

Università degli Studi di Padova

Facoltà di Ingegneria



Metodi di Implementazione del Controllo Predittivo

Relatore: Prof. Mauro Bisiacco

Laureando: Alberto Gobbo

Corso di laurea triennale in Ingegneria dell'Automazione

26 Marzo 2010

Anno Accademico 2009/2010

Indice

1 Introduzione	5
1.1 Motivazione	5
2 Caratteristiche principali dei controllori predittivi	6
2.1 Strategia di controllo	6
2.2 Modelli di previsione	6
2.2.1 Modello del processo	6
2.2.2 Modello del disturbo	9
2.3 Funzione obiettivo	10
2.4 Legge di controllo	10
3 Principali MPC commerciali	12
3.1 Dynamic Matrix Control	12
3.2 Model Algorithmic Control	15
4 Generalized Predictive Control	18
4.1 Modello del processo e del disturbo	18
4.2 Funzione obiettivo	19
4.3 Formulazione dell'algoritmo	19
4.4 Disturbi misurabili	22
4.6 Presenza di vincoli nel GPC	24
4.6.1 Vincoli di uguaglianza	24
4.6.2 Vincoli di disuguaglianza	26
5 Progettazione e simulazione di un controllore GPC	27
5.1 Descrizione dell'impianto	27
5.2 Descrizione del modello	28
5.3 Discretizzazione del modello	29
5.4 Implementazione del controllore GPC.	30
5.5 Simulazioni	34
5.5.1 Simulazioni con tempo di campionamento $T_c = 60$ [sec].	34
5.5.2 Simulazioni con tempo di campionamento $T_c = 20$ [sec].	35
6 Bibliografia	37

1 Introduzione

1.1 Motivazione

Il controllo predittivo rappresenta una delle più avanzate tecniche di controllo digitale in campo industriale. Esso viene denominato con la sigla **MBPC** (*Model based predictive control*) o **MPC** e non rappresenta una specifica legge di controllo, ma bensì una classe di leggi di controllo. Nati alla fine degli anni settanta in ambito industriale (Industria petrolchimica), hanno avuto poi una considerevole evoluzione fino ai nostri giorni, anche se inizialmente questo metodo non ha avuto l'interesse della comunità accademica, ma solo uno sviluppo in campo industriale.

La sigla MPC non delinea una specifica legge di controllo, bensì un'ampia gamma di metodi di controllo che fanno un uso esplicito di un modello del sistema per ottenere il segnale di controllo con l'uso di una funzione obiettivo che lo minimizza. Le caratteristiche generali che si possono riscontrare nelle varie famiglie di controllori predittivi sono:

- L'uso esplicito di un modello del sistema da controllare, per prevedere l'uscita agli istanti di tempo futuri (orizzonte degli eventi);
- La possibilità di introdurre dei vincoli nel calcolo dei segnali di controllo dovuti all'uso di attuatori;
- Calcolare una sequenza di segnali di controllo attraverso l'uso di una funzione obiettivo;
- In ogni istante di campionamento, l'orizzonte si sposta verso il futuro, comportando l'applicazione del primo segnale di controllo della sequenza calcolato ad ogni passo.

I principali vantaggi nell'uso dei controllori MPC rispetto ad altri metodi sono:

- Facilità nella gestione di **sistemi-multivariabili**;
- Facilità nella gestione di **dinamiche-complesse** (risposta inversa, ritardi, rampe, ecc...);
- Possibilità di applicarli a schemi **“feedforward”** (disturbi misurabili);
- Facilità nella gestione di eventuali **vincoli** sulle variabili controllate e su quelle manipolate (portando l'impianto verso i suoi limiti senza saturare le sue variabili).

2 Caratteristiche principali dei controllori predittivi

2.1 Strategia di controllo

La strategia di controllo, generalmente adottata nelle varie implementazioni di controllori MPC si può descrivere attraverso queste fasi:

1. Attraverso l'ausilio del **modello del sistema** da controllare, vengono determinate le uscite future per ogni istante t fino ad N , chiamato orizzonte di predizione. Queste uscite predette $y(t+k|t)$ con $k = 1 \dots N$ dipendono dalle uscite e dagli ingressi passati (precedenti all'istante t) e dagli ingressi di controllo futuri $u(t+k|t)$ con $k = 0 \dots N-1$.
2. Gli ingressi di controllo futuri vengono calcolati attraverso un criterio di ottimizzazione in modo tale da mantenere l'uscita del sistema più vicino possibile alla traiettoria di riferimento $w(t+k)$. Un criterio comunemente usato è una funzione quadratica degli errori calcolati tra le uscite future e le traiettorie di riferimento future.
3. Solo il segnale di controllo $u(t|t)$ viene inviato all'impianto, mentre i successivi segnali calcolati vengono eliminati. Ciò in quanto al nuovo istante di campionamento si ottiene l'uscita dal sistema $y(t+1)$ che verrà utilizzata per effettuare nuovamente i calcoli descritti al passo 1. Dove i segnali di comando futuri saranno $u(t+k+1|t+1)$ con $k = 0 \dots N-1$.

2.2 Modelli di previsione

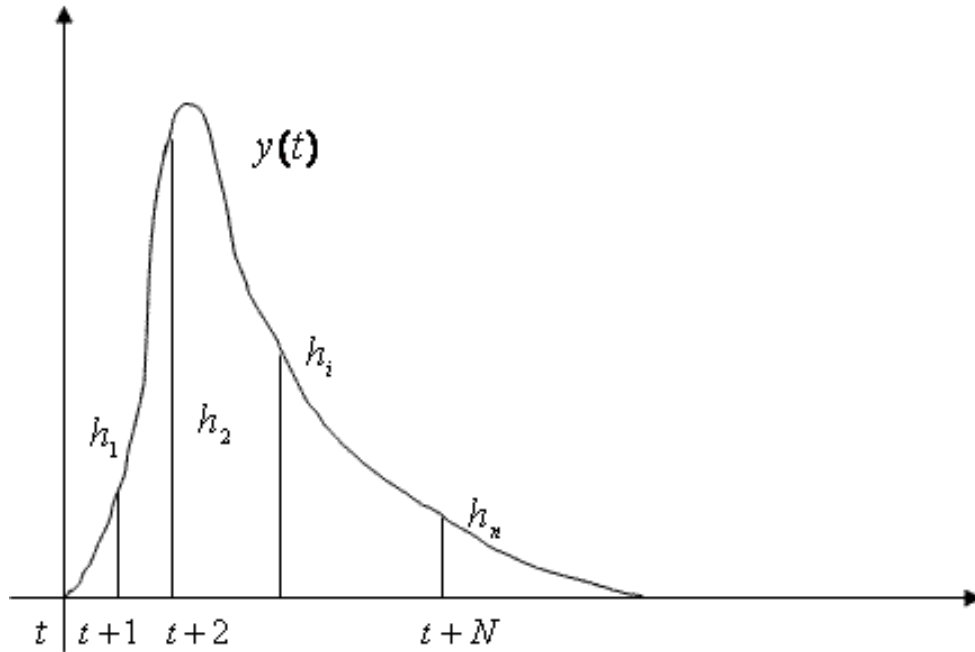
Come si può notare, il modello del sistema, gioca un ruolo fondamentale nei controllori MPC. Più accurato è il modello rispetto al sistema reale più la previsione delle uscite future sarà precisa. Le diverse strategie implementate nei controllori MPC possono adoperare vari modelli per rappresentare le relazioni tra gli ingressi e le uscite misurabili. Alcune vengono trattate come variabili manipolabili e altre come misurazioni dei disturbi da compensare come in una retroazione classica. Il modello del disturbo, viene preso in considerazione, al fine di descrivere alcuni comportamenti che non si riflettono nel modello del sistema, come l'errore e il rumore presenti negli ingressi non misurabili. Si capisce quindi che entrambi i modelli sono necessari nella progettazione, per ottenere dal MPC una previsione precisa.

2.2.1 Modello del processo

1. **Risposta all'impulso**, è molto diffusa in ambito industriale in quanto nella pratica è molto facile da ottenere, può anche essere utilizzata su sistemi multi-variabile. Il modello del sistema da controllare viene ottenuto, eccitando il sistema con un ingresso ad impulso. L'uscita è legata all'ingresso dalla relazione:

$$y(t) = \sum_{i=1}^{\infty} h_i u(t-i)$$

Dove h_i , è l'insieme dei campioni ottenuti in uscita quando nel sistema agisce un segnale d'ingresso di impulso unitario (si veda la successiva figura).



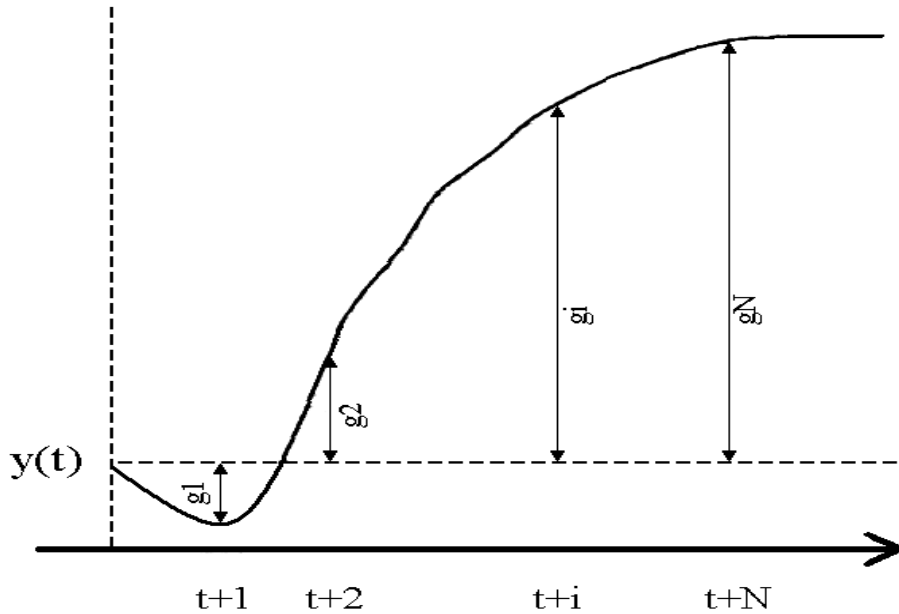
Nella pratica però solo i primi N campioni vengono utilizzati, che comunque sono lo stesso un numero considerevole, e questo può essere considerato un inconveniente. Un altro limite imposto da questo algoritmo come si può intuire è la stabilità a catena aperta del sistema. Nonostante tali svantaggi l'algoritmo può essere utilizzato in sistemi multi-variabile e il suo utilizzo non richiede nessuna informazione a priori sul processo. La previsione è data da:

$$\hat{y}(t+k|t) = \sum_{i=1}^N h_i u(t+k-i|t) = H(z^{-1})u(t+k|t)$$

2. **Risposta al gradino**, valgono le stesse considerazioni fatte per la risposta all'impulso, solo che in questo caso il sistema viene stimolato con un ingresso a gradino. Per un sistema stabile quindi si avrà come risposta:

$$y(t) = y_0 + \sum_{i=1}^N g_i \Delta u(t-i) = y_0 + G(z^{-1})(1-z^{-1})u(t)$$

Dove g_i è il valore della risposta campionata del processo e $\Delta u(t) = u(t) - u(t-1)$, come mostrato in figura.



Senza perdita di generalità il valore y_0 si può porre uguale a zero, cosicché l'uscita predetta diventa:

$$\hat{y}(t+k|t) = \sum_{i=1}^N g_i \Delta u(t+k-i|t)$$

3. **Funzione di trasferimento**, è un modello utilizzato in molte categorie di MPC, il modello del processo è descritto dalla sua funzione di trasferimento $G = \frac{B}{A}$ in modo che l'uscita sia data da:

$$A(z^{-1})y(t) = B(z^{-1})u(t)$$

Con $A(z^{-1})$ e $B(z^{-1})$ polinomi in z^{-1} e con le uscite predette calcolate con:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})} u(t+k|t)$$

In questa rappresentazione si possono anche considerare i processi instabili e si ha il vantaggio di richiedere pochi parametri. Il suo maggiore svantaggio è rappresentato dalla conoscenza a priori del processo considerato nonché dell'ordine dei polinomi A e B , che in molti sistemi rappresenta un grosso problema.

