

Titolo: Procedura per la progettazione automatica di circuiti pneumatici di comando

Autore: A.Macor - A.Mirandola - R.Pignatelli - G.Zamboni

Abstract: Testo non disponibile

PROCEDURA PER LA PROGETTAZIONE AUTOMATICA DI CIRCUITI PNEUMATICI DI COMANDO

A. Macor - A. Mirandola - R. Pignatelli - G. Zamboni

Dipartimento di Ingegneria Meccanica - Università di Padova - Padova

SOMMARIO

La progettazione industriale dei circuiti pneumatici di comando è attualmente basata su procedure e criteri che il singolo progettista desume dalla letteratura e dall'esperienza.

Nell'ambito dell'automazione delle procedure di progettazione, ormai diffusa in tutti i settori dell'ingegneria, si inserisce il presente lavoro che si propone come obiettivo la formalizzazione di un procedimento particolarmente efficace che possa essere facilmente automatizzato in un calcolatore.

La procedura che qui si propone fornisce una concreta risoluzione del problema tramite un insieme di equazioni logiche che consentono di minimizzare il numero dei componenti necessari per il corretto funzionamento del circuito di comando.

1. INTRODUZIONE

Come è noto, una macchina automatica, chiamata a realizzare una sequenza di movimenti secondo un programma di lavoro prestabilito, è generalmente costituita da due sezioni distinte:

- la sezione di potenza, che comprende gli attuatori e gli elementi funzionali che realizzano direttamente i movimenti;
- la sezione di comando, che comprende tutti gli elementi necessari per la coordinazione dei movimenti della sezione di potenza.

Limitando l'attenzione alle macchine con circuiti di potenza di tipo pneumatico, si può dire che, a seconda della soluzione tecnologica adottata per la parte di comando, sono possibili due configurazioni principali:

- macchine elettropneumatiche: la sezione di comando è di tipo elettrico o elettronico;
- macchine interamente pneumatiche: entrambe le sezioni sono di tipo pneumatico.

La scelta più appropriata della tecnologia di comando è cosa di una certa delicatezza, poichè in essa intervengono vari fattori quali la funzionalità, i costi della macchina, i costi di gestione e manutenzione, i costi di riparazione, ecc... Per facilitare tale scelta si può far riferimento ad opportuni criteri reperibili nella letteratura tecnica, si veda ad esempio Bouteille e Belforte [1987], dai quali si deduce che, nonostante la crescente importanza assunta dall'elettronica nelle macchine automatiche, il comando pneumatico risulta conveniente in certi campi applicativi, tipicamente quelli a basso tasso di complessità e con configurazioni di macchina non troppo estese; inoltre, il comando pneumatico è una scelta d'obbligo per le applicazioni in ambiente esplosivo (industrie chimiche e petrolchimiche).

La progettazione dei circuiti di comando è materia abbastanza complessa: le tipologie di problemi posti dagli automatismi industriali sono le più svariate e per un singolo problema spesso le soluzioni sono molteplici. Sarebbe molto vantaggioso automatizzare le procedure di progettazione dei sistemi di comando.

In questo lavoro si presenta, appunto, una procedura automatica di progettazione dei circuiti di comando pneumatici, basata sulla ricerca delle equazioni logiche che realizzano la sequenza richiesta minimizzando la complessità di configurazione del circuito. L'intendimento del lavoro è la costruzione di

una base per lo sviluppo di procedimenti più completi adatti ad essere inseriti nei programmi CAD per oleostatica e pneumatica, nei quali già è presente una parte riguardante la progettazione delle sezioni di potenza.

In questo modo si otterrebbe uno strumento completo di progettazione che, partendo dalla definizione del ciclo di lavoro, fornirebbe automaticamente lo schema del circuito di comando; parallelamente opportune procedure di calcolo provvederebbero al dimensionamento della sezione di potenza; il database del programma CAD fornirebbe infine la lista dei materiali.

2. I CIRCUITI PNEUMATICI DI COMANDO

In un sistema pneumatico, se l'azione voluta è immediata conseguenza della serie di condizioni prestabilite per il corretto funzionamento, si parla di *logica combinatoria* e *combinatori* sono definiti i circuiti che la realizzano; in essi le uscite sono in ogni istante funzione degli ingressi [Falzone 1970].

Quando invece il diagramma operativo della macchina prevede una serie di movimenti da effettuarsi in ordine prestabilito, si parla di circuiti sequenziali. In essi le uscite sono funzione della combinazione presente dei segnali di input e delle combinazioni passate; gli elementi che ricordano la storia delle combinazioni passate sono detti *memorie*, in genere identificabili con valvole bistabili.

In base ai principi costruttivi e di funzionamento si può operare la seguente classificazione dei comandi sequenziali:

- comandi sequenziali a tempo
- comandi sequenziali a camme
- comandi sequenziali con controllo di posizione.

I primi due sono comandi del tipo ad anello aperto in cui lo svolgimento del ciclo dipende, nel primo caso, dalla scansione temporale imposta dagli elementi temporizzatori, nel secondo, dalla sequenza generata dalle camme. Un imprevisto qualsiasi, che impedisca il compiersi di un'azione del ciclo, può falsare tutta la sequenza e nessun dispositivo è in grado di accorgersene. Questi circuiti, perciò, anche se di grande semplicità ed economia, sono piuttosto rigidi e non sempre possono essere impiegati, specialmente laddove sono richiesti alti livelli di automazione.

