

Simulazione dei Sistemi dinamici con Matlab-Simulink

**Modellazione di un satellite
con appendici flessibili**

Ing. Alessandro Pisano
`apisano@unica.it`

in cui:

$\omega(t) = \begin{bmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{bmatrix}$ è il vettore delle tre componenti della velocità angolare della parte rigida del satellite,

$\eta(t) = \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \eta_3(t) \end{bmatrix}$ contiene i modi oscillatori della parte flessibile,

$u(t) = \begin{bmatrix} u_x(t) \\ u_y(t) \\ u_z(t) \end{bmatrix}$ è il vettore che contiene le coppie applicate alla parte rigida del satellite,

$u_p(t) = \begin{bmatrix} u_{p1}(t) \\ u_{p2}(t) \\ u_{p3}(t) \end{bmatrix}$ sono le tensioni applicate agli attuatori piezoelettrici posizionati sulle appendici flessibili.

Le matrici del modello sono le seguenti

$$J = \begin{bmatrix} 420.8 & 3.6 & -4.2 \\ 3.6 & 410.6 & 9.4 \\ -4.2 & 9.4 & 690.7 \end{bmatrix} \quad \delta = \begin{bmatrix} 2.62 & 0.007 & -0.003 \\ -0.001 & 0.124 & -2.73 \\ -0.001 & 0.437 & -0.051 \end{bmatrix}$$

$$P = \begin{bmatrix} 70.26 & -4.23 & 2.34 \\ 4.8 & 31.93 & 1.24 \\ -1.05 & 2.55 & 29.84 \end{bmatrix} \quad M_{\omega}(t) = \begin{bmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix}.$$

$$C = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.3 \end{bmatrix} \quad K = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

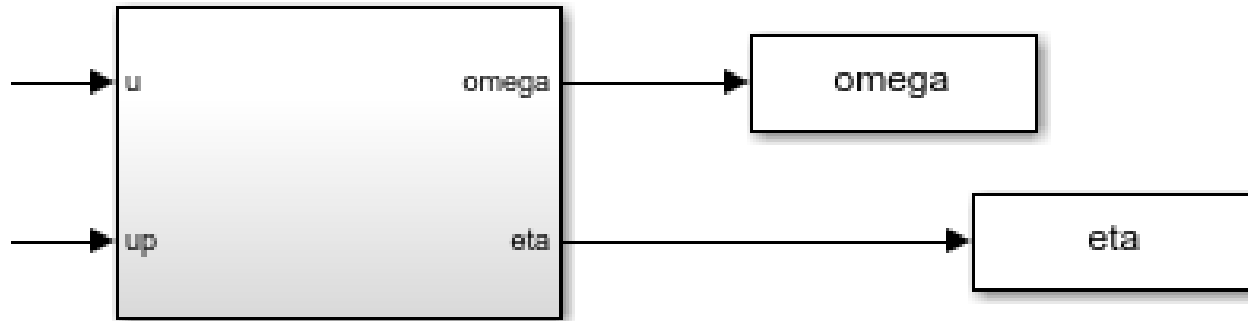
Realizzare un modello Simulink del sistema dinamico considerando le seguenti condizioni iniziali

$$\omega(0) = \begin{bmatrix} 0.1 \\ -0.1 \\ 0 \end{bmatrix} \quad \eta(0) = \begin{bmatrix} 0.01 \\ 0.02 \\ 0.03 \end{bmatrix} \quad \dot{\eta}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

ed applicando i seguenti vettori in ingresso:

$$u(t) = \begin{bmatrix} 10\cos(2t) \\ 5\sin(3t) \\ 20\sin(5t) \end{bmatrix} \quad u_P(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Il modello dovrà contenere un Subsystem che riceve in ingresso i vettori $u(t)$ ed $u_p(t)$ e produca in uscita i vettori $\omega(t)$ ed $\eta(t)$, come mostrato nella figura seguente



Simulare il sistema per 30 secondi con passo di campionamento $T_s = 0.01s$ e creare due grafici il primo dei quali mostri sovrapposte le tre velocità angolari ed il secondo le tre componenti del vettore $\eta(t)$

Realizzare uno script che avvii in automatico il modello Simulink e crei i due grafici richiesti.

Scrivere quindi una **function** che acquisisca in ingresso i vettori $\omega_x(t)$, $\omega_y(t)$ ed $\omega_z(t)$

esportati nel workspace e costruisca il vettore $\Omega(t) = \sqrt{\omega_x^2(t) + \omega_y^2(t) + \omega_z^2(t)}$.

Costruire quindi un terzo grafico che mostri l'evoluzione temporale del vettore $\Omega(t)$.