4. **State space (modello in spazio di stato)**, il processo è descritto dalle equazioni di stato qui rappresentate

$$\begin{cases} x(t+1) = Mx(t) + Nu(t) \\ y(t) = Qx(t) \end{cases}$$

Con x spazio di stato e M , N e Q le matrici di stato, di ingresso e di uscita del sistema. La predizione delle uscite è data da:

$$\hat{y}(t+k|t) = Q\hat{x}(t+k|t) = Q[M^k x(t) + \sum_{i=1}^k M^{i-1} Nu(t+k-i|t)]$$

Questo modello introduce molti vantaggi, soprattutto nei sistemi multi-variabile che in questo modo sono facili da implementare e si può usare anche qualora i processi siano instabili. I maggiori svantaggi sono dovuti dalla necessità di una buona conoscenza del processo da controllare e in alcuni casi dall'implementazione del modello stesso.

5. **Modelli non-lineari**, possono essere utilizzati per la rappresentazione del processo, ma introducono difficoltà nell'ottimizzazione del problema.

2.2.2 Modello del disturbo

La scelta del modello che rappresenta il disturbo, è importante tanto quanto quella del modello del processo. Un modello ampiamente utilizzato è il CARIMA (Controlled Auto-Regressive and Integrated Moving Average), dove il disturbo, dato dalla differenza tra l'uscita misurata del processo e quella calcolata nel modello è:

$$n(t) = \frac{C(z^{-1})}{D(z^{-1})} e(t)$$

dove $D(z^{-1})$ contiene un integratore $\Delta = 1 - z^{-1}$ e $C(z^{-1})$ solitamente si pone uguale a uno, mentre il termine $e(t)$ descrive un processo di rumore bianco con media nulla. Per predire il disturbo ad un istante k determinato, occorre sviluppare l'equazione sopra scritta:

$$\hat{n}(t+k|t) = \frac{C(z^{-1})}{D(z^{-1})} e(t+k) = \left[Q_k(z^{-1}) + z^{-k} \frac{C_k(z^{-1})}{D(z^{-1})} \right] e(t+k)$$

preso $Q_k(z^{-1}) = q_0 + q_1 z^{-1} + \dots + q_{k-1} z^{-(k-1)}$ si ha:

$$\hat{n}(t+k|t) = q_0 e(t+k) + q_1 e(t+k-1) + \dots + q_{k-1} e(t+1) + \frac{C_k(z^{-1})}{D(z^{-1})} e(t)$$

Ora, come si può notare, il termine $e_k = q_0 e(t+k) + q_1 e(t+k-1) + \dots + q_{k-1} e(t+1)$ è una combinazione lineare di termini che rappresentano un rumore bianco che non può essere predetto, quindi viene scartato, mentre:

$$\hat{n}(t+k|t) = \frac{C_k(z^{-1})}{D(z^{-1})} e(t) = \frac{C_k(z^{-1}) D(z^{-1})}{D(z^{-1}) C(z^{-1})} n(t) = \frac{C_k(z^{-1})}{C(z^{-1})} n(t)$$

rappresenta qualcosa di perfettamente noto, ponendo $C(z^{-1}) = 1$ la predizione del disturbo sarà data da:

$$\hat{n}(t+k|t) = C_k(z^{-1})n(t)$$

tale ultima formula verrà combinata con la funzione di trasferimento del modello, per ottenere la predizione delle uscite future.

2.3 Funzione obiettivo

I vari algoritmi per l'implementazione di un MPC, forniscono diverse funzioni di costo per ottenere una legge di controllo ottima. Lo scopo generale è che l'uscita futura inseguì un determinato segnale di riferimento ω , ottimizzando il segnale di comando, e tenendo in considerazione, se presenti, i vincoli sulle variabili. L'espressione generale della funzione obiettivo è:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2$$

Dalla formula si può notare che sono presenti due termini:

- un termine che tiene conto dello scostamento della traiettoria del sistema da quella di riferimento;
- un termine che tiene conto dell'intensità della variabile di controllo o della sua derivata: si può scegliere se ridurre il valore assoluto di u o la sua derivata o entrambe le cose; nella funzione sopra riportata si è scelto di ridurre la derivata di u , infatti

$$\Delta u(t+j-1) = u(t+j) - u(t+j-1)$$

N_1 e N_2 rappresentano il minimo e il massimo costo dell'orizzonte di controllo, stabilendo i limiti entro i quali, l'uscita del sistema deve inseguire il riferimento mentre N_u rappresenta l'orizzonte di controllo, $\omega(t)$ è la traiettoria di riferimento. Se questa è nota a priori, il sistema può reagire prima che il cambiamento sia avvenuto, evitando così il problema del ritardo nella risposta del processo. I coefficienti $\delta(j)$ e $\lambda(j)$ invece sono sequenze che considerano l'evoluzione e il comportamento del sistema, possono essere presi costanti o esponenziali, ad esempio:

$$\delta(j) = \alpha^{N_2-j}$$

Se $0 < \alpha < 1$, gli errori predetti negli intervalli più lontani sono più penalizzati rispetto a quelli prossimi a t , dando luogo a un controllo più agevole, minimizzando gli sforzi. Inoltre, una scelta opportuna di $\delta(j)$ e $\lambda(j)$ consente al controllore MPC di ottenere un equilibrio efficiente tra prestazioni e consumi. Ad esempio $\delta(j)$ prossimo a 1 conduce il sistema verso un *andamento veloce*, mentre $\lambda(j)$ prossimo a 1 permette un basso consumo del controllo. Ovviamente non è possibile ottenere entrambe le cose contemporaneamente, ed in generale si otterrà una soluzione di compromesso tra i due casi limite, appena discussi.

2.4 Legge di controllo

Per ottenere il valore di $u(t+k|t)$ basta minimizzare la funzione di costo $J(N_1, N_2, N_u)$. I valori delle uscite predette $\hat{y}(t+k|t)$, sono calcolati in funzione dei valori passati di ingressi e uscite e dei segnali di controllo futuri, facendo uso del modello scelto e sostituendolo nella funzione di

costo, ottenendo così un'espressione la cui minimizzazione porta ai valori cercati del segnale di controllo. La soluzione analitica, può essere ottenuta attraverso un criterio quadratico, se il modello è lineare e non vi sono vincoli nelle variabili, altrimenti va calcolata attraverso l'ausilio di metodi computazionali iterativi (ad esempio un problema di programmazione lineare). Qualunque sia il metodo, ottenere una soluzione non è facile, per il grande numero di variabili da considerare. E questo è uno degli aspetti più negativi dei controllori MPC, in quanto richiede l'utilizzo di una potenza di calcolo maggiore e quindi di un maggiore costo di costruzione, rispetto a controllori classici come ad esempio i PID.

Come già accennato, le strategie di controllo basate sul MPC, possono prevedere l'introduzione di vincoli sulle variabili, sebbene questo comporti degli oneri computazionali. Nella pratica infatti tutti i processi di produzione sono soggetti a dei vincoli. Gli attuatori hanno un limitato campo di azione, come ad esempio il caso delle valvole, limitate dalla posizione totalmente aperta o totalmente chiusa. Mentre nel caso delle uscite bisogna tenere presente i limiti fisici del processo, come massima temperatura, massima pressione, ecc ...

Quindi il modello matematico completo per determinare la legge di controllo sarà dato da:

$$\begin{cases} \min J \\ U_{min} \leq u(t) \leq U_{max} \\ dU_{min} \leq \Delta u(t) \leq dU_{max} \\ Y_{min} \leq y(t) \leq Y_{max} \end{cases}$$

Dove $\Delta u(t) = u(t) - u(t - 1)$ è un vincolo sulla derivata del segnale di controllo, inserito se necessario nella formulazione del problema.

3 Principali MPC commerciali

Tra i principali algoritmi di implementazione del MPC, non si posso scartare quelli che hanno avuto un'ampia diffusione negli ambienti industriali. Una grande sottofamiglia di controllori MPC, è rappresentata da quelli che impiegano modelli di convoluzione, detti anche metodi non parametrici. Basati sulla risposta impulsiva o a gradino, come precedentemente detto, hanno il vantaggio di essere di facile realizzazione. Tra i principali algoritmi si ricordano:

3.1 Dynamic Matrix Control

Identificato con la sigla DMC, questo controllore predittivo, fu utilizzato per la prima volta alla fine degli anni settanta, in un'industria petrolchimica. Al giorno d'oggi ha avuto un grandissimo successo dovuto alla sua facile implementazione e utilizzo. Infatti il DMC utilizza una risposta al gradino per la modellizzazione del processo, basata sull'acquisizione di un numero finito di termini N . Sempre assumendo che il processo sia stabile con la presenza di integratori al suo interno. I valori dei disturbi invece, sono considerati uguali lungo tutto l'orizzonte e il loro valore viene ricavato all'istante t come la differenza tra l'uscita misurata e quella predetta, cioè:

$$\hat{n}(t + k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t)$$

Per ricavare l'algoritmo per la predizione del segnale, come appena accennato, si parte dalla descrizione dell'impianto ricavata dalla sua risposta ad un gradino unitario:

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t - i)$$

I valori predetti, lungo l'orizzonte di predizione saranno:

$$\hat{y}(t + k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t + k - i|t) + \hat{n}(t + k|t)$$

Ricordando che il disturbo è considerato costante su tutto l'orizzonte, la predizione dei valori delle uscite diventa:

$$\begin{aligned} \hat{y}(t + k|t) &= \sum_{i=1}^k g_i \Delta u(t + k - i|t) + \sum_{i=k+1}^{\infty} g_i \Delta u(t + k - i|t) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t - i|t) \\ &= \sum_{i=1}^k g_i \Delta u(t + k - i|t) + f_u(t + k) \end{aligned}$$

dove il termine $f(t + k)$ rappresenta la risposta libera del sistema, indipendente dalle future azioni di controllo:

$$f_u(t + k) = y_m(t) - \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(t - i)$$

L'ipotesi di avere un processo da controllare stabile asintoticamente è imposta dai limiti fisici della memoria del controllore. Infatti ipotizzato stabile il sistema da controllare, i coefficienti g_i della

risposta al gradino unitario tenderanno ad un valore costante dopo un certo istante di campionamento N , e quindi si potrà considerare la sottrazione

$$g_{k+i} - g_i \approx 0, \quad \forall i > N$$

richiedendo quindi la memorizzazione, da parte del controllore, dei primi N campioni.

Preso ora $k = 1 \dots p$ (con p orizzonte di predizione delle uscite) e considerata m come il numero delle azioni di controllo:

$$\hat{y}(t+1|t) = g_1 \Delta u(t) + f_u(t+1)$$

$$\hat{y}(t+2|t) = g_1 \Delta u(t) + g_2 \Delta u(t+1) + f_u(t+2)$$

\vdots

$$\hat{y}(t+p|t) = \sum_{i=1}^m g_i \Delta u(t+p-i|t) + f_u(t+p)$$

e definita \mathbf{f} come il vettore della risposta libera e la matrice dinamica \mathbf{G} come:

$$\mathbf{G} = \begin{pmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \dots & g_1 \\ \vdots & \vdots & \vdots & \vdots \\ g_p & g_{p-1} & \dots & g_{p-m+1} \end{pmatrix}$$

la predizione sarà:

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}_u$$

Per quanto riguarda i disturbi misurabili, essi possono essere aggiunti molto semplicemente alle equazioni di predizione in quanto possono essere trattati come ingressi di sistema. Il contributo per la predizione dell'uscita da parte del disturbo misurabile sarà quindi:

$$\hat{\mathbf{y}}_d = \mathbf{D}\mathbf{d} + \mathbf{f}_d$$

dove \mathbf{D} è una matrice simile a \mathbf{G} , la quale contiene i coefficienti della risposta a gradino del sistema nel disturbo, \mathbf{d} è il vettore di incremento del disturbo e \mathbf{f}_d è la parte di risposta del sistema che non dipende dal disturbo.

