

POLITECNICO DI TORINO

**COLLEGIO DI INGEGNERIA INFORMATICA,
DEL CINEMA E MECCATRONICA**



Tesi di Laurea Magistrale

Digital Factory

**Studio di un processo produttivo e sviluppo di
un'interfaccia software di supervisione**

Relatore

Prof. MICHELE TARAGNA

Co-relatore

Ing. LUCIO BONANDINI

Candidato

DAVIDE LANZA

Dicembre 2020

Abstract

In passato, il mondo dell'industria è stato governato dalla meccanica, dall'elettronica e dall'automazione.

Oggi, si sta imponendo un quarto protagonista, non meno importante: l'informatica. L'utilizzo di supervisori, software gestionali e database, in stretta collaborazione con il processo produttivo, porta ad un'inedita cooperazione tra l'infrastruttura informatica aziendale ed il reparto di produzione.

La grande varietà dei potenziali partner di comunicazione, in uno scenario in cui l'informatica assume un ruolo così importante, rende necessaria l'introduzione di una lingua comune che sia condivisa e compresa da tutti.

Nel 2008, un importante aggiornamento del protocollo OPC da parte di OPC Foundation si propone di risolvere tutte le problematiche di interfacciamento tra i diversi PLC e Motion Controller di campo con i server aziendali.

Nasce così OPC Unified Architecture che, qualche anno più tardi, in contesto Industria 4.0, viene scelto come protocollo standard di comunicazione.

In questa tesi viene presentata un'applicazione sviluppata in **C#** che, installata e configurata su un dispositivo, permette da una parte il collegamento in OPC UA con una macchina e dall'altra la storicizzazione su un database delle variabili desiderate.

Vista la varietà di settori industriali ai quali tale applicazione può risultare utile, è stato scelto un ambito particolarmente interessante per quanto riguarda il controllo dell'avvolgimento: il *Converting*.

Dopo una panoramica del settore, viene presentata nel dettaglio la teoria dell'avvolgimento che, come ogni lavorazione, non è esente da difetti e scarti legati ad una cattiva qualità dell'automazione.

Si apre quindi un'ampia sezione relativa al controllo della tensione, discutendo pregi e difetti delle tre tipologie di controllo più comuni: il controllo in anello aperto, il controllo con retroazione da cella di carico e quello mediante l'utilizzo di un ballerino.

La scelta dell'approccio più adeguato è sempre dettata da un'attenta analisi del processo produttivo, tenendo conto dei costi implementativi, delle necessità e delle caratteristiche della lavorazione.

Le sezioni successive dell'elaborato sono dedicate all'analisi dell'applicazione realizzata e all'introduzione dei concetti chiave dell'Industria 4.0 e del protocollo OPC UA. Dopo una panoramica delle funzionalità principali e dei loro utilizzi, viene descritta tecnicamente l'architettura informatica e gli algoritmi implementativi delle funzionalità introdotte.

Partendo dal controllo di avvolgimento e svolgimento di bobine nel settore del Converting, passando per le innovazioni dell'Industria 4.0 e dell'OPC UA, per concludere con l'applicazione Motion SQL Connector, si può immaginare un caso reale di implementazione di tutte queste tecnologie, combinate per sviluppare una macchina da un lato ottimizzata dal punto di vista dell'automazione e dall'altro conforme ai nuovi dettami dell'Industria 4.0 e integrata con i sistemi informatici aziendali.

Ringraziamenti

Vorrei innanzitutto esprimere la mia gratitudine a Michele Taragna, relatore di questo elaborato, per la sua disponibilità ed i suoi preziosi consigli.

Un ringraziamento speciale va all'azienda Motion Engineering e soprattutto a Lucio Bonandini che, oltre ad essere co-relatore di questa tesi, è anche stata la persona che mi ha dato l'opportunità di scoprire il settore lavorativo in cui voglio crescere.

Colgo questa occasione anche per ringraziare tutte le persone che mi hanno supportato in questo percorso accademico.

Primi fra tutti i miei genitori, che mi hanno accompagnato in questi anni, insegnandomi a raggiungere gli obiettivi che mi ero prefissato, a non procrastinare, a diventare grande.

Grazie a Silvia, mia sorella che nonostante la grande differenza di età, è stata per me grande fonte di ispirazione, aiutandomi a capire chi sarei voluto essere in futuro. A te Alice, al mio fianco in questi anni, che continui a supportarmi nei momenti di difficoltà tirando sempre fuori il meglio di me.

Infine, ringrazio i miei compagni di studio Giovanni e Carmelo che, fin dai primi giorni al Politecnico, hanno reso più leggera le giornate di studio in biblioteca e Riccardo, con il quale ho condiviso la preparazione degli esami della magistrale.

Indice

Elenco delle tabelle	8
Elenco delle figure	9
Acronimi	11
Dal sistema di controllo al Database	13
1 Il Converting	15
1.1 L'industria del Converting	15
1.2 Converting: il modello di macchina	16
1.3 Converting: alcuni esempi	16
1.3.1 Riavvolgimento (<i>Slitter machine</i>)	17
1.3.2 Taglio bobine (<i>Roll Saw machine</i>)	17
1.3.3 Sfogliatura (<i>Sheeter machine</i>)	18
1.3.4 Goffratura (<i>Embossing machine</i>)	19
1.3.5 Rivestimento (<i>Coating machine</i>)	19
1.3.6 Laminazione (<i>Lamination machine</i>)	20
1.3.7 Confezionamento: Flow Pack	21
2 L'avvolgimento	23
2.1 Qualità dell'avvolgimento	23
2.1.1 Durezza dell'avvolgimento e diametro	24
2.1.2 Difetti del rotolo	24
2.2 Principi di avvolgimento	29
2.2.1 Principio della tensione di avvolgimento	29
2.2.2 Principio del Nip	30
2.2.3 Principio della coppia di avvolgimento	31
2.2.4 L'avvolgitore di Pfeiffer e il WIT	31
2.3 Tipologie di avvolgitori (<i>Winder</i>)	33
2.3.1 <i>Winder</i> centrali	34

2.3.2	<i>Winder</i> di superficie	34
2.3.3	<i>Winder</i> ibridi	35
3	Il controllo di tensione	37
3.1	La tensione di un materiale	37
3.2	Relazione tra tensione e diametro	38
3.3	Misura della tensione	39
3.4	Tecniche di controllo	40
3.4.1	Controllo manuale della tensione	41
3.4.2	Controllo in anello aperto: diametro misurato	42
3.4.3	Controllo in anello aperto: diametro calcolato	44
3.4.4	Controllo in anello chiuso: ballerino	46
3.4.5	Controllo in anello chiuso: cella di carico	49
3.4.6	Tecniche ibride di controllo	50
3.4.7	Tecnologie a confronto	51
4	Industria 4.0	53
4.1	Dal passato fino ai giorni nostri	53
4.2	La nuova rivoluzione industriale	54
4.3	Principali innovazioni	54
4.4	Norme Industria 4.0	55
4.5	Il protocollo OPC UA	56
4.5.1	OPC Legacy	56
4.5.2	OPC UA	57
4.5.3	Architettura	58
4.5.4	OPC UA <i>Client</i>	60
4.5.5	OPC UA <i>Server</i>	60
4.5.6	OPC UA – Tecnologia	63
5	Motion SQL Connector	65
5.1	Motivazioni	65
5.2	Scenari d'uso	66
5.3	Panoramica dell'applicazione	67
5.3.1	Layout	68
5.3.2	File di configurazione	70
5.3.3	<i>Trigger</i> di processo	72
5.3.4	<i>Heartbit</i> di segnalazione alla macchina	73
5.3.5	Gestione errori	75
5.3.6	Connessione automatica	77
5.3.7	Gestione timeout di connessione e riconnessione automatica	78
5.3.8	<i>Logging</i> delle variabili monitorate	78

5.3.9	Invio della ricetta remota	78
5.4	Processo e sviluppo dell'applicazione	80
5.4.1	Introduzione	80
5.4.2	Gestione database	80
5.4.3	Avvio dell'applicazione	81
5.4.4	Funzione di connessione	82
5.4.5	Thread Worker in background	83
5.4.6	Caricamento variabili in memoria	83
5.4.7	Funzione di gestione del cambio di valore	84
5.4.8	Gestione del Login	87
6	Conclusione e sviluppi futuri	89
	Bibliografia	93

Elenco delle tabelle

2.1	Moduli elastici di alcuni materiali [12].	30
3.1	Criteri empirici di selezione della tipologia di controllo più adatta [22].	52

Elenco delle figure

1.1	Tipico flusso funzionale di un macchinario per Converting.	16
1.2	Funzionamento di un rullo flessocurvo allargatore. Rielaborazione da [2].	16
1.3	Schema di funzionamento di uno Slitter. Traduzione da [3].	17
1.4	Macchina Roll Saw YYS-II [5].	18
1.5	Componenti principali di una macchina Sheeter. Traduzione da [6].	18
1.6	Esempio di goffratura del metallo. Rielaborazione da [7].	19
1.7	Tre rulli con diverse texture di goffratura [8].	19
1.8	Esempio di Roller Coater e schema di funzionamento [9].	20
1.9	Modello di macchina TM-coat per rivestimento e laminazione a caldo [10].	20
1.10	Struttura di una Flow Pack. Traduzione da [11].	21
2.1	Esempio di anima schiacciata [14].	25
2.2	Possibile conseguenza sul rotolo di Poor Start [15].	26
2.3	Telescoping [16].	27
2.4	Dishing [16].	27
2.5	Carta con evidenti fenomeni di Bagginess [17] e rotolo di carta igienica affetto da Buckling [18].	27
2.6	Sezione laterale di un rotolo con stellatura [19].	28
2.7	Principio della tensione. Traduzione da [12].	29
2.8	Principio del Nip. Traduzione da [12].	31
2.9	Winder di Pfeiffer. Traduzione da [20].	32
2.10	Tipologie di avvolgitori. Traduzione da [21].	33
3.1	Esempi di materiale composto da due fogli accostati a tensioni differenti [16].	38
3.2	Schema delle principali zone di controllo del tiro. Rielaborazione da [16].	38
3.3	Schema strutturale di un'implementazione con cella di carico e una con ballerino. Rielaborazioni da [16].	40

3.4	Esempio di controllo manuale di tensione. Rielaborazione da [16]. . .	41
3.5	Controllo in anello aperto di tensione di tipo <i>Follower Arm</i> : il primo implementato con un braccio <i>pivot</i> e il secondo con un <i>Lay-On Roll</i> . Rielaborazioni da [16].	43
3.6	Controllo in anello aperto di tensione con sensore ad ultrasuoni. Rielaborazione da [16].	43
3.7	Controllo in anello aperto di tensione con calcolo del diametro. Rielaborazione da [16].	45
3.8	Esempio di controllo con ballerino a braccio rotante. Rielaborazione da [16].	48
3.9	Esempio di controllo con ballerino lineare. Rielaborazione da [16]. . .	48
3.10	Esempio di controllo con ballerino rotativo. Rielaborazione da [16].	49
3.11	Esempio di controllo con cella di carico. Rielaborazione da [16]. . .	50
3.12	Esempio di controllo ibrido ad anello chiuso e aperto con sensore a ultrasuoni e ballerino rotativo. Rielaborazione da [16].	51
4.1	Le quattro rivoluzioni industriali [24].	53
4.2	Grafico che mostra il numero di dispositivi connessi nel corso degli anni, espresso in miliardi [26].	55
4.3	Il protocollo OPC UA integrato con l'organizzazione aziendale [28].	57
4.4	Possibile modello architetturale del protocollo OPC UA [29].	59
4.5	Architettura di un OPC UA <i>client</i> [29].	60
4.6	Architettura di un OPC UA <i>server</i> [29].	61
4.7	Rappresentazione della tecnologia del protocollo OPC UA [25]. . . .	63
5.1	Layout dell'applicazione.	70
5.2	Lista delle variabili monitorate.	70
5.3	Porzione di un esempio di file di configurazione.	72
5.4	Cattura di due variabili <i>trigger</i> durante l'esecuzione.	73
5.5	Schema di funzionamento dell' <i>heartbeat</i> tra applicazione e macchina.	74
5.6	Esempio di esecuzione senza alcun errore e visualizzazione della ricetta da produrre.	75
5.7	Popup di errore associato al codice -1.	76
5.8	Popup di errore associato al codice -2.	76
5.9	Popup di errore associato al codice -3.	77
5.10	Esempio di ricetta comandata da remoto corredata di alcuni parametri utili.	79
5.11	Flowchart delle fasi di avvio dell'applicazione.	81
5.12	Flowchart della gestione del cambio valore.	85

Acronimi

SCADA

Supervisory control and data acquisition

PLC

Programmable logic controller

OPC UA

Open platform communications unified architecture

TNT

Tension nip torque

WIT

Wound-in tension

IIoT

Industrial internet of things

IEC

International electrotechnical commission

API

Application programming interface

IT

Information technology

HMI

Human-machine interface

Introduzione

Dal sistema di controllo al Database

Il controllo dell'avvolgimento e dello svolgimento di bobine di materiale è uno degli aspetti più importanti dell'automazione di macchine per la lavorazione di materiale sotto forma di film. Il settore industriale nel quale si raggruppano tutti questi processi produttivi in cui si elaborano film di diverso materiale per ottenere un prodotto finito è il Converting.

Fanno parte di questo settore la produzione di film plastici, la stampa di notiziari e parte del settore tessile. A differenza di altri settori industriali, l'automazione può variare sensibilmente da una macchina all'altra. Differenze di materiale e caratteristiche di processo spesso obbligano i progettisti a scelte progettuali molto diverse tra loro, in termini di struttura della macchina e costi.

Spesso, non esiste il sistema perfetto per ogni tipologia di avvolgitore, in quanto il miglior risultato è determinato da una buona scelta del sistema di controllo in seguito ad un'accurata analisi delle variabili in gioco della particolare applicazione. Durante l'avvolgimento è infatti importante eliminare disturbi legati alla conformazione del materiale e alla tipologia di controllo realizzato.

Verranno quindi presentate diverse tecniche di controllo partendo dalle più semplici in anello aperto, fino alle tecniche più raffinate in anello chiuso, per poi prestare particolare attenzione ai controlli mediante ballerino. A prescindere dalla tecnologia usata, lo scopo comune è il controllo della tensione del materiale, andando a determinare così la qualità del prodotto finito. L'obiettivo della prima parte del presente elaborato è quindi un'analisi dettagliata delle varie tipologie di avvolgimento, analizzando vantaggi e svantaggi di ognuna e concentrandosi su alcune di esse.

Dopo aver dettagliato tutte le tecnologie di controllo viene spostata l'attenzione

sulla parte più informatizzata del processo produttivo. Nel passato il mondo dell'industria è stato governato dalla meccanica, dall'elettronica e dall'automazione. Oggi si sta imponendo un quarto protagonista, non meno importante: l'informatica. Sono sempre più comuni, infatti, supervisor SCADA e in generale sistemi informativi in cooperazione con i PLC o i Motion Controller. Grazie all'integrazione tra hardware di controllo e sistemi di supervisione è possibile ottenere sistemi utili a monitorare le prestazioni e l'efficienza del processo. Sempre più spesso ci sono motivazioni aziendali per interconnettere le macchine tra loro e con i livelli superiori delle aziende, come software gestionali e sistemi informativi. Sull'onda di queste richieste, in Germania nel 2010 si è cominciato a parlare di Industria 4.0.

Dopo un'introduzione ai nuovi standard dettati da questa ultima "rivoluzione industriale", viene trattato nel dettaglio il protocollo principe di questa innovazione: l'OPC UA. Grazie a questo protocollo è infatti possibile instaurare delle connessioni tra il campo e lo strato informativo superiore.

Dopo un'analisi del modello Client-Server dell'OPC UA, verrà descritta un'applicazione, realizzata in **C#** proprio per rispondere a queste nuove richieste aziendali. L'applicazione sarà quindi in grado di collegare macchine controllate da un Siemens Simotion D, configurato come server OPC UA, con un database aziendale. L'Industria 4.0 prevede uno scambio bilaterale di informazioni tra la macchina e il gestionale. Grazie a questa applicazione è possibile storicizzare tutte le variabili desiderate su un database. Inoltre, è possibile impartire il "cambio ricetta" alla macchina modificando sul database la successiva ricetta da produrre, soddisfacendo così anche lo scambio dati in direzione opposta.

Partendo quindi dal controllo di avvolgimento e svolgimento di bobine nel settore Converting, passando per le innovazioni dell'Industria 4.0 e dell'OPC UA, per concludere con l'applicazione OPC UA client sviluppata in **C#**, si può immaginare un caso reale di implementazione di tutte queste tecnologie, combinate per sviluppare una macchina da un lato ottimizzata dal punto di vista dell'automazione e dall'altro conforme ai nuovi dettami dell'Industria 4.0 e integrata con i sistemi informatici aziendali.

Capitolo 1

Il Converting

1.1 L'industria del Converting

Tutte le industrie che producono bobine di prodotti lavorati fanno parte del settore del Converting. Rientrano in questo settore i processi produttivi nei quali vengono avvolti materiali in film che, essendo in generale molto fini, sono suscettibili alle variazioni di tensione, causa di difetti o nei casi peggiori di rotture.

I materiali più comuni per questo settore industriale sono carta, tessuto, alluminio o diverse materie plastiche trattate a nastro. I principali settori, per questo tipo di lavorazioni, sono il tessile, l'editoriale e l'imballaggio.

Le principali lavorazioni del settore sono:

- il rivestimento, che molto spesso è adesivo, per la produzione di etichette e nastri
- il taglio, per ottenere un prodotto finito o un passaggio intermedio delle dimensioni desiderate
- la laminazione, utile per una finitura superficiale, solitamente eseguita a caldo
- la stampa, che può avvenire come deposizione di inchiostro oppure per impressione, in questo caso si parla di goffatura

Nella maggior parte delle situazioni produttive, l'obiettivo principale del Converting è quello di lavorare il film mentre viene trasportato lungo la linea. Non mancano però esempi di puro riavvolgimento, come nel caso del *Roll Slitting* nel quale una grossa bobina viene riavvolta su bobine di dimensioni inferiori.

1.2 Converting: il modello di macchina

Le macchine per il Converting, pur essendo parecchie e diverse tra loro, sono tutte riconducibili a un modello ben preciso. All'inizio del processo produttivo di quasi tutte le tipologie vi è infatti uno svolgitore, adibito all'introduzione del materiale mediante un controllo di tensione più o meno complesso. A valle delle operazioni di svolgimento e tensionamento è presente la lavorazione vera e propria del materiale, seguita infine dall'avvolgimento dello stesso su una nuova bobina. Spesso anche nelle fasi finali di avvolgimento è presente un controllo di tiro.

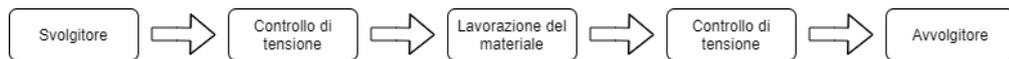


Figura 1.1: Tipico flusso funzionale di un macchinario per Converting.

Solitamente, una macchina di questo genere presenta un traino che stabilisce la velocità di avanzamento del film e diversi altri rulli, motorizzati e non, adibiti a diverse funzionalità.

Sono presenti, in gran numero, rulli folli che servono semplicemente per guidare il materiale e non farlo deviare dalla direzione opportuna. Utili anche i guidabanda per cambiare direzione al velo e i flessocurvi allargatori che, stirando il materiale in senso ortogonale alla direzione, evitano la formazione di pieghe.

Sono presenti anche rulli per la lavorazione vera e propria del materiale come per la stampa o per il taglio. Abbiamo infine dei rulli utili al controllo di tensione del materiale come il ballerino. [1]

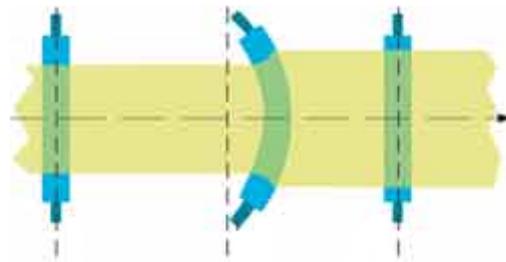


Figura 1.2: Funzionamento di un rullo flessocurvo allargatore. Rielaborazione da [2].

1.3 Converting: alcuni esempi

L'industria del Converting copre una vasta gamma di settori merceologici, vengono quindi introdotti alcuni dei macchinari più comuni per la produzione di giornali,

riviste, carta igienica e di materiali per il mondo del packaging, come varie tipologie di imballaggi e prodotti plastici.

Tutte queste tipologie di macchine sono prodotte da diverse aziende ma in questa sede viene data solamente una panoramica sul modello specifico di macchina.

Sono quindi elencate alcune delle operazioni più comuni e le relative macchine adibite a quel particolare compito.

1.3.1 Riavvolgimento (*Slitter machine*)

Si tratta di uno dei macchinari più utilizzati in tutta l'industria del Converting, utile per svolgere una bobina di grandi dimensioni su più bobine della misura desiderata. Operazioni di questo genere possono essere utili come passo intermedio di lavorazione oppure come ultima lavorazione prima dello stoccaggio del prodotto finito. Il materiale viene svolto, tagliato e riavvolto; sono spesso presenti dei rulli di riciclo per recuperare il materiale di scarto a valle della stazione di taglio.

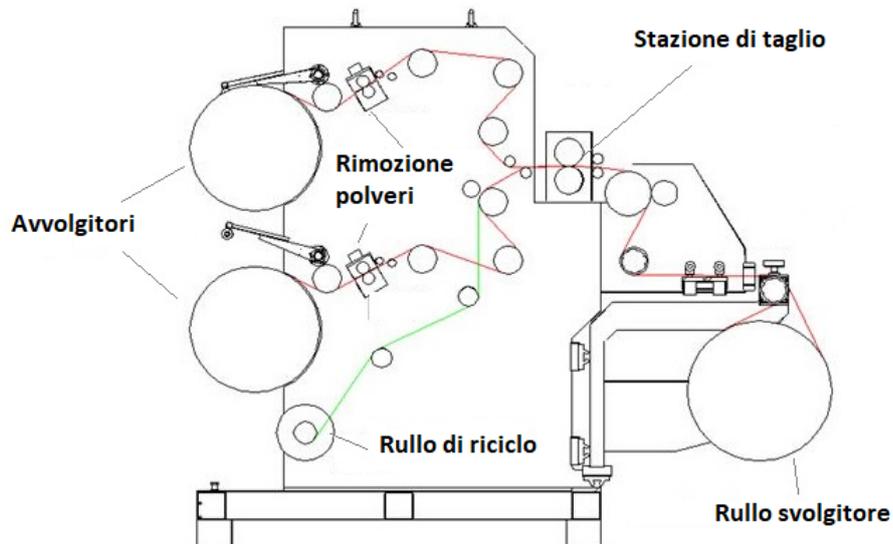


Figura 1.3: Schema di funzionamento di uno Slitter. Traduzione da [3].

1.3.2 Taglio bobine (*Roll Saw machine*)

Nei primi anni 2000 è stata introdotta questa nuova tipologia di macchina, in particolare nell'industria della carta. La funzione finale è simile a quella dello Slitter, con la differenza che invece di svolgere e riavvolgere il materiale, esso viene tagliato direttamente sotto forma di rotolo. È presente infatti una sega che taglia di netto il rullo avvolto in direzione ortogonale al diametro: si hanno quindi più

bobine di ugual diametro ma di dimensioni ridotte. Questa strategia ha prestazioni fino a quattro volte superiori in termini di velocità di processo rispetto a uno slitter standard con lo svantaggio che segnando le bobine vengono introdotte possibili imperfezioni sui bordi. [4]



Figura 1.4: Macchina Roll Saw YYS-II [5].

1.3.3 Sfogliatura (*Sheeter machine*)

Le macchine sfogliatrici sono adibite alla trasformazione di rotoli di materiale in fogli o lastre di dimensione specifica. Solitamente, dopo aver ottenuto le lastre, una ad una vengono riposte in una pila, questa operazione richiede estrema precisione nei movimenti e nel taglio. [4]

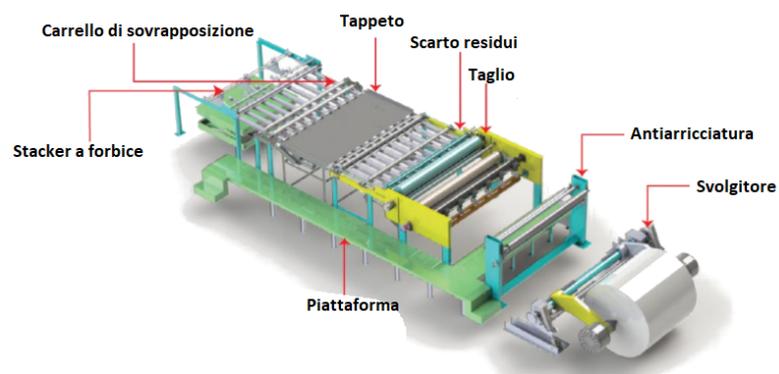


Figura 1.5: Componenti principali di una macchina Sheeter. Traduzione da [6].

