

Indice

Introduzione	4
I Il controllo dei sistemi manifatturieri	8
I.1 Introduzione ai sistemi manifatturieri	8
I.2 Classificazione dei sistemi manifatturieri	10
I.3 Indici di prestazione, figure di merito	18
I.4 Disturbi agenti sul sistema, problemi di modellazione	21
I.5 I setup	22
I.6 Confronto tra lo scheduling dinamico e quello statico	26
I.7 Approccio al controllo utilizzando il modello fluido	28
II La struttura gerarchica del controllo	30
II.1 Introduzione	30
II.2 Concetti base	32
II.3 Lo scheduling dei setup nel contesto gerarchico	37
II.4 Caratterizzazione dei cicli limite	41
II.5 Posizione ottima dei cicli limite nel caso di sistemi “ <i>make to order</i> ”	49

III Il controllo mediante le politiche corridoio	62
III.1 Uso delle politiche corridoio	62
III.2 Comportamento dinamico della traiettoria nel caso di un cor- ridoio iperconico simmetrico	68
III.3 Scelta dei corridoi e stabilità dei cicli limite	90
IV Il controllo durante il transitorio	96
IV.1 Introduzione del problema	96
IV.2 Uso della programmazione dinamica come tecnica risolutiva	99
IV.3 Un indice di costo per ottimizzare il comportamento del siste- ma durante il transitorio	109
V Il controllo durante il transitorio di un sistema “<i>make to order</i>”	113
V.1 Formulazione del problema	113
V.2 Soluzione teorica: il caso con buffer infiniti	118
V.3 Soluzione teorica: il caso con buffer finiti	122
V.4 Soluzione numerica del problema	126
V.5 Proprietà della soluzione ottima ricavata con l’approccio nu- merico	133
Conclusioni	141
A Listato del programma	143
Ringraziamenti	160

Bibliografia

161

Introduzione

La manifattura fa parte dell'insieme di tutte quelle attività attraverso le quali l'uomo produce beni volti a migliorare la qualità della vita. Essa è una delle attività più antiche dell'umanità ed è piena di tradizioni. Come tutte le attività umane più importanti, è soggetta ad un costante studio ed a un continuo cambiamento tecnologico. Tuttavia in alcune aree dei sistemi manifatturieri, tecniche *ad hoc* e intuizione risultano essere prevalenti rispetto ai metodi sistematici e, soprattutto per motivi economici, risulta difficile introdurre innovazioni. Accade perciò che metodi moderni (come ad esempio l'uso dei computer) siano a volte applicati solamente per realizzare e consolidare le tecniche già esistenti.

In questa tesi lo studio dei sistemi manifatturieri e delle tecniche per il loro controllo è stato attuato da un punto di vista metodologico e l'attenzione è stata rivolta a sistemi costituiti da una o più macchine che sono in grado di produrre più tipi di parti, e le cui macchine nel commutare la lavorazione da un tipo di parte all'altro impiegano un certo intervallo di tempo, detto tempo di setup, non trascurabile rispetto a tutti gli altri intervalli temporali caratteristici del sistema. La procedura adottata in questa tesi è la seguente: dopo aver definito una misura di prestazione, si cerca di individuare una

tecnica di controllo in retroazione dallo stato del sistema che ottimizzi le prestazioni del sistema stesso.

Lo studio è stato affrontato utilizzando un modello continuo per il sistema manifatturiero, in cui le parti in transito nel sistema sono approssimate da flussi continui. Il sistema sarà considerato non perfettamente affidabile, ossia soggetto a guasti.

Controllare i sistemi manifatturieri del tipo preso in considerazione significa scegliere per ogni macchina l'istante nel quale terminare la produzione del tipo di parte che si sta lavorando e decidere con quale altro tipo di parte far ripartire la produzione dopo le operazioni del setup. Tutto ciò deve essere fatto rispettando le capacità di tutti i magazzini e cercando di limitare il numero delle domande in attesa dei clienti, e quindi il tempo di attesa del generico ordine. Se il controllo è realizzato in retroazione, allora tutte le decisioni appena descritte devono essere prese in tempo reale, sulla base delle informazioni disponibili sullo stato attuale del sistema, come ad esempio il livello dei magazzini, il numero di ordini in attesa, il numero di macchine funzionanti, etc.

L'argomento, di notevole complessità, è stato affrontato in letteratura mediante una struttura gerarchica (si veda [1]) che permette di suddividere il problema in sottoproblemi definiti su scale temporali diverse. La descrizione della struttura gerarchica verrà data nel capitolo II. Gli eventi che influenzano lo stato del sistema, in base alle frequenze con cui si manifestano, vengono raggruppati in livelli differenti. Il livello gerarchico esaminato in questa tesi è lo scheduling dinamico dei cambiamenti dei setup. Il problema

dello scheduling per questo tipo di sistemi è stato estensivamente studiato in letteratura ([2, 3, 4, 5, 6, 7], solo per citarne alcuni).

Dopo un esame generale delle tecniche presenti in letteratura, l'attenzione è stata focalizzata sui sistemi costituiti da una singola macchina che rappresentano un caso molto significativo all'interno del contesto della produzione reale (si veda il capitolo 14 di [8]). Per questo tipo di sistemi, due problematiche ampiamente studiate in letteratura al livello dello scheduling dinamico dei setup sono l'inseguimento di traiettorie nello spazio di stato del sistema definite dal livello gerarchico superiore, e la determinazione di politiche al fine di ottimizzare una funzione di prestazione definita sul sistema.

Una possibile tecnica per risolvere il problema dell'inseguimento di traiettorie è quella delle politiche corridoio ([4]). I contributi della tesi a riguardo, sono riportati nel capitolo II e nel capitolo III, e sono stati rispettivamente l'analisi del comportamento a regime e lo studio del transitorio del sistema di produzione al variare del tipo di corridoio utilizzato.

Per la determinazione di politiche ottime in relazione a funzioni di costo definite sul sistema di produzione (tipicamente associate alle domande in attesa e a quelle perse nel caso di buffer con capacità limitata), sono stati presi in esame diversi lavori ([9] e [10] per il caso a buffer infiniti, [11] e [12] a buffer finiti ma senza tempi di setup). In questa tesi si è considerato il caso di sistemi manifatturieri caratterizzati da tempi di setup non trascurabili e da buffer a capacità finita. Tale problematica è risultata essere poco presente in letteratura.

Il sistema studiato nel presente lavoro è costituito da una macchina capace

di lavorare due tipi differenti di oggetti o parti, uno alla volta. La macchina è soggetta a guasti ed è caratterizzata da tempi di setup non trascurabili. Come indice di costo utilizzato per l'ottimizzazione del comportamento del sistema si è considerata una funzione lineare a tratti del livello dei magazzini e del numero degli ordini in attesa.

La maggior parte della tesi è dedicata ai sistemi “make to order”, cioè che producono soltanto se ci sono dei clienti in attesa che richiedono il prodotto finito.

Nel caso dei sistemi “make to order”, i risultati sono stati l'individuazione del comportamento ottimo del sistema a regime, la caratterizzazione analitica del comportamento ottimo durante il transitorio con buffer infiniti, e la caratterizzazione mediante un approccio numerico del comportamento ottimo transitorio del sistema nel caso di buffer finiti. In quest'ultimo caso, la piena caratterizzazione analitica delle soluzioni è molto più complessa. Saranno comunque presentati alcuni risultati di interesse, come ad esempio la scelta ottima del setup iniziale.

In futuro ci si propone di estendere i risultati analitici trovati nel problema a buffer infiniti anche al caso più realistico di buffer finiti, considerando anche un numero di parti da lavorare maggiore di due.

Capitolo I

Il controllo dei sistemi manifatturieri

I.1 Introduzione ai sistemi manifatturieri

Letteralmente, il termine manifattura sta ad indicare la trasformazione di materiali grezzi in prodotti finiti trasportabili e utili per l'uomo. Le caratteristiche fondamentali di una produzione manifatturiera sono due: la *producibilità* e la *ripetibilità*. Un'azienda manifatturiera vuole sapere in anticipo, e con il massimo livello di accuratezza possibile, le caratteristiche finali di un dato prodotto. Inoltre, essa è interessata a ripetere tante volte, e nello stesso modo, una certa operazione. In altre parole, l'azienda fa una promessa ai suoi consumatori sulle caratteristiche finali del prodotto (o meglio, molte copie di un prodotto). Se l'aspettativa dei consumatori verrà disattesa, sicuramente chi ha prodotto il manufatto ne subirà le conseguenze economiche non certo positive. Oltre che sulle caratteristiche, l'azienda fa una promessa anche sui tempi di rilascio del prodotto. Essa dovrà produrre un certo numero di copie del suo prodotto in un dato intervallo di tempo, se non ci riuscirà

dovrà subire l'insoddisfazione dei suoi consumatori. Quindi la precisione dei tempi di rilascio di un manufatto è importante tanto quanto la bontà delle sue caratteristiche finali.

La manifattura può essere suddivisa in due categorie :

- la manifattura *discreta*, che produce unità distinte di prodotto (per esempio frigoriferi, automobili,...);
- la manifattura *continua*, che produce quantità continue di un certo materiale (raffinazione del petrolio, produzione di liquidi in generale).

Per *sistema manifatturiero* si intende l'insieme dei macchinari, elementi di trasporto, computer, magazzini, e tutti gli apparati che vengono usati per la manifattura. Da questo punto di vista, anche le persone fanno parte di tale sistema. Le sottoparti degli insiemi manifatturieri, che sono a loro volta insiemi, vengono talvolta chiamate *centri di lavoro*, *stazioni di lavoro*, oppure *celle*.

Lo *scheduling* è la scelta degli istanti di tempo futuri in cui si verificheranno gli *eventi controllabili*. Questi ultimi comprendono tutti gli eventi di cui è possibile scegliere l'istante di inizio : il caricamento di parti è un evento controllabile, la rottura di una macchina non lo è. Ogni volta che si verifica un evento incontrollabile lo stato del sistema cambia, e quindi lo scheduling deve essere ricalcolato sulla base della nuova situazione venutasi a formare. Per questo motivo le decisioni devono essere prese in *tempo reale*.

I.2 **Classificazione dei sistemi manifatturieri**

In questa sezione sono riportate due classificazioni dei sistemi manifatturieri: la prima, tratta da [13], è basata sulle caratteristiche dell'impianto e sul suo controllo, la seconda, tratta da [14], è basata sulla modalità della domanda, dell'offerta e della realizzazione del prodotto. Entrambe le classificazioni hanno un carattere statico.

Linee di produzione ed impianti job shop

Le due forme tradizionali di organizzazione di un sistema manifatturiero sono la linea di produzione (flow line) e il job shop. Gli impianti di tipo job shop sono costituiti diversi tipi di macchine, alcune delle quali sono in grado di eseguire operazioni su differenti tipi di parte, anche se ciò può richiedere operazioni di inizializzazione e riconfigurazione. La movimentazione dei materiali negli impianti job shop permette a parti diverse di seguire percorsi diversi. Tale organizzazione consente quindi di lavorare più tipi di parti, ma la maggiore complessità di gestione legata sia alla necessità di controllare il movimento delle parti sia alla necessità di riconfigurazione delle macchine rende tali sistemi inadatti ad alti volumi di produzione.

Le linee di produzione (flow line) sono organizzate in modo tale che tutte le parti visitino le macchine ed i centri di lavorazione che costituiscono il sistema manifatturiero nello stesso ordine, semplificando quindi in modo drastico la gestione della movimentazione, ma riducendo anche la diversità di parti che possono essere lavorate. Viceversa, la semplicità della movimentazione

e il ridotto numero di parti diverse lavorate rendono possibile alti volumi di produzione, con bassi costi. Le linee di produzione, qualora producano un solo tipo di componente, possono essere migliorate tramite l'automazione delle singole macchine in modo rigido, cioè fatta attraverso meccanismi classici di trasmissione del moto, come ad esempio organi a camme.

Linee di produzione automatizzate

Un ulteriore miglioramento, in termini di efficienza rispetto alle linee di produzione automatizzate in modo rigido, può essere ottenuto sincronizzando il funzionamento delle varie macchine in modo tale che le lavorazioni inizino contemporaneamente su tutte le macchine, e la movimentazione delle parti sia sincronizzata (tali sistemi, in letteratura, sono indicati come *automatic transfer lines*). In tal modo, la quantità di parti "semilavorate" presenti nel sistema può essere molto bassa, ottenendo un'alta produttività. Per contro, tali sistemi sono in grado di lavorare pochi tipi di parti, spesso un solo tipo di parte; inoltre, la riconfigurazione del sistema è estremamente lunga e costosa. Il loro uso è divenuto comune intorno agli anni cinquanta.

Linee di produzione flessibili

L'introduzione degli elaboratori nel controllo delle macchine utensili (*macchine a controllo numerico*) ha consentito forme di automazione molto più semplici dal punto di vista della gestione e della riconfigurazione. Ciò ha portato alla realizzazione di *linee di produzione flessibili*, sia nella versione sincronizzata (*flexible transfer lines*) sia nella versione non sincronizzata (*flexible transfer lines*).

xible flow lines). In quest'ultimo caso, è indispensabile disporre di magazzini tra le varie macchine, per consentire un flusso di parti più agevole, che altrimenti sarebbe vincolato dai tempi di lavorazione della macchina più lenta. Le linee flessibili di produzione consentono la produzione di un numero di parti più elevato rispetto alle linee automatizzate, sebbene, a meno di parti assai simili tra loro, la riconfigurazione può richiedere comunque molto tempo. Inoltre, tutte le parti devono comunque visitare le macchine nello stesso ordine.

Sistemi flessibili di produzione

L'uso congiunto di macchine a controllo numerico e di sistemi di movimentazione delle parti ha consentito di realizzare, a partire dagli anni settanta, sistemi flessibili di produzione (*Flexible Manufacturing Systems, FMS*). Un FMS è essenzialmente un job shop automatizzato. In tali sistemi le parti possono visitare le macchine in qualsiasi ordine, e quindi vi è la possibilità di produrre parti di tipi differenti. Tuttavia, la necessità di utilizzare elaboratori per il controllo dell'intero sistema (cioè, sia per il controllo delle macchine che per il controllo della movimentazione), rende i sistemi flessibili di produzione non vantaggiosi per tutti quei casi in cui sono necessari alti tassi di produzione.

Sistemi di produzione cellulare

Una ulteriore forma di organizzazione di un sistema di produzione consiste nella interconnessione lasca di più isole (o *celle*) automatizzate. Mentre cia-

scuna cella è fortemente automatizzata, l'interazione tra le varie celle può anche essere debole, in modo da garantire un maggior livello di flessibilità dell'intero sistema: infatti il numero, la capacità, e la configurazione delle celle può cambiare nel tempo in accordo con le esigenze della produzione. In questo caso, è di fondamentale importanza capire a fondo le relazioni che intercorrono tra le capacità produttive delle singole celle ed i metodi di coordinamento tra le celle, nonché il loro impatto sulle prestazioni complessive del sistema.

Sistemi flessibili di assiematura

Una delle operazioni elementari tipiche di un impianto manifatturiero, che però è difficile da automatizzare in modo economico, è quella di *assiematura*. Ad iniziare dai primi anni ottanta, prima nell'industria tessile e delle calzature, e poi anche in quella elettronica, si è capito come l'introduzione di sistemi flessibili di assiematura, basati su apparati automatici per la movimentazione, il montaggio, ed anche il riconoscimento dei pezzi, possa dar luogo a significativi vantaggi in termini di flusso dei materiali.

Una seconda possibile classificazione dei sistemi manifatturieri, utile ai fini descrittivi, utilizza tre parametri fondamentali di differenziazione:

1. *modalità di risposta alla domanda*;
2. *modalità di risposta dell'offerta*, ovvero di realizzazione del volume di produzione;

3. *modalità di realizzazione del prodotto.*

Modalità di risposta alla domanda

Secondo questo elemento, la produzione può essere organizzata in tre modalità fondamentali:

1. a magazzino (*make to stock*);
2. su commessa (*make to order*);
3. su commessa con magazzini intermedi (*assemble to order*).

I sistemi produttivi a magazzino sono generalmente adottati da imprese che producono *su previsione* prodotti standardizzati o comunque pochi modelli in grandi volumi; in questo caso l'impresa immagazzina quantità (*stock*) di prodotti finiti sulla base della previsione di vendita. Successivamente, a seguito della ricezione degli ordini da parte dei clienti, si prendono dal magazzino le merci.

La produzione su commessa è, al contrario, orientata alla evasione di un ordine specifico del cliente: solo ad ordine acquisito viene avviata la produzione. Nell'ambito della produzione su commessa si possono distinguere due modalità di produzione:

1. su commessa singola (*engineer to order*);
2. su commesse ripetitive (*make to order* propriamente detto).

Nel primo caso si tratta di produzioni specifiche per singoli prodotti in cui alla ricezione dell'ordine è connessa non solo la produzione, ma anche l'attività di

ideazione e progettazione. La commessa è cioè per un prodotto unico, ancora da progettare per rispondere alle richieste della domanda.

Il make to order si riferisce invece, alla produzione di modelli di prodotto che l'impresa ha già nel suo catalogo. La ricezione dell'ordine può quindi comportare eventuali personalizzazioni del prodotto, ma non la progettazione completa, che invece viene effettuata secondo le previsioni e gli studi sulla domanda.

L'assemble to order è una forma intermedia di produzione che prevede la predisposizione di un magazzino di componenti standardizzati, che vengono assemblati solamente alla ricezione di un ordine da parte dei clienti. La componentistica è quindi realizzata in base a previsioni della domanda, ma l'assemblaggio su ordine permette di mantenere una certa flessibilità a fronte di variazioni della domanda. Inoltre, la varietà dei prodotti aumenta perché è funzione delle diverse combinazioni modulari possibili dei componenti di base.

Modalità di predisposizione dell'offerta

Secondo questo aspetto la produzione può essere:

1. unitaria;
2. a lotti (intermittente);
3. continua.

Nelle produzioni unitarie il ciclo produttivo è organizzato in base alle specifiche del singolo prodotto, e quindi i flussi produttivi sono molto vari.

Esempi di produzioni unitarie sono la cantieristica navale o la produzione di abbigliamento di tipo sartoriale.

La produzione intermittente o a lotti permette di mantenere una certa varietà dei prodotti, ma per accrescere l'efficienza, pur mantenendo la flessibilità, la produzione viene programmata in sequenze di lotti di prodotti. Gran parte della produzione industriale è a lotti, di maggiori o minori dimensioni in relazione alle caratteristiche del prodotto e allo specifico mercato dell'impresa.

La produzione continua, tipica dell'industria chimica, per esempio, è caratterizzata da produzioni in grandissimi lotti di prodotti standardizzati. Si tratta in genere di produzioni per processo in cui l'impianto viene considerato come un'unica macchina nel quale entrano le materie prime e dal quale esce il prodotto finito.

Modalità di realizzazione del prodotto

Questo aspetto è in stretta relazione con le caratteristiche del prodotto, e può avvenire:

1. per parti
2. per processo

La produzione per processo, o *continua*, come già sottolineato in precedenza, è applicata a prodotti realizzati soprattutto attraverso trasformazioni chimiche e fisiche delle materie prime, per cui è difficile risalire ai singoli

componenti. La produzione per processo è propria dell'industria chimica, farmaceutica, siderurgica.

La produzione per parti è la più comune produzione manifatturiera, e consiste nella fabbricazione e/o assemblaggio di componenti di prodotto, per cui è possibile risalire da un assieme (prodotto o componente) alle sue parti costituenti.

I.3 Indici di prestazione, figure di merito

La valutazione quantitativa del comportamento di un sistema manifatturiero può essere ottenuta tramite l'introduzione di opportuni indici di prestazione o figure di merito. Tali indici dovrebbero consentire di valutare in modo globale il sistema, e quindi la loro ottimizzazione dovrebbe corrispondere alla conduzione ottima del sistema. Gli indici illustrati in seguito rappresentano alcuni aspetti chiave, ma particolari, del comportamento di un sistema manifatturiero, e la loro combinazione, con coefficienti di peso che dipendono dallo specifico sistema, può rappresentare una modalità di valutazione globale dell'impianto. In effetti, il peso relativo dei vari aspetti, cioè dei vari indici, dipende anche dalle strategie aziendali, e può essere effettivamente determinato (eventualmente in modo "aggregato") solo dai livelli decisionali più alti.

Tasso di produzione (Throughput)

Il tasso di produzione è il numero di parti che vengono prodotte dal sistema (od anche da un gruppo di macchine) nell'unità di tempo. Nel caso di sistemi che producono più tipi di parti, è una grandezza vettoriale. Se la domanda di beni prodotti è nota, le tecniche di controllo devono minimizzare lo scostamento tra la domanda ed il tasso di produzione. Spesso, si assume che il problema di controllo consista nel massimizzare il tasso di produzione, cioè, implicitamente, si assume una domanda di beni infinita, o comunque un sistema che in termini di risorse è sottodimensionato rispetto alla domanda richiesta. Il throughput, propriamente, indica il tasso di produzione nel caso

di sistemi che producono un solo tipo di parti.

Livelli dei magazzini intermedi (Work in Progress)

I magazzini intermedi sono utili per rendere più fluido il flusso all'interno del sistema, e consentono, ad esempio, di disaccoppiare, in caso di guasti, le macchine bloccate da quelle operative. Vi sono però anche svantaggi associati alla presenza di tali magazzini. In particolare, le loro dimensioni dovrebbero essere contenute perché:

- le parti contenute nei magazzini hanno un costo, e non producono profitto finché non escono dal sistema;
- il pericolo di danneggiamento o deperimento delle parti è proporzionale al tempo di permanenza nel sistema;
- eventuali problemi legati alle lavorazioni precedenti possono essere evidenziati in ritardo, e quindi dopo aver prodotto molte parti difettose;
- lo spazio per i magazzini, ed i dispositivi di movimentazione, costano.

È facile convincersi, sulla base delle considerazioni fatte finora, che l'ottenimento di bassi scostamenti tra domanda e tasso di produzione è un requisito in contraddizione con l'ottenimento di bassi livelli dei magazzini intermedi.

Rispetto delle date di consegna (Due dates)

Dal punto di vista del soddisfacimento dei clienti, il rispetto delle date di consegna è uno degli indici di prestazione più importante. Una scarsa at-

tenzione a tale aspetto può determinare una "disaffezione" dei clienti, con conseguente riduzione degli ordini, e quindi del fatturato.

Capacità del sistema

La capacità di un sistema manifatturiero indica il massimo tasso di produzione, per ciascuna delle parti producibili. Per sistemi complessi, che producono più tipi di parti, la definizione di tale concetto è piuttosto complessa, infatti, la capacità di produzione di una parte è fortemente legata ai livelli di produzione di altre parti. Un possibile approccio alla definizione del concetto di capacità è quello di considerare l'insieme di tutti i possibili vettori di tassi di produzione (ove ogni componente indica il tasso della rispettiva parte), compatibili con lo stato del sistema. Tale insieme dipende anche dall'orizzonte temporale rispetto al quale viene calcolato. Se si considerano intervalli brevi, allora lo stato di ciascuna macchina (e cioè, l'essere la macchina attiva o bloccata) influenza l'insieme dei tassi di produzione ammissibili, se si considera un intervallo più ampio, la capacità dipenderà dalle prestazioni medie delle macchine.

I.4 Disturbi agenti sul sistema, problemi di modellazione

In linea di principio, e cioè se il funzionamento di un sistema manifatturiero fosse assolutamente prevedibile, se la disponibilità di materiali "grezzi" fosse adeguata, e se la domanda futura di beni fosse nota con assoluta precisione, sarebbe possibile (almeno teoricamente) pianificare in anticipo la sequenza delle operazioni su tutte le macchine, e la loro temporizzazione. Nella realtà, i sistemi reali sono soggetti a disturbi esterni, come la variabilità della domanda, ed interni, come i guasti, che rendono necessaria l'introduzione di meccanismi per il controllo in linea (cioè, in retroazione dallo stato). Le fonti di incertezza ed i disturbi sono molteplici, ed agiscono su scale temporali anche molto diverse tra loro. Ciò rende utile, ed anzi, spesso necessaria, l'introduzione di schemi gerarchici di controllo.

I.5 I setup

Molti sistemi manifatturieri sono in grado di lavorare differenti tipi di parti. In alcuni casi però, è necessario considerare un costo significativo in tempo e/o denaro ogni volta che una risorsa di produzione viene configurata per eseguire operazioni su un tipo di parte differente da quella attualmente in lavorazione. Vediamo alcuni esempi:

- nella produzione di parti meccaniche, spesso gli utensili di taglio devono essere cambiati ogni volta che cambia il tipo di pezzo da lavorare. Inoltre, se la macchina è controllata attraverso un computer, bisogna cambiare anche il programma di controllo, o perlomeno reimpostarne i parametri. Questo cambiamento può richiedere un intervallo di tempo non trascurabile : si devono rimuovere i vecchi utensili, montare i nuovi, che in genere hanno bisogno di una procedura di calibrazione;
- nella verniciatura dei veicoli, è opportuno raggruppare insieme quelli che dovranno avere lo stesso colore. Questo perché l'operazione di cambio colore richiede un lavaggio lungo e costoso dell'impianto . Inoltre il lavaggio sarà più semplice se si dovrà passare da un colore chiaro ad uno scuro, piuttosto che fare il contrario.

Il problema dello *scheduling* dei setup consiste nel decidere quando fermare la produzione del tipo di parte che attualmente si sta lavorando, e stabilire con quale altro tipo riprendere a produrre.

Quindi effettuare uno scheduling significa risolvere assieme un problema di temporizzazione e uno di sequenziamento. Purtroppo nel vocabolario della

lingua italiana non esiste un termine per esprimere il concetto di temporizzazione più quello di sequenziamento, per questo motivo, in questo lavoro verrà sempre usata la parola anglosassone, e ormai quasi internazionale, *scheduling*.

Se i cambiamenti di setup vengono effettuati troppo spesso, si va incontro ad alti costi. Infatti, se una grande quantità di tempo viene impiegata nel cambiare configurazione, non c'è tempo sufficiente per produrre la quantità di prodotto necessaria a soddisfare la domanda. Viceversa, se i cambiamenti di setup sono poco frequenti, si alzano i livelli dei magazzini.

In questo lavoro si prendono in considerazione soltanto quei sistemi manifatturieri nei quali si devono fare molte copie di uno stesso prodotto. Se si deve produrre soltanto una copia di ogni prodotto, allora il tempo di setup può essere inglobato nel tempo di lavorazione del prodotto stesso, risolvendo banalmente il problema. Analogamente, si suppone che le parti non sono e non devono essere raggruppate in lotti.

Si suppone che il cambiamento del setup attuale sia un evento controllabile, quindi che sia possibile decidere l'istante in cui terminare la lavorazione attuale, per effettuarne un'altra differente dopo la riconfigurazione della macchina.

Il problema dello scheduling, in alcuni casi, può essere molto complicato, come si può evincere dai seguenti esempi :

- i tempi di setup possono essere funzione del numero di pezzi prodotti, del numero di parti da lavorare, o del numero di risorse al momento disponibili. Inoltre la durata di un setup può essere casuale, ad esempio a causa delle caratteristiche variabili dei pezzi grezzi;

- in presenza di rilavorazioni, i tempi di setup possono dipendere dal fatto che il pezzo passa nella macchina per la prima volta oppure no;
- alcuni cambiamenti di setup esigono la sostituzione di utensili. Essi possono essere cambiati singolarmente oppure raggruppati e cambiati insieme. Durante il cambiamento dei setup, ci possono essere verifiche e controlli;
- l'istante in cui cambiare i setup potrebbe anche essere funzione dei cambiamenti dei turni di lavoro, o della vicinanza dei giorni festivi. Ciò impone vincoli aggiuntivi sullo scheduling;
- molte operazioni manifatturiere possono essere divise in tre fasi distinte: il setup, la lavorazione vera e propria, e lo *smantellamento* o *dismissione*. Quest'ultimo non deve essere confuso con il cambiamento del setup, piuttosto esso si occupa di riportare la macchina in uno stato standard, in genere dal punto di vista della pulizia. I setup e gli smantellamenti possono richiedere differenti risorse;
- i costi dei setup possono includere il consumo di materiali, dovuto o a operazioni di test, o al tempo necessario alla stabilizzazione del processo (i primi pezzi prodotti non soddisfano le specifiche richieste, e devono essere eliminati).

In questa introduzione al concetto dei setup, non potevano mancare alcune brevi considerazioni sulla riduzione dei tempi dei setup, e dei loro costi. Il modo più semplice, ma anche il più costoso, di effettuare tale riduzione è

quello di comprare molte macchine, e per ogni operazione, dedicarne una per ogni tipo di parte, o una per ogni insieme di tipi di parti che richiedono lo stesso setup. Lo studio delle tecniche che si possono usare per raggruppare in modo conveniente i tipi delle parti è chiamato *group technology* (Smith, 1989; Kusiak 1990). Un'altra idea è quella di cambiare il setup durante il cambio del turno di lavoro, quando nessuna lavorazione è in opera. È anche opportuno:

- tenere vicino alla macchina i materiali interessati dalla lavorazione, come le parti nuove e gli utensili;
- istruire operai specificatamente per effettuare i cambi dei setup;
- effettuare off-line, mentre la macchina sta ancora lavorando il tipo di parte precedente, il maggior numero di operazioni possibili;
- studiare nei dettagli l'operazione del cambio setup, al fine di eliminare fasi inutili.

I.6 Confronto tra lo scheduling dinamico e quello statico

Nella letteratura tecnica esistono due approcci al problema dello scheduling dei setup su di una macchina capace di lavorare più tipi di parti : l'approccio dinamico e quello statico.

Il primo è incentrato su decisioni prese in tempo reale in base alle informazioni al momento disponibili sullo stato della macchina. In ogni istante t , in funzione del livello dei magazzini, del valore della domanda e della capacità del sistema, bisogna decidere due cose: se cambiare la parte che attualmente si sta producendo, e in caso affermativo, il tipo di parte da usare per cominciare la nuova lavorazione.

Il secondo invece, tratta una situazione di regime: tutte le richieste e tutte le informazioni sono disponibili sin dall'inizio, e non saranno modificate per nessun motivo; non compare più il generico istante t , e ogni decisione (non più in tempo reale) è indipendente dallo stato della macchina (scompare quindi il concetto di retroazione). Un tipico esempio di scheduling statico è il seguente : si supponga di avere un insieme di configurazioni possibili per una macchina e di conoscere il tempo o il costo necessario per passare da una configurazione all'altra. Si consideri il problema di scegliere un ciclo, cioè una sequenza chiusa dei setup, che dovrà essere ripetuto molte volte. Si assuma che ogni setup dovrà apparire esattamente una volta durante un ciclo. Lo scopo è minimizzare il tempo totale, o il costo totale del ciclo. Questo tipo di problema può essere modellato, e risolto, attraverso la formulazione

del commesso viaggiatore (*traveling salesman problem*, *TSP*), in cui un commesso viaggiatore (la macchina) deve visitare un insieme di città (i setup) e ritornare, alla fine del ciclo, esattamente da dove era partito. L'obiettivo è quello di minimizzare il costo o il tempo totale.

Altri tipici esempi di scheduling statico dei setup sono:

1. l'*economic order quantity (EOQ)*, che affronta il problema di come ordinare (e quindi non produrre) il lotto migliore per un solo tipo di parte;
2. l'estensione di *Wagner-Within* dell'EOQ; si considera sempre un solo tipo di parte, ma la domanda non rimane costante, ma è variabile nel tempo;
3. l'*economic lot-sizing problem (ELSP)*, che fa uso di un modello più dettagliato delle risorse che dovranno essere schedulate.

Per approfondimenti su questi temi si può consultare [15], e i numerosi riferimenti presenti nella sua bibliografia.

I.7 Approccio al controllo utilizzando il modello fluido

In un sistema manifatturiero tutti gli eventi, controllabili o non, importanti dal punto di vista del controllo, hanno una natura discreta. Ad esempio gli istanti di inizio delle lavorazioni, il verificarsi di una rottura, o il cambiamento della domanda. Di conseguenza, sembrerebbe naturale studiarli utilizzando dei modelli aventi la medesima caratteristica strutturale, ossia con i modelli a tempo discreto. Usando questi modelli però, è molto difficile trovare delle politiche di controllo ottime, a causa della natura combinatoria del problema decisionale. Per questo motivo si è scelto di usare in questo lavoro un approccio al controllo del flusso (*flow control*). Tale approccio usa una approssimazione continua (anche detta *fluida*) per modellare il flusso discreto delle parti in un sistema manifatturiero. Il modello approssimato che si ottiene è molto più semplice da risolvere, e la sua soluzione può essere utilizzata per determinare lo scheduling discreto delle decisioni. Ciò può essere fatto, ad esempio, usando delle semplici politiche di inseguimento della soluzione del modello continuo.

Per derivare un modello continuo bisogna focalizzare l'attenzione sui tassi con cui gli eventi accadono piuttosto che sulle singole operazioni. Questo modo di procedere risulta particolarmente semplice ed efficace quando il problema del controllo del sistema manifatturiero può essere affrontato per mezzo di una struttura gerarchica, dove i vari eventi possono essere divisi in base alle loro frequenze di accadimento. Quindi, vista l'importanza dell'argo-

mento, il prossimo capitolo è dedicato alla struttura gerarchica del controllo e ai suoi dettagli.