MODELLO IN FORMA ESPLICITA

$$J \dot{\omega}(t) + \delta^T \ddot{\eta}(t) + M_{\omega}(t) (J \omega(t) + \dot{\eta}(t)) = u(t)$$

$$\ddot{\eta}(t) + C \dot{\eta}(t) + K \eta(t) + \delta \omega(t) = P u_p(t)$$

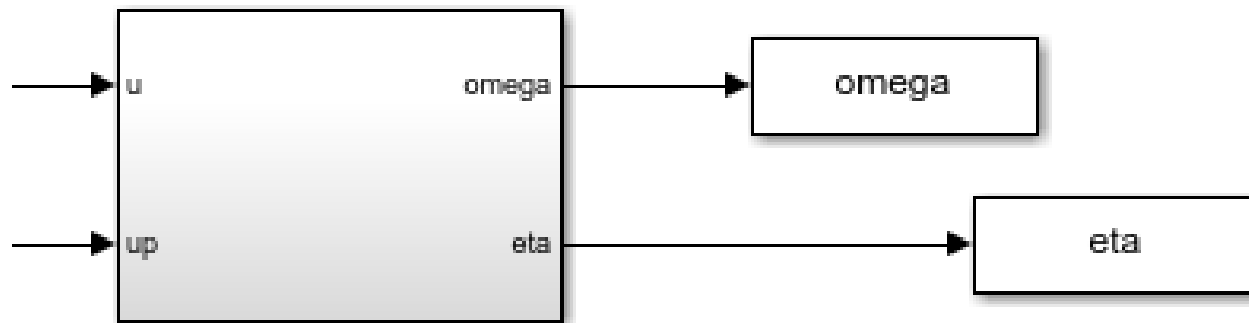


$$\dot{\omega}(t) = J^{-1} \{ u(t) - \delta^T \ddot{\eta}(t) - M_{\omega}(t) (J \omega(t) + \dot{\eta}(t)) \}$$

$$\ddot{\eta}(t) = P u_p(t) - C \dot{\eta}(t) - K \eta(t) - \delta \omega(t)$$

N.B. il modello in forma esplicita prevederebbe che si sostituisse nella prima equazione l'espressione di $\ddot{\eta}(t)$ ricavata dalla seconda. Si otterrebbe un modello più complesso. La forma «pseudo esplicita» rappresentata dalle equazioni in rosso risulta maggiormente vantaggiosa ai fini della realizzazione del modello Simulink.

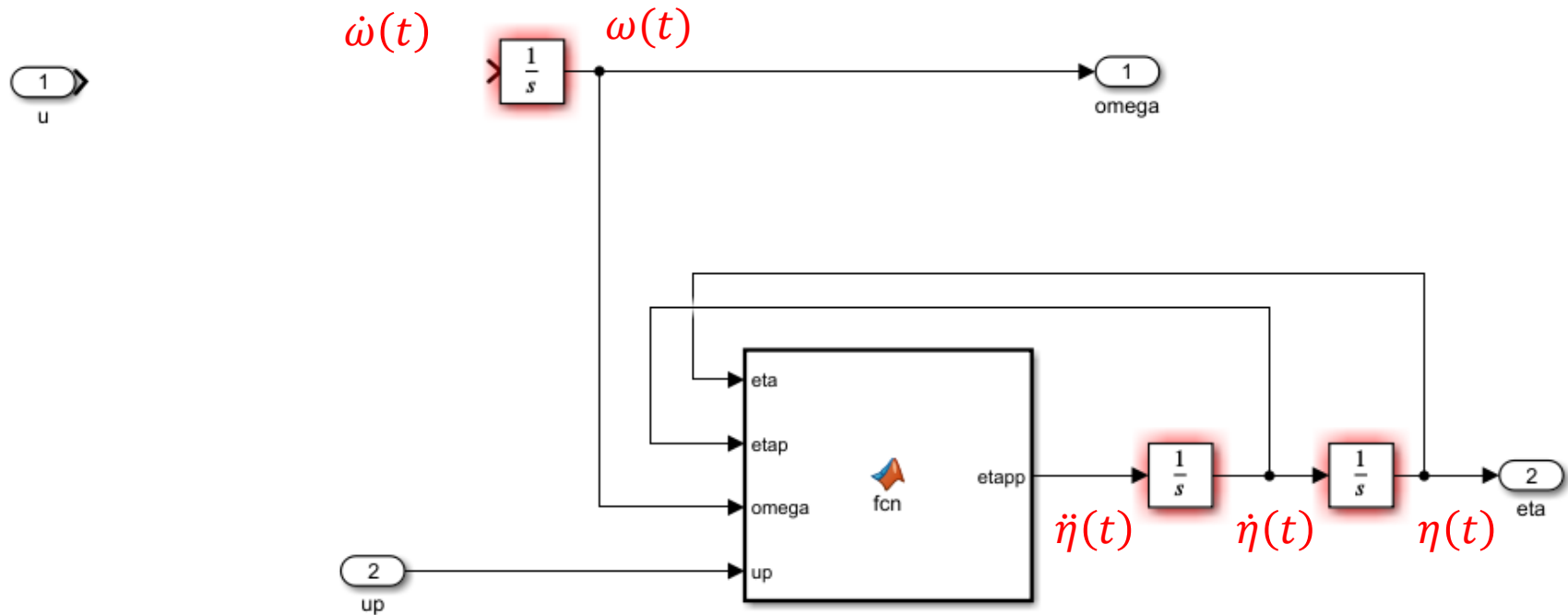
Realizziamo in primis il subsystem principale:



$$\dot{\omega}(t) = J^{-1}\{u(t) - \delta^T \dot{\eta}(t) - M_{\omega}(t) (J \omega(t) + \dot{\eta}(t))\} = F_1(\omega(t), \dot{\eta}(t), \ddot{\eta}(t), u(t))$$

$$\ddot{\eta}(t) = P u_p(t) - C \dot{\eta}(t) - K \eta(t) - \delta \omega(t) = F_2(\eta(t), \dot{\eta}(t), \omega(t), u_p(t))$$

