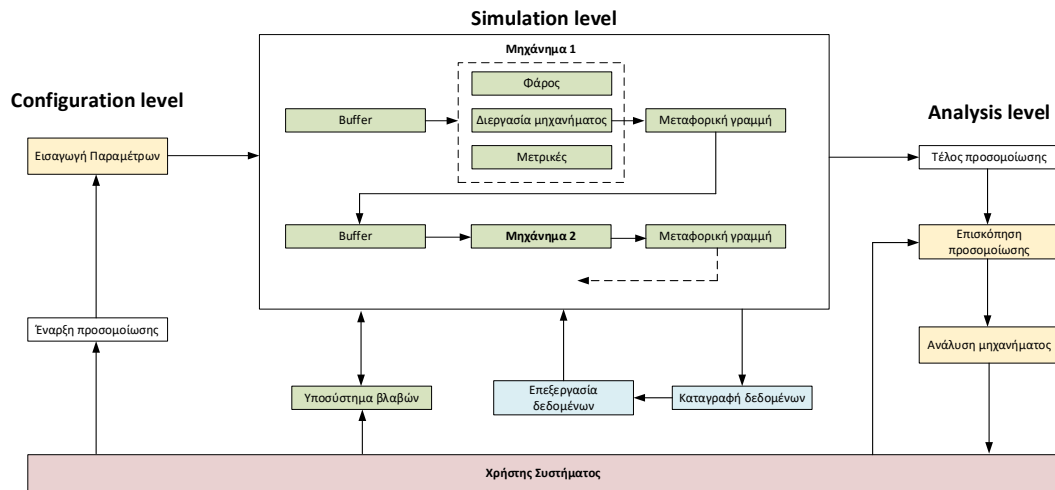
4.1.1 Αρχιτεκτονική συστήματος προσομοίωσης

Η αρχιτεκτονική του συστήματος προσομοίωσης στη παρούσα Διπλωματική Εργασία υλοποιήθηκε με βάση την αρχιτεκτονική των υποσυστημάτων σε συνδυασμό με την οντότητα. Συγκεκριμένα, ο χρήστης κατά τη διάρκεια της προσομοίωσης έρχεται σε επαφή με **τέσσερα παράθυρα** γραφικού περιβάλλοντος. Όπως φαίνεται από το σχήμα 4.1, το πρώτο παράθυρο αποτελεί την εισαγωγή των βασικών παραμέτρων με βάση τις οποίες θα τρέξει η προσομοίωση, στο οποίο ο χρήστης εισάγει τον χρόνο που επιθυμεί να εκτελεστεί η προσομοίωση και το είδος εμφάνισης βλαβών.

Το δεύτερο παράθυρο αφορά το κύριο παράθυρο προσομοίωσης και αναπαριστά σε ζωντανή απεικόνιση (**live monitoring**) την παραγωγική διαδικασία που προσομοιώνεται. Το υποσύστημα προσομοίωσης διαθέτει το υποσύστημα buffer, την αρχιτεκτονική της οντότητας, το υποσύστημα βλαβών και μεταφορικής γραμμής. Κατά τη διάρκεια της προσομοίωσης το σύστημα καταγράφει τα απαραίτητα δεδομένα και τα επεξεργάζεται.

Μετά την ολοκλήρωση της προσομοίωσης, εμφανίζεται στον χρήστη το τρίτο παράθυρο UI το οποίο αφορά την σύνοψη της γραμμής παραγωγής. Σε αυτό αναγράφονται τα στατιστικά δεδομένα της παραγωγής με τα οποία ο χρήστης κατανοεί πλήρως τη συμπεριφορά της.

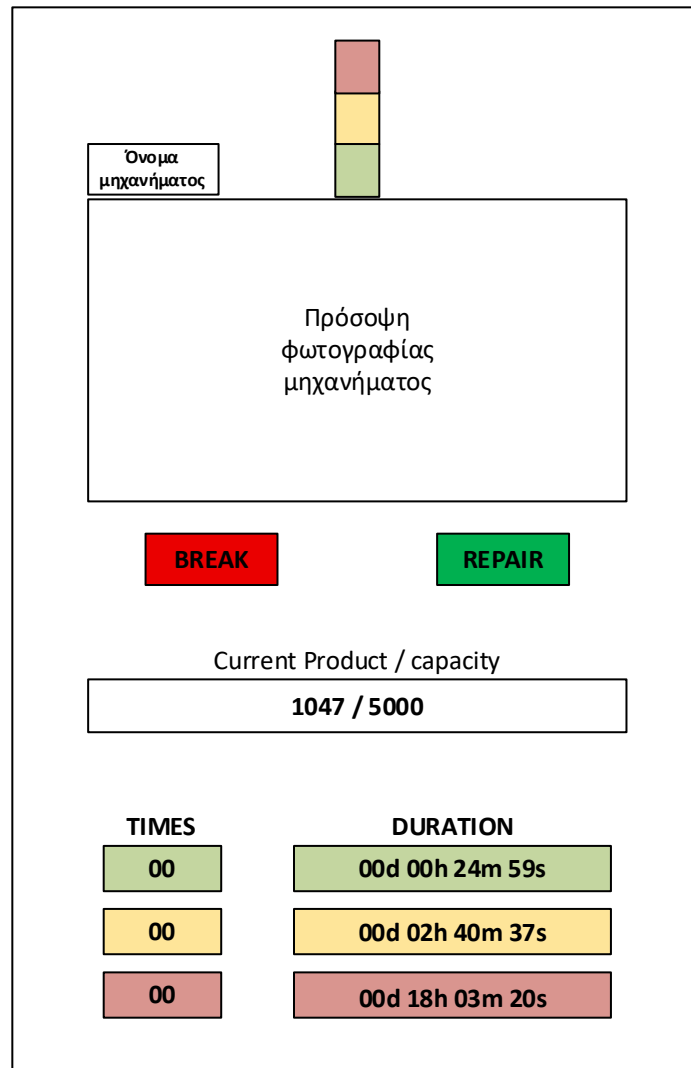
Τέλος, παρέχεται η δυνατότητα στο χρήστη, μέσω του τρίτου παραθύρου (παράθυρο επισκόπησης γραμμής παραγωγής), να επιλέξει την ανάλυση οποιουδήποτε μηχανήματος της παραγωγής επιθυμεί. Έτσι, εμφανίζεται το παράθυρο της ανάλυσης του μηχανήματος. Ο χρήστης μέσα από αυτό βλέπει τα δεδομένα που αφορούν αποκλειστικά την παραγωγική διαδικασία του μηχανήματος κατά την διάρκεια της προσομοίωσης. Η αρχιτεκτονική του συστήματος απεικονίζεται στο σχήμα 4.1:



Σχήμα 4.1: Αρχιτεκτονική συστήματος

4.1.2 Αρχιτεκτονική UI οντότητας

Η αρχιτεκτονική της οντότητας περιλαμβάνει όλα τα απαραίτητα γραφικά χαρακτηριστικά για την πλήρη κατανόηση της συμπεριφοράς της γραμμής παραγωγής κατά τη διάρκεια της εκτέλεσης της προσομοίωσής της. Στο παρακάτω σχήμα απεικονίζεται η οντότητα και διαχωρίζεται σε υποσυστήματα τα οποία αναλύονται στο κεφάλαιο 5.4.



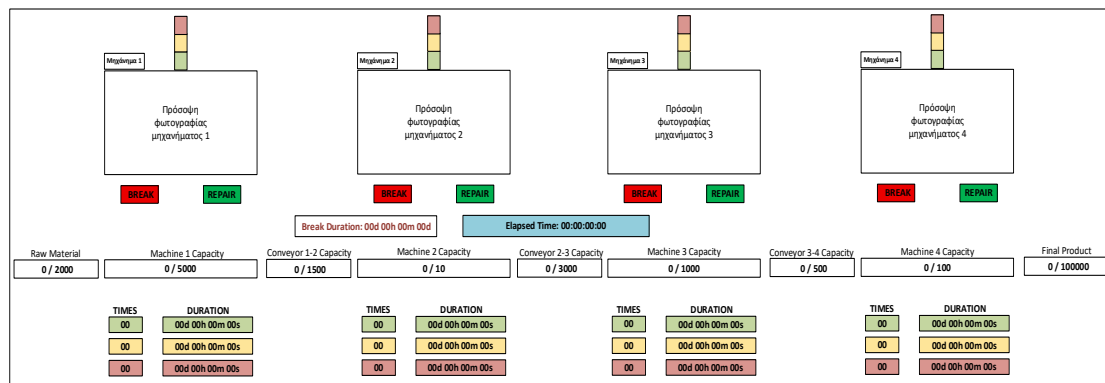
Σχήμα 4.2: Αρχιτεκτονική οντότητας

Η ανάλυση της οντότητας γίνεται με **top-down** προσέγγιση. Αρχικά, υπεύθυνος για την αναπαράσταση των τριών καταστάσεων του μηχανήματος είναι ο φάρος. Παράλληλα με την αναπαράσταση του φάρου ορίζεται και το όνομα του μηχανήματος. Στη συνέχεια, αποτυπώνεται η πρόσοψη του μηχανήματος στο οποίο αναφέρεται η οντότητα και βοηθά το χρήστη να αντιλαμβάνεται τη ροή της παραγωγικής διαδικασίας. Έπειτα, αν ο χρήστης επιλέξει την χειροκίνητη δημιουργία βλαβών, εμφανίζονται τα δύο κουμπιά: break και repair. Το break με το πάτημα θέτει το μηχάνημα σε κατάσταση stop και το repair βγάζει το μηχάνημα από τη κατάσταση stop και το θέτει σε run. Κάτω από την απεικόνιση των χειροκίνητων βλαβών υπάρχει η χωρητικότητα του μηχανήματος και του buffer. Δηλαδή η αριστερά τιμή ορίζει πόσα προϊόντα έχει στο εσωτερικό του το μηχάνημα και επεξεργάζεται και η δεξιά ορίζει το ανώτατο όριο του buffer, μέχρι πόσα προϊόντα δηλαδή μπορεί να εισάγει το μέγιστο. Τέλος, στην οντότητα απεικονίζονται διαδοχικά οι μετρικές που αφορούν τις καταστάσεις του μηχανήματος: run, stand by, stop. Στις μετρικές, η πρώτη στήλη αφορά το πλήθος των εμφανίσεων που το μηχάνημα τέθηκε στην εκάστοτε κατάσταση και η δεύτερη στήλη αφορά το συνολικό χρόνο για τον οποίο το μηχάνημα ήταν σε αυτή τη κατάσταση. Τα δεδομένα και οι μετρικές τρέχουν σε live αναπαράσταση κατά την εκτέλεση της προσομοίωσης.

4.1.3 Αρχιτεκτονική UI προσομοίωσης

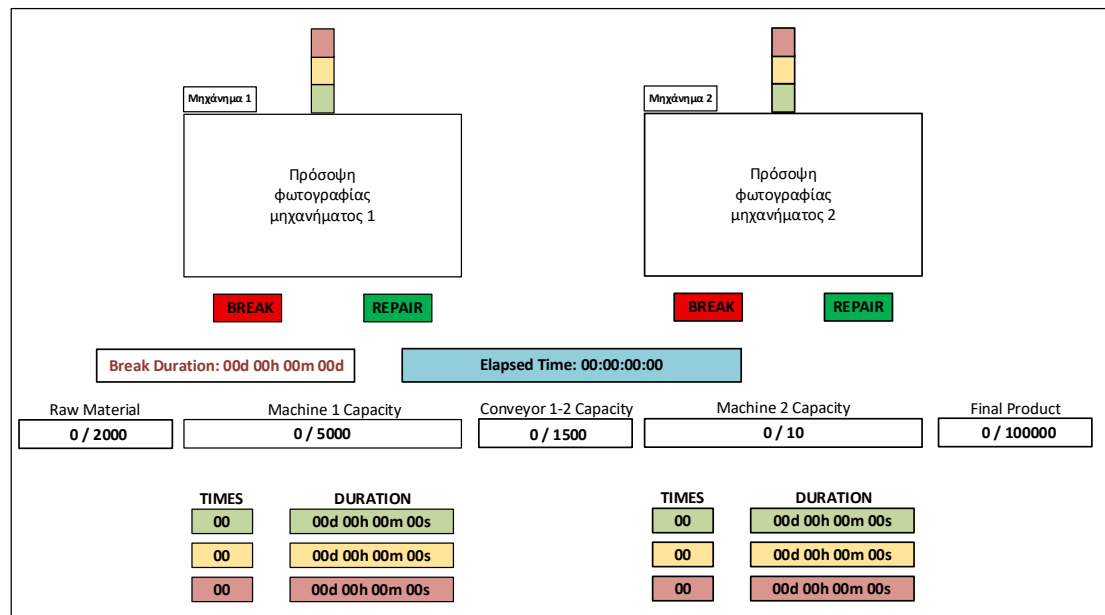
Για το γραφικό περιβάλλον της προσομοίωσης σχεδιάστηκε η αρχιτεκτονική της με σκοπό να προσομοιώνει με ευκολία οποιαδήποτε γραμμή παραγωγής. Ο αριθμός των μηχανημάτων μπορεί να διαφέρει σε κάθε παραγωγική διαδικασία. Ο αλγόριθμος του γραφικού περιβάλλον προσομοίωσης υλοποιήθηκε σύμφωνα με την αρχιτεκτονική της οντότητας. Κάθε σύστημα προσομοίωσης βασίζεται στην αρχιτεκτονική της προσομοίωσης. Η αρχιτεκτονική της προσομοίωσης βασίζεται με τη σειρά της στην οντότητα που υλοποιείται στην ενότητα 4.1.2.

Στο σχήμα 4.3 απεικονίζεται η αρχιτεκτονική της προσομοίωσης μια γραμμής παραγωγής τεσσάρων μηχανημάτων.



Σχήμα 4.3: Αρχιτεκτονική 4 μηχανημάτων

Στο σχήμα 4.4 ορίζεται η αρχιτεκτονική δυο μηχανημάτων μιας απλής παραγωγικής διαδικασίας.



Σχήμα 4.4: Αρχιτεκτονική 2 μηχανημάτων

4.2 Ανάλυση αρχιτεκτονικής

Στην ενότητα «Ανάλυση αρχιτεκτονικής» παρουσιάζεται η ανάλυση του συστήματος και ο διαχωρισμός του σε υποσυστήματα όσον αφορά την αρχιτεκτονική.

4.2.1 Διαχωρισμός υποσυστημάτων

Τα τμήματα και τα υποσυστήματα στα οποία διασπάται το σύστημα είναι:

- Υποσύστημα εισαγωγής βασικών παραμέτρων
- Υποσύστημα καταχώρησης δεδομένων
- Υποσύστημα προσομοίωσης
 - Υποσύστημα φάρου
 - Υποσύστημα μεταφορικής γραμμής
 - Υποσύστημα buffer
 - Υποσύστημα διεργασίας μηχανήματος
 - Υποσύστημα live χρόνου προσομοίωσης
 - Υποσύστημα μετρικών
- Υποσύστημα βλαβών
- Υποσύστημα καταγραφής δεδομένων
- Υποσύστημα επεξεργασίας δεδομένων
- Υποσύστημα live monitoring
- Υποσύστημα επισκόπησης προσομοίωσης
- Υποσύστημα ανάλυσης απόδοσης μηχανήματος

4.2.2 Περιγραφή υποσυστημάτων

Εκτός των υποσυστημάτων και των τμημάτων, στην αρχιτεκτονική αντιστοιχούν οι οντότητες των μηχανημάτων, οι ρυθμίσεις του συστήματος και τα δεδομένα που παράγονται από την εκτέλεση του αλγορίθμου. Σε αυτό τη σημείο μπορεί να γίνει η περιγραφή λειτουργίας κάθε υποσυστήματος.

Υποσύστημα εισαγωγής βασικών παραμέτρων

Το τμήμα αυτό αποτελεί ένα γραφικό περιβάλλον φιλικό προς τον χρήστη. Ο χρήστης μέσω του παραθύρου εισάγει τις βασικές παραμέτρους που απαιτούνται από τον προσομοιωτή για να εκτελεστεί με επιτυχία. Οι παράμετροι αυτές είναι: ο χρόνος προσομοίωσης (SIM_TIME) δηλαδή ο χρόνος που χρειάζεται για να εκτελεστεί η παραγωγική διαδικασία. Ο διακριτός χρόνος βήματος (DT) ο οποίος αφορά την οπτικοποίηση των αποτελεσμάτων στο υποσύστημα live monitoring και το ρυθμό με τον οποίο ανανεώνεται. Και τέλος στο υποσύστημα αυτό εμπεριέχεται η επιλογή του χρήστη για την αυτόματη ή την χειροκίνητη δημιουργία βλαβών.

Υποσύστημα καταχώρησης δεδομένων

Στο υποσύστημα καταχώρησης δεδομένων περιέχονται οι βασικές παράμετροι των μηχανημάτων. Οι παράμετροι αυτές είναι σημαντικές για την ορθή λειτουργία και αρχικοποίηση του μηχανήματος και κατ' επέκταση της παραγωγής. Το μεγαλύτερο πλήθος των παραμέτρων ορίζονται από τον κατασκευαστή και τα υπόλοιπα ορίζονται από τον προγραμματιστή κατόπιν συνεννόησης με τον χρήστη του συστήματος. Σκοπός του υποσυστήματος είναι να καταχωρεί τις παραμέτρους των μηχανημάτων στη διαδικασία που επιτελούν κατά τη διάρκεια της προσομοίωσης. Χωρίς αυτές, ο αλγόριθμος θα αδυνατούσε να προσομοιώσει τη πραγματική γραμμή παραγωγής. Οι παράμετροι έχουν περιγραφεί αναλυτικά στο κεφάλαιο των απαιτήσεων του συστήματος και θα εξηγηθούν με παραδείγματα κώδικα στο κεφάλαιο της υλοποίησης.

Υποσύστημα προσομοίωσης

Το παρόν υποσύστημα είναι το κύριο μέρος προσομοίωσης στο οποίο αποτυπώνεται η ροή της παραγωγικής διαδικασίας. Αποτελεί το κεντρικό γραφικό περιβάλλον και έχει ως οντότητες κάθε μηχανήμα της γραμμής που προσομοιώνεται. Στο τμήμα προσομοίωσης συγκαταλέγονται έξι υποσυστήματα τα οποία απαρτίζουν κάθε οντότητα. Τα συστήματα αυτά είναι: υποσύστημα φάρου, μεταφορικής γραμμής, buffer, διεργασίας μηχανήματος, live χρόνου προσομοίωσης και τέλος το υποσύστημα μετρικών.

Υποσύστημα Φάρου

Το υποσύστημα φάρου είναι υπεύθυνο για την αποτύπωση των καταστάσεων του μηχανήματος. Αποτελείται από 3 καταστάσεις (**Run**, **Stand by**, **Stop**) οι οποίες περιεγράφηκαν στο κεφάλαιο 3.1. Σκοπός του συστήματος είναι να προσομοιώνει τη λειτουργία ενός πραγματικού φάρου που έχει τοποθετηθεί σε ένα μηχανήμα με τη λειτουργία της προσομοίωσης. Δηλαδή κατά τη ροή μιας γραμμής παραγωγής, στο φάρο ανάβει η αντίστοιχη ένδειξη χρώματος ανάλογα με την κατάσταση στην οποία βρίσκεται το μηχανήμα. Για παράδειγμα, κατά τη διάρκεια της παραγωγής έστω ότι ένα μηχανήμα είναι σε κατάσταση Run τότε και ο φάρος θα έχει αναμμένο το χρώμα πράσινο. Σε περίπτωση που το ίδιο μηχανήμα τεθεί σε κατάσταση stop για οποιοδήποτε λόγο, αυτόματα ο φάρος τίθεται σε χρώμα κόκκινο (κατάσταση stop).

Υποσύστημα μεταφορικής γραμμής

Οι μεταφορικές γραμμές διέπουν κάθε παραγωγική διαδικασία. Είναι υπεύθυνες για την μεταφορά των προϊόντων από το ένα μηχανήμα προς το επόμενο. Η υλοποίησή τους είναι απαραίτητη διότι επιτελούν μια ξεχωριστή διαδικασία κατά τη διάρκεια της παραγωγής. Αυτή η διαδικασία είναι σύνθετη και όχι απλή. Η λειτουργία και ο στόχος του υποσυστήματος γραμμών είναι να εκτελούν διαρκώς ελέγχους στο μηχανήμα που σερβίρουν το προϊόν. Δηλαδή κάθε υποσύστημα μεταφορικής γραμμής είναι υπεύθυνο να ελέγχει και να τροφοδοτεί με προϊόν το αμέσως επόμενο υποσύστημα διεργασίας μηχανήματος.

Υποσύστημα buffer

Στις γραμμές παραγωγής η έννοια του buffer ορίζεται ως η διατήρηση αρκετών αποθεμάτων για την ομαλή λειτουργία των λειτουργιών. Αυτές οι προμήθειες συχνά περιλαμβάνουν τις πρώτες ύλες που χρειάζονται για την παραγωγή, καθώς και τα αποθέματα των τελικών προϊόντων που περιμένουν την αποστολή. Οι εγκαταστάσεις παραγωγής διατηρούν αυτά τα αποθέματα προσωρινής αποθήκευσης για να βοηθήσουν στη σταθεροποίηση τυχόν διακυμάνσεων που αντιμετωπίζουν με τις αλυσίδες προσφοράς και ζήτησης, τις παραγωγικές ικανότητες και τους χρόνους παράδοσης. Χωρίς την κατάλληλη αποθήκευση, οι διαδικασίες παραγωγής θα επιβραδύνονταν, τα έξοδα θα αυξάνονταν και τα κέρδη θα μειωνόταν. Ο σκοπός του buffer είναι να ληφθεί υπόψη η μεταβλητότητα στις διαδικασίες παραγωγής, ενώ παράλληλα να μεγιστοποιηθεί η αποτελεσματικότητα και τα κέρδη. Σε έναν ιδανικό κόσμο, η προσωρινή αποθήκευση δεν θα ήταν απαραίτητη επειδή η μεταβλητότητα δεν θα υπήρχε. Ωστόσο, δεδομένου ότι η μεταβλητότητα υπάρχει, είναι απαραίτητο να χρησιμοποιηθεί η προσωρινή αποθήκευση (**buffering**) ως μέσο για την ελαχιστοποίηση του αντίκτυπου αυτών των μεταβλητών. Μέσω της προσωρινής αποθήκευσης, οι κατασκευαστές μπορούν να αλλάξουν τις διαδικασίες τους μέσω του χειρισμού των αποθεμάτων, των χωρητικότητων και των χρόνων. Στον αλγόριθμο το υποσύστημα του buffering αποτελεί μια διεργασία που εκτελεί το πρόγραμμα και ελέγχει συχνά το απόθεμα κάθε μηχανήματος.

Υποσύστημα διεργασίας μηχανήματος

Το υποσύστημα μηχανήματος αποτελεί την κεντρική λειτουργία της οντότητας – μηχανήματος. Είναι υπεύθυνο για την διεργασία που επιτελεί στον αλγόριθμο. Δηλαδή απεικονίζει τον τρόπο με τον οποίο το μηχάνημα δέχεται προϊόντα, τα επεξεργάζεται και τα εξάγει έτοιμα για να τα υποδεχτεί το επόμενο μηχάνημα. Είναι η διαδικασία η οποία επιτελεί κάθε μηχάνημα που έχει προσαρμοστεί σύμφωνα με τις απαιτήσεις της γραμμής παραγωγής. Η οντότητα του μηχανήματος περιέχει επίσης και την απεικόνιση του ίδιου του μηχανήματος. Διαθέτει δηλαδή μια φωτογραφία που αναπαριστά πιστά την πρόσοψη του πραγματικού μηχανήματος. Τέλος, εκτός από την απεικόνιση του μηχανήματος παρέχεται και το όνομα του μηχανήματος στον χρήστη.

Υποσύστημα live χρόνου προσομοίωσης

Είναι το υποσύστημα για την απεικόνιση του χρόνου προσομοίωσης. Ο χρόνος αυτός αυξάνεται σύμφωνα με την βιβλιοθήκη SimPy απεικονίζει live τον χρόνο παραγωγής της γραμμής. Πιο συγκεκριμένα, απεικονίζεται σε μέρες, ώρες, λεπτά και δευτερόλεπτα και δείχνει τον χρόνο στον οποίο η προσομοίωση είναι εν ενεργή και παράγει. Ο χρόνος αυτός προκαθορίζεται από τον χρήστη στο τμήμα εισαγωγής παραμέτρων. Είναι σημαντικός διότι ο χρήστης κατανοεί κάθε χρονική στιγμή την συμπεριφορά της γραμμής. Ο χρόνος προσομοίωσης ανανεώνεται σύμφωνα με το στατικό χρόνο βήματος (**DT**). Κάθε DT χρονικές στιγμές εμφανίζεται η νέα τιμή του στο υποσύστημα.