In generale l'espressione di predizione dovrà quindi tenere conto delle risposte libere del sistema e dei disturbi misurabili e non, che agiscono sull'impianto da controllare. Considerandola come una somma di effetti si avrà:

$$\mathbf{f} = \mathbf{f}_u + \mathbf{D}\mathbf{d} + \mathbf{f}_d + \mathbf{f}_n$$

da cui si potrà avere una formulazione generale dell'espressione di predizione

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

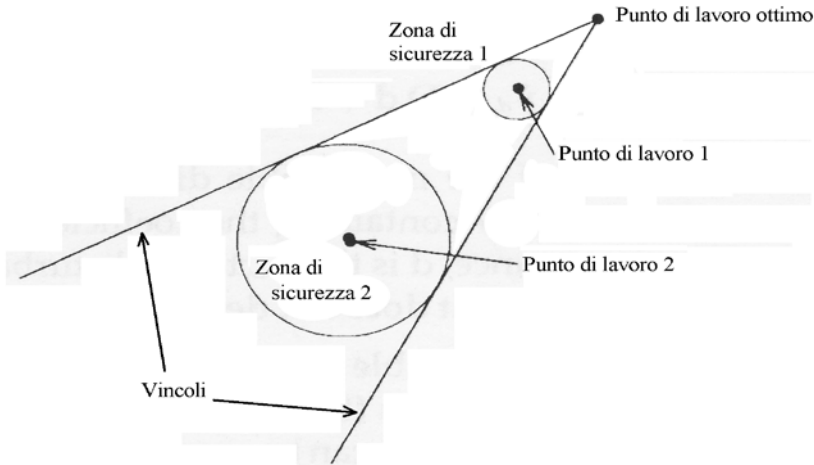
Come detto fin'ora l'obiettivo principale di un controllore predittivo è quello di portare il valore dell'uscita il più vicino possibile al valore stabilito nella traiettoria di riferimento operando in una zona ristretta a causa dei vincoli utilizzati. Tutto ciò viene fatto minimizzando la funzione obiettivo descritta precedentemente, che nel caso di un controllore DMC diventa:

$$J = \sum_{j=1}^p [\hat{y}(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^m \lambda [\Delta u(t+j-1)]^2$$

Se per ipotesi escludiamo la presenza di vincoli, la funzione di costo da minimizzare prende la forma conosciuta $J = \mathbf{e}\mathbf{e}^T + \lambda \mathbf{u}\mathbf{u}^T$, con \mathbf{e} vettore di previsione degli errori futuri lungo l'orizzonte di predizione. Derivando il termine J rispetto ad \mathbf{u} ed eguagliando a zero si ottiene così il valore delle azioni di controllo future:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f})$$

In presenza di vincoli, come maggiormente possibile nel caso di utilizzo su un impianto reale del controllore, il DMC ha il pregio di trattarli con maggiore semplicità rispetto ad altri algoritmi per il controllo predittivo. Come si può vedere dalla figura sottostante, il punto economico di lavoro si trova nell'intersezione dei due vincoli, però è necessario mantenere una zona di sicurezza intorno al punto di funzionamento, poiché l'effetto di perturbazioni potrebbe portare il processo a violare i vincoli. Le dimensioni di tale area possono essere ridotte in base alla capacità del controllore nella gestione dei vincoli.



I vincoli sugli ingressi e quelli sulle uscite, sono entrambi considerati nella seguente forma:

$$\sum_{i=1}^N C_{yi}^j \hat{y}(t+k|t) + C_{ui}^j u(t+k-i) + c^j \leq 0, \quad j = 1 \dots N_c$$

dove C^j sono coefficienti reali.

Dato che le uscite future possono essere messe in relazione direttamente col vettore dei segnali di controllo attraverso la matrice dinamica del sistema, possiamo tener conto di tutti i vincoli relativi all'ingresso e all'uscita mediante la seguente disuguaglianza:

$$Ru \leq c$$

Come si può notare non è altro che la formulazione di un problema quadratico, risolvibile con tecniche idonee.

3.2 Model Algorithmic Control

Conosciuto anche come *Model predictive Heuristic Control*, il cui software è chiamato IDICOM (*Identification-command*), è un metodo simile a quello illustrato in precedenza ma che esibisce alcune piccole differenze. In primo luogo questo controllore utilizza un modello basato sulla risposta impulsiva del sistema.

$$y(t) = \sum_{i=1}^{\infty} h_i u(t-i) = H(z^{-1})u(t)$$

Con questo modello si calcolano le uscite ad un determinato istante temporale come combinazione lineare dei valori d'ingresso passati, i cui pesi sono i coefficienti della risposta impulsiva h_i . Troncando ai primi N campioni il modello e assumendo che il sistema sia stabile e causale, si può facilmente ottenere la predizione delle uscite all'istante k , data dalla seguente formula

$$\hat{y}(t+k|t) = \sum_{i=1}^N h_i u(t+k-i|t) + \hat{n}(t+k|t)$$

dove la sommatoria può essere scomposta in due sommatorie più semplici:

$$f_r(t+k) = \sum_{i=k+1}^N h_i u(t+k-i) \quad f_0(t+k) = \sum_{i=1}^k h_i u(t+k-i)$$

con f_r che rappresenta la risposta libera mentre f_0 la risposta forzata del sistema. Assumendo che il disturbo rimanga costante lungo tutto l'intervallo futuro di predizione ed uguale al valore calcolato all'istante t , come differenza tra uscita misurata e uscita predetta:

$$\hat{n}(t+k|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t) = y_m(t) - \sum_{i=1}^N h_i u(t-i)$$

la predizione sarà quindi data da:

$$\hat{y}(t+k|t) = f_r + f_0 + \hat{n}(t|t)$$

Definendo M come l'orizzonte di predizione, \mathbf{u}_+ come il vettore delle azioni di controllo proposte e \mathbf{u}_- come il vettore delle azioni di controllo passate si avranno le matrici:

$$\mathbf{u}_+ = \begin{pmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+M-1) \end{pmatrix} \quad \mathbf{u}_- = \begin{pmatrix} u(t-N+1) \\ u(t-N+2) \\ \vdots \\ u(t-1) \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} \hat{y}(t+1) \\ \hat{y}(t+2) \\ \vdots \\ \hat{y}(t+M) \end{pmatrix} \quad \mathbf{n} = \begin{pmatrix} \hat{n}(t+1) \\ \hat{n}(t+2) \\ \vdots \\ \hat{n}(t+M) \end{pmatrix}$$

e definendo le matrici:

$$\mathbf{H}_1 = \begin{pmatrix} h_1 & 0 & \dots & 0 \\ h_2 & h_1 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ h_M & h_{M-1} & \dots & h_1 \end{pmatrix} \quad \mathbf{H}_2 = \begin{pmatrix} h_N & \dots & h_i & \dots & h_1 \\ 0 & \dots & h_j & \dots & h_2 \\ \dots & \ddots & \dots & \dots & \dots \\ 0 & \dots & h_N & \dots & h_{M+1} \end{pmatrix}$$

l'espressione della predizione sarà:

$$\mathbf{y} = \mathbf{H}_1 \mathbf{u}_+ + \mathbf{H}_2 \mathbf{u}_- + \mathbf{n}$$

Il principale obiettivo dei controllori MAC consiste nel determinare una sequenza di segnali di controllo, in grado di minimizzare la somma delle deviazioni quadratiche delle uscite predette rispetto una traiettoria di riferimento

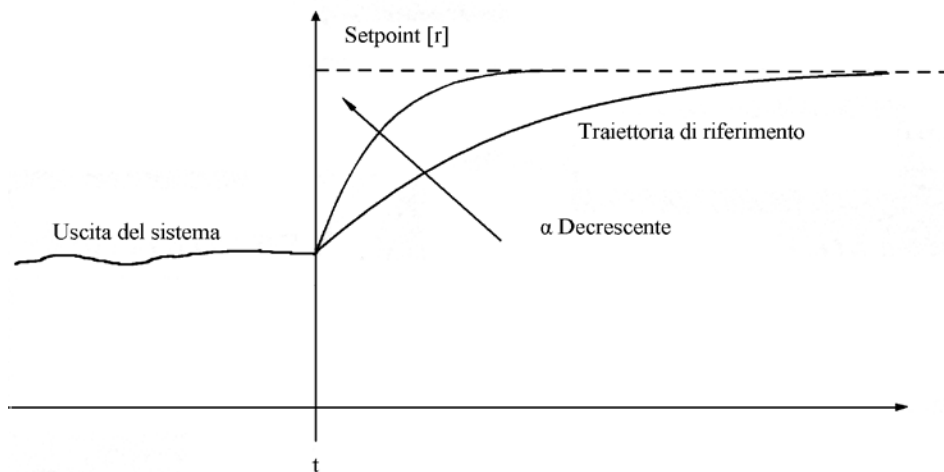
$$\mathbf{w} = \begin{pmatrix} w(t+1) \\ w(t+2) \\ \vdots \\ w(t+N) \end{pmatrix}$$

Normalmente, in questo tipo di controllori predittivi, la traiettoria di riferimento è un'approssimazione graduale delle uscite del sistema, rispetto a un segnale di riferimento, data dalla seguente equazione di riferimento:

$$w(t+k) = \alpha w(t+k-1) + (1-\alpha)r(t+k) \quad k = 1 \dots N$$

con $w(t) = y(t)$.

Come si può notare dalla figura a pagina seguente, la scelta del parametro reale α determina la forma della traiettoria di riferimento, nonché la velocità con cui si vuole raggiungere un determinato punto di lavoro nel sistema. Nella pratica questo risulta un approccio molto naturale e intuitivo perché modificando il valore del parametro α , si possono facilmente ottenere alcune delle principali caratteristiche richieste: ad esempio tempo di assestamento, tempo di salita e sovraelongazione della sua risposta.



Se i futuri errori sono espressi come

$$\mathbf{e} = \mathbf{w} - \mathbf{y} = \mathbf{w} - \mathbf{H}_2 \mathbf{u}_- - \mathbf{n} - \mathbf{H}_1 \mathbf{u}_+ = \mathbf{w} - \mathbf{f} - \mathbf{H}_1 \mathbf{u}_+$$

la funzione obiettivo che minimizza l'errore e il costo di controllo, una volta scritta nella seguente forma:

$$J = \mathbf{e}^T \mathbf{e} + \lambda \mathbf{u}_+^T \mathbf{u}_+$$

nell'ipotesi che non ci siano vincoli, la predizione dei segnali d'ingresso futuri sarà:

$$\mathbf{u}_+ = (\mathbf{H}_1^T \mathbf{H}_1 + \lambda \mathbf{I})^{-1} \mathbf{H}_1^T (\mathbf{w} - \mathbf{f})$$

4 Generalized Predictive Control

In questo capitolo verrà trattato uno dei più popolari algoritmi per il controllo predittivo. Oltre ad illustrare la strategia di controllo verrà fornita un'implementazione del controllore predittivo, eseguita e testata su un modello appositamente scelto tramite l'ausilio del programma di calcolo Matlab.

Il metodo GPC, fu proposto per la prima volta da D.W. Clarke, C. Mohtadi e P.S. Tutts, nel 1987. Ebbe da subito un grande successo sia nell'industria che nel mondo accademico, esso infatti fornisce buone prestazioni e un certo grado di robustezza. I motivi della sua grande popolarità stanno nella possibilità di gestire diversi tipi di controllo e obiettivi per una vasta gamma di impianti, mantenendo limitato il numero delle variabili da tenere in considerazione. A differenza degli altri MPC commerciali illustrati fino ad ora, il GPC richiede a priori una conoscenza dell'impianto da controllare, esso infatti come modello del sistema utilizza una **funzione di trasferimento** della quale sono già stati precedentemente elencati pregi e difetti. L'idea di base del controllore GPC consiste nel calcolare una sequenza di segnali di controllo futuri ottimizzati attraverso una funzione di costo multistadio definita attraverso l'orizzonte di predizione, utilizzando molti degli accorgimenti visti fino ad ora, ma avendo delle differenze che si potranno notare più avanti durante la descrizione di questo controllore. Sebbene i controllori GPG possono essere usati in sistemi multivariabili, sarà preso in considerazione l'implementazione dell'algoritmo per il solo caso dei sistemi SISO.

4.1 Modello del processo e del disturbo

La funzione di trasferimento ottenuta attraverso la modellizzazione e linearizzazione dell'impianto da controllare dovrà avere la seguente forma:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})e(t)$$

dove $u(t)$ e $y(t)$ sono i segnali di ingresso e di uscita, $e(t)$ un rumore bianco di media nulla e il termine d rappresenta il ritardo del sistema, mentre i termini A , B e C rappresentano polinomi in z^{-1} così descritti:

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na}$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}$$

$$C(z^{-1}) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc}$$

La modellizzazione del disturbo viene effettuata con l'ausilio di un modello CARIMA (precedentemente descritto) e il modello della funzione di trasferimento diventa:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta}$$

con $\Delta = 1 - z^{-1}$, inoltre per semplificare i prossimi passaggi $C(z^{-1})$ verrà assunto uguale a 1 senza perdita di generalità.

4.2 Funzione obiettivo

Nell'algoritmo GPC, viene minimizzata una funzione di costo nella forma già vista:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - \omega(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2$$

4.3 Formulazione dell'algoritmo

Da quanto detto fino ad ora, si può notare come la creazione delle matrici che permetteranno di calcolare il segnale di controllo, attraverso la minimizzazione di una funzione di costo, è da ritenersi un procedimento iterativo. Il quale in molti casi può rivelarsi non completamente ottimale come formulazione dell'algoritmo. Sebbene il calcolo di molte variabili venga effettuato prima della messa in funzione dell'impianto ("fase fuori linea") a gravare principalmente sul calcolo delle variabili è la scelta dell'orizzonte di predizione. Sarà quindi sempre utile cercare algoritmi in grado di ottimizzare le prestazioni dell'hardware che implementerà il controllore MPC.