Importiamo al suo interno tre blocchi integratori ed un blocco Matlab Function che genererà $\ddot{\eta}(t)$



$$\ddot{\eta}(t) = P u_p(t) - C\dot{\eta}(t) - K\eta(t) - \delta\omega(t)$$

```
function etapp = fcn(eta,etap,omega,up)
```

```
C=[0.5 0 0;0 0.4 0; 0 0 0.3];
```

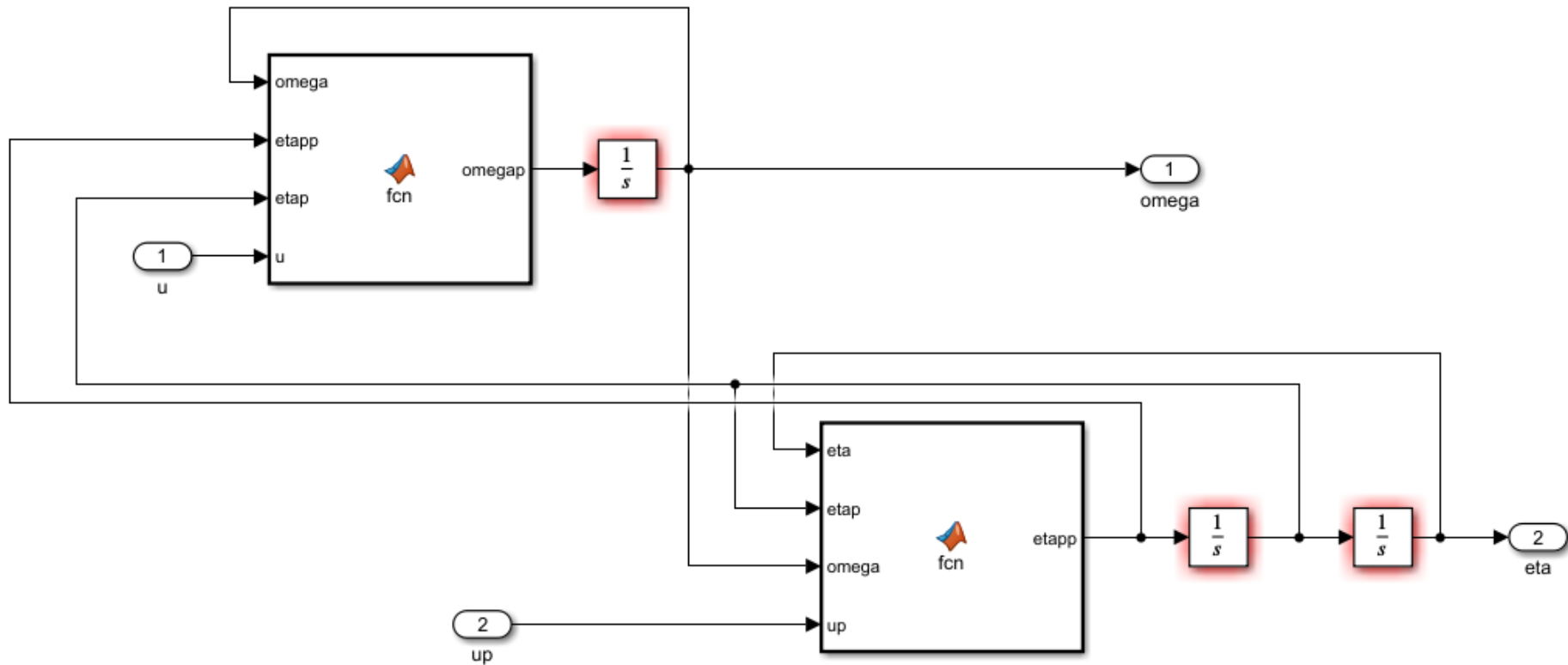
```
K=[2 0 0;0 3 0;0 0 1];
```

```
delta=[2.62 0.007 -0.003; -0.001 0.124 -2.73; -0.001 0.437 -0.051];
```

```
P=[70.26 -4.23 2.34; 4.8 31.93 1.24; -1.05 2.55 29.84];
```

```
etapp=P*up-C*etap-K*eta-delta*omega;
```