La terza categoria di comandi è di tipo ad anello chiuso ed ogni fase viene distinta e riconosciuta dal sistema secondo due modalità:

- utilizzando memorie e segnalatori di posizione (finecorsa); questa soluzione elimina la possibilità di corse aleatorie. L'aspetto negativo di questa modalità è rappresentato dall'elevato numero di componenti necessari (memorie e sensori di finecorsa);
- utilizzando solo sensori di finecorsa che con il loro segnale confermano la fine di un'azione; l'azione successiva inizia solo dopo il completamento della precedente. Qualora essa non si avvii l'intero comando rimane bloccato.

Quest'ultimo tipo di comando è senza dubbio il più economico ma non è sempre applicabile, specialmente nei casi di sequenze con segnali interferenti, noti anche col nome di segnali bloccanti.

Può accadere, infatti, che il segnale che comanda la fuoriuscita di un attuatore rimanga presente sul relativo distributore anche se la sequenza prevede l'azione del segnale opposto che determina il rientro; in questo caso i due segnali, agendo sui lati opposti del distributore, impediscono la commutazione e quindi provocano l'arresto del ciclo. Il primo dei due segnali è chiamato perciò segnale bloccante.

L'eliminazione dei segnali bloccanti è necessaria, quindi, per il corretto svolgimento della sequenza e costituisce uno dei momenti più importanti della progettazione dei circuiti di comando.

Un approccio non empirico al problema dei sistemi di comando per le applicazioni pneumatiche richiede l'uso dell'algebra booleana, in quanto i segnali in gioco sono di tipo binario [Belforte 1987, Colombo 1987, Romiti e Belforte 1975]. La combinazione di questi segnali attraverso opportune funzioni logiche elementari porta alla definizione delle condizioni che devono essere verificate affinché si possa realizzare un determinato evento, che in pneumatica si identifica con un'azione (avanzamento di un cilindro per esempio). L'insieme delle equazioni logiche così ottenute costituisce il circuito logico di comando, che, tradotto impiegando gli elementi pneumatici (valvole, memorie ecc.), diventa il circuito di comando pneumatico.

Per comprendere meglio quanto verrà esposto in seguito sono necessarie alcune definizioni.

La *sequenza* di funzionamento di una macchina automatica è costituita dai movimenti degli attuatori necessari a svolgere una prestabilita successione di azioni.

Ai fini della progettazione, la sequenza deve essere suddivisa in *passi o fasi*, cui sono associate le azioni, ed in *transizioni*, cui sono associate delle condizioni.

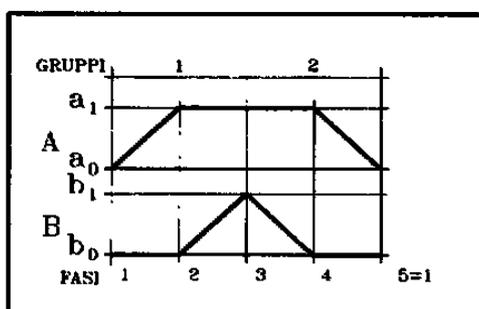


Fig. 1: Diagramma corsa-passo di una sequenza con segnale bloccante.

- a_0, b_0 segnali dei finecorsa dei cilindri A e B posti in posizione retratta;
 a_1, b_1 segnali dei finecorsa dei cilindri A e B posti in posizione estratta.
 a_1 segnale bloccante.

Il *passo* caratterizza un comportamento invariabile del sistema di comando e resta attivo finché non si siano verificate le nuove condizioni di transizione.

La *transizione* indica la possibilità di passaggio da un passo a quello successivo e la condizione di transizione (detta anche *ricettività*) è un'equazione logica i cui fattori sono segnali di pressione generati dai sensori.

I passi sono indicati con il simbolo letterale del cilindro che deve essere mosso seguito da un segno + o da un segno - a seconda che si intenda la corsa di uscita o di rientro del pistone.

La sequenza può essere rappresentata graficamente tramite i diagrammi corsa-passo (fig.1), i quali mostrano, a meno della grandezza tempo, la successione delle varie fasi.

Un altro interessante metodo di rappresentazione della sequenza è offerto dal linguaggio simbolico-grafico GRAFCET [Belforte, 1987; Bouteille e Belforte, 1987]

3. IL PROBLEMA DELLA PROGETTAZIONE DEI CIRCUITI DI COMANDO

Da quanto visto si capisce come la progettazione dei circuiti di comando richieda non solo il corretto svolgimento della sequenza, ed a tal fine fondamentale è lo studio atto ad eliminare i segnali bloccanti, ma anche l'economicità della soluzione circuitale. È il soddisfacimento di questi due aspetti a rendere il problema della progettazione assai delicato.

In definitiva il problema generale posto dalla progettazione dei circuiti di comando si può sintetizzare in questo modo: *ricercare le equazioni di ricettività, e di conseguenza la configurazione del circuito, che consentano la realizzazione corretta della sequenza data, cioè senza arresti o corse doppie, utilizzando il minor numero di dispositivi e collegamenti.*

Vediamo, allora in estrema sintesi, come viene fatta attualmente la progettazione, quali sono le principali difficoltà ed i metodi per superarle; al punto successivo vedremo, invece, come potrebbe svolgersi la progettazione utilizzando una procedura nuova, basata sull'impiego delle equazioni logiche.

