UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

Tesina di Complementi di controllo

Controllo di un quadricottero

Professore
Alfredo Pironti

Candidato
Dario Di Francesco
P38000171

Anno Accademico 2021/2022

Indice

1. Introduzione

Nel seguente elaborato verrà trattata la **modellazione** e il **controllo** di un **sistema multivariabile**, in particolare un quadricottero.

Il modello del quadricottero a cui si giungerà sarà non lineare, da questo modello non lineare si otterrà un modello linearizzato.

Utilizzando il sistema linearizzato verranno progettati le seguenti tipologie di controlli:

1) **Controllore** in **retroazione dello stato** e **precompensatore** con **allocazione** degli **autovalori**, **l'allocazione** verrà fatta utilizzando i seguenti metodi:
   a. **Luogo delle radici**
   b. **Metodo ITAE**
2) **Controllore** in **retroazione** dello **stato**, **precompensatore** e **osservatore** e con allocazione degli **autovalori**, l'**allocazione** verrà fatta utilizzando i seguenti metodi:
   a. **Luogo delle radici**
   b. **Metodo ITAE**
3) Controllore in **retroazione** dello **stato**, **integratore** e **osservatore** e con **allocazione degli autovalori**, **l'allocazione** verrà fatta utilizzando i seguenti metodi:
   a. **Luogo delle radici**
   b. **Metodo ITAE**
4) **Controllore ottimo LQR**, i valori della **matrice Q** verranno scelti seguendo due metodologie:
   a. **Sperimentalmente**
   b. **Scelta dei pesi**
5) **Controllore ottimo LQG,** i valori della **matrice Q** verranno scelti seguendo due metodologie:
   a. **Sperimentalmente**
   b. **Scelta dei pesi**

2. Modello matematico

2.1Meccanica del Volo

I **quadricotteri** sono una classe di velivoli dotati di quattro **motori** posti a croce sul **telaio** o **frame** ,ovvero dei **motori elettrici brushless** che vengono scelti per la loro semplicità e per il loro peso, questi motori fanno ruotare delle **eliche rotanti** con **passo fisso**, composte tipicamente da due pale per avere un buon compromesso tra semplicità, peso e potenza sviluppata.

Il **quadricottero** possiede **sei gradi di libertà** (**GdL**) dei quali:

- <u>**3 GdL**</u> per la **traslazione** lungo gli assi $\boldsymbol{x}$ $\boldsymbol{y}$ e $\boldsymbol{z}$;
- <u>**3 GdL**</u> per la **rotazione** attorno agli **assi**;

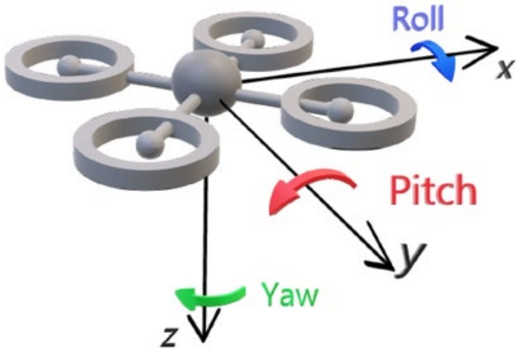


Figura 1: Rotazione del quadricottero attorno ai tre assi

I **droni** essendo dotati di quattro motori sono dei sistemi **sotto-attuato** perché il numero dei **gradi di libertà** è superiore al numero degli **attuatori**.

Il movimento di **rotazione** attorno all'asse $\boldsymbol{x}$ del velivolo è chiamato **rollio** o **roll**, la rotazione attorno all'asse $\boldsymbol{y}$ è detta **beccheggio** o **pitch** ed infine la rotazione attorno all'asse $\boldsymbol{z}$ è chiamata **imbardata** o **yaw.**

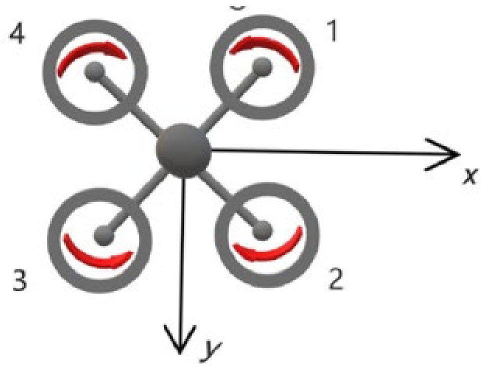


Figura 2: Senso di rotazione dei propulsori di un quadrirotore

Per garantire il movimento e la rotazione attorno ai **tre assi** occorre variare la **velocità** di **rotazione** delle **eliche**, queste ultime non ruoteranno tutte nello stesso **senso** infatti due coppie di **eliche** diametralmente opposte ruoteranno in senso **orario** e le altre due in senso antiorario, questo servirà per garantire la stabilità (come mostrato in figura7)

Ogni elica ruotando genera una **forza**, diretta lungo l'**asse** $\boldsymbol{z}$ ed una **coppia**, se tutti i propulsori ruotassero alla stessa velocità si otterrebbe che tutte le **coppie** generate saranno di uguale intensità e dunque si annulleranno, l'accelerazione angolare risultante sarà quindi essere nulla

Da quanto detto si deduce che modificando la velocità di rotazione delle eliche il quadrirotore è in grado di creare scompensi di forze e momenti che permettono ad esso di spostarsi e ruotare attorno agli assi.
Quando tutte le eliche girano alla stessa velocità e generano una forza totale diretta verso l'alto pari al peso del quadrirotore si ottiene la condizione di volo a punto fisso, detta **hovering**.

