

Motore in corrente continua

Controllo in retroazione dello stato e Osservatore dello stato

Controllo ottimo

Esercitazioni di Controlli Automatici LS (Prof. C. Melchiorri)

Si consideri il motore elettrico in corrente continua a magneti permanenti rappresentato in Figura 1:

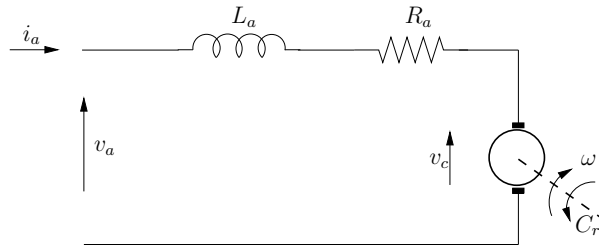


Figure 1: Motore elettrico in corrente continua.

Assumendo come ingresso v_a , come stato i_a , ω e ϑ , come uscita ϑ , le eq. di stato del sistema sono:

$$\begin{bmatrix} \frac{di_a(t)}{dt} \\ \frac{d\omega(t)}{dt} \\ \frac{d\vartheta(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_m}{L_a} & 0 \\ \frac{K_c}{J} & -\frac{b}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a(t) \\ \omega(t) \\ \vartheta(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} [v_a(t)]$$

$$[\vartheta(t)] = [0 \ 0 \ 1] \begin{bmatrix} i_a(t) \\ \omega(t) \\ \vartheta(t) \end{bmatrix}$$

Considerando come valori numerici:

$$R_a = 1, \quad L_a = 0.1, \quad K_c = 0.1$$

$$J = 10, \quad b = 1, \quad K_m = 0.1$$

si ottengono le seguenti matrici dinamiche:

$$A = \begin{bmatrix} -10 & -1 & 0 \\ 0.01 & -0.1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix}, \quad C = [0 \ 0 \ 1],$$

1. Verificare la raggiungibilità, l'osservabilità e la scomposizione di Kalman del sistema (utilizzare le funzioni matlab `ctrb`, `obsv`, `mininv`, `maxinv` e `kalmcd`);
2. Studiare la stabilità asintotica, la stabilità ingresso limitato - stato limitato (i.l.s.l) e ingresso limitato - uscita limitata (i.l.u.l.) del sistema;
3. Si realizzi una retroazione statica dello stato tale che il sistema in retroazione presenti gli autovalori $\lambda_1 = -0.1$, $\lambda_2 = -0.2$ e $\lambda_3 = -10$ (utilizzare le funzioni matlab `place` o `acker`). Si verifichi la risposta del sistema per diverse condizioni iniziali;
4. Si realizzi una legge di controllo ottimo a tempo infinito che minimizzi la corrente assorbita, la velocità del motore e la tensione in ingresso. Si verifichi la risposta del sistema per diverse condizioni iniziali;

5. Dal momento che non si possiede un accesso alla misura della velocità e della corrente del motore si realizzi un osservatore dello stato e lo si inserisca nello schema di controllo;
6. Si progetti la legge di controllo ottima a minima energia tale da portare il sistema dallo stato iniziale $x_0 = [0, 0, 0]^T$ allo stato finale $x_f = [0, 0, 2]^T$ in $t = t_f = 1$ [s]. Si esegua la simulazione in simulink (vedere file allegati).
7. Si consideri il problema della regolazione del valore dell'uscita (inseguimento del setpoint);
8. Si implementi il regolatore progettato in ambiente RTAI e si confronti la risposta del sistema con quella di Matlab.

SOLUZIONE

Retroazione dello stato

Per il progetto della retroazione dello stato, si procede considerando il sistema dato dalle matrici A , B , C , D . La matrice K può quindi essere ricavata con il comando `K=place(A,B,autovalori)`.

Legge di controllo ottimo

Come richiesto dal testo, al fine di realizzare un controllore ottimo, si desidera minimizzare il seguente indice:

$$J = \int_0^{\infty} [x_1 \quad x_2 \quad x_3] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + Ru_1^2 dt$$

dove $R = 1$.