Nel caso del controllore GPC per la fase "fuori linea" si può, inizialmente, prendere in considerazione la seguente equazione Diofantea:

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_j(z^{-1}) \quad \text{con} \quad N_1 \leq j \leq N_2$$

ricavata dalla divisione

$$\frac{1}{\tilde{A}(z^{-1})} = E_j(z^{-1}) + \frac{z^{-j}F_j(z^{-1})}{\tilde{A}(z^{-1})}$$

fino a quando il resto viene fattorizzato come $z^{-j}F_j(z^{-1})$. Qui vale l'uguaglianza $\tilde{A}(z^{-1}) = A(z^{-1})\Delta$, mentre E_j e F_j sono polinomi rispettivamente di grado $j-1$ e na .

Ora moltiplicando ambo i membri dell'equazione, che descrive il modello del sistema con il disturbo, per il termine $\Delta E_j(z^{-1})z^j$:

$$\Delta E_j(z^{-1})z^j [A(z^{-1})y(t)] = \Delta E_j(z^{-1})z^j \left[z^{-d}B(z^{-1})u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \right]$$

si ottiene con una serie di passaggi e sostituzioni:

$$[1 - z^{-j}F_j(z^{-1})]y(t+j) = E_j(z^{-1})B(z^{-1})\Delta u(t-1-d+j) + E_j(z^{-1})e(t+j)$$

che una volta riscritta diventa:

$$y(t+j) = F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t-1-d+j) + E_j(z^{-1})e(t+j)$$

Ora definendo $G_j(z^{-1}) \triangleq E_j(z^{-1})B(z^{-1})$ e scartando i termini $E_j(z^{-1})e(t+j)$, in quanto combinazione lineare di rumori bianchi futuri che non possono essere predetti. Il valore dell'uscita predetta all'istante j sarà:

$$\hat{y}(t+j|t) = G_j(z^{-1})\Delta u(t-1-d+j) + F_j(z^{-1})y(t)$$

Il punto chiave nell'utilizzare l'equazione Diofantea, per il calcolo di E_j e di F_j , consiste nello sfruttare la sua ricorsione. In pratica, partendo da $1/\tilde{A}(z^{-1})$, si ottengono:

$$E_j(z^{-1}) = e_{j,0} + e_{j,1}z^{-1} + \dots + e_{j,j-1}z^{-(j-1)}$$

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + \dots + f_{j,na}z^{-na}$$

Ora supponendo di voler calcolare $E_{j+1}(z^{-1})$ e $F_{j+1}(z^{-1})$ bisognerebbe ripetere la divisione del polinomio $\tilde{A}(z^{-1})$ fino ad ottenere il resto fattorizzato $z^{-(j+1)}F_{j+1}(z^{-1})$ e questo comporterebbe un onere computazionale che non si può non considerare. Si può dimostrare però che all'iterazione successiva il polinomio $E(z^{-1})$ diventerà:

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + e_{j+1,j}z^{-j}$$

pertanto sarà necessario per ogni $N_1 < i \leq N_2$:

$$\frac{F_i(z^{-1})}{\tilde{A}(z^{-1})} = e_{i+1,0} + \frac{f_{i+1,0} + f_{i+1,1}z^{-1} + \dots + f_{i+1,na}z^{-na}}{\tilde{A}(z^{-1})}$$

$$E_{i+1}(z^{-1}) = E_i(z^{-1}) + e_{i+1,0}$$

Per tale scopo, in Matlab è stato implementato una funzione ("*divisione.m*"), che ha il compito di ricevere sotto forma di vettore $\tilde{A}(z^{-1})$, $F(z^{-1})$ e il valore n per ottenere i polinomi resto $z^{-(n)}F_n(z^{-1})$ e quoziente $E_n(z^{-1})$, dove n , nella prima iterazione sarà uguale a N_1 , mentre nelle successive basterà trovare solo il primo quoziente e il suo resto, calcolati tramite il resto precedente fino a che $n \leq N_2$.

Nell'algoritmo GPC, si cerca dunque di ottenere, tramite la minimizzazione della funzione di costo, l'insieme dei segnali d'uscita $u(t), u(t+1), \dots, u(t+N)$ e tenendo presente che la funzione di trasferimento considera anche modelli in cui esiste un tempo di ritardo della risposta rispetto ad un segnale di ingresso, le seguenti variabili saranno definite come: $N_1 \triangleq d+1$, $N_2 \triangleq N+d$ e $N_u \triangleq N$.

Le predizioni delle uscite future saranno:

$$\begin{aligned}\hat{y}(t+d+1|t) &= G_{d+1}(z^{-1})\Delta u(t) + F_{d+1}(z^{-1})y(t) \\ \hat{y}(t+d+2|t) &= G_{d+2}(z^{-1})\Delta u(t+1) + F_{d+2}(z^{-1})y(t) \\ &\vdots \\ \hat{y}(t+d+N|t) &= G_{d+N}(z^{-1})\Delta u(t+N-1) + F_{d+N}(z^{-1})y(t)\end{aligned}$$

che posso essere riscritte come

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{F}(z^{-1})\mathbf{y}(t) + \mathbf{G}'(z^{-1})\Delta\mathbf{u}(t)$$

dove:

$$\mathbf{y} = \begin{pmatrix} \hat{y}(t+d+1|t) \\ \hat{y}(t+d+2|t) \\ \vdots \\ \hat{y}(t+d+N|t) \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} g_0 & 0 & 0 & 0 \\ g_1 & g_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \dots & g_0 \end{pmatrix} \quad \mathbf{F}(z^{-1}) = \begin{pmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \vdots \\ F_{d+N}(z^{-1}) \end{pmatrix}$$

$$\mathbf{G}'(z^{-1}) = \begin{pmatrix} [G_{d+1}(z^{-1}) - g_0]z \\ [G_{d+2}(z^{-1}) - g_0 - g_1 z^{-1}]z^2 \\ \vdots \\ [G_{d+N}(z^{-1}) - g_0 - g_1 z^{-1} - \dots - g_{N-1} z^{-(N-1)}]z^N \end{pmatrix}$$

Si può notare come i termini $\mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t)$ dipendono solo dall'evoluzione passata del sistema definita con il termine \mathbf{f} . Si avrà quindi:

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

Inoltre definendo il vettore della traiettoria di riferimento come:

$$\mathbf{w} \triangleq \begin{pmatrix} w(t+d+1) \\ w(t+d+2) \\ \vdots \\ w(t+d+N) \end{pmatrix}$$

e ipotizzando l'assenza di vincoli nel problema, la funzione di costo si può scrivere come:

$$\mathbf{J} = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u}$$

Dalla quale derivando \mathbf{J} rispetto a \mathbf{u} e uguagliando a zero, una volta fatti i dovuti raccoglimenti, si otterrà come vettore di previsione dei segnali di controllo incrementali futuri:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) = \mathbf{K}(\mathbf{w} - \mathbf{f})$$

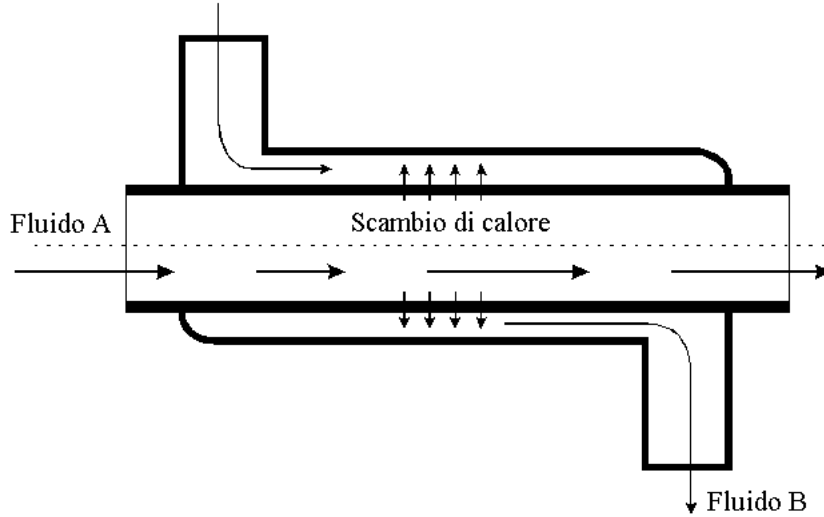
Ricordando inoltre che $\Delta u(t) = u(t) - u(t-1)$, il segnale di comando mandato all'impianto sarà:

$$u(t) = \Delta u(t) + u(t-1)$$

L'incremento dell'azione di controllo sarà dunque dato dal fattore \mathbf{K} in proporzione agli errori futuri, diversamente a quanto accade in un controllore a retroazione classico, in cui l'azione di controllo viene effettuata in base agli errori passati.

4.5 Disturbi misurabili

Molti dei processi industriali da controllare, possono essere soggetti a disturbi esterni causati dalle variazioni sulle variabili misurabili. Queste situazioni si verificano tipicamente in sistemi le cui uscite sono affette da variazioni sul regime del sistema. Un esempio semplice è dato dal caso di uno scambiatore di calore, di cui un semplice schema è mostrato nella figura sottostante.



Il compito del controllore consiste nel regolare il flusso del fluido B, affinché all'uscita dello scambiatore si abbia una certa temperatura desiderata, ottenuta tramite lo scambio di calore con il flusso del fluido A. Se quest'ultimo ha delle variazioni di portata, allora ci saranno anche delle variazioni di temperatura sul fluido B che il controllore andrà a correggere variandone il flusso. Nel caso di controllori a retroazione classici è facile implementare una legge di controllo basata sui disturbi misurabili, mentre del controllore predittivo GPC verrà illustrato qui di seguito un procedimento per implementare il controllo in presenza di questo tipo di problemi.

Nella funzione di trasferimento fino ad ora utilizzata, viene inserito il disturbo misurabile in questo modo:

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) + D(z^{-1})v(t) + C(z^{-1})\frac{e(t)}{\Delta}$$

dove la variabile $v(t)$ rappresenta il disturbo misurato al tempo t , mentre il polinomio $D(z^{-1})$ è definito come

$$D(z^{-1}) \triangleq d_0 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{nd} z^{-nd}$$

Una volta moltiplicate la funzione di trasferimento per il termine $\Delta E_j(z^{-1})z^j$ si ottiene:

$$\Delta E_j(z^{-1})z^j [A(z^{-1})y(t)] = \Delta E_j(z^{-1})z^j \left[B(z^{-1})u(t-1) + D(z^{-1})v(t) + C(z^{-1})\frac{e(t)}{\Delta} \right]$$

dalla quale si ricava

$$y(t+j) = F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t-1+j) + E_j(z^{-1})D(z^{-1})\Delta v(t+j) + E_j(z^{-1})e(t+j)$$

Ora definendo $H_j(z^{-1}) \triangleq E_j(z^{-1})D(z^{-1})$ e scartando i termini $E_j(z^{-1})e(t+j)$ che sono, come già detto una combinazione lineare di rumori bianchi futuri e quindi non possono essere predetti. si avrà che la predizione dell'uscita all'istante j sarà:

$$\hat{y}(t+j|t) = G_j(z^{-1})\Delta u(t-1+j) + H_j(z^{-1})\Delta v(t+j) + F_j(z^{-1})y(t)$$

da cui verranno sviluppate le predizioni date da tutti gli istanti lungo l'orizzonte:

$$\begin{aligned}\hat{y}(t+1|t) &= G_1(z^{-1})\Delta u(t) + H_1(z^{-1})\Delta v(t+1) + F_1(z^{-1})y(t) \\ \hat{y}(t+2|t) &= G_2(z^{-1})\Delta u(t+1) + H_2(z^{-1})\Delta v(t+2) + F_2(z^{-1})y(t) \\ &\vdots \\ \hat{y}(t+N|t) &= G_N(z^{-1})\Delta u(t+N-1) + H_N(z^{-1})\Delta v(t+N) + F_N(z^{-1})y(t)\end{aligned}$$

le quali posso essere riscritte come

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{v} + \mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t) + \mathbf{H}'(z^{-1})\Delta v(t)$$

dove le nuove variabili aggiunte, sono definite in questo modo

$$\begin{aligned}\mathbf{H} &= \begin{pmatrix} h_0 & 0 & 0 & 0 \\ h_1 & h_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ h_{N-1} & h_{N-2} & \dots & h_0 \end{pmatrix} \\ \mathbf{H}' &= \begin{pmatrix} [H_1(z^{-1}) - h_0]z \\ [H_2(z^{-1}) - h_0 - h_1z^{-1}]z^2 \\ \vdots \\ [H_N(z^{-1}) - h_0 - h_1z^{-1} - \dots - h_{N-1}z^{-(N-1)}]z^N \end{pmatrix} \\ \mathbf{v} &= \begin{pmatrix} \Delta v(t+1) \\ \Delta v(t+2) \\ \vdots \\ \Delta v(t+N) \end{pmatrix}\end{aligned}$$

Dall'equazione di predizione si osserva che i termini $\mathbf{F}(z^{-1})y(t)$, $\mathbf{G}'(z^{-1})\Delta u(t)$ e $\mathbf{H}'(z^{-1})\Delta v(t)$ dipendono rispettivamente dai valori d'uscita, d'ingresso e del disturbo misurato passati. Si può quindi definire la variabile \mathbf{f}' come evoluzione passata del sistema, cioè:

$$\mathbf{f}' \triangleq \mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t) + \mathbf{H}'(z^{-1})\Delta v(t)$$

Per quanto riguarda la predizione dei disturbi misurabili si può, conoscendo le caratteristiche dell'impianto da controllare creare una legge che cerchi di predire come si svilupperà l'evoluzione futura del disturbo. Oppure rendendo tutti i termini di \mathbf{v} uguali al valore del disturbo misurato all'istante t iniziale. Quindi, non essendovi un modo preciso che permetta di determinare i valori futuri del disturbo misurabili, bisognerà prestare un po' d'attenzione durante lo studio dell'impianto da controllare. Determinati i valori di \mathbf{v} e definendo

$$\mathbf{f} = \mathbf{H}\mathbf{v} + \mathbf{f}'$$

la predizione ora sarà data da

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

che in assenza di vincoli, permette di trovare una soluzione ottimale delle uscite di controllo con le stesse modalità spiegate in precedenza.