Se andiamo ad aumentare la spinta di ogni propulsore della stessa quantità otterremo una manovra di salita mentre al contrario se diminuiamo in ugual modo la spinta di ogni propulsore il quadrirotore si immetterà in una traiettoria di discesa.
Le manovre più complesse di rotazione attorno agli assi di **rollio** e **beccheggio** sono ottenute incrementando la potenza di un rotore e riducendo quella del suo opposto:

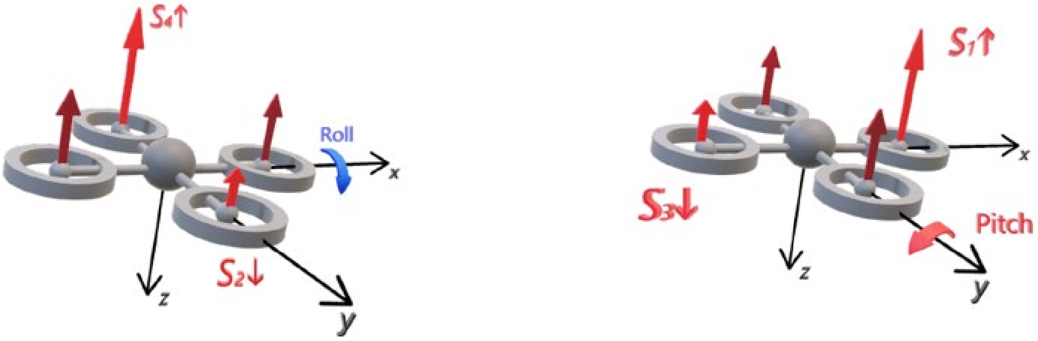


Figura 3 Manovra di rollio o roll e di beccheggio o pitch

La manovra di rotazione attorno all'asse di **imbardata** viene invece ottenuta andando ad incrementare la potenza di una coppia di **motori opposti** e riducendo quella degli altri due, la portanza complessiva dovrà essere comunque pari al peso del quadrirotore per poter mantenere la quota costante:

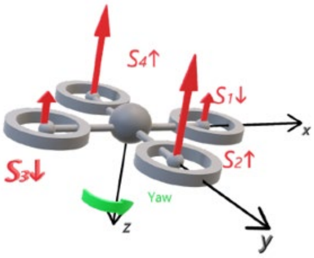


Figura 4: Manovra di imbardata

In questo modo lo squilibrio di momenti, causato dalla differenza tra le coppie dei rotori, provoca una rotazione attorno all'asse verticale.

2.2 Angoli di Eulero

Gli angoli di Eulero permettono di descrivere l'assetto di un velivolo nello spazio, ossia l'inclinazione del quadrirotore rispetto al **sistema di riferimento globale**. Questi **angoli** si ottengono una serie di rotazioni elementari partendo da un sistema di riferimento fisso $\boldsymbol{x}$ $\boldsymbol{y}$ e $\boldsymbol{z}$ e arrivare al sistema di riferimento solidale $\boldsymbol{x'}$ $\boldsymbol{y'}$ e $\boldsymbol{z'}$:

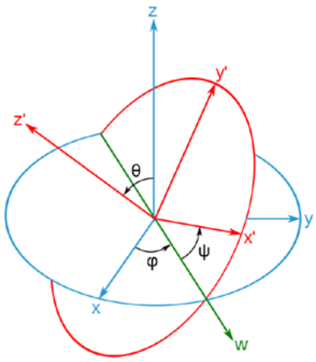


Figura 5: Rappresentazione angoli di Eulero

Si possono definire i seguenti angoli:

- $\boldsymbol{\varphi}$: è l'angolo di **rollio** ottenuto facendo ruotare la **terna** solidale attorno all'**asse** $\boldsymbol{x'}$
- $\boldsymbol{\theta}$: è l'angolo di **beccheggio** ottenuto facendo ruotare la **terna** solidale attorno all'**asse** $\boldsymbol{y'}$
- $\boldsymbol{\psi}$: è l'angolo di **imbardata** ottenuto facendo ruotare la **terna** solidale attorno all'**asse** $\boldsymbol{z'}$

Si nota che le **rotazioni positive** sono in senso antiorario e il dominio dei **tre angoli** sono:

$$\boldsymbol{\varphi} \in \left]-\mathbf{180}^{\circ}, \mathbf{180}^{\circ}\right[\tag{2.2.1}$$

$$\boldsymbol{\theta} \in \left]-\mathbf{90}, \mathbf{90}^{\circ}\right[\tag{2.2.2}$$

$$\boldsymbol{\psi} \in \left]-\mathbf{180}^{\circ}, \mathbf{180}^{\circ}\right[\tag{2.2.3}$$