Υποσύστημα μετρικών

Το υποσύστημα των μετρικών αποτελεί αποκλειστικά την απεικόνιση της συμπεριφοράς του μηχανήματος κατά τη διάρκεια της παραγωγής. Αποτελείται από 3 καταστάσεις (run,

stand by, stop). Χρησιμοποιείται για την καταμέτρηση του χρόνου που βρίσκεται το μηχάνημα καθώς και για το πλήθος των φορών που βρέθηκε στην ίδια κατάσταση. Πιο συγκεκριμένα, εκτελείται στο παρασκήνιο καθώς εκτελείται η live προσομοίωση. Είναι, δηλαδή, ο πίνακας στον οποίο καταγράφεται η διάρκεια σε χρόνο και το πλήθος που βρέθηκε στη κάθε κατάσταση το μηχάνημα.

Υποσύστημα βλαβών

Το υποσύστημα βλαβών είναι μια διαδικασία υπεύθυνη για την εμφάνιση βλαβών στην προσομοίωση. Η ύπαρξη του έχει ιδιαίτερη σημασία γιατί σε μια πραγματική γραμμή παραγωγής το πιο σύνηθες φαινόμενο είναι η εμφάνιση βλαβών και δυσλειτουργιών στα μηχανήματα της παραγωγής. Από τον προσομοιωτή, ο χρήστης έχει τη δυνατότητα να επιλέξει ανάμεσα σε δύο είδη βλαβών: την **χειροκίνητη** βλάβη και την **αυτόματη**. Η **χειροκίνητη βλάβη** ενεργοποιείται από τον χρήστη με το πάτημα ενός κουμπιού το οποίο θέτει σε κατάσταση stop το μηχάνημα. Έχει την δυνατότητα να επιλέξει τον χρόνο τον οποίο ένα μηχάνημα θα παραμείνει στη κατάσταση stop μέσω του γραφικού πλαισίου καταμέτρησης χρόνου. Επιπλέον, όταν επιλέξει το κουμπί επιδιόρθωσης μηχανήματος, το μηχάνημα βγαίνει από την κατάσταση stop και τίθεται πάλι σε run. Η λειτουργία αυτή είναι απαραίτητη για τον χρήστη διότι επιλέγει ο ίδιος να παρατηρήσει τη συμπεριφορά της γραμμής παραγωγής σε συγκεκριμένο χρόνο της παραγωγής θέτοντας μηχανήματα εκτός λειτουργίας. Επίσης, ο χρήστης έχει τη δυνατότητα να επιλέξει τον χρόνο που θα βρεθεί σε κατάσταση σφάλματος κάθε μηχάνημα. Για την ευκολία της προσομοίωσης, κατά την έναρξη της κατάσταση stop εκκινεί το πλαίσιο καταμέτρησης για να επιλέξει ο χρήστης τον χρόνο στον οποίο θέλει να θέσει το μηχάνημα σε κατάσταση βλάβης. Με την επιδιόρθωση ο χρόνος αυτός μηδενίζεται.

Ο χρήστης έχει, ακόμη, την δυνατότητα να επιλέξει τον **αυτόματο τρόπο** εμφάνισης βλαβών αντί για τον χειροκίνητο. Αυτή η λειτουργία προσομοιώνει πιο σωστά την πραγματική γραμμή παραγωγής διότι σε μια πραγματική γραμμή παραγωγής η εμφάνιση βλαβών συμβαίνει με τυχαίο τρόπο σε κάθε μηχάνημα. Για το πρόγραμμα προσομοίωσης ο χρήστης πρέπει να δώσει την πιθανότητα με την οποία επιθυμεί να εμφανιστεί βλάβη σε κάθε μηχάνημα. Η εισαγωγή των τιμών των πιθανοτήτων για κάθε μηχάνημα γίνεται από το τμήμα εισαγωγής βασικών παραμέτρων.

Υποσύστημα καταγραφής δεδομένων

Το υποσύστημα αυτό είναι υπεύθυνο για την καταγραφή των δεδομένων που παράγει όλο το πρόγραμμα προσομοίωσης. Σκοπός του είναι να αποθηκεύει τα δεδομένα της παραγωγής στις μεταβλητές καταχώρησης δεδομένων. Κατά την διάρκεια εκτέλεσης του αλγορίθμου προσομοίωσης δημιουργούνται πληθώρα από δεδομένα και συγχρόνως καταγράφονται από το υποσύστημα. Έπειτα από την ολοκλήρωση της προσομοίωσης, ο αλγόριθμος επεξεργάζεται πολύ γρήγορα αυτά τα δεδομένα και τα στέλνει προς απεικόνιση στο υποσύστημα ολοκλήρωσης προσομοίωσης. Το υποσύστημα καταγραφής δεδομένων λειτουργεί ανεξάρτητα από το κυρίως πρόγραμμα προσομοίωσης χωρίς να το επηρεάζει.

Παράλληλα, το υποσύστημα αυτό διαθέτει έναν αλγόριθμο καταγραφής των δεδομένων σε αρχείο. Τα δεδομένα αυτά αφορούν τις καταστάσεις του φάρου. Η καταγραφή και το πλήθος των εγγραφών είναι ανάλογο του στατικού χρόνου βήματος (DT). Κάθε DT χρόνο καταγράφονται στο αρχείο τα δεδομένα: η χρονική στιγμή προσομοίωσης που καταγράφηκε αυτή η εγγραφή, το όνομα του μηχανήματος, η κατάσταση στην οποία βρίσκεται κατά τη διάρκεια της παραγωγής (MACHINE_STATUS), η περιγραφή για την κατάσταση της μηχανής

(MACHINE_DESCRIPTION) και τέλος, το πλήθος των προϊόντων που έχει παραχθεί από το μηχάνημα μέχρι εκείνη της στιγμή προσομοίωσης.

Δημιουργείται επιπλέον ένα **pandas.dataframe** δεδομένων το οποίο διατηρείται στον αλγόριθμο μέχρι την ολοκλήρωσή του και συμβάλει συνοδευτικά στα συμπεράσματα της προσομοίωσης και στον υπολογισμό των μετρικών. Η καταγραφή, επίσης, των δεδομένων στο dataframe γίνεται κάθε DT χρονικές στιγμές. Η χρήση του βοηθά στην γρήγορη καταγραφή των δεδομένων από άποψη πολυπλοκότητας του συστήματος και σκοπό έχει να αποφευχθεί ο σύνθετος υπολογισμός των δεδομένων κατά την ολοκλήρωση της προσομοίωσης.

Υποσύστημα επεξεργασίας δεδομένων

Στο υποσύστημα επεξεργασίας δεδομένων, όπως περιγράφεται και από τον τίτλο του, γίνεται η τροποποίηση και η επεξεργασία των δεδομένων του συστήματος. Σε αυτό, τα δεδομένα εισέρχονται από το κυρίως πρόγραμμα, επεξεργάζονται παράλληλα και ανεξάρτητα από την εκτέλεση της προσομοίωσης. Τέλος επιστρέφονται στο αρχικό πρόγραμμα για να αναπαρασταθούν ως στατιστικά στα υποσυστήματα: live monitoring, ολοκλήρωσης δεδομένων και ανάλυσης απόδοσης μηχανήματος.

Υποσύστημα live monitoring

Είναι ένα σύστημα υπεύθυνο για την ζωντανή παρακολούθηση (live monitoring) της γραμμής παραγωγής που προσομοιώνεται. Λειτουργεί με βάση τα υποσυστήματα καταγραφής δεδομένων, μετρικών και μηχανήματος. Απεικονίζει πρακτικά όσα αναφέρθηκαν σε αυτά τα τρία υποσυστήματα. Ο σκοπός του είναι να παρέχει τη διευκόλυνση στον χρήστη να παρακολουθεί σε live απεικόνιση τη προσομοίωση της παραγωγής, τη συμπεριφορά της γραμμής και τις μετρικές των μηχανημάτων.

Είναι ένα σημαντικό σύστημα για την γραφική απεικόνιση (UI) της γραμμής παραγωγής που προσομοιώνεται και το κυρίως εργαλείο για την διεπαφή του χρήστη με την γραμμή παραγωγής.

Υποσύστημα επισκόπησης προσομοίωσης

Το υποσύστημα ολοκλήρωσης προσομοίωσης απεικονίζει την τελική εικόνα της γραμμής παραγωγής. Εκτελείται λίγο πριν τη λήξη της προσομοίωσης και ολοκληρώνεται κατά τη λήξη της, όπου και εμφανίζεται ως διεπαφή στον χρήστη. Αποτυπώνει τα στατιστικά δεδομένα, δηλαδή τη συμπεριφορά της γραμμής κατά τη διαδικασία της παραγωγής, έπειτα από την ολοκλήρωση του χρόνου προσομοίωσης. Ο χρήστης, μέσω της γραφικής απεικόνισης (UI), έχει την δυνατότητα να κατανοήσει και να εξάγει συμπεράσματα για την διαδικασία της παραγωγής. Το γραφικό περιβάλλον του υποσυστήματος προσομοίωσης εμφανίζεται αυτόματα σε διαφορετικό περιβάλλον από αυτό της προσομοίωσης.

Το υποσύστημα αυτό αποτελείται από δύο πλαίσια καταγραφής στατιστικών. Το **πρώτο πλαίσιο** ορίζει τους δείκτες απόδοσης της γραμμής προσομοίωσης. Σε αυτό αναγράφεται η συνολική διάρκεια προσομοίωσης της παραγωγικής διαδικασίας. Αναγράφεται επίσης, η ταχύτητα της παραγωγής, δηλαδή πόσο προϊόντα παράγει κάθε δευτερόλεπτο η γραμμή, ο οποίος υπολογίζεται από το υποσύστημα προσομοίωσης. Υπολογίζονται και καταγράφονται τα συνολικά προϊόντα που παράχθηκαν από τον προσομοιωτή συγκριτικά με τα αναμενόμενα προϊόντα που θα παρήγαγε μια ιδανική γραμμή παραγωγής χωρίς

σταματήματα, βλάβες και μικρο-προβλήματα. Και τέλος, εκτιμάται η συνολική απόδοση της παραγωγής (**Overall Equipment Effectiveness – OEE**) σε ποσοστό. Αυτό βοηθά τον χρήστη να κατανοήσει πόσο παραγωγική ήταν η γραμμή.

Το **δεύτερο πλαίσιο** αφορά τις καταστάσεις των μηχανημάτων και έχει τη δομή πίνακα. Καταγράφονται σε στήλες το όνομα κάθε μηχανήματος, το ποσοστό στο οποίο ήταν στην κατάσταση run, stand by και stop. Τα στατιστικά αυτά καταγράφονται και επεξεργάζονται από το υποσύστημα καταγραφής δεδομένων. Σκοπό έχουν να ενισχύσουν στην εκτίμηση του χρήστη για την συνολική συμπεριφορά και απόδοση της γραμμής παραγωγής. Τέλος, παρέχεται και η δυνατότητα επιλογής της ανάλυσης κάθε μηχανήματος σε ξεχωριστό παράθυρο, με το πάτημα του αντίστοιχου κουμπιού. Υπεύθυνο για την ανάλυση είναι το υποσύστημα ανάλυσης απόδοσης μηχανήματος.

Υποσύστημα ανάλυσης απόδοσης μηχανήματος

Το υποσύστημα ανάλυσης μηχανήματος αποτελείται από τρία τμήματα. Σκοπός του υποσυστήματος είναι η ανάλυση των στατιστικών κάθε μηχανήματος ξεχωριστά. Δίνει στον χρήστη την ακριβή λεπτομέρεια και λειτουργία του μηχανήματος της παραγωγής για το οποίο επέλεξε να δει την ανάλυση. Ο υπολογισμός των στατιστικών γίνεται αυτόματα μετά τη λήξη της προσομοίωσης και υπεύθυνο είναι το υποσύστημα προσομοίωσης και καταγραφής δεδομένων.

Το **πρώτο τμήμα** αποτελείται από γενικές πληροφορίες για το συγκεκριμένο μηχάνημα που επιλέχθηκε προς ανάλυση. Πιο συγκεκριμένα, καταγράφεται το ποσοστό παραγωγής του μηχανήματος, δηλαδή πόσο παραγωγικό ήταν το μηχάνημα κατά τη διαδικασία της παραγωγής, καθώς και ο χρόνος για τον οποίο ήταν σε κατάσταση παραγωγής (run). Επιπλέον εκτιμάται και η τιμή του δείκτη **MTBS** (Mean Time Between Stop). Ο δείκτης MTBS είναι ίδιος με τον **MTBF** (Mean Time Between Failures).

Ο μέσος χρόνος μεταξύ βλαβών (**MTBF**) είναι μια πρόβλεψη του χρόνου μεταξύ των αστοχιών ενός μηχανήματος κατά το κανονικό χρόνο παραγωγής του. Με άλλα λόγια, το MTBF είναι μια μέτρηση συντήρησης, που αναπαρίσταται σε ώρες, που δείχνει πόσο χρόνο λειτουργεί ένα μηχάνημα χωρίς διακοπή. Είναι σημαντικό να σημειωθεί ότι το MTBF χρησιμοποιείται μόνο για αντικείμενα και μηχανήματα που δέχονται επιδιορθώσεις καθώς και ως ένα εργαλείο που βοηθά στο σχεδιασμό για να αποφευχθεί η επισκευή του εξοπλισμού. Πριν υπολογιστεί το MTBF, πρέπει να κατανοηθεί πώς επηρεάζει την αξιοπιστία και τη διαθεσιμότητα. Η υψηλή αξιοπιστία και η διαθεσιμότητα συνήθως πάνε μαζί, αλλά οι όροι δεν είναι εναλλάξιμοι. Η αξιοπιστία είναι η ικανότητα ενός μηχανήματος να εκτελεί τις απαιτούμενες λειτουργίες του υπό ορισμένες συνθήκες για μια προκαθορισμένη χρονική περίοδο. Είναι δηλαδή, η πιθανότητα ότι ένα κομμάτι του μηχανήματος θα κάνει αυτό που προορίζεται να κάνει χωρίς αστοχίες.

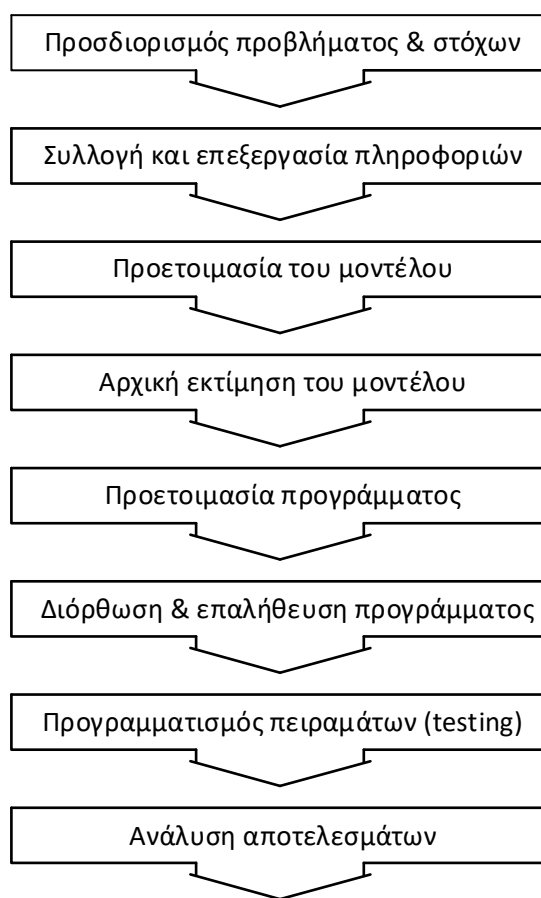
Στο **δεύτερο τμήμα** καταγράφονται τα στατιστικά που αφορούν την ανάλυση του μη παραγωγικού χρόνου. Δηλαδή τα στατιστικά που σχετίζονται με τις καταστάσεις των μηχανημάτων stand by (κατάσταση αναμονής). Αναπαρίσταται ο συνολικός μη παραγωγικός χρόνος, το πλήθος στο οποίο το μηχάνημα τέθηκε στη κατάσταση stand by και η συνολική διάρκεια του χρόνου που το μηχάνημα βρέθηκε σε κατάσταση αναμονής.

Επιπλέον στο ίδιο τμήμα καταγράφεται ο πίνακας του **Root Cause Analysis (RCA)**. Εκτενής αναφορά του RCA δίνεται στο κεφάλαιο 2.3. Στο συγκεκριμένο πίνακα αναπαρίστανται όλα τα μηχανήματα της γραμμής, το πλήθος των σταματημάτων, η διάρκεια και το ποσοστό για τα αντίστοιχα. Για το κάθε μηχάνημα που γίνεται η ανάλυση, τα δεδομένα της γραμμής του πίνακα αφορούν τις καταστάσεις που το μηχάνημα βρίσκεται σε stand by και όλα τα υπόλοιπα υπολογίζονται από την κατάσταση stop των μηχανημάτων, σε συνδυασμό με τα δεδομένα του μηχανήματος που γίνεται η ανάλυση.

Στο **τελευταίο τμήμα**, το οποίο αποτελεί τα στατιστικά των σταματημάτων του μηχανήματος, αναγράφονται όλες οι καταστάσεις stop και stand by του μηχανήματος. Κάθε γραμμή δεδομένων διαθέτει το όνομα του μηχανήματος, τη στιγμή της προσομοίωσης που δημιουργήθηκε, ο τύπος (stop ή stand by) και η διάρκεια.

4.3 Βασικά στάδια προσομοίωσης

Κάθε προσομοίωση μιας πραγματικής γραμμής παραγωγής ακολουθεί 8 βασικά στάδια. Μια επιτυχής προσομοίωση που προσεγγίζει την πραγματική παραγωγική διαδικασία προϋποθέτει την εφαρμογή όλων των παρακάτω βημάτων. Στα πλαίσια τη διπλωματικής εργασίας τα στάδια προσομοίωσης αποτέλεσαν θεμέλιο για τον ορθό ορισμό του αλγορίθμου και της μελέτης περίπτωσης.



Σχήμα 4.5

Κατά την σχεδίαση μιας διαδικασίας προσομοίωσης, απαιτείται πρώτα η πλήρης **κατανόηση του προβλήματος** που πάμε να λύσουμε. Κάθε γραμμή παραγωγής παρουσιάζει διαφορετικά προβλήματα και κατηγορίες εμφάνισης βλαβών, για αυτό είναι απαραίτητο να καθοριστούν οι **στόχοι της προσομοίωσης**. Στη συνέχεια, είναι απαραίτητο να κατανοηθεί πλήρως η λειτουργία της γραμμής παραγωγής, δηλαδή ο τρόπος με τον οποίο κάθε μηχανήμα εισάγει το προϊόν το επεξεργάζεται και έπειτα το εξάγει. Έτσι, **συλλέγονται τα δεδομένα και η πληροφορία** της γραμμής και στη συνέχεια επεξεργάζονται και αναλύονται.

Μετά την ολοκλήρωση των παραπάνω βημάτων, αρχίζει η **προετοιμασία του μοντέλου**, ο σχεδιασμός του. Κάθε μοντέλο προσομοίωσης βασίζεται σε μια συγκεκριμένη αρχιτεκτονική που αποτελεί πρότυπο για κάθε είδους προσομοίωση. Με αυτό τον τρόπο, προετοιμάζεται το μοντέλο για να δημιουργηθεί η **πρώτη εκτίμηση της προσομοίωσης**. Η

αρχική εκτίμηση του μοντέλου ενισχύει τον προγραμματισμό στην σφαιρική κατανόηση της παραγωγικής διαδικασίας και τον αποτρέπει από την εμφάνιση σημαντικών σφαλμάτων. Έπειτα, ο προγραμματιστής με βάση την αρχιτεκτονική και την προετοιμασία του μοντέλου, προετοιμάζει το πρόγραμμα υλοποίησης της προσομοίωσης. Το πρόγραμμα είναι πρακτικά ο αλγόριθμος που αντιστοιχεί στην αναπαράσταση της πραγματικής γραμμής.

Στη συνέχεια, **επαληθεύεται και διορθώνεται** το πρόγραμμα υλοποίησης. Σκοπός αυτού του βήματος είναι να βεβαιώσει την πλήρη εκτέλεση του αλγορίθμου χωρίς εμφάνιση σφαλμάτων, δυσλειτουργιών ή διαφορετικής προσέγγισης της προσομοίωσης από τη πραγματική γραμμή. Μετά την επαλήθευση και της επιβεβαίωσης ότι ο αλγόριθμος έχει συνταχθεί ορθά, εκτελούνται μια **σειρά πειραμάτων (testing)**. Το testing της υπηρετεί τις ανάγκες του προγραμματίζει για να επιβεβαιώσει την απόδοτης προσέγγισης της πραγματικής γραμμής παραγωγής. Στόχος είναι η προσομοίωση να προσεγγίζει όσο το δυνατόν περισσότερο την πραγματική λειτουργία της γραμμής.

Τέλος, **αναλύονται τα αποτελέσματα** κάθε πειράματος που εκτελέστηκε. Με αυτόν το τρόπο, εκτιμάται η αποδοτικότητα και η λειτουργία του αλγορίθμου. Είναι σημαντικό, να ελεγχθεί αρκετές φορές έτσι ώστε να εξασφαλιστεί η ορθή εκτέλεσή του. Ο προγραμματιστής μέσα από τα αποτελέσματα της ανάλυσης επιβεβαιώνει την προσέγγιση της προσομοίωσης. Αν η προσέγγιση είναι μικρή, τότε τα βασικά στάδια της προσομοίωσης εκτελούνται από την αρχή. Διαφορετικά, το πρόγραμμα είναι έτοιμο να παραδοθεί στον χρήστη.

4.4 Συνεχής βελτίωση προσομοιωτή

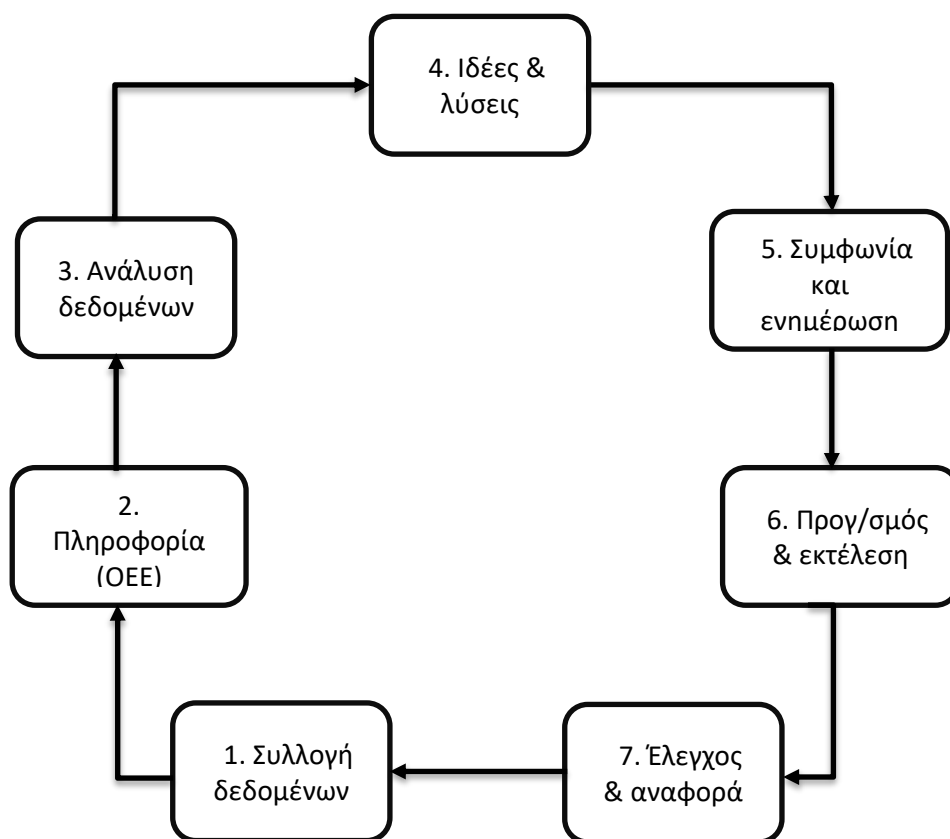
Έχοντας λοιπόν αναπτύξει την θεωρία αλλά και την εφαρμογή των υποσυστημάτων καθώς και τα βασικά στάδια της προσομοίωσης, το επόμενο βήμα που ακολουθεί είναι η εφαρμογή μεθόδων συνεχούς βελτίωσης. Η εφαρμογή των μεθόδων αυτών, αφορά διαδικασίες οι οποίες επιφέρουν σημαντικές απώλειες απόδοσης και ως εκ τούτου αποτελούν ευκαιρίες βελτίωσης. Αξίζει να σημειωθεί, πως για την ορθή και επιτυχημένη εφαρμογή μιας μεθόδου βελτίωσης, είναι απαραίτητη η γνώση των χειριστών σε θέματα υπολογισμού των δεικτών απόδοσης **GE** και **OEE**. Έχοντας αυτό σαν δεδομένο, θα πρέπει στη συνέχεια να αναπτυχθεί μια διαδικασία, η οποία θα βασίζεται στις απώλειες που εντοπίζονται σε μια γραμμή παραγωγής και θα στοχεύει στην ελαχιστοποίηση τους και κατ'επέκταση στην αύξηση της απόδοσης. Αν και οι περισσότερες βιομηχανίες διαθέτουν τα απαιτούμενα εργαλεία και γνωρίζουν αρκετές μεθόδους αντιμετώπισης και επίλυσης προβλημάτων, έχει παρατηρηθεί μια σχετική αδυναμία στην ορθή κατανόηση και εφαρμογή αυτών. Για την βελτίωση της συνολικής απόδοσης του εξοπλισμού (OEE) μπορούν να απαιτηθούν ενέργειες προς πολλές κατευθύνσεις και τμήματα εντός της εταιρίας, αλλά και παράλληλη εφαρμογή διαφορετικών τεχνικών. Η πολυπλοκότητα αυτή, απαιτεί την εξ'αρχής στοχευμένη επιλογή της κατάλληλης μεθοδολογίας, βάσει της υπό βελτίωση διαδικασίας. Δηλαδή, πρέπει να γίνει μια αξιολόγηση του προβλήματος και να επιλεγεί η μέθοδος προσέγγισης που μπορεί να εφαρμοστεί και να αποδώσει καλύτερα. Φυσικά, η δυσκολία των προβλημάτων ποικίλει βάσει της αιτίας που τα προκαλεί. Αν θέλαμε να κατατάξουμε με αύξοντα βαθμό δυσκολίας τις αιτίες που προκαλούν μειωμένη απόδοση, θα είχαμε την ακόλουθη σειρά [9]:

1. Απώλειες ταχύτητας
2. Αλλαγή κωδικού παραγωγής
3. Μικροσταματήματα
4. Βλάβες – Σταματήματα
5. Παραγωγή ελαττωματικών προϊόντων

Ως εκ τούτου, θα πρέπει πρώτα να γίνει αντιληπτή η φύση του προβλήματος και στη συνέχεια να καθοριστούν ρεαλιστικοί στόχοι. Όπως έχει ήδη αναφερθεί, η συνολική απόδοση του εξοπλισμού μετριέται σε μονάδες χρόνου και γι' αυτό θα πρέπει να γνωρίζουμε το τι μας στοιχίζει μια παύση ενός λεπτού, προκειμένου να ορίσουμε και τον χρόνο που μπορούμε να διαθέσουμε για την περάτωση του έργου βελτίωσης. Μια ανάλυση κόστους – κέρδους (Cost Benefit Analysis) μπορεί να μας υποδείξει τον βέλτιστο χρόνο, ωστόσο δεν πρέπει να λησμονείται το γεγονός ότι ο δείκτης ΟΕΕ δεν είναι οικονομικός και η μετατροπή του σε παραγωγικότητα και κατ' επέκταση σε χρηματικό όφελος δεν μπορεί να θεωρηθεί υπολογισμός με υψηλή ακρίβεια. Για τον λόγο αυτό, απαιτείται μια εκτίμηση του κόστους διακοπής της παραγωγής για ένα λεπτό. Για παράδειγμα, αν σε μια γραμμή οι απώλειες στοιχίζουν €100/λεπτό και ο χαμένος παραγωγικός χρόνος ημερησίως είναι 100 λεπτά, τότε το ημερήσιο κόστος των απωλειών καθορίζεται στα €10.000. Το ποσό αυτό αποτελεί μια ένδειξη για το όφελος που μπορεί να προέλθει από την επιτυχημένη εφαρμογή ενός έργου, μιας μεθοδολογίας συνεχούς βελτίωσης και κατά συνέπεια το κόστος των πόρων που μπορούν να διατεθούν σε αυτό.