Capitolo II

La struttura gerarchica del controllo

II.1 Introduzione

La maggior parte dei sistemi manifatturieri è grande e molto complessa. Viene naturale cercare di dividere il controllo e la gestione in una *gerarchia* costituita da più *livelli* differenti. Ogni livello sarà caratterizzato dalla *lunghezza* del suo orizzonte di pianificazione, e dal tipo di dati necessari per prendere le decisioni. I livelli più alti della gerarchia avranno dei lunghi orizzonti e useranno dei dati molto aggregati, relativi ad aspetti globali del sistema. I livelli più bassi invece, lavoreranno su orizzonti brevi e con informazioni molto dettagliate. Anche la natura delle incertezze varierà con i livelli.

Le politiche che opereranno sui sistemi manifatturieri dovranno essere capaci di gestire le occorrenze di importanti eventi discreti come i guasti delle macchine, i cambiamenti dei setup, le variazioni della domanda, etc. Per questo motivo, la politica di controllo dovrà essere basata su dei modelli realistici, ed essere trattabile dal punto di vista computazionale. La struttura

della gerarchia sarà basata sulle caratteristiche dello specifico tipo di produzione che si dovrà controllare. I livelli della gerarchia corrisponderanno agli insiemi degli eventi con differenti frequenze di occorrenza.

II.2 Concetti base

Nella letteratura tecnica esistono diversi esempi di scheduling gerarchico (ad esempio [16],[17]). In questo lavoro si farà riferimento alla struttura gerarchica introdotta nel campo dei sistemi manifatturieri da S.B. Gershwin (per approfondimenti è possibile consultare [18], [1], [19], e [5]). In questa gerarchia ogni livello si occupa soltanto degli eventi controllabili e non, che si manifestano approssimativamente con la stessa frequenza temporale. Gli eventi meno frequenti sono assegnati ai livelli più alti della gerarchia, mentre gli eventi con frequenze maggiori sono assegnati progressivamente ai livelli più bassi.

Si assume che le frequenze dei vari livelli siano ben separate. Sotto la precedente assunzione il problema generale dello scheduling può essere diviso in problemi più piccoli, ognuno dei quali impostato e risolto in un ben preciso livello della gerarchia.

Ciò è possibile se si considera che ad ogni livello, gli stati dei livelli più in alto nella gerarchia cambiano molto lentamente, e quindi non si commette un grande errore se li si considera costanti. Gli stati dei livelli più bassi invece, cambiano molto spesso, e quindi possono essere ben descritti attraverso dei tassi di variazione. Ogni livello si occupa soltanto dello scheduling dei *suoi* eventi controllabili e determina i tassi ai quali i livelli più in basso dovranno schedulare i corrispondenti eventi controllabili. Questi tassi costituiscono l'obiettivo dei problemi di scheduling dei livelli inferiori.

Un livello della gerarchia è costituito dai cambiamenti dei setup. L'ap-

proccio è del tutto generale e non si fanno assunzioni sul resto della struttura della gerarchia. Per fissare le idee comunque, si suppone:

- che il livello sopra a quello dei setup corrisponda ai guasti e alle riparazioni delle macchine, e che questi eventi si manifestino meno frequentemente dei cambiamenti dei setup;
- che il livello immediatamente più in basso si occupi dello smistamento (*dispatching*) delle parti, e che questo compito debba essere eseguito molto più frequentemente dei cambiamenti dei setup.

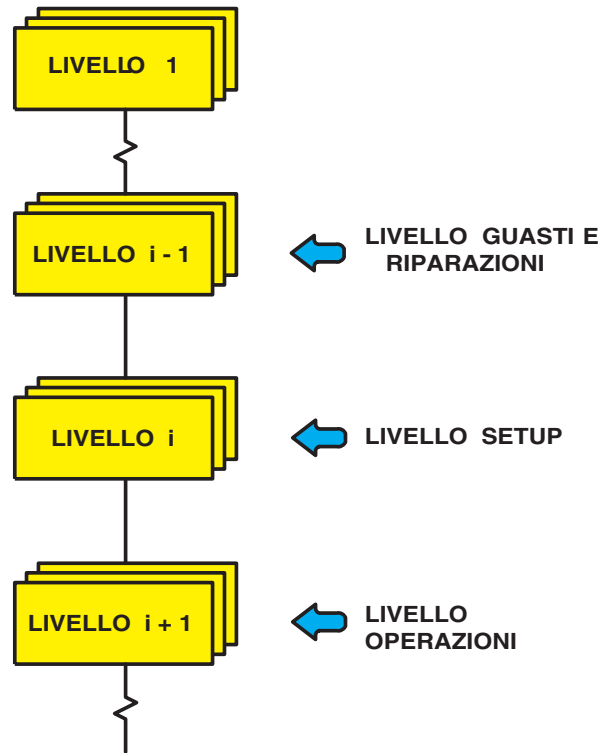


Figura II.1: Esempio dei livelli di una gerarchia

Il livello dei guasti/riparazioni determina ad ogni guasto e ad ogni ripa-

razione del sistema i tassi che costituiscono l'obiettivo per il livello dei setup. Questi tassi includono sia i tassi dei cambiamenti dei setup che i tassi di produzione per i vari tipi di parti. Nel decidere i tassi di cambio setup, il livello superiore non solo può tenere conto dei suoi tassi di riferimento, ma anche di altri fattori come ad esempio un eventuale costo del cambio dei setup. Questo livello usa un modello di ottimizzazione per calcolare i tassi obiettivo per il livello dei setup. I tassi *bersaglio* o *di riferimento* dipendono dalla componente discreta dello stato del livello guasti/riparazioni, e in generale sono diversi per differenti stati della macchina.

Il livello dei setup deve decidere l'esatta temporizzazione, e l'esatto sequenziamento dei cambi dei setup, e deve determinare il tasso di produzione dei tipi di parti in ogni setup, al fine di seguire i tassi di riferimento forniti dal livello dei guasti-riparazioni. Successivamente, i tassi di produzione in ogni setup vengono passati al livello inferiore, il quale deve temporizzare le spedizioni delle parti al fine di seguire i tassi di produzione imposti dal livello immediatamente superiore.

Riassumendo, il generico compito di ogni livello della gerarchia è lo scheduling dei propri eventi controllabili al fine di seguire gli obiettivi fissati dal livello superiore, e il raffinamento di questi obiettivi che costituirà il riferimento per il livello inferiore. Ogni livello, quindi, deve risolvere un problema di controllo del flusso.

Un aspetto fondamentale della teoria gerarchica del controllo è rappresentato dagli insiemi di capacità dei vari livelli. In un determinato livello della gerarchia, l'insieme di capacità è composto da tutti i punti ammissibili

nello spazio dei tassi delle attività con frequenza maggiore, quindi relative al livello inferiore.

Per esempio, si prenda in considerazione una sola macchina che lavora due tipi di parti. Si supponga che la macchina impieghi un certo tempo di setup per passare dalla parte 1 alla parte 2 e viceversa. Particolarmente importanti sono gli insiemi di capacità del livello dei setup e del livello immediatamente superiore, che si suppone essere quello dei guasti/riparazioni.

Al livello dei setup, mentre si sta producendo la parte di tipo i (con $i=1,2$), la capacità è compresa nell'intervallo $[0, \mu_i]$, dove μ_i è il tasso di produzione massimo per la parte di tipo i ammissibile dalla macchina.

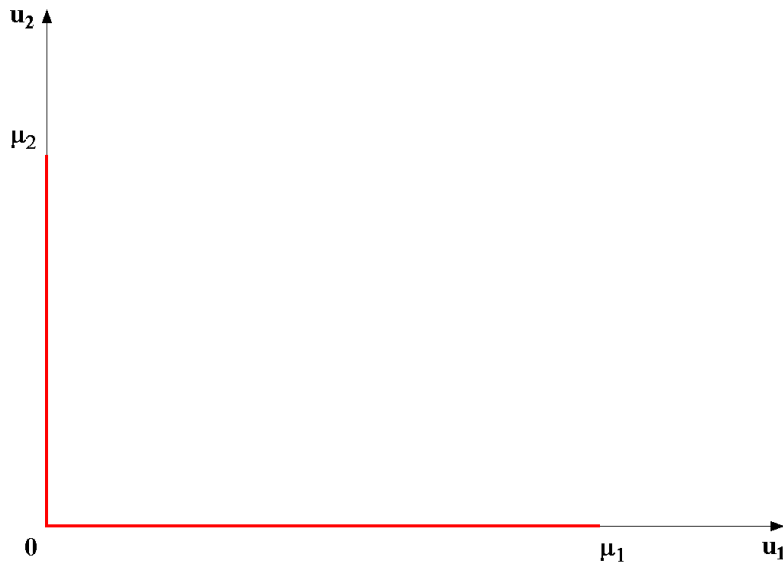


Figura II.2: Insieme di capacità al livello dei setup

Al livello superiore, bisogna decidere sia i tassi di produzione, sia i tassi di cambiamento dei setup per ogni stato della macchina (funzionante e guasto). Quando la macchina è funzionante l'insieme di capacità sarà tridimensionale,

con un asse per ogni tasso di produzione dei tipi di parte, e un terzo asse per il tasso di cambiamento dei setup. Quando la macchina è guasta, ovviamente, la capacità della macchina è 0.

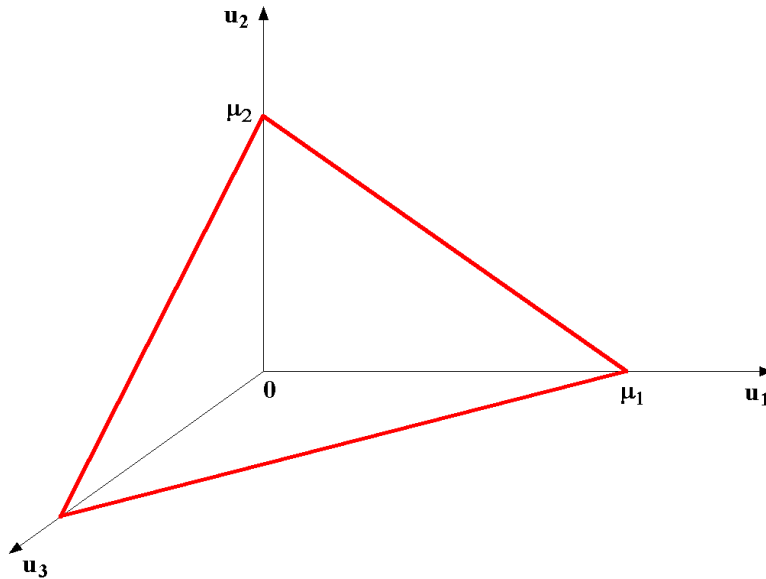


Figura II.3: Insieme di capacità al livello dei guasti e riparazioni

In generale, il tasso di cambiamento dei setup determinato al livello superiore può essere funzione dello stato di quel livello. Per esempio, si può decidere di fare pochi cambi di setup quando si è molto lontani dai tassi di produzione di riferimento tollerando ampie escursioni dei livelli dei magazzini. Viceversa, quando si è molto vicini all'obiettivo, potrebbe essere preferibile destinare una parte maggiore della capacità produttiva ai cambi dei setup, al fine di evitare grosse oscillazioni delle scorte e/o dell'arretrato.

II.3 Lo scheduling dei setup nel contesto gerarchico

Il compito del livello dei setup è di schedulare i cambiamenti dei setup e di determinare per il livello inferiore i tassi di riferimento delle attività in ogni setup. Nel fare questo, deve esso stesso seguire nel miglior modo possibile i tassi forniti dal livello superiore, e ovviamente rispettare i propri vincoli di capacità.

Il surplus di una attività è definito come la differenza tra la produzione cumulativa e la domanda cumulativa. Per esempio il surplus per la produzione di un certo tipo di parte rappresenta l'eccesso di produzione rispetto alla domanda. Quindi se si ottengono surplus negativi (e in questo contesto è possibile), significa che per il tipo di parte in questione è presente un lavoro *arretrato*, ovvero che ci sono domande in attesa che ancora non sono state soddisfatte. Sia \mathbf{x} il vettore dei surplus delle varie attività. Le dinamiche dei surplus possono quindi essere descritte dall'equazione vettoriale:

$$\dot{\mathbf{x}} = \mathbf{u} - \mathbf{d} \quad (\text{II.1})$$

dove \mathbf{u} è il vettore dei tassi di produzione (cioè il numero di attività processate nell'unità di tempo), e \mathbf{d} è il vettore dei tassi con cui arrivano le domande.

Adesso si consideri lo spazio dei vettori \mathbf{x} al livello dei setup. Data una condizione iniziale per \mathbf{x} , è possibile disegnare una traiettoria basata sui tassi di riferimento provenienti dal livello superiore. Il compito del livello dei setup è di seguire *nel miglior modo possibile* questa traiettoria di riferimento.

Sotto ragionevoli assunzioni e approssimazioni, ogni livello della gerarchia

genererà traiettorie di riferimento *lineari a tratti* nello spazio dei surplus per il livello inferiore. Questo segue dalla formulazione della programmazione dinamica del problema del controllo del flusso. Per maggiori dettagli vedi [4].

La soluzione proposta dagli autori consiste in una partizione dello spazio di stato in aree o regioni in cui il tasso di produzione si mantiene costante. Ogni area è associata strettamente ad un vertice dell'insieme di capacità, e quindi, il tasso di produzione ottimo è pari a quello del vertice corrispondente.

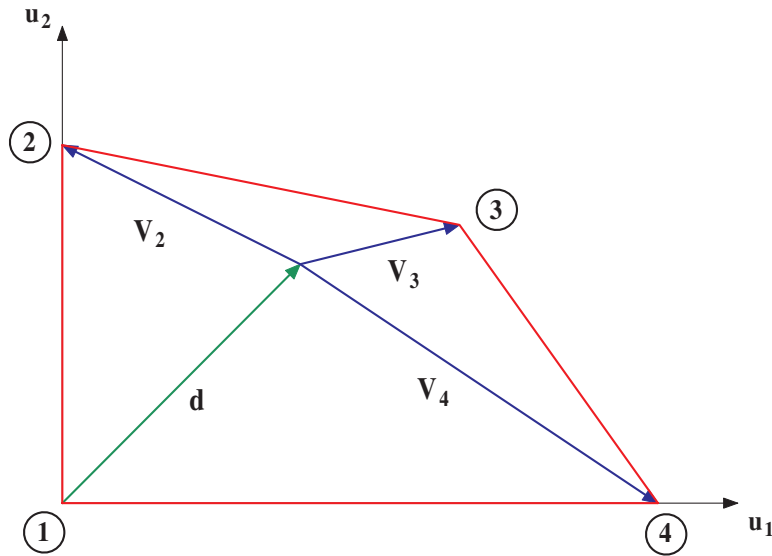


Figura II.4: Vertici e spigoli di un insieme di capacità di una macchina priva dei tempi di setup

Dunque, lo spazio sarà diviso in tante aree quanti sono i vertici. Le frontiere tra le varie aree corrispondono invece agli spigoli dell'insieme di capacità, e sotto alcune approssimazioni, possono essere considerate lineari. Le frontiere

convergono verso un unico punto, universalmente conosciuto come “*hedging point*”. All’interno di una regione, la politica trovata fa muovere la traiettoria del surplus lungo delle linee rette finché non raggiunge una frontiera. Una volta toccata una frontiera il tasso di produzione commuta al valore del vertice corrispondente, e la traiettoria comincia a muoversi all’interno della nuova regione.

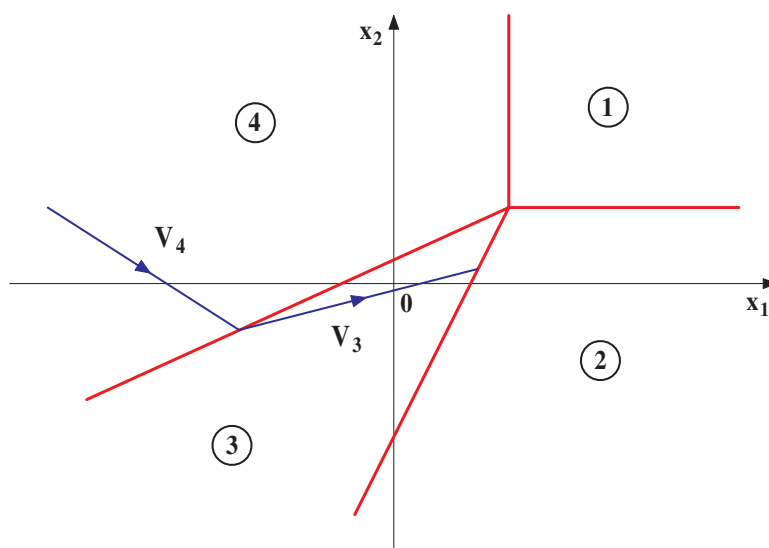


Figura II.5: Esempio di movimento della traiettoria nelle varie regioni

Questo finché la direzione del movimento in una regione è tale da riportare la traiettoria verso la frontiera appena attraversata. Una frontiera con tale caratteristica è chiamata *attrattiva*.

Nasce un problema quando la traiettoria di riferimento va a coincidere con una frontiera attrattiva: bisogna fare in modo che la traiettoria segua la frontiera, evitando il più possibile le oscillazioni (“*chattering*”).

Per fare ciò occorrerebbe fissare il tasso di produzione su un valore appar-

tenente ad uno spigolo dell'insieme di capacità, e non ad un suo vertice. Ma mentre il tasso di produzione associato ad un vertice impone alla macchina di lavorare come *macchina dedicata*, cioè di processare un solo tipo di parte, il tasso associato ad un qualsiasi punto su di uno spigolo implica la lavorazione contemporanea di entrambe le parti, cosa pressoché impossibile in una macchina che presenta dei tempi di setup. Questo problema verrà affrontato attraverso l'uso delle *politiche corridoio*, vedi il capitolo III.

II.4 Caratterizzazione dei cicli limite

Si consideri una traiettoria di riferimento nello spazio n -dimensionale, dove n è il numero delle parti che la macchina può lavorare, che porta all'*hedging point*. Si supponga che tale traiettoria debba essere seguita da una macchina con tempi di setup non trascurabili. Ammesso che la macchina raggiunga l'*hedging point*, per poterci rimanere indefinitamente nel tempo (almeno fino alla prossima rottura della macchina), è necessario porre tutti i tassi di produzione pari ai rispettivi tassi delle domande. Se i tempi dei setup fossero nulli, o perlomeno trascurabili, ciò sarebbe possibile, e la traiettoria rimarrebbe ferma sull'*hedging point*. Però, non essendo questa la situazione presa in considerazione, sicuramente, la traiettoria non rimarrà ferma sul punto, ma tenderà ad allontanarsi.

Nonostante ciò, scegliendo opportunamente gli istanti in cui si cambia la lavorazione dei vari tipi di parti, si può cercare di far ritornare la traiettoria sullo stesso punto dopo aver completato una data sequenza dei cambiamenti dei setup. In questo modo la traiettoria effettuerà dei cicli nello spazio dei surplus, chiamati *cicli limite*. Come esempio di ciclo limite si veda la figura II.6.

Sia:

$$\mathbf{u}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mu_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

il vettore usato per denotare la produzione della macchina nel setup i ; μ_i è il valore massimo del tasso di produzione quando la macchina è nella configurazione i .

Si denoti con :

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}$$

il tasso di domanda che la macchina deve soddisfare. Quindi il vettore :

$$\mathbf{v}_i = \mathbf{u}_i - \mathbf{d} \tag{II.2}$$

rappresenta la velocità della traiettoria nello spazio dei surplus quando la macchina è nel setup i .

Facendo riferimento alla figura II.7, \mathbf{v}_1 è la velocità della traiettoria nello spazio dei surplus (o semplicemente velocità del surplus) quando la macchina è nel setup 1. Naturalmente per \mathbf{v}_2 vale la stessa cosa con le opportune modifiche. Invece, $-\mathbf{d}$ è la velocità del surplus quando la macchina effettua un cambiamento di setup.

Per fare in modo che la traiettoria della macchina effettui un ciclo completo, è necessario che il tasso di produzione medio su di un ciclo sia esattamente pari al tasso di domanda \mathbf{d} .

Si chiami con T il tempo impiegato dalla traiettoria per effettuare un

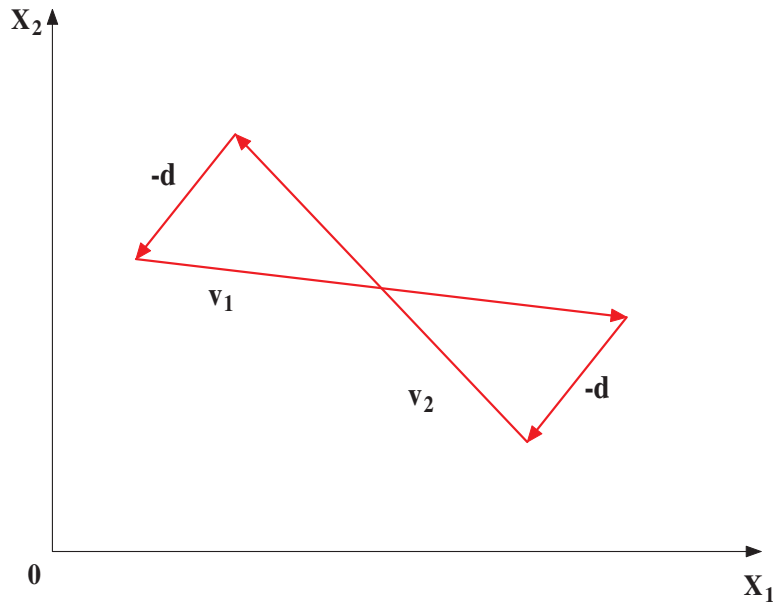


Figura II.6: Esempio di ciclo limite

ciclo, e con β_i la parte di T che la macchina trascorre nel setup i . È possibile, quindi, scomporre \mathbf{d} come segue:

$$\mathbf{d} = \sum_{i=1}^n \beta_i \mathbf{u}_i + \beta_0 \mathbf{0} \quad (\text{II.3})$$

dove $\mathbf{0}$ rappresenta il vettore nullo e β_0 la parte di tempo in cui la macchina non lavora nessuna parte. Vista la particolare forma del vettore \mathbf{u}_i , è possibile risolvere la II.3 rispetto ai β_i , ottenendo:

$$\beta_i = \frac{d_i}{\mu_i}, \quad i = 1, 2, \dots, n. \quad (\text{II.4})$$

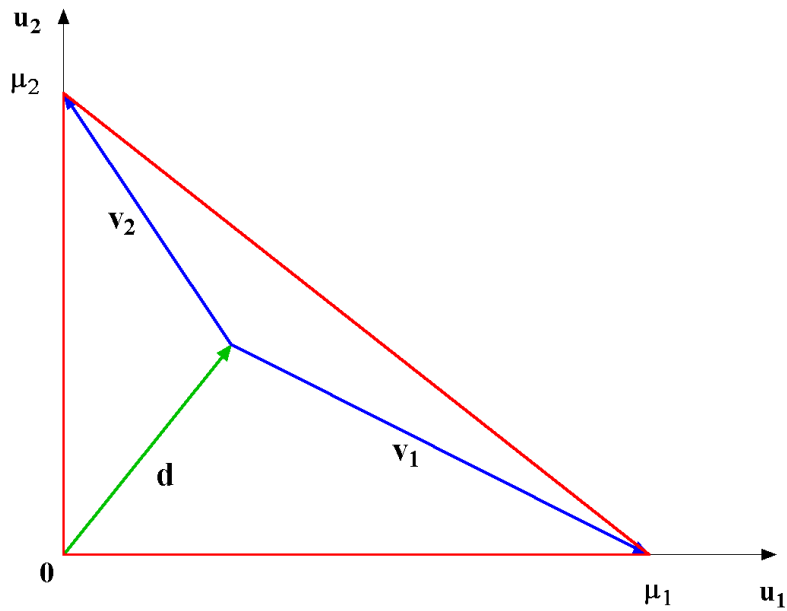


Figura II.7: Velocità del surplus

Sia τ_i il tempo che la macchina impiega per processare una sola parte di tipo i . Risulta quindi :

$$\tau_i = \frac{1}{\mu_i}$$

Se tutti i tempi di setup fossero stati nulli, l'insieme di capacità della macchina sarebbe stato caratterizzato dalla seguente disuguaglianza :

$$\sum_{i=1}^n \tau_i u_i = \sum_{i=1}^n \frac{u_i}{\mu_i} \leq 1 \quad (\text{II.5})$$

Se il vettore della domanda \mathbf{d} è tutto all'interno di questo insieme, si avrà

$$\sum_{i=1}^n \tau_i d_i = \sum_{i=1}^n \frac{d_i}{\mu_i} < 1 \quad (\text{II.6})$$

la quale, utilizzando la II.4, diventa:

$$\sum_{i=1}^n \beta_i < 1 \quad (\text{II.7})$$

o anche

$$\beta_0 = 1 - \sum_{i=1}^n \beta_i > 0 \quad (\text{II.8})$$

È possibile concludere quindi, che quando il vettore della domanda \mathbf{d} è *interno* all'insieme di capacità corrispondente a tempi di setup nulli, sia β_0 che i β_i esistono, sono unici e possono essere determinati usando rispettivamente la II.8, e la II.4. Questo implica l'esistenza di un ciclo limite. Poiché per una politica di controllo stabile il vettore \mathbf{d} deve essere ammissibile per il livello superiore, la condizione sull'esistenza del ciclo limite sarà verificata. Per ulteriori approfondimenti su quest'ultima affermazione si veda [4].

Si consideri una ben precisa sequenza S dei setup, dove ogni setup compare almeno una volta. Sia τ_S il tempo totale che la macchina spende per effettuare tutti i cambiamenti dei setup nella sequenza. Si può scrivere quindi

$$\tau_S = \beta_0 T \quad (\text{II.9})$$

oppure

$$T = \frac{\tau_S}{\beta_0} . \quad (\text{II.10})$$

Il tempo speso nel setup i è quindi pari a

$$t_i = \beta_i T = \frac{\beta_i}{\beta_0} \tau_S . \quad (\text{II.11})$$

È interessante notare che la durata del ciclo limite e il tempo speso in ogni setup dipendono solamente dal tempo totale τ_S speso per i cambiamenti dei setup nella sequenza S . Essi non dipendono né dall'ordine dei setup nella sequenza né dal loro numero. Questo segue direttamente dal fatto che i β_i e β_0 non dipendono dall'ordine dei setup nella sequenza S . In particolare, se il tempo richiesto per cambiare setup dalla configurazione i alla j , dipende solo da j , allora τ_S sarà indipendente dall'ordine dei setup, e tutti i cicli limite corrispondenti ai vari ordinamenti dei setup avranno la stessa durata. Una cosa che dipende ancora dall'ordine dei setup è la forma del ciclo limite .

In generale, un ciclo limite può consistere in una sequenza di setup in cui alcuni si presentano per più di una volta. Un caso speciale è rappresentato dal ciclo limite associato ad una sequenza *round-robin* dei setup, cioè una sequenza nella quale la macchina lavorerà tutti i tipi di parti esattamente una volta sola, e sempre nello stesso ordine. Il ciclo limite corrispondente ad

una data sequenza round-robin sarà completamente determinato una volta note le durate delle lavorazioni t_i , siccome è nota la sequenza e la durata di ogni cambio dei setup.

Dalla II.10 emerge che il ciclo limite sarà più breve per una sequenza per cui τ_S è minimo. Questo può essere un motivo per l'uso delle sequenze round-robin, essendo il tempo totale speso nei cambi dei setup più piccolo di qualsiasi altra sequenza con ripetizioni. Comunque, in alcuni casi può essere preferibile usare differenti tipi di sequenze.

La forma di un ciclo limite per una data sequenza di setup è indipendente dalla sua posizione nello spazio \mathbf{x} . In altre parole, i cicli limite corrispondenti alla stessa sequenza S , con gli stessi t_i e gli stessi β_i possono essere realizzati in posizioni differenti nello spazio, ma avranno tutti la stessa forma.

Come esempio, si consideri un sistema manifatturiero costituito da una macchina capace di lavorare due differenti tipi di parti che verranno dette di tipo 1 e di tipo 2. Si identifichi con:

- τ_{12} il tempo di setup associato al cambio della lavorazione da 1 a 2, e con τ_{21} il viceversa;
- d_1 e d_2 i tassi delle rispettive domande, assunti costanti in accordo con il concetto di gerarchia;
- μ_1 e μ_2 i tassi massimi di produzione.

In accordo con la II.4 si ottengono i valori:

$$\beta_1 = \frac{d_1}{\mu_1}, \quad e \quad \beta_2 = \frac{d_2}{\mu_2} \quad (\text{II.12})$$

che sostituiti nella II.8 forniscono:

$$\beta_0 = \frac{\mu_1\mu_2 - d_2\mu_1 - d_1\mu_2}{\mu_1\mu_2} \quad (\text{II.13})$$

In questo caso particolare τ_S è pari a:

$$\tau_S = \tau_{12} + \tau_{21} \quad (\text{II.14})$$

e quindi per la durata T del ciclo limite si ottiene il valore:

$$T = \frac{\tau_S}{\beta_0} = \frac{(\tau_{12} + \tau_{21}) \cdot \mu_1\mu_2}{\mu_1\mu_2 - d_2\mu_1 - d_1\mu_2} \quad (\text{II.15})$$

mentre per le durate delle lavorazioni si ottengono le espressioni:

$$t_1 = \frac{(\tau_{12} + \tau_{21}) \cdot d_1\mu_2}{\mu_1\mu_2 - d_2\mu_1 - d_1\mu_2} \quad (\text{II.16})$$

$$t_2 = \frac{(\tau_{12} + \tau_{21}) \cdot \mu_1 d_2}{\mu_1\mu_2 - d_2\mu_1 - d_1\mu_2} \quad (\text{II.17})$$

II.5 Posizione ottima dei cicli limite nel caso di sistemi “make to order”

Si prenda in considerazione un sistema manifatturiero composto da una macchina che deve lavorare due differenti tipi di parti. Si assuma che la macchina sia in grado di lavorare soltanto un tipo alla volta, e nel commutare la lavorazione da un tipo all'altro impieghi un certo tempo di setup.

La produzione del sistema manifatturiero è organizzata per mezzo di uno schema *make to order*, (si veda la sezione I.2). Inoltre, il sistema è fornito di due magazzini, uno per ogni tipo, per raccogliere e mettere in coda le domande che man mano arrivano al sistema. Per il corretto funzionamento del sistema è necessario supporre che la macchina abbia una disponibilità pressoché infinita di pezzi grezzi di entrambi i tipi. Il sistema in considerazione può essere schematizzato come:

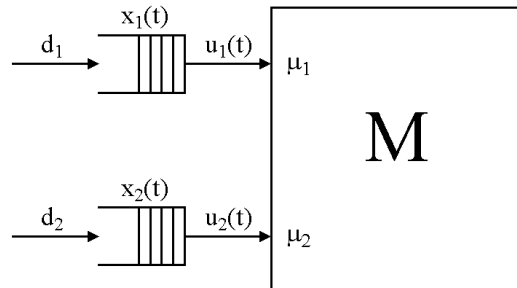


Figura II.8: Il sistema manifatturiero

Un tale sistema manifatturiero ha una caratteristica particolare: sfrutta soltanto il terzo quadrante dello spazio dei surplus (si veda la figura II.9). Infatti esso, producendo soltanto in presenza di domande, è impossibilitato ad im-

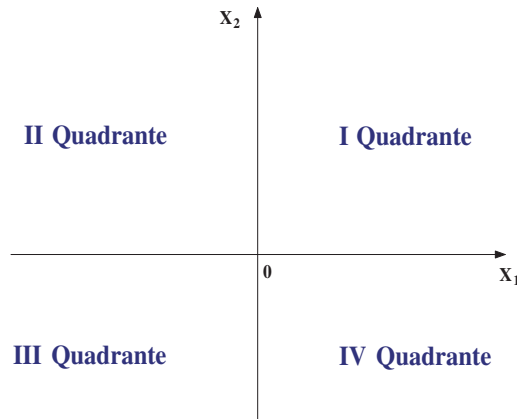


Figura II.9: I quadranti nello spazio dei surplus

magazzinare pezzi finiti. Il funzionamento del sistema è il seguente: in base a come è configurata al momento la macchina, si controlla se c'è una domanda in attesa. Se sì, il pezzo comincia ad essere lavorato. Una volta terminata la lavorazione, lo si fa uscire dal sistema e si toglie la domanda, oramai soddisfatta, dalla coda. Quindi se $x_i(t)$ rappresenta la differenza tra produzione cumulativa e domanda cumulativa, essa non potrà mai essere positiva, ma soltanto negativa (il che indica la presenza di domande in attesa) o al massimo nulla. Ciò esclude che lo stato del sistema possa stare nel primo, nel secondo e nel quarto quadrante.