1.3.4 Goffratura (*Embossing machine*)

Questa tipologia di macchine imprime dei pattern su carta o su tessuto non tessuto; principale esempio di utilizzo di questa tecnologia è la carta igienica che infatti presenta quasi sempre un motivo decorativo impresso.

Altro esempio di impiego della goffratura sono i rotoli da cucina. In generale non è solo un miglioramento estetico ma, come in questo caso, viene favorito l'assorbimento dei liquidi.

Il numero di prodotti finiti che hanno subito questo tipo di lavorazione è decisamente superiore alle aspettative: cinture, scarpe, contenitori per bevande in alluminio e materiali plastici sono solo pochi esempi di quanto il settore sia eterogeneo. Attraverso questa tecnica possono anche essere creati dei tessuti formati da più strati, il processo di incollaggio dei bordi è noto come *ply-bonding*. [4]

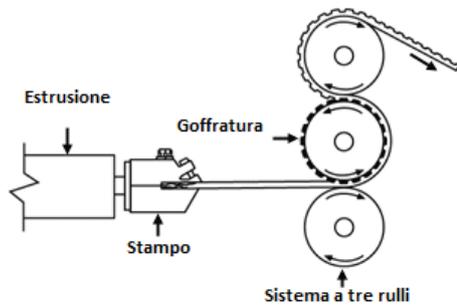


Figura 1.6: Esempio di goffratura del metallo. Rielaborazione da [7].



Figura 1.7: Tre rulli con diverse texture di goffratura [8].

1.3.5 Rivestimento (*Coating machine*)

Con l'uso di questi macchinari viene applicato del materiale liquido, o gassoso, come rivestimento per uno o entrambi i lati del materiale prodotto. La difficoltà nella realizzazione dell'automazione di questi macchinari sta nel distribuire il rivestimento su tutta la superficie e in modo uniforme. Questo obiettivo è raggiunto regolando opportunamente il controllo di velocità della linea.

Spesso i rivestimenti possono essere colla e silicone, oppure prodotti antistatici o resistenti ai raggi ultravioletti: per questi ultimi tipi di rivestimento potrebbe essere necessario del tempo dopo l'applicazione per l'asciugatura o l'indurimento. Nell'industria del legno non è raro l'utilizzo di macchine per rivestire i fogli con olio o lacca.



Figura 1.8: Esempio di Roller Coater e schema di funzionamento [9].

1.3.6 Laminazione (*Lamination machine*)

La laminazione viene applicata a diversi tipi di materiale per ottenere un prodotto dello spessore desiderato. Il materiale viene fatto passare all'interno di una coppia di rulli caldi con distanza tra loro minore dello spessore del film. Il risultato finale è un prodotto dello spessore desiderato e uniforme.

La laminazione può anche essere operata a freddo, in questo caso la lavorazione dona una maggior resistenza meccanica al materiale, migliorando la sua superficie. La differenza principale con la laminazione a caldo è l'assenza di ossidazione del materiale; questo porta quindi a una migliore qualità superficiale del prodotto finale. [4]



Figura 1.9: Modello di macchina TM-coat per rivestimento e laminazione a caldo [10].

1.3.7 Confezionamento: Flow Pack

Un'ampia gamma di prodotti che spazia dal cibo al settore medicale vengono imbustati attraverso una tecnologia a moto continuo orizzontale.[11] Il film viene infatti srotolato durante il passaggio del materiale da imbustare che, senza alcuna fermata, viene confezionato. Il modello della Flow Pack è uno dei più utilizzati.

Il materiale viene piazzato su un nastro trasportatore, comunemente chiamato *Conveyor*, per essere portato verso una zona di piegamento del film. Viene infatti utilizzato un singolo strato di film che, srotolato da un cilindro sul quale è avvolto, viene guidato verso la zona di piegamento, chiamata *Former*, che gli donerà una forma tubolare in modo da avvolgere il prodotto. Appena dopo il piegamento, i due lembi di film vengono riposti adiacenti tra loro e saldati al di sotto del centro del prodotto. Una volta che la saldatura longitudinale è completa, il prodotto viene trasportato verso una ganascia di taglio e saldatura rotante. Durante il taglio tra un prodotto e l'altro avviene anche la saldatura contemporanea che implica la chiusura del retro del prodotto precedente e del fronte del prodotto successivo. Grazie a un ulteriore nastro orizzontale il prodotto finito viene trasportato per essere eventualmente riposto automaticamente in scatole o altri contenitori. Tutte queste operazioni sono eseguite in un moto continuo del materiale e del film, questo implica notevoli vantaggi nell'uso di questi macchinari in termini di tempo ed efficienza. [11]

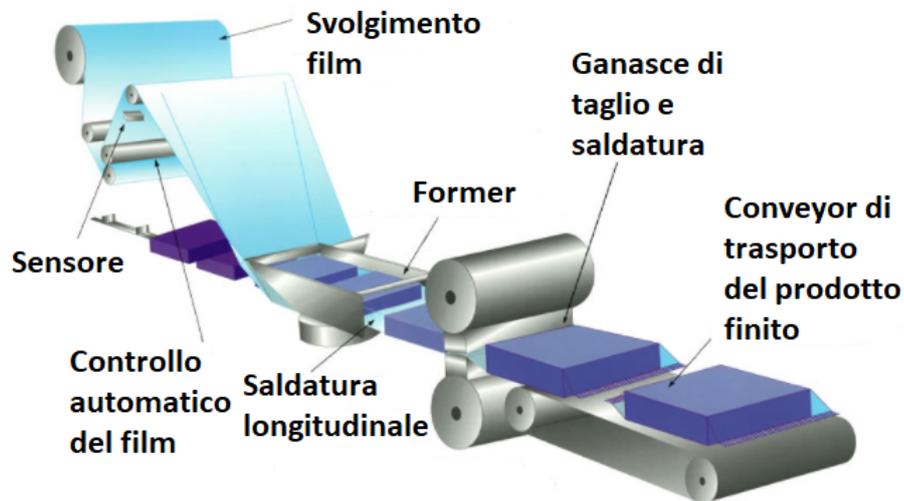


Figura 1.10: Struttura di una Flow Pack. Traduzione da [11].

Capitolo 2

L'avvolgimento

2.1 Qualità dell'avvolgimento

Un avvolgimento si può definire "buono" nel momento in cui le bobine sono della forma corretta, del giusto diametro e della corretta consistenza. La densità della bobina e la tensione sono i fattori che caratterizzano la bontà dell'avvolgimento.

Una bobina avvolta con tensioni esigue risulterà in una forma non cilindrica oppure verrà deformata una volta riposta in magazzino. Difetti di questo tipo hanno diversi problemi connessi: nel momento in cui una bobina non cilindrica venga svolta, produrrà delle oscillazioni di tensione che possono rovinare il materiale e, nel caso in cui ci sia un'unità di stampa a valle, incidere sulla precisione del processo. Anche bobine avvolte con tensioni troppo elevate non causano meno problemi. Il principale problema legato ad avvolgimenti troppo vigorosi è il bloccaggio del materiale; diversi tipi di film infatti, se avvolti con troppa forza, aderiscono agli strati inferiori fino a fondersi. Non è raro che, dopo avvolgimenti di questo tipo, collassi il centro della bobina creando così diversi problemi nella rimozione dell'albero dalla stessa. Inoltre, avvolgimenti troppo intensi possono peggiorare i difetti già presenti nel film. Tipicamente, il materiale avvolto presenta diverso spessore in diversi punti risultando quindi non perfettamente uniforme. Nel momento dell'avvolgimento, le varie zone spesse e più sottili vengono sommate tra di loro. Se l'avvolgimento risulta troppo saldo verrà deformato il materiale in prossimità delle zone più spesse. Questo crea un prodotto che, una volta svolto, avrà un difetto chiamato *Bagginess*, proprio per la presenza di queste zone ondulate. Difetti di questo tipo possono essere evitati aggiungendo aria mentre si avvolgono zone sottili e riducendo la tensione di avvolgimento per le zone più spesse.

L'avvolgimento di materiali elastici, durante varie fasi di processo, presenta ancora più problematiche. Questi, infatti, creano variazioni di spessore talmente grandi, tra una lavorazione e l'altra, da non essere controllabili semplicemente variando la

tensione di avvolgimento. Per cercare di contrastare questi effetti il materiale, o i rulli avvolgitori, vengono mossi avanti e indietro creando un'oscillazione. Questo movimento, che nelle applicazioni più ottimizzate può essere di tipo sinusoidale, aiuta a rendere casuale la distribuzione degli eventuali difetti presenti. Bisogna prestare però attenzione che la velocità di oscillazione sia abbastanza veloce da distribuirli uniformemente ma senza essere eccessiva e rovinare il materiale. [12]

2.1.1 Durezza dell'avvolgimento e diametro

L'avvolgimento di un rotolo di materiale genera al suo interno tensioni sempre più grandi, note come stress residuo. Queste, se troppo elevate, espongono il centro della bobina a carichi di compressione così elevati da causare un difetto del materiale, noto come *Buckling*, caratterizzato da zone corrugate e deformate. Se il materiale avvolto non gode di buona elasticità, gli strati più centrali potrebbero perdere tensione e allentarsi. Questo fenomeno causa un effetto indesiderato noto come *Telescoping*, contraddistinto da un progressivo disallineamento dei bordi, visibile durante lo svolgimento. Per diminuire questo difetto è possibile controllare la tensione di avvolgimento in modo da diminuirla mano a mano che il diametro aumenta: è possibile quindi delineare una relazione tra il diametro del rotolo e la relativa diminuzione della durezza dell'avvolgimento. [12]

2.1.2 Difetti del rotolo

Un controllo di tensione adeguato durante l'avvolgimento e lo svolgimento è la chiave per ottenere dei rotoli di materiale opportunamente formati. Spesso, infatti, i principali difetti del prodotto finito sono dovuti ad avvolgimenti troppo intensi o poco precisi.

I difetti che caratterizzano un rotolo non si limitano a peggiorare l'estetica del prodotto ma ne inficiano il processo produttivo e, nei casi più gravi, compromettono l'integrità del materiale creando scarto.

Segue una panoramica dei principali difetti di avvolgimento e svolgimento con le relative cause e conseguenze.[13]

Danni al rotolo

Questi difetti sono i più visibili, in quanto sono causati da eventi posteriori alla creazione della bobina. Anche se non comuni, si possono verificare tagli superficiali visibili alla rimozione dell'imballaggio. Le cause di questi difetti sono ricercabili nel processo a valle dell'avvolgimento, in generale nella filiera per il trasporto dei rotoli.

Difetti del *core*

Sono problemi legati alla zona centrale o anima del rotolo avvolto. Questa zona infatti è quella più esposta a errori di controllo della tensione di avvolgimento subendo i carichi di compressione più elevati. Possiamo ricondurre questa tipologia di difetti a tre principali fenomeni. [13]

1. *Crushed core*: uno dei difetti più comuni, causato da un avvolgimento con tensione eccessiva. L'albero su cui è avvolto il materiale collassa a causa dell'eccessiva forza di compressione agente su di esso. Utilizzare un materiale più resistente alle compressioni oppure iniziare l'avvolgimento con tensione minore potrebbero essere semplici soluzioni al problema.
2. *Loose core*: causato da una differenza di velocità di rotazione tra l'asse e il materiale avvolto. Questo potrebbe causare uno scivolamento del materiale dall'albero che creerebbe inevitabilmente problemi ai processi a valle. La causa di questo difetto è generata da un avvolgimento troppo debole nella fasi iniziali.
3. *Offset core*: l'albero sul quale viene avvolto il materiale non è allineato con i bordi del materiale. La causa di questo problema è da ricercare in un allineamento tra l'asse e il film poco preciso.



Figura 2.1: Esempio di anima schiacciata [14].

Poor Start

Il materiale presenta notevoli differenze visive tra le parti più interne e vicine al cilindro e quelle più esterne. Il difetto può essere prodotto da diverse problematiche relative alle prime fasi di avvolgimento. Un materiale corrugato o allentato prima di essere avvolto può favorire questa anomalia. Inoltre, può succedere che il film oscilli all'entrata sul cilindro, questo può essere evitato aumentando la tensione di arrotolamento. [15]

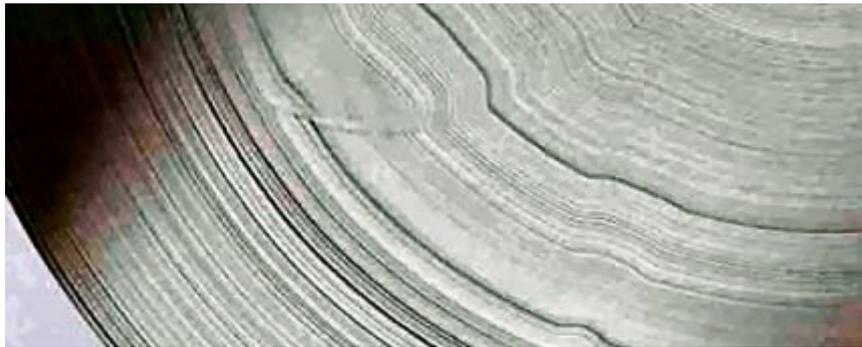


Figura 2.2: Possibile conseguenza sul rotolo di Poor Start [15].

Tagli superficiali

La presenza di tagli sulla superficie del film, soprattutto in direzione di marcia della macchina è spesso causata da zone in cui il materiale si accumula a causa di un avvolgimento troppo debole. Nel momento in cui l'eccesso di materiale raggiunge un punto stretto di passaggio, come potrebbero essere due calandre, la bolla di materiale scoppia lacerandolo. Oltre all'incisione sono a volte presenti zone traslucide, causate da un eccessivo stiramento.

Dishing e Telescoping

Come già anticipato, quando l'albero non è perfettamente allineato con il materiale in entrata, si creano dei disallineamenti concavi o convessi. Se il difetto risulta facilmente visibile su un rotolo prodotto si parla di Dishing, se invece l'anomalia si nota solo durante lo svolgimento dello stesso indica Telescoping. Può capitare che l'albero avvolgitore non sia ben saldo durante il processo e subisca delle vibrazioni che potrebbero muovere la posizione relativa tra asse e materiale.

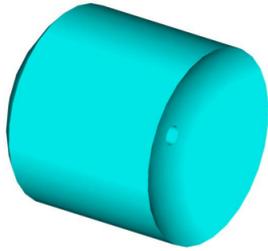


Figura 2.3: Telescoping [16].

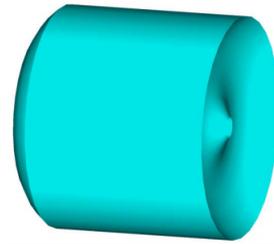


Figura 2.4: Dishing [16].

Bagginess e Buckling

Nessun materiale è perfettamente uniforme in termini di spessore. Spesso infatti, per varie motivazioni, sono presenti zone di materiale con spessore maggiore rispetto ad altre. Questo, associato ad un avvolgimento con tensioni ingenti, crea un difetto noto come *Bagginess*, il materiale infatti risulta ondulato a causa dello stiramento in corrispondenza delle zone più spesse, accumulate l'una sull'altra. Altra imperfezione, causata da un avvolgimento troppo vigoroso, è il *Buckling*, caratterizzato da zone corrugate e causato dall'eccessiva pressione subita dagli strati più interni. [13]



Figura 2.5: Carta con evidenti fenomeni di Bagginess [17] e rotolo di carta igienica affetto da Buckling [18].

Starring

Il lato del rotolo può essere caratterizzato da deformazioni a forma di stella. Questo è dovuto a uno slittamento relativo degli strati di materiale, causato da un aumento di tensione di avvolgimento rispetto agli strati più interni. Una possibile soluzione è quella di operare in modo opposto: iniziare l'avvolgimento con una tensione maggiore e, mano a mano che il diametro aumenta, diminuirla. A volte invece la causa è da ricercare nel trasporto dei rotoli, forti impatti possono infatti causare questa imperfezione. [13]

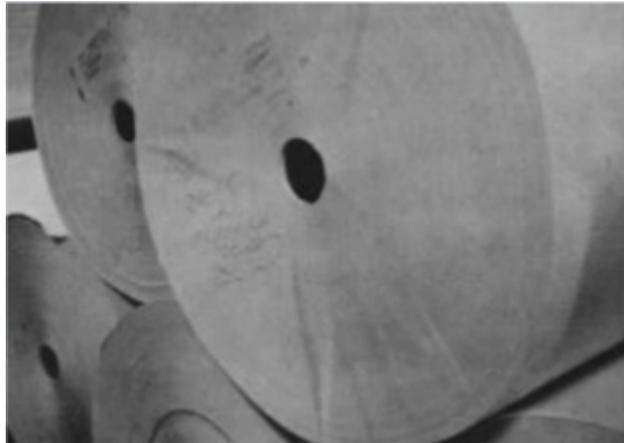


Figura 2.6: Sezione laterale di un rotolo con stellatura [19].

2.2 Principi di avvolgimento

La qualità dell'avvolgimento e la sua durezza sono determinate dalla cooperazione di tre parametri, riassumibili nell'acronimo TNT che sta per *Tension*, *Nip* e *Torque*. La precisione di controllo di queste tre variabili determina la buona riuscita del processo.

2.2.1 Principio della tensione di avvolgimento

La tensione impartita al materiale appena prima di essere avvolto sul rullo è la variabile dominante nel caso di materiali elastici. Maggiore è la tensione e più vigoroso sarà l'avvolgimento. Ogni materiale ha un modulo di elasticità che può essere usato per il calcolo della massima tensione sopportabile dal film. Nel sistema internazionale il modulo elastico è espresso in kilogrammi su metro quadrato, si preferisce utilizzare per comodità i kg/cm^2 . Empiricamente è stato dimostrato che tensioni dell'ordine dell'1.5% del modulo elastico sono sopportabili dai materiali senza che essi subiscano deformazioni permanenti. Per i motivi descritti in 2.1.2, è opportuno diminuire progressivamente la tensione di avvolgimento all'aumentare del diametro per evitare difetti come *Buckling*, *Starring* o *Dishing*. Sono considerate corrette diminuzioni del 25% a rotolo completato. Tra i materiali più comuni nell'industria dei Converting troviamo diversi polimeri ricavati dal petrolio: vengono quindi riportati alcuni di essi e i rispettivi moduli elastici. [12]

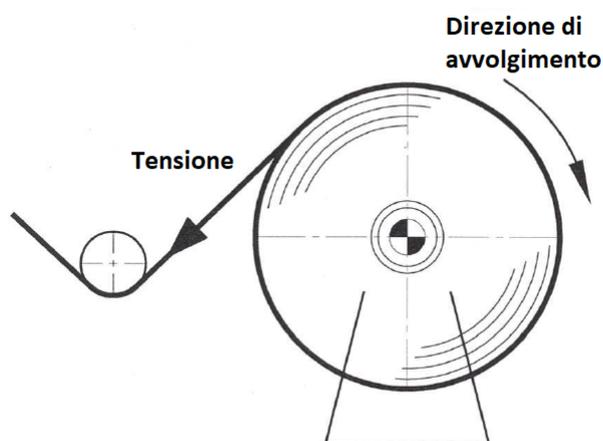


Figura 2.7: Principio della tensione. Traduzione da [12].

Tabella 2.1: Moduli elastici di alcuni materiali [12].

Materiale	Modulo elastico
PVC	350 – 560 kg/cm^2
LDPE	1000 – 1800 kg/cm^2
LLDPE	1400 – 2100 kg/cm^2
HDPE	8400 – 14000 kg/cm^2
BOPP	7700 – 24000 kg/cm^2
OPET	35000 – 42000 kg/cm^2

2.2.2 Principio del Nip

Se nel caso di materiali elastici la variabile predominante per un buon avvolgimento è la tensione del film in entrata sul tamburo, per i materiali rigidi lo è il Nip. Il Nip è la forza esercitata da un secondo tamburo, agente sul tamburo primario di avvolgimento, diretta verso il centro del rotolo. Il secondo tamburo agisce come guida per il materiale stesso in entrata sul tamburo di avvolgimento.

L'utilità primaria di questa particolare struttura è quella di rimuovere lo strato d'aria che accompagna il film nel processo di avvolgimento, evitando quindi che si infiltri tra gli strati avvolti. Inoltre, grazie a questo secondo tamburo premente, il rotolo acquisisce tensione interna.

La durezza di avvolgimento è quindi direttamente proporzionale alla forza premente o al Nip. Se il materiale è flessibile, come alcuni film per il confezionamento, la sfida è complicata da un bilanciamento per un Nip che da una parte rimuova la quantità d'aria necessaria e dall'altra non deformi il film, causando il blocco del rotolo e la fusione degli strati. Come per la tensione di avvolgimento, è opportuno diminuire gradatamente il Nip all'avvicinarsi del fine rotolo per evitare difetti di *Starring* e *Telescoping*. A volte, soprattutto per materiali particolarmente stretti, l'infiltrazione d'aria tra uno strato e l'altro può essere un vantaggio per prevenire deformazioni. In questa particolare situazione, nota come *Gap-Winding*, il rullo appoggiato sulla superficie del rotolo, dal quale viene esercitato il Nip, viene mantenuto leggermente distaccato dalla superficie. [12]

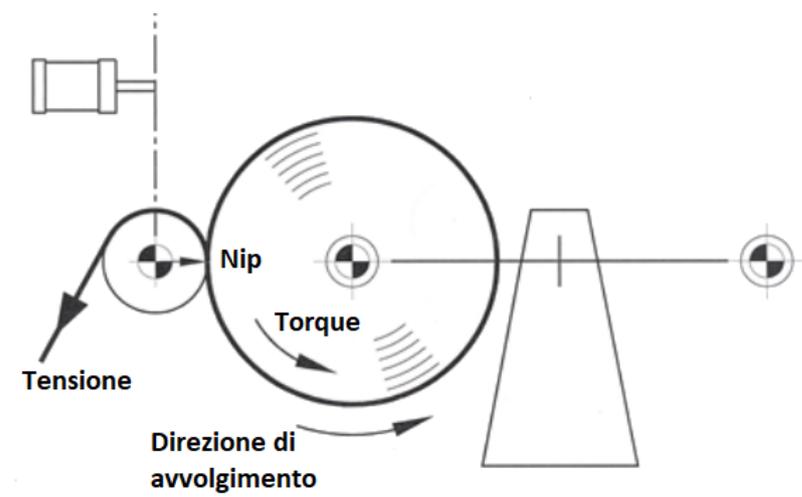


Figura 2.8: Principio del Nip. Traduzione da [12].

2.2.3 Principio della coppia di avvolgimento

La coppia di avvolgimento, detta *Torque*, è indotta dal centro del rotolo e trasmessa a tutti gli strati del materiale, stringendo i più interni. Nel caso di un avvolgimento eseguito solamente con il movimento dell'asse centrale del rotolo, il principio della coppia coincide con quello della tensione di avvolgimento.

Differente, invece, è nel caso degli avvolgimenti guidati da tamburi adiacenti al rotolo di materiale: in questi casi, mentre con il rullo esterno viene controllata la tensione del film, la coppia generata dall'asse del rotolo può raffinare ulteriormente la durezza dello stesso, combinando quindi i risultati dei due principi ora resi indipendenti. [12]

2.2.4 L'avvolgitore di Pfeiffer e il WIT

Un protagonista dello studio dell'avvolgimento è stato Pfeiffer, il quale determinò empiricamente una relazione tra le tre componenti TNT. Sviluppando un particolare *winder*, utile per estrapolare la relazione desiderata, riuscì a dimostrare che la forza dovuta al Nip è responsabile dell'aumento di tensione interna, quindi di rigidità del rotolo, in misure decisamente più importanti rispetto al contributo della sola tensione di avvolgimento.

L'avvolgitore sviluppato da Pfeiffer, di tipo superficiale, è in grado di misurare la tensione interna del materiale estrapolando dal rotolo l'ultimo strato in modo che, prima di essere immesso nuovamente nella bobina, venga misurata la tensione tramite una cella di carico. Con questa strategia risulta quindi possibile misurare

la tensione dell'ultimo strato dopo aver superato il rullo motorizzato dal quale viene esercitato il Nip.