Inseriamo un secondo blocco Matlab Function che genererà $\dot{\omega}(t)$



$$\dot{\omega}(t) = J^{-1}\{u(t) - \delta^T \ddot{\eta}(t) - M_{\omega}(t) (J \omega(t) + \dot{\eta}(t))\}$$

$$M_{\omega}(t) = \begin{bmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix}$$

```
function omegap = fcn(omega,etapp,etap,u)
```

```
J=[420.8 3.6 -4.2; 3.6 410.6 9.4; -4.2 9.4 690.7];
```

```
delta=[2.62 0.007 -0.003; -0.001 0.124 -2.73; -0.001 0.437 -0.051];
```

```
omegax=omega(1);
```

```
omegay=omega(2);
```

```
omegaz=omega(3);
```

```
Momega=[0 -omegaz omegay; omegaz 0 -omegax; -omegay omegax 0];
```

```
omegap=inv(J)*(u-Momega*(J*omega+etap)-delta'*etapp);
```

Definiamo in uno script le condizioni iniziali $\omega(0)$, $\eta(0)$ e $\dot{\eta}(0)$

$$\omega(0) = \begin{bmatrix} 0.1 \\ -0.1 \\ 0 \end{bmatrix} \quad \eta(0) = \begin{bmatrix} 0.01 \\ 0.02 \\ 0.03 \end{bmatrix} \quad \dot{\eta}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

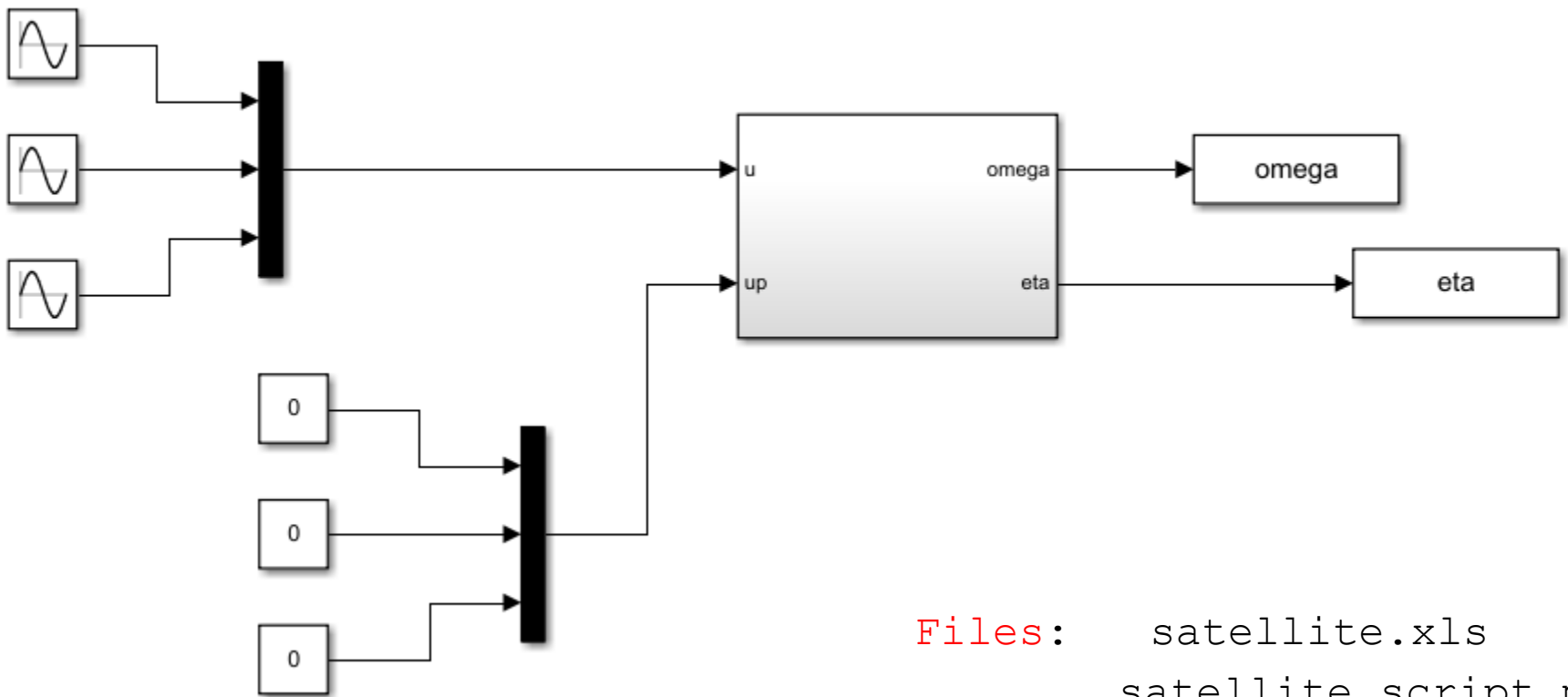
```
omega0=[0.1; -0.1; 0];
eta0=[0.01; 0.02; 0.03];
etap0=[0;0;0];
```

Vettori colonna

e inseriamo le tre variabili $\omega(0)$, $\eta(0)$, $\dot{\eta}(0)$ nel campo «Initial Condition» del rispettivo integratore