Come noto dalla teoria, la legge di controllo cercata sarà della forma:

$$\begin{aligned} u(t) &= Kx(t) \\ &= -R^{-1}B^T Sx(t) \end{aligned}$$

in cui la S è data dalla soluzione dell'equazione algebrica di Riccati (ARE):

$$SA + A^T S - SBR^{-1}S + Q = 0$$

Mediante il comando Matlab `care` si ottiene:

$$S = \begin{bmatrix} 0.0414 & 0.006 & 0 \\ 0.0006 & 4.9938 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

in cui si è considerata solo la prima colonna della matrice B .

Si ricava quindi la legge di controllo:

$$\begin{aligned} u(t) &= Kx'(t) \\ \dots &= -R^{-1}B^T Sx(t) \\ \dots &= [-0.4142 \quad -0.006 \quad 0]x(t) \end{aligned}$$

Inseguimento del setpoint

Si consideri il problema di controllo con retroazione dello stato con setpoint non-nullo. Nel caso in esame il sistema si comporta da servomeccanismo in quanto si hanno un ingresso ed una uscita. Il blocco di ingresso del sistema può essere quindi determinato tramite la relazione:

$$K_i = -[C(A + BK)^{-1}B]^{-1}$$

NOTA: Il blocco K_i va calcolato tramite le matrici del sistema tempo continuo e non varia considerando il sistema a dati campionati.

Lo schema simulink del controllore con retroazione dello stato e inseguimento del setpoint è riportato in figura 2.

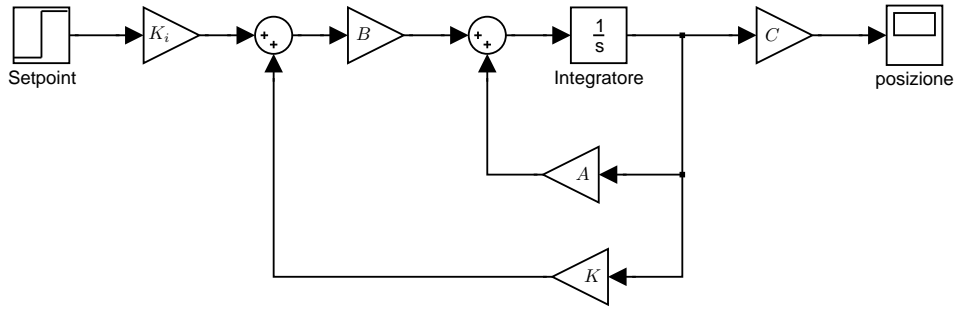


Figure 2: Schema del controllore con retroazione dello stato e inseguimento del setpoint.

Osservatore dello stato

Per il progetto dell'osservatore dello stato, si procede considerando il sistema duale A^T , C^T , B^T , D^T . A questo punto, il progetto della matrice di retroazione dell'osservatore si riduce al problema di sintesi della matrice di retroazione dello stato del sistema duale. La matrice L può quindi essere ricavata con il comando `L=place(A',C',autovalori)'`.

Gli autovalori assegnati al sistema tramite la legge di controllo ottimo possono essere ricavati tramite il comando matlab `eig(A+B*K)`. Lo schema completo del sistema di controllo con osservatore dello stato tempo continuo è riportato in figura 3.

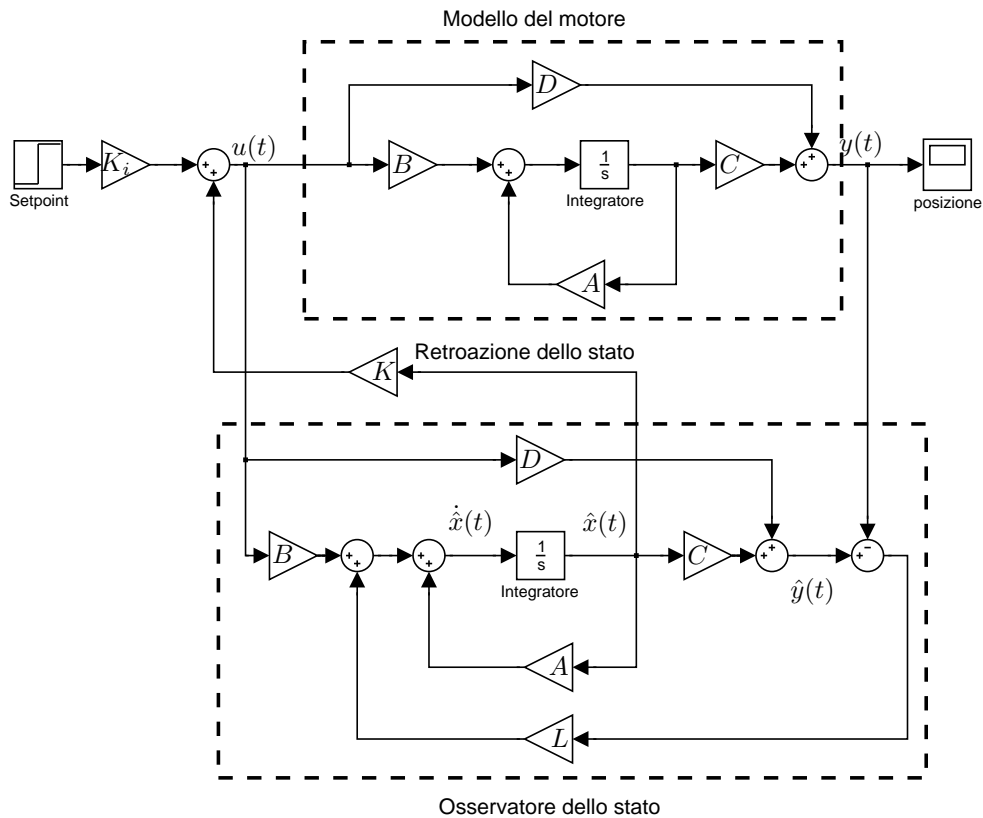


Figure 3: Schema completo del sistema di controllo con osservatore dello stato tempo continuo.

Controllo ottimo tempo finito: caso tempo continuo

L'indice di comportamento per problemi di controllo a minima energia del tipo:

$$J = \int_{t_0}^{t_f} u^T(t) R u(t) dt \quad (1)$$

Poniamo per semplicità $R = 1$. La funzione Hamiltoniana del sistema è:

$$H(t) = u^2(t) + \lambda^T(t)(Ax(t) + Bu(t)) \quad (2)$$

Sono noti lo stato iniziale x_0 , lo stato finale x_f e il tempo finale $t_f = 1$. Considerando le equazioni del sistema aggiunto e le condizioni di stazionarietà possiamo scrivere:

$$\dot{\lambda}(t) = -\frac{\partial H(t)}{\partial x(t)} = -A^T \lambda(t) \quad (3)$$

$$u(t) = -\frac{1}{2}B^T \lambda(t) \quad (4)$$

e sostituendo l'espressione di $u(t)$ nell'equazione dinamica del sistema:

$$\dot{x}(t) = Ax(t) - \frac{1}{2}BB^T \lambda(t) \quad (5)$$

Il sistema dinamico così ottenuto può essere riscritto nella forma:

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -\frac{1}{2}BB^T \\ 0 & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = H \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (6)$$

La dipendenza delle variabili dal tempo è stata omessa per semplicità. Il sistema (6) viene comunemente detto sistema hamiltoniano, da cui è possibile ricavare:

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = e^{Ht} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} \quad (7)$$

ed in particolare, valutando lo stato del sistema Hamiltoniano per $t = t_f$:

$$\begin{bmatrix} x_f \\ \lambda_f \end{bmatrix} = e^{Ht_f} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} = H_{t_f} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} \quad (8)$$

La matrice $H_{t_f} = e^{Ht_f}$ può essere calcolata in generale passando attraverso la forma di Jordan della matrice H . Scomponendo la matrice H_{t_f} in blocchi di dimensione opportuna, è possibile scrivere:

$$\begin{bmatrix} x_f \\ \lambda_f \end{bmatrix} = \begin{bmatrix} H_{t_f 11} & H_{t_f 12} \\ H_{t_f 21} & H_{t_f 22} \end{bmatrix} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} \quad (9)$$