Le **matrici associate** alle **rotazioni** saranno:

- <u>**Matrice associata alla rotazione intorno a $\boldsymbol{\phi}$**</u>**:**

$$\boldsymbol{R_{\varphi}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{cos(\varphi)} & \boldsymbol{-sin(\varphi)} \\ \mathbf{0} & \boldsymbol{sin(\varphi)} & \boldsymbol{cos(\varphi)} \end{bmatrix} \tag{2.2.4}$$

- <u>**Matrice associata alla rotazione intorno a $\boldsymbol{\theta}$**</u>**:**

$$\boldsymbol{R_{\theta}} = \begin{bmatrix} \boldsymbol{cos(\theta)} & \mathbf{0} & \boldsymbol{sin(\theta)} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{-sin(\theta)} & \mathbf{0} & \boldsymbol{cos(\theta)} \end{bmatrix} \tag{2.2.5}$$

- <u>**Matrice associata alla rotazione intorno a $\boldsymbol{\psi}$**</u>**:**

$$\boldsymbol{R_{\psi}} = \begin{bmatrix} \boldsymbol{cos(\psi)} & \boldsymbol{-sin(\psi)} & \mathbf{0} \\ \boldsymbol{sin(\psi)} & \boldsymbol{cos(\psi)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{2.2.6}$$

Moltiplicando $\boldsymbol{R_{\varphi}}, \boldsymbol{R_{\theta}}, \boldsymbol{R_{\psi}}$ fra loro ottengo la matrice di **rotazione DCM** (**direction cosine matrix**):

$$\boldsymbol{DCM} = \begin{bmatrix} \boldsymbol{cos(\theta)cos(\psi)} & \boldsymbol{cos(\theta)sin(\psi)} & \boldsymbol{-sin(\theta)} \\ \boldsymbol{sin(\varphi)sin(\theta)cos(\psi) - cos(\varphi)sin(\psi)} & \boldsymbol{sin(\varphi)sin(\theta)sin(\psi) + cos(\varphi)sin(\psi)} & \boldsymbol{sin(\varphi)cos(\theta)} \\ \boldsymbol{sin(\varphi)cos(\theta)cos(\psi) + sin(\varphi)sin(\psi)} & \boldsymbol{sin(\varphi)cos(\theta)sin(\psi) - sin(\varphi)cos(\psi)} & \boldsymbol{cos(\varphi)cos(\theta)} \end{bmatrix} \tag{2.2.1}$$

2.3 SISTEMI DI RIFERIMENTO

I **sistemi** di **riferimento** o **SDR** che vengono utilizzati nel moto di un velivolo sono i seguenti:

- <u>SDR Inerziale NED (North-East-Down)</u>**:** questo sistema non è realmente **inerziale** ma lo diventa considerando delle ipotesi semplificative in cui la terra viene considera piatta e non rotante

- <u>SDR Assi Verticali Locali</u>**:** questo sistema viene fissata l'origine nel baricentro del drone, gli assi di questo sistema sono sempre paralleli al **SDR Inerziale**.

- <u>SDR Assi Corpo</u> o <u>Body</u>**:** anche in questo sistema di riferimento viene fissata l'origine nel baricentro del drone, ma in questo caso gli assi rimangono solidali al drone e quindi quando il drone si inclina questo sistema di riferimento si inclina con esso.

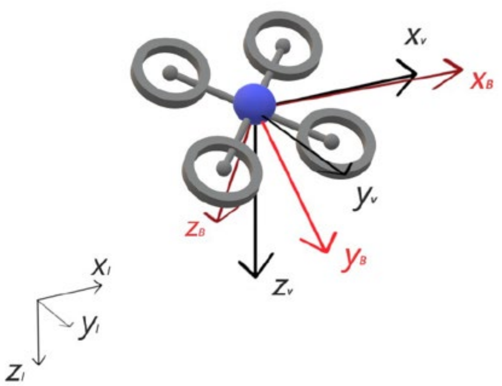


Figura 6: Rappresentazione dei vari SDR del quadricottero

2.4. VELOCITA' NEL SISTEMA BODY CON ANGOLI DI EULERO

In questo capitolo si analizza come sintetizzare la posizione e il movimento del quadricottero nello spazio sotto forma di equazioni differenziale, ovvero si procede con la modellizzazione vera e propria del caso reale.

Nella teoria del controllo le dinamiche del sistema modellato sono espresse tipicamente attraverso lo stato del sistema che nel caso del quadrirotore corrispondono a **12 equazioni** che ne descrivono l'assetto e la posizione a **sei gradi** di **libertà**.

Quindi si evince immediatamente che **6 gradi di libertà** appartengono al sistema di **riferimento NED** e i restanti **6** al sistema di riferimento **Body**.