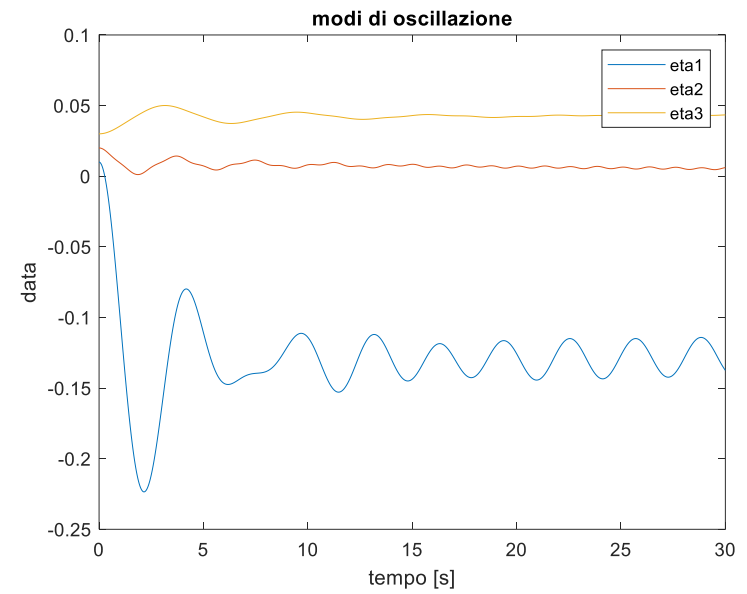
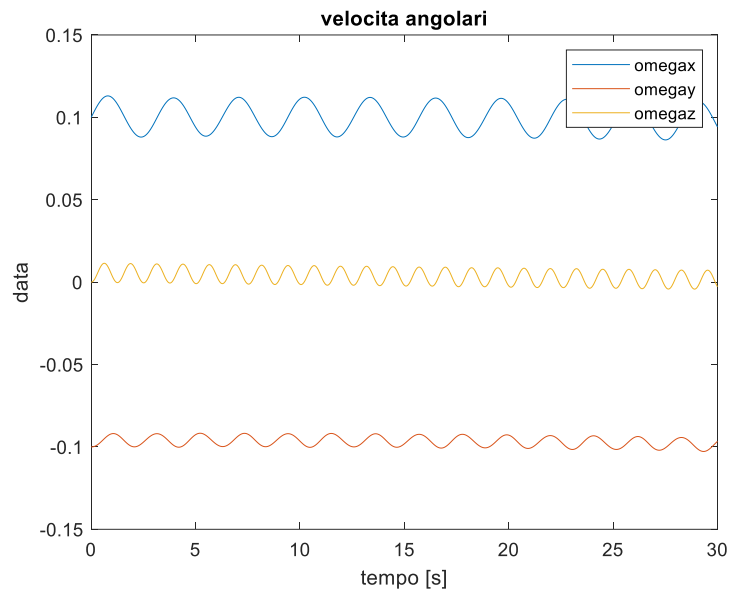
Ora costruiamo, esternamente al subsystem, i vettori $u(t)$ e $u_p(t)$

$$u(t) = \begin{bmatrix} 10\cos(2t) \\ 5\sin(3t) \\ 20\sin(5t) \end{bmatrix} \quad u_p(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$



Files: satellite.xls
 satellite_script.m

```
omega0=[0.1; -0.1; 0];  
eta0=[0.01; 0.02; 0.03];  
etap0=[0;0;0];  
  
sim('satellite')  
%%  
figure(1)  
plot(omega)  
xlabel('tempo [s]')  
title('velocita angolari')  
legend('omegax','omegay','omegaz')  
  
figure(2)  
plot(eta)  
xlabel('tempo [s]')  
title('modi di oscillazione')  
legend('eta1','eta2','eta3')
```



Scrivere quindi una **function** che acquisisca in ingresso i vettori $\omega_x(t)$, $\omega_y(t)$ ed $\omega_z(t)$ esportati nel workspace e costruisca il vettore $\Omega(t) = \sqrt{\omega_x^2(t) + \omega_y^2(t) + \omega_z^2(t)}$.

```
function out=VELTOT(in)

omegax=in.Data(:,1);
omegay=in.Data(:,2);
omegaz=in.Data(:,3);

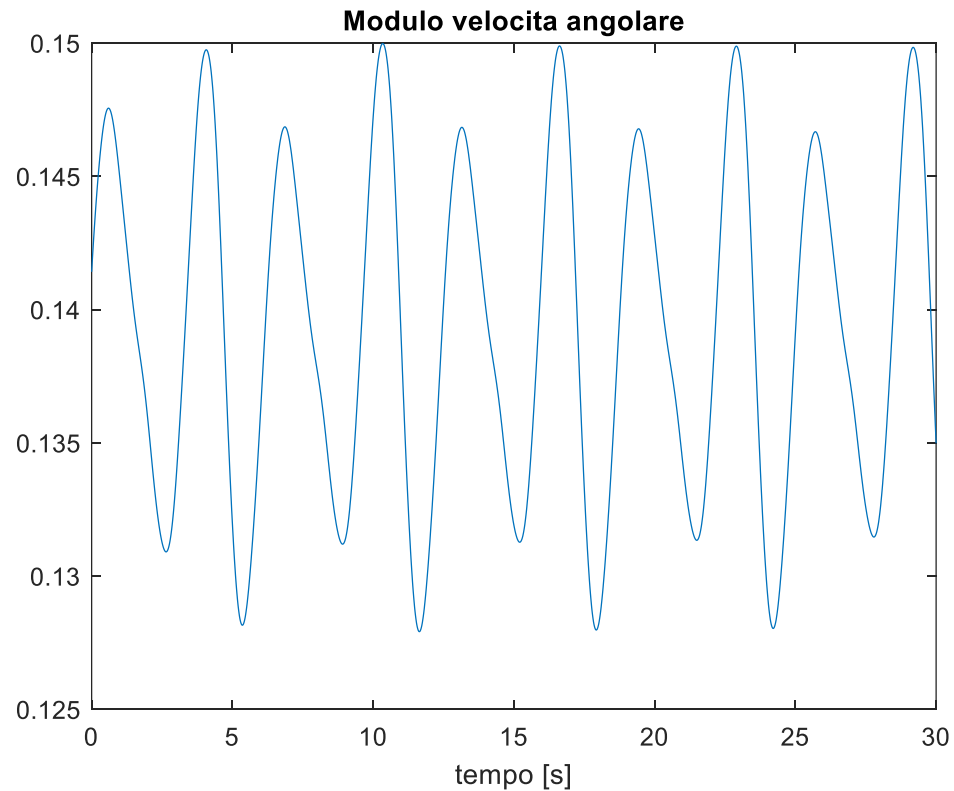
out=sqrt(omegax.^2+omegay.^2+omegaz.^2);
end
```