4.6 Presenza di vincoli nel GPC

La presenza di vincoli sulle variabili da controllare è diffusa nella gran parte dei problemi di controllo. Si pensi infatti ai limiti imposti dagli attuatori i quali hanno un campo preciso di funzionamento, oltre il quale il segnale di uscita nel migliore dei casi satura mentre nel peggiore potrebbe causare guasti nell'impianto e oneri nella sua riparazione. Anche le uscite del sistema da controllare potrebbero essere soggette a dei vincoli, come ad esempio l'impostazione di una determinata soglia di guardia minima (massima) sulla pressione o sul livello di liquido in un serbatoio. Il problema dei vincoli come si può intuire, in molti casi non può essere trascurato e quindi necessita di una particolare attenzione in fase di progettazione. L'introduzione di vincoli però, non porta più ad una soluzione ottenuta per via analitica bensì ad un problema quadratico standard *QP* che determina, se esiste, una soluzione ottima tramite un processo iterativo. In commercio si possono trovare svariati algoritmi che risolvono iterativamente il problema *QP* ad ogni istante di campionamento, dopo di che il segnale $u(t)$ viene inviato al processo come avviene nella formulazione senza vincoli.

Anche Matlab al suo interno dispone di una funzione, chiamata *quadprog* per l'implementazione di un problema di programmazione quadratica della forma:

$$\begin{cases} \min \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{q}^T \mathbf{u} \\ \mathbf{A} \mathbf{u} \leq \mathbf{b} \\ \mathbf{A}_{eq} \mathbf{u} = \mathbf{b}_{eq} \\ \mathbf{lb} \leq \mathbf{u} \leq \mathbf{ub} \end{cases}$$

dove \mathbf{b} , \mathbf{b}_{eq} , \mathbf{ub} , \mathbf{lb} , \mathbf{u} , \mathbf{q} sono vettori e \mathbf{Q} , \mathbf{A} , \mathbf{A}_{eq} sono matrici. E la sua sintassi è

$$\mathbf{u} = \text{quadprog}(\mathbf{Q}, \mathbf{q}, \mathbf{A}, \mathbf{b}, \mathbf{A}_{eq}, \mathbf{b}_{eq}, \mathbf{lb}, \mathbf{ub})$$

Questo tipo di risolutore che viene anche chiamato “*metodo di proiezione*”, fa parte di una categoria di algoritmi che impiegano strategie di tipo attivo: per raggiungere la soluzione i vincoli di disuguaglianza vengono inizialmente trasformati in vincoli di uguaglianza, mediante variabili di slack. Ottenendo così un sistema formato da vincoli d'uguaglianza, che prende il nome di “*active set*”, da cui ne deriva il nome di strategie di tipo attivo. Poiché i vincoli di disuguaglianza vengono, come detto trasformati in vincoli di uguaglianza, partiremo nella trattazione di quest'ultimi.

4.6.1 Vincoli di uguaglianza

Rappresentati nella forma:

$$\mathbf{A}_{eq} \mathbf{u} = \mathbf{b}_{eq}$$

con \mathbf{A}_{eq} matrice $m \times n$ e \mathbf{b}_{eq} vettore di dimensione m . Assumendo per ipotesi che $m \leq n$, si possono sfruttare le n variabili d'ingresso in funzione di $n - m$ variabili, in modo da ottenere:

$$\mathbf{u} = \mathbf{Y}\mathbf{b}_{eq} + \mathbf{Z}\mathbf{v}$$

dove \mathbf{Y} e \mathbf{Z} sono rispettivamente matrici di dimensione $n \times m$ e $n \times (n - m)$, tali che $\mathbf{A}_{eq}\mathbf{Y} = \mathbf{I}$ e $\mathbf{A}_{eq}\mathbf{Z} = \mathbf{0}$ formando così una matrice $[\mathbf{Y} \ \mathbf{Z}]$ di rango n . Ora prendendo la funzione obbiettivo:

$$J = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u}$$

dalla quale, riscritta in questa forma

$$J = \frac{1}{2} \mathbf{u}^T 2(\mathbf{G}^T - \lambda \mathbf{I}) \mathbf{u} + 2(\mathbf{f} - \mathbf{w})^T \mathbf{G} \mathbf{u} + (\mathbf{f} - \mathbf{w})^T (\mathbf{f} - \mathbf{w})$$

e inserendo l'equazione $\mathbf{u} = \mathbf{Y}\mathbf{b}_{eq} + \mathbf{Z}\mathbf{v}$ al suo interno si ottiene

$$\begin{aligned} J &= \frac{1}{2} [\mathbf{Y}\mathbf{b}_{eq} + \mathbf{Z}\mathbf{v}]^T \mathbf{Q} [\mathbf{Y}\mathbf{b}_{eq} + \mathbf{Z}\mathbf{v}] + \mathbf{q}^T [\mathbf{Y}\mathbf{b}_{eq} + \mathbf{Z}\mathbf{v}] + f_0 \\ &= \frac{1}{2} \mathbf{v}^T \mathbf{Z}^T \mathbf{Q} \mathbf{Z} \mathbf{v} + [\mathbf{q}^T + \mathbf{b}_{eq}^T \mathbf{Y}^T \mathbf{Q}] \mathbf{Z} \mathbf{v} + \left[\frac{1}{2} \mathbf{b}_{eq}^T \mathbf{Y}^T \mathbf{Q} + \mathbf{q}^T \right] \mathbf{Y} \mathbf{b}_{eq} + f_0 \end{aligned}$$

dove

$$\mathbf{Q} = 2(\mathbf{G}^T \mathbf{G} - \lambda \mathbf{I})$$

$$\mathbf{q}^T = 2(\mathbf{f} - \mathbf{w})^T \mathbf{G}$$

$$f_0 = (\mathbf{f} - \mathbf{w})^T (\mathbf{f} - \mathbf{w})$$

cioè un problema QP con $n - m$ variabili, nel quale, se la matrice $\mathbf{Z}^T \mathbf{Q} \mathbf{Z}$ è definita positiva esiste solo un punto di ottimo, che può essere ottenuto tramite la risoluzione del seguente sistema di equazioni lineari

$$\mathbf{Z}^T \mathbf{Q} \mathbf{Z} \mathbf{v} = -\mathbf{Z}^T (\mathbf{q} + \mathbf{Q} \mathbf{Y} \mathbf{b}_{eq})$$

Si può inoltre osservare che se esiste un punto \mathbf{u}_k tale che soddisfi l'equazione $\mathbf{A}_{eq} \mathbf{u}_k = \mathbf{b}_{eq}$, allora ogni altro punto \mathbf{u} che soddisfa i vincoli del problema, può cioè essere espresso come $\mathbf{u} = \mathbf{u}_k + \mathbf{Z}\mathbf{v}$ facendo così ottenere per costruzione il vettore $\mathbf{Y}\mathbf{b}_{eq}$, tale per cui i vincoli siano soddisfatti. Infine si può ottenere il vettore \mathbf{v} risolvendo la seguente equazione lineare:

$$\mathbf{Z}^T \mathbf{Q} \mathbf{Z} \mathbf{v} = -\mathbf{Z}^T g(\mathbf{u}_k)$$

dove $g(\mathbf{u}_k) = \mathbf{Q}\mathbf{u}_k + \mathbf{q}$ rappresenta il gradiente di $J(\mathbf{u})$ nel punto \mathbf{u}_k .

La variabile f_0 non viene considerata nell'ottimizzazione in quanto un termine costante e quindi, che non contribuisce alla minimizzazione del segnale in quanto

$$\min[f(x) + c_0] = \min f(x) + c_0$$

Un modo generale per ottenere \mathbf{Y} e \mathbf{Z} opportune consiste nella scelta di una matrice \mathbf{W} di dimensione $n \times (n - m)$ tale da ottenere una matrice $\begin{bmatrix} \mathbf{A}_{eq} \\ \mathbf{W} \end{bmatrix}$ non singolare, la cui inversa può essere espressa come

$$\begin{bmatrix} \mathbf{A}_{eq} \\ \mathbf{W} \end{bmatrix}^{-1} = [\mathbf{Y} \ \mathbf{Z}]$$

da cui come si può notare si ottiene: $\mathbf{A}_{eq}\mathbf{Y} = \mathbf{I}$ e $\mathbf{A}_{eq}\mathbf{Z} = \mathbf{0}$.

4.6.2 Vincoli di disuguaglianza

Si parte dalla formulazione generale in cui i vincoli di disuguaglianza vengono espressi, cioè come $\mathbf{A}\mathbf{u} \leq \mathbf{b}$. Bisogna trovare un punto ammissibile \mathbf{u}_{am} in grado di soddisfare, tramite l'inserimento di opportune variabili slack, la disuguaglianza $\mathbf{A}\mathbf{u}_{am} \leq \mathbf{b}$ in modo tale da ottenere per ogni riga i di \mathbf{A} l'uguaglianza $\mathbf{a}_i\mathbf{u}_{am} = \mathbf{b}_i$. Dove le righe \mathbf{a}_i e i coefficienti \mathbf{b}_i verranno adoperati per creare la matrice \mathbf{A}_{eq} e il vettore \mathbf{b}_{eq} permettendo di trovare una soluzione ottima nel modo visto in precedenza per i vincoli di uguaglianza. Questo metodo richiede di trovare un punto ammissibile iniziale, nel senso che prima di far partire il calcolo iterativo per la ricerca di un punto ottimo, è necessario ricercare un punto \mathbf{u}_0 per partire all'interno della regione ammissibile. Un algoritmo possibile per la ricerca di questa soluzione è il seguente descritto in pseudocodice:

1. si fissa un punto iniziale \mathbf{u}_0 ;
2. si pone $\mathbf{a} = \mathbf{A}\mathbf{u}_0 - \mathbf{b}$;
3. se $\mathbf{a} \leq \mathbf{0}$ STOP (\mathbf{u}_0 è punto ammissibile);
4. $a_{max} = \max(\mathbf{a})$;
5. si risolve il seguente problema di ottimizzazione aumentato

$$\begin{cases} \min_{\mathbf{u}_1} J_1(\mathbf{u}_1) \\ \mathbf{A}_1\mathbf{u}_1 \leq \mathbf{b} \end{cases}$$

dove

$$\mathbf{u}_1 = \begin{bmatrix} \mathbf{u}_0 \\ a_{max} \end{bmatrix} \quad \text{e} \quad \mathbf{A}_1 = [\mathbf{A} \quad -\mathbf{1}]$$

e quindi si riparte dal punto 1 utilizzando come punto iniziale \mathbf{u}_1 .