Αφού λοιπόν προσδιορίσαμε την αντιστοιχία απωλειών – κόστους, το επόμενο βήμα είναι ο καθορισμός μιας συγκεκριμένης προσέγγισης για την επίλυση του δεδομένου προβλήματος. Στο παρακάτω Διάγραμμα, απεικονίζονται τα επιμέρους βήματα που απαρτίζουν μια ολοκληρωμένη εφαρμογή ενός έργου συνεχούς βελτίωσης. Πολύ σημαντική συνθήκη για την επιτυχημένη εφαρμογή του, είναι η τήρηση της διαδοχικότητας των βημάτων. Αν παραβιαστεί αυτή η συνθήκη, τότε τα αποτελέσματα της διαδικασίας δεν θα είναι ικανοποιητικά.

Αναλυτικά, τα βήματα ενός έργου Συνεχούς Βελτίωσης, είναι τα ακόλουθα:



1. Συλλογή Δεδομένων

Η συλλογή των δεδομένων θα πρέπει να γίνεται όσο το δυνατόν πιο «κοντά» στη γραμμή παραγωγής, ιδανικά από τους χειριστές των ίδιων των γραμμών. Παράλληλα, πολύ σημαντικό στοιχείο αποτελεί η μορφή της φόρμας συλλογής των δεδομένων, που θα πρέπει να είναι λιτή, περιεκτική και κατανοητή. Σε αντίθετη περίπτωση, όπου οι χειριστές θα δυσκολεύονταν να καταγράψουν τα δεδομένα, δεν θα υπήρχαν οι κατάλληλες βάσεις για την συνέχιση του έργου βελτίωσης.

2. Πληροφορία (ΟΕΕ)

Τα πρωτογενή δεδομένα που συλλέχθηκαν, απαιτούν αρχικά αξιολόγηση και εν συνεχεία ανάλυση, προκειμένου να εξαχθούν βάσιμα συμπεράσματα. Όπως και στο πρώτο βήμα της διαδικασίας, έτσι και σε αυτό, η απλότητα και η περιεκτικότητα των πληροφοριών είναι χαρακτηριστικά ιδιαίτερης σημασίας. Η πολυπλοκότητα και η εκτεταμένη ανάλυση των πληροφοριών είναι βέβαιο πως θα αποδυναμώσει το ενδιαφέρον και την εμπλοκή των χειριστών και κατά συνέπεια η πολύτιμη συμβολή τους θα ατονήσει. Η καλύτερη πρακτική, που εφαρμόζεται από βιομηχανίες με εμπειρία σε έργα συνεχούς βελτίωσης, είναι η άμεση μετατροπή των μετρήσεων σε πληροφορίες μέσω διαγραμμάτων. Ωστόσο για να συμβεί αυτό, προϋποθέτει γνώση των διαδικασιών αλλά και σωστό σχεδιασμό κατά την συλλογή των δεδομένων.

3. Ανάλυση Δεδομένων

Αποτελεί ουσιαστικά το στάδιο αποκρυπτογράφησης των πληροφοριών. Η διαδικασία της ανάλυσης, θα πρέπει να είναι προσεκτικά καθορισμένη και να ακολουθεί μια συγκεκριμένη τεχνική. Στόχος της, είναι η απόλυτη κατανόηση του προβλήματος και των αιτιών που το προκαλούν.

4. Ιδέες & Λύσεις

Η συλλογή ιδεών αποτελεί το αμέσως επόμενο βήμα της κατανόησης του προβλήματος. Μεθοδολογία για την άντληση ιδεών δεν μπορεί να εφαρμοστεί, ωστόσο μια σωστή διαχείριση των λύσεων που θα αναφερθούν, μπορεί να φανεί πολύ χρήσιμη για την συνέχεια του έργου. Για τον λόγο αυτό, προτείνεται η καταγραφή όλων των ιδεών και λύσεων που θα γεννηθούν σε μια συνάντηση, προκειμένου να δημιουργηθεί μια βάση – βοηθός για την επίλυση αντίστοιχων προβλημάτων.

5. Συμφωνία & Ενημέρωση

Πρόκειται για το στάδιο κατά το οποίο αρχικοποιούνται οι προθέσεις της ομάδας για την επίλυση του προβλήματος. Αξίζει να σημειωθεί η σημαντικότητα της συμφωνίας με τους υπόλοιπους χειριστές και εργαζόμενους στην παραγωγή για τις ενέργειες που πρόκειται να εκτελεστούν, ώστε να ελαχιστοποιηθούν οι πιθανότητες αποτυχίας μιας φαινομενικά καλής λύσης. Για το λόγο αυτό, μπορεί να έχει προηγηθεί μια αξιολόγηση των αιτιών αποτυχίας μια λύσης, προκειμένου να δοθεί έμφαση στους παράγοντες που την επηρεάζουν.

6. Προγραμματισμός & Εκτέλεση

Το βασικό στοιχείο αυτού του σταδίου είναι η αποφασιστικότητα. Δεν χρειάζονται χρονοτριβές και αναβολές στην εφαρμογή του πλάνου. Δεν είναι λίγες οι περιπτώσεις βιομηχανιών που η γραφειοκρατία των διαδικασιών τους στοίχισε εβδομάδες για την αγορά ενός νέου πάγκου εργασίας. Οι τεχνικές συνεχούς βελτίωσης στηρίζονται στην ενεργοποίηση του ανθρώπινου παράγοντα αλλά και στην αποφασιστικότητα της διοίκησης για αλλαγές που θα συμβάλουν στην αύξηση της αποδοτικότητας.

7. Έλεγχος και Αναφορά

Το τελευταίο στάδιο είναι αυτό του ελέγχου και της αναφοράς. Η διαδικασία συλλογής μετρήσεων πρέπει να διατηρηθεί έως ότου τα αποτελέσματα του έργου συνεχούς βελτίωσης αποτυπωθούν σε αριθμούς. Με τις νέες μετρήσεις της βελτιωμένης διαδικασίας, μπορεί να υπολογιστεί η μεταβολή των αποδόσεων και φυσικά να μετατραπεί σε οικονομικό όφελος για την εταιρία. Μια τέτοια συνολική αναφορά στη διοίκηση, που θα παρουσιάζει την αρχική κατάσταση, τις επεμβάσεις και φυσικά το τεκμηριωμένο αποτέλεσμα, αποτελεί ουσιαστικά, την επιτυχή ολοκλήρωση ενός έργου συνεχούς βελτίωσης. Συμπερασματικά, η επιτυχημένη εφαρμογή ενός έργου συνεχούς βελτίωσης εγγυείται σε δύο βασικούς παράγοντες. Αφ' ενός, στον έξυπνο καθορισμό του στόχου και αφ' ετέρου στην ολοκλήρωση σε σύντομο, προκαθορισμένο χρονικό διάστημα. Αν το πρόβλημα είναι πιο σύνθετο και κριθεί πως απαιτείται περισσότερος χρόνος, τότε προτείνεται η τμηματοποίηση του σε επιμέρους στόχους.

5. Υλοποίηση του συστήματος

Στο προηγούμενο κεφάλαιο παρουσιάστηκε σχηματικά η αρχιτεκτονική και ο σχεδιασμός του συστήματος προσομοίωσης και στη συνέχεια περιεγράφηκαν οι λειτουργίες των υποσυστημάτων που το αποτελούν. Στο κεφάλαιο αυτό γίνεται μια λεπτομερής περιγραφή της υλοποίησης κάθε υποσυστήματος, με σκοπό την κατανόηση της λειτουργίας ολόκληρου του λογισμικού. Στην πρώτη υποενότητα περιγράφεται η δομή της υλοποίησης και στις επόμενες υποενότητες αναπτύσσονται λεπτομερώς όλα τα συστατικά στοιχεία του λογισμικού προσομοίωσης που ανταποκρίνονται στις απαιτήσεις του συστήματος.

Το σύστημα προσομοίωσης που υλοποιείται στη παρούσα διπλωματική εργασία εστιάζει στην ανάπτυξη λογισμικού για γραμμές παραγωγής. Μπορεί να υλοποιηθεί και να εφαρμοστεί στο μεγαλύτερο ποσοστό των γραμμών παραγωγής και των παραγωγικών διαδικασιών που υπάρχουν σήμερα. Για την προσομοίωση των παραγωγικών διαδικασιών καθίσταται απαραίτητο το αλγοριθμικό κομμάτι του συστήματος το οποίο αποτελεί την «καρδιά» του συστήματος. Η διεπαφή χρήστη είναι μια λειτουργία που βοηθά και ενισχύει την οπτικοποίηση και την αποτύπωση της γραμμής παραγωγής σε ένα φιλικό περιβάλλον για τον χρήστη.

Αρχικά, ένας προσομοιωτής μπορεί να διαφοροποιηθεί με βάση τη χρήση για την οποία προορίζεται και τα δεδομένα που απαιτούνται για την ορθή λειτουργία του. Οι βασικές διαφορές μεταξύ του αλγόριθμου και της διεπαφής είναι ότι το σύστημα παράγει και επεξεργάζεται τα δεδομένα τα οποία είναι απαραίτητα για την οπτικοποίηση της παραγωγικής διαδικασίας.

5.1 Δομή υλοποίησης συστήματος

Όπως αναφέρθηκε σε προηγούμενα κεφάλαια η γλώσσα που υλοποιείται το σύστημα προσομοίωσης είναι η Python 3.9 και το περιβάλλον υλοποίησης που χρησιμοποιήθηκε είναι το **PyCharm**. Το PyCharm είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης που χρησιμοποιείται στον προγραμματισμό υπολογιστών, ειδικά για τη γλώσσα προγραμματισμού Python. Αναπτύχθηκε από την εταιρεία JetBrains.

Ο αλγόριθμος προσομοίωσης οργανώνεται στα εξής αρχεία – φακέλους:

Ο **Main** αποτελεί τον κύριο φάκελο που περιέχει όλα τα απαραίτητα αρχεία για την εκτέλεση του αλγορίθμου. Στο φάκελο αυτό:

- **images:** Είναι ο φάκελος που περιέχει όλες τις απαραίτητες φωτογραφίες του συστήματος. Οι φωτογραφίες αυτές αποτυπώνονται στο υποσύστημα προσομοίωσης και σε επιμέρους υποσυστήματα. Αναπαριστούν τις προσόψεις των μηχανημάτων της πραγματικής γραμμής παραγωγής.
- **output:** Ο output φάκελος περιέχει όλα τα αρχεία εξόδου που παράγει ο αλγόριθμος. Αυτά τα αρχεία είναι αρχεία δεδομένων και περιέχουν όλη την απαραίτητη πληροφορία του προσομοιωτή.
- **output.csv:** Είναι το αρχείο δεδομένων του συστήματος. Περιέχει την πληροφορία για τη συμπεριφορά της γραμμής, την οποία ο χρήστης μπορεί να διατηρήσει έπειτα

από την εκτέλεση της προσομοίωσης. Πιο συγκεκριμένα, οι τιμές καταγράφονται κάθε DT χρονικές στιγμές για όλα τα μηχανήματα της παραγωγής. Άρα, αν τα μηχανήματα της παραγωγής είναι X, οι εγγραφές θα είναι:

$$\#records = \frac{X * SIM_TIME}{DT}$$

Το αρχείο αποτελείται από **πέντε στήλες**. Η πρώτη αφορά τον χρόνο προσομοίωσης που δημιουργήθηκε αυτή η εγγραφή, η δεύτερη ορίζει το όνομα του μηχανήματος για το οποίο αναφέρεται, η τρίτη περιγράφει τον λόγο για τον οποίο βρέθηκε σε μία κατάσταση, η τέταρτη εμφανίζει το πλήθος των προϊόντων που έχουν παραχθεί από το συγκεκριμένο μηχάνημα τη δεδομένη στιγμή και η πέμπτη αφορά το status, την κατάσταση (run, stand by, stop) στην οποία βρίσκεται το μηχάνημα τη δεδομένη στιγμή προσομοίωσης.

- **Converters.py:** Είναι το υποσύστημα μετατροπών και επεξεργασίας δεδομένων. Αποτελείται από συναρτήσεις που τροποποιούν και επεξεργάζονται τα δεδομένα που παράγει ο προσομοιωτής. Έπειτα από την επεξεργασία τους, τα δεδομένα επιστρέφονται στο κυρίως πρόγραμμα για να εμφανιστούν στο υποσύστημα προσομοίωσης ως στατιστικά και μετρικές.
- **OutputCSV.py:** Ο αλγόριθμος αυτός είναι υπεύθυνος για τον έλεγχο του αρχείου εγγραφής δεδομένων. Εκτελείται κατά την έναρξη του αλγορίθμου προσομοίωσης και ο ρόλος του είναι να ελέγχει το output.csv αν υπάρχουν εγγραφές. Στη περίπτωση που εντοπίσει εγγραφές στο αρχείο, τις αδειάζει. Με αυτόν τον τρόπο δίνει την δυνατότητα στον προσομοιωτή να εγγράψει τις νέες τιμές των μεταβλητών, χωρίς να εμπεριέχει τις τιμές των μεταβλητών της προηγούμενης εκτέλεσης.
- **WindowTkinter.py:** Το πρόγραμμα αυτό αφορά αποκλειστικά το γραφικό περιβάλλον και την απεικόνιση των παραμέτρων του συστήματος. Αποτελείται από συναρτήσεις, όπου η κάθε μια είναι ένα ξεχωριστό παράθυρο διαφορετικής λειτουργίας. Ο χρήστης μέσα από το κάθε παράθυρο γραφικού περιβάλλοντος, κατανοεί πλήρως την λειτουργία της προσομοίωσης, την ανάλυση των μηχανημάτων και της παραγωγής. Περιέχει τα συστήματα ενημέρωσης του χρήστη σε γραφικό περιβάλλον, δηλαδή στο σύστημα εισαγωγής βασικών παραμέτρων.
- **mainDiploma.py:** Είναι το κύριο πρόγραμμα προσομοίωσης του συστήματος. Το αρχείο αυτό αφορά τον αλγόριθμο ο οποίος είναι υπεύθυνος για την προσομοίωση της παραγωγικής διαδικασίας και για ένα μέρος του γραφικού περιβάλλοντος του υποσυστήματος live monitoring. Το **mainDiploma** αποτελείται από 1200+ γραμμές κώδικα και έχουν φτιαχτεί και βελτιστοποιηθεί για να αναπαριστούν πιστά την πραγματική παραγωγική διαδικασία που προσομοιώνεται. Περιλαμβάνει το μεγαλύτερο αριθμό των υποσυστημάτων. Πιο συγκεκριμένα περιλαμβάνει το υποσύστημα προσομοίωσης, βλαβών, καταγραφής δεδομένων, live monitoring και ολοκλήρωσης προσομοίωσης.

Στις παρακάτω υποενότητες παρουσιάζεται ο τρόπος λειτουργίας του κάθε μέρους της διπλωματικής εργασίας ξεχωριστά. Επίσης πραγματοποιούνται εκτενείς αναφορές σε πολλά σημαντικά σημεία της υλοποίησης. Στις προηγούμενες ενότητες αναλύθηκαν η γλώσσα προγραμματισμού, τα εργαλεία που χρησιμοποιήθηκαν και τα υποσυστήματα που απαρτίζουν τον αλγόριθμο στο σύνολο της ανάπτυξης του συστήματος προσομοίωσης.

5.2 Υποσύστημα εισαγωγής βασικών παραμέτρων

Το υποσύστημα εισαγωγής βασικών παραμέτρων είναι το παράθυρο που αποτελεί το γραφικό περιβάλλον, φιλικό προς τον χρήστη. Σε αυτό, ο χρήστης εισάγει τις βασικές παραμέτρους του προσομοιωτή. Περιέχεται στο αρχείο WindowTkinter.py και εκτελείται πριν την έναρξη της προσομοίωσης.

```
def welcomeScreen()
```

Η συνάρτηση αυτή εκτελείται με την εκκίνηση του προγράμματος προσομοίωσης. Είναι η πρώτη συνάρτηση που εκτελείται στο πρόγραμμα και εμφανίζει στον χρήστη το γραφικό περιβάλλον εισαγωγής παραμέτρων. Το υποσύστημα αποτελείται από δυο τμήματα. Στο πρώτο τμήμα ο χρήστης εισάγει το στατικό χρόνο βήματος DT και το συνολικό διάστημα προσομοίωσης (T ή SIM_TIME).

Το δεύτερο τμήμα αφορά την εμφάνιση βλαβών είτε χειροκίνητες είτε αυτόματες, με την επιλογή ενός checkbox. Αν η επιλογή είναι αυτόματη εμφάνιση βλαβών στα μηχανήματα, τότε ο χειριστής πρέπει να εισάγει την πιθανότητα για κάθε μηχανήμα σύμφωνα με την οποία επιθυμεί να τεθεί κάθε μηχανήμα σε κατάσταση stop.

Η συνάρτηση έχει προγραμματιστεί με κώδικα για την εμφάνιση γραφικού περιβάλλοντος που βασίζεται στην βιβλιοθήκη tkinter. Δεν έχει καμία παράμετρο ως όρισμα και η λειτουργία της είναι αφενός να εμφανίζει το γραφικό περιβάλλον και αφετέρου, να επιστρέφει τις τιμές των μεταβλητών που καταχώρησε ο χρήστης, στο κυρίως πρόγραμμα. Τέλος, περιέχει το κουμπί εκκίνησης, το οποίο, αφού ο χρήστης έχει εισάγει τα απαραίτητα δεδομένα, το επιλέγει και ξεκινάει η προσομοίωση. Σε περίπτωση που ο χρήστης δεν εισάγει τα απαραίτητα δεδομένα, το σύστημα επιστρέφει μήνυμα λάθους.

Στο εσωτερικό της περιλαμβάνει την συνάρτηση:

```
def probability_set(machine_name, row)
```

Η συνάρτηση probability_set επιστρέφει την τιμή της πιθανότητας εμφάνισης βλαβών για κάθε μηχανήμα. Λειτουργεί σαν μια γενική συνάρτηση κληρονομικής παραμέτρων και ορίζεται για κάθε μηχανήμα που υπάρχει στη παραγωγική διαδικασία. Τέλος, όπως προαναφέρθηκε, αυτές οι τιμές επιστρέφονται από την κεντρική συνάρτηση στο κυρίως πρόγραμμα.

Στο παρακάτω σχήμα απεικονίζεται ενδεικτικά το γραφικό περιβάλλον 3 μηχανημάτων που απαρτίζουν τη γραμμή παραγωγής.

Εικόνα 5.1: Welcome Screen

5.3 Υποσύστημα καταχώρησης δεδομένων

Ιδανικά το υποσύστημα καταχώρησης δεδομένων θα υλοποιούνταν με την τεχνική της εισαγωγής αρχείου δεδομένων. Η υλοποίηση αυτή όμως αποτελεί μελλοντική επέκταση για το παρόν σύστημα προσομοίωσης. Στη παρούσα διπλωματική, οι παράμετροι των μηχανημάτων αρχικοποιούνται στην αρχή του αλγορίθμου και αποτελούν global μεταβλητές για ολόκληρο το πρόγραμμα. Οι παράμετροι αυτοί έχουν περιγραφεί αναλυτικά στο κεφάλαιο 3.1 «Παράμετροι συστήματος».

```
# ---Machine Name---
# standard vars
machine_capacity = 50000
initial_machineProd = 0
machine_batch = 5000
current_machine_level = 0
machine_critical_buffer = machine_batch
# dynamic vars
MACHINE_SPEED = PRODUCTION_SPEED # product / second
machine_input = machine_batch # input prod/sec
machine_output = 1 # output cans/sec
machine_ptime = machine_output / MACHINE_SPEED # Process Time
MTBF_machine = 31.03 # standard error in min (MTBS)
# status
machine_status = 'green'
machine_description = "Έναρξη παραγωγής"
```

Επιπλέον απαραίτητες για την έναρξη της προσομοίωσης, είναι οι γενικές μεταβλητές του συστήματος:

```
# --GENERAL VARIABLES--  
RANDOM_SEED = 42  
REPAIR_DURATION = 10.0  
NUM_MACHINES = 5
```

Η παράμετρος `RANDOM_SEED` χρησιμοποιείται για την αναπαραγωγή διαφορετικών δεδομένων από τις παραμέτρους που χρησιμοποιούν την βιβλιοθήκη `random` κατά την εκτέλεση της προσομοίωσης. Ο ρόλος της είναι να εισάγει την τιμή στην συνάρτηση `random.seed(RANDOM_SEED)`.