Inseriamo la function al termine dello script, e la richiamiamo prima della sua definizione passandogli come argomento di ingresso la variabile omega di tipo Timeseries.

```
Omega=VELTOT(omega);
```



```
figure(3)  
plot(0:0.01:30, Omega)  
xlabel('tempo [s]')  
title('Modulo velocita angolare')
```



Forma differente per la **function** preposta per ricevere in ingresso le tre componenti della velocità angolare

```
function
out=VELTOT2(in1,in2,in3)

out=sqrt(in1.^2+in2.^2+in3.^2);
end
```

Istruzioni da inserire nello script:

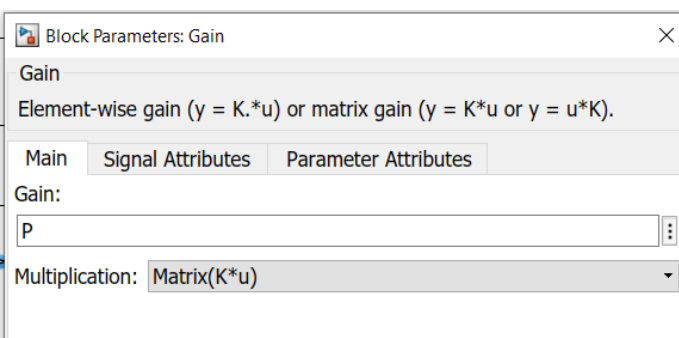
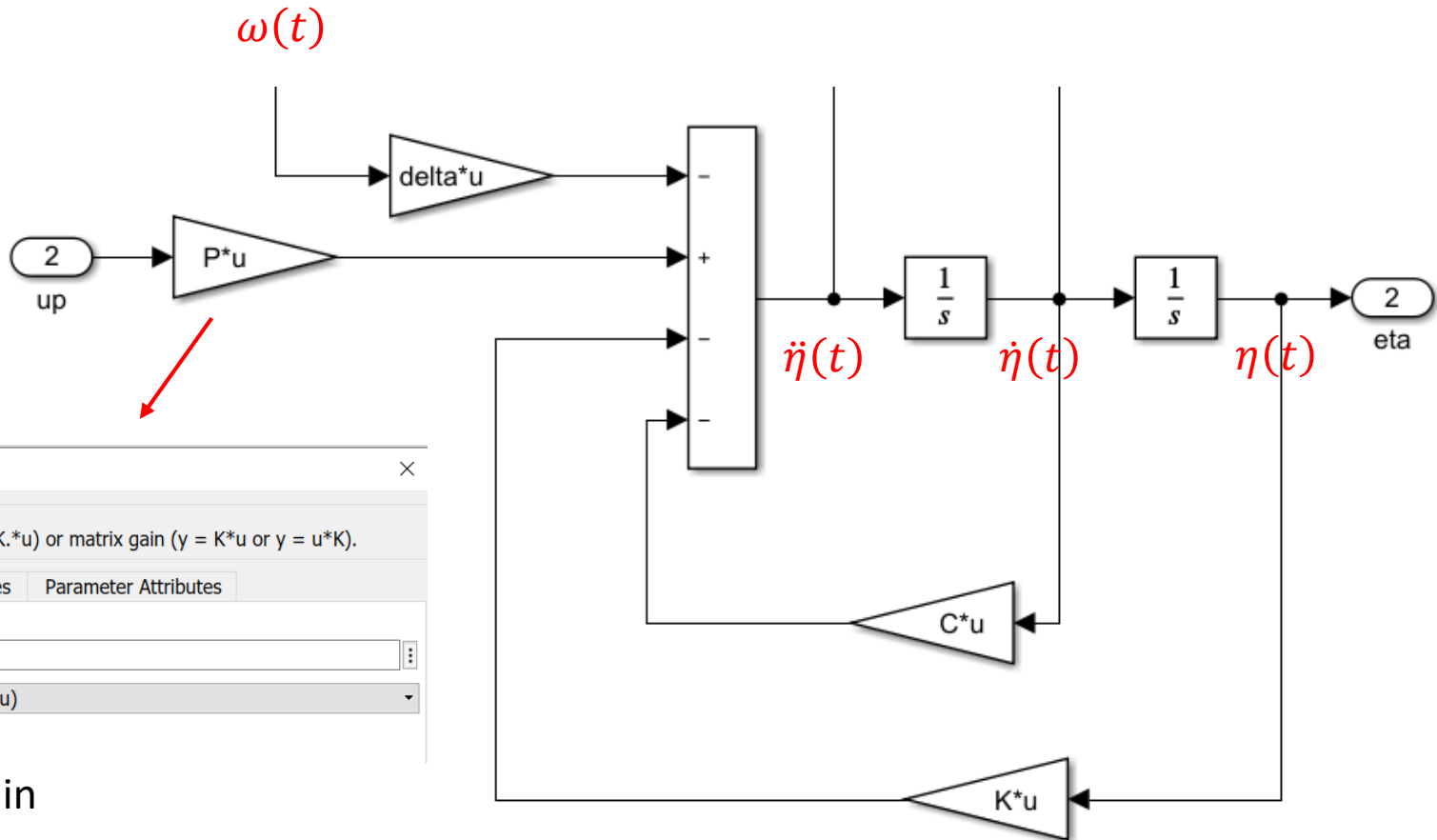
```
omegax=omega.Data(:,1);
omegay=omega.Data(:,2);
omegaz=omega.Data(:,3);

Omega=VELTOT2(omegax,omegay,omegaz);

figure(4)
plot(0:0.01:30,Omega)
xlabel('tempo [s]')
title('Modulo velocità angolare')
```