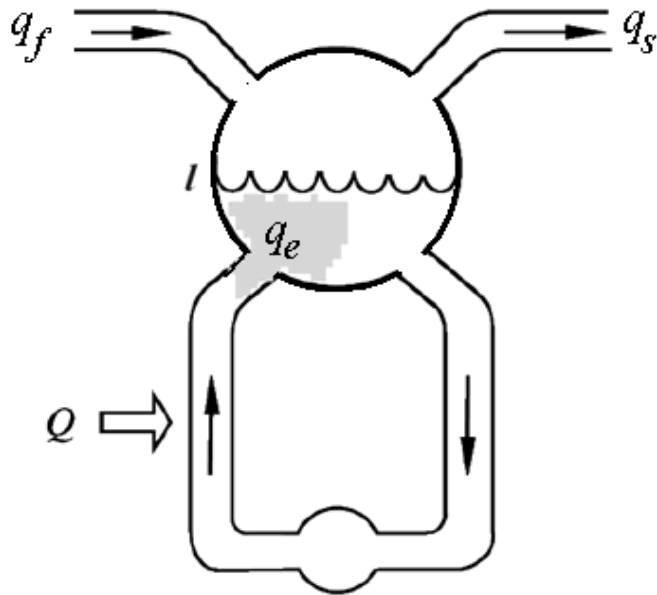
Supponendo di aver trovato \mathbf{u}_1 tale che soddisfi i vincoli di uguaglianza e di aver verificato che soddisfa anche quelli di disuguaglianza, viene allora effettuato un test per stabilire se il punto è quello di ottimo, il quale consiste nel controllare se tutti i moltiplicatori di Lagrange associati ai vincoli di uguaglianza sono positivi ($\lambda_i \geq 0, \forall i$). Se non lo sono, i vincoli che hanno il moltiplicatore di Lagrange più negativo vengono eliminati dal sistema di vincoli di uguaglianza e il passo viene ripetuto. Se invece \mathbf{u}_1 è ammissibile per i vincoli di uguaglianza ma non per quelli di disuguaglianza, viene creata una nuova regione ammissibile, con l'intersezione dei vincoli di disuguaglianza e il segmento che collega i punti \mathbf{u}_1 e \mathbf{u}_0 . Con questa nuova area vengono ripetuti i calcoli per la ricerca del punto di ottimo.

5 Progettazione e simulazione di un controllore GPC

In questo capitolo verrà fornita un'implementazione pratica di un controllore predittivo GPC le cui caratteristiche si apprestano ad essere ottimali per il seguente tipo di impianto: si vuole effettuare un controllo di livello in un *drum-boiler* (ossia un generatore di vapore).

5.1 Descrizione dell'impianto

Il *drum-boiler* ha trovato un grande utilizzo nelle centrali termoelettriche, le quali ovviamente necessitano di un flusso di vapore costante per far muovere le turbine e generare così energia elettrica. La sua importanza si riscontra anche nella grande quantità di articoli che ne trattano il funzionamento e la modellizzazione ai fini del controllo, infatti la sua complessità ha richiesto anni di studi per fornire modelli sempre più precisi per simulare la risposta dell'impianto e progettare il sistema di controllo. Tralasciando la trattazione teorica del funzionamento dell'impianto (la quale viene descritta molto bene in [3]) nella pratica, come si può vedere in figura, il calore Q fornito al fascio tubifero surriscalda l'acqua portandola in ebollizione, facendo così risalire il vapore formato verso l'alto. Nella figura l rappresenta il valore del livello equivalente d'acqua nel corpo cilindrico, q_f il flusso d'acqua di alimentazione in ingresso, q_s il valore nominale del flusso di vapore in uscita richiesto dalla turbina e q_e è una portata equivalente di mistura acqua-vapore proveniente dal fascio tubifero in risalita.



Si vuole quindi cercare di mantenere costante il livello dell'acqua presente nel cilindro, rispetto alle variazioni della richiesta di vapore q_s . Il problema principale nel controllo di questo tipo di sistemi è dato dal fenomeno dello *shrink and swell*: se aumenta la richiesta di vapore, avviene un abbassamento della pressione nel cilindro la quale causa un innalzamento del livello del liquido e un ribollito dato dalla presenza di vapore sotto la superficie del pelo d'acqua. Il sistema di controllo tende quindi a diminuire il flusso di alimentazione, anche se data la maggiore richiesta di vapore, dovrebbe aumentarlo. Viceversa diminuendo la richiesta di vapore, aumenta la pressione e il livello si abbassa. Il compito principale del controllore consiste quindi nell'evitare innalzamenti del livello d'acqua nel cilindro che nella peggiore delle ipotesi potrebbero portare allo straripamento del serbatoio ed andare a rovinare le turbine, oppure lo svuotamento totale che causerebbe un sovrariscaldamento del fascio tubifero rovinandolo o portandolo alla rottura. Inoltre i tempi di risposta di sistemi di questo genere sono espressi in minuti, quindi il più delle volte risulta difficoltoso il controllo con metodi classici, i quali effettuano correzioni sugli errori passati impossibili da stabilire nel breve tempo, va ricordato tuttavia che buone prestazioni si possono ottenere tramite l'ausilio della tecnica *three-element* basata sulla progettazione di controllori PID.

5.2 Descrizione del modello

Per il progetto del sistema di controllo è stato utilizzato un modello lineare semplificato, ottenuto tramite tecniche di identificazione simulando un modello più complesso come quello descritto in [3]:

$$\hat{l}(s) = \frac{1}{T_1 s} [\hat{q}_f(s) - \hat{q}_e(s)]$$

$$\hat{q}_f(s) = \frac{1}{(1 + \tau s)(1 + T_v s)} \hat{u}(s) + \hat{d}(s)$$

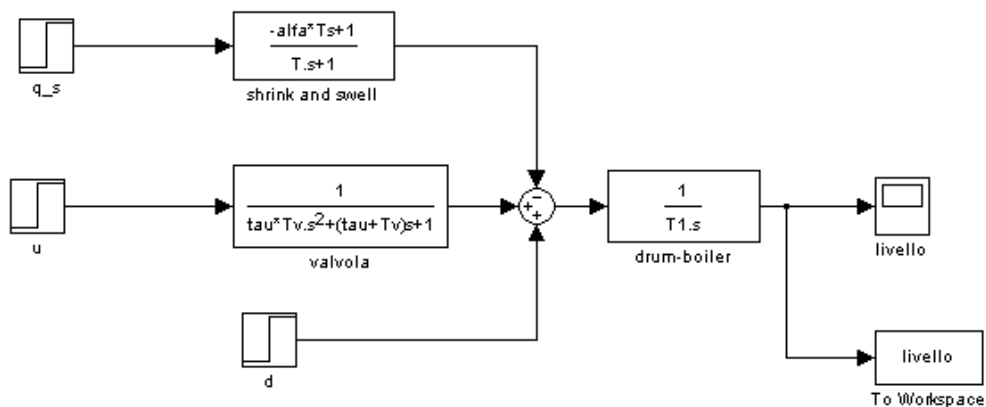
$$\hat{q}_e(s) = \frac{1 - \alpha T s}{1 + T s} \hat{q}_s(s)$$

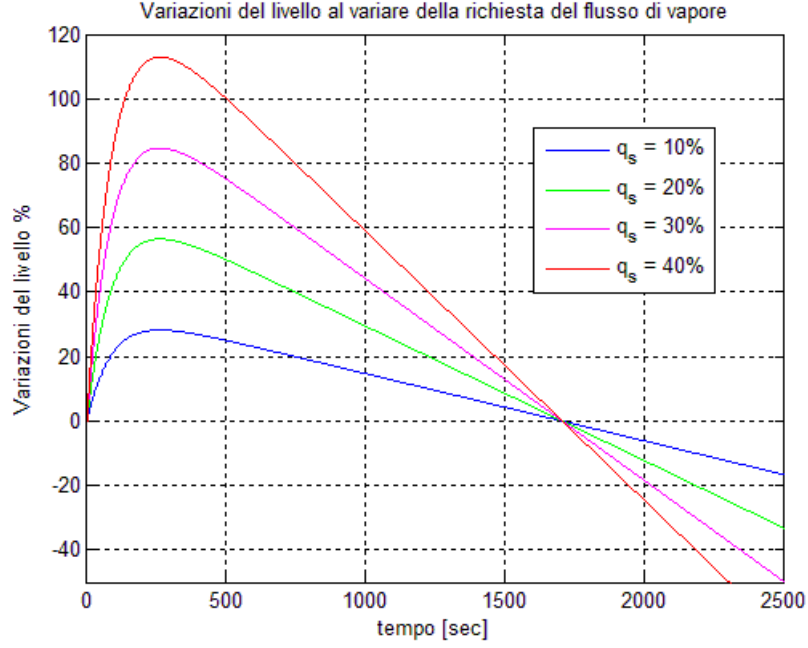
dove le variabili $\hat{l}(s)$, $\hat{q}_f(s)$, $\hat{q}_e(s)$ e $\hat{q}_s(s)$ sono già state illustrate in precedenza, $\hat{u}(s)$ rappresenta il valore dell'azione di comando applicata alla valvola di alimentazione mentre $\hat{d}(s)$ rappresenta i valori dei disturbi che descrivono ad esempio variazioni di flusso d'acqua di alimentazione dovuti a cambiamenti imprevedibili di portata delle pompe di alimentazione ed errori di modellizzazione di varia natura. Le costanti di tempo τ e T_v si riferiscono alla valvola di alimentazione e al ritardo di trasporto dell'acqua di alimentazione. Il termine $1 - \alpha T s$ tiene conto dell'effetto *shrink and swell*. I valori nominali del modello sono:

$$\tau = 1.5[\text{min}], \quad T_v = 8[\text{min}], \quad T_1 = 8[\text{min}], \quad T = 1.5[\text{min}], \quad \alpha = 18$$

Sebbene questo è un modello semplificato che trascurava molti aspetti, come ad esempio la relazione tra q_s e Q o la variazione di pressione all'interno del corpo cilindrico, risulta essere più che valido per la progettazione del sistema di controllo. Le variabili in gioco sono considerate come variazioni percentuali rispetto al loro valore nominale, il quale rappresenta un punto di lavoro dell'impianto. Il controllore dovrà quindi cercare di mantenere il livello al suo valore nominale cioè $\hat{l}(s) = 0$ che in questo caso sarà considerato a metà nel corpo cilindrico.

Prima di iniziare con la progettazione del controllore è stata effettuata una simulazione sul modello descritto dalle equazioni precedenti, mantenendo il valore del riferimento a zero e cambiando la richiesta di vapore q_s con variazioni del 10% rispetto al valore nominale del punto di lavoro. Lo schema in figura rappresenta il modello implementato in Simulink con cui sono state effettuate le prove di simulazione.





Questa figura ci mostra invece, come per piccole variazioni di q_s si hanno elevate variazioni del livello d'acqua all'interno del corpo cilindrico. Infatti come si può notare per $q_s = 40\%$ il livello aumenta del 115% provocando l'inevitabile fuoriuscita dell'acqua al suo interno. Inoltre mantenendo inalterato il flusso dell'acqua di alimentazione q_f si avrà uno svuotamento del corpo cilindrico e quindi l'esaurimento progressivo del vapore generato richiesto dalle turbine.

5.3 Discretizzazione del modello

Come si può intuire dalla figura precedente, questi tipi di impianti hanno al suo interno modi molto lenti, che ne determinano tempi di salita, di risposta e di assestamento dell'ordine delle decine di minuti. Per una buona scelta del tempo di campionamento si vogliono avere 10-20 campioni nel tempo di assestamento, che equivale circa a tre volte la costante di tempo più grande presente nel modello, in questo caso $3T_v = 24[\text{min}] = 1440[\text{sec}]$. Ora essendo $3T_v/20 = 72[\text{sec}]$ come tempo di campionamento è stato scelto $T_c = 60[\text{sec}]$. Anche se come tempo di campionamento potrebbe sembrare elevato, bisogna tenere presente che il controllore al suo interno implementa una funzione di ottimizzazione di un problema QP con vincoli, la quale richiede un certo numero di iterazioni e quindi un certo tempo per trovare una soluzione ottima. Tuttavia sono state eseguite delle prove anche con un tempo di campionamento inferiore $T_c = 20[\text{sec}]$ ed in seguito sono stati commentati i risultati ottenuti durante le simulazioni con i due diversi tempi di campionamento utilizzati.