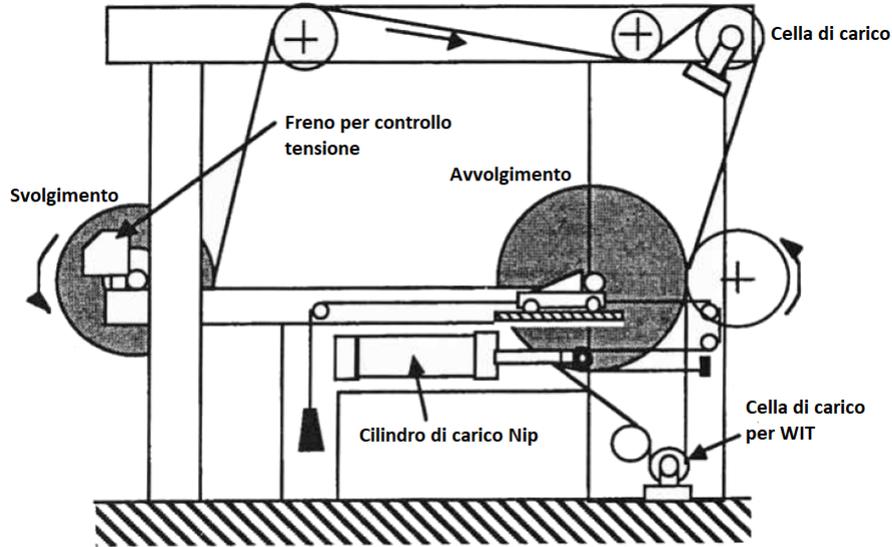


Figura 2.9: Winder di Pfeiffer. Traduzione da [20].

La relazione, ricavata empiricamente, che descrive la rigidità del rotolo o WIT (Wound-In Tension) è la seguente:

$$WIT = \frac{1}{B} \ln \left(\frac{N + A}{A} \right) + \frac{TN}{C + DN} \quad (2.1)$$

Dove N è il Nip [PLI: Pounds per Linear Inch], T è la tensione [PLI] e A, B, C, D sono coefficienti che dipendono dal nip e dal diametro del rotolo. L'avvolgitore di Pfeiffer mostra malfunzionamenti a bassi carichi di Nip, questo implica una relazione tra il WIT e il coefficiente di attrito statico tra il rullo avvolgitore e quello avvolto. Viene quindi dedotta da Pfeiffer, dopo esperimenti con materiali diversi, la seguente relazione:

$$WIT \leq \mu_{st} N \quad (2.2)$$

dove μ_{st} è il coefficiente di attrito statico e N è il Nip [PLI].

Studi più recenti basati su avvolgitori puramente centrali e condotti da Good portano una relazione più semplice tra Nip e WIT rispetto a quella della formula trovata da Pfeiffer. La seguente formula è rielaborata, rispetto all'originale, per avere delle unità di misura più consone al sistema internazionale.

$$WIT = T + \frac{\mu N}{10C} \quad (2.3)$$

dove WIT è la durezza del rotolo [Mpa], T è il tensionamento del materiale [MPa], μ è il coefficiente di attrito statico, N è il Nip [N/cm] e C è lo spessore del web [mm]. [1] Relazioni più complicate, che esulano dalla seguente tesi, includono anche la terza variabile TNT, la coppia, nello studio della durezza del rotolo (WIT). Grazie a queste dipendenze, tra tensione e Nip, è comunque possibile prevedere informazioni qualitative rispetto alla durezza dell'avvolgimento in corso. [20]

2.3 Tipologie di avvolgitori (*Winder*)

Sul mercato esistono diversi tipi di *winder*, ognuno di essi utile per particolari situazioni produttive. Scegliere la più corretta tipologia di avvolgitore spesso determina la buona riuscita del prodotto, in termini di qualità di avvolgimento e assenza di difetti.[21]

L'avvolgimento di una bobina crea delle forze in grado di mantenere il materiale coeso permettendo così il trasporto; le stesse forze, però, potrebbero danneggiarlo. Questo può capitare nel caso di un utilizzo di un avvolgitore non adeguato al materiale avvolto o ad una scorretta automazione del processo. Vengono quindi introdotte le tre principali tipologie di avvolgitori, mettendo in luce i principi TNT utilizzati da ognuna e relativi vantaggi e svantaggi.[12]

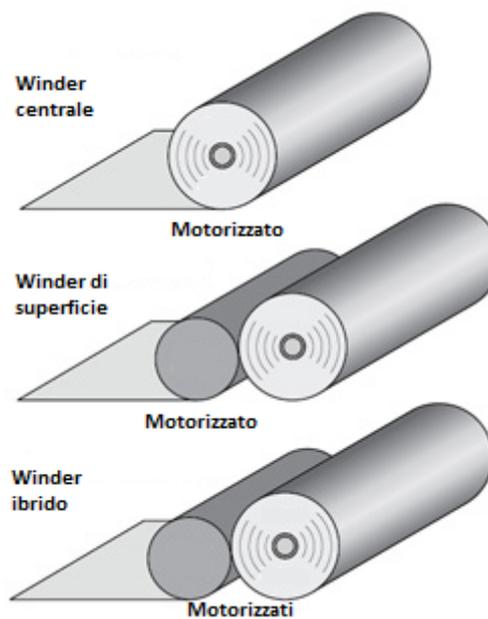


Figura 2.10: Tipologie di avvolgitori. Traduzione da [21].

2.3.1 *Winder* centrali

In generale un avvolgitore di tipo centrale presenta un albero avvolgitore motorizzato dal quale viene prodotta tutta la coppia utile al processo. Nel caso descritto viene utilizzata soltanto la tensione per il controllo della durezza di avvolgimento. A volte può essere comunque introdotto un rullo di pressione adiacente al rotolo, in questo caso entra in gioco anche il principio del Nip.

Grazie ai winder centrali è possibile produrre dei rotoli piuttosto soffici e avere una velocità di produzione particolarmente elevata. Questa tipologia di avvolgitore presenta invece delle limitazioni sul diametro massimo processabile in quanto tutto lo sforzo di avvolgimento è da imputare all'asse centrale senza avere collaborazione da coppie esterne, come nel caso di rulli appoggiati al materiale. Inoltre, a fine avvolgimento, il cambio bobina implica la rimozione dell'albero, operazione che può danneggiare il rotolo producendo scarto.

Un altro fattore da tenere in considerazione è la dimensione del rotolo, infatti a velocità e tensione del materiale costanti, la velocità di avvolgimento è inversamente proporzionale al diametro. Questo perché il valore minimo definisce la velocità massima del motore, mentre la massima coppia richiesta è data dal valore massimo dello stesso.

L'avvolgitore centrale rimane comunque una delle soluzioni più comuni. Oltre a permettere avvolgimenti rapidi e morbidi consente la lavorazione in entrambi i versi di rotazione, operazione non possibile nelle versioni base degli avvolgitori di superficie. [22]

2.3.2 *Winder* di superficie

L'avvolgitore di superficie comprende un rullo non motorizzato sul quale viene avvolto il *web* e uno o più rulli motorizzati posti accanto al primo. Con questa tipologia di avvolgitori, il materiale viene avvolto tramite contatto sulla superficie dello stesso, questo però spesso implica poter avvolgere il film in una direzione sola. Mentre la tensione svolge un ruolo fondamentale nel caso di materiali elastici, il Nip domina l'avvolgimento di materiali inelastici.

Il vantaggio principale di un avvolgimento superficiale è la sorgente della tensione applicata al materiale: questa infatti è impartita dal rullo superficiale e non trasmessa tramite gli strati di materiale, come nel caso dell'avvolgitore centrale, permettendo così un controllo molto più preciso.

Questo tipo di avvolgitore non è adatto ai processi nel quale è utile effettuare *Gap-Winding*, l'iniezione d'aria tra uno strato e l'altro è infatti sfavorita dalla necessità di contatto tra il rullo di traino e il rotolo. Gli avvolgitori superficiali sono utili quindi per processare rotoli di durezza e diametro elevati a costi ridotti. Inoltre, non hanno problemi relativi al cambio bobina che viene effettuato su un albero non motorizzato. [12]

2.3.3 *Winder* ibridi

Esistono avvolgitori ibridi, una soluzione mista tra quelli centrali e quelli di superficie. Questa tipologia fa uso di tutti i tre principi TNT. La tensione del materiale è utilizzata, in combinazione con il Nip impartito dal rullo superficiale, per ottimizzare l'avvolgimento. Nel contempo è possibile seguire un profilo desiderato di durezza del rotolo controllando, in modo indipendente dalle due precedenti, la coppia impartita dal rullo centrale.

Particolarmente interessante è il disaccoppiamento tra la tensione del materiale e quella di avvolgimento.

Il *winder* ibrido è quindi utilizzato nei processi produttivi in cui è richiesta una riduzione controllata della tensione di avvolgimento all'avvicinarsi del fine rotolo. Tuttavia la scelta della soluzione mista grava sul costo del macchinario, rendendola la soluzione meno economica. [12]

Capitolo 3

Il controllo di tensione

3.1 La tensione di un materiale

La tensione è definita come la forza applicata a un film di materiale nella direzione della macchina. Nel sistema anglosassone la tensione di un materiale è generalmente misurata in PLI (Pounds per Linear Inch), nel sistema internazionale l'unità di misura standard è invece il N/m (Newton al metro). Moltiplicando il valore della tensione in N/m per la larghezza in metri del film si ottiene la tensione totale applicata al materiale, la quale si misura in Newton.

La tensione applicata a un film può essere descritta come una forza pari al peso di un oggetto appeso al materiale stesso. Dividendo questo valore per la larghezza del materiale in metri avremmo la tensione misurata in N/m.

Per poter applicare un controllo sul film avvolto è necessario trazionarlo, il controllo di tensione rimane quindi uno degli aspetti essenziali per un buon processo di avvolgimento. Studi ed evidenze dimostrano che il materiale si allinea perpendicolarmente ai rulli avvolgitori che siano folli o motorizzati. Questo comportamento viene eluso nel momento in cui non sia applicata una tensione appropriata, in questi casi infatti il materiale può avere un comportamento inatteso rendendo inefficaci i contatti con i tamburi utili all'orientamento dello stesso. D'altra parte la tensione applicata non deve eccedere una certa soglia per evitare fenomeni di stiratura.

Un cattivo controllo di tensione nelle fasi di svolgimento e avvolgimento porta, come già dettagliato nella sezione 2.1.2, rispettivamente a fenomeni di *Telescoping* e *Dishing*. In processi più delicati come la stampa e la laminazione il contributo di una buona padronanza del controllo di tensione è ancora più evidente.

Vengono riportati due esempi di materiale sottoposto a processi di lavorazione affetti da un controllo di tensione non adeguato, si nota infatti una curvatura causata dalla differenza di tensione dei due materiali accostati. [16]

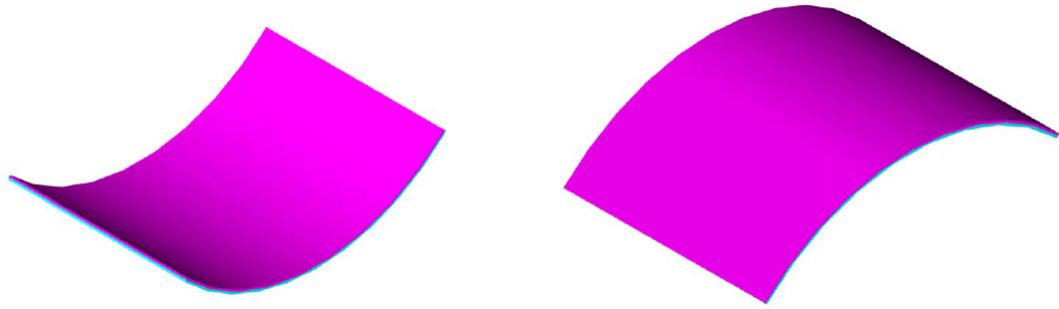


Figura 3.1: Esempi di materiale composto da due fogli accostati a tensioni differenti [16].

3.2 Relazione tra tensione e diametro

Nell'industria del Converting e, in generale, in qualunque processo che prevede la lavorazione di un film di qualsiasi materiale, ci sono tre zone strategiche nelle quali è necessario un controllo di tensione. L'obiettivo, durante l'intero processo, è quello di mantenere costante la tensione applicata al film in qualunque punto.

Sui rulli di avvolgimento e svolgimento è di primaria importanza seguire il processo con un controllo di trazione dipendente dal raggio attuale; spesso anche nelle zone intermedie della lavorazione è necessario ribadire una tensione adeguata alle lavorazioni in atto.

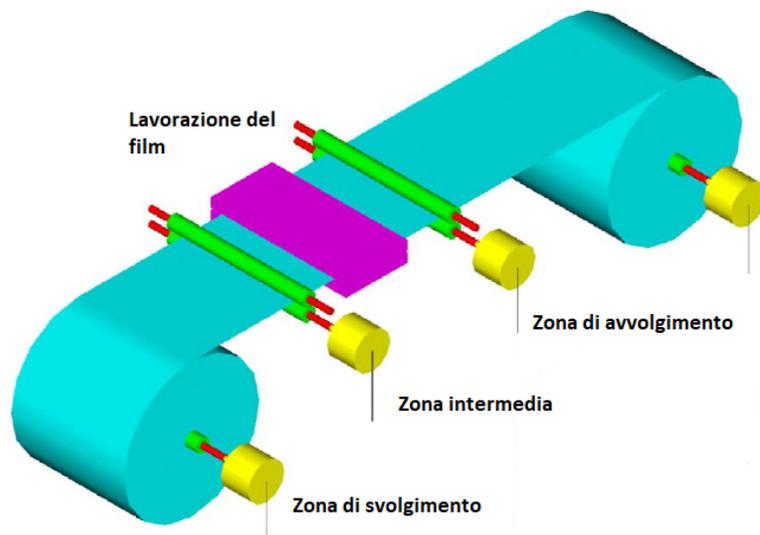


Figura 3.2: Schema delle principali zone di controllo del tiro. Rielaborazione da [16].

In generale, per applicare una tensione a un materiale è necessario imprimere una coppia al rullo, la relazione tra i moduli di tensione e coppia è lineare e dipende dal raggio secondo la legge

$$|\tau| = |T| \cdot |r| \quad (3.1)$$

dove τ è la coppia (*Torque*) [Nm], T è la tensione [N] e r il raggio [m]. Nelle zone iniziali e finali del processo, dove avviene l'avvolgimento o lo svolgimento del film è quindi indispensabile tenere presente questa relazione per adattare il controllo di tensione al raggio attuale del materiale avvolto o svolto. Nel caso di uno svolgimento, infatti, per mantenere costante la tensione, è necessario ridurre la coppia erogata dal motore del rullo in maniera proporzionale alla riduzione del diametro. Al contrario, è utile aumentarla nel caso di un avvolgimento con conseguente aumento del diametro avvolto. Di conseguenza, nelle zone di periferia è necessario un controllo di tensione molto dinamico, in quanto la grandezza dei rotoli varia repentinamente e spesso, mentre in quelle centrali della lavorazione è presente un'automazione molto più stabile; non si deve però sottovalutare che difetti del materiale o dell'avvolgimento potrebbero richiedere, anche se in misura inferiore, dinamicità di risposta nell'adeguare la tensione a quella desiderata. Per ogni zona è necessario quindi conoscere la tensione adeguata: spesso i valori più adatti vengono dedotti empiricamente osservando il processo. Esistono comunque tabelle riassuntive di buone stime dei diversi valori a seconda del materiale e del processo impiegato. Alcune linee guida prevedono *setpoint* di tensione pari al 20% del valore di trazione massimo del materiale.

In generale è sempre utile ricordare che ogni processo di avvolgimento è unico nel suo genere; questo spesso implica la supervisione di un tecnico esperto che sappia configurare in maniera ottimale i riferimenti di tensione utili ai fini di ottenere un buon prodotto.

La diminuzione di coppia erogata dal motore nel caso di uno svolgimento non è da confondere con la riduzione di tensione nel processo di avvolgimento menzionata in 2.1.2. Nel primo caso, come descritto, l'adattamento della coppia è necessario per mantenere costante la tensione del materiale; al contrario, nel secondo si vuole diminuire gradualmente la tensione stessa per evitare che, strato dopo strato, si accumuli una forza di compressione eccessiva provocando il collasso del rotolo oppure difetti come *Bagginess* o *Buckling*. Per quanto riguarda invece le zone di svolgimento e in qualunque punto intermedio del processo è perentorio garantire tensione costante. [16]

3.3 Misura della tensione

La tensione di un materiale può essere determinata con l'utilizzo di una cella di carico o di un ballerino. Nel primo caso è possibile inserire, nella zona opportuna,

un ulteriore cilindro collegato a una cella di carico: un trasduttore che converte una forza in un opportuno segnale elettrico che viene visualizzato a video o opportunamente interpretato da un PLC. Diversamente, se viene utilizzato un ballerino con sistemi a molla o pneumatici, è possibile calcolare la tensione del materiale nota la geometria del sistema e la dinamica dello stesso. Conoscendo le condizioni a vuoto del ballerino è possibile associare il carico agente su di esso alla tensione effettiva del materiale in quel punto.

Esistono altri metodi meno precisi per ricavare la tensione di un materiale: in particolare è possibile stimarla, noto il modello matematico del sistema oppure, nel caso di assi di avvolgimento non motorizzati, ricavarla tramite un dinamometro. Questi ultimi tuttavia non sono modelli di misura particolarmente efficaci e attuabili; nella maggior parte delle applicazioni non è possibile riprodurre un modello matematico preciso del sistema o non è possibile interagire fisicamente con parti di macchina per un calcolo manuale della tensione. Per questo motivo nella maggioranza delle macchine industriali viene effettuata la misura di tensione tramite ballerini o celle di carico.

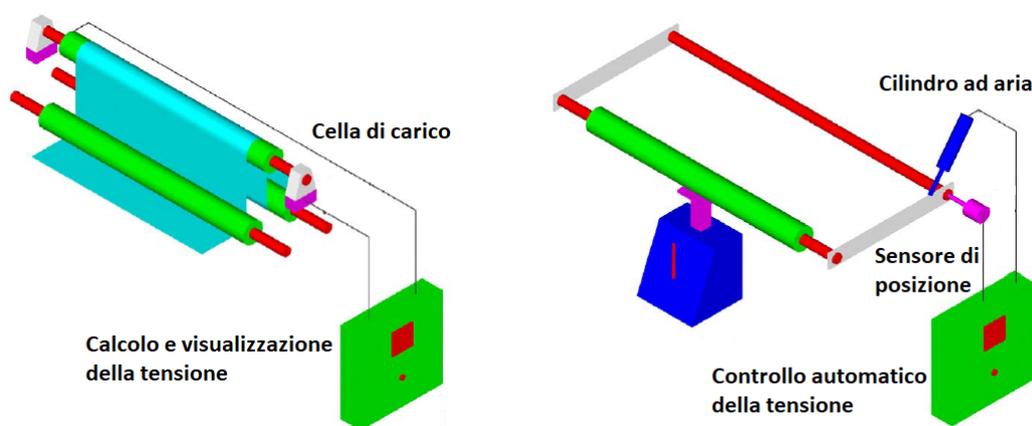


Figura 3.3: Schema strutturale di un'implementazione con cella di carico e una con ballerino. Rielaborazioni da [16].

3.4 Tecniche di controllo

Nelle macchine da Converting e, in generale in qualunque applicazione che preveda la movimentazione di un film da processare da un rullo svolgitore a uno avvolgitore, sono previste diverse stazioni di controllo. Le posizioni strategiche per attuarlo sono, come descritto in 3.2, in prossimità dell'avvolgimento e svolgimento ma anche in zone intermedie del processo. In generale però, sono le prime a richiedere

particolare attenzione in quanto vi è un continuo adattamento della coppia erogata ai motori per ottenere un tiro costante sul materiale.

In questa sezione verranno trattati i metodi principali, usati in industria, per garantire un controllo del tiro ottimale. In particolare, dopo una panoramica sul controllo manuale della tensione, verranno affrontati alcuni metodi in anello aperto con diametro misurato o calcolato e altri in anello chiuso utilizzando ballerini o celle di carico in retroazione. Questi ultimi, essendo in catena chiusa, sono molto più precisi e hanno la possibilità di compensare diverse situazioni non prevedibili come cambi di tensione improvvisi e difetti del materiale. Inoltre, rilevando direttamente la tensione del materiale, sono utilizzabili in tutte le zone del macchinario, anche in quelle interne, nelle quali non sono possibili i metodi in anello aperto. Questo tipo di controllo ha però lo svantaggio di non tenere traccia del diametro attuale e quindi non può garantire autonomamente un allentamento della tensione nella stazione di riavvolgimento per evitare i problemi al rotolo descritti in 2.1.2.

3.4.1 Controllo manuale della tensione

Considerabile come un caso degenere di controllo, il metodo più economico e semplice è un controllo manuale tramite un potenziometro agente sulla coppia del motore dell'asse di avvolgimento o svolgimento.

La buona riuscita del processo è completamente dipendente dalla perizia degli operatori che lo utilizzano, spesso infatti questa tipologia di controllo viene attuata nelle zone intermedie del processo e meno critiche rispetto a zone di avvolgimento o svolgimento. Il controllo del tiro nelle zone iniziali e terminali dell'applicazione, a causa dell'eccessiva dinamicità della coppia da erogare, è di solito affidato a tecniche automatiche.

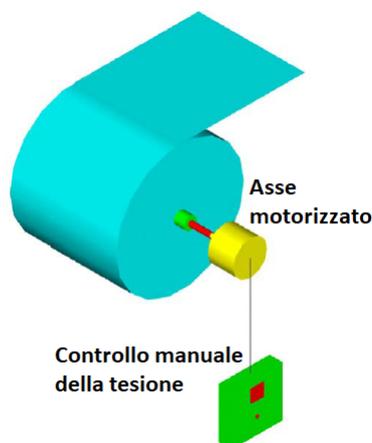


Figura 3.4: Esempio di controllo manuale di tensione. Rielaborazione da [16].

3.4.2 Controllo in anello aperto: diametro misurato

Come è già stato descritto precedentemente, noto il diametro attuale avvolto o svolto è possibile trovare la coppia da erogare al motore per mantenere costante il tiro del materiale; è quindi possibile misurare il diametro del rotolo attraverso opportuni sensori con fotocellule, laser o ultrasuoni. Una volta noto il diametro attuale e comunicato al controllore di tensione, verrà comandata la coppia adeguata all'asse che manterrà costante la tensione del film.

Tutte le tipologie di controllo basate sulla misura del diametro sono in anello aperto. Non esiste infatti alcuna retroazione per aggiustare opportunamente il tiro risultante; noto il diametro infatti viene direttamente erogata la coppia corrispondente al motore.

Per il processo di riavvolgimento è possibile modificare il controllore per ottenere anche una diminuzione graduale del tiro, evitando i danni al rotolo discussi in 2.1.2.

Follower arm

Questa tecnica di misurazione presenta un braccio in grado di muoversi con una ruota o un rotolo fissati all'estremità in appoggio al film. La parte meccanica è mobile e, appoggiando l'estremità sul rotolo, seguirà l'aumento o diminuzione del suo diametro. Per poter aderire alla superficie del materiale, il braccio è spinto verso la bobina con meccanismi a molla o pneumatici. Grazie a sensori come potenziometri o proximity è possibile misurare la sua posizione e quindi il diametro attuale del rotolo che, segnalato al controllore di tensione, permetterà il calcolo della coppia da erogare. Nel caso in cui la struttura meccanica del braccio sostenga un rotolo in appoggio al materiale, si parla di *Lay-On Roll*.

La facilità di questa soluzione insieme a un basso costo implementativo, rendono molto comune l'utilizzo di questo sistema. Nonostante questi vantaggi il sistema presenta parti meccaniche da mantenere, non compensa rotoli non perfettamente tondi generando oscillazioni e rende più difficoltose le operazioni di cambio bobine. Inoltre, essendo un controllo in catena aperta e basandosi unicamente sulla misura del diametro, non prevede nessun feedback e correzione alla tensione comandata.

Sensore a ultrasuoni

La misurazione del diametro attraverso un braccio mobile, come descritto, presenta alcuni svantaggi sia dal punto di vista meccanico che gestionale: è quindi molto comune una soluzione di misura tramite sensori.

Grazie a un rilevatore, generalmente a ultrasuoni, posto perpendicolarmente alla superficie del rotolo in avvolgimento o svolgimento, è possibile determinare con precisioni elevate il diametro attuale.

Nonostante sia di facile implementazione è necessario controllare che il materiale

non assorba la radiazione ultravioletta e occorre prestare particolare attenzione al posizionamento del sensore. Utilizzando quindi 3.1 è possibile determinare la coppia da erogare al motore per ottenere tiro costante.

Un vantaggio considerevole di questa strategia implementativa è la pura linearità del segnale relativamente al diametro misurato. L'assenza di parti meccaniche aggiuntive e il basso costo della sensoristica moderna contribuiscono a rendere questa soluzione la più comune tra le tecniche di controllo basate sulla misura del diametro. [16]

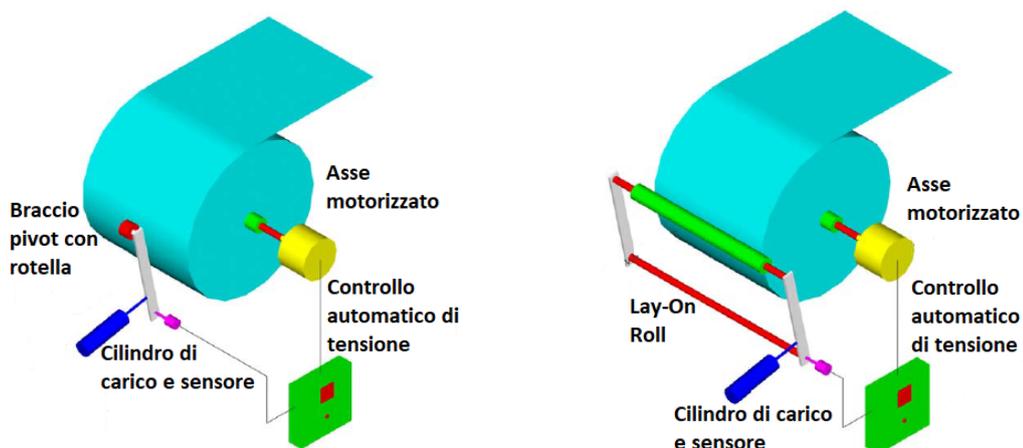


Figura 3.5: Controllo in anello aperto di tensione di tipo *Follower Arm*: il primo implementato con un braccio *pivot* e il secondo con un *Lay-On Roll*. Rielaborazioni da [16].

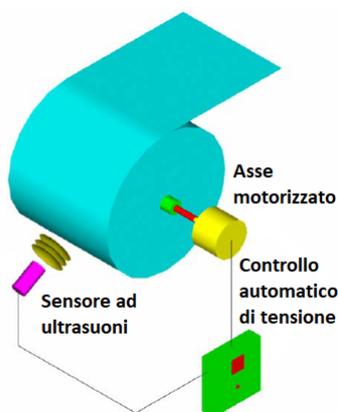


Figura 3.6: Controllo in anello aperto di tensione con sensore ad ultrasuoni. Rielaborazione da [16].

3.4.3 Controllo in anello aperto: diametro calcolato

In questo caso non vengono utilizzati sensori per misurare direttamente il diametro attuale, esso viene infatti calcolato matematicamente. La rilevazione quindi non è direttamente sulla grandezza di interesse quanto su altre caratteristiche del sistema utili al calcolo.

Spesso, in industria, non è facile avere le condizioni ideali per poter misurare il diametro direttamente con uno dei metodi descritti in 3.4.2. Le tecniche di misura del diametro, precise soprattutto per rilevare il diametro iniziale, vengono quindi spesso sostituite da tecniche di calcolo dello stesso. Il calcolo può essere eseguito utilizzando la velocità del materiale e dell'asse oppure, con una soluzione poco utilizzabile nei casi reali, conoscendo le rivoluzioni del motore e lo spessore del film.

Rapporto di velocità

Il diametro attuale è computabile come rapporto di due velocità: quella del materiale e quella dell'asse avvolgitore o svolgitore. La velocità del film è infatti costante e nota, mentre quella di avvolgimento è dipendente dal diametro.

Sulle macchine che implementano questa strategia di controllo vengono installati due sensori di giri al minuto (RPM), di solito degli encoder: uno su un rullo qualunque sul quale passa il materiale e l'altro sull'asse di avvolgimento. Il calcolo si basa sulla relazione tra velocità tangenziale e angolare di un moto circolare.

Vengono quindi riportati alcuni semplici passaggi matematici per ricavare il diametro, noto il modulo delle quantità descritte.

$$v = \omega \cdot r = 2\pi n \cdot r = \pi n \cdot D$$

Quindi si ottiene:

$$D = \frac{v}{\pi n} \tag{3.2}$$

dove D è il diametro [m], v è la velocità del film [m/min] e n gli RPM dell'asse di avvolgimento. Il risultato è un calcolo del diametro attuale della bobina note le due velocità in questione.

Esistono diversi metodi ottimizzati basati su questa soluzione: è infatti possibile integrare la posizione e la velocità nel tempo per avere risultati più precisi soprattutto nelle fasi di accelerazione e decelerazione.

Tra le soluzioni possibili in anello aperto, nonostante venga aumentata la complessità di controllo rispetto a soluzioni con *follower arm* o sensore ad ultrasuoni, è una delle più utilizzate grazie al basso costo realizzativo.

Come già accennato, il limite di questa versione basica, superato poi dalle ottimizzazioni, è la bassa precisione in accelerazione e decelerazione; inoltre, potrebbero essere utili dei filtri in uscita per avere un segnale ripulito dal rumore. [23]

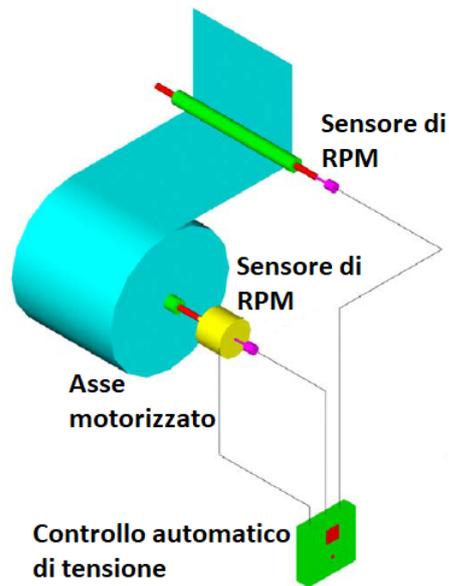


Figura 3.7: Controllo in anello aperto di tensione con calcolo del diametro. Rielaborazione da [16].

Rivoluzioni motore e spessore film

Un altro metodo di calcolo del diametro è basato sulla misura dello spessore del materiale. Questa strategia richiede l'utilizzo di un solo sensore posto sull'asse di avvolgimento per rilevare le rivoluzioni motore effettuate da un certo momento iniziale.

Nota questa informazione è possibile utilizzare la seguente formula per ricavare il diametro attuale del materiale avvolto.

$$D_{new} = D_{old} + (Thickness * 2) \quad (3.3)$$

Grazie a questa semplice relazione iterativa è possibile ottenere in ogni istante l'informazione aggiornata del diametro attuale. La formula è la stessa utilizzata nel caso di materiale svolto, avendo cura di sottrarre il secondo addendo.

Dal punto di vista teorico la soluzione sarebbe la più precisa possibile: oltre a non richiedere filtri, in quanto esente da rumore, il calcolo non è suscettibile ad accelerazioni e decelerazioni. Nei casi reali, al contrario, non si ha quasi mai una precisione tale sulla misura dello spessore da poterlo utilizzare nella formula. Lo spessore infatti è influenzato da diversi fattori come il tiro applicato, l'elasticità del materiale e la forza con cui è stato avvolto il materiale sulla bobina.

Questa serie di considerevoli svantaggi rende la soluzione di difficile implementazione e la confina a settori molto specifici in cui anche l'informazione di spessore, è una variabile ben nota e con una precisione accettabile. [23]

3.4.4 Controllo in anello chiuso: ballerino

Nel caso di una soluzione con ballerino, il controllo diventa un posizionamento. Nonostante le diverse tipologie di ballerini, tutte si possono ricondurre a un principio operativo comune: è presente un rullo folle che viene caricato con una forza in una direzione mentre il materiale tende a portarlo nella direzione opposta. All'interno del processo produttivo viene aggiunto un cilindro attorno al quale viene deviato il film di materiale. Il rullo viene controllato da un braccio o da un cilindro pneumatico che, con una forza regolabile, mette in tensione il web ora vincolato da questo nuovo percorso. Se il ballerino rimane in posizione centrale e stabile, la tensione del materiale è uguale alla forza selezionata dal braccio che ne controlla la sua posizione. Per controllare il tiro, il sistema di controllo deve assicurare che il ballerino rimanga nella sua posizione centrale. Se la tensione nel sistema cambia, anche la posizione del ballerino varia: quest'ultima viene quindi rilevata da un sensore di posizione che può essere di tipo diverso a seconda del ballerino installato. Se viene rilevato un movimento del ballerino dalla sua posizione centrale, il sistema risponderà con una variazione della velocità o della coppia del motore dell'asse di avvolgimento. Grazie quindi a una variazione della velocità dell'avvolgitore, il controllo di posizione del ballerino correggerà il suo errore di posizione, riportandolo a convergere verso il *setpoint*. La posizione del ballerino è solitamente determinata usando un encoder di posizione, il cui risultato verrà confrontato con il riferimento di posizione. La tensione viene quindi determinata esclusivamente usando il ballerino. Se la tensione cambia, la posizione del ballerino cambia di conseguenza.

Grazie a questo sistema, brevi fluttuazioni della velocità che producono un effetto sulla posizione del ballerino, difficilmente hanno degli effetti sulla tensione.

Questa tecnica è comunque sempre preceduta da un accurato studio delle caratteristiche fisiche del ballerino e delle sue possibilità di movimento.

Il principale vantaggio di questa soluzione è l'assorbimento di brevi fluttuazioni della tensione. Infatti il ballerino agisce come *buffer* per l'eccedenza o carenza istantanea di materiale. Questo è uno dei maggiori vantaggi per l'implementazione di questa tecnica, se rapportata ad altri tipi di controllo in anello chiuso o aperto. Grazie a questa importante caratteristica, il sistema funge da smorzatore per rotoli eccentrici e salti tra strati che avvengono in particolare quando si avvolgono cavi. In generale vengono attenuati gli strappi di tiro dovuti a qualunque genere di difetto del materiale o del rotolo. I limiti sono invece dovuti principalmente all'implementazione meccanica del ballerino ed alle sue caratteristiche dinamiche.

[16]

Il funzionamento

Il principio di funzionamento del ballerino si basa sul deviare il materiale con una forza definita. In passato veniva caricato il ballerino stesso con dei pesi per applicare la forza utile all'applicazione. In applicazioni più recenti sono stati introdotti cilindri ad aria comandati da opportune valvole. Oggi vengono utilizzati cilindri elettrici o pneumatici a basso attrito, spesso equipaggiati con controlli elettronici della pressione dell'aria. La deviazione del ballerino dalla sua posizione standard è rilevata usando un sensore e utilizzata per il calcolo della tensione del materiale. Essa dipende infatti dalla forza di richiamo del ballerino.

Spesso ci sono altri fattori che influenzano il valore attuale della tensione, risultanti dalla disposizione geometrica dei vari cilindri, dal peso del ballerino stesso e dal momento d'inerzia della struttura in movimento. Con un'accurata progettazione meccanica, questi effetti possono essere drasticamente ridotti se non eliminati. Il controllo di posizione correggerà l'attuale posizione del ballerino per riportarlo alla sua posizione di riferimento. Esso, in generale, è internamente scelto come parametro fisso. Per un ballerino la cui forza di supporto può essere regolata pneumaticamente o idraulicamente, può essere diminuita la durezza dell'avvolgimento.

Tipi di ballerino

Per la realizzazione di un ballerino esistono diverse strutture geometriche più o meno complesse; in industria spesso il collocamento di nuove parti meccaniche non è facile e nel tempo si sono affermate solo alcune delle configurazioni possibili.

- **Ballerino a braccio rotante:** una delle soluzioni più facili e per questo motivo molto comune. Viene collocato un cilindro aggiuntivo collegato a un braccio che può ruotare. La posizione del ballerino può essere rilevata tramite un encoder sull'asse di rotazione del braccio. Questa soluzione richiede attenzione nell'includere la forza di gravità nei conti ed è sensibile al momento di inerzia della struttura.
- **Ballerino lineare:** la soluzione con la potenzialità di immagazzinamento del materiale maggiore. Solitamente è una soluzione di facile applicazione anche su impianti già esistenti in quanto strutturalmente più compatta. Anche questa soluzione risente della forza di gravità e del momento d'inerzia della struttura oltre ad essere particolarmente costosa, soprattutto per l'utilizzo di trasduttori lineari.

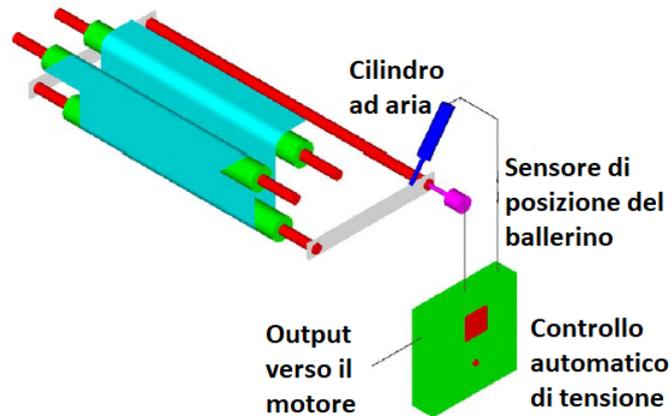


Figura 3.8: Esempio di controllo con ballerino a braccio rotante. Rielaborazione da [16].

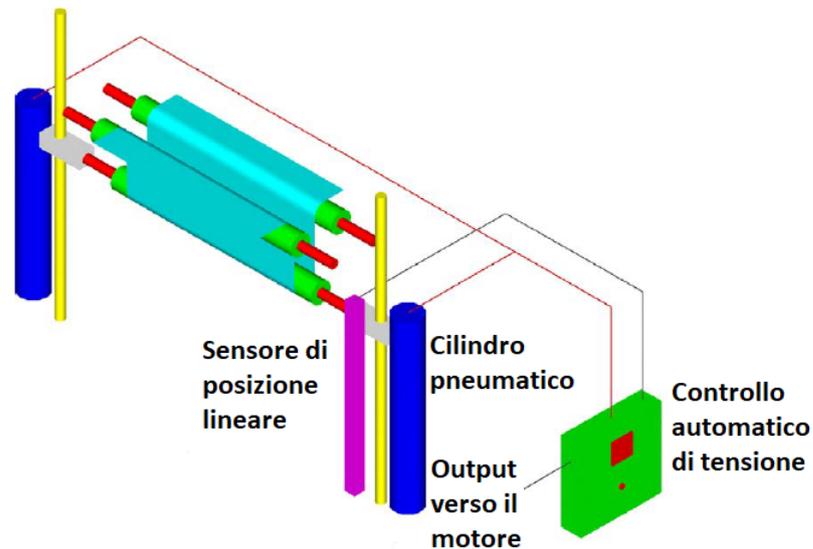


Figura 3.9: Esempio di controllo con ballerino lineare. Rielaborazione da [16].

- **Ballerino rotativo:** una soluzione indipendente dalla gravità. Questa soluzione oltre ad annullare l'effetto della gravità, riduce di molto quello del momento d'inerzia. Il cilindro ballerino infatti effettua un movimento traslatorio avanti e indietro definito da un asse rotativo ed è quindi ancorato ad un perno dove scarica la forza peso. Uno svantaggio nell'utilizzo di questa terza tipologia di ballerini è che la sua struttura implica un magazzino di materiale fisso e deciso in fase di progettazione.

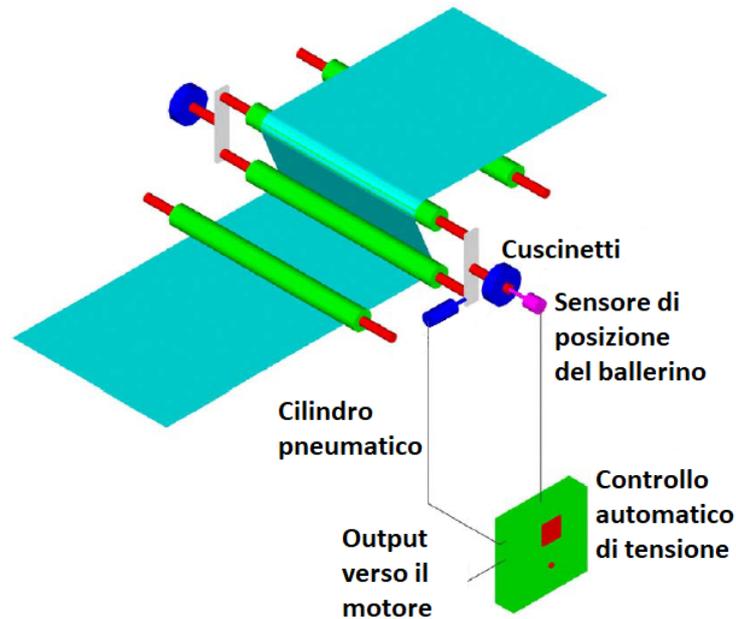


Figura 3.10: Esempio di controllo con ballerino rotativo. Rielaborazione da [16].

3.4.5 Controllo in anello chiuso: cella di carico

Questa tipologia di controllo utilizza una cella di carico per misurare la forza di richiamo di un cilindro attorno al quale viene fatto transitare il materiale. Il segnale in uscita dal trasduttore di tensione è proporzionale alla tensione e viene passato al controllore come valore attuale che quindi controllerà direttamente la tensione del materiale. La forza che viene esercitata sul rullo a causa della tensione del film dipende dall'angolo con cui viene avvolto il materiale su di esso.

Sono possibili diverse soluzioni apportando semplici modifiche goniometriche al calcolo, anche se generalmente l'angolo di avvolgimento è 180 gradi. Con questa configurazione fisica è necessario guidare il materiale sul rotolo in questione attraverso dei rulli di accompagnamento. Viene quindi selezionato un riferimento di tensione desiderato e il controllo, sulla base della tensione attuale misurata, adatterà la tensione per convergere al *setpoint*. Questa implementazione gode di una precisione molto elevata e per questo viene accolta positivamente da molte realtà industriali.

A differenza della soluzione con ballerino non è però possibile avere i vantaggi di stabilità di cui quest'ultima gode grazie alla funzione di *buffer* descritta in 3.4.4.

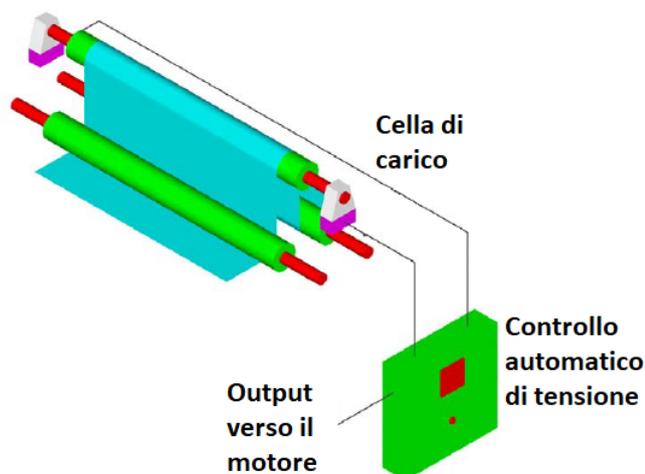


Figura 3.11: Esempio di controllo con cella di carico. Rielaborazione da [16].

3.4.6 Tecniche ibride di controllo

Nel settore del Converting non è raro che per certi tipi di applicazioni non siano raggiunti degli standard adeguati di produzione utilizzando solo un tipo di controllo. Spesso, infatti, è comune combinare diverse strategie per colmare le carenze reciproche. Alte velocità di produzione e forti accelerazioni possono rendere instabili i controlli ad anello chiuso; d'altra parte i controlli in anello aperto godono di scarsa precisione proprio per la mancanza di una retroazione. In soluzioni ibride abbiamo quindi un controllo in catena aperta, responsabile dell'informazione più grossolana, e uno in catena chiusa per raffinare i risultati ottenuti dal primo.

Implementazioni così costituite sono molto stabili e precise e hanno l'unico svantaggio di essere costose. Per un'azienda del settore è sempre un investimento concentrarsi su soluzioni di controllo innovative, soprattutto per le zone di avvolgimento e svolgimento. La maggior parte degli scarti nel settore è proprio dovuta a una cattiva automazione del controllo del tiro.

Al giorno d'oggi, grazie alla tecnologia sempre più avanzata dei sistemi di controllo, è possibile raggiungere livelli di accuratezza tali da sgravare gli operatori dal compito di controllare i processi di avvolgimento e svolgimento e permettere loro di concentrare gli sforzi sulle aree più critiche del processo.

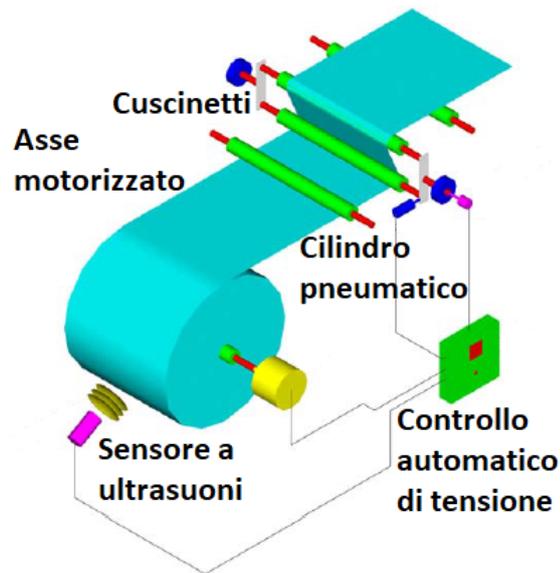


Figura 3.12: Esempio di controllo ibrido ad anello chiuso e aperto con sensore a ultrasuoni e ballerino rotativo. Rielaborazione da [16].

3.4.7 Tecnologie a confronto

Come già anticipato, scegliere il corretto tipo di automazione per un processo di avvolgimento, non è mai un compito semplice. Ogni tipologia di approccio ha i suoi vantaggi e svantaggi a seconda dell'applicazione. Spesso alcune caratteristiche geometriche della macchina vincolano i progettisti a rinunciare alla soluzione con ballerino in favore dell'installazione di una cella di carico. In termini di occupazione spaziale, il ballerino, è infatti la tipologia più svantaggiata, soprattutto in casi di montaggio a posteriori. In altri occasioni sono le caratteristiche tecniche del processo a vincolare la scelta: processi in cui sono presenti forti picchi di tensione sul materiale e frequenti cambi di velocità suggeriscono l'installazione di un sistema a ballerino, sfruttando le sue buone capacità di assorbimento e accumulo del materiale. D'altra parte, la staticità della soluzione con cella di carico implica facilità di installazione e minor manutenzione meccanica.

Per facilitare la scelta della strategia implementativa sono state nel tempo redatte diverse tabelle riepilogative dei parametri in gioco. Si riporta in 3.1 una tabella riassuntiva delle tre tipologie di automazione descritte, corredata dei rispettivi intervalli di valori utili per scegliere la miglior implementazione.

Tabella 3.1: Criteri empirici di selezione della tipologia di controllo più adatta [22].

Parametro	Anello aperto	Ballerino	Cella di carico
Rapporto diametri $R_d = D_{max}/D_{min}$	Fino a 10:1	Fino a 15:1	Fino a 15:1
Rapporto tensioni $R_t = T_{max}/T_{min}$	Fino a 6:1	Dipende dal supporto	Fino a 20:1
Prodotto rapporti $R_p = R_d * R_t$	Fino a 40:1	Fino a 40:1	Fino a 100:1
Velocità film	Fino a 600 m/min	Fino a 2000 m/min	Fino a 2000 m/min
Materiali consigliati	tessuto, carta, lamiere	gomma, cavi, fili, tessuto, carta	carta, fogli sottili

Capitolo 4

Industria 4.0

4.1 Dal passato fino ai giorni nostri

L'industria ha acquisito un sempre crescente ruolo di importanza nella comunità, un pilastro fondamentale per il lavoro e per l'economia. Per questo nel tempo i governi hanno continuamente cercato di migliorarla. In questi anni il mondo sta assistendo a una vera e propria rivoluzione industriale da paragonare a quelle avvenute nel passato.

Nel 1784 avvenne la prima, con l'introduzione del vapore, poi la seconda nel 1870 con l'introduzione della produzione di massa e dell'energia elettrica e infine la terza dal 1969 ad oggi con l'automazione, i computer e l'elettronica. Oggi stiamo quindi assistendo ad una quarta rivoluzione con l'introduzione dell' "Internet of Things" e delle grandi possibilità derivanti dall'uso esteso delle reti di connessione.

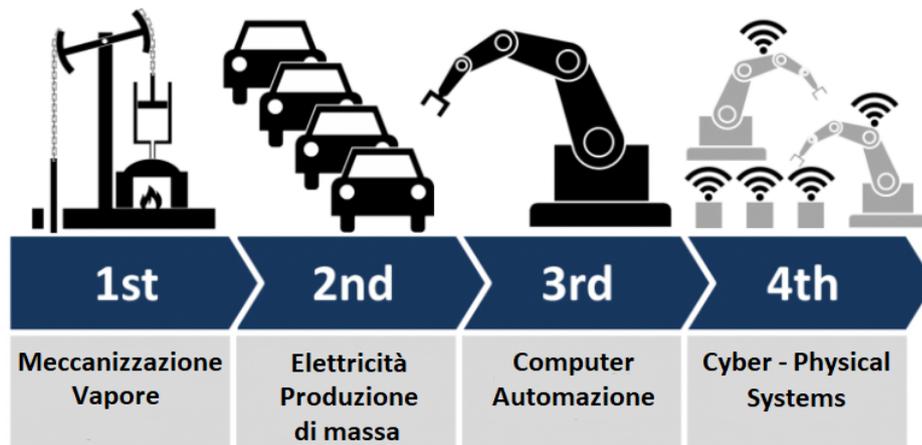


Figura 4.1: Le quattro rivoluzioni industriali [24].

4.2 La nuova rivoluzione industriale

La Germania, nel 2010, ha cominciato a usare il termine “Industria 4.0”. Questo concetto introduce l’uso dell’insieme delle tecnologie note come *Internet of Things*, abbreviato con IoT, e in generale di servizi Internet che consentono la collaborazione tra il mondo fisico e i sistemi digitali.

Il termine Industria 4.0 indica una tendenza dell’automazione industriale ad integrare alcune nuove tecnologie produttive per migliorare le condizioni di lavoro e aumentare la produttività e la qualità degli impianti.

La *Smart Factory* è il concetto chiave di questa rivoluzione. Essa richiede un’integrazione verticale, dalle entità fisiche ai sistemi informativi. Questo significa che tutti i dispositivi al più basso livello produttivo come PLC, macchine, Scada e robot possono comunicare e prendere decisioni con i livelli più alti dell’organizzazione aziendale come database e gestionali. In collaborazione con la *Smart Factory* abbiamo:

- *Smart Production*: nuove tecnologie produttive che creano collaborazione tra tutti gli elementi presenti nella produzione
- *Smart Service*: tutte le infrastrutture informatiche e tecniche che permettono di integrare i sistemi
- *Smart Energy*: creare sistemi più performanti riducendo gli sprechi di energia.

Tutto questo porta alla necessità di comunicazione con un protocollo standard che possa rendere possibile l’integrazione di sistemi eterogenei provenienti da diversi produttori.

Questo ha portato a scegliere l’ OPC UA come protocollo principe dell’Industria 4.0. Tra i vari motivi che hanno contribuito alla scelta di questo protocollo come standard ci sono la sicurezza e lo scambio dati e informazioni tra dispositivi ormai standardizzato.

L’innovazione tecnologica portata dall’Industria 4.0 avrà ripercussioni positive sulla produttività, sul fatturato e sul tasso di occupazione. [25]

4.3 Principali innovazioni

Big Data

Per *Big Data* si intende la gestione e l’analisi di un grande numero di dati che non possono essere gestiti con metodi convenzionali. La capacità di gestire una grande quantità di dati può portare a notevoli vantaggi strategici, invisibili con un’analisi di un campione più ristretto di informazioni. L’aumento dei dispositivi connessi ha

aumentato in maniera esponenziale la quantità di informazioni scambiate facendo emergere così la nuova figura del *Big Data Analyst*.

Internet Of Things

Internet of Things è la connessione a internet di un numero sempre crescente di dispositivi legati alla vita quotidiana di ogni persona. Il numero di oggetti interconnessi sta aumentando di anno in anno.

Mentre nell'ultima decade il numero di telefoni e pc connessi è pressoché rimasto costante con un lieve rialzo, le cosiddette “*Things*” sono più che raddoppiate.

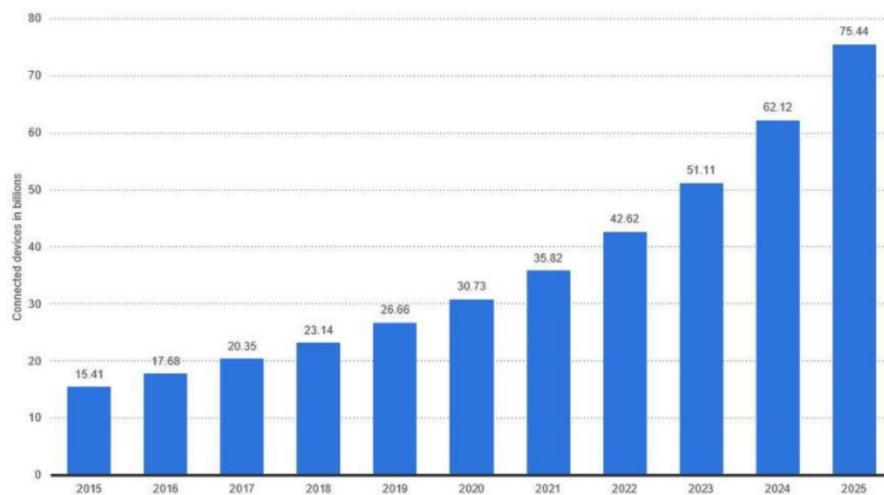


Figura 4.2: Grafico che mostra il numero di dispositivi connessi nel corso degli anni, espresso in miliardi [26].

Cyber-Physical Systems

I *Cyber-Physical Systems* si riferiscono al fatto che la produzione è accompagnata da processi di elaborazione informatici, tra cui sensori e attuatori per raccogliere e spedire dati.

4.4 Norme Industria 4.0

Un macchinario, per essere conforme alle norme dell'Industria 4.0 deve essere interconnesso. Affinché un bene possa essere definito interconnesso è necessario e sufficiente che:

1. Scambi informazioni in modo bidirezionale con sistemi interni come un sistema gestionale, di pianificazione, di sviluppo, di monitoraggio o anche con altre macchine dello stabilimento tramite un collegamento su specifiche documentate come TPC-IP, HTTP, MQTT, OPC UA o simili.
2. Sia identificato univocamente, al fine di riconoscere l'origine delle informazioni, mediante l'utilizzo di standard di indirizzamento internazionalmente riconosciuti come l'indirizzo IP.

Per una fabbrica, rendere i propri macchinari conformi all'Industria 4.0 porta notevoli vantaggi economici. Una perizia, corredata da analisi tecnica, potrà attestare la conformità delle linee di produzione al nuovo standard.

4.5 Il protocollo OPC UA

4.5.1 OPC Legacy

OPC è un insieme di standard sviluppato dalla OPC Foundation per il controllo e il monitoraggio di sistemi industriali, rilasciato nel 1996, basato su Microsoft DCOM e operativo sui sistemi Windows.

La base del protocollo è l'architettura *Client-Server*. La inevitabile presenza di alcune limitazioni hanno in seguito portato allo sviluppo della nuova versione dello standard.

Uno dei problemi principali era la sola compatibilità con Windows: questo escludeva tutte le altre piattaforme come Linux, Unix e Mac OS dal poterlo utilizzare. Inoltre esistevano diverse limitazioni all'accesso remoto ai dati che non potevano essere trasferiti tra reti di diverso tipo. Veniva quindi meno l'interscambio di informazioni tra i diversi livelli dell'infrastruttura informatica aziendale, uno dei cardini dell'Industria 4.0. Infine non era stata prestata particolare attenzione alla semplicità di configurazione e alla sicurezza dei dati.

Consci delle limitazioni sopra descritte si è cercata un'evoluzione dello standard che andasse incontro alle necessità delle aziende moderne.

Partendo quindi dallo standard sono stati sviluppati quattro tipi di *server* OPC:

- OPC DA (*Data Access*) per la trasmissione di dati real time
- OPC A&E (*Alarms and events*) per spedire allarmi e eventi al *client*
- OPC HDA (*Historical Data*) per l'accesso a dati storici da parte del *client*
- OPC UA (*Unified Architecture*) per rendere possibile lo scambio di qualsiasi tipo di dati

4.5.2 OPC UA

L'interoperabilità semantica è considerata la chiave per l'implementazione dell'Industria 4.0. Una comunicazione sicura dal campo ai database o al cloud può essere implementata con l'OPC UA. OPC UA, acronimo di *Open Platform Communications United Architecture*, è uno standard per lo scambio dati per le comunicazioni industriali (*Machine-to-Machine* o *PC-to-Machine*).

Punto di forza è l'indipendenza dalla marca del sistema di controllo, dall'applicazione, dai linguaggi di programmazione e dai sistemi operativi sui quali è in esecuzione l'applicazione stessa.

Uno dei più grandi pilastri dell'implementazione dell'Industria 4.0 è lo scambio di informazioni e dati in modo sicuro e standardizzato tra diversi dispositivi e livelli dell'industria.

Così, nell'aprile del 2015 venne raccomandato come il modello architetturale standard per l'Industria 4.0 l'IEC Standard OPC UA. [27]

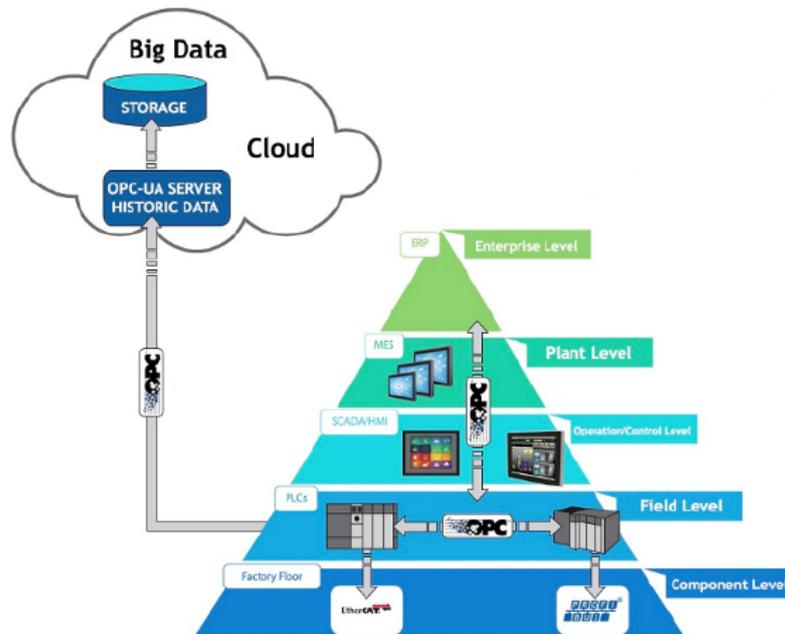


Figura 4.3: Il protocollo OPC UA integrato con l'organizzazione aziendale [28].

OPC UA è la versione più recente dei *server* OPC, rilasciata nel 2008, e apporta tutte le evoluzioni necessarie per superare i punti deboli del vecchio standard. La più grande differenza dalla versione passata è che i dati macchina sono ora descritti semanticamente in una maniera comprensibile alla macchina. Questa versione non fa più uso dell'interfaccia DCOM e questo la porta a essere compatibile con molti altri sistemi operativi.

OPC UA può essere mappato su una larga scala di protocolli di comunicazione e i dati possono essere codificati in vari modi per bilanciare portabilità e efficienza. Inoltre, i dati possono essere visualizzati in molti formati differenti tra loro, includendo strutture binarie e documenti XML o JSON.

Alcune delle caratteristiche più importanti e degne di nota dell'OPC UA sono:

- Il protocollo implementa un *Address Space* e un modello che permette di accedere ad allarmi, dati storici e altro, facendo utilizzo del medesimo set di servizi e organizzando in maniera gerarchica i dati.
- Offre il supporto per JAVA, C++,C# e altri linguaggi di programmazione superando quindi tutte le problematiche legate al sistema operativo e diventando *cross-platform*.
- Integra tutte le caratteristiche dei predecessori salvando informazioni storiche e attuali in variabili gestibili con metodi all'interno dello stesso *server*.
- Migliora la sicurezza usando certificati e chiavi pubbliche per scambiare messaggi sicuri attraverso TCP e HTTP.
- Viene usato OpenSSL per i processi di autenticazione
- Permette la creazione di dati complessi e strutturati

4.5.3 Architettura

Uno degli obiettivi dell'Industria 4.0 e dell'*Industrial Internet of Things (IIoT)* è lo scambio di informazioni tra le macchine e i servizi in modo sicuro e standardizzato. Il requisito primario per usare OPC UA per l'Industria 4.0 è l'uso di una rete basata sul protocollo IP. Ci sono due meccanismi per scambiare i dati:

- Un modello *Client-Server* nel quale i *client* utilizzano servizi dedicati e offerti dal *server*
- Un modello *Publisher-Subscriber* dove un *server* pubblica informazioni all'esterno

Entrambi i meccanismi sono slegati dai protocolli utilizzati che possono essere quindi TCP e HTTPS per il primo modello e UDP o MQTT per il secondo. Con il modello *Pub-Sub*, le applicazioni OPC UA non scambiano direttamente richieste e risposte. Al contrario, i *Publisher* mandano messaggi a un *Message Oriented Middleware*, senza la conoscenza di quanti *Subscriber* ci siano. Similmente, i *Subscriber* esprimono interesse per tipi di dato e messaggi senza la conoscenza dei *Publisher*.

Il modello maggiormente usato rimane comunque quello *Client-Server*, e per questo motivo verrà approfondito in questo testo. *UA-server* è il dispositivo che propone le informazioni, l'*UA-client* è quello che le utilizza.

In un'organizzazione aziendale, grande o piccola che sia, ogni livello organizzativo può contenere diversi *client* e/o diversi *server*. Ogni *client* può essere connesso a diversi *server* contemporaneamente e lavorare sotto diversi sistemi operativi, garantendo interoperabilità tra tutti loro.

In molti casi, i *client* non hanno alcuna conoscenza del formato dei dati, ma saranno capaci di determinarlo *runtime* così da poterli utilizzare in modo appropriato. OPC UA dà supporto a relazioni tra i nodi invece di limitarsi a una singola gerarchia. In questo modo, un *server* può presentare i dati nel formato più consono per i *client*. Questa flessibilità, combinata con il supporto per la definizione di tipi, rende OPC UA applicabile a una vasta gamma di architetture.

OPC UA fornisce un meccanismo ai *client* per rilevare e recuperare velocemente errori di comunicazione senza dover aspettare lunghi tempi di attesa derivanti dai protocolli sottostanti.

OPC UA dà supporto a una vasta gamma di *server*, dai PLC di macchina a *server* di piccole o grosse dimensioni, senza preoccuparsi di quanto siano eterogenei in termini di dimensioni, performance, piattaforme di esecuzione e capacità di calcolo. Il protocollo quindi definisce dei subset, chiamati profili, dedicati alle varie tipologie di *server*. I *client* quindi possono scoprire i profili di un *server* e creare una rete di interazioni con esso tramite il profilo più adeguato.

Le specifiche dell'OPC UA sono stratificate, questo per isolare il cuore dell'applicazione dalla tecnologia e dalle reti sottostanti. Questo permette al protocollo di essere mappato a tecnologie future, senza dover stravolgere l'architettura base.

Sono possibili anche architetture miste dove sono presenti *server* combinati con *client*. [29]

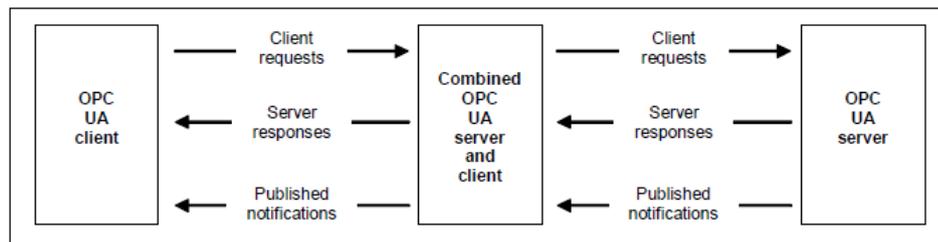


Figura 4.4: Possibile modello architetturale del protocollo OPC UA [29].

4.5.4 OPC UA *Client*

Ogni *client* può accedere alle interfacce messe a disposizione dai *server* per scambiare con essi dati in entrambe le direzioni. Spesso i *client* sono delle applicazioni software per elaborare dati inviati dai *server*.

Tra le applicazioni più comuni troviamo SCADA o gestionali. L'OPC UA *client* include *OPC UA Client Structure*, *OPC UA Client Application* e *OPC UA Client Communication Stack* per interagire con il *server*. La *Communication Stack* converte le chiamate *API* in messaggi: ottiene messaggi di risposta e notifica dal *server* e tramite l'*API* li passa all'applicazione.

Come descritto dalla figura 4.5, *Client Application* è il codice che implementa la funzione del *client*. Esso utilizza il *Client API* per spedire e ricevere le richieste e le risposte con il *server*.

Il *Client API* è un'interfaccia interna che isola il codice dell'applicazione *client* dallo *stack* di comunicazione OPC UA. Lo *stack* di comunicazione OPC UA converte le chiamate alla *client API* in messaggi che vengono poi spediti al *server* attraverso l'entità di comunicazione sottostante. Inoltre, riceve da quest'ultima le risposte e i messaggi di notifica che, attraverso il *Client API*, inoltra all'applicazione *client*. [29]

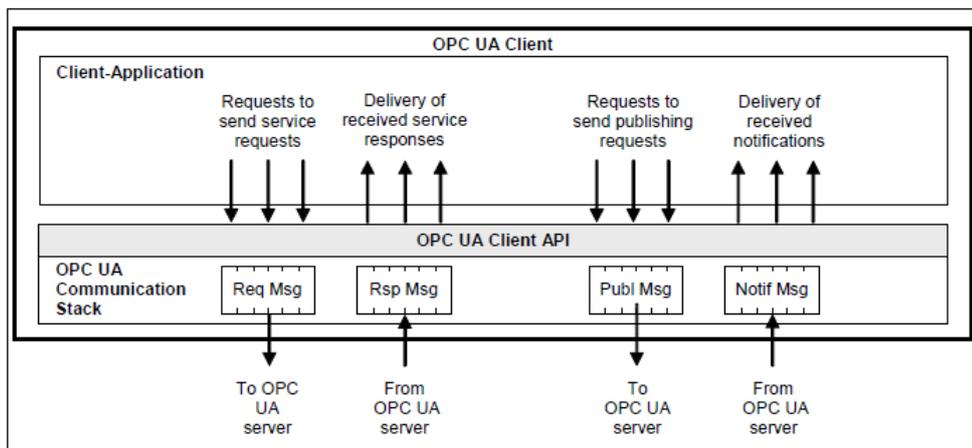


Figura 4.5: Architettura di un OPC UA *client* [29].

4.5.5 OPC UA *Server*

Il *server* OPC UA è il software che implementa lo standard del protocollo e mette a disposizione l'interfaccia OPC per il mondo esterno.[29]

Nel mondo dell'industria la situazione più comune è che un produttore hardware integri all'interno dei suoi dispositivi un *server* OPC, in modo da rendere standard

l'accesso alla sua apparecchiatura.

In questo testo, verrà preso in considerazione il controllore Siemens Simotion, che integra al suo interno un *server* OPC UA per il monitoraggio e il controllo delle sue variabili. L'OPC UA *server* include la *OPC UA Server Application*, i *real objects*, *OPC UA Address Space*, *OPC UA Server API* e il *Communication Stack*. [29]

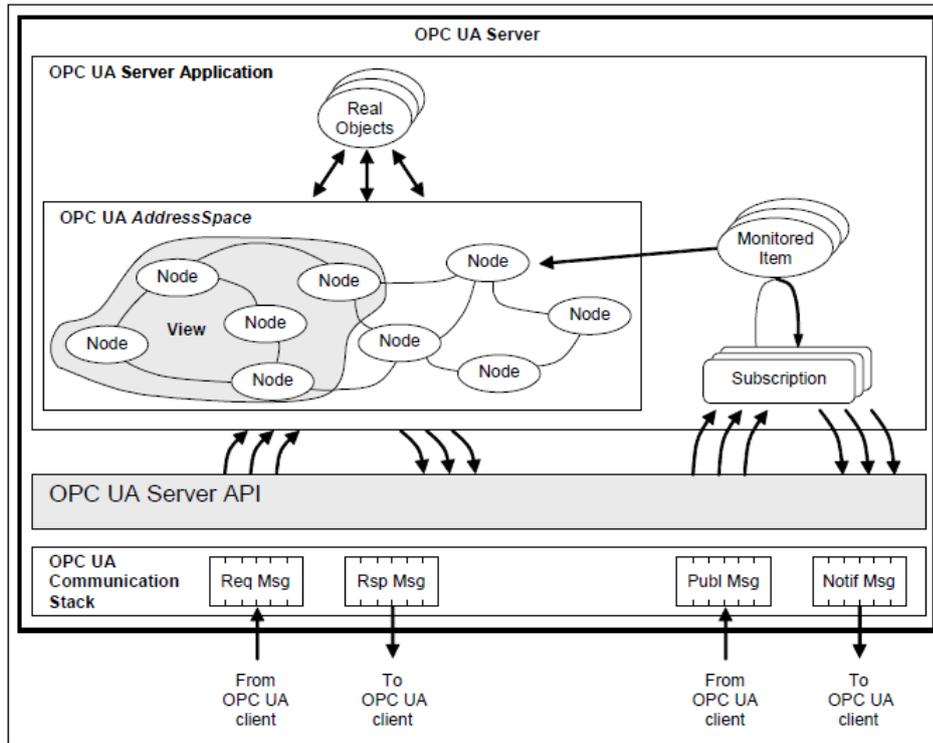


Figura 4.6: Architettura di un OPC UA *server* [29].

OPC UA *Server Application*

L'applicazione *server* è il codice che implementa le funzioni di *server*. Usa la *server API* per spedire e ricevere messaggi OPC UA dai *client*. Anche in questo caso la *server API* è un'interfaccia interna di separazione tra l'applicazione *server* e lo *stack* di comunicazione OPC UA.

Real objects

I cosiddetti oggetti reali sono oggetti fisici o software che sono accessibili dall'applicazione *server*. Esempi possono essere i dispositivi fisici o contatori diagnostici.

OPC UA *Address Space*

L'*address space* è modellizzato come un set di nodi accessibili dai *client* attraverso i servizi OPC UA (interfacce e metodi). I nodi nell'*address space* sono usati per rappresentare gli oggetti reali e le loro interazioni. I *server* sono liberi di organizzare i loro nodi all'interno dell'*address space*. L'uso dei riferimenti tra i nodi permette ai *server* di organizzare l'*address space* in gerarchie o qualunque architettura si desidera.

Address Space views

Una vista è un sottoinsieme dell'*address space*. Le viste sono usate per restringere i nodi che il *server* rende visibili ai *client*, restringendo così la dimensione dell'*address space* che può essere soggetto ai servizi richiesti da essi. Le viste celano alcuni dei nodi o dei riferimenti dell'*address space*. I *client* possono sfogliare le views esistenti che molte volte sono gerarchiche e permettono una navigazione ad albero.

Monitored items

I monitored items sono entità nel *server* create dal *client* che monitora i nodi dell'*address space* e le loro controparti nel mondo reale. Quando viene rilevato un cambio di valore o occorre un evento o un allarme, essi generano una notifica che è trasferita al *client* attraverso una *Subscription*.

Nel capitolo successivo, viene calato questo concetto nel caso reale dell'applicazione sviluppata in cui diverse variabili di macchina sono monitorate e loro variazioni scatenano diversi comportamenti del software.

Subscription

Una subscription è un endpoint nel *server* che pubblica notifiche ai *client*. I *client* controllano la frequenza con cui la pubblicazione avviene spedendo *Publish messages*.

Ridondanza

OPC UA fornisce le strutture dati e i servizi grazie ai quali la ridondanza può essere raggiunta in maniera standard. Essa può essere usata per avere una certa tolleranza ai guasti e al bilancio del carico.

4.5.6 OPC UA – Tecnologia

Le principali tecnologie implementate da OPC UA per garantire interoperabilità tra i vari attori del protocollo sono: *Encoding*, *Secure Channel* e *Transport*.

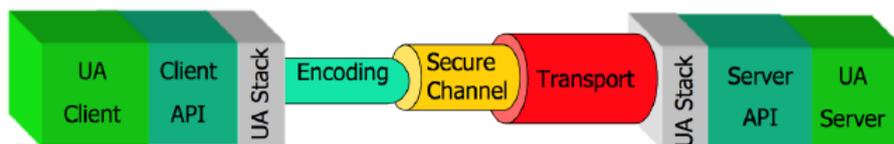


Figura 4.7: Rappresentazione della tecnologia del protocollo OPC UA [25].

Encoding

Basandosi sulle prestazioni di rete, OPC UA determina la soluzione ottimale per l’encoding tra XML e formato binario. Il formato XML può essere strutturato in diversi modi dipendenti dai dati, se invece le prestazioni della rete sono alte può essere usato il formato UA binario. Inoltre XML è usato per lo scambio dati tra diverse applicazioni.

Secure Channel

Quando l’informazione è codificata, è spedita automaticamente al livello successivo: il *Secure Channel*, utile per tenere al sicuro l’informazione. Il canale usa firme digitali, autenticazione e connessioni socket sicure. Tutto questo viene gestito da due protocolli: WS-Secure Conversation e UA-Secure Conversation.

Transport

La comunicazione in OPC UA si basa sulla messaggistica. OPC UA definisce due protocolli di trasporto: UA TCP e SOAP/HTTP, entrambi usati per instaurare una connessione tra *server* e *client*. UA TCP oltre a un header, presenta un corpo nel quale sono codificate le informazioni di connessione al socket e i dati veri e propri criptati. SOAP/HTTP è invece conosciuto come “*Firewall Friendly*”. HTTP usa TCP, SOAP invece combina la tecnologia HTTP con documenti XML. Questo, ancora una volta, garantisce interoperabilità tra diverse applicazioni e piattaforme. [25]

Capitolo 5

Motion SQL Connector

5.1 Motivazioni

Negli ultimi anni, nel mondo dell'automazione, è sempre crescente la necessità di storicizzare i dati provenienti dalle macchine e, più in generale, dall'intero processo produttivo.

A causa del grande numero di protocolli di comunicazione presenti sul mercato, legati alle diverse marche dei PLC e dei Motion Controller, ci si imbatte spesso nella grande difficoltà di interfacciamento tra il campo e i server aziendali.

L'OPC UA, universalmente riconosciuto, fornisce questa interfaccia superando tutti i limiti di protocolli proprietari di comunicazione. Risulta quindi possibile collegare tutti i dispositivi che lo supportano in modo da rendere accessibili i dati.

L'Industria 4.0, come descritto in 4.1, prevede uno scambio bilaterale di informazioni tra le macchine e i software gestionali o database.

Sono presenti sul mercato già diverse soluzioni di supervisione per ovviare a questi requisiti e poter certificare le macchine compatibili con i nuovi dettami. Lo scoglio più grande, per una gran numero di aziende, è il costo dei software classici di supervisione.

L'azienda Motion Engineering, specializzata in automazione e supervisione industriale, presso la quale ho frequentato il tirocinio curriculare e svolto la presente tesi, ha colto l'esigenza di molti clienti di accedere alle nuove tecnologie senza dover affrontare un investimento eccessivo.

L'azienda, come argomento di studio e di sviluppo, mi ha dato la possibilità di realizzare un'applicazione che, senza riprodurre le svariate funzionalità di un supervisore, avesse i requisiti minimi per poter certificare un sistema produttivo che la utilizza come *"Industry 4.0 compliant"*.

L'OPC UA Motion SQL Connector è un'applicazione sviluppata in **C#** che, installata e configurata, permette da una parte, il collegamento in OPC UA con

una macchina e dall'altra la storicizzazione su un Database (MsSQL o MySQL) delle variabili desiderate.

Inoltre, per soddisfare in modo esaustivo i requisiti dettati dall'Industria 4.0 permette anche di comandare alcune variabili della macchina direttamente dal Database.

In particolare, è previsto il cambio ricetta da gestionale configurando l'ID, il nome della nuova ricetta, il numero di pezzi da produrre con tale ricetta e eventualmente altre informazioni.

5.2 Scenari d'uso

Sempre più spesso, nelle realtà industriali, si nota un primo timido tentativo di integrazione tra sistemi IT e macchinari.

Oggi giorno sono disponibili sul mercato diverse soluzioni di *Data Collecting* dal campo, primo tra tutti l'OPC UA.

Molte realtà già conoscono questo tipo di approccio che sposta l'attenzione dall'automazione vera e propria a una visione più ampia, integrando supervisor e gestionali.

L'applicazione sviluppata è quindi un tentativo di fornire al cliente finale uno strumento con il quale, in modo automatico, interagire con un database aziendale, fino a questo momento utilizzato solamente dai sistemi gestionali. Inoltre, grazie alla comunicazione bidirezionale tra PLC e database, fornire un prodotto che sia al passo con i tempi e che aiuti a rientrare negli standard dettati dall'Industria 4.0 per accedere ai relativi vantaggi economici.

Con l'avvento della quarta rivoluzione industriale un gran numero di aziende ha rotto le barriere con l'innovazione tecnologica. Molte di loro, tra le più conservatrici, cercavano di mantenere inalterate le loro realtà: automazioni vecchie ma collaudate e funzionanti, sistemi di supervisione rudimentali o manuali, pannelli operatore di vecchio stampo.

Per le aziende più storicamente affermate è molto difficile fare un salto verso l'innovazione tecnologica che prevede incertezze, spese e complicazioni concettuali. Grazie alle agevolazioni che lo stato prevede per coloro che si cimentano nel rendere la propria realtà industriale più moderna, si sta notando entusiasmo e forte interesse verso le nuove tecnologie, anche da parte di realtà industriali storiche.

Gli incentivi per l'Industria 4.0, come descritto in 4.4, dettano specifiche regole per i singoli macchinari. Essi infatti devono scambiare informazioni con i sistemi informatici di livello superiore e essere identificati univocamente su una rete.

In molti casi si assiste già ad un invio di dati dalla macchina verso un supervisore o un gestionale per implementare soluzioni di supervisione più o meno complesse; questo però, malgrado a molti non sia ancora chiaro, non è sufficiente a garantire

una certificazione di conformità ai dettami dell'Industria 4.0.

Oltre a un invio di dati verso livelli informativi superiori, è imprescindibile uno scambio di informazioni anche nel verso opposto.

Gestionali o supervisori devono introdurre delle funzionalità che permettano ai reparti di produzione e logistica un invio di dati verso il campo.

Un esempio di questo requisito può essere la gestione di uno schedatore di ricette centralizzato. La ricetta di macchina è da intendere come l'insieme dei parametri di lavoro che permettono il funzionamento della stessa. Una macchina può avere più ricette, in funzione di quanti tipi di prodotto o di lavorazione siano presenti. Prima dell'avvento dei supervisori, l'insieme delle ricette era salvato sulla macchina e reso disponibile agli operatori per la selezione. Oggi questo compito, che spesso richiede l'intervento di un esperto, può essere gestito in modo centralizzato. Il responsabile di produzione può, per esempio, gestire le ricette di ogni singola macchina da un eventuale ufficio, comunicando quindi i dati dal supervisore alle macchine. Nel caso sopra descritto di un supervisore è naturale conseguenza anche il soddisfacimento del secondo requisito di identificazione univoca sulla rete di ogni macchinario.

Oggigiorno i supervisori SCADA stanno raggiungendo un grande numero di aziende, anche grazie agli incentivi economici di cui si è descritto. Non è però da sottovalutare che in molti casi di piccole o medie aziende, come già anticipato, il costo di un supervisore SCADA non sia comunque sostenibile, pur beneficiando degli sgravi economici dell'Industria 4.0. L'applicazione sviluppata ha come obiettivo proprio queste realtà aziendali. I servizi di scambio dati che il Motion SQL Connector mette a disposizione sono una versione semplificata e decisamente più economica di quelli offerti da un supervisore SCADA. Questo quindi non implica che tutte le necessità soddisfatte da un supervisore lo siano attraverso l'uso di questa applicazione. Il Motion SQL Connector garantisce uno scambio dati minimo tra un Database e una macchina in entrambe le direzioni ma sufficiente per aggiudicare la macchina come conforme all'Industria 4.0.

5.3 Panoramica dell'applicazione

L'applicazione è sviluppata in **C#** e, dopo la sua installazione, è configurabile tramite un file testuale. All'interno del file sono presenti i dati e gli indirizzi principali per la comunicazione e l'intera lista delle variabili di macchina che si desidera monitorare. Si è preferito inserire tutte le variabili tramite il file di configurazione rispetto a farlo mentre l'applicazione è già in esecuzione.

Il motivo di questa scelta va ricercato nel fatto che le variabili da monitorare e il loro numero è deciso a priori all'interno di un accordo con l'azienda cliente. Inoltre il percorso di queste variabili all'interno dell'albero OPC UA di Simotion non è di facile intuizione e per questo si è preferito dare accesso a un'applicazione

auto-configurante.

Il collegamento all'eseguibile è riposto nella cartella di avvio automatico in modo tale che venga eseguito a ogni riavvio del pc. Al momento dell'esecuzione, il file testuale viene caricato in memoria e l'applicazione tenta una connessione a intervalli regolari di tempo verso la macchina. L'intervallo di tempo è configurabile e adattato alla situazione desiderata, evitando così un inutile spreco di risorse nel caso in cui la macchina non venga collegata per un lungo periodo di tempo. Se invece la macchina può essere raggiunta, tutte le variabili desiderate verranno raggiunte tramite OPC UA e verranno elencate come variabili monitorate. L'applicazione ha due modalità di funzionamento: la prima con livello di privilegio utente e la seconda a livello di amministratore, riservata a Motion Engineering. Senza essere autenticati, il sistema oscura tutti i nomi delle variabili e non permette alcuna interazione con le stesse. Autenticandosi come amministratori, invece, i nomi delle variabili saranno in chiaro ed è possibile variarne il numero e la posizione.

Grazie alla sezione delle variabili monitorate è possibile quindi vedere tutti i loro valori in tempo reale. Tutte le variabili sono quindi storicizzate sul database con una cadenza regolata da alcune variabili *trigger* presenti nella medesima lista. È presente un secondo *trigger* che scatena la lettura dal database e la conseguente scrittura in macchina dei nuovi valori comandati.

La storicizzazione sul database di tutti i valori monitorati da una parte e la scrittura in macchina di valori comandati dal database dall'altra, costituiscono il raggiungimento delle norme necessarie a decorare la macchina come "*Industry 4.0 compliant*".

5.3.1 Layout

L'interfaccia grafica dell'applicazione ricalca gli standard delle applicazioni WPF. All'avvio del programma viene presentata all'utente la possibilità di collegarsi alla sorgente OPC UA con o senza autenticazione. Se avviene una connessione senza autenticazione, il sistema si collegherà alla macchina con il livello di privilegio più basso, nel quale non saranno disponibili in chiaro i nomi delle variabili monitorate ma solo i loro valori. Se invece viene inserita la password corretta per l'autenticazione come amministratore, il sistema renderà visibili tutti i nomi delle variabili. L'accesso con privilegi da amministratore è riservato ai tecnici della società fornitrice.

Nella zona inferiore della pagina è presente un testo di controllo, utile per notificare diversi stati in cui l'applicazione e i suoi servizi si trovano; inoltre, sono visualizzati tutti gli eventuali messaggi di errore o di allarme. Una volta che la connessione con la sorgente OPC UA è avvenuta con successo, viene visualizzato nella parte sinistra della finestra l'albero di oggetti di Simotion. La navigazione all'interno delle sottocartelle è consentita solo in modalità amministratore.

Nella parte inferiore dell'applicazione viene visualizzata l'intera lista delle variabili monitorate, correlate di diverse informazioni di diagnostica e controllo, primo tra tutti il valore in tempo reale. Nello specifico le colonne sono:

- *NodeId*: il nome della variabile, oscurato e sostituito con un generico *Field_i*
- *Sampling*: il periodo di campionamento della variabile
- *Value*: il valore in tempo reale della variabile
- *Quality*: la qualità di ricezione del dato
- *Timestamp*: la marca temporale dell'ultima variazione del dato
- *Last Error*: la stringa di descrizione dell'ultimo errore avvenuto per quella variabile, se presente

Nella parte superiore, in prossimità della zona di autenticazione è presente un *header* con diverse informazioni utili per l'utente. Come prima informazione viene visualizzato il numero di tentativi di riconnessione effettuati fino al momento presente. Il numero di tentativi cresce di un'unità ogni volta che il sistema tenta di riconnettersi alla macchina: questo può succedere sia durante un avvio dell'applicazione a macchina spenta o non raggiungibile, sia durante un periodo di disconnessione temporanea dalla macchina. Vengono inoltre visualizzati altri due contatori: le righe della tabella utilizzata come *logger* delle variabili monitorate e quelle della tabella utilizzata per comandare il successivo cambio ricetta della macchina. Questi due contatori sono utili all'utente per monitorare lo stato di memoria del database e per controllare che non venga comandata più di una ricetta per volta, situazione di errore che comunque viene monitorata e segnalata dall'applicazione mediante un messaggio di errore specifico.

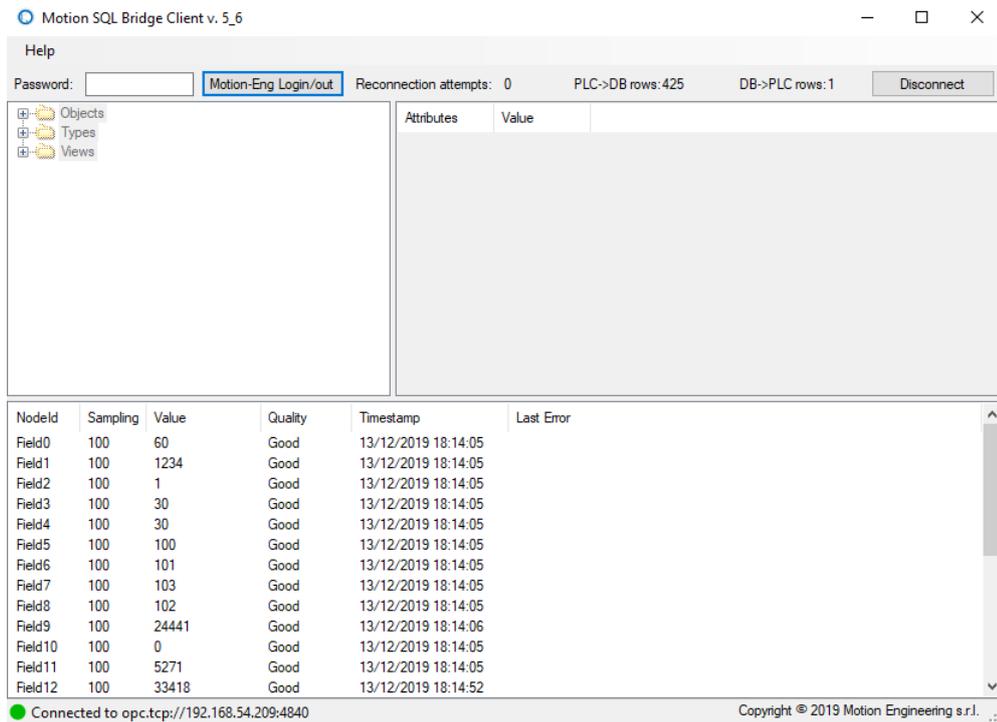


Figura 5.1: Layout dell'applicazione.

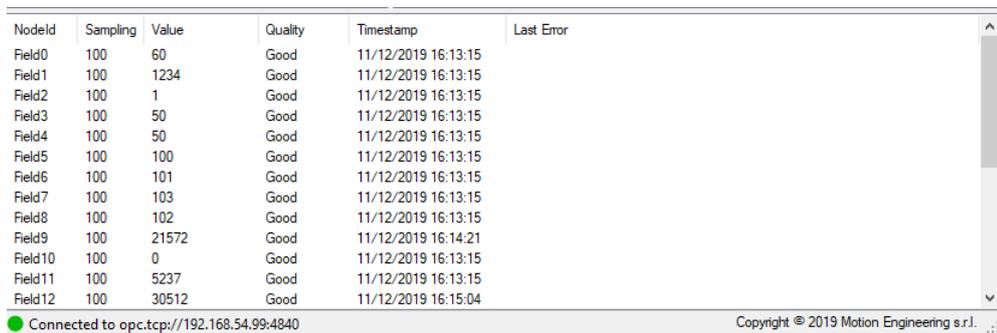


Figura 5.2: Lista delle variabili monitorate.

5.3.2 File di configurazione

Il file di configurazione viene caricato in memoria a ogni nuovo avvio dell'applicazione. Essa è stata pensata per essere sempre in esecuzione su una macchina virtuale o fisica dedicata del cliente. Per essere robusta a riavvii improvvisi o programmati viene configurata completamente tramite file di testo; infatti, non avendo a disposizione variabili ritentive come sui sistemi PLC, il file di testo rimane la soluzione più comoda e efficiente. Nel file sono contenute tutte le informazioni

per instaurare una connessione sia con il database che con la macchina.

Come requisito fondamentale per il corretto funzionamento del programma vi è una corretta gestione delle reti. In molte realtà aziendali la rete intranet viene suddivisa principalmente in due sezioni: la rete di campo e quella aziendale. Oggi, con l'avvento dell'Industria 4.0, c'è la necessità di far interagire queste due sottoreti, che fino a poco tempo fa potevano rimanere isolate. Questo comporta uno sviluppo della gestione IT aziendale per garantire efficienza e sicurezza.

L'applicazione quindi va installata su un computer in rete aziendale che però ha accesso alla rete di campo per poter raggiungere il server OPC UA installato su Simotion. Nel file testuale sono anche contenute le informazioni riguardanti le variabili che, tramite il protocollo OPC UA, verranno monitorate. Il file è organizzato nel seguente modo:

- Le informazioni del database come nome, indirizzo, istanza, credenziali d'accesso, porte e nomi delle tabelle
- Numero di variabili da monitorare
- Stringa di connessione OPC UA
- Stringa criptata
- Variabili da monitorare in lettura e scrittura (le seconde indicate con la sequenza di caratteri jolly "#W#").

La stringa criptata è una sequenza di caratteri alfanumerici generata randomicamente con al suo interno codificata la matricola della macchina dalla quale si vogliono estrarre le variabili. Questo preclude al cliente la possibilità di prelevare i dati da una macchina diversa da quella prevista. Nel caso non ci sia coerenza tra la matricola prevista e quella monitorata, il sistema continua a monitorare i valori delle variabili ma viene meno lo scambio dati con il database, in entrambi i versi.

Nell'attuale versione dell'applicazione sono presenti due tabelle di scambio dati: una per il Log dei dati sul database e l'altra, che contiene solo una riga, per l'invio della successiva ricetta da produrre sulla macchina.

Le variabili di Simotion sono localizzate all'interno di un sistema gerarchico di cartelle standard per tutte le macchine dotate di questo controllore. Le variabili da monitorare sono quindi da inserire nel file testuale con il path corretto: solo in questo modo saranno raggiungibili dal *client* OPC UA dell'applicazione. Anche per questo motivo, la configurazione del file di testo è riservata all'azienda fornitrice.

```

//-----//
// DB info [ip,port,name,user,password,PLC->DB table,DB->PLC table] //
//-----//
192.168.0.105\\SQLEXPRESS,1434,OpcDb,MotionAdmin,*****,opcRead,opcWrite

//-----//
// Number of variables, triggers and writable included //
//-----//
17

//-----//
// OPC UA Server address [opc.tcp://ip:port] //
//-----//
opc.tcp://192.168.0.100:4840

//-----//
// Key String //
//-----//
$$4a3e333f84b1a4fc852486ad0610e245b0f55ff393c5b9c6362ee165359596850$$

//-----//
// Variables (the last three are in order iErrorCode, DB->PLC trigger var,PLC->DB trigger var
//-----//
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsMinuteMachineAutoRunning
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsMaterialCounter1
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsMaterialCounter2
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsMaterialCounter3
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsMaterialCounter4
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsCycles
ns=2;s=unit/Def_Supervisor.Gst_Machine.stCounters.udiAbsPowerCounter
ns=2;s=unit/Def_Glob.G_stOPC.stData.iSerialNbr
ns=2;s=unit/Def_Glob.G_stOPC.stData.sMachineName
ns=2;s=unit/Def_Glob.G_stOPC.stReceive.stRecipe.sArticle #W#
ns=2;s=unit/Def_Glob.G_stOPC.stReceive.stRecipe.sOrderNr #W#
ns=2;s=unit/Def_Glob.G_stOPC.stReceive.stRecipe.sPaperLotNbr #W#

```

Figura 5.3: Porzione di un esempio di file di configurazione.

5.3.3 *Trigger* di processo

Per garantire al cliente finale la maggior flessibilità possibile, si è deciso di non imporre tempistiche fisse e decise a priori.

Le due funzionalità principali dell'applicazione: quella di Log sul database e quella di segnalazione della nuova ricetta, sono entrambe degli eventi legati alla variazione di due variabili di macchina. Scegliere a priori il periodo di esecuzione di questi due eventi sarebbe stato limitante e soprattutto avrebbe reso l'applicazione dipendente dal settore industriale in cui viene installata. Diverse aziende, infatti, hanno tempistiche di lavoro ed esigenze di risposta molto diverse tra loro.

Motion SQL Connector è un'applicazione che può essere installata su una popolazione molto eterogenea di realtà industriali. Ci sono variabili monitorate come velocità e temperature, che hanno bisogno di un'elevata velocità di supervisione mentre altre, come variazioni di parametri o quantità secondarie al processo, che possono essere controllate con cadenza decisamente inferiore. Queste diverse necessità di risposta sono quindi garantite dalla configurazione di due variabili, chiamate *trigger*, modificabili sul pannello operatore della macchina.

Tra le variabili monitorate e quindi elencate nel file di testo, nelle ultime posizioni, se ne annoverano due che ricoprono proprio questo scopo: il *trigger* di scrittura su macchina della nuova ricetta (da DATABASE a PLC) e il *trigger* di scrittura sul database (da PLC a DATABASE). Queste due variabili sono dei booleani che scatenano le funzioni di lettura/scrittura a ogni variazione; il periodo con il quale vengono comandate è, come già annunciato, configurabile dal pannello operatore. Il primo *trigger* salva in macchina su apposite variabili il nome e l'ID della nuova ricetta, eventualmente il lotto, l'ordine e i pezzi da produrre con tale ricetta. Per evitare ambiguità, la tabella OpcWrite deve contenere al più una riga. Il secondo *trigger*, invece, scatena una lettura istantanea di tutti i valori delle variabili monitorate e li salva sotto forma di una nuova riga nella tabella OpcRead, alla quale viene aggiunto un timestamp. Sull'applicazione è presente il numero di righe di entrambe le tabelle Opc per controllarne la dimensione.

ns=2:s=unit/UN_Global_Def.UN_stOPC.stClock.xRXSync	100	True	Good	11/05/2020 17:39:28
ns=2:s=unit/UN_Global_Def.UN_stOPC.stClock.xTXSync	100	False	Good	11/05/2020 17:39:34

● Connected to opc.tcp://192.168.0.100:4840

Figura 5.4: Cattura di due variabili *trigger* durante l'esecuzione.

5.3.4 *Heartbit* di segnalazione alla macchina

Nel mondo dell'automazione industriale viene seguito uno standard oramai affermato da tempo riguardo all'*handshaking* di comunicazione tra macchine e i più semplici sistemi di supervisione collegati ad esse. Viene utilizzata una variabile, spesso booleana, chiamata *heartbit* che, dalla macchina o dal sistema soprastante, viene continuamente cambiata di stato. Nel momento in cui l'interlocutore si accorge della permanenza dello stato di questa variabile per più di un tempo di timeout configurato, viene subito segnalato un problema di comunicazione. È stato quindi introdotto un meccanismo simile nell'applicazione. All'interno della lista di variabili da monitorare, scritte nel file di configurazione, oltre ai *trigger* e alla variabile di errore è presente la variabile di *heartbit*. Questa viene scritta dal programma in ogni scansione di scrittura delle variabili del PLC. In particolare, viene cambiata di stato a ogni scrittura in modo tale che la macchina veda continuamente una sua variazione. Nel momento in cui il programma di macchina non rilevi più la sua continua variazione, viene notificata al pannello la perdita di connessione con l'applicazione. La causa di questa anomalia sarà quindi da ricercare nella connessione di rete o nello stato della virtual machine su cui l'applicazione è installata. È stata quindi aggiunta questa funzionalità per migliorare la robustezza del sistema stesso. È utile infatti che l'operatore a bordo macchina venga avvisato della perdita di connessione con l'applicazione in quanto fino all'eventuale riconnessione non

verranno più storicizzati i dati sul database. Inoltre, grazie alla segnalazione sul pannello di macchina, l'operatore ha la possibilità di accorgersi che l'eventuale ricetta comandata potrebbe essere un'informazione scaduta e non più attendibile.

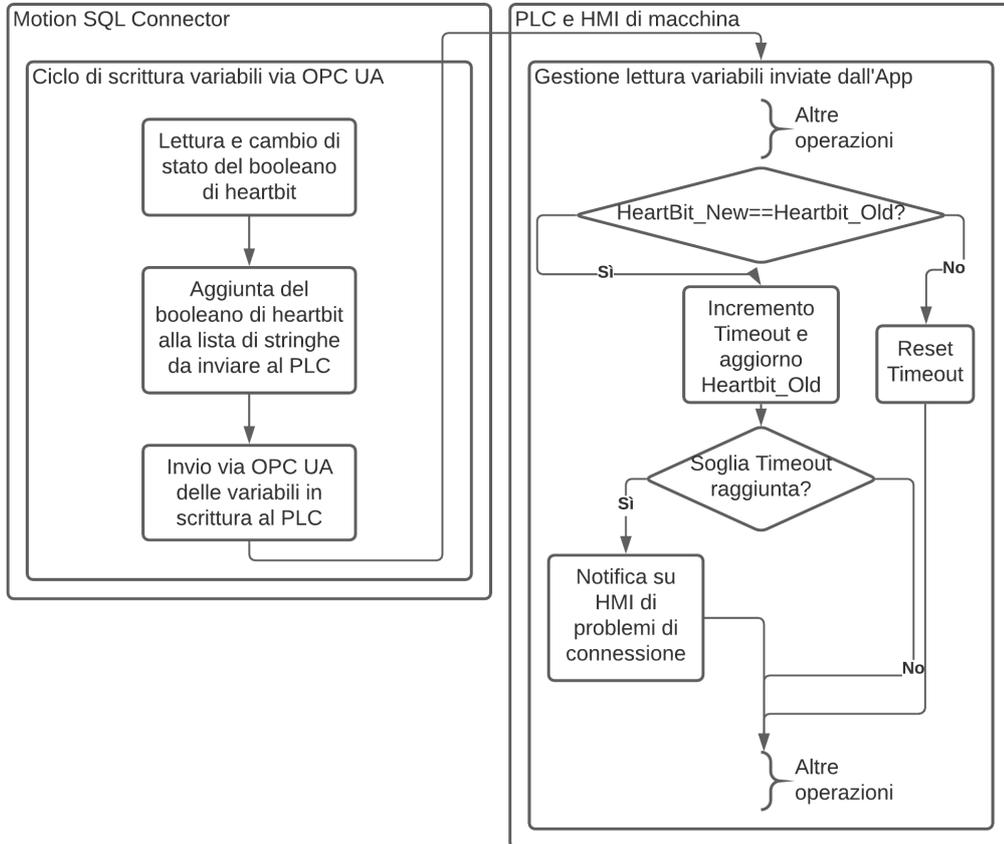


Figura 5.5: Schema di funzionamento dell'*heartbeat* tra applicazione e macchina.

5.3.5 Gestione errori

Per gestire gli errori relativi al processo, viene usata una variabile che viene scritta con un valore utile a definire lo stato attuale dell'applicazione. Per errori non si intende eccezioni software, quanto situazioni in cui la macchina e il sistema in generale si vengono a trovare, che impedirebbero il corretto funzionamento dell'applicazione o ne inibirebbero una parte. Questa variabile è configurata, alla stregua delle altre, attraverso il file di testo e viene sempre posta per convenzione appena sopra alle due variabili *trigger*. La variabile di errore è un intero che può avere diversi valori a seconda dello stato o errore:

- 0: l'applicazione è in esecuzione e non ha alcun errore. In questo stato è possibile che la ricetta in macchina sia coerente con quella comandata dal gestionale oppure che ci sia incongruenza. Nel caso di incongruenza viene segnalata a pannello la nuova ricetta comandata e eventualmente i pezzi da produrre con tale ricetta.

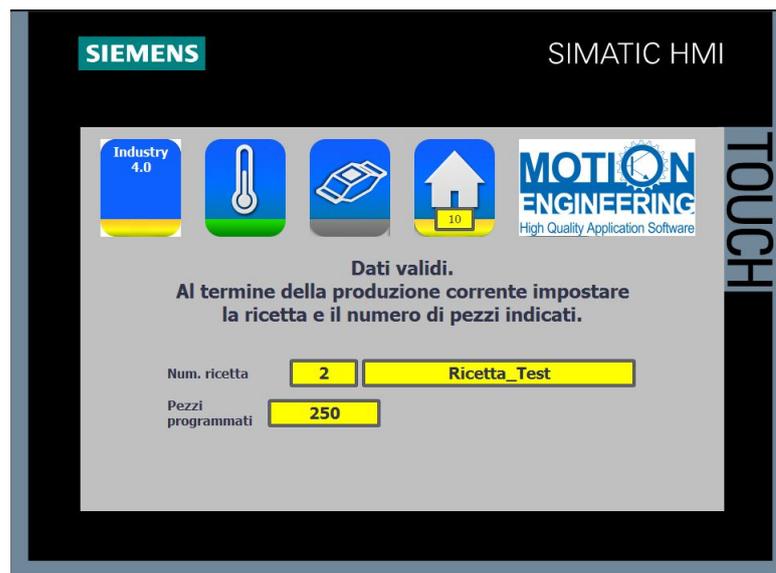


Figura 5.6: Esempio di esecuzione senza alcun errore e visualizzazione della ricetta da produrre.

- -1: l'applicazione è in esecuzione ma la tabella di DB → PLC ha più di una riga e quindi è segnalata ambiguità sulla prossima ricetta. In questa situazione viene temporaneamente disabilitata la funzione di scrittura in macchina fino al momento in cui le righe in questa tabella si riducono a una soltanto.

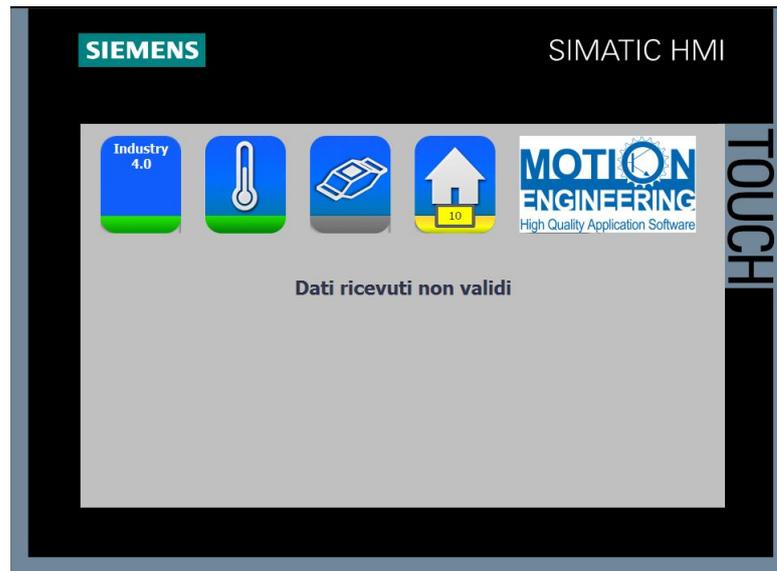


Figura 5.7: Popup di errore associato al codice -1.

- -2: l'applicazione è in esecuzione ma la tabella DB → PLC è vuota, non si hanno quindi informazioni riguardo al cambio ricetta, viene temporaneamente disabilitata questa funzione in attesa che si immetta una riga.

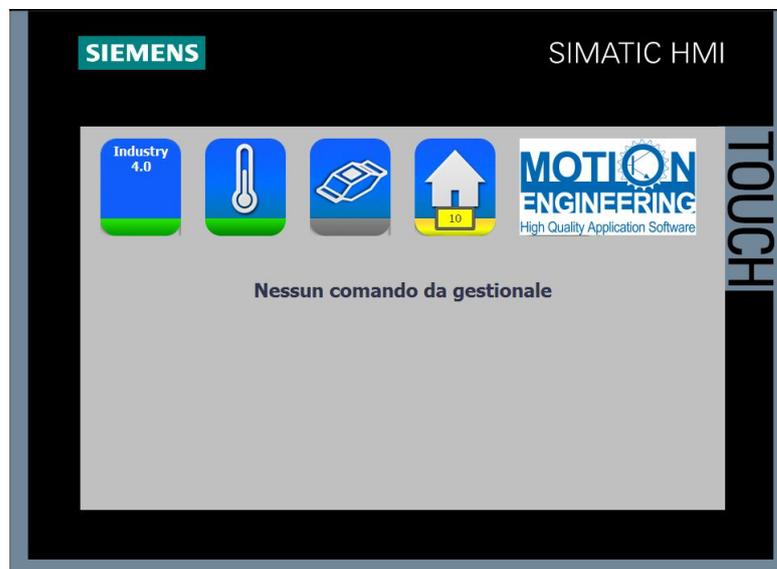


Figura 5.8: Popup di errore associato al codice -2.

- -3: l'applicazione è stata installata e connessa a una macchina non riconosciuta.

In questo stato l'applicazione continua a funzionare come OPC client inibendo l'interfacciamento con il database. Grazie al fatto che la variabile di errore sia anch'essa una delle variabili di macchina, è possibile informare gli operatori in produzione degli eventuali errori presenti. All'interno del software di macchina e di HMI è possibile quindi programmare dei popup informativi come reazioni ai valori delle variabile in questione. Informando direttamente l'operatore attraverso il pannello di macchina viene meno la necessità di controllare l'applicazione, rendendo il processo più automatico.



Figura 5.9: Popup di errore associato al codice -3.

5.3.6 Connessione automatica

Grazie all'inserimento dell'applicazione nella cartella di startup automatico, come già descritto in 5.3, avviene un avvio automatico a ogni reboot dell'ambiente su cui è installata. Questo meccanismo protegge da eventuali riavvii previsti, e non, del sistema operativo ma lascia comunque scoperta l'eventualità di macchina spenta durante l'avviamento dell'applicazione. Per questo motivo è stato introdotto un meccanismo di riconnessione automatica.

Una volta avviato l'applicativo, viene controllata a intervalli regolari la disponibilità della macchina alla connessione OPC UA. Fino a quando essa non risulta presente, viene ritentata la connessione. Si è optato per una soluzione *polling* con periodo di interrogazione piuttosto lungo per non appesantire il sistema.

5.3.7 Gestione timeout di connessione e riconnessione automatica

Essendo un'applicazione in stretta collaborazione con un macchinario industriale, sono necessari dei meccanismi di risposta a eventuali perdite di connessione o, in generale, a tutte le problematiche che presentandosi, andrebbero a interrompere, anche momentaneamente, il collegamento Ethernet con la macchina. Per ovviare a queste situazioni è stato creato una porzione di codice eseguita nel continuo in background, utile per controllare lo stato della connessione tra macchina e applicazione. Grazie a un meccanismo di controllo del timeout di comunicazione viene monitorato lo stato del collegamento tra i due sistemi. Nel caso in cui venga rilevato un mal funzionamento, dopo un tempo di timeout configurabile, viene effettuato un riavvio della sessione. Nel caso in cui esso non sia possibile viene ritentato un collegamento a intervalli regolari e configurabili fino alla sua buona riuscita. Grazie a queste funzionalità introdotte nel codice dell'applicazione si ha quindi una tolleranza più alta ai problemi di connessione, frequenti in ambiti industriali per svariati motivi, anche solo per macchina spenta. La gestione automatica di tutte le problematiche collegate porta l'applicazione a un livello di automazione superiore, evitando attenzione continua da parte degli operatori.

5.3.8 *Logging* delle variabili monitorate

Tutte le variabili monitorate dall'applicazione vengono scritte sulla cartella dedicata al *logging* a periodi di tempo regolari e cadenzati dal *trigger*. Nel mondo della supervisione, è di cruciale importanza storicizzare i dati per permettere l'accesso e la manipolazione a posteriori.

La reportistica di impianto, la manutenzione predittiva e i trend di macchina sono solo poche tra le molte applicazioni che le aziende moderne richiedono per migliorare la loro gestione aziendale e il loro prestigio.

L'implementazione di una storicizzazione su un database sviluppata nell'applicazione non è quindi da vedere come fine a se stessa; è possibile infatti rendere disponibile la tabella a programmi di reportistica o supervisione per analisi dettagliate dei dati.

5.3.9 Invio della ricetta remota

Soprattutto per grandi realtà industriali, l'invio delle ricette da remoto sta diventando una priorità. L'applicazione è quindi in grado di informare gli operatori di macchina del prossimo cambio di ricetta da eseguire. Nel file di configurazione, come descritto, sono presenti alcune variabili in scrittura come nome e id della ricetta e pezzi da produrre con tale ricetta.

L'invio di una ricetta è solo una delle possibilità di integrazione dei sistemi gestionali con i reparti di produzione, in questa trattazione verrà presa come esempio. Nel database aziendale, oltre alla tabella di *logging*, è presente una tabella utile alla notifica alla macchina della prossima ricetta da produrre. Questa tabella deve contenere al più una riga per evitare ambiguità e essere sempre nella situazione di poter notificare l'operatore con la ricetta desiderata. Oltre al numero univoco di ricetta e alla stringa che ne definisce il nome sono presenti altre informazioni come la quantità di pezzi previsti con quella ricetta. In generale, è possibile corredare la ricetta con tutte le informazioni che il cliente desidera. Sul fronte del *trigger* dedicato, avviene la lettura del record sul database e la conseguente scrittura in macchina dei valori. Come evidenziato in figura 5.6, nel caso in cui la ricetta attualmente operativa in macchina sia diversa dalla ricetta comandata dal database, gli operatori sono prontamente avvisati tramite un popup sul pannello del prossimo cambio di ricetta, corredato di tutte le informazioni utili. Se, al contrario, il cambio di ricetta è già avvenuto e dal database non proviene ancora la nuova ricetta da produrre, all'operatore verrà notificata congruenza di ricetta.

Field8	100	WBM17 C492	Good	15/05/2020 18:12:52
Field9	100	Articolo_Test	Good	15/05/2020 18:12:52
Field10	100	Ordine_Test	Good	15/05/2020 18:12:52
Field11	100	Lotto_Test	Good	15/05/2020 18:12:52
Field12	100	1500	Good	15/05/2020 18:12:52

Figura 5.10: Esempio di ricetta comandata da remoto corredata di alcuni parametri utili.

5.4 Processo e sviluppo dell'applicazione

5.4.1 Introduzione

L'applicazione, sviluppata in C#, segue il concetto di architettura modulare, presentando diversi file. In questa sezione verrà presentata una panoramica delle principali funzioni software implementate all'interno del sorgente. In particolare verranno trattate: la gestione dei database, le funzioni adibite al caricamento in memoria del file di configurazione con le relative informazioni, l'avvio dell'applicazione, il ciclo principale di processo, la gestione degli errori, il Log sul database, il comando della successiva ricetta da produrre, l'*heartbeat* di comunicazione e i meccanismi di riconnessione automatica.

Per quanto riguarda la gestione dell'OPC UA è stata utilizzata la libreria "MIT License 1.00", distribuita direttamente da OPC Foundation e disponibile per qualunque sviluppatore voglia integrare funzioni OPC UA nelle proprie applicazioni. Non è argomento di questa tesi approfondire il codice di suddetta libreria, lasciando invece spazio alle funzionalità implementate e sopra citate. Si riporta in figura 5.11 un *flowchart* per schematizzare la prima parte del processo dell'applicazione. In particolare le fasi che seguono l'avvio della stessa fino al compimento del caricamento delle variabili desiderate. Nella suddetta figura si possono evincere i contributi dei due *thread*, eseguiti in *background*, adibiti alla connessione e alla riconnessione automatiche.

Verrà in seguito presentato un ulteriore *flowchart* per quanto riguarda la parte centrale dell'applicazione, adibita al controllo della variazione dei valori e ai processi implicati.

5.4.2 Gestione database

L'applicazione è al momento compatibile con due database: MySQL e Microsoft SQL Server. La struttura del codice è comunque favorevole ad un'eventuale futura integrazione di ulteriori database come Oracle o altri.

Per quanto riguarda l'integrazione del database MySQL è stata importata la libreria `MySql.Data`, mentre per quanto riguarda il database MSSQL è stata utilizzata la libreria di sistema `System.Data.SqlClient`.

Per ogni motore database è stata creata una classe adibita alle funzioni base di interazione con il esso. In particolare la classe `GestioneMySQL` per l'interfaccia con MySQL e `GestioneMsSql` per quella con Microsoft SQL Server.

Sono state quindi implementate, all'interno delle rispettive classi, le funzioni base di interfacciamento con i database come connessione e disconnessione.

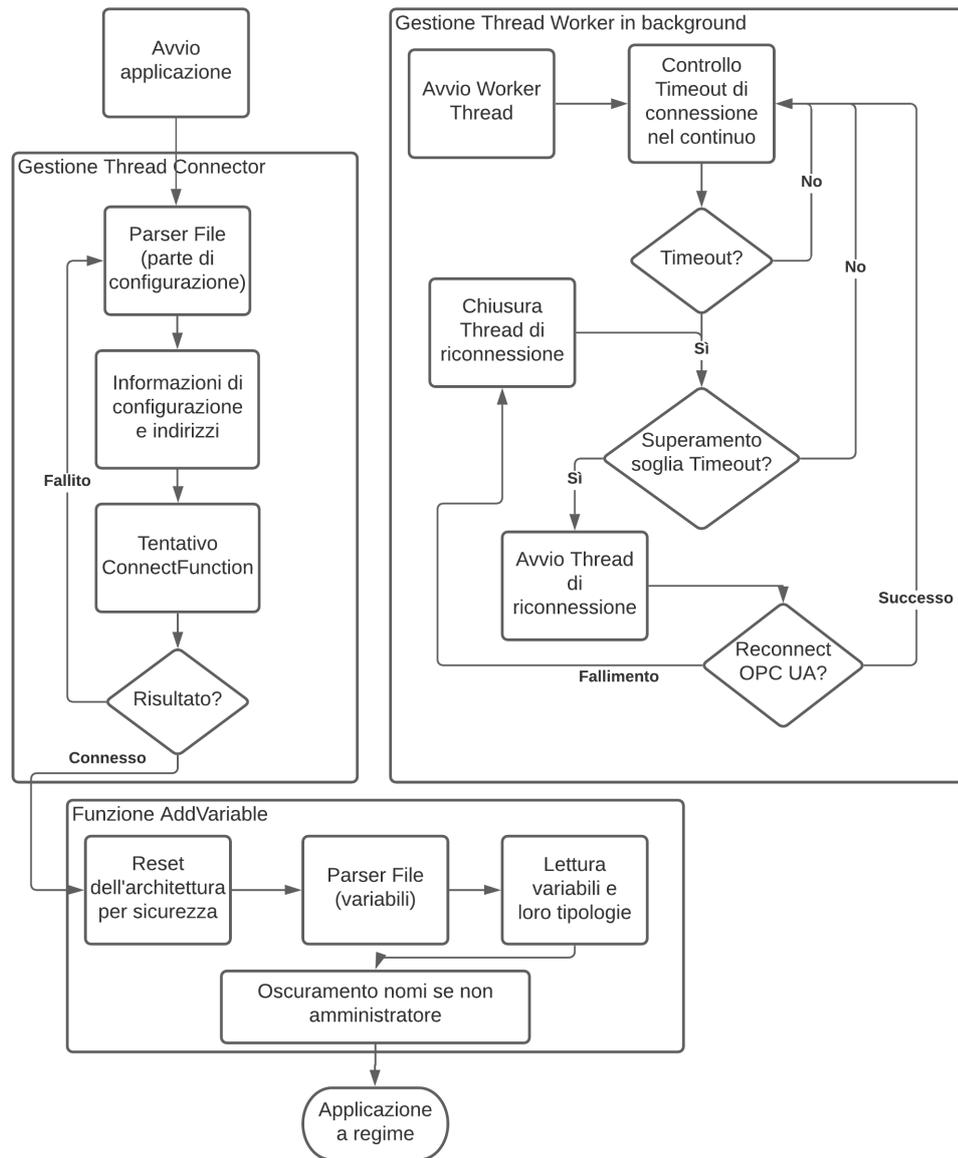


Figura 5.11: Flowchart delle fasi di avvio dell'applicazione.

5.4.3 Avvio dell'applicazione

Nel metodo costruttore del form principale sono presenti alcune inizializzazioni di variabili e oggetti della libreria OPC UA. Tra le varie operazioni di secondaria importanza c'è l'esecuzione di un thread chiamato `Connector`, utile per tentare di stabilire la connessione con la macchina. La disponibilità alla connessione da

parte della macchina in reparto non è infatti garantita nel momento dell'avvio dell'applicazione.

L'applicazione deve essere completamente indipendente dalle tempistiche di macchina in quanto, molto spesso, gli orari di lavoro e di gestione possono essere molto diversi tra reparti IT e ambienti produttivi.

L'applicazione quindi gestisce dei tentativi di connessione a intervalli regolari e piuttosto distanziati nel tempo. Tramite una chiamata asincrona, gestita ad alto livello con una `RunWorkerAsync`, viene eseguito un thread secondario con il compito di eseguire i tentativi di connessione attraverso l'esecuzione della funzione `connector_DoWork`. Tale funzione chiama la `ConnectFunction`, *wrapper* della `Connect` vera e propria. Questo doppio passaggio, a prima vista ridondante, è utile perchè il `Connector` è, come descritto, un thread secondario; risulta quindi necessario passare da un `Invoke` per chiamare la `Connect` del thread primario. Nel caso di tentativo fallito, dopo una certa tempistica configurabile, il `Connector` tenta un'altra connessione fino al successo con conseguente ritorno del thread.

5.4.4 Funzione di connessione

Nel momento in cui la macchina risulti raggiungibile da parte dell'applicazione, il thread di connessione `Connector` completerà il lavoro chiamando la funzione `Connect`. Una volta avviata la procedura di connessione, il thread in background `Connector` cesserà di esistere e ritornerà in maniera gestita. La funzione di connessione carica il file di configurazione in memoria. Per quanto riguarda la posizione di tale file è stato previsto che il file testuale venga salvato allo stesso livello dell'eseguibile, in questo modo attraverso i metodi della classe di sistema `Directory` è possibile svincolarsi da qualunque *path* assoluto, prelevando i file direttamente nella cartella dove risiede l'eseguibile dell'applicazione. Una volta aperto il file viene eseguito un *parser* per estrapolare tutti i dati necessari alla parte di configurazione. Il file viene letto riga per riga e vengono salvate in memoria le informazioni relative al database, all'OPC UA e al numero di variabili. Viene inoltre decodificata la stringa alfanumerica, utile per segnalare situazioni non previste dall'applicazione. Al termine della lettura del file, viene aperta la connessione OPC UA con la macchina tramite la funzione `Connect` di sistema della libreria OPC UA. Nel caso in cui tutte le operazioni sopra citate siano state completate, viene chiamata la funzione adibita all'inserimento in memoria di tutto il listato di variabili da monitorare. Infine, subito dopo aver caricato tutte le variabili in memoria, viene eseguito, se non ancora in esecuzione, un thread chiamato `Worker`, adibito alla riconnessione automatica dopo un'eventuale perdita di connessione con la macchina causata da svariate ragioni.

5.4.5 Thread Worker in background

Nelle realtà industriali non è raro che avvengano delle disconnessioni improvvise dei macchinari, anche solo per manutenzioni o per spegnimento. L'applicazione, non essendo costantemente controllata da operatori, deve essere robusta a ogni casistica possibile; è stato quindi introdotto un meccanismo di riconnessione automatica in caso di perdita di connessione con il campo. Per poter ovviare a questo problema è stato implementato un secondo thread in esecuzione in background oltre al Connector.

Questo thread è stato chiamato `Worker` e viene eseguito per la prima volta al termine senza errori della funzione di connessione, dettagliata nella sezione precedente. Il corpo della funzione in esecuzione da questo thread è continuamente attivo e controlla due marche temporali.

Chiave per l'implementazione di questo controllo è la variabile `LastKeepAlive`, una delle proprietà utili per diagnosticare lo stato della sessione OPC UA instaurata con il Simotion D. In particolare, viene confrontato, in modo asincrono e su un thread separato, il suo valore con quello del ciclo precedente, controllandone quindi la progressione temporale. Nel caso in cui le due marche temporali coincidano, viene incrementato un contatore di timeout, in caso contrario viene resettato il suddetto contatore. Il contatore di timeout è configurabile a un valore scelto a priori che sia significativo per la particolare applicazione.

Diverse realtà industriali potrebbero avere diversi tempi di timeout che indicano un'effettiva perdita di comunicazione tra due sistemi. Quando questo contatore raggiunge la soglia preconfigurata si presuppone un problema di connessione e viene creato un nuovo thread adibito al tentativo di riconnessione. Il thread viene configurato con una *lambda expression* nella quale, dopo opportune verifiche, viene programmata una riconnessione invocando la `ConnectFunction`, *wrapper* della `Connect` vera e propria.

Se il collegamento alla rete risulti ancora assente, viene eseguita una disconnessione forzata, ristabilendo le condizioni per il prossimo tentativo di riconnessione. Viene quindi eseguito il thread e atteso il suo ritorno entro un tempo prefissato di tre secondi, giudicato sufficiente per un'esecuzione corretta senza errori della funzione assegnata ad esso. Nel caso in cui non ci sia ancora una connessione stabile tra la macchina e l'applicazione, la *join* non avrà successo. In questo caso si attiva una parte di codice relativa ai tentativi di riconnessione, incrementandoli di un'unità ogni fallimento della *join*. Questa situazione viene risolta nel momento in cui la connessione ritorni disponibile.

5.4.6 Caricamento variabili in memoria

Come prima operazione viene richiamata una procedura di inizializzazione di tutte le variabili in memoria, utile per iniziare il processo di configurazione da

un'architettura nota. Viene quindi eseguito un *parser* del file di testo per estrapolare tutte le variabili desiderate.

Le variabili possono essere semplici variabili in lettura dalla macchina che verranno storicizzate sul database oppure variabili in scrittura, identificate attraverso la sequenza di caratteri "#W#", posta dopo il *path* assoluto della variabile. All'interno della medesima lista, per convenzione poste in ultima posizione, ci sono le variabili utili al processo: l'*heartbeat*, i due *trigger* e la variabile di errore. Segue quindi una sezione di codice di gestione delle varie tipologie di variabili monitorate in base alla loro funzione. Viene posta particolare attenzione alla variabile che identifica la matricola della macchina, infatti verranno eseguiti dei controlli su quest'ultima per evitare che l'utente finale installi l'applicazione per il controllo di una macchina non prevista in sede di contratto. Come già descritto in 5.3.5, è presente un errore specifico per notificare questa situazione di anomalia. Come ultima operazione della funzione vengono oscurati tutti i nomi delle variabili monitorate nel caso di privilegi non sufficienti a visualizzarli. Vengono quindi eseguite delle funzioni adibite a nascondere le stringhe delle variabili, sostituendole con dei nomi fittizi del tipo `Fieldz`. Quest'operazione viene eseguita per tutte le variabili presenti sul file di testo, anche quelle di *trigger* e di errore. Nel caso in cui i privilegi di accesso nel momento dell'avvio dell'applicazione siano invece sufficienti per la visualizzazione totale delle stesse, questa sezione non verrà eseguita.

La possibilità di accedere come amministratori viene comunque lasciata solamente a tecnici designati.

5.4.7 Funzione di gestione del cambio di valore

Tutte le variabili caricate in memoria dal file di configurazione sono inserite in apposite strutture di `MonitoredItem`. La libreria OPC UA mette infatti a disposizione questa classe, utile a gestire delle variabili soggette a un controllo costante da parte dell'applicazione. La funzione di gestione del cambio di valore, accettando due parametri in ingresso, è utile proprio a questo scopo. In particolare riceve come ingressi un oggetto di tipo `MonitoredItem` e l'evento di variazione di classe `MonitoredItemNotificationEventArgs`. All'interno della definizione della suddetta funzione viene gestito l'intero processamento scatenato dalla variazione di una qualunque delle variabili. Il puntatore all'oggetto `MonitoredItem` viene salvato in una variabile, generalmente definita `Item`. La funzione viene eseguita a ogni cambio di valore di una qualunque delle variabili monitorate, è stato quindi necessario introdurre un meccanismo di riconoscimento delle variabili utili al processo anche all'interno di questa procedura. Una volta salvato l'oggetto nella variabile generica `Item`, grazie a una semplice comparazione del testo del *tag* che ha generato la variazione con il nome delle variabili più significative del processo, è possibile identificare ogni tipologia di variabili e gestirle in maniera adeguata. La

stringa utilizzata per la comparazione è contenuta nella variabile oggetto Item, in particolare accedendo a `item.SubItems[0].Text`.

Segue una descrizione delle varie sezioni della procedura adibite alle singole tipologie di variabili e un flowchart riassuntivo in figura 5.12.

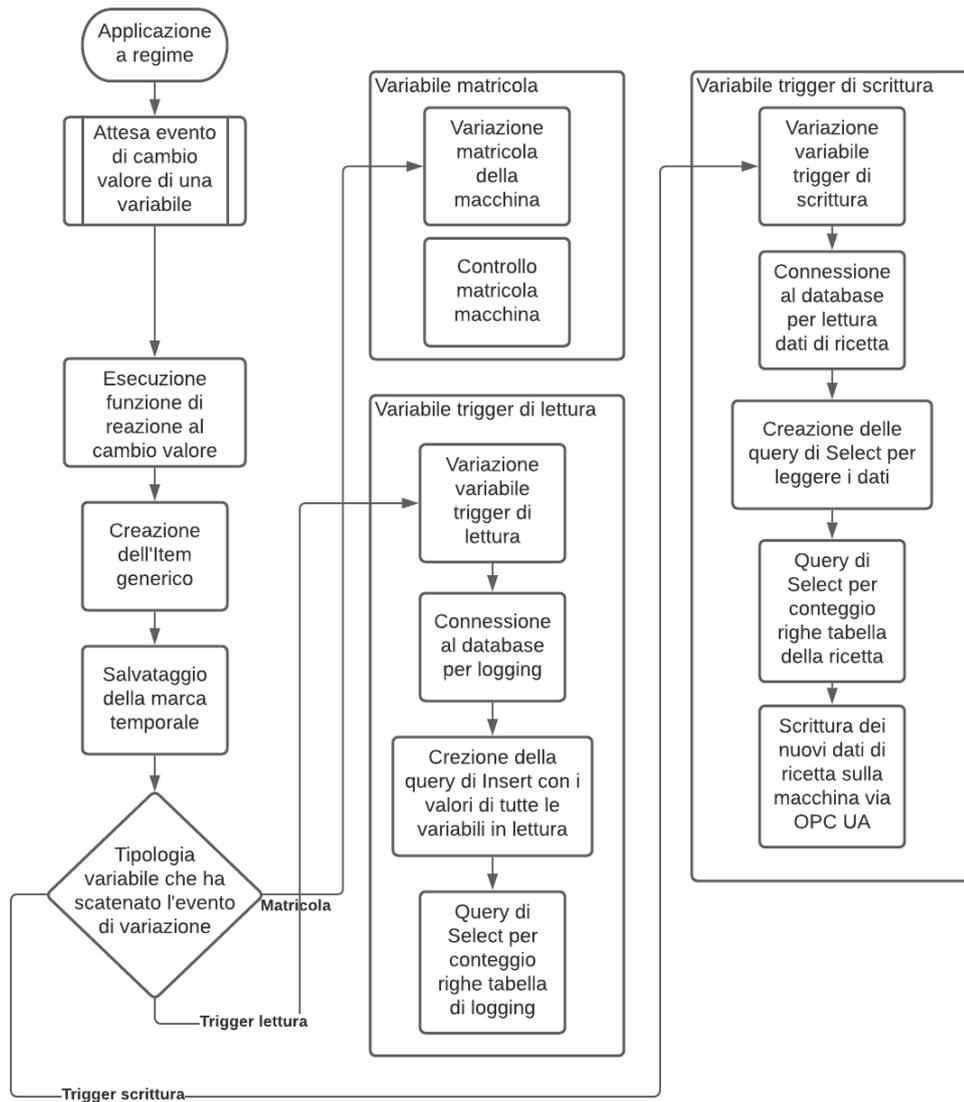


Figura 5.12: Flowchart della gestione del cambio valore.

Matricola della macchina

Il controllo di questa variabile è utile unicamente per gestire la situazione già descritta in cui il cliente installi l'applicazione per il controllo di una macchina non prevista e diversa da quella decisa in sede di contratto. La matricola della macchina prevista è infatti annegata nella stringa criptata presente nel file testuale. Nel caso in cui, all'accensione, la variabile matricola non coincida con la decodifica della stringa, viene segnalata l'anomalia e bloccate tutte le operazioni di interfacciamento con il Database. In particolare viene utilizzato un booleano **Blocked** per notificare il riscontro dell'anomalia e viene imputato il valore -3 alla variabile di errore **ErrorCode**, che verrà poi scritto sulla variabile di errore monitorata e visualizzata nel listato di variabili controllate.

Variabile di *trigger* per la lettura

Una delle due variabili di *trigger* è utile al Log dei dati sul database; viene quindi indicata per convenzione come *trigger* in lettura in quanto scrive sul database tutte le variabili lette dalla macchina. Solo nel momento di un'esecuzione della funzione a causa di una variazione di questa variabile viene scatenata la scrittura sul database; tutte le altre variazioni vengono ignorate. La scelta del periodo di questa variabile, come già spiegato, è di cruciale importanza.

A questo punto della trattazione risulta ancora più chiaro quali siano le conseguenze di un'erronea scelta delle tempistiche con cui vengono scatenati questi fronti di variazione. Tutte le variazioni delle variabili sono ignorate e i valori scritti sul database sono cadenzati dalla variazione del *trigger* di lettura. Da questo consegue che variazioni istantanee delle quantità monitorate potrebbero essere perse se non coincidono con le variazioni del *trigger*.

Risulta quindi necessario uno studio del processo per scegliere adeguatamente il periodo di questa variabile in modo da ripulire il Log dal rumore senza però privarsi delle variazioni più significative delle variabili di processo. Il controllo della variazione del *trigger* è correlato con un ulteriore controllo del booleano **Blocked**; nel caso in cui venga riscontrata l'anomalia della matricola, tutta la sezione di *logging* sul database viene omessa. All'interno del corpo della funzione viene creata dinamicamente la query che verrà eseguita sul database. L'inserimento di una riga contiene tutti i valori delle variabili monitorate, catturati nel momento della variazione del *trigger*. Si ottiene quindi un nuovo record che viene inserito nel database con l'esecuzione di una **Insert**, nella quale ogni variabile è generalmente chiamata **Field*i***.

Ogni installazione dell'applicazione è corredata di un manuale tecnico con il quale viene informato il cliente finale della corrispondenza tra la colonna *i*-esima della tabella di Log e la variabile effettivamente monitorata. Vengono storicizzate quindi tutte le variabili tranne quelle in scrittura dalla macchina, l'*heartbit* e la variabile di

errore. Inoltre, prima dell'esecuzione della query, viene aggiunto in prima posizione un campo contenente la marca temporale che identificherà quel record. Dopo la corretta esecuzione della query di inserimento, viene eseguita una **Select** per contare le righe attuali della tabella di Log. Grazie a questa query è possibile quindi aggiornare il contatore presente sulla pagina principale dell'applicazione.

Variabile di *trigger* per la scrittura

Oltre a storicizzare i dati desiderati, l'applicazione è in grado di notificare i cambi di ricetta da remoto. L'interazione bidirezionale con i sistemi gestionali è infatti uno dei requisiti dell'Industria 4.0 e il cambio ricetta centralizzato è uno dei "desiderata" più comuni. Quando viene rilevata la variazione del *trigger* per la scrittura viene eseguita la funzione **WriteValues** che, per prima cosa, crea una lista di stringhe: una per ogni variabile in scrittura definita nel file di configurazione. Per non perdere in generalità e per comodità dei clienti finali, tutte le variabili sono salvate sul database come stringhe. Viene quindi eseguita una query di **Select** sulla tabella opportuna, utile a caricare in memoria le informazioni da inviare tramite OPC UA alla macchina. Durante l'esecuzione della query e, in particolare, nella gestione dei risultati, viene implementato un meccanismo di controllo per quanto riguarda il numero di righe della tabella.

Per evitare ambiguità sulla ricetta successiva da inviare alla macchina, il numero di righe di questa tabella deve essere sempre pari a uno. Nel caso in cui il numero di righe non sia congruo e quindi vi siano più righe o nessuna, vengono generati rispettivamente i valori -1 e -2 che vengono in seguito assegnati alla variabile **ErrorCode**. In entrambi i casi verrà mantenuto il *logging* sul database ma comandati a 0 i valori della ricetta per la macchina. Si è preferito inviare degli zeri piuttosto che delle stringhe vuote per una gestione lato macchina più favorevole. Per ragioni di controllo viene comunque eseguita un'ulteriore query per contare le righe di questa tabella, in modo da informare gli operatori di eventuali incongruenze.

Tramite la funzione di libreria **WriteValues** alla quale viene passata la lista di stringhe di valori e quella delle variabili monitorate, vengono inviati i valori desiderati alla macchina. Oltre alle variabili utili per il cambio ricetta come nome e identificativo della ricetta e pezzi da produrre, viene aggiunto anche il codice di errore. Grazie all'invio di quest'ultimo è possibile generare degli eventi collegati ai valori della variabile di errore che servano da avviso agli operatori della macchina come descritto in 5.3.5.

5.4.8 Gestione del Login

L'applicazione, come già descritto, gestisce un login che garantisce l'accesso come amministratore ai servizi messi a disposizione.

L'accesso con o senza privilegi contraddistingue la visibilità o l'oscuramento dei nomi dei *tag*. Tutte le variabili, se non si è autenticati come amministratori, non sono visibili e sono sostituite da nomi fittizi del tipo `Fieldi`. La funzione che gestisce l'evento associato alla pressione del pulsante di login, dopo aver controllato la correttezza delle credenziali di accesso, gestisce e implementa il codice per oscurare o rendere visibili tutti i nomi. Il nome delle variabili è in realtà il *path* assoluto di ognuna all'interno dell'albero di progettazione Simotion. Il motivo principale dell'autenticazione e oscuramento delle variabili è, da una parte scoraggiare l'utente finale a una ricerca manuale delle stesse all'interno di un albero di progetto sconosciuto, dall'altra evitare di esporre il *know-how* aziendale sull'organizzazione dei dati e della memoria di un programma Simotion.

Capitolo 6

Conclusione e sviluppi futuri

I concetti chiave dell'Industria 4.0, in molte occasioni, non sono ancora del tutto compresi nelle realtà industriali: solo negli ultimi anni la supervisione e l'interconnessione del campo con i sistemi informativi aziendali stanno acquisendo un ruolo fondamentale nell'industria.

L'applicazione sviluppata, per alcuni clienti, risulta utile per approfondire il tema dell'Industria 4.0 e comprendere i requisiti per poter ottenere la certificazione di conformità dei loro macchinari. Oltre al conseguente vantaggio economico, le aziende cominciano ad apprezzare i benefici che si possono trarre da un sistema tecnologicamente all'avanguardia. La supervisione di un impianto, indipendentemente dal tipo e dalla complessità del software utilizzato, oltre a portare agevolazioni di tipo gestionale e organizzativo, migliora e incrementa il processo produttivo.

Il progetto Motion SQL Connector, con alle spalle quasi un anno di storia, è nato come una soluzione economica di supervisione per permettere una prima interazione tra il campo e l'infrastruttura informatica aziendale. Grazie a continui riscontri da parte dei clienti che stanno utilizzando il software, l'applicazione è in costante aggiornamento per rispondere alle loro diverse esigenze.

Il testo ha trattato nel dettaglio l'ultima versione rilasciata, nella quale sono state introdotte la riconnessione automatica e la gestione del timeout di connessione. Un'applicazione in così stretto contatto con il processo produttivo deve essere robusta agli spegnimenti dei macchinari e alle perdite di connessione, fenomeni molto frequenti in ambienti industriali. L'obiettivo di instaurare un rapporto duraturo con le aziende che stanno attualmente utilizzando il software è proprio quello di creare un circolo virtuoso, per permettere lo sviluppo di un'applicazione utile al cliente e al passo con le necessità del mondo industriale odierno.

Al momento è previsto un ulteriore aggiornamento dell'applicazione, nel quale

verrà introdotta la possibilità di gestire ricette multiple comandate dal database. La gestione centralizzata delle ricette ha infatti due possibili utilità. Da una parte sgrava gli operatori dal compito di decidere quale prodotto caricare in macchina, dall'altra permette una schedulazione della produzione decisamente più comoda e veloce. Alcune aziende hanno l'effettiva necessità di risolvere solo il secondo dei due punti, lasciando agli operatori la facoltà di decidere la schedulazione delle ricette in modo autonomo e conseguente alle necessità del momento. Per questo tipo di esigenza è quindi necessario che l'applicazione sviluppata preveda l'invio alla macchina di più di una ricetta per volta. Grazie a questa miglioria, l'operatore avrà la possibilità di selezionare, sull'HMI di macchina, la ricetta desiderata da produrre.

Oltre alle funzionalità operative descritte, Motion SQL Connector funge da *Historian* per i dati di produzione. Per rendere fruibile l'informazione non è da escludere la possibilità di sviluppare un'interfaccia grafica minima, che permetta la visualizzazione delle informazioni storicizzate. Nella versione attuale, i dati vengono salvati su un database del cliente, il quale, se lo desidera, può visualizzarli o rielaborarli mediante opportuni software.

La storicizzazione dell'informazione non deve essere fine a se stessa: reportistica e post-elaborazione sono solo due esempi di utilizzo dei dati salvati. Potrebbe quindi risultare comodo avere, a livello di applicazione, già una prima visualizzazione attraverso grafici e trend.

L'eventuale sviluppo di un'interfaccia grafica deve essere però conseguente ad un'attenta analisi di costi di sviluppo e effettiva necessità dei clienti. Uno dei punti di forza del Motion SQL Connector è il soddisfacimento delle richieste minime dell'Industria 4.0 con una spesa contenuta da parte del cliente. Nel momento in cui l'applicazione subisca un'evoluzione importante dal punto di vista grafico, potrebbe non essere più competitiva sul mercato rispetto ad altre soluzioni più standard e complete come i supervisori SCADA.

Oltre agli aggiornamenti funzionali, in futuro saranno possibili delle modifiche sostanziali al codice dell'applicazione. Il software Motion SQL Connector è stato programmato in **C#** per questioni di competenze personali ma, con l'avvento di **Qt** e delle sue indiscusse potenzialità, non è da escludere una riscrittura del *core* dell'applicazione con questa nuova collezione di librerie.

Nel corso dell'esperienza formativa di quest'anno lavorativo, è capitata l'occasione di dover progettare altri programmi dedicati ad alcune specifiche esigenze, e per lo sviluppo di tali software è stato scelto proprio **Qt**, integrato in **C++**.

In futuro potrebbe rendersi necessaria e funzionale la fusione del Motion SQL Connector con gli altri applicativi sviluppati in **Qt**: in questo caso la conversione del software risulterà necessaria.

Con l'avvento delle nuove tecnologie, anche il mondo dell'industria, seppur più lentamente di altre realtà, sta evolvendo verso un approccio più informatico

e organizzato. La reportistica di impianto, la schedulazione delle ricette e la manutenzione predittiva sono solo alcuni esempi di come le aziende si stiano "digitalizzando".

Il termine *Digital Factory* rappresenta i cambiamenti presenti e futuri che le industrie nel mondo stanno subendo e evidenzia, ancora una volta, come l'informatica stia diventando una scienza predominante anche in questi settori.

Bibliografia

- [1] Caravita A. *Macchine per il converting: Funzionalità e implementazione di tecniche di controllo*. URL: http://tesi.cab.unipd.it/49489/1/Tesi_di_Laurea_-_Iacopo_Tamellin_Matricola_1046040.pdf (cit. alle pp. 16, 33).
- [2] Double e Company. *Rulli allargatori gobbi*. URL: <https://www.ee-co.com/it/products/rulli-allargatori-gobbi> (cit. a p. 16).
- [3] MTI Corporation. *MSK-B750*. URL: <https://www.mtixtl.com/MSK--DSC-B750.aspx> (cit. a p. 17).
- [4] Dan. *Types of equipment used in the converting industry*. URL: <http://www.sizemic.org/types-of-equipment-used-in-the-converting-industry> (cit. alle pp. 18–20).
- [5] YYS. *China Paper Roll Cutting Machine With Blade*. URL: <https://www.yys-machinery.com/china-paper-roll-cutting-machine-with-blade.html> (cit. a p. 18).
- [6] Parason. *Synchronous - On The Fly Sheeters*. URL: <https://parason.com/products/paper-machine/synchronous-fly-sheeters/> (cit. a p. 18).
- [7] Advantage Fabricated Metals. *Embossing*. URL: <http://www.advantagefabricatedmetals.com/embossing-process.html> (cit. a p. 19).
- [8] Utecs. *MYW720/1300*. URL: <http://www.utecsinter.com/products/paper-embossing-machine-94.html> (cit. a p. 19).
- [9] Soest Ingenieursbureau en Handelonderneming bv. *Roller Coating*. URL: <https://www.soestbv.com/coating/coating-machines/roller-coating/> (cit. a p. 20).
- [10] Zappa Macchine. *Hot melt multi-roller coating and laminating machine mod. TM-coat*. URL: <http://zappamacchine.com/en/prodotto/hot-melt-lamination-machine/> (cit. a p. 20).
- [11] Campbell Wrapper Corporation. *Flow Wrapper Basics*. URL: <https://www.campbellwrapper.com/machinery/horizontal-flow-wrappers/flow-wrapper-basics/> (cit. a p. 21).

- [12] R. Duane Smith. «Challenges in Winding Flexible Packaging Film». In: *Coating International* 61 (set. 2015) (cit. alle pp. 24, 29–31, 33–35).
- [13] Valmet Forward. *Troubleshooting roll defects - causes and corrective actions*. URL: <https://www.valmet.com/media/articles/up-and-running/reliability/RTWRDWUG/> (cit. alle pp. 24, 25, 27, 28).
- [14] Jarshire Corrugating & Board. *Reel savers and core restorers*. URL: <http://www.jarshire.co.uk/reel-savers/4594150985> (cit. a p. 25).
- [15] Valmet Forward. *Avoiding roll defects at the winder - causes and solutions*. URL: <https://www.valmet.com/media/articles/up-and-running/performance/AvoidWindingDefects/> (cit. a p. 26).
- [16] Jeff Damour. *The mechanics of tension control*. URL: <https://www.convertersaccessory.com/papers/tcpaper1.pdf> (cit. alle pp. 27, 37–41, 43, 45, 46, 48–51).
- [17] Cecilia Land. *Baggy paper webs: Effect of uneven moisture and grammage profiles in different process steps*. URL: <http://www.diva-portal.org/smash/get/diva2:354209/FULLTEXT01.pdf> (cit. a p. 27).
- [18] Geertje Hek. *Buckling of roll of toilet paper having accidentally been axially compressed*. URL: https://shellbuckling.com/presentations/unstiffenedCylinders/pages/page_36.html (cit. a p. 27).
- [19] Pekka Komulainen. *Focus on paper roll quality, roll defects and roll hardness testing*. URL: <https://www.slideshare.net/Peeke/paper-roll-quality-and-roll-hardness-1-jan-2018-85460531> (cit. a p. 28).
- [20] Rupesh Kumar Garg. *The effects of nip and wound roll diameter on wound-on-tensione in surface wound production size roll*. URL: <https://shareok.org/bitstream/handle/11244/10891/Thesis-2002-G231e.pdf?sequence=1> (cit. alle pp. 32, 33).
- [21] Parkinson technologies inc. *Surface of center winder: which is the best for your application*. URL: <https://parkinsonstechnologies.com/index.php/blog/surface-or-center-winder-which-is-best-for-your-application> (cit. a p. 33).
- [22] Siemens. «Standard Winder Application with SIMOTION». In: (feb. 2005) (cit. alle pp. 34, 52).
- [23] Siemens. *Roll Diameter Requirements in Converting Processes*. URL: https://www.aimcal.org/uploads/4/6/6/9/46695933/roll_diameter_requirements_in_converting_processes__2802.16.17_29.pdf (cit. alle pp. 44, 45).
- [24] Christoph Roser. *A Critical Look at Industry 4.0*. URL: <https://www.allaboutlean.com/industry-4-0/> (cit. a p. 53).

- [25] Edgar Gustavo Delgado Tello. *Industry 4.0: Application of advanced services in logistics*. URL: <https://upcommons.upc.edu/handle/2117/121967> (cit. alle pp. 54, 63).
- [26] Statista Research Department. *Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025*. URL: <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/> (cit. a p. 55).
- [27] Stefan Hoppe / Alexander Stark. *IoT Basics: What is OPC UA?* URL: <https://www.spotlightmetal.com/index.php/iot-basics-what-is-opc-ua-a-842878/> (cit. a p. 57).
- [28] KVM Contractor. *OPC UA HMI SCADA MES Panels*. URL: <http://www.kvmcontractor.com/cloud-automation/opc-ua-hmi-scada-mes-panels/> (cit. a p. 57).
- [29] OPC Foundation. *OPC Unified Architecture Part 1: Overview and Concepts*. URL: <https://reference.opcfoundation.org/v104/Core/docs/Part1/> (cit. alle pp. 59–61).