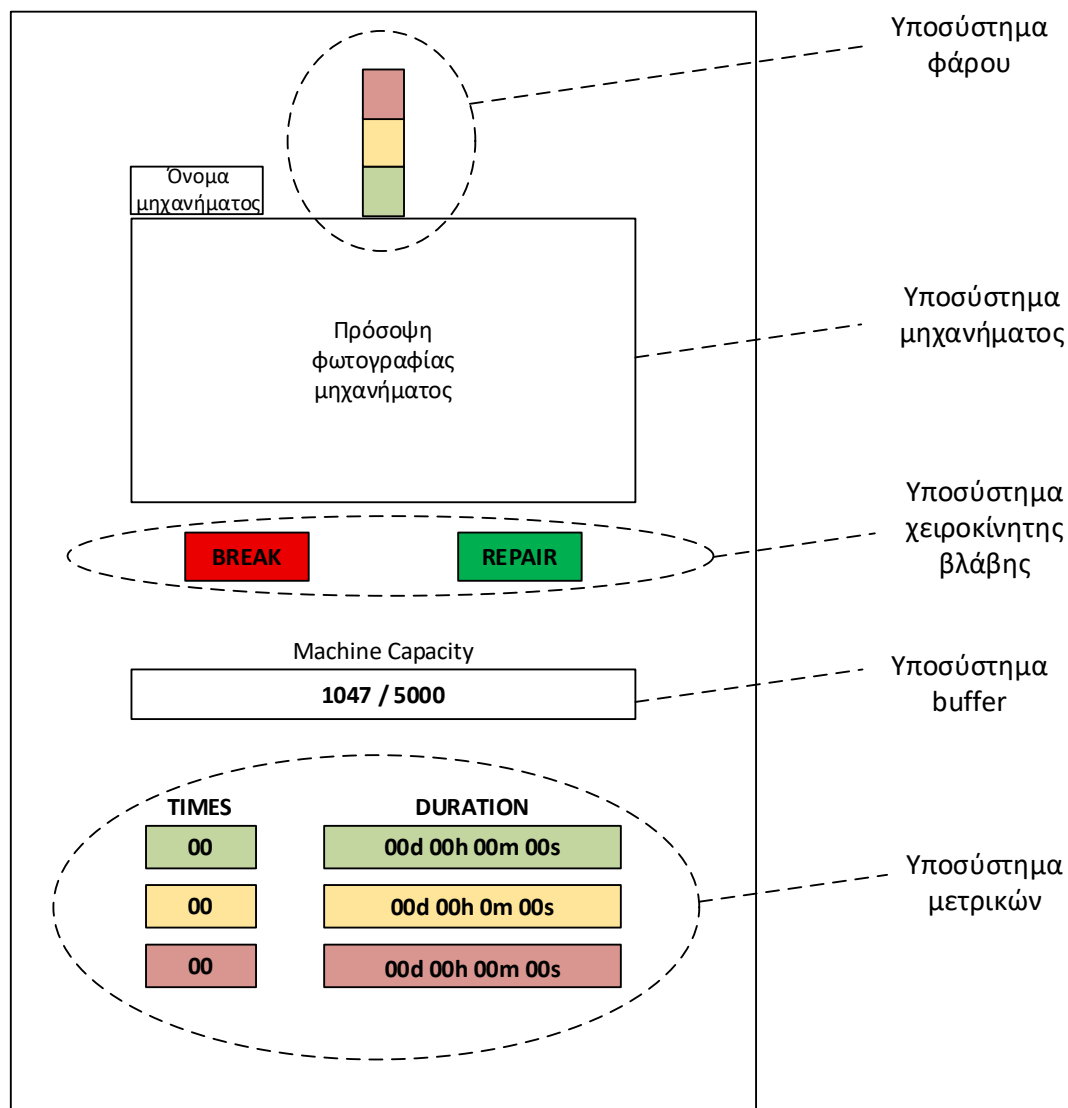
Η `REPAIR_DURATION` είναι μια παράμετρος που ορίζει τα λεπτά για το οποία ένα μηχάνημα χρειάζεται να επισκευαστεί. Έστω ότι εμφανίστηκε μια βλάβη σε κάποιο μηχάνημα, ο χειριστής του μηχανήματος θα χρειαστεί κατά μέσο όρο `REPAIR_DURATION` δευτερόλεπτα για να επαναφέρει το μηχάνημα σε κατάσταση παραγωγής (`run`).

Το `NUM_MACHINES` ορίζει το πλήθος των μηχανημάτων που υπάρχουν στη γραμμή παραγωγής ανεξάρτητα από τις μεταφορικές γραμμές. Ο σκοπός του είναι για να εισάγει στον αλγόριθμο το πλήθος των οντοτήτων που θα χρειαστεί για να κατασκευάσει το γραφικό περιβάλλον. Επίσης, επιστρέφεται σαν τιμή στον προγραμματιστή για τον ορθή απεικόνιση της παραγωγής στον αλγόριθμο.

Το υποσύστημα εισαγωγής βασικών παραμέτρων με το υποσύστημα καταχώρησης δεδομένων συσχετίζονται, αφού η καταχώρηση των βασικών δεδομένων της προσομοίωσης γίνεται από το γραφικό περιβάλλον του υποσυστήματος εισαγωγής παραμέτρων.

5.4 Υποσύστημα προσομοίωσης

Είναι το κύριο μέρος του αλγορίθμου προσομοίωσης στο οποίο επιτελούνται όλες οι διαδικασίες της παραγωγής. Περιλαμβάνεται στο αρχείο `mainDiploma.py`. Η υλοποίηση του διαχωρίζεται σε **τέσσερα υποσυστήματα** και αποτελούν την οντότητα – κλάση του προσομοιωτή. Στον όρο οντότητα περιλαμβάνονται τα υποσυστήματα του φάρου, μηχανήματος, `buffer` και μετρικών. Περιλαμβάνεται επίσης και η γραφική απεικόνιση του υποσυστήματος βλαβών με την χειροκίνητη επιλογή. Με τον όρο οντότητα, στο γραφικό περιβάλλον της προσομοίωσης ορίζεται το αντικείμενο που περιλαμβάνει τα γραφικά `components` των τεσσάρων παραπάνω υποσυστημάτων. Αυτή η τεχνική σχετίζεται με την τεχνική της δημιουργίας κλάσεων. Στο πρόγραμμα κάθε διεργασία μηχανήματος υλοποιείται με κληρονομικότητα από την κύρια κλάση `class ProductionClass`, η οποία περιγράφεται στο υποσύστημα μηχανήματος. Μέσω αυτής της τεχνικής, δίνεται η ευκολία στο πρόγραμμα να επεκτείνει είτε να μειώνει την γραμμή παραγωγής που προσομοιώνει. Δηλαδή με άμεσο τρόπο μπορεί να προστεθεί είτε να αφαιρεθεί στον προσομοιωτή οποιοδήποτε μηχάνημα. Ο σκοπός της τεχνικής της δημιουργίας οντοτήτων είναι να αναπροσαρμόζεται με ευκολία η προσομοίωση της παραγωγικής διαδικασίας. Με αυτό τον τρόπο επίσης επιτυγχάνεται η βελτιστοποίηση του κώδικα προγραμματισμού και μειώνεται η υπολογιστική πολυπλοκότητα του συστήματος. Στο σχήμα 5.1 αναπαρίσταται η οντότητα που υλοποιείται στα πλαίσια της διπλωματικής.



Σχήμα 5.1: Οντότητα μηχανήματος

Σε αυτό το σημείο σημαντικό είναι να οριστεί το **περιβάλλον** υλοποίησης της προσομοίωσης δηλαδή η **κλάση** του αλγορίθμου. Η κλάση αρχικοποιεί κάθε διεργασία του προγράμματος και είναι υπεύθυνη για την έναρξη της προσομοίωσης. Στη συνέχεια, δίνεται η δομή της κλάσης που απαρτίζει μια απλή γραμμή παραγωγής τριών μηχανημάτων.

```
class ProductionName(object):
    def __init__(self):
        self.env = env

    # Raw material
    self.rawProd = simply.Container(env, capacity=rawProd_capacity,
        init=initial_rawProd)
```

```

# Machine -1- initialization
self.machin1Prod = simpy.Container(env, capacity=machine1_capacity,
init=initial_machine1)
self.machine1Buffer = env.process(self.machine1_buffer())
self.machine1Process = env.process(self.machine1_process())

# Machine -1-2- Conveyor
self.machine1_2Conv = env.process(self.machine1_machine2_conveyor
())

# Machine -2- initialization
self.machin2Prod = simpy.Container(env, capacity=machine2_capacity,
init=initial_machine2)
self.machine2Buffer = env.process(self.machine2_buffer())
self.machine2Process = env.process(self.machine2_process())

# Machine -2-3- Conveyor
self.machine2_3Conv = env.process(self.machine2_machine3_conveyor
())

# Machine -3- initialization
self.machin3Prod = simpy.Container(env, capacity=machine3_capacity,
init=initial_machine3)
self.machine3Buffer = env.process(self.machine3_buffer())
self.machine3Process = env.process(self.machine3_process())

```

Η κλάση αρχικοποιεί τις διεργασίες τριών μηχανημάτων, τις διεργασίες δύο μεταφορικών γραμμών και τρία Container των τριών μηχανημάτων. Επιπλέον, αρχικοποιείται το Container της πρώτης ύλης. Πιο αναλυτικά, ορίζεται το Container της πρώτης ύλης (rawProd) το οποίο έχει χωρητικότητα (rawProd_capacity) και υπάρχει απόθεμα (initial_rawProd). Το Container περιγράφει το πρώτο προϊόν της παραγωγής που είναι διαθέσιμο προς επεξεργασία. Αυτό το προϊόν καλείται rawProd και είναι έτοιμο να εισαχθεί στο machine 1 ώστε να επεξεργαστεί.

Στη συνέχεια ορίζονται όλες οι απαραίτητες διεργασίες του μηχανήματος 1. Αρχικά, κάθε μηχανήμα εισάγει ένα συγκεκριμένο τύπο προϊόντος και εξάγει ένα διαφορετικό, επεξεργασμένο προϊόν. Για τον λόγο αυτό, ορίζεται το **Container** του machine1Prod το οποίο αποτελεί την επεξεργασία που έχει επιτελέσει το μηχανήμα 1 στο προϊόν rawProd. Το προϊόν machine1Prod που εξάγεται από το μηχανήμα 1 είναι έτοιμο να εισαχθεί στο επόμενο μηχανήμα (μηχανήμα 2) για να επεξεργαστεί. Για το ίδιο μηχανήμα είναι απαραίτητο να οριστεί το υποσύστημα του **buffer** του. Το υποσύστημα αυτό, όπως έχει ήδη αναλυθεί, αποτελεί το σύστημα ελέγχου του μηχανήματος 2. Η διεργασία που επιτελεί είναι να ελέγχει το πλήθος των αποθεμάτων που υπάρχει στο μηχανήμα 1. Τροφοδοτεί και συντονίζει το προϊόν, δηλαδή την πρώτη ύλη, του μηχανήματος 1. Η συνάρτηση της διεργασίας του buffer ορίζεται από την συνάρτηση machine1Buffer, και την διεργασία τη επιτελεί η συνάρτηση machine1_buffer().

Επίσης, σημαντικό και καθοριστικό για την ορθή λειτουργία της γραμμής παραγωγής είναι να οριστούν οι **διαδικασίες** που επιτελούν τα **μηχανήματα** της παραγωγής. Η διεργασία του μηχανήματος ορίζεται από τη συνάρτηση machine1Process και η διεργασία του εκτελείται από τη συνάρτηση machine1_process(). Η διεργασία των μηχανημάτων περιγράφεται στο κεφάλαιο του υποσυστήματος διεργασίας μηχανήματος και αναλύεται η υλοποίηση της διαδικασίας επεξεργασία προϊόντων. Η διαδικασία του μηχανήματος 1 είναι

να εισάγει τη πρώτη ύλη (rawProd), να την επεξεργάζεται και έπειτα να εξάγει το επεξεργασμένο προϊόν machine1Prod.

Τέλος, υπεύθυνο για την μεταφορά των προϊόντων είναι το υποσύστημα μεταφορικών γραμμών, οι μεταφορικές γραμμές δηλαδή της παραγωγικής διαδικασίας. Η μεταφορική γραμμή εξάγει τα επεξεργασμένα προϊόντα από το μηχάνημα 1 και τα εισάγει στο μηχάνημα 2. Η διεργασία της ορίζεται από τη συνάρτηση machine1_2Conv και η διεργασία της επιτελείται από τη συνάρτηση machine1_machine2_conveyro().

Στη συνέχεια αναλύονται όλα τα υποσυστήματα της προσομοίωσης και προσδιορίζεται η υλοποίησή τους.

5.4.1 Υποσύστημα φάρου

Το υποσύστημα του φάρου αποτελεί την προσομοίωση ενός πραγματικού φάρου. Η λειτουργία του είναι να εκπέμπει το αντίστοιχο χρώμα σύμφωνα με την κατάσταση στην οποία είναι το μηχάνημα. Η κόκκινη ένδειξη ορίζει την κατάσταση του μηχανήματος σε βλάβη (stop), η κίτρινη ορίζει την κατάσταση αναμονής (stand by) του μηχανήματος χωρίς να παράγει και το πράσινο ορίζει την ομαλή λειτουργία (run) του μηχανήματος δηλαδή τη κατάσταση παραγωγής. Στη παρακάτω φωτογραφία απεικονίζεται η αντιστοιχία του πραγματικού φάρου με τον φάρο προσομοίωσης.



Εικόνα 5.2: Πραγματικός φάρος – Φάρος προσομοίωσης

```
def machine_beacon(col, machine_name)
    # Beacon
    LB3 = Label(window, bg='#FAB3B3', bd=4, width=3, height=2,
relief=RAISED) # Red
    LB3.grid(row=1, column=col, padx=180, sticky=W)
    LB2 = Label(window, bg='#FBF3C1', bd=4, width=3, height=2,
relief=RAISED) # Yellow
    LB2.grid(row=2, column=col, padx=180, sticky=W)
    LB1 = Label(window, bg='#D9F2CC', bd=4, width=3, height=2,
relief=RAISED) # Green
    LB1.grid(row=3, column=col, padx=180, sticky=W)
    # Machine Name
    Button(window, bg='#8DCDBA', bd=4, relief=RAISED,
text=machine_name, font="Arial 13 bold").grid(row=3, column=col,
padx=20, sticky=W)

    return LB3, LB2, LB1
```

Για την υλοποίηση της προσομοίωσης του φάρου είναι απαραίτητο να κληθεί η συνάρτηση `machine_beacon` για κάθε οντότητα - μηχανήμα. Η συγκεκριμένη συνάρτηση αφορά αποκλειστικά την γραφική απεικόνιση τους φάρου και τοποθετείται στο γραφικό περιβάλλον πάνω από την πρόσοψη του μηχανήματος, όπως φαίνεται και στο διάγραμμα της οντότητας. Το όρισμα `col` αφορά την στήλη στην οποία θα τοποθετηθεί ο φάρος στο αντίστοιχο μηχανήμα σύμφωνα με την αλληλουχία της πραγματικής γραμμής παραγωγής. Η παράμετρος `machine_name` αφορά την ονομασία του μηχανήματος η οποία είναι απαραίτητη για τον διαχωρισμό του από τα υπόλοιπα. Υπεύθυνη για την μετάβαση των καταστάσεων και την live απεικόνιση των καταστάσεων είναι η συνάρτηση `update()`.

```
def update()
```

Η συγκεκριμένη συνάρτηση αναπαριστά σε live απεικόνιση την προσομοίωση μιας παραγωγικής διαδικασίας. Είναι μια συνάρτηση στην οποία καλούνται οι υποσυναρτήσεις των επιμέρους υποσυστημάτων. Εκτελείται σύμφωνα με τον στατικό χρόνο βήματος (DT). Δηλαδή κάθε DT χρονικές στιγμές ανανεώνεται το γραφικό περιβάλλον της προσομοίωσης και επιστρέφονται οι νέες τιμές και οι γραφικές απεικονίσεις των συναρτήσεων που καλούνται στο εσωτερικό της ίδιας. Για παράδειγμα, αν ένα μηχανήμα μεταβεί από την κατάσταση `run` στη κατάσταση `stop` η συνάρτηση `update()` θα θέσει σε κατάσταση `off` την πράσινη ένδειξη και θα ορίσει την κόκκινη σε `on`, με απλά λόγια θα σβήσει η ένδειξη του πράσινου χρώματος και θα ανάψει το κόκκινο.

Η δομή της συνάρτησης καθορίζεται από την δομή της πραγματικής γραμμής παραγωγής που προσομοιώνεται. Πριν από κάθε υποσύστημα μηχανήματος υπάρχει το υποσύστημα του `buffer`. Στο γραφικό περιβάλλον κάτω από κάθε υποσύστημα μηχανήματος απεικονίζεται το υποσύστημα των μετρικών. Πάνω από τα μηχανήματα τοποθετείται το υποσύστημα του φάρου. Η αρχιτεκτονική του γραφικού περιβάλλοντος ορίζεται στο κεφάλαιο 4.1 Αρχιτεκτονική συστήματος.

5.4.2 Υποσύστημα μεταφορικής γραμμής

Το υποσύστημα μεταφορικής γραμμής υλοποιείται με σκοπό να ελέγχει συνεχώς την κατάσταση και το απόθεμα του μηχανήματος που σερβίρει – εξάγει το προϊόν. Η λειτουργία της είναι να εντοπίζει το πλήθος του αποθέματος του μηχανήματος προτού γεμίσει είτε εμφανιστεί βλάβη στο ίδιο το μηχανήμα. Στην περίπτωση που ο `buffer` του μηχανήματος είναι γεμάτος, δηλαδή το προϊόν που είναι σε αναμονή για να εισαχθεί στο μηχανήμα και να επεξεργαστεί έχει φτάσει στο ανώτατο όριο της χωρητικότητας, τότε το υποσύστημα της μεταφορικής γραμμής θέτει σε κατάσταση αναμονής το προηγούμενο μηχανήμα. Στη συνέχεια, δίνεται η δομή της συνάρτησης της μεταφορικής γραμμής.

```
def machine1_machine2_conveyor(self):  
    # Ορισμός global μεταβλητών  
  
    machine1_standby_start = 0  
    flag = False  
  
    while True:  
        yield env.timeout(1)  
        if not flag and self.machine1Prod.level == machine2_capacity and  
            machine2_status != Status.green.name:  
            machine1_standby_start = env.now
```

```

# Set machine1 Status
machine1_status = Status.yellow.name
machine1_description = Status.yellow.value
yield env.timeout(0)

flag = True

elif flag and depall_status == Status.green.name:
    MACHINE1_RUN_TIMES += 1
    MACHINE1_STANDBY_TIMES += 1
    MACHINE1_STANDBY_DURATION[machine1_standby_start] = env.now -
machine1_standby_start
    flag = False

```

Η παραπάνω συνάρτηση εκτελείται διαρκώς κατά τη διάρκεια της προσομοίωσης αφού διαθέτει τον βρόγχο `while True`. Στην αρχή ορίζονται οι `global` μεταβλητές που είναι υπεύθυνες για το υποσύστημα των μετρικών, αποτυπώνουν δηλαδή τα στατιστικά και την κατάσταση του μηχανήματος. Η συνάρτηση πρώτα ελέγχει αν το πλήθος των προϊόντων του μηχανήματος 1, το μηχανήμα πριν τη μεταφορική γραμμή, έχει φτάσει στο όριο της χωρητικότητας του μηχανήματος 2. Παράλληλα, ελέγχει την κατάσταση του μηχανήματος 2, έτσι ώστε σε περίπτωση που δεν βρίσκεται σε κατάσταση παραγωγής (`run`), ο αλγόριθμος να κατανοήσει ότι το μηχανήμα 2 δεν παράγει. Η περίπτωση αυτή δηλώνει ότι ο `buffer` του μηχανήματος 2 έχει γεμίσει και δεν μπορεί να δεχτεί άλλα προϊόντα. Άρα, έχει ως αποτέλεσμα, ο αλγόριθμος να δώσει την εντολή στο μηχανήμα 1 να σταματήσει την παραγωγή μιας και το παραγόμενο προϊόν που εξάγει δεν μπορεί να μεταφερθεί στο επόμενο μηχανήμα (2). Έτσι, το μηχανήμα 1 τίθεται σε κατάσταση αναμονής (`stand by`) μέχρις ότου το απόθεμα του μηχανήματος 2 να αρχίσει να μειώνεται.

Η ίδια συνάρτηση περιέχει την λειτουργία της καταχώρησης δεδομένων. Συλλέγει τα δεδομένα για την κατάσταση του μηχανήματος 1 και τα καταχωρεί στο υποσύστημα `live` προσομοίωσης για την απεικόνισή τους. Πιο συγκεκριμένα, ορίζει και καταχωρεί ως δεδομένο το πλήθος για που το μηχανήμα τέθηκε σε κατάσταση `run` και `stand by` καθώς επίσης και τη διάρκεια που το μηχανήμα 1 ήταν σε αναμονή (`stand by`).

Η γραφική απεικόνιση των μετρικών της μεταφορικής γραμμής γίνεται από την συνάρτηση:

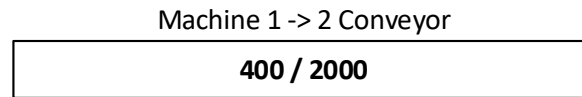
```

def current_conveyor_capacity_buffer (col, i_padx, bg_colour,
current_level, machine_capacity, machine_description):
    # Current level capacity status of conveyors
    Label(window, text=machine_description, font="Arial 14",
bg=window_colour, wraplength=140).grid(row=6, column=col, pady=0,
sticky=S)
    Label(window, text=str(current_level) + ' / ' +
str(machine_capacity), bg=bg_colour, relief=RIDGE, font="Arial 15
bold", justify=CENTER).grid(row=7, column=col, ipadx=i_padx,
ipady=15)

```

Η συνάρτηση χρησιμοποιείται από κάθε οντότητα ξεχωριστά με τα αντίστοιχα ορίσματά της. Καλείται στο εσωτερικό της συνάρτησης `update()` που υπάρχει στο υποσύστημα `live monitoring`. Το όρισμα `col`, αφορά την στήλη που θα τοποθετηθεί στο γραφικό περιβάλλον, το `current_level` είναι η τιμή των προϊόντων που μεταφέρει την εκάστοτε στιγμή η μεταφορική και η οποία αλλάζει σε πραγματικό χρόνο καθώς η προσομοίωση εκτελείται.

Επίσης, ορίζονται το `machine_capacity` που είναι το όριο των προϊόντων που χωράει η μεταφορική και τέλος χρησιμοποιείται η περιγραφή της μεταφορικής γραμμής για να διευκρινίζει στον χρήστη τα ονόματα των μετρικών. Η γραφική απεικόνιση του υποσυστήματος είναι:



5.4.3 Υποσύστημα buffer

Κάθε υποσύστημα διεργασίας μηχανήματος διαθέτει ένα υποσύστημα buffer. Ο buffer υλοποιήθηκε για να ελέγχει διαρκώς το πλήθος των αποθεμάτων που έχει το μηχάνημα για το οποίο είναι υπεύθυνος. Σκοπός του είναι να ορίζει τη κατάσταση του μηχανήματος σε stand by σε περίπτωση που το μηχάνημα δεν έχει προϊόν να εισάγει προς επεξεργασία. Η δομή της συνάρτησης ορίζεται ως:

```
def machine_batch_buffer(self):
# Ορισμός global μεταβλητών

yield env.timeout(machine1_ptime + machine2_ptime + 4)

while True:
    if self.machine2Prod.level < machine2_critical_buffer:

        if MACHINE2_STANDBY_FLAG:
            MACHINE2_STANDBY_TIMES += 1
            MACHINE2_STANDBY_FLAG = False

        # Set Buffer Status
        machine2_status = Status.yellow.name
        machine2_description = Status.yellow.value

        yield env.timeout(machine1_ptime + 2)

        # Stand by απο Machine 1
        if machine1_status == Status.red.name:
            MACHINE1_STANDBY_DURATION[machine2_standby_start] = env.now -
machine2_standby_start

    else:
        yield env.timeout(machine1_ptime)
```

Αρχικά, η συνάρτηση του buffer ελέγχει διαρκώς το επίπεδο των προϊόντων που παράγει το μηχάνημα κατά τη διάρκεια της προσομοίωσης με το κρίσιμο σημείο που έχει ορίσει ο προγραμματιστής. Το κρίσιμο σημείο είναι το ελάχιστο πλήθος προϊόντων που χρειάζεται το μηχάνημα για να εισάγει στη παραγωγική του διαδικασία. Είναι δηλαδή το ελάχιστο πλήθος που πρέπει να έχει ώστε να βρίσκεται σε κατάσταση παραγωγής. Ο έλεγχος ξεκινάει αφού έχει αρχίσει η παραγωγή όλων των προηγούμενων μηχανημάτων και εκτελείται διαρκώς.

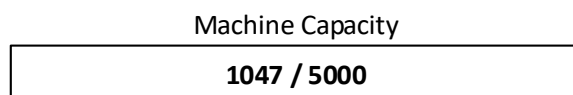
Αυτή η καθυστέρηση εισάγεται διότι διαφορετικά, με την εκκίνηση του αλγορίθμου προσομοίωσης, το μηχάνημα θα βρισκόταν ήδη στη κατάσταση αναμονής.

Στη συνέχεια, η συνάρτηση ελέγχει την κατάσταση του μηχανήματος. Αν είναι σε κατάσταση παραγωγής (run) και το απόθεμα είναι μικρότερο από το κρίσιμο σημείο τότε το μηχάνημα τίθεται σε κατάσταση αναμονής (stand by). Και περιμένει σε αυτή την κατάσταση έως ότου υπάρχουν ξανά διαθέσιμα προϊόντα προς επεξεργασία. Τέλος, ελέγχει και τη κατάσταση του προηγούμενου μηχανήματος το οποίο συμβάλει στην υπολογισμό του χρόνου αναμονής στον οποίο βρέθηκε το μηχάνημα.