Come primo step identifichiamo tutte le **velocità** di **traslazione** $\boldsymbol{V_B}$ e **velocità** di **rotazione** $\boldsymbol{\omega_B}$ del **drone**, ovvero le componenti nel sistema **Body**, ovvero:

$$\boldsymbol{V_B} = [\boldsymbol{u} \quad \boldsymbol{v} \quad \boldsymbol{w}]^{\boldsymbol{T}} \tag{2.4.1}$$

$$\boldsymbol{\omega_B} = [\boldsymbol{p} \quad \boldsymbol{q} \quad \boldsymbol{r}]^{\boldsymbol{T}} \tag{2.4.1}$$

2.5. EQUAZIONI STANDARD DEL VOLO

Per il **sistema NED** o termini lineare invece avremo:

$$\boldsymbol{\Gamma}_{\mathbf{NED}} = [\boldsymbol{x} \quad \boldsymbol{y} \quad \boldsymbol{z}]^{\boldsymbol{T}} \tag{2.5.1}$$

$$\boldsymbol{\Theta}_{\mathbf{NED}} = [\boldsymbol{\varphi} \quad \boldsymbol{\theta} \quad \boldsymbol{\psi}]^{\boldsymbol{T}} \tag{2.5.2}$$

Quindi per i termini lineari avremo:

$$\dot{\boldsymbol{\Gamma}}_{\mathbf{NED}} = \boldsymbol{DCM} \cdot \boldsymbol{V_B} \tag{2.5.3}$$

E analogamente per le rotazioni varrà:

$$\dot{\boldsymbol{\Theta}}_{\mathbf{NED}} = \boldsymbol{T} \cdot \boldsymbol{\omega_B} \tag{2.5.4}$$

Dove la matrice $\boldsymbol{T}$ analogamente alla $\boldsymbol{R}$ è una matrice per le **trasformazioni angolari**:

$$\boldsymbol{T} = \begin{bmatrix} \mathbf{1} & \boldsymbol{sin(\varphi)tan(\theta)} & \boldsymbol{cos(\varphi)tan(\theta)} \\ \mathbf{0} & \boldsymbol{cos(\varphi)} & -\boldsymbol{sin(\varphi)} \\ \mathbf{0} & \dfrac{\boldsymbol{sin(\varphi)}}{\boldsymbol{cos(\theta)}} & \dfrac{\boldsymbol{cos(\varphi)}}{\boldsymbol{cos(\theta)}} \end{bmatrix} \tag{2.5.5}$$

Esplicitando le **equazioni matriciali** delle **traslazioni** otteniamo il seguente sistema:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{u}\, \boldsymbol{cos(\theta)sin(\psi)} + \boldsymbol{v}\big(\boldsymbol{sin(\theta)sin(\phi)cos(\psi)} - \boldsymbol{cos(\phi)sin(\psi)}\big) + \boldsymbol{w}\big(\boldsymbol{sin(\theta)cos(\phi)cos(\psi)} + \boldsymbol{sin(\phi)sin(\psi)}\big); \\ \dot{\boldsymbol{y}} = \boldsymbol{u}\, \boldsymbol{cos(\theta)sin(\psi)} + \boldsymbol{v}\big(\boldsymbol{sin(\theta)sin(\phi)sin(\psi)} + \boldsymbol{cos(\phi)}\, \boldsymbol{cos(\psi)}\big) + \boldsymbol{w}\big(\boldsymbol{sin(\theta)cos(\phi)sin(\psi)} - \boldsymbol{sin(\phi)cos(\psi)}\big); \\ \dot{\boldsymbol{z}} = -\boldsymbol{u}\, \boldsymbol{sin(\theta)} + \boldsymbol{v}\, \boldsymbol{cos(\theta)sin(\phi)} + \boldsymbol{w}\, \boldsymbol{cos(\theta)cos(\phi)} \end{cases} \quad (2.5.6$$

Esplicitando le **equazioni matriciali** delle **rotazioni** otteniamo il seguente sistema:

$$\begin{cases} \dot{\boldsymbol{\phi}} = \boldsymbol{p} + \boldsymbol{q}\, \boldsymbol{tan(\theta)sin(\phi)} + \boldsymbol{r}\, \boldsymbol{tan(\theta)cos(\phi)}; \\ \dot{\boldsymbol{\theta}} = \boldsymbol{q}\, \boldsymbol{tan(\theta)sin(\phi)} - \boldsymbol{r}\, \boldsymbol{sin(\phi)}; \\ \dot{\boldsymbol{\psi}} = \dfrac{\boldsymbol{sin(\phi)}}{\boldsymbol{cos(\theta)}}\boldsymbol{q} + \dfrac{\boldsymbol{cos(\phi)}}{\boldsymbol{cos(\theta)}}\boldsymbol{r} \end{cases} \tag{2.5.7}$$

Avendo definito le equazioni che permettono di ricavare le **posizioni** $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ e le **posizioni angolari** $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, $\boldsymbol{\psi}$, tramite le **matrici trasformazioni** $\boldsymbol{DCM}$ e $\boldsymbol{T}$ delle coordinate, è possibile applicare la **seconda legge** di **Newton** $\boldsymbol{F} = \boldsymbol{m\ddot{a}}$ avvalendosi delle seguenti ipotesi:

- L'**origine** della **terna BODY** coincide con l'**origine** $\boldsymbol{NED}$;
- Il **centro** delle **terna BODY** coincide con il **centro** gli **assi principali** di **inerzia**.

