



**UNIVERSITÀ DI PISA**

**SCUOLA DI INGEGNERIA**

Corso di Laurea in Ingegneria Informatica

Anno Accademico 2018/2019

**Classificazione degli stati funzionali di  
una linea produttiva del settore tissue  
attraverso i dati di log**

Candidato

Davide Falcone

Relatori

Prof. Mario G.C.A. Cimino

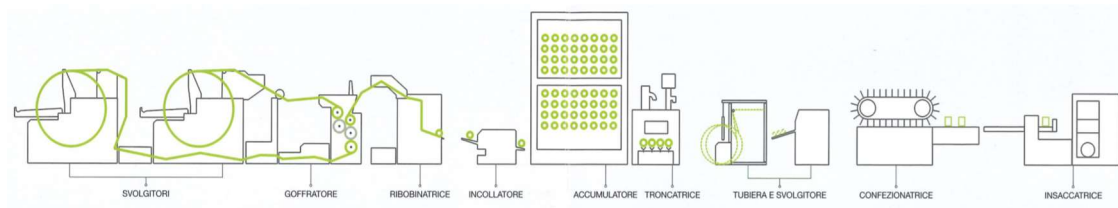
Prof. Ssa Gigliola Vaglini

Ing. Marco Omeri

# Sommario

1. Introduzione	3
2. Primo passo : downtime e uptime	4
3. Secondo passo : rilevare il cambio di lama	6
1. Primo tentativo	6
2. Secondo tentativo	7
3. Terzo tentativo	9
4. Quarto tentativo	11
5. Quinto tentativo	15
4. Terzo passo : analisi degli intervalli di downtime	17
5. Design della caratterizzazione data-driven dei downtimes della macchina	22
6. Introduzione allo pseudocodice	26
7. Pseudocodice	29
8. Risultati e conclusioni	32
9. Sitografia	33
10. Ringraziamenti	34

# 1. INTRODUZIONE



*Figura 1: Linea di produzione del settore tissue*

Una linea produttiva del settore tissue è formata da molti macchinari in serie, ognuno con la sua funzione.

La troncatrice è quella macchina, all'interno della linea produttiva, che ha il compito di tagliare in tanti rotoli di dimensione prefissata, quei lunghi rotoli che riceve in ingresso dall'accumulatore.

La troncatrice è dotata di una lama che si consuma nel tempo ed è dotata di sensori che rilevano la velocità della lama, la sua condizione (in percentuali di utilizzo) ed altri dati che vengono inviati sistematicamente ogni 4 secondi ad un database.

L'obiettivo di questa tesi è quello di studiare i dati prodotti da questi sensori e inviati al database per poter riconoscere in che stato si trova la troncatrice in ogni momento della sua produzione.

Più precisamente, l'obiettivo è quello di calcolare la durata per la quale la troncatrice non è in produzione a causa di un cambio lama.

Per cambiare la lama, nelle troncatrici, è necessaria infatti una procedura effettuata da un operatore esperto che deve rispettare determinate norme di sicurezza.

È possibile effettuare un upgrade della troncatrice attraverso l'utilizzo di un macchinario che permette di effettuare il cambio lama automaticamente, senza aver bisogno della presenza di un operatore.

Lo scopo finale di questa tesi è infatti il confronto tra un cambio di lama effettuato da un operatore e un cambio di lama effettuato automaticamente dalla macchina. Questo confronto è fondamentale per capire quando conviene effettuare l'upgrade ad una troncatrice.

## 2. PRIMO PASSO : DOWNTIME E UPTIME

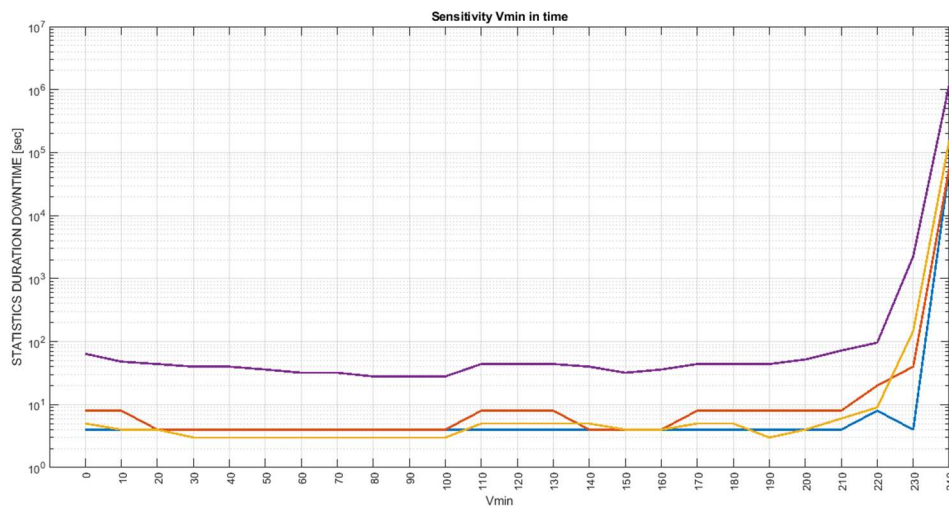
Il primo passo per l'obiettivo finale è studiare i dati prelevati dal database per stabilire quando la troncatrice è in funzione (uptime) e quando la troncatrice non è in funzione (downtime). Il dato più importante per questa fase d'analisi è sicuramente la velocità della troncatrice.

Per uno studio corretto dei tempi di downtime, è necessario stabilire una velocità (chiamata  $V_{min}$ ) di soglia sotto la quale la troncatrice viene ritenuta in downtime. Per stabilire  $V_{min}$ , occorre prima effettuare un'analisi di sensibilità della soglia, in modo tale da studiare come varia il tempo di downtime e il numero di cambi di stato al variare di  $V_{min}$ .

Occorre inoltre verificare la velocità di Jog con un tecnico della macchina. Per velocità di Jog si intende quel micro avanzamento della macchina usato durante gli interventi manuali.

L'analisi di sensibilità della soglia viene effettuata su dati prodotti dalla troncatrice riguardo la sua velocità di esecuzione.

Con l'utilizzo di uno script di Matlab si genera un plot in cui vengono evidenziati il tempo totale di downtime al variare di  $V_{min}$ .



*Figura 2 : analisi sensibilità per determinare  $V_{min}$*

Dopo un confronto con un tecnico della macchina si è deciso di settare la velocità di soglia  $V_{min}$  a 15 tagli al minuto e una velocità di Jog di 5 tagli al minuto. Utilizzando quindi uno script di Matlab (che chiameremo `extractAllVariablesAndClockLS1.m`) progettato per lo studio degli stati funzionali della ribobinatrice (altra macchina nella linea produttiva) è possibile modificare tale script per implementare la nuova logica della velocità di soglia per ottenere un plot dello stato funzionale della troncatrice.

extractAllVariablesAndClockLS1.m ci permette inoltre di avere un plot (come in Figura 3) per ogni macchina presa in studio (in totale sono 14).

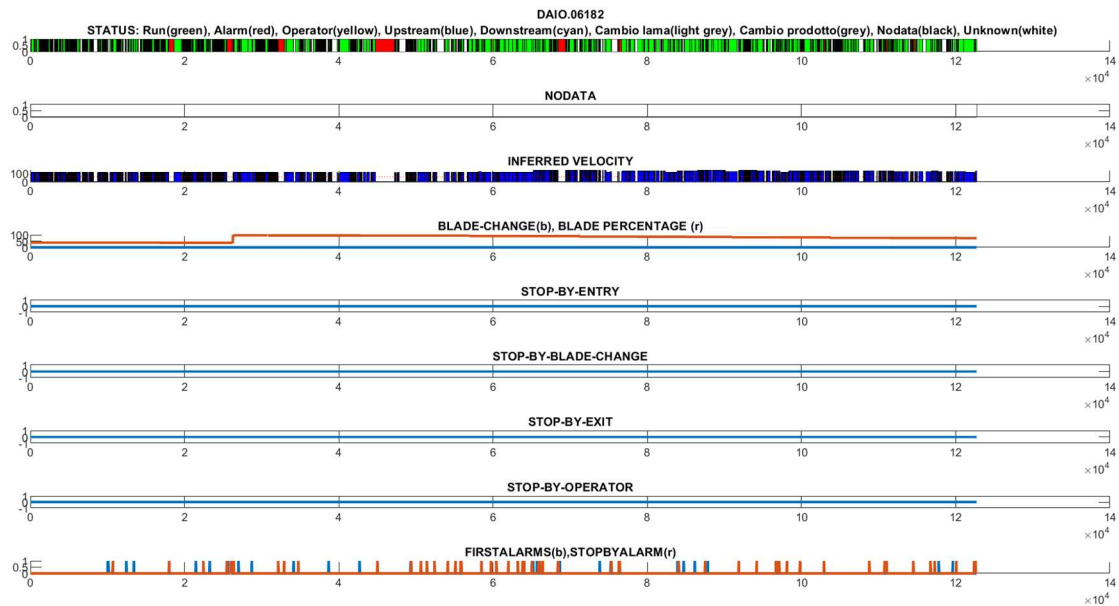


Figura 3: Uno dei plot ottenuti da extractAllVariablesAndClockLS1.m

Come si vede in Figura 3, il plot riconosce quando la macchina è in uptime grazie all'analisi della velocità.

Inoltre è possibile notare dal 4° plot a partire dall'alto che vi è un cambio di lama non riconosciuto. Infatti la percentuale della lama cresce a dismisura in un momento ma la variabile BLADECHANGE non viene settata. Ciò è causato, come nel caso precedente, al fatto che lo script è nato per una macchina diversa (la ribobinatrice) e quindi si basa su una logica di funzionamento differente.

Questo sarà il prossimo problema da affrontare.

### 3. SECONDO PASSO : RILEVARE IL CAMBIO DI LAMA

#### 3.1 Primo tentativo

Si ipotizza quindi una logica per rilevare il momento in cui viene cambiata una lama. Una lama può essere sostituita solo se inferiore ad una certa percentuale (Beta) e una lama nuova può essere immessa nel macchinario solo se superiore ad una certa percentuale (Alpha).

Per un istante T, si campionano i valori percentuali della lama 1, 5, 10 minuti prima di T e 1,5,10 minuti dopo T.

Il cambio di lama è vero se soddisfa la seguente logica :

- I valori a T(i-1minuto), T(i-5minuti), T(i-10 minuti) devono essere minori di Beta
- I valori a T(i+1minuto), T(+5minuti), T(i+10 minuti) devono essere maggiori di Alpha
- Le differenze [T(i+1minuto) - T(i-1minuto)], [T(i+5minuti) - T(i-5minuti)], [T(i+10minuti) - T(i-10minuti)] devono essere maggiori di (Alpha-Beta)

$$CL = \left\{ \begin{array}{l} \exists t_i \in D: PercBlade(t_{i-1min}) < Beta, \\ PercBlade(t_{i-5mi}) < Beta, \\ PercBlade(t_{i-1}) < Beta, \\ PercBlade(t_{i+1m}) > Alpha, \\ PercBlade(t_{i+5mi}) > Alpha, \\ PercBlade(t_{i+10m}) > Alpha, \\ PercBlade(t_{i+1mi}) - PercBlade(t_{i-1min}) > (Alpha - Beta), \\ PercBlade(t_{i+5min}) - PercBlade(t_{i-5min}) > (Alpha - Beta), \\ PercBlade(t_{i+10m}) - PercBlade(t_{i-10min}) > (Alpha - Beta) \end{array} \right\}$$

*Formula 1: Prima formula per rilevazione del cambio lama (Alpha 70 Beta 25)*

Dopo un'analisi manuale sui grafici riguardo l'andamento della percentuale della lama, si decide di utilizzare, al primo tentativo, come soglie Alpha e Beta rispettivamente 70 e 25.