Per comodità, l'attenzione verrà focalizzata sul terzo quadrante, cambiando il segno di $x_i(t)$ (che quindi assume il significato di differenza tra domanda cumulativa e produzione cumulativa), e di conseguenza anche il verso di tutti i vettori introdotti nelle sezioni precedenti, come si può anche osservare dalla figura II.10. Dopo aver ristretto l'attenzione soltanto al terzo quadrante, è necessario cercare all'interno di esso la posizione ottima del ciclo limite, che

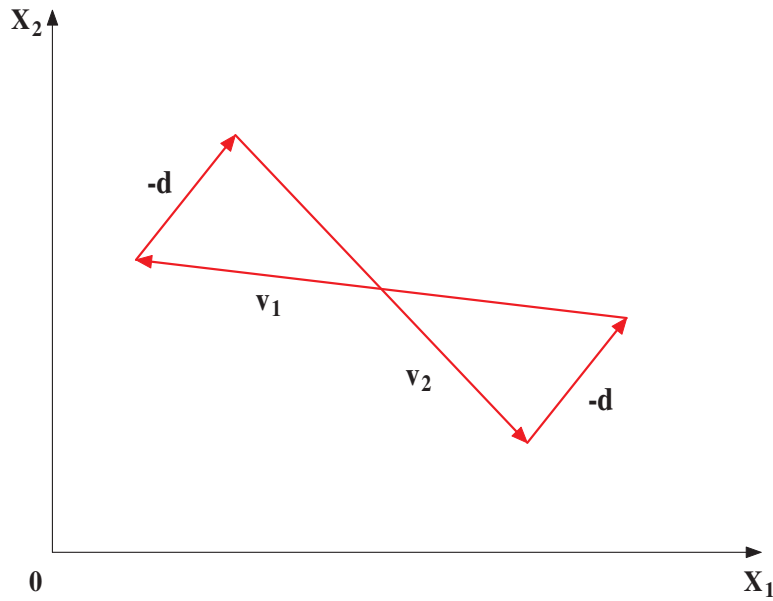


Figura II.10: Cambiamento del verso dei vettori

rappresenta il miglior comportamento del sistema nella situazione di regime. Una posizione ottima “in assoluto” non esiste, ma dipende dalla funzione di costo scelta per premiare le situazioni favorevoli e per penalizzare quelle sfavorevoli.

In questo contesto si sceglie di premiare dei bassi livelli dei magazzini, e di penalizzare quindi degli elevati valori di domande in attesa, che corrispondono all’avere dei clienti insoddisfatti. Il costo puntuale scelto è di tipo lineare, ed è una delle forme più trattate nella letteratura scientifica:

$$g[x(t)] = c_1 x_1(t) + c_2 x_2(t) \quad (\text{II.18})$$

con $c_1 \geq 0, c_2 \geq 0$.

Quello che si vuole minimizzare è il costo medio del ciclo limite, ovvero il costo che si ottiene percorrendolo per intero e una volta soltanto, diviso il tempo di percorrenza. Da notare, che in questo caso, dovendo lavorare soltanto due tipi differenti di parti, esiste soltanto un ciclo limite possibile, e la sequenza è per forza di cose round-robin (se si decide di terminare la produzione delle parti di tipo 1, necessariamente si deve iniziare a lavorare quelle di tipo 2, e viceversa). Come detto nel paragrafo II.4, fissato τ_S , che in questo caso è pari a $\tau_{12} + \tau_{21}$, i parametri che caratterizzano il ciclo limite e la sua forma sono univocamente determinati. La sua posizione nello spazio dei surplus invece è libera e dipende solamente da due parametri: le coordinate di un punto qualunque sul ciclo limite, scelto arbitrariamente come inizio. Si osservi la figura II.11.

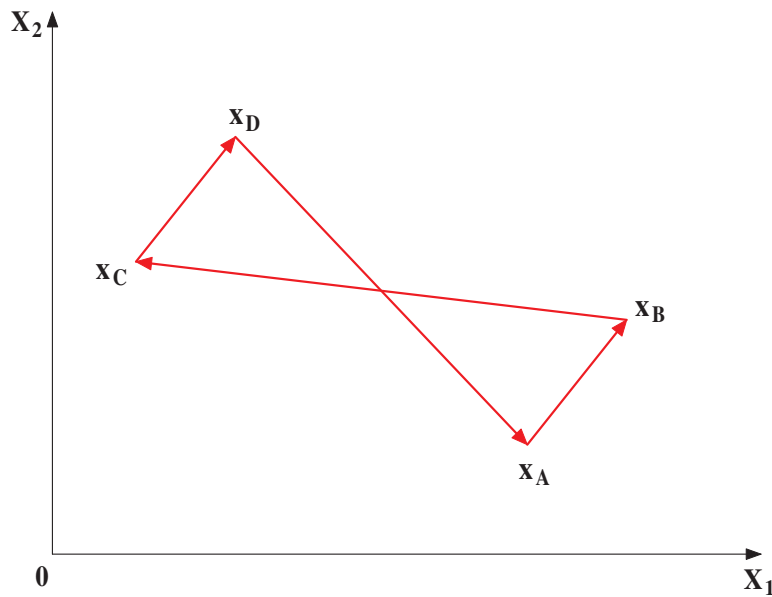


Figura II.11: Figura di riferimento per il calcolo della posizione ottima

Con riferimento alla figura II.11:

- i punti $\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C$, e $\mathbf{x}_D \in \mathbb{R}^2$;
- \mathbf{x}_A è stato scelto come punto di partenza;
- il tratto $\overline{\mathbf{x}_A \mathbf{x}_B}$ corrisponde al cambio del setup da 2 a 1, e ha durata τ_{21} ;
- il tratto $\overline{\mathbf{x}_B \mathbf{x}_C}$ è la lavorazione della parte di tipo 1 e dura t_1 ;
- il tratto $\overline{\mathbf{x}_C \mathbf{x}_D}$ corrisponde al cambio del setup da 1 a 2, e ha durata τ_{12} ;
- il tratto $\overline{\mathbf{x}_D \mathbf{x}_A}$ è la lavorazione della parte di tipo 2 e dura t_2 .

Naturalmente, gli intervalli temporali t_1 e t_2 non sono qualsiasi, ma sono pari a quelli determinati nel paragrafo II.4, rispettivamente tramite la II.16 e la II.17, in modo da far compiere alla traiettoria del surplus un ciclo chiuso. In questo paragrafo comunque, non è importante esplicitarne la forma, e quindi le si lascerà indicate come t_1 e t_2 per non appesantire troppo la scrittura.

I punti $\mathbf{x}_B, \mathbf{x}_C$, e \mathbf{x}_D possono essere facilmente scritti in funzione di x_{A1} e x_{A2} , ovvero le componenti di \mathbf{x}_A rispetto agli assi coordinati, osservando che:

$$\begin{aligned} \mathbf{x}_B &= \mathbf{x}_A + \mathbf{v}_0 \tau_{21} \\ \mathbf{x}_C &= \mathbf{x}_B + \mathbf{v}_1 t_1 \\ \mathbf{x}_D &= \mathbf{x}_C + \mathbf{v}_0 \tau_{12} \end{aligned} \tag{II.19}$$

con:

$$\mathbf{v}_0 = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \quad \text{e} \quad \mathbf{v}_1 = \begin{bmatrix} d_1 - \mu_1 \\ d_2 \end{bmatrix}$$

le quali risolte forniscono i valori:

$$\begin{aligned} x_{B_1} &= x_{A_1} + \tau_{21}d_1 \\ x_{B_2} &= x_{A_2} + \tau_{21}d_2 \end{aligned} \quad (\text{II.20})$$

$$\begin{aligned} x_{C_1} &= x_{A_1} + \tau_{21}d_1 + t_1(d_1 - \mu_1) \\ x_{C_2} &= x_{A_2} + \tau_{21}d_2 + t_1d_2 \end{aligned} \quad (\text{II.21})$$

$$\begin{aligned} x_{D_1} &= x_{A_1} + \tau_{21}d_1 + t_1(d_1 - \mu_1) + \tau_{12}d_1 \\ x_{D_2} &= x_{A_2} + \tau_{21}d_2 + t_1d_2 + \tau_{12}d_2 \end{aligned} \quad (\text{II.22})$$

dove x_{Y_1} e x_{Y_2} indicano rispettivamente la prima e la seconda componente del generico vettore \mathbf{x}_Y . Come già precedentemente detto, si vuole minimizzare il costo medio del ciclo limite in funzione della sua posizione nello spazio dei surplus, analiticamente ciò equivale a risolvere:

$$\min_{\mathbf{x}_A} \lim_{t \rightarrow +\infty} \frac{1}{t} \int_{t_0}^{t_0+t} g[x(\tau)] d\tau \quad (\text{II.23})$$

con $t_0 \geq 0$, istante iniziale qualsiasi. La II.23, con $t = nT + \Delta$, dove T è la durata del ciclo limite, può essere scritta come:

$$\min_{\mathbf{x}_A} \lim_{n \rightarrow +\infty} \frac{1}{nT + \Delta} \int_{t_0}^{t_0+nT+\Delta} g[x(\tau)] d\tau. \quad (\text{II.24})$$

Quindi è sufficiente calcolare il costo medio lungo un ciclo limite. Osservando la figura II.11, ci si rende conto che l'integrale lungo il ciclo limite può essere diviso in quattro parti:

$$\frac{1}{T} \int_{t_0}^{t_0+T} g[x(\tau)] d\tau = \frac{1}{T} (\alpha + \beta + \gamma + \delta) \quad (\text{II.25})$$

con ogni parte pari a :

$$\alpha = \int_{t_0}^{T_1} \mathbf{c}^t [\mathbf{x}_A + \mathbf{v}_0(\tau - t_0)] d\tau$$

$$\beta = \int_{T_1}^{T_2} \mathbf{c}^t [\mathbf{x}_B + \mathbf{v}_1(\tau - T_1)] d\tau$$

$$\gamma = \int_{T_2}^{T_3} \mathbf{c}^t [\mathbf{x}_C + \mathbf{v}_0(\tau - T_2)] d\tau$$

$$\delta = \int_{T_3}^{t_0+T} \mathbf{c}^t [\mathbf{x}_D + \mathbf{v}_2(\tau - T_3)] d\tau$$

dove:

- $T_1 = t_0 + \tau_{21}$;
- $T_2 = t_0 + \tau_{21} + t_1$;
- $T_3 = t_0 + \tau_{21} + t_1 + \tau_{12}$;
- $T = \tau_{21} + t_1 + \tau_{12} + t_2$;
- $\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$

Risolvendo gli integrali si ottiene:

$$\begin{aligned}
 \alpha &= \left(\frac{1}{2}c_1d_1 + \frac{1}{2}c_2d_2 \right) \tau_{21}^2 + (c_1x_{A1} + c_2x_{A2})\tau_{21} \\
 \beta &= \left[\frac{1}{2}c_1(d_1 - \mu_1) + \frac{1}{2}c_2d_2 \right] t_1^2 + \\
 &\quad + (c_1x_{A1} + c_2x_{A2} + c_1d_1\tau_{21} + c_2d_2\tau_{21})t_1 \\
 \gamma &= \left(\frac{1}{2}c_1d_1 + \frac{1}{2}c_2d_2 \right) \tau_{12}^2 + \\
 &\quad + (c_1x_{A1} + c_2x_{A2} + c_1d_1\tau_{21} + c_2d_2\tau_{21} + \\
 &\quad + c_1d_1t_1 + c_2d_2t_1 - c_1\mu_1t_1)\tau_{12} \\
 \delta &= \left[\frac{1}{2}c_1d_1 + \frac{1}{2}c_2(d_2 - \mu_2) \right] t_2^2 + \\
 &\quad + (c_1x_{A1} + c_2x_{A2} + c_1d_1\tau_{21} + c_2d_2\tau_{21} + c_1d_1t_1 + c_2d_2t_1 + \\
 &\quad - c_1\mu_1t_1 + c_1d_1\tau_{12} + c_2d_2\tau_{12})t_2
 \end{aligned} \tag{II.26}$$

Effettuando la somma dei quattro integrali si ha:

$$\begin{aligned}
 \frac{1}{T} \int_{t_0}^{t_0+T} g[x(\tau)] d\tau &= \frac{1}{T} \left[(c_1\tau_{12} + c_1\tau_{21} + c_1t_1 + c_1t_2)x_{A1} + \right. \\
 &\quad + (c_2\tau_{21} + c_2\tau_{12} + c_2t_1 + c_2t_2)x_{A2} + \\
 &\quad \left. + \mathcal{M} \right]
 \end{aligned} \tag{II.27}$$

dove \mathcal{M} è la collezione dei termini rimanenti, tutti comunque indipendenti sia da x_{A_1} e sia da x_{A_2} .

Con riferimento alla figura II.11, si noti che la minimizzazione della II.27 deve essere effettuata rispettando i vincoli:

1. $x_{A_2} \geq 0$
2. $x_{C_1} \geq 0$

Il secondo vincolo può essere esplicitato in funzione di x_{A_1} utilizzando la prima delle II.21, che fornisce:

$$x_{C_1} = x_{A_1} + \tau_{21}d_1 - t_1(\mu_1 - d_1)$$

quindi il vincolo diventa:

$$x_{A_1} \geq t_1(\mu_1 - d_1) - \tau_{21}d_1$$

Il problema da risolvere è quindi un problema di minimo vincolato, che trascurando la costante $\frac{1}{T}$, può essere posto nella forma:

$$\left\{ \begin{array}{l} \min_{\substack{x_{A_1} \\ x_{A_2}}} c_1(\tau_{12} + \tau_{21} + t_1 + t_2)x_{A_1} + c_2(\tau_{21} + \tau_{12} + t_1 + t_2)x_{A_2} + \mathcal{M} \\ \text{s.t. } x_{A_1} \geq t_1(\mu_1 - d_1) - \tau_{21}d_1 \\ x_{A_2} \geq 0 \end{array} \right. \quad (\text{II.28})$$

Come prima cosa da fare per risolvere il problema è opportuno andare ad indagare sulla forma della regione ammissibile, cioè l'insieme dei punti nello

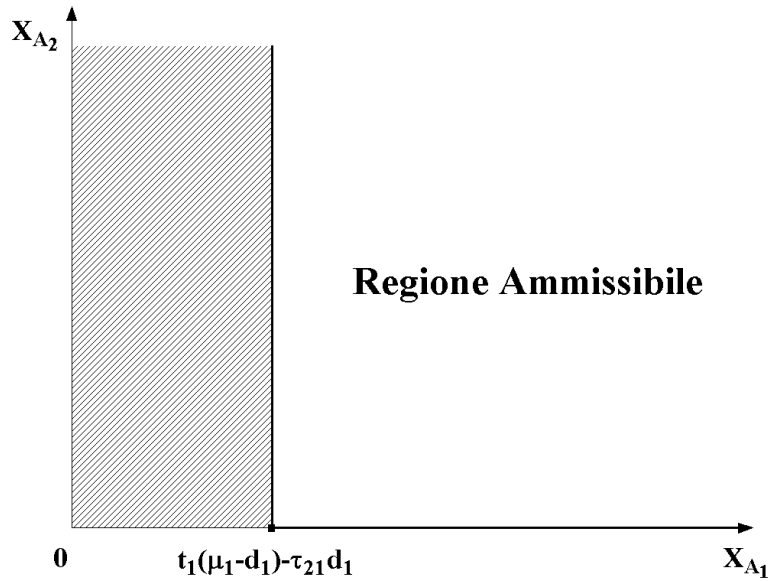


Figura II.12: Regione ammissibile

spazio dei surplus che non violano le equazioni dei vincoli.

Dal disegno si nota che la regione ammissibile è un *insieme convesso* non *limitato*. È opportuno, prima di proseguire nella soluzione del problema di minimizzazione, richiamare alcune definizioni e alcuni risultati fondamentali:

- Un insieme \mathcal{X} di punti si dice *convesso* se, presi due qualsiasi punti $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, collegando \mathbf{x}_1 e \mathbf{x}_2 con un segmento, risulta sempre che tutti i punti del segmento appartengono a \mathcal{X} .
- Un punto \mathbf{x} di un insieme convesso \mathcal{X} è un punto *estremo* di \mathcal{X} se non è esprimibile come combinazione convessa di due punti $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, essendo $\mathbf{x}_1 \neq \mathbf{x}_2 \neq \mathbf{x}$, se cioè non è esprimibile come combinazione convessa stretta di \mathbf{x}_1 e \mathbf{x}_2 .

- Un vettore \mathbf{d} , non nullo, è detto *direzione* di un insieme convesso \mathcal{X} se, per ogni punto $\mathbf{x} \in \mathcal{X}$ il raggio $\{\mathbf{x} + \lambda \mathbf{d} : \lambda \geq 0\}$ appartiene all'insieme.
- Tra le direzioni di un insieme convesso, le direzioni *estreme* non possono essere rappresentate come combinazione conica di n direzioni *distinte*, con n dimensione geometrica dell'insieme. Viceversa, ogni direzione di \mathcal{X} è rappresentabile come combinazione conica di n direzioni estreme.
- Sia \mathcal{X} un insieme convesso e non limitato, allora $\mathbf{x} \in \mathcal{X}$ se e solo se può essere espresso come combinazione convessa dei suoi punti estremi, più una combinazione con costanti positive (conica) delle sue direzioni estreme.

Nel caso in esame, come si può anche notare dalla figura II.12, la regione ammissibile ammette un unico punto estremo:

$$\begin{bmatrix} t_1(\mu_1 - d_1) - \tau_{21}d_1 \\ 0 \end{bmatrix}$$

e due direzioni estreme:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

con ciò, il generico punto \mathbf{x} di \mathcal{X} può essere espresso come:

$$\mathbf{x} = \begin{bmatrix} t_1(\mu_1 - d_1) - \tau_{21}d_1 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (\text{II.29})$$

che semplificato diventa:

$$\mathbf{x} = \begin{bmatrix} t_1(\mu_1 - d_1) - \tau_{21}d_1 + \lambda_1 \\ \lambda_2 \end{bmatrix} \quad (\text{II.30})$$

Se si sostituisce la II.30 all' interno del problema di minimizzazione II.28 si ottiene un nuovo problema del tutto equivalente al precedente:

$$\begin{cases} \min_{\substack{\lambda_1 \\ \lambda_2}} c_1(\tau_{12} + \tau_{21} + t_1 + t_2)(t_1(\mu_1 - d_1) - \tau_{21}d_1 + \lambda_1) + c_2(\tau_{21} + \tau_{12} + t_1 + t_2)\lambda_2 + \mathcal{M} \\ \text{s.t.} \quad \lambda_1 \geq 0 \\ \quad \quad \lambda_2 \geq 0 \end{cases} \quad (\text{II.31})$$

Analizzando la funzione obiettivo, ci si rende conto abbastanza facilmente che sia λ_1 che λ_2 moltiplicano delle quantità positive. Quindi il minimo sarà ottenuto in corrispondenza dei più bassi valori possibili, cioè per $\lambda_1 = 0$ e $\lambda_2 = 0$, ai quali corrispondono i valori

- $x_{A1} = t_1(\mu_1 - d_1) - \tau_{21}d_1$
- $x_{A2} = 0$

Per cui, la posizione ottima del ciclo limite, nel caso trattato è quella mostrata in figura II.13, cioè il più possibile vicino all'origine.

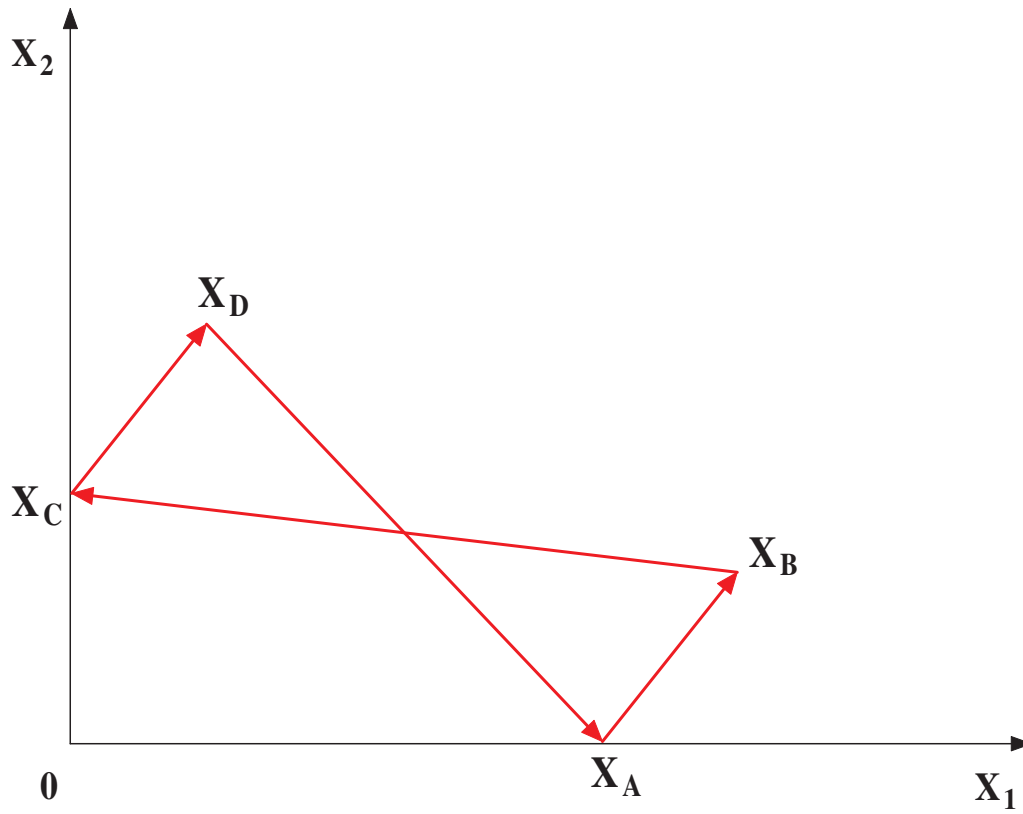


Figura II.13: Posizione ottima del ciclo limite

Capitolo III

Il controllo mediante le politiche corridoio

III.1 Uso delle politiche corridoio

Una tecnica per seguire la traiettoria di riferimento, ampiamente descritta in letteratura (si veda per esempio [4],[7],[20]) è quella delle *politiche corridoio*. L'idea che sta alla base di queste politiche è molto semplice: si costruisce un corridoio attorno alla traiettoria che bisogna seguire. Il corridoio serve per determinare l'esatta temporizzazione dei cambiamenti dei setup, e nel caso che il numero delle parti da lavorare sia maggiore di due, anche a scegliere il successivo tipo di parte da lavorare. La traiettoria del surplus si muove all'interno del corridoio, e ogni volta che raggiunge una faccia dello stesso, si effettua un cambiamento di setup della macchina. Usando questa politica la traiettoria segue la direzione generale del riferimento, "zigzagando" all'interno del corridoio (vedi figura III.1). La larghezza di un corridoio dipende dai tassi di cambiamento dei setup determinati al livello superiore. Bisogna tener presente che un corridoio molto largo genera dei cambiamenti di setup

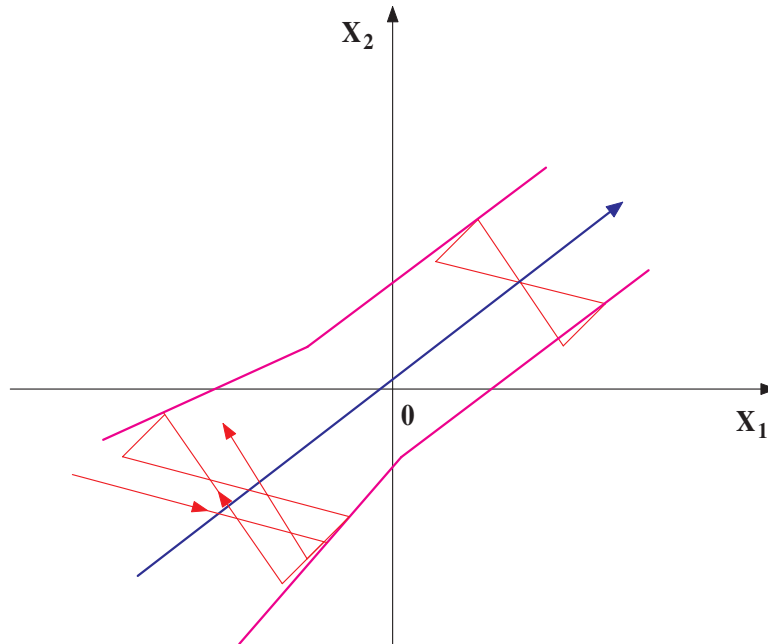


Figura III.1: Un corridoio costruito intorno alla traiettoria di riferimento, e il ciclo limite corrispondente.

poco frequenti e quindi delle ampie oscillazioni nello spazio dei surplus che inducono degli alti valori dei livelli massimi dei magazzini e delle domande in attesa.

Esistono due grandi tipologie di corridoi:

- i corridoi *paralleli* o *prismatici*
- i corridoi *iperconici*

I corridoi prismatici

La forma più semplice per un corridoio è sicuramente quella costituita da degli iperpiani paralleli alla traiettoria da seguire. Nello spazio n -dimensionale,

un corridoio prismatico è costituito da n iperpiani che racchiudono la traiettoria. Ogni faccia del prisma formato dagli iperpiani è associata ad una ben precisa configurazione della macchina. La traiettoria del surplus si muove tutta all'interno del prisma, e quando raggiunge una faccia dà il via al cambiamento del setup: il nuovo tipo di parte che verrà lavorato sarà quello associato alla faccia del corridoio appena raggiunto. Come già detto, un pa-

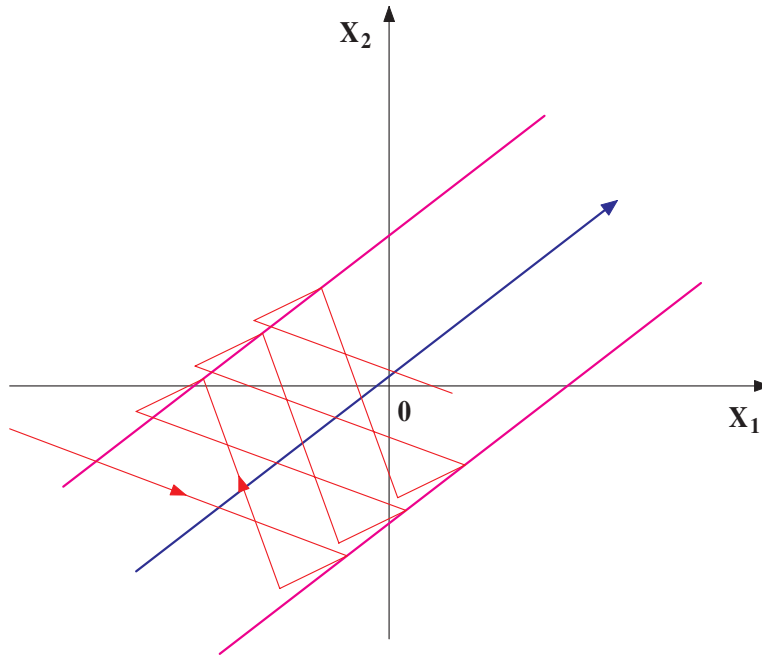


Figura III.2: Un corridoio prismatico.

rametro fondamentale del corridoio è la sua larghezza. Questo parametro è sicuramente limitato inferiormente, infatti esisterà un valore al di sotto del quale la traiettoria del surplus è impossibilitata a seguire quella di riferimento. Questo perché se i cambiamenti dei setup vengono effettuati troppo frequentemente, una parte molto consistente della capacità produttiva della

macchina non viene impiegata nella lavorazione delle parti, e quindi la traiettoria si muoverà nella direzione opposta a quella desiderata. Se il corridoio è scelto propriamente, la traiettoria avanzerà con un passo uniforme verso il punto desiderato (ad esempio l'hedging point).

In assenza di vincoli aggiuntivi, usando un corridoio prismatico, è possibile sorpassare (cioè andare oltre) il punto desiderato, accumulando del surplus sicuramente non voluto. È quindi necessario limitare il movimento della traiettoria nell'intorno del punto obiettivo, ad esempio modificando i tassi di produzione oppure la forma del corridoio stesso, introducendo degli iperpiani aggiuntivi.

I corridoi iperconici

I problemi legati ai corridoi prismatici possono essere superati passando da una forma prismatica ad una iperconica del corridoio. Analogamente al caso precedente, nello spazio n -dimensionale, il corridoio iperconico è costituito da n iperpiani che cingono la traiettoria di riferimento formando, questa volta, un ipercono le cui facce sono associate ognuna ad una particolare configurazione della macchina. La politica di controllo rimane invariata: ogni volta che la traiettoria del surplus raggiunge una faccia dell'ipercono, si cambia il setup, e si configura la macchina in accordo con il tipo di parte associato con la faccia appena toccata. La caratteristica fondamentale di un corridoio iperconico è che se lo si sceglie in modo appropriato, la traiettoria converge ad un ciclo limite stabile senza dover modificare i tassi di produzione o la forma del corridoio stesso nell'intorno dell'hedging point, come è necessario

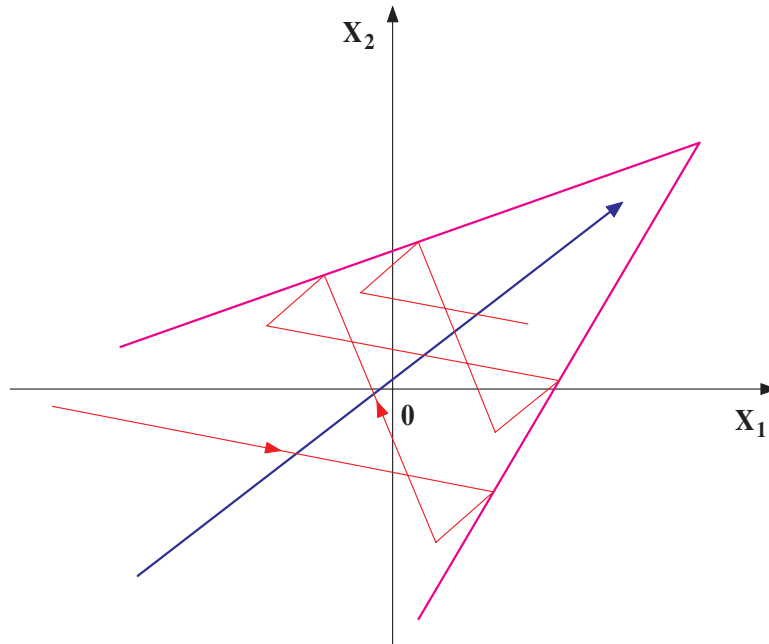


Figura III.3: Un corridoio iperconico.

fare nel caso dei corridoi prismatici.

Con i corridoi iperconici, la traiettoria del surplus si avvicina al riferimento con una velocità che man mano diminuisce, fino a quando la traiettoria viene intrappolata in un ciclo limite.

I corridoi iperconici di tipo “clearing”

I corridoi iperconici di tipo “clearing” costituiscono un caso particolare di quelli iperconici: gli iperpiani che formano il corridoio sono ortogonali agli assi coordinati, interessano solo il terzo quadrante, e l’apice del corridoio coincide con l’origine degli assi. Inoltre, quando la traiettoria raggiunge la faccia:

$$x_{(i-2+n) \bmod(n)+1} = 0$$

il sistema cambia configurazione e il nuovo setup sarà l' i -esimo. Ciò comporta l'azzeramento (o il soddisfacimento) di tutte le domande in attesa della parte attualmente in lavorazione prima di cambiare il setup. Inoltre la sequenza dei setup sarà ciclica e del tipo : $1, 2, \dots, n$.

Una politica molto simile la si può trovare in letteratura in [2], dove vengono trattate le politiche “*clear-a-fraction*”, o più brevemente *CAF*. Queste politiche hanno in comune l'azzeramento delle domande in attesa. Bisogna precisare però, che le politiche corridoio sono più generali, in quanto possono ammettere anche dei soddisfacimenti “parziali” delle domande in attesa: un corridoio ottimale dipende dalla funzione di costo scelta, e non è detto quindi che una politica di tipo *clearing* sia ottimale.

III.2 Comportamento dinamico della traiettoria nel caso di un corridoio iperconico simmetrico

Nella sezione III.1 è stato introdotto il concetto di corridoio di controllo della traiettoria del surplus assieme alle sue tipologie più importanti. Adesso si vuole determinare, in modo analitico, la dinamica della traiettoria del surplus controllata attraverso un particolare corridoio iperconico, simmetrico e parametrizzato attraverso un angolo α .

Per semplicità, ma senza nulla togliere alla generalità della trattazione, si consideri un corridoio iperconico avente il suo apice nell'origine del sistema di coordinate. Questa condizione può essere soddisfatta per mezzo di una semplice traslazione degli assi delle coordinate.

Il sistema manifatturiero oggetto dello studio è in grado di lavorare due tipi differenti di parti, un tipo alla volta, presentando dei tempi non trascurabili per il cambio della configurazione da un tipo all'altro. Si consideri per il controllo della traiettoria un corridoio iperconico formato da due iperpiani che, considerando le dimensioni geometriche del problema, altro non sono che due rette passanti per l'origine. Sia α l'angolo compreso tra ogni retta e la bisettrice Π_B del primo e del terzo quadrante. Si chiami con Π_S la retta che sta al di sopra della bisettrice, e con Π_I quella che sta al di sotto: il corridoio considerato è illustrato nella figura III.4. Riassumendo, si vuole studiare un corridoio iperconico, simmetrico rispetto alla bisettrice del primo e del terzo quadrante di equazione $x_1 = x_2$, e con un angolo di apertura complessivo

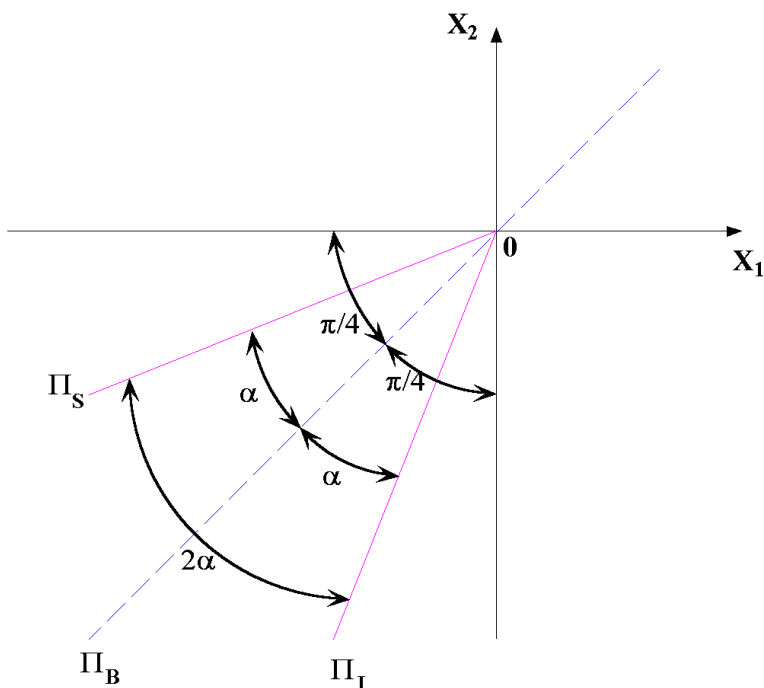


Figura III.4: Il corridoio di controllo.

pari a 2α .