Queste ipotesi permettono di scrivere un **tensore** di **inerzia diagonale** annullando i **prodotti centrifughi**.

Applicando le **leggi cardinali** della **dinamica** si ottiene che:

$$\boldsymbol{m\,\ddot{\Gamma}_{NED}} = \sum_{i=0}^{n} \boldsymbol{F_i} \tag{2.5.8}$$

$$\boldsymbol{I\,\ddot{\Theta}_{NED}} = \sum_{i=0}^{n} \boldsymbol{\tau_i} \tag{2.5.9}$$

Derivando la (2.5.3) si ottiene:

$$\boldsymbol{m\,\ddot{\Gamma}_{NED}} = \boldsymbol{m}\frac{\boldsymbol{d}}{\boldsymbol{dt}}(\boldsymbol{DCM\,V_B}) = \boldsymbol{DCM\,F_B} \tag{2.5.10}$$

Applicando la **proprietà** della **derivata** del **prodotto** di **funzioni** si ottiene:

$$\boldsymbol{m}\left(\boldsymbol{D\dot{C}M\,V_B} + \boldsymbol{\dot{V}_B\,DCM}\right) = \boldsymbol{DCM\,F_B} \tag{2.5.11}$$

Il termine $\boldsymbol{D\dot{C}M\,V_B}$ che contiene la **velocità** della **terna traslazione** $\boldsymbol{DCM}$ che si può esprimere come il **prodotto vettoriale** tra la **velocità angolare** $\boldsymbol{\omega_B}$ e **velocità** di **traslazione** $\boldsymbol{V_B}$, ovvero:

$$\boldsymbol{D\dot{C}M\,V_B} = \boldsymbol{DCM}\,(\boldsymbol{\omega_B} \times \boldsymbol{V_B}) \tag{2.5.12}$$

Dalla (2.5.12) si ottiene:

$$\boldsymbol{m\,DCM}\left(\boldsymbol{\omega_B} \times \boldsymbol{V_B} + \boldsymbol{\dot{V}_B}\right) = \boldsymbol{DCM\,F_B} \tag{2.5.13}$$

Da cui si ottiene che:

$$\boldsymbol{m}\left(\boldsymbol{\omega_B} \times \boldsymbol{V_B} + \boldsymbol{\dot{V}_B}\right) = \boldsymbol{F_B} \tag{2.5.14}$$

Esplicitando la (2.5.9) si ottiene:

$$\boldsymbol{I\,\ddot{\Theta}_{NED}} = \boldsymbol{I}\,\frac{\boldsymbol{d}}{\boldsymbol{dt}}\,(\boldsymbol{T\omega_B}) = \boldsymbol{T\,\tau_B} \tag{2.5.15}$$

Applicando la **proprietà** della **derivata** del **prodotto** di **funzioni** si ottiene:

$$\boldsymbol{I}\left(\boldsymbol{\dot{T}\omega_B} + \boldsymbol{T\dot{\omega}_B}\right) = \boldsymbol{T\,\tau_B} \tag{2.5.16}$$

Il termine $\boldsymbol{\dot{T}\omega_B}$ che contiene la **velocità** della **terna traslazione** $\boldsymbol{DCM}$ che si può esprimere come il **prodotto vettoriale** tra il **momento angolare** $\boldsymbol{I\omega_B}$ e il prodotto tra la **velocità** di **angolare** $\boldsymbol{\omega_B}$ e la **terna traslazione** $\boldsymbol{T}$, ovvero:

$$\boldsymbol{\dot{T}\omega_B} = \boldsymbol{\omega_B T} \times (\boldsymbol{I\omega_B}) \tag{2.5.17}$$

Dalla (2.5.12) si ottiene:

$$\boldsymbol{I\omega_B T} \times (\boldsymbol{I\omega_B}) + \boldsymbol{I\,T\dot{\omega}_B} = \boldsymbol{T\,\tau_B} \tag{2.5.18}$$

Da cui si ottiene che:

$$\boldsymbol{I\omega_B \times (I\omega_B) + I\,\dot{\omega}_B = \tau_B} \tag{2.5.19}$$

Sviluppando i **prodotti vettoriali** e si ottiene l'**equazione vettoriale** in forma **matriciale** della forza :

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} = F_B \tag{2.5.20}$$

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = m \begin{bmatrix} \dot{p}I_x + qr(I_z - I_y) \\ \dot{q}I_y + pr(I_x - I_y) \\ \dot{r}I_z + pq(I_y - I_x) \end{bmatrix} = \tau_B \tag{2.5.21}$$

2.6. MATRICE DI INERZIA

Per facilitare il calcolo della **matrice** d'**inerzia** è stato stilizzato il corpo del **drone**, come mostrato in **figura 7.**