Viene effettuata quindi una modifica a `extractAllVariablesAndClockLS1.m` per implementare la logica di riconoscimento del cambio lama e fatto girare lo script per ottenere i plot delle singole macchine.



*Figura 4 : Particolare di uno dei plot ottenuti da `extractAllVariablesAndClockLS1.m` dopo la modifica per implementare la logica per il rilevamento del cambio lama*

La Figura 4 mostra un plot generato precedentemente dallo script. In questo plot non si riesce a distinguere bene la variabile BladeChange, quindi, per poter studiare

il suo comportamento, viene effettuata una modifica al codice dello script per avere 2 grafici separati.

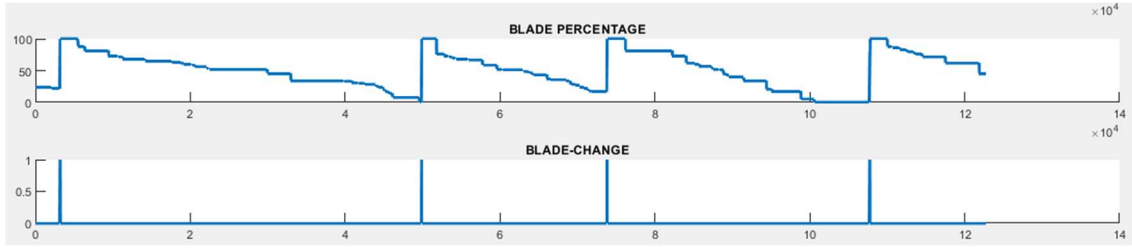


Figura 5 : Particolare del plot, diviso in due sottoplot per una migliore lettura

Ad un primo sguardo della Figura 5, può sembrare di aver scelto una soluzione adeguata. In realtà ci sono vari momenti (come si vede dalla Figura 6) in cui BladeChange è settato al valore logico alto, quindi per ogni intervallo si sceglie di selezionare soltanto il primo valore.

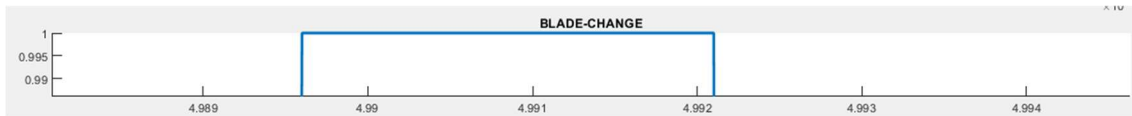


Figura 6 : dettaglio zoomato della variabile BladeChange

### 3.2 Secondo tentativo

Per risolvere il problema sorto con il primo tentativo, si decide di selezionare soltanto il primo valore di ogni intervallo.

Per fare ciò, effettuiamo uno shift di una posizione del vettore dei valori di BladeChange e facciamo un AND logico tra il vettore originale e l'inverso (NOT) del vettore shiftato.

$$CL = \left\{ \begin{array}{l} \exists t_i \in D: PercBlade(t_{i-1min}) < Beta, \\ PercBlade(t_{i-5min}) < Beta, \\ PercBlade(t_{i-10min}) < Beta, \\ PercBlade(t_{i+1m}) > Alpha, \\ PercBlade(t_{i+5min}) > Alpha, \\ PercBlade(t_{i+10min}) > Alpha, \\ PercBlade(t_{i+1mi}) - PercBlade(t_{i-1min}) > (Alpha - Beta), \\ PercBlade(t_{i+5min}) - PercBlade(t_{i-5mi}) > (Alpha - Beta), \\ PercBlade(t_{i+10min}) - PercBlade(t_{i-10min}) > (Alpha - Beta) \end{array} \right\}$$

Formula 2: Prima parte della formula del secondo tentativo (Alpha 70 Beta 25)

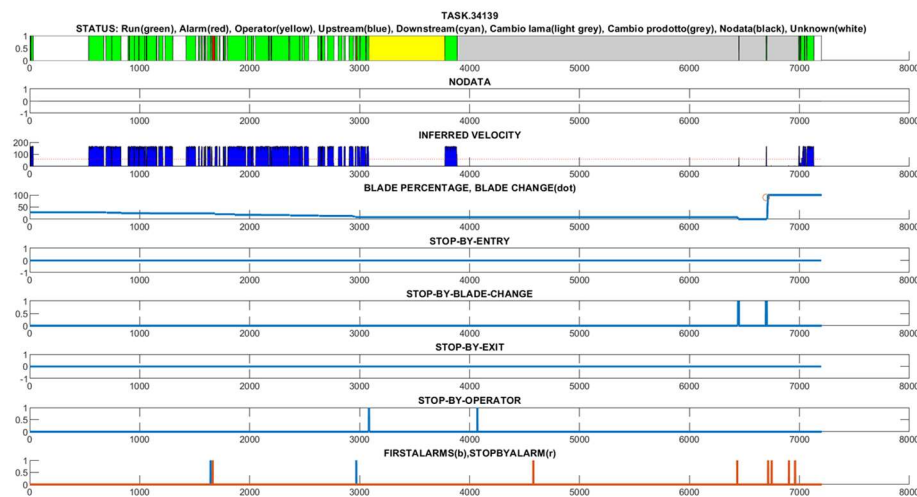
$$CL = CL \wedge \neg ShiftedCL$$

Formula 3: Operazione per selezionare il primo valore di ogni intervallo

Oltre a questa modifica, si decide di modificare nuovamente lo script `extractAllVariablesAndClockLS1.m` per una visione più facile della variabile `BladeChange`.

Si riduce il numero di plot e si inserisce un pallino rosso dove la variabile `BladeChange` viene settata.

Si analizza quindi un intervallo ridotto per vedere meglio la classificazione del tipo di downtime e la rilevazione del cambio lama.



*Figura 7: intervallo di 8 ore della macchina TASK in cui il cambio lama è rilevato correttamente*

Nella Figura 7 viene riconosciuto correttamente (e una sola volta) il momento in cui vi è un cambio lama e l'intervallo di downtime ad esso associato viene riconosciuto correttamente.

In seguito ad un'analisi dei risultati dello script su tutte le macchine, sono stati rilevati 4 casi in cui lo script non ha correttamente rilevato il cambio lama (ovvero non è stata settata la variabile `BLADECHANGE` a 1). Questi casi sono illustrati in Figura 8, 9, 10, 11.



*Figura 8: macchina Aktul con un cambio lama perso*



*Figura 9: macchina Daio con un cambio lama perso*





Figura 10: macchina Task con un cambio lama perso



Figura 11: macchina Tedim con un cambio lama perso

Quindi, come si vede dalle figure, la logica ipotizzata al secondo tentativo non è corretta.

### 3.3 Terzo tentativo

Analizzando i plot delle macchine che presentano i problemi, si ipotizzano due cause :

- La soglia di Beta è troppo alta, infatti come si vede nella Figura 12 è stato effettuato un cambio lama quando la vecchia lama aveva una percentuale di XX, superiore a 25.
- Nella logica per il riconoscimento del cambio lama, non vanno considerati i campioni precedenti e successivi di un minuto. Come si vede in Figura 13, è presente uno spike della durata maggiore di 1 minuto.

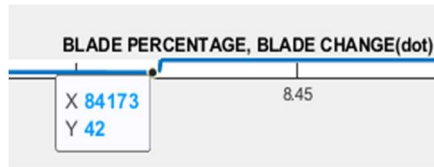


Figura 12: particolare di un cambio lama perso a causa di Beta

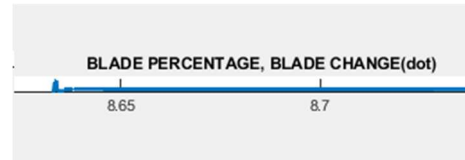


Figura 13: particolare di un cambio lama perso a causa di uno spike maggiore di 1 minuto

Lo spike è causato da un movimento involontario della fotocellula che analizza lo status della lama. Infatti è probabile che durante il cambio lama, l'operatore possa spostare tale fotocellula che rileva dati non corretti.

Si decide quindi di alzare la soglia di Beta a 45 e di eliminare i campioni precedenti e successivi di un minuto.

$$CL = \left\{ \begin{array}{l} \exists t_i \in D: PercBlade(t_{i-5min}) < Beta, \\ PercBlade(t_{i-10min}) < Beta, \\ PercBlade(t_{i+5mi}) > Alpha, \\ PercBlade(t_{i+10min}) > Alpha, \\ PercBlade(t_{i+5min}) - PercBlade(t_{i-5min}) > (Alpha - Beta), \\ PercBlade(t_{i+10min}) - PercBlade(t_{i-10min}) > (Alpha - Beta) \end{array} \right\}$$

Formula 4: Prima parte della formula del terzo tentativo (Alpha 75 Beta 45)

$$CL = CL \wedge \neg \text{ShiftedCL}$$

*Formula 5: Operazione per selezionare il primo valore di ogni intervallo*

Si modifica quindi nuovamente lo script per implementare la nuova logica per il riconoscimento del cambio lama.

In seguito alla modifica dello script, possiamo vedere (nelle Figure da 14 a 17) che i problemi precedenti sono stati risolti.



*Figura 14: macchina Aktul corretto*



*Figura 15: macchina Daio corretto*



*Figura 16: macchina Task corretto*



*Figura 17: macchina Tedim corretto*

Ricordiamo che l'obiettivo finale dello studio dei comportamenti della troncatrice è quello di avere una figura chiara sulla durata del tempo medio che un operaio impiega a cambiare una lama.

Tutto ciò serve poiché c'è la possibilità di installare un meccanismo all'interno della troncatrice che riesce ad effettuare il cambio lama in 3 minuti. Il passo successivo sarà quindi quello di modellare il tempo complessivo di downtime cambio lama per ogni macchina e il tempo medio per ogni cambio lama di ogni macchina.

Si crea quindi un primo prototipo dello script finale che sarà spiegato meglio nei capitoli successivi. Questo script ci sarà utile anche in questa fase per poter concludere se la logica del cambio lama può considerarsi definitiva o meno. Una prima bozza dello script appena creato (che chiameremo d'ora in poi *anDowntimeCL*), rileva però una situazione anomala.