Η απεικόνιση του υποσυστήματος buffer γίνεται από το υποσύστημα live monitoring και έχει την δομή:

```
def current_machine_buffer(col, i_padx, bg_colour, current_level,
machine_capacity, machine_description):
    # Current machine level capacity
    Label(window, text=machine_description, font="Arial 14",
bg=window_colour).grid(row=6, column=col, pady=0, sticky=S)
    Label(window, text=str(current_level) + ' / ' +
str(machine_capacity), bg=bg_colour, relief=RIDGE, font="Arial 15
bold", justify=CENTER).grid(row=7, column=col, pady=0,
ipadx=i_padx, ipady=15)
```

Η λειτουργία της συνάρτησης είναι να απεικονίζει γραφικά το πλήθος των προϊόντων που παράγει το μηχάνημα κάθε στιγμή κατά τη διάρκεια της προσομοίωσης και τη συνολική χωρητικότητα του μηχανήματος. Η απεικόνιση είναι:



5.4.4 Υποσύστημα διεργασίας μηχανήματος

Το υποσύστημα αυτό είναι πολύ σημαντικό για την εκτέλεση του αλγορίθμου προσομοίωσης και ορίζει την διεργασία που επιτελεί το μηχάνημα. Δηλαδή η διαδικασία με την οποία επεξεργάζεται και παράγει τα προϊόντα. Η διεργασία αυτή διαφέρει από μηχανήμα σε μηχανήμα και ορίζεται από τον κατασκευαστή του. Ο προγραμματισμός της προσομοιώνει τη πραγματική παραγωγική διαδικασία που επιτελεί το μηχάνημα. Η κεντρική δομή που ορίζεται για μια διεργασία μηχανήματος είναι η παρακάτω.

```
def machine_process():
# Ορισμός global μεταβλητών

while True:
    try:
        if production_class.rawProd.level >= machine1_batch and not
MACHINE1_IS_BROKEN:
            machine1_run_start = env.now
            yield env.timeout(0) # Τα βάζει ακαριαία η μεταφορική
            machine1_status = Status.green.name
            machine1_description = Status.green.value
            yield production_class.rawProd.get(machine1_input)
```

```

for i in range(machine1_batch):
    current_machine1_level = machine1_batch - i
    # Set Machine Status
    machine1_status = Status.green.name
    machine1_description = Status.green.value

    yield env.timeout(machine1_ptime)
    yield production_class.machine1Prod.put(machine1_output)

if MACHINE1_STOP_FLAG:
    MACHINE1_RUN_DURATION.append(env.now - machine1_run_start)

else:
    yield env.timeout(0)

except simpy.Interrupt:
    yield env.timeout(0.01)
    print("Machine 1 has stopped")
    start_machine1_break = env.now
    machine1_status = Status.red.name
    machine1_description = Status.red.value
    MACHINE1_STOP_TIMES += 1
    MACHINE2_STANDBY_FLAG = MACHINE3_STANDBY_FLAG = True

# Manual Breakdown
if AUTO_BRAKE_VAR == 0:
    while MACHINE1_STOP_FLAG:
        # Set Machine Status
        yield env.timeout(0.01)
        machine1_status = Status.red.name
        machine1_description = Status.red.value

        MACHINE1_STOP_FLAG = True

# Automated Breakdown
else:
    # Brake Duration. (Repair After 5-30 min)
    repair_at = start_machine1_break + random.uniform(300, 1800)

    while env.now <= repair_at:
        yield env.timeout(0.01)
        machine1_status = Status.red.name
        machine1_description = Status.red.value

    MACHINE1_STOP_DURATION[start_machine1_break] = env.now - start_
machine1_break
    MACHINE1_IS_BROKEN = False
    IN_BREAKDOWN = False

```

Στη παραπάνω υλοποίηση του κώδικα δομείται η συνάρτηση μια διεργασίας μηχανήματος που αποτελεί μέρος μιας παραγωγικής διαδικασίας τριών μηχανημάτων. Αρχικά, στη συνάρτηση ορίζονται οι global μεταβλητές που είναι υπεύθυνες για την κατάσταση και τις μετρικές του μηχανήματος. Στη συνέχεια, εκτελείται η διεργασία του μηχανήματος μέσω του άπειρου βρόγχου `while True`. Στη διεργασία, ελέγχεται εάν το επίπεδο του πλήθους των προϊόντων του προηγούμενου μηχανήματος είναι μεγαλύτερο ή ίσο από το ελάχιστο πλήθος που μπορεί να επεξεργαστεί το μηχάνημα. Το `machine1_batch` ορίζει δηλαδή την ελάχιστη ποσότητα που μπορεί να εισάγει το μηχάνημα. Έπειτα ξεκινάει η διαδικασία επεξεργασίας του μηχανήματος. Ορίζεται η κατάσταση του σε `run` (`machine1_status= Status.green.name`) και εισάγει τα πρώτα προϊόντα προς επεξεργασία.

Η εισαγωγή των προϊόντων γίνεται με την εντολή `yield production_class.rawProd.get(machine1_input)`. Σύμφωνα με την εντολή αυτή, η κλάση του αλγορίθμου αφαιρεί το πλήθος (`#machine1_input`) των προϊόντων από το Container του προηγούμενου μηχανήματος ή της πρώτης ύλης με την εντολή `get`. Η ενέργεια αυτή ορίζει την εισαγωγή των προϊόντων στο μηχάνημα για να επεξεργαστούν. Στη συνέχεια, ανάλογα με την διεργασία του μηχανήματος, ορίζεται και το κυρίως `process` του. Για παράδειγμα, στη διεργασία του παραπάνω μηχανήματος, τα προϊόντα εισάγονται μαζικά ανά `#machine1_batch`, δηλαδή ανά **batches**. Τα επεξεργάζεται για χρόνο όσο έχει οριστεί από το υποσύστημα εισαγωγής βασικών παραμέτρων δηλαδή ίσο με `machine1_ptime`. Παράλληλα ορίζει την κατάσταση του μηχανήματος σε `run` ενώ παράγει. Κατά την ολοκλήρωση του χρόνου επεξεργασίας τα προϊόντα εξάγονται διαθέσιμα για να μεταφερθούν στο επόμενο μηχάνημα. Η διενέργεια αυτή επιτελείται από την εντολή `yield production_class.machine1Prod.put(machine1_output)` η οποία με την εντολή **put** εισάγει στο επόμενο Container του επόμενου μηχανήματος την ποσότητα `#machine1_output`. Η προσθήκη των επεξεργασμένων προϊόντων του μηχανήματος 1 στο Container του μηχανήματος 2 ορίζει και την άμεση μεταφορά των προϊόντων μέσω του υποσυστήματος μεταφορικής γραμμής. Για την ευκολία της υλοποίησης, στη διπλωματική εργασία, η μεταφορά των προϊόντων από ένα μηχάνημα σε ένα άλλο θεωρούμε πως γίνεται ακαριαία. Τέλος, η διεργασία μετά την ολοκλήρωσή της επαναλαμβάνεται συνεχώς έως ότου η κατάστασή του τεθεί διαφορετική του `run` ή τερματιστεί η προσομοίωση.

Στο εσωτερικό της διεργασίας του μηχανήματος υπάρχει η εξαίρεση **except Interrupt**. Η εξαίρεση ενεργοποιείται όταν το μηχάνημα τίθεται εκτός παραγωγικής διαδικασίας δηλαδή σε κατάσταση βλάβης (`status = stop`). Η ενεργοποίηση γίνεται από το υποσύστημα βλαβών του αλγορίθμου και η δημιουργία των βλαβών περιγράφεται στο ίδιο. Η Κατά την εκκίνηση της διακοπής, η κατάσταση του μηχανήματος ορίζεται σε βλάβη (`machine1_status = Status.red.name`), το πλήθος των σταματημάτων αυξάνεται κατά ένα και ο αλγόριθμος λαμβάνει το `flag` του μηχανήματος ως κατάσταση βλάβης. Αυτή η λειτουργία υλοποιείται με σκοπό το πρόγραμμα να μην θέσει το μηχάνημα ξανά σε κατάσταση `stop` αν είναι ήδη σε αυτή τη κατάσταση. Έπειτα, η συνάρτηση ελέγχει αν ο χρήστης του συστήματος έχει επιλέξει τη χειροκίνητη ή την αυτόματη εμφάνιση βλαβών στο σύστημα. Στη περίπτωση της αυτόματης δημιουργίας βλάβης, η συνάρτηση ελέγχει διαρκώς το `flag` του μηχανήματος μέχρι αυτό να τεθεί σε κατάσταση `run`, να βγει από τον βρόγχο αυτόματης βλάβης και να τεθεί πάλι σε κατάσταση παραγωγής. Στη περίπτωση της χειροκίνητης δημιουργίας βλάβης, ορίζεται ξανά η κατάσταση του μηχανήματος και η συνάρτηση εισέρχεται σε έναν άπειρο βρόγχο μέχρι χρήστης του συστήματος να πατήσει του κουμπι επισκευής του μηχανήματος (**REPAIR**). Έτσι, καταγράφεται ως μετρική, η διάρκεια στην οποία το μηχάνημα ήταν σε κατάσταση `stop` με σκοπό να αθροιστεί στη συνολική διάρκεια της κατάστασης και να εμφανιστεί στο γραφικό περιβάλλον από το υποσύστημα `live monitoring`.

5.4.5 Υποσύστημα live χρόνου προσομοίωσης

Για την εκτέλεση της προσομοίωσης είναι απαραίτητο να οριστεί ο χρόνος προσομοίωσης (SIM_TIME). Είναι ο πραγματικός χρόνος στον οποίο εκτελείται η παραγωγική διαδικασία. Η υλοποίηση του υποσυστήματος live χρόνου προσομοίωσης έγινε για την live αναπαράσταση κατά την εκτέλεση του αλγορίθμου. Ο χρόνος προσομοίωσης ορίζεται από το υποσύστημα εισαγωγής βασικών παραμέτρων και εκτελείται με την εντολή:

```
env.run(until=SIM_TIME)
```

που σηματοδοτεί την έναρξη της προσομοίωσης. Υπεύθυνη για την live απεικόνιση είναι η συνάρτηση update() όπου κάθε DT χρονικές μονάδες ο χρόνος προσομοίωσης αυξάνεται. Στο γραφικό περιβάλλον ο χρόνος προσομοίωσης αναπαρίσταται στο κέντρο του παραθύρου προσομοίωσης με το παρακάτω τρόπο:

Elapsed Time: 00:12:38:47

όπου δηλώνεται αντίστοιχα οι μέρες, ώρες, λεπτά και δευτερόλεπτα που διήρκεσε η προσομοίωση. Κατά τη διάρκεια τη προσομοίωσης ο χρόνος τρέχει σε ζωντανή αναπαράσταση.

5.4.6 Υποσύστημα μετρικών

Το υποσύστημα των μετρικών είναι μέρος του υποσυστήματος της προσομοίωσης. Στο γραφικό περιβάλλον αναπαρίσταται τελευταίο στο γραφικό περιβάλλον και απεικονίζει το πλήθος και την διάρκεια των τριών καταστάσεων κάθε μηχανήματος. Ο σκοπός που υλοποιήθηκε είναι να αναπαριστά live κατά την εκτέλεση της προσομοίωσης τις μετρικές των μηχανημάτων, δηλαδή τη συμπεριφορά της γραμμής παραγωγής. Το υποσύστημα μετρικών είναι μέρος της του υποσυστήματος της προσομοίωσης και δομείται στο υποσύστημα live monitoring με τον παρακάτω τρόπο:

```
def kpis(col, run_times, run_duration, standby_times,
standby_duration, stop_times, stop_duration):

    # Label (TIMES, DURATION)
    Label(window, bg=window_colour, text='TIMES', font="Arial 15
bold").grid(row=10, column=col, sticky=W, padx=50)
    Label(window, bg=window_colour, text='DURATION', font="Arial 15
bold").grid(row=10, column=col, sticky=E, padx=80)

    # Run
    Label(window, bg='#9FE849', padx=30, relief=RIDGE, text=run_times,
font="Arial 15 bold").grid(row=12, column=col, sticky=W, padx=40)
    Label(window, bg='#9FE849', padx=30, relief=RIDGE,
text=run_duration, font="Arial 15 bold").grid(row=12, column=col,
sticky=E, padx=30)

    # Stand By
```



```

Label(window, bg='#F5F372', padx=30, relief=RIDGE,
text=standby_times, font="Arial 15 bold").grid(row=13, column=col,
sticky=W, padx=40)
Label(window, bg='#F5F372', padx=30, relief=RIDGE,
text=standby_duration, font="Arial 15 bold").grid(row=13,
column=col, sticky=E, padx=30)

# Stop
Label(window, bg='#F57272', padx=30, relief=RIDGE, text=stop_times,
font="Arial 15 bold").grid(row=14, column=col, sticky=W, padx=40)
Label(window, bg='#F57272', padx=30, relief=RIDGE,
text=stop_duration, font="Arial 15 bold").grid(row=14, column=col,
sticky=E, padx=30)

```

Η συνάρτηση αυτή αποτελεί την γενικευμένη μορφή και έτσι ακριβώς ορίζεται στον αλγόριθμο. Κάθε οντότητα καλεί την συνάρτηση `kris()` με τα αντίστοιχα ορίσματά της. Όσα μηχανήματα – οντότητες υπάρχουν στη γραμμή παραγωγής, τόσες φορές θα καλείται και η συνάρτηση. Μία για κάθε μηχανήμα. Πιο συγκεκριμένα, η συνάρτηση αποτυπώνει τον τίτλο κάθε στήλης (πλήθος και διάρκεια) και χωρίζεται σύμφωνα με τις τρεις καταστάσεις των μηχανημάτων. Κάθε διαφορετική κατάσταση περιέχει δύο παιδιά, το πρώτο έχει το πλήθος που το μηχανήμα εισήλθε στη κατάσταση και το δεύτερο έχει την διάρκεια που βρέθηκε το μηχανήμα στην αντίστοιχη κατάσταση. Το πεδίο μετρικών κάθε οντότητας απεικονίζεται στο γραφικό περιβάλλον προσομοίωσης με τον παρακάτω τρόπο:

TIMES	DURATION
05	00d 7h 24m 59s
03	00d 02h 40m 37s
02	00d 00h 50m 20s

5.5 Υποσύστημα βλαβών

Για να προσεγγίζει η πραγματική γραμμή παραγωγής την αναπαράσταση της προσομοίωσης, κρίθηκε απαραίτητη η υλοποίηση του υποσυστήματος βλαβών. Σκοπός του υποσυστήματος είναι να δημιουργεί βλάβες, σταματήματα σε οποιοδήποτε μηχανήμα της παραγωγής. Το υποσύστημα χωρίζεται σε δύο κατηγορίες βλαβών, τις χειροκίνητες και τις αυτόματες. Οι χειροκίνητες βλάβες υλοποιούνται με την παρακάτω συνάρτηση:

```

def manual_break_and_repair_machine(machine_name):
# Ορισμός global μεταβλητών

if AUTO_BRAKE_VAR == 0:
    IN_BREAKDOWN = True
    BREAK_START = env.now

# Set Status
if machine_name == 'MACHINE1':
    print("Machine1 break pressed")
    production_class.machine1_gen.interrupt()

```

```

elif machine_name == 'MACHINE2':
    print("Machine2 break pressed")
    production_class.machine2_gen.interrupt()

elif machine_name == 'MACHINE3':
    print("Machine3 break pressed")
    production_class.machine3_gen.interrupt()

else:
    print("Κανένα απο τα 3 if")

```

Η **συνάρτηση χειροκίνητης δημιουργία βλάβης** έχει ως όρισμα το όνομα του μηχανήματος και καλείται ξεχωριστά για κάθε μηχανήμα στο υποσύστημα live monitoring. Αρχικά, ελέγχει αν ο χρήστης έχει επιλέξει την χειροκίνητη δημιουργία βλάβης. Αν ναι, τότε ελέγχει το μηχανήμα για το οποίο έχει κληθεί, εμφανίζει την οθόνη ότι ο χρήστης πάτησε το κουμπί BREAK του παραθύρου προσομοίωσης και τότε το μηχανήμα τίθεται εκτός λειτουργίας. Η ενέργεια αυτή επιτυγχάνεται με την εντολή `production_class.machine1_gen.interrupt()`, όπου το πρόγραμμα μόλις την διαβάσει διακόπτει την διαδικασία επεξεργασίας του μηχανήματος 1 και μεταβαίνει στην εξαίρεση που περιεγράφηκε στο υποσύστημα διεργασία μηχανήματος. Δηλαδή, με την διακοπή της διεργασία το μηχανήμα τίθεται αυτόματα σε κατάσταση stop και επιτελείται η διαδικασία που περιεγράφηκε στο υποσύστημα αυτό. Η επιλογή της χειροκίνητης δημιουργίας βλάβης στην οντότητα του γραφικού περιβάλλοντος εμφανίζεται στο παρακάτω σχήμα:



Όπου ο χρήστης επιλέγει τότε να θέσει εκτός λειτουργίας ένα μηχανήμα με το κουμπί BREAK και τότε να το επαναφέρει σε κατάσταση λειτουργίας με το κουμπί REPAIR. Για την διευκόλυνση του χρήστη και την αντίληψη του χρόνου στον οποίο θέτει ένα μηχανήμα σε κατάσταση stop, έχει υλοποιηθεί η τεχνική της καταμέτρησης χρόνου. Κατά την οποία με την πίεση του κουμπιού BREAK αρχίζει και μετράει σύμφωνα με την εκτέλεση του χρόνου προσομοίωσης και με την πίεση του REPAIR, τίθεται ξανά στην αρχική του κατάσταση, δηλαδή μηδέν. Ο τρόπος που απεικονίζεται στο παράθυρο προσομοίωσης είναι ο παρακάτω:



Η **συνάρτηση αυτόματης εμφάνισης βλαβών** υλοποιήθηκε για να προσεγγίσει την πραγματική γραμμή παραγωγής αφού οι περιπτώσεις βλαβών εμφανίζονται με τυχαίο τρόπο σε ένα μηχανήμα και για τυχαίο χρόνο. Η δομή της συνάρτησης είναι η εξής:

```

def auto_brake_machine():
    # Ορισμός global μεταβλητών

    if AUTO_BRAKE_VAR == 1:
        print("Auto breakdown mode enabled")
        while True:
            # CHECK PERIOD
            yield env.timeout(100)
            # Machine1 Brake Probability

```

```

    if np.random.binomial(1, MACHINE1_BRAKE_PROB/100) == 1 and not
MACHINE1_IS_BROKEN:
        MACHINE1_IS_BROKEN = True
        production_class.machine1_gen.interrupt()
        print("Inside def probability, MACHINE1")

    # Machine2 Brake Probability
    if np.random.binomial(1, MACHINE2_BRAKE_PROB/100) == 1 and not
MACHINE2_IS_BROKEN:
        MACHINE2_IS_BROKEN = True

        production_class.machine2_gen.interrupt()
        print("Inside def probability, MACHINE2")

    # Machine3 Brake Probability
    if np.random.binomial(1, MACHINE3_BRAKE_PROB/100) == 1 and not
MACHINE3_IS_BROKEN:
        MACHINE3_IS_BROKEN = True
        production_class.machine3_gen.interrupt()
        print("Inside def probability, MACHINE3")

```

Η συνάρτηση `auto_break_machine()` εκτελείται σαν μια διεργασία (process) στον αλγόριθμο προσομοίωσης και ορίζεται στην κλάση του. Εκτελείται συνεχώς κατά την διάρκεια της προσομοίωσης και σε κάθε εκτέλεση ελέγχει αν ο χρήστης έχει επιλέξει αυτόματη εμφάνιση βλαβών. Αν ναι, τότε ο χρήστης έχει ορίσει την πιθανότητα για κάθε μηχανήμα που επιθυμεί να εμφανιστούν οι βλάβες στο ίδιο. Στη συνέχεια η συνάρτηση εκτελεί Διωνυμική κατανομή με ελάχιστη πιθανότητα εμφάνισης ίση με 1 και μέγιστη τη διαίρεση της πιθανότητας που εισήγαγε με το 100 για να υπολογιστεί σε ποσοστό. Αν το αποτέλεσμα της κατανομής έχει τιμή ίση με τη μονάδα και το μηχανήμα για το οποίο πάει να εκτελέσει δεν βρίσκεται ήδη σε βλάβη (`MACHINE_IS_BROKEN = False`) τότε εισέρχεται στο υποπρόγραμμα δημιουργίας βλάβης. Σε αυτό, θέτει την παράμετρο του μηχανήματος `MACHINE_IS_BROKEN` σε αληθή, με σκοπό να μην ξανά τεθεί σε κατάσταση stop ενώ είναι ήδη και έπειτα εκτελεί την εντολή `production_class.machine1_gen.interrupt()`. Η εντολή `interrupt()`, όπως περιεγράφηκε και στην χειροκίνητη δημιουργία βλαβών, διακόπτει τη διαδικασία επεξεργασίας του μηχανήματος και εισέρχεται στο exception, θέτοντας το μηχανήμα σε κατάσταση stop.

5.6 Υποσύστημα καταγραφής δεδομένων

Το υποσύστημα καταγραφής δεδομένων υλοποιήθηκε για την συλλογή, καταγραφή και αποθήκευση των δεδομένων που παράγει ο αλγόριθμος προσομοίωσης κατά τη διάρκεια της εκτέλεσής του. Η καταγραφή των δεδομένων γίνεται παράλληλα σε δυο επιμέρους συστήματα. Το **πρώτο σύστημα** αφορά την καταγραφή των δεδομένων σε global παραμέτρους των μηχανημάτων, ξεχωριστές για το καθένα. Οι παράμετροι αυτές αρχικοποιούνται κατά την έναρξη του αλγορίθμου προσομοίωσης και έχουν την παρακάτω δομή:

```

# --- RCA VARIABLES ---
# Machine KPIs
MACHINE_IS_BROKEN = False

```

```
# Run
MACHINE_RUN_TIMES = 1
MACHINE_RUN_DURATION = []

# Stand By
MACHINE_STANDBY_FLAG = True
MACHINE_STANDBY_TIMES = 0
MACHINE_STANDBY_DURATION = {}

# Stop
MACHINE_STOP_FLAG = True
MACHINE_STOP_TIMES = 0
MACHINE_STOP_DURATION = {}
```

Οι μεταβλητές καταχώρησης δεδομένων αποτελούν το σημαντικότερο στοιχείο του αλγορίθμου για την απεικόνιση των μετρικών και των αποτελεσμάτων της προσομοίωσης. Συγκεκριμένα:

- **MACHINE_IS_BROKEN:** δηλώνει αν το μηχάνημα βρίσκεται σε κατάσταση βλάβης ή όχι. Ο ρόλο της μεταβλητής είναι να μη τεθεί το μηχάνημα ξανά σε κατάσταση stop ενώ βρίσκεται ήδη.
- **MACHINE_\$_TIMES:** είναι το πλήθος στο οποίο το μηχάνημα βρέθηκε σε κάθε κατάσταση. Αντίστοιχα \$ = run | standby | stop.
- **MACHINE_\$_DURATION:** είναι η χρονική διάρκεια που το μηχάνημα τέθηκε στην εκάστοτε κατάσταση. Ανάλογα την λειτουργία που επιτελεί κάθε μηχάνημα ορίζεται είτε σε python list είτε σε python dictionary. Αντίστοιχα \$ = run | standby | stop.
- **MACHINE_\$_FLAG:** ορίζει την Boolean τιμή που έχει το flag του μηχανήματος και σκοπός του είναι να αποτρέψει τον αλγόριθμο να εισέλθει στην ίδια κατάσταση που βρίσκεται τη δεδομένη στιγμή ένα μηχάνημα.

Το **δεύτερο σύστημα** αφορά την συνάρτηση παρακολούθησης των καταστάσεων κάθε μηχανήματος της παραγωγής. Η δομή το είναι:

```
def status_monitoring()
# Ορισμός global μεταβλητών
# Παρακολούθηση μηχανών παραγωγής

while True:
yield env.timeout(DT) # Κάθε χρόνο dt, εισάγει μια εγγραφή
df = df.append({'ENV_NOW': env.now, 'Machine1':
machine1_description, 'Machine2': machine2_description, 'Machine3':
machine3_description}, ignore_index=True)

write_row({'Env_now': env.now, 'Machine': 'machine_name',
'Description': machine_description, 'Current_cans':
production_class.machineProd.level, 'Status': machine_status})
```

Η συνάρτηση status+monitoring() εκτελείται κάθε DT χρονικές στιγμές και αρχικοποιείται στη κλάση του συστήματος προσομοίωσης ως simpry.process. Στο pandas.dataframe καταχωρούνται οι καταστάσεις όλων των μηχανημάτων της παραγωγής. Επίσης, χρησιμοποιείται το αρχείο καταγραφής δεδομένων output.csv, στο οποίο

καταγράφονται ο χρόνος, το όνομα του μηχανήματος, η περιγραφή που δίνεται, το πλήθος των προϊόντων που επεξεργάζεται την δεδομένη στιγμή και τέλος η κατάστασή του.