Il **modello stilizzato** del **drone** è un **modello virtuale** e questo **modello** è stato reso quanto più realistico andando a porre delle **masse** laddove sono concentrati i pesi del **drone**, ovvero:

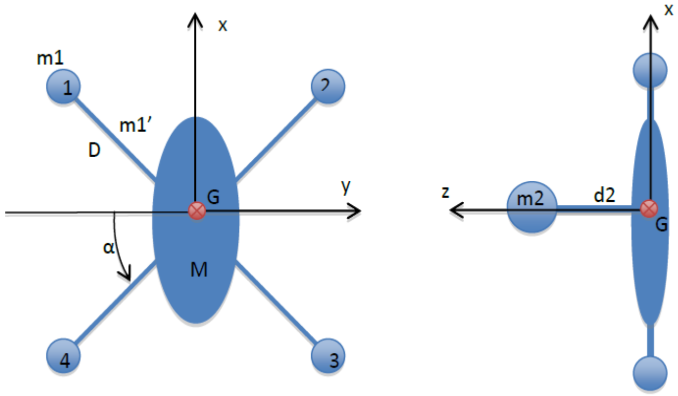


Figura 7: Stilizzazione del quadricottero

- $\boldsymbol{m_1}$: sono le **quattro masse** che rappresentano i **rotori** del **drone**
- $\boldsymbol{m_1}'$: sono le **quattro masse** che rappresentano i **bracci** del **drone**
- $\boldsymbol{M}$: è una **massa** che rappresenta il nucleo del drone che include l'elettronica e il corpo del drone stesso che è stato stilizzato in forma **ellissoidale**
- $\boldsymbol{m_2}$: è una **massa** che rappresenta il peso della **gimbal** a cui è collegata ad esempio una telecamera e questa massa è stata stilizzata con una forma **sferica**

Il **modello virtuale** sono state utilizzate delle **masse concentrate** per il **corpo centrale** e il **gimbal** per rendere trascurabili **momenti centrifughi** $\boldsymbol{I_{xy}, I_{xz}, I_{yz}}$ nella rotazione attorno all'**asse verticale.** Inoltre, i **rotori** sono stati approssimati con delle **masse concentrate** e questo non riporta grandi variazioni nel calcolo della **matrice** di **inerzia** $\boldsymbol{I}$ in quanto i **rotori** hanno piccole dimensioni e i **termini quadratici** quindi risultano trascurabili.

Sono state inoltre definite le seguenti grandezze:

- $\boldsymbol{D}$: **lunghezza** dei **bracci**
- $\boldsymbol{d_2}$: **lunghezza** della **gimbal**
- $\boldsymbol{\alpha}$: **angolo** tra l'**asse baricentrico** $\boldsymbol{x}$ e i **bracci** del drone
- $\boldsymbol{r}$: **raggio** della **sfera** della **massa** $\boldsymbol{m_2}$ che rappresenta la **gimbal**
- $\boldsymbol{a, b, c}$: **semiassi** dell' **sfera** della **massa** $\boldsymbol{M}$ che rappresenta il corpo del **drone**
- $\boldsymbol{p_a}$: $\boldsymbol{lunghezza \times passo}$ dell'elica del **drone**

Nel **modello virtuale**, essendo il corpo complessivo del **drone simmetrico** rispetto alla **verticale**, ovvero:

$$\begin{cases} X_g = 0; \\ X_g = 0; \end{cases} \tag{2.6.1}$$

Il **calcolo** del **baricentro** G quindi sarà fatto lungo l'**asse** z e quindi si ottiene:

$$G = \frac{\sum m_i Z_i}{\sum m_i} = \frac{m_1 d_2}{4m_1 + M + 4m_1' + m_2} \tag{2.6.2}$$

La **matrice** di **inezia** è pari a:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{2.6.3}$$

Per come è stato è stato definito il **modello virtuale** del drone i **momenti centrifughi** saranno pari a:

$$\begin{cases} I_{xy} = I_{yx} = 0 \\ I_{xz} = I_{xz} = 0 \\ I_{yz} = I_{zy} = 0 \end{cases} \tag{2.6.4}$$

Mentre i **momenti di inerzia di massa**

$$I_{xx} = 4\left(m_1 d_1^2 cos^2(\alpha) + \frac{1}{3} m_1' d_1^2 cos^2(\alpha)\right) + \frac{M}{5}(b^2 + c^2) + [4(m_1 + m_1') + M]G^2 + \frac{2}{5} m_2 r^2 + m_2(d_2 - G)^2 \tag{2.6.5}$$

$$I_{yy} = 4\left(m_1 d_1^2 sin^2(\alpha) + \frac{1}{3} m_1' d_1^2 cos^2(\alpha)\right) + \frac{M}{5}(a^2 + c^2) + [4(m_1 + m_1') + M]G^2 + \frac{2}{5} m_2 r^2 + m_2(d_2 + G)^2 \tag{2.6.7}$$

$$I_{zz} = 4\left(m_1 d_1^2 + \frac{1}{3} m_1' d_1^2\right) + \frac{M}{5}(a^2 + b^2) + \frac{2}{5} m_2 r^2 \tag{2.6.8}$$