4. LA PROGETTAZIONE DEI CIRCUITI SECONDO I METODI TRADIZIONALI

I dati del problema sono costituiti dalla sequenza delle fasi e dallo stato iniziale dei sensori di finecorsa.

Il primo passo verso la risoluzione è la rappresentazione delle sequenze: ciò può essere fatto tramite il diagramma corsa-passo, il quale offre la possibilità di individuare i segnali bloccanti come quei segnali che restano attivi durante la doppia corsa del cilindro che hanno comandato (segnale a_1 di fig. 1).

L'eliminazione dei segnali bloccanti può essere effettuata secondo varie tecniche [Belladonna e Mombelli 1982] nelle quali si tende a diminuire la durata del segnale in modo che esso agisca solo quando serve; tali tecniche sono note come:

- tecnica dei collegamenti;
- utilizzo dei sensori unidirezionali.

Un altro gruppo di metodi prevede l'utilizzo di memorie le quali, commutando in opportune fasi della sequenza, generano dei segnali in grado di diversificare gli stati della sequenza stessa. In tal modo, stati identici, cioè caratterizzati dalla stessa configurazione degli ingressi, possono distinguersi attraverso lo stato delle memorie.

Le modalità con cui queste memorie vengono inserite in un circuito definiscono i seguenti metodi:

- metodo delle memorie in cascata
- metodo del contatore binario
- metodo delle memorie "passo-passo"
- sequenziatori modulari e microsequenziatori.

Il metodo più usato, quando è possibile la sua applicazione, è la tecnica dei collegamenti [Belladonna e Mombelli 1982], che consente di eliminare i segnali bloccanti senza l'inserimento di memorie. Essa consiste nel collegare l'alimentazione del finecorsa che genera il segnale bloccante in serie con il segnale di un altro finecorsa che:

- sia presente quando inizia il segnale bloccante,
- sia assente quando deve iniziare la corsa opposta del cilindro il cui movimento è stato avviato dal segnale bloccante,
- non si ripresenti fintantoché il segnale bloccante rimane presente.

L'individuazione del segnale sbloccante non è sempre facile, anche utilizzando il diagramma corsa-passo, specialmente per le sequenze con molti cilindri e con più segnali bloccanti.

Quando non è possibile ricorrere a questa tecnica si passa, allora, ad una delle altre.

Esaminiamo, a scopo esemplificativo, solo due dei precedenti metodi: il metodo delle memorie in cascata, usato specialmente in passato, ed il metodo delle memorie passo-passo.

4.1 METODO DELLE MEMORIE IN CASCATA

Si deve procedere alla suddivisione della sequenza in gruppi per annullare i segnali bloccanti.

Se si suddividesse la sequenza in tanti gruppi quante sono le fasi, in ogni gruppo non sarebbe mai presente un segnale bloccante; in questo modo però sarebbero necessarie troppe memorie e la soluzione non sarebbe più ottimale.

Nella progettazione corrente si preferisce allora seguire una via più pratica: individuare i gruppi in maniera che in ognuno di essi sia presente un solo comando per cilindro. Ad esempio la sequenza:

$$A+ / B+ / A \rightarrow C+ / B \rightarrow C- / T / C+ / C-$$

può essere scomposta nei gruppi seguenti:

$$A+ / B+ ; \quad A \rightarrow C+ ; \quad B \rightarrow C- / T ; \quad C+ ; \quad C-$$

È evidente che questo criterio può non essere quello ottimale al fine di ridurre al minimo il numero dei componenti: infatti la sequenza precedente potrebbe essere suddivisa più convenientemente in tre gruppi, nessuno dei quali contiene segnali bloccanti:

$$A+ / B+ / A \rightarrow C+ ; \quad B \rightarrow C- / T / C+ ; \quad C-$$

In ognuno dei gruppi così individuati agisce un segnale esclusivo generato da un "blocco memorie", come quello di fig. 2, costituito da tre valvole di tipo bistabile; se con P_i si intendono i segnali di pilotaggio e con U_i le uscite verso i distributori, è immediato capirne il funzionamento:

- quando arriva il segnale di pilotaggio P_1 , il segnale in uscita è U_1 che comanderà il relativo distributore;
- quando arriva il segnale P_2 , il distributore in alto commuta e l'uscita diventa U_2 , annullando il segnale U_1 ; nel frattempo il pilotaggio P_1 non è più attivo, ma la valvola mantiene la posizione (valvole bistabili);
- analogamente per il segnale P_3 , il quale commuta il distributore su cui agisce e "resetta" la memoria precedente (quella relativa al segnale P_2); il pilotaggio P_1 è inattivo, ma la posizione viene mantenuta;
- il pilotaggio P_4 commuta la valvola, genera U_4 e "resetta" la memoria precedente.

Con questa tecnica, ad ogni pilotaggio corrisponde una sola uscita dato che quella precedente viene sempre annullata. Ne consegue che i segnali generati dalla cascata non possono essere bloccanti in quanto è presente solo l'uscita che deve pilotare il distributore principale interessato, mentre tutte le altre, e quindi i pilotaggi degli altri distributori principali che potrebbero essere in conflitto, sono assenti.