Avendo cambiato la logica di rilevazione (da second a terzo tentativo) del cambio lama allo script principale (ovvero non si fa nessun confronto con campioni precedenti e seguenti 1 minuto), il tempo che viene considerato come downtime legato al cambio di lama scende drasticamente in alcuni casi.

DATI CON SCRIPT ORIGINALE, OVVERO SENZA AVER TOLTO IL CONTROLLO SUI CAMPIONI PRIMA E DOPO 1 MINUTO			
DATI CON SCRIPT MODIFICATO, OVVERO DOPO AVER TOLTO IL CONTROLLO SUI CAMPIONI PRIMA E DOPO 1 MINUTO			
Macchina	Downtime Cambio Lama Totale	Nr Cambi Lama	Cambi lama assenti secondo il grafico
AKTUL.34112_AllVariablesLS1.csv	29096	4	0
CARDINAL_TISSUE.20360_AllVariablesLS1.csv	0	0	0
CARTINDUSTRIA_VENETA.33071_AllVariablesLS1.csv	288	1	0
DAIO.06181_AllVariablesLS1.csv	0	0	0
DAIO.06182_AllVariablesLS1.csv	672	1	0
EUROVAST.33123_AllVariablesLS1.csv	0	0	0
FADERCO.34155_AllVariablesLS1.csv	14404	1	1
FAPSA.8568_AllVariablesLS1.csv	32	1	0
FLORYCART.33007_AllVariablesLS1.csv	0	0	0
SUAVECEL.34150_AllVariablesLS1.csv	1404	1	0
TASK.34096_AllVariablesLS1.csv	116	2	0
TASK.34139_AllVariablesLS1.csv	15196	4	0
TASK.34186_AllVariablesLS1.csv	36	4	0
TEDIM.33117_AllVariablesLS1.csv	8540	1	0

Figura 19: risultati script con logica del terzo tentativo : in verde ci sono i valori che sono aumentati dopo la modifica, in rosso quelli che sono diminuiti (o persi). In giallo è evidenziato la situazione di Faderco, in cui è stato rilevato un nuovo cambio lama ma ce n'è comunque uno assente (vedere Figura 20)

Nella Figura 18 si possono vedere i dati prodotti dallo script con la logica del secondo tentativo, relativi al numero di cambi lama rilevati, al numero di cambi lama mancanti e il tempo totale (in secondi) in cui la macchina è considerata in downtime causata da cambio lama.

Come si può vedere, il terzo tentativo ha portato un miglioramento nel rilevamento del numero dei cambi lama, ma ha introdotto un calo drastico (e irrealistico ad esempio nel caso di TASK 34816 in cui il tempo totale di downtime è 36 secondi per 4 cambi di lama) del tempo totale di downtime.



Figura 20: Caso Faderco, un cambio lama viene rilevato e uno no

Inoltre la modifica del terzo tentativo, come si vede nel caso illustrato a Figura 20, non ha risolto completamente la mancanza di rilevazione del cambio lama (ovvero la transazione a valore alto della variabile BLADECHANGE).

### 3.4 Quarto tentativo

Col quarto tentativo cerchiamo di risolvere i problemi riguardo la durata non possibile dei downtime totali della maggior parte delle macchine.

Per avere un'immagine più dettagliata del problema, si decide di studiare l'andamento della macchina TASK 34096 dato che essa, con entrambe le versioni dello script, non ha nessun problema di rilevamento di cambio lama ma presenta il problema della durata ridotta con lo script modificato.

Si analizza soltanto il periodo di 4 ore intorno al primo cambio lama per avere una visione più dettagliata.

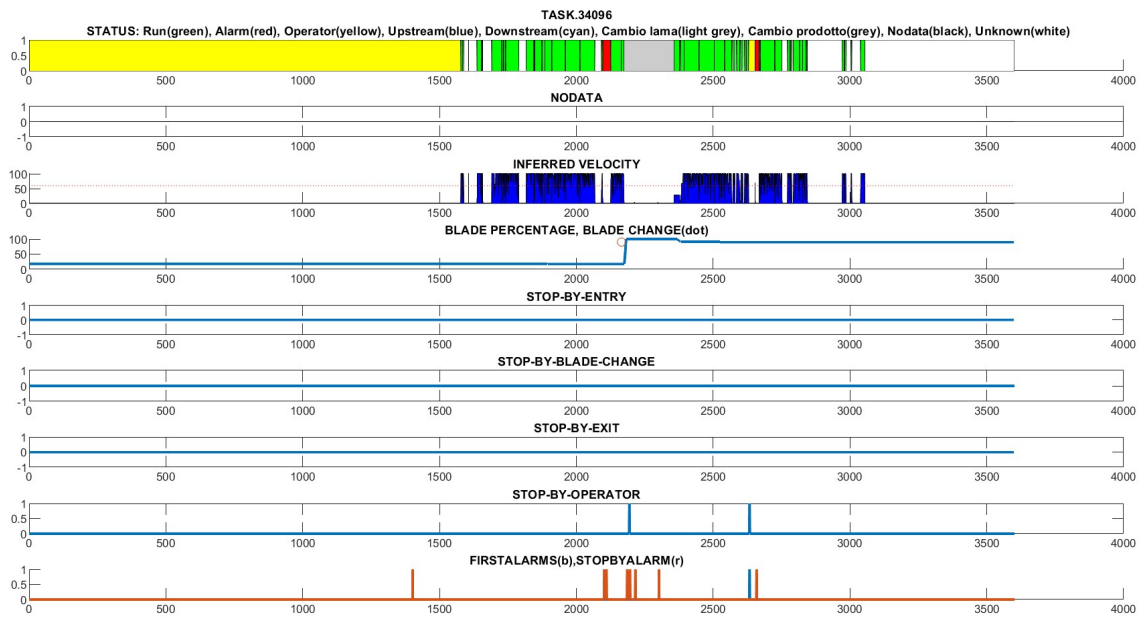


Figura 21 : TASK 34096 intervallo di 4 ore con script del secondo tentativo

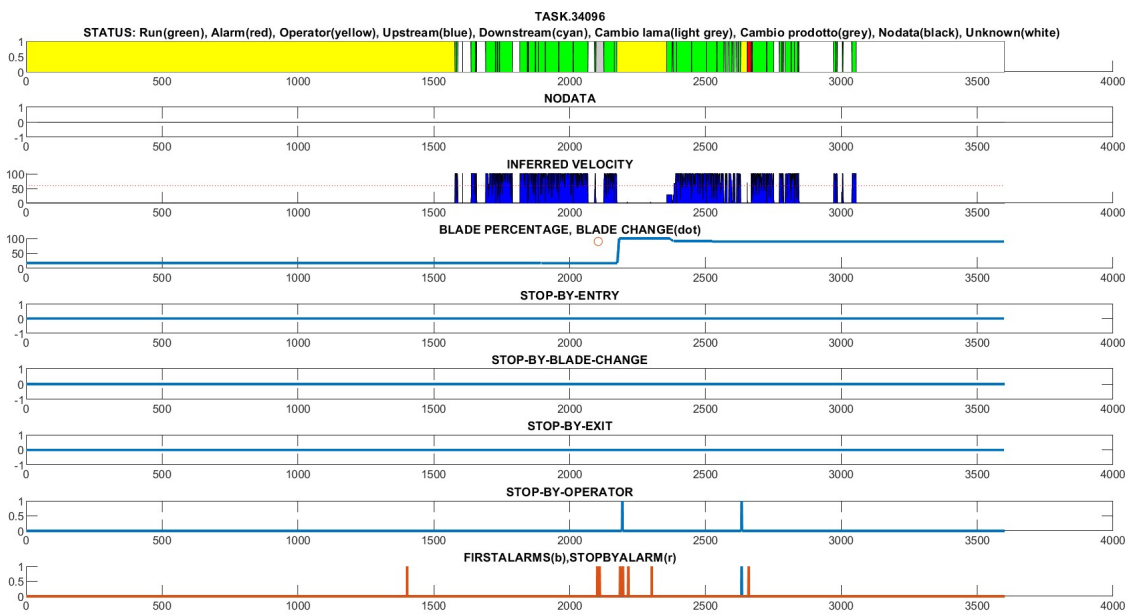


Figura 22 : TASK 34096 intervallo di 4 ore con script del terzo tentativo

	A	B	C	D	E
1	TIMESTAMP	VELOCITY	BLADE	BLADECHANGE	DOWNTIMETYPE
50361	2019-06-24 07:57:17	102	17	0	0
50362	2019-06-24 07:57:21	66	17	0	0
50363	2019-06-24 07:57:25	101	17	0	0
50364	2019-06-24 07:57:29	0	17	0	8
50365	2019-06-24 07:57:33	102	17	1	0
50366	2019-06-24 07:57:37	102	17	0	0
50367	2019-06-24 07:57:41	102	17	0	0
50368	2019-06-24 07:57:45	102	17	0	0
50369	2019-06-24 07:57:49	102	17	0	0
50370	2019-06-24 07:57:53	102	17	0	0
50371	2019-06-24 07:57:57	91	17	0	0
50372	2019-06-24 07:58:01	102	17	0	0
50373	2019-06-24 07:58:05	97	17	0	0
50374	2019-06-24 07:58:09	9	17	0	5
50375	2019-06-24 07:58:13	0	17	0	5
50376	2019-06-24 07:58:17	0	24	0	5
50377	2019-06-24 07:58:21	0	36	0	5
50378	2019-06-24 07:58:25	0	48	0	5
50379	2019-06-24 07:58:29	0	60	0	5
50380	2019-06-24 07:58:33	0	75	0	5
50381	2019-06-24 07:58:37	0	86	0	5
50382	2019-06-24 07:58:41	0	100	0	5
50383	2019-06-24 07:58:45	0	100	0	5
50384	2019-06-24 07:58:49	0	100	0	5
50385	2019-06-24 07:58:53	0	100	0	5

Figura 23 : momento in cui viene settata la variabile BladeChange con logica del secondo tentativo

	A	B	C	D	E
1	TIMESTAMP	VELOCITY	BLADE	BLADECHANGE	DOWNTIMETYPE
50292	2019-06-24 07:52:41	48	17	0	0
50293	2019-06-24 07:52:45	68	17	0	0
50294	2019-06-24 07:52:49	101	17	0	0
50295	2019-06-24 07:52:53	102	17	0	0
50296	2019-06-24 07:52:57	21	17	0	0
50297	2019-06-24 07:53:01	9	17	0	5
50298	2019-06-24 07:53:05	0	17	0	5
50299	2019-06-24 07:53:09	0	17	0	5
50300	2019-06-24 07:53:13	0	17	0	5
50301	2019-06-24 07:53:17	0	17	0	5
50302	2019-06-24 07:53:21	0	17	0	5
50303	2019-06-24 07:53:25	0	17	0	5
50304	2019-06-24 07:53:29	0	17	0	5
50305	2019-06-24 07:53:33	0	17	1	5
50306	2019-06-24 07:53:37	0	17	0	5
50307	2019-06-24 07:53:41	0	17	0	5
50308	2019-06-24 07:53:45	0	17	0	5
50309	2019-06-24 07:53:49	0	17	0	5
50310	2019-06-24 07:53:53	0	17	0	5
50311	2019-06-24 07:53:57	0	17	0	5
50312	2019-06-24 07:54:01	0	17	0	5
50313	2019-06-24 07:54:05	0	17	0	5
50314	2019-06-24 07:54:09	0	17	0	5
50315	2019-06-24 07:54:13	0	17	0	5

Figura 14 : momento in cui viene settata la variabile BladeChange con logica del terzo tentativo



Come si vede dalle Figure 21 e 22, utilizzando il vecchio script (secondo tentativo) viene classificato come tempo di downtime dovuto al cambio lama un periodo maggiore (e diverso) rispetto all'utilizzo del nuovo (terzo tentativo) script. In entrambi gli script, il periodo classificato con DOWNTIMETYPE = 5 è il periodo di downtime successivo al settaggio della variabile BLADECHANGE.

Andando ad analizzare in Figura 23 e 24 il file CSV prodotto dagli script, si può vedere come col vecchio script, la variabile BLADECHANGE (ad indicare il rilevamento del cambio lama) viene settata in periodi diversi a seconda dello script. Infatti con lo script vecchio veniva settata alla riga 50365 mentre con lo script nuovo alla riga 50305, ovvero 60 istanti prima (4 minuti prima).

Si ipotizza che questo sia la causa della diminuzione del tempo rilevato di downtime dovuto al cambio lama, poiché nella versione nuova dello script, BLADECHANGE viene settata prima di un altro periodo di downtime (quello che nella Figura 21 è evidenziato in rosso).

Si prova quindi a cambiare la logica con cui si setta la variabile BLADECHANGE nel nuovo script, infatti invece di prendere il primo valore dell'intervallo che rispetta le condizioni, proviamo a prendere l'ultimo.

Per fare ciò, ispirandoci alla logica per trovare il primo istante dell'intervallo, si scopre che per prendere l'ultimo istante di ogni intervallo occorre fare un NOR tra output e shiftedinput.

$$CL = \left\{ \begin{array}{l} \exists t_i \in D: PercBlade(t_{i-5min}) < Beta, PercBlade(t_{i-10min}) < Beta, \\ PercBlade(t_{i+5min}) > Alpha, PercBlade(t_{i+10min}) > Alpha, \\ PercBlade(t_{i+5min}) - PercBlade(t_{i-5min}) > (Alpha - Beta), \\ PercBlade(t_{i+10min}) - PercBlade(t_{i-10min}) > (Alpha - Beta) \end{array} \right\}$$

Formula 6 : Prima parte della formula del quarto tentativo

$$CL = \neg(CL \vee \neg ShiftedCL)$$

Formula 7 : serve a prendere l'ultimo istante di ogni intervallo

Si modifica quindi lo script extractAllVariablesAndClockLS1.m e si analizzano i dati del downtime con la bozza dello script anDownTimeCL.m per trarre conclusioni sulla logica.

Macchina	Downtime Cambio Lama Totale	Nr Cambi Lama	Cambi lama assenti secondo il grafico	Tempo medio
AKTUL.34112	28872	4	0	7218 Troppo
CARDINAL_TISSUE.20360	0	0	0	
CARTINDUSTRIA_VENETA.33071	1260	1	0	1260 Possibile
DAIO.06181	0	0	0	
DAIO.06182	92	1	0	92 Troppo poco
EUROVAST.33123	0	0	0	
FADERCO.34155	14404	1	1	14404 Troppo
FAPSA.8568	2812	1	0	2812 Possibile
FLORYCART.33007	0	0	0	
SUAVECEL.34150	1404	1	0	1404 Possibile
TASK.34096	3132	2	0	1566 Possibile
TASK.34139	16508	4	0	4127 Possibile?
TASK.34186	2428	4	0	607 Possibile
TEDIM.33117	9208	1	0	9208 Troppo

Figura 25: analisi manuale dei dati prodotti da anDownTimeCL con logica del quarto tentativo

Come si vede dalla Figura 25, in una buona parte dei casi, il tempo medio per cambio lama ottenuto dallo script sembra essere un valore possibile e in linea con i tempi effettivi che un operaio impiega ad effettuare il cambio lama (dai 15 ai 30 minuti circa secondo gli esperti).

Emergono però situazioni problematiche opposte.

C'è un caso in cui il valore restituito dallo script `anDownTimeCL` è decisamente troppo basso.

Ci sono altri casi in cui i valori sono decisamente troppo alti.

### 3.5 Quinto tentativo

Si decide quindi di modificare la soglia di Beta (ovvero la soglia minima per cui una lama viene cambiata) alzandola a 50 e inoltre si decide di cambiare istante per il riconoscimento del cambio lama.

In un primo momento infatti avevamo deciso di prendere il primo istante dell'intervallo che rispettasse la logica del cambio lama.

Successivamente avevamo preso l'ultimo, adesso invece si sceglie quello che sta nel mezzo tra primo e ultimo.

$$CL = \left\{ \begin{array}{l} \exists t_i \in D: PercBlade(t_{i-5min}) < Beta, PercBlade(t_{i-10min}) < Beta, \\ PercBlade(t_{i+5min}) > Alpha, PercBlade(t_{i+10min}) > Alpha, \\ PercBlade(t_{i+5min}) - PercBlade(t_{i-5min}) > (Alpha - Beta), \\ PercBlade(t_{i+10min}) - PercBlade(t_{i-10min}) > (Alpha - Beta) \end{array} \right\}$$

*Formula 8 : Prima parte della formula del terzo tentativo (Alpha 75 Beta 50)*

$$First = CL \wedge \neg ShiftedCL$$

*Formula 9 : Primo istante*

$$Last = \neg(CL \vee \neg ShiftedCL)$$

*Formula 10 : Ultimo istante*

$$CL = First + \left( \frac{Last - First}{2} \right)$$

*Formula 11 : Istante che si trova nel mezzo tra primo e ultimo*

Si procede quindi ad una nuova modifica dello script

`extractAllVariablesAndClockLS1.m` e ad una nuova esecuzione dello script `anDowntimeCL` per un'analisi approfondita dei risultati ottenuti.

DATI CON SCRIPT MODIFICATO3 --- PRENDENDO VALORE DI INTERVALLO NEL MEZZO					
Macchina	Downtime Cambio Lama Totale	Nr Cambi Lama	Cambi lama assenti secondo il grafico	Tempo medio	
AKTUL.34112	28872	4	0	7218	Troppo
CARDINAL_TISSUE.20360	0	0	0		
CARTINDUSTRIA_VENETA.33071	1260	1	0	1260	Possibile
DAIO.06181	0	0	0		
DAIO.06182	688	1	0	688	Possibile
EUROVAST.33123	0	0	0		
FADERCO.34155	14404	2	0	7202	Troppo
FAPSA.8568	2812	1	0	2812	Possibile
FLORYCART.33007	0	0	0		
SUAVECEL.34150	1404	1	0	1404	Possibile
TASK.34096	3132	2	0	1566	Possibile
TASK.34139	16508	4	0	4127	Possibile?
TASK.34186	2428	4	0	607	Possibile
TEDIM.33117	9208	1	0	9208	Troppo

*Figura 26 : analisi manuale dei dati prodotti da anDownTimeCL con logica del quinto tentativo*

Nella Figura 26 si vedono i risultati dello script anDownTimeCL analizzati manualmente e possiamo ritenerci soddisfatti del risultato.

Possiamo dire che la ricerca della logica con la quale riconoscere il cambio lama può concludersi qua.

Infatti adesso rimane soltanto il problema dei valori medi troppi alti per ogni cambio lama. Questo problema però non riguarda la logica del cambio lama, ma soltanto la logica dello script anDownTimeCL con il quale si analizzano gli intervalli di downtime dovuti al cambio lama.

Si passa quindi alla fase successiva.



## 4. TERZO PASSO : ANALISI DEGLI INTERVALLI DI DOWNTIME

Riguardo quest'ultimo problema, possiamo dire sicuramente che si tratta di situazioni non comuni, per esempio : se la macchina finisce la lama ma non c'è nessun operaio (perché è finito il turno) che può cambiare la lama, il tempo di downtime attribuito al cambio lama è di varie ore.

Si decide quindi di affrontare questo problema eliminando periodi di downtime superati ad un determinato valore.

Si realizza uno script (chiamato findDtLongerThan.m) che, data una soglia di tempo, analizza per ogni macchina quanti intervalli di Downtime (causati dal cambio lama) superino tale soglia.

Questo script ci sarà utile perché così riusciremo a classificare tali intervalli come Unknown e non sporcheranno l'analisi del tempo medio per downtime causato da cambio lama.

Viene quindi testato lo script findDtLongerThan.m con le soglie a 20, 30, 40, 60, 80 e 100 minuti.

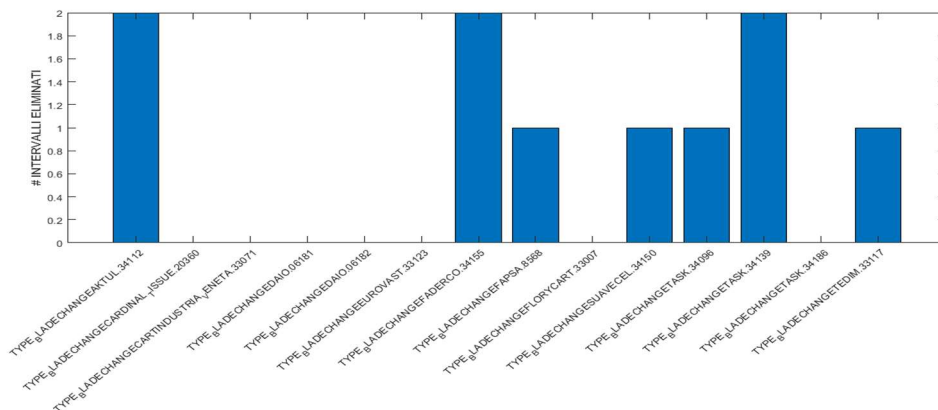


Figura 27: Numero intervalli downtime per cambio lama superiori a 20 minuti

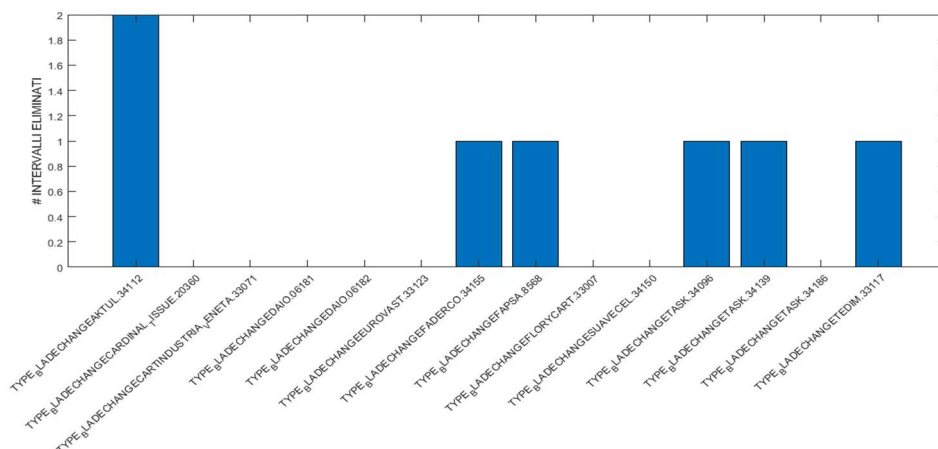


Figura 28: Numero intervalli downtime per cambio lama superiori a 30 minuti

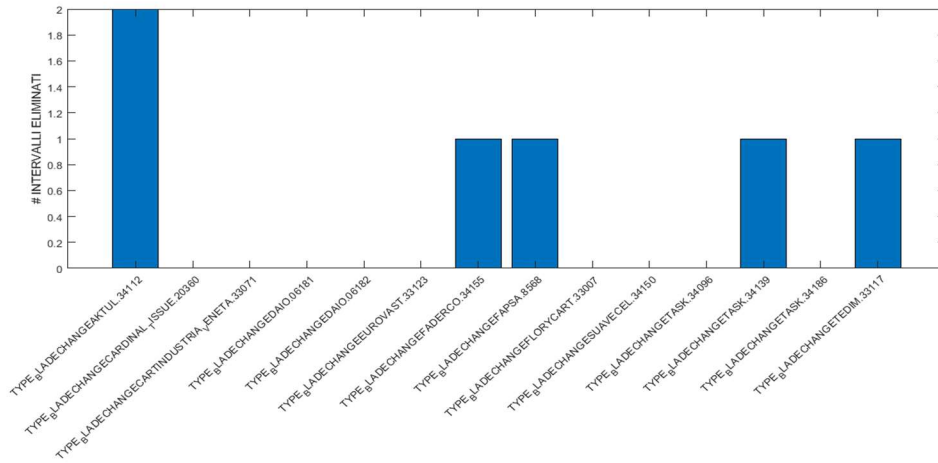


Figura 29: Numero intervalli downtime per cambio lama superiori a 40 minuti

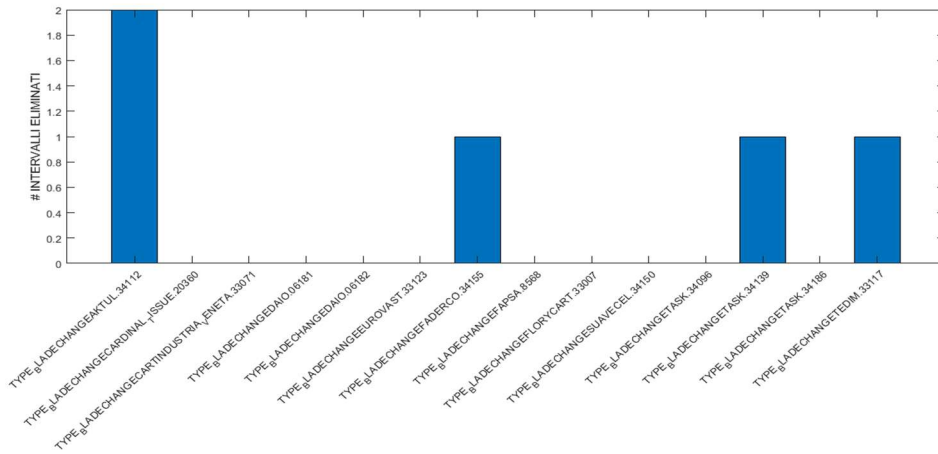


Figura 30: Numero intervalli downtime per cambio lama superiori a 60 minuti

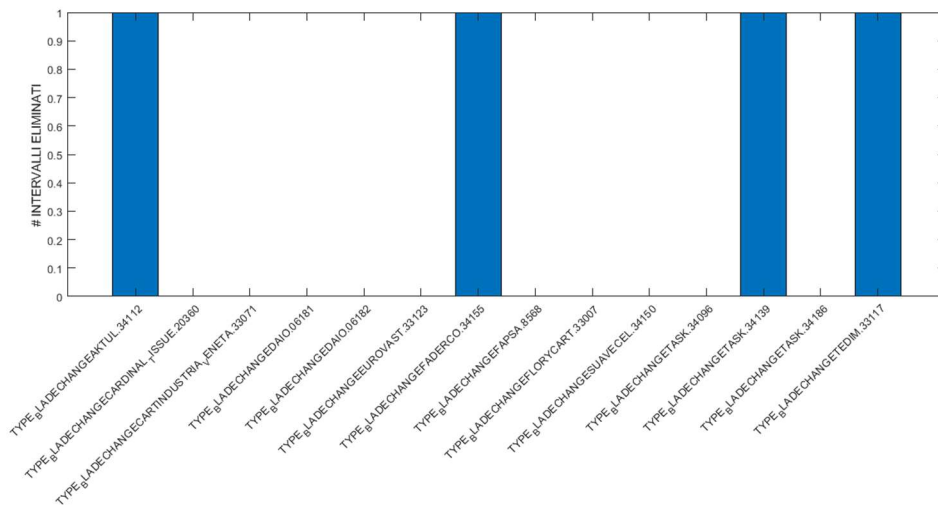


Figura 31: Numero intervalli downtime per cambio lama superiori a 80 minuti

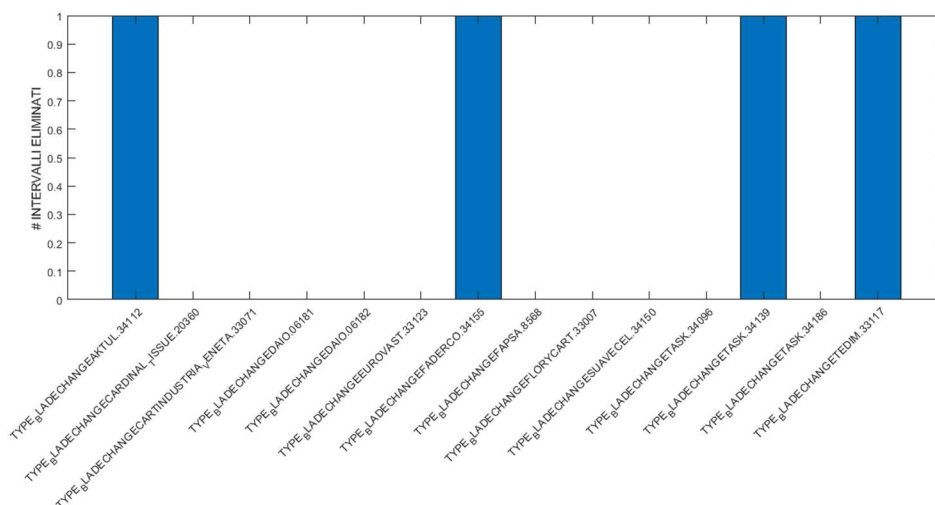


Figura 32: Numero intervalli downtime per cambio lama superiori a 100 minuti

Come si vede dalle Figure da 27 a 32 il numero di periodi eliminati rimane quasi costante dopo i 60 minuti, quindi scegliamo questo valore come soglia.

Una volta scelta la soglia di 60 minuti, si decide di creare un nuovo script partendo da anDownTimeCL e implementando le funzionalità di findDtLongerThan.

Questo nuovo script avrà quindi la funzione di analizzare, per ogni macchina, i seguenti dati :

- Il tempo totale di downtime causato dal cambio lama
- Il tempo medio di ogni downtime per il cambio lama
- Tutto ciò, eliminando quegli intervalli di downtime (causati dal cambio lama) superiori alla soglia scelta (60 minuti in questo caso).

Chiameremo questo script anDownTimeCLThreshold.

La media di ogni downtime per il cambio lama sarà calcolata quindi come segue :

- Prima eliminiamo dai downtime gli intervalli che superano la soglia
- Successivamente calcoliamo quanti di questi intervalli han superato la soglia
- La media sarà quindi  $\text{TempoDowntimeRimanente} / (\text{NumeroCambiLama} - \text{IntervalliTolti})$

DATI CON SCRIPT MODIFICATO 4 --- PRENDENDO VALORE DI INTERVALLO NEL MEZZO --- SENZA INTERVALLI TAGLIATI				
Macchina	Downtime rimanente	Nr Cambi Lama	Periodi Eliminati	
AKTUL.34112	3356	4	2	
CARDINAL_TISSUE.20360	0	0	0	
CARTINDUSTRIA_VENETA.33071	1260	1	0	
DAIO.06181	0	0	0	
DAIO.06182	688	1	0	
EUROVAST.33123	0	0	0	
FADERCO.34155	2136	2	1	
FAPSA.8568	2812	1	0	
FLORYCART.33007	0	0	0	
SUAVECEL.34150	1404	1	0	
TASK.34096	3132	2	0	
TASK.34139	6264	4	1	
TASK.34186	2428	4	0	
TEDIM.33117	852	1	1	

Figura 33: analisi manuale dati prodotti da anDownTimeCLThreshold

In Figura 33 possiamo vedere un'analisi dei risultati prodotti dallo script anDownTimeCLThreshold con soglia a 60 minuti.

Quasi tutti i valori di media rientrano in un intervallo di valori che può essere considerato corretto.

Una situazione anomala è quella della macchina TEDIM, poiché essa ha un cambio lama (ovvero la variabile BLADECHANGE viene settata una volta), viene eliminato un intervallo maggiore di 60 minuti ma rimane comunque un periodo considerato per il cambio lama, quindi viene una divisione per zero.

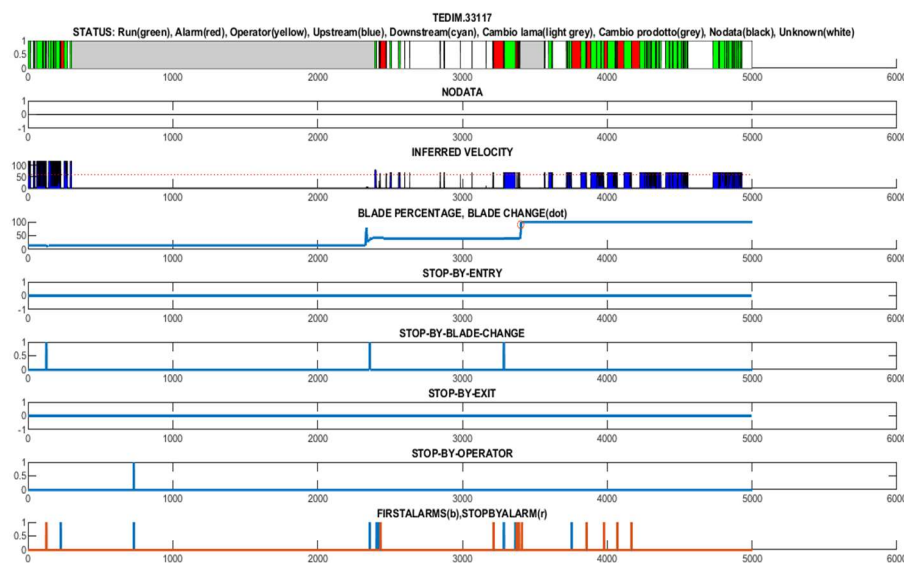


Figura 34: particolare dell'andamento della macchina TEDIM

La causa di ciò sembra essere illustrata in Figura 34.

Infatti il primo periodo Grigio viene eliminato dalla soglia, però esso è causato da un STOP BY OPERATOR, non dal settaggio della variabile BLADECHANGE. Il periodo successivo (che è quello che non viene eliminato) invece è derivato dal settaggio di BLADECHANGE, quindi la sottrazione tra (NumeroCambiLama – IntervalliTolti) non è l'operazione corretta in questo caso particolare.

Si cambia quindi logica con cui si decrementa il numero di cambi lama durante il calcolo della media.

Non si sottrae più quindi il numero di intervalli tolti, ma soltanto il numero di intervalli tolti i quali all'interno contengono un passaggio a livello alto della variabile BLADECHANGE

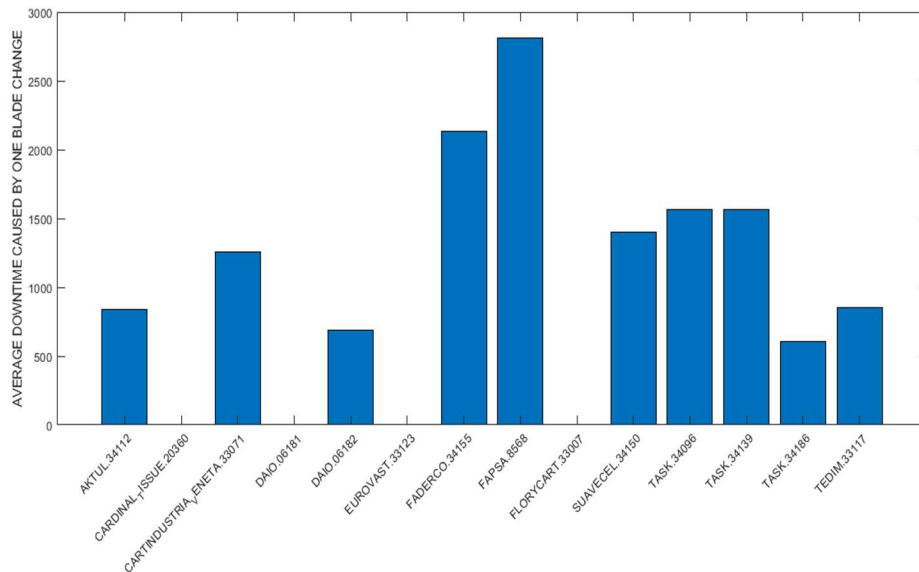


Figura 35: plot prodotto da *anDownTimeCLThreshold* in cui si evidenzia la durata media dei downtime dovuti a cambio lama per ogni macchina

In Figura 35 v'è il risultato grafico prodotto dal nuovo script *anDownTimeCLThreshold* in cui possiamo vedere che tutte le macchine che contengono downtime dovuti al cambio lama hanno valori di downtime medi plausibili.

Macchina	Downtime rimanente	Nr Cambi Lama	Periodi Eliminati	CambiLamaTolti	Tempo medio
AKTUL.34112	3356	4	2	0	839
CARDINAL_TISSUE.20360	0	0	0	0	0
CARTINDUSTRIA_VENETA.33071	1260	1	0	0	1260
DAIO.06181	0	0	0	0	0
DAIO.06182	688	1	0	0	688
EUROVAST.33123	0	0	0	0	0
FADERCO.34155	2136	2	1	1	2136
FAPSA.8568	2812	1	0	0	2812
FLORYCART.33007	0	0	0	0	0
SUAVECEL.34150	1404	1	0	0	1404
TASK.34096	3132	2	0	0	1566
TASK.34139	6264	4	1	0	1566
TASK.34186	2428	4	0	0	607
TEDIM.33117	852	1	1	0	852

Figura 36: analisi manuale dati prodotti dal nuovo script modificato *anDownTimeCLThreshold*

In Figura 36 invece v'è un analisi manuale dei dati ottenuti. Confrontando con la Figura 33 possiamo notare infatti che il numero di intervalli tolti è maggiore del numero di intervalli tolti che contengono un vero BLADECHANGE, come ci aspettavamo.

Si è quindi risolta la situazione anomala della macchina TEDIM.

## 5. DESIGN DELLA CARATTERIZZAZIONE DATA-DRIVEN DEI DOWNTIMES DELLA MACCHINA

Si definiscono 7 stati NON produttivi della macchina. Si elencano in ordine di specificità: DOWNTIME, NODATA, STOP BY ALARM, STOP BY UPSTREAM MACHINE, STOP BY DOWNSTREAM MACHINE, STOP BY OPERATOR, CAMBIO LAMA, CAMBIO PRODOTTO. L'intervallo in cui la macchina è in uno di questi stati può essere individuato mediante la seguente euristica:

- **DOWNTIME LS:** la velocità della macchina è talmente bassa che non può essere in produzione alcun prodotto. Le ragioni di tale malfunzionamento sono ignote. Può essere identificato come quell'intervallo in cui la velocità della macchina  $V_{LS}$  è sempre più bassa di una determinata soglia  $V_{MIN}$ . Tale soglia è certamente più alta della velocità di JOG, micro-avanzamento della macchina usato durante gli interventi manuali e caratterizzato da una velocità di 5 tagli/minuto per ragioni di sicurezza. A seguito di un colloquio con un tecnico della macchina si decide di settare  $V_{MIN}$  a 15 tagli al minuto. Infine, in concomitanza ad un downtime può avvenire il passaggio a zero della variabile STATUS\_RUN.

$$D = \{t_i \in [t_{START}, t_{END}] \forall t_i: V_{LS}(t_i) \leq V_{min} \vee \exists t_i: RUN_{LS}(t_i) = 0\}$$

- **NO DATA:** la ricezione dei dati dalla macchina è compromessa. In condizioni ideali, la macchina invia messaggi di clock ogni 4 secondi circa, al contrario di tutte le altre grandezze che vengono inviate solo al cambiare del loro valore. Questo consente di determinare quando la connessione sia in uno stato funzionante, e solo in questo caso di ricostruire le altre serie temporali applicando una funzione a gradino. Essendo tuttavia non garantito l'arrivo del clock ogni 4 secondi esatti, è necessario determinare un intervallo di tempo (ovvero una "durata minima", maggiore di 4 secondi) entro il quale, se non arriva un segnale di clock possiamo ritenere la macchina in condizione NODATA, condizione in cui non è garantita la correttezza di una ricostruzione delle altre serie temporali mediante applicazione di un segnale a gradino. A seguito di analisi di sensibilità effettuate per lo studio degli stati della ribobinatrice, l'ampiezza dell'intervallo selezionata è di 20 secondi: se non si riceve almeno un segnale di clock in un intervallo di almeno 20 secondi, la macchina si trova in uno stato NODATA

$$.ND = \{t_i \in [t_{START}, t_{END}] \forall t_i: \neg \exists clock_{LS}(t_i), t_{END} - t_{START} \geq 20sec\}$$

- **STOP BY ALARM:** La macchina si è fermata automaticamente perché ha rilevato un problema. Non si hanno informazioni sulla componente

interessata dal malfunzionamento. È possibile individuare tale intervallo come un downtime avvenuto in concomitanza ad almeno una delle seguenti condizioni:

- Il passaggio a 1 di una delle seguenti variabili STATUS\_STOPBYFAULT, STATUS\_FASTOP, STATUS\_STOPBYEMERGENCY
- Variazione della variabile FIRST\_ALARM, ma solo quando la velocità della tagliatrice è superiore a 5 tagli/minuto (diversamente siamo in modalità “jog” e diversi allarmi qui vengono generati dagli interventi dell’operatore durante la risoluzione dei guasti)

$$S_{AL} = \left\{ \begin{array}{l} \exists t_i \in D: FIRST\_ALARM_{LS}(t_i) \neq FIRST\_ALARM_{LS}(t_{i+1}) \vee \\ FASTOP_{LS}(t_i) = 1 \vee \dots \vee STOPBYFAULT_{LS}(t_i) = 1 \end{array} \right\}$$

- **STOP BY OPERATOR:** La macchina è ferma perché l’operatore ha notato un problema e ne ha comandato lo stop. Lo *stop by operator* è un downtime avvenuto in concomitanza a:

- Il passaggio della variabile STATUS\_STOPBYOPERATOR a 1

$$S_{OP} = \{\exists t_i \in D: STOPBYOPERATOR_{LS}(t_i) = 1\}$$

- **STOP BY UPSTREAM MACHINE:** La macchina è ferma perché una delle macchine a monte della troncatrice ha un problema. Lo *stop by upstream machine* è un downtime avvenuto in concomitanza a:

- Il passaggio della variabile STATUS\_STOPBYENTRY a 1

$$S_{US} = \{\exists t_i \in D: STOPBYENTRY_{LS}(t_i) = 1\}$$

- **STOP BY DOWNSTREAM MACHINE:** La macchina è ferma perché una delle macchine a valle della troncatrice ha un problema. Lo *stop by downstream machine* è un downtime avvenuto in concomitanza ad almeno una delle seguenti condizioni:

- Livello dell’accumulatore superiore al 95%
- Il passaggio della variabile STATUS\_STOPBYEXIT a 1

$$S_{DS} = \{\exists t_i \in D: STOPBYEXIT_{LS}(t_i) = 1 \vee PercAcc_{LS}(t_i) > 95\%\}$$

- **CAMBIO LAMA:** la macchina si è fermata per cambiare la lama. E' possibile individuare un cambio lama osservando la percentuale di lama rimanente. Tale grandezza è monitorata mediante sensori e il relativo segnale ha un comportamento a dente di sega. Il cambio lama è il downtime avvenuto in concomitanza alla risalita della percentuale di lama.

Una lama può essere sostituita anche se la sua percentuale non è arrivata a 0. Una nuova lama può essere sostituita può non avere la sua percentuale a 100. Durante l'operazione del cambio di lama, la fotocellula che monitora la percentuale della lama può essere spostata, ciò può causare dei valori non coerenti durante l'operazione.

Per modellare il cambio lama in modo che sia robusto a questi disturbi, si adotta un euristica basata sulle seguenti considerazioni : il "vero" cambio lama è quell'evento che aumenta "Considerevolmente" la percentuale di lama rimanente e questa condizione deve essere vera anche tempo dopo il cambio stesso.

Vengono scelte quindi due soglie, chiamate Alpha e Beta secondo la seguente logica : una lama può essere sostituita solo se inferiore ad una certa percentuale (Beta) e una lama nuova può essere immessa nel macchinario solo se superiore ad una certa percentuale (Alpha). Si modella quindi il cambio lama come il primo istante  $t_i$  tale che  $PercBlade(t_{i-5\text{minuti}}) < Beta$  e tale condizione è contestualmente vera anche per  $t_{i-10\text{minuti}}$ . Inoltre deve valere che  $PercBlade(t_{i+5\text{minuti}}) < Alpha$  e tale condizione è contestualmente vera anche per  $t_{i+10\text{minuti}}$ .