Dati x_0 e x_f , è quindi possibile calcolare λ_0 e λ_f :

$$\lambda_0 = H_{t_f 12}^{-1}(x_f - H_{t_f 11}x_0) \quad (10)$$

$$\lambda_f = H_{t_f 21}x_0 + H_{t_f 22}\lambda_0 \quad (11)$$

Una volta noto λ_0 , dalla (7) possibile ricavare in modo analogo l'espressione di $\lambda(t)$:

$$\begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} = e^{Ht} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} = H_t(t) \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} = \begin{bmatrix} H_{t_{11}}(t) & H_{t_{12}}(t) \\ H_{t_{21}}(t) & H_{t_{22}}(t) \end{bmatrix} \begin{bmatrix} x_0 \\ \lambda_0 \end{bmatrix} \quad (12)$$

da cui:

$$x(t) = H_{t_{11}}(t)x_0 + H_{t_{12}}(t)\lambda_0 \quad (13)$$

$$\lambda(t) = H_{t_{21}}(t)x_0 + H_{t_{22}}(t)\lambda_0 \quad (14)$$

$$u(t) = -\frac{1}{2}B^T(H_{t_{21}}(t)x_0 + H_{t_{22}}(t)\lambda_0) \quad (15)$$

Nel controllo a minima energia, si ha solitamente $H_{t_{21}}(t) = 0$, per cui la legge di controllo ottima in catena aperta diviene:

$$u(t) = -\frac{1}{2}B^T H_{t_{22}}(t)\lambda_0 \quad (16)$$

Per la simulazione del sistema in ambiente Simulink, si vedano i file allegati.

Problema: Progettare la legge di controllo ottima a minima energia che, a partire dallo stato x_f in $t = t_f$, riporti il sistema nelle condizioni iniziali per $t = t_{f_2} = 2$ [s]. Eseguire la simulazione completa del sistema da $t = 0$ a $t = t_{f_2}$.

Discretizzazione del controllore e dell'osservatore

Una volta determinate le matrici di retroazione dello stato e dell'osservatore, si procede alla discretizzazione del sistema tramite il comando matlab `sysd=c2d(ss(A,B,C,D),Tc)`, dove T_c è il tempo di campionamento scelto sulla base dell'autovalore maggiore in modulo del sistema. La struttura `sysd` contiene quindi le matrici del sistema discretizzato $A_d=sysd.a$, $B_d=sysd.b$, $C_d=sysd.c$ e $D_d=sysd.d$.

In particolare le matrici C e D restano invariate mentre cambiano le matrici A e B .

Anche la matrice di retroazione dello stato K resta invariata ma cambia la matrice di retroazione dell'osservatore L .

Per la sintesi della matrice L_d con Matlab può essere utilizzato, anche in questo caso, il comando `sysd2=c2d(ss(A,L,C,D),Tc)`. La struttura `sysd2` contiene quindi la matrice di retroazione dell'osservatore discretizzata $L_d=sysd2.b$.

Lo schema simulink dell'osservatore dello stato tempo discreto è riportato in figura 4.

Condizionamento dei segnali di ingresso ed uscita

Allo scopo di riprodurre il più fedelmente possibile la risposta di un sistema di controllo reale utilizzando attraverso la simulazione in ambiente simulink, è necessario tenere esplicitamente in considerazione, nello schema del controllore, della scalatura, limitazione (saturazione) e discretizzazione sia delle variabili di ingresso che di quelle di uscita. Questi (ed altri) effetti, dovuti alla realizzazione pratica dei sistemi di misura e attuazione, vengono comunemente indicati con il termine "condizionamento" dei segnali di ingresso ed uscita.

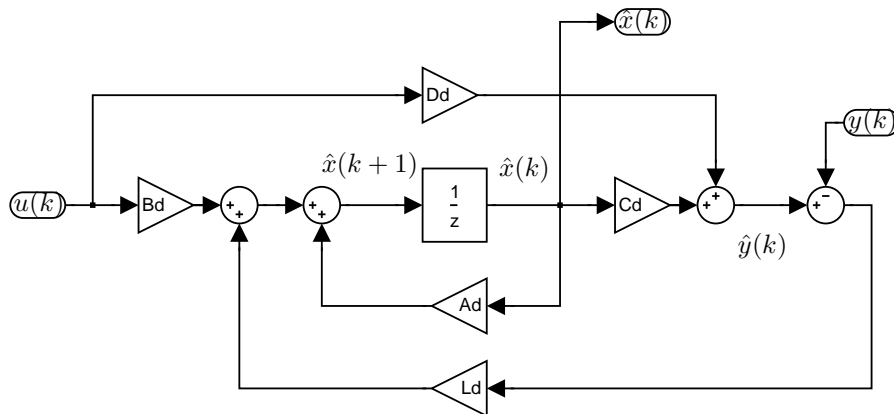


Figure 4: Schema di un generico osservatore tempo discreto.