Η συνάρτηση `write_row()` εισάγει τις παραπάνω τιμές στο αρχείο, υλοποιείται στο `Main/OutputCSV.py` και έχει την εξής δομή:

```
def write_row(row):
    df = pd.DataFrame(data=row, index=[None], columns=headerList)
    df.to_csv("output/output.csv", sep=',', mode='a', encoding='utf-8',
             header=False, index=False)
```

Στο ίδιο πρόγραμμα υπάρχει επίσης και η συνάρτηση ελέγχου του αρχείου. Ο ρόλος της είναι να αδειάζει τα δεδομένα που πιθανά θα υπάρχουν στο αρχείο `output.csv`, πριν την εκτέλεση του αλγορίθμου προσομοίωσης με σκοπό να γράψει τα νέα. Η συνάρτηση υλοποιείται στο `Main/OutputCSV.py`

```
def check_csv()
```

5.7 Υποσύστημα επεξεργασίας δεδομένων

Το υποσύστημα επεξεργασίας δεδομένων βρίσκεται στο αρχείο `Main/Converters.py` και απαρτίζεται από τις εξής συναρτήσεις:

```
def duration_converter_to_DHMS(total_ptime):
    if type(total_ptime) is dict:
        total_proc_time = sum(total_ptime.values())
    elif type(total_ptime) is list:
        total_proc_time = sum(total_ptime)
    else:
        total_proc_time = total_ptime

    days = int(total_proc_time / 86400)
    hours = int((total_proc_time - (days * 86400)) // 3600)
    minutes = int(total_proc_time % 3600 / 60)
    seconds = int((total_proc_time % 3600) % 60)
    return '{:02d}d {:02d}h {:02d}m {:02d}s'.format(days, hours,
    minutes, seconds)
```

Η συνάρτηση **`duration_converter_to_DHMS()`** εισάγει σαν όρισμα λεξικό, λίστα ή πραγματικό αριθμό. Οι τύποι αυτοί διαθέτουν δεδομένα πραγματικών αριθμών σε δευτερόλεπτα και είναι οι χρόνοι του κάθε μηχανήματος στις τρεις καταστάσεις. Σκοπός της συνάρτησης είναι να εξάγει τα αποτελέσματα με την μορφή `00d 00h 00m 00s` για να είναι αναγνώσιμα στο γραφικό περιβάλλον. Με αυτό τον τρόπο ο χρήστης έχει μια πιο ευανάγνωστη μορφή των μετρικών της προσομοίωσης.

Αντίστοιχη υλοποίηση και λογική έχει η συνάρτηση **`duration_converter_to_M()`** με την μόνη διαφορά ότι τα αποτελέσματα που επιστρέφει είναι μόνο με την μορφή λεπτών (`00m`) και χρησιμοποιείται για μεταβλητές που χρειάζονται απεικόνιση λεπτών στο γραφικό περιβάλλον.

```
def duration_converter_to_M(total_ptime):
    if type(total_ptime) is dict:
```

```

    total_proc_time = sum(total_ptime.values())
elif type(total_ptime) is list:
    total_proc_time = sum(total_ptime)
else:
    total_proc_time = total_ptime

minutes = int(total_proc_time // 60)

return '{:02d} min'.format(minutes)

```

Επιπλέον, χρησιμοποιείται η συνάρτηση **sum_converter()** η οποία μετατρέπει τα δεδομένα του χρόνου σε πραγματικούς αριθμούς με αναπαράσταση δύο δεκαδικών ψηφίων. Η δομή της είναι:

```

def sum_converter(total_ptime):
    if type(total_ptime) is dict:
        total_proc_time = sum(total_ptime.values())
    elif type(total_ptime) is list:
        total_proc_time = sum(total_ptime)
    else:
        total_proc_time = total_ptime

    return round(total_proc_time, 2)

```

Τέλος, γίνεται χρήση της συνάρτησης **machine_duration_conv_to_perc()** η οποία εισάγει ένα πραγματικό αριθμό, λεξικό ή λίστα που αποτελούν τη χρονική διάρκεια των μηχανημάτων σε κάθε κατάσταση. Έχει επιπλέον, ως όρισμα τον συνολικό χρόνο προσομοίωσης. σε συνδυασμό αυτών των δυο ορισμάτων η συνάρτηση τα μετατρέπει σε ποσοστό. Η δομή της είναι η εξής:

```

def machine_duration_conv_to_perc(DURATION, SIM_TIME):
    print(type(DURATION))
    if type(DURATION) is dict:
        # if dict is empty
        if not DURATION:
            total_sum = 0
        elif isinstance(DURATION.values(), float):
            total_sum = DURATION.values()
        else:
            total_sum = sum(DURATION.values())

    else:
        # if list is empty
        if not DURATION:
            total_sum = 0
        # if list has one number
        elif isinstance(DURATION, float):
            total_sum = DURATION
        else:
            total_sum = sum(DURATION)

    return round(total_sum*100/SIM_TIME, 2)

```

5.8 Υποσύστημα live monitoring

Το υποσύστημα live monitoring είναι υπεύθυνο για την γραφική απεικόνιση της προσομοίωσης όπως αυτή περιγράφεται στο κεφάλαιο 4.1.3 «Αρχιτεκτονική UI προσομοίωσης». Αποτελείται από μια κεντρική συνάρτηση η οποία εκτελείται ως process στην κλάση του συστήματος και ορίζεται ως εξής:

```
# Status & Variable Function Monitoring
self.status_monitoring_gen = env.process(status_monitoring())
```

```
def live_monitoring()
```

Στο εσωτερικό της περιέχει το υποσύστημα του φάρου και τις συναρτήσεις που το διέπουν. Περιέχει επίσης, το υποσύστημα του live χρόνο προσομοίωσης και των μετρικών. Η δομή της είναι εκτεταμένη ωστόσο θα αναλυθούν οι συναρτήσεις που εμπεριέχονται στο εσωτερικό της. Κάθε συνάρτηση της live_monitoring() έχει αναλυθεί στις επιμέρους συναρτήσεις και ορίζονται ως:

- **machine_beacon:** Είναι η γραφική απεικόνιση των τριών καταστάσεων του υποσυστήματος του φάρου.
- **current_conveyor_capacity_buffer:** Αποτυπώνει τα προϊόντα της μεταφορικής γραμμής που μεταφέρει κατά τη διάρκεια της προσομοίωσης καθώς και την χωρητικότητά της. Η συνάρτηση περιγράφεται στο υποσύστημα μεταφορικής γραμμής.
- **current_machine_buffer:** Είναι ο buffer του μηχανήματος και εμφανίζει το πλήθος που επεξεργάζεται το μηχάνημα κατά τη διάρκεια της προσομοίωσης σε σύγκριση με το μέγιστο μέγεθος των προϊόντων που μπορεί να παράξει ταυτόχρονα το μηχάνημα, δηλαδή τη χωρητικότητά του. Η συνάρτηση περιγράφεται στο υποσύστημα διεργασίας μηχανήματος.
- **manual_repair_fun:** Είναι η συνάρτηση που ελέγχει τα flags όλων των μηχανημάτων της παραγωγής και τα αποδεσμεύει μόλις ο χρήστης πιέσει το κουμπί REPAIR.
- **manual_brake_button:** Αποτυπώνει τη γραφική αναπαράσταση του κουμπιού BREAK για κάθε μηχάνημα.
- **kpis:** Είναι η συνάρτηση υπεύθυνη για τις μετρικές του συστήματος και ορίζεται από το υποσύστημα των μετρικών.
- **beacon_status:** Η συνάρτηση αυτή ορίζει την αρχική γραφική απεικόνιση των τριών καταστάσεων του φάρου. Είναι υπεύθυνη για την αλλαγή των χρωμάτων των καταστάσεων καθώς το μηχάνημα αλλάζει status.

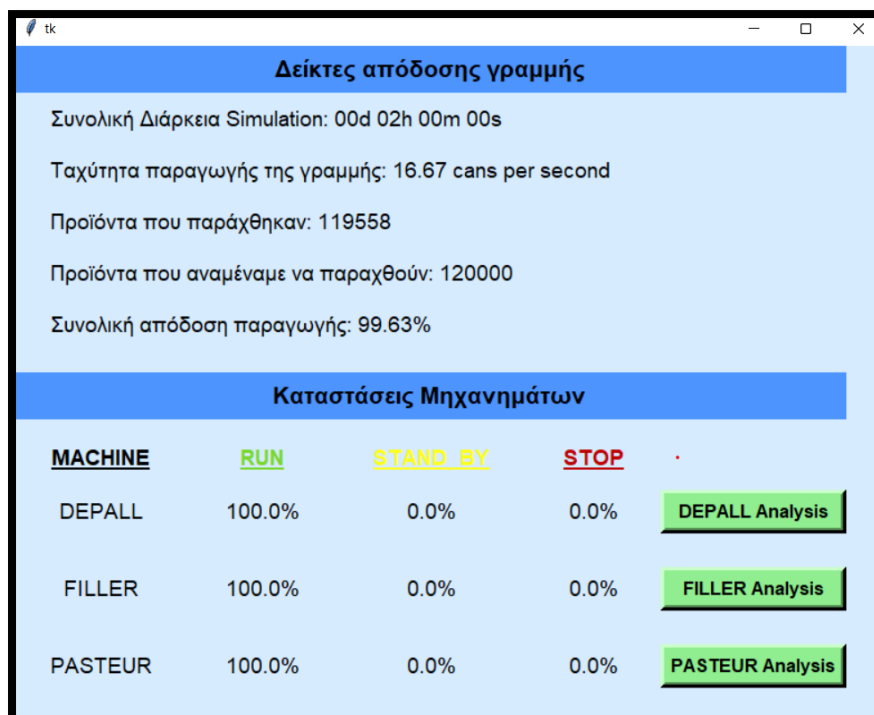
- **sim_time_converter:** Είναι ένας απλός μετατροπέας, για την αναπαράσταση των δεδομένων του χρόνου προσομοίωσης κατά την εκτέλεση της προσομοίωσης. Η λειτουργία της και η δομή ορίζεται στο υποσύστημα live χρόνου προσομοίωσης.
- **colour_palette_calc:** Η συνάρτηση αυτή αποτελεί την εναλλαγή των χρωματικών προτύπων στο buffer των μηχανημάτων και των μεταφορικών. Υλοποιήθηκε βασισμένη σε μια παλέτα χρωμάτων με κλίμακα από 1 έως το 10. Η μικρότερη τιμή ορίζει το πιο ανοιχτό χρώμα της παλέτας και σηματοδοτεί ότι ο buffer είναι σχεδόν άδειος. Η μεγαλύτερη τιμή (10) ορίζει το πιο σκούρο χρώμα της παλέτας και σηματοδοτεί ότι ο buffer είναι γεμάτος. Υλοποιήθηκε για να διευκολύνει τη διεπαφή του χρήστη κατά την εκτέλεση της προσομοίωσης να αντιλαμβάνεται πόσο γεμάτο είναι κάθε μηχανήμα και μεταφορική γραμμή.
- **update:** Η συνάρτηση update είναι υπεύθυνη να τρέχουν όλα τα υποσυστήματα σε ζωντανή απεικόνιση. Κατά την διάρκεια της προσομοίωσης η update ανανεώνει το γραφικό της περιβάλλον (κύριο περιβάλλον προσομοίωσης) ανά DT χρονικές στιγμές. Έτσι, ο χρήστης έχει live αναπαράσταση της γραμμής παραγωγής που προσομοιώνεται.

5.9 Υποσύστημα επισκόπησης προσομοίωσης

Το υποσύστημα επισκόπησης προσομοίωσης ενεργοποιείται αμέσως μετά την λήξη του προγράμματος προσομοίωσης. Υλοποιείται με την μέθοδο δημιουργίας νέου παραθύρου ξεχωριστό από το παράθυρο της προσομοίωσης και αποτυπώνει την τελική σύνοψη της γραμμής παραγωγής. Ο χρήστης μέσα από αυτό το γραφικό περιβάλλον έχει τη δυνατότητα να κατανοήσει τη συμπεριφορά της γραμμής παραγωγής και να δει τις μετρικές όλων των μηχανημάτων. Η συνάρτηση που υλοποιείται είναι:

```
def outroScreen()
```

Μέσω της συνάρτησης παρέχεται η συνολική εικόνα της παραγωγικής διαδικασίας. Στη παρακάτω φωτογραφία απεικονίζεται η σύνοψη τριών μηχανημάτων έπειτα από την εκτέλεση της προσομοίωσης.



Εικόνα 5.3: Σύνοψη γραμμής παραγωγής

Το υποσύστημα επισκόπησης της προσομοίωσης παρέχει στον χρήστη του βασικούς δείκτες απόδοσης της γραμμής. Πιο συγκεκριμένα, ο χρήστης βλέπει την συνολική διάρκεια της προσομοίωσης και την ταχύτητα παραγωγής των μηχανημάτων. Επίσης, απεικονίζεται το πλήθος των προϊόντων που παράχθηκαν με την ολοκλήρωση της προσομοίωσης σε συνδυασμό με τα αναμενόμενα προϊόντα που θα παρήγαγε μια ιδανική γραμμή παραγωγής. Τέλος, εκτιμάται η συνολική απόδοση της παραγωγής **OEE** (Overall Equipment Effectiveness).

Στο ίδιο παράθυρο δίνονται τα ποσοστά των καταστάσεων που τέθηκε κάθε μηχανήμα. Δηλαδή, για κάθε μια από τις τρεις καταστάσεις (run, stand by, stop) παρέχεται το ποσοστό που το μηχάνημα παρέμεινε σε αυτή την κατάσταση συνολικά κατά την διάρκεια της παραγωγικής διαδικασίας.

Τέλος, ο χρήστης έχει την δυνατότητα να επιλέξει μεμονωμένα την ανάλυση κάποιου μηχανήματος πατώντας το αντίστοιχο κουμπί «Machine analysis» που βρίσκεται δεξιά από τα ποσοστά των καταστάσεων. Με αυτό τον τρόπο, ο χρήστης οδηγείται σε ένα άλλο διαφορετικό γραφικό περιβάλλον – παράθυρο για το οποίο είναι υπεύθυνο το υποσύστημα ανάλυσης απόδοσης του μηχανήματος.

5.10 Υποσύστημα ανάλυσης απόδοσης μηχανήματος

Το υποσύστημα ανάλυσης απόδοσης υλοποιήθηκε για να δίνει λεπτομερή στατιστικά δεδομένα στον χρήστη. Σε αυτό το υποσύστημα εφαρμόζεται η τεχνική του RCA – Root Cause Analysis. Η συνάρτηση που υλοποιεί το υποσύστημα είναι η:

```
def additionalAnalysisScreen(machine_name, STANDBY_PERCENT_, STOP_PERCENT_)
```

Αποτελεί την αρχικοποίηση του γραφικού παραθύρου ανάλυσης κάθε μηχανήματος. Έχει ως ορίσματα: τη ονομασία του μηχανήματος για το οποίο ο χρήστης επέλεξε να γίνει η

ανάλυση, το ποσοστό που βρέθηκε στη κατάσταση stand by καθώς και το ποσοστό της κατάστασης stop. Στην αρχή του παραθύρου απεικονίζονται οι γενικές πληροφορίες του μηχανήματος. Αφορούν το ποσοστό και το χρόνο που το μηχάνημα ήταν συνολικά σε παραγωγή δηλαδή σε κατάσταση run, και το MTBS (**MTBF**) δηλαδή τον συνεχόμενο παραγωγικό χρόνο μέχρι να τεθεί σε stop. Περιέχει τις παρακάτω συναρτήσεις που καλούνται για κάθε οντότητα μηχανήματος ξεχωριστά:

- **rca_general_info:** Είναι οι γενικές πληροφορίες του μηχανήματος για την ανάλυση του μη παραγωγικού χρόνου. Δηλαδή του χρόνου που τέθηκε σε stand by, κατάσταση αναμονής. Αρχικά, δίνεται ο συνολικός μη παραγωγικός χρόνος, το πλήθος των φορών που το μηχάνημα ήταν σε αναμονή και τέλος ο συνολικός χρόνος που διήρκεσαν τα stand by's.
- **RCA_Table:** Είναι η συνάρτηση που αποτυπώνει τον πίνακα του Root Cause Analysis του μηχανήματος. Ο πίνακας διαθέτει όλα τα μηχανήματα της παραγωγής στη στήλη μηχανήματα καθώς και τις στήλες: Πλήθος, Διάρκεια, Ποσοστό. Για το μηχάνημα το οποίο γίνεται η ανάλυση, η γραμμή των δεδομένων του αφορά μόνο τη κατάσταση που τέθηκε σε stop. Για όλα τα υπόλοιπα, οι γραμμές δεδομένων τους αφορούν την κατάσταση αναμονής. Στον παρακάτω πίνακα δίνεται η κατανόηση του υπολογισμού των δεδομένων του πίνακα RCA.

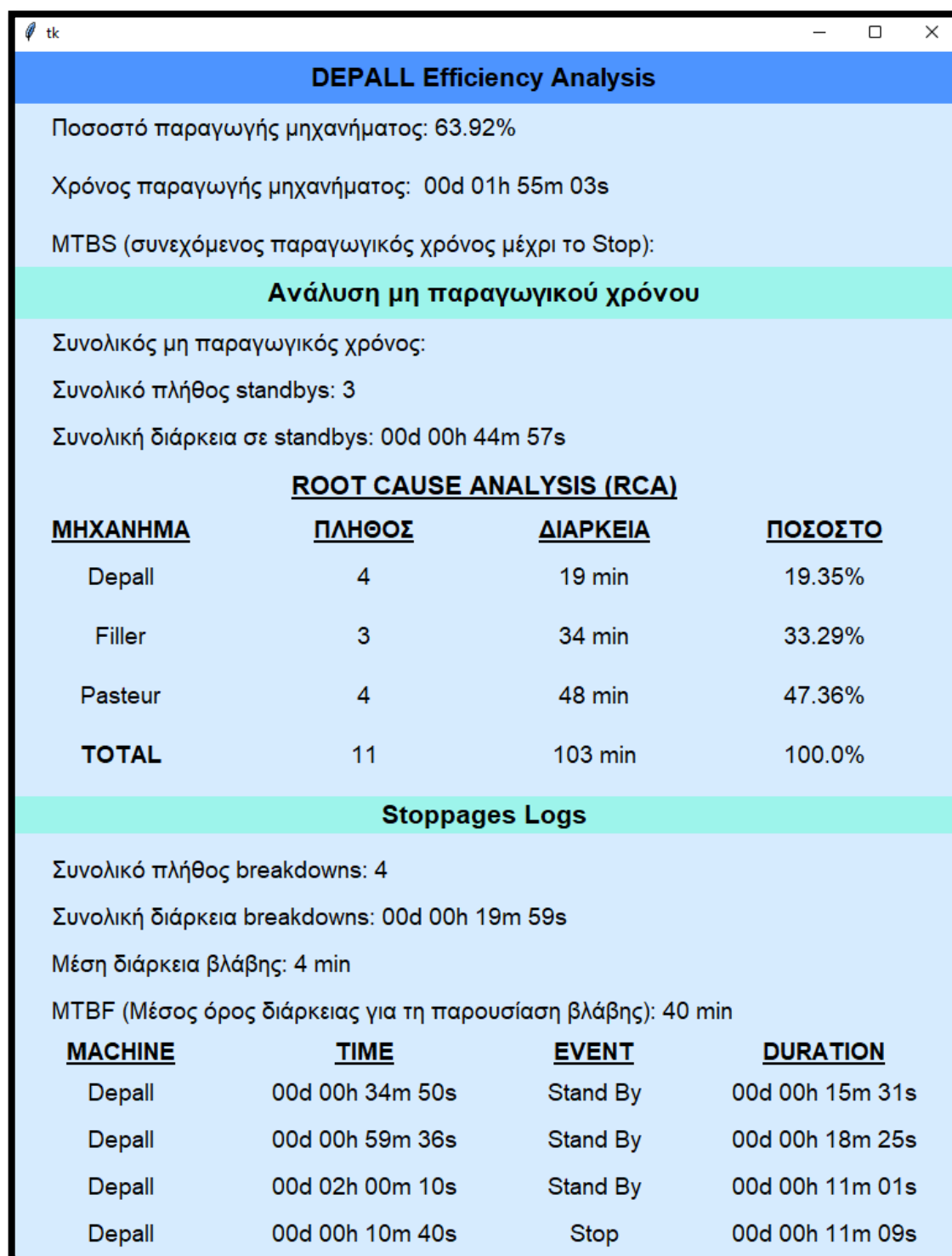
ΜΗΧΑΝΗΜΑΤΑ	ΠΛΗΘΟΣ	ΔΙΑΡΚΕΙΑ	ΠΟΣΟΣΤΟ
Μηχάνημα 1	A	B	C
Μηχάνημα 2	D	E	F
Μηχάνημα 3	G	H	I
Σύνολο	J	K	L

$$J = A + D + G, \quad K = B + E + H, \quad L = C + F + I$$

$$C = \frac{B}{K}, \quad F = \frac{E}{K}, \quad I = \frac{H}{K}$$

- **stoppages_general_info:** Η συνάρτηση αυτή επιστρέφει τα στατιστικά δεδομένα που αφορούν τα σταματήματα του μηχανήματος, την κατάσταση stop δηλαδή. Ορίζονται: το συνολικό πλήθος των breakdowns (βλαβών του μηχανήματος), η συνολική διάρκεια που το μηχάνημα ήταν σε βλάβη, η μέση διάρκεια που το μηχάνημα παρέμενε σε κατάσταση stop. Και τέλος, το **MTBF**, ο μέσος όρος διάρκειας για παρουσίαση βλάβης. Δηλαδή, ο χρόνος που το μηχάνημα παράγει μέχρι να ξανά τεθεί σε stop.
- **standbys_print:** Επιστρέφεται μια λίστα με κάθε stand by του μηχανήματος. Για κάθε εγγραφή, δίνεται η χρονική στιγμή του προσομοιωτή που πραγματοποιήθηκε το event, ο τύπος του event (stand by ή stop) και ο χρόνος που διήρκεσε.
- **breakdowns_print:** Επιτελεί την ίδια λειτουργία με την συνάρτηση standby_print, αλλά για τα σταματήματα του μηχανήματος.

Στη συνέχεια, δίνεται ένα σχήμα εκτέλεσης μηχανήματος σε παράδειγμα για την ανάλυση του υποσυστήματος:



Εικόνα 5.4: Ανάλυση μηχανήματος

Ολόκληρος ο κώδικας της εφαρμογής μπορεί να βρεθεί στο **Github** κάνοντας κλικ [εδώ](#).

6. Μελέτη περίπτωσης

Ως μελέτη περίπτωσης χρησιμοποιήθηκε το εργοστάσιο παραγωγής ζύθου – Αθηναϊκή Ζυθοποιία. Η γραμμή που προσομοιώθηκε είναι η συσκευασία κουτιού μπίρας.

Ιδιαίτερα χρήσιμη για την μελέτη περίπτωσης αποτέλεσε η έρευνα για τη γραμμή παραγωγής μπίρας στο εργοστάσιο της Heineken: **“A practical model of Heineken’s bottle filling line with dependent failures”** [31]. Οι γραμμές παραγωγής ή οι μεταφορικές γραμμές ερευνώνται για τουλάχιστον 40 χρόνια. Τα μοντέλα θεωρίας των συστημάτων επισκευής έχουν ερευνηθεί από τις πηγές: [32] [33] [34]. Στις πηγές, [34] [35] παρέχεται η πληροφορία ότι ο συνολικός χρόνος που ένα μηχάνημα είναι εκτός λειτουργίας για διεργασία εξαρτώμενης βλάβης και επισκευής είναι συμπτωτικά κανονικός.

6.1 Η γραμμή παραγωγής

Ένα αριθμός μηχανημάτων λειτουργούν σε σειρά με σκοπό να παράγουν γεμάτα κουτιά μπίρας. Άδεια κουτιά μπίρας φτάνουν σε παλέτες των 50.000 κουτιών. Ο depalletizer παίρνει τα κουτιά από την παλέτα και τα τοποθετεί στη μεταφορική γραμμή. Ο filler πλένει, γεμίζει τα κουτιά με μπίρα και τα σφραγίζει. Στη συνέχεια, ο παστεριωτής (Pasteur) παστεριώνει τα κουτιά για 45 λεπτά. Ο packer τοποθετεί τα κουτιά σε συσκευασία και ο palletizer τοποθετεί τις συσκευασίες σε παλέτες. Μεταξύ κάθε μηχανήματος υπάρχουν μεταφορικές γραμμές. Οι μεταφορικές έχουν διπλή λειτουργία, τη λειτουργία της μεταφοράς και του buffer. Η πρώτη λειτουργία μεταφέρει τα κουτιά από το ένα μηχάνημα στο άλλο και ο buffer αποθηκεύει τα κουτιά μπίρας με σκοπό ένα μηχάνημα να συνεχίσει να παράγει αν το προηγούμενο ή το επόμενο από αυτό μηχάνημα είναι προσωρινά σε κατάσταση βλάβης. Κάθε μηχάνημα έχει μια ονομαστική ταχύτητα παραγωγής, αλλά συνήθως παράγει με χαμηλότερο ρυθμό καθώς εξαρτάται από το πλήθος των breakdowns. Το core μηχάνημα (bottleneck) είναι αυτό που παράγει με την μικρότερη ταχύτητα από όλα τα μηχανήματα της γραμμής. Σε αυτή την περίπτωση είναι ο filler με 667 κουτιά το λεπτό. Το core μηχάνημα είναι τυπικά το πιο ακριβό και δύσκολο να αναβαθμιστεί. Η χωρητικότητα και το εύρος των buffer αναλύεται στον παρακάτω πίνακα.

Μηχάνημα	Εύρος κουτιών	Χωρητικότητα (κουτιά)
Depalletizer	900	50000
Filler	667	4400
Pasteur	667	3900
Labeller	800	6700
Packer	912	9600
Palletizer	900	inf

Πίνακας 6.1

Όλα τα μηχανήματα μοντελοποιούνται να παράγουν στην προκαθορισμένη ταχύτητα. Αν τα μηχανήματα δεν παρουσίαζαν προβλήματα, τότε η γραμμή θα παρήγαγε με τη ταχύτητα του core μηχανήματος και το σύστημα του buffer δεν θα χρησιμοποιούνταν. Η αποδοτικότητα των μηχανημάτων εξαρτάται από την διαθεσιμότητά τους κατά τη διάρκεια της παραγωγής. Όπου η διαθεσιμότητα είναι η πιθανότητα τα μηχανήματα να μην είναι σε κατάσταση βλάβης (stop). Όταν ένα μηχάνημα τίθεται σε κατάσταση βλάβης, το επόμενο μηχάνημα θα συνεχίσει να λειτουργεί έως ότου να εξαντληθεί το απόθεμα του buffer του. Όταν ο buffer αδειάσει τότε και αυτό το μηχάνημα με τη σειρά του θα σταματήσει και θα

τεθεί σε κατάσταση αναμονής. Όταν ένα μηχάνημα σταματήσει καλείται breakdown. Τα μηχανήματα πριν από το breakdown μηχανήματα θα συνεχίσουν να λειτουργούν μέχρι ο buffer τους να γεμίσει.

Στο πλαίσιο της διπλωματικής εργασίας, η μελέτη περίπτωσης που υλοποιήθηκε αφορά ένα μέρος της γραμμής παραγωγής κουτιού και πιο συγκεκριμένα τα μηχανήματα Depall, Filler και Pasteur. Τα δεδομένα των μηχανημάτων και των μεταφορικών γραμμών του αναλύονται στον πίνακα 6.2. Ο πίνακας περιέχει συγκεντρωτικά τις τιμές των παραμέτρων που αφορούν την υλοποίηση του υποσυστήματος καταχώρησης δεδομένων. Η ταχύτητα παραγωγής της γραμμής έχει επιλεγεί να είναι 60.000 κουτιά την ώρα. Άρα 16.6 κουτιά το δευτερόλεπτο.

	Palet Cans Conveyor	Depall	Depall Cans Conveyor	Filler	Filled Cans Conveyor	Pasteur
Capacity (cans)	50000	5000	4000	4000	200	3000
Initial Cans	Inf	0	0	0	0	0
Batch (cans)	5000	5000	1	1	1	1..600
Input	5000	5000	1	1	1	1..600
Output	1	1	1	1	1	1..600
Speed	16,66	16,66	16,66	16,66	16,66	16,67
Process Time (seconds)	0,06	0,06	0	0,06	0	1800
Critical Buffer (cans)	5000	500	0	1	0	1..600

Πίνακας 6.2: Δεδομένα γραμμής

6.2 Προσομοίωση γραμμής

Στην ενότητα αυτή, μελετάται η περίπτωση της προσομοίωσης της γραμμής κουτιού. Αφού έχουν περιγράψει η λειτουργία των μηχανημάτων της παραγωγής και έχουν καταχωρηθεί τα δεδομένα κάθε μηχανήματος που χρειάζεται ο αλγόριθμος για την έναρξη της προσομοίωσης, αρχίζει με την εκκίνηση του υποσυστήματος εισαγωγής βασικών παραμέτρων.

Ο χρήστης επιλέγει να δημιουργήσει μια προσομοίωση συσκευασία μπύρας σε κουτί. Εισάγει αρχικά στο βήμα της προσομοίωσης (DT) την τιμή 10. Κάθε 10 δευτερόλεπτα θα ανανεώνεται το γραφικό περιβάλλον της προσομοίωσης. Στη συνέχεια επιλέγει η παραγωγή να διαρκέσει 6 ώρες όπως φαίνεται στην εικόνα 6.1.

tk

Προσομοίωση Γραμμής Παραγωγής

Επιλογές Προσομοίωσης

Βήμα προσομοίωσης (dt):

10

sec

Συνολικό διάστημα προσομοίωσης (T):

6

hours

Έναρξη Προσομοίωσης

Εμφάνιση Βλαβών

Κατά τη διάρκεια της προσομοίωσης ο χρήστης δημιουργεί βλάβες στα μηχανήματα και τις επιδιορθώνει όποτε επιθυμεί.
Για αυτόματη δημιουργία βλαβών από τον simulator επιλέξτε το παρακάτω checkbox και συμπληρώστε τις πιθανότητες εμφάνισης ανα μηχανήμα.

☐ Αυτόματη Εμφάνιση Βλαβών

Πιθανότητα βλάβης:

Depall

0

%

Filler

0

%

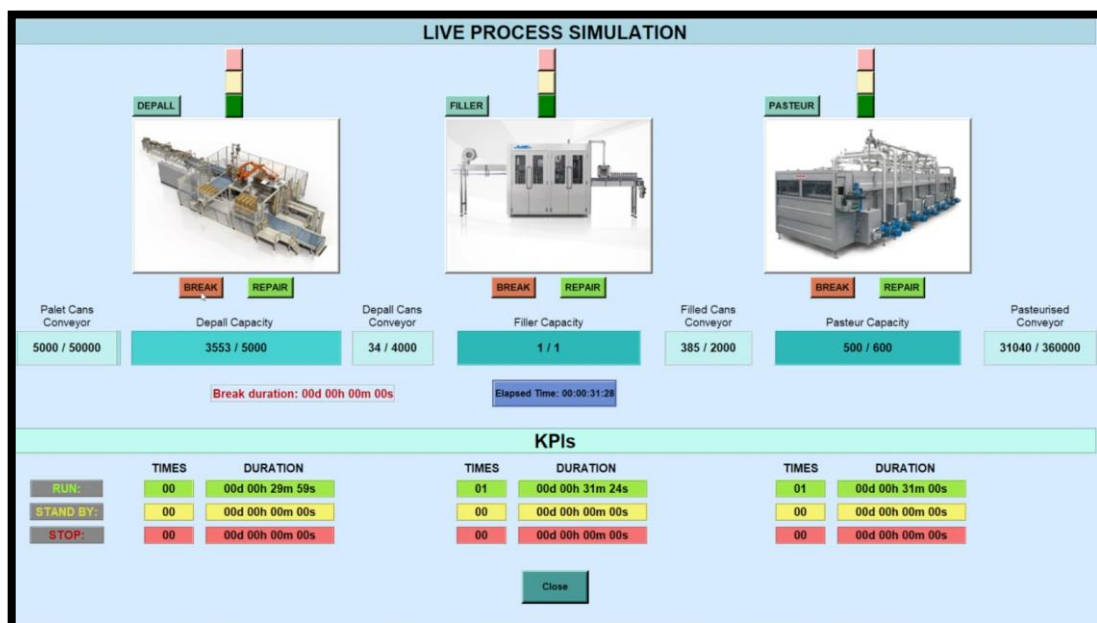
Pasteur

0

%

Εικόνα 6.1 Υποσύστημα εισαγωγής βασικών παραμέτρων

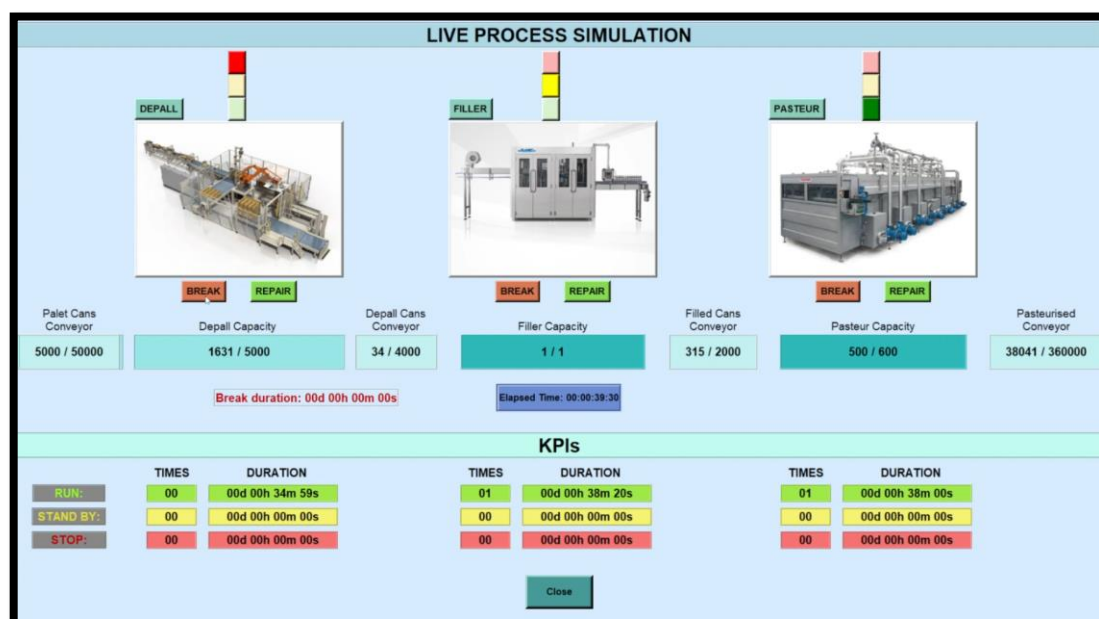
Χωρίς να επιλέξει το checkbox «αυτόματη εμφάνιση βλαβών», ενεργοποιείται η χειροκίνητη δημιουργία βλάβης από το υποσύστημα βλαβών. Και έπειτα εκκινεί την προσομοίωση πιέζοντας το κουμπί «Έναρξη προσομοίωσης». Κατόπιν της εντολής, καλείται το υποσύστημα προσομοίωσης και το υποσύστημα live monitoring τα οποία είναι υπεύθυνα για την γραφική αναπαράσταση της γραμμής παραγωγής.



Εικόνα 6.2: Υποσύστημα προσομοίωσης

Η εικόνα 6.2 είναι το γραφικό περιβάλλον της προσομοίωσης των τριών μηχανημάτων **Depall – Filler – Pasteur**. Η προσομοίωση έχει εκτελέσει 31 λεπτά και 28 δευτερόλεπτα παραγωγικής διαδικασίας. Μέχρι την στιγμή αυτή ο χρήστης δεν έχει επιλέξει να

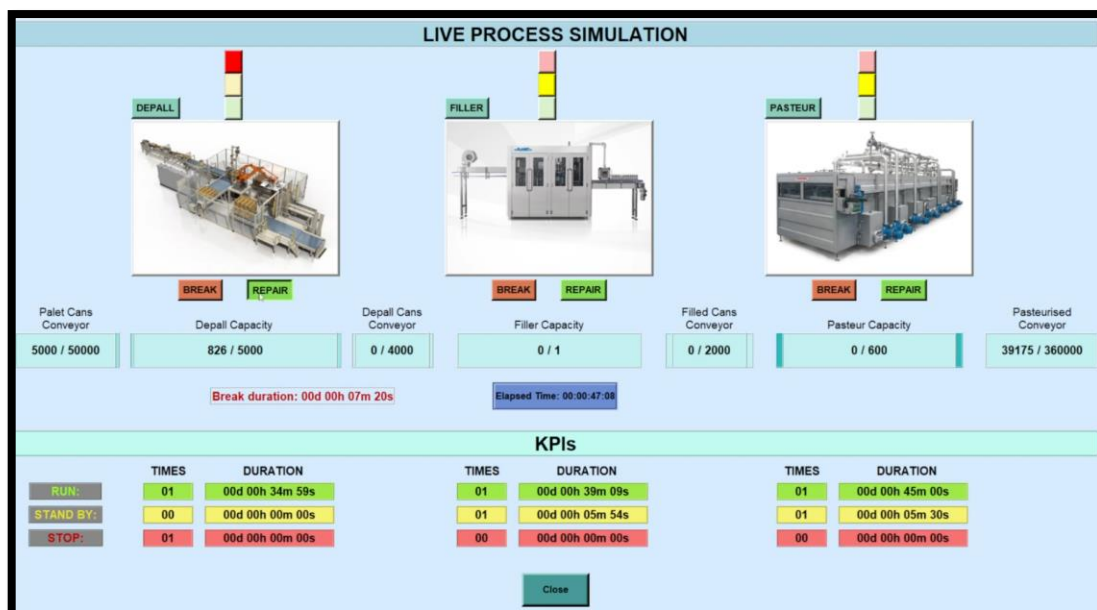
δημιουργηθεί βλάβη σε κάποιο μηχάνημα. Στο πλαίσιο KPIs αποτυπώνονται οι τρεις καταστάσεις των μηχανημάτων, το πλήθος και ο χρόνος για την κάθε μια.



Εικόνα 6.3: Χρονική Στιγμή 39:30

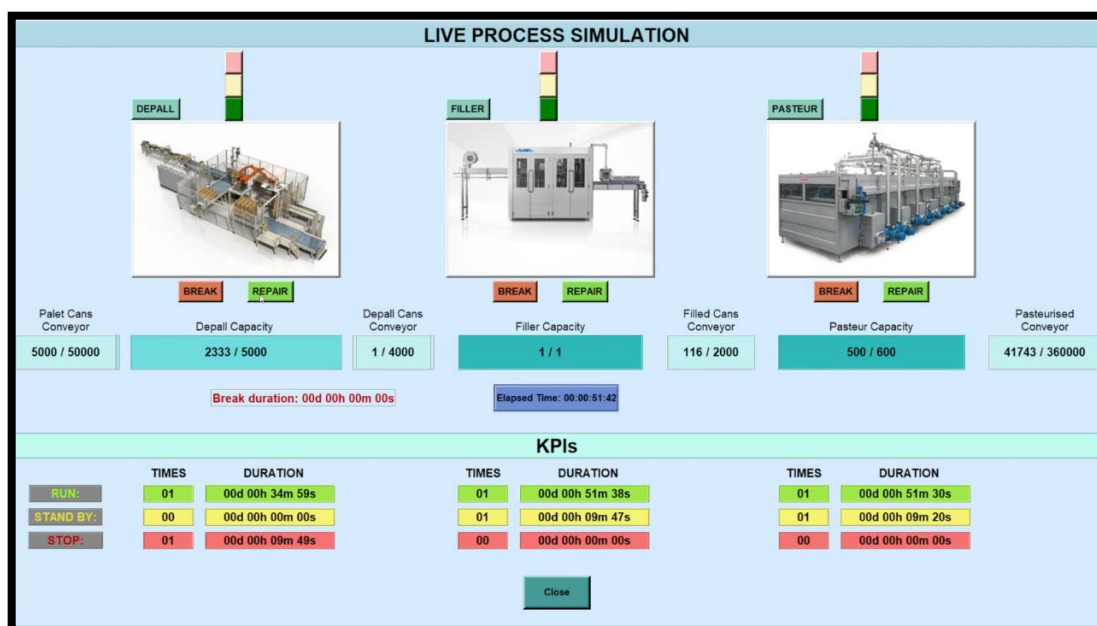
Την χρονική στιγμή 39:30, ο χρήστης επιλέγει να δημιουργήσει την πρώτη βλάβη στο μηχάνημα Depall. Ο χρόνος της παραγωγής σταματάει και τίθεται αυτόματα στην κατάσταση stop. Η κατάστασή φαίνεται από τον φάρο του μηχανήματος, στον οποίο έχει ανάψει η κόκκινη ένδειξη. Παράλληλα ξεκινάει η μέτρηση του βοηθητικού ρολογιού “Breakdown duration” το οποίο βοηθά τον χρήστη να επιλέξει τον χρόνο breakdown κάθε μηχανήματος που θέλει να προκαλέσει. Η προσομοίωση εκτελείται με μεγάλη ακρίβεια ($DT = 10$), όμως επειδή ο χρόνος εκτέλεση της παραγωγικής διαδικασίας εκτελείται πολύ γρήγορα σε σχέση με τον πραγματικό χρόνο, αμέσως ο Filler τίθεται σε κατάσταση stand by. Ο Pasteur συνεχίζει να λειτουργεί αφού όπως φαίνεται στην εικόνα η μεταφορική γραμμή (Buffer) του Filler έχει ακόμα απόθεμα που μπορεί να το επεξεργαστεί ο Pasteur.

Στην εικόνα 6.4 απεικονίζεται η κατάσταση των μηχανημάτων τη στιγμή 47:08 της προσομοίωσης. Στη συγκεκριμένη στιγμή, ο Filler βρίσκεται 8 λεπτά σταματημένος, Η μεταφορική γραμμή του Depall (buffer) δεν έχει απόθεμα, επομένως ο Filler παραμένει στην κατάσταση αναμονής και ο Pasteur τίθεται σε κατάσταση stand by αφού δεν υπάρχει απόθεμα στη μεταφορική του Filler.



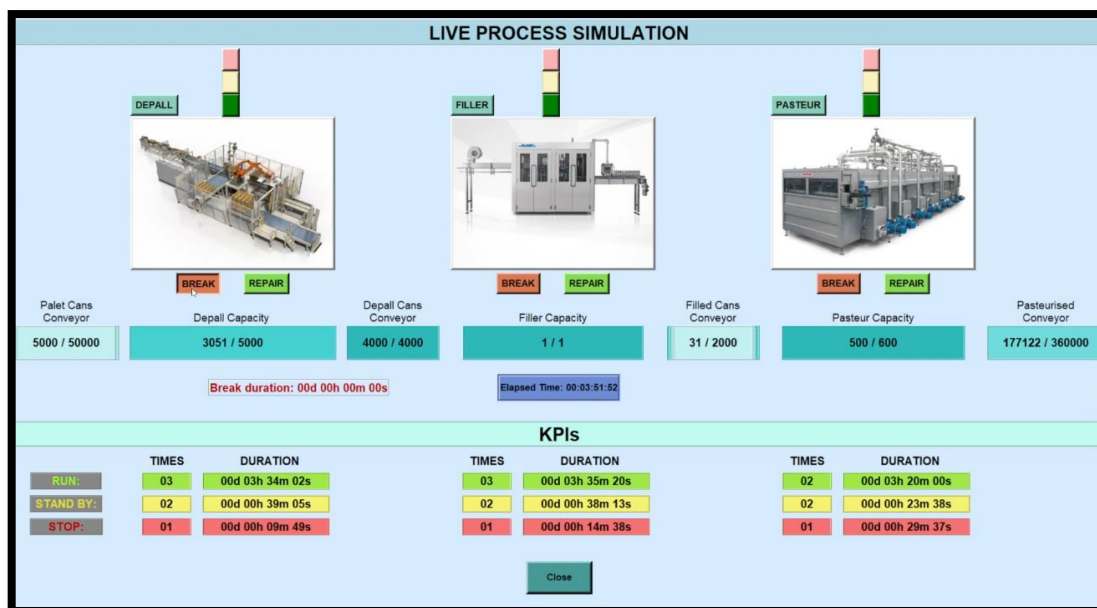
Εικόνα 6.4: Χρονική Στιγμή 47:08

Ο χρήστης επιλέγει τη χρονική στιγμή 50 της προσομοίωσης το κουμπί REPAIR. Με το πάτημα το μηχάνημα Depall τίθεται πάλι σε κατάσταση run και συνεχίζει να παράγει. Αμέσως τα μηχανήματα Filler και Pasteur επανέρχονται σε κατάσταση λειτουργίας αλλάζοντας τη κατάστασή τους σε run, όπως φαίνεται στη εικόνα 6.5 και τις καταστάσεις του φάρου.



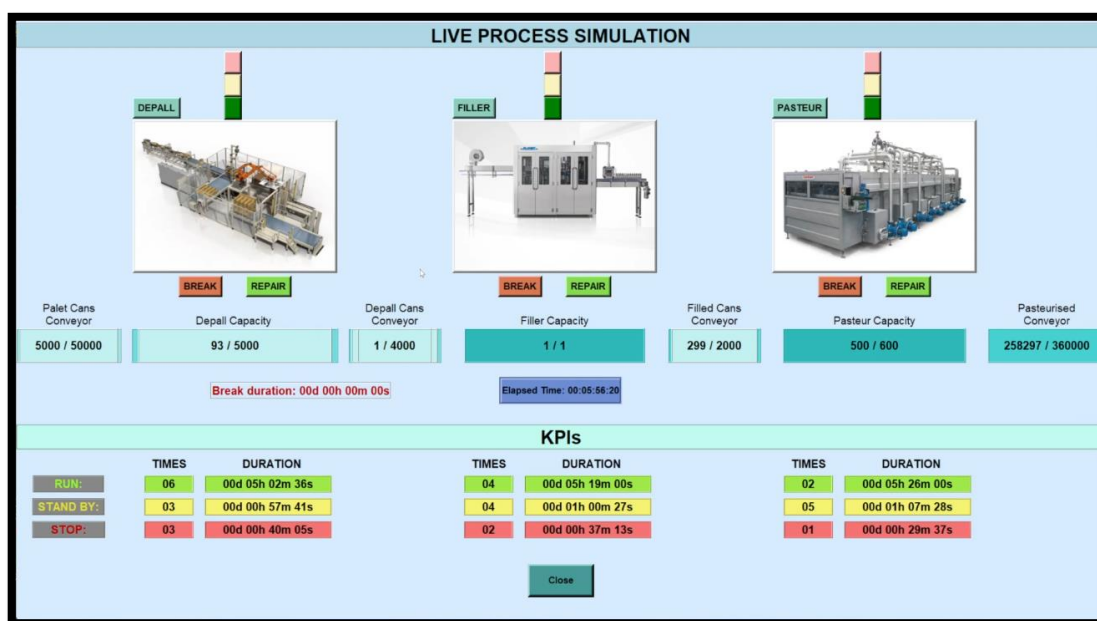
Εικόνα 6.5: Χρονική Στιγμή 51:42

Στη εικόνα 6.7 ο προσομοιωτής έχει φτάσει στη χρονική στιγμή 03:51:52, δηλαδή έχει εκτελέσει περίπου 4 ώρες προσομοίωσης της παραγωγικής διαδικασίας. Στην εικόνα αυτή φαίνονται οι μετρικές και οι καταστάσεις των μηχανημάτων τη δεδομένη στιγμή. Το Depall έχε τεθεί 1 φορά σε stop με διάρκεια 10 λεπτών, ο Filler έχει τεθεί 1 φορά με διάρκεια 15 λεπτών και ο Pasteur 1 φορά για χρόνο μισή ώρας.



Εικόνα 6.6: Χρονική Στιγμή 3:51:52

Η προσομοίωση συνεχίζεται και μέχρι την ολοκλήρωσή της ο χρήστης εκτελεί πρόσθετα σταματήματα σε ορισμένα μηχανήματα.



Εικόνα 6.7: Τέλος προσομοίωσης

Με την εικόνα 6.7 ολοκληρώνεται η διαδικασία της προσομοίωσης 6 ωρών. Στην εικόνα απεικονίζεται η συμπεριφορά της γραμμής κατά τη παραγωγή της. Μια πραγματική γραμμή παραγωγής, κατά τη διάρκεια 6 ωρών παρουσιάζει τουλάχιστον 5 σταματήματα συνολικά στη γραμμή. Η προσομοίωση που εκτελέστηκε προσεγγίζει επιτυχώς την πραγματική γραμμή. Η διάρκεια των σταματημάτων και της κατάστασης αναμονής είναι ανάλογη με την διάρκεια που θα παρουσίαζε μια πραγματική γραμμή με ακριβώς την ίδια συμπεριφορά. Κατά την ολοκλήρωση το Depall έχει τεθεί 6 φορές σε run με διάρκεια παραγωγής 5 ωρών, 3 φορές σε κατάσταση αναμονής με διάρκεια 60 λεπτών και 3 φορές σε κατάσταση stop με διάρκεια 40 λεπτών συνολικά. Για το Filler και το Pasteur απεικονίζονται στην εικόνα 6.7 αντίστοιχα.

Με την ολοκλήρωση της προσομοίωσης αυτόματα αναδύεται το παράθυρο του υποσυστήματος επισκόπησης προσομοίωσης. Το παράθυρο αυτό όπως φαίνεται από την εικόνα 6.8 παρέχει τα στατιστικά δεδομένα όλης της προσομοίωσης.