Si osserva che, se il numero dei gruppi in cui si è suddivisa la sequenza è n , sono necessarie $n-1$ memorie.

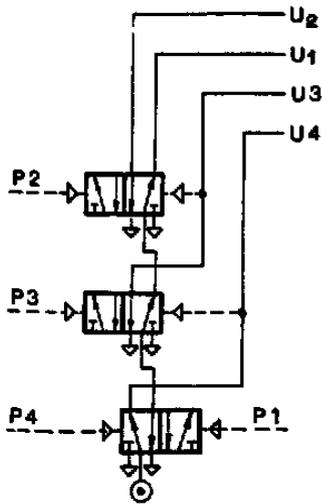


Fig. 2: Blocco di memorie in cascata.

L'inserimento del circuito in cascata in quello di comando va fatto in modo che il finecorsa in testa ad ogni gruppo, anziché pilotare il distributore principale interessato, piloti sempre i distributori ausiliari collegati in cascata (nella figura il pilotaggio P1); l'uscita corrispondente piloterà poi il distributore principale.

Gli altri finecorsa, cioè quelli successivi al primo, pilotano direttamente i distributori principali.

Per evitare che variazioni contemporanee di segnali creino malfunzionamenti, i segnali di comando della cascata si considerano avvenire secondo fasi supplementari separate dalle altre fasi della sequenza.

4.2 METODO DELLE MEMORIE "PASSO-PASSO"

Anche in questo caso è prevista la divisione in gruppi della sequenza principale e ad ogni gruppo resta associata una singola memoria. Ad esempio la sequenza di figura 1:

A+/B+/B-/A-

avente il segnale a₁ bloccante, può essere suddivisa in due gruppi:

A+/B+ ; B-/A-

Le equazioni logiche sono le seguenti (con m si intendono i segnali in uscita dalle memorie e con M i segnali in ingresso che commutano le stesse):

gruppo 1	fase 1	$m_1 = A+ * NOT (M2)$	--->	stato : $m_1 a_0 b_0$
	fase 2	$m_1 * a_1 = B+$	--->	stato : $m_1 a_1 b_0$
	fase supplementare	$m_1 * b_1 = M2$	--->	stato : $m_1 a_1 b_1$
gruppo 2	fase 3	$m_2 = B- * NOT (M1)$	--->	stato : $m_1 m_2 a_1 b_1$
	fase 4	$m_2 * b_0 = A-$	--->	stato : $m_2 a_1 b_0$
	fase supplementare	$m_2 * a_0 = M1$	--->	stato : $m_2 a_0 b_0$

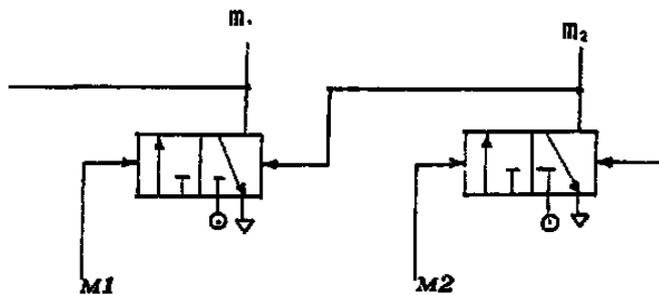


Fig. 3: Disposizione delle memorie

Il segnale m₁ emesso dalla memoria 1 attiva la corsa positiva del cilindro A ed azzerla la memoria 2 (fase 1); quando il pistone del cilindro A arriva in posizione "tutto fuori" attiva il segnale a₁, che, assieme ad m₁, attiva la corsa in uscita di B. Quando il pistone di B arriva alla fine della sua corsa attiva il segnale b₁, il quale non può attivare direttamente la corsa negativa di B, perché sul distributore principale agisce ancora il segnale m₁*a₁. Esso va ad azionare, invece, la seconda memoria la quale azzerla la memoria precedente, cosicché il segnale m₁*a₁ ora

non è più attivo ed il segnale m_2 può così attivare liberamente la corsa negativa di B.

La stessa logica è alla base del funzionamento dei sequenzatori modulari, i quali sono costituiti da tante unità simili a quella appena descritta che possono essere facilmente collegabili a seconda della sequenza desiderata.

A questo punto è possibile ricavare le equazioni di ricettività per ogni fase e procedere alla stesura dello schema logico e pneumatico del circuito di comando.

Si dovrà studiare, infine, l'inserzione di circuiti combinatori per le operazioni di marcia ed arresto, per le sicurezze e l'azzeramento del circuito.

5. METODO PROPOSTO

Se in una transizione si includono tutti i segnali attivi, si osserva che le disfunzioni (p. es. segnali bloccanti) si presentano ogni volta che fasi diverse vengono attivate da transizioni uguali. Il sistema non è in grado di distinguere gli stati uguali ed evolverà verso altri stati distinti realizzando più fasi contemporaneamente. Per evitare ciò basterà diversificare, con il segnale di una memoria, le combinazioni di segnali attivi uguali tra loro.

Più precisamente: si divide la sequenza in sottosequenze o gruppi che non includano due transizioni con tutti i segnali attivi uguali e ad ognuno di tali gruppi si associa il segnale di una memoria.

Si devono ora ricercare, all'interno di ogni gruppo, le corrette equazioni di ricettività. In altre parole si pone il seguente problema: tra tutti i segnali presenti in una transizione quale loro combinazione prendere come equazione di ricettività affinché la fase che deve seguire non venga avviata anche in altri punti del gruppo? Ed ancora, quale combinazione se si vuole minimizzare il numero dei sensori?

La risposta a questo problema può essere espressa in termini puramente matematici: per evitare che la combinazione logica dei segnali presenti nella transizione a non si verifichi anche nella transizione b bisognerà che la prima contenga almeno un segnale non presente nella seconda; in altri termini, la somma logica costruita con i segnali di a non presenti in b deve entrare a far parte dell'equazione di ricettività. Se si vuole che la transizione a non si ripeta, le somme logiche così ottenute in corrispondenza di tutte le altre transizioni dovranno essere tra loro collegate secondo un prodotto logico (AND) per formare un'unica equazione logica che è l'equazione di ricettività cercata.

Infine, se si vuole minimizzare il numero dei sensori, basterà semplificare quest'ultima equazione tramite le regole dell'algebra logica.

Si esprimerà ora in termini matematici quanto sopra illustrato.

1) Suddivisione in gruppi

La prima fase è la ricerca dei prodotti uguali dei segnali attivi dei fincorsa in ogni transizione.

Chiamando con $j_{i,k}$ il segnale del fincorsa relativo al cilindro j , di posizione i ($i=0$ posizione retratta, $i=1$ posizione estratta), nella transizione k , si definiscono i seguenti prodotti logici.

$$P_k = \prod_{j=1}^N j_{i,k} \quad \text{con } j_{i,k} = 1 \text{ ed } N = \text{numero degli attuatori e altri dispositivi comandati}$$

Transizioni i cui prodotti P siano uguali devono essere incluse in gruppi diversi.

2) Determinazione equazioni di ricettività

Dopo la definizione dei gruppi, si passa alla ricerca delle equazioni di ricettività per ogni transizione del gruppo. Come detto in precedenza, se si vuole che la transizione r non attivi anche la fase relativa alla transizione s , basterà scegliere tra tutti i segnali attivi in r quelli che non lo saranno più in s , cioè:

$$j_{i,r} = 1 \rightarrow j_{i,s} = 0$$

Non è necessario considerarli tutti, uno solo è sufficiente, perciò l'equazione di ricettività deve essere una somma logica OR:

$$R_r = \sum_{j=1}^N j_{i,s}$$

Se si vuole, ora, che la transizione della fase r non provochi movimenti anche nelle altre fasi, basterà considerare le somme logiche simili alla precedente, calcolate in corrispondenza delle altre transizioni, ed unirle secondo un prodotto logico AND. In definitiva l'equazione logica cercata ha la seguente forma:

$$R_r = \left(\prod_{s=1}^F \sum_{j, s \neq r} j_{i,s} \right) \quad \text{con } F \text{ numero delle fasi del gruppo}$$

È sufficiente ora semplificare l'equazione precedente secondo le regole dell'algebra logica per ottenere l'equazione di ricettività per la transizione alla fase r che, per quanto detto, ha la caratteristica di essere univoca.

Vale la pena, ora, di fare alcune osservazioni sulle conseguenze di quanto esposto.

La ricerca dei segnali adatti per l'annullamento dei segnali bloccanti, che, come visto nella tecnica dei collegamenti, può essere anche faticosa, ora non è più necessaria, perché tutta questa operazione è compresa nella scrittura delle equazioni di ricettività, che come vedremo è assai semplice.

Inoltre, dette equazioni, in virtù del modo con cui vengono ricavate, sono univoche ed attive in una sola transizione: perciò non possono generare segnali bloccanti, né corse contemporanee di più cilindri.

6. IL PROGRAMMA DI PROGETTAZIONE AUTOMATICA

In base alla procedura esposta al capitolo precedente è stato stilato un programma per personal computer che consente la risoluzione automatica dei circuiti di comando sequenziali.

Il programma è suddiviso in tre blocchi.

Il primo blocco permette l'iscrizione dei dati di ingresso relativi agli attuatori, alla sequenza da realizzare ed allo stato iniziale dei sensori di finecorsa.

Il secondo blocco esegue il calcolo del circuito la cui sequenza è stata definita nel primo blocco. Per dare maggior flessibilità alla risoluzione del circuito sono state concepite tre modalità di calcolo:

- modalità automatica di minimo: il programma esegue i calcoli secondo la procedura descritta al punto 5, pertanto la configurazione circuitale prodotta sarà minima;
- modalità automatica di massimo: il programma cerca la soluzione che divide la sequenza in tanti gruppi quante sono le fasi e per ogni fase inserisce il segnale di una memoria;
- modalità interattiva: tramite il dialogo con l'operatore, la procedura permette la ricerca di soluzioni intermedie, per esempio con un maggior numero di memorie o di sensori.

Il terzo blocco è destinato alla rappresentazione grafica ed alla simulazione di funzionamento. Partendo dalle equazioni di ricettività ricavate nel secondo blocco il programma è in grado di disegnare a video il circuito con la sezione di potenza, quella di comando e il blocco memorie. Inoltre i dati forniti nel primo blocco consentono di tracciare il diagramma corsa-passo. Un'ultima opzione, infine, consente l'animazione del circuito. In tal modo, l'operatore è in grado di verificare la rispondenza della sequenza realizzata dal programma con quella impostata.

7. APPLICAZIONI

Si consideri la sequenza ciclica di fig.4 che presenta due fasi con movimenti contemporanei ed i segnali bloccanti a_0 nei confronti del cilindro B e b_0 nei confronti del cilindro A.

La sequenza non presenta transizioni uguali; pertanto non si procede ad alcuna suddivisione in gruppi.

Il calcolo delle equazioni di ricettività parte dalla transizione alla fase 1.

I segnali attivi disponibili sono a_0 , b_0 , e c_0 ; nella seconda transizione, di questi è inattivo solo il primo, a_0 ; nella terza sono inattivi a_0 e b_0 ; nella quarta a_0 e c_0 .

Allora l'equazione di ricettività che avvia la fase 1 ma non le fasi 2,3,4, è la seguente:

$$R_1 = a_0 * (a_0 + b_0) * (a_0 + c_0)$$

che semplificata fornisce: $R_1 = a_0$

Perciò l'equazione di ricettività per la prima transizione è la seguente:

$$a_0 = A +$$

- Similmente per la seconda transizione:
- segnali disponibili: $a_1, b_0, e c_0$
 - segnali inattivi nelle altre transizioni: a_1 nella prima, b_0 nella terza e c_0 nella quarta.
 - equazione di ricettività : $R_2 = a_1 * b_0 * c_0$

Procedendo allo stesso modo si ricavano le seguenti equazioni di ricettività:

- fase 1 $a_0 = A+$
- fase 2 $a_1 * b_0 * c_0 = B+$
- fase 3 $b_1 = B- * C+$
- fase 4 $c_1 = A- * C-$

Da queste equazioni è immediato ricavare lo schema del circuito (figura 5).

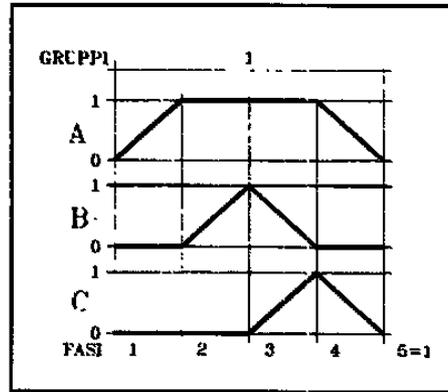


Fig. 4 : Diagramma corsa-passe del primo esempio

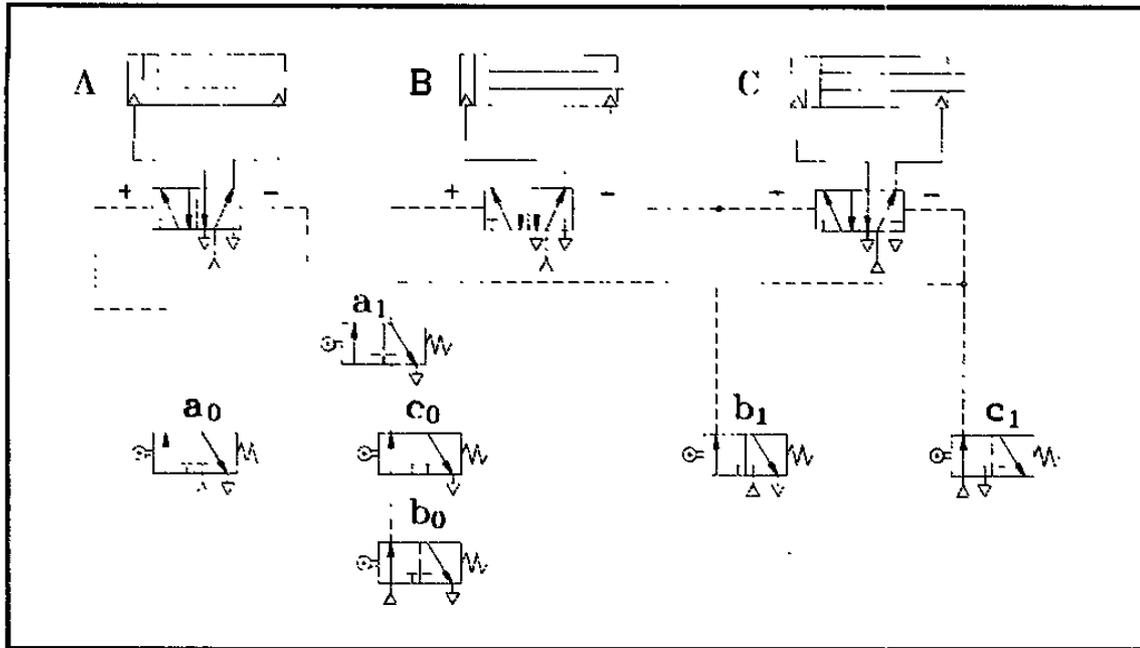


Fig.5: Schema semplificato del circuito del primo esempio.

Si noti la estrema semplicità del procedimento che richiede solo la conoscenza delle elementari regole dell'algebra logica.

Come secondo esempio si consideri la sequenza di figura 6, realizzata da quattro attuatori in 7 fasi. Dato che le transizioni uguali sono tre (la prima, la sesta e l'ottava.), il programma, ricorrendo alla risoluzione automatica, spezza la sequenza in tre gruppi :

- $A+, B+, A- * C+, C-$; $B-, D+$; $D-$

formando le equazioni di ricettività seguenti ed il circuito di figura 7.