In figura 5 viene riportato lo schema simulink del sistema di controllo del motore in corrente continua. In questa figura, i blocchi DAC e ADC, rappresentati in dettaglio in figura 6, effettuano il condizionamento dei segnali, mentre il blocco `Motore` contiene il modello dinamico tempo continuo del motore in corrente continua. Questi tre blocchi rappresentano come il sistema viene "visto" dal controllore real-time. Notare che la variabile di controllo viene saturata prima di essere messa in ingresso al sistema e all'osservatore dello stato. Per il sistema in esame, si assuma $K_{in}=1$ e $K_{out}=1$.

Implementazione del controllore

Viene esaminato il problema relativo all'implementazione in ambiente RTAI-Linux dell'osservatore e della retroazione dello stato per un generico sistema dinamico.

Con riferimento alle figure 4 e 5, l'algoritmo di controllo realtime, ad ogni istante di campionamento, dovrà calcolare l'innovazione, l'evoluzione dello stato dell'osservatore e l'azione di controllo. Viene riportato di seguito uno schema per la soluzione del problema.

```
...
/* discrete time controller output, plant input */
```

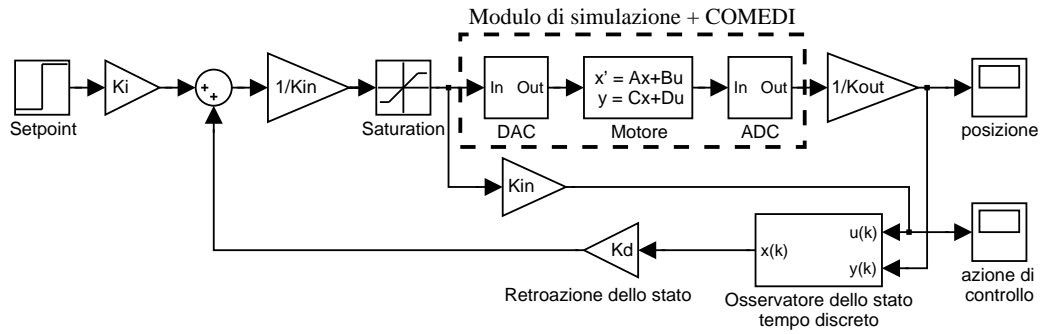


Figure 5: Schema di un generico osservatore tempo discreto.

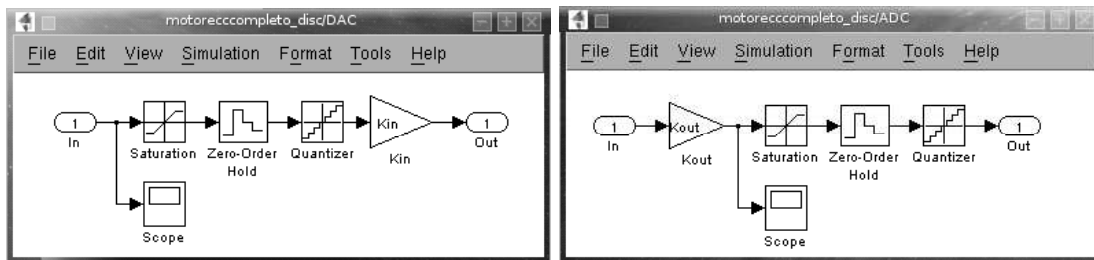


Figure 6: Schemi di scalatura, saturazione e discretizzazione delle variabili di ingresso ed uscita.

```

double dcmotor_ctrl_input [INPUT_DIM]={0.0};

/* plant output y(k) */
double dcmotor_ctrl_output [OUTPUT_DIM]={0.0};

/* discrete time observer matrices */

static double Ad[STATE_DIM] [STATE_DIM]={...};
static double Bd[STATE_DIM] [INPUT_DIM]={...};
static double Cd[OUTPUT_DIM] [STATE_DIM]={...};
static double Ld[STATE_DIM] [OUTPUT_DIM]={...};
static double Kd[INPUT_DIM] [STATE_DIM]={...};
static double Ki[INPUT_DIM] [OUTPUT_DIM]={...};
...

/* inizializzazione dello stato e del setpoint */
...

while(isrunning){

    /* calcolo dell'azione di controllo, da attuare al ciclo successivo */
    // u(k) = Kd*x(k) + ki*r(k)

    /* attesa dell'istante di campionamento */
    rt_task_wait_period();

```

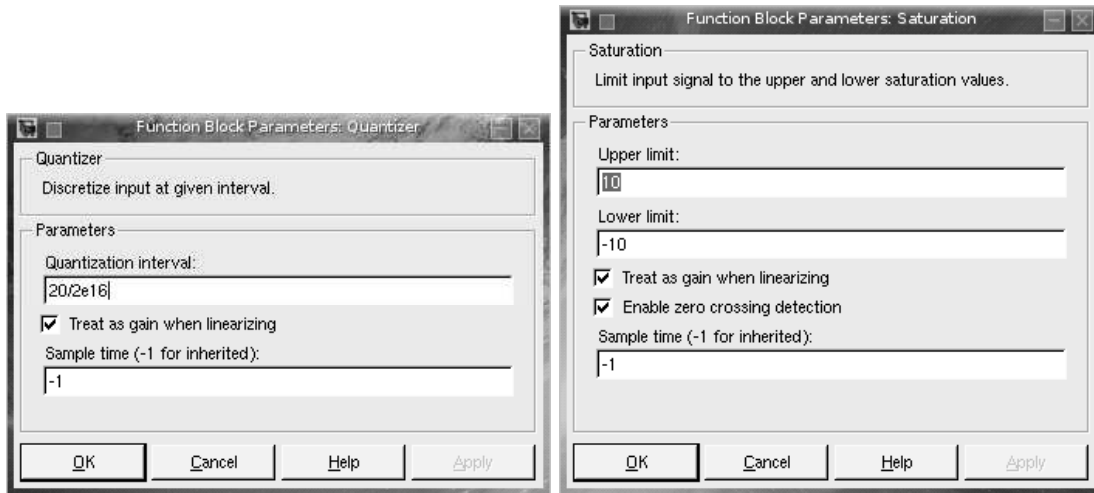


Figure 7: Parametri dei blocchi di scalatura e discretizzazione delle variabili di ingresso ed uscita.

```

/* attuazione del controllo u(k) */
for(i=0;i<INPUT_DIM;i++){
    /* scalatura delle variabili di controllo */
    // dcmotor_ctrl_input[i] = u[i]/kin[i];

    /* conversione da grandezza fisica a registro del DAC */
    ao_data=comedi_from_phys(dcmotor_ctrl_input[i],ao_rangetype,ao_maxdata);
    /* scrittura sul registro del DAC */
    comedi_data_write(cf,ao_subdev,ao_chan+i,ao_range,ao_aref,ao_data);

    /* saturazione e discretizzazione delle variabili di controllo */
    dcmotor_ctrl_input[i]=comedi_to_phys(ao_data,ao_rangetype,ao_maxdata);

    /* riscalatura delle variabili di controllo */
    // u[i] = dcmotor_ctrl_input[i]*kin[i];
}

/* lettura dell'output y(k) */
for(i=0;i<OUTPUT_DIM;i++){
    /* lettura del registro dell'ADC */
    comedi_data_read(cf,ai_subdev,ai_chan+i,ai_range,ai_aref,&ai_data);

    /* conversione da registro del ADC a grandezza fisica*/
    dcmotor_ctrl_output[i]=comedi_to_phys(ai_data,ai_rangetype,ai_maxdata);

    /* scalatura delle variabili di uscita */
    // y[i] = dcmotor_ctrl_output[i]/kout[i];
}

/* calcolo dell'evoluzione libera */
// x(k+1) = Ad*x(k)

/* calcolo dell'errore di inseguimento dello stato */
// e(k) = Cd*x(k) + Dd*u(k) - y(k)

/* calcolo dell'innovazione */

```

```
// x(k+1) += Ld*e(k)

/* calcolo dell'evoluzione forzata */
// x(k+1) += Bd*u(k)

/* aggiornamento dello stato */
// x(k) = x(k+1)

/* calcolo e aggiornamento dello setpoint */
// r(k+1) = ...
// r(k) = r(k+1)
}
...
```