Per cui la **matrice inerziale** sarà una **matrice diagonale** del **sistema** sarà pari a:

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.6.9}$$

2.7. FORZA THRUST E MOMENTI

Il **modello virtuale** del **drone**, introdotto nel precedente capitolo, è stato utilizzato per semplificare il **sistema inerziale** e renderlo **baricentrico**, questo permette di far sì che il **sistema** di **riferimento BODY** vada non solo a coincidere con il **baricentro** G del **drone** che è anche **centro** degli **assi principali** di **inerzia** (questo perché $I_{xy} = I_{zx} = I_{yz} = 0$) ma il **baricentro** G è anche il **punto** di **applicazione** di tutte le **forze** F e i **momenti** M.

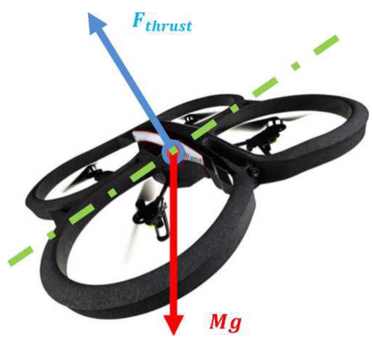


Figura 8

Per poter far sollevare il **drone** bisogna vincere la **forza gravitazionale** che attrae il corpo verso la terra dovuta proprio alla massa del drone, questa **forza gravitazionale** può essere modellata come un **vettore verticale** e **diretto** verso il **basso**. Per poter vincere questa forza il drone deve applicare una **forza uguale** e di **verso opposto** alla **forza gravitazionale** e questa forza è chiamata **forza** di **propulsione** o **Thrust** o di **spinta** dovuta all'azione dei rotori che generano un **vettore forza perpendicolare** al **piano** che contiene il **BODY** del **drone,** come mostrato in **Figura 8**.

Per questi motivi la **forza Thrust** avrà componente soltanto lungo l'**asse z** in quanto le **eliche** sono **fisse** e quindi la **forza** generata è sempre **perpendicolare** al **piano xy**, mentre **momenti** legati al **Trust** saranno invece presenti su tutte e tre le **componenti** della **terna** di **riferimento**.

$$\begin{cases} m(\dot{u} + qw - rv) = -mgsin(\theta) \\ m(\dot{v} + ru - pw) = mgcos(\theta)sin(\phi) \\ m(\dot{w} + pv - qu) = mgcos(\theta)cos(\phi) - F_{tz} \end{cases} \tag{2.7.1}$$

$$\begin{cases} \dot{p}I_x + I_z qr - I_y qr = M_{tx} \\ \dot{q}I_y + I_x pr - I_y pr = M_{ty} \\ \dot{r}I_z + I_y pq - I_x pq = M_{tz} \end{cases} \tag{2.7.2}$$

2.8. VARIBILI DI STATO E INGRESSI DEL SISTEMA

La **variabile vettoriale** degli **stati x** del **sistema** può essere definita come:

$$x = [\phi \quad \theta \quad \psi \quad phi \quad p \quad q \quad r \quad x \quad y \quad z \quad u \quad v \quad w] \tag{2.8.1}$$

La **variabile vettoriale** degli **ingressi stato u** del **sistema** può essere definita come:

$$u = [F_{tz} \quad M_{tx} \quad M_{ty} \quad M_{tz}] \tag{2.8.2}$$

È possibile esplicitare le equazioni scritte nei capitoli precedenti rispetto alle **variabili** di **stato**:

$$\begin{gathered} \dot{u} = -qw + rv - gsin(\theta) \\ \dot{v} = pw - ru + gcos(\theta)si\,n(\phi) \\ \dot{w} = qu - pv + gcos(\theta)cos(\phi) - \frac{F_{tz}}{m} \\ \dot{p} = \frac{(I_y - I_z)}{I_x}qr + \frac{M_{tx}}{I_x} \\ \dot{q} = \frac{(I_z - I_x)}{I_y}pr + \frac{M_{ty}}{I_x} \\ \dot{r} = \frac{(I_y - I_x)}{I_z}pq + \frac{M_{tz}}{I_z} \\ \dot{\phi} = p + q\,tan(\theta)sin(\phi) + r\,tan(\theta)cos(\phi); \\ \dot{\theta} = q\,tan(\theta)sin(\phi) - r\,sin(\phi); \\ \dot{\psi} = \frac{sin(\phi)}{cos(\theta)}q + \frac{cos(\phi)}{cos(\theta)}r \\ \dot{x} = u\,cos(\theta)sin(\psi) + v\big(sin(\theta)sin(\phi)cos(\psi) - cos(\phi)sin(\psi)\big) + w\big(sin(\theta)cos(\phi)cos(\psi) + sin(\phi)sin(\psi)\big); \\ \dot{y} = u\,cos(\theta)sin(\psi) + v\big(sin(\theta)sin(\phi)sin(\psi) + cos(\phi)\,cos(\psi)\big) + w\big(sin(\theta)cos(\phi)sin(\psi) - sin(\phi)cos(\psi)\big); \\ \dot{z} = -u\,sin(\theta) + v\,cos(\theta)sin(\phi) + w\,cos(\theta)cos(\phi) \end{gathered} \tag{2.8.3}$$

2.9. SCOMPOSIZIONE DELLE FORZE E DEI MOMENTI

Le **eliche** sono la causa scatenante della **forza** di **trhust F_{tz}** sia dei **momenti M_t** che permettono il sollevamento del **drone.** La **spinta** delle **eliche** sono dei **profili alari rotanti** che sono caratterizzati da una certa **portanza** ed una **resistenza**, come mostrato in Figura 9. In particolare, la **portanza** è diretta lungo la **normale** al **piano** contenente le **pale** delle **eliche**, mentre la **resistenza** genererà una **coppia rotante** a risultante nulla contenuta nel **piano** delle **eliche**, in verso opposto a quello di **rotazione**.

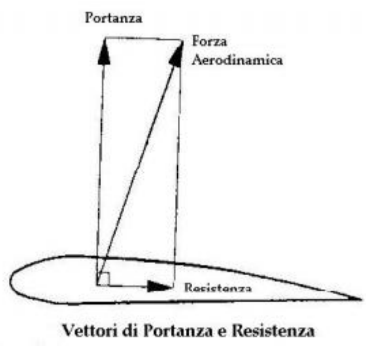


Figura 9

La **forza** di **trhust F_{tz}** è generata da ogni singola **elica** del **drone** ed è dovuta alla **portanza** che essa genera ed è definita dalla seguente relazione:

$$P = K_s \omega^2 \tag{2.9.2}$$

Dove:

- $K_s = 2L^4 \rho \tau C_s$: dove:
  - L: è **lunghezza** della **pala** (lama)
  - ρ: è la **densità** dell'**aria** (supposta costante)
  - τ: il **passo geometrico** dell'**elica**
  - C_s: è un parametro caratteristico dell'elica.

Il **drag** dell'**elica** (effetto della resistenza) è dato invece dal seguente sistema:

$$\begin{cases} R_x = R_y = K_s \omega^2 D \\ R_y = K_d \omega^2 = K_s \omega^2 L \end{cases} \tag{2.9.3}$$

Dove:

- $K_d = 2L^5 \rho \tau C_d$: dove:
  - L: è **lunghezza** della **pala** (lama)
  - ρ: è la **densità** dell'**aria** (supposta costante)
  - τ: il **passo geometrico** dell'**elica**
  - C_s: è un parametro caratteristico dell'elica.
- D: è la **lunghezza** dei **bracci**

Il **coefficiente K_s** è genere variabile in funzione di alcuni parametri caratteristici dell'elica e dalle condizioni di volo, ma a basse velocità può essere ritenuto costante, in particolare in questo caso è stato scelto pari a **0.9**.

2.10 . ASSETTO DEL DRONE

Per come è stato progettato il quadricottero si muoverà nella stessa **direzione** degli assi su cui giacciono i braccetti, ma la sua direzione è ruotata di in un angolo $\alpha = 45^\circ$, infatti il suo assetto è quello mostrato in Figura 10.

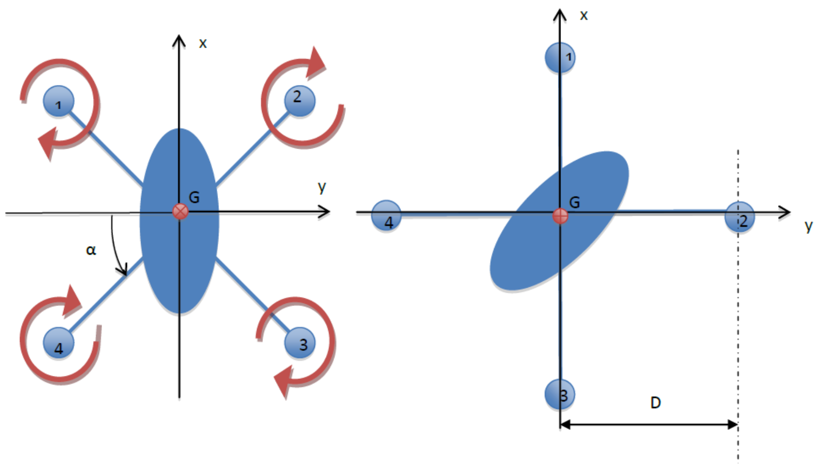


Figura 10 Assetto quadricottero

L'angolo α quindi definisce l'**angolo** di **rotazione** dell'**assetto** sistema **BODY** del **drone** rispetto ad il **sistema** di **riferimento fisso**.

3.1 MODELLO NON LINEARE DEL DRONE

$$
\begin{gathered}
\dot{u} = -qw + rv - g sin(\theta) \\
\dot{v} = pw - ru + g cos(\theta) sin(\phi) \\
\dot{w} = qu - pv + g cos(\theta) cos(\phi) - \frