

UNIVERSITÀ DEGLI STUDI DI PADOVA

—

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

—

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA
MECCANICA

ANALISI E OTTIMIZZAZIONE DI CELLA ROBOTIZZATA FLESSIBILE

RELATORE: CH.MO PROF. ING. GIULIO ROSATI

LAUREANDO: NICOLA GIORDANI

ANNO ACCADEMICO 2013-2014

Ai miei genitori

“ Dal momento che non possiamo evitare una fine improvvisa, almeno cerchiamo di non morire un poco ogni giorno, ricordando che essere vivi richiede uno sforzo ben maggiore del semplice respirare. ”

MARTHA MEDEIROS - A MORTE DEVAGAR, 2000

Indice

| | |
|---|-------------|
| Sommario | XIII |
| Introduzione | XV |
| 1 Robotica industriale | 1 |
| 1.1 Robotica flessibile | 2 |
| 1.1.1 Cella di lavoro | 3 |
| 1.2 Stato dell'arte | 5 |
| 1.3 Layout di cella in esame | 7 |
| 1.3.1 Robot e pinza | 7 |
| 1.3.2 Feeder flessibile | 7 |
| 1.3.3 Postazione di lavorazione | 9 |
| 1.3.4 Postazione di scarico | 10 |
| 1.3.5 Sequenziamento dinamico | 11 |
| 1.3.6 Buffer | 12 |
| 1.3.7 Disposizione delle postazioni | 13 |
| 1.3.8 Posizione di home | 14 |
| 1.3.9 Parametri in esame | 14 |
| 2 Analisi di cella robotizzata | 17 |
| 2.1 Classificazione di Graham-Dawande | 17 |
| 2.1.1 Parametro α | 17 |
| 2.1.2 Parametro β | 18 |
| 2.1.3 Tempistiche di movimentazione | 19 |
| 2.1.4 Funzione obiettivo γ | 21 |

| | | |
|----------|--|-----------|
| 2.1.5 | Classificazione della cella in esame | 21 |
| 2.2 | Metodi di analisi | 23 |
| 2.2.1 | Stati della cella | 23 |
| 2.2.2 | Attività della cella | 25 |
| 2.3 | Tempistiche fra stati della cella | 26 |
| 2.3.1 | Carta robot-macchina | 27 |
| 2.4 | Risoluzione analitica | 29 |
| 3 | Modello analitico | 31 |
| 3.1 | Notazione | 31 |
| 3.2 | Cella senza buffer | 32 |
| 3.2.1 | Tempo di scarico trascurabile | 32 |
| 3.2.2 | Influenza del tempo di scarico | 36 |
| 3.3 | Cella con buffer | 37 |
| 3.4 | Priorità feeder | 38 |
| 3.4.1 | Componenti depositati nel buffer | 38 |
| 3.4.2 | Svuotamento del feeder | 40 |
| 3.4.3 | Tempo limite di rinnovo feeder | 42 |
| 3.4.4 | Componenti prelevati dal buffer | 43 |
| 3.4.5 | Tempo ciclo | 44 |
| 3.4.6 | Flusso nullo nel buffer | 46 |
| 3.4.7 | Post-buffering | 48 |
| 3.4.8 | Influenza del tempo di scarico | 50 |
| 3.5 | Priorità buffer | 51 |
| 3.5.1 | Componenti depositati nel buffer | 52 |
| 3.5.2 | Svuotamento del feeder | 52 |
| 3.5.3 | Tempo limite di rinnovo feeder | 52 |
| 3.5.4 | Componenti prelevati dal buffer | 54 |
| 3.5.5 | Tempo ciclo | 54 |
| 3.5.6 | Post-buffering | 56 |
| 3.5.7 | Flusso nullo nel buffer | 56 |
| 3.5.8 | Feeder unitario | 58 |

| | | |
|----------|--|-----------|
| 3.5.9 | Influenza del tempo di scarico | 59 |
| 4 | Implementazione del modello | 61 |
| 4.1 | Implementazione in Matlab | 61 |
| 4.1.1 | Ricorsività | 62 |
| 4.2 | Confronto con cicli limite | 64 |
| 4.3 | Dimensione del feeder variabile | 66 |
| 4.3.1 | Priorità feeder | 67 |
| 4.3.2 | Priorità buffer | 69 |
| 5 | Apparato sperimentale | 71 |
| 5.1 | Robot e controller | 71 |
| 5.2 | Postazioni di lavoro | 73 |
| 5.3 | Software per la gestione della cella | 76 |
| 5.3.1 | Comunicazione con Matlab | 78 |
| 5.3.2 | Segnali I/O | 81 |
| 5.4 | Cella di lavoro | 83 |
| 5.5 | Accuratezza dei movimenti | 83 |
| 5.6 | Prove sperimentali | 86 |
| 5.6.1 | Impostazione delle prove | 86 |
| 5.6.2 | Problematiche riscontrate | 87 |
| 5.7 | Analisi dell'incertezza | 88 |
| 5.7.1 | Fonti di incertezza | 88 |
| 6 | Analisi dei risultati | 95 |
| 6.1 | Validazione ipotesi iniziali | 95 |
| 6.1.1 | Movimenti additivi | 95 |
| 6.1.2 | Postazioni puntiformi | 98 |
| 6.2 | Confronto dei risultati con il modello | 101 |
| 6.2.1 | Parametri sperimentali | 101 |
| 6.2.2 | Dimensione del feeder | 103 |
| 6.2.3 | Tempo di scarico | 103 |
| 6.2.4 | Tempo di rinnovo feeder | 106 |

| | | |
|----------|--|------------|
| 6.2.5 | Distanza fra postazioni | 107 |
| 6.2.6 | Tempo di lavorazione | 108 |
| 6.2.7 | Dimensione buffer | 110 |
| 7 | Ottimizzazione della produttività | 113 |
| 7.1 | Curve parametriche | 113 |
| 7.1.1 | Confronto con prove sperimentali | 118 |
| 7.2 | Posizionamento del buffer | 118 |
| 7.2.1 | Assunzioni per la movimentazione | 122 |
| 7.2.2 | Risultati | 124 |
| 7.2.3 | Confronto con risultati sperimentali | 127 |
| 7.3 | Numero di componenti variabile | 128 |
| 7.3.1 | Confronto con risultati sperimentali | 131 |
| 7.4 | Ottimizzazione della sequenza di buffering | 132 |
| | Conclusioni | 135 |
| | A Notazione | 137 |
| | B Confronto SPEL+, PAC, V+ | 139 |
| B.1 | Locazioni | 140 |
| B.1.1 | Epson <i>SPEL+</i> | 141 |
| B.1.2 | Denso PAC | 144 |
| B.1.3 | Adept V+ | 146 |
| B.2 | Comandi di movimento | 148 |
| B.2.1 | <i>SPEL+</i> Language | 150 |
| B.2.2 | <i>PAC</i> Language | 152 |
| B.2.3 | Adept - V+ Language | 153 |
| | C Programmazione in Epson | 155 |
| C.1 | Multitasking | 155 |
| C.1.1 | Background tasks | 156 |
| C.2 | Sense, Parallel statements e CP | 157 |
| C.2.1 | Sense | 157 |

| | | |
|-------|---------------------------------------|------------|
| C.2.2 | Parallel statements | 158 |
| C.2.3 | Influenza sui tempi di moto | 159 |
| | Bibliografia | 163 |

Sommario

La presente tesi si pone l'obiettivo di analizzare la produttività di uno specifico layout di cella robotizzata al variare dei relativi parametri caratteristici. Nella cella sono presenti un feeder flessibile, un singolo macchinario di lavorazione e una postazione di scarico pezzo. Si considera il caso con robot singolo e un'unica tipologia di componenti da movimentare. Vi è inoltre un buffer, posizionato fra il feeder e il macchinario di lavorazione, per contrastare la presenza incerta di un componente nella postazione di input. La gestione della cella viene effettuata dinamicamente: non vi è un ciclo di lavoro stabilito a priori e la scelta della successiva operazione viene realizzata in base alla disponibilità istantanea delle postazioni. Tale approccio è consentito dall'assegnazione di differenti priorità alle possibili operazioni eseguibili.

I principali obiettivi raggiunti consistono nell'ottenimento di curve parametriche del tempo ciclo al variare di uno o più parametri caratteristici e la determinazione dell'assegnazione ottimale di priorità alle operazioni che prevedono un prelievo da feeder o da buffer in funzione di tali parametri.

Introduzione

Il lavoro di tesi presentato nelle pagine seguenti si inserisce nel contesto della gestione del ciclo di lavoro di celle robotizzate flessibili. Per sopperire alle frequenti variazioni di lavorazione e produzione, è necessario poter disporre di una logica che faciliti la riconfigurabilità del sistema, generando automaticamente il ciclo di lavoro, una volta nota l'architettura della cella necessaria all'ottenimento del prodotto desiderato. La gestione della cella, secondo tale approccio, avviene dinamicamente e non è necessaria la definizione a priori di un ciclo di lavoro. Sorge tuttavia il problema della stima della produttività nella fase preliminare di progetto in funzione dei parametri che caratterizzano la cella.

Il presente studio analizza un determinato layout di cella robotizzata elaborando un modello analitico per valutarne la produttività prima della realizzazione. Si vuole determinare l'interazione fra i parametri caratteristici dell'architettura per fornire criteri di valutazione utili nella fase preliminare del progetto.

Nel capitolo 1 è riportata una descrizione della robotica in ambito industriale e del trend attuale di orientamento verso l'automazione flessibile con celle robotizzate. Segue la descrizione dello stato dell'arte relativo allo studio di celle robotizzate e viene introdotto il layout della postazione considerata in questa tesi. I problemi inerenti le celle robotizzate, infatti, non sono analizzabili per un caso generico, ma necessitano la specificazione del layout in esame.

Il capitolo 2 si occupa pertanto di inquadrare questo studio fra tutti i possibili casi d'esame tramite la classificazione di Graham-Dawande. Si esplicita ciascuno dei parametri previsti da essa, ponendo particolare attenzione alle assunzioni sui tempi di movimentazione fra le stazioni interne della cella. A tal riguardo vengono esposte le assunzioni comunemente adottate in letteratura e si giustifica la scelta

effettuata fra queste. Sono presentati, inoltre, i possibili metodi di analisi di cella robotizzata e viene elaborato il diagramma degli stati della cella con le relative tempistiche. Esso risulta particolarmente utile per la realizzazione del modello analitico.

Nel capitolo 3 viene sviluppato il modello teorico per valutare il tempo ciclo della cella, dati i parametri caratteristici. Inizialmente viene valutato il tempo ciclo della cella gestita senza l'utilizzo del buffer, in modo da ottenere un valore di confronto rispetto all'utilizzo del buffer. Successivamente sono elaborati i modelli per la gestione della cella con buffering, analizzando due casi differenziati da una diversa priorità associata al feeder e al buffer. Se un componente, infatti, è disponibile per il prelievo in entrambe le postazioni, è necessario indicare quale sia la stazione prioritaria fra le due. La differente priorità di svuotamento, infatti, modifica notevolmente il ciclo di lavoro della cella. In seguito, nel capitolo 4, si analizzano l'implementazione del modello teorico in un algoritmo utilizzando il software MATLAB e le problematiche riscontrate in tale operazione.

Il capitolo 5 analizza la strumentazione utilizzata per verificare sperimentalmente l'attendibilità del modello. L'apparato strumentale riguarda il robot, il controller, i relè utilizzati per simulare le postazioni e il computer annesso alla cella. Esso, tramite la comunicazione con il controller, si occupa di gestire la cella. In questo capitolo è inclusa anche la descrizione generale del codice utilizzato per raggiungere lo scopo. Per non appesantire la trattazione si è preferito riportare in appendice C, anziché nel testo principale, un'analisi dettagliata di alcuni aspetti secondari del codice *SPEL+* in questione. Essi riguardano l'utilizzo di *background tasks*, *parallel statements* e altre opzioni per il moto. In appendice B è invece riportata l'analisi fra tre differenti linguaggi di programmazione per quanto riguarda la gestione di locazioni e la movimentazione. Si desidera porre l'attenzione sui vantaggi e le problematiche derivanti dall'utilizzo di robot Epson rispetto alla programmazione di dispositivi prodotti da aziende concorrenti.

I capitoli 6 e 7, infine, si occupano di esporre i risultati dei test sperimentali confrontandoli con i risultati analitici. All'inizio si verifica sperimentalmente l'attendibilità delle ipotesi semplificative avanzate nell'analisi della cella in esame. Vengono successivamente illustrate le curve caratteristiche del tempo ciclo in re-

lazione ai principali parametri della cella giustificandone gli andamenti risultanti. Si riportano, inoltre, una serie di curve parametriche che ben riassumono i risultati delle analisi, utili in fase di progettazione preliminare della cella di lavoro. Infine si analizza l'influenza sul tempo ciclo del posizionamento della postazione di buffering, per determinare la convenienza rispetto ad una sua vicinanza al feeder o al macchinario di lavorazione.

Capitolo 1

Robotica industriale

Affinchè un'azienda industriale possa avere successo nel mondo attuale, altamente competitivo, un aspetto essenziale è rappresentato dall'aumento della produttività. Un modo per fare ciò è velocizzare il ciclo di lavoro tramite l'automazione delle catene produttive, la quale consente di gestire i processi e i macchinari tramite sistemi di controllo. Tale approccio permette inoltre di ridurre l'apporto di manodopera e migliorare la qualità del prodotto finale. Per tali motivi l'automazione è attualmente implementata nella maggior parte dei sistemi produttivi. Un sistema automatizzato di produzione consiste in una serie di macchinari che gestiscono la movimentazione del prodotto e le sue successive lavorazioni fino all'ottenimento del prodotto finito. Questo consente una lavorazione accurata e ripetibile assicurando al prodotto finale caratteristiche standard riducendo, inoltre, il numero di scarti. La progettazione e realizzazione di una linea industriale apposita, o *macchina dedicata*, per un determinato ciclo di produzione richiede un investimento notevole. Un elevato indice di ritorno dell'investimento (ROI) e la durata pluridecennale del sistema ne giustificano tuttavia il costo iniziale.

La progettazione di un sistema automatizzato dedicato consente di produrre un determinato prodotto finito sempre uguale a sè stesso per l'intero tempo di utilizzo del sistema di produzione. Un sistema siffatto ottiene una resa elevata a scapito della flessibilità e risulta perciò sconveniente in una situazione di produzione limitata nel tempo o con necessità di frequenti modifiche al processo. Il trend attuale in ambito industriale consiste nella produzione di lotti di dimensio-

ni ridotte e con notevole varietà nel prodotto: in tale scenario la realizzazione di macchinari dedicati non rappresenta la scelta ottimale.

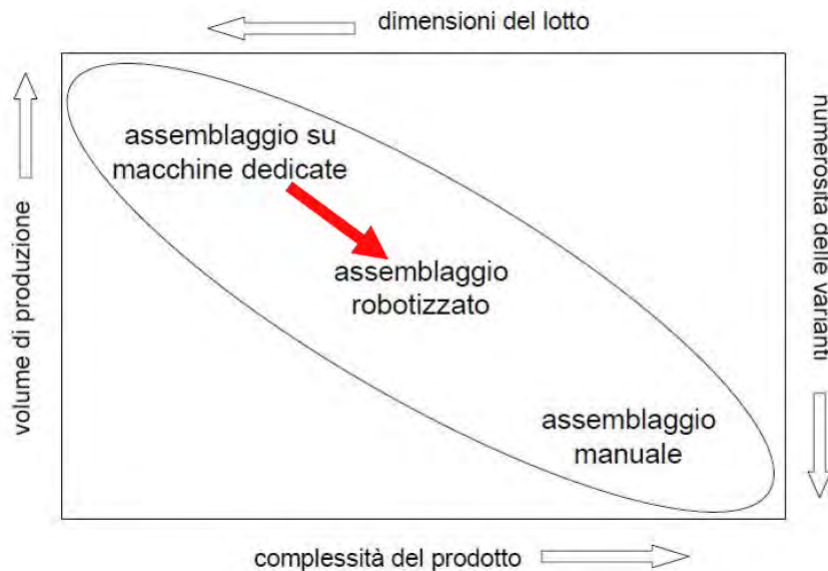


Figura 1.1: Tecnologie di assemblaggio in funzione dei parametri di produzione. La freccia scura rappresenta il trend attuale verso lotti di dimensione ridotta e numerose varianti del prodotto finale. [1]

La flessibilità in un sistema industriale automatizzato viene ottenuta tramite l'utilizzo di celle di lavoro robotizzate anzichè macchinari dedicati. La gestione dell'automazione in celle robotizzate è nota come *robotica flessibile*.

1.1 Robotica flessibile

La tendenza attuale in ambito industriale è determinata dalla realizzazione di sistemi di produzione formati da una successione di singole unità di base robotizzate estremamente flessibili. Una cella robotizzata è un sistema che contiene uno o più robot e un numero di dispositivi ausiliari dipendente dalle necessità produttive. I robot utilizzati nelle celle sono manipolatori industriali, macchine programmabili e pertanto in grado di effettuare operazioni molto differenti fra di

loro. Per la maggior parte delle applicazioni è utilizzato un robot singolo come semplice *handling device*, ovvero per gestire la movimentazione dei pezzi nella cella: al robot è assegnato il solo compito di prelevare e depositare i componenti sui macchinari disposti all'interno della cella.

Una cella robotizzata è un ambiente estremamente flessibile: è sufficiente riprogrammare il manipolatore e variare la disposizione delle postazioni all'interno della cella per soddisfare eventuali nuove esigenze del ciclo produttivo. I componenti della cella possono essere in gran parte riutilizzati, riducendo così il problema della progettazione ad hoc dell'attrezzatura che si presenta nel caso di macchine dedicate. In particolare il manipolatore, il relativo controller e il quadro elettrico sono riutilizzabili. Essendo questi i componenti di costo maggiore è comprensibile la tendenza verso l'utilizzo della robotica flessibile.

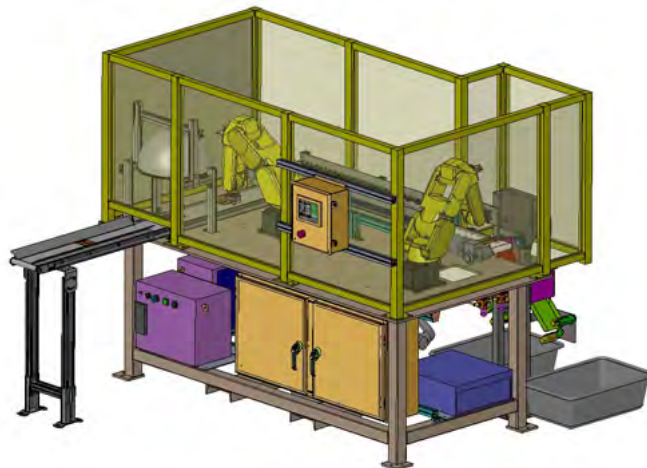


Figura 1.2: Esempio di cella robotizzata. Design per una cella di ispezione in una catena di produzione alimentare.

1.1.1 Cella di lavoro

Il termine *cella* viene utilizzato per identificare un'unità base in un processo complesso. Più celle indipendenti vengono disposte in successione per realizzare il ciclo produttivo e ottenere il prodotto finito. Ciascuna cella è fisicamente caratterizzata da una struttura chiusa solitamente inagibile per gli operatori. Per motivi di

sicurezza, infatti, l'ambiente attorno al manipolatore è chiuso da strutture di profilati e pannelli di plexiglass. L'apertura di una qualsiasi porta della cella impone l'arresto delle attività del robot per garantire la sicurezza degli operatori.

I manipolatori più frequentemente utilizzati in celle robotizzate sono i robot antropomorfi a 6 assi e i robot di tipo SCARA, *Selective Compliance Assembly Robot Arm*, caratterizzati da 4 gradi di libertà. Il numero di assi, o gradi di libertà, indica la molteplicità di giunti indipendentemente controllabili. Il manipolatore SCARA è tipicamente utilizzato per le semplici operazioni di *pick-and-place* che prevedono l'afferraggio e prelievo di pezzi con movimenti perpendicolari al piano di lavoro. I robot antropomorfi consentono una più ampia gamma di operazioni.



Figura 1.3: Esempi di manipolatori industriali. Robot a 6 assi Adept Viper s650 (sinistra) e robot a 4 assi Adept Cobra s800 di tipologia SCARA (destra).

L'utilizzo di celle robotizzate richiede attenzione alla gestione del layout interno e alla sequenza dei movimenti da imporre al robot in modo da ottenere la massima produttività possibile. Un tipico problema di ottimizzazione di cella robotizzata richiede di specificare l'ottimale sequenza di movimenti del robot ed eventualmente la sequenza di prelievo di parti differenti dalla stazione di input.

1.2 Stato dell'arte

Lo studio della sequenza ottimale di movimenti è ampiamente discusso in bibliografia. Generalmente ciascuno studio si sviluppa attorno ad un ben preciso layout della cella, poichè si ritiene impossibile analizzare genericamente tutte le situazioni possibili. Un'estesa descrizione dello stato dell'arte nei problemi di sequenziamento in celle robotizzate è stata riportata da Dawande et al. [2]. Un'analisi approfondita dei cosiddetti *Robotic Flowshops*, cicli in cui ciascun componente deve essere processato da differenti macchinari in un ordine preciso, è riportata in Crama et al. [3].

I primi studi sul sequenziamento di celle robotiche sono stati condotti con layout lineari e risalgono agli inizi degli anni '90. Tra questi rientra lo studio analitico condotto da Sethi et al. [4]. Egli dimostra che ci sono $m!$ cicli di lavoro possibili in una cella con m macchinari di lavorazione e una sola tipologia di componenti. Nel caso di esattamente 3 distinti macchinari interni alla cella identifica inoltre le condizioni sotto le quali ciascun caso risulta quello ottimale. Negli ultimi decenni, grazie alla diffusione dei manipolatori industriali, si è potuto assistere ad una crescita di interesse verso l'argomento e un numero sempre maggiore di pubblicazioni al riguardo. In particolare è aumentato l'interesse verso particolari situazioni presenti all'interno della cella comparse negli ultimi anni. Fra queste si possono citare l'utilizzo di macchinari in parallelo, postazioni di buffering, utilizzo di pinze multiple e presenza contemporanea di componenti differenti.

L'introduzione della possibilità di disporre di buffer di input e/o output per i macchinari di lavorazione ad esempio modifica completamente l'impostazione richiesta dai problemi di sequenziamento. Brauner et al. [5] studiano il problema della sola presenza di buffer di output su ciascun macchinario per un layout lineare, Drobouchevitch et al. [6] analizzano il medesimo problema per il layout circolare, dove con layout circolare si intende la capacità del robot di ruotare completamente su sè stesso. Le analisi di celle con entrambi i tipi di buffer sui macchinari sono più recenti [7] e si concentrano in particolare sul confronto fra l'uso di buffer o quello di *dual-gripper robots*. Disporre di un end-effector che consenta la movimentazione di due componenti contemporaneamente è un notevole

vantaggio per la cella robotizzata, introduce tuttavia una complicazione nell'identificazione della sequenza ottimale di movimenti. Anche gli studi che analizzano l'utilizzo di pinze doppie si differenziano in base al layout: circolare [8] o lineare [9] [10].

Un'ulteriore complessità della configurazione può essere fornita dal numero di macchinari presenti e dalla loro configurazione. Un'estensiva analisi per il caso con 3 macchinari di lavorazione è stata realizzata da Gultekin et al. [11]. Tale studio analizza una cella in cui le lavorazioni possono essere effettuate in ordine qualsiasi e dove ciascun macchinario è in grado di realizzare una qualunque delle operazioni tramite un veloce cambio utensile tipico dei macchinari CNC. Un'ulteriore complicazione ai problemi di sequenziamento deriva dall'utilizzo di macchinari in parallelo [12], ovvero l'utilizzo di postazioni multiple identiche. Tipicamente la postazione raddoppiata è quella che rappresenta il collo di bottiglia per la produzione. Non mancano gli equivalenti studi con robot multiplo all'interno della stessa cella o con la necessità di movimentare componenti differenti fra loro [13].

Si accenna infine ad una categoria di articoli che analizzano la sequenza di movimenti ottimale nel caso di macchinari di lavorazione di tipologia *no-wait*. Tali macchinari richiedono il prelievo del pezzo immediatamente al termine della lavorazione e non ammettono un'attesa per ragioni legate a lavorazioni a caldo che non consentono il raffreddamento del pezzo. Agnetis [14] analizza il caso per due postazioni di lavorazione e dimostra la complessità computazionale (*NP-difficile*) per tutti i casi con un numero maggiore di macchinari.

Si noti che in tutti gli studi citati si presume che un nuovo componente risulti sempre disponibile e prelevabile sulla postazione di input. La postazione di scarico è inoltre considerata sempre disponibile ad accogliere un nuovo componente al termine del ciclo. Tali circostanze non sono rispettate da numerose celle di lavoro industriali e sono argomento di analisi della presente tesi.

1.3 Layout di cella in esame

Il compito della cella di lavoro in esame è quello di prelevare componenti da una postazione di ingresso pezzi e porli su un macchinario designato ad una generica lavorazione. Al termine dell'operazione il componente lavorato deve essere riposto su una postazione adibita allo scarico. Poichè, come si avrà modo di specificare, la disponibilità di componenti in ingresso risulta non costante, è utile implementare una postazione ausiliaria che funga da buffer. Tale postazione consente il deposito temporaneo di componenti non lavorati in vista di un prelievo successivo al momento della mancanza di pezzi nella postazione di input. Si analizzeranno ora i vari dispositivi e postazioni presenti nella cella e i parametri da loro dipendenti che verranno utilizzati nei successivi capitoli.

1.3.1 Robot e pinza

La cella considerata prevede l'utilizzo di un unico manipolatore senza limitazioni sulla sua tipologia. E' sufficiente che il robot sia in grado di afferrare e rilasciare i componenti da movimentare e di raggiungere qualsiasi postazione presente all'interno della cella. Sulla flangia robot è previsto l'utilizzo di una pinza singola, ovvero un *end-effector* in grado di effettuare presa e rilascio di un singolo componente alla volta.

Le pinze industriali sono costituite da una flangia con annesse valvole pneumatiche o oleodinamiche controllate tramite segnali output del controller. Il gruppo pinza viene in seguito reso solidale alla flangia robot tramite appositi fori disposti sulla flangia robot dal produttore di manipolatori. I parametri considerati relativi al robot sono le tempistiche di movimento tra le postazioni e il tempo di presa/rilascio pezzi.

1.3.2 Feeder flessibile

Nella cella in esame è presente una postazione di prelievo pezzi, o *feeder*, che rende disponibili i componenti al manipolatore tramite movimentazione dall'esterno della cella. I componenti forniti sono tutti uguali fra loro. Il feeder considerato è flessibile, ovvero i componenti vengono forniti in numero limitato e, in seguito al



Figura 1.4: Esempio di robot Mitsubishi con diversi end-effector.

prelievo dell'ultimo di essi, è necessario attendere un tempo di rinnovo prima di disporre di ulteriori componenti. Il feeder flessibile si contrappone ai tradizionali *bowl-feeder*, dispositivi in grado di fornire costantemente un pezzo al manipolatore, garantendo per di più posizione e orientamento desiderati. Questi ultimi sono dispositivi che consistono in tazze vibranti riempite con componenti alla rinfusa da un operatore o da sistemi ausiliari automatizzati. Tramite un apposito moto sussultorio i componenti risalgono lungo un percorso a spirale presente all'interno della tazza. Lungo tale percorso sono disposte delle trappole appositamente progettate. Tramite esse i componenti che presentano un'orientazione non conforme a quella desiderata vengono riconosciuti e obbligati a cadere nuovamente al centro della tazza. I bowl-feeder sono dispositivi che richiedono una progettazione apposita a seconda del componente movimentato, di conseguenza costosi e per nulla flessibili.

L'utilizzo di feeder flessibili consente di ridurre il costo di realizzazione della cella robotizzata e utilizzare la medesima strumentazione nell'eventuale riconfigurazione del ciclo di lavoro. Un esempio di feeder flessibile è un semplice nastro trasportatore: a monte di esso un operatore carica i componenti. Solo un'estremità raggiunge l'interno della cella, consentendo un agevole caricamento esterno

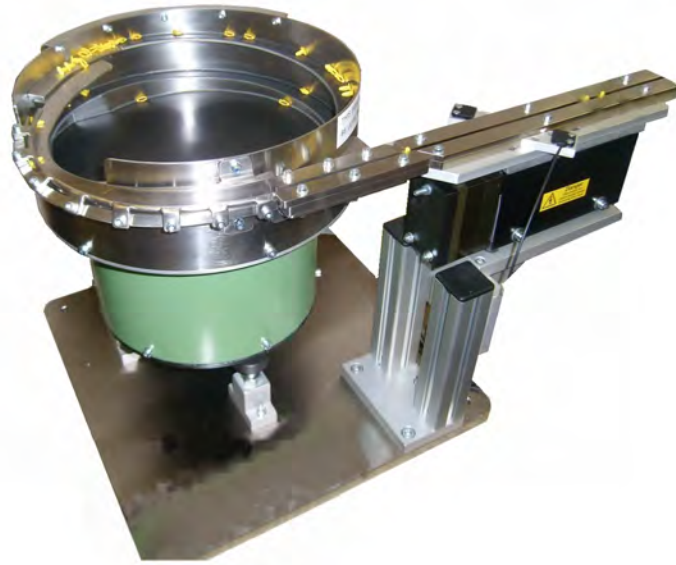


Figura 1.5: Bowl-Feeder prodotto dall'azienda Prometica Ltd.

senza richiedere la presenza dell'operatore nella cella. Un sistema di visione inquadra la porzione di nastro interna alla cella, individua i componenti e fornisce al controller le relative coordinate per consentirne la presa. Ciclicamente viene richiesto l'avanzamento del nastro di una quantità fissata. Il numero di componenti presenti ad ogni rinnovo del feeder, o in questo caso avanzamento del nastro, può variare a seconda della tipologia di caricamento a monte e a seconda dell'effettivo riconoscimento da parte del sistema di visione. Un tale sistema non richiede alcuna progettazione e consente totale flessibilità. Un altro esempio di feeder flessibile è dato da alimentazione di componenti tramite pallet distinti.

La stazione di ingresso pezzi considerata nella presente tesi è caratterizzata pertanto dai seguenti parametri: tempo richiesto per il rinnovo, numero di pezzi disponibile ad ogni rinnovo e, se questo non è costante, l'eventuale distribuzione di probabilità di numero di pezzi disponibile.

1.3.3 Postazione di lavorazione

La postazione di lavorazione è costituita da un generico macchinario che realizza una lavorazione sul componente, non si è interessati, nella situazione in esame, a

quale possa essere la natura dell'operazione. La postazione è in grado di lavorare su un unico pezzo per volta. Inoltre è richiesto il rilascio del pezzo da parte del robot. Il manipolatore, in seguito all'apertura delle griffe, può pertanto effettuare altre operazioni, non è richiesta la sua sosta sopra la postazione. Si considera per semplicità una postazione in cui la posizione di rilascio pezzo grezzo e afferraggio dello stesso in seguito alla lavorazione coincidano. La postazione è caratterizzata da una tempistica necessaria per terminare l'operazione e fino al termine di essa non è possibile recuperare il pezzo.

Ci sono numerose operazioni industriali riconducibili al caso in esame. Per citare alcuni esempi essa può consistere in una stazione di piegatura, di saldatura o di montaggio.

L'unico parametro che caratterizza la postazione di lavorazione nel presente studio è la tempistica richiesta per la lavorazione del pezzo.

1.3.4 Postazione di scarico

La postazione di scarico rappresenta la locazione in cui è necessario posizionare i pezzi lavorati per consentirne la movimentazione all'esterno della cella. In seguito al rilascio del pezzo è necessario attendere un tempo di ricambio della postazione prima di poter scaricare il successivo componente lavorato. Un esempio ampiamente implementato in ambito industriale è la giostra girevole, un dispositivo in grado di accogliere più pezzi lungo la periferia e di ruotare per consentire il deposito di un nuovo componente. È frequente anche l'utilizzo di supporti scorrevoli su rotaia in grado di contenere un pezzo lavorato ciascuno. Solitamente queste soluzioni sono utilizzate se vi è un'ulteriore cella robotizzata poichè consentono un preciso posizionamento del pezzo, semplificando la presa per la successiva operazione. Se il prodotto lavorato non deve subire ulteriori trattamenti automatizzati è utilizzato anche lo scarico su generico nastro scorrevole. A valle di esso gli operatori effettuano il controllo qualità e gestiscono il confezionamento.

Generalmente la postazione di scarico è posizionata all'estremità della cella opposta a quella di ingresso pezzi per questioni di flusso, fra queste due viene invece posto il macchinario designato alla lavorazione.

Si presume la postazione di scarico in grado di accogliere un solo componente alla volta. Pertanto l'unico parametro considerato relativo alla postazione è il tempo richiesto per la movimentazione all'esterno della cella del pezzo lavorato.



Figura 1.6: Esempio di sistema di scarico pezzo su rotaia prodotto dall'azienda Rollon SPA. Dopo il deposito del componente lavorato il relativo supporto avanza sul percorso verso la stazione successiva.

1.3.5 Sequenziamento dinamico

La cella in esame è caratterizzata da una gestione dinamica dei movimenti del robot. Non vi è un ciclo di lavoro stabilito a priori: i comandi impartiti al robot vengono stabiliti al termine di ciascuna operazione a seconda della situazione nella cella. Nel codice che gestisce la cella viene assegnata una priorità per ciascuna operazione e, in base alla disponibilità istantanea delle postazioni, si determina la seguente operazione del robot. Tale gestione viene ottenuta analizzando ciclicamente la situazione dei segnali forniti dai vari sistemi implementati nella cella. A livello pratico le postazioni presenti all'interno di una cella dialogano con il manipolatore tramite PLC o collegamento diretto al controller. Una postazione può ad esempio assumere gli stati occupata/disponibile, caso tipico di un macchinario di lavorazione che stia lavorando o meno un componente. È indispensabile implementare la possibilità di inviare segnali hardware per dialogare con il sistema di controllo del manipolatore e indicare lo stato di ciascuna postazione. Un comune segnale hardware utilizzato per effettuare la comunicazione è la variazione di

voltaggio ad un capo della connessione. Tale segnale viene confrontato con una tensione di riferimento e in base all'esito può assumere i valori definiti, in termini elettronici, *alto* o *basso*. Tipicamente la gestione dei segnali avviene con tensioni di 24V.

1.3.6 Buffer

Se all'interno della cella non fosse posizionato un buffer le operazioni che costituiscono il ciclo sarebbero le seguenti:

- Prelievo di un pezzo dal feeder;
- Deposito sul macchinario di lavorazione;
- Prelievo del componente al termine dell'operazione;
- Deposito nella postazione di scarico.

Le possibilità di ottimizzazione del tempo ciclo sono nulle, il robot è vincolato ad effettuare tali movimenti in sequenza. Poichè il feeder è flessibile un tale approccio implicherebbe un'attesa del robot durante il tempo di ricambio senza la possibilità di effettuare altre operazioni. Risulta pertanto vantaggiosa l'aggiunta di una postazione di bufferizzazione in cui depositare temporaneamente i pezzi, nel caso preso in considerazione non ancora lavorati, in modo da svuotare la postazione di ingresso in anticipo e disporre di componenti grezzi anche durante il suo rinnovo. L'operazione di cambio feeder avviene in questo modo in *tempo mascherato*. Un'operazione è detta avvenire in tempo mascherato se il manipolatore non deve attendere il completamento. Tale situazione costituisce un vantaggio per la produttività della cella.

La sequenza di movimenti è determinata dinamicamente. Utilizzando una postazione di buffering sorge pertanto il problema di quale priorità assegnare alle diverse postazioni: nel caso in cui un pezzo grezzo sia disponibile sia nel feeder sia sul buffer è necessario stabilire quale prelievo ha la precedenza. La priorità di prelievo globale è associata alla lavorazione: se un pezzo ha terminato la lavorazione e la postazione di scarico è in grado di accoglierlo il manipolatore preleva il componente lavorato. In caso negativo il manipolatore preleva il componente grezzo,

se disponibile, dalla postazione a cui è stata attribuita la priorità successiva. Uno degli scopi del presente studio è valutare quando è conveniente l'assegnazione alla priorità di svuotamento al buffer e quando al feeder.

Non è di secondaria importanza la determinazione della dimensione del buffer, ovvero il numero di slot di cui dispone la postazione. Tale valore rappresenta il numero di componenti che è in grado di accogliere contemporaneamente la postazione. Questa è costituita generalmente da una serie di supporti progettati ad hoc per il pezzo da movimentare distanziati in modo da consentire il deposito e prelievo senza incorrere in collisioni indesiderate. A seconda della forma del componente è possibile tuttavia realizzare supporti che consentano di sovrapporre parzialmente i pezzi. Le possibili strategie di sequenza per il prelievo e deposito nel buffer sono irrilevanti per il presente studio. La scelta fra gestione FIFO (First-in-first-out) o FILO (First-in-last-out) non verrà trattata in quanto si assumeranno per semplicità le differenti posizioni di buffering coincidenti in un solo punto.

1.3.7 Disposizione delle postazioni

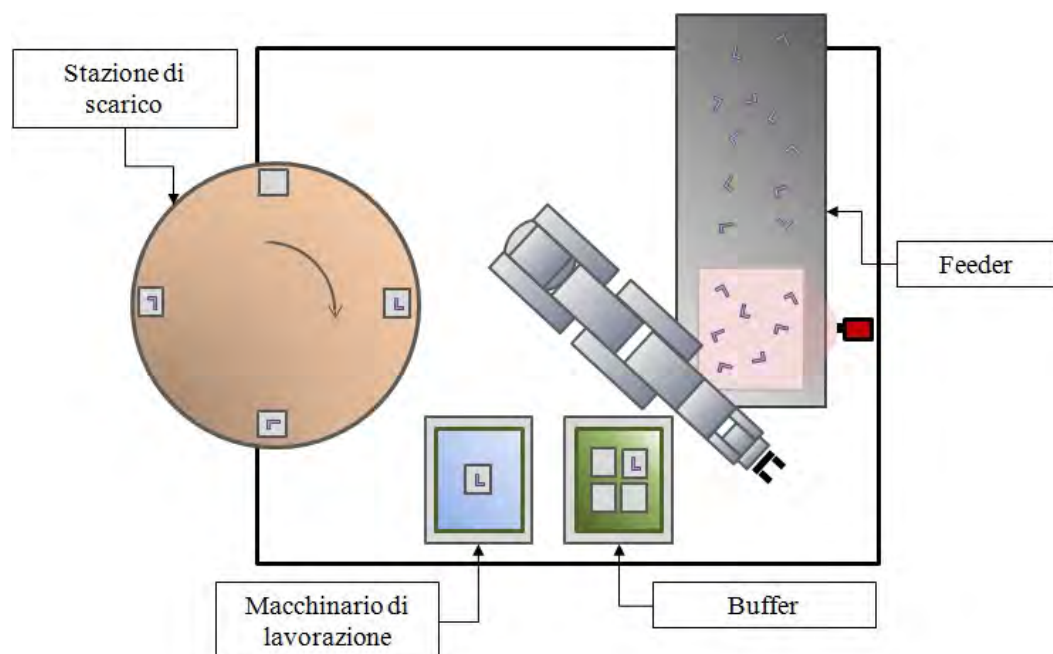


Figura 1.7: Layout della cella in esame.

Il metodo di analisi che si utilizzerà per il calcolo del tempo ciclo potrebbe essere utilizzato indipendentemente dalla disposizione relativa delle postazioni all'interno della cella. Tuttavia, in accordo con la pratica industriale, le postazioni si suppongono posizionate a semicerchio o in linea. Per ragioni dettate da ingombri, la postazione del feeder e quella di scarico sono situate agli estremi delle relative configurazioni. Lo scopo del buffer è ospitare componenti provenienti dal feeder e fornire pezzi per portarli in lavorazione. Di conseguenza la scelta più razionale consiste nel vincolare la sua posizione fra la postazione di lavorazione e il feeder. I vantaggi derivanti da un suo posizionamento ravvicinato a una delle due postazioni anziché in posizione intermedia vengono analizzati in sezione 7.2.

1.3.8 Posizione di home

Vi possono essere casi in cui il manipolatore non ha operazioni disponibili da compiere e rimane in attesa di comandi. Questo caso si presenta, ad esempio, dopo aver rilasciato un componente sul macchinario di lavorazione se il feeder è in fase di rinnovo. Nella gestione di celle robotizzate è buona prassi indicare una posizione di *home* in cui posizionare il manipolatore in tali situazioni. È possibile indicare più posizioni di home complicando notevolmente il codice robot e la sequenza di movimenti ma si trascura questa possibilità. La posizione di home si suppone situata nei pressi della lavorazione, ad esempio ad una quota z superiore. Tale scelta consente di afferrare un componente nel minor tempo possibile al termine della lavorazione e risulta pertanto la soluzione ottimale.

1.3.9 Parametri in esame

Si riporta un riassunto dei parametri variabili nel layout di cella in esame.

- Tempistiche robot:
 - Tempo di spostamento fra postazioni;
 - Tempo di presa/rilascio pezzi.
- Tempistiche delle postazioni:

- Tempo di rinnovo feeder;
- Tempo di lavorazione;
- Tempo di scarico pezzi.

- Dimensione postazioni:
 - Numero di pezzi presenti sul feeder con eventuale distribuzione di probabilità se tale valore non è costante;
 - Dimensione del buffer.

- Scheduling:
 - Priorità di svuotamento associate alla postazione di buffering o al feeder.

Capitolo 2

Analisi di cella robotizzata

2.1 Classificazione di Graham-Dawande

Un qualsiasi problema di analisi di cella robotizzata può essere descritto in forma compatta utilizzando la classificazione di Graham-Dawande. Tale metodo di classificazione, ampiamente adottato in letteratura per indicare il layout della cella in esame, è proposto da Dawande [15], estendo quello introdotto da Graham [16] per i problemi generici di sequenziamento. Il problema viene descritto mediante tre distinti parametri: le caratteristiche della cella (α), quelle del ciclo produttivo (β) e la funzione obiettivo (γ). Il problema in analisi viene indicato nella forma compatta $\alpha|\beta|\gamma$. Si esplicherà ora nel dettaglio il significato dei parametri.

2.1.1 Parametro α

Il parametro α riporta tutto ciò che è inerente all'ambiente di lavoro: numero di manipolatori presenti nella cella, tipologia di pinza montata, numero di macchinari presenti e relativo layout. L'aspetto del parametro è:

$$\alpha = RF_{m,r,b}^{g,l}(m_1, \dots, m_m)$$

- RF = Robotic Flowshop;
- m = numero di postazioni di lavorazione;
- m_i = numero di macchinari in parallelo nella postazione i -esima;

- r = numero di manipolatori;
- $\bar{b} = (b_1, \dots, b_m)$ = dimensione dei buffer in uscita a ciascuna postazione, se presenti;
- g = numero di pinze sulla flangia robot;
- l = layout (\sqcup per lineare o semicircolare, \circ per circolare).

Il layout circolare non è considerato in questo progetto poichè si suppone l'utilizzo di manipolatori industriali incapaci di ruotare completamente su sè stessi. La caratteristica che contraddistingue il layout circolare in una cella robotizzata è la possibilità di movimentare i componenti verso una postazione utilizzando due differenti percorsi. La scelta ottimale fra le due opportunità ad ogni movimento complica ulteriormente l'analisi della cella.

2.1.2 Parametro β

Nel parametro β sono riportate le caratteristiche del ciclo produttivo. Ogni macchinario di lavorazione è caratterizzato da un criterio di prelievo che lo contraddistingue. Si parla di *free pick-up* se non vi è un limite di tempo oltre cui il pezzo lavorato può rimanere sulla postazione di lavorazione, di *no-wait pick-up* se è richiesta la rimozione non appena termina il processo, e infine di *interval pick-up* se l'afferraggio del componente può avvenire in una finestra temporale successiva al termine dell'operazione. In quest'ultimo caso, si indica con $[a_i, b_i]$ la finestra di lavoro relativa al macchinario M_i , dove il componente richiede una lavorazione di durata a_i e il tempo concesso per il prelievo risulta pari a $(b_i - a_i)$. Esempi per il caso no-wait sono le lavorazioni con materiale che richiede il mantenimento ad una data temperatura come i processi inerenti la lavorazione di metalli a caldo o lo stampaggio di plastiche. La modalità interval pick-up è utilizzata ad esempio in processi con immersione in solventi.

Nel parametro beta è indicato inoltre il numero di componenti differenti processati dalla cella. Vi possono infatti essere situazioni in cui i pezzi in ingresso siano differenti fra di loro e richiedano tempistiche di lavorazione variabili. Le

restanti informazioni in β riguardano le strategie di produzione, ininfluenti nel caso in esame, e le assunzioni in merito alle tempistiche di movimentazione.

Il parametro β viene espresso nella forma:

$$\beta = (\text{pick-up}, \text{travel-metric}, \text{part-type}, \text{production-strategy})$$

- pick-up = *free, no-wait, interval*;
- travel-metric = E (*Euclidean*), C (*constant*), A (*additive*);
- part-type = MP (*multiple parts*), SP (*single part*);
- production-strategy = *all* nel caso in esame.

2.1.3 Tempistiche di movimentazione

Un importante aspetto indicato nel parametro β consiste nel valore *travel-metric* che descrive la tempistica di spostamento fra stazioni. Uno schema comunemente adottato nell'analisi di celle robotizzate prevede la disposizione delle postazioni numerate in linea o in semicerchio, M_0, \dots, M_{m+1} , dove M_0 rappresenta la postazione di input, M_{m+1} quella di output e le altre i macchinari. Indicheremo con $d(M_i, M_j) = \delta_{ij}$ il tempo di spostamento fra le stazioni M_i e M_j . Il modello più generale che si può considerare assume ciascun tempo di movimentazione δ_{ij} un parametro indipendente. Brauner et al. [17] analizzano tale modello generale e definisce la cella robotizzata *Euclidea* se sono soddisfatte le seguenti condizioni:

- Il tempo richiesto per la movimentazione da una postazione a sè stessa è nullo ($\delta_{ii} = 0, \forall i$);
- Il tempo di movimentazione fra più stazioni soddisfa la disuguaglianza triangolare ($\delta_{ij} + \delta_{jk} \geq \delta_{ik}, \forall i, j, k$).

Il termine cella robotizzata *a simmetria Euclidea* è attribuito se è soddisfatta anche la condizione di simmetria $\delta_{ij} = \delta_{ji}, \forall i, j$. Considerare tempi di movimentazione indipendenti tra loro complica l'analisi, a causa dell'elevato numero di parametri in gioco. Per questa ragione è diffuso l'utilizzo di assunzioni relativi a tempistiche di movimentazione costanti o additive. A tali assunzioni è usualmente

associata la condizione di simmetria e l'invarianza delle tempistiche rispetto alla presenza o meno del payload sull'end effector.

In alcuni casi particolari la disposizione delle postazioni all'interno della cella è tale da poter considerare costante il tempo richiesto da qualsiasi movimentazione [18] con un ridotto errore di approssimazione. In questi casi il moto fra una coppia di postazioni, anche se non adiacenti, è considerato pari a δ .

$$d(M_i, M_j) = \delta, \quad 0 \leq i, j \leq m + 1, \quad i \neq j. \quad (2.1)$$

Tale assunzione è indicata con *metrica costante di movimentazione*. Nei pratici è più frequentemente verificata invece l'additività delle tempistiche di movimentazione. Sia δ_i il tempo richiesto per la movimentazione fra le postazioni adiacenti M_{i-1} e M_i , $\forall i = 1, \dots, m + 1$. La tempistica richiesta per effettuare un movimento tra postazioni non adiacenti è calcolabile come somma dei distinti moti.

$$d(M_i, M_j) = \sum_{k=i+1}^j \delta_k, \quad \forall i < j \quad (2.2)$$

È possibile semplificare ulteriormente il modello considerando uguali le tempistiche fra due stazioni adiacenti $\delta_i = \delta$, $\forall i = 1, \dots, m + 1$. In tal caso

$$d(M_i, M_j) = |i - j|\delta \quad (2.3)$$

Come si avrà modo di giustificare con risultati sperimentali, l'ipotesi di movimentazione additiva non sempre è sufficientemente accurata. Una migliore assunzione proposta da Logendran et al. [19] prendono in considerazione un contributo di accelerazione e decelerazione fra moti che riguardano postazioni non adiacenti. La proposta consiste nell'introduzione del parametro η indicante la quota temporale risparmiata, nel moto tra due stazioni non adiacenti, grazie al passaggio da una postazione senza effettuare la sosta (esempio: $\delta_{13} = \delta_{12} + \delta_{23} - \eta$).

$$d(M_i, M_j) = \sum_{k=\min(i,j)+1}^{\max(i,j)} \delta_k - (|i - j| - 1)\eta, \quad 0 \leq i, j \leq m + 1 \quad (2.4)$$

L'assunzione che il valore η sia indipendente dalla postazione in esame deriva dall'ipotesi che il moto del robot fra due stazioni adiacenti sia tale da prevedere un tratto centrale a velocità costante. Tale ipotesi è in accordo con la tipologia

di movimentazione utilizzata nei manipolatori industriali. Nei sistemi di controllo di questi robot vengono infatti implementate leggi primitive di moto a *trapezio di velocità* caratterizzate da un tratto lineare centrale [20].

Si può rappresentare l'equazione 2.4 nel caso di tempistiche fra stazioni adiacenti costanti e pari a δ .

$$d(M_i, M_j) = |i - j|\delta - (|i - j| - 1)\eta \quad (2.5)$$

Secondo tale assunzione ad esempio $\delta_{13} = 2\delta - \eta$ e $\delta_{14} = 3\delta - 2\eta$. La verifica sperimentale della sua attendibilità verrà analizzata in sezione 6.1.1.

2.1.4 Funzione obiettivo γ

γ indica l'obiettivo di minimizzazione dell'analisi. In una cella robotizzata è frequente porre come scopo la minimizzazione del tempo ciclo o, analogamente, la massimizzazione della produttività μ . Questa richiesta non è tuttavia da considerarsi l'unica possibile in ambito industriale. La convenienza economica in alcuni casi non è legata alla sola cadenza di produzione. È possibile ad esempio effettuare l'ottimizzazione di ciclo produttivo in cui l'obiettivo è la minimizzazione dell'usura del manipolatore. L'aumento della manutenzione richiesta dal robot e una vita di utilizzo minore dello stesso possono infatti risultare sconvenienti rispetto ad una maggiore produttività. Altri studi considerano invece il costo della fornitura elettrica e ricercano il valore minimo di corrente assorbita dal manipolatore ad ogni ciclo. In particolare Shrouf et al. [21] hanno analizzato la minimizzazione della spesa per la fornitura elettrica suddividendo il lavoro del sistema robotizzato in base al differente costo di un kWh nell'arco della giornata.

2.1.5 Classificazione della cella in esame

Il layout preso in esame è schematizzabile come segue:

- α = un robot, pinza singola, un solo macchinario di lavorazione e layout semicircolare. La notazione non prevede una postazione di solo buffering, pertanto arbitrariamente si decide di indicare con $b(0) = n$ tale aspetto;

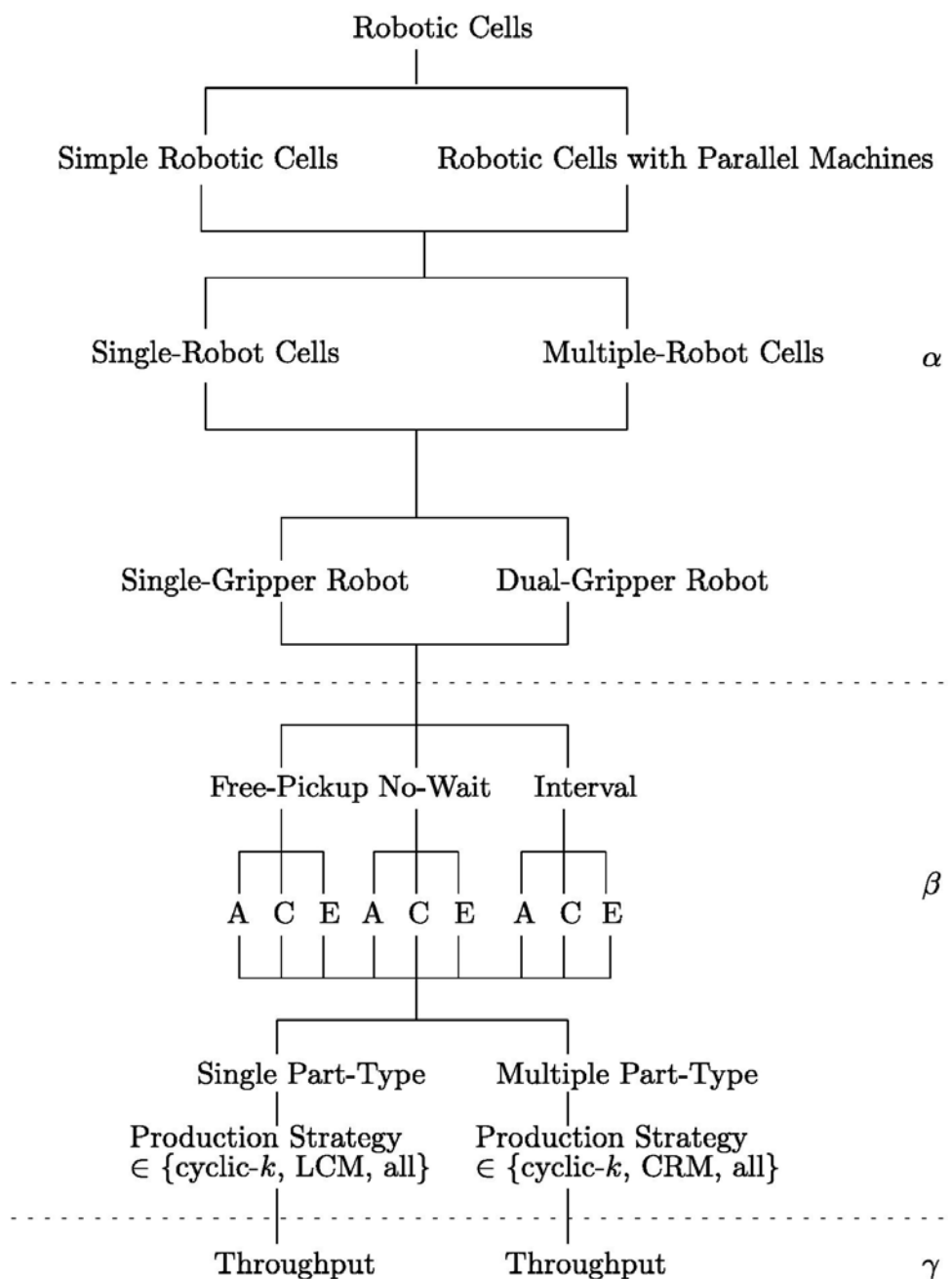


Figura 2.1: Schema riassuntivo della classificazione di celle robotizzate [2].

- β = free pick-up, movimentazione additiva, una sola tipologia di componenti, qualsiasi strategia di produzione ammessa;
- γ = tempo ciclo di lavoro.

In definitiva il problema è classificabile nel seguente modo:

$$RF_{1,1,b(0)=n}^{1,\sqcup} | (free, A, SP, all) | \mu \quad (2.6)$$

Tale classificazione è prevista per indicare analisi di celle in cui i componenti sono sempre disponibili nella stazione di input e con costante disponibilità della postazione di scarico. Non è tuttavia questa la situazione del caso in esame. In assenza di convenzioni per descrivere tale situazione ci si limita alla sola descrizione testuale di tali aspetti.

2.2 Metodi di analisi

2.2.1 Stati della cella

Dopo aver classificato il problema in esame si procede con la descrizione delle modalità di analisi della cella. La sequenza di movimenti del robot e di conseguenza la produttività della cella può essere analizzata come successione di stati. Una descrizione matematicamente accurata di uno stato della cella è rappresentata dalle informazioni seguenti:

- Configurazione spaziale del robot;
- Presenza del componente o meno sull'end effector;
- Numero di componenti su ciascuna postazione;
- Tempo rimanente per le postazioni che prevedono una tempistica (ad esempio la lavorazione del componente).

Parte delle informazioni elencate sono superflue per la maggioranza dei problemi di analisi di cella robotizzata. Ci si limita di conseguenza ad indicare il solo stato delle postazioni e l'ultima azione compiuta dal robot. Considerando tale semplificazione una notazione proposta per descrivere uno stato della cella [4] prevede l'utilizzo di un vettore di lunghezza pari al numero di postazioni mag-

giorato di un'unità. Ciascuna componente del vettore descrive la situazione per la postazione i -esima e l'ultima rappresenta l'azione più recente del robot.

$$(e_1, e_2, e_3, \dots, e_{m+1}) \quad (2.7)$$

Nel caso in esame si assume una notazione simile, indicando con e_1 lo stato della postazione di input, con e_2 invece lo stato del buffer. I valori assunti rappresentano il numero di componenti presenti nella postazione. e_3 rappresenta la postazione di lavorazione e e_4 quella di scarico. Tali parametri assumono il valore Ω se la rispettiva postazione risulta non disponibile, \emptyset in caso contrario. Ad esempio lo stato rappresentato dalla notazione $(2, 4, \Omega, \emptyset, L_L)$ rappresenta la situazione in cui sono presenti 2 componenti nel feeder e 4 nel buffer. La postazione di lavorazione risulta occupata e quella di scarico disponibile ad accogliere un componente. L'ultima azione effettuata dal manipolatore coincide con il rilascio di un componente sulla postazione di lavorazione. In generale, nota la situazione iniziale della cella, una successione di movimenti del robot può essere completamente definita utilizzando il solo termine $m+1$ del vettore di stato, ovvero l'ultima azione compiuta dal manipolatore.

Nel caso in esame le operazioni possibili sono:

- prova
- Prelievo da feeder (F_U);
- Deposito su buffer (B_L);
- Prelievo da buffer (B_U);
- Deposito sul macchinario di lavorazione (L_L);
- Prelievo dal macchinario di lavorazione (L_U);
- Deposito su postazione di scarico (S_L).

I pedici corrispondono alle abbreviazioni dei termini load e unload. A queste operazioni risulta utile aggiungere due azioni inerenti la postazione fittizia di home. In particolare si considera il raggiungimento della posizione di home e

l'allontanamento da questa (H_R - Home reach e H_L - Home leave). In figura 2.2 sono riportati gli stati previsti e la loro possibile successione.

2.2.2 Attività della cella

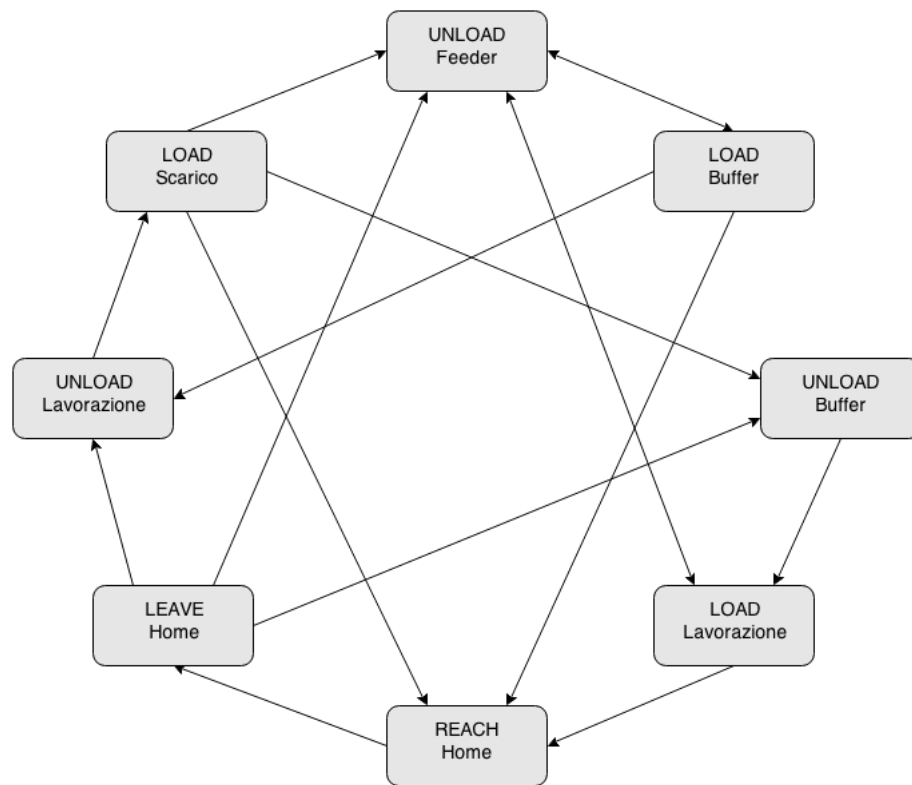


Figura 2.2: Diagramma con gli stati della cella e le relazioni fra di essi.

Si accenna alla possibilità di utilizzare una differente notazione per analizzare la cella sfruttando il concetto di attività anziché di stato. Un'analisi basata su attività semplificherebbe il problema nel caso di macchinari multipli e necessità di processare un medesimo componente sui differenti macchinari in ordine. Nel caso in esame si dispone di un solo macchinario pertanto tale metodo di analisi della cella risulta inadeguato. Un'attività A_i viene definita come successione delle seguenti operazioni:

- Prelievo di un pezzo dalla postazione M_i ;
- Spostamento da M_i a M_{i+1} ;

- Deposito di un pezzo sulla postazione M_{i+1} .

2.3 Tempistiche fra stati della cella

Come esposto in sezione 2.1.3 si considera una movimentazione con tempistiche additive utilizzando l'espedito del parametro η per considerare la quota di accelerazione derivante dal moto fra stazioni non adiacenti. Il tempo richiesto per il rilascio e l'afferraggio dei componenti è indicato con ϵ e si suppone indipendente dalla postazione in cui tale operazione viene effettuata. Con questa notazione la tempistica richiesta per il passaggio dallo stato $(2, 4, \Omega, \emptyset, L_L)$ allo stato $(1, 4, \Omega, \emptyset, F_U)$ è indicata con $2\delta - \eta + \epsilon$.

In precedenza la posizione di home è stata arbitrariamente posizionata nei dintorni della lavorazione. A livello pratico questa affermazione indica il posizionamento, come frequente, ad una quota z superiore al piano di lavoro. Questa assunzione permette di considerare un solo parametro per gli spostamenti che riguardano la locazione di home. Con il termine τ_H ci si riferisce al tempo richiesto per effettuare il movimento dalla posizione di home alla posizione di lavorazione e viceversa. I movimenti da una postazione diversa da quella di lavorazione verso la posizione di home si suppongono pari ai tempi richiesti per muoversi verso la postazione di lavorazione. La stessa considerazione può essere espressa per i moti dalla posizione di home verso le altre postazioni. La semplice assunzione di considerare la home nei pressi della lavorazione permette una facile gestione delle tempistiche di movimenti senza introdurre 4 nuovi parametri (δ_{HF} , δ_{HB} , δ_{HL} e δ_{HS}).

In figura 2.3 sono riportate le tempistiche fra le attività previste per la cella. L'attesa nella posizione di home è funzione della situazione presente nella cella e non è determinabile a priori. Nel diagramma è stata indicata con il generico termine w , *wait*.

Si può notare che le tempistiche considerate sono indipendenti dal numero di componenti presenti nella stazione di partenza e destinazione. Le postazioni di feeder e buffer possono infatti accogliere più componenti e ciascuno di questi avrà una propria locazione spaziale. Considerando una tempistica unica si suppone

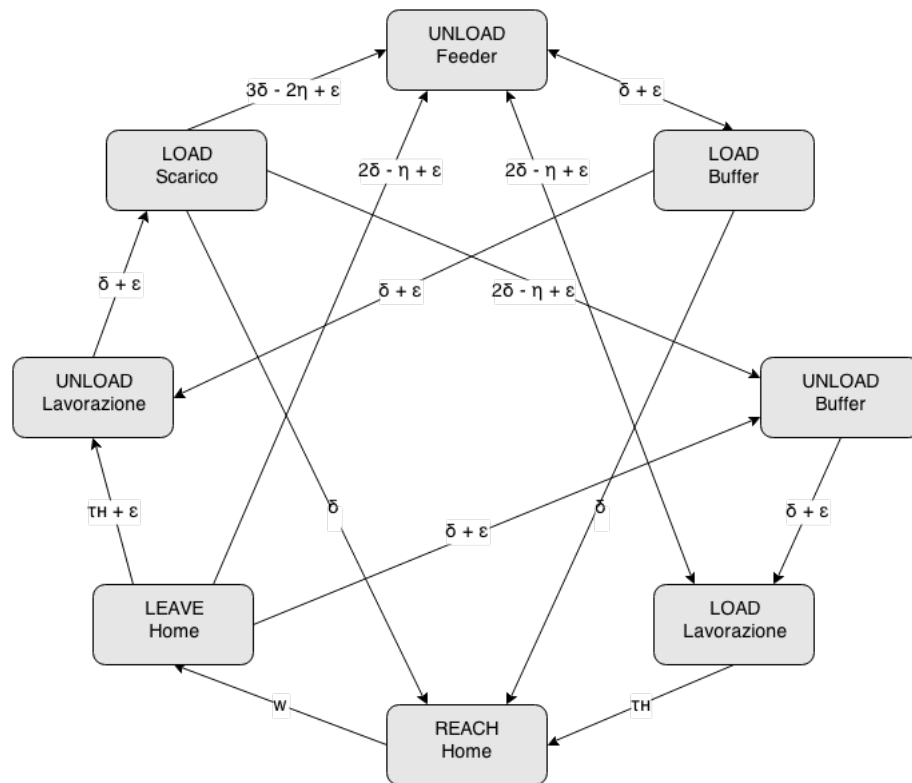


Figura 2.3: Diagramma con le tempistiche richieste per passare da uno stato all'altro.

che le postazioni possano essere trattate come puntiformi. Si presume che i componenti vengano cioè prelevati e rilasciati nella medesima locazione. Tale punto può essere con buona approssimazione essere identificato con il centro geometrico delle differenti locazioni reali. L'attendibilità di tale assunzione è verificata sperimentalmente in sezione 6.1.2.

2.3.1 Carta robot-macchina

Un approccio alternativo all'analisi della cella è l'utilizzo della carta robot-macchina. Tale metodo consiste in una rappresentazione visiva delle attività di ciascun dispositivo interno alla cella nel corso del tempo. Tale approccio è frequentemente utilizzato in fase preliminare di sistemi produttivi per valutare il numero di macchinari supervisionabili da uno o più operatori (da qui *carta uomo-macchina*). Nel caso in esame non sono previsti operatori ma è comunque possibile sostituire

alla loro rappresentazione il manipolatore. La carta robot-macchina consiste in una successione di linee incolonnate, ciascuna riferita ad un differente dispositivo interno alla cella. Ciascuna linea rappresenta lo scorrere del tempo e indica le attività in corso per quel dispositivo [22]. La carta robot-macchina consente una valutazione visiva del processo e l'identificazione immediata dei relativi aspetti critici.

In figura 2.4 è rappresentato un esempio di carta robot-macchina per la cella in esame. Si supponga che lo stato iniziale nella cella sia $(3, 0, \emptyset, \emptyset, H_L)$. Viene prelevato un componente nel feeder, riposto in lavorazione e in seguito, durante l'attesa del termine dell'operazione, due componenti vengono immessi nel buffer in modo da svuotare il feeder. Si può notare la dicitura *Working* per il dispositivo di input in seguito al prelievo dell'ultimo componente, indicante il processo di rinnovo. È interessante osservare come l'attesa del macchinario di lavorazione in seguito al termine dell'operazione dipenda dall'entità della bufferizzazione. Nella prima lavorazione rappresentata vengono inseriti due componenti nel buffer, nella seconda vi è l'attesa in home. L'attesa del componente lavorato è maggiore nel primo caso, poichè la doppia bufferizzazione è realizzata solo in parte in tempo mascherato.

| | | | | | | | | | | | | | | | |
|-------------|----------------|---------------|----------------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|----------|---------------|--|
| Feeder | | U | | U | | U | WO | | | | | | | | |
| Buffer | | | | | | | L | | | | | | | L | |
| Macchinario | | | | | L | WO | | | | | | WA | | U | |
| Scarico | | | | | | | | | | | | | | | |
| Robot | $2\delta-\eta$ | ε | $2\delta-\eta$ | ε | δ | ε | δ | ε | δ | ε | δ | ε | δ | ε | |

| | | | | | | | | | | | | | | | | |
|-------------|----------|---------------|----------------|---------------|----------|---------------|----------|--|--|--|--|----|--|----------|---------------|----------|
| Feeder | WO | | | | | | | | | | | | | | | |
| Buffer | | | | | | | | | | | | U | | | | |
| Macchinario | | | | | L | WO | | | | | | WA | | U | | |
| Scarico | | L | WO | | | | | | | | | | | | | |
| Robot | δ | ε | $2\delta-\eta$ | ε | δ | ε | τ^H | | | | | | | τ^H | ε | δ |

Figura 2.4: Esempio di carta robot-macchina per la cella in esame. U = Unload, L = Load, WO = Working, WA = Waiting, attesa di prelievo del componente lavorato.

2.4 Risoluzione analitica

Tramite l'analisi del diagramma in figura 2.3 è possibile ottenere formule matematiche rappresentanti il tempo ciclo della cella. Un problema comune di analisi di cella robotizzata consiste nella ricerca della sequenza di movimenti ottimale. Un problema di questo tipo può risultare NP-difficile (*nondeterministic polynomial-time hard*) ed è necessario utilizzare metodi di ricerca non convenzionali, principalmente euristici, per determinare la soluzione desiderata. Un possibile metodo consiste nell'utilizzare algoritmi genetici [23, 24] che ricercano la soluzione traendo ispirazione da aspetti naturalistici. Essi si basano sui principi di evoluzione di una specie e i criteri di selezione naturale considerando i tentativi di soluzione come individui. Un altro esempio di metodo di soluzione di problemi NP-Hard in ambito robotizzato è il metodo di Taguchi [25] basato su fondamenti statistici.

Nel caso in esame tuttavia l'obiettivo non è identificare la sequenza di operazioni ottimale, poichè questa è definita automaticamente dati i parametri della cella. Si ricorda infatti che la gestione della cella in esame avviene dinamicamente, senza una sequenza stabilita a priori ma basata unicamente sulla disponibilità delle postazioni. Definiti i parametri che caratterizzano la cella tale sequenza è automaticamente imposta. Lo scopo della presente analisi è identificare tale sequenza e determinarne il tempo ciclo risultante al variare dei parametri della cella stessa. Non si tratta di determinare un valore minimo assoluto quanto piuttosto ottenere rappresentazioni dell'andamento del tempo ciclo in funzione dei parametri caratteristici della cella. Di conseguenza si desidera impostare un modello teorico e implementarlo con un algoritmo in Matlab.

Esempio di studio analitico

La disponibilità di una formula finale del tempo ciclo consente inoltre il confronto fra differenti alternative di layout possibili. Utilizzando la medesima impostazione del presente lavoro, Drobouchevitch et Al. [7] hanno analizzato la produttività di una cella con manipolatore a doppia pinza confrontando le tempistiche con quelle fornite da una configurazione analoga a pinza singola aggiungendo in quest'ultima un buffer in ingresso al macchinario di lavorazione. Le soluzioni a pinza multipla,

prevalentemente doppia, con pinze indipendenti solidali alla flangia robot consentono di saturare maggiormente i macchinari di lavorazione in quanto in seguito al prelievo di un pezzo lavorato è possibile depositare il pezzo grezzo contenuto nella seconda pinza. Nello studio in questione in entrambi i casi il macchinario di lavorazione dispone di un buffer in uscita, ovvero è in grado di movimentare a lato il pezzo lavorato per consentire il deposito di un nuovo componente grezzo senza necessariamente prelevare prima quello già lavorato. Nella seconda configurazione è presente anche una locazione di buffering in ingresso al macchinario di lavorazione, cioè un componente non lavorato può essere depositato a lato del macchinario anche durante una lavorazione in corso. Un apposito sistema si occupa del caricamento del macchinario di lavorazione. I risultati dello studio, effettuato analiticamente, affermano la totale equivalenza dei due casi in termini di produttività. Tali risultati consentono di orientare la valutazione del layout da adottare in base a scelte economiche dettate dal costo della pinza doppia e del sistema di autoalimentazione del macchinario di lavorazione.

Capitolo 3

Modello analitico

Questo capitolo è dedicato allo studio matematico della cella in esame. Lo scopo è arrivare ad esprimere il tempo ciclo di produttività della cella in funzione dei parametri caratteristici di quest'ultima. Per comodità di analisi e per poter anche verificare l'effettivo vantaggio della presenza del buffer, si parte dall'analisi della cella più semplice per poi passare a quella più complicata con presente anche il buffer.

3.1 Notazione

Per consentire una più facile comprensione della trattazione si riportano i simboli utilizzati nel capitolo e il loro significato:

- δ = tempo di spostamento del robot fra stazioni adiacenti;
- η = quota di accelerazione e decelerazione relativa al moto fra stazioni non adiacenti;
- ϵ = tempo di prelievo o rilascio del pezzo;
- τ_F = tempo di rinnovo del feeder;
- τ_L = tempo di lavorazione del componente;
- τ_S = tempo di rinnovo della postazione di scarico;

- τ_H = tempo richiesto per raggiungere la posizione di home dalla postazione di lavorazione e viceversa;
- N_F = numero di componenti disponibili nel feeder ad ogni rinnovo, supposto costante se non diversamente specificato;
- N_B = numero di slot presenti nella postazione di buffer.

I tempi in esame si considerano deterministici per semplicità.

3.2 Cella senza buffer

Per questo caso di cella semplificata si presume, come già accennato nel capitolo precedente, che le postazioni di input e output si trovino ad estremi opposti. Si considera inoltre la posizione del macchinario analoga al caso con buffer presente, ovvero nelle vicinanze della postazione di scarico.

3.2.1 Tempo di scarico trascurabile

Nel caso in cui non venga utilizzata la stazione di bufferizzazione, il movimento del robot prevede, in ordine, la presa del pezzo dal feeder, il posizionamento sul macchinario, l'attesa del termine della lavorazione in home, la presa e il trasporto alla stazione di scarico. Chiude il ciclo il ritorno al feeder per prendere un altro componente, se questo è disponibile, o il movimento in home per attendere il segnale di disponibilità della postazione di input. Si consideri innanzitutto il caso in cui il tempo richiesto per il rinnovo della stazione di scarico sia trascurabile e pertanto non rientri nel calcolo del tempo ciclo. In altre parole, quando un pezzo ha terminato la lavorazione e viene afferrato per essere riposto nella stazione di scarico, questa è sempre disponibile ad accogliere il componente lavorato. In tale situazione identifichiamo con β il tempo, costante per ogni singolo pezzo, necessario perchè esso venga portato in lavorazione e al termine di quest'ultima posizionato nel punto adibito allo scarico. Per il calcolo di β si considera che il robot abbia appena afferrato un componente dalla postazione di input e sia pronto ad allontanarsi da essa.

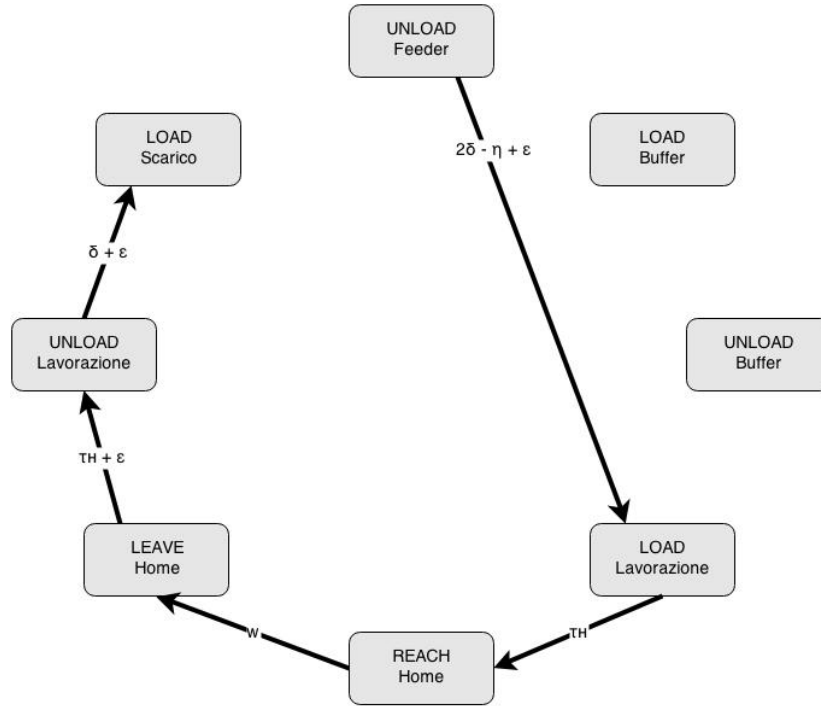


Figura 3.1: Rappresentazione nel diagramma degli stati della sequenza identificata dal parametro β . L'attesa w è pari a $\tau_L - \tau_H$.

$$\beta = (2\delta - \eta + \epsilon) + \tau_H + (\tau_L - \tau_H) + (\tau_H + \epsilon) + (\delta + \epsilon) = 3\delta - \eta + 3\epsilon + \tau_L + \tau_H \quad (3.1)$$

Per permettere una più facile comprensione nella trattazione si avrà cura di raggruppare fra parentesi tonde i distinti movimenti quando possibile. In questo caso, partendo dalla postazione di input, sono richiesti $2\delta - \eta$ per raggiungere la postazione di lavorazione a cui viene aggiunto ϵ per considerare il tempo di rilascio del pezzo. Segue l'attesa di un tempo pari a quello di lavorazione τ_L muovendosi nel frattempo verso la posizione di home. Si assume per semplicità che τ_L sia superiore al tempo richiesto per effettuare tale movimento, ovvero il manipolatore giunge alla posizione di home prima del termine dell'operazione di lavorazione. Grazie a tale assunzione $w = \tau_L - \tau_H > 0$. Segue il movimento dalla posizione di home verso la stazione di lavorazione τ_H e si procede con l'afferraggio del pezzo impiegando un tempo ϵ . Il robot conclude muovendosi verso la stazione di scarico in un tempo $\delta + \epsilon$.

Supponendo ora che il feeder sia disponibile, cioè contenga almeno un pezzo, finita l'azione di scarico, il robot si muoverà immediatamente verso di esso e caricherà un nuovo pezzo non lavorato. Quest'ultima azione avviene in un tempo di $3\delta - 2\eta + \epsilon$. Si introduce ora α che sta a indicare il tempo di un ciclo completo in questa situazione. Si ha quindi:

$$\alpha = \beta + (3\delta - 2\eta + \epsilon) = 6\delta - 3\eta + 4\epsilon + \tau_L + \tau_H \quad (3.2)$$

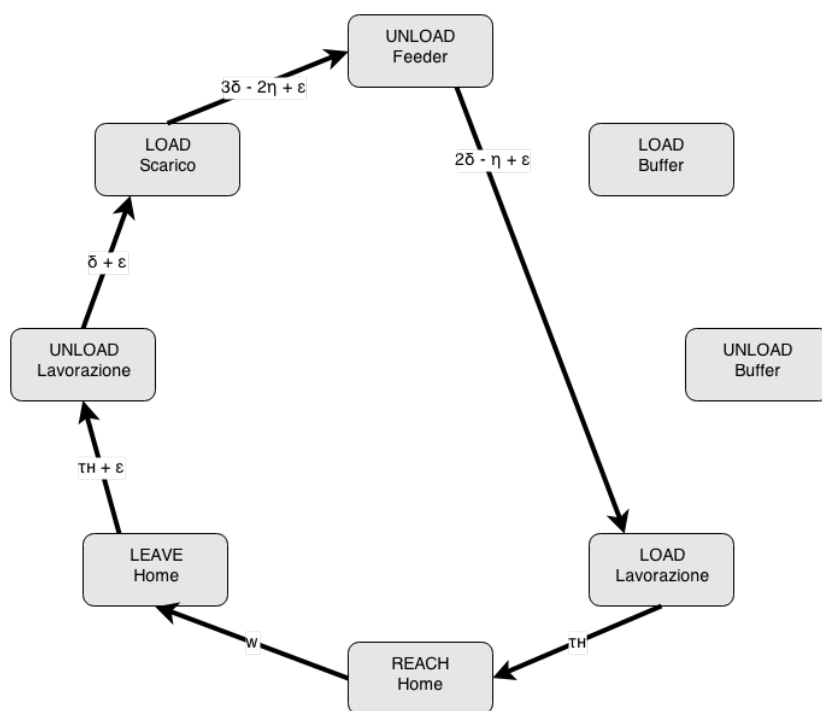


Figura 3.2: Rappresentazione nel diagramma degli stati della sequenza identificata dal parametro α , partendo da UNLOAD Feeder. L'attesa w è pari a $\tau_L - \tau_H$.

Il tempo ciclo viene calcolato dividendo il tempo che intercorre fra due medesime situazioni nella cella per il numero di pezzi passati in essa. Nel caso in esame si considera il tempo intercorrente tra due rinnovi feeder e si divide quindi per il numero di pezzi forniti in ogni vassoio, N_F . L'idea è quella di moltiplicare il valore α per il numero di pezzi presenti inizialmente in ciascun vassoio prestando però attenzione all'eventuale contributo di un cambio feeder che non avviene in tempo mascherato. Infatti, poichè il robot si posiziona in home nel caso in cui

debba attendere la disponibilità di pezzi in ingresso, si può affermare che il feeder viene rinnovato completamente in tempo mascherato solo se:

$$\tau_F \leq \beta = (2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) \quad (3.3)$$

In questa situazione, il tempo ciclo della cella di lavoro corrisponde ad α e non vi sarebbe alcun beneficio derivante dall'utilizzo di un buffer, in quanto un nuovo pezzo risulta sempre disponibile in seguito al deposito nel punto di scarico del pezzo precedentemente elaborato.

$$T_C = \frac{\alpha N_F}{N_F} = \alpha \quad (3.4)$$

Se parte del cambio feeder avviene invece in tempo non mascherato, ovvero $\tau_F > \beta$, il tempo ciclo finale sarà inevitabilmente superiore ad α . In altri termini, quando il robot posiziona allo scarico l'ultimo pezzo di un determinato vassoio, è poi costretto a raggiungere la home e attendere il segnale di disponibilità in quanto l'arrivo di nuovi componenti richiede ulteriore tempo. Per calcolare il tempo ciclo si avrebbe quindi il robot che parte dalla posizione di home. La prima azione effettuata è lo spostamento verso la postazione di input unita all'afferraggio del primo pezzo, questo in un tempo di $(2\delta - \eta + \epsilon)$. Seguono $N_F - 1$ cicli di lavorazione di durata α . Successivamente allo scarico dell'ultimo pezzo, il robot si posiziona in home impiegando un tempo δ e attende il segnale in questione. Per concludere si deve calcolare l'attesa nella posizione di home. Questa è nulla se il segnale viene ricevuto durante il movimento verso di essa, altrimenti è pari a $(\tau_F - \beta - \delta)$ che corrisponde al tempo rimanente per completare il ricambio della postazione di ingresso pezzi. Riassumendo quanto appena spiegato:

$$N_F T_C = (2\delta - \eta + \epsilon) + (N_F - 1)\alpha + (\beta + \delta) + w \quad (3.5)$$

$$w = \max(\tau_F - \beta - \delta, 0) \quad (3.6)$$

Rielaborando le equazioni si giunge alla formula del tempo ciclo nel caso di tempo di scarico trascurabile e rinnovo feeder in tempo non completamente mascherato.

$$T_C = \frac{(2\delta - \eta + \epsilon) + (N_F - 1)\alpha + \max(\tau_F, \beta + \delta)}{N_F} \quad (3.7)$$

3.2.2 Influenza del tempo di scarico

Si consideri ora il caso in cui il tempo richiesto dal sistema di scarico per diventare nuovamente disponibile ad accogliere un pezzo lavorato sia tale da influenzare il tempo ciclo. Questo significa che terminata la lavorazione il robot non ha ancora ricevuto il segnale di disponibilità della stazione di scarico e si posiziona in home in attesa di quest'ultimo. All'arrivo del segnale si muove verso la postazione di lavorazione per prelevare il componente lavorato e lo colloca allo scarico. Questa situazione avviene se:

$$\begin{aligned}\tau_S &> (3\delta - 2\eta + \epsilon) + (2\delta - \eta + \epsilon) + \tau_L \\ &> 5\delta - 3\eta + 2\epsilon + \tau_L\end{aligned}\tag{3.8}$$

Se la disequazione non è soddisfatta si ricade nel caso analizzato in precedenza in cui si è considerato τ_S irrilevante per la produttività della cella. In caso opposto il tempo ciclo è influenzato da τ_S . Nuovamente è necessario considerare se il cambio feeder avviene interamente in tempo mascherato o meno, tuttavia la condizione che distingue i due casi è differente da quella analizzata precedentemente. Si indichi con Δ_S il tempo di attesa nella posizione di home richiesto prima di ricevere la disponibilità della stazione di scarico: $\Delta_S = \tau_S - [(3\delta - 2\eta + \epsilon) + (2\delta - \eta + \epsilon) + \tau_L]$. La condizione cercata risulta:

$$\tau_F \leq (2\delta - \eta + \epsilon) + \tau_L + \Delta_S + (\tau_H + \epsilon) + (\delta + \epsilon)\tag{3.9}$$

Il secondo membro della disequazione 3.9 rappresenta il trasporto dell'ultimo pezzo dal feeder verso la lavorazione, l'operazione di lavorazione e successivo afferraggio del pezzo lavorato, l'attesa in home e lo scarico del pezzo. Sostituendo l'espressione di Δ_S si può semplificare la disequazione 3.9 come segue.

$$\tau_F \leq \tau_S - 2\delta + 2\eta + \epsilon + \tau_H\tag{3.10}$$

Il ricambio feeder avviene in tempo mascherato e il tempo ciclo è pari a:

$$\tau_S + (\tau_H + \epsilon) + (\delta + \epsilon) = \phi\tag{3.11}$$

Tale valore risulta utile nella trattazione dei cicli con buffering e verrà indicato con ϕ .

Nel caso in cui l'espressione non sia soddisfatta il tempo totale fra l'arrivo di due vassoi si può pensare come una successione di $N_F - 1$ cicli di durata ϕ a cui segue un ciclo comprendente l'attesa del ricambio feeder. Quest'ultimo si può analizzare considerando i movimenti del manipolatore a partire dalla postazione di scarico: con un movimento δ viene raggiunta la posizione di home, si attende il segnale del feeder per un tempo Δ_F , viene prelevato uno dei nuovi pezzi giunti nella postazione di input per portarlo in lavorazione e in seguito il manipolatore torna alla stazione di scarico per riporre il pezzo lavorato. Nel caso in cui il tempo risultante sia inferiore a ϕ anche questo ciclo avrà tale durata.

$$\begin{aligned}\Delta_F &= \max(0, \tau_F - [(2\delta - \eta + \epsilon) + \tau_L + \Delta_S + (\tau_H + \epsilon) + (\delta + \epsilon) + (\delta)]) \\ &= \max(0, \tau_F - 4\delta + \eta - 3\epsilon - \tau_L - \Delta_S - \tau_H)\end{aligned}\quad (3.12)$$

$$\begin{aligned}T_C N_F &= (N_F - 1)(\phi) + [\max(\phi, \delta + \Delta_F + (2\delta - \eta + \epsilon) + \\ &\quad + (2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)) \\ T_C &= \frac{(N_F - 1)(\phi) + \max(\phi, 6\delta + \Delta_F - 2\eta + 4\epsilon + \tau_L + \tau_H)}{N_F}\end{aligned}\quad (3.13)$$

3.3 Cella con buffer

Di maggiore complessità è il modello sviluppato per simulare una cella funzionante con buffer interno. L'identificazione di un ciclo che si ripete in un periodo $N_F T_C$ come è stato effettuato nel precedente caso esaminato non è un'operazione immediata poichè questo ciclo potrebbe interessare un numero di rinnovi feeder superiore all'unità. Per semplicità si suppone che in seguito ad un eventuale tempo iniziale richiesto per saturare il buffer la cella lavori ogni gruppo di pezzi effettuando le medesime operazioni. Si assume cioè che il periodo fra due medesime situazioni nella cella corrisponde all'intervallo fra l'arrivo di due feeder successivi. Si analizza la situazione a regime: fra un cambio feeder ed il seguente il buffer si ritrova con lo stesso numero di pezzi. Non vengono prelevati più componenti di quelli immessi nello stesso ciclo e, se nella cella entrano dal feeder N_F pezzi, ne vengono scaricati lo stesso numero.

La priorità di svuotamento considerata in sezione 3.4 è attribuita alla stazione con il feeder: nel caso in cui il robot si trovi ad aver scaricato un componente

lavorato e debba scegliere se portare in lavorazione un pezzo prelevandolo dalla postazione di input o dal buffer, la scelta ricade sulla postazione con il feeder. Il caso duale viene considerato in sezione 3.5. La priorità di svuotamento generale è comunque sempre attribuita alla stazione di lavorazione: se ci dovesse essere un pezzo lavorato pronto ad essere riposto allo scarico tale operazione ha la precedenza su tutte quelle possibili.

3.4 Priorità feeder

3.4.1 Componenti depositati nel buffer

Si consideri un caso generico di lavorazione: il pezzo viene prelevato nella stazione di input e posizionato dove richiesto per poter effettuare la lavorazione. Durante l'attesa del termine lavorazione ovvero finchè il robot non riceve un segnale di ready da parte del macchinario preposto a tale operazione esso preleva ulteriori pezzi dal feeder posizionandoli nel buffer. La bufferizzazione durante l'attesa del termine lavorazione è un'operazione che non avviene completamente in tempo mascherato poiché procede fino alla ricezione del rispettivo segnale. In condizioni ideali il manipolatore dovrebbe trovarsi già in prossimità della stazione di lavorazione nel momento di ricezione del segnale, pronto a prelevare il componente. Se però il segnale di fine lavorazione viene ricevuto durante il tragitto verso la presa di un nuovo pezzo dal feeder il manipolatore è costretto a terminare l'operazione, cioè prelevare il pezzo e depositarlo nel buffer, nonostante il componente lavorato sia disponibile ad essere scaricato.

Il numero di pezzi che teoricamente è possibile depositare nella stazione di buffer durante ogni singola attesa del termine lavorazione viene indicato con N_{BL} e può essere calcolato considerando la successione di operazioni che il robot è in grado di effettuare durante il tempo di lavorazione.

$$\tau_L \leq (2\delta - \eta + \epsilon) + (\delta + \epsilon) + (N_{BL} - 1)[(\delta + \epsilon) + (\delta + \epsilon)] \quad (3.14)$$

Il termine $[(\delta + \epsilon) + (\delta + \epsilon)]$ corrisponde al tempo richiesto per bufferizzare un singolo pezzo e considera il viaggio dal buffer al feeder per la presa di un pezzo sommato al medesimo viaggio di ritorno per il rilascio. Come si può notare in

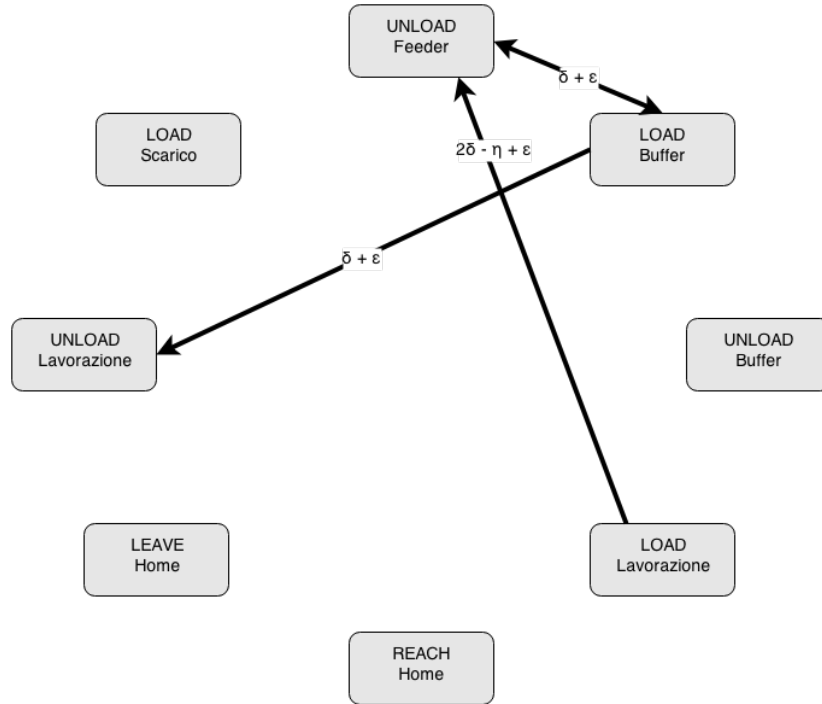


Figura 3.3: Rappresentazione nel diagramma degli stati della sequenza identificata dall'equazione 3.16.

questo calcolo non si considera il tempo richiesto per ritornare alla stazione di lavorazione ma solo quello richiesto dalle operazioni compiute prima della ricezione del segnale di fine lavorazione. Tale affermazione è in accordo con la gestione dinamica della cella: le operazioni sono stabilite in base ai segnali ricevuti. Il valore N_{BL} è ottenibile cercando il più piccolo intero che soddisfi 3.14. Poichè il pezzo in corso di buffering deve essere depositato prima di poter procedere verso la stazione di lavorazione abbiamo:

$$N_{BL} = \left\lceil \frac{\tau_L - \delta + \eta}{2\delta + 2\epsilon} \right\rceil \quad (3.15)$$

Risulta utile in seguito la definizione di un tempo di lavorazione fittizio τ_{L_A} che rappresenta il tempo apparente di lavorazione nel caso in cui venga effettuato il buffering. Come precedentemente esposto l'operazione di riempimento del buffer può non avvenire in tempo completamente mascherato: il parametro apparente di lavorazione rappresenta il tempo intercorso fra il rilascio di un pezzo sulla stazione di lavorazione e l'arrivo del manipolatore sulla postazione di lavorazione per

prelevare quello stesso pezzo nel caso in cui, nel frattempo, avvenga l'operazione di buffering. Una rappresentazione di tale situazione è riportata in figura 3.3.

$$\begin{aligned}\tau_{LA} &:= (2\delta - \eta + \epsilon) + (\delta + \epsilon) + (N_{BL} - 1)[(\delta + \epsilon) + (\delta + \epsilon)] + (\delta) \\ &= 2\delta - \eta + N_{BL}(2\delta + 2\epsilon)\end{aligned}\tag{3.16}$$

Per comodità in seguito verrà utilizzato il termine *ciclo-lavorazione* per indicare l'intera operazione di prelievo da feeder, rilascio in lavorazione e attesa con eventuale buffering intermedio. Con il termine *ciclo-feeder* si indica invece la successione di operazioni fra l'arrivo di un feeder e quello successivo. Un *ciclo-feeder* risulta composto da una serie di *cicli-lavorazione* a cui seguono i cicli di svuotamento del buffer durante l'attesa di rinnovo della stazione di input.

3.4.2 Svuotamento del feeder

Per procedere con lo sviluppo del modello è necessario definire dei parametri rappresentativi dello stato in cui viene raggiunta la condizione di svuotamento del feeder. Si vuole avere a disposizione un parametro che indichi se l'ultimo pezzo del feeder venga prelevato nel movimento verso la stazione di lavorazione o nel corso dell'operazione di buffering. In quest'ultimo caso è necessario conoscere anche a che punto è stato interrotto il processo di buffering. La bufferizzazione durante l'ultimo ciclo-lavorazione avviene completamente in tempo mascherato se i pezzi disponibili nel feeder sono in numero minore di quelli teoricamente posizionabili nel buffer N_{BL} . Il tempo di attesa del pezzo sulla stazione di lavorazione sarà in questo caso $\tau_L + \tau_H$ anzichè τ_{LA} . Si suppone che effettuando una bufferizzazione di $N_{BL} - 1$ pezzi l'intero ciclo di buffering durante l'ultima attesa lavorazione avvenga in tempo mascherato, ovvero non si considera che il segnale di fine lavorazione non possa arrivare durante il movimento dal buffer alla posizione di home. Con una tale ipotesi si sta trascurando un possibile contributo al tempo ciclo del viaggio di ritorno verso la lavorazione nel caso in cui il segnale di fine lavorazione dovesse arrivare durante il movimento stesso.

I parametri introdotti per raggiungere lo scopo illustrato sono indicati con γ e λ .

$$\gamma := \lceil \frac{N_F}{(N_{BL} + 1)} \rceil \quad (3.17)$$

γ rappresenta il numero teorico di *cicli-lavorazione* con buffering intermedio pari a N_{BL} nel caso in cui non ci sia un limite alla dimensione del buffer. Risulta immediata la definizione di un duale parametro λ che rappresenta il numero di pezzi mancanti perché γ sia una variabile intera senza bisogno di approssimazione, ovvero il numero di componenti non immessi nel buffer, durante l'ultimo dei cicli sopra citati, a causa dello svuotamento del feeder.

$$\lambda := \gamma(N_{BL} + 1) - N_F \quad (3.18)$$

Per come è stato definito λ può assumere valori compresi tra 0 e N_{BL} . Si faccia riferimento alla figura 3.4 per dare una valenza fisica ai termini definiti. Si supponga di disporre di un pallet in ingresso costituito da 12 pezzi come nella figura a sinistra e sia il valore N_{BL} risultante dall'equazione 3.15 pari a 2.

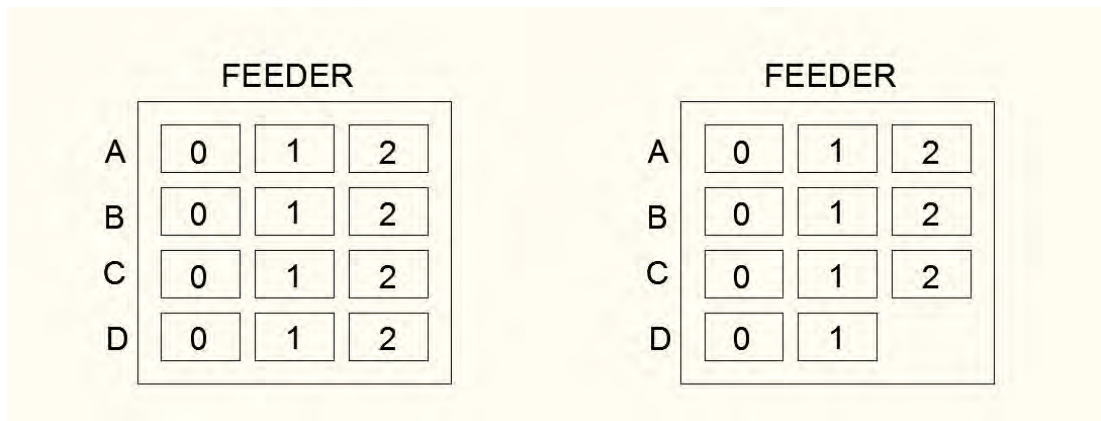


Figura 3.4: Esempio per fornire una spiegazione pratica dei parametri λ e γ .

Posso suddividere i pezzi disponibili sul feeder in γ gruppi (A-B-C-D, $\gamma = 4$) costituiti da $N_{BL} + 1$ pezzi (0,1,2, $N_{BL} = 2$). Ciascun gruppo corrisponde ad un *ciclo-lavorazione* come segue:

1. Prelievo pezzo 0 e posizionamento su lavorazione;
2. Bufferizzazione pezzi 1, ..., $N_{BL}(1 - 2)$;

3. Afferraggio componente lavorato per scarico;
4. Ritorno al feeder.

In questo esempio il parametro λ assume un valore nullo, ovvero il feeder viene svuotato con un numero esatto γ di operazioni di bufferizzazione massima teorica pari a N_{BL} . Se il numero di pezzi sul feeder è pari a 11 come nella figura a destra (mantenendo il medesimo valore di N_{BL} dall'equazione 3.15) per $(\gamma - 1)$ volte vengono effettuate le operazioni descritte mentre le operazioni per l'ultimo gruppo di pezzi sono:

1. Prelievo pezzo 0 e posizionamento su lavorazione;
2. Bufferizzazione pezzi $1, \dots, N_{BL} - \lambda$;
3. Afferraggio componente lavorato per scarico;
4. Ritorno al feeder.

Il parametro λ in questo caso assume il valore unitario. Si può dedurre che, se $\lambda = 0$, i cicli-lavorazione presentano tutti un'attesa del pezzo sulla lavorazione pari a τ_{LA} e sono in numero pari a γ . Nel caso contrario i cicli di lavorazione totali sono in numero pari a γ ma l'ultimo di questi è caratterizzato da un'attesa del pezzo sulla lavorazione pari a $\tau_L + \tau_H$.

3.4.3 Tempo limite di rinnovo feeder

Così come il buffer viene riempito nella sola attesa del termine lavorazione, allo stesso modo esso viene sfruttato durante la sola attesa di cambio feeder. Questa considerazione è in accordo con l'ipotesi di assegnare la priorità di svuotamento al feeder. Prima di calcolare il valore del numero di pezzi che si riescono a prelevare dal buffer ad ogni cambio feeder è necessario definire un tempo limite di cambio feeder: se τ_F è minore di questo valore il buffer non viene utilizzato in quanto successivamente allo scarico di un qualsiasi pezzo è sempre disponibile un componente nella postazione di ingresso. Questo tempo limite è strettamente legato al parametro λ poiché a seconda del momento in cui avviene il prelievo dell'ultimo

pezzo dal feeder si assiste ad un tempo di raggiungimento della stazione di scarico variabile. Si calcolerà ora τ_F per i vari casi.

- se $\lambda = N_{BL}$ l'ultimo pezzo prelevato dal feeder viene portato in lavorazione e non si ha modo di bufferizzare durante l'attesa: il tempo limite è calcolato considerando il movimento dal feeder verso la lavorazione, l'attesa del termine di quest'ultima e il movimento verso lo scarico.

$$\begin{aligned}\tau_{Flim} &= (2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) \\ &= 3\delta - \eta + 3\epsilon + \tau_L + \tau_H\end{aligned}\quad (3.19)$$

- se $\lambda = 0$ la bufferizzazione dell'ultimo ciclo viene completata interamente, con la ricezione del segnale di fine lavorazione mentre l'ultimo pezzo del feeder viene portato al buffer. Il tempo limite è calcolato considerando il movimento dal feeder al buffer, da qui alla lavorazione, e da quest'ultima allo scarico.

$$\begin{aligned}\tau_{Flim} &= (\delta + \epsilon) + (\delta + \epsilon) + (\delta + \epsilon) \\ &= 3\delta + 3\epsilon\end{aligned}\quad (3.20)$$

- se $\lambda \neq 0$ e $\lambda \neq N_{BL}$ è necessario determinare il tempo di lavorazione rimasto nel momento in cui il robot si posiziona in home in attesa del termine dell'operazione.

$$w = \tau_L - [(2\delta - \eta + \epsilon) + (\delta + \epsilon) + ((N_{BL} - \lambda + 1)(\delta + \epsilon) + (\delta + \epsilon)) + (\delta)] \quad (3.21)$$

$$\begin{aligned}\tau_{Flim} &= (\delta + \epsilon) + (\delta + w + \tau_H + \epsilon) + (\delta + \epsilon) \\ &= 3\delta + 3\epsilon + w + \tau_H\end{aligned}\quad (3.22)$$

3.4.4 Componenti prelevati dal buffer

Noto il tempo limite inferiore di cambio feeder oltre il quale conviene l'utilizzo del buffer, se $\tau_{Flim} - \tau_F < 0$ è possibile determinare il numero di componenti prelevabili dal buffer, N_{out} . Ciascun pezzo è prelevato, depositato sulla postazione di lavorazione e in seguito scaricato. La procedura è simile a quella seguita per la

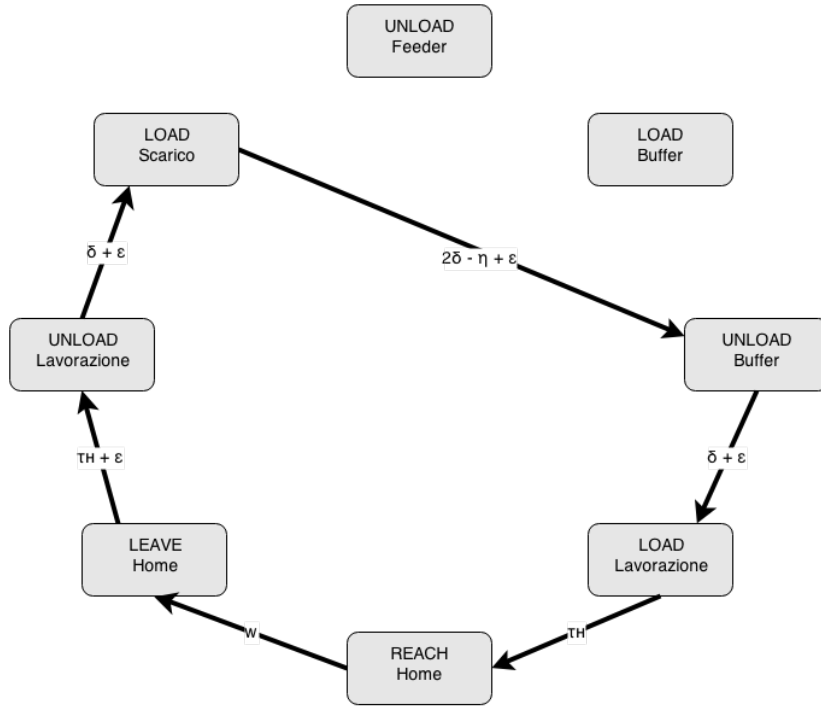


Figura 3.5: Rappresentazione delle operazioni di prelievo dal buffer nel diagramma degli stati.

determinazione di N_{BL} , compreso l'arrotondamento all'intero superiore. In figura 3.5 è rappresentata tale situazione nel diagramma degli stati.

$$\tau_F \leq \tau_{Flim} + N_{out}[(2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)] \quad (3.23)$$

$$N_{out} = \left\lceil \frac{\tau_F - \tau_{Flim}}{4\delta + 4\epsilon - \eta + \tau_L + \tau_H} \right\rceil \quad (3.24)$$

3.4.5 Tempo ciclo

Il tempo ciclo è calcolabile sommando tutte le operazioni effettuate dal robot tra due successivi rinnovi del feeder dividendo il risultato per il numero di componenti presenti in esso. Infatti, con la supposizione in sezione 3.3, il numero di pezzi lavorati e scaricati dalla cella nel tempo ciclo è esattamente N_F . Le operazioni compiute dal robot si possono per semplicità suddividere in tre macro-operazioni. Ciò consente di utilizzare una sola formula per il calcolo del tempo ciclo indipendentemente dai parametri della cella.

1. La prima macro-operazione consiste nel prelievo di un pezzo dal feeder, deposito in lavorazione, attesa di un tempo τ_{LA} , scarico e ritorno al feeder. In questa successione di operazioni il robot inserisce pezzi nel buffer ma la loro entità non rientra nella formula grazie all'utilizzo di un tempo di lavorazione fittizio. Il numero di volte in cui questa macro-operazione viene effettuata è indicato con γ_A .

$$\begin{aligned} T_{CA} &= (2\delta - \eta + \epsilon) + (\tau_{LA} + \epsilon) + (\delta + \epsilon) + (3\delta - 2\eta + \epsilon) \\ &= 6\delta - 3\eta + 4\epsilon + \tau_{LA} \end{aligned} \quad (3.25)$$

2. La seconda macro-operazione è simile alla prima con la differenza che non avviene bufferizzazione o, se questa è presente, è effettuata in tempo totalmente mascherato, cioè riguardante un numero di componenti inferiore a N_{BL} . Nelle situazioni analizzate fino ad ora questo corrisponde all'ultimo *ciclo-lavorazione* se $\lambda \neq N_{BL}$. La molteplicità con cui è compiuta tale macro-operazione viene in seguito indicata con il parametro γ_R (si è usata R per *real*, in contrapposizione alla A di *apparente* del caso precedente).

$$\begin{aligned} T_{CR} &= (2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) + (3\delta - 2\eta + \epsilon) \\ &= 6\delta - 3\eta + 4\epsilon + \tau_L + \tau_H \end{aligned} \quad (3.26)$$

Nelle condizioni fin qui analizzate risulta immediata l'attribuzione dei parametri γ_A e γ_R :

- se $\lambda = 0$: $\gamma_A = \gamma$ e $\gamma_R = 0$
- se $\lambda \neq 0$: $\gamma_A = \gamma - 1$ e $\gamma_R = 1$

Risulta evidente come la somma fra γ_A e γ_R debba coincidere con il parametro γ calcolato precedentemente.

3. L'ultima parte del ciclo di lavoro corrisponde alle operazioni effettuate durante il cambio del feeder. In accordo con quanto illustrato, questo ciclo corrisponde alla presa di N_{out} pezzi dal buffer, che vengono portati in lavorazione e, in seguito ad un'attesa τ_L , riposti allo scarico.

$$T_{CO} = N_{out}[(\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) + (2\delta - \eta + \epsilon)] \quad (3.27)$$

É possibile ora definire la formula generale del tempo ciclo:

$$T_C = \frac{\gamma_A T_{CA} + \gamma_R T_{CR} + T_{CO}}{N_F} \quad (3.28)$$

3.4.6 Flusso nullo nel buffer

Come precisato nelle ipotesi del modello, si desidera analizzare la situazione a regime: il bilancio dei pezzi in entrata e quelli in uscita nel buffer ad ogni cambio feeder deve essere nullo. Questo non si verifica in due casi: o i pezzi prelevati nel buffer sono in numero inferiore a quelli depositati complessivamente nello stesso ciclo di lavoro o i pezzi riposti sono minori di quelli teoricamente prelevabili in fase di ricambio feeder.

Per il primo caso, dopo un numero variabile di cicli iniziali, a livello pratico si raggiunge la condizione di saturazione del buffer, in quanto questo non è di dimensioni illimitate. Da quel momento in poi inizia effettivamente la fase a regime considerata dal modello. Si trascura il tempo che intercorre fra l'avvio della cella e la saturazione del buffer in quanto si suppone che il tempo ciclo venga calcolato su un periodo di tempo tale da rendere trascurabile il transitorio iniziale.

Per il secondo caso, invece, parte del tempo di attesa ricambio feeder avviene con il robot fermo in quanto non sono disponibili ulteriori pezzi nel buffer per essere lavorati: viene modificato il valore di T_{CO} introdotto nel capitolo precedente.

In generale si vuole imporre che il numero di pezzi totali inseriti nel buffer complessivamente sia uguale a quello prelevato in fase di ricambio feeder. Il numero di pezzi inseriti nel buffer fra l'arrivo di due feeder successivi viene indicato con N_{in} ed è pari a:

$$N_{in} := (\gamma N_{BL}) - \lambda \quad (3.29)$$

Sia Ω il numero di buffer effettivamente usati in un ciclo a regime, $\Omega = \min(N_{in}, N_{out})$, si vuole analizzare come determinare il tempo ciclo nei casi in cui non ci sia il flusso nullo, in altri termini se $N_{in} \neq \Omega$ o $N_{out} \neq \Omega$. E' possibile inoltre che sia richiesto un determinato limite del buffer derivante da limitazioni interne alla cella. Indicando con N_B il numero di pezzi massimo che il buffer è in grado di

accomodare è possibile riformulare l'espressione di Ω come segue:

$$\Omega = \min(N_{in}, N_{out}, N_B) \quad (3.30)$$

1. $N_{in} \neq \Omega$

Non vi è modo di effettuare la bufferizzazione teorica nella sua interezza. Aumentano i cicli di lavorazione senza inserimento di pezzi nel buffer. Si può determinare il numero di pezzi che eccedono la bufferizzazione imposta indicando tale valore con e (*exceed*).

$$e := (\gamma N_{BL} - \lambda) - \Omega \quad (3.31)$$

La somma dei due parametri γ_A e γ_R in questo caso non equivale a γ come in precedenza ma a $\gamma + e$. Questo accade perchè i pezzi che non è possibile depositare nel buffer a causa del limite raggiunto o dell'impossibilità di prelevarli in seguito costituiscono ulteriori cicli di macro-operazioni di tipo 2. Il numero di macro-operazioni di tipo 1 che il robot è in grado di realizzare prima di saturare il buffer, ovvero il valore corretto di γ_A , corrisponde a:

$$\gamma_A = \lfloor \frac{\Omega}{N_{BL}} \rfloor \quad (3.32)$$

Di conseguenza:

$$\gamma_R = \gamma + e - \gamma_A \quad (3.33)$$

Sorge un problema non trascurabile: modificando il numero di pezzi inseriti nel buffer si modifica inevitabilmente anche il parametro λ , rappresentativo della situazione in cui avviene lo svuotamento del feeder. Di conseguenza varia il tempo limite inferiore di ricambio feeder utile alla bufferizzazione e con questo il numero di pezzi utilmente prelevati dal buffer: può di conseguenza variare anche Ω . Nell'algoritmo implementato in Matlab sono state utilizzate parti ricorsive: la funzione richiama sè stessa variando uno dei parametri in ingresso fino al raggiungimento della convergenza. L'implementazione in Matlab del modello viene discussa in sezione 4.1.

2. $N_{out} \neq \Omega$

E' necessario modificare il valore del parametro T_{CO} . Esso in tale situazione è indipendente dal buffer. Il suo valore è fornito dal tempo richiesto per muoversi dallo scarico alla home e attendere il segnale di feeder disponibile. Si considera inoltre che il moto verso il feeder avviene ora dalla home e non dalla postazione di scarico.

$$\begin{aligned} T_{CO} &= \max(\tau_F - \tau_{Flim}, \delta) + (2\delta - \eta + \epsilon) - (3\delta - 2\eta + \epsilon) \\ &= \max(\tau_F - \tau_{Flim} + \eta - \delta, \eta) \end{aligned} \quad (3.34)$$

3.4.7 Post-buffering

Nel caso in cui $N_{out} = \Omega$ può avvenire una bufferizzazione non ancora considerata e in seguito indicata con il termine *post-buffering*. Il segnale di disponibilità del feeder potrebbe giungere durante la lavorazione dell'ultimo pezzo N_{out} prelevato dal buffer. Di conseguenza il robot, in accordo con la risposta ai segnali, preleva i componenti dal feeder e li ripone nel buffer finchè non riceve la disponibilità del pezzo lavorato sulla stazione di lavorazione. Il numero di pezzi prelevati dal feeder e riposti nel buffer in questa fase verrà indicato con N_P . Terminata l'operazione prosegue il ciclo come descritto in precedenza.

É necessario considerare che il numero dei pezzi disponibili nel feeder è ora minore ($N_F - N_P$) perchè parte di essi sono già stati riposti nel buffer al termine del ciclo precedente. Di conseguenza varia λ il quale determina τ_{Flim} che influenza a sua volta il numero di pezzi N_{out} prelevabili dal buffer durante il ciclo. A seconda della relazione fra N_{out} e N_{BL} avviene o meno la post-bufferizzazione e può variare N_P rispetto al valore inizialmente calcolato. Anche in questo caso è richiesta la ricorsività dell'algoritmo per raggiungere la convergenza. Nella nuova situazione

$$\lambda = \gamma(N_{BL} + 1) - (N_F - N_P) \quad (3.35)$$

$$\gamma = \lceil \left(\frac{N_F - N_P}{N_{BL} + 1} \right) \rceil \quad (3.36)$$

La condizione affinchè avvenga il post-buffering è la ricezione del segnale di feeder disponibile successivamente al prelievo dell'ultimo pezzo dal buffer e prima

del termine della lavorazione dello stesso.

$$\begin{aligned} \tau_F < \tau_{Flim} + (N_{out} - 1)[(2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)] + \\ + (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L \end{aligned} \quad (3.37)$$

Il tempo di lavorazione rimasto nel momento in cui il manipolatore inizia la post-bufferizzazione, τ_{LR} , è la differenza fra i due membri della disequazione.

$$\begin{aligned} \tau_{LR} &= (N_{out} - 1)[(2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)] + \\ &\quad + (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L - (\tau_F - \tau_{Flim}) \\ &= (N_{out} - 1)(4\delta + 4\epsilon - \eta + \tau_L + \tau_H) + 3\delta - \eta + 2\epsilon + \tau_L - (\tau_F - \tau_{Flim}) \end{aligned} \quad (3.38)$$

Per determinare il numero di pezzi inseriti nel buffer prima della ricezione del termine lavorazione si segue una procedura simile a quella già utilizzata in precedenza per determinare N_{BL} e N_{out} .

$$\tau_{LR} \leq (2\delta - \eta + \epsilon) + (\delta + \epsilon) + (N_P - 1)[(\delta + \epsilon) + (\delta + \epsilon)] \quad (3.39)$$

Il valore ottenuto dall'equazione viene arrotondato all'intero superiore in quanto il pezzo prelevato va rilasciato nel buffer prima di poter procedere con le operazioni di scarico del pezzo in attesa nella postazione di lavorazione.

$$N_P = \lceil \left(\frac{\tau_{LR} - 3\delta + \eta + 2\epsilon}{2\delta + 2\epsilon} \right) \rceil + 1 \quad (3.40)$$

Se N_{out} è nullo la post-bufferizzazione non può avvenire in quanto non vi è alcun utilizzo del buffer, così come non avviene se $N_{out} \neq \Omega$ poichè in quel caso il manipolatore termina lo scarico dei pezzi dal buffer e attende l'arrivo del feeder in home. Nel solo caso in cui $N_{out} = \Omega$ si considera la post-bufferizzazione. Il tempo fittizio di lavorazione τ_{LP} dell'ultimo componente prelevato dal buffer è determinabile come segue:

$$\begin{aligned} \tau_{LP} &= \tau_L - \tau_{LR} + (2\delta - \eta + \epsilon) + (\delta + \epsilon) + (N_P - 1)[(\delta + \epsilon) + (\delta + \epsilon)] + \delta \\ &= \tau_L - \tau_{LR} + 2\delta - \eta + N_P(2\delta + 2\epsilon) \end{aligned} \quad (3.41)$$

$$\begin{aligned}
T_{CO} &= (N_{out} - 1)[(2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)] + \\
&\quad + (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_{LP} + \epsilon + (\delta + \epsilon) \quad (3.42) \\
&= (N_{out} - 1)(4\delta - \eta + 4\epsilon + \tau_L + \tau_H) + 4\delta - \eta + 4\epsilon + \tau_{LP}
\end{aligned}$$

Se il parametro N_P risulta maggiore di Ω è necessaria l'imposizione $N_P = \Omega$ e tutti i depositi nel buffer vengono pertanto effettuati in questa fase. Di conseguenza se $N_P \geq \Omega$ allora γ_A risulta necessariamente nulla e $\gamma_R = N_F - N_P$. Inoltre se i pezzi teoricamente inseribili in pre-bufferizzazione sono superiori a $\Omega - N_P$ è necessario utilizzare nuovamente l'approccio con il parametro e come in precedenza. Più precisamente se $\gamma N_{BL} - \lambda > \Omega - N_P$ allora:

$$e = (\gamma N_{BL} - \lambda) - (\Omega - N_P) \quad (3.43)$$

$$\gamma_A = \left\lfloor \frac{\Omega - N_P}{N_{BL}} \right\rfloor \quad (3.44)$$

$$\gamma_R = \gamma + e - \gamma_A \quad (3.45)$$

3.4.8 Influenza del tempo di scarico

Fino ad ora si è considerato trascurabile il tempo richiesto per il ricambio della postazione di output. Quando il tempo di scarico assume valori rilevanti influisce inevitabilmente sul tempo ciclo totale diventando il collo di bottiglia del ciclo. La massima priorità di svuotamento è attribuita alla stazione di lavorazione. Tuttavia finchè non è disponibile la stazione di scarico il manipolatore non afferra il pezzo lavorato ma attende in home o se possibile deposita ulteriori pezzi nel buffer. Nella formula 3.11 è stato introdotto il parametro ϕ , utile per semplificare le espressioni riguardanti il tempo di scarico. L'influenza di τ_S sul tempo ciclo può essere schematizzata come segue:

- Il numero di pezzi N_{out} che il manipolatore riesce a prelevare dal buffer durante il cambio feeder può diminuire a causa di τ_S . Se $\phi > (2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)$ allora:

$$\tau_F \leq \tau_{Flim} + N_{out}\phi \quad (3.46)$$

$$N_{out} = \lceil \frac{\tau_F - \tau_{Flim}}{\phi} \rceil \quad (3.47)$$

Generalizzando:

$$N_{out} = \lceil \frac{\tau_F - \tau_{Flim}}{\max(\phi, 4\delta + 4\epsilon - \eta + \tau_L + \tau_H)} \rceil \quad (3.48)$$

- Se τ_S è tale da influenzare il tempo richiesto per effettuare la macro-operazione di tipo 1 allora $T_{CA} = \phi$. Più precisamente tale condizione avviene se $\tau_S > (3\delta - 2\eta + \epsilon) + (2\delta - \eta + \epsilon) + \tau_{LA}$. Si può generalizzare la situazione definendo T_{CA} come il massimo fra i due valori possibili.

$$T_{CA} = \max(6\delta - 3\eta + 4\epsilon + \tau_{LA}, \phi) \quad (3.49)$$

- Se τ_S è tale da influenzare il tempo richiesto per effettuare la macro-operazione di tipo 2 allora $T_{CR} = \phi$. Questa condizione corrisponde a $\tau_S > (3\delta - 2\eta + \epsilon) + (2\delta - \eta + \epsilon) + \tau_L$.

$$T_{CR} = \max(6\delta - 3\eta + 4\epsilon + \tau_L + \tau_H, \phi) \quad (3.50)$$

- Se il tempo richiesto durante il cambio feeder per prelevare un pezzo dal buffer dalla posizione di scarico, posizionarlo in lavorazione, attendere la lavorazione è inferiore al tempo di scarico ne risente il parametro T_{CO} . Pertanto se $\tau_S > (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L$ allora $T_{CO} = N_{out}\phi$. Riassumendo:

$$T_{CO} = \max(N_{out}T_{CR}, \tau_F - \tau_{Flim} + \eta - \delta, \eta); \quad (3.51)$$

3.5 Priorità buffer

Si desidera ora analizzare il caso in cui la priorità di svuotamento sia attribuita al buffer. Il modello realizzato per il caso in esame utilizza gli stessi parametri introdotti nei precedenti paragrafi, tuttavia è necessario modificare parte delle formule. La maggiore differenza con il modello presentato precedentemente consiste nel fatto che i pezzi inseriti nel buffer durante l'attesa di una lavorazione non vengono interamente utilizzati durante l'attesa di ricambio feeder, bensì uno

di questi pezzi viene prelevato successivamente e posto in lavorazione anche se vi è disponibilità sul feeder. Se il numero di pezzi inseriti nel buffer durante ciascuna lavorazione è unitario allora non vi sarà accumulo di pezzi nel buffer in quanto la sequenza di movimenti permette solamente di avvicinare i componenti al macchinario di lavorazione.

3.5.1 Componenti depositati nel buffer

Il valore di N_{BL} è indipendente dalla priorità di svuotamento in esame, le equazioni 3.15 e 3.16 rimangono valide. La differente priorità di svuotamento non influisce sulla sequenza di movimenti del robot durante l'attesa di termine lavorazione.

3.5.2 Svuotamento del feeder

I parametri γ e λ , rappresentativi della condizione di svuotamento del feeder, ovvero dell'operazione in atto nel momento del prelievo dell'ultimo pezzo dal feeder assumono i seguenti valori:

$$\gamma = \lceil \left(\frac{N_F}{N_{BL}} \right) \rceil \quad (3.52)$$

$$\lambda = \gamma(N_{BL}) - N_F \quad (3.53)$$

Per entrambe le formule la differenza rispetto al caso precedente consiste nella sostituzione di $N_{BL} + 1$ con il solo termine N_{BL} . Questo è dovuto al fatto che il primo pezzo di ciascun ciclo γ viene prelevato dal buffer anzichè dal feeder, pertanto ciascun ciclo γ è costituito dall'inserimento di N_{BL} componenti nel buffer con iniziale prelievo di un componente da esso. Nel modello con la priorità di svuotamento attribuita al feeder invece ciascun ciclo γ comprendeva l'inserimento di N_{BL} pezzi nel buffer e il prelievo di un pezzo dal feeder. Tali affermazioni giustificano l'utilizzo del termine $N_{BL} + 1$ in quel caso.

3.5.3 Tempo limite di rinnovo feeder

In precedenza è stato considerato come tempo limite inferiore di ricambio feeder il tempo necessario, in seguito al prelievo dell'ultimo componente dal feeder

per raggiungere la postazione di scarico. Se il tempo di rinnovo dovesse essere inferiore a tale valore allora l'utilizzo del buffer diventerebbe superfluo. Nel caso in esame invece si considera come tempo limite il tempo richiesto in seguito allo svuotamento dello stesso per raggiungere la stazione di scarico e, in seguito, processare anche un successivo pezzo presente nel buffer. Se il feeder dovesse tornare disponibile in un tempo minore di quello calcolabile in questo modo allora il numero di postazioni sul buffer utilizzate sarebbe pari a 1 e l'utilità della postazione diventerebbe esclusivamente quella di avvicinare i componenti alla stazione di lavorazione. Non necessariamente tale situazione è un vantaggio per la cella. Questo dipende dalla relazione fra il guadagno di tempo derivante dall'afferraggio del componente in una locazione più vicina alla lavorazione e la percentuale di mascheramento dell'operazione di inserimento nel buffer del pezzo durante la lavorazione.

- se $\lambda = 0$ la bufferizzazione durante l'attesa di fine lavorazione avviene nella sua interezza, è possibile inserire N_{BL} componenti nel buffer. L'ultimo pezzo presente sul feeder viene prelevato, riposto sul buffer e non vi è attesa nella locazione di home.

$$\begin{aligned}\tau_{Flim} &= (\delta + \epsilon) + (\delta + \epsilon) + (\delta + \epsilon) + (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + \\ &\quad + (\tau_H + \epsilon) + (\delta + \epsilon) \\ &= 7\delta - \eta + 7\epsilon + \tau_L + \tau_H\end{aligned}\tag{3.54}$$

- se $\lambda \neq 0$ l'ultimo pezzo presente nel feeder viene prelevato durante la bufferizzazione ma questa viene effettuata in tempo totalmente mascherato, ovvero per un numero di componenti inferiore a N_{BL} . E' necessario determinare il tempo di lavorazione rimasto nel momento in cui il robot raggiunge la posizione di home:

$$\begin{aligned}w &= \tau_L - [(2\delta - \eta + \epsilon) + (\delta + \epsilon) + (N_{in} - \lambda)((\delta + \epsilon) + (\delta + \epsilon)) + (\delta)] \\ &= \tau_L - [(N_{in} - \lambda)(2\delta + 2\epsilon) + 4\delta + 2\epsilon - \eta]\end{aligned}\tag{3.55}$$

In seguito all'inserimento nel buffer dell'ultimo pezzo disponibile sul feeder il robot attende in home per un tempo w e procede con lo scarico del pezzo

successivamente. Se $w < 0$ il pezzo lavorato si rende disponibile durante il tragitto verso la posizione di home. Per i motivi esposti in precedenza il robot raggiunge comunque la locazione di home e w assume un valore nullo poichè il valore negativo non avrebbe alcun significato fisico.

$$\begin{aligned}\tau_{Flim} &= (\delta + \epsilon) + (\delta + w + \tau_H + \epsilon) + (\delta + \epsilon) + (2\delta - \eta + \epsilon) + \\ &\quad + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) \quad (3.56) \\ &= 7\delta + 7\epsilon + w + 2\tau_H - \eta + \tau_L\end{aligned}$$

3.5.4 Componenti prelevati dal buffer

Noto il tempo limite inferiore di rinnovo feeder è possibile determinare il numero di pezzi che è possibile prelevare dal buffer, lavorare e scaricare prima della segnalazione della nuova disponibilità del feeder. La procedura è identica a quella presentata in precedenza e la formula 3.24 per determinare N_{out} mantiene la propria validità. Rispetto al caso precedente tuttavia nel parametro τ_{Flim} è compresa anche la lavorazione di un pezzo disponibile nel buffer successivamente allo svuotamento del feeder. Per questo motivo con N_{out} si intendono tutti i pezzi prelevati dal buffer, lavorati e scaricati prima del termine della procedura di rinnovo del feeder escludendo il primo pezzo già disponibile nel buffer. La lavorazione e scarico di questo pezzo infatti sono indipendenti dallo stato del feeder, in accordo con l'attribuzione della priorità al buffer per lo svuotamento.

3.5.5 Tempo ciclo

Il tempo ciclo viene calcolato utilizzando un approccio simile al precedente. Le operazioni compiute dal robot si possono suddividere in tre macro-operazioni ed una correzione che in alcuni casi assume valore nullo. Come osservato in precedenza questo approccio consente di utilizzare una sola formula per il calcolo del tempo ciclo in tutti i casi possibili, semplificando la gestione dell'espressione finale.

1. La prima macro-operazione consiste nel prelievo di un pezzo dal buffer, deposito in lavorazione, attesa di un tempo τ_{L_A} in home, scarico e ritorno al feeder. In questa successione di operazioni il robot inserisce pezzi nel buffer

ma la loro entità non rientra nella formula grazie all'utilizzo di un tempo di lavorazione fittizio. La molteplicità di compimento di tale operazione, in accordo con la notazione utilizzata precedentemente, viene indicata con γ_A .

$$\begin{aligned} T_{CA} &= (\delta + \epsilon) + (\tau_{LA} + \epsilon) + (\delta + \epsilon) + (2\delta - \eta + \epsilon) \\ &= 4\delta - \eta + 4\epsilon + \tau_{LA} \end{aligned} \quad (3.57)$$

2. La seconda macro-operazione è simile alla prima con la differenza che non avviene bufferizzazione o, se questa è presente, è effettuata in tempo totalmente mascherato, cioè per un numero di pezzi inferiore a N_{BL} . La molteplicità con cui è compiuta tale macro-operazione in accordo con il modello precedentemente illustrato è definita γ_R .

$$\begin{aligned} T_{CR} &= (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) + (2\delta - \eta + \epsilon) \\ &= 4\delta - \eta + 4\epsilon + \tau_L + \tau_H \end{aligned} \quad (3.58)$$

3. L'ultima parte del ciclo di lavoro corrisponde alle operazioni effettuate durante il ricambio del feeder. In accordo con quanto illustrato questo ciclo corrisponde alla presa di N_{out} pezzi dal buffer, il trasporto in lavorazione e in seguito ad un'attesa τ_L i pezzi lavorati vengono riposti allo scarico.

$$T_{CO} = N_{out}[(2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)] \quad (3.59)$$

4. Si considera anche un'eventuale correzione al tempo ciclo nel caso in cui il primo pezzo venga prelevato dal feeder anzichè dal buffer a causa dell'assenza di componenti in esso. T_{CR} e T_{CA} infatti considerano il solo prelievo del pezzo dal buffer. I casi in cui viene utilizzata tale correzione al tempo ciclo verranno discussi nei seguenti paragrafi, per il momento ci si limita ad introdurre tale valore. La correzione è ininfluente dal valore di γ_A e γ_R in quanto solo il primo ciclo sarà caratterizzato da tale correzione, sia esso di durata T_{CR} o T_{CA} .

$$\begin{aligned} \Delta_F &= \delta_{FS} - \delta_{BS} \\ &= (3\delta - 2\eta + \epsilon) - (2\delta - \eta + \epsilon) \\ &= \delta - \eta \end{aligned} \quad (3.60)$$

La formula generale del tempo ciclo per il modello con priorità di svuotamento del buffer è la seguente:

$$T_C = \frac{\gamma_A T_{CA} + \gamma_R T_{CR} + T_{CO} + \Delta_F}{N_F} \quad (3.61)$$

3.5.6 Post-buffering

La post-bufferizzazione è stata definita in precedenza come l'operazione di inserimento di pezzi nel buffer nel caso in cui il feeder dovesse rendersi disponibile prima del termine di una lavorazione. Le formule 3.37, 3.38 e 3.40 introdotte per il modello precedente rimangono valide poichè il parametro τ_{Flim} , presente in esse, è stato adattato in modo da tenere conto delle differenze fra i modelli. La post-bufferizzazione non influisce sul numero di pezzi inseriti nel buffer in ciascun ciclo, tuttavia impone un differente sequenziamento delle operazioni del robot. L'espressione corretta di γ e λ , determinato N_P , è la seguente:

$$\gamma = \lceil \left(\frac{N_F - N_P}{N_{BL}} \right) \rceil \quad (3.62)$$

$$\lambda = \gamma N_{BL} - (N_F - N_P) \quad (3.63)$$

Il valore N_P modifica γ e λ i quali hanno influenza sul parametro τ_{Flim} e di conseguenza su N_{out} e N_P . È possibile iterare il modello fino al raggiungimento della convergenza.

3.5.7 Flusso nullo nel buffer

Il numero di pezzi immessi nel buffer ad ogni *ciclo-feeder* è indicato con N_{in} . Tenendo conto di un'eventuale dimensione limite del buffer imposta N_B si può determinare la seguente espressione:

$$N_{in} = \min(\gamma(N_{BL} - 1) - \lambda - 1, N_B - 1) \quad (3.64)$$

Il parametro Ω viene utilizzato per identificare il numero di postazioni utilizzate nella stazione di buffering durante il ricambio feeder. Il dimensionamento della

stazione di conseguenza richiede un valore minimo pari a $\Omega + 1$ per considerare lo slot utilizzato per l'avvicinamento del componente.

$$\Omega = \min(N_{in}, N_{out}) \quad (3.65)$$

Si desidera analizzare come gestire il modello nei casi in cui non ci sia il flusso nullo, in altri termini se $N_{in} \neq \Omega$ o $N_{out} \neq \Omega$.

1. Se $N_{in} \neq \Omega$ si procede con l'iterazione del modello imponendo una dimensione del buffer pari a $\Omega + 1$.
2. Se $N_{out} = 0$ il buffer viene utilizzato esclusivamente per avvicinare i pezzi alla stazione di lavorazione.
 - Se $N_P = 0$ il primo pezzo di ciascun ciclo è prelevato dal feeder e si aggiunge la correzione prima esposta per il tempo ciclo. Inoltre $T_{CO} = 0$.
 - Se $N_P \neq 0$ impongo per questo un valore unitario, poichè a regime vi sarebbe un solo slot libero. Il primo pezzo è prelevato dal buffer.

$$T_{CO} = \tau_{LP} - (\tau_L + \tau_H) \quad (3.66)$$

Indipendentemente dal valore di N_P :

- Se $N_{BL} \neq 1$: $\gamma_A = 0$ e $\gamma_R = N_F$
- Se $N_{BL} = 1$: $\gamma_A = N_F$ e $\gamma_R = 0$

3. Se $N_{out} \neq \Omega$

Il primo ciclo prevede sicuramente il prelievo del pezzo dal feeder anzichè dal buffer, non vi è modo di avere post-bufferizzazione ($N_P = 0$). Per la definizione di γ_A e γ_R si utilizza l'approccio già esposto in precedenza sfruttando il parametro e . Tale approccio è generale e valido per ciascun caso in esame. Se $\gamma(N_{BL} - 1) - \lambda - 1 > \Omega - N_P$ allora:

$$e = \gamma(N_{BL} - 1) - \lambda - 1 - (\Omega - N_P) \quad (3.67)$$

$$\gamma_A = \lfloor \frac{\Omega - N_P}{N_{BL} - 1} \rfloor \quad (3.68)$$

$$\gamma_R = \gamma + e - \gamma_A + 1 - N_P \quad (3.69)$$

Se invece la disequazione non è soddisfatta:

- se $\lambda = 0$: $\gamma_A = \gamma$ e $\gamma_R = 1$
- se $\lambda \neq 0$: $\gamma_A = \gamma - 1$ e $\gamma_R = 2$

In generale:

$$\begin{aligned} T_{CO} &= \max(\max(\tau_F - \tau_{Flim}, \delta) + (2\delta - \eta + \epsilon) - (3\delta - 2\eta + \epsilon), \Omega T_{CR}) \\ &= \max(\tau_F - \tau_{Flim} + \eta - \delta, \eta, \Omega T_{CR}) \end{aligned} \quad (3.70)$$

4. Negli altri casi possibili, ovvero se $\Omega = N_{out} = N_{in}$ e $\Omega \neq 0$:

Se $N_P = 0$ l'equazione di T_{CO} è la precedente. Altrimenti viene considerato anche il contributo della post-bufferizzazione, ovvero $\Omega - 1$ cicli di durata T_{CR} e un ultimo ciclo di durata $T_{CR} - \tau_L + \tau_{LP}$.

$$T_{cout} = \max(\tau_F - \tau_{Flim} + \eta - \delta, \eta, \Omega T_{CR}, \Omega T_{CR} + \tau_{LP} - \tau_L) \quad (3.71)$$

Se $N_P > N_{out} + 1$ è necessario imporre $N_P = N_{out} + 1$. Se $N_P = \Omega + 1$ allora:

- Se $N_{BL} \neq 1$: $\gamma_A = 0$ e $\gamma_R = N_F - N_P$
- Se $N_{BL} = 1$: $\gamma_A = N_F - N_P$ e $\gamma_R = 0$

Altrimenti valgono le considerazioni precedenti su γ_R e γ_A .

3.5.8 Feeder unitario

Si considera il caso particolare in cui N_F sia pari a 1. E' necessario determinare se quell'unico pezzo viene prelevato dal feeder o se questo viene immesso nel buffer durante la lavorazione e prelevato successivamente dal buffer.

Se $\tau_F > (2\delta - \eta + \epsilon) + \tau_L$ il segnale di pezzo disponibile sul feeder viene ricevuto dopo il termine della lavorazione del pezzo precedente e si ricade nel caso del

modello senza buffer. Se invece la disequazione non è verificata si determina il tempo apparente di lavorazione dovuto alla bufferizzazione dell'unico pezzo e in seguito il tempo ciclo:

$$\tau_{L_A} = \max(\tau_L + \tau_H, (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \delta) \quad (3.72)$$

$$\begin{aligned} T_C &= (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_{L_A} + \epsilon + (\delta + \epsilon) \\ &= 4\delta + 4\epsilon - \eta + \tau_{L_A} \end{aligned} \quad (3.73)$$

3.5.9 Influenza del tempo di scarico

In accordo con lo sviluppo dei modelli precedenti si è considerato fino ad ora trascurabile il tempo richiesto per il ricambio della postazione di output. Quando il tempo di scarico assume valori rilevanti influisce inevitabilmente sul tempo ciclo totale diventando il collo di bottiglia del ciclo. Risulta di indubbia utilità il parametro ϕ definito precedentemente nella formula 3.11, utile per semplificare le espressioni riguardanti il tempo di scarico. L'influenza di τ_S sul tempo ciclo può essere schematizzata come segue:

- Il numero di pezzi N_{out} che il manipolatore riesce a prelevare dal buffer durante il cambio feeder può diminuire a causa di τ_S . Se $\phi > (2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon)$ allora:

$$\tau_F \leq \tau_{Flim} + N_{out}\phi \quad (3.74)$$

$$N_{out} = \lceil \frac{\tau_F - \tau_{Flim}}{\phi} \rceil \quad (3.75)$$

Generalizzando:

$$N_{out} = \lceil \frac{\tau_F - \tau_{Flim}}{\max(\phi, 4\delta + 4\epsilon - \eta + \tau_L + \tau_H)} \rceil \quad (3.76)$$

- Se τ_S è tale da influenzare il tempo richiesto per effettuare la macro-operazione di tipo 1 allora $T_{CA} = \phi$. Più precisamente tale condizione avviene se $\tau_S > (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_{L_A}$ allora $T_{CA} = \phi$. Si può generalizzare la situazione definendo T_{CA} come il massimo fra i due valori possibili.

$$T_{CA} = \max(4\delta - \eta + 4\epsilon + \tau_{L_A}, \phi) \quad (3.77)$$

- Se τ_S è tale da influenzare il tempo richiesto per effettuare la macro-operazione di tipo 2 varia T_{CR} . Se $\tau_S > (2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L$ allora $T_{CR} = \phi$.

$$T_{CR} = \max(4\delta - \eta + 4\epsilon + \tau_L + \tau_H, \phi) \quad (3.78)$$

Capitolo 4

Implementazione del modello

In questo capitolo viene analizzata l'implementazione del modello esposto nel capitolo 3. Viene esposta l'impostazione dell'implementazione in Matlab e le problematiche riscontrate in tale operazione. Si determina inoltre un valore limite di confronto del risultato finale come controllo di eventuali errori presenti nell'algoritmo. Viene poi analizzata separatamente la gestione del caso in cui il numero di componenti nel feeder sia variabile in ogni ciclo in quanto tale caso richiede una simulazione con script apposito.

4.1 Implementazione in Matlab

Il modello è stato implementato in Matlab con lo scopo di ottenere una funzione che abbia come argomento i parametri caratteristici della cella. I valori assunti dalla funzione sono il tempo ciclo calcolato e il numero di slot del buffer realmente utilizzati a regime. È stata realizzata una differente funzione per ciascuno dei tre casi in esame: ciclo senza buffer e ciclo con buffer considerando le due possibili priorità di gestione.

Alcuni controlli iniziali vengono effettuati sulle variabili in ingresso per valutarne l'effettivo significato fisico e interrompere eventualmente l'algoritmo. Successivamente questo manipola le variabili in ingresso secondo il modello esposto tramite una serie di cicli *if*, valutando il risultato delle disequazioni in esame. Non sono state riscontrate particolari problematiche nella stesura del codice ad

eccezione della gestione della sua ricorsività, argomento trattato nella prossima sezione.

4.1.1 Ricorsività

Come esposto durante l'analisi del modello teorico è necessario in alcune situazioni ricercare la convergenza della soluzione. A livello di algoritmo tale richiesta è stata gestita utilizzando il concetto di ricorsività: all'interno della funzione vi è un richiamo a sè stessa con nuovi parametri in ingresso e l'arresto della computazione precedente. La nuova funzione può tuttavia richiamare nuovamente sè stessa e avviare un ciclo che non necessariamente raggiunge la convergenza della soluzione. Matlab interrompe di default un processo ricorsivo al raggiungimento delle 500 iterazioni e consente l'impostazione di un valore differente all'occorrenza. Negli algoritmi sviluppati, se la convergenza è raggiungibile, essa richiede un numero di iterazioni inferiore alla decina, altrimenti il ciclo instaurato si ripete uguale a sè stesso ogni m iterazioni. Nel caso di mancato raggiungimento della convergenza risulta pertanto inutile aumentare oltre 500 il numero di iterazioni concesse di default da Matlab.

Due sono le ricorsività introdotte nel modello. La prima riguarda il caso $N_{in} \neq \Omega$ (sezione 3.4.6 e 3.5.7). In tale situazione si potrebbero inserire più componenti nel buffer rispetto a quelli prelevabili in seguito e pertanto è necessario imporre una limitazione nel processo di buffering. La definizione dei parametri γ e λ è strettamente dipendente dal processo di bufferizzazione utilizzato (3.35), tuttavia la loro determinazione avviene prima di determinare la condizione $N_{in} \neq \Omega$. Tali parametri condizionano il resto del modello e quindi il valore del tempo ciclo. Poichè uno dei parametri in ingresso alla funzione corrisponde alla dimensione del buffer l'espedito utilizzato per gestire tale inconveniente consiste nel richiamare la funzione imponendo una dimensione del buffer pari a Ω nel caso di priorità associata al feeder e $\Omega+1$ nella duale situazione. Tale ricorsività non crea problemi di convergenza poichè non è prevista la possibilità di richiamare la funzione con una dimensione del buffer superiore a quella in questione. Per questo motivo il raggiungimento della convergenza avviene in ogni caso.

La seconda situazione che richiede ricorsività riguarda il post-buffering. Essa è caratterizzata da una maggiore complessità di implementazione rispetto alla prima. Si faccia riferimento alla sezione 3.4.7 per il caso con priorità attribuita al feeder. Il caso duale è analogo e non verrà preso in considerazione. Determinato N_P con l'equazione 3.40 è necessario considerare che il numero dei pezzi disponibili concettualmente nel feeder al ciclo successivo sia ora pari a $N_F - N_P$. Questo perchè parte di essi sono già stati riposti nel buffer al termine del ciclo precedente. Di conseguenza varia λ (equazione 3.35), parametro rappresentativo della situazione di svuotamento del feeder. Esso determina τ_{Flim} (equazioni 3.19, 3.20 e 3.22) il quale a sua volta influenza il numero di pezzi N_{out} prelevabili dal buffer durante il ciclo (3.24). A seconda della relazione fra N_{out} e N_{BL} avviene o meno la post-bufferizzazione e può variarne la sua entità. In altre parole può variare N_P rispetto al valore iniziale. Anche in questo caso è richiesta la ricorsività dell'algoritmo per determinare il valore a convergenza dei parametri citati.

Un approccio ricorsivo a tale problema in Matlab si è dimostrato insoddisfacente in quanto il raggiungimento della convergenza si dimostra in alcuni casi impossibile. La principale problematica è rappresentata dalle tre differenti equazioni che determinano τ_{Flim} (3.19, 3.20 e 3.22). È stato in tali occasioni osservato che il valore di λ varia ad ogni iterazione in modo tale da costituire un ciclo continuo di tre o meno iterazioni successive, determinando di volta in volta un valore differente di τ_{Flim} .

In caso di mancato raggiungimento della convergenza si è cercato di determinare un criterio per la scelta del valore di N_P con cui proseguire l'algoritmo. Si è pensato di considerare tutti i valori di N_P ottenuti durante le iterazioni e utilizzare tali valori per calcolare i rispettivi tempi ciclo ed effettuare la media fra i risultati o prenderne il valore minore. Tutti gli approcci presi in considerazione hanno portato all'ottenimento di curve instabili e non sufficientemente compatibili con i dati sperimentali. Per le ragioni esposte la post-bufferizzazione è stata affrontata in modo approssimativo con il solo scopo di evitare la complicazione derivante da un'ulteriore ricorsività del modello. Si determina cioè N_P e si prosegue con l'algoritmo senza ricercarne la convergenza.

4.2 Confronto con cicli limite

Per avere un controllo sull'attendibilità del valore fornito dagli algoritmi è possibile confrontare quest'ultimo con il valore ottenuto da un ciclo limite. Con il termine *ciclo limite* si intende un ciclo ideale che consente il raggiungimento di una tempistica che rappresenta il limite inferiore ottenibile dalla cella. Si potrebbe pensare che il ciclo con il tempo teorico minore sia quello ottenibile da una cella in cui il pezzo nel feeder risulta sempre disponibile. In figura 4.1 è rappresentata tale sequenza di movimenti. Essa coincide con la presenza di un bowl-feeder all'interno della cella.

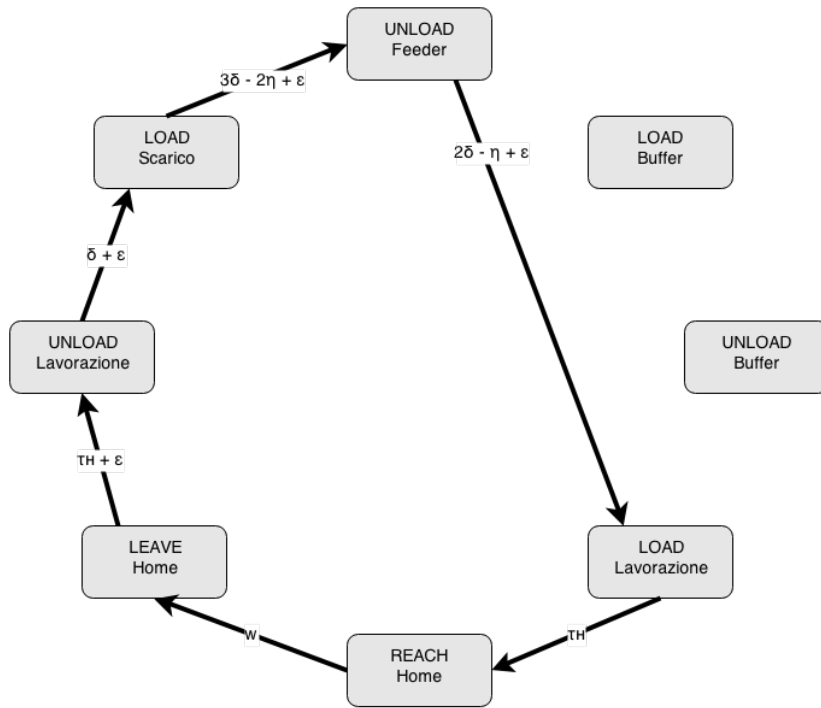


Figura 4.1: Rappresentazione nel diagramma degli stati del ciclo limite con bowl-feeder.

$$\begin{aligned}
 T_{limF} &= \max((2\delta - \eta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) + (3\delta - 2\eta + \epsilon), \phi) \\
 &= \max(6\delta - 3\eta + 4\epsilon + \tau_L + \tau_H, \phi)
 \end{aligned}
 \tag{4.1}$$

Con il layout in esame tuttavia si possono ottenere tempi ciclo inferiori al limite appena esposto. Questo è dovuto alla sola considerazione che il buffer si

trova ad una distanza minore dalla stazione di lavorazione rispetto al feeder. I movimenti da e verso il buffer richiedono meno tempo rispetto a quelli riguardanti la postazione di ingresso componenti. In una situazione limite ideale è possibile riempire il buffer infinitamente e prelevare i pezzi esclusivamente da esso. La sequenza di movimenti in tal caso è rappresentata in figura 4.2. Il tempo limite ottenibile in questa teorica situazione risulta:

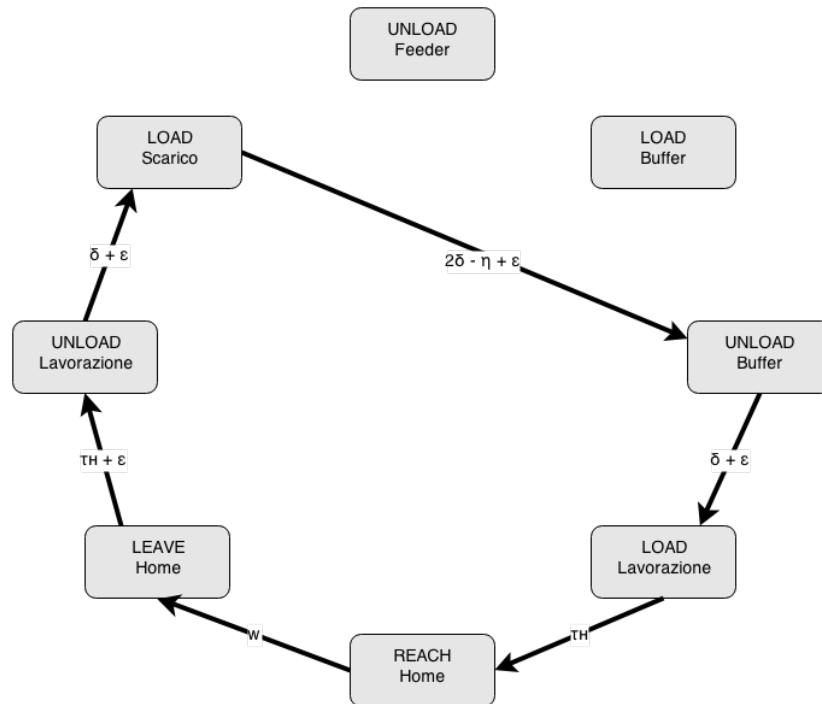


Figura 4.2: Rappresentazione nel diagramma degli stati del ciclo limite con bowl-feeder nella posizione del buffer.

$$\begin{aligned}
 T_{limB} &= \max((\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon) + (2\delta - \eta + \epsilon), \phi) \\
 &= \max(4\delta - \eta + 4\epsilon + \tau_L + \tau_H, \phi)
 \end{aligned}
 \tag{4.2}$$

Al termine dell'algorithmo implementato in Matlab, in seguito al calcolo del tempo ciclo viene valutato T_{limB} tramite l'equazione 4.2 e si confrontano i due risultati. Nell'eventualità in cui il risultato sia inferiore al tempo limite il modello restituisce un errore. Tale accorgimento è stato ampiamente utilizzato nelle prime

fasi delle stesura dell'algoritmo per correggere gli errori o identificare situazioni non ancora prese in considerazione dal modello.

4.3 Dimensione del feeder variabile

Gli algoritmi implementati fino ad ora considerano la sola situazione a regime e non prevedono la possibilità di imporre una disponibilità variabile di componenti ad ogni rinnovo del feeder. La successione del numero di componenti disponibili nel feeder ad ogni ciclo può essere simulata, nota la distribuzione di probabilità rispetto al valore massimo, e racchiusa in un array in Matlab. Si potrebbe valutare il tempo ciclo a regime per ciascuno di questi valori ed effettuarne in seguito la media, tuttavia un tale approccio risulterebbe eccessivamente approssimato e non coerente con la sequenza reale di movimenti. Nel modello base si è imposto infatti il bilancio nullo all'interno del buffer per ciascun ciclo di lavoro. Questa assunzione non rappresenta la situazione nel caso di N_F variabile poichè, verosimilmente, nei cicli in cui la dimensione del feeder è elevata vi sarà accumulo nel buffer mentre nei cicli con dimensione ridotta vi sarà maggiore prelievo di componenti da esso. Il bilancio nel buffer non è pertanto nullo per ciascun ciclo.

Vengono ora presentate le modifiche al modello realizzate per valutare la situazione in esame. Alla funzione in Matlab vengono forniti due ulteriori parametri: lo stato di riempimento del buffer e un flag per indicare la modalità richiesta. Se tale flag indica l'imposizione di flusso nullo, l'algoritmo procede come esposto nel capitolo precedente; se il flag richiede invece la valutazione del tempo ciclo non a regime, l'algoritmo segue le formule presentate in questa sezione e stima la produttività per il solo ciclo considerato. In questo caso, il secondo valore in uscita dalla funzione non corrisponde al numero di slot del buffer utilizzati come in precedenza. Esso rappresenta il bilancio complessivo per quel ciclo: positivo se il numero di componenti immessi è superiore all'entità del prelievo e viceversa. Uno script di Matlab gestisce il calcolo della successione del numero di componenti disponibili nel feeder, valuta lo stato del buffer al termine di ciascun ciclo e si occupa di richiamare in successione le funzioni, fornendo loro i corretti argomenti in input.

Il tempo ciclo è calcolabile al termine della simulazione come segue:

$$T_C = \frac{\sum_{i=1}^n T_{c_i} N_{feed_i} + T_b N_b}{\sum_{i=1}^n N_{feed_i}} \quad (4.3)$$

Si è indicato con T_{c_i} il tempo del singolo ciclo i -esimo e con N_{feed_i} il numero di componenti presenti nel feeder per lo stesso ciclo. Ai cicli simulati, tuttavia, ne va aggiunto uno ulteriore che consideri lo svuotamento del buffer per ricondursi alle condizioni iniziali. Si considera un numero di cicli da buffer verso la lavorazione pari agli slot rimasti occupati, altrimenti il valore di T_C ottenuto non corrisponde a quello reale. Infatti, per calcolare il tempo ciclo si deve avere la situazione finale coincidente con quella iniziale. In 4.3 si indica con N_b il numero di pezzi nel buffer in eccesso alla situazione iniziale. Il tempo in quest'ultimo ciclo è dato da

$$\begin{aligned} T_b &= \max((2\delta - \eta + \epsilon) + (\delta + \epsilon) + \tau_L + (\tau_H + \epsilon) + (\delta + \epsilon), \phi) \\ &= \max(4\delta - \eta + 4\epsilon + \tau_L + \tau_H, \phi) \end{aligned} \quad (4.4)$$

Il tempo ciclo può anche essere calcolato trascurando al denominatore il contributo di N_b .

$$T_C = \frac{\sum_{i=1}^n T_{c_i} N_{feed_i}}{\left(\sum_{i=1}^n N_{feed_i}\right) - N_b} \quad (4.5)$$

I tempi limite non possono essere un controllo efficace nel caso di singoli cicli in quanto l'immissione di pezzi nel buffer senza un successivo prelievo determina tempistiche inferiori a queste. Il controllo sul tempo ciclo è invece effettuabile sul tempo finale complessivo.

4.3.1 Priorità feeder

Nell'algoritmo si può trascurare il parametro Ω poichè il flusso nullo non è richiesto. Si indichi con b il numero di postazioni del buffer occupate all'inizio del ciclo. Il numero di pezzi effettivamente immessi nel buffer nel ciclo è il valore minimo fra quelli teoricamente consentiti e lo spazio disponibile nel buffer.

- Se $\lambda = 0$

$$N_{in} = \min(\gamma_A N_{BL} - \lambda + N_P, N_B - b) \quad (4.6)$$

- Se $\lambda \neq 0$

$$N_{in} = \min((\gamma_A + 1)N_{BL} - \lambda + N_P, N_B - b) \quad (4.7)$$

Il numero di pezzi effettivamente prelevati dal buffer invece è il valore minimo fra i pezzi idealmente prelevabili e quelli realmente disponibili nel buffer.

$$N_{out} = \min(N_{out}, b + N_{in}) \quad (4.8)$$

Il bilancio complessivo del buffer, non necessariamente nullo, per il ciclo in esame è pari alla differenza fra i due parametri appena calcolati.

$$\Delta = N_{in} - N_{out} \quad (4.9)$$

Tale parametro viene fornito in uscita alla funzione per calcolare il parametro b in ingresso alla funzione rappresentante il successivo ciclo.

Nella situazione a regime la formulazione di T_{CO} dipendeva dal parametro Ω . In questo caso è invece generalizzabile in un'unica espressione che racchiude tutti i casi possibili.

$$T_{CO} = \max(N_{out}(4\delta - \eta + 4\epsilon + \tau_L + \tau_H), \tau_F - \tau_{Flim} + \eta - \delta, \eta, N_{out}(4\delta - \eta + 4\epsilon + \tau_L + \tau_H) + \tau_{LP} - \tau_L - \tau_H) \quad (4.10)$$

L'ultimo termine è trascurabile se $N_P = 0$ in quanto in tal caso coincide con il primo, tuttavia l'equazione non perde validità.

La gestione dei parametri γ_A e γ_R è gestita analogamente al modello a regime:

- Se $\lambda = 0$: $\gamma_A = \gamma$, $\gamma_R = 0$
- Se $\lambda \neq 0$:
 - Se $\gamma = 0$: $\gamma_A = 0$, $\gamma_R = 1$
 - Se $\gamma \neq 0$: $\gamma_A = \gamma - 1$, $\gamma_R = 1$

Va considerato tuttavia il caso in cui si raggiunga la saturazione del buffer considerando anche la condizione iniziale dello stesso. Se $\gamma N_{BL} - \lambda > N_B - b - N_P$ infatti:

$$e = (\gamma N_{BL} - \lambda) - (N_B - b - N_P) \quad (4.11)$$

Di conseguenza:

$$\gamma_A = \left\lfloor \frac{N_B - b - N_P}{N_{BL}} \right\rfloor \quad (4.12)$$

$$\gamma_R = \gamma - \gamma_A + e \quad (4.13)$$

4.3.2 Priorità buffer

Analogamente al caso precedente il numero di componenti effettivamente immessi nel buffer durante il ciclo-feeder è il valore minimo fra quelli teoricamente bufferizzabili e lo spazio disponibile nel buffer, tenendo conto delle differenze nel processo di buffering.

- Se $\gamma = 0$: $N_{in} = \min(N_P, N_B - b)$
- Se $\gamma \neq 0$:
 - Se $\lambda = 0$: $N_{in} = \min(\gamma(N_{BL} - 1) + N_P, N_B - b)$
 - Se $\lambda \neq 0$: $N_{in} = \min(\gamma(N_{BL} - 1) - \lambda + N_P, N_B - b)$

Il numero di componenti effettivamente prelevati dal buffer è equivalente al caso precedente (4.8) così come il calcolo del bilancio nel buffer (4.9) e l'equazione per il calcolo di T_{CO} (4.10).

Più complessa risulta invece l'assegnazione dei parametri γ_A e γ_R :

- Se $N_{in} = N_P$:
 - Se $N_{BL} = 1$: $\gamma_A = N_F - N_{in} - 1$, $\gamma_R = 1$
 - Se $N_{BL} \neq 1$: $\gamma_A = 0$, $\gamma_R = N_F - N_{in}$
- Se $N_{in} \neq N_P$:

- Se $N_{BL} = 1$: $\gamma_A = \gamma + N_F - N_{in}$, $\gamma_R = 1$
- Se $N_{BL} \neq 1$: $\gamma_A = \gamma$, $\gamma_R = 1 + N_F - N_{in}$

Capitolo 5

Apparato sperimentale

Un'intensiva campagna di prove sperimentali è stata effettuata per valutare la compatibilità dei risultati teorici con le tempistiche misurabili in una situazione reale. Per tale motivo una cella di lavoro è stata allestita presso il dipartimento di robotica dell'università di Padova. Si desidera infatti convalidare il modello, per quanto possibile, con dati sperimentali così da fornire attendibilità alle curve parametriche risultanti dallo studio. Nei prossimi paragrafi viene descritto l'apparato sperimentale allestito per effettuare tali prove.

5.1 Robot e controller

Il robot utilizzato è il modello C4 prodotto dalla Epson Seiko. E' il più recente manipolatore a 6 assi prodotto dall'azienda ed è distribuito in due versioni differenti: C4 e C4L. Il secondo è caratterizzato da uno sbraccio maggiore, 900 mm anziché 600 mm, a scapito della ripetibilità, ± 0.03 mm contro i ± 0.02 mm del modello C4. Una panoramica delle principali specifiche della serie C4 è riportata in figura 5.2. Il modello utilizzato è quello con sbraccio minore. Per la precisione è stato utilizzato un robot Epson C4-A601S come il modello riportato in figura 5.1.

Il controller associato al robot è l'Epson RC700, modello più recente della ditta e unico dispositivo che supporta il robot C4. Altri prodotti a 6 assi della stessa azienda consentirebbero invece il supporto da parte di diversi modelli di controller



Figura 5.1: Modello Epson C4 e controller Epson RC700.

permettendo all'utente la relativa scelta. Ad esempio, sempre considerando i robot a 6 assi, sia la serie C3 sia la serie S5 sono supportate da entrambe le versioni RC180 e RC620.

La pinza montata sul manipolatore è progettata e realizzata da un'azienda per la quale il dipartimento sta realizzando un progetto commerciale. L'end effector consente l'afferraggio di un determinato modello di occhiali tramite valvola bistabile. Tra i segnali forniti dal robot vi sono quelli relativi all'apertura e alla chiusura della pinza. La terna utensile associata alla pinza è ottenuta tramite la trasformazione $(0, 0, 90, 0, 0, 0)$ applicata alla terna della flangia robot. Tale affermazione indica che la pinza ha un'estensione di 90 mm lungo l'asse z della flangia robot. Disponendo la terna utensile il controller è in grado di calcolare quale posizione della flangia consente di raggiungere la locazione desiderata e di conseguenza i valori in coordinate di giunto. Per questioni commerciali non viene riportata una foto dettagliata dell'end-effector in questione.

| EPSON ROBOT MODEL | | C4-A60*** | C4L-A90*** |
|-------------------------|------------------|---|-------------|
| Configuration | | Articulated 6 Axis | |
| Mounting Configurations | | Table Top, Ceiling | |
| Payload | | Rated 1 kg / Max. 4 kg (5 kg*) | |
| Repeatability | | ±0.02 mm | ±0.03 mm |
| Axis Rotation | J1 (Turning) | ±170 deg | |
| | J2 (Lower Arm) | -160 ~ +65 deg | |
| | J3 (Upper Arm) | +225 ~ -51 deg | |
| | J4 (Wrist Roll) | ±200 deg | |
| | J5 (Wrist Bed) | ±135 deg | |
| | J6 (Wrist Twist) | ±360 deg | |
| Horizontal Reach | to mounting face | 665 | 965 |
| | to wrist center | 600 | 900 |
| Vertical Reach | to mounting face | 885 | 1185 |
| | to wrist center | 820 | 1120 |
| Speed | J1 | 450 deg/sec | 275 deg/sec |
| | J2 | 450 deg/sec | 275 deg/sec |
| | J3 | 514 deg/sec | 289 deg/sec |
| | J4 | 555 deg/sec | |
| | J5 | 555 deg/sec | |
| | J6 | 720 deg/sec | |
| Cycle Time | 1 kg workload | 0.37 sec | 0.47 sec |
| Allowable Moment | J4 | 4.41 N*m | |
| | J5 | 4.41 N*m | |
| | J6 | 2.94 N*m | |
| Moment of Inertia | J4 | 0.15 kg*m ² | |
| | J5 | 0.15 kg*m ² | |
| | J6 | 0.1 kg*m ² | |
| Motor Ratings | Axis 1 | 400 watts | |
| | Axis 2 | 400 watts | |
| | Axis 3 | 150 watts | |
| | Axis 4 | 50 watts | |
| | Axis 5 | 50 watts | |
| | Axis 6 | 50 watts | |
| Environment | Temperature | 0 ~ 40 deg C | |
| | Humidity | 20 ~ 80% RH (no condensation) | |
| | Other: | Keep away from: *Flammable/corrosive gases *Flammable/corrosive solvents *Water or oil and powder dust *Sources of electric noise | |
| Weight | | 27 kg | 29 kg |

Figura 5.2: Principali specifiche dei manipolatori industriali della serie C4 Epson Seiko.

5.2 Postazioni di lavoro

Le postazioni di lavoro all'interno della cella sono state considerate puntiformi in accordo con le ipotesi avanzate durante la realizzazione del modello. Tale affermazione implica l'utilizzo di un'unica locazione istruita al robot per ciascuna postazione. La locazione delle postazioni in coordinate cartesiane è stata scelta in seguito a considerazioni sullo spazio di lavoro del robot utilizzato. Un manipolatore industriale infatti garantisce le migliori prestazioni in corrispondenza di locazioni non in prossimità dei limiti dell'area raggiungibile. In figura 5.3 è

riportato un estratto dello spazio di lavoro del robot Epson C4. Le postazioni sono state collocate lungo una circonferenza con raggio pari alla media geometrica fra i valori minimi e massimi raggiungibili all'altezza del secondo giunto. In particolare:

$$r_{av} = \sqrt{\frac{r_{min}^2 + r_{max}^2}{2}} = 461.66mm \quad (5.1)$$

Le postazioni sono state distribuite simmetricamente e uniformemente lungo tale

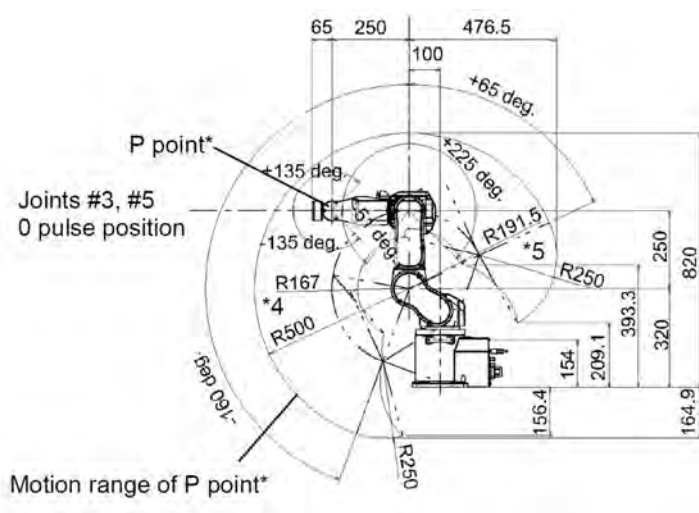


Figura 5.3: Schema dello spazio di lavoro raggiungibile dal manipolatore. Nel manuale sono riportate le restanti viste.

circonferenza. La maggior parte dei test sono realizzati considerando le postazioni distribuite su un angolo complessivo pari a 120° . Per valutare l'influenza del parametro δ è stata considerata anche una configurazione in cui l'angolo occupato è pari a 60° .

L'altezza imposta per le postazioni è pari a 140 mm rispetto alla terna world, in accordo con l'intenzione di posizionarle in una zona centrale rispetto allo spazio di lavoro del robot. L'orientazione della terna di destinazione è stata scelta costante, indipendentemente dalla postazione in esame: si considera cioè che il pezzo venga prelevato e riposto mantenendo sempre la medesima orientazione, per semplicità, sebbene tale considerazione richieda un maggiore lavoro per il se-



Figura 5.4: Quote per le due configurazioni di cella considerate. I valori riportati sono sufficienti per identificare le coordinate cartesiane nel riferimento xy del manipolatore.

sto giunto. In sezione 6.1.2 viene analizzata l'influenza di tale considerazione sulle tempistiche di movimentazione.

A causa del numero elevato di ore in cui è stato utilizzato il manipolatore per effettuare le prove, si è preferito effettuare la movimentazione impostando ogni motore in modo da non superare l'85% della velocità e dell'accelerazione massime da loro raggiungibili in modo da limitarne l'usura.

Lo stato delle postazioni di lavoro è stato gestito tramite relè temporizzati installati nel quadro elettrico. Ad ogni relè sono associati due segnali robot, uno di tipologia input e uno output per il robot. Il segnale di output è utilizzato per sollecitare il relè e simulare il cambio di stato da disponibile ad occupato. Nel momento in cui il robot alza tale segnale, vi è un immediato cambio di stato del relativo segnale di input. Al termine del tempo impostato manualmente sul relè, il segnale di input cambia nuovamente stato, indicando che la postazione è nuovamente disponibile. Il segnale di input indica pertanto al robot lo stato della postazione. Ciascun relè è connesso inoltre a due indicatori luminosi situati all'interno della cella per consentire l'osservazione dello stato delle postazioni, sebbene tale accorgimento non abbia alcuno scopo pratico.

Sono stati impiegati due differenti tipologie di relè in quanto già disponibili in laboratorio: due modelli TU93 prodotti dalla CdcElettromeccanica e due modelli M1MFT prodotti dalla Broyce Control. Entrambi consentono diverse opzioni di regolazione del tempo di switch per valori inferiori al minuto, in accordo con le

prove che si desidera effettuare.



Figura 5.5: Relè temporizzati CdcElettromeccanica TU93 (sinistra) e Broyce Control M1MFT (destra).

5.3 Software per la gestione della cella

Un robot è un dispositivo che può ricevere istruzioni tramite comandi. Nella forma in cui viene consegnato dai distributori esso non è programmato per compiere alcuna operazione. Vi sono aziende, note anche come *integratori di sistemi*, il cui lavoro consiste nel ricevere commesse dal cliente, acquistare robot, programmarli e gestire l'ambiente attorno ad essi in modo da consegnare un sistema pronto all'utilizzo. La programmazione del manipolatore avviene tramite software e linguaggi proprietari, sviluppati e distribuiti dall'azienda che li produce. Non esiste un linguaggio generico per la programmazione, ciascun produttore di manipolatori industriali implementa un codice proprietario necessario per impartire i comandi. Il manipolatore viene consegnato completo di software e manuali per la programmazione ed è frequente che il cliente abbia occasione di assistere ad una lezione introduttiva alla relativa programmazione.

Vi sono notevoli differenze fra codici sviluppati da aziende concorrenti, nonostante questi consentano di fatto di realizzare le medesime operazioni. Con-

siderandoli da un punto di vista prettamente generale, differenti codici di programmazione sono simili tra loro: sono linguaggi che consentono di movimentare il robot, gestire segnali di input e output, comunicare con altri dispositivi, gestire processi paralleli. Nonostante ciò, ciascun codice proprietario è caratterizzato da una sintassi propria, differenti nomi di comandi ma, soprattutto, possibilità e limiti di utilizzo notevolmente diversi. La stesura di un codice per realizzare un desiderato movimento può risultare assai più semplificata utilizzando un codice piuttosto che un altro. Il codice che ne risulta penalizzato potrebbe invece semplificare la questione in altri campi di utilizzo, quali la gestione di segnali I/O o la comunicazione con altri dispositivi. Il linguaggio di programmazione può essere uno dei criteri di scelta fra marche differenti di robot. In appendice B sono riportate a titolo di esempio le differenze fra tre linguaggi robot per quanto riguarda la gestione di locazioni e la movimentazione.

Poichè è stato utilizzato un manipolatore prodotto dalla Epson Seiko, la programmazione avviene utilizzando il linguaggio *SPEL+*. Il codice necessario per la gestione della cella in esame potrebbe essere scritto interamente in tale linguaggio, tuttavia in questo caso, la movimentazione del manipolatore è stata gestita utilizzando in parte anche Matlab. Il controller e Matlab, posti in comunicazione fra di loro, consentono un'indubbia flessibilità di utilizzo. Parte del codice è elaborato dal controller e parte è gestito da Matlab. Ho contribuito alla realizzazione di questo codice di gestione cella durante i primi mesi di permanenza nel laboratorio di robotica e, vista la sua flessibilità, ho optato per utilizzarlo anche per effettuare le prove del presente progetto.

Il codice in linguaggio robot prevede una totale flessibilità di utilizzo in quanto non vi è alcun riferimento al compito da svolgere. Esso è interpretato dal controller e prevede un ciclo continuo di lettura sulla porta di comunicazione Ethernet e l'interpretazione dell'eventuale pacchetto di dati presente in essa. Le indicazioni presenti in tali dati consentono di indicare un qualsiasi compito al manipolatore dall'esterno, utilizzando per la gestione delle operazioni un software più flessibile. La sequenza di pacchetti e il loro contenuto sono infatti gestiti tramite Matlab. L'analisi di una serie di pacchetti di dati forniti dall'esterno e lo svolgimento di azioni in base ai valori forniti in tali pacchetti consente di svincolare dal controller

la gestione della successione di operazioni. La gestione della cella viene in questo modo relegata a Matlab. Ciascun pacchetto contiene una serie di dati intervallati da separatori: il primo valore (*header*) identifica la tipologia di comando richiesto e i successivi valori indicano relative proprietà di quel comando. Ad esempio, un eventuale pacchetto di dati 34/2 indica il cambio di terna utensile (34) e la scelta dell'utensile 2. L'associazione fra header del pacchetto e tipologia di comando è stabilita tramite un protocollo interno al laboratorio. Nei casi più complessi, come quelli di comandi di movimento, ciascun pacchetto dati contiene numerose decine di valori separati, in quanto è necessario indicare la destinazione, le modalità di moto, la configurazione della posa e così via. Per la realizzazione del suddetto programma si è ricorso all'utilizzo di strumenti avanzati offerti da *SPEL+* quali *background tasks* e *parallel statements*. La descrizione di questi aspetti è esposta in appendice C.

5.3.1 Comunicazione con Matlab

La comunicazione con un dispositivo esterno è fondamentale per utilizzare un manipolatore al massimo delle sue capacità. I robot prodotti dalla ditta Epson consentono di stabilire comunicazioni con un dispositivo esterno tramite porta Ethernet e porta seriale EIA RS-232. La comunicazione tramite porta Ethernet consente il trasferimento di dati ad una velocità maggiore, la connessione via cavo seriale RS-232 è tuttavia implementata da un numero maggiore di dispositivi. L'invio e la ricezione di informazioni dal controller può essere effettuato anche utilizzando entrambe le porte di comunicazione se il dispositivo esterno, nel caso in esame un computer, dispone di entrambe le uscite. Un possibile vantaggio derivante dall'utilizzo di due connessioni contemporanee è la possibilità di selezionare e instradare i dati a seconda delle informazioni in essi contenuti.

Tale metodo è quello utilizzato nel codice implementato per la gestione della cella. Tramite porta RS232 vengono inviate informazioni secondarie che non richiedono la massima velocità di trasferimento, mentre il ciclo principale di lavoro del robot viene gestito dalla connessione ethernet, più prestante. In questo modo è possibile, ad esempio, ricevere informazioni sui banchi di bit di input/output senza intaccare la tempistica di comunicazione dei comandi del ciclo principale.

L'implementazione della doppia comunicazione in *SPEL+* richiede attenzione alla gestione dei task multipli. Non è infatti possibile che due task abbiano il permesso di comunicare su una medesima porta nonostante la richiesta avvenga in istanti distinti. In altre parole, aperta una porta di comunicazione, qualunque sia la sua tipologia, non è consentita la sua utilizzazione da parte di un processo parallelo. Se il task che ha effettuato l'inizializzazione della comunicazione esegue un processo tramite comando *CALL*, sospendendo la propria attività in attesa del termine di quest'ultimo, la gestione della comunicazione viene assegnata al nuovo task. Se invece l'esecuzione del processo avviene tramite comando *XQT*, ovvero in parallelo, la gestione della comunicazione non subisce modifiche. Queste problematiche possono complicare la stesura del codice se è prevista la comunicazione con il dispositivo esterno da più processi paralleli. È indispensabile, in questo caso, ricorrere all'utilizzo di variabili globali o flag per indicare al task gestore della comunicazione la necessità di inviare dati e le informazioni da trasferire.

Tempistiche di comunicazione

Nelle tabelle 5.1 e 5.2 sono presentati i risultati di un test sperimentale effettuato con lo scopo di valutare le tempistiche di comunicazione fra Matlab e Epson RC+ utilizzando le due differenti comunicazioni disponibili. Il test consiste in:

1. Invio di un comando di movimento da Matlab tramite porta ethernet;
2. Invio di una conferma di avvenuta ricezione da parte del controller;
3. Esecuzione del comando di movimento;
4. Invio della conferma del termine movimento.

Le conferme 2 e 4 sono state inviate utilizzando differenti metodologie di comunicazione. I valori misurati in Matlab sono infine confrontati con le tempistiche richieste per la semplice esecuzione del comando di movimento tramite *SPEL+* senza comunicazione con l'esterno (*target*).

Si può osservare come la modalità di conferma intermedia non influisca sensibilmente sulle tempistiche di comunicazione, in quanto effettuata in parallelo

| Conferma 2 | Conferma 4 | Tempo misurato [s] | StDev [s] | Target [s] | Δt [ms] |
|-------------|------------|--------------------|-----------|------------|-----------------|
| Ethernet | Ethernet | 0,736 | 0,003 | 0,729 | 6,9 |
| RS-232 | Ethernet | 0,736 | 0,004 | 0,729 | 7,1 |
| Non inviato | Ethernet | 0,736 | 0,003 | 0,729 | 6,7 |
| RS-232 | RS-232 | 0,746 | 0,008 | 0,729 | 17,0 |
| Non inviato | RS-232 | 0,751 | 0,009 | 0,729 | 22,2 |

Tabella 5.1: Risultati del test per valutare i tempi richiesti dalla comunicazione fra controller e Matlab. Velocità impostata pari al 50%.

| Conferma 2 | Conferma 4 | Tempo misurato [s] | StDev [s] | Target [s] | Δt [ms] |
|-------------|------------|--------------------|-----------|------------|-----------------|
| Ethernet | Ethernet | 0,500 | 0,006 | 0,496 | 4,9 |
| RS-232 | Ethernet | 0,501 | 0,007 | 0,496 | 6,0 |
| Non inviato | Ethernet | 0,501 | 0,007 | 0,496 | 5,7 |
| RS-232 | RS-232 | 0,516 | 0,008 | 0,496 | 20,2 |
| Non inviato | RS-232 | 0,514 | 0,009 | 0,496 | 18,6 |

Tabella 5.2: Risultati del test per valutare i tempi richiesti dalla comunicazione fra controller e Matlab. Velocità impostata pari al 100%.

rispetto al movimento. Si può apprezzare, inoltre, la maggiore velocità di trasmissione dati della porta ethernet, stimabile in 3 – 3.5 ms per comunicazione, in quanto i tempi del test considerano una doppia comunicazione. La porta seriale richiede invece un tempo dell'ordine di grandezza di 15 – 20 ms per singola comunicazione.

5.3.2 Segnali I/O

I segnali di Input e Output consentono di gestire i dispositivi presenti all'interno della cella, tra cui l'apertura e chiusura della pinza solidale alla flangia robot. Gestire gli I/O tramite controller equivale a disporre di un PLC, limitato nel numero di segnali, gestito tramite linguaggio robot anzichè linguaggio PLC. Il controller è infatti dotato di una serie di banchi di input e output: con il termine banco si intende un blocco di 8 ingressi o uscite collegati allo stesso riferimento di tensione comune. Il controller Epson RC700 in versione base monta tre banchi di input e due banchi di output, per eventuali esigenze è possibile montare schede aggiuntive. In questo caso non è stato necessario.

La gestione degli I/O tramite programmazione viene effettuata considerando come un interruttore ciascun bit del banco, potendo alzare o abbassare il segnale o richiederne lo stato. Il codice Epson *SPEL+*, tuttavia, fornisce anche metodi di gestione degli I/O a banchi, ovvero a gruppi di otto bit, così come è osservabile fisicamente nel quadro elettrico della cella robotizzata. Al programmatore vengono messi a disposizione i comandi *IN* e *OUT* che consentono di leggere o, nel solo caso output impostare, un intero banco di I/O anzichè il singolo bit. Per poter operare su interi banchi di bit è indispensabile un'accurata scelta dei collegamenti con i dispositivi nella relativa morsettiera. Risulta inappropriato, ad esempio, collegare parte degli I/O di un dispositivo in un banco e la restante parte in un altro.

Si supponga di voler monitorare un banco di bit di Input e comunicare ad un dispositivo esterno qualsiasi cambiamento di stato nel banco in osservazione. Con la gestione di interi banchi di bit tale operazione risulta semplificata. L'esempio in figura 5.6 illustra uno script per effettuare la ricezione sulla porta 201 del valore decimale rappresentante la situazione del primo banco di Input (bit 0-7).

Figura 5.6: Esempio inerente la gestione della comunicazione di segnali nel codice utilizzato in laboratorio.

```
temp = In(0)
Do While True
    Wait In(0) <> temp
    temp = In(0)
    Print #201, Str$(temp)
Loop
```

Se ciò non dovesse essere sufficiente, *SPEL+* offre uno strumento ancora più versatile per la gestione degli I/O: la lettura o scrittura di due banchi con un solo comando (*InW* e *OutW*).

Memory I/O

Non è di secondaria importanza la possibilità di utilizzare I/O software, denominati Memory I/O. Si tratta di *built-in memory bits* che consentono una facile comunicazione fra task. Si consideri il caso in cui un processo debba segnalare un'informazione binaria ad un altro task o in un'altra porzione di codice dello stesso task. Tale situazione in gergo informatico è nota come *flag*. Una comune soluzione, adottabile in qualsiasi linguaggio di programmazione, consiste nel dichiarare una variabile globale, inizializzarla al valore nullo e all'occorrenza imporre un differente valore. Tale operazione è poco elegante e aggiunge inutilmente una variabile globale al progetto.

In *SPEL+* i banchi di Memory I/O consentono di evitare le operazioni precedentemente descritte: l'utente può semplicemente modificare il valore di un bit fittizio tra i 256 a disposizione e controllarne lo stato quando necessario. Questi segnali non sono connessi con alcun dispositivo esterno e si comportano contemporaneamente come segnali sia di input sia di output.

Remote input

Gli input possono essere utilizzati per gestire le operazioni dello stesso controller. Con il termine *controllo remoto* si intende l'operazione di utilizzare segnali esterni per impartire comandi al manipolatore, siano essi input fisici o informazioni ricevute tramite una porta di comunicazione attiva. Le funzioni ottenibili tramite controllo remoto possono essere, per citarne alcune, l'avvio del progetto o lo spegnimento dei motori. In questo modo è possibile evitare che l'operatore debba utilizzare il computer per avviare il programma di gestione del manipolatore. È sufficiente disporre a lato della cella un interruttore fisico connesso ad uno dei bit di input del controller e configurare il programma Epson RC+ in modo che l'avvio del progetto avvenga con l'utilizzo dello switch. Nel codice utilizzato in laboratorio si è fatto uso di remote input per l'inizializzazione del progetto.

5.4 Cella di lavoro

In figura 5.7 è riportato lo schema complessivo dell'apparato sperimentale adottato. Il robot è connesso al controller, il quale comunica con il computer tramite due differenti porte di comunicazione. I connettori del controller, tra cui quello inerente i segnali di emergenza e i banchi I/O, sono connessi alle relative morsettiere nel quadro elettrico. L'alimentatore a 24V è inserito nel quadro elettrico, così come i relè temporizzati. Nelle figure 5.8 e 5.9 sono riportate le foto dell'allestimento complessivo con cui sono stati realizzati i test.

5.5 Accuratezza dei movimenti

Il comando *FINE*, nei robot Epson, viene utilizzato per indicare l'accuratezza con cui è richiesto il posizionamento dei motori al termine di un determinato movimento. I valori indicati nel comando corrispondono al numero di tacche encoder entro cui il movimento viene considerato concluso dal sistema di controllo. È possibile specificare un valore differente per ciascun giunto: la destinazione è considerata raggiunta quando ciascun giunto è posizionato nel range ad esso specificato. Solo in seguito a tale verifica viene processato il comando successivo

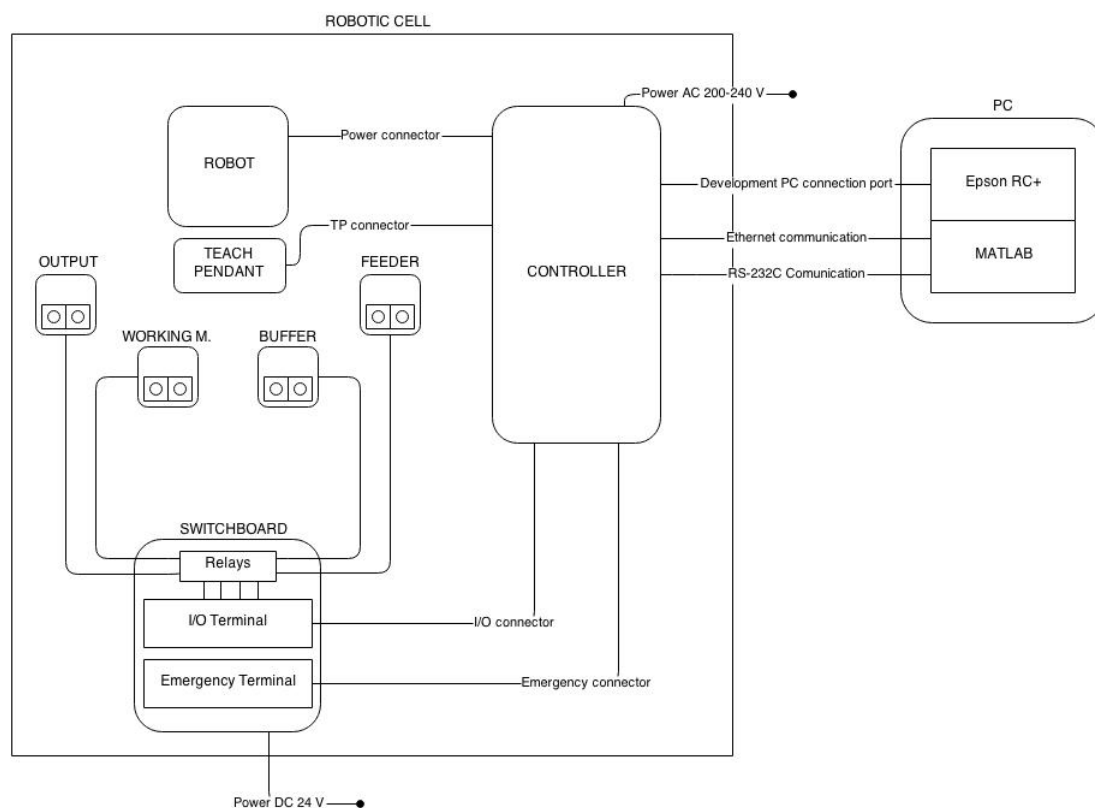


Figura 5.7: Schema dell'apparato sperimentale utilizzato per effettuare i test.

di movimento. Il valore di default per il manipolatore Epson C4 è pari a 10000. Poiché l'entità del parametro *fine* ha notevole importanza nella tempistica di movimentazione del robot, si specifica che le prove sono state condotte mantenendo il valore di default.

Per giustificare l'importanza di tale parametro si riportano in figura 5.10 i risultati di un test effettuato per valutarne l'influenza sulle tempistiche di movimento. Si è scelto un movimento ed è stato fatto compiere al robot variando progressivamente la velocità imposta e l'accuratezza richiesta, mantenendo il valore specificato uguale per tutti gli assi.

Come si può osservare, valori elevati del parametro *fine* consentono di ridurre notevolmente i tempi di movimentazione, a scapito tuttavia dell'accuratezza di posizionamento.

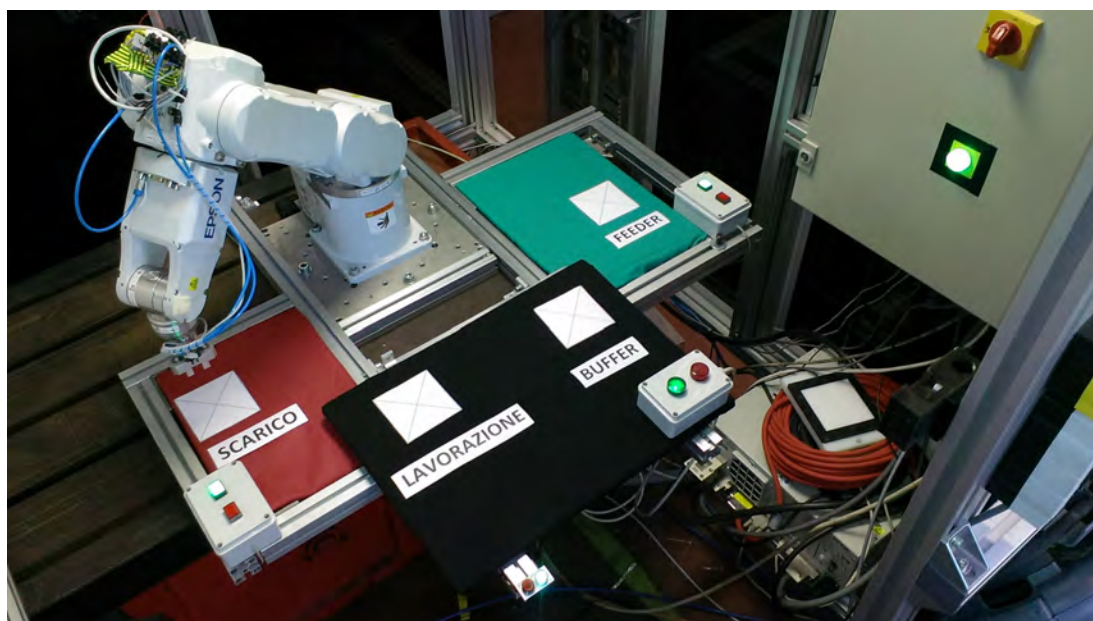


Figura 5.8: Foto dell'interno della cella di lavoro allestita.



Figura 5.9: Foto dell'esterno della cella di lavoro allestita.

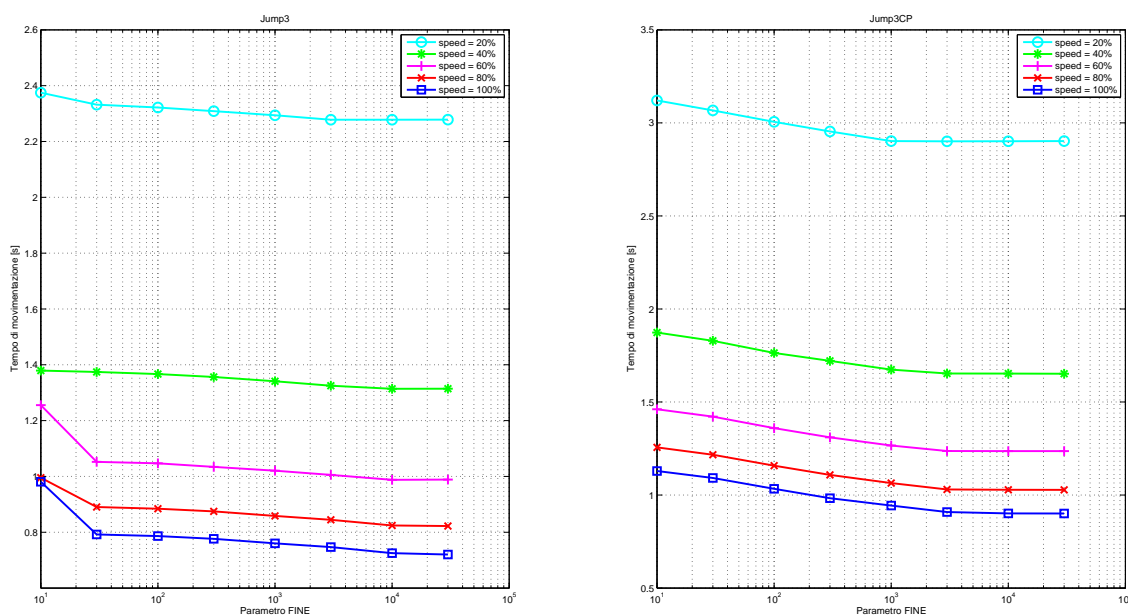


Figura 5.10: Risultati del test condotto per valutare l'influenza del parametro di accuratezza sulle tempistiche di movimentazione. A sinistra un movimento di tipo Jump3, a destra Jump3CP (Appendice C)

5.6 Prove sperimentali

5.6.1 Impostazione delle prove

Ciascuna prova è stata realizzata settando come desiderato i relè temporizzati e simulando un numero di rinnovi feeder successivi consono al tipo di prova. Vista l'assunzione di postazioni puntiformi non sono stati movimentati componenti: il robot raggiunge le locazioni, apre o chiude la pinza ma non afferra o rilascia alcun componente. La movimentazione di componenti reali risulterebbe infatti in contrasto con l'assunzione di associare una sola locazione alle postazioni che possono accogliere più di un pezzo.

Non è stato definito un tempo prefissato per la realizzazione di ciascuna prova e nemmeno un numero di rinnovi feeder. Stabilire uno di questi due criteri per decretare il termine di qualsiasi prova risulterebbe inadeguati. A seconda dei parametri in gioco, infatti, essi possono non essere appropriati. Se il numero di

componenti nel feeder è elevato, è richiesto un tempo complessivo della prova superiore per poter ottenere una corretta misurazione del tempo ciclo. Viceversa, se la dimensione del buffer è notevole, si avrà un transitorio iniziale più lungo e il criterio di numero di vassoi prefissato non risulta corretto. Si è optato pertanto arbitrariamente per la decisione di concludere una prova rispettando il seguente criterio: stimato ad un dato istante, con prova ancora in corso, il tempo ciclo corrispondente, essa si considera conclusa se la lavorazione dei successivi N_F componenti comporta una variazione della stima inferiore al centesimo di secondo. A livello pratico tale metodo è stato applicato avvalendosi di un'interfaccia grafica, così da poter visionare costantemente il valore in questione senza attendere il termine della prova.

Si è parlato di stima poichè la semplice media di tutti i tempi rilevati considera anche il transitorio iniziale e un numero di lavorazioni non necessariamente multiple di N_F , come a rigore dovrebbe essere. A prova terminata, infatti, dal listato di tempi rilevati vengono eliminati i tempi precedenti al raggiungimento della situazione a regime. Dal listato sono inoltre esclusi gli ultimi tempi rilevati eventualmente in eccesso in modo da ottenere un numero di dati multiplo di N_F . Tali accortezze, idealmente, non sarebbero necessarie per una prova di durata infinita.

5.6.2 Problematiche riscontrate

La principale difficoltà incontrata nella realizzazione delle prove sperimentali consiste nella regolazione dei relè temporizzati. La regolazione dei relè, secondo il relativo manuale, avviene in maniera discreta. Nel modello TU93 sono implementati segnali luminosi per indicare all'utente quale sia il settaggio scelto. Tuttavia, a parità di segnali luminosi, e quindi idealmente della regolazione del relè, una minima variazione di settaggio modifica le tempistiche del dispositivo. In altre parole una generica regolazione discreta è stata effettivamente riscontrata, ma questa consente comunque ampio margine di variazione. La problematica derivante da ciò consiste nella difficoltà di effettuare prove diverse con la stessa tempistica di un relè se fra queste è necessario modificarne il settaggio. Questa situazione si verifica, ad esempio, durante una campagna di prove in cui si vuole misurare il

tempo ciclo per tutte le combinazioni possibili fra n valori di τ_L e m valori di τ_F ($n, m > 1$).

La tempistica di un relè varia inoltre sensibilmente a seconda delle condizioni ambientali e della tensione nel quadro elettrico. A parità di settaggio sono state misurate tempistiche notevolmente differenti ($\pm 6\%$) a distanza di 10 ore fra due misurazioni. Per i motivi esposti ciascuna serie di prove è stata preceduta e seguita da una rilevazione di 3 valori delle tempistiche dei relè.

A causa dei problemi citati la realizzazione delle prove ha richiesto più tempo del previsto. Nel caso di proseguimento del presente studio si consiglia una diversa gestione delle tempistiche delle postazioni con dispositivi più idonei. La gestione tramite software anzichè relè fisici potrebbe risultare un'ottima scelta.

5.7 Analisi dell'incertezza

Per ottenere una corretta rappresentazione dei dati sperimentali e valutare la compatibilità fra risultati sperimentali e modelli analitici, è indispensabile valutare l'incertezza dei risultati delle misurazioni. La principale fonte di incertezza, come appena accennato, è determinata dai relè temporizzati usati per simulare le tempistiche delle differenti postazioni.

5.7.1 Fonti di incertezza

Le fonti di incertezza considerate sono:

1. Risoluzione di rilevazione dei singoli tempi tramite software Matlab;
2. Incertezza dei relè utilizzati nella stazione di scarico e lavorazione;
3. Incertezza del relè temporizzato utilizzato per simulare il rinnovo della stazione di feeder;
4. Errori di sequenziamento dovuti alla fluttuazione dei valori dei relè;
5. Incertezza dovuta alla ripetibilità di ciascun movimento del robot;
6. Incertezza nella comunicazione fra Matlab e il controller;

7. Propagazione degli errori sui singoli tempi rilevati per calcolare il tempo ciclo.

Vengono ora analizzati uno ad uno gli aspetti elencati fino a giungere alla formula utilizzata per determinare l'incertezza sul tempo ciclo calcolato.

Risoluzione di rilevazione

Il software Matlab rende disponibile all'utente i comandi *tic* e *toc* grazie ai quali è possibile effettuare misure di tempo assoluto. Il comando *tic* indica l'istante di inizio del conteggio temporale e il duale comando *toc* esprime l'istante in cui si desidera effettuare la misura. Nelle versioni di Matlab precedenti alla R2006 [26] i comandi *tic* e *toc* erano connessi alla funzione *clock*, sfruttavano cioè il wall clock interno del sistema. Ogni comando *tic* o *toc* prelevava il tempo istantaneo riportato in tale metodo di conteggio effettuato dal sistema operativo. Matlab, in seguito, valutava la differenza fra i due valori rilevati per fornire una stima della misura richiesta. Analizzando il caso in Windows, la funzione *clock* chiama il servizio di sistema *GetLocalTime* usando l'API (Application Programming Interface). Il servizio in esame, tuttavia, ha scarsa accuratezza e un richiamo immediatamente successivo dello stesso può addirittura determinare un valore negativo del tempo trascorso. Dalla versione R2006b di Matlab è stato modificato il sistema di misura delle tempistiche. Il servizio di sistema attualmente usato per effettuare misure di tempo è *QueryPerformanceCounter* che permette di avviare un counter a 64-bit in unità di sistema anziché secondi. La differenza fra le due rilevazioni viene poi divisa per la frequenza restituita dal servizio di sistema *QueryPerformanceFrequency* grazie al quale si può giungere al risultato in secondi. Per osservare l'attuale risoluzione dei comandi *tic* e *toc* in Matlab è stato effettuato il seguente test nella command window:

```
> tic;
> toc

Elapsed time is 0.000001 seconds.
```

In definitiva l'incertezza dovuta allo strumento di rilevazione, ovvero il software Matlab, di ciascun tempo si considera trascurabile.

Dalla versione R2008b in poi la sintassi *tic-toc* inoltre è stata estesa per consentire l'avvio di contatori multipli. La nuova sintassi consente di salvare un preciso contatore in una variabile per poi richiamare tale parametro in una successiva funzione *toc*. Tale metodo è stato ampiamente utilizzato nel codice implementato per la gestione della cella di lavoro.

Incertezza associata ai relè

Ad ogni rilevazione effettuata si misura un ciclo di lavoro che comprende necessariamente una lavorazione e uno scarico del pezzo. Rispetto al valore dichiarato, o ,più precisamente, rispetto al valore medio misurato, i relè utilizzati presentano una ripetibilità tutt'altro che trascurabile. I relè sono costituiti da circuiti elettrici e di conseguenza le prestazioni sono influenzate dalle condizioni ambientali e sono suscettibili alle variazioni di temperatura dovute al passaggio di corrente. Non di minor importanza vi è l'aspetto della tensione fornita ai relè: è stato osservato che l'utilizzo del robot influisce in maniera misurabile sulla tensione fornita dagli alimentatori che sono connessi ai relè. I relè utilizzati per le stazioni di lavorazione e scarico sono entrambi dei modelli TU93 prodotti dalla CdcElettromeccanica. Nel manuale del prodotto è riportata una ripetibilità pari all'1% del fondo scala selezionato. Il prodotto consente infatti di regolare non solo il comportamento desiderato a livello di gestione dei segnali ma anche la temporizzazione degli stessi. Due sono le regolazioni consentite per quanto riguarda la temporizzazione: la scelta del fondo scala e l'impostazione di una percentuale del fondo scala selezionato. Nella gran parte delle prove sperimentali effettuate il fondo scala settato per questi relè è pari a 1 o 5 s. Di conseguenza l'incertezza rispetto al valore medio dichiarata dal produttore è pari a 1 o 5 centesimi di secondo. Tali valori sono stati verificati sperimentalmente tramite prove con un numero di misurazioni dell'ordine del migliaio utilizzando un programma scritto in *SPEL+* appositamente per testare i relè senza movimentare il robot.

In figura 5.11 è riportato parte del codice utilizzato, dove con *j* è indicato il numero del relè in esame e *#30* è il numero associato al file di testo su cui si desidera salvare i risultati. Si considerano pertanto verificate le indicazioni del produttore.

```
For i = 1 To n_prove
    ResetElapsedTime
    On outputsig(j)
    Wait Sw(inputsig(j)) = 0
    Wait Sw(inputsig(j)) = 1
    tempo = ElapsedTime
    Print #30, tempo, Chr$(10)
    Off outputsig(j)
    Print tempo
    Wait 0.3
Next
```

Figura 5.11: Esatratto di codice utilizzato per testare i tempi del relè.

Verrà in seguito definito con δt_{LS} o δt_1 l'incertezza relativa ai relè della postazione di lavorazione e scarico. Ciascun tempo rilevato è affetto da un'incertezza δt_{LS} in quanto il fenomeno misurato prevede la dipendenza lineare da uno solo dei due relè in esame. Se la stazione di lavorazione detta il ritmo alla cella l'incertezza sul relè di scarico è ininfluenza e viceversa. Fra i test effettuati non sono presenti situazioni in cui non è stato possibile determinare quali dei due relè fosse quello che influenza il tempo del ciclo, ovvero non ci sono casi in cui vi è alternanza fra l'influenza di uno o l'altro dei relè fra componenti successivi.

Ogni test prevede la lavorazione di un numero finito di vassoi disponibili sul feeder. In altre parole il numero totale di rilevazioni per ogni test è un multiplo del numero di pezzi presenti nel feeder. Il tempo di rinnovo del feeder pertanto ha influenza solamente su una rilevazione ogni N_F . Il relè utilizzato per simulare la postazione di ingresso pezzi è in modello M1MFT della Broyce Control. Nel manuale del prodotto è riportata una ripetibilità pari allo 0.5% del tempo impostato. Viene indicata con δt_F l'incertezza relativa al relè dedicato alla postazione di feeder. Tale incertezza sarà tanto meno influente sul tempo ciclo finale misurato tanto maggiore è la dimensione di un vassoio di pezzi. La rilevazione in esame

presenterà un'incertezza che considera sia δt_{LS} sia δt_F , stimabile come segue [27]:

$$\delta t_2 = \sqrt{\delta t_{LS}^2 + \delta t_F^2} \quad (5.2)$$

Errori di sequenziamento

Gli errori di sequenziamento dovuti alla variazione dei tempi dei relè fra un ciclo e l'altro riguardano una differente gestione della sequenza di movimenti del robot dovuti all'anticipo o ritardo del segnale fra un ciclo e l'altro. La gestione di tali errori risulta complicata e si è preferito utilizzare l'osservazione del movimento del robot per valutare se ci fosse differenza di sequenziamento fra un ciclo e il successivo dovuto a questo problema. Di conseguenza questi errori non vengono considerati nel calcolo dell'incertezza in quanto i cicli non coerenti con il resto del test non sono stati considerati. Per il risultato finale tali cicli sono comunque stati rilevati in numero nettamente inferiore rispetto ai cicli totali misurati.

Ripetibilità dei movimenti

Il controller Epson RC700 e il manipolatore C4 utilizzati garantiscono un'ottima ripetibilità dei tempi di movimento del robot. Come verrà esposto in sezione 6.1.1, il manipolatore è in grado di compiere un determinato movimento con una ripetibilità di 1-2 ms. Visti i contributi all'incertezza dovuti ai relè dell'ordine di grandezza di 5 centesimi di secondo, il numero di movimenti robot per ciascun ciclo è tale da influenzare la sola seconda cifra significativa dell'incertezza finale. Si è pertanto preferito proseguire la trattazione trascurando questo contributo per semplicità.

Comunicazione con Matlab

Come è stato presentato nelle tabelle 5.1 e 5.2 la deviazione standard delle prove sulla comunicazione è dell'ordine dei millesimi di secondo. Considerando i valori dell'incertezza dovuta ai relè si è preferito trascurare questo contributo.

Propagazione degli errori

Il tempo ciclo della cella deriva dalla media di tutti i tempi rilevati, tuttavia si tenga presente che ciascun tempo rilevato corrisponde ad un ciclo differente del robot. Non si rileva pertanto una serie di valori che rappresentano lo stesso misurando. Note le incertezze sulle singole misure derivanti dalle considerazioni fin qui esposte è possibile determinare l'incertezza finale sul valore del tempo ciclo calcolato tramite la legge di propagazione delle singole incertezze.

$$\begin{aligned}\delta t &= \frac{\sum_{i=1}^N \delta T_i}{N} = \frac{N_{cicli}(N_F - 1)\delta t_1 + N_{cicli}\delta t_2}{N_{cicli}N_F} = \frac{(N_F - 1)\delta t_1 + \delta t_2}{N_F} \\ &= \frac{\delta t_2 - \delta t_1}{N_F} + \delta t_1 = \delta t_{LS} + \left[\frac{\sqrt{\delta t_{LS}^2 + \delta t_F^2} - \delta t_{LS}}{N_F} \right]\end{aligned}\quad (5.3)$$

Si precisa che alcuni grafici presentati nei prossimi capitoli vengono riportati con il tempo ciclo diviso il tempo di lavorazione sulle ordinate, anzichè il solo tempo ciclo. Tale scelta ha lo scopo di semplificare la consultazione ed è equivalente a scalare l'intero grafico di un fattore costante. Questa modalità di visualizzazione corrisponde alla consultazione della saturazione del macchinario di lavorazione anzichè al tempo ciclo assoluto della cella e semplifica la valutazione dell'ottimizzazione fra più situazioni. La divisione per il tempo di lavorazione non è considerata nell'analisi dell'incertezza per le ragioni esposte. Il risultato della misura è infatti il tempo ciclo e tutti i risultati vengono scalati. Anche i risultati sperimentali vengono ridimensionati di una costante pari al valore di τ_L usato nei modelli teorici se non diversamente specificato.

Capitolo 6

Analisi dei risultati

In questo capitolo si analizzano i risultati di una campagna di prove volta alla determinazione della compatibilità sperimentale con il modello teorico. In sezione 6.1 vengono analizzate le prove effettuate per giustificare le ipotesi iniziali assunte nella stesura del modello teorico per i criteri di movimentazione. In sezione 6.2 invece si confrontano alcune curve scelte a campione in cui viene analizzata l'influenza sul tempo ciclo di ciascun parametro, mantenendo costanti i restanti.

6.1 Validazione ipotesi iniziali

Le ipotesi avanzate in precedenza che richiedono una verifica sperimentale riguardano le caratteristiche dello spostamento fra postazioni. In sezione 2.1.3 sono state esposte le possibili assunzioni semplificative per il modello ed è stata adottata la soluzione additiva considerando la quota di accelerazione e decelerazione richiesta per effettuare la sosta in una postazione. In sezione 2.3 si è supposto inoltre che ciascuna postazione possa essere considerata come entità puntiforme, trascurando la distribuzione spaziale dei componenti. Queste due assunzioni vengono ora giustificate sperimentalmente per dimostrarne l'attendibilità fisica.

6.1.1 Movimenti additivi

L'equidistanza fra postazioni su una circonferenza non verifica necessariamente la supposizione di tempistiche costanti per movimenti fra stazioni adiacenti. No-

nostante esse siano equidistanti nello spazio le tempistiche richieste per ciascun moto dipendono dalla loro locazione nello spazio di lavoro del robot. Nel modello teorico si è considerato il criterio additivo seguente:

$$d(M_i, M_j) = \sum_{k=\min(i,j)+1}^{\max(i,j)} \delta_k - (|i-j|-1)\eta, 0 \leq i, j \leq m+1 \quad (6.1)$$

ricordando $\delta_k = d(M_{i-1}, M_i)$ e supponendo che il parametro η sia costante e indipendente dalla postazione in cui non viene effettuata la sosta. Inoltre esso è stato supposto indipendente dal numero di postazioni intermedie fra quelle di partenza e di arrivo.

La verifica sperimentale è stata effettuata tramite la misura del tempo necessario per spostare il componente fra una coppia di stazioni. Si indichi con δ_{ij} il tempo richiesto per la movimentazione fra la postazione i e j e si utilizzino i pedici F, B, L e S rispettivamente per feeder, buffer, lavorazione e scarico. Sono state effettuate le misure dei tempi δ_{FB} , δ_{BL} , δ_{LS} , δ_{FL} , δ_{BS} e δ_{FS} per determinare l'attendibilità delle assunzioni in esame. Utilizzando le formule che verranno ora esposte, sono stati ottenuti i risultati riportati in tabella 6.1.

I valori δ_{FB} , δ_{BL} e δ_{LS} rappresentano, senza necessità di manipolazione, tre distinte tempistiche fra coppie di stazioni adiacenti. Con $\bar{\delta}$ si rappresenti la media fra i tre distinti tempi.

$$\bar{\delta} = \frac{\delta_{FB} + \delta_{BL} + \delta_{LS}}{3} \quad (6.2)$$

Verificata la validità dell'assunzione di $\delta = \bar{\delta}$ come valore costante per δ_{FB} , δ_{BL} e δ_{LS} , si può passare a η . Noto $\bar{\delta}$ è possibile determinare per ciascuna tempistica δ_{FL} , δ_{BS} e δ_{FS} il calcolo del relativo parametro η , rispettivamente η_{1A} , η_{1B} e η_2 . Considerando le coppie di postazioni a distanza doppia:

$$\eta_{1A} = 2\bar{\delta} - \delta_{FL} \quad (6.3)$$

$$\eta_{1B} = 2\bar{\delta} - \delta_{BS} \quad (6.4)$$

Analizzando, invece, il movimento fra feeder e scarico:

$$\eta_2 = \frac{3\bar{\delta} - \delta_{FS}}{2} \quad (6.5)$$



Figura 6.1: Rappresentazione dei due layout considerati per i test sperimentali.

Si può infine determinare un valore medio del parametro η considerando i risultati delle equazioni 6.3, 6.4 e 6.5.

$$\bar{\eta} = \frac{\eta_{1A} + \eta_{1B} + \eta_2}{3} \quad (6.6)$$

I risultati di tale analisi ottenuti con la configurazione in semicerchio sono riportati in tabella 6.1. Il medesimo test è stato realizzato per due configurazioni differenziate fra di loro dall'angolo formato fra due postazioni adiacenti: 40° o 20° . Si rimanda alla figura 6.1 per una rappresentazione di tali layout.

| Layout | | δ_{FB} | δ_{BL} | δ_{LS} | $\bar{\delta}$ | η_{1A} | η_{1B} | η_2 | $\bar{\eta}$ |
|------------|------------|---------------|---------------|---------------|----------------|-------------|-------------|----------|--------------|
| 40° | Media [s] | 0.391 | 0.392 | 0.390 | 0.391 | 0.291 | 0.291 | 0.287 | 0.289 |
| | StDev [ms] | 1.6 | 1.1 | 1.6 | 1.8 | 2.4 | 2.2 | 1.9 | 2.4 |
| 20° | Media [s] | 0.331 | 0.332 | 0.328 | 0.330 | 0.273 | 0.272 | 0.270 | 0.271 |
| | StDev [ms] | 1.3 | 1.9 | 1.5 | 1.4 | 2.4 | 2.3 | 2.1 | 2.3 |

Tabella 6.1: Risultati sperimentali del test per valutare l'attendibilità delle ipotesi sulle tempistiche di movimentazione.

Come si può notare l'assunzione del parametro δ costante per movimentazioni fra postazioni adiacenti è compatibile con la realtà sperimentale. La variazione dei tre valori rispetto alla media è inferiore allo 0.5% e si considera, pertanto, trascurabile. Il medesimo ragionamento può essere effettuato per il parametro η . Si può notare in particolare come in entrambi i casi il valore η_2 sia il minore fra i tre considerati. La deviazione standard dei risultati è dell'ordine del millesimo di

secondo e questa caratteristica sarà riscontrabile anche in altri test presentati nei prossimi capitoli. Tali valori indicano la ripetibilità dei tempi di movimentazione del manipolatore, notevolmente inferiore rispetto alle tempistiche di lavoro che verranno impostate per le postazioni.

Un test semplice come questo consente di affermare la plausibilità dell'ipotesi di movimentazione additiva con η indipendente dal numero e tipologia di postazioni coinvolte nel movimento.

6.1.2 Postazioni puntiformi

Le postazioni che possono contenere più di un componente sono il feeder e il buffer. Queste sono precedentemente state supposte puntiformi, cioè si è imposta sempre la medesima locazione di destinazione del manipolatore, indipendentemente dallo stato di occupazione della postazione. Nel reale layout in esame entrambe le postazioni sono costituite da più slot in modo da poter accogliere diversi componenti. Le differenti posizioni dei componenti in ciascuna di esse possono essere casuali o stabilite a priori. Il primo è il caso, ad esempio, di nastri trasportatori con annessi sistemi di visione ed è tipico nella postazione di input. Il secondo caso ben si adatta alla postazione di buffer in cui un sistema di visione è inadeguato a causa della necessità di elaborare immagini con maggiore frequenza.

Si è supposto di considerare come locazione della postazione il solo centro geometrico delle possibili configurazioni assumibili dal componente reale. L'orientazione della posa unica, inoltre, è posta pari alla media di quelle realmente presenti nella postazione.

Il test effettuato per valutare l'attendibilità di queste assunzioni consiste nella misura di movimentazioni fra due postazioni considerando ciascun movimento con punto di partenza e arrivo variabile. Una funzione random assegna la quota x e y al punto di destinazione mantenendo la quota z costante. In questo modo si può simulare la disposizione casuale dei componenti su entrambe le postazioni. Le posizioni raggiungibili sono comprese in un quadrato di lato 2Δ e centrato nel punto di riferimento per il caso puntiforme. In seguito è stato condotto il medesimo test variando anche la componente di orientamento del componente tramite assegnazione random di una rotazione rispetto all'asse z pari a $\pm\psi$.

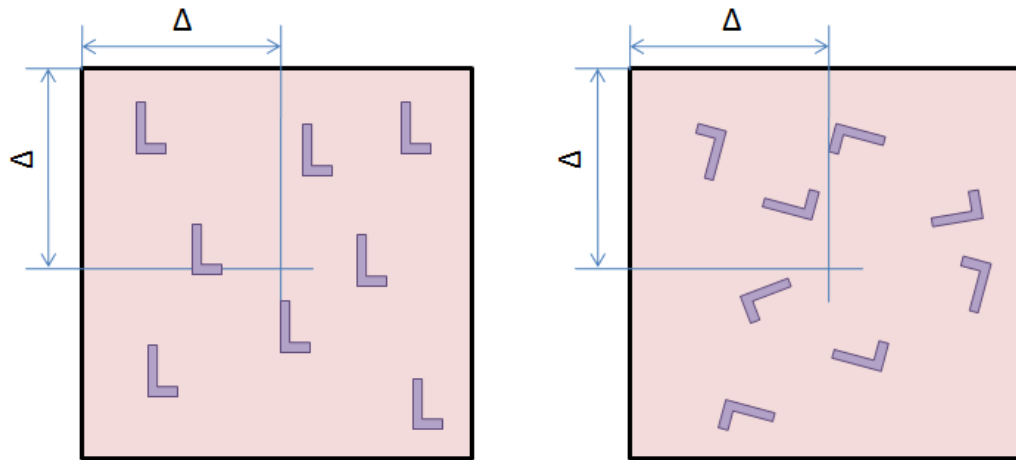


Figura 6.2: Schema della postazione non puntiforme considerata nei test. A sinistra l'orientazione del componente è costante, a destra variabile.

```

randNumu = psi - Int(Rnd(2 * psi))
randNumx = delta - Int(Rnd(2 * delta))
randNumy = delta - Int(Rnd(2 * delta))
P6 = P3 +X(randNumx) +Y(randNumy) :U(randNumu)

```

Figura 6.3: Estratto di codice per la realizzazione del test sulle postazioni puntiformi.

In figura 6.3 si riporta per chiarezza una parte del codice utilizzato per effettuare il test. In figura 6.2 è rappresentata invece la situazione considerata.

Si può osservare dai risultati riportati in tabella 6.3 come, variando la sola posizione dei componenti nella postazione, l'errore sul tempo di movimentazione rispetto all'assunzione di posizione puntiforme sia minore di 0.5%. Aumentando eccessivamente l'area dedicata alla postazione tale valore aumenta oltre il 2% perchè vi è una notevole variazione delle pose assunte dal manipolatore fra i successivi movimenti. Il manipolatore cioè, in alcuni casi, lavora al limite del proprio spazio di lavoro. Tale situazione rappresenta una pessima scelta di progetto della cella poichè il manipolatore dovrebbe evitare, se possibile, di lavorare agli

| | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 |
|-----------------------|--------|--------|--------|--------|--------|
| Δ [mm] | 20 | 40 | 60 | 80 | 100 |
| ψ [°] | 0 | 0 | 0 | 0 | 0 |
| Target [s] | 0,424 | 0,424 | 0,424 | 0,424 | 0,424 |
| Media [s] | 0,424 | 0,426 | 0,425 | 0,426 | 0,433 |
| DevSt [s] | 0,006 | 0,013 | 0,020 | 0,023 | 0,032 |
| (Target-Media)/Target | 0,10% | 0,39% | 0,33% | 0,45% | 2,01% |

Tabella 6.2: Risultati sperimentali per valutare l'attendibilità dell'ipotesi di postazioni puntiformi. Δ compreso fra 20 e 100 mm. L'orientazione dei componenti non varia.

| | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 |
|-----------------------|--------|--------|--------|--------|---------|
| Δ [mm] | 20 | 40 | 60 | 80 | 100 |
| ψ [°] | 45 | 45 | 45 | 45 | 45 |
| Target [s] | 0,424 | 0,424 | 0,424 | 0,424 | 0,424 |
| Media [s] | 0,426 | 0,426 | 0,430 | 0,431 | 0,431 |
| DevSt [s] | 0,009 | 0,014 | 0,018 | 0,023 | 0,029 |
| (Target-Media)/Target | 0,56% | 0,37% | 1,42% | 1,66% | 1,64% |

Tabella 6.3: Risultati sperimentali per valutare l'attendibilità dell'ipotesi di postazioni puntiformi. Δ compreso fra 20 e 100 mm. L'orientazione dei componenti varia fra $\pm 45^\circ$.

estremi di tale spazio. Il continuo cambio di configurazione impone una maggiore rotazione dei giunti e pertanto il tempo medio aumenta.

La variazione dell'orientazione dei componenti, i cui relativi risultati sono riportati in tabella 6.3, aumenta l'errore rispetto all'assunzione puntiforme principalmente a causa della limitazione imposta dal giunto 6. Tale giunto è il responsabile della rotazione lungo l'asse z poichè si assume l'end effector perpendicolare al piano su cui sono situati i componenti. Il motore del giunto 6, costretto ad effettuare ampie rotazioni, risulta quello con il più elevato *tempo di azionamento minimo*. A livello di controllo esso è il tempo rispetto al quale vengono scalate le

leggi di moto degli altri giunti per garantire il medesimo istante di inizio e fine movimento su tutti gli assi.

In definitiva l'assunzione di considerare le stazioni di feeder e buffer puntiformi determina un errore stimabile ad un valore pari a 1-2%. I risultati riportati da questa stima dell'errore, inoltre, riguardano un caso più gravoso di quello tipico della pratica industriale, poichè entrambe le postazioni sono state considerate caratterizzate da una componente random. Nella pratica invece la postazione di buffering è tipicamente una postazione con componenti risposti in slot prestabiliti e con il medesimo orientamento. La distanza fra essi varia in funzione della dimensione del componente movimentato. Solo la postazione di ingresso pezzi presenta una componente random determinata dal sistema di visione.

6.2 Confronto dei risultati con il modello

Un'intensiva campagna di prove è stata condotta per valutare l'attendibilità del modello teorico. Vist l'alto numero dei parametri in esame non si vuole avere la pretesa di dichiarare la completa validazione del modello teorico. La compatibilità dei risultati sperimentali eventualmente convalida il modello per i soli parametri considerati.

6.2.1 Parametri sperimentali

Nei grafici che verranno presentati sono stati sovrapposti i risultati sperimentali con la relativa incertezza alla curva del modello teorico a cui fanno riferimento. Si è preferito rappresentare una sola curva teorica, generata utilizzando i valori medi rilevati sperimentalmente senza considerare le relative incertezze. Tale scelta è dovuta alla non linearità del modello analitico: si potrebbe pensare di ottenere due curve estreme considerando i valori minimi e massimi misurati. Tuttavia non è necessariamente verificata la condizione di tempo ciclo maggiore (minore) utilizzando parametri maggiori (minori). Sarebbe necessario valutare il tempo ciclo per ogni combinazione di valori assumibili per determinare le curve del modello teorico. Per semplificare la rappresentazione si preferisce pertanto rappresentare

la sola curva teorica derivante dai valori medi sperimentali e indicare l'incertezza sui soli dati sperimentali.

Se non diversamente specificato i valori dei principali parametri della cella allestita sono quelli riportati in tabella 6.4. Va specificato che tali valori possono essere determinati anche senza disporre fisicamente di un manipolatore. È possibile valutare in fase preliminare di progetto il valore assunto dalle tempistiche di movimentazione utilizzando simulatori, ovvero software in grado di stimare la risposta dinamica del sistema. Essi sono appositamente forniti dal produttore di robot.

| | Media [s] | StDev [s] |
|----------|-----------|-----------|
| δ | 0,391 | 0.001 |
| η | 0,290 | 0.001 |
| τ_H | 0,194 | 0.001 |

Tabella 6.4: Valori sperimentali utilizzati nel modello teorico.

Per determinarne il valore del parametro ϵ , rappresentante il tempo di presa e rilascio, è stata rilevata la differenza di tempo fra un singolo movimento senza presa del pezzo e il medesimo movimento con afferraggio finale e un movimento successivo. Determinando il parametro tramite il test descritto, esso viene utilizzato per considerare anche le tempistiche di comunicazione fra computer e controller e quelle richieste per effettuare la scelta dell'operazione successiva. Queste sarebbero altrimenti trascurate perchè non previste dal modello. Il parametro ϵ così determinato assume il valore di 0.05 s.

Va precisato che alcune operazioni non prevedono la presa o rilascio del pezzo al termine del movimento. In tali situazioni il modello non prevede il contributo di ϵ pertanto, inglobando in esso le tempistiche di elaborazione dati, queste vengono necessariamente trascurate in alcuni casi.

Se non diversamente specificato il buffer dispone di un numero di slot disponibili superiore al numero di componenti nel feeder. Tale situazione equivale ad avere un buffer con un numero di slot superiore a quello effettivamente utilizzato a regime. L'eventuale differenza nel tempo ciclo utilizzando valori diversi

è determinata dai primi cicli, essendo questi dedicati al riempimento del buffer. Considerando il solo tempo a regime tale differenza non rientra nei risultati.

Il numero di componenti presenti ad ogni rinnovo feeder è sempre considerato costante. L'analisi del caso probabilistico è effettuata solamente in sezione 7.3.

6.2.2 Dimensione del feeder

In figura 6.4 si riportano alcuni grafici con il numero di pezzi nel feeder sulle ascisse. Le linee tratteggiate indicano i cicli limite esposti in sezione 4.2 e rappresentano la situazione limite raggiungibile con i parametri in esame. In questi grafici si può notare l'utilità delle curve limite tratteggiate: il tempo ciclo con priorità assegnata al feeder tende al valore limite per elevati numeri di componenti ad ogni rinnovo del feeder. Tale situazione è infatti comparabile alla presenza di un bowl-feeder nella postazione di ingresso pezzi. Le curve con priorità assegnata al buffer invece tendono alla rispettiva curva limite solo nel caso in cui l'inserimento nel buffer risulti completamente mascherato.

Si può notare come la curva del tempo ciclo senza l'utilizzo del buffer sia decrescente. Questo avviene perchè il tempo di attesa ricambio feeder, sebbene non vari in durata temporale, viene suddiviso su un numero maggiore di pezzi movimentati. Per $N_F \rightarrow \infty$ anche tale curva tenderà al valore limite rappresentato tratteggiato.

Per i valori considerati il ciclo con priorità assegnata al buffer risulta generalmente il più conveniente a livello di tempo ciclo. Tale affermazione non è tuttavia da considerarsi generale poichè con altri parametri potrebbe risultare più difficile la bufferizzazione, operazione imposta per ogni componente nel caso con priorità assegnata al buffer. Un esempio di tale caso è presente nella prima figura per un numero di componenti ridotto.

6.2.3 Tempo di scarico

In figura 6.5 sono riportati alcuni test effettuati per valutare l'influenza del tempo di scarico sul tempo ciclo. Si può notare dall'andamento delle curve l'effettiva presenza di un tempo di scarico limite sotto il quale il suo valore è trascurabile. Tale

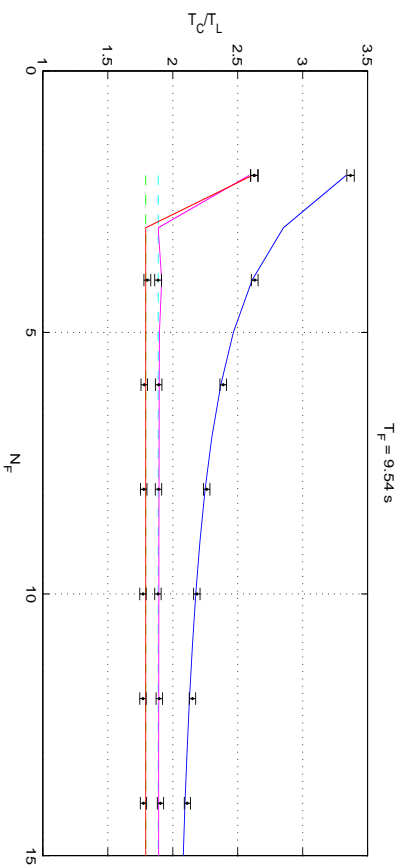
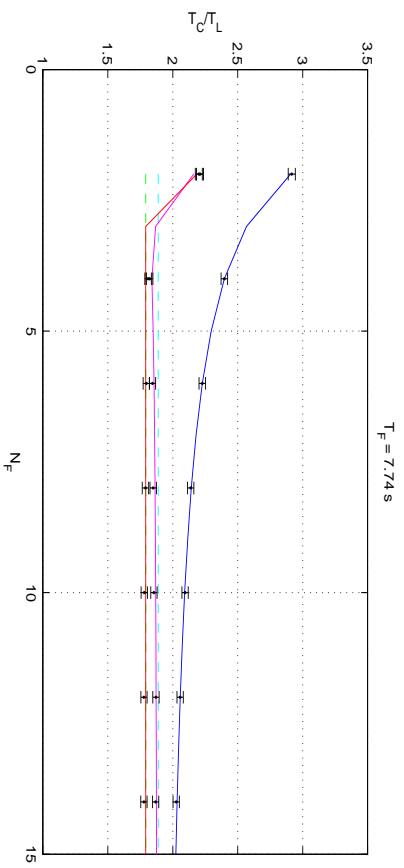
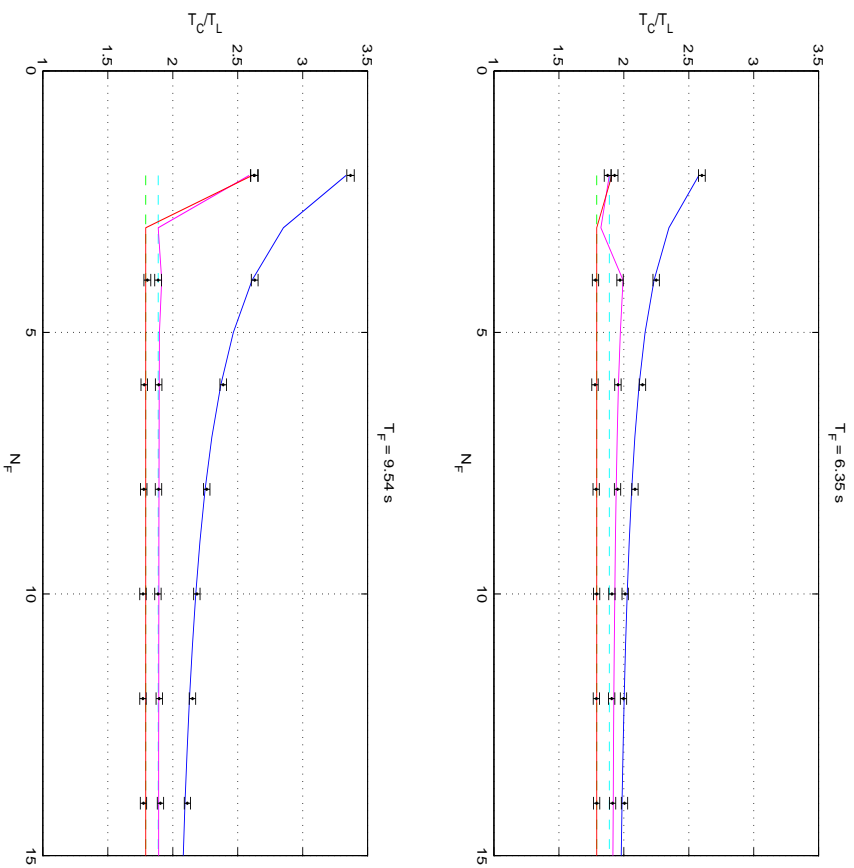
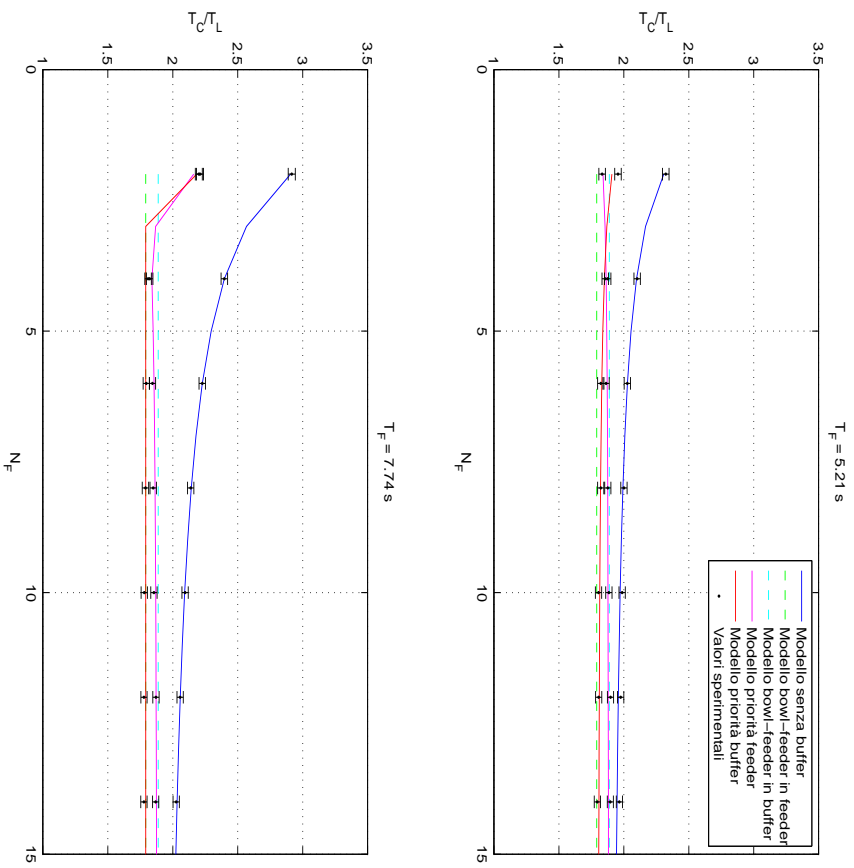


Figura 6.4: Influenza del numero di componenti nel feeder sul tempo ciclo. $\tau_S = 0.197$ s, $\tau_L = 2.11$ s. In senso orario $\tau_F = 5.21$ s, 6.35 s, 7.74 s e 9.54 s.

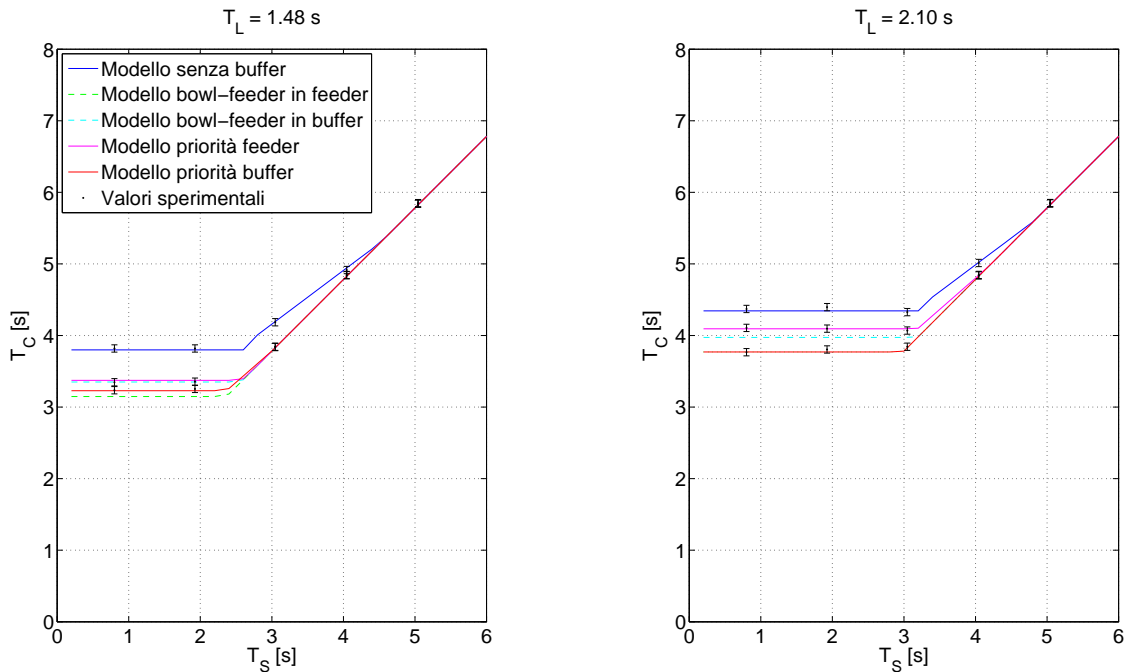


Figura 6.5: Influenza del tempo di scarico sul tempo ciclo. $\tau_F = 6.40$ s, $N_F = 8$. A sinistra $\tau_L = 1.48$ s, a destra $\tau_L = 2.10$ s.

situazione è in accordo con quanto esposto durante la realizzazione del modello. Superato il valore soglia di tempo di scarico il tempo ciclo assume un andamento transitorio verso un asintoto lineare crescente. Tale asintoto rappresentato in un grafico $T_C - \tau_S$ con assi caratterizzati dalla medesima scala come in figura assume un andamento con inclinazione pari a 45° . Tale affermazione indica che il tempo di scarico, per valori oltre la soglia, influisce linearmente sul tempo ciclo.

Va considerato che la soglia oltre cui il tempo ciclo varia l'andamento costante è differente a seconda della tipologia di gestione della cella. Come si è visto durante l'elaborazione del modello il tempo di scarico influisce su una serie di parametri in maniera differente e con diverse equazioni. Si può identificare come soglia il valore a cui il tempo di scarico contribuisce, seppur in maniera limitata, al tempo ciclo. Vi è un intervallo del tempo di scarico oltre tale soglia in cui la sua influenza è verificata solo per alcuni parametri (ad esempio N_{out}), e questo intervallo rappresenta il transitorio verso l'asintoto. Quando il tempo di scarico è tale da soddisfare tutte le disequazioni relative allora la sua influenza è strettamente

lineare.

Il numero discreto di prove sperimentali effettuate per ciascun caso è dovuto alle limitazioni imposte dai relè temporizzati. La selezione dell'intervallo di tempo non è continuo e pertanto si è limitati nelle possibili prove realizzabili.

6.2.4 Tempo di rinnovo feeder

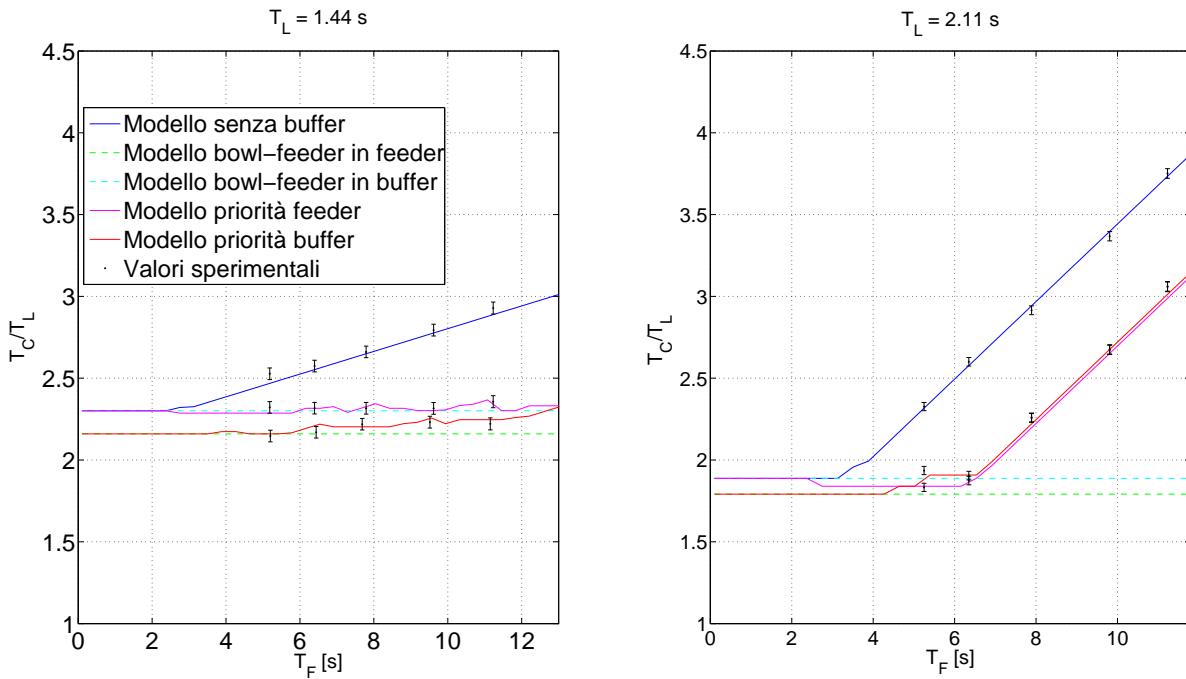


Figura 6.6: Influenza del tempo di rinnovo feeder sul tempo ciclo. $\tau_S = 0.20$ s. A sinistra $\tau_L = 1.44$ s e $N_F = 10$, a destra $\tau_L = 2.11$ s e $N_F = 2$.

In figura 6.6 è rappresentato il caratteristico andamento del tempo ciclo in funzione del tempo di rinnovo del feeder. Si può notare un andamento quasi costante ad eccezione di leggere fluttuazioni a cui segue un comportamento lineare con inclinazione a 45° come visto nel caso del tempo di scarico. Tuttavia vi sono notevoli differenze rispetto al caso precedente: l'asintoto non è il medesimo per le tre curve. Il valore superiore è rappresentato dal caso con assenza di bufferizzazione. Le due differenti priorità di svuotamento nella figura a destra concorrono verso lo stesso asintoto ma non necessariamente questa situazione accade. Sono state osservate combinazioni di parametri della cella in cui questo non si verifica.

A differenza del caso precedentemente analizzato non vi sono transitori fra l'andamento orizzontale e quello lineare crescente. Questo avviene perchè l'aumento del tempo ciclo è dovuto solo ad un'attesa in home del termine ricambio feeder. Il limite di ricambio feeder è funzione degli altri parametri della cella. Per mostrare gli asintoti anche per i casi con bufferizzazione nella figura a destra il numero di componenti nel feeder è stato ridotto fino a 2.

6.2.5 Distanza fra postazioni

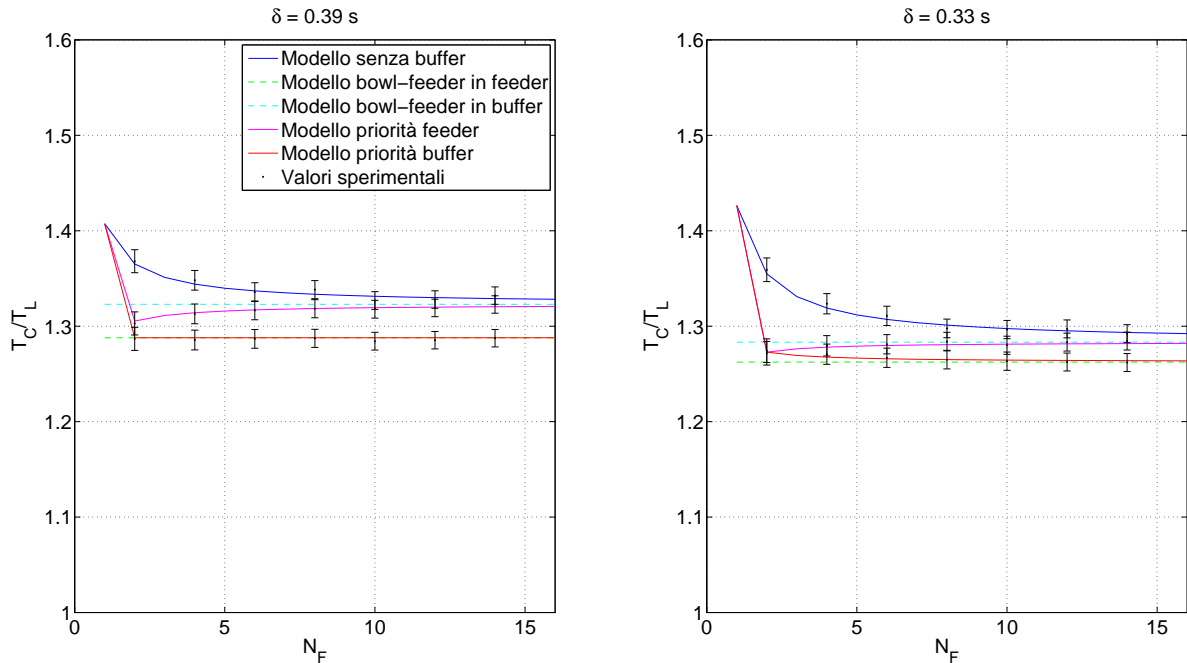


Figura 6.7: Influenza delle tempistiche di movimentazione sul tempo ciclo. $\tau_S = 0.24$ s, $\tau_L = 5.80$ s, $\tau_F = 7.62$ s. A sinistra $\delta = 0.39$ s a destra $\delta = 0.33$ s.

In figura 6.7 sono riportati i risultati ottenuti tramite due distinte configurazioni spaziali della cella. Come esposto in sezione 6.1.1 le configurazioni analizzate sono rappresentate da una distanza angolare fra postazioni pari a 20° e 40° . I due grafici rappresentano due situazioni in cui varia solamente tale parametro. In accordo con quanto esposto fino ad ora il parametro rappresentativo della movimentazione è δ . Dai due grafici si può notare che le curve limite traslano verso tempi minori poichè minore è la distanza fra le postazioni. Inoltre le due curve

limite sono più ravvicinate fra loro perchè la differenza fra il prelievo di un componente dal feeder o dal buffer ha meno influenza sul tempo finale. Come ci si aspettava una distanza minore fra postazioni riduce la differenza di tempo ciclo fra le priorità differenti.

È interessante notare come nel caso a distanza maggiore, figura a sinistra, la gestione con priorità attribuita al buffer coincide con la curva limite inferiore e pertanto si utilizza il buffer in maniera ottimale. In un'eventuale rappresentazione tempociclo/curvalimite anzichè tempociclo/tempolavorazione il caso con $\delta = 0.39$ s indicherebbe una migliore ottimizzazione della cella rispetto alle potenzialità limite rispetto al caso con $\delta = 0.33$ s. In termini di tempo ciclo assoluto comunque la distanza minore offre vantaggi nonostante non si raggiunga l'ottimizzazione ideale.

Le figure inoltre riportano un caso interessante per la gestione con priorità di svuotamento attribuita al feeder. Il raggiungimento dell'asintoto orizzontale rappresentante il caso ideale con componente sempre disponibile avviene con andamento crescente anzichè decrescente. Tale situazione indica che l'aumento di pezzi nel feeder è controproducente per il tempo ciclo della cella e potrebbe sembrare errato. In realtà la giustificazione di tale andamento è dovuta alla sola maggiore distanza dalla lavorazione. Per un numero minore di componenti nel feeder, a parità di postazioni di buffering utilizzate, il contributo al tempo ciclo dei pezzi prelevati dal buffer è maggiore. Di conseguenza aumentando il numero di componenti presenti nel feeder aumentano i cicli di prelievo da feeder, più distante rispetto alla lavorazione, e pertanto anche il tempo ciclo.

6.2.6 Tempo di lavorazione

In figura 6.8 sono rappresentati i risultati di alcuni test con il tempo di lavorazione variabile. Sulle ascisse è stato scelto di porre il tempo ciclo rapportato al tempo di lavorazione per poter apprezzare l'andamento iperbolico delle curve. La caratteristica principale dei grafici è rappresentata dalle iperboli dei cicli limite e del ciclo in assenza di buffering. Le curve che prevedono l'utilizzo del buffer si discostano dai valori limite per tempi ridotti della lavorazione. Per tempistiche elevate le curve convergono ad un unico asintoto iperbolico. L'utilizzo delle ascis-

se come esposto precedentemente consente inoltre di valutare la saturazione del macchinario di lavorazione. Fissata la tempistica richiesta da tale macchinario ottimizzare il tempo ciclo equivale a massimizzare la saturazione di tale macchinario. Per tempistiche elevate le ascisse tendono al valore unitario indicando che la postazione di lavorazione viene saturata maggiormente, in quanto le tempistiche di movimentazione assumono minor influenza.

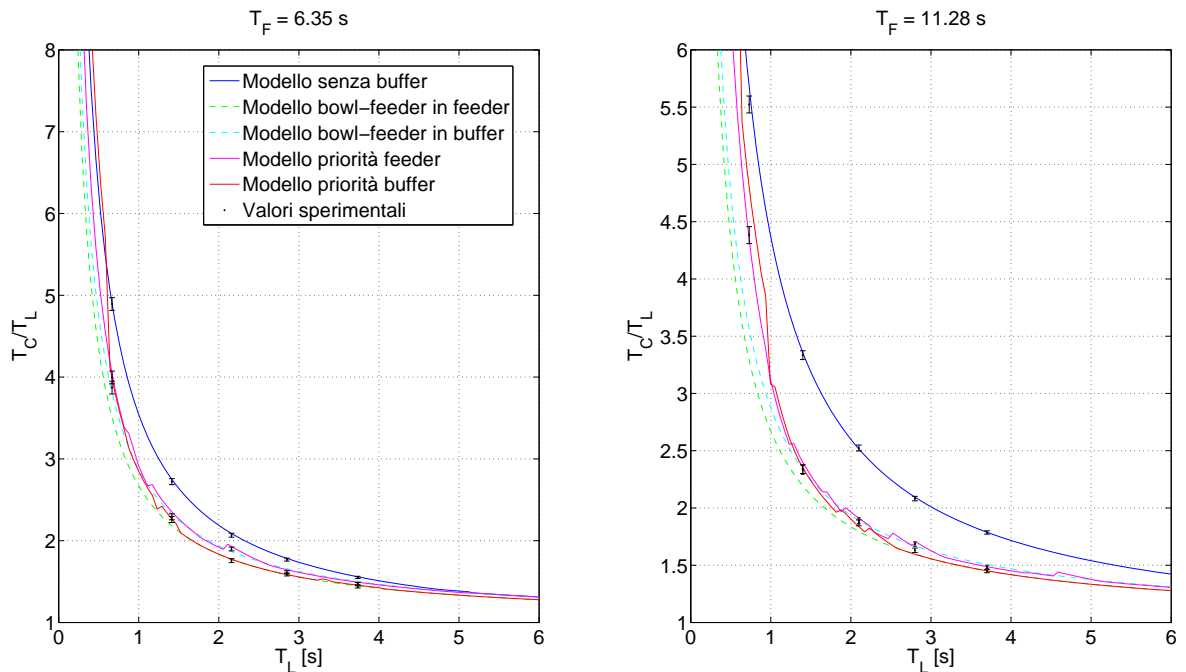


Figura 6.8: Influenza del tempo di lavorazione sul tempo ciclo. $\tau_S = 0.24$ s, $N_F = 6$. A sinistra $\tau_F = 6.35$, a destra $\tau_F = 11.29$ s.

Si può in particolare notare che per tempi di lavorazione ridotti la gestione con priorità associata al buffer risulta sconveniente addirittura rispetto alla gestione senza buffer. Tale comportamento è giustificato dalla necessità di bufferizzare un componente non appena un pezzo è stato rilasciato sul macchinario di lavorazione. Tale inserimento avviene in tempo completamente non mascherato, poichè nonostante la lavorazione termini poco dopo il rilascio del componente grezzo il manipolatore è costretto a terminare il buffering di un pezzo avendo già cominciato tale operazione. Considerando la situazione in figura, una gestione senza buffer risulta più conveniente per tempi ridotti della postazione di lavorazione. Il

buffer in tale caso offre maggiore utilità per coprire le sole tempistiche durante il cambio feeder.

6.2.7 Dimensione buffer

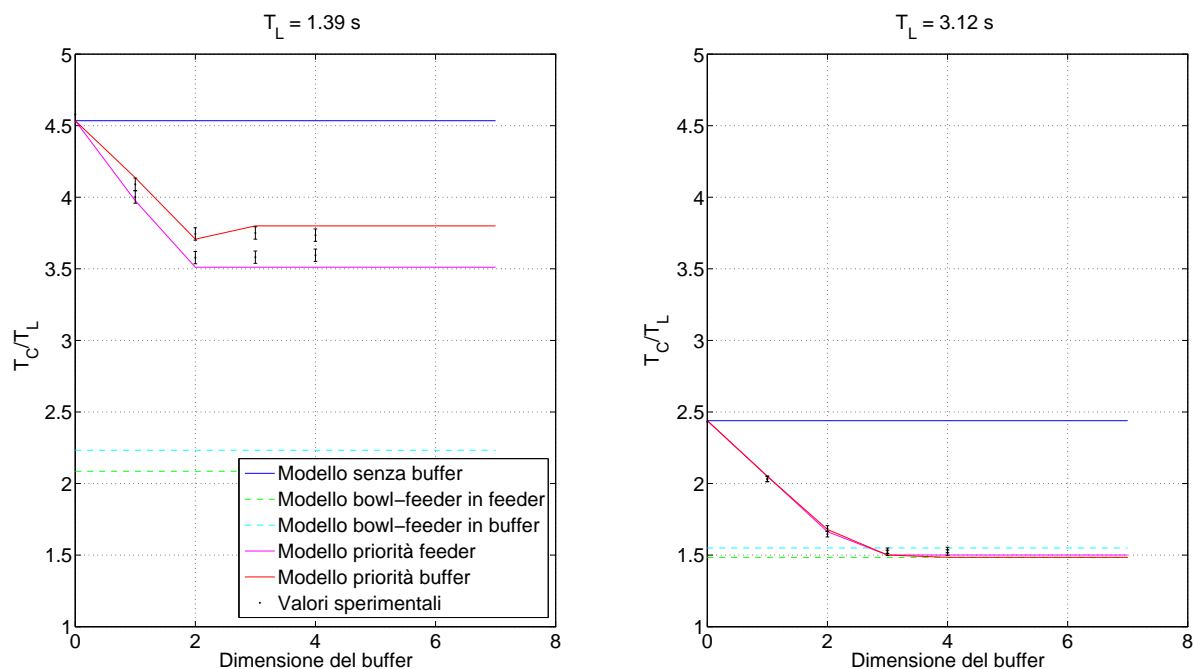


Figura 6.9: Influenza della dimensione della postazione di buffering sul tempo ciclo. $N_F = 4$, $\tau_S = 0.25$, $\tau_F = 15.40$ s. A sinistra $\tau_L = 1.39$ s, a destra $\tau_L = 3.12$ s.

La figura 6.9 rappresenta i risultati sperimentali di due prove in cui è stato variata la sola dimensione del buffer. Tale prova è stata effettuata principalmente con lo scopo di giustificare la validità di un'affermazione precedente. Le altre prove esposte nel presente capitolo sono state realizzate infatti indicando una dimensione del buffer ampiamente superiore al numero di componenti presenti nel feeder. Tale situazione equivale a fornire una dimensione tale al buffer che, superato il transitorio iniziale, consenta di raggiungere la situazione a regime cercata. Come si può osservare dalle figure il tempo ciclo rimane costante per una dimensione del buffer superiore al numero di componenti nel feeder. Questo comportamento è maggiormente giustificato dalla non considerazione dei tempi

rilevati per i primi cicli di lavoro, ovvero del transitorio iniziale. Se si attribuisce una dimensione elevata al buffer i primi cicli saranno caratterizzati da un massivo deposito di componenti sulla relativa postazione fino a raggiungere il riempimento totale. Tuttavia parte di questi componenti non vengono prelevati in alcun caso dal manipolatore, ad eccezione di guasti a monte del sistema di ingresso pezzi non considerato nella presente tesi.

Capitolo 7

Ottimizzazione della produttività

Nel presente capitolo vengono presentati i risultati relativi all'ottimizzazione del tempo ciclo. Sono riportate le curve parametriche, obiettivo principale della presente tesi, che consentono di effettuare graficamente la scelta dei parametri della cella stabilito un tempo ciclo target. Vengono, inoltre, riportati i risultati dello studio sul posizionamento del buffer per determinarne la locazione ottimale. Si riportano, infine, i risultati dello studio con N_F variabile. Al termine del capitolo viene proposto un metodo di ulteriore ottimizzazione che richiede solamente l'imposizione di una limitazione nella gestione dinamica della cella.

7.1 Curve parametriche

La rappresentazione in curve parametriche dei risultati dello studio consente di valutare graficamente l'influenza reciproca dei differenti parametri. È sufficiente determinare i parametri di progetto non vincolati per poi ricercarne la combinazione ottimale tramite l'utilizzo delle corrette curve parametriche. È possibile, inoltre, valutare quale sia l'influenza sulla produttività di un'eventuale scostamento di tali parametri rispetto al valore di progetto.

Visto l'elevato numero di parametri che caratterizza l'architettura della cella in esame si possono determinare, a seconda delle esigenze progettuali, curve parametriche notevolmente differenti fra di loro. È possibile variare la combinazione di parametri rappresentati e il loro range considerato. Si riportano ora solamente

alcune delle combinazioni possibili. Le curve non rappresentate sono, comunque, ricavabili utilizzando i modelli teorici esposti in precedenza.

Per limitare il numero di parametri da rappresentare sono state effettuate alcune assunzioni per le relazioni fra essi. Si desidera, inoltre, esporre tutti i risultati in funzione di δ , indicando, ad esempio, il tempo di lavorazione come $k\delta$. Il tempo di rinnovo feeder viene espresso, anzichè in relazione a δ , in funzione del tempo di lavorazione poichè la correlazione fra τ_L e τ_F ha maggiore significato fisico. Si suppone, inoltre, che il tempo di scarico sia trascurabile. I risultati verranno rappresentati in funzione non del tempo ciclo τ_C ma di τ_C/τ_L , principalmente per vedere la saturazione del macchinario di lavorazione e, in secondo luogo, per avere una variabile reale, libera da dimensioni e dipendenza da ulteriori parametri come δ .

Per quanto riguarda i parametri di movimento, le curve che verranno ora esposte sono realizzate assumendo le seguenti relazioni:

$$\eta = \frac{3\delta}{4} \quad (7.1)$$

$$\tau_H = \frac{\delta}{2} \quad (7.2)$$

$$\epsilon = \frac{\delta}{8} \quad (7.3)$$

Queste rappresentano un'approssimazione dei valori utilizzati nei test precedenti in cui $\delta = 0.391s$, $\eta = 0.290s$, $\tau_H = 0.194s$ e $\epsilon = 0.05s$ (tabella 6.4). Tale scelta, arbitraria, è dettata dalla sola impossibilità di rappresentare in questa tesi tutte le possibili interazioni fra parametri.

Le curve parametriche riportate in figura 7.1, relative alla priorità attribuita al feeder, presentano sulle ascisse il numero di componenti presenti ad ogni rinnovo. Sulle ordinate è rappresentato il tempo ciclo rapportato al tempo di lavorazione. In ciascun grafico le differenti curve indicano una dimensione differente del buffer, ovvero il numero di slot disponibili per l'utilizzo. Ciascuna riga di grafici indica un valore differente di tempistiche di lavorazione e le colonne sono invece relative al tempo di rinnovo del feeder.

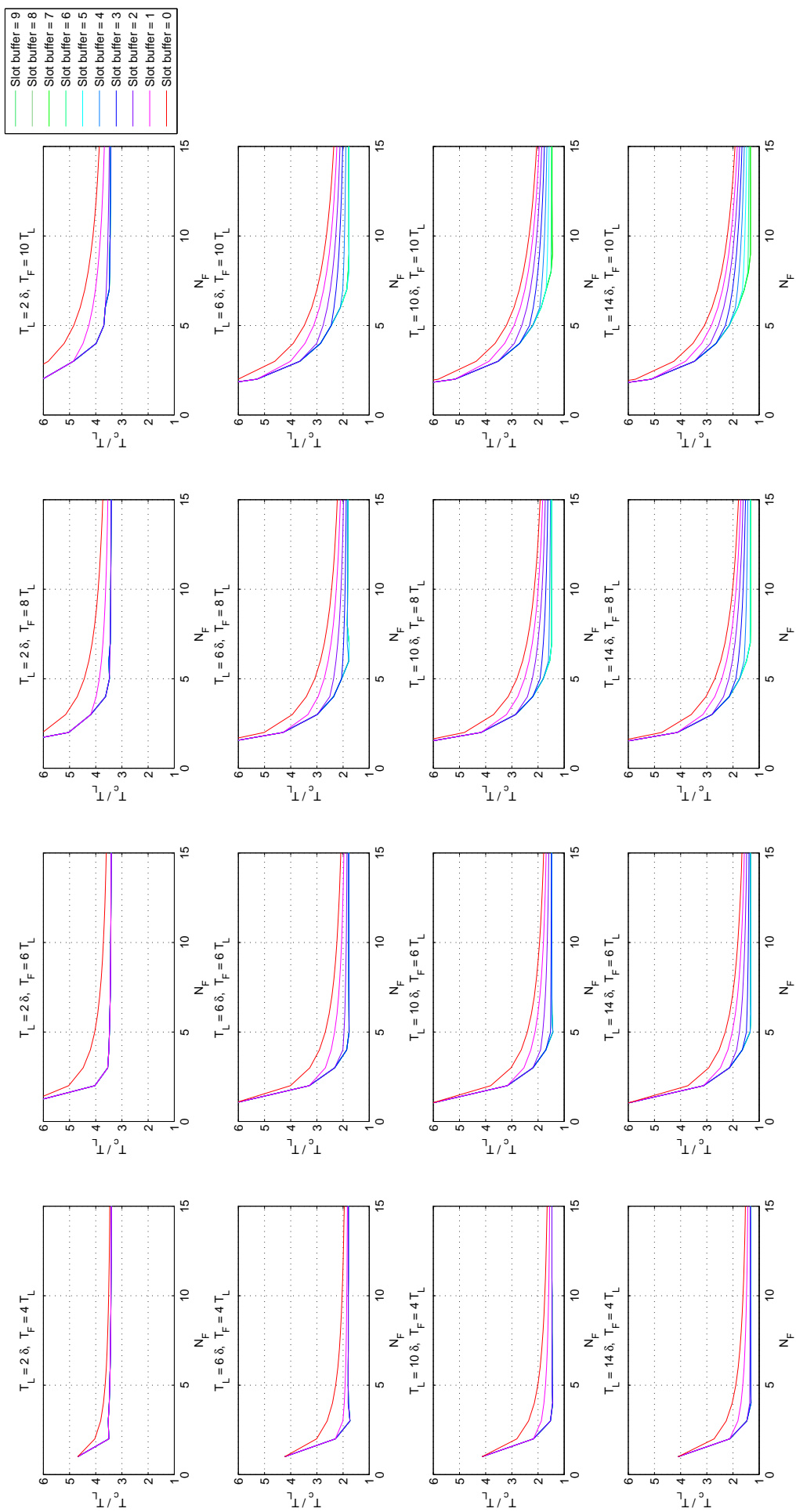


Figura 7.1: Curve parametriche. $\eta = \frac{3\delta}{4}$, $\tau_H = \frac{\delta}{2}$, $\epsilon = \frac{\delta}{8}$. τ_S trascurabile.

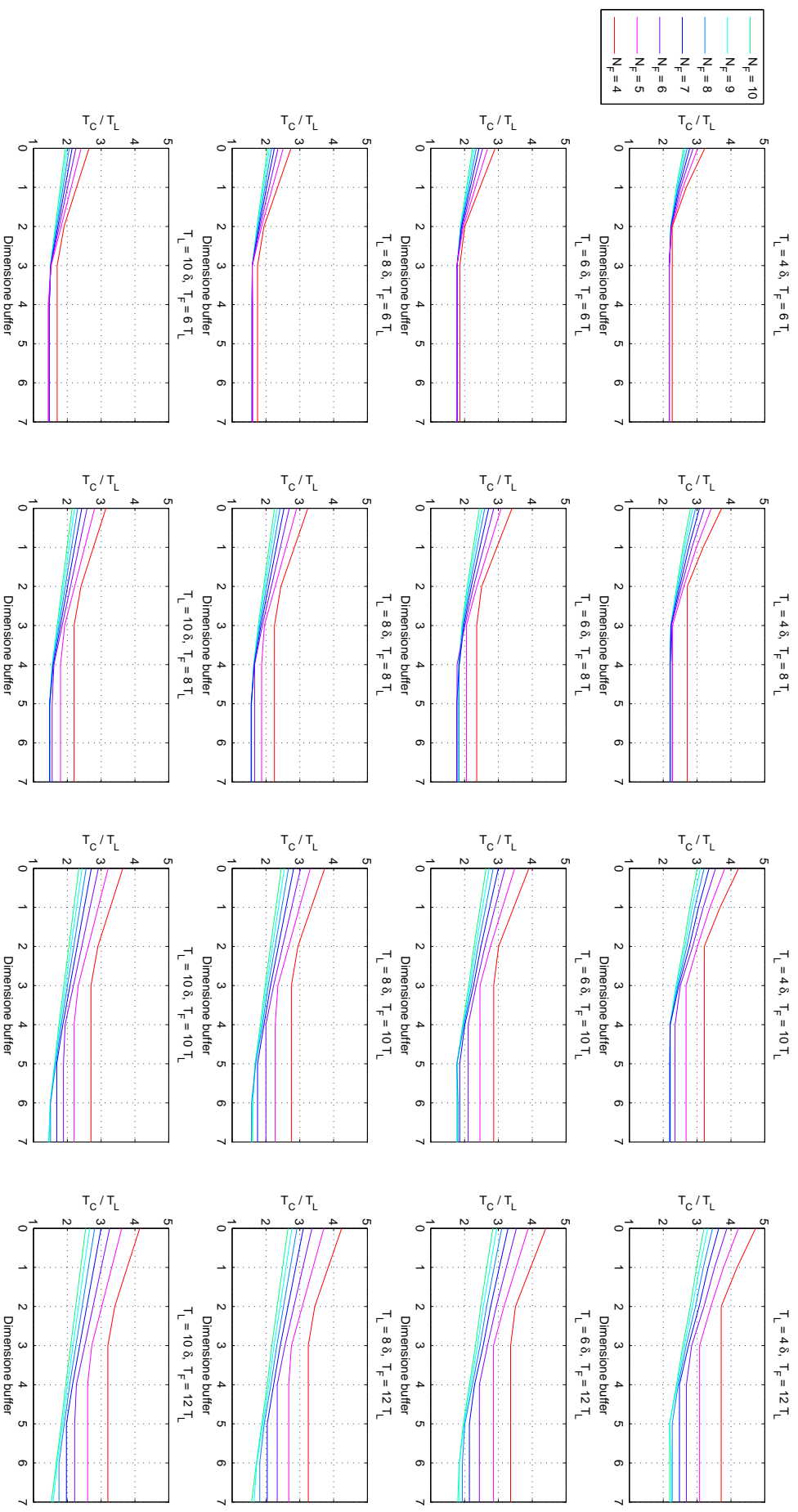


Figura 7.2: Rappresentazione alternativa delle curve parametriche riportate in figura 7.1.

Si nota come un aumento del tempo di lavorazione o di rinnovo feeder determini un maggiore utilizzo di slot nel buffer. Dove, infatti, è rappresentato un numero minore di curve significa che le restanti sono sovrapposte. In altre parole, la disponibilità di un numero maggiore di slot per il buffering non implica benefici alla produttività della cella.

Aumentando il rapporto τ_L/δ le curve traslano verso valori inferiori: la postazione di lavorazione è maggiormente saturata poichè le tempistiche di movimentazione assumono un peso, complessivamente, minore. Viceversa, aumentando il rapporto τ_F/τ_L la curva inferiore raggiunge l'asintoto orizzontale per valori di N_F maggiori, poichè maggiore è il numero teorico di componenti processabili dal buffer durante il rinnovo.

Ogni curva a N_B maggiore può essere vista come un'estensione di quella precedente. Si può cioè, per ogni valore di N_B , identificare un valore di N_F prima del quale la curva coincide con quella di $N_B - 1$. Per valori minori invece esse coincidono poichè la dimensione del feeder è tale da rendere superfluo un numero elevato di slot nel buffer.

Un metodo alternativo di rappresentazione delle medesime curve parametriche è riportato in figura 7.2. La dimensione del buffer è stata invertita con quella del feeder. Sulle ascisse è, infatti, ora riportato il numero di slot disponibili nel buffer e la parametrizzazione viene effettuata rispetto alla dimensione del feeder. Si nota in particolare l'andamento orizzontale dei grafici superato un valore caratteristico di N_B . Individuata la curva di progetto, tale valore indica il numero minimo di slot nel buffer richiesti per ottenere la produttività massima della cella.

Si può apprezzare, inoltre, la correlazione fra il numero di componenti nel feeder e il tempo di rinnovo dello stesso. Per valori ridotti di τ_F , le curve a differente N_F assumono l'andamento orizzontale alla medesima dimensione del buffer. Non è, in questo caso, il numero di componenti nel feeder a determinare un maggiore utilizzo della postazione di buffer. Per valori superiori di τ_F , invece, le differenti curve assumono asintoti orizzontali differenti. Un tempo di rinnovo elevato, infatti, consente un miglior utilizzo del buffer, soprattutto in combinazione con un N_F elevato.

Si riporta, a titolo puramente esemplificativo, anche una serie di curve pa-

rametriche per valutare l'influenza del tempo di scarico. In figura 7.3 è, infatti, riportata tale situazione, in cui, rispetto ai casi precedenti, la dimensione del feeder è mantenuta costante e pari a 10 pezzi e si impone un differente tempo di scarico.

Si conclude la sezione affermando che le curve parametriche ottenibili per il caso con priorità di svuotamento attribuita al buffer assumono un andamento analogo a quelle presentate e, pertanto, non vengono riportate. La scelta della modalità di attribuzione della priorità non può essere effettuata utilizzando le curve parametriche. Tale decisione risulta notevolmente semplificata dalla consultazione di curve come quelle presentate nel precedente capitolo (ad esempio figura 6.5) in cui entrambi gli andamenti sono riportati nello stesso grafico. In questo modo risulta immediata la determinazione di quale curva assuma il valore inferiore del tempo ciclo.

7.1.1 Confronto con prove sperimentali

Si riportano in figura 7.4 i risultati di una campagna di prove realizzata per valutare l'attendibilità delle curve sperimentali derivanti dal modello teorico. Si può notare come i dati sperimentali siano, in gran parte, compatibili con il modello teorico e condividano con questo il medesimo andamento.

7.2 Posizionamento del buffer

Fino ad ora è stato arbitrariamente considerato un layout in cui le postazioni sono uniformemente distribuite lungo un arco di circonferenza. Si vuole analizzare l'influenza del posizionamento del buffer sul tempo ciclo rispetto a tale situazione. Un posizionamento ravvicinato al feeder consente senza dubbio l'inserimento di un numero maggiore di componenti nel buffer durante ciascuna lavorazione. Tuttavia, tale caso allunga inevitabilmente i cicli di scarico del buffer, in quanto quest'ultimo si trova ora più distante sia dalla postazione di scarico che dalla postazione di lavorazione. Si desidera determinare, a parità degli altri parametri caratteristici della cella, quale sia il posizionamento ideale per il buffer rispetto al caso equidistante.

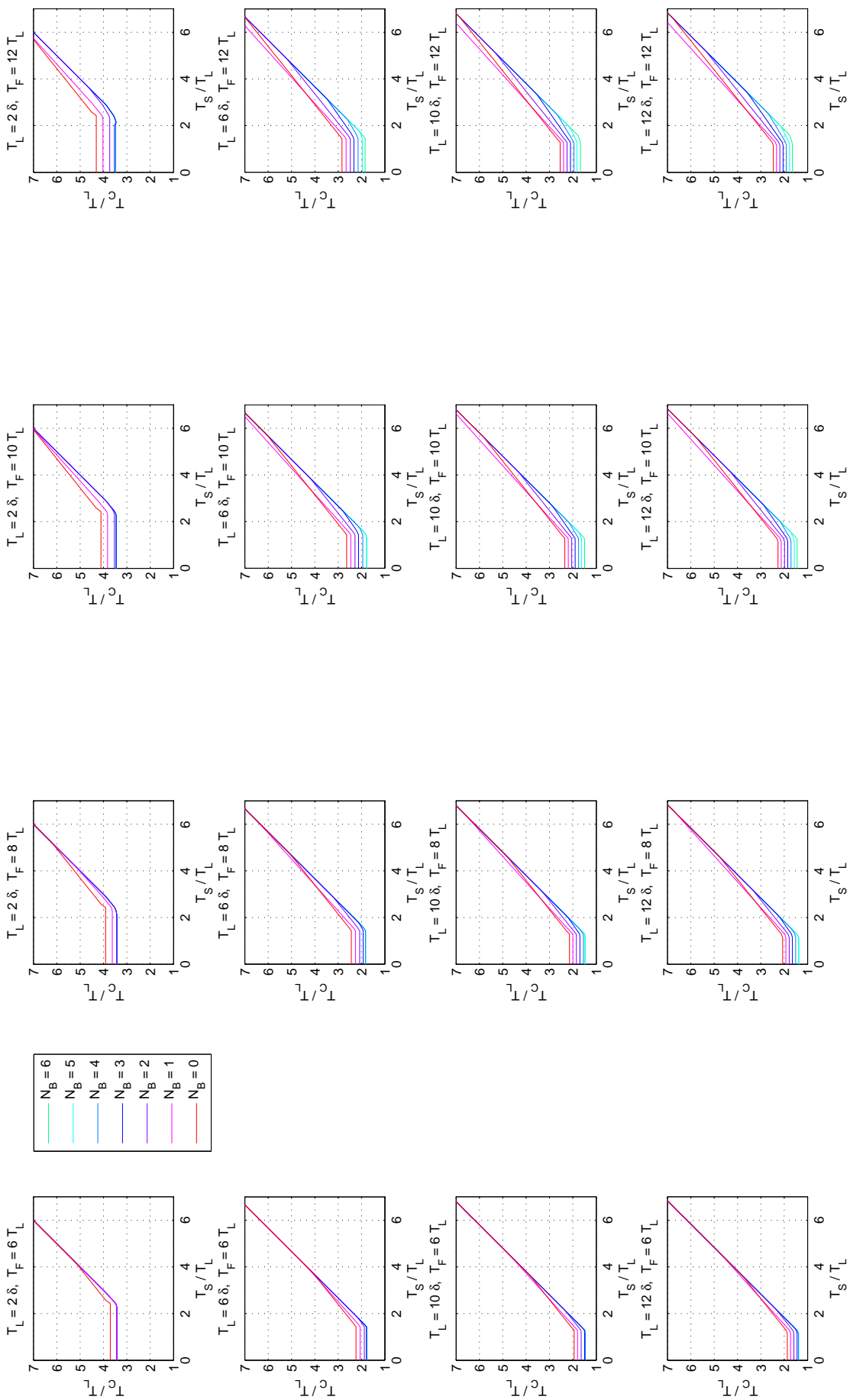


Figure 7.3: Curve parametriche. $N_F = 10$, $\eta = \frac{3\delta}{4}$, $\tau_H = \frac{\delta}{2}$, $\epsilon = \frac{\delta}{8}$.

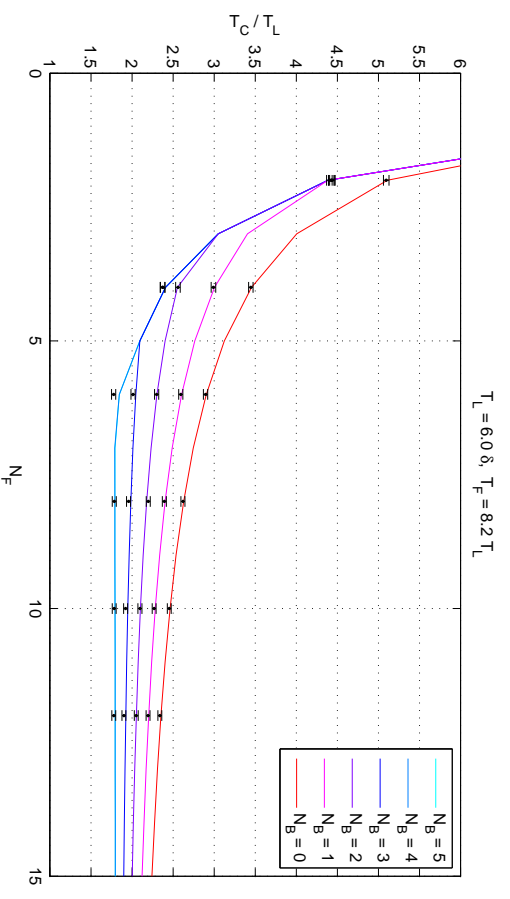
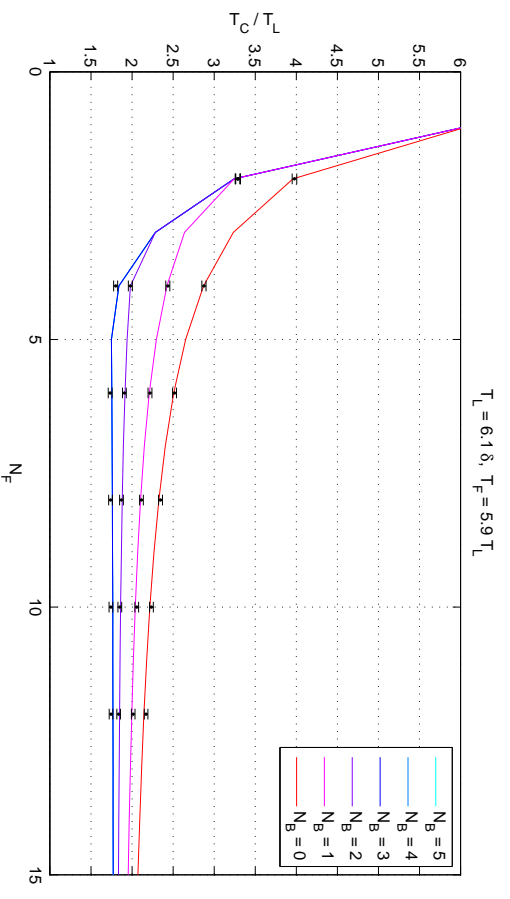
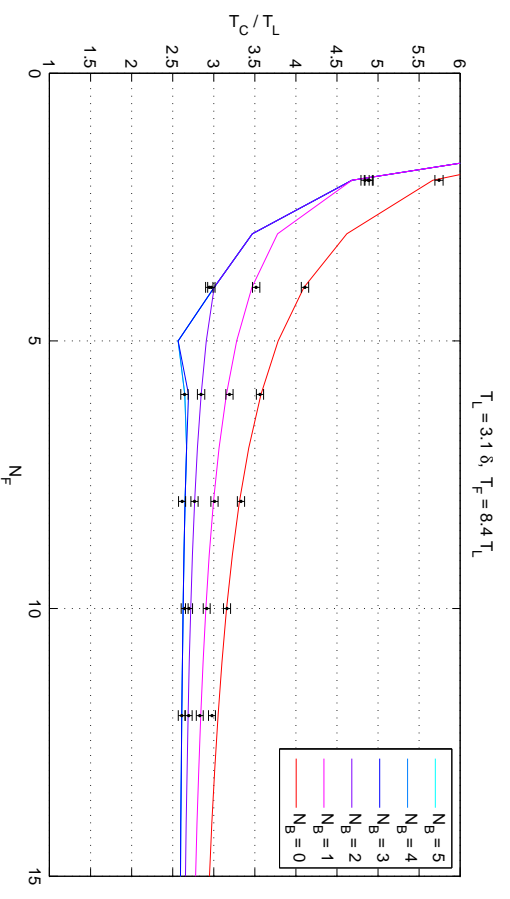
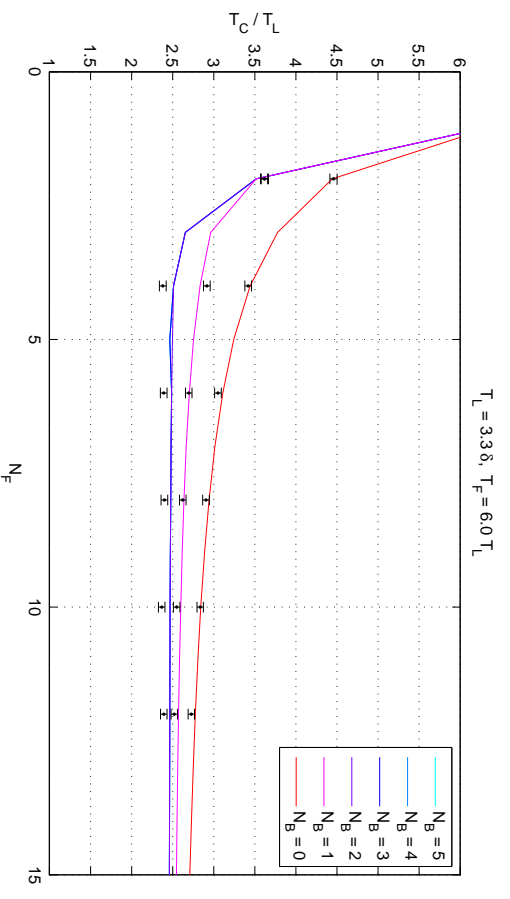


Figura 7.4: Determinazione sperimentale di curve parametriche.

A livello analitico è stato generalizzato l'algoritmo utilizzando non più la semplice assunzione di tempistiche additive, ma il caso più generale di tempistiche Euclidee indipendenti fra loro (sezione 2.1.3). Il modello non viene riportato in quanto la procedura è la medesima di quella esposta precedentemente con l'utilizzo di un maggior numero di parametri di tempo anzichè i soli δ e η .

L'algoritmo così elaborato è più generale: permette infatti la valutazione di un numero maggiore di situazioni a scapito però di un numero di variabili di input superiore. Nel capitolo 3, si è preferito esporre il caso additivo per semplicità di notazione ed esposizione.

Il diagramma degli stati utilizzato è riportato in figura 7.5. I parametri δ_{FS} , δ_{LS} e δ_{FL} sono costanti indipendentemente dalla posizione del buffer poichè non dipendono da esso. I valori di δ_{FB} , δ_{BL} e δ_{BS} , invece, sono variabili. In altre parole, i parametri non costanti riguardano i tempi di movimentazione per le operazioni da e verso il buffer.

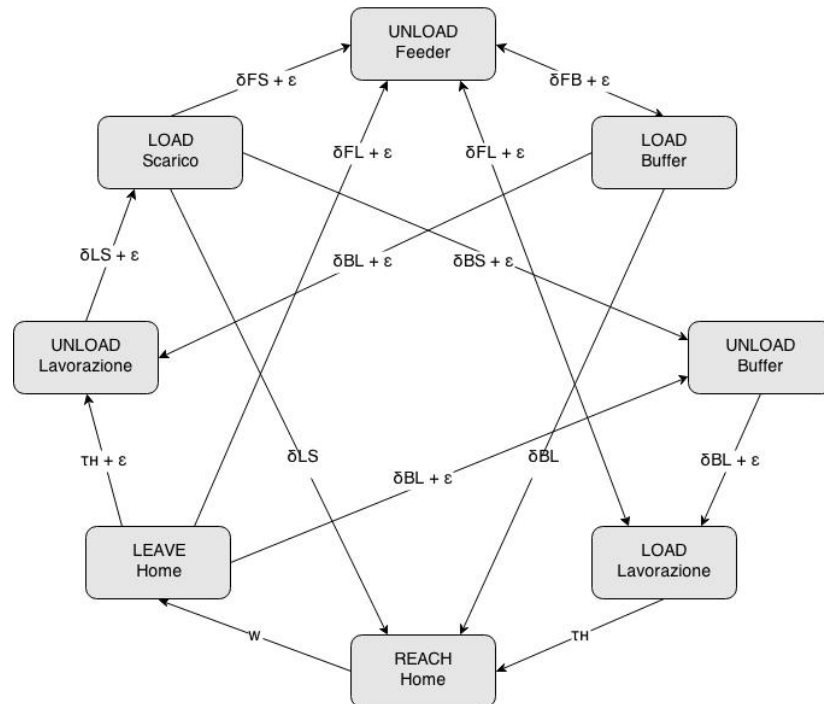


Figura 7.5: Diagramma con la notazione utilizzata nella gestione dell'algoritmo con tempistiche non additive per valutare il posizionamento ottimale del buffer.

A livello pratico sono state effettuate simulazioni variando la posizione del

buffer pur mantenendo la sua locazione sulla semicirconferenza a cui appartengono le altre postazioni. Il numero di posizioni differenti considerate è pari a 7, uniformemente distribuite lungo l'arco individuato da postazione di lavorazione e feeder. Per la precisione, poichè per le altre postazioni è stata utilizzata la configurazione complessiva disposta su 120° come per i precedenti test, le locazioni del buffer sono state disposte imponendo un angolo pari a 10° fra di esse. In figura 7.6 è riportato lo schema della cella così descritta.

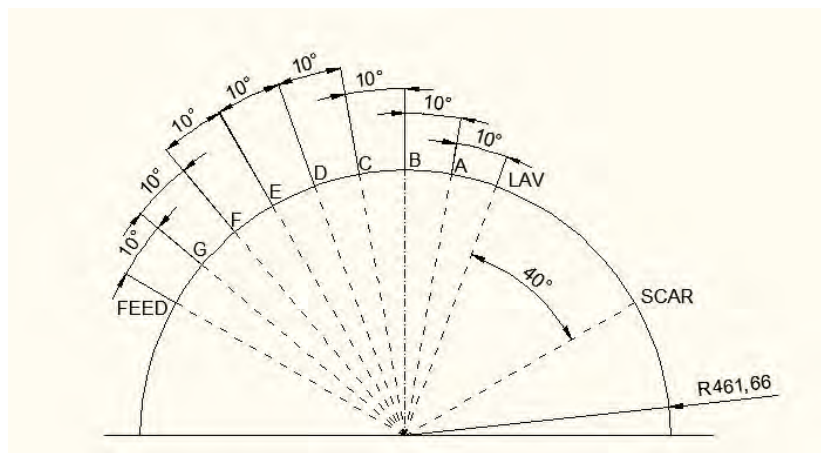


Figura 7.6: Schema con quote delle posizione del buffer considerate

7.2.1 Assunzioni per la movimentazione

In seguito alla realizzazione dei test relativi al caso esposto, si desidera esporre i risultati riportando sulle ascisse un parametro indicativo della posizione del buffer. Definito tale valore con ξ , si vuole rappresentare con $\xi = 1$ il caso di buffer coincidente con il macchinario di lavorazione. Viceversa $\xi = 0$ rappresenta il caso di buffer coincidente con il feeder. $\xi = 0.5$, invece, rappresenta la situazione considerata fino ad ora con postazioni equidistanti. I parametri variabili a seconda della posizione del buffer, tuttavia, sono tre: δ_{FF} , δ_{BL} e δ_{BS} . Sorge il problema di associare ad un solo valore di ξ una terna dei parametri in esame. Successivamente è richiesta l'assegnazione ad ogni prova sperimentale o, più precisamente, ad ogni posizione del buffer considerata, del relativo valore di ξ .

Per semplificare la trattazione si consideri una movimentazione additiva per i parametri non variabili e un'interpolazione lineare fra le tempistiche note per i restanti. In altre parole, utilizzando nuovamente i parametri δ e η :

$$\delta_{LS} = \delta \quad (7.4)$$

$$\delta_{FL} = 2\delta - \eta \quad (7.5)$$

$$\delta_{FS} = 3\delta - 2\eta \quad (7.6)$$

Per i parametri restanti si considerano funzioni lineari di ξ in modo da rispettare le condizioni note. Si supponga ad esempio di voler determinare $\delta_{BS} = f(\xi)$. Per $\xi = 1$ il valore assunto deve essere pari a $3\delta - 2\eta$ poichè equivalente alla situazione rappresentata da δ_{FS} . Per $\xi = 0.5$ invece $\delta_{BS} = 2\delta - \eta$ in accordo con le tempistiche utilizzate fino ad ora. Di conseguenza, effettuando una semplice interpolazione lineare, si ottiene:

$$\delta_{BS} = 3\delta - 2\eta - (2\delta - 2\eta)\xi \quad (7.7)$$

Lo stesso ragionamento può essere effettuato per gli altri due parametri funzioni di ξ :

$$\delta_{BL} = 2\delta - \eta - (2\delta - 2\eta)\xi \quad (7.8)$$

$$\delta_{FB} = \eta + (2\delta - 2\eta)\xi \quad (7.9)$$

Si specifica che per valori di ξ prossimi a 0 o 1 il senso fisico della rappresentazione viene a mancare, poichè l'assunzione di linearità del moto non è utilizzabile se la stazione di partenza e destinazione coincidono. Tali casi non sono tuttavia considerati nelle prove sperimentali e ci si limita ad affermare che il modello non ha valenza fisica per i valori estremi assumibili dal parametro ξ .

L'assegnazione di ξ ai casi sperimentali considerati è stata realizzata tramite prove sulle tempistiche dei movimenti, i cui risultati sono riportati in tabella 7.1. Sono stati rilevati i tempi δ_{FB} , δ_{BL} e δ_{BS} e, noto δ e ξ , si è determinato

| | A | B | C | D | E | F | G |
|-------------------|-------|-------|-------|-------|-------|-------|-------|
| δ_{BL} [s] | 0.276 | 0.308 | 0.340 | 0.368 | 0.392 | 0.420 | 0.444 |
| δ_{FB} [s] | 0.444 | 0.420 | 0.392 | 0.368 | 0.340 | 0.308 | 0.276 |
| δ_{BS} [s] | 0.392 | 0.420 | 0.444 | 0.472 | 0.500 | 0.524 | 0.552 |
| ξ_1 | 0.94 | 0.79 | 0.63 | 0.50 | 0.38 | 0.25 | 0.13 |
| ξ_2 | 0.87 | 0.75 | 0.62 | 0.50 | 0.37 | 0.21 | 0.06 |
| ξ_3 | 0.88 | 0.75 | 0.63 | 0.50 | 0.37 | 0.25 | 0.12 |
| $\bar{\xi}$ | 0.90 | 0.76 | 0.63 | 0.50 | 0.37 | 0.24 | 0.10 |

Tabella 7.1: Risultati del test realizzato per associare ad ogni posizione del buffer considerata (A-G) il relativo valore ξ . $\delta = 0.368$, $\eta = 0.264s$.

il rispettivo valore di ξ invertendo le equazioni 7.7, 7.8 e 7.9. Il valore medio di questi è stato poi assunto come valore del parametro ξ relativo a tale posizione del buffer. Grazie a quanto esposto, utilizzando un solo parametro, è possibile rappresentare sia il modello teorico sia i risultati sperimentali.

7.2.2 Risultati

In figura 7.7 si riportano le curve parametriche per il caso in esame. Si può notare che tali curve hanno un andamento fortemente dipendente dai parametri della cella. Non è possibile determinare una posizione assoluta ottimale del buffer in termini di tempo ciclo. Non si può cioè stabilire a priori, indipendentemente dai parametri della cella, quale sia la posizione del buffer che consenta l'ottenimento della maggiore produttività.

Gli scalini presenti nelle curve sono determinati da un utilizzo di un numero differente di slot nella postazione di buffering. Una distanza minore fra buffer e lavorazione, infatti, velocizza il prelievo dei componenti durante il rinnovo feeder. Una distanza maggiore velocizza, invece, il processo di inserimento di componenti dal feeder al buffer. Nel momento in cui la relazione fra queste due situazioni è

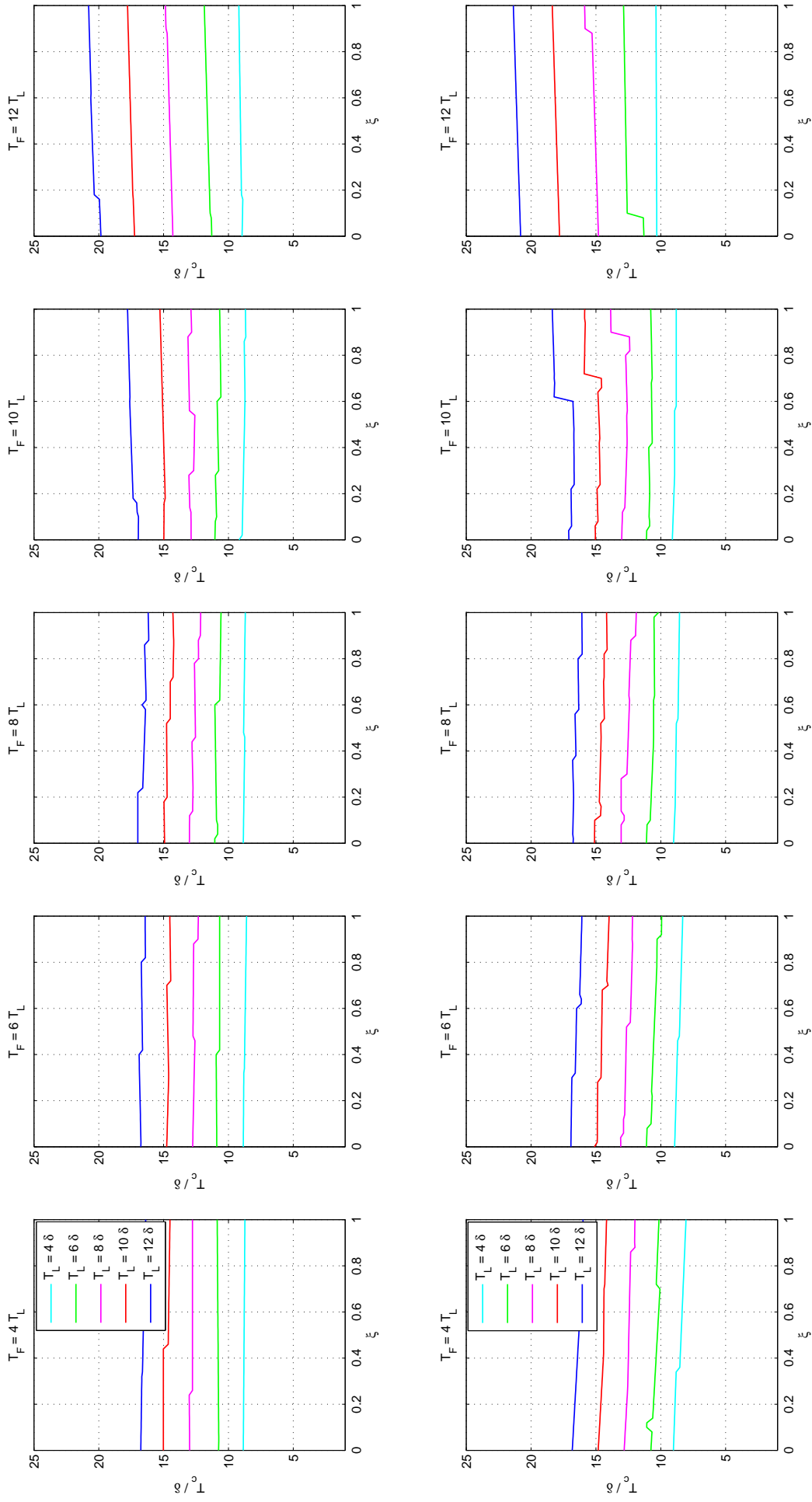


Figura 7.7: Curve parametriche: tempo ciclo in funzione della posizione del buffer, del parametro di spostamento e del tempo di lavorazione e di rinnovo feeder. $N_F = 10$.

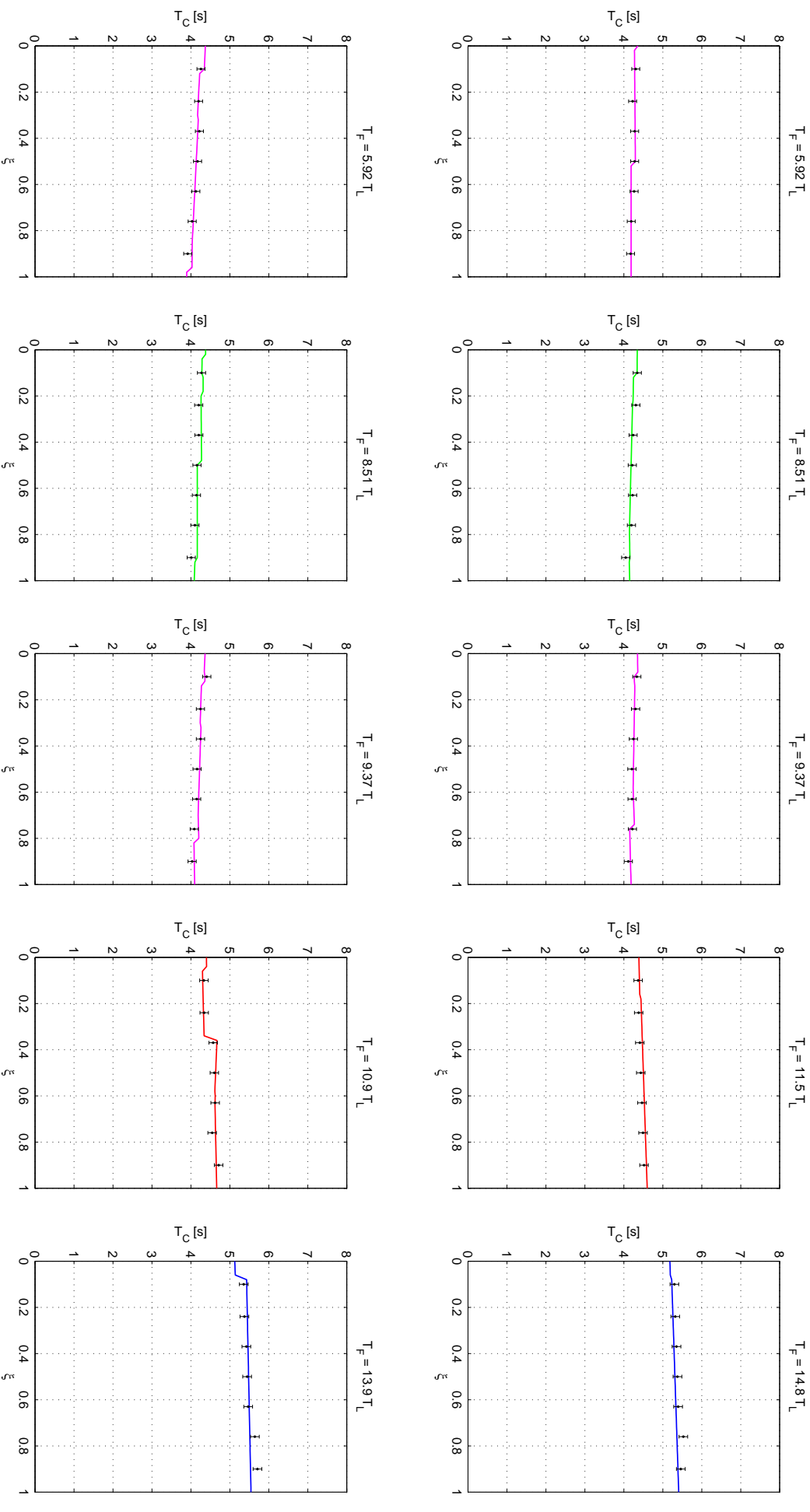


Figura 7.8: Risultati con posizione del buffer variabile. Priorità di svuotamento attribuita al feeder (sopra) e al buffer (sotto). $N_F = 10$, $\tau_L = 6\delta$.

tale da modificare il valore di slot del buffer utilizzati, si assiste ad uno scalino nelle relative curve.

Per valori ridotti del tempo di rinnovo feeder, l'andamento del tempo ciclo è decrescente con ξ . Il numero di slot del buffer utilizzati è ridotto, poichè vi è meno tempo a disposizione per processare componenti durante il rinnovo del feeder. Il collo di bottiglia non corrisponde all'inserimento di componenti nel buffer, bensì al suo svuotamento. Di conseguenza, il minore tempo ciclo si ottiene utilizzando una locazione in prossimità del macchinario di lavorazione. Tale condizione consente di ridurre la tempistica di ciascun ciclo di operazioni relative ad un componente prelevato dal buffer.

Per valori elevati del tempo di rinnovo feeder si osserva un andamento crescente del tempo ciclo con ξ . La giustificazione di tale comportamento va ricercata nella necessità di avviare il ricambio feeder il prima possibile, a causa della sua tempistica solo parzialmente gestibile in tempo mascherato. L'avvicinamento del buffer al feeder consente di effettuare l'inserimento dei componenti in un intervallo minore e questo, a parità di componenti, consente di avviare anticipatamente lo svuotamento del feeder. Tale situazione consente di effettuare in tempo mascherato una porzione maggiore del tempo di rinnovo.

7.2.3 Confronto con risultati sperimentali

Si riportano i risultati di una campagna di prove volta a verificare la compatibilità, rispetto ai dati sperimentali, del modello con posizionamento del buffer variabile. In figura 7.8 sono riportati i risultati per il caso con priorità assegnata al feeder e al buffer.

I risultati sperimentali sono compatibili con il modello teorico, in particolare si può osservare un ottimo fitting in termini di pendenza delle curve. I commenti esposti nella precedente sezione sono pertanto verificati sperimentalmente: non vi è una posizione conveniente in senso assoluto per il buffer ma essa dipende dai parametri caratteristici della cella.

7.3 Numero di componenti variabile

Fino ad ora sono stati riportati risultati con numero di componenti costanti ad ogni rinnovo feeder. È possibile, tuttavia, che per motivazioni legate al sistema di movimentazione del feeder questa situazione non sia verificata. Si consideri ad esempio un nastro trasportatore, caricato a monte da operatori, coadiuvato da un sistema di visione. Il numero di componenti forniti da tale sistema è determinato dal numero di pezzi presenti nell'area inquadrata e dall'effettivo loro riconoscimento. Se i componenti sono parzialmente sovrapposti, ad esempio, essi non vengono riconosciuti correttamente dal sistema di visione. Inoltre, a causa di distrazioni degli operatori, il numero di componenti disposti può variare nel tempo. La dimensione incerta del feeder giustifica a maggior ragione una gestione dinamica della cella poichè consente un adattamento alla situazione in esame. L'imposizione di una sequenza di movimenti stabilita a priori infatti risulterebbe inappropriata nel caso di N_F variabile. La medesima affermazione può essere riferita ad eventuali tempistiche delle postazioni variabili.

La situazione in esame viene per semplicità caratterizzata da due parametri: il numero di componenti massimo riscontrabile e la distribuzione di probabilità rispetto a tale valore, rappresentativa della successione di componenti disponibili. Si vuole determinare l'andamento del tempo ciclo in seguito a un numero elevato di rinnovi feeder, confrontandolo con la produttività nel caso di dimensione del feeder costante nel tempo.

Si considera in questa tesi la sola distribuzione rettangolare di probabilità. Qualsiasi valore compreso fra l'unità e il massimo N_F è, cioè, equiprobabile ad ogni rinnovo del feeder. In figura 7.9 sono riportati i risultati di una simulazione in cui si assume tale distribuzione di probabilità.

Si può osservare innanzitutto che, per valori di $N_F = 1$, le due curve coincidono, poichè anche nel caso probabilistico questo rimane costante e pari all'unità. Per valori superiori la curva con componenti in numero variabile indica un tempo ciclo maggiore: tale differenza si riduce all'aumentare di N_F fino a ridursi a 0. Il comportamento descritto è giustificato dal fatto che, per valori ridotti di N_F , il buffer non raggiunge la saturazione, anche nel caso di valore costante. Qualsiasi

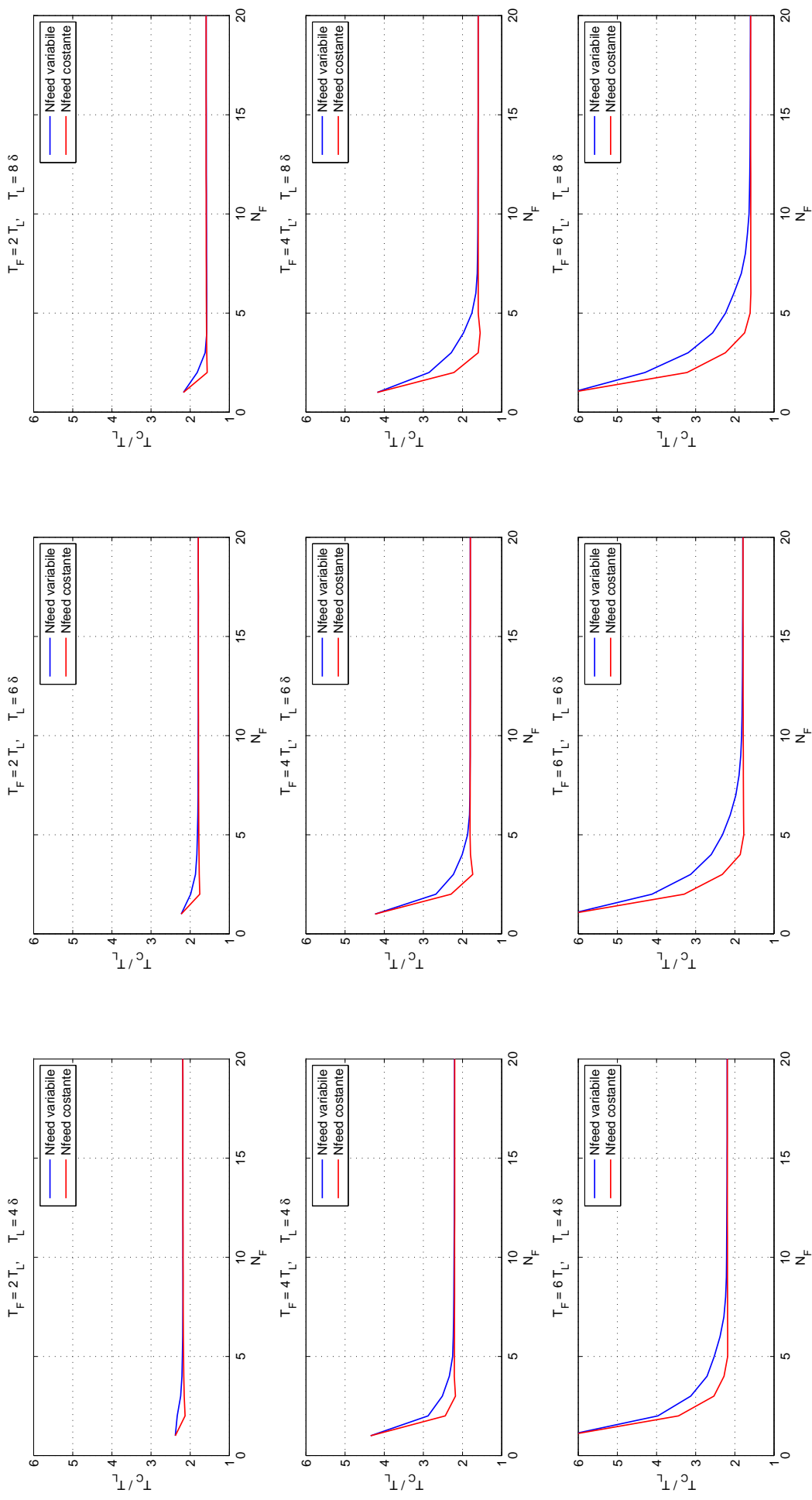


Figura 7.9: Risultati della simulazione con numero di componenti variabili ad ogni rinnovo feeder. Priorità di svuotamento assegnata al feeder. Dimensione del buffer imposta pari a 10.

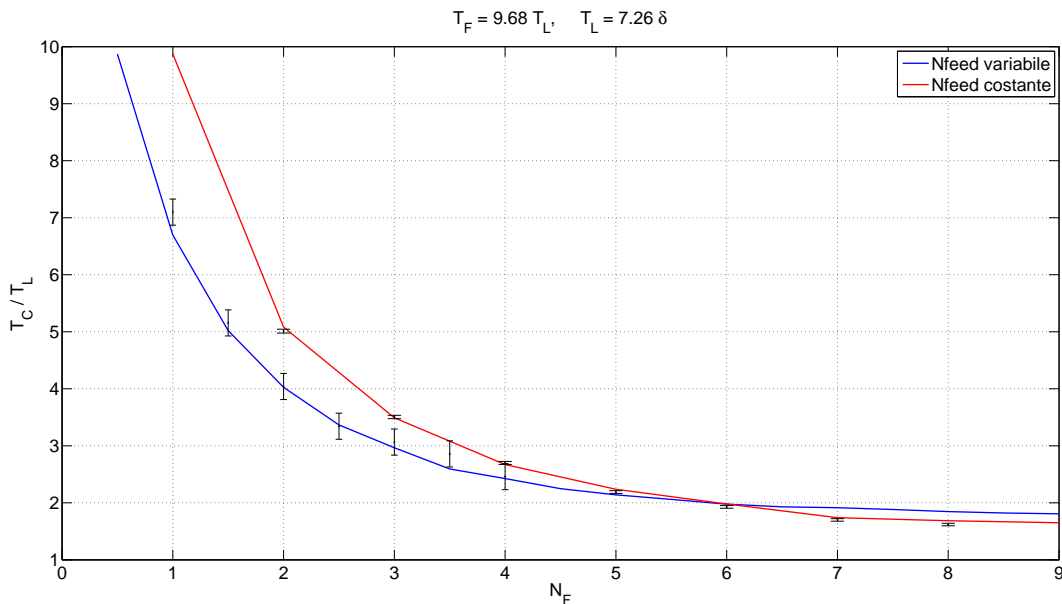


Figura 7.10: Risultati della simulazione con numero di componenti variabili ad ogni rinnovo feeder. Priorità di svuotamento assegnata al feeder. Dimensione del buffer imposta pari a 6.

sia il ciclo in esame, al termine di esso non vi è alcun accumulo di componenti nel buffer. Di conseguenza in entrambi i casi analizzati vi è attesa per il ricambio feeder. Nel caso a N_F variabile, tuttavia, tale attesa è funzione del numero di componenti presenti nel feeder in quel ciclo. Per valori superiori a quello descritto, invece, vi è accumulo nel buffer. La minima differenza fra il caso variabile e quello costante è data comunque da eventuali successioni di rinnovi feeder con un numero ridotto di componenti. L'accumulo di componenti nel buffer non è illimitato e, poichè la simulazione è realizzata con $N_B = 10$, sono sufficienti pochi cicli successivi con $N_F = 1$ per obbligare il manipolatore ad attendere in home il ricambio del feeder.

Si può, infine, osservare che la differenza fra le due curve rappresentate è tanto marcata quanto maggiore è il tempo di rinnovo feeder, mentre è poco influenzata da un aumento del tempo di lavorazione. Tale affermazione è in accordo con quanto esposto precedentemente: è il numero di componenti lavorabili durante l'attesa per il rinnovo del feeder che influisce su tale differenza. Questo compor-

tamento non si potrebbe osservare se il tempo feeder fosse riportato in valore assoluto anzichè in relazione al tempo di lavorazione.

7.3.1 Confronto con risultati sperimentali

Si riportano ora i risultati di un test effettuato per verificare l'attendibilità del modello con N_F variabile. La realizzazione di tale test si è rivelata più difficile del previsto. I risultati teorici presentati, infatti, sono stati ottenuti con una simulazione di 5000 rinnovi feeder per ciascun valore di N_F considerato. Una simulazione in cui tale valore assume un ordine di grandezza inferiore al migliaio non garantisce la convergenza al valore finale e, di conseguenza, i risultati grafici variano tra due medesime simulazioni. Tale inconveniente ha limitato la possibilità di verificare i risultati analitici nel caso di N_F variabile. Una prova sperimentale che preveda centinaia di rinnovi feeder contenenti in media una decina di componenti richiede un lasso di tempo pari ad una giornata di lavoro con il robot continuamente in funzione. Per questioni di sicurezza, in laboratorio non è consentito lasciare il manipolatore al lavoro senza sorveglianza. Inoltre si è preferito evitare di utilizzare intensivamente il manipolatore per questioni legate all'usura dello stesso. Il robot, infatti, non è di proprietà del dipartimento ma di un'azienda per la quale lo stesso sta effettuando una commissione.

Per i motivi esposti, i test effettuati per questo caso sono limitati alla verifica di una sola curva e a valori di N_F massimi inferiori alla decina. Questa situazione è in contrasto con quanto realizzato per gli altri aspetti in esame della tesi. In tali casi, infatti, il numero di prove effettuate è superiore a quelle riportate nel presente documento per questioni di spazio.

Si è preferito, inoltre, valutare l'attendibilità di una curva che rappresenta una situazione più semplice da verificare. Si è, infatti, considerato il caso di $N_B = 6$ e τ_F sufficientemente elevato in modo da ottenere $N_{out} \geq N_B$. Tale affermazione equivale a sostenere che al termine di ciascun ciclo feeder non vi è presenza di componenti nel buffer, indipendentemente dalla dimensione del feeder. Tale situazione è stata scelta appositamente per consentire di sostenere una prova con meno di 100 rinnovi feeder per ciascun valore da esaminare sapendo di ottenere un risultato apprezzabile. Nell'analisi dell'incertezza di tale prove è stata utilizzata

l'equazione 5.3 imponendo $N_F = 1$, considerando, cioè, il solo caso più gravoso. All'incertezza finale, inoltre, è stato arbitrariamente aggiunto un valore pari a $0.5s$ per considerare il numero limitato di prove effettuate.

I risultati di figura 7.10 sono riportati in un grafico differente da quelli presentati in figura 7.9. La curva con N_F variabile, infatti, è ora riportata ponendo sulle ascisse la media dei valori possibili assunti dal numero di componenti presenti nel feeder. Nel caso precedente, invece, sulle ascisse è riportato il valore massimo di tale parametro. La differente rappresentazione illustra, pertanto, i risultati a parità di media di N_F anzichè a parità di valore massimo. Le curve così rappresentate si intersecano in un punto e indicano con più semplicità quale caso risulta conveniente. Si è preferito, tuttavia, riportare le curve precedenti con la rappresentazione precedentemente esposta. Si è ricorso a tale scelta in quanto l'intersezione, per i casi riportati, è individuabile per valori di N_F poco superiori all'unità e, pertanto, difficilmente apprezzabili graficamente.

7.4 Ottimizzazione della sequenza di buffering

Si accenna alla possibilità di ottimizzare il processo di buffering nel caso di tempistiche deterministiche delle postazioni, modificando in parte la gestione dinamica della cella. Questa possibilità viene analizzata dal solo punto di vista teorico poichè richiede una gestione differente da quella prevista dall'apparato sperimentale allestito. Può verificarsi che l'operazione di inserimento componenti nel buffer avvenga interamente durante i primi cicli lavorazione e durante i restanti non vi sia il deposito di alcun componente a causa del raggiungimento della saturazione della postazione. É possibile limitare il numero di componenti immessi durante i primi cicli e distribuire il buffering sui successivi in modo da aumentare l'entità delle operazioni in tempo mascherato.

Tale limitazione non è compatibile con una gestione dinamica della cella basata su un ordine di priorità delle operazioni da effettuare. Tuttavia, se a tale priorità si potesse associare un limite di molteplicità per cui un'operazione può essere effettuata in sequenza, sarebbe possibile ottimizzare ulteriormente il tempo ciclo, pur mantenendo globalmente una gestione che preveda la risposta ai segnali

hardware. La limitazione andrebbe imposta solamente all'operazione di prelievo componente da feeder per il deposito nel buffer.

Lo scopo dell'operazione è aumentare γ_R diminuendo γ_A , in quanto nella formula del tempo ciclo (equazione 3.28) il primo termine sarebbe così moltiplicato per un valore minore ($T_{CR} < T_{CA}$). Si considera per semplicità il solo caso con priorità assegnata al feeder. Si vuole realizzare una bufferizzazione il più possibile in tempo mascherato depositando un numero di pezzi massimo $N_{BL} - 1$ ad ogni attesa lavorazione anziché N_{BL} . Questa ottimizzazione è possibile solo se il numero di componenti inseriti complessivamente nel buffer in ciascun *ciclo-feeder* è limitato da N_{out} o N_B . Se così non fosse, il numero di componenti inseriti ad ogni lavorazione sarebbe sempre N_{BL} . In tale situazione non vi è possibilità di ottimizzare il ciclo con le modalità che verranno ora esposte.

Il numero di componenti massimo per cui è possibile ottimizzare l'inserimento nel buffer è pari a γ_A . I componenti in questione, infatti, corrispondono all'ultimo pezzo depositato in ciascun ciclo di lavorazione. Si indica con N_{opt} gli slot disponibili per accogliere i γ_A componenti citati. La relazione fra i due parametri indica se sia possibile o meno effettuare l'ottimizzazione in esame.

- se $\lambda = 0$: $N_{opt} = \gamma_R(N_{in} - 1)$.
 - Se $N_{opt} \geq \gamma_A$ l'ottimizzazione è applicabile a tutti i componenti.

$$\gamma_R^* = \gamma_A + \gamma_R$$

$$\gamma_A^* = 0$$

- Nel caso contrario, invece, l'ottimizzazione riguarda solamente un numero di pezzi pari a N_{opt} :

$$\gamma_R^* = \gamma_R + N_{opt} = \gamma_R N_{in}$$

$$\gamma_A^* = \gamma_A - N_{opt}$$

- se $\lambda \neq 0$, il primo ciclo di durata T_{CR} prevede già una parziale bufferizzazione, il numero di slot disponibili ad accogliere pezzi si riduce: $N_{opt} = \gamma_R(N_{in} - 1) - (\lambda - 1)$.

– Se $N_{opt} \geq \gamma_A$:

$$\gamma_R^* = \gamma_A + \gamma_R$$

$$\gamma_A^* = 0$$

– Se $N_{opt} < \gamma_A$:

$$\gamma_R^* = \gamma_R + \gamma_R(N_{in} - 1) = \gamma_R N_{in}$$

$$\gamma_A^* = \gamma_A - \gamma_R(N_{in} - 1)$$

Conclusioni

In questa tesi sono stati sviluppati diversi modelli analitici allo scopo di stimare il tempo ciclo di lavoro di una cella robotizzata caratterizzata da un determinato layout. I modelli, utilizzando funzioni ricorsive, sono stati implementati in MATLAB in modo da poter disporre di algoritmi per calcolare la produttività in funzione dei parametri caratteristici della cella.

É stata allestita una cella di lavoro per valutare l'attendibilità di tali algoritmi e simulare il caso analizzato. Le tempistiche richieste dalle varie postazioni sono state gestite tramite relè temporizzati posizionati all'interno del quadro elettrico della cella. Le prove hanno fornito risultati compatibili con quelli derivanti dal modello analitico, tuttavia non è possibile affermare la completa verifica del modello quanto piuttosto la sua attendibilità per i soli valori utilizzati nelle prove sperimentali.

Uno dei possibili sviluppi di questo lavoro è rappresentato dall'analisi di differenti layout di cella robotizzata mantenendo la gestione dinamica di essa. É possibile includere ulteriori macchinari di lavorazione, disporre postazioni in parallelo per raddoppiarne la capacità e, in questi casi, valutare inoltre la convenienza di utilizzo di una pinza multipla come end-effector. L'analisi di tempistiche non deterministiche per le postazioni può rappresentare un ulteriore sviluppo dello studio.

In futuri sviluppi è tuttavia necessaria una gestione differente delle tempistiche delle postazioni. L'utilizzo di relè temporizzati si è rivelato insoddisfacente a causa della difficoltà di settaggio e della scarsa ripetibilità nelle prestazioni fornite. Si consiglia l'implementazione di una gestione software di tali tempistiche per ridurre le problematiche derivanti dalle prove sperimentali.

Appendice A

Notazione

In questa sezione è riportato il riepilogo dei simboli utilizzati nella tesi:

- τ_L - Tempo richiesto dal macchinario di lavorazione.
- τ_F - Tempo richiesto per il rinnovo del feeder flessibile.
- τ_S - Tempo richiesto per rinnovare la postazione di scarico.
- δ - Tempo per il movimento fra due postazioni adiacenti.
- η - Quota temporale di accelerazione/decelerazione relativa al movimento fra stazioni non adiacenti.
- ϵ - Tempo richiesto per effettuare la presa e il rilascio del componente;
- τ_H - Tempo per il movimento dalla posizione di home al macchinario di lavorazione;
- N_F - Numero di componenti disponibili nella postazione di feeder;
- N_B - Slot disponibili nella postazione di buffering.

Appendice B

Confronto SPEL+, PAC, V+

Per completare il discorso relativo alla programmazione di robot e alla differenza fra prodotti realizzati da aziende diverse vengono analizzati tre linguaggi di programmazione. Si desidera porre attenzione sulle differenze di utilizzo e le limitazioni relative alla definizione di locazioni e alle possibilità offerte dalla movimentazione del robot. Si rimanda ai manuali ufficiali rilasciati dai produttori per la descrizione dettagliata dei linguaggi di programmazione e la sintassi richiesta per ottenere un codice compilabile e correttamente eseguibile.

Nel laboratorio di robotica dell'Università, negli ultimi due decenni, sono stati utilizzati per ricerca robot prodotti dalle aziende Adept Technology, Denso Corporation ed Epson Seiko. Grazie alla disponibilità di manipolatori, software e manuali di questi tre differenti produttori e al know-how conseguito dal dipartimento negli anni, è stato possibile realizzare il confronto presente in questa appendice.

Il linguaggio implementato sui robot prodotti dall'azienda Adept Technology è il *V+*. I manipolatori prodotti dalla ditta Denso Corporation implementano invece il linguaggio *PAC* e, come già esposto, lo *SPEL+* è il linguaggio dei robot della Epson Seiko. Nelle prossime sezioni ci si sofferma ad esporre le differenti modalità di gestione delle locazioni e di movimentazione del robot per i tre codici proprietari in esame, avendo cura di indicare in particolar modo i vantaggi offerti da ciascun linguaggio.

Per completezza si informa che l'analisi del linguaggio Adept è effettuata su

manuali e software risalenti al 2005 a causa dell'assenza di acquisti più recenti nel laboratorio di robotica. Documentazioni e software più recenti vengono infatti forniti solo in occasioni di nuovi acquisti, in quanto eventuali software e linguaggi aggiornati richiedono anche una versione successiva del controller annesso al manipolatore.

B.1 Locazioni

Con il termine *locazione* si intende una variabile adibita a rappresentare una posa del manipolatore. Si tratta di variabili tipicamente utilizzate nei comandi di movimento per indicare le destinazioni richieste. Nei linguaggi di programmazione robot le locazioni possono essere istruite tramite coordinate cartesiane o di giunto. Una locazione fornita in coordinate di giunto consiste nell'insieme dei valori assunti da ciascun motore rispetto al proprio riferimento, tipicamente in gradi, e identifica un'univoca configurazione del manipolatore. Una locazione fornita in coordinate cartesiane, invece, consiste nell'indicazione delle coordinate dell'end effector in una terna di riferimento, non necessariamente quella utensile o quella relativa alla base robot. Annesse alle coordinate cartesiane, indicate in mm, vi sono le coordinate dell'orientazione desiderata per l'end effector riportate secondo la convenzione scelta dal produttore. Considerando manipolatori a 6 gradi di libertà con gli ultimi tre assi perpendicolari, rotoidali e concorrenti in un punto, caso tipico dei robot industriali, una posa indicata in coordinate cartesiane può essere raggiunta con 8 configurazioni differenti, identificabili in forma chiusa [28]. In particolare è possibile raggiungere una determinata posizione con due differenti configurazioni della mano o, secondo alcuni autori spalla, indicate con i termini *righty* e *lefty*. Per ciascuna di esse vi sono inoltre due possibili configurazioni del gomito, *above* o *below*, e del polso, *flip* o *no flip*. Una variabile di locazione definita in coordinate cartesiane contiene tipicamente anche informazioni riguardo alla configurazione desiderata. Gli aspetti citati, per quanto teoricamente generici e indipendenti dal robot in esame, sono gestiti in maniera differente dai differenti produttori come si avrà cura di esporre ora.

B.1.1 Epson *SPEL+*

Il software Epson RC+, interprete del linguaggio *SPEL+* e interfaccia fra utente e controller, rende disponibili all'utilizzatore le variabili P0 - P999 già dichiarate come locazioni. Indipendentemente dalla modalità di assegnazione dei valori della posa, ciascun punto P(i) contiene le coordinate dell'end effector nello spazio cartesiano, le indicazioni sulla configurazione assunta dal robot, un'etichetta e un'eventuale descrizione. Se l'assegnazione è stata effettuata in coordinate cartesiane e la configurazione non è specificata il software assegna automaticamente la posa Righty - Above - NoFlip: la configurazione salvata in una locazione può essere modificata in un momento successivo.

Tabella B.1: Struttura delle variabili di locazione in *SPEL+*

| Num | Lab | X | Y | Z | U | V | W | Loc | Hand | Elb | Wri | J4F | J6F | Desc |
|-----|-----|---|---|---|---|---|---|-----|------|-----|-----|-----|-----|------|
|-----|-----|---|---|---|---|---|---|-----|------|-----|-----|-----|-----|------|

É possibile assegnare un'etichetta ad una locazione, ovvero associare un nome alternativo alla numerazione imposta da *SPEL+*. L'assegnazione di etichette non impone variazioni alla numerazione P(i), di conseguenza una stessa locazione è richiamabile sia tramite l'eventuale relativa etichetta sia tramite la numerazione già esistente.

Ciascun progetto genera un file con estensione *.pts* contenente le informazioni relative ai punti utilizzati. E' consentito effettuare qualsiasi comune operazione tra file come creare una copia di salvataggio o importare un differente file *.pts*, operazioni eseguibili anche tramite comandi diretti. La GUI *Points manager* disponibile nel software EPSON RC+ consente inoltre di visualizzare e modificare il file in esame.

Il codice *SPEL+* impone l'utilizzo di uno dei punti P0 - P999 per definire una locazione. Nella stesura di codici con un utilizzo massivo di differenti locazioni la scelta del numero del punto a cui associare la nuova locazione può creare confusione, in quanto, anche lavorando con etichette, è necessario associare tale nome ad un punto specifico dell'elenco a disposizione.

I parametri J4Flag e J6Flag indicano le convenzioni di angolo assunte per le rotazioni dei giunti 4 e 6. Durante l'utilizzo di un robot a 6 assi è infatti

Figura B.1: SPEL+: Assegnazione della locazione P1 in coordinate cartesiane con configurazione lefty - below - flip. Assegnazione della locazione P2 in coordinate di giunto e imposizione di una configurazione lefty alla locazione con etichetta "esempio" .

```
P1 = XY(x,y,z,u,v,w) /L /B /F
P2 = JA(j1,j2,j3,j4,j5,j6)
PLABEL 3, "esempio"
HAND esempio, LEFTY
```

possibile ottenere una medesima posa e configurazione anche se uno di questi giunti è ruotato di 360 gradi. Per distinguere le due situazioni in *SPEL+* si sfruttano gli attributi JFlag. Un valore nullo di tale attributo indica che l'angolo del giunto in esame è compreso fra -180 e 180° . Il valore unitario indica invece $-360 < \alpha < -180^\circ$ o $180^\circ < \alpha < 360^\circ$. Il parametro è settato di default per ciascuna locazione pari a 0.

L'assegnazione delle coordinate nello spazio cartesiano avviene tramite il comando XY. La convenzione utilizzata per la terna (u, v, z) di rotazione corrisponde ad una prima rotazione attorno all'asse z locale (*roll*) seguita da una rotazione sull'asse y (*pitch*) e sull'asse x (*yaw*). JA impone invece l'assegnazione di coordinate espresse nello spazio dei giunti. Alcuni esempi sono riportati in figura B.1.

Sebbene in memoria ciascuna locazione sia gestita come un array di parametri in un preciso ordine la locazione in *SPEL+* non è direttamente manipolabile come un vettore. Non è possibile, ad esempio, inserire la definizione della locazione in un costrutto *for* specificando ad ogni ciclo una differente coordinata della posizione. In altre parole la sintassi $P1(i)$ non viene interpretata dal compilatore. Per trasformare una locazione in vettore è necessario utilizzare più comandi distinti come riportato in figura B.2. L'elenco di punti tuttavia è interpretabile come array, ovvero la sintassi $P(i)$ è ammessa.

Le operazioni consentite fra locazioni in *SPEL+* si limitano alla semplice som-

Figura B.2: SPEL+: Esempio di codice per trasformare una locazione in vettore. Movimento verso un punto posizionato +20 mm rispetto al punto P1 in coordinate world (x), -50mm rispetto alla terna utensile (z) e ad una quota pari a 150mm rispetto alla terna world (y)

```
vettore = [CX(P1), CY(P1), CZ(P1), CU(P1), CV(P1), CW(P1)]  
Go P1 +X(20) :Y(150) -TLZ(50)
```

ma. L'espressione $P3 = P1 + P2$ effettua automaticamente la somma delle coordinate cartesiane e degli angoli di rotazione. Anche se le locazioni sono state definite in coordinate di giunto la somma così espressa viene effettuata in coordinate cartesiane, in accordo con quanto esposto in merito alla modalità di salvataggio in memoria delle locazioni. E' prevista inoltre la possibilità di usare offsets relativi esprimibili con i comandi $\pm[X, Y, Z, U, V, W]$ per gli spostamenti rispetto alla terna base robot e $\pm[TLX, TLY, TLZ, TLU, TLV, TLW]$ per quelli rispetto alla terna utensile. Per operare su una locazione modificandone le coordinate assolute invece si utilizza la sintassi : $[X, Y, Z, U, V, W]$.

In *SPEL+* il calcolo di locazioni relative è indipendente dal modello di robot in uso. La nuova locazione mantiene le caratteristiche di quella che è stata elaborata per ottenerla. Quando un manipolatore a 6 assi effettua un movimento verso una locazione definita utilizzando offsets come appena descritto è possibile che la configurazione del polso finale non corrisponda a quella voluta. Si rivela pertanto particolarmente utile la modalità *least joint mode* implementata nei robot Epson che consente la gestione di tali inconvenienti. Ad esempio, il comando `G0 LJM(Here -Y(100))` comporta il calcolo della configurazione del punto finale che richiede la minore rotazione dei giunti. Vi è inoltre un'opzione attivabile nelle impostazioni del controller che consente di mantenere perennemente attiva la modalità LJM senza necessità di dichiararla ad ogni movimento (*AutoLJM*).

B.1.2 Denso PAC

Nel linguaggio *PAC* non vi sono variabili di locazione già impostate. Le locazioni richiedono la dichiarazione della variabile che ne conterrà i valori e questo avviene tramite tre differenti comandi poichè tale è il numero di tipologie di locazioni implementate in *PAC*. Nel relativo manuale ci si riferisce ad esse con i termini Type P, Type J e Type T.

Ciascuna è identificata da una sintassi differente nonostante indichino tutte una locazione.

Tabella B.2: Struttura delle variabili di locazione in *PAC*

| | | | | | | | | | | |
|--------|----|----|----|----|----|----|-----|----|----|-----|
| Type P | x | y | z | Rx | Ry | Rz | fig | | | |
| Type J | j1 | j2 | j3 | j4 | j5 | j6 | | | | |
| Type T | Px | Py | Pz | Ox | Oy | Oz | Ax | Ay | Az | fig |

Nella sintassi di alcuni comandi, ad esempio quelli di movimento, sono accettate solo alcune delle tipologie di locazione. Per questo motivo l'utilizzo di tre distinte tipologie richiede particolare attenzione da parte del programmatore. La conversione da una tipologia all'altra è comunque possibile utilizzando i comandi P2J, P2T, J2P, T2P, T2J.

La locazione di tipo P è quella rappresentata da parametri definiti nello spazio cartesiano. Ciascuna locazione di questa tipologia è costituita da tre parti: un vettore posizione, un vettore rotazione e un intero che rappresenta la configurazione della posa. Vi è la possibilità di modificare la locazione usando un determinato vettore di tre componenti per assegnare i parametri di posizione o rotazione. Il procedimento inverso, PVEC e RVEC, consente invece di trasformare automaticamente una locazione in un vettore.

L'intero finale rappresentativo della posa è posto pari al valore -1 se non diversamente specificato dall'utente. Tale valore indica la mancata assegnazione della posa, pertanto durante il moto verso tale locazione il robot ne assumerà una dipendente dalla locazione in cui è iniziato il movimento. Il comando LETF consente successivamente di specificare la sola configurazione voluta imponendo un valore tra 0 e 31. Le possibilità offerte sono 32, 2^5 , in accordo con quanto

Figura B.3: PAC: Dichiarazione di tre distinte tipologie di locazioni e esempio di assegnazione dei relativi valori.

```
DEFPOS loc_esempio
LET loc_esempio = (200, 200, 200, 0, 0, 0, -1)
DEFJNT loc_esempio2
LET loc_esempio2 = (10,0,80,5,10,90)
DEFTRN trans_esempio3
LET trans_esempio3 = (100,100,100,1,0,0,0,1,0,4)
```

esposto in merito alla posa in *SPEL+*. Tale soluzione consente di semplificarne la definizione purchè si disponga della tabella riassuntiva delle pose corrispondenti a ciascun intero. Con un solo valore è possibile specificare contemporaneamente la configurazione di polso, gomito, spalla e le convenzioni da utilizzare per i giunti 4 e 6. L'equivalente del JFlag presente in Epson è indicato in Denso con *axis figure*. Il parametro 4th-axis figure può assumere le connotazioni Single 4 e Double 4, e il parametro 6th-axis figure i valori Single - Double. Per single in tale contesto si intende $-180 < \alpha < 180^\circ$, con il termine double invece $-360 < \alpha < -180^\circ$ o $180^\circ < \alpha < 360^\circ$. Per completezza si fa notare che la configurazione destra-sinistra della posa del manipolatore è definita come configurazione della spalla anzichè della mano come in *SPEL+*.

La convenzione per la definizione delle rotazioni è indicata con Rx, Ry, Rz, ovvero rotazioni in sequenza attorno all'asse x, y e z.

La locazione di tipo J è costituita solamente dai valori assunti da ciascun giunto nel proprio sistema di riferimento. La posa è univocamente definita.

La terza possibilità offerta da *PAC* corrisponde alla trasformazione omogenea, o locazione di tipo T. Tale tipologia di variabile è costituita da un vettore posizione, un vettore orientazione e un vettore *approach*, ciascuno di tre componenti, a cui segue l'indicazione sulla figura desiderata. Il vettore di posizione è equivalente a quello esposto in precedenza e indica la posizione in coordinate cartesiane dell'end effector. Il vettore approach e quello di orientazione sono in-

vece versori che indicano rispettivamente l'orientazione dell'asse z e dell'asse y utensile. Il loro requisito, oltre alla necessità di essere normalizzati, è la reciproca perpendicolarità. In quanto trasformazione questa tipologia di locazione ben si presta ad essere utilizzata per effettuare operazioni fra locazioni. E' sufficiente l'utilizzo dell'operatore $[*]$ per moltiplicare due trasformazioni e ottenerne una terza. (Es. $T3 = T1 * T2$).

L'alternativa offerta da *PAC* per effettuare operazioni su locazioni è come il *SPEL+* l'utilizzo di offset. Per esprimere delle locazioni dipendenti da quelle già definite è sufficiente utilizzare l'operatore $+$ ed effettuare un'operazione vettoriale. Con il simbolo H al termine del vettore si indica che l'offset è relativo alle coordinate nel sistema utensile. É inoltre possibile effettuare somme tra locazioni purchè esse corrispondano alla medesima tipologia di locazione.

Risulta particolarmente utile la possibilità di dichiarare qualsiasi tipologia di variabile come array, locazioni comprese.

B.1.3 Adept V+

V+ è un linguaggio che non richiede la dichiarazione di variabili. Il compilatore assegna automaticamente la tipologia adatta in base all'utilizzo nel codice stesso. É consentito comunque dichiarare le variabili se vi sono esigenze particolari di allocazione di memoria o necessità di limitazione nel numero di variabili globali. Di default infatti tutte le variabili sono considerate globali, in contrasto con le regole di buon senso per una corretta programmazione che consigliano l'utilizzo di esse solo quando strettamente necessario. Anche le locazioni, in quanto variabili, rispecchiano quanto appena esposto e non richiedono allocazione preliminare di memoria.

Tabella B.3: Struttura delle locazioni in *V+*

| | | | | | | |
|-----------------|----|----|----|----|----|-----|
| Transformation | x | y | z | Rz | Ry | Rz' |
| Precision point | j1 | j2 | j3 | j4 | j5 | j6 |

Vi sono due distinte tipologie di locazioni: trasformazioni e *precision points*. Le trasformazioni sono formate da sei componenti che rappresentano la posizione

Figura B.4: V+: Definizione di una locazione di tipo *transformation* e attribuzione della posizione corrente ad un *precision point*.

```
SET esempio = TRANS(150, 200, 180, 50, 0, 20)
HERE #esempio2
```

nello spazio cartesiano e l'orientazione richiesta in quella posizione. Le prime tre componenti identificano nello spazio cartesiano l'origine della nuova terna di riferimento rispetto a quella di partenza. Le tre successive componenti identificano invece le rotazioni necessarie per ottenere la terna desiderata. La convenzione utilizzata da V+ corrisponde agli angoli di Eulero ZYZ': la successione delle rotazioni corrisponde alla sequenza *yaw*, *pitch* e *roll*. La prima è una rotazione attorno all'asse Z, la seconda attorno all'asse Y e l'ultima nuovamente attorno all'asse Z. Per come è definita, questa tipologia di locazione consente operazioni fra locazioni semplificate e flessibili.

Un *precision point* è costituito dai valori assunti da ciascun giunto nelle proprie coordinate. Sono coordinate assolute, misurate rispetto allo zero del dispositivo di rotazione/traslazione. Per tale motivo non è consentito definire movimenti relativi o effettuare operazioni fra due *precision points*. L'utilizzo di un *precision point* richiede l'utilizzo del simbolo # prima del nome della variabile (figura B.4).

Il vantaggio di lavorare prevalentemente con locazioni di trasformazione è senza dubbio l'efficienza e la semplicità delle operazioni fra locazioni. L'operatore [:] è utilizzato per effettuare un prodotto fra matrici ed è consentito effettuare prodotti fra più di due trasformazioni. La destinazione del moto MOVE loc_1:loc_2:loc_3, ad esempio, corrisponde al prodotto delle tre matrici di trasformazione. Il comando INVERSE inoltre fornisce la trasformazione inversa rispetto a quella indicata e semplifica eventuali operazioni successive. L'impostazione del linguaggio favorisce l'utilizzo di locazioni relative e la combinazione fra di loro rispetto a SPEL+ e PAC.

La gestione delle locazioni è ulteriormente semplificata tramite il comando SHIFT che consente di ottenere una nuova trasformazione imponendo la sola tra-

slazione di una già definita. **SCALE** consente inoltre di scalare le componenti di posizione di un fattore specificato. La combinazione fra di loro dei comandi citati permette indubbiamente la maggiore flessibilità nelle operazioni fra locazioni differenti rispetto agli altri linguaggi analizzati.

Uno dei principali svantaggi di questo metodo di gestione delle locazioni è la difficoltà di controllo delle pose assunte dal manipolatore. In *SPEL+* e *PAC* a ciascuna locazione è associata una determinata figura, ovvero un'univoca configurazione del robot. Operando con *V+* tale possibilità viene a meno, non c'è modo di associare ad una trasformazione una figura desiderata. La configurazione assunta dal manipolatore in *V+* non è infatti associata alla locazione ma al movimento. E' una concezione notevolmente differente rispetto agli altri linguaggi in cui i comandi di gestione della posa richiedevano nella loro sintassi l'identificazione di una locazione. In *V+* sono disponibili invece dei comandi che specificano la configurazione richiesta al termine del successivo movimento: **ABOVE/BELOW**, **LEFTY/RIGHTY** e **FLIP/NOFLIP**. L'utilizzo di precision points evita tale problema in quanto la configurazione del manipolatore è definita univocamente dai valori assunti dalle coordinate di giunto.

La decomposizione di una locazione in un vettore è implementata tramite il comando **DECOMPOSE** ed è concesso specificare anche la posizione dell'array di destinazione in cui inserire i valori.

B.2 Comandi di movimento

Come già accennato, il movimento che può compiere un manipolatore per raggiungere una locazione, a parità di posizione di partenza, non è unico. Non vi è un solo motore in movimento, il moto dell'end effector è determinato dal moto di distinti motori che lavorano su assi differenti. Il moto lineare nello spazio cartesiano dell'end effector non è normalmente quello che consente le tempistiche minori, nonostante il percorso lineare fra due punti sia quello di lunghezza minore. Per mantenere tale traiettoria ciascun motore è costretto infatti ad effettuare rotazioni superiori a quelle strettamente indispensabili. Questa tipologia di moto è comunque contemplata dai differenti produttori in quanto utile per operazioni

che richiedono linearità cartesiana, come la saldatura di un componente. A livello di controllo un moto di questo tipo risulta però notevolmente complesso e richiede elevate tempistiche computazionali.

Il movimento più rapido da effettuare per un manipolatore è invece quello che minimizza le rotazioni di ciascun motore. Il sistema di controllo, tramite analisi cinematica inversa, calcola le coordinate dei giunti nella posizione finale e associa una legge primitiva del moto per ciascuno di essi. Tali leggi vengono in seguito scalate temporalmente rispetto al motore che richiede il più elevato *tempo minimo* di azionamento. Il tempo minimo di azionamento dipende dalla rotazione complessiva richiesta al motore e dalle sue caratteristiche. Le tempistiche richieste sono le minime possibili, tuttavia il moto dell'end effector risultante è di difficile previsione. Tale tipologia di moto è nota come *point-to-point motion*.

In ambito robotico industriale è ampiamente utilizzato un moto combinato delle due modalità descritte. Tale soluzione è imposta dalla necessità di allontanarsi dalla posizione di partenza con moto lineare lungo l'asse z utensile per evitare collisioni. Un discorso analogo vale per il raggiungimento della posizione di arrivo desiderata. Il moto risultante è composto da un allontanamento lineare iniziale (depart), un moto point-to-point veloce intermedio e un avvicinamento lineare (approach) fino al raggiungimento della destinazione imposta. Esso è noto come *gate motion*.

Tale moto è ottenibile tramite combinazione dei movimenti base definiti precedentemente tuttavia, vista la frequenza di utilizzo in ambito industriale, disporre del relativo comando già implementato nel linguaggio di programmazione semplifica la stesura del codice al programmatore. È questo il caso del linguaggio *SPEL+* come si avrà cura di esporre nei prossimi paragrafi.

Un'ulteriore tipologia di moto ampiamente utilizzata nel lavoro con i manipolatori è il passaggio attraverso punti di via. Vi possono essere dei movimenti imposti al manipolatore dettati dalla sola necessità di evitare la collisione con un ingombro interno alla cella. Tali movimenti non richiedono il passaggio esatto per la destinazione indicata ma è richiesto il solo avvicinamento ad un punto, scelto spesso con ampia approssimazione. A livello pratico tale scopo viene raggiunto raccordando le traiettorie del movimento verso il punto di via e da lì verso quello

successivo. Il moto risultante è indicato con il termine *continuous path*.

Le tipologie di movimentazione implementate dai produttori di robot sono quelle precedentemente esposte precedentemente e rappresentate graficamente in figura B.5. Variano tuttavia le opzioni che possono essere specificate nei comandi di moto e la flessibilità di utilizzo dei relativi comandi.

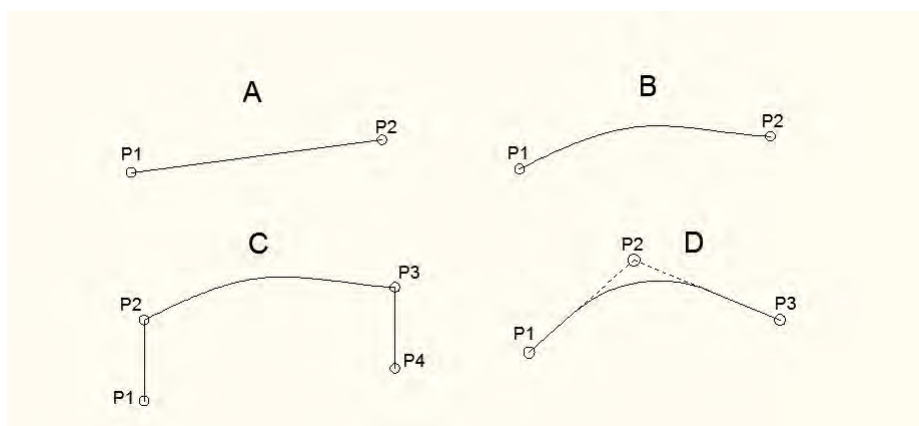


Figura B.5: Rappresentazione grafica delle tipologie di moti implementate nei linguaggi di programmazione robot. A: linear, B: point-to-point, C: gate motion, D: punti di via.

B.2.1 *SPEL+* Language

Il movimento point-to-point è ottenuto tramite il comando **GO**, quello con interpolazione lineare nello spazio cartesiano invece con il comando **MOVE**. L'opportunità più vantaggiosa offerta da *SPEL+* è la presenza di tre distinti comandi che consentono di realizzare il gate motion. Sono comandi implementati nel solo linguaggio *SPEL+* e consentono di semplificare notevolmente la stesura del codice per la movimentazione. Jump è il nome utilizzato indicare il moto in esame.

Il comando **JUMP** è previsto solo per i manipolatori a 4 assi. Essendo questi robot di tipo SCARA il movimento lungo z è gestito da un solo asse e il controllo risulta semplificato: i moti di approach e depart vengono effettuati controllando un solo motore e trascurando i restanti, consentendo tempi di esecuzione ridotti. Il movimento intermedio avviene ad una quota z specificata precedentemente ed

è di tipo PTP. È sufficiente indicare la quota z a cui effettuare il moto intermedio poichè il controller gestisce automaticamente il calcolo dei punti intermedi. Nel caso di manipolatori a 6 gradi di libertà i comandi analoghi sono invece **JUMP3** e **JUMP3CP**. I movimenti di approach e depart possono avvenire in qualsiasi direzione nello spazio e il movimento intermedio è di tipo PTP nel caso di **Jump3** e interpolato nel caso del **Jump3CP**. È necessario indicare entrambi i punti intermedi in questi casi vista l'assenza di limitazioni sulla destinazione dei moti lineari.

Alquanto pratica la possibilità di specificare il parametro *arch* durante un qualsiasi moto di tipo **Jump**. Tale approccio consente di raccordare i tratti lineari con quello intermedio specificando l'entità del raccordo: quello per il tratto di approach può essere differente da quello di depart. L'utilizzo dell'opzione *arch* è analogo a considerare i punti intermedi del jump come punti di via, la sua utilità consiste nel poter effettuare il movimento intermedio ad una quota senza imporre che i movimenti lineari, lenti, avvengano fino al raggiungimento di essa.

I singoli comandi di movimento in *SPEL+* consentono vaste opportunità di controllo del moto grazie ai numerosi argomenti facoltativi in coda allo stesso. È possibile ad esempio imporre operazioni di I/O parallele al movimento o, prima di iniziare il depart, valutare se completare il moto o meno a seconda dell'esito di una condizione specificata. La gestione dell'accuratezza con cui la posizione finale è considerata raggiunta non può tuttavia essere espressa all'interno del movimento ma richiede un comando apposito (**FINE**).

La gestione di un numero limitato di punti di via può essere effettuata aggiungendo la dichiarazione **CP** ai singoli comandi di moto o con il comando **PASS**. Se il numero di punti è elevato è consigliato utilizzare la funzione di moto curvilinea. Specificando i punti in esame viene creato un file **.CVT** contenente indicazioni sulla traiettoria da seguire. La curva desiderata viene calcolata una sola volta dal controller ed è sufficiente richiamarla in seguito con **CVMOVE**. È interessante la possibilità di indicare la necessità di ottenere una curva chiusa, ovvero includere nell'interpolazione il ritorno al punto di partenza. E' possibile inoltre specificare quanti assi controllare durante la curva: 2, 3, 4 o 6: gli assi non selezionati mantengono la posizione degli encoder precedente al moto.

B.2.2 PAC Language

Il comando a disposizione dell'utente per effettuare i movimenti si limita alla sola sintassi MOVE: la tipologia di movimento viene specificata come argomento del comando stesso. Per il movimento PTP la sintassi corretta è MOVE P, per il movimento lineare nello spazio cartesiano invece è MOVE L. L'equivalente del Jump in *SPEL+* non è presente in questo linguaggio e per effettuare un gate motion è necessario concatenare più movimenti distinti.

Vi sono tuttavia dei comandi di moto relativo che consentono di semplificare l'operazione, per quanto comunque laboriosa, e permettono di non calcolare esplicitamente i punti intermedi del gate motion. Si tratta dei comandi DEPART e APPROACH, ciascuno con l'eventuale attributo L o P specificato per definire la tipologia di moto.

In *PAC* è presenta una libreria di funzioni già implementate che possono essere richiamate dall'utente. In essa è presente un moto simile al jump, utilizzabile esclusivamente con robot a 4 assi che tuttavia consente minima flessibilità. Le poche opzioni specificabili, le lunghezze di depart e approach minime richieste, necessitano inoltre l'utilizzo di un altro comando richiamato dalla libreria.

Un vantaggio nella gestione della movimentazione in *PAC* è la possibilità di specificare direttamente nel comando di movimentazione quale sia la condizione di conferma del raggiungimento della posizione finale. E' possibile identificare tre differenti tipologie di condizioni e queste vengono indicate con @E, @P o @0. Il controllo encoder @E considera il movimento terminato quando i valori delle tacche encoder rientrano in un range specificato rispetto a quelli relative alla posizione desiderata. Si richiede in altre parole il passaggio per il punto specificato, indicandone eventualmente il range di accuratezza consentito. Il pass motion @P impone un moto che passi nelle sole vicinanze del punto specificato, considerando la destinazione come punto di via. @0 considera il moto terminato se il sistema di controllo valuta raggiunta la destinazione finale, quando cioè le oscillazioni di assestamento diventano trascurabili.

La gestione di un numero elevato di punti di via avviene tramite generazione (SETSPLINEPOINT) di una *free-curve*, analogamente al caso Epson, richiamata con la sintassi MOVE S. E' possibile specificare fino a 20 diverse free curves con 200

punti ciascuna.

B.2.3 Adept - V+ Language

Dal punto di vista dei comandi di movimento *V+* è un linguaggio estremamente semplice. Tutti i comandi hanno una sintassi breve e accettano come argomenti la sola destinazione o al più un valore numerico. Da un punto di vista della leggibilità del codice è sicuramente un vantaggio poichè esso presenta righe brevi e facilmente comprensibili. Tuttavia nella stesura di movimenti complessi il numero di righe di codice richieste diventa non trascurabile. Qualsiasi comando associato al movimento, come ad esempio la richiesta del continuous path o dell'accuratezza nel raggiungimento della destinazione, richiede una riga di comando a parte e ha valenza solo per il movimento successivo se non diversamente specificato.

I comandi messi a disposizione dell'utente per i movimenti point-to-point sono MOVE, APPRO e DEPART. Appro e Depart sono analoghi agli omonimi comandi in *PAC*. Aggiungendo una S ai comandi sopracitati si ottiene il rispettivo movimento con interpolazione lineare.

Curiosamente *V+* implementa il forward processing per tutte le operazioni di movimento: i comandi successivi a quello di moto vengono processati non appena comincia il movimento. È necessario utilizzare un apposito comando di interruzione (BREAK) per impedire il forward processing se non desiderato. E' una concezione opposta a quanto presente negli altri linguaggi in cui, se richiesto, è necessario dichiararne l'utilizzo.

Analogamente al codice *PAC* non sono presenti comandi di movimento composti per i robot a 6 assi. MOVEF/MOVESF identifica un movimento simile al jump ma è utilizzabile esclusivamente con i robot a 4 assi e consente relativa flessibilità. È permesso ad esempio specificare la distanza minima da compiere lungo il solo asse z nei movimenti verticali e la percentuale di coppia massima ammessa durante ciascuno dei tre movimenti. Uno svantaggio di tale comando è dato dalla necessità di utilizzare un utility program fornito dal produttore prima di poterne fare uso. Tramite dei brevi movimenti prestabiliti esso calcola, a seconda della risposta dinamica del sistema, alcuni coefficienti dinamici con cui effettuare il moto composto. Senza aver precedentemente eseguito l'utility il comando non

Figura B.6: V+: Moto per punti di via utilizzando un array di locazioni.

```
CPON ALWAYS
FOR i = 0 TO LAST(traiettoria[])
MOVE traiettoria[i]
END
```

è interpretabile. E' uno svantaggio non indifferente poichè, nonostante sia consentita la memorizzazione di più modelli dinamici, ad ogni cambio di payload o utensile non registrati è necessaria una nuova esecuzione dell'utility.

Le possibilità offerte da *V+* per la gestione dei punti di via sono minime e si limitano all'utilizzo del comando *CPON/CPOFF* con cui si abilita o disattiva il continuous path per il successivo movimento. É comunque possibile ricorrere ad array di locazioni che consentono di compattare il moto come rappresentato in figura B.6, sebbene la creazione di siffatto array possa non risultare immediata per un numero elevato di punti di via.

Un'interessante possibilità a disposizione dell'utente è determinata dall'*Alter Mode*, indicato anche con *real-time path-modification*. Questo consente di realizzare un'intera traiettoria, purchè realizzata in modalità lineare, con un prefissato spostamento rispetto a quella nominale.

Un'altra particolare funzione proposta da *V+* riguarda l'imposizione di una durata minima per un movimento. Il valore del comando è impostato di default pari a 0 e l'utente può settarlo a seconda delle esigenze.

Appendice C

Programmazione in Epson

In questa appendice sono riportate alcune funzioni offerte dal linguaggio *SPEL+* implementate nel codice utilizzato per effettuare i test sperimentali. La presente sezione si pone lo scopo di completare l'analisi introdotta in sezione 5.3, illustrando alcuni aspetti avanzati offerti da Epson e utilizzati in laboratorio. Il codice è infatti stato redatto per uno scopo più variegato delle semplici prove incluse nella presente tesi. Essendo un completamento del testo principale si ritiene corretto inserire la presente sezione in appendice.

C.1 Multitasking

I linguaggi di programmazione robot consentono l'utilizzo di più task attivi contemporaneamente. Tale possibilità è tipicamente utilizzata per la gestione di device differenti, uno per ogni processo parallelo. La gestione della pinza, ad esempio, è comunemente gestita da un task differente rispetto a quello relativo al sistema di visione. Il linguaggio *SPEL+* implementa un multitasking in cui ciascun processo può avere differenti privilegi, priorità e limitazioni a seconda delle esigenze dell'utente. Se non diversamente specificato i processi vengono eseguiti con tipologia normale. La loro interruzione avviene in caso di utilizzo di uno degli stop fisici di emergenza presenti all'esterno della cella, se il controller riscontra un errore, se le porte della cella vengono aperte o se l'operatore termina l'esecuzione del programma.

Risulta di indubbia utilità la possibilità di disporre di processi in esecuzione anche in seguito ad alcune delle situazioni citate. Questi processi per questioni di sicurezza e rispetto delle normative limitano l'utilizzo dei comandi impedendo, ad esempio, l'implementazione di qualsiasi comando di movimento. I task indicati come tipologia *No Emergency Abort* consentono la non interruzione del processo in seguito ad emergenze. Sono task pensati per la gestione degli I/O senza necessità di comunicazione con dispositivi esterni. Tra le tipologie di processi implementati in *SPEL+* le maggiori opportunità pratiche sono offerte dai task in background.

C.1.1 Background tasks

Uno dei principali limiti dei task *No Emergency Abort* citati è la loro interruzione nel caso di utilizzo della teach pendant. Quest'ultima, simile ad un joystick, è lo strumento utilizzato dagli operatori per entrare nella cella e impartire comandi al robot. E' utilizzata principalmente per istruire manualmente al robot nuove locazioni di presa. Quando l'operatore è all'interno della cella il controller limita la potenza fornita ai motori mantenendo velocità e accelerazioni al di sotto di valori pericolosi per l'operatore stabiliti in normativa.

Per sopperire all'inconveniente dei processi terminati dalla teach pendant in *SPEL+* sono disponibili, dalla versione 7.0 di Epson RC+, i task in background. Questi sono processi non terminabili finchè il controller è attivo. Un operatore non è consapevole della presenza di task in background che non vengono interrotti dalla pressione degli stop di emergenza o dai segnali di "Pause" o "Abort". Di conseguenza possono essere considerati come se fossero parte del sistema piuttosto che processi paralleli del codice. Per poter essere utilizzati, vista la loro peculiarità, richiedono l'attivazione manuale nelle impostazioni del controller in cui vengono eseguiti.

Applicazione pratica

In laboratorio i task in background sono stati utilizzati per consentire l'acquisizione di punti manualmente tramite teach pendant senza dover interrompere il programma in esecuzione e, in seguito, riavviarlo. Si desidera consentire all'ope-

ratore l'acquisizione di nuove locazioni tramite teach pendant entrando nella cella senza successivamente dover operare al computer per riavviare l'intera sequenza di inizializzazione. Nel momento in cui viene azionata la chiave di utilizzo della teach pendant infatti tutti i task eccetto quelli in background cessano di operare. Il task in background è stato implementato in modo da fornire indicazioni sullo schermo della teach pendant e guidare una corretta procedura di acquisizione punti. Al termine di questa esso rileva lo switch della chiave sulla teach pendant e la chiusura delle porte della cella. In seguito inizializza nuovamente i task principali di tipologia normale senza necessità di intervento dell'operatore.

C.2 Sense, Parallel statements e CP

I comandi di movimento in *SPEL+* consentono la specifica di impostazioni avanzate come ad esempio l'esecuzione di comandi in parallelo o il completamento del moto a seconda della verifica o meno di una condizione specificata. In questa sezione vengono analizzati i principali attributi delle funzioni di movimentazione e la loro influenza sulle tempistiche di lavoro.

C.2.1 Sense

L'attributo Sense è limitato ai soli movimenti che comprendono approach e depart, ovvero Jump, Jump3 e Jump3CP. Tramite questa opzione è possibile specificare una condizione che richiede verifica nel momento in cui il robot è in procinto di iniziare l'approach verso la destinazione finale. Il momento in cui avviene la valutazione della condizione è il termine del tratto orizzontale del moto composto. Se essa è soddisfatta, il robot termina il movimento senza effettuare l'approach. Il moto in questo caso viene considerato terminato nonostante non venga raggiunta la locazione indicata. Nel caso in cui invece la condizione specificata non sia soddisfatta il manipolatore termina il movimento come specificato dal comando ed il moto è considerato terminato quando viene raggiunta la posizione di destinazione.

Le possibilità che vengono offerte al programmatore con tale comando sono notevoli e consentono di velocizzare le operazioni del robot in alcuni scenari. Si supponga di dover afferrare un componente da un feeder flessibile. E' possibile

in un qualsiasi momento inviare al manipolatore un comando di movimento che preveda l'afferraggio di un pezzo da questa postazione indipendentemente dalla presenza o meno di componenti. Utilizzando in modo appropriato il comando Sense il robot raggiungerà la destinazione come richiesto solo nel caso in cui, prima di effettuare l'approach, il macchinario preposto alla gestione della postazione abbia inviato il segnale di disponibilità. In caso contrario il movimento terminerà senza effettuare l'approach, consentendo il ricambio della stazione e il processamento di altri comandi. Il comando JS consente di verificare se il Sense dell'ultimo movimento è stato verificato o meno. Con tale informazione è possibile instradare correttamente la scelta dei successivi comandi da fornire al manipolatore.

C.2.2 Parallel statements

I parallel statements, o parallel processing, consentono di indicare dei comandi da eseguire contemporaneamente ad un movimento specificandone l'istante esecuzione. Il momento viene specificato rispetto alla percentuale di distanza percorsa del movimento totale. Le opzioni consentite sono limitate ad una lista di comandi principalmente riguardanti la gestione degli I/O. Il comando `G0 P1 !D10, On 2, D90, Off 3!` ad esempio consente di alzare l'output numero 2 nel momento in cui la percentuale di distanza percorsa nel movimento verso P1 è pari al 10%. Inoltre raggiunta il 90% della traiettoria viene abbassato il segnale dell'output numero 3.

Se in seguito al termine di un movimento non sono stati completati tutti i comandi in parallelo, l'esecuzione del programma è sospesa fino al completamento degli stessi. Nel manuale Epson si riporta che tale situazione avviene quando il moto copre una ridotta distanza e vi sono numerosi comandi in parallelo. Tuttavia a livello pratico si è rilevato che l'utilizzo di parallel statements, in particolare se combinato con CP e/o Sense, determina una tempistica di movimentazione notevolmente superiore anche nei casi in cui il movimento sia tutt'altro che breve.

Figura C.1: Estratto di codice utilizzato per effettuare il test sull'influenza delle opzioni di moto sulle tempistiche finali.

```
Go P1
WaitPos
Wait (0.1)
ResetElapsedTime
Jump3CP P3, P4, P2 CP Sense ! D0; MemOn (0); D100; MemOn (1) !
WaitPos
Print #30, ElapsedTime, Chr$(10)
MemOff (0)
MemOff (1)
```

C.2.3 Influenza sui tempi di moto

Per i motivi illustrati è stato realizzato un test per valutare l'influenza dei comandi CP, Sense e i parallel statements sulle tempistiche dei movimenti. L'opzione CP è stata analizzata precedentemente e consente il passaggio per un punto di via. Vengono ora presentati i risultati di un test effettuato in laboratorio in cui un medesimo movimento è stato realizzato con differenti condizioni imposte nel comando principale. In figura C.1 si riporta un estratto del codice utilizzato per illustrare la modalità di esecuzione delle prove.

I risultati per il caso Jump3 sono riportati in tabella C.1.

Si può notare come il comando CP, nonostante il movimento sia singolo e dunque non venga utilizzata la relativa opzione di raccordo, abbia influenza sulle tempistiche di moto. In altre parole se non segue un ulteriore movimento a quello in esame il termine CP viene comunque computato dal controller. Si assiste così ad un aumento di tempo ingiustificato.

Dai risultati si può stimare l'influenza del comando Sense pari a 3 ms e quella dei parallel statements pari a 9 ms. È interessante notare come l'utilizzo del parallel processing D0 o D100 ha una limitata influenza sulla tempistica rilevata, nonostante il primo venga effettuato all'inizio del moto e l'ultimo al termine di

Tabella C.1: Risultati del test sull'influenza delle opzioni delle opzioni aggiuntive nel comando di movimento JUMP3

| <i>Speed</i> | | 20% | 40% | 60% | 80% | 100% |
|-------------------------|------------|-------|-------|-------|-------|-------|
| - | Media [s] | 2.256 | 1.301 | 0.977 | 0.813 | 0.714 |
| | StDev [ms] | 0.4 | 1.3 | 1.4 | 1.7 | 1.1 |
| <i>CP</i> | Media [s] | 2.313 | 1.357 | 1.032 | 0.862 | 0.753 |
| | StDev [ms] | 1.2 | 0.5 | 2.2 | 1.8 | 1.0 |
| <i>D0</i> | Media [s] | 2.264 | 1.308 | 0.985 | 0.819 | 0.723 |
| | StDev [ms] | 1.6 | 2.0 | 2.5 | 1.8 | 1.1 |
| <i>D100</i> | Media [s] | 2.263 | 1.307 | 0.984 | 0.820 | 0.722 |
| | StDev [ms] | 0.5 | 1.1 | 3.0 | 1.7 | 0.0 |
| <i>D0 D100</i> | Media [s] | 2.263 | 1.308 | 0.983 | 0.820 | 0.722 |
| | StDev [ms] | 0.5 | 2.2 | 1.1 | 0.9 | 0.9 |
| <i>D0 D100 CP</i> | Media [s] | 2.320 | 1.364 | 1.038 | 0.868 | 0.760 |
| | StDev [ms] | 0.9 | 1.5 | 1.8 | 1.7 | 1.7 |
| <i>SENSE</i> | Media [s] | 2.260 | 1.304 | 0.980 | 0.817 | 0.720 |
| | StDev [ms] | 1.1 | 1.1 | 1.4 | 1.5 | 1.6 |
| <i>CP SENSE</i> | Media [s] | 2.316 | 1.360 | 1.033 | 0.864 | 0.756 |
| | StDev [ms] | 0.5 | 1.1 | 1.0 | 2.1 | 2.2 |
| <i>D0 SENSE</i> | Media [s] | 2.267 | 1.311 | 0.987 | 0.824 | 0.726 |
| | StDev [ms] | 0.8 | 1.3 | 1.0 | 1.5 | 1.7 |
| <i>D100 SENSE</i> | Media [s] | 2.268 | 1.311 | 0.987 | 0.823 | 0.724 |
| | StDev [ms] | 0.5 | 1.3 | 1.1 | 1.3 | 0.9 |
| <i>D0 D100 SENSE</i> | Media [s] | 2.267 | 1.311 | 0.987 | 0.824 | 0.725 |
| | StDev [ms] | 0.8 | 1.3 | 1.1 | 1.7 | 1.1 |
| <i>D0 D100 SENSE CP</i> | Media [s] | 2.321 | 1.366 | 1.039 | 0.870 | 0.761 |
| | StDev [ms] | 0.8 | 1.1 | 1.1 | 2.5 | 0.9 |

esso. Il tempo richiesto non dipende pertanto dall'esecuzione dello switch sul segnale quanto alla computazione del processo parallelo.

A velocità elevate le tempistiche richieste per le opzioni di moto sono le medesime, tuttavia hanno maggiore influenza sul tempo totale in quanto il movimento è più veloce.

Bibliografia

- [1] M. Fabris, “Ottimizzazione di cella robotizzata flessibile,” Tesi di Laurea in Ingegneria Meccanica, Università degli Studi di Padova, 2012-2013.
- [2] M. Dawande, H. Geismar, S. Sethi, and C. Sriskandarajah, “Sequencing and scheduling in robotic cells: recent developments,” *Journal of Scheduling*, pp. 387–426, 8 2005.
- [3] Y. Crama, V. Kats, J. V. de Klundert, and E. Levner, “Cyclic scheduling in robotic flowshops,” *Annals of Operations Research*, pp. 97–124, 2000.
- [4] S. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, and W. Kubiak, “Sequencing of parts and robot moves in a robotic cell,” *International Journal of Flexible Manufacturing Systems*, vol. 4, pp. 331–358, 1992.
- [5] N. Brauner, G. Finke, and C. Gueguen, “Optimal cyclic production in robotic flowshops with buffer,” *Technical Report RR982-I, Laboratoire Leibniz, Institut IMAG, Grenoble, France*, 1997.
- [6] I. G. Drobouchevitch, S. Sethi, and C. Sriskandarajah, “Scheduling dual gripper robot cell: One-unit cycles,” *European Journal of Operational Research*, vol. 171, pp. 598–631, 2006.
- [7] I. G. Drobouchevitch, H. N. Geismar, and C. Sriskandarajah, “Throughput optimization in robotic cells with input and output machine buffers: A comparative study of two key models,” *European Journal of Operational Research*, vol. 206, pp. 622–633, 2010.

-
- [8] S. Sethi, J. Sidney, and C. Sriskandarajah, "Scheduling in dual gripper robotic cells for productivity gains," *IEEE Transactions on Robotics and Automation*, vol. 17, p. 324?341, 2001.
- [9] Q. Su and F. Chen, "Optimal sequencing of double-gripper gantry robot moves in tightly-coupled serial production systems," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 22–30, 1996.
- [10] H. Geismar, M. Dawande, and C. Sriskandarajah, "Throughput optimization in constant travel-time dual gripper robotic cells with parallel machines," *Production and Operations Management*, vol. 15, p. 311?328, 2006.
- [11] H. Gultekin, M. S. Akturk, and O. E. Karasan, "Scheduling in a three-machine robotic flexible manufacturing cell," *Computers and Operations Research*, vol. 34, pp. 2463–2477, 2007.
- [12] H. N. Geismar, M. Dawande, and C. Sriskandarajah, "Robotic cells with parallel machines: Throughput maximization in constant travel-time cells," *Journal of Scheduling*, vol. 7, p. 375?395, 2004.
- [13] M. Dawande, M. Pinedo, and C. Sriskandarajah, "Multiple part-type production in robotic cells: Equivalence of two real-world models," *Manufacturing and service operations management*, vol. 11, p. 210?228, 2009.
- [14] A. Agnetis, "Scheduling no-wait robotic cells with two and three machines," *European Journal of Operational Research*, vol. 123, pp. 303–314, 2000.
- [15] M. Dawande, N. Geismar, S. Sethi, and C. Sriskandarajah, *Throughput Optimization in Robotic Cells*. Springer Science+Business Media, LLC, 2007.
- [16] R. Graham, E. Lawler, J. Lenstra, and A. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Annals of Discrete Mathematics*, pp. 287–326, 1979.
- [17] N. Brauner, G. Finke, and W. Kubiak, "Complexity of one-cycle robotic flow-shops," *Journal of Scheduling*, pp. 355–371, 2003.

- [18] M. Dawande, C. Sriskandarajah, and S. Sethi, “On throughput maximization in constant travel-time robotic cells,” *Manufacturing and Service Operations Management*, vol. 4, pp. 296–312, 2002.
- [19] R. Logendran and C. Sriskandarajah, “Sequencing of robot activities and parts in two-machine robotic cells,” *International Journal of Production Research*, vol. 34, pp. 3447–3463, 1996.
- [20] G. Legnani, *Robotica industriale*. Italy: CEA, 2003.
- [21] F. Shrouf, J. Ordieres-Meré, A. García-Sánchez, and M. Ortega-Mier, “Optimizing the production scheduling of a single machine to minimize total energy consumption costs,” *Journal of Cleaner Production*, vol. 67, pp. 197–207, 2014.
- [22] A. Pareschi, *Impianti Industriali - Criteri di scelta, progettazione e realizzazione*. Editrice Esculapio, 1994.
- [23] J. Carlier, H. Mohamed, Kharbeche, K. Mohamed, and M. Aziz, “An optimization-based heuristic for the robotic cell problem,” *European journal of operational research*, vol. 202, pp. 636–645, 2010.
- [24] C. Tarek, S. Chaabane, L. Taicir, and T. Damien, “A genetic algorithm for robust hybrid flow shop scheduling,” *International journal of computer integrated manufacturing*, vol. 24, pp. 821–833, 2011.
- [25] A. Khalid, A. Kazem, and M. Romeo, “Simulation modelling and analysis of scheduling in robotic flexible assembly cells using taguchi method,” *International journal of production research*, vol. 52, pp. 2654–2666, 2014.
- [26] M. Knapp-Cordes and B. McKeeman, *Improvements to tic and toc Functions for Measuring Absolute Elapsed Time Performance in MATLAB*, MathWorks, 2011.
- [27] E. O. Doebelin, *Strumenti e metodi di misura*. McGraw-Hill, 2008.

- [28] D. L. Pieper, "The kinematics of manipulators under computer control," PhD thesis, Department of Mechanical Engineering, Stanford University, 1968.