Per descrivere il sistema con un modello del tipo CARIMA previsto per l'implementazione del controllore GPC, sono state considerate le equazioni che ne descrivono il modello approssimato:

$$\begin{aligned} \hat{l}(s) &= \frac{1}{T_1 s} [\hat{q}_f(s) - \hat{q}_e(s)] = \frac{1}{T_1 s} \left[\frac{1}{(1 + \tau s)(1 + T_v s)} \hat{u}(s) - \frac{1 - \alpha T s}{1 + T s} \hat{q}_s(s) \right] \\ &= \frac{(1 + T s) \hat{u}(s) - (1 + \tau s)(1 + T_v s)(1 - \alpha T s) \hat{q}_s(s)}{T_1 s(1 + \tau s)(1 + T_v s)(1 + T s)} \end{aligned}$$

e una volta effettuata la discretizzazione si è ottenuto

$$l(t) = Z[\hat{l}(s)] = \frac{B(z^{-1})u(t-1) - D(z^{-1})q_s(t-1)}{A(z^{-1})}$$

mentre il disturbo $\hat{d}(s)$ per semplificare i calcoli, senza però avere perdita di informazioni, è stato trattato come un rumore bianco di media nulla. La funzione di trasferimento dalla quale è stato implementato il controllore GPC ottenuta è

$$A(z^{-1})l(t) = B(z^{-1})u(t-1) + D(z^{-1})q_s(t-1) + \frac{d(t)}{\Delta}$$

La discretizzazione del modello è stata effettuata con il metodo zero-holder hold il quale viene descritto dalle seguenti equazioni

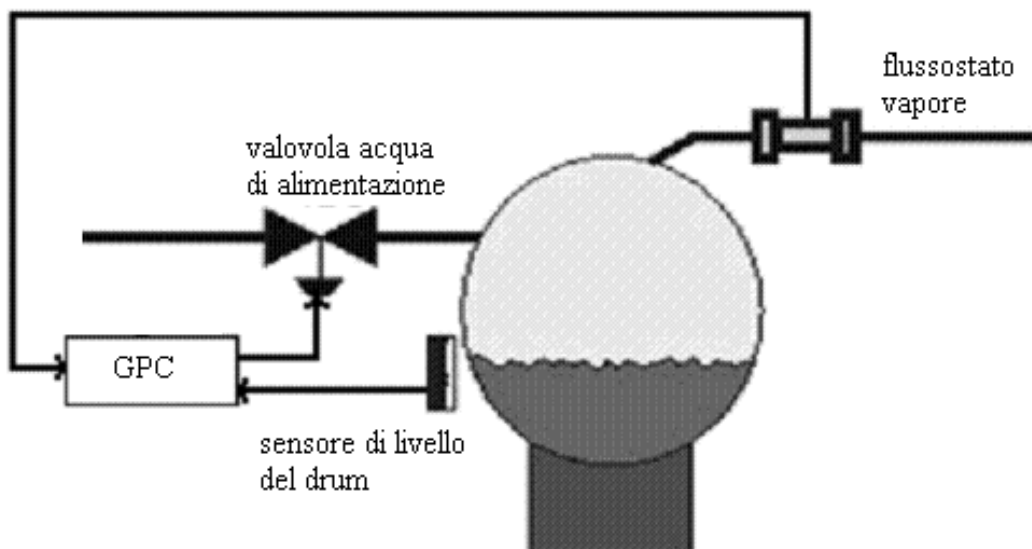
$$G(s) = H_0(s)G'(s)$$

$$G(z) = Z\left[\frac{(1 - e^{-sT_c/2})}{s}G'(s)\right] = (1 - z^{-1})Z\left[\frac{G'(s)}{s}\right]$$

dove $G'(s)$ è la funzione di trasferimento che si vuole discretizzare e $H_0(s)$ è l'interpolatore di ordine zero (Holder). In Matlab, la discretizzazione di un sistema continuo sys in uno discreto $sysd$, con questo metodo, viene effettuata con il comando $sysd = c2d(sys, T_c, 'zoh')$, con T_c tempo di campionamento.

5.4 Implementazione del controllore GPC

Il controllore GPC, come si può vedere dall'impianto schematizzato, dovrà effettuare un controllo di livello dell'acqua all'interno del corpo cilindrico, comandando la valvola dell'acqua di alimentazione. Il flusso di vapore richiesto dalle turbine verrà trattato come un disturbo misurabile con le modalità viste nel paragrafo 4.5, mentre il livello come uscita misurata.



Una volta ottenuto il modello CARIMA, si passa alla creazione del controllore GPC, dove qui di seguito è stato fornito e commentato il codice principale implementato in Matlab.

```
%% Dati modello

global A As d N Rif Yp qss delta lambda;

As = poly([roots(A)' roots([1 -1])]); % polinomio  $A(z^{-1})*(1-z^{-1})$ 

d = 0; % Ritardo
N = 0; % Orizzonte di predizione

Yp = zeros(length(A),1); % vettore uscite passate
Rif = zeros(N, 1); % vettore traiettoria di riferimento
qss = 0; %  $q_s(t-1)$ 

delta = 1; % regola la velocità del sistema
lambda = 1; % regola i consumi del segnale di controllo
```

Questa prima parte di codice definisce alcune delle variabili fondamentali per la costruzione del controllore e permette una rapida consultazione e modifica di alcuni parametri durante le varie simulazioni. Le variabili *delta* e *lambda* sono i coefficienti $\delta(j)$ e $\lambda(j)$ appartenenti alla funzione obiettivo da ottimizzare per il calcolo delle uscite, in questo caso sono stati definiti costanti lungo tutto l'orizzonte di predizione, la variabile *d* rappresenta il ritardo del sistema che in questo caso è 0 mentre la variabile *N*, che rappresenta l'orizzonte di predizione, è stata scelta in base al tempo di campionamento utilizzato. Il polinomio *As* è definito come $\tilde{A}(z^{-1}) = A(z^{-1})\Delta$ necessario per il calcolo delle matrici tramite l'ausilio dell'equazione Diofantea come spiegato nel capitolo 4, di cui il codice in Matlab è:

```
global G1 G B F dU H1 H Q qpas delta lambda ;

%% Prima iterazione per il calcolo di  $y_s(t+d+1|t)$ 

[E R] = divisione(As, 1, d+1);

lA = length(A); % numero di termini di A
g1 = poly([roots(E)' roots(B)'])*B(1); % polinomio g1
G = eye(N)*g1(1); % matrice G
G1 = [g1(2:length(g1)); zeros(N-1, length(g1)-1)]; % matrice G'
F = [R; zeros(N-1, lA)]; % matrice F

h1 = poly([roots(E)' roots(D)'])*D(1); % polinomio h
H = eye(N)*h1(1); % matrice H
H1 = [h1(2:length(h1)); zeros(N-1, length(h1)-1)]; % matrice H'

%% iterazioni successive per il calcolo di  $y_s(t+d+i|t)$  con  $d+1 < i \leq N+d$ 

for (i=2:N)

    [e R] = divisione(As, R, 1);
    E = [E e];

    gi = poly([roots(E)' roots(B)'])*B(1); % polinomio gi
    j = length(gi);
    G = G + [[zeros(i-1, N-i+1); eye(N-i+1)*gi(i)] zeros(N, i-1)];
```

```

G1 = G1 + [zeros(i-1, j-i); gi(i+1:j); zeros(N-i, j-i) ];
F = F + [zeros(i-1, lA); R; zeros(N-i, lA)];

hi = poly([roots(E)' roots(D)'])*D(1);          % polinomio hi
k = length(hi);
H = H + [[zeros(i-1, N-i+1); eye(N-i+1)*hi(i)] zeros(N, i-1)];
H1 = H1 + [zeros(i-1, k-i); hi(i+1:k); zeros(N-i, k-i) ];

end

%% calcolo di Q per la soluzione del problema QP

Q = 2*(delta*G'*G + eye(N)*lambda);

%% vettori

dU = zeros(length(g1)-1,1);          % vettore segnali di comando passati
qpas = zeros(length(h1)-1,1);        % vettore delle variazioni di q_s passate

```

Questa parte di codice implementa la parte definita “fuori linea” (si vedano i paragrafi 4.3, 4.4 e 4.5), dove vengono create tutte le matrici ed i vettori necessari per la funzione “ottimizzatrice” che è l’effettivo cuore del controllore. La funzione

$$[Q(z^{-1}) \ R(z^{-1})] = \text{divisione}(P_1(z^{-1}), P_2(z^{-1}), n)$$

restituisce il quoziente e il resto della divisione tra i polinomi $P_1(z^{-1})$ e $P_2(z^{-1})$ rispettivamente di grado n_1 e n_2 , in modo tale da ottenere

$$\frac{P_2(z^{-1})}{P_1(z^{-1})} = Q(z^{-1}) + \frac{z^{-n}R(z^{-1})}{P_1(z^{-1})}$$

dove $Q(z^{-1})$ e $R(z^{-1})$ hanno rispettivamente grado $n - 1$ e $n_1 - 1$.

L’ultima parte di codice riguarda la funzione ottimizzatrice, la quale riceve in ingresso i valori percentuali delle variabili $q_s(t)$ (variazione della richiesta di vapore), $l(t)$ (variazione del livello dell’acqua all’interno del corpo cilindrico) e restituisce il segnale di comando da inviare alla valvola. Inoltre è previsto un altro ingresso nel caso si volesse modificare il livello nominale del corpo cilindrico in base ad un comando di riferimento esterno. Il codice della funzione *ottimizzatore* è il seguente:

```

function du = ottimizzatore(v)

global  Rif Yp F G1 dU N H H1 qpas Q G qsS delta

qsF = [v(1) - qsS; zeros(N-1,1)];          % variazioni del flusso di
                                           % vapore future
qsS = v(1);
Rif = [ Rif(2:length(Rif)); v(2)];          % aggiornamento traiettoria di
                                           % riferimento
Yp = [v(3); Yp(1:length(Yp)-1)];          % aggiornamento uscite passate

f = G1*dU + F*Yp + H*qsF + H1*qpas;        % evoluzione passata del sistema
b = 2*delta*[(f-Rif)'*G]';                % termine q' per la funzione di
                                           % ottimizzazione

```



```

deltaU =
    quadprog(Q,b',G,ones(length(Rif),1),[],[],ones(length(Rif),1),
                                                    ones(length(Rif),1));

dU = [deltaU(1); dU(1:length(dU)-1)];           % aggiornamento segnali di
                                                    comando passati
qpas = [qsf(1); qpas(1:length(qpas)-1)];         % aggiornamento valori del
                                                    flusso di vapore passati

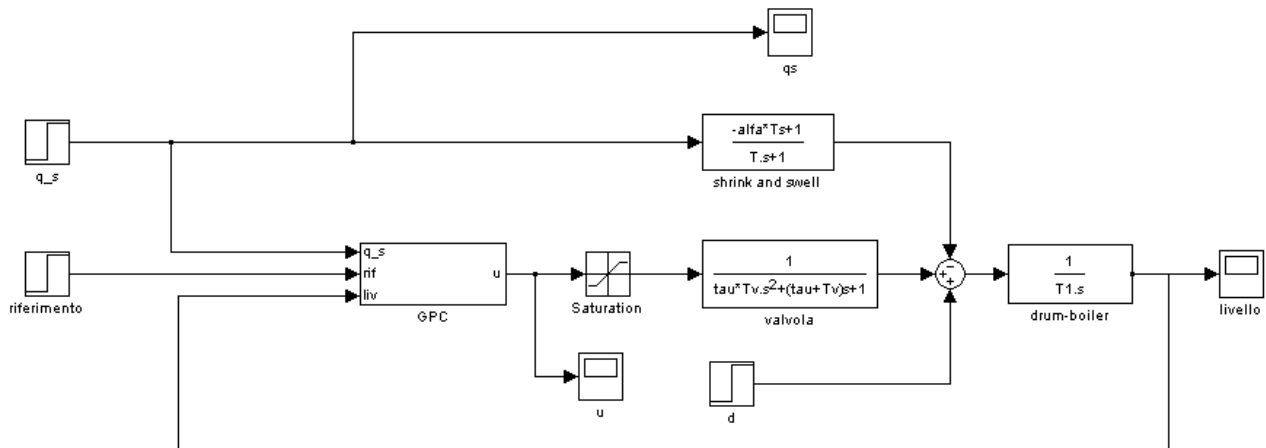
du = deltaU(1);                                   % segnale di comando da inviare
                                                    alla valvola

```

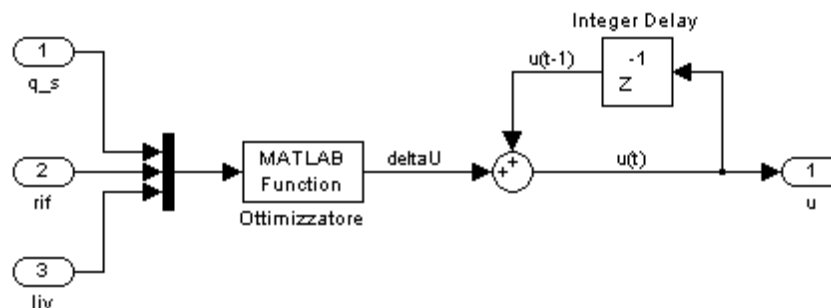
Come si può vedere, viene implementata la funzione *quadprog* (si veda il paragrafo 4.6) in cui sono stati inseriti i vincoli:

$$\begin{aligned}
 -1 &\leq u \leq 1 \\
 -1 &\leq y \leq 1 \rightarrow -1 \leq Gu \leq 1
 \end{aligned}$$

In fine sono stati creati gli schemi in Simulink tramite i quali sono poi state effettuate le simulazioni



Questo schema mostra come il controllore e l'impianto vengono collegati, è stato inserito inoltre un blocco di saturazione per rendere più reali le caratteristiche della valvola, il quale satura l'azione del segnale di comando nel caso in cui $|u| > 1$. Mentre il controllore GPC è stato così costruito:



La funzione *ottimizzatore* implementata in Matlab restituisce $\Delta u(t)$, è necessario quindi ricavare il segnale di comando in questo modo $u(t) = \Delta u(t) + u(t - 1)$.