Modo alternativo per realizzare il Subsystem. Costruzione di $\ddot{\eta}(t)$.

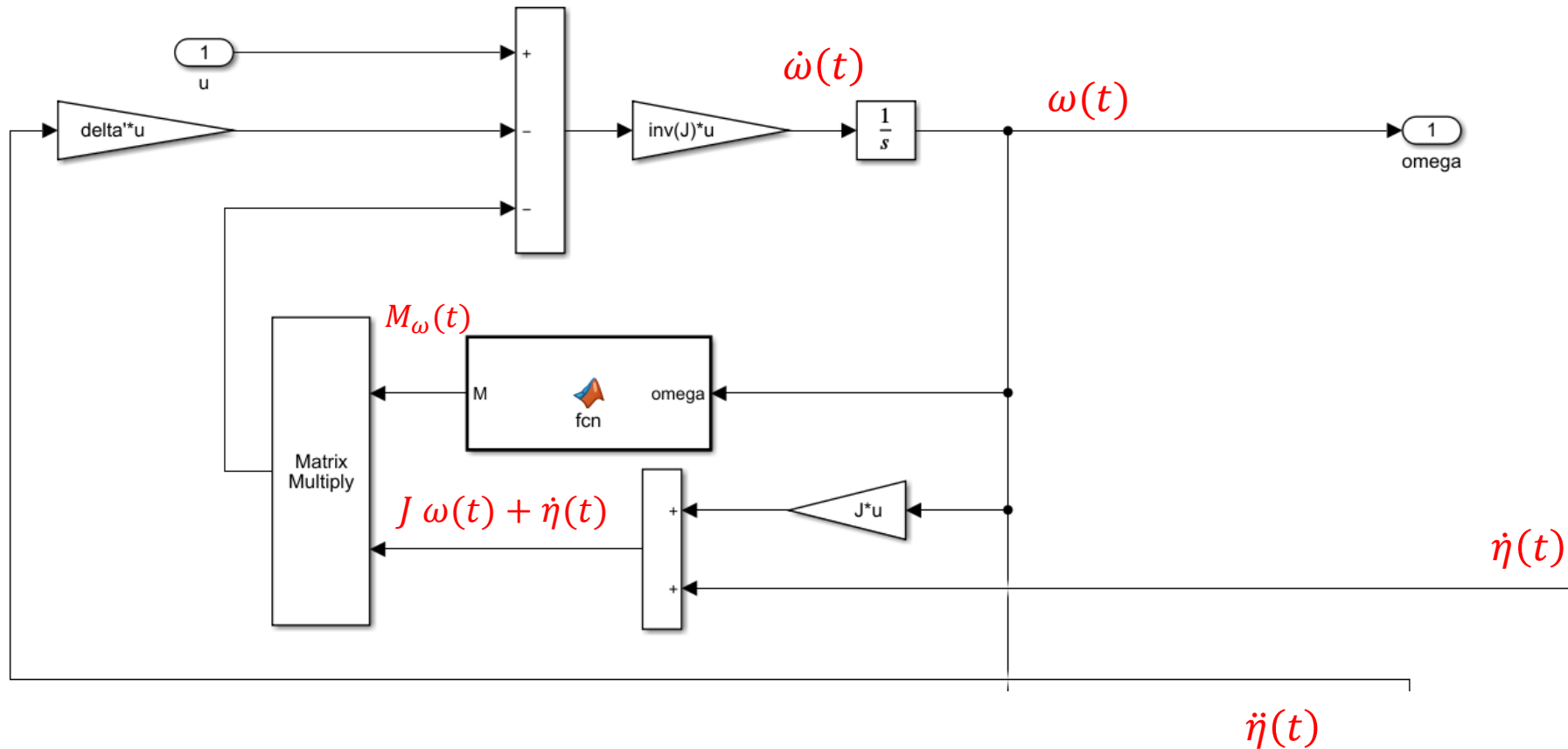
$$\ddot{\eta}(t) = P u_p(t) - C\dot{\eta}(t) - K\eta(t) - \delta\omega(t)$$