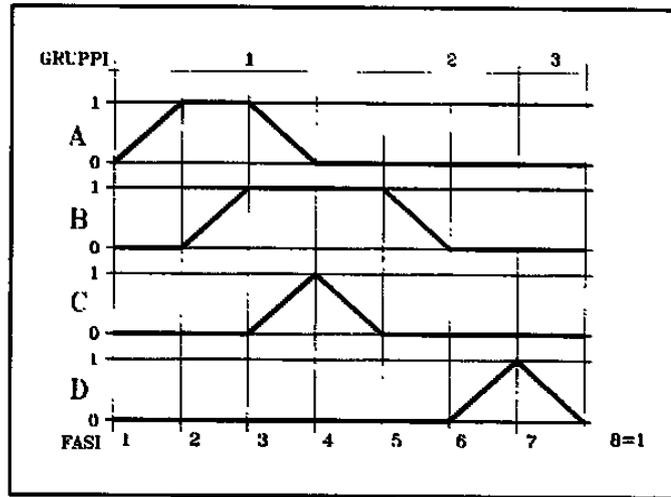


Fig.6: Diagramma corsa-passo relativo al secondo esempio.

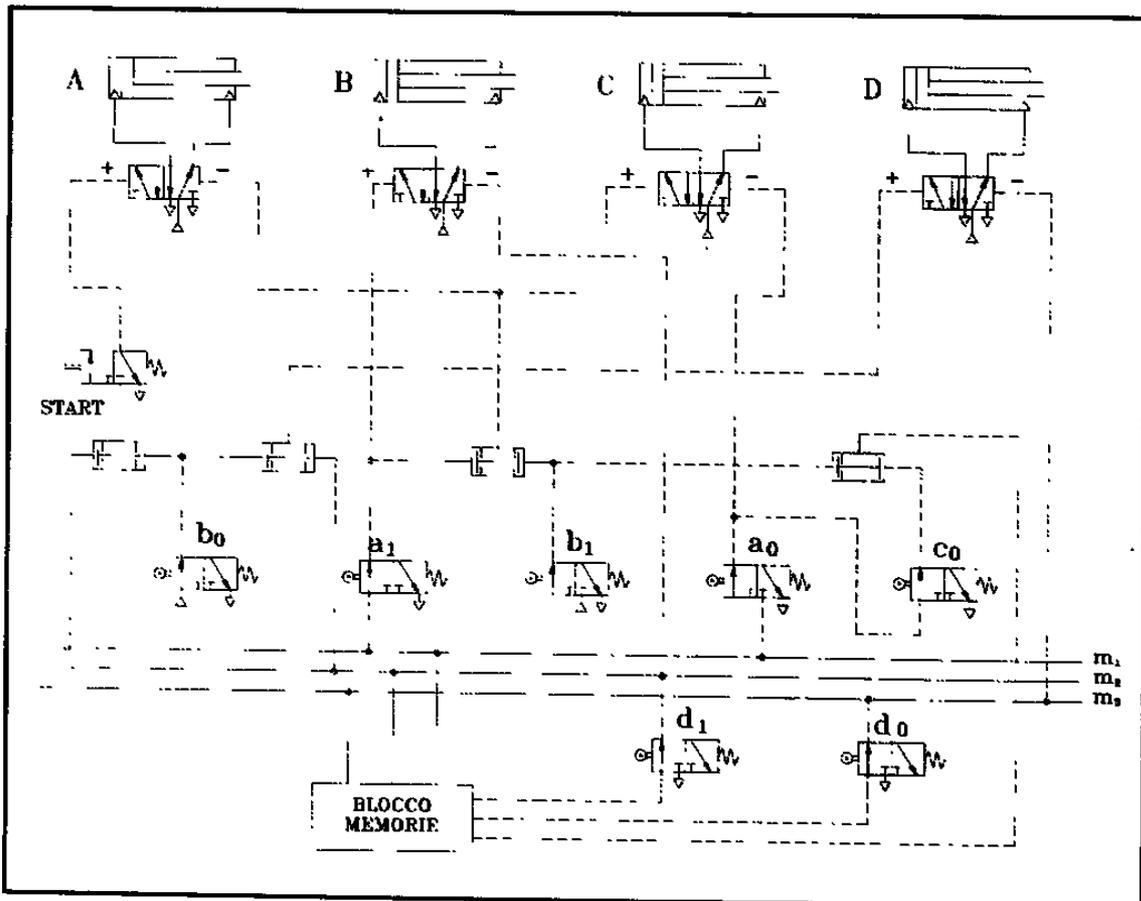


Fig.7: Schema del circuito relativo al secondo esempio. (N.B. le posizioni dei finecorsa e dei pistoni non sono congruenti con nessuna delle fasi della sequenza)

gruppo 1

fase 1	$m_1 * b_0$	=	$A+ * NOT (M_3)$
fase 2	$m_1 * a_1$	=	$B+$
fase 3	$m_1 * a_1 * b_1$	=	$A- * C+$
fase 4	$m_1 * a_0$	=	$C-$
fase supplementare	$m_1 * b_1 * a_0 * c_0$	=	M_2

gruppo 2

fase 5	m_2	=	$B- * NOT (M_1)$
fase 6	$m_2 * b_0$	=	$D+$
fase supplementare	$m_1 * d_1$	=	M_3

gruppo 3

fase 7	m_3	=	$D- * NOT (M_2)$
fase supplementare	$m_3 * d_0$	=	M_1

Si riportano, infine, le equazioni di ricettività di un terzo esempio la cui sequenza è rappresentata in figura 8.