Le equazioni delle rette Π_S e Π_I in funzione del parametro α si possono ricavare considerando le due seguenti matrici di rotazione:

- rotazione antioraria di α radianti

$$RA(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$$

- rotazione oraria di α radianti

$$RO(\alpha) = RA(-\alpha) = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) \\ \sin(-\alpha) & \cos(-\alpha) \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}.$$

Si consideri una retta Π'_S coincidente con la bisettrice Π_B . L'equazione della bisettrice può essere espressa in due modi differenti:

1. $x_1 - x_2 = 0$, con gradiente pari a $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$;
2. $x_2 - x_1 = 0$, con gradiente pari a $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$.

Poiché il corridoio è formato dalla parte di piano compresa tra la retta Π_S e la retta Π_I (si veda la figura III.4), per il controllo della traiettoria bisogna considerare il semipiano che giace al di sotto di Π'_S . Questo semipiano, scegliendo di rappresentare tutti i semipiani con una forma del tipo $\mathbf{a}^t \mathbf{x} \leq 0$, ha come gradiente il vettore diretto verso l'alto, cioè quello presente al punto 2. Quindi come gradiente di Π'_S bisogna considerare il vettore:

$$\mathbf{a}'_S = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

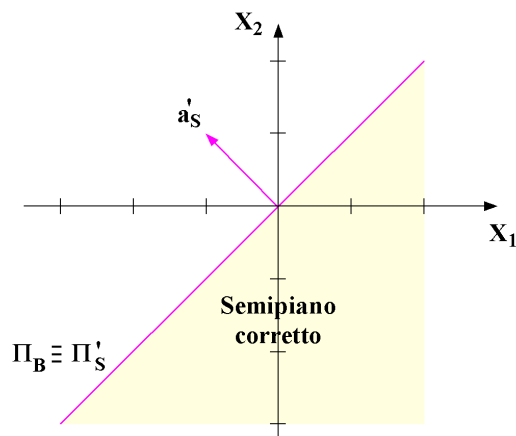


Figura III.5: Iperpiano superiore prima della rotazione

Ruotando \mathbf{a}'_S in senso orario di α radianti si ottiene il gradiente \mathbf{a}_S di un iperpiano Π_S che risulta essere ruotato rispetto all'iperpiano Π_B in senso orario di α radianti.

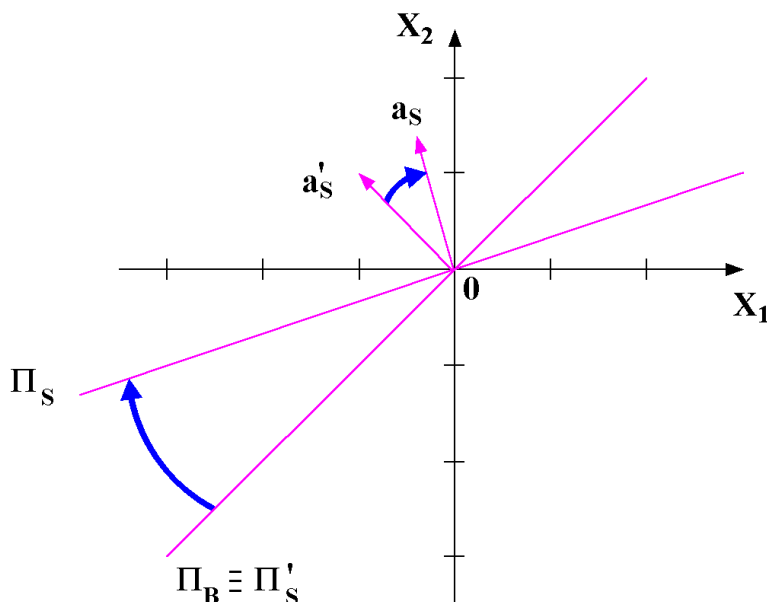


Figura III.6: Iperpiano superiore dopo la rotazione

In formule:

$$\mathbf{a}_S = RO(\alpha)\mathbf{a}'_S = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -\cos \alpha + \sin \alpha \\ \cos \alpha + \sin \alpha \end{bmatrix} \quad (\text{III.1})$$

l'equazione cercata di Π_S è perciò:

$$\mathbf{a}_S^t \mathbf{x} = 0 \quad \Leftrightarrow \quad (-\cos \alpha + \sin \alpha) x_1 + (\cos \alpha + \sin \alpha) x_2 = 0. \quad (\text{III.2})$$

Un analogo procedimento applicato ad un altro iperpiano Π'_i sempre coincidente con Π_B , ma considerando questa volta il semipiano al di sopra della bisettrice e una rotazione antioraria di α radianti, porta al risultato:

$$\mathbf{a}_I = RA(\alpha)\mathbf{a}'_I = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \cos \alpha + \sin \alpha \\ -\cos \alpha + \sin \alpha \end{bmatrix} \quad (\text{III.3})$$

e all'equazione dell'iperpiano:

$$\mathbf{a}_I^t \mathbf{x} = 0 \quad \Leftrightarrow \quad (\cos \alpha + \sin \alpha) x_1 + (-\cos \alpha + \sin \alpha) x_2 = 0. \quad (\text{III.4})$$

La situazione complessiva è quella mostrata in figura III.7, dove la parte di piano evidenziata è il corridoio di controllo.

Caratterizzato in funzione del parametro α il corridoio, non rimane che descrivere in modo efficiente la dinamica della traiettoria del surplus al suo interno. Per fare questo si supponga che i tempi per il cambio dei setup siano tra loro uguali, e pari a una costante δ :

$$\tau_{12} = \tau_{21} = \delta.$$

Sia $\mathbf{x}(t_k)$ la k -esima intersezione della traiettoria con la bisettrice $x_1 = x_2$. Lo scopo è quello di cercare una relazione che legghi due intersezioni immediatamente successive tra la traiettoria del surplus e la retta Π_B ; cioè una

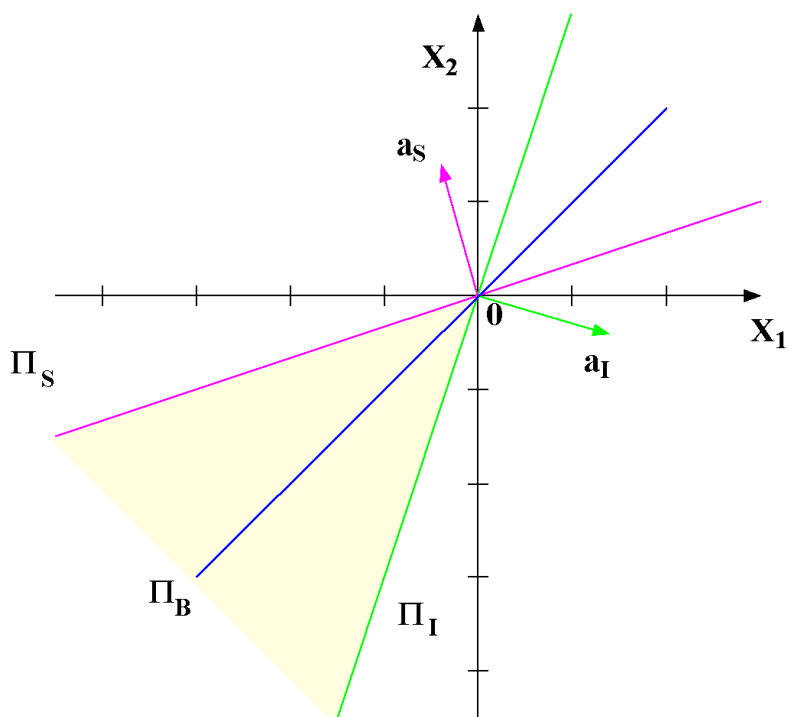


Figura III.7: Corridoio iperconico simmetrico

relazione del tipo:

$$\mathbf{x}(t_{k+1}) = f[\mathbf{x}(t_k)].$$

A tal fine si consideri un pezzo generico di traiettoria all'interno del corridoio con punto iniziale e finale sulla bisettrice, come quello mostrato in figura III.8. Il punto \mathbf{x}_A è pari a:

$$\mathbf{x}_A = \mathbf{x}_O + t_1 \mathbf{v}_1, \quad (\text{III.5})$$

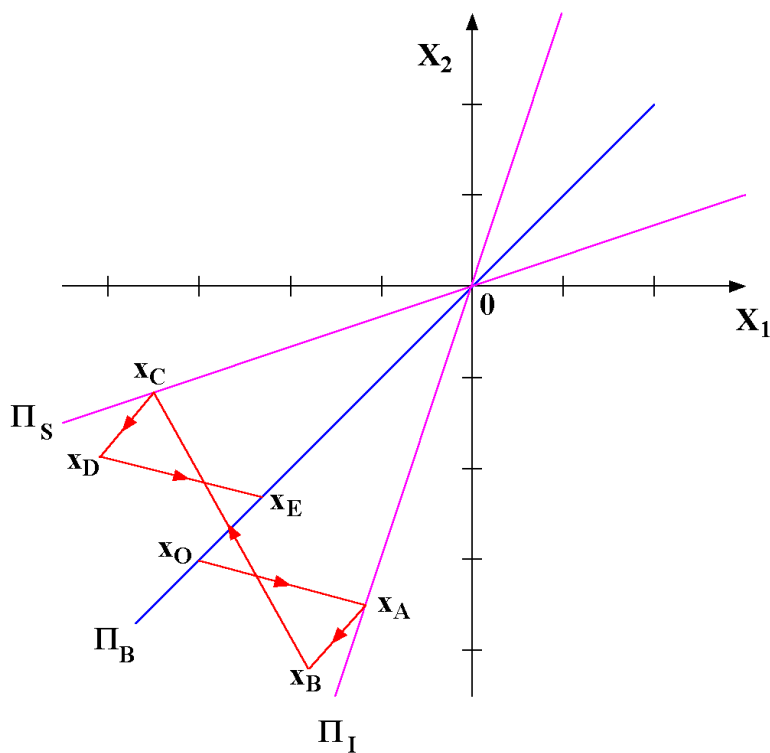


Figura III.8: La traiettoria considerata.

dove t_1 è la durata temporale del tratto $\overline{x_O x_A}$, e \mathbf{v}_1 è la velocità della traiettoria del surplus quando la macchina è nella configurazione 1. Per determinare t_1 è sufficiente osservare che il punto \mathbf{x}_A appartiene alla retta Π_I , e quindi ne soddisfa l'equazione:

$$\mathbf{a}_I^t \mathbf{x}_A = 0 \quad \Leftrightarrow \quad \mathbf{a}_I^t (\mathbf{x}_O + t_1 \mathbf{v}_1) = 0 \quad \Leftrightarrow \quad \mathbf{a}_I^t \mathbf{x}_O + t_1 \mathbf{a}_I^t \mathbf{v}_1 = 0 \quad (\text{III.6})$$

da cui risulta:

$$t_1 = -\frac{\mathbf{a}_I^t \mathbf{x}_O}{\mathbf{a}_I^t \mathbf{v}_1}, \quad (\text{III.7})$$

che sostituita nella equazione III.5 fornisce il valore di \mathbf{x}_A in funzione solo del punto iniziale \mathbf{x}_O :

$$\mathbf{x}_A = \mathbf{x}_O - \frac{\mathbf{a}_I^t \mathbf{x}_O}{\mathbf{a}_I^t \mathbf{v}_1} \mathbf{v}_1. \quad (\text{III.8})$$

Manipolando in modo opportuno la III.8 è possibile ottenere una forma matriciale molto compatta: si osservi che i risultati dei prodotti $\mathbf{a}_I^t \mathbf{x}_O$ e $\mathbf{a}_I^t \mathbf{v}_1$ sono degli scalari, quindi è lecito scrivere:

$$\mathbf{x}_A = \mathbf{x}_O - \frac{1}{\mathbf{a}_I^t \mathbf{v}_1} \mathbf{v}_1 \mathbf{a}_I^t \mathbf{x}_O,$$

raccogliendo a destra nel secondo membro il vettore \mathbf{x}_O si ottiene:

$$\mathbf{x}_A = \left(I - \frac{\mathbf{v}_1 \mathbf{a}_I^t}{\mathbf{a}_I^t \mathbf{v}_1} \right) \mathbf{x}_O = A_1 \mathbf{x}_O, \quad (\text{III.9})$$

dove la matrice A_1 è pari a:

$$A_1 = \left(I - \frac{\mathbf{v}_1 \mathbf{a}_I^t}{\mathbf{a}_I^t \mathbf{v}_1} \right),$$

e la matrice I è ovviamente la matrice identità 2×2 :

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Per il tratto di traiettoria successivo, cioè $\overline{\mathbf{x}_A \mathbf{x}_B}$, non è necessario calcolarne la durata in quanto questa è pari al tempo δ del cambio del setup, noto per ipotesi:

$$\mathbf{x}_B = \mathbf{x}_A + \delta \mathbf{v}_0.$$

Ripetendo lo stesso ragionamento per i vari tratti del pezzo di traiettoria considerato, si arriva ad una espressione che lega il punto finale \mathbf{x}_E al punto iniziale \mathbf{x}_O :

$$\mathbf{x}_E = A\mathbf{x}_O + B\Delta, \quad (\text{III.10})$$

con:

- $A = A_1 A_2 A_3,$
- $B = [A_3 A_2 \mathbf{v}_0, A_3 \mathbf{v}_0],$
- $A_2 = \left(I - \frac{\mathbf{v}_2 \mathbf{a}_S^t}{\mathbf{a}_S^t \mathbf{v}_2} \right),$
- $A_3 = \left(I - \frac{\mathbf{v}_1 \mathbf{a}_B^t}{\mathbf{a}_B^t \mathbf{v}_1} \right),$

dove \mathbf{a}_B è pari al gradiente della bisettrice $\Pi_B,$

- $\Delta = \begin{bmatrix} \delta \\ \delta \end{bmatrix}.$

In pratica l'espressione lineare trovata lega due intersezioni immediatamente successive della traiettoria del surplus con la bisettrice, cioè il risultato che si stava cercando. Quindi generalizzando è possibile scrivere:

$$\mathbf{x}(t_{k+1}) = A\mathbf{x}(t_k) + B\Delta. \quad (\text{III.11})$$

Si osservi che il generico punto $\mathbf{x}(t_k)$, per ogni k , appartiene per costruzione alla bisettrice Π_B . Per questo motivo, l'equazione vettoriale III.11 sarà formata da due equazioni scalari (le righe corrispondenti) esattamente uguali tra loro. Quindi per l'analisi della dinamica del punto di intersezione è sufficiente considerarne una soltanto.

L'ultimo passo da fare è relazionare l'istante t_{k+1} con l'istante t_k , ovvero trovare la dinamica degli istanti in cui avviene l'intersezione tra la traiettoria e la bisettrice. Per fare ciò si identifichi con:

- t_0 l'istante iniziale;
- t_1 la durata del tratto $\overline{\mathbf{x}_O\mathbf{x}_A}$;
- t_2 la durata del pezzo $\overline{\mathbf{x}_B\mathbf{x}_C}$;
- t_3 la durata temporale tra \mathbf{x}_D e \mathbf{x}_E .

Introducendo le variabili T_A, T_B, T_C pari a:

- $T_A = t_0 + t_1$;

- $T_B = T_A + \delta$;
- $T_C = T_B + t_1$;
- $T_D = T_C + \delta$;
- $T_E = T_D + t_3$;

le quali rappresentano gli istanti in cui la traiettoria raggiunge il punto nello spazio del surplus con il pedice uguale (per esempio T_C è l'istante nel quale la traiettoria raggiunge il punto \mathbf{x}_C), e sostituendoci le espressioni di t_1, t_2 , e t_3 si ottiene:

$$T_E = t_0 + 2\delta + \left(\frac{\mathbf{a}_S^t \mathbf{x}_O}{\mathbf{a}_S^t \mathbf{v}_2} + \delta \frac{\mathbf{a}_S^t \mathbf{v}_O}{\mathbf{a}_S^t \mathbf{v}_2} - \frac{\mathbf{a}_I^t \mathbf{x}_O}{\mathbf{a}_I^t \mathbf{v}_1} \cdot \frac{\mathbf{a}_S^t \mathbf{v}_1}{\mathbf{a}_S^t \mathbf{v}_2} \right) \left(\frac{\mathbf{a}_B^t \mathbf{v}_2}{\mathbf{a}_B^t \mathbf{v}_1} - 1 \right) \quad (\text{III.12})$$

che scritta in termini generali e in modo compatto diventa:

$$t_{k+1} = t_k + \mathcal{C}, \quad (\text{III.13})$$

con

$$\begin{aligned} \mathcal{C} &= 2\delta + \left(\frac{\mathbf{a}_S^t \mathbf{x}(t_k)}{\mathbf{a}_S^t \mathbf{v}_2} + \delta \frac{\mathbf{a}_S^t \mathbf{v}_O}{\mathbf{a}_S^t \mathbf{v}_2} - \frac{\mathbf{a}_I^t \mathbf{x}(t_k)}{\mathbf{a}_I^t \mathbf{v}_1} \cdot \frac{\mathbf{a}_S^t \mathbf{v}_1}{\mathbf{a}_S^t \mathbf{v}_2} \right) \left(\frac{\mathbf{a}_B^t \mathbf{v}_2}{\mathbf{a}_B^t \mathbf{v}_1} - 1 \right) \\ &= P\mathbf{x}(t_k) + Q. \end{aligned}$$

In conclusione la dinamica della particolare traiettoria considerata è:

$$\begin{cases} \mathbf{x}(t_{k+1}) &= A\mathbf{x}(t_k) + B\Delta \\ t_{k+1} &= t_k + P\mathbf{x}(t_k) + Q \end{cases} \quad (\text{III.14})$$

Si prenda in considerazione l'autovalore di una qualsiasi delle equazioni scalari che costituiscono la prima delle III.14. Esso è pari a:

$$\lambda = -\frac{-\mu^2 - 4d^2 + 4d^2 \cos^2 \alpha + 4d\mu - 4d\mu \cos^2 \alpha + 2\mu^2 \sin \alpha \cos \alpha - 4d\mu \sin \alpha \cos \alpha}{\mu^2 + 2\mu^2 \sin \alpha \cos \alpha - 4d\mu + 4d\mu \cos^2 \alpha - 4d\mu \sin \alpha \cos \alpha + 4d^2 - 4d^2 \cos^2 \alpha} \quad (\text{III.15})$$

cioè una funzione del tasso d di arrivo delle domande, del tasso μ di produzione, e infine dell'angolo α di apertura del corridoio. Da notare che l'autovalore è del tutto indipendente dalla durata δ del cambio dei setup.

A titolo esemplificativo sono riportate di seguito alcune figure ottenute per differenti valori di d, μ e α , rappresentanti sia l'evoluzione del punto $\mathbf{x}(t_k)$ nello spazio del surplus e sia la dinamica del generico istante t_k . I valori dei parametri utilizzati per generare le varie figure sono riassunti nella tabella III.1.

Figura	α	μ	d	\mathbf{x}_O
III.9, III.10	$\frac{\pi}{4}$	3	1	-50
III.11, III.12	$\frac{\pi}{6}$	3	1	-50
III.13, III.14	$\frac{\pi}{4}$	10	2	-50

Tabella III.1: I valori dei parametri delle figure

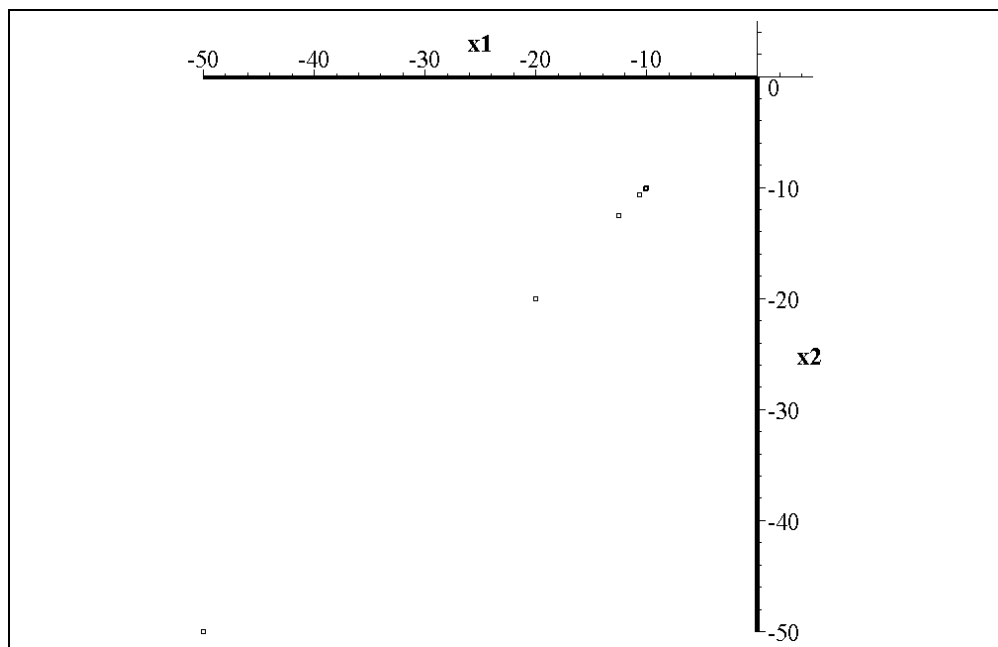


Figura III.9: $\alpha = \pi/4$, $\mu = 3$, $d = 1$

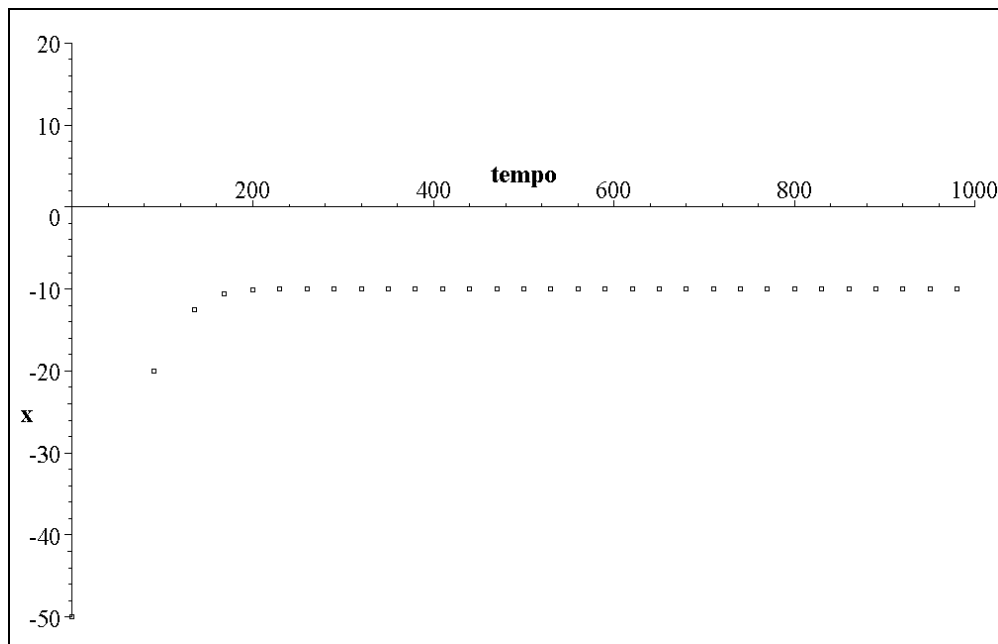


Figura III.10: $\alpha = \pi/4$, $\mu = 3$, $d = 1$

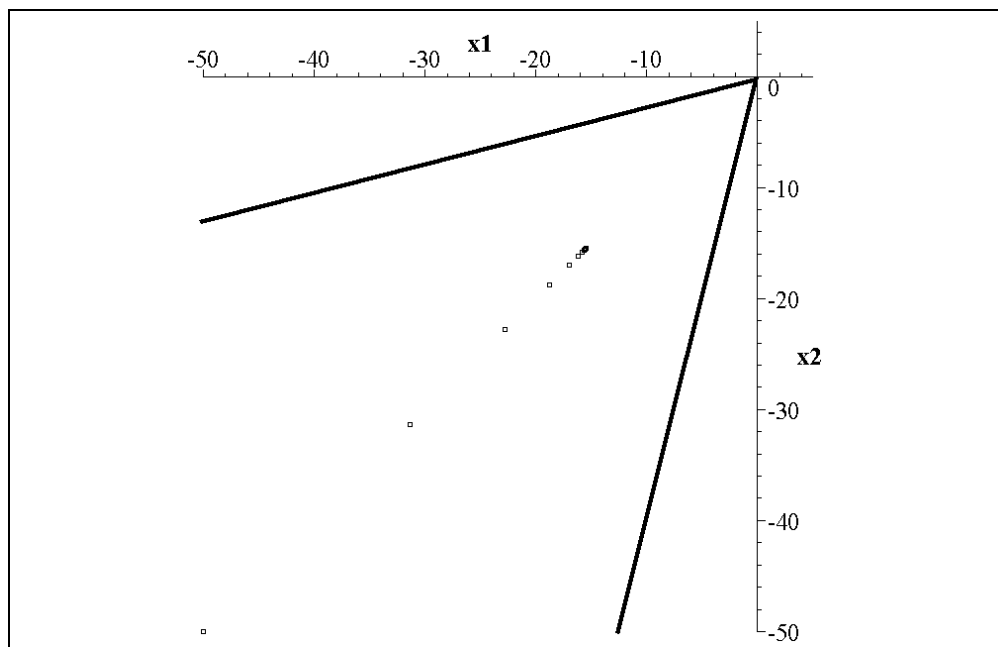


Figura III.11: $\alpha = \pi/6, \mu = 3, d = 1$

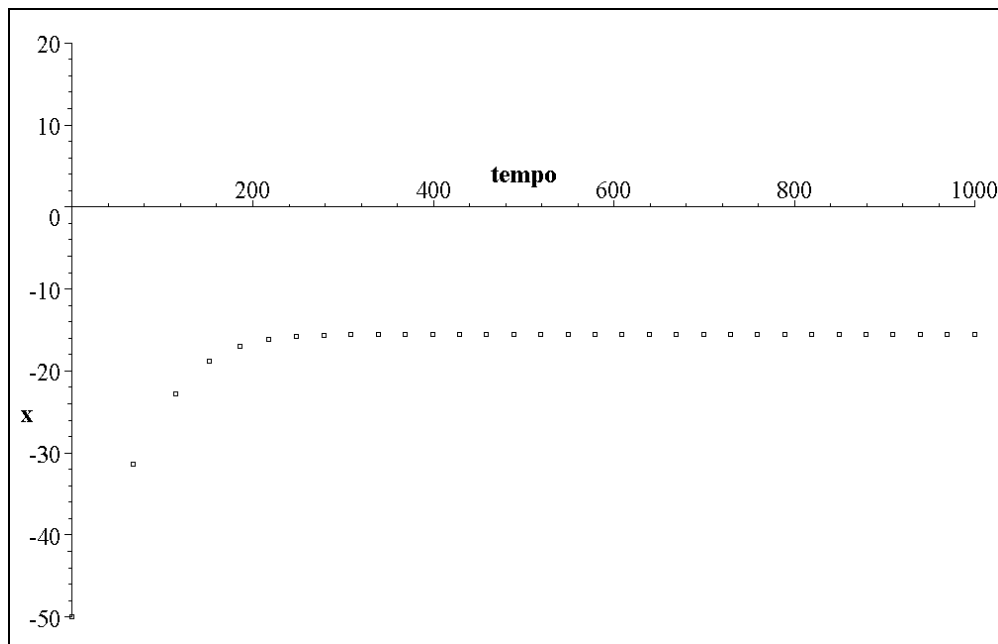


Figura III.12: $\alpha = \pi/6, \mu = 3, d = 1$

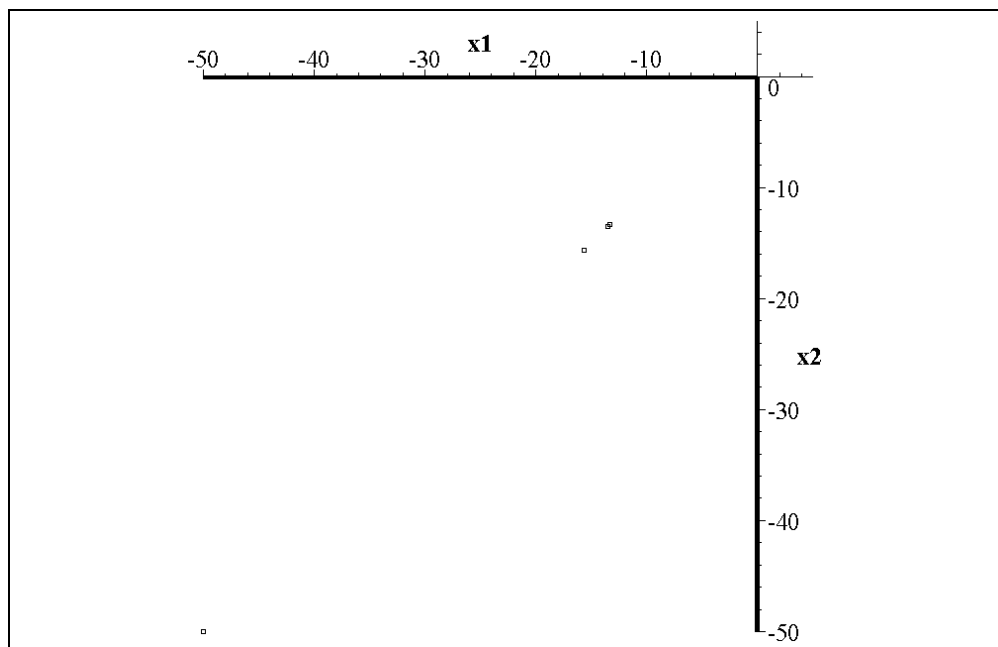


Figura III.13: $\alpha = \pi/4$, $\mu = 10$, $d = 2$

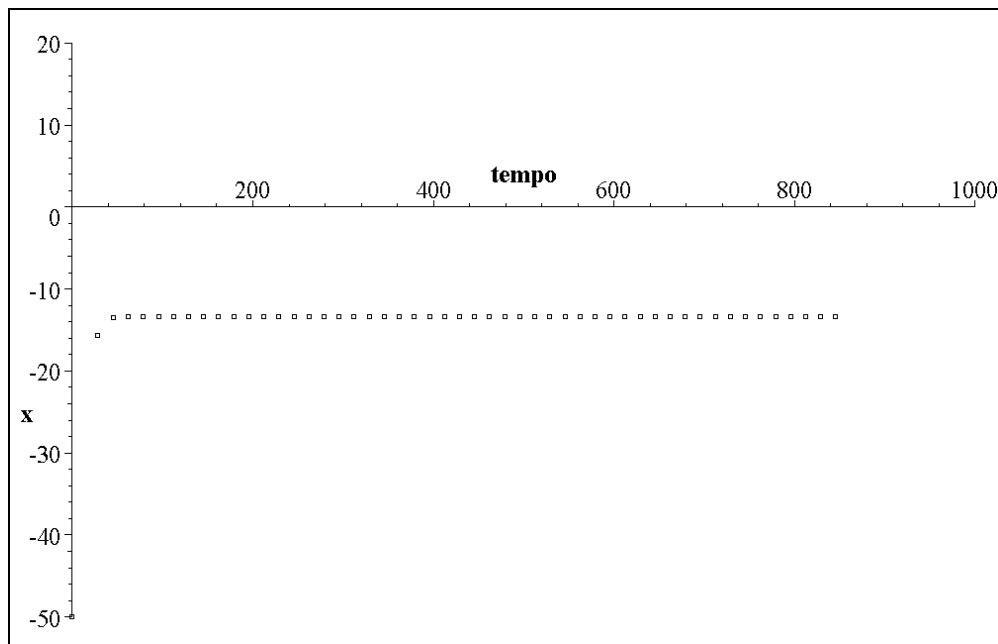


Figura III.14: $\alpha = \pi/4$, $\mu = 10$, $d = 2$

Si noti che la seconda equazione delle III.14 converge ad un valore costante pari alla durata del ciclo limite.

Una cosa interessante, sulla quale concentrare l'attenzione, è la dipendenza dell'autovalore III.15 dall'angolo α del corridoio una volta fissati i tassi μ e d . Ad esempio per gli stessi valori da cui sono state ottenute le figure III.9 e III.10 si ottiene l'andamento in figura III.15, dal quale si evince

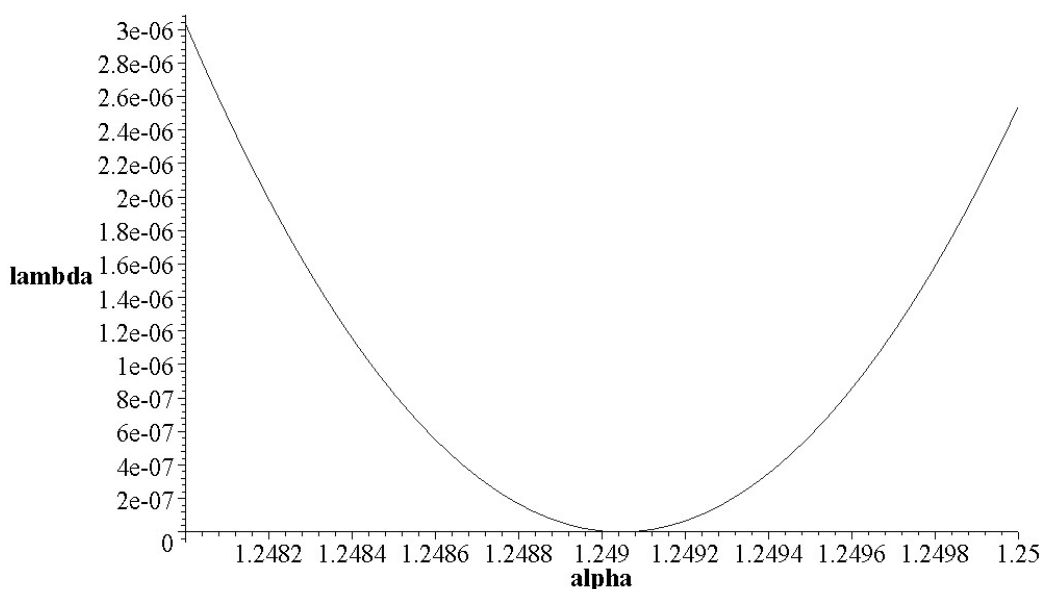


Figura III.15: Andamento dell' autovalore in funzione dell'angolo.

l'esistenza di un valore minimo per λ pari a 0. Ciò significa che scegliendo un corridoio di controllo simmetrico, con un angolo di apertura totale pari al doppio del valore di α per il quale risulta $\lambda = 0$, è possibile raggiungere il punto di regime in un numero finito di passi, in questo caso uno, ricordando la teoria dei sistemi lineari a tempo discreto (si veda ad esempio [21]). Per

esempio ponendo α pari al valore 1,249, e usando per d e μ gli stessi valori del grafico delle figure III.9 e III.10, si ottengono gli andamenti della figura III.16 a pagina 89.

In realtà, è possibile risolvere in forma chiusa l'equazione $\lambda = 0$, ottenendo una funzione $\alpha = f(\mu, d)$ che fornisce per la generica coppia (μ, d) il valore dell'angolo di apertura del corridoio, per mezzo del quale è possibile raggiungere in un unico passo il ciclo limite corrispondente (cioè quello che si *incastra* nel corridoio iperconico), partendo da un qualsiasi punto iniziale sulla bisettrice.

Le soluzioni di $\lambda = 0$ sono le stesse che annullano il numeratore dell'autovalore. La discussione delle radici del denominatore sarà fatta successivamente, adesso si ponga l'attenzione sugli zeri del numeratore, cioè sulle soluzioni dell'equazione:

$$\mu^2 + 4d^2 - 4d^2 \cos^2\alpha - 4d\mu + 4d\mu \cos^2\alpha - 2\mu^2 \sin\alpha \cos\alpha + 4d\mu \sin\alpha \cos\alpha = 0.$$

Dividendo tutto per $\cos^2\alpha$ e ricordando che $\frac{1}{\cos^2\alpha} = 1 + \tan^2\alpha$, si ottiene l'equazione:

$$\tan^2\alpha (\mu^2 + 4d^2 - 4d\mu) + \tan\alpha (4d\mu - 2\mu^2) + \mu^2 = 0, \quad (\text{III.16})$$

che è equivalente alla precedente per tutti quei valori di α per i quali risulta $\cos^2\alpha \neq 0$, cioè per $\alpha \neq \frac{\pi}{2} + k\pi$ con $k = 0, 1, 2, \dots$

Se si effettua la sostituzione:

$$x = \tan\alpha,$$

l'equazione III.16 diventa una semplice equazione di II grado in x :

$$ax^2 + bx + c = 0$$

con:

$$a = \mu^2 + 4d^2 - 4d\mu$$

$$b = 4d\mu - 2\mu^2$$

$$c = \mu^2$$

le cui soluzioni sono:

$$x_{1-2} = \frac{2\mu^2 - 4d\mu}{2\mu^2 + 8d^2 - 8d\mu},$$

cioè l'equazione $\lambda = 0$ ammette due soluzioni coincidenti date da:

$$\alpha_{1-2} = \arctan\left(\frac{2\mu^2 - 4d\mu}{2\mu^2 + 8d^2 - 8d\mu}\right).$$

Ad esempio, per $\mu = 3$ e $d = 1$, si ottiene il valore:

$$\alpha = 1.249045\dots$$

che corrisponde con quello ricavato per ispezione dalla figura III.15.

Con un analogo procedimento si ricavano le radici del denominatore dell'autovalore:

$$\bar{x}_{1-2} = \frac{4d\mu - 2\mu^2}{2\mu^2 + 8d^2 - 8d\mu},$$

e quindi:

$$\arctan\left(\frac{4d\mu - 2\mu^2}{2\mu^2 + 8d^2 - 8d\mu}\right),$$

che risultano essere opposte a quelle trovate per il numeratore.

Se si considera un sistema manifatturiero con una struttura di tipo *make to order*, la cui traiettoria evolve soltanto all'interno del III quadrante, è possibile dimostrare che partendo da una condizione iniziale posta sulla bisettrice non esiste nessun corridoio, o meglio nessun valore di α , con $0 < \alpha \leq \frac{\pi}{4}$, capace di portare la traiettoria sul ciclo limite in un passo solo. In questa situazione i vincoli sono:

$$0 < \alpha \leq \frac{\pi}{4}$$

$$\mu \geq 0$$

$$d > 0$$

$$\mu > 2d.$$

Si vuole dimostrare che non è possibile trovare nessuna coppia (d, μ) che soddisfa i vincoli e che rende nullo l'autovalore; essendo $x = \tan\alpha$ si ha che:

$$0 < \alpha \leq \frac{\pi}{4} \quad \Leftrightarrow \quad 0 < x \leq 1.$$

Si deve dimostrare che $x_{1-2} > 1$ per ogni (d, μ) :

$$\frac{2\mu^2 - 4d\mu}{2\mu^2 + 8d^2 - 8d\mu} \stackrel{?}{>} 1,$$

da cui:

$$2\mu^2 - 4d\mu \stackrel{?}{>} 2\mu^2 + 8d^2 - 8d\mu,$$

semplificando:

$$\mu - 2d \stackrel{?}{>} 0.$$

La quale essendo verificata per ipotesi dimostra la validità della proposizione $x_{1-2} > 1$ per ogni (d, μ) .

Con ciò non esiste nessun valore di α , con α compreso tra 0 e $\frac{\pi}{4}$, e quindi nessun corridoio contenuto totalmente nel III quadrante, capace di generare delle traiettorie del surplus che raggiungono il ciclo limite la cui posizione nello spazio è funzione dell'apertura del corridoio, in un numero finito di passi.

Da notare che se si sceglie $\alpha = \frac{\pi}{4}$, cioè un corridoio coincidente con la parte di piano compresa tra gli assi coordinati, si ottiene un ciclo limite la cui posizione coincide con quella che minimizza l'indice di costo II.23, a pagina

54. Da ciò emerge che non è possibile utilizzare una politica corridoio basata su un corridoio simmetrico per raggiungere il ciclo limite ottimo in un numero finito di passi. La convergenza perciò, sarà sempre di tipo asintotico.

Controllando la traiettoria del surplus con un corridoio che permette di raggiungere il ciclo limite in un numero finito di passi si minimizza il numero dei cambiamenti di setup, e quindi si sfrutta al massimo la capacità produttiva della macchina impiegando il più breve tempo possibile per raggiungere la condizione stazionaria di regime, ovvero in questo caso il ciclo limite. Ma ciò ha anche un risvolto negativo: si effettua un numero ridotto di cambi della configurazione della macchina, ottenendo durante il transitorio delle sequenze di lavorazione di pezzi dello stesso tipo molto lunghe, e di conseguenza delle ampie oscillazioni sia dei livelli dei magazzini che del numero delle domande in attesa.

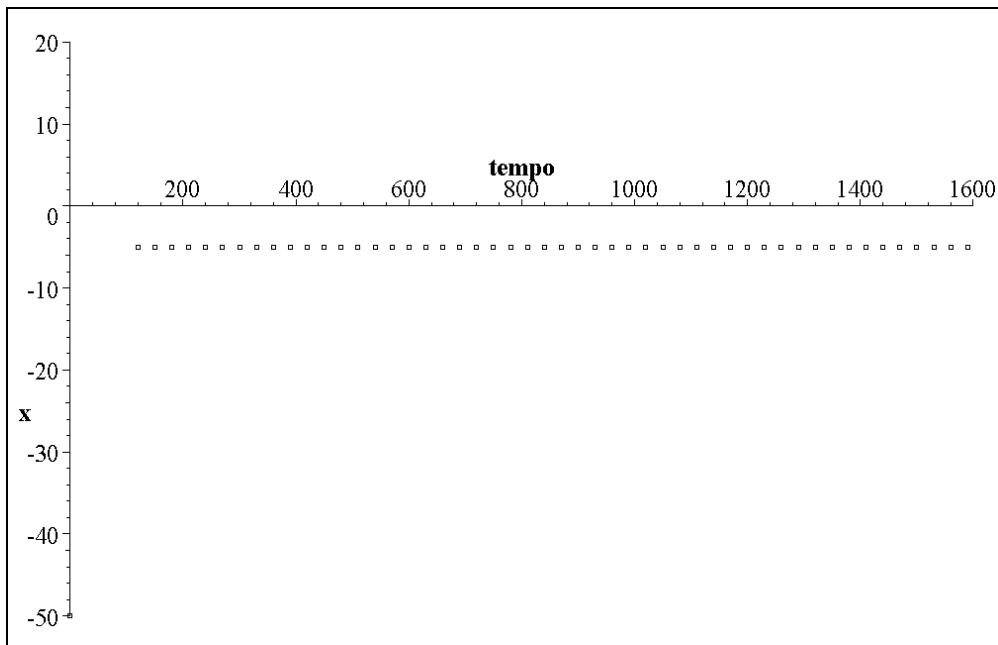
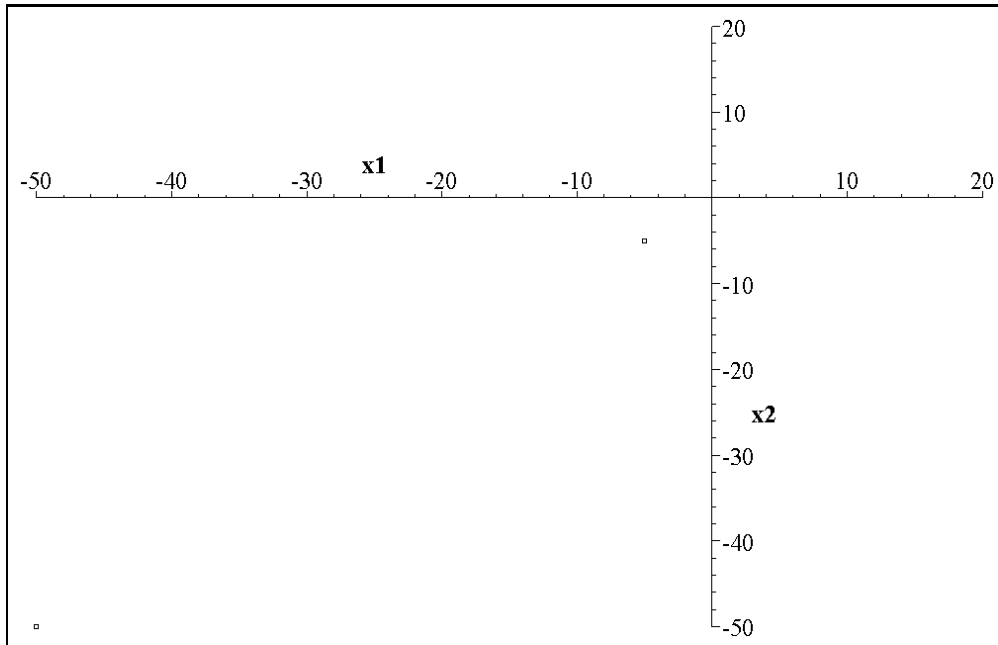


Figura III.16: Dinamiche in un passo con $\alpha = 1,249$, $\mu = 3$, $d = 1$.

III.3 Scelta dei corridoi e stabilità dei cicli limite

Si prenda in considerazione un sistema di produzione composto da una sola macchina, capace di lavorare n tipi differenti di parti, una alla volta. Si assuma che la macchina impieghi dei tempi di setup nel cambiare la sua configurazione. Inoltre, si assuma che la macchina debba seguire una traiettoria di riferimento.

Si costruisca un corridoio iperconico, convesso e lineare, con n facce attorno al riferimento. Si associ ogni faccia del corridoio ad un setup: ciò serve a determinare l'istante di inizio del cambio di configurazione della macchina in quel particolare setup. È facile rendersi conto che, per un dato corridoio, la traiettoria del surplus è una funzione del tempo “lineare a tratti” le cui discontinuità corrispondono alle intersezioni con le facce del corridoio. Per determinare la dinamica di \mathbf{x} , quindi, è sufficiente trovare il comportamento dei punti di intersezione della traiettoria con le facce del corridoio. Sia $\mathbf{x}(t_k)$, con $k = 1, 2, \dots, n$, la k -esima intersezione della traiettoria con una faccia del corridoio arbitrariamente scelta. È stata dimostrata (per approfondimenti sui teoremi e sulle proposizioni riportati in questa sezione si veda [4]) la seguente proposizione:

Proposizione 1 *Se le facce di un corridoio convesso lineare sono tali da generare una sequenza dei setup round-robin e periodica, allora $\mathbf{x}(t_k)$ soddisfa una equazione di stato lineare, invariante rispetto al tempo*

$$\mathbf{x}(t_{k+1}) = A\mathbf{x}(t_k) + B\mathbf{s} + C\mathbf{b} \quad (\text{III.17})$$

dove A, B , e C sono matrici costanti, \mathbf{s} è il vettore dei tempi dei setup, e \mathbf{b} è un vettore costante associato agli iperpiani che formano il corridoio.

A questo punto è necessario fare delle ipotesi, comunque non restrittive, sui corridoi. È stato detto in precedenza che il corridoio è iperconico con n facce, si assuma inoltre che il suo vertice coincida con l'origine degli assi. Questa situazione può essere ottenuta, ad esempio, semplicemente traslando il sistema delle coordinate. L'ipercono può essere descritto analiticamente dalle equazioni:

$$\mathbf{a}_i^t \mathbf{x} \leq 0 \quad i = 1, 2, \dots, n \quad (\text{III.18})$$

Quando la traiettoria raggiunge dall'interno del corridoio la faccia $\mathbf{a}_i^t \mathbf{x} = 0$ per un certo i , scatta il cambiamento del setup della macchina, e il nuovo setup sarà quello associato all'iperpiano appena toccato. Il prossimo teorema fornisce una condizione sufficiente sui vettori \mathbf{a}_i (che sono i gradienti dei vari iperpiani che costituiscono il corridoio), quindi sulla forma del corridoio, affinché la sequenza dei setup sia periodica e round-robin:

Teorema 1 *Un corridoio iperconico genera una sequenza $1, 2, \dots, n$ dei setup round-robin e periodica, se valgono le seguenti condizioni:*

1. Per tutti gli indici i e j , con $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$, e $i \neq j$,

$$\begin{aligned} \mathbf{a}_j^t \mathbf{v}_i &> 0 \quad \text{se } j = i \pmod{n} + 1, \\ \mathbf{a}_j^t \mathbf{v}_i &\leq 0 \quad \text{se } j \neq i \pmod{n} + 1. \end{aligned} \quad (\text{III.19})$$

2. Per tutti gli indici i ,

$$\mathbf{a}_i^t \mathbf{v}_0 \leq 0, \quad (\text{III.20})$$

dove \mathbf{v}_i e \mathbf{v}_j sono le velocità della traiettoria del surplus rispettivamente nel setup i e nel setup j , e \mathbf{v}_0 è la velocità della macchina quando c'è un cambiamento di setup in corso.

È opportuno dare un'interpretazione geometrica delle condizioni di sufficienza del teorema 1.

Condizione 1. Se l'indice j è immediatamente successivo all'indice i (ad esempio $i = 2$ e $j = 3$) allora deve essere soddisfatta la relazione $\mathbf{a}_j^t \mathbf{v}_i > 0$, cioè l'angolo compreso tra il gradiente dell'iperpiano j (il vettore \mathbf{a}_j) e la velocità del surplus nel setup i deve essere inferiore a $\frac{\pi}{2}$; ciò deriva direttamente dalla definizione di prodotto scalare tra due vettori, che qui si richiama:

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = |\mathbf{v}_1| |\mathbf{v}_2| \cos(\widehat{\mathbf{v}_1 \mathbf{v}_2}) = \mathbf{v}_1^t \mathbf{v}_2$$

dove $\widehat{\mathbf{v}_1 \mathbf{v}_2}$ è l'angolo minore di π compreso tra i vettori \mathbf{v}_1 e \mathbf{v}_2 . Quando la macchina è nel setup i , questa condizione assicura alla traiettoria la possibilità di raggiungere l'iperpiano $j = i + 1$. Viceversa, se l'indice j non è successivo all'indice i (ad esempio $i = 2$ e $j = 4$) allora deve valere $\mathbf{a}_j^t \mathbf{v}_i \leq 0$, cioè l'angolo tra il gradiente dell'iperpiano e la velocità deve essere maggiore, o al massimo pari a $\frac{\pi}{2}$. Questo significa geometricamente che quando la macchina è nel setup i , la traiettoria si allontana dall'iperpiano j , o al limite

procede parallelamente ad esso.

Condizione 2. Per tutti gli indici i deve valere $\mathbf{a}_i^t \mathbf{v}_0 \leq 0$, cioè, sempre ricorrendo alla interpretazione tramite il prodotto scalare tra due vettori, la traiettoria del surplus durante un qualsiasi cambio del setup della macchina non deve mai uscire al di fuori del corridoio iperconico.

Non è difficile scegliere degli iperpiani che soddisfano le ipotesi del teorema 1, infatti, i corridoi di tipo “*clearing*” per esempio, le soddisfano, come si può facilmente verificare.

A questo punto è possibile introdurre e dimostrare l’esistenza e la stabilità dei cicli limite associati ai corridoi iperconici. Prima però è opportuno specificare meglio il significato che assume la parola stabilità nel contesto del controllo in retroazione dei sistemi manifatturieri. In letteratura, una ottima definizione di stabilità di una politica di scheduling dei setup, è formulata in [2], ed è riportata qui di seguito:

Definizione 1 *Una politica di controllo è stabile se risulta*

$$\sup_t x_i(t) \leq M < \infty, \quad \text{per ogni } i = 1, 2, \dots, n$$

dove con $x_i(t)$ si intende la differenza tra produzione cumulativa e la domanda cumulativa della parte di tipo i , e n rappresenta il numero totale dei tipi di parte.

Si noti che $x_i(t)$ rappresenta il surplus della produzione della parte di tipo i all’istante t , ovvero il livello del suo *work in progress*, cioè il livello

istantaneo del magazzino associato. Quindi in altre parole, la definizione 1 afferma che una politica di controllo dello scheduling dei setup è stabile se riesce a mantenere limitati nel tempo i livelli di tutti i magazzini. Nel prossimo teorema, sempre tratto da [4], vengono enunciate e dimostrate condizioni sufficienti affinché la traiettoria del surplus, controllata attraverso un corridoio iperconico, converga ad *un ciclo limite stabile*, cioè sarà dimostrata la stabilità asintotica del sistema lineare III.17 a pagina 90. Questo significa che il vettore $\mathbf{x}(t)$ tenderà, per t che tende all'infinito, a un ben preciso punto dello spazio dei surplus, differente da infinito. Ciò assicura che i livelli dei magazzini saranno limitati da una costante finita, concordando perfettamente con la definizione 1 di stabilità.

Teorema 2 *Per una politica di controllo che usa un corridoio iperconico il quale soddisfa le ipotesi del teorema 1, la traiettoria del surplus converge ad un ciclo limite stabile e unico, ammesso che il vettore dei tassi delle domande della macchina soddisfi la condizione*

$$\sum_{i=1}^m \tau_i d_i < 1, \quad (\text{III.21})$$

dove τ_i è il tempo di processamento della parte i sulla macchina.

La condizione III.21 esprime il fatto che la domanda, o meglio il vettore \mathbf{d} delle domande, deve stare all'interno dell'insieme di capacità associato ad una macchina simile a quella considerata, ma senza i tempi di setup, quindi perfettamente flessibile. Un esempio di questa situazione è riportato in figura

II.7 a pagina 44.

Stare su un ciclo limite comporta principalmente due vantaggi:

1. la domanda è automaticamente soddisfatta;
2. i cambiamenti della configurazione della macchina sono periodici secondo un fissato ciclo di produzione.

La seconda proprietà implica una grande regolarità di funzionamento della macchina, e quindi una facilità di implementazione.

È importante notare che non sempre l'obiettivo primario del controllo della macchina è di far rimanere il surplus sul ciclo limite, ad esempio il verificarsi di un evento discreto ai livelli superiori della gerarchia può cambiare i riferimenti per il livello dei setup in qualsiasi istante. Questo può comportare due differenti tipi di conseguenze:

1. il ciclo limite cambia forma (ad esempio in seguito ad un mutamento del tasso della domanda), e la traiettoria del surplus viene a trovarsi in un punto che non appartiene più al ciclo limite ottimo;
2. la forma del ciclo limite ottimo rimane invariata assieme alla sua posizione nello spazio dei surplus, ma la traiettoria è costretta ad allontanarsi (ad esempio in seguito a una rottura della macchina).

Quando la frequenza di questi eventi aumenta, allora diventa molto più importante migliorare il modo in cui la traiettoria raggiunge il ciclo limite. Questa problematica sarà affrontata nei capitoli IV e V.

Capitolo IV

Il controllo durante il transitorio

IV.1 Introduzione del problema

Il controllo di un sistema manifatturiero composto da una sola macchina caratterizzata da tempi di setup non trascurabili, il più delle volte, ha per obiettivo quello di far stare la traiettoria del surplus sul ciclo limite che minimizza l'indice di costo scelto.

Come si è detto alla fine del capitolo III, lo stare sul ciclo limite ottimo comporta numerosi vantaggi. Non sempre però è possibile condurre la macchina su un ciclo limite per lunghi periodi di tempo. Il verificarsi di eventi associati ai livelli superiori della gerarchia, quindi caratterizzati da frequenze inferiori a quelle con le quali si cambia il setup, può costringere la macchina ad abbandonare il ciclo limite oppure può cambiare la forma stessa del ciclo limite.

L'esempio classico è quello della rottura della macchina, o di una parte di essa che comunque ne impedisce il completo funzionamento. Se si ipotizza che la traiettoria si trova nel punto G nell'istante in cui si verifica il guasto alla macchina, l'andamento del surplus sarà del tipo: Il punto F corrisponde alla

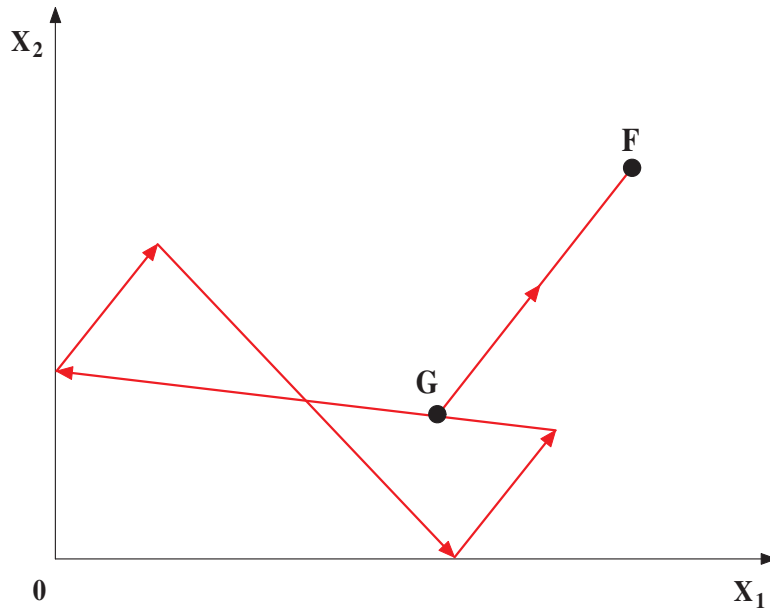


Figura IV.1: Andamento della traiettoria in caso di guasto

fine della riparazione, dove la macchina è pronta per riprendere a lavorare.

Si noti che:

- la durata del tratto \overline{GF} è pari al tempo impiegato per riparare la macchina. È irrilevante in questo contesto specificare la natura deterministica o stocastica di questo intervallo temporale;
- il segmento \overline{GF} è sempre parallelo al vettore \mathbf{v}_O che rappresenta la velocità del sistema quando tutti i tassi di produzione della macchina sono nulli, quindi sia durante un cambio del setup, sia dopo un guasto.

Un'altro esempio di quello che potrebbe accadere è rappresentato da un'improvviso cambiamento del valore della domanda. La variazione di un qualsiasi tasso di domanda ha per effetto il cambio della forma del ciclo limite,

quindi anche di quello nella posizione ottima. Se si generalizza il significato dei punti G e F precedentemente introdotti assumendo che G sia l'ultimo punto che il surplus percorre sulla traiettoria ottima e F sia il punto da cui riparte la macchina per raggiungere il ciclo limite, allora tali punti in questo caso vengono a coincidere, e la macchina si viene a trovare in un punto dello spazio dei surplus che repentinamente non appartiene più alla traiettoria di regime ottima.

Una volta che la macchina è nel punto F pronta per lavorare, indipendentemente dalla natura dell'evento che ha mutato la situazione, nasce un nuovo problema: quale è il modo migliore per ritornare sul ciclo limite ottimo? Ovvero come si può ottimizzare il comportamento del sistema anche durante il transitorio?

Purtroppo la funzione di costo scelta per ricavare il ciclo limite ottimo non impone alcuna condizione sul transitorio, perciò si devono necessariamente seguire altre strade. Il problema che si vuole risolvere è quindi il seguente:

ammesso di aver trovato la traiettoria ottima per quanto riguarda il regime stazionario, dato un punto iniziale F che non appartiene a tale traiettoria, quale è il modo migliore di condurre la macchina da F alla traiettoria ottima stazionaria?

A questo interrogativo si cercherà di dare una risposta nei prossimi capitoli utilizzando lo strumento della *programmazione dinamica* e ricavando successivamente delle proprietà analitiche della soluzione ottima.

IV.2 Uso della programmazione dinamica come tecnica risolutiva

La programmazione dinamica è un approccio generale alla soluzione dei problemi di ottimizzazione. Essa è molto utile quando è necessario prendere delle decisioni lungo un certo intervallo temporale. I problemi di programmazione dinamica sono delle generalizzazioni dei problemi del *calcolo delle variazioni* e di *teoria del controllo*. Questa tecnica è particolarmente interessante perché conduce a delle soluzioni caratterizzate da una struttura a retroazione: le decisioni vengono prese in risposta al verificarsi dei vari eventi.

All'interno della formulazione di un problema di programmazione dinamica, come in tutti i problemi di ottimizzazione, è possibile distinguere i seguenti elementi fondamentali:

- lo *stato*;
- il *controllo*;
- la *funzione obiettivo*;
- le *dinamiche*;
- i *vincoli*;
- le *condizioni iniziali*.

Lo *stato* è formato da tutte le informazioni necessarie per la determinazione dell'evoluzione temporale del sistema. Ma anche quando lo stato è conosciuto, il futuro non può essere previsto con assoluta certezza, in quanto possono

sempre accadere eventi imprevedibili.

Le azioni di *controllo* sono scelte da colui che nel problema ha il compito di prendere le decisioni, ovvero il *decisore*. Esse sono conosciute anche con il termine di *variabili decisionali*.

La *funzione obiettivo* rappresenta la quantità che deve essere minimizzata, ovvero la misura di quanto buona è la soluzione proposta. Essa è anche chiamata *funzione di costo* o *criterio*.

Le *dinamiche* sono necessarie per la descrizione dell'evoluzione dello stato in funzione delle variabili di controllo e degli eventi stocastici.

I *vincoli* limitano l'insieme dei controlli (cioè delle soluzioni) ammissibili.

Le *condizioni iniziali* sono i valori delle variabili di stato all'inizio dell'intervallo temporale considerato dal problema. Alcune volte, al loro posto è possibile trovare le *condizioni finali*.

Esistono differenti tipi di problemi che possono essere affrontati con la programmazione dinamica. Ad esempio il tempo può essere *continuo* o *discreto*, lo stato *continuo*, *discreto*, o *misto*. Infine, i problemi possono avere un carattere *deterministico* oppure *stocastico*.

I problemi deterministici a tempo continuo e con le variabili di stato continue sono chiamati *calcolo delle variazioni*, e sono stati lungamente studiati in passato. Invece i problemi con lo stato continuo e il tempo discreto, oppure vincoli di disuguaglianza, oppure elementi stocastici, vengono chiamati problemi di *controllo ottimo*.

Esiste una differenza concettuale molto profonda tra l'approccio della programmazione dinamica e quello del calcolo delle variazioni. Il calcolo

delle variazioni ha per scopo la ricerca di soluzioni $\mathbf{u}(t)$ che sono solamente funzioni del tempo. Ciò è talvolta anche chiamato *controllo ad anello aperto* (*open loop control*). La programmazione dinamica invece, cerca soluzioni del problema aventi la forma $\mathbf{u}(\mathbf{x})$, ossia delle funzioni dello stato. Il termine usato per caratterizzare questo tipo di soluzioni è *controllo a ciclo chiuso* (*closed loop control*).

Se il sistema che si sta studiando è deterministico, il metodo del calcolo delle variazioni fornisce una soluzione a ciclo aperto $\mathbf{u}(t)$ che è più che soddisfacente. Se invece lo stesso problema lo si affronta utilizzando la programmazione dinamica, si trova una soluzione $\mathbf{u}[\mathbf{x}(t), t]$ il cui andamento temporale coincide con la precedente soluzione, ma la tecnica utilizzata per ricavarla è differente, ed è quella più appropriata quando il problema è stocastico.

Talvolta la retroazione è utile quando il sistema reale è stocastico ma può essere descritto attraverso un modello deterministico. In questo caso accade che l'aspetto aleatorio del sistema è sufficientemente limitato da consentire una predizione soddisfacente del comportamento del modello deterministico. Ma allo stesso tempo l'aleatorietà è grande abbastanza da generare dei discostamenti tra il comportamento effettivo del sistema e la predizione deterministica: il controllo a retroazione è in grado di correggere questi scostamenti. In tutti questi casi non è necessario modellare con precisione gli aspetti aleatori del sistema, il controllo calcolato attraverso il modello deterministico può comunque assicurare un *buon* comportamento del sistema.

A causa della diversità dei tipi di problemi non esiste una formulazione generale di programmazione dinamica che comprende tutti i problemi come

casi particolari. Una classe importante è quella deterministica a tempo continuo e stato continuo. La sua formulazione è del tipo:

$$\min_{\substack{\mathbf{u}(t) \\ 0 \leq t \leq T}} \int_0^T g[\mathbf{x}(t), \mathbf{u}(t)] dt + F[\mathbf{x}(T)]$$

tale che

$$\frac{d\mathbf{x}(t)}{dt} = f[\mathbf{x}(t), \mathbf{u}(t), t] \quad (\text{IV.1})$$

$\mathbf{x}(0)$ dato

$$h[\mathbf{x}(t), \mathbf{u}(t)] \leq 0$$

dove: $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $f \in \mathbb{R}^n$, $h \in \mathbb{R}^k$, g e F sono scalari. La funzione F è chiamata costo finale. I dati del problema sono: T , $\mathbf{x}(0)$, le funzioni f , g , h e F . L'obiettivo è trovare le funzioni \mathbf{u} e \mathbf{x} .

Una seconda classe, altrettanto importante quanto la precedente, è quella dei problemi sempre deterministici con stato continuo ma a tempo discreto:

$$\min_{\substack{\mathbf{u}(i) \\ 0 \leq i \leq T-1}} \sum_{i=0}^{T-1} g[\mathbf{x}(i), \mathbf{u}(i)] dt + F[\mathbf{x}(T)]$$

tale che

$$\mathbf{x}(i+1) = f[\mathbf{x}(i), \mathbf{u}(i), i] \quad (\text{IV.2})$$

$\mathbf{x}(0)$ dato

$$h[\mathbf{x}(i), \mathbf{u}(i)] \leq 0$$

dove valgono gli stessi commenti fatti per il problema IV.1. Si noti che il

problema IV.2 è un problema di programmazione non lineare. Se f , g , h e F fossero lineari, esso sarebbe un problema di programmazione lineare. Se fosse risolto attraverso i metodi normali esso fornirebbe le funzioni $\mathbf{x}(i)$ e $\mathbf{u}(i)$ ma non la funzione $\mathbf{u}[\mathbf{x}(i)]$, la quale può essere usata per implementare una politica di controllo a retroazione.

Cost-to-go

Si supponga di aver risolto un problema di programmazione dinamica, il costo ottimo ottenuto lo si chiami J . Si può risolvere un identico problema partendo da una differente condizione iniziale: il costo ottimo probabilmente cambierà. E cambierà anche se si parte da un differente istante iniziale. Sia dunque $J[\mathbf{x}(s), s]$ il valore della funzione di costo che si ottiene risolvendo il problema di programmazione dinamica partendo all'istante s e dal punto $\mathbf{x}(s)$. Questa funzione è chiamata in letteratura *cost-to-go* o *funzione valore*.

Tempo discreto, stato continuo, deterministico

Si supponga di aver risolto un problema di programmazione dinamica. È possibile considerare un nuovo problema, identico al primo in tutto tranne che per lo stato iniziale e l'istante iniziale: li si scelga sulla traiettoria ottima trovata per il primo problema. Siano:

- $\mathbf{x}(0), 0$ lo stato e l'istante iniziale del primo problema;
- $\mathbf{x}(t_1), t_1$ lo stato e l'istante iniziale del secondo problema;
- $\mathbf{x}^*(t), \mathbf{u}^*(t)$ le soluzioni del primo problema;

allora $\mathbf{x}(t_1)$ e t_1 sono tali che:

$$\mathbf{x}(t_1) = \mathbf{x}^*(t_1).$$

Il principio di ottimalità assicura che la soluzione del secondo problema è una parte della soluzione del primo problema: per la precisione è la parte che comincia all'istante t_1 . In altre parole, se si viene a sapere che una certa coppia $\mathbf{x}(t_1), t_1$ appartiene alla traiettoria ottima, allora si può ricavare la soluzione dopo t_1 risolvendo lo stesso problema di partenza ma con condizione iniziale $\mathbf{x}(t_1)$ all'istante t_1 . Ciò ha un'importante conseguenza: se si suppone di conoscere un punto $(\mathbf{x}(t_1), t_1)$ della traiettoria ottima e anche la soluzione del problema con condizione iniziale in $(\mathbf{x}(t_1), t_1)$, allora è necessario calcolare soltanto la parte di traiettoria che rimane, cioè quella compresa tra gli istanti 0 e t_1 .

Si supponga di conoscere $J(\mathbf{x}, t_1)$ per qualche $0 < t_1 < T$ e per ogni valore di \mathbf{x} . Per determinare la traiettoria da 0 a t_1 è necessario risolvere il seguente problema:

$$\min_{\substack{\mathbf{u}(i) \\ 0 \leq i \leq t_1 - 1}} \sum_{i=0}^{t_1-1} g[\mathbf{x}(i), \mathbf{u}(i)] dt + J[\mathbf{x}(t_1), t_1]$$

tale che

$$\mathbf{x}(i+1) = f[\mathbf{x}(i), \mathbf{u}(i), i] \tag{IV.3}$$

$$\mathbf{x}(0) \quad \text{dato}$$

$$h[\mathbf{x}(i), \mathbf{u}(i)] \leq 0$$

Si noti che questo problema ha la stessa forma di IV.2 con t_1 al posto di T e $J[\mathbf{x}(t_1), t_1]$ al posto di $F[\mathbf{x}(T)]$. Il principio di ottimalità vincola il valore di $\mathbf{x}(t_1)$ dell'ultimo problema ad essere pari al valore $\mathbf{x}(t_1)$ del problema originario, di conseguenza il valore $J[\mathbf{x}(0), 0]$ dell'ultimo problema sarà uguale al $J[\mathbf{x}(0), 0]$ del problema originario. In formule:

$$J[\mathbf{x}(0), 0] = \min_{\substack{\mathbf{u}(i) \\ 0 \leq i \leq t_1 - 1}} \left\{ \sum_{i=0}^{t_1-1} g[\mathbf{x}(i), \mathbf{u}(i)] + J[\mathbf{x}(t_1), t_1] \right\}. \quad (\text{IV.4})$$

Della IV.4 è possibile ricavare anche la versione un passo (*one-step*), infatti partendo dall'istante $t_1 - 1$ invece che da 0 si ottiene:

$$J[\mathbf{x}(t_1 - 1), t_1 - 1] = \min_{\mathbf{u}(t_1-1)} \left\{ g[\mathbf{x}(t_1 - 1), \mathbf{u}(t_1 - 1)] + J[\mathbf{x}(t_1), t_1] \right\}. \quad (\text{IV.5})$$

Quest'ultima equazione afferma che se si conosce $J[\mathbf{x}(t_1), t_1]$ allora è possibile ricavare $J[\mathbf{x}(t_1 - 1), t_1 - 1]$. Ponendo:

$$J[\mathbf{x}(T), T] = F[\mathbf{x}(T)]$$

è possibile, almeno in linea di principio, generare l'intera funzione J e il relativo controllo ottimo partendo dall'istante finale T e risolvendo ricorsivamente all'indietro.

Tempo continuo, stato continuo, deterministico

Si supponga di dover considerare un problema del tipo IV.1. È possibile definire:

$$J[\mathbf{x}(0), 0] = \min \int_0^T g[\mathbf{x}(t), \mathbf{u}(t)] dt + F[\mathbf{x}(T)] \quad (\text{IV.6})$$

considerando anche tutti i vincoli presenti nella IV.1. Generalizzando:

$$J(\mathbf{x}, t) = \min \int_t^T g[\mathbf{x}(s), \mathbf{u}(s)] ds + F[\mathbf{x}(T)]. \quad (\text{IV.7})$$

Si prenda un istante t_1 tale che $0 < t_1 < T$. Analogamente al caso a tempo discreto si supponga di conoscere la funzione $J(\mathbf{x}, t_1)$. Allora è possibile scrivere:

$$J[\mathbf{x}(0), 0] = \min_{\substack{\mathbf{u}(t) \\ 0 \leq t \leq T}} \left\{ \int_0^{t_1} g[\mathbf{x}(t), \mathbf{u}(t)] dt + \int_{t_1}^T g[\mathbf{x}(t), \mathbf{u}(t)] dt + F[\mathbf{x}(T)] \right\} \quad (\text{IV.8})$$

spezzando la minimizzazione si ottiene:

$$J[\mathbf{x}(0), 0] = \min_{\substack{\mathbf{u}(t) \\ 0 \leq t \leq t_1}} \left\{ \int_0^{t_1} g[\mathbf{x}(t), \mathbf{u}(t)] dt + \min_{\substack{\mathbf{u}(t) \\ t_1 \leq t \leq T}} \left[\int_{t_1}^T g[\mathbf{x}(t), \mathbf{u}(t)] dt + F[\mathbf{x}(T)] \right] \right\} \quad (\text{IV.9})$$

da cui:

$$J[\mathbf{x}(0), 0] = \min_{\substack{\mathbf{u}(t) \\ 0 \leq t \leq t_1}} \int_0^{t_1} g[\mathbf{x}(t), \mathbf{u}(t)] dt + J[\mathbf{x}(t_1), t_1] \quad (\text{IV.10})$$

questo problema ha la stessa forma di IV.1 ma con t_1 al posto di T e $J[\mathbf{x}(t_1), t_1]$ al posto di $F[\mathbf{x}(t)]$.

Se si perturba un po' t_1 si ottiene l'equazione *un passo* corrispondente alla IV.5:

$$J[\mathbf{x}(t_1), t_1] = \min_{\substack{\mathbf{u}(t) \\ t_1 \leq t \leq t_1 + \delta t}} \left\{ \int_{t_1}^{t_1 + \delta t} g[\mathbf{x}(t), \mathbf{u}(t)] dt + J[\mathbf{x}(t_1 + \delta t), t_1 + \delta t] \right\} \quad (\text{IV.11})$$

Partendo dall'ultima equazione scritta ed effettuando una serie di manipolazioni e approssimazioni (per i dettagli si veda [15]) è possibile arrivare all'equazione di Hamilton-Jacobi-Bellman (*HJB equation*):

$$-\frac{\partial J}{\partial t}(\mathbf{x}, t) = \min_{\mathbf{u}} \left\{ g(\mathbf{x}, \mathbf{u}) + \frac{\partial J}{\partial \mathbf{x}}(\mathbf{x}, t) f(\mathbf{x}, \mathbf{u}, t) \right\} \quad (\text{IV.12})$$

con condizione iniziale nell'istante finale:

$$J(\mathbf{x}, T) = F(\mathbf{x}).$$

La IV.12 risolta fornisce la funzione $J(\mathbf{x}, t)$.

Sia l'equazione di Bellman, sia la versione un passo sono molto importanti poiché trasformano un problema di ottimizzazione esteso su un intervallo temporale in un problema di ottimizzazione in un singolo istante di tempo. L'equazione di Bellman è impossibile da risolvere in forma chiusa, tranne che per qualche eccezione.

Nel prossimo capitolo, i concetti introdotti in questa sezione saranno usati per risolvere numericamente il problema del controllo dello scheduling di un particolare sistema manifatturiero con tempi di setup non trascurabili. Le soluzioni numeriche trovate saranno poi usate come spunto e come verifica per la caratterizzazione analitica di alcuni aspetti delle soluzioni teoriche del problema.

IV.3 Un indice di costo per ottimizzare il comportamento del sistema durante il transitorio

La ricerca della traiettoria migliore è costituita da due fasi: l'impostazione di un problema di ottimizzazione e la sua soluzione attraverso la programmazione dinamica.

Il problema di ottimizzazione è composto da una funzione obiettivo o indice di costo o cifra di merito da minimizzare o massimizzare a secondo del problema, e da dei vincoli che definiscono la regione ammissibile, ovvero l'insieme di tutte le soluzioni tra le quali cercare la migliore, cioè quella che rende minimo o massimo l'indice scelto. Mentre l'individuazione dei vincoli il più delle volte non rappresenta un compito eccessivamente oneroso in quanto si tratta di escludere tutte le soluzioni che sicuramente non sono ammissibili, la caratterizzazione dell'indice di costo è un'operazione assai delicata perché da essa dipendono la forma e le proprietà della soluzione ottima. Infatti, se per esempio si considera un primo problema di ottimizzazione con un indice di costo A e un insieme di vincoli V , e poi un secondo problema con lo stesso insieme V di vincoli ma con un indice B differente da A , allora le soluzioni ottime del primo problema molto probabilmente non saranno più ottime per il secondo.

In questa sezione si cercherà di impostare analiticamente un problema di ottimizzazione il cui scopo è trovare una traiettoria con un buon comportamento durante il regime transitorio della macchina, ossia dal punto F fino

al ciclo limite. In particolare si tenterà di impostare la forma dell'indice di costo.

In letteratura esistono molti esempi di indici di costo. La maggior parte hanno una forma del tipo:

$$\lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T g[\mathbf{x}(t)] dt, \quad (\text{IV.13})$$

forma utilizzata anche nella sezione II.5, dove $g[\mathbf{x}(t)]$ è il costo puntuale che in genere viene scelto lineare. A titolo di esempio è possibile vedere l'articolo [1], oppure [22], o anche [9]. Questo tipo di costo però non fornisce alcuna informazione sul comportamento che il sistema deve seguire durante il regime transitorio.

Un lavoro in cui viene usato un indice di costo particolarmente adatto al problema del transitorio è l'articolo [10] di Hu e Caramanis del 1992. Al suo interno si trova il seguente problema di ottimizzazione:

$$\min_{\mu(t)} \lim_{T \rightarrow +\infty} \int_0^T \{g[\mathbf{x}(t)] - \bar{J}\} dt \quad (\text{IV.14})$$

soggetto ai vincoli

$$\begin{aligned} \mu(t) &= \mu[\mathbf{x}(t), \sigma(t)] \\ \dot{\mathbf{x}}(t) &= \mathbf{u}(\sigma) - \mathbf{d} \\ \sigma(t) &= [\xi(t), \tau(t)] \\ \xi(t) &= \mu(t^-) \end{aligned}$$

$$\dot{\tau}(t) = \begin{cases} \begin{cases} 0 & \text{se } \tau(t) = 0 \\ -1 & \text{se } \tau(t) > 0 \end{cases} & \text{se } \mu(t^-) = \mu(t) \\ \delta(t)\tau_{ji} & \text{se } \mu(t^-) = j \text{ e } \\ & \mu(t) = i \text{ con } i \neq j \end{cases}$$

dove:

$$\bar{J} = \min_{\mu(t)} \lim_{T \rightarrow +\infty} \frac{1}{T} \int_0^T g(\mathbf{x}) dt$$

è il costo medio per periodo, $\xi(t)$ è la più recente richiesta di setup e $\tau(t)$ è il tempo necessario affinché il setup richiesto diventi operativo sulla macchina.

Indipendentemente dalla forma del costo puntuale $g(\mathbf{x})$, \bar{J} rappresenta il costo medio minimo per periodo per T che tende a $+\infty$, ossia il costo ottimo del ciclo limite per tutti i problemi la cui soluzione ottima a regime coincide con un ciclo limite. Da questo punto di vista gli autori, utilizzando la struttura di costo IV.14, cercano di minimizzare l'integrale della differenza tra il costo puntuale $g[\mathbf{x}(t)]$ e il costo ottimo a regime. Se ad esempio la traiettoria parte da un punto F che presenta un costo puntuale maggiore del costo limite, la minimizzazione tenderà a portare la traiettoria sul ciclo limite ottimo facendole attraversare zone con un costo puntuale più basso, in modo da minimizzare l'integrale della differenza. Viceversa, se F presenta un costo puntuale minore di \bar{J} e se non esistono zone da attraversare con un costo puntuale ancora più piccolo, la minimizzazione tenderà a far rimanere la traiettoria nell'intorno del punto F di partenza, dove sarà locato il ciclo limite ottimo. Se invece esistono zone migliori dal punto di vista del costo, è ovvio che la traiettoria abbandonerà l'intorno di F per raggiungerle. Questa è la filosofia che sta dietro l'indice IV.14.

Esso però presenta un problema analitico di esistenza. Se si ammette che la soluzione ottima del problema dello scheduling dei setup converge a un

ciclo limite (come ad esempio accade nel caso di 2 soli tipi di parti scegliendo un costo del tipo IV.13), allora una volta che questo sarà raggiunto dalla traiettoria, il costo puntuale $g[\mathbf{x}(t)]$ assumerà un andamento periodico, con periodo esattamente pari alla durata del ciclo limite. Questo farà in modo che anche la differenza $g - \bar{J}$ sarà periodica per T che tende a $+\infty$, e quindi anche il suo integrale. Di conseguenza il limite dell'integrale per T che tende a $+\infty$ non sarà definito in quanto oscillerà permanentemente e in maniera continua tra due valori (si veda per esempio la figura V.3, a pagina 133).

Questo problema può essere superato sostituendo il limite con un limite inferiore:

$$\liminf_{T \rightarrow +\infty} \int_0^T \{g[\mathbf{x}(t)] - \bar{J}\} dt \quad (\text{IV.15})$$

in questo modo, anche se il comportamento dell'integrale rimane periodico, non sussistono più i problemi di esistenza del limite poiché il nuovo indice considera il limite inferiore di tale integrale.

Capitolo V

Il controllo durante il transitorio di un sistema “*make to order*”

V.1 Formulazione del problema

In questo capitolo verrà affrontato un problema di controllo specifico. Esso è totalmente ripreso dall'articolo [23], sottomesso all'attenzione della 40st IEEE Conference on Decision and Control. Si consideri un sistema manifatturiero costituito da una sola macchina soggetta a guasti, capace di produrre due tipi differenti di parti con tempi di cambio del setup non trascurabili. La struttura del sistema è di tipo *make to order*, quindi l'evoluzione della traiettoria del surplus sarà vincolata all'interno del III quadrante (vedi la sezione II.5. Lo schema del sistema è identico a quello presente a pagina 49, che per comodità è riportato a pagina 114. Sia:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

il vettore dello stato del sistema, dove la generica componente $x_i(t)$ rappresenta il contenuto del buffer i all'istante t . In accordo con la struttura di

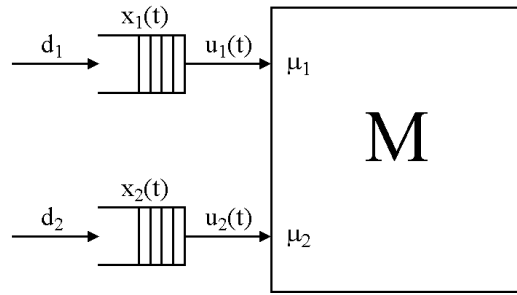


Figura V.1: Il sistema manifatturiero

controllo gerarchica, sia μ_i il tasso di produzione massimo per la parte di tipo i , e d_i il tasso di arrivo delle domande, supposto costante.

Si focalizzi l'attenzione sul livello dei setup, assumendo che il livello superiore della gerarchia si occupi della gestione dei guasti e delle riparazioni della macchina. Perciò, al livello dei setup, la macchina può essere considerata completamente affidabile rispetto ai guasti. La dinamica della parte i -esima è data da:

$$\begin{aligned} \dot{x}_i(t) &= d_i - s_i(t)u(t) \\ x_i(0) &= x_{0i} \end{aligned} \tag{V.1}$$

dove $u(t)$ è il tasso di produzione della macchina all'istante t mentre si sta lavorando la parte i , con $i = 1, 2$, e:

$$s(t) = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

se la macchina è nel setup 1,

$$s(t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

se la macchina è nel setup 2,

$$s(t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

durante un cambiamento del setup. Inoltre, sia τ_{ij} il tempo di setup richiesto dalla macchina per passare dalla configurazione i a quella j , con $i \neq j$. Si assume di avere a disposizione dei buffer limitati dal punto di vista della capacità.

Si consideri il seguente indice di costo:

$$J_a = \lim_{\tau \rightarrow +\infty} \frac{1}{\tau} \int_0^\tau g[\mathbf{x}(s)] ds, \quad (\text{V.2})$$

dove $g(\mathbf{x})$ è il costo puntuale, assunto positivo, associato al contenuto del buffer \mathbf{x} . Se la macchina fosse stata completamente affidabile, la traiettoria ottima rispetto all'indice V.2, a regime, sarebbe andata sul ciclo limite posizionato nello spazio di stato il più vicino possibile all'origine degli assi, come mostrato nella sezione II.5. In questo caso invece, la macchina è soggetta a guasti, e si assume che l'intervallo temporale tra due rotture consecutive sia sufficientemente grande da poter considerare il sistema in regime stazionario quando si verificherà il successivo guasto.

L'obiettivo è trovare un controllo ottimo per il regime transitorio. Si assume che nella riparazione sia compresa la scelta del setup iniziale della macchina. Come detto in conclusione della sezione IV.3, si consideri l'indice di costo:

$$J_t = \liminf_{\tau \rightarrow +\infty} \int_0^\tau \{g[\mathbf{x}(s)] - \bar{J}\} ds \quad (\text{V.3})$$

dove \bar{J} è il costo medio J_a del ciclo limite. In questo modo, per ottimizzare il transitorio, si suppone che il prossimo guasto accada all'infinito.

Per risolvere il problema appena formulato è necessario utilizzare dei risultati intermedi validi per lo stesso tipo di problema ma con buffer infiniti.

Si osservi che il punto di partenza della traiettoria, cioè il punto F, è assunto essere sempre a destra del ciclo limite ottimo, in quanto si suppone sempre che il sistema stia ripartendo dopo aver subito un guasto e la successiva riparazione. Ciò può essere espresso in modo formale attraverso la seguente assunzione:

Assunzione 1 Sia $\mathbf{x}_0 = (x_{01}, x_{02})$ lo stato iniziale per il problema del transitorio. Allora esiste un punto $P = (x_1, x_2)$ alla destra del ciclo limite ottimo, e un intervallo finito di tempo Δt tale che

$$x_{0i} = x_i + d_i \Delta t \quad , \quad i = 1, 2.$$

Il problema sarà studiato nella sua forma simmetrica, ossia:

Assunzione 2 Il sistema è simmetrico, cioè $\mu_1 = \mu_2 = \mu$, $d_1 = d_2 = d$, $c_1 = c_2 = c$, e $\tau_{12} = \tau_{21} = \tau$.

Concludendo sarà affrontato il seguente problema:

Problema 1 *Dato il sistema in considerazione, sotto le Assunzioni 1 e 2, assumendo che la prossima rottura del sistema accada all'infinito, trovare il setup iniziale, le curve dei setup e il tasso di produzione $u(\cdot)$ al fine di minimizzare l'indice J_t .*

V.2 Soluzione teorica: il caso con buffer infiniti

La soluzione del Problema 1, considerando un sistema manifatturiero fornito di buffer infinito, rappresenta un punto di partenza per la soluzione del problema più realistico con i buffer finiti.

Nel caso dei buffer infiniti il costo puntuale analizzato è del tipo:

$$g[\mathbf{x}(t)] = c_1x_1(t) + c_2x_2(t) = c[x_1(t) + x_2(t)], \quad (\text{V.4})$$

con c_1 e c_2 costanti positive. Il seguente teorema afferma che la soluzione ottima del problema sarà costituita unicamente da lavorazioni eseguite dalla macchina al massimo tasso di produzione possibile.

Teorema 3 *Per un dato valore dello stato iniziale \mathbf{x}_0 , sotto le Assunzioni 1 e 2, sia $u(t)$, con $t \in (0, +\infty)$, il controllo che minimizza l'indice J_t . Allora $u(t) = \mu$ per ogni $s(t) \neq [0, 0]$, e per tutti i $t \geq 0$.*

Prova Senza perdere di generalità, si assuma che esista un intervallo temporale (t_1, t_2) , $t_2 > t_1$, dove $u(t) = u$ con $u < \mu$. È facile costruire un'altra funzione di controllo $u(t) = \mu$ per $t \in (t_1, t_1 + \frac{\mu-d}{\mu-d}(t_2 - t_1))$ e $u(t) = d$ per $t \in (t_1 + \frac{\mu-d}{\mu-d}(t_2 - t_1), t_2)$. Essa fornisce un valore di J_t più piccolo. \square

Un'altra proprietà fondamentale della soluzione ottima del problema è quella di cambiare il setup corrente della macchina esclusivamente sugli assi coordinati dello spazio di stato. Da ciò deriva la sicura non ottimalità di

tutte le soluzioni che presentano uno o più cambi del setup all'interno dello spazio di stato.

Teorema 4 *Si assuma che $\mathbf{x}(t)$ sia una traiettoria ottima. Sia $\mathbf{x}(\bar{t}) = (\bar{x}_1, \bar{x}_2)$ un punto della traiettoria $\mathbf{x}(t)$ dove viene eseguito un cambio del setup. Allora, o $\bar{x}_1 = 0$ oppure $\bar{x}_2 = 0$, cioè, il setup viene cambiato rispettivamente o sull'asse $x_2 = 0$ oppure sull'asse $x_1 = 0$.*

Prova Senza perdere di generalità, si supponga che la traiettoria ottima $\mathbf{x}(t)$ cambi la lavorazione, dalla parte di tipo 1 a quella di tipo 2, nell'istante \bar{t} e nel punto $B_0 = (a, b)$, con $a \neq 0$ e $b \neq 0$. Si consideri una seconda traiettoria $\tilde{\mathbf{x}}(t)$, coincidente con $\mathbf{x}(t)$ fino all'istante \bar{t} , dopodiché cambi il setup dalla parte di tipo 1 a quella di tipo 2 nell'istante $\bar{t} + \Delta t$ e nel punto $R_0 = (a - (\mu - d)\Delta t, b + d\Delta t)$, con $a \geq (\mu - d)\Delta t$. La rimanente parte della traiettoria $\tilde{\mathbf{x}}(t)$ è realizzata eseguendo l' i -esimo cambio del setup dalla parte l alla parte m all'istante $t_{lm}^{(i)}$, dove $\tilde{x}_l(\tilde{t}_{lm}^{(i)}) = x_l(t_{lm}^{(i)})$ e $t_{lm}^{(i)}$ è l'istante dove la traiettoria ottima esegue lo stesso cambio del setup. Come è possibile verificare, $\tilde{t}_{lm}^{(i)} > t_{lm}^{(i)}$ per tutti i valori di i, l, m , e ciò implica per $\tilde{\mathbf{x}}(t)$ un valore più basso di J_t . □

Infine, l'ultimo teorema fornisce un'ulteriore caratterizzazione della soluzione ottima, cioè il setup iniziale sarà scelto massimizzando l'istante del primo cambio del setup, ovvero seguendo una politica di tipo LQF (*Longest Queue First*).

Teorema 5 Sia $\mathbf{x}_0 = (a, b)$ lo stato iniziale e si assuma $a > b$ ($a < b$). Allora il setup iniziale ottimo è $\mathbf{s}(0) = (1, 0)$ ($\mathbf{s}(0) = (0, 1)$).

Prova Si consideri $\mathbf{x}_0 = (a, b)$. La prova di questo teorema è basata sul confronto tra il costo della traiettoria T_1 che inizia da \mathbf{x}_0 con il setup 1 e la traiettoria T_2 che inizia sempre da \mathbf{x}_0 ma con il setup 2. È noto dal Teorema 4 che entrambi le traiettorie per essere ottime devono cambiare il setup sugli assi coordinati. Sia α_k la coordinata differente da 0 del k -esimo cambio del setup della traiettoria T_2 , e β_k la corrispondente di T_1 . Sia $\gamma_k = \beta_k - \alpha_k$ e $\Delta t_k = t_k^{(1)} - t_k^{(2)}$, dove $t_k^{(1)}$ è l'istante in cui la traiettoria T_1 effettua il k -esimo cambio del setup e $t_k^{(2)}$ è l'istante dove T_2 esegue il k -esimo cambio del setup. È possibile ricavare le seguenti dinamiche:

$$\begin{aligned}\Delta t_{k+1} &= \Delta t_k + \frac{\gamma_k}{\mu - d}, \\ \gamma_{k+1} &= \frac{d}{\mu - d} \gamma_k,\end{aligned}$$

con condizioni iniziali $\Delta t_0 = \frac{a-b}{\mu-d}$, $\gamma_0 = -(\mu - 2d)\Delta t_0$. Poiché $\frac{d}{\mu-d} < 1$, $\gamma_k \rightarrow 0$ per $k \rightarrow +\infty$, e risulta anche che $\Delta t_k \rightarrow 0$ per $k \rightarrow +\infty$. Inoltre, Δt_k raggiunge lo 0 in maniera monotona. Tutto questo implica che il costo di J_t relativo alla traiettoria T_2 è maggiore del costo della traiettoria T_1 . \square

Riassumendo, in questa sezione si sono presentati tre teoremi che forniscono tre proprietà fondamentali che caratterizzano completamente la soluzione del problema 1 nel caso di buffer infiniti:

- quando il sistema deve produrre lo deve fare sempre con il massimo tasso di produzione possibile;
- tutti i cambi dei setup devono avvenire esclusivamente lungo gli assi coordinati;
- il setup iniziale sarà scelto seguendo una politica di tipo LQF.

V.3 Soluzione teorica: il caso con buffer finiti

Questa sezione è basata sulla rimozione dell'ipotesi di buffer infiniti. Si supponga che ogni buffer i abbia una capacità massima L_i . L'introduzione di questa modifica obbliga a considerare un costo aggiuntivo all'interno dell'espressione analitica V.4 del costo puntuale per le domande perse. Assumendo che $R_1 = R_2 = R$ sia il costo per unità di domanda persa dei due tipi di parti, si ponga:

$$g[\mathbf{x}(t)] = c(x_1 + x_2) + Rd[I(x_1 = L) + I(x_2 = L)], \quad (\text{V.5})$$

dove $I(\cdot) = 1$ se il suo argomento è vero, $I(\cdot) = 0$ altrimenti.

In questo caso, è necessario aggiungere due assunzioni alle due già fatte in precedenza. Si assume che il ciclo limite corrispondente al problema affrontato sia completamente contenuto nello spazio di stato, descritto analiticamente da:

$$\mathcal{X} = \{x \in \mathbb{R}^2 : 0 \leq x_i \leq L, i = 1, 2\}.$$

Ciò è abbastanza ragionevole, in quanto una scelta oculata della capacità del buffer dovrebbe tener conto anche delle caratteristiche stazionarie del sistema. In aggiunta, si assumerà che il costo istantaneo del punto (L, L) sarà maggiore del costo medio \bar{J} sul ciclo limite, altrimenti il problema potrebbe divenire inconsistente.

Assunzione 3 *Il ciclo limite non raggiunge mai le frontiere dello spazio di stato \mathcal{X} , cioè $\frac{2(\mu-d)}{\mu-2d}d\tau \leq L$.*

Assunzione 4 Il costo del punto (L, L) è maggiore del costo medio del ciclo limite, cioè $2(cL + dR) > \bar{J}$.

La soluzione ottima, come nel caso a buffer finiti, è caratterizzata da delle lavorazioni eseguite al massimo tasso di produzione possibile.

Teorema 6 Con le capacità dei buffer finite, per un dato stato iniziale \mathbf{x}_0 , sotto le assunzioni 1, 2, 3, e 4, sia $u(t)$, con $t \in (0, +\infty)$, il controllo che minimizza l'indice J_t . Allora, per tutti i valori di t , $u(t) = \mu$ se $\mathbf{s}(t) \neq [0, 0]$.

La prova è omessa, ma è molto simile a quella del teorema 3.

Nel seguito si riporteranno alcune proprietà analitiche delle soluzioni ottime del problema a buffer finiti, che sono state determinate per spiegare alcune caratteristiche della soluzione ottima determinata mediante un approccio numerico (si vedano i paragrafi V.4, e V.5)

Teorema 7 Si assuma che R sia sufficientemente piccolo, che lo stato iniziale sia $\mathbf{x}_0 = (a, b)$, e che la traiettoria ottima viaggi lungo il bordo fino agli assi coordinati. Allora sotto le Assunzioni 1, 2, 3 e 4, esiste una soglia:

$$R_t(\mathbf{x}_0) = \frac{c\mu(a+b)}{2d(\mu-2d)} - \frac{2cd\tau(\mu-d)}{(\mu-2d)^2} - \frac{Lc}{d}, \quad (\text{V.6})$$

tale che, se $R < R_t(\mathbf{x}_0)$ risulta conveniente partire da \mathbf{x}_0 con il setup 1 se $a < b$, altrimenti se $R > R_t(\mathbf{x}_0)$ con il setup 2.

Prova L'Assunzione 3 implica che una volta che la traiettoria lascia il bordo dello spazio di stato, essa rimane sempre all'interno della regione \mathcal{X} e non

raggiunge più nessun bordo. L'espressione della soglia è ottenuta attraverso un confronto diretto tra il costo J_t della traiettoria che inizia da \mathbf{x}_0 con il setup 1 e la traiettoria che comincia da \mathbf{x}_0 con il setup 2. Il costo puntuale delle due traiettorie in funzione del tempo è riportato in figura V.2. Questo confronto fornisce:

$$R_t(\mathbf{x}_0) = \frac{\Delta A - \bar{J} \cdot (t_{01} + t_{1s} - t_{02} - t_{2s})}{d(t_{2s} - t_{1s})},$$

dove, in accordo con la figura, t_{01} è l'istante in cui la traiettoria LQF raggiunge il bordo, t_{02} l'istante in cui la traiettoria SQF raggiunge il bordo, $t_{1s} = t_{01} + \tau$, $t_{2s} = t_{02} + \tau$, \bar{J} il costo medio sul ciclo limite. Il termine ΔA è l'area ombreggiata rappresentata nella figura V.2. Manipolando il risultato del confronto si arriva all'espressione V.6 riportata nell'enunciato. \square

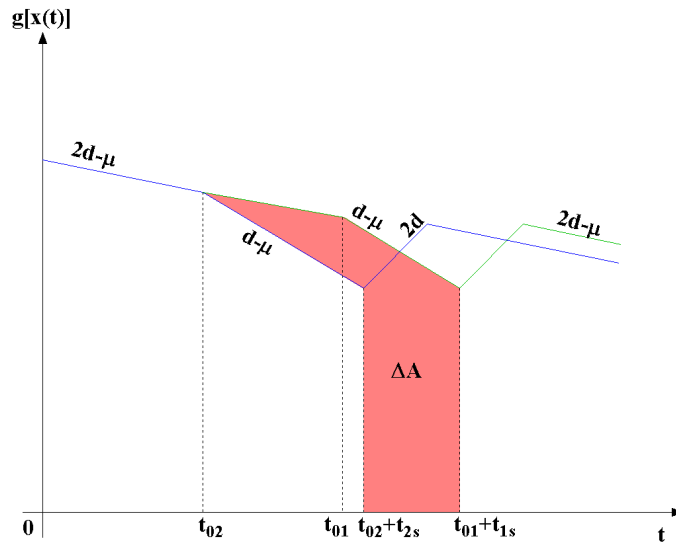


Figura V.2: Confronto tra le due traiettorie nella prova del Teorema 7

Dall'espressione di $R_t(x_0)$ emerge che i punti iniziali che hanno lo stesso valore di $a + b$ hanno anche la stessa soglia $R_t(x_0)$. Come $a + b$ aumenta, anche $R_t(x_0)$ aumenta. L'inversione comincia dall'angolo (L, L) di \mathcal{X} e si propaga verso l'interno man mano che R diminuisce.

V.4 Soluzione numerica del problema

In questa tesi si considera la soluzione al problema 1 nel caso di buffer finiti basata sulla formulazione discreta dell'equazione “un passo” che affronta il problema attraverso la tecnica della programmazione dinamica. Con ciò, il primo problema da risolvere è l'impostazione attraverso la programmazione dinamica di una equazione simile alla IV.5, presente a pagina 105.

Sia $J_i^{(k)}(\mathbf{x})$ la funzione *cost-to-go* partendo dal punto \mathbf{x} nello spazio di stato con il setup i ($i = 1, 2$), e muovendosi per un intervallo pari a $k \delta t$. Tali funzioni rappresentano il valore ottimo dell'integrale V.3 prima dell'operazione \liminf . Le condizioni iniziali sono $J_i^{(0)}(\mathbf{x}) = 0$, per ogni i e per ogni $\mathbf{x} \in \mathcal{X}$. Il valore δt rappresenta il più piccolo intervallo di tempo che l'algoritmo risolutivo è in grado di apprezzare, esso è quindi il quanto di tempo della discretizzazione temporale, ed è scelto tale che:

$$\tau = N \delta t.$$

Lo scopo è quello di trovare le due funzioni $J_1^{(M)}(\mathbf{x})$ e $J_2^{(M)}(\mathbf{x})$ con M molto grande in modo tale da essere approssimato con l'infinito e quindi associato a dei costi di regime.

L'idea da cui nasce la formulazione un passo è assai semplice. Se la macchina è in un certo punto \mathbf{x} dello spazio di stato in un generico passo k e nel setup i , le decisioni possibili sono due, e alternative tra loro:

1. si fa proseguire la produzione della parte di tipo i per un altro passo di durata δt ;

2. si interrompe la produzione della parte di tipo i e si dà inizio alla procedura del cambio del setup, interrompibile, e di durata pari a τ .

Questa decisione tradotta in formule diventa:

$$\begin{aligned}
 J_i^{(k)}(\mathbf{x}) &= \\
 &= \min \left\{ \left[\int_0^{\delta t} g(\mathbf{x} + \mathbf{v}_i s) ds + J_i^{(k-1)}(\mathbf{x} + \mathbf{v}_i \delta t) + (k-1)\bar{J}\delta t \right] - k\bar{J}\delta t, \right. \\
 &\quad \left. \left[\int_0^\tau g(\mathbf{x} + \mathbf{v}_0 s) ds + J_j^{(k-N)}(\mathbf{x} + \mathbf{v}_0 \tau) + (k-N)\bar{J}\delta t \right] - k\bar{J}\delta t \right\}.
 \end{aligned}
 \tag{V.7}$$

L'equazione V.7 a tempo discreto, ancora non è implementabile su di un calcolatore, è necessario discretizzare anche lo spazio di stato. Infatti dalla V.7 emerge che per calcolare il valore $J_i^{(k)}(\mathbf{x})$ è necessario conoscere il valore del costo al passo precedente e in un altro punto differente da \mathbf{x} . A causa della continuità dello spazio \mathcal{X} , non è possibile memorizzare al passo $k-1$ un numero di valori praticamente infinito per poi calcolare i costi del passo k . Per questo motivo nasce la necessità della discretizzazione dello spazio di stato, che però non risulta indipendente da quella fatta per l'asse temporale. Se infatti δt rappresenta il più piccolo intervallo temporale trattato dall'algoritmo risolutivo, la discretizzazione dello spazio di stato deve essere tale che se si prende un qualsiasi punto \bar{x} discretizzato, qualunque sia l'azione che la

macchina compie, lo stato alla fine dell'azione deve coincidere con un altro punto discretizzato dello spazio. In altre parole, qualunque azione compia la macchina, il punto finale non deve mai cadere in un punto che non appartiene alla discretizzazione, altrimenti non sarà possibile memorizzarne il costo. Per fare ciò è necessario individuare le singole azioni che la macchina può fare. Esse sono in questo caso quattro:

1. la macchina è nel setup 1 e si decide di continuare a produrre 1 per un altro δt ;
2. la macchina sta lavorando la parte di tipo 1 e si decide di cambiare il setup;
3. la macchina è nel setup 2 e si decide di continuare a produrre 2 per un altro δt ;
4. la macchina sta lavorando la parte di tipo 2 e si decide di cambiare il setup.

Alla luce dell'Assunzione 1, le azioni 2 e 4 coincidono. Per ogni possibile azione si impone un sistema del tipo:

$$\begin{cases} \mathbf{x}_F = k\Delta\mathbf{x} + \mathbf{x}_I \\ \mathbf{x}_F = v\Delta t + \mathbf{x}_I \end{cases}$$

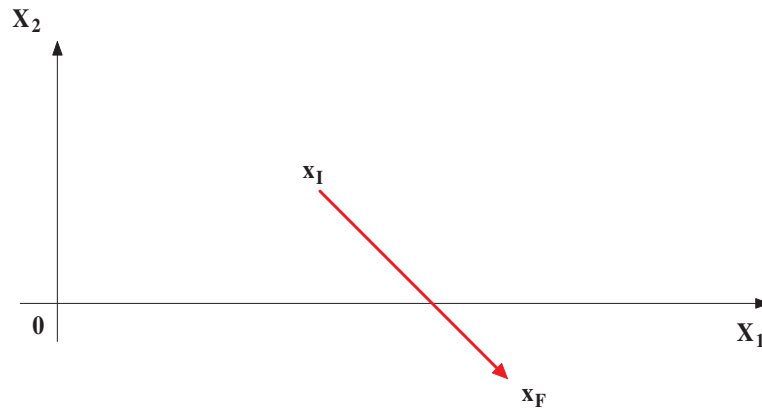
dove \mathbf{x}_I è il generico punto iniziale, \mathbf{x}_F quello finale, e Δt è pari a δt nel caso delle azioni 1 e 3, e pari a τ nel caso di 2 e 4. Risolvendo i 3 sistemi collegati alle tre azioni possibili che la macchina può compiere, si ottengono i

valori delle varie costanti k , le quali forniscono il quanto $\Delta \mathbf{x}$ per lo spazio di stato. In questo modo si è sicuri che indipendentemente dal comportamento della macchina, il punto finale disti orizzontalmente e verticalmente uno o più multipli (il numero preciso è fornito dai valori di k) di $\Delta \mathbf{x}$. Quindi $\Delta \mathbf{x}$ può essere visto come una misura di quanto fitta sia la discretizzazione dello spazio. Più è piccolo $\Delta \mathbf{x}$, più punti bisogna considerare nelle elaborazioni, più attendibile sarà la soluzione numerica.

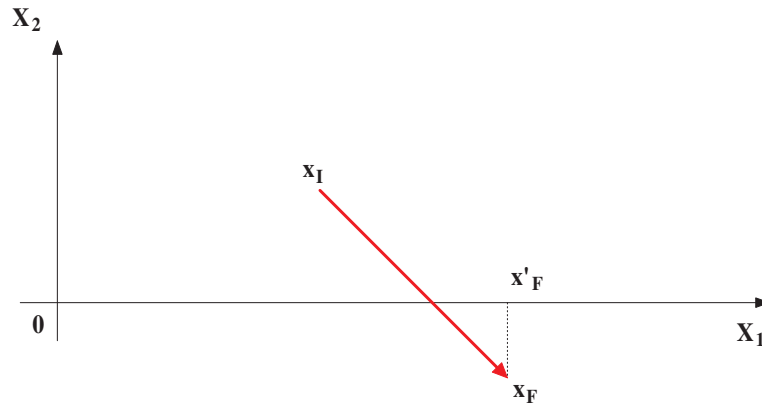
Si noti che i primi N passi sono leggermente differenti, in quanto considerano degli intervalli temporali più piccoli del tempo τ del cambio del setup.

Gestione delle traiettorie sul bordo

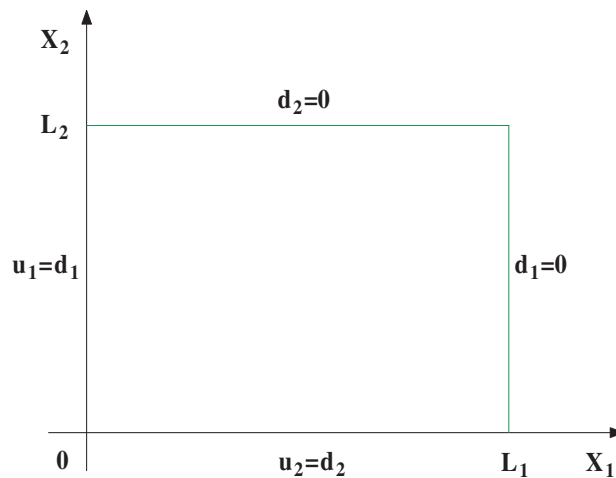
Alcuni problemi nascono quando il punto finale \mathbf{x}_F della traiettoria è sito al di fuori dello spazio di stato:



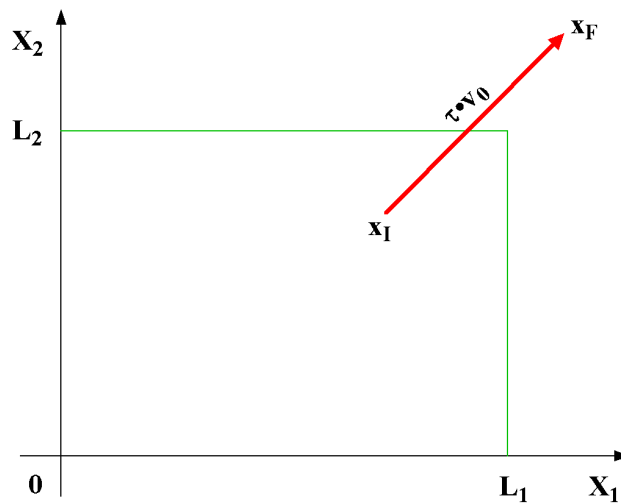
In questi casi si considera come nuovo punto finale la proiezione ortogonale del punto finale effettivo sul bordo oltrepassato dalla traiettoria:



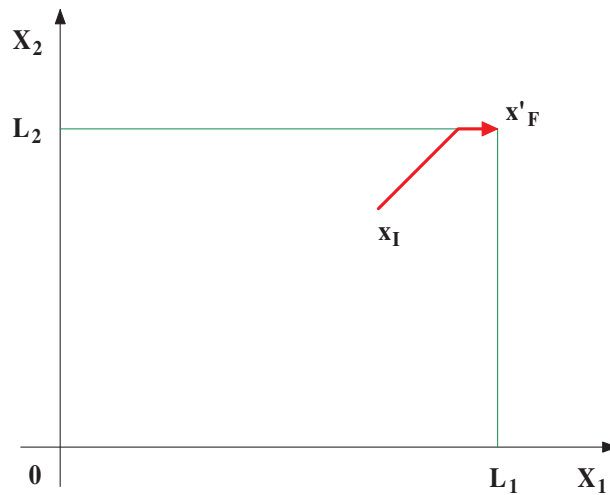
Ciò trova giustificazione nel fatto che quando la traiettoria raggiunge una qualsiasi delle quattro frontiere dello spazio di stato, inevitabilmente a causa dei vincoli strutturali del sistema, o si annullano i tassi di arrivo delle domande, o i tassi di produzione u_i uguagliano i corrispondenti d_i :



In alcune situazioni può non essere possibile considerare la proiezione ortogonale. Ad esempio durante un cambio del setup, il sistema è obbligato a non produrre per un intervallo di tempo pari a $\tau = N\delta t$.



In questo caso, la traiettoria è obbligata a rimanere ferma nel punto (L, L) fino alla fine della riconfigurazione:



Anche se le discretizzazioni del tempo e dello spazio sono state necessarie per il calcolo delle soluzioni numeriche, inevitabilmente hanno introdotto delle approssimazioni, che in alcuni casi sono state la causa di anomalie nel

comportamento delle traiettorie-soluzioni. Tali anomalie sono state di volta in volta corrette durante la fase di stesura del codice in C^{++} dell'algoritmo risolutivo. Il codice finale e commentato è presente nell'appendice A.

V.5 Proprietà della soluzione ottima ricavata con l'approccio numerico

Una caratteristica della soluzione numerica è quella di non convergere ad un valore limite. Quando il numero delle iterazioni è sufficientemente elevato, le funzioni *cost-to-go* oscillano tra due valori molto vicini tra loro. Ad esempio partendo dal punto $(50, 50)$, cioè con 50 domande in attesa per le parti di tipo 1 e 50 per quelle di tipo 2, con $\mu = 3$, $d = 1$, $L = 62$, $R = 150$ e $\tau = 5$ si ottiene:

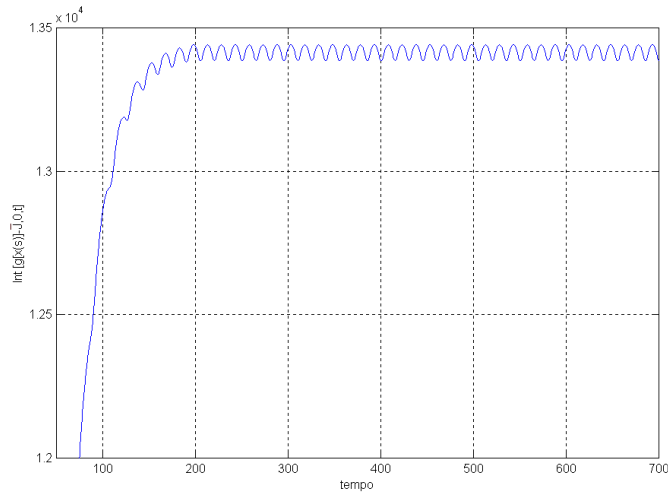


Figura V.3: Il costo integrale lungo una traiettoria che raggiunge il ciclo limite a regime

Dal grafico si può osservare che l'ampiezza dell'oscillazione, cioè la differenza tra un picco massimo e uno minimo, è molto più piccola dell'ordine di grandezza dei valori stessi, e che il periodo è esattamente pari alla metà della durata del ciclo limite, in questo caso: $T = 30$. Questo perché si è considera-

to un problema completamente simmetrico, per il quale la traiettoria, dopo aver percorso metà del ciclo limite, si trova in dei punti dello spazio di stato che presentano lo stesso costo di quelli iniziali.

La soluzione numerica della V.3 risulta possedere tutte le caratteristiche della soluzione teorica del caso con buffer infiniti ($R = 0$). Ovvero che i cambiamenti del setup vengono effettuati sugli assi, il setup iniziale è scelto in modo da massimizzare l'istante del prossimo cambio di setup, e che la traiettoria converge al ciclo limite.

Calcolando la soluzione numerica della V.3 per vari valori dei parametri si sono riscontrate delle caratteristiche interessanti, in particolare al variare del parametro R . Tali caratteristiche sono qui di seguito raccolte in forma di risultati.

Risultato 1 *In funzione del costo R delle domande perse, il controllo ottimo può evitare di raggiungere i bordi, caratterizzati da $x_1 = L$ o $x_2 = L$, se R è sufficientemente grande, eseguendo il cambio del setup prima di toccare il bordo. Se il valore di R viene ridotto, la traiettoria ottima si muoverà lungo il bordo fino ad un punto che dipende dal valore di R . Se R è sufficientemente piccolo, la traiettoria percorrerà l'intero bordo cambiando il setup sugli assi, cioè nel punto $(0, L)$ o nel punto $(L, 0)$.*

I comportamenti descritti nel Risultato 1 sono illustrati nelle figure V.4, V.5, V.6, e V.7.

Le traiettorie sono state ricavate utilizzando i parametri $\mu = 3$, $d = 1$, $L = 62$, $\tau = 5$, e a partire dal punto iniziale $(50, 50)$. Per la figura V.4 si è

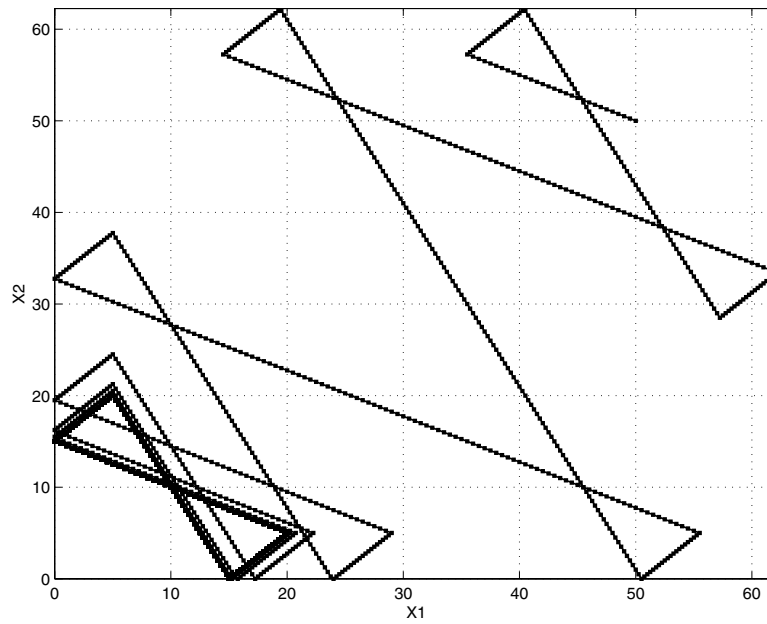


Figura V.4: Il comportamento sul bordo $x_2 = L$, con R molto grande.

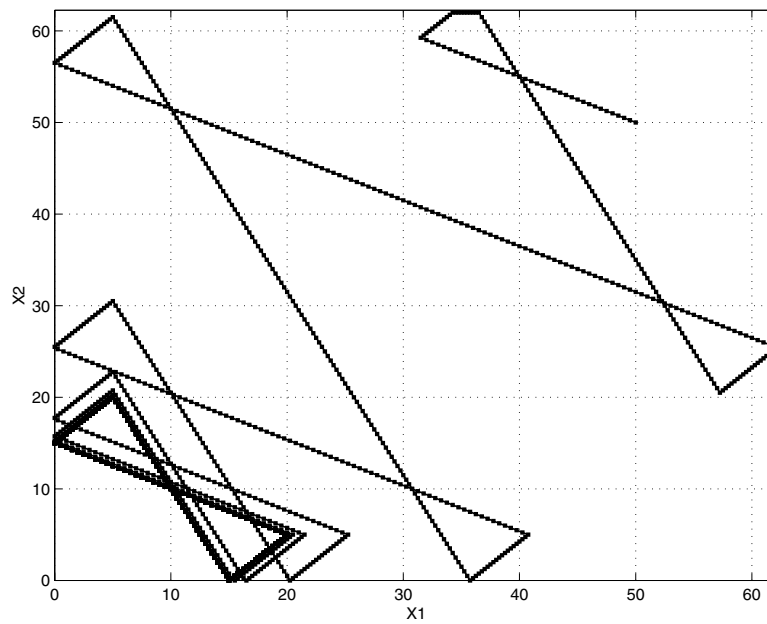


Figura V.5: Il comportamento sul bordo $x_2 = L$, con $R = 1000$.

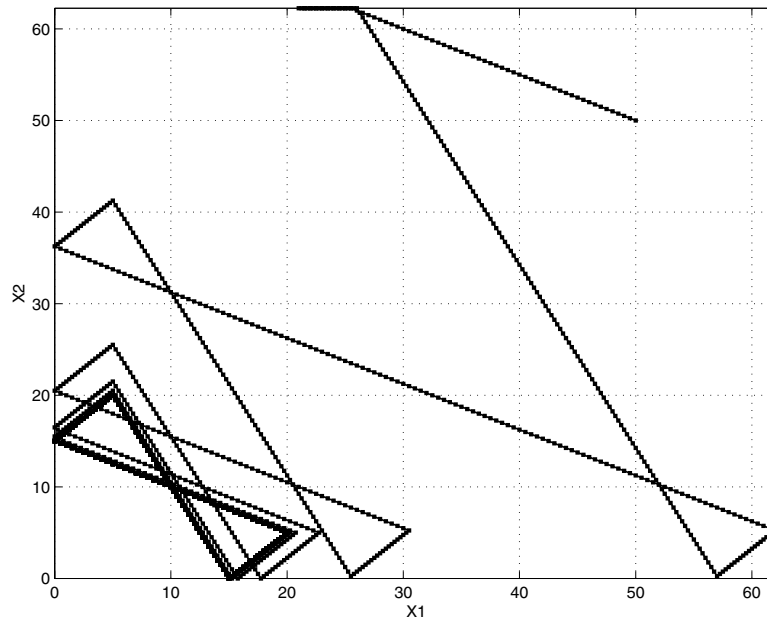


Figura V.6: Il comportamento sul bordo $x_2 = L$, con $R = 600$.

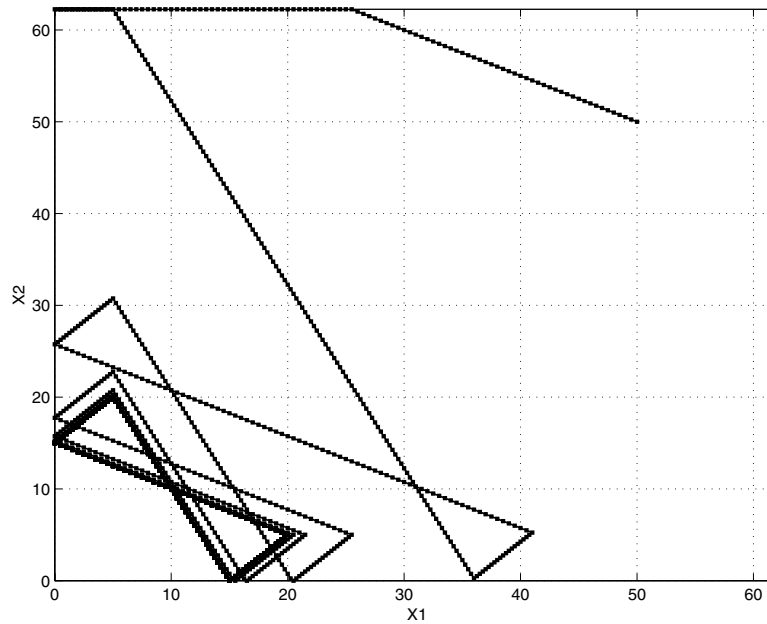


Figura V.7: Il comportamento sul bordo $x_2 = L$, con $R = 150$.

usato un grande valore di R , $R = 10^6$, ed è chiaro che la traiettoria ottima non raggiunge il bordo al fine di evitare di pagare un grande costo per le domande perse. Man mano che R diminuisce la traiettoria comincia a muoversi lungo il bordo: la figura V.5 è caratterizzata da un valore $R = 1000$, la figura V.6 $R = 600$, e la figura V.7 $R = 150$. Il motivo per cui la traiettoria ottima percorre tratti di bordo sempre più lunghi man mano che R diminuisce è che andare lungo il bordo permette alla traiettoria di muoversi più velocemente poiché un tasso di arrivo delle domande è nullo. Ciò non è possibile se la traiettoria si muove all'interno dello spazio di stato, dove sono presenti entrambi i tassi di arrivo delle domande, e la velocità della traiettoria durante le lavorazioni è globalmente pari a $\mu - 2d$. Questa maggiore velocità permette alla traiettoria di raggiungere più velocemente le zone dello spazio di stato favorevoli dal punto di vista del costo, ottimizzando quindi il comportamento transitorio del sistema. Questo contributo positivo, quando diventa maggiore di quello corrispondente al costo da pagare per le domande perse, permette alla traiettoria ottima di scivolare lungo il bordo.

Risultato 2 *Esiste un punto P sul bordo $x_2 = L$, con $P = (x_1, L)$, dove x_1 è funzione dei parametri, tale che per tutti gli $x > x_1$ la traiettoria ottima si muove lungo il bordo nell'intervallo $[x_1, L]$, e cambia il setup se $x < x_1$. Una proprietà simile vale anche per il bordo $x_1 = L$.*

Questo risultato è illustrato nelle figure V.8 e V.9 per differenti condizioni iniziali. Il punto P risulta essere sempre nella stessa posizione, cioè esso non dipende dalle condizioni iniziali come ci si attendeva.

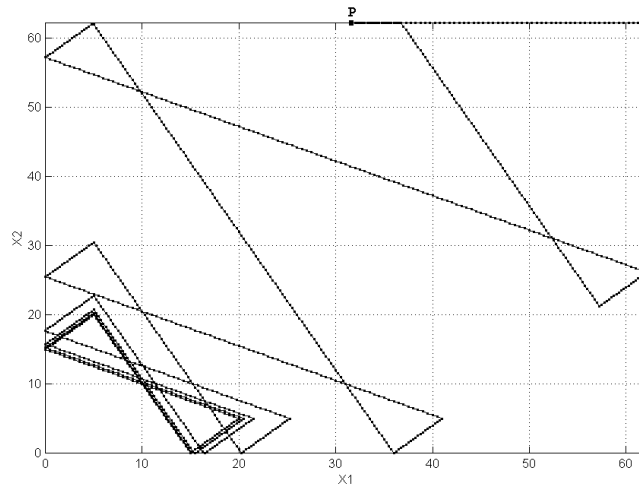


Figura V.8: Il punto P con condizioni iniziali $(62, 62)$.

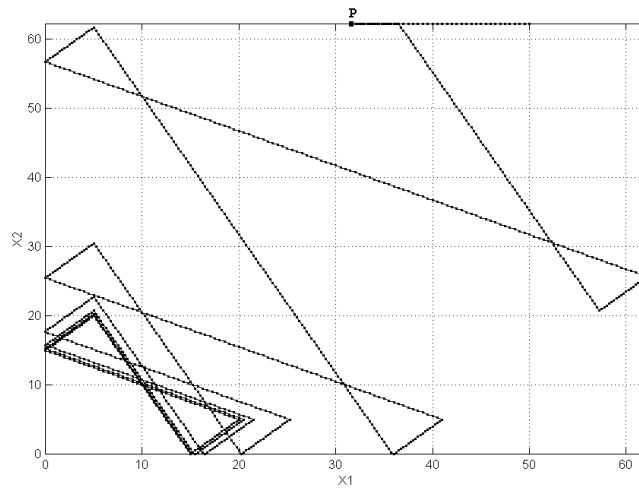


Figura V.9: Il punto P con condizioni iniziali $(50, 62)$.

Infine l'ultimo risultato, il quale ha ispirato il teorema 7, a pagina 123.

Risultato 3 *Si assuma che la traiettoria ottima si muova lungo il bordo fino a toccare gli assi (cioè, richiamando il risultato 1 se R è sufficientemente piccolo). Allora esiste una soglia R_t per il valore di R e per ogni punto iniziale $\mathbf{x}_0 = (a, b)$, cioè $R_t = R_t(\mathbf{x}_0)$, tale che se $R > R_t(\mathbf{x}_0)$ il setup ottimo iniziale da \mathbf{x}_0 è 1(2) se $a > b$ ($a < b$), come nel caso a buffer infiniti. Se $R < R_t(\mathbf{x}_0)$ la soluzione ottima percorre il bordo il più a lungo possibile, quindi inizia con l'altro setup, cioè 2(1) se $a > b$ ($a < b$), come una politica SQF (Shortest Queue First). Poiché $R_t(\mathbf{x}_0)$ è costante sulla retta $a + b = \text{costante}$, è possibile osservare una inversione della scelta del setup iniziale.*

Le caratteristiche enunciate nel risultato 3 sono illustrate nelle figure V.10 e V.11 a pagina 140, ricavate con i valori dei parametri pari a quelli delle figure precedenti, ma con $R = 50$ nella figura V.10, e $R = 150$ nella figura V.11.

Nelle figure, la zona colorata in nero corrisponde a scegliere il setup 1 come setup iniziale, quella in bianco a scegliere il setup 2. L'insieme dei punti colorati di rosso invece, rappresenta il luogo all'interno dello spazio di stato dove i due setup si equivalgono. Tale luogo coincide con la bisettrice. Si osservi come l'estensione della regione dell'inversione sia funzione del valore di R .

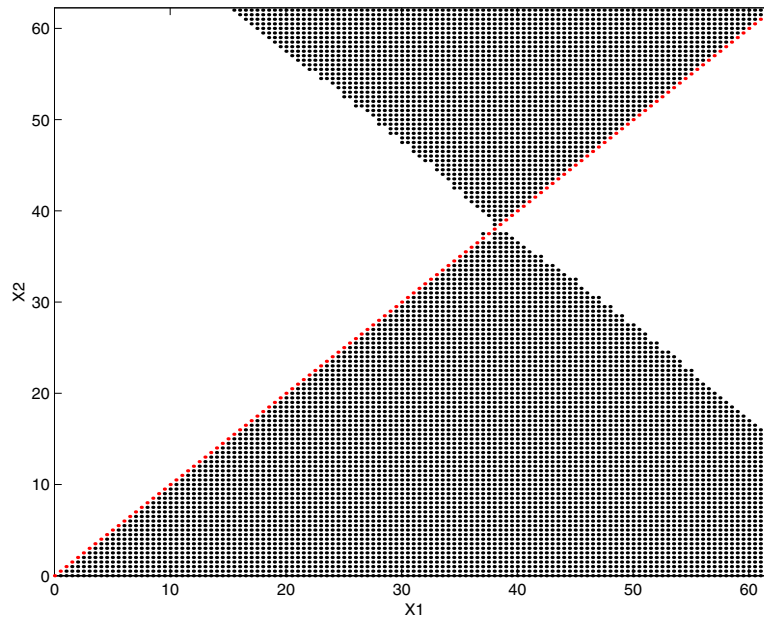


Figura V.10: L'inversione illustrata nel Risultato 3 con $R = 50$.

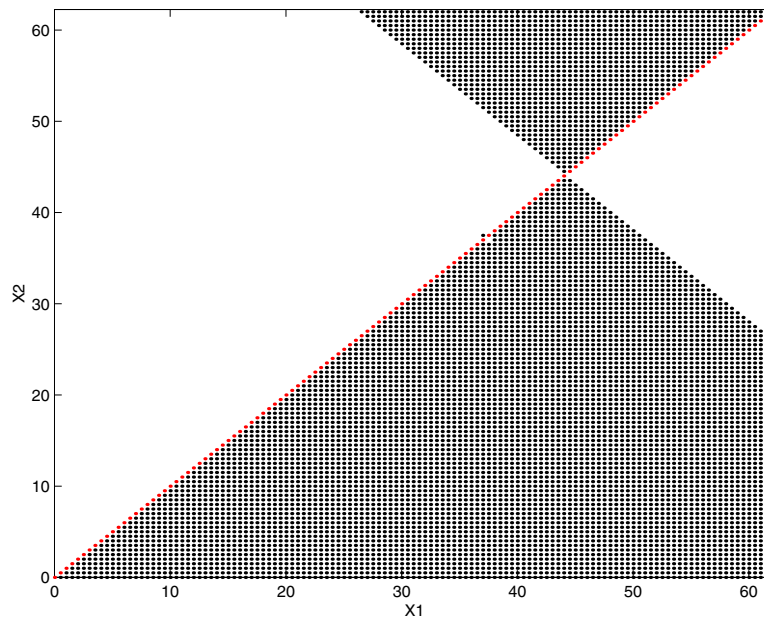


Figura V.11: L'inversione illustrata nel Risultato 3 con $R = 150$.

Conclusioni

In questa tesi sono stati affrontati vari aspetti del controllo in retroazione dei sistemi manifatturieri.

Si sono presi in considerazione quei sistemi costituiti da una sola macchina, in grado di lavorare più tipi di parti, impiegando però un certo tempo di setup nel cambio della lavorazione tra i differenti tipi. Si è assunto che i sistemi non fossero perfettamente affidabili e si è considerata un'approssimazione continua per i modelli.

Il controllo di questo tipo di sistemi è stato ampiamente studiato da diversi anni e una struttura di tipo gerarchico ha permesso la scomposizione del problema generale e complesso del controllo di un sistema manifatturiero nei suoi aspetti principali. Ciò ha consentito di isolare la problematica relativa allo scheduling dinamico dei setup.

In questo contesto, sono state utilizzate le politiche “corridoio”, che permettono di risolvere il problema dello scheduling dinamico.

Come primo contributo della tesi, si è determinato il comportamento ottimo a regime nell'ambito delle politiche “corridoio” e si è analizzato analiticamente il comportamento transitorio delle traiettorie relative a questo problema.

Il principale contributo della tesi riguarda l'ottimizzazione del comportamento transitorio dei sistemi manifatturieri di tipo "make to order" capaci di produrre due tipi di parti. L'aspetto innovativo è stato la scelta di un indice di prestazione che permette di valutare la bontà del regime transitorio del sistema. Relativamente a questo indice, sono state determinate le proprietà analitiche della politica ottima nel caso dei buffer infiniti. Estendendo il problema anche al caso con buffer finiti si sono incontrati evidenti problemi di complessità, ed è stato possibile ricavare solo alcune proprietà.

Le soluzioni numeriche ottenute hanno fornito lo spunto necessario alla individuazione delle proprietà analitiche determinate per il problema nel caso di buffer finiti.

Le direzioni verso le quali orientare le future ricerche sono molteplici. Innanzitutto sono necessarie ulteriori ricerche nel caso dei buffer finiti. È anche opportuno estendere il problema al caso di più di due parti e rimuovere l'assunzione di simmetria qui fatta per snellire la trattazione.

Per quanto riguarda le politiche corridoio bisognerebbe considerare corridoi più complessi di quelli lineari. Questi ultimi corridoi infatti, sono stati gli unici ad essere trattati in questo lavoro.

Appendice A

Listato del programma

```
#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <except.h>
#include <iostream.h>

// Dichiarazione delle funzioni
double calcola_integrale(double i,double j,int prod1,int prod2,int dom1,
    int dom2,double intervallo);
double integrale1(int i,int j,double intervallo);
double integrale2(int i,int j,double intervallo);
double integrale0(int i,int j,double intervallo);
int MCD(int divisore,int dividendo);
double min(double primo,double secondo);
double min3(double primo,double secondo,double terzo);
void dealloca(double **matrice);
void dealloca_i(int **matrice);
void dealloca_stradario(int ***matrice);
void display(double **matrice);
void display_i(int **matrice);
void display_stradario(int ***matrice);
void uguale(double **bersaglio,double **sorgente);
double calcola_scarto(int *riga,int *colonna,double **a,double **b);
double **nuova_matrice();
void crea_buffer();
void aggiorna_buffer();

// Dichiarazione delle variabili globali
int Passi = 1000;
int NumeroPunti = 250;
int d1 = 1;
```

```

int d2 = 1;
int u1 = 3;
int u2 = 3;
int Passi_Minimo =120;
const N = 20;
double tau12 = 5.0;
double tau21 =5.0;
double deltat = tau12/N;
int k1,k2,k3,k4,k5,k6,riga,colonna;
double deltax,c1,c2,R1,R2,gxij,costo,costo_medio,T1,T2,Xa1,Xa2,Xb1,Xb2,
        Xc1,Xc2,Xd1,Xd2,parte1,parte2,parte3,parte4,tempo_limite;
double **J1[N+1];
double **J2[N+1];
double **J1min;
double **J2min;

FILE *fp1;
FILE *fp2;
FILE *fp3;
FILE *fp4;
FILE *fp5;
FILE *fp6;
FILE *fp7;
FILE *fp8;

// Programma Principale
main()
{
    int ris1,ris2,ris,i,j,k,k1bis,k2bis,k3bis,k4bis,k5bis,k6bis;

    k1 = u1-d1;
    k2 = d2;
    k3 = d1;
    k4 = u2-d2;
    clrscr();
    printf("K1 = %d, K2 = %d, K3 = %d, K4 = %d.\n",k1,k2,k3,k4);
    ris1 = MCD(k1,k2);
    printf("Il max.com.div tra K1 e K2 e' : %d\n",ris1);
    ris2 = MCD(k3,k4);
    printf("Il max.com.div tra K3 e K4 e' : %d\n",ris2);
    ris = MCD(ris1,ris2);
    printf("Quindi...\n");
    printf("Il max.com.div tra K1,K2,K3 e K4 e' : %d\n",ris);
    if (ris != 1)
    {
        k1 = k1/ris;
        k2 = k2/ris;
        k3 = k3/ris;
        k4 = k4/ris;
    }
}

```

```

}
printf("K1 = %d, K2 = %d, K3 = %d, K4 = %d.\n",k1,k2,k3,k4);
deltax = (u1-d1)*deltat/k1;
printf("deltax = %.10f\n",deltax);
printf("deltat = %.10f\n",deltat);
k6 = tau21*d2/deltax;
k5 = tau12*d1/deltax;
printf("K5 = %d, K6 = %d\n",k5,k6);
c1 = 3.0;
c2 = 6.0;
R1 = 0.0;
R2 = 0.0;
printf("C1 = %f, C2 = %f\n",c1,c2);
printf("R1 = %f, R2 = %f\n",R1,R2);
printf("tau = %f\n",tau12);
printf("N = %d\n",N);

/* Calcolo del costo MEDIO del ciclo limite */
//Intervallo di produzione della parte 1
T1 = ((u2*d1)*(tau12+tau21))/(u2*u1-d2*u1-u2*d1);
//Intervallo di produzione della parte 2
T2 = ((u1*d2)*(tau12+tau21))/(u2*u1-d2*u1-u2*d1);
//Coordinate degli angoli del ciclo limite
Xa1 = T1*(u1-d1)-tau21*d1;
Xa2 = 0.0;
Xb1 = T1*(u1-d1);
Xb2 = tau21*d2;
Xc1 = 0.0;
Xc2 = T1*d2+tau21*d2;
Xd1 = tau12*d1;
Xd2 = tau12*d2+T1*d2+tau21*d2;
//Costo del setup 2-1
parte1 = (c1*d1 + c2*d2)*tau21*tau21/2.0 + (c1*Xa1 + c2*Xa2)*tau21;
//Costo della produzione della parte 1
parte2 = (c1*(d1 - u1) + c2*d2)*T1*T1/2.0 + (c1*Xb1 + c2*Xb2)*T1;
//Costo del setup 1-2
parte3 = (c1*d1 + c2*d2)*tau12*tau12/2.0 + (c1*Xc1 + c2*Xc2)*tau12;
//Costo della produzione della parte 2
parte4 = (c1*d1 + c2*(d2 - u2))*T2*T2/2.0 + (c1*Xd1 + c2*Xd2)*T2;
//Costo del ciclo limite
costo = parte1+parte2+parte3+parte4;
//Durata del ciclo limite
tempo_limite = T1 + T2 + tau12 + tau21;
//Costo MEDIO del ciclo limite
costo_medio = costo/tempo_limite;
printf("Il costo del ciclo limite e' : %f\n",costo);
printf("La durata del ciclo limite e' : %f\n",tempo_limite);
printf("Il costo MEDIO del ciclo limite e' : %f\n",costo_medio);
/*Creazione Dinamica delle matrici di calcolo */

```

```

    crea_buffer();
/* La prima iterazione in assoluto */
    for ( i = 0; i < NumeroPunti; i++)
        for ( j = 0; j < NumeroPunti; j++)
            {
                gxij=c2*(i)*deltax+c1*(j)*deltax;
                J1[0][i][j]=min(integrale1(i,j,deltat),integrale0(i,j,
                    deltat))-(costo_medio*deltat);
                J2[0][i][j]=min(integrale2(i,j,deltat),integrale0(i,j,
                    deltat))-(costo_medio*deltat);
            }

    printf("> 1/%d : J1(0,119)=%f J1(0,120)=%f J1(0,121)=%f\n",Passi,
        J1[0][0][119],J1[0][0][120],J1[0][0][121]);
    aggiorna_buffer();
/* Dalla seconda fino alla N-esima iterazione. */
    for ( k = 2; k <= N; k++ )
        {
            for ( i = 0; i < NumeroPunti; i++)
                for ( j = 0; j < NumeroPunti; j++)
                    {
                        gxij=c2*(i)*deltax+c1*(j)*deltax;
                        if (i+k2 > NumeroPunti-1) k2bis = NumeroPunti-i-1;
                        else k2bis = k2;
                        if (j-k1 < 0) k1bis = j;
                        else k1bis = k1;
                        if (i-k4 < 0) k4bis = i;
                        else k4bis = k4;
                        if (j+k3 > NumeroPunti-1) k3bis = NumeroPunti-j-1;
                        else k3bis = k3;
                        if (i+k6 > NumeroPunti-1) k6bis = NumeroPunti-i-1;
                        else k6bis = k6;
                        if (j+k5 > NumeroPunti-1) k5bis = NumeroPunti-j-1;
                        else k5bis = k5;
                        if ((i == 60)&&(j == 1))
                            J1[0][i][j] = min3((integrale1(i,j,deltat)+
                                (costo_medio*(k-1)*deltat)+
                                J1[1][i+k2bis][j-k1bis])-
                                (k*costo_medio*deltat),
                                integrale0(i,j,k*deltat)-
                                (k*costo_medio*deltat),
                                (integrale1(i,j,deltat)+
                                (costo_medio*(k-1)*deltat)+
                                J1[1][60][0])-
                                (k*costo_medio*deltat));
                            else
                                J1[0][i][j] = min((integrale1(i,j,deltat)+
                                    (costo_medio*(k-1)*deltat)+
                                    J1[1][i+k2bis][j-k1bis])-

```

```

                (k*costo_medio*deltat),
                integrale0(i,j,k*deltat)-
                (k*costo_medio*deltat));
if ((i == 1)&&(j == 60))
    J2[0][i][j] = min3((integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][i-k4bis][j+k3bis])-
        (k*costo_medio*deltat),
        integrale0(i,j,k*deltat)-
        (k*costo_medio*deltat),
        (integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][0][60])-
        (k*costo_medio*deltat));
    else
        J2[0][i][j] = min((integrale2(i,j,deltat)+
            (costo_medio*(k-1)*deltat)+
            J2[1][i-k4bis][j+k3bis])-
            (k*costo_medio*deltat),
            integrale0(i,j,k*deltat)-
            (k*costo_medio*deltat));
}

printf("> %d/%d : J1(0,119)=%f J1(0,120)=%f J1(0,121)=%f\n",
    k,Passi,J1[0][0][119],J1[0][0][120],J1[0][0][121]);
//scarto = calcola_scarto(&riga,&colonna,J1[0],J1[1]);
//printf("Iterazione %d / %d ,Scarto : %f, %d, %d\n",k,Passi,
    scarto,riga,colonna);
aggiorna_buffer();
}

/* Dalla N+1-esima iterazione, fino alla penultima */ /* Dalla
N+1-esima iterazione, fino alla ultima */
for ( k = N+1; k <= Passi; k++ )
{
    for ( i = 0; i < NumeroPunti; i++)
        for ( j = 0; j < NumeroPunti; j++)
            {
                gxij=c2*(i)*deltax+c1*(j)*deltax;
                if (i+k2 > NumeroPunti-1) k2bis = NumeroPunti-i-1;
                else k2bis = k2;
                if (j-k1 < 0) k1bis = j;
                else k1bis = k1;
                if (i-k4 < 0) k4bis = i;
                else k4bis = k4;
                if (j+k3 > NumeroPunti-1) k3bis = NumeroPunti-j-1;
                else k3bis = k3;
                if (i+k6 > NumeroPunti-1) k6bis = NumeroPunti-i-1;
                else k6bis = k6;
                if (j+k5 > NumeroPunti-1) k5bis = NumeroPunti-j-1;

```

```

else k5bis = k5;
if ((i == 60)&&(j == 1))
    J1[0][i][j] = min3((integrale1(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J1[1][i+k2bis][j-k1bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau12)+
        (costo_medio*(k-N)*deltat)+
        J2[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau12)),
        (integrale1(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J1[1][60][0])-
        (k*deltat*costo_medio));
else
    J1[0][i][j] = min((integrale1(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J1[1][i+k2bis][j-k1bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau12)+
        (costo_medio*(k-N)*deltat)+
        J2[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau12)));
if ((i == 1)&&(j == 60))
    J2[0][i][j] = min3((integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][i-k4bis][j+k3bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau21)+
        (costo_medio*(k-N)*deltat)+
        J1[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau21)),
        (integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][0][60])-
        (k*deltat*costo_medio));
else
    J2[0][i][j] = min((integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][i-k4bis][j+k3bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau21)+
        (costo_medio*(k-N)*deltat)+
        J1[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau21)));

```

```

    }

    printf("> %d/%d : J1(0,119)=%f J1(0,120)=%f J1(0,121)=%f\n",
           k,Passi,J1[0][0][119],J1[0][0][120],J1[0][0][121]);
    aggiorna_buffer();
}
J1min = new double*[NumeroPunti];
for ( i = 0; i < NumeroPunti; i++)
    J1min[i] = new double[NumeroPunti];
for ( i = 0; i < NumeroPunti; i++)
    for ( j = 0; j < NumeroPunti; j++)
        J1min[i][j] = 100000000.0;
J2min = new double*[NumeroPunti];
for ( i = 0; i < NumeroPunti; i++)
    J2min[i] = new double[NumeroPunti];
for ( i = 0; i < NumeroPunti; i++)
    for ( j = 0; j < NumeroPunti; j++)
        J2min[i][j] = 100000000.0;
for ( k = Passi+1; k <= Passi+Passi_Minimo; k++ )
{
    for ( i = 0; i < NumeroPunti; i++)
        for ( j = 0; j < NumeroPunti; j++)
        {
            gxij=c2*(i)*deltax+c1*(j)*deltax;
            if (i+k2 > NumeroPunti-1) k2bis = NumeroPunti-i-1;
            else k2bis = k2;
            if (j-k1 < 0) k1bis = j;
            else k1bis = k1;
            if (i-k4 < 0) k4bis = i;
            else k4bis = k4;
            if (j+k3 > NumeroPunti-1) k3bis = NumeroPunti-j-1;
            else k3bis = k3;
            if (i+k6 > NumeroPunti-1) k6bis = NumeroPunti-i-1;
            else k6bis = k6;
            if (j+k5 > NumeroPunti-1) k5bis = NumeroPunti-j-1;
            else k5bis = k5;
            if ((i == 60)&&(j == 1))
                J1[0][i][j] = min3((integrale1(i,j,deltat)+
                                     (costo_medio*(k-1)*deltat)+
                                     J1[1][i+k2bis][j-k1bis])-
                                     (k*deltat*costo_medio),
                                     (integrale0(i,j,tau12)+
                                     (costo_medio*(k-N)*deltat)+
                                     J2[N][i+k6bis][j+k5bis])-
                                     (costo_medio*((k-N)*
                                     deltat+tau12)),
                                     (integrale1(i,j,deltat)+
                                     (costo_medio*(k-1)*deltat)+
                                     J1[1][60][0])-

```

```

                                (k*deltat*costo_medio));
else
    J1[0][i][j] = min((integrale1(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J1[1][i+k2bis][j-k1bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau12)+
        (costo_medio*(k-N)*deltat)+
        J2[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau12)));
if ((i == 1)&&(j == 60))
    J2[0][i][j] = min3((integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][i-k4bis][j+k3bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau21)+
        (costo_medio*(k-N)*deltat)+
        J1[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau21)),
        (integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][0][60])-
        (k*deltat*costo_medio));
else
    J2[0][i][j] = min((integrale2(i,j,deltat)+
        (costo_medio*(k-1)*deltat)+
        J2[1][i-k4bis][j+k3bis])-
        (k*deltat*costo_medio),
        (integrale0(i,j,tau21)+
        (costo_medio*(k-N)*deltat)+
        J1[N][i+k6bis][j+k5bis])-
        (costo_medio*((k-N)*
        deltat+tau21)));
if ( J1[0][i][j] < J1min[i][j] )
    J1min[i][j] = J1[0][i][j];
if ( J2[0][i][j] < J2min[i][j] )
    J2min[i][j] = J2[0][i][j];
}

printf("M> %d/%d : J1(0,119)=%f J1(0,120)=%f J1(0,121)=%f\n"
    ,k-Passi,Passi_Minimo,J1[0][0][119],J1[0][0][120],
    J1[0][0][121]);
aggiorna_buffer();
}

```

```

//Salvataggio dei risultati trovati su file,
in modo da poterli far elaborare anche da MATLAB

```

```

fp7 = fopen("j1min.m", "w");
fprintf(fp7, "J1min=[");
for (i=0; i<NumeroPunti; i++)
{
    for (j=0; j<NumeroPunti; j++)
        fprintf(fp7, "%f ", J1min[i][j]);
    fprintf(fp7, "\n");
}
fprintf(fp7, "];");
fclose(fp7);
fp8 = fopen("j2min.m", "w");
fprintf(fp8, "J2min=[");
for (i=0; i<NumeroPunti; i++)
{
    for (j=0; j<NumeroPunti; j++)
        fprintf(fp8, "%f ", J2min[i][j]);
    fprintf(fp8, "\n");
}
fprintf(fp8, "];");
fclose(fp8);
fp4 = fopen("generale.m", "w");
fprintf(fp4, "NumeroPunti=%d;\n", NumeroPunti);
fprintf(fp4, "kappa(1)=%d;\n", k1);
fprintf(fp4, "kappa(2)=%d;\n", k2);
fprintf(fp4, "kappa(3)=%d;\n", k3);
fprintf(fp4, "kappa(4)=%d;\n", k4);
fprintf(fp4, "kappa(5)=%d;\n", k5);
fprintf(fp4, "kappa(6)=%d;\n", k6);
fprintf(fp4, "u1=%d\n", u1);
fprintf(fp4, "u2=%d\n", u2);
fprintf(fp4, "d1=%d\n", d1);
fprintf(fp4, "d2=%d\n", d2);
fprintf(fp4, "tau12=%f\n", tau12);
fprintf(fp4, "tau21=%f\n", tau21);
fprintf(fp4, "deltax=%f\n", deltax);
fprintf(fp4, "deltat=%f\n", deltat);
fprintf(fp4, "c1=%f\n", c1);
fprintf(fp4, "c2=%f\n", c2);
fprintf(fp4, "R1=%f\n", R1);
fprintf(fp4, "R2=%f\n", R2);
fprintf(fp4, "costo=%f\n", costo);
fprintf(fp4, "tempo_limite=%f\n", tempo_limite);
fprintf(fp4, "costo_medio=%f\n", costo_medio);
fclose(fp4);
fp5 = fopen("j1.m", "w");
fprintf(fp5, "J1=[");
for (i=0; i<NumeroPunti; i++)
{
    for (j=0; j<NumeroPunti; j++)

```

```

        fprintf(fp5,"%f ",J1[0][i][j]);
        fprintf(fp5,"\n");
    }
    fprintf(fp5,"];");
    fclose(fp5);
    fp6 = fopen("j2.m","w");
    fprintf(fp6,"J2=[");
    for (i=0;i<NumeroPunti;i++)
    {
        for (j=0;j<NumeroPunti;j++)
            fprintf(fp6,"%f ",J2[0][i][j]);
        fprintf(fp6,"\n");
    }
    fprintf(fp6,"];");
    fclose(fp6);
    for (i=0;i<=N;i++)
    {
        de_alloca(J1[i]);
        de_alloca(J2[i]);
    }
    de_alloca(J1min);
    de_alloca(J2min);
    return 0;
}
// Corpo delle Funzioni -----

// I parametro colonna, II parametro riga
double calcola_integrale(double xcolonna,double xriga,int
prod1,int prod2,int dom1,int dom2,double intervallo) {
    double integrale;

    integrale = (c2*xcolonna + c1*xriga)*intervallo +
        (c1*(dom1 -prod1) +
        c2*(dom2 - prod2))*intervallo*intervallo/2.0;
    return integrale;
}

double integrale1(int i,int j,double intervallo) {
    double risultato,tempo,x,tempo1,tempo2,x1,x2;

    if ((i+k2<= NumeroPunti-1)&&(j-k1>=0))
        risultato = calcola_integrale(i*deltax,j*deltax,u1,0,
d1,d2,intervallo);
    if ((i+k2 > NumeroPunti-1)&&(j-k1 >= 0))
    {
        tempo = (NumeroPunti-1-i)*deltax/d2;
        x = j*deltax + (d1-u1)*tempo;
        risultato = calcola_integrale(i*deltax,j*deltax,u1,0,d1,d2,
tempo)+

```

```

        calcola_integrale((NumeroPunti-1)*deltax,x,u1,0,
        d1,0,intervallo-tempo)+R2*d2*(intervallo-tempo);
    }
    if ((i+k2 <= NumeroPunti-1)&&(j-k1 < 0))
    {
        tempo = -j*deltax/(d1-u1);
        x = i*deltax + d2*tempo;
        risultato = calcola_integrale(i*deltax,j*deltax,u1,0,d1,d2,
        tempo)+
        calcola_integrale(x,0.0,d1,0,d1,d2,
        intervallo-tempo);
    }
    if ((i+k2 > NumeroPunti-1)&&(j-k1 < 0))
    {
        tempo1 = (NumeroPunti-1-i)*deltax/d2;
        x1 = j*deltax + (d1-u1)*tempo1;
        tempo2 = -j*deltax/(d1-u1);
        x2 = i*deltax + d2*tempo2;
        if (tempo1 <= tempo2)
            risultato = calcola_integrale(i*deltax,j*deltax,u1,0,
            d1,d2,tempo1) +
            calcola_integrale((NumeroPunti-1)*deltax,
            x1,u1,0,d1,0,tempo2-tempo1) +
            calcola_integrale((NumeroPunti-1)*deltax,
            0.0,d1,0,d1,0,intervallo-tempo2)+
            R2*d2*(intervallo-tempo1);
        else
            risultato = calcola_integrale(i*deltax,j*deltax,u1,0,
            d1,d2,tempo2) +
            calcola_integrale(x2,0.0,d1,0,d1,d2,
            tempo1-tempo2) +
            calcola_integrale((NumeroPunti-1)*deltax,
            0.0,d1,0,d1,0,intervallo-tempo1)+
            R2*d2*(intervallo-tempo1);
    }
    return risultato;
}

```

```

double integrale2(int i,int j,double intervallo) {
    double risultato,tempo,x,tempo1,tempo2,x1,x2;

    if ((i-k4 >= 0)&&(j+k3<=NumeroPunti-1))
        risultato = calcola_integrale(i*deltax,j*deltax,0,u2,d1,d2,
        intervallo);
    if ((i-k4 < 0)&&(j+k3 <= NumeroPunti-1))
    {
        tempo = - i*deltax/(d2-u2);
        x = j*deltax + d1*tempo;
    }
}

```

```

        risultato = calcola_integrale(i*deltax,j*deltax,0,
            u2,d1,d2,tempo)+
            calcola_integrale(0.0,x,0,d2,d1,d2,
                intervallo-tempo);
    }
    if ((i-k4 >= 0)&&(j+k3 > NumeroPunti-1))
    {
        tempo = (NumeroPunti-1-j)*deltax/d1;
        x = i*deltax + (d2-u2)*tempo;
        risultato = calcola_integrale(i*deltax,j*deltax,0,
            u2,d1,d2,tempo)+
            calcola_integrale(x,(NumeroPunti-1)*deltax,
                0,u2,0,d2,intervallo-tempo)+
            R1*d1*(intervallo-tempo);
    }
    if ((i-k4 < 0)&&(j+k3 > NumeroPunti-1))
    {
        tempo1 = - i*deltax/(d2-u2);
        x1 = j*deltax + d1*tempo1;
        tempo2 = (NumeroPunti-1-j)*deltax/d1;
        x2 = i*deltax + (d2-u2)*tempo2;
        if (tempo1 <= tempo2)
            risultato = calcola_integrale(i*deltax,j*deltax,0,
                u2,d1,d2,tempo1) +
                calcola_integrale(0.0,x1,0,d2,d1,d2,
                    tempo2-tempo1) +
                calcola_integrale(0.0,(NumeroPunti-1)*
                    deltax,0,d2,0,d2,intervallo-tempo2)+
                R1*d1*(intervallo-tempo2);
        else
            risultato = calcola_integrale(i*deltax,j*deltax,0,
                u2,d1,d2,tempo2) +
                calcola_integrale(x2,(NumeroPunti-1)*deltax,
                    0,u2,0,d2,tempo1-tempo2) +
                calcola_integrale(0.0,(NumeroPunti-1)*
                    deltax,0,d2,0,d2,intervallo-tempo1)+
                R1*d1*(intervallo-tempo2);
    }
    return risultato;
}

double integrale0(int i,int j,double intervallo) {
    double risultato,tempo,x,tempo1,tempo2,x1,x2;

    if ((i+k6 <= NumeroPunti-1)&&(j+k5<=NumeroPunti-1))
        risultato = calcola_integrale(i*deltax,j*deltax,0,0,d1,d2,
            intervallo);
    if ((i+k6 > NumeroPunti-1)&&(j+k5 <= NumeroPunti-1))
    {

```

```

        tempo = (NumeroPunti-1-i)*deltax/d2;
        x = j*deltax + d1*tempo;
        risultato = calcola_integrale(i*deltax,j*deltax,0,0,d1,d2,
            tempo)+
            calcola_integrale((NumeroPunti-1)*deltax,x,0,0,
            d1,0,intervallo-tempo)+R2*d2*(intervallo-tempo);
    }
    if ((i+k6 <= NumeroPunti-1)&&(j+k5 > NumeroPunti-1))
    {
        tempo = (NumeroPunti-1-j)*deltax/d1;
        x = i*deltax + d2*tempo;
        risultato = calcola_integrale(i*deltax,j*deltax,0,0,
            d1,d2,tempo)+
            calcola_integrale(x,(NumeroPunti-1)*deltax,
            0,0,0,d2,intervallo-tempo)+
            R1*d1*(intervallo-tempo);
    }
    if ((i+k6 > NumeroPunti-1)&&(j+k5 > NumeroPunti-1))
    {
        tempo1 = (NumeroPunti-1-i)*deltax/d2;
        x1 = j*deltax + d1*tempo1;
        tempo2 = (NumeroPunti-1-j)*deltax/d1;
        x2 = i*deltax + d2*tempo2;
        if (tempo1 <= tempo2)
            risultato = calcola_integrale(i*deltax,j*deltax,0,0,
                d1,d2,tempo1) +
                calcola_integrale((NumeroPunti-1)*deltax,
                x1,0,0,d1,0,tempo2-tempo1) +
                calcola_integrale((NumeroPunti-1)*deltax,
                (NumeroPunti-1)*deltax,0,0,0,0,
                intervallo-tempo2)+
                R2*d2*(intervallo-tempo1)+
                R1*d1*(intervallo-tempo2);
        else
            risultato = calcola_integrale(i*deltax,j*deltax,0,0,
                d1,d2,tempo2) +
                calcola_integrale(x2,(NumeroPunti-1)*deltax,
                0,0,0,d2,tempo1-tempo2) +
                calcola_integrale((NumeroPunti-1)*deltax,
                (NumeroPunti-1)*deltax,0,0,0,0,
                intervallo-tempo1)+
                R2*d2*(intervallo-tempo1)+
                R1*d1*(intervallo-tempo2);
    }
    return risultato;
}

int MCD(int dividendo,int divisore) {
    int appoggio;

```

```

    div_t x;

    if (dividendo < divisore)
    {
        appoggio = dividendo;
        dividendo = divisore;
        divisore = appoggio;
    }
    x = div(dividendo,divisore);
    while (x.rem != 0)
    {
        dividendo = divisore;
        divisore = x.rem;
        x = div(dividendo,divisore);
    }
    return divisore;
}

double min(double primo,double secondo) {
    double confronto;

    confronto=primo - secondo;
    if (confronto >= 0.0)
        return secondo;
    else
        return primo;
}

double min3(double primo,double secondo,double terzo) {
    double minimo12,minimo;

    minimo12 = min(primo,secondo);
    minimo = min(minimo12,terzo);
    return minimo;
}

void dealloca(double **matrice) {
    for (int i = 0; i < NumeroPunti; i++)
        delete[] matrice[i];
    delete[] matrice;
}

void dealloca_i(int **matrice) {
    for (int i = 0; i < NumeroPunti; i++)
        delete[] matrice[i];
    delete[] matrice;
}

```

```

void dealloca_stradario(int ***matrice) {
    for (int i = 0; i < NumeroPunti; i++)
        for(int j = 0; j <NumeroPunti; j++)
            delete[] matrice[i][j];
    for (i=0; i<NumeroPunti; i++)
        delete[] matrice[i];
    delete[] matrice;
}

void display(double **matrice) {
    for (int i = 0; i < NumeroPunti; i++)
    {
        for (int j = 0; j < NumeroPunti; j++)
            printf("%.6f ",matrice[i][j]);
        printf("\n");
    }
}

void display_i(int **matrice) {
    for (int i = 0; i < NumeroPunti; i++)
    {
        for (int j = 0; j < NumeroPunti; j++)
            printf("%d ",matrice[i][j]);
        printf("\n");
    }
}

void display_stradario(int ***matrice) {
    int i,j,m;

    for (m = 0;m<4;m++)
    {
        printf("Sezione (:,:,%d) \n",m);
        for (i=0;i<NumeroPunti;i++)
        {
            for (j=0;j<NumeroPunti;j++)
                printf("%d ",matrice[i][j][m]);
            printf("\n");
        }
        getchar();
    }
    return;
}

double calcola_scarto(int *riga,int *colonna,double **a,double
**b) {
    double differenza,massimo;
    int i,j;

```

```

    massimo = 0.0;
    for (i = 0; i < NumeroPunti; i++)
        for (j = 0; j < NumeroPunti; j++)
            {
                differenza = fabs(a[i][j] - b[i][j]);
                if (differenza > massimo)
                    {
                        massimo = differenza;
                        *riga = i;
                        *colonna = j;
                    }
            }
    return massimo;
}

double **nuova_matrice() {
    double **nuovam;
    int i;

    try
    {
        nuovam = new double*[NumeroPunti];
        for ( i = 0; i < NumeroPunti; i++)
            nuovam[i] = new double[NumeroPunti];
    }
    catch (xalloc)
    {
        printf("Non c' e' memoria sufficiente per creare
            tutte le matrici");
        exit(-1);
    }
    return nuovam;
}

void crea_buffer() {
    int i;

    for (i=0;i<=N;i++)
    {
        J1[i] = nuova_matrice();
        J2[i] = nuova_matrice();
    }
    return;
}

void aggiorna_buffer() {
    int i,j,m;

    for (m=N;m>=1;m--)

```

```
    for (i=0;i<NumeroPunti;i++)
      for (j=0;j<NumeroPunti;j++)
      {
        J1[m][i][j]=J1[m-1][i][j];
        J2[m][i][j]=J2[m-1][i][j];
      }
  return;
}
```

Ringraziamenti

La realizzazione di questa tesi sarebbe stata impossibile, o certamente molto più complicata, senza l'apporto e l'aiuto di alcune persone.

Desidero ringraziare il professore Salvatore Nicosia, sempre pronto a mettere a disposizione di noi studenti la sua impagabile esperienza, e il professore Paolo Valigi, il cui contributo è stato essenziale ai fini della riuscita del lavoro.

Un particolare ringraziamento va all'Ing. Francesco Martinelli, per il suo costante aiuto e la sua disponibilità.

Inoltre, non posso non ringraziare tutte le persone che lavorano presso il dipartimento di informatica, sistemi e produzione (DISP) dell'università di Roma Tor Vergata, nel settore automatica e sistemi di automazione industriali.

Infine, ringrazio tutti i miei amici, i quali si sono rivelati degli insostituibili compagni di viaggio durante tutta la mia carriera universitaria.

Bibliografia

- [1] S.B. Gershwin. “Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems”. In *Proc. of IEEE, Special Issue on Dynamics of Discrete Event Systems*, vol. 77, pp. 195-209, Jan. 1989.
- [2] J. Perkins and P.R. Kumar. “Stable, Distributed, Real-time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems”. *IEEE Transactions on Automatic Control*, vol. 1:pp. 139–148, 1989.
- [3] P.R. Kumar and T.I. Seidman. “Dynamic Instabilities and Stabilization Methods in Distributed Real-Time Scheduling of Manufacturing Systems”. *IEEE Transactions on Automatic Control*, vol. 35:pp. 289–298, 1990.
- [4] A. Sharifnia, M. Caramanis, and S.B. Gershwin. “Dynamic Setup Scheduling and Flow Control in Manufacturing System”. *Discrete Event Dynamic Systems*, vol. 1:pp. 149–175, 1991.
- [5] N. Srivatsan and S.B. Gershwin. “Selection of Setup Times in a Hierarchical Controlled Manufacturing System”. In *Proceedings of the 29st Conference on Decision and Control*, pp. 575-581, Dec. 1990.

- [6] M. Caramanis, A. Sharifnia, J. Hu, and S.B. Gershwin. "Development of a Science Base for Planning and Scheduling Manufacturing Systems". In *NSF Grantees meeting, Austin, TX*, Jan. 1991.
- [7] T.W.E. Lau and A. Sharifnia. "Dynamic Capacity Allocation in Manufacturing Systems with Significant Setup Times ". In *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, pp. 436-442, 1994.
- [8] M.P. Groover. *Automation, production systems, and computer integrated manufacturing*. Prentice-Hall, second edition, 2001.
- [9] S. Connolly, Y. Dallery, and S.B. Gershwin. "A Real-time Policy for Performing Setup Changes in a Manufacturing System". In *Proceedings of 31st Conf. on Decision and Control (CDC '92), Tucson, AZ*, pp. 764-770, 1992.
- [10] J. Hu and M. Caramanis. "Near Optimal Set-up Scheduling for Flexible Manufacturing Systems". In *Proceedings of the Third International Conference on Computer Integrated Manufacturing*, pp. 192-201, 1992.
- [11] F. Martinelli and P. Valigi. "A dynamic control problem for two part-type pull manufacturing system". In *Proceedings of 37st Conference on Decision and Control, Tampa, Florida*, pp. 2171-2172, Dec. 1998.
- [12] F. Martinelli and P. Valigi. "A scheduling problem for a finite buffer capacity pull manufacturing system". In *1998 IEEE Int. Conf. on Robotics and Automation, (Leuven, Belgium)*, pp. 2753-2758, May 1998.

- [13] J.A. Buzacott and J.G. Shanthikumar. *Stochastic models of manufacturing systems*. Prentice-Hall, 1993.
- [14] M. Chiarvesio. Organizzazione e gestione della produzione e Gestione dei materiali. Traccia delle lezioni.
- [15] S.B. Gershwin. *Manufacturing systems engineering*. PTR Prentice-Hall, 1994.
- [16] C.M. Libosvar. “Hierarchies in production management and control: a survey”. MIT Laboratory for information and decision systems rep. LIDS-P-1734, 1988.
- [17] C.M. Libosvar. “Hierarchical production management, the flow-control layer”. Thesis for the degree of Docteur de l’Universite de Metz, 1988.
- [18] J. Kimemia and S.B. Gershwin. “An algorithm for the computer control of a flexible manufacturing system”. *IIE Transactions*, vol. 15(No. 4):pp. 353–362, Dec. 1983.
- [19] S.B. Gershwin, M. Caramanis, and P. Murray. “Simulation Experience with a Hierarchical Scheduling Policy for a Simple Manufacturing System”. In *Proceedings of 27st Conference on Decision and Control, Austin, Texas*, pp. 1841-1849, Dec. 1988.
- [20] M.P. Garcia and L.O. Brandão. “When a Cone Policy is Coherent”. In *Proceedings of 35st Conference on Decision and Control, Kobe, Japan*, pp. 3330-3331, Dec. 1996.

- [21] E. Fornasini e G. Marchesini. *Appunti di teoria dei sistemi*. Edizioni Libreria Progetto, 1992.
- [22] S.X. Bai and M. Elhafsi. “Optimal Feedback Control of a Manufacturing System with Setup Changes”. In *Proceedings of the Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, pp. 191-196, 1994.
- [23] M. Del Gaudio, F. Martinelli, and P. Valigi. “A scheduling problem for two competing queues with finite capacity and non negligible setup times”. Sottomesso alla 40st IEEE Conference on Decision and Control, 2001.