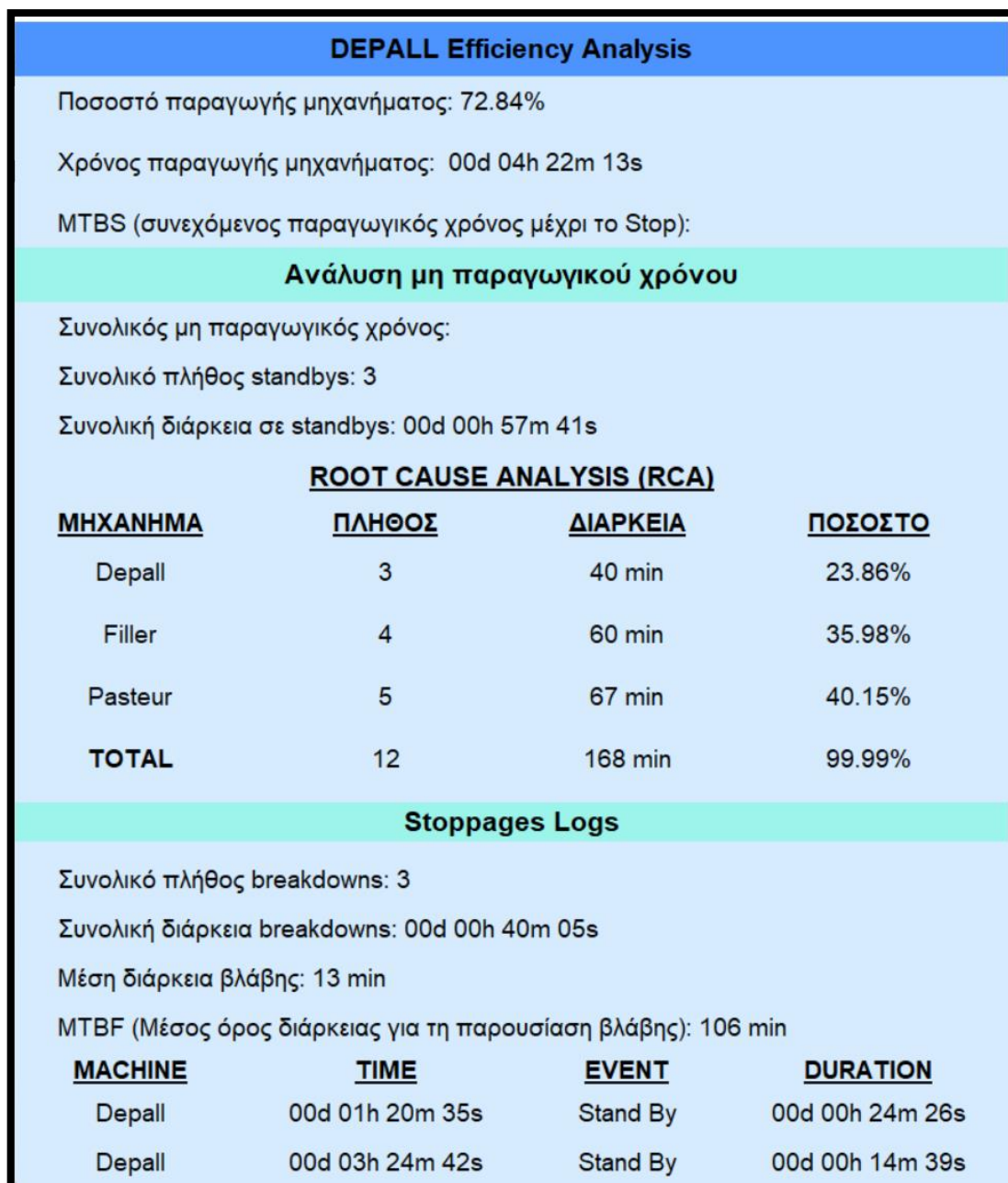
Δείκτες απόδοσης γραμμής				
Συνολική Διάρκεια Simulation: 00d 06h 00m 00s				
Ταχύτητα παραγωγής της γραμμής: 16.67 cans per second				
Προϊόντα που παράχθηκαν: 261798				
Προϊόντα που αναμέναμε να παραχθούν: 360000				
Συνολική απόδοση παραγωγής: 72.72%				
Καταστάσεις Μηχανημάτων				
<u>MACHINE</u>	<u>RUN</u>	<u>STAND BY</u>	<u>STOP</u>	
DEPALL	72.84%	16.02%	11.14%	DEPALL Analysis
FILLER	72.87%	16.79%	10.34%	FILLER Analysis
PASTEUR	73.03%	18.74%	8.23%	PASTEUR Analysis

Εικόνα 6.8: Επισκόπηση προσομοίωσης

Σύμφωνα με τα υποσύστημα επισκόπησης γραμμής, ο χρήστης κατανοεί πλήρως τη παραγωγική διαδικασία της γραμμής, που εκτελέστηκε για χρόνο 6 ωρών. Στο παράθυρο 6.8 απεικονίζεται η ταχύτητα παραγωγής που είχαν όλα τα μηχανήματα της γραμμής και η οποία είναι κατά μέσο όρο 17 κουτιά το δευτερόλεπτο, δηλαδή 1020 κουτιά το λεπτό. Στη συνέχεια, παρουσιάζεται το πλήθος των προϊόντων που παράξε συνολικά η γραμμή παραγωγής κατά τη διάρκεια της προσομοίωσης. Ο χρήστης έχει τη δυνατότητα να συγκρίνει την γραμμή με την παραγωγή μιας ιδανικής γραμμής χωρίς εμφάνιση δυσλειτουργιών και προβλημάτων. Συγκρίνοντας δηλαδή το πλήθος των αναμενόμενων προϊόντων. Και με την αναλογία των δυο τιμών υπολογίζεται η συνολική απόδοση της γραμμής η οποία ήταν 73%. Το ποσοστό ορίζει ότι η παραγωγή στο 73% του συνολικού χρόνου παραγωγής ήταν σε κατάσταση run και εκτελούσε τις διαδικασίες των μηχανημάτων χωρίς σταματήματα.

Στο πλαίσιο των καταστάσεων των μηχανημάτων απεικονίζεται ο πίνακας των τριών μηχανημάτων της γραμμής. Για κάθε μηχανήμα ορίζεται το ποσοστό της κάθε κατάστασης που βρισκόταν. Για το Depall, το ποσοστό παραγωγής του ήταν 72.84%, το ποσοστό αναμονής σε κατάσταση stand by: 16.02% και συνολικά 11.14% του χρόνου βρέθηκε σε κατάσταση βλάβης. Για το Filler και τον Pasteur φαίνονται στον πίνακα.

Ο χρήστης στη συνέχεια επιλέγει να γίνει η ανάλυση κάθε μηχανήματος. Ο αλγόριθμος του εμφανίζει το αντίστοιχο παράθυρο το οποίο αφορά αποκλειστικά τα στατιστικά δεδομένα του μηχανήματος που επέλεξε. Για το Depall, όπως αποδεικνύεται από την εικόνα 6.9 ήταν σε ποσοστό παραγωγής 73%, συνολικό χρόνο ίσο με 4 ώρες και 22 λεπτά.



Εικόνα 6.9: Depall analysis

Στη συνέχεια αναλύεται ο μη παραγωγικός χρόνος του μηχανήματος, δηλαδή η κατάσταση αναμονής (stand by). Στην κατάσταση αυτή βρέθηκε για 58 λεπτών. Ο αριθμός που τέθηκε σε αναμονή ήταν 3 φορές που συνολικά διήρκεσαν 57 λεπτά.

Επίσης, παρουσιάζεται ο πίνακας **RCA (Root Cause Analysis)** του μηχανήματος Depall. Κατά τον οποίο το Depall είχε συνολικά 3 σταματήματα 40 λεπτών. Αποτέλεσε το 24% του συνολικού μη παραγωγικού χρόνου των 3 μηχανημάτων. Ο Filler τέθηκε 4 φορές σε κατάσταση αναμονής για χρόνο 1 ώρας και αποτελεί το 36% του συνολικού μη παραγωγικού χρόνου του Depall. Τέλος, ο Pasteur τέθηκε 5 φορές σε κατάσταση αναμονής με συνολική διάρκεια 67 λεπτών το οποίο σπατάλησε το 40% του μη παραγωγικού χρόνου του Depall. Συνολικά υπήρξαν 12 stops και standby, τα οποία διήρκεσαν 168 λεπτά από το συνολικό χρόνο παραγωγής.

Αντίστοιχα εφαρμόζεται η τεχνική ανάλυσης για το Filler στην εικόνα 6.10 και για τον Pasteur στην εικόνα 6.11.

FILLER Efficiency Analysis			
Ποσοστό παραγωγής μηχανήματος: 72.87%			
Χρόνος παραγωγής μηχανήματος: 00d 04h 22m 19s			
MTBS (συνεχόμενος παραγωγικός χρόνος μέχρι το Stop):			
Ανάλυση μη παραγωγικού χρόνου			
Συνολικός μη παραγωγικός χρόνος:			
Συνολικό πλήθος standbys: 629			
Συνολική διάρκεια σε standbys: 00d 01h 00m 27s			
ROOT CAUSE ANALYSIS (RCA)			
<u>ΜΗΧΑΝΗΜΑ</u>	<u>ΠΛΗΘΟΣ</u>	<u>ΔΙΑΡΚΕΙΑ</u>	<u>ΠΟΣΟΣΤΟ</u>
Depall	3	57 min	35.52%
Filler	2	37 min	22.93%
Pasteur	5	67 min	41.55%
TOTAL	10	162 min	100.0%
Stoppages Logs			
Συνολικό πλήθος breakdowns: 2			
Συνολική διάρκεια breakdowns: 00d 00h 37m 13s			
Μέση διάρκεια βλάβης: 18 min			
MTBF (Μέσος όρος διάρκειας για τη παρουσίαση βλάβης): 161 min			
<u>MACHINE</u>	<u>TIME</u>	<u>EVENT</u>	<u>DURATION</u>
Filler	00d 00h 39m 11s	Stand By	00d 00h 00m 03s
Filler	00d 00h 39m 15s	Stand By	00d 00h 00m 03s

Εικόνα 6.10: Filler analysis

PASTEUR Efficiency Analysis			
Ποσοστό παραγωγής μηχανήματος: 73.03%			
Χρόνος παραγωγής μηχανήματος: 00d 04h 22m 54s			
MTBS (συνεχόμενος παραγωγικός χρόνος μέχρι το Stop):			
Ανάλυση μη παραγωγικού χρόνου			
Συνολικός μη παραγωγικός χρόνος:			
Συνολικό πλήθος standbys: 4048			
Συνολική διάρκεια σε standbys: 00d 01h 07m 28s			
ROOT CAUSE ANALYSIS (RCA)			
<u>ΜΗΧΑΝΗΜΑ</u>	<u>ΠΛΗΘΟΣ</u>	<u>ΔΙΑΡΚΕΙΑ</u>	<u>ΠΟΣΟΣΤΟ</u>
Depall	3	57 min	39.04%
Filler	4	60 min	40.92%
Pasteur	1	29 min	20.05%
TOTAL	8	147 min	100.01%
Stoppages Logs			
Συνολικό πλήθος breakdowns: 1			
Συνολική διάρκεια breakdowns: 00d 00h 29m 37s			
Μέση διάρκεια βλάβης: 29 min			
MTBF (Μέσος όρος διάρκειας για τη παρουσίαση βλάβης): 330 min			
<u>MACHINE</u>	<u>TIME</u>	<u>EVENT</u>	<u>DURATION</u>
Pasteur	00d 00h 39m 39s	Stand By	00d 00h 00m 01s
Pasteur	00d 00h 39m 40s	Stand By	00d 00h 00m 01s

Εικόνα 6.11: Pasteur Analysis

7. Συμπεράσματα και μελλοντικές

επεκτάσεις

7.1 Συμπεράσματα

Συνοψίζοντας, στην παρούσα εργασία επιτεύχθηκε η έρευνα και η ανάπτυξη ενός ολοκληρωμένου συστήματος προσομοίωσης, με εφαρμογές σε βιομηχανική διαδικασία παραγωγής. Η προσομοίωση δούλεψε ικανοποιητικά και μέσω του παρόντος συστήματος προσεγγίστηκε η πραγματική ροή της παραγωγικής διαδικασίας της βιομηχανίας.

Αρχικά, ο χρήστης της προσομοίωσης εισήγαγε τα δεδομένα που αντιστοιχούν στην πραγματική γραμμή παραγωγής, με ταχύτητα παραγωγής 60.000 κουτιά την ώρα, σύμφωνα με την δυναμικότητα των τριών μηχανημάτων και της διαδικασίας παραγωγής κάθε μηχανήματος.

Το σύστημα προσομοίωσης βοήθησε στην πλήρη απεικόνιση της συμπεριφοράς της πραγματικής γραμμής παραγωγής. Μέσα από αυτή, ο υπεύθυνος παραγωγής παρατήρησε πως επηρεάζονται τα μηχανήματα της παραγωγής όταν ένα μηχάνημα τεθεί σε κατάσταση stop. Το γεγονός αυτό, τον βοήθησε να εξάγει τα εξής συμπεράσματα:

1. Εντοπισμός συγκεκριμένων μηχανημάτων που είχαν συχνά σταματήματα λόγω δυσλειτουργιών και προβλημάτων
2. Χρειάζεται βελτιστοποίηση μηχανικών ρυθμίσεων σε ορισμένα μηχανήματα
3. Χρειάζεται να κάνει καλύτερη ρύθμιση παραμέτρων σε συγκεκριμένα μηχανήματα
4. Πρόσθετη προληπτική συντήρηση των μηχανημάτων και ανάπτυξη προβλεπτικής
5. Από τη συμπεριφορά της γραμμής αναδείχτηκαν οι εξωτερικοί παράγοντες που σταμάταγαν τη λειτουργία της παραγωγής και δεν οφείλοντας στα μηχανήματα
6. Εντοπισμός κρυφών αιτιών που δημιουργούσαν άλλα μηχανήματα
7. Αιτίες σταματημάτων που οφείλονταν είτε σε ανθρώπινο παράγοντα είτε σε μηχάνημα

Στη πραγματική παραγωγή, για την εκτέλεση πολλών και διαφορετικών σεναρίων θα απαιτούνταν ένα πλήθος από ενέργεια και κυρίως μεγάλο οικονομικό κόστος. Ο ρόλος του προσομοιωτή ήρθε να απαλλάξει το κόστος από τις πραγματικές γραμμές. Μέσα από αυτόν, εντοπίστηκαν τα βασικά προβλήματα της γραμμής. Πιο συγκεκριμένα, ο Buffer του Filler, είχε μικρή χωρητικότητα με αποτέλεσμα το Derall, να τίθεται σε κατάσταση αναμονής πολλαπλές φορές. Μέσω του προβλήματος αυτού, εντοπίστηκε ένα σημαντικό πρόβλημα το οποίο υπήρχε και στην πραγματική γραμμή. Το αποτέλεσμα ήταν να παρθούν μέτρα στο πραγματικό μηχάνημα του Filler έτσι ώστε να αποτραπούν τα συχνά σταματήματα.

Τέλος, ένας προσομοιωτής είναι σίγουρα χρήσιμος για ένα εργοστάσιο και μια βιομηχανία. Αναπαριστά κάθε παραγωγική διαδικασία και τη βελτιστοποιεί. Μέσω της εκτέλεσης διαφορετικών σεναρίων εντοπίζονται τα κυρίαρχα προβλήματα της γραμμής και λαμβάνονται τα αντίστοιχα μέτρα. Έτσι, αυξάνεται η απόδοση της γραμμής.

7.2 Μελλοντικές επεκτάσεις

Με βάση την έρευνα και τις παρατηρήσεις που έγιναν κατά την ανάπτυξη της παρούσας εργασίας, προτείνονται μερικές ενδιαφέρουσες ιδέες που θα μπορούσαν να επεκτείνουν το σύστημα που αναπτύχθηκε στην παρούσα διπλωματική εργασία:

- 1. Εισαγωγή στατικού αρχείου ρυθμίσεων:** Μέσω ενός αρχείου .json θα γίνεται η εισαγωγή των μεταβλητών που είναι υπεύθυνες για τη λειτουργία των μηχανημάτων. Όπως αυτές περιγράφονται στο κεφάλαιο Απαιτήσεις συστήματος. Ορθό θα ήταν να αποφευχθεί η αρχικοποίηση των τιμών τους στο κυρίως πρόγραμμα του συστήματος με σκοπό να αποτρέψει οποιαδήποτε παρεμβολή στο κώδικα είτε δυσκολία στο προγραμματισμό. Αυτές οι μεταβλητές δίνονται μια φορά από τον χρήστη και είναι στατικές είτε δυναμικές για την εκτέλεση του προγράμματος. Άρα η μορφή αρχείου json θα εισάγει τις αυτές μεταβλητές στο κυρίως πρόγραμμα το οποίο με βάσει αυτές θα τρέχει την προσομοίωση. Το αρχείο ρυθμίσεων έχει υλοποιηθεί σε μεγάλο βαθμό στην παρούσα διπλωματική, όμως απαιτείται να επεκταθεί και χρησιμοποιείται αποκλειστικά και μόνο, χωρίς να εισάγονται μεταβλητές και παράμετροι στο κυρίως πρόγραμμα.
- 2. Επέκταση του συστήματος Graphical User Interface – GUI:** Σε μελλοντική επέκταση της παρούσα διπλωματική, αποτελεί η δημιουργία νέου γραφικού περιβάλλοντος σε διαδικτυακή μορφή. Σκοπός της υλοποίησης είναι να ενσωματωθεί σε cloud based πλατφόρμες έτσι ώστε να είναι προσβάσιμη από οποιοδήποτε σημείο. Η λειτουργία αυτή θα διευκολύνει το προγραμματισμό του UI και θα υλοποιηθεί σε γλώσσες προγραμματισμού ή frameworks που αφορούν αποκλειστικά GUI, όπως Angular. Έτσι, το γραφικό περιβάλλον θα γίνει πολύ πιο φιλικό προς το χρήστη, θα προσαρμόζεται εύκολα στις ανάγκες του και θα είναι πιο interactive στο να εκτελεί ενέργειες κατά τη διάρκεια εκτέλεσης της προσομοίωσης.
- 3. Εισαγωγή δεδομένων σε σύστημα βάσης δεδομένων:** Η ορθή καταγραφή τις συμπεριφοράς του προσομοιωτή, των μηχανημάτων και η καταγραφή των μετρικών – αποτελεσμάτων θα γίνεται σε σύστημα διαχείρισης δεδομένων δηλαδή σε βάση δεδομένων. Εκεί θα υπάρχει ολόκληρο το ιστορικό της γραμμής παραγωγής που προσομοιώνει το σύστημα. Έτσι θα αποθηκεύεται το ιστορικό της γραμμής παραγωγής ύστερα από πολλαπλές εκτελέσεις της προσομοίωσης. Για να προσεγγιστεί πιστά η πραγματική γραμμή παραγωγής από τον προσομοιωτή, πρέπει να εκτελεστούν ένα πλήθος από προσομοιώσεις. Για αυτό το λόγο όλα τα δεδομένων πρέπει να καταχωρούνται στη βάση δεδομένων και έπειτα από τον αλγόριθμο να επεξεργάζονται. Με αυτόν το τρόπο θα παρέχονται ακριβή δεδομένα και η πιο πραγματική προς τη προσεγγίζουσα συμπεριφορά της γραμμής. Η υλοποίηση έχει εφαρμοστεί μέχρι ένα σημείο στη παρούσα διπλωματική, όμως είναι απαραίτητο να επεκταθεί.
- 4. Περιβάλλον πραγματικού χρόνου:** Μέσα από επιλογή του γραφικού περιβάλλοντος ο χρήστης θα έχει την δυνατότητα να εναλλάσσει το περιβάλλον υλοποίησης του προσομοιωτή. Πιο συγκεκριμένα, αν επιθυμεί η εκτέλεση της προσομοίωσης να

ολοκληρωθεί σε πολύ σύντομο χρονικό διάστημα που δεν αναπαριστά τον πραγματικό χρόνο, μπορεί να το επιλέξει. Επίσης, θα παρέχεται η δυνατότητα για τον χρήστη να επιλέγει την προσομοίωση της παραγωγικής διαδικασίας σε πραγματικό χρόνο. Δηλαδή, ο χρόνος με τον οποίο θα εκτελείται η διαδικασία της παραγωγής θα περατώνεται αντίστοιχα με την περάτωση του πραγματικού χρόνου.

8. Βιβλιογραφία

- [1] J. Banks, «Simulation languages and simulators,» WSC, December 1992.
- [2] E. Sujova, H. Cierna και R. Bambura, «Simulation model of production as tool for industry 4.0 implementation into practice,» *18th International Scientific Conference Engineering for Rural Development*, May 2019.
- [3] W. C. Glassey, *The Theory and Practice of Time Study*, London: Cox & Wyman Limited, 1966.
- [4] M. Baudin, *Lean Assembly: The Nuts and Bolts of Making Assembly Operations Flow.*, New York: Productivity Press, 2002.
- [5] T. ARTUN, *Assembly line design and optimization*, Sweden: CHALMERS UNIVERSITY OF TECHNOLOGY, 2009.
- [6] Anon, Researching Capacity Utilization, [Online] http://www.acclaropartners.com/pubs/Acclaro_White_Paper_Researching_Capacity_Util, 2010.
- [7] S. Nakajima, *Introduction to TPM: Total Productive Maintenance*, Portland: Productivity Pres, 1988.
- [8] F. Wauters και J. Mathot, *OEE Overall Equipment Effectiveness.*, [Online] <http://www05.abb.com/global/scot/scot296.nsf/veritydisplay/4581d5d1ce980419c1256bf>, 2002.
- [9] T. McLoughlin, *Guide to using Overall Equipment Effectiveness (OEE) and a process for Continuous Improvement*, Kraft Foods, 2009.
- [10] R. & R. Division, *Root Cause Analysis Handbook: A Guide to Effective Incident Investigation*, ABS Group, 2005.
- [11] A. Lokrantz, E. Gustavsson και M. Jirstrand, *Root cause analysis of failures and quality deviations in manufacturing using ML*, Gothenburg: Elsevier, 2018.
- [12] W. Sherrie A., Tromp και Diana, *Root Cause Analysis for Higher Education*.
- [13] D. Cmpanga, *Product and Production Process Modeling and Configuration*, Univerisity of Perugia, 2012.
- [14] A. Downey, *Modeling and Simulation in Python*, Massachusetts: Green Tea Press, 2017.
- [15] H. D. Karatza, *Simulation Modelling Practice and Theory*, Thessaloniki: Science Direct.
- [16] K. Ehmann, S. G. Kapoor, R. E. DeVor και I. Lazoglu, *Machining process modeling: A review*, Mechanical Science and Engineering Micro and Nanotechnology Lab, 1997.
- [17] Q. Meredith, V. Katherine και R. Robert, *The Accuracy and Reliability of Measurements Made on Computer-Based Digital Models*, The Angle, 2004.
- [18] J. Schoenwaelder, *On the Difference between Information Models and Data Models*, University of Osnabrueck, 2003.
- [19] W. Kritzinger, M. Karner, G. Traar, J. Henjes και W. Sihn, *Digital Twin in manufacturing: A categorical literature review and classification*, Viena: Elsevier, 2018.
- [20] F. Aidan, F. Zhong και C. Barlow, *Digital Twin: Enabling Technologies, Challenges and Open Research*, IEEE, 2020.
- [21] D. Solomatine, L. See και R. Abrahart, *Data-Driven Modelling: Concepts, Approaches and Experiences*, UK, 2008.

- [22] N. Ali, K. Petersen και C. Wohlin, A Systematic Literature Review on the Industrial Use of Software Process Simulation, *Journal of Systems and Software*. 97: 65–85., 2014.
- [23] K. Marc, J. Raymond, B. Madachy, M. David και C. Raffo, *Software process simulation modeling: Why? What? How?*, Elsevier, 1999.
- [24] «Tecnomatix,» Siemens, [Ηλεκτρονικό]. Available: <https://www.plm.automation.siemens.com/global/en/products/tecnomatix/>.
- [25] «Aspen Plus,» AspenTech, [Ηλεκτρονικό]. Available: <https://www.aspentech.com/en/products/engineering/aspen-plus>.
- [26] I. Rossetti, *Process Simulation for Industrial Process Design*, Italy: Omics International, 2017.
- [27] «Anylogic,» [Ηλεκτρονικό]. Available: <https://www.anylogic.com/>.
- [28] L. Dapeng, W. Qing και X. Junchao, The role of software process simulation modeling in software risk management: A systematic review, China: ESEM, 2009.
- [29] T. Fei, Z. He, L. Ang και A. Y. C. Nee, Digital Twin in Industry: State-of-the-Art, IEEE, 2019.
- [30] SimPy. [Ηλεκτρονικό]. Available: <https://simpy.readthedocs.io/en/latest/>.
- [31] R. Cooke, A. Bosma και F. Harte, A practical model of Heineken’s bottle filling line with dependent failures.
- [32] K. Funaki και K. Yoshimoto, Distribution of total uptime during a given time interval, IEEE, 1994.
- [33] J. Muth, A method for predicting system downtime (97-102), IEE, 1968.
- [34] L. Takacs, On certain sojourn time problems in the theory of stochastic processes (169-191), *Acta Mathematica*, 1957.
- [35] A. Renyi, On the asymptotic distribution of the sum of a random number of independent random variacles, *Acta Mathematica*, 1995.
- [36] H. MAnning, *Computer simulation of packaging lines*, Brewery, 1985.
- [37] H. Krosender και P. Hinterwimmer, Problems arising within a bottle line, *MBA A Technical Quarterly*, 1978.
- [38] R. Murphy, «Examining the distribution of buffer protection,» *AIIE Trans*, pp. 113-120, 1979.