gruppo 1

fase 1	$m_1 * b_0$	=	$A+ * NOT (M_2)$
fase 2	$m_1 * a_1$	=	$B+ * C+$
fase 3	$m_1 * b_1$	=	$A-$
fase supplementare	$m_1 * b_1 * a_0$	=	M_2

gruppo 2

fase 4	$m_2 * b_1$	=	$A+ * NOT (M_1)$
fase 5	$m_2 * a_1$	=	$C-$
fase 6	$m_2 * c_0$	=	$B-$
fase 7	$m_2 * b_0$	=	$A-$
fase supplementare	$m_2 * b_0 * a_0$	=	M_1

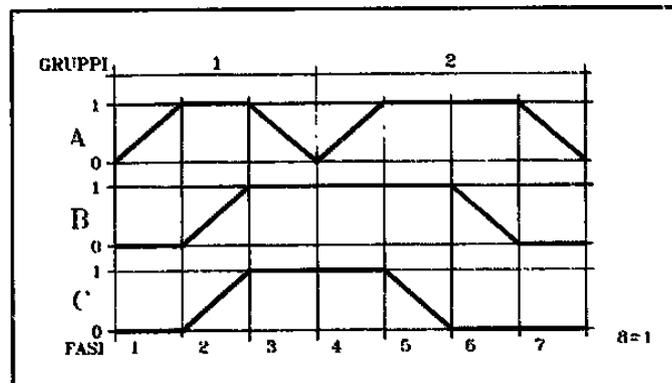


Fig. 8: Diagramma corsa-passo relativo al terzo esempio.

CONCLUSIONI

In questo lavoro è stato presentato un metodo matematico di risoluzione dei circuiti pneumatici di comando.

Attualmente la progettazione dei circuiti viene fatta in base al criterio generale della transizione alla fase successiva tramite il consenso del sensore attivato dall'attuatore relativo alla fase precedente. Tale criterio comporta il vantaggio di un utilizzo minimo di dispositivi nel circuito, ma crea il problema dei segnali bloccanti, che viene risolto con varie tecniche: tecnica dei collegamenti, introduzione di memorie ecc... In ogni caso il progettista è costretto ad una serie di ragionamenti e di scelte che aumentano con la complessità della sequenza.

Il metodo proposto, invece, è completamente automatico: infatti esso si basa su una procedura matematica che agisce all'interno di gruppi di fasi con transizioni diverse. Il procedimento mostra tutta la sua potenza anche nell'applicazione manuale: partendo dal diagramma corsa-passo, con pochi passaggi logico-matematici e senza ragionamenti complessi, che spesso sono fonte di errore, si perviene alle equazioni di ricettività cercate.

Nella seconda parte del lavoro si sono studiati alcuni circuiti utilizzando un programma al calcolatore che concretizza la procedura proposta.

Il lavoro è stato svolto con il contributo del M.U.R.S.T. (Ministero dell'Università e della Ricerca Scientifica e Tecnologica).

BIBLIOGRAFIA

- BELFORTE G.: *Pneumatica. Tecniche nuove*, Milano, 1987
- BELLADONNA U, MOMBELLI A.: *Tecniche circuitali pneumatiche*. Ed. U. Hoepli, Milano, 1991
- BOUTEILLE D., BELFORTE G.: *Automazione flessibile elettropneumatica e pneumatica. Tecniche Nuove*, Milano, 1987
- COLOMBO G.: *Automazione industriale. VOL. I,II,III*. Ed. Dott. Giorgio, Torino, 1987
- FALZONE V.: *Il progetto logico dei circuiti digitali*. Edizioni Scientifiche Siderea, Roma, 1970.
- MARIE G.: *L'applicazione dei controllori programmabili industriali*. Tecniche nuove, Milano, 1987
- PIGNATELLI R.: *Progettazione automatica di sistemi di comando per macchine oleopneumatiche..Tesi di Laurea - Relatore prof. G. Zamboni - Dipartimento di Ingegneria Meccanica - Università di Padova*, 1988
- ROMITI A., BELFORTE G.: *Automazione a fluido. vol. I Fluidodinamica. Controlli; Vol. II Componenti. Esercizio dei circuiti*. Patron editore, Bologna, 1975.

ABSTRACT

A mathematical procedure for solving pneumatic control circuits is presented. The base concept is the univocity of each transition condition, so that only one phase at one time can be activated, avoiding multiple strokes of the pistons. To fulfill this request, the signals active in a transition and inactive in the other ones must be used; a logical combination of them produces a univocal transition equation. The procedure breaks the sequence in two or more groups containing unequal transition conditions; for each of these groups the previous logical procedure has to be applied. To minimize the devices in the circuit, a logical simplification of all the transition equations can be made; the set of logical equations produced in this way can easily be transformed in a scheme of the circuit.

The method is simple and can be automatically applied without mental efforts, which could cause some errors.

The procedure has been translated into a computer program, which has been used to study the examples shown in the paper.