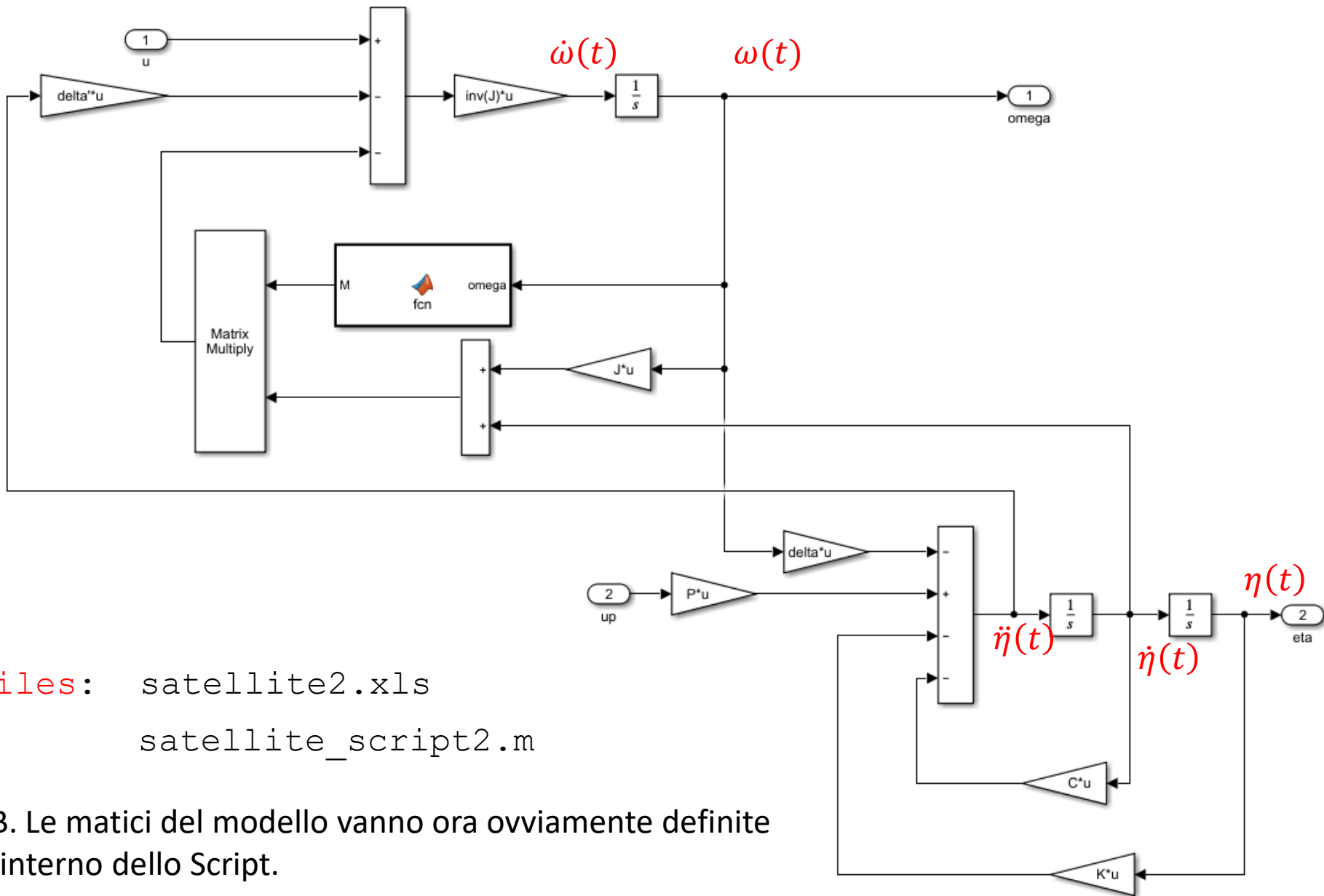


Blocco Gain

Modo alternativo per realizzare il Subsystem. Costruzione di $\dot{\omega}(t)$

$$\dot{\omega}(t) = J^{-1}\{u(t) - \delta^T \ddot{\eta}(t) - M_{\omega}(t) (J \omega(t) + \dot{\eta}(t))\}$$





Files: satellite2.xls
satellite_script2.m

N.B. Le matici del modello vanno ora ovviamente definite all'interno dello Script.