Infine deve anche valere che  $PercBlade(t_{i+5\text{minuti}}) - PercBlade(t_{i-5\text{minuti}}) < Alpha - Beta$  e tale condizione è contestualmente vera anche per  $t_{i+/-10\text{minuti}}$ .

$$CL = \left\{ \begin{array}{l} \exists t_i \in D: PercBlade(t_{i-5mi}) < Beta, PercBlade(t_{i-10m}) < Beta, \\ PercBlade(t_{i+5mi}) > Alpha, PercBlade(t_{i+10m}) > Alpha, \\ PercBlade(t_{i+5mi}) - PercBlade(t_{i-5mi}) > (Alpha - Beta), \\ PercBlade(t_{i+10min}) - PercBlade(t_{i-10min}) > (Alpha - Beta) \end{array} \right\}$$

Questa logica ci ritorna un insieme di intervalli di istanti t. Per comodità, scegliamo il istante intermedio t di ogni intervallo applicando la seguente logica :

- Si trova il primo istante dell'intervallo effettuando uno shift di una posizione del vettore dei valori di CL e facciamo un AND logico tra il vettore originale e l'inverso (NOT) del vettore shiftato.

$$First = CL \wedge \neg ShiftedCL$$



- Si trova l'ultimo istante dell'intervallo effettuando uno shift di una posizione del vettore dei valori di CL e facciamo un NOR logico tra il vettore originale e l'inverso (NOT) del vettore shiftato.

$$Last = \neg(CL \vee \neg ShiftedCL)$$

- L'elemento che sta nel mezzo sarà dato da :

$$CL = First + ((Last - First)/2)$$

- **CAMBIO PRODOTTO:** cambia il tipo di prodotto in lavorazione. Può essere individuato come il downtime avvenuto in concomitanza alla modifica del codice ricetta nei dati. C'è anche la possibilità di modificare la ricetta manualmente, cambiandone i parametri relativi. Un'altra occasione in cui i parametri della ricetta vengono cambiati manualmente sono i micro-aggiustamenti fatti dall'operatore e finalizzati al mantenimento della qualità del prodotto finale. Questa, può variare per motivazioni ambientali (es. umidità). Inoltre questi cambi (il numero e l'entità) variano a seconda dell'esperienza dell'operatore. Ne deriva una intrinseca difficoltà nell'associare una certa variabilità di tali segnali al cambio ricetta piuttosto che ad un micro-aggiustamento. Per questo motivo il cambio prodotto viene modellato come il downtime associato alla modifica esplicita del codice della ricetta e/o alla presenza ad 1 della variabile CHANGE PRODUCTION ON GOING.

$$CP = \{\exists t_i \in D: Recipe_{LS}(t_i) \neq Recipe_{LS}(t_{i+1}) \vee CHANGEONGOINGPRODUCTION_{LS}(t_i) = 1\}$$

## 6. INTRODUZIONE ALLO PSEUDOCODICE

Lo script principale main.m ha lo scopo di chiamare i due principali script creati per lo studio del comportamento della troncatrice.

Chiama quindi in sequenza prima `extractAllVariablesAndClockLS1.m` e poi `anDownTimeCLThreshold.m`

`extractAllVariablesAndClockLS1` viene invocata passandogli come parametri i dati ricevuti dal database della troncatrice, la soglia Beta e la soglia Alpha. Queste due soglie sono parametrizzate e modificabili, tuttavia dopo lo studio del paragrafo 3 è stato deciso che 70 (per Alpha) e 50 (per Beta) sono soglie accettabili) `extractAllVariablesAndClockLS1` estrae dai dati ricevuti come database tutti i parametri che sono utili per lo studio della macchina (velocità, stato lama, i vari allarmi)

Successivamente li inserisce in una struttura dati creata appositamente. Classifica come tempi di downtime tutti quei momenti in cui la velocità è al di sotto di 15 tagli al minuto e come momenti di jog tutti quei momenti dove la velocità è minore di 5 tagli al minuto.

Successivamente chiama la funzione `getBladeChange` (passandogli la struttura appena creata e le due soglie Alpha e Beta) che ha lo scopo di riconoscere e rilevare i momenti in cui è stato effettuato un cambio lama.

La funzione `getBladeChange` segue la logica descritta dall'euristica nel capitolo 5. In seguito, tramite `getDowntimeType` classifica i vari tempi di downtime, anch'esso secondo la logica descritta dall'euristica nel capitolo 5. Importante sottolineare che l'ordine con cui viene riconosciuto il tipo di downtime nel codice (e nello pseudocodice) non è affatto casuale.

Infatti alcune delle condizioni presentate possono verificarsi contemporaneamente nello stesso downtime e tale ambiguità viene risolta stabilendo delle priorità in base alla specificità della causa del downtime:

- Lo STOP BY OPERATOR, STOP BY UPSTREAM MACHINE e lo STOP BY DOWNSTREAM MACHINE avvengono in concomitanza con lo STOP BY ALARM, dato che certamente ci sarà una variazione di FIRST\_ALARM legata a tali occorrenze. Definiamo quindi STOP BY UPSTREAM MACHINE >> STOP BY ALARM e STOP BY DOWNSTREAM MACHINE >> STOP BY ALARM.
- Inoltre ai fini dell'analisi lo stop generato dall'operatore può essere successivo o contestuale a quelli da downstream o upstream, che però sono più specifici, quindi si stabilisce una priorità del tipo STOP BY UPSTREAM MACHINE >> STOP BY OPERATOR e STOP BY DOWNSTREAM MACHINE >> STOP BY OPERATOR.
- Lo STOP BY UPSTREAM MACHINE può inoltre verificarsi in concomitanza col CAMBIO LAMA, poiché tra le macchine a monte della troncatrice c'è

anche lo svolgitoro, interessato dal CAMBIO LAMA. Definiamo quindi CAMBIO LAMA >> STOP BY UPSTREAM MACHINE

- Il CAMBIO PRODOTTO può essere contestuale ad un CAMBIO LAMA, ma non è vero il contrario. Definiamo quindi CAMBIO PRODOTTO >> CAMBIO LAMA.
- Il CAMBIO PRODOTTO e il CAMBIO LAMA hanno la maggiore priorità sugli altri downtime.

Il NODATA ha priorità su tutti gli altri stati in quanto in questo stato, le altre grandezze su cui basare l'euristica per il riconoscimento dello stop della linea non sono attendibili.

Poi utilizza la funzione `replaceIfNoData` per inserire i NODATA dove è necessario. Infine, dopo aver prodotto il grafico dello stato funzionale della troncatura tramite la funzione `Plot`, restituisce `LS.bladeChange` e `LS.downtimetype`, ovvero le variabili che serviranno a `anDownTimeCLThreshold` per ricavare i dati su cui effettuare le sue operazioni.

Lo script `anDownTimeCLThreshold` ha la funzione di classificare come periodi di DOWNTIME UNKNOWN quei periodi di tempo precedentemente classificati come DOWNTIME dovuti a cambio lama ma che superano una certa soglia di tempo.

Questa soglia di tempo, dopo vari test, è stata settata a 60 minuti.

La funzione, dopo aver modificato questi periodi, produce un grafico, macchina per macchina, in cui viene illustrata la differenza tra la durata dei DOWNTIME in condizioni normali (ovvero quando il cambio lama è effettuato da un operatore) e la durata teorica se il cambio lama fosse fatto automaticamente da un upgrade della macchina.

In sintesi funziona come segue :

- Riceve in ingresso due matrici, *bladechange* e *downtimetype*, dove ogni colonna rappresenta l'andamento temporale delle variabili *BLADECHANGE* e *DOWNTIMETYPE*

- Per ogni macchina (quindi per ogni colonna di una matrice) prima trova tutti i periodi causati da cambio lama (tipo = 5)

- In seguito, cerca gli istanti in cui la macchina entra in un intervallo con *downtimetype* uguale a 5 e quando esce da questo periodo. Lo fa attraverso uno studio degli indici (l'array *typecinque* nello pseudocodice) in cui la macchina si trova in questi periodi.

Shiftando infatti l'array in avanti di una posizione, è possibile trovare l'istante in cui inizia un periodo sovraccitato : se nella stessa posizione, l'indice contenuto in *typecinque* è superiore dell'indice contenuto in *shiftedFor + 1*, vuol dire che quell'indice segna l'inizio di un periodo.

Shiftando invece l'array indietro di una posizione, è possibile trovare l'istante in cui finisce un periodo : se nella stessa posizione, l'indice contenuto in *shiftInd* è superiore dell'indice contenuto in *typecinque + 1*, vuol dire che quell'indice segna la fine di un periodo.

- Se un periodo dura solo un istante, è necessario quindi duplicare quell'istante per

classificarlo sia come istante di inizio che come istante di fine intervallo

Per fare ciò è sufficiente sfruttare le due proprietà citate sopra. Un indice è sia di inizio che fine se soddisfa le due proprietà contemporaneamente.

- Una volta trovati gli istanti in cui vi è un cambio, si cercano i relativi indici nella serie temporale.

- L'insieme degli indici è formato a coppie. Gli elementi dispari dell'insieme degli indici totali appena ottenuti è l'insieme degli indici che segnano l'inizio di un intervallo. Gli elementi pari invece segnano la fine di un intervallo.

- Si studiano quindi questi intervalli : se la loro durata (Fine - Inizio) è superiore alla soglia, viene classificato come unknown

- Se all'interno di questo intervallo, vi è un passaggio a livello alto della variabile *BLADECHANGE*, si incrementa un contatore (*contInterEl*) che ci servirà per il calcolo della media. Per vedere se viene settata, basta fare la somma della variabile *BLADECHANGE* dall'indice di inizio all'indice di fine. Se questa somma è uguale a 1, vuol dire che c'è stato un passaggio a livello alto (visto che può assumere solo valori 0 oppure 1)

- La media verrà quindi calcolata come il tempo totale di downtime (ovvero la somma dei periodi con *DOWNTIME* uguale a 5) diviso la differenza tra il numero di cambi lama (ovvero quante volte *BLADECHANGE* è salita a livello alto) e il contatore *contInterEl*

- Il confronto dei tempi di downtime viene effettuato tra il tempo totale calcolato e il tempo totale stimato se fosse presente l'upgrade della macchina per il cambio di lama.

Per calcolare ciò, è sufficiente moltiplicare la differenza tra il numero di cambi lama e *contInterEl* per una durata stimata che ci metterebbe l'upgrade della macchina, ovvero 3 minuti.

- Infine produce i grafici necessari per il confronto tra cambio lama effettuato manualmente e cambio lama effettuato automaticamente

Per semplicità le seguenti funzioni sono dettagliate a parole:

- *getNoData*: applica la logica descritta per il riconoscimento dello stato NODATA. Successivamente alla classificazione dei downtimes questi vengono aggiunti alla Colonna LS.DOWNTIMETYPE mediante la procedura *replaceAllNoData*
- *getDowntimeBoundaries*: torna un insieme di [istanti iniziali - istanti finali] dei downtimes, riconosciuti mediante la logica specificata
- *checkAlarmExistence*: torna gli allarmi generati a velocità superiore a 5 tagli/minuto
- *Plot*: produce un grafico dello stato funzionale della troncatrice e delle grandezze monitorate come mostrato a pagina seguente
- *Bar* : produce due grafici. Il primo mostra un confronto per ogni macchina tra il tempo totale di DOWNTIME dovuto al cambio lama e il tempo stimato in caso di utilizzo di una macchina automatica del cambio lama.

Il secondo mostra la durata media del DOWNTIME dovuta ad ogni cambio lama.

## 7. PSEUDOCODICE

```
Function main(data, Beta, Alpha, Soglia)
    bladeChange, DowntimeType = extractAllVariablesAndClockLS1(data, Beta, Alpha)
    anDownTimeCLThreshold(bladeChange, DowntimeType, Soglia)

end
```

```
Function extractAllVariablesAndClockLS1(data, Beta, Alpha)
    LS = createNewColumns(LS, ["NODATA" "DOWNTIMES" "JOG" "BLADECHANGES"
    "DOWNTIMESTYPE"])
    LS.NODATA = getNoData(LS.STATUS_CLOCK, 20)
    LS.DOWNTIMES = LS.VELOCITY < 15
    LS.JOG = LS.VELOCITY < 5
    LS.BLADECHANGES = geBladeChange(LS, Beta, Alpha)
    [startTime, endTime] = getDowntimeBoundaries(LS.DOWNTIMES, LS.STATUS_RUN)
    For i in 1:length(endTime)
        LS.DOWNTIMESTYPE(startTime(i), endTime(i)) = getDowntimeType(LS,
        startTime(i), endTime(i))
    End
    LS.DOWNTIMESTYPE = replaceIfNoData(LS.DOWNTIMESTYPE, LS.NODATA)
    Plot(LS)
    return LS.bladeChange, LS.DowntimeType
end
```

```
Function getBLADEChange(data, Beta, Alpha)
    inputs = [data.STATISTIC_UNW1_BLADE_ACTPERC]
    output = zeros(size(tmpBLADE))
    currentBLADE = inputs
    if exists(currentBLADE)
        output = shift(currentBlade, -300) < Beta AND
        shift(currentBlade, -600) < Beta AND
        shift(currentBlade, 300) > Alpha AND
        shift(currentBlade, 600) > Alpha AND
        shift(currentBlade, 300) - shift(currentBlade, -300) > (Alpha - Beta)
        AND shift(currentBlade, 600) - shift(currentBlade, -600) > (Alpha - Beta)

    end
    first = output AND !(shift(output, 1))
    last = output NOR !(shift(output, 1))
    output = first + ((last - first) / 2)
    return output
end
```

```

Function getDowntimeType(data, startTime, endTime)
currentData = selectData(data,startTime-seconds(40),endTime)
  if checkValue(currentData.STATUS_CHANGEPRODUCTONGOING,1) OR
  checkExistence(currentData.PRODUCTDATA_ACTUALRECIPE_INDICE)
    downtimeType = "CAMBIO PRODOTTO"
  else
    if checkValue(currentData.BLADECHANGES,1)
      downtimeType = "CAMBIO LAMA"
    else
      if checkValue(currentData.STOPBYEXIT,1) OR
      max(currentData.ACCUMULATOR)>95
        downtimeType = "STOP BY DOWNSTREAM MACHINE"
      else
        if checkValue(currentData.STOPBYMATERIAL,1)
          downtimeType = "STOP BY UPSTREAM MACHINE"
        else
          if checkValue(currentData.STOPOPERATOR,1)
            downtimeType = "STOP BY OPERATOR"
          else
            if checkValue(currentData.STOPBYALARM,1) OR
            checkValue(currentData.STOPBYALARM,1) OR
            checkValue(currentData.STOPBYALARM,1) OR
            checkAlarmExistence(currentData.STATISTIC_FIR
            ST_ALARM, currentData.JOG)
              downtimeType = "STOP BY ALARM";
            end
          end
        end
      end
    end
  end
end
end
end
end
end

```

*Function* anDownTimeCLThreshold(bladechange, downtimetype, soglia)

```
For i in 1:length(bladechange)

    contInterEl = 0
    bChange = bladechange(i,:)
    dtType = downtimetype(i,:)
    typecinque = find(dtType==5)

    If isempty(typecinque)
        continue
    end

    shiftInd = shift(dtType, -1)
    shiftFor = shift(dtType, 1)
    instChange = (typecinque - shiftFor > 1 || shiftInd - typecinque > 1)
    instDoppi = (typecinque - shiftFor > 1 && shiftInd - typecinque > 1)

    indDoppi = typecinque(instDoppi)
    indChange = typecinque(instChange)

    For i2 = 1: length(indDoppi)
        val = find(indChange==indDoppi(i2));
        indChange = [indChange(1:val); indDoppi(i2); indChange(val+1:end)];
    end

    indInizInter = indChange(1:2:end)
    indFineInter = indChange(2:2:end)

    indInizInterNew = indInizInter
    indFineInterNew = indFineInter

    For i3 = 1: length(indInizInter)

        If indFineInter(i3) - indInizInter(i3) > soglia
            indInizInterNew(i3) = indFineInter(i3) - soglia
        end
    end

    For i5 = 1 : length(indInizInterNew)

        If sum(bChange(indInizInterNew(i4):indFineInterNew(i4))) == 1
            contInterEl = contInterEl + 1;
        end

        dtType(indInizInterNew(i4):indFineInterNew(i4)) = 8;
    end

    matVal(i, 1) = contatoreEl;
    matVal(i, 2) = sum(bChange);
    matVal(i, 3) = sum(dtType == 5)
    matVal(i, 4) = matVal(i, 3) / (matVal(i, 2) - matVal(i, 1))
    matVal(i, 5) = (matVal(i, 2) - matVal(i, 1)) * 3

end

bar(matVal)

end
```

## 8. RISULTATI E CONCLUSIONI

Il risultato finale è espresso dalle Figure 37 e 38.

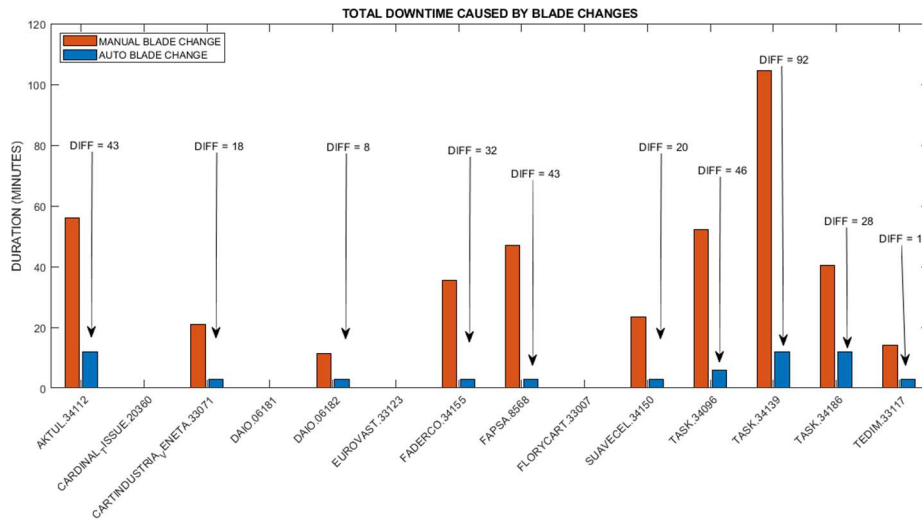


Figura 37: Confronto tra cambio lama manuale e cambio lama automatico

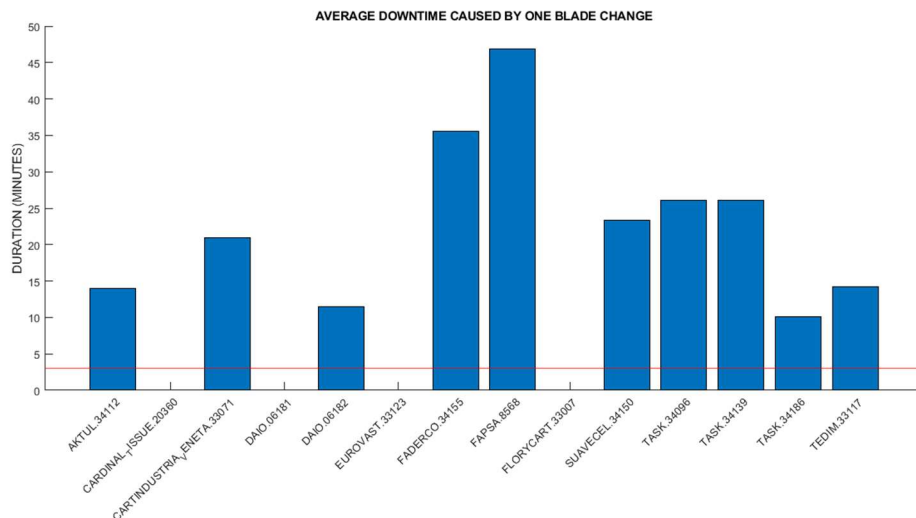


Figura 38: Tempo medio impiegato per ogni macchina per un cambio lama

Nella Figura 37 vi è un confronto tra il tempo totale di downtime causato dal cambio lama manuale e il tempo ipotetico, con lo stesso numero di cambi lama, che ci avrebbe impiegato l'upgrade della macchina con un cambio automatico. Nella Figura 38 invece per ogni macchina è possibile guardare in media quanto tempo di downtime ha causato un cambio lama, dove la riga rossa orizzontale indica il tempo stimato con l'upgrade.

Affiancando questi risultati ad un'analisi economica è possibile stabilire in quanto tempo l'upgrade per il cambio lama automatico è conveniente per una troncatrice.



## **9. SITOGRAFIA**

- Documentazione Matlab

<https://www.mathworks.com/help/matlab/>

## **10. RINGRAZIAMENTI**

Voglio ringraziare il mio relatore, il professor Mario Cimino, il quale mi ha aiutato a scegliere come finire il mio percorso di studi e mi ha aiutato con grandi consigli per il mio futuro.

Voglio ringraziare l'Ing. Luca Alfeo, il suo supporto tecnico e morale (soprattutto quest'ultimo) mi ha permesso di concludere il mio periodo di studi con serenità.

Voglio ringraziare il Sig. Marco Omeri e tutto lo staff del TomorrowLab di Fabio Perini che mi ha fatto sentire a mio agio per tutto il periodo della mia tesi. Sarà difficile trovare in futuro un ambiente come questo.

Ringrazio soprattutto la mia famiglia che mi ha sostenuto e aiutato in ogni scelta (alcune opinabili) durante questi 4 anni.

Infine voglio ringraziare tutti i compagni che hanno condiviso con me quest'avventura :

Nicola, Riccardo, Luca, Andrea, Alessandro, Filippo, Simone, Bill, Niccolò, Claudio e Francesco.