5.5 Simulazioni

Qui di seguito sono stati riportati i risultati ottenuti dalle simulazioni eseguite con due periodi di campionamento scelti precedentemente, ipotizzando una variazione massima di $q_s = 20\%$.

5.5.1 Simulazioni con tempo di campionamento $T_c = 60$ [sec]

I polinomi del modello discreto ottenuto con $T_c = 60$ [sec] sono:

$$A(z^{-1}) = 1 - 2.9z^{-1} + 3.1z^{-2} - 1.4z^{-3} + 0.2z^{-4}$$

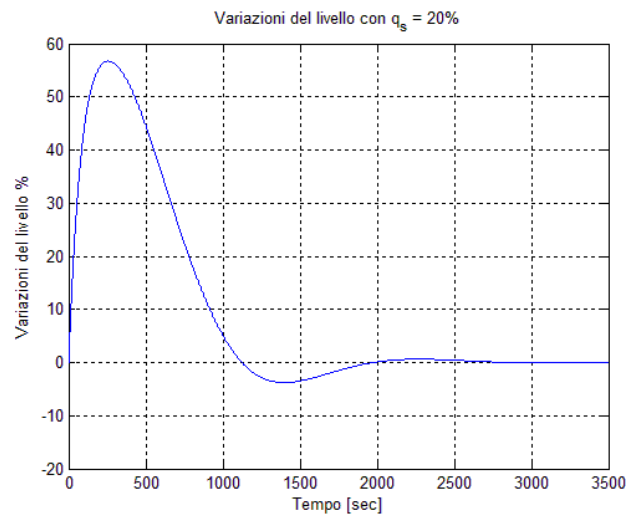
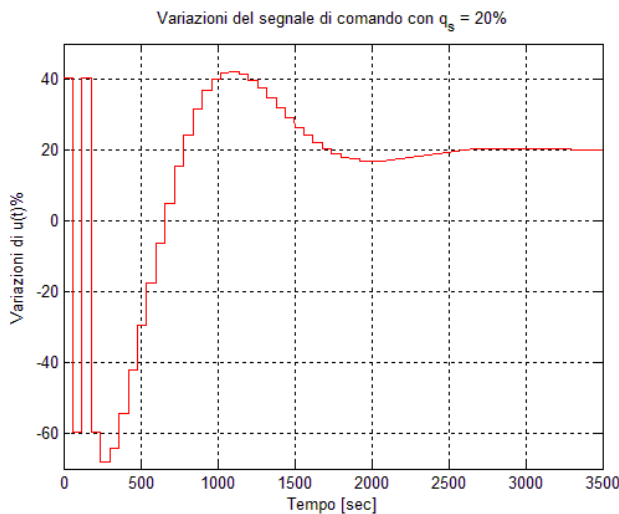
$$B(z^{-1}) = 0.0014 + 0.0040z^{-1} - 0.0015z^{-2} - 0.0005z^{-3}$$

$$D(z^{-1}) = 1.6 - 3.9z^{-1} + 3.1z^{-2} - 0.8z^{-3}$$

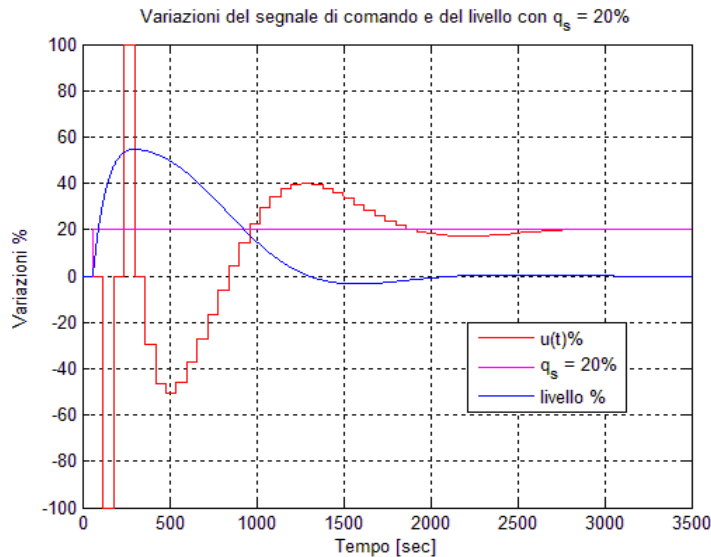
Il ritardo d è stato posto uguale a 0, mentre l'orizzonte di predizione N è stato scelto in modo tale che, tenendo in considerazione anche il tempo di campionamento, fosse uguale a 10 volte la costante di tempo più grande del modello considerato:

$$N = \frac{10 \times T_v}{T_c} = 80$$

Gli ultimi due parametri da fissare, necessari per l'ottimizzazione della funzione obiettivo, sono stati tarati durante le simulazioni iniziali. La migliore risposta, come si vede nelle figure sottostanti è stata ottenuta con $\delta = 1$ e $\lambda = 0.9$.



Come si può notare, l'innalzamento improvviso del livello nel corpo cilindrico causato dall'effetto *shrink and swell*, non viene attenuato se non di pochi valori percentuali. Il controllore però fornisce un'azione di comando sufficientemente buona per portare il livello al suo valore nominale, in un tempo piuttosto rapido considerando il tipo di impianto e le sue costanti di tempo. Inoltre l'onere computazionale richiesto è piuttosto basso in quanto le matrici utilizzate per la soluzione del problema con la funzione *quadprog*, hanno come dimensione massima 80×80 e una volta avviata, la simulazione termina in poche decine di secondi. Con un tempo di campionamento così elevato però, potrebbe accadere che il valore di q_s cambi, nel peggiore dei casi, un istante dopo il campionamento del segnale e quindi che il sistema agisca indisturbato senza un'azione di comando per un tempo pari a $2T_c = 120$ [sec]. Non essendo un problema da sottovalutare è stata effettuata una simulazione facendo partire il gradino che simula q_s al tempo $t = 60.1$ [sec], di cui il risultato ottenuto è mostrato nella prossima figura.



Anche se inizialmente il segnale di controllo oscilla un po' si nota come il controllore in pochi passi riesca a formulare una legge di controllo che riporta il livello al valore nominale con un tempo di assestamento uguale al caso precedente.

5.5.2 Simulazioni con tempo di campionamento $T_c = 20$ [sec]

I polinomi del modello discreto ottenuto con $T_c = 20$ [sec] sono:

$$A(z^{-1}) = 1 - 3.6z^{-1} + 4.7z^{-2} - 2.8z^{-3} + 0.6z^{-4}$$

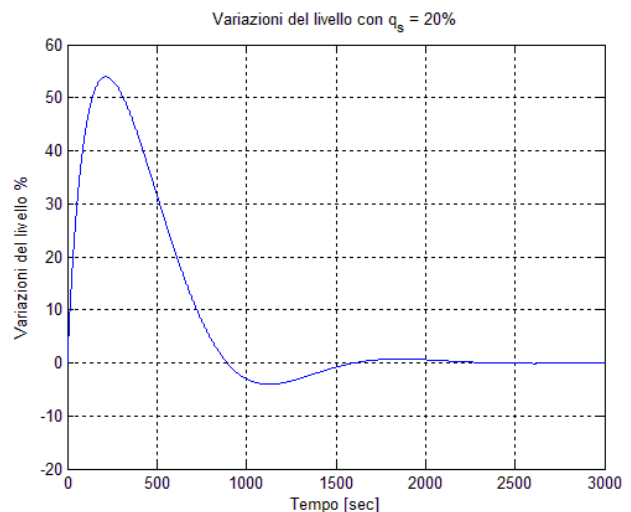
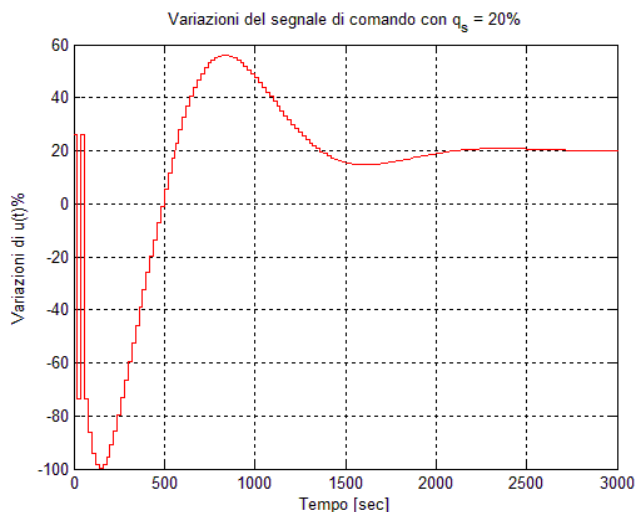
$$B(z^{-1}) = 0.06 \times 10^{-3} + 0.18 \times 10^{-3}z^{-1} - 0.13 \times 10^{-3}z^{-2} - 0.04 \times 10^{-3}z^{-3}$$

$$D(z^{-1}) = 0.67 - 1.85z^{-1} + 1.7z^{-2} - 0.52z^{-3}$$

mentre l'orizzonte di predizione è

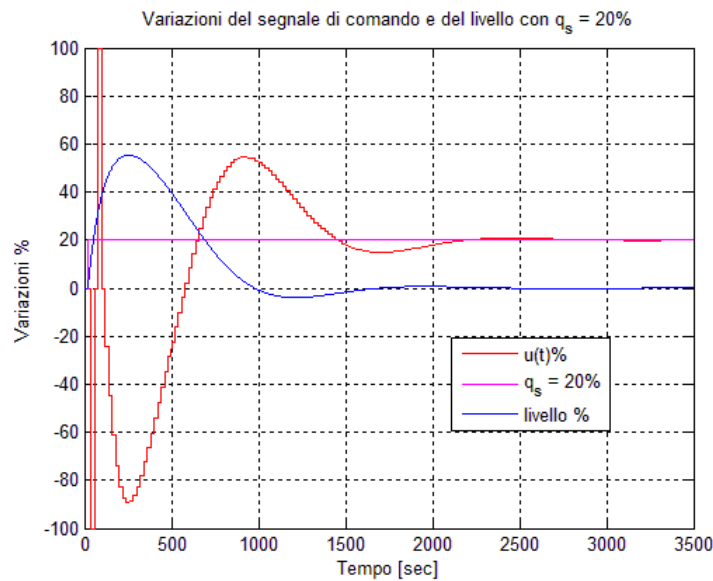
$$N = \frac{10 \times T_v}{T_c} = 240$$

Dopo alcune simulazioni, sono stati fissati $\delta = 0.32$ e $\lambda = 0.98$ e si sono così ottenuti gli andamenti del segnale di controllo e del livello all'interno del corpo cilindrico mostrati nelle prossime figure.



L'azione di comando in questo caso risulta molto più accentuata rispetto al caso precedente, ma questo oltre a diminuire il tempo di assestamento tende ad attenuare meglio l'effetto iniziale dovuto al fenomeno *shrink and swell*. Ottenere prestazioni migliori dall'impianto però, ha comportato un aumento dell'onere computazionale da parte del controllore molto più elevato rispetto al caso precedente, infatti la ricerca di un punto di ottimo da parte della funzione *quadprog* ha rallentato visibilmente i tempi delle simulazioni (circa $1.5 / 2 [min]$) basta considerare il fatto che le matrici utilizzate per l'ottimizzazione del segnale di comando hanno una dimensione 3 volte superiore rispetto al caso precedente e questo non è un problema trascurabile.

Con la stessa procedura di prima è stata effettuata una simulazione nel caso in cui q_s variesse un istante dopo il campionamento del controllore cioè a $t = 20.1 [sec]$, ottenendo la risposta illustrata in figura.



In questo caso il controllore non riesce ad attenuare l'effetto iniziale dello *shrink and swell*, tuttavia riesce a riportare il livello al suo valore nominale in un tempo ragionevole anche se l'azione di comando, in termini energetici risulta più dispendiosa.

6 Bibliografia

- [1] E. F. Camacho, C. Bordons, *Model Predictive Control*, Springer, 2007
- [2] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2000
- [3] Åstrom-Bell, *Automatica*, 36, (2000) 363-378
- [4] D. Ciscato, *Appunti di controllo digitale*. Libreria Progetto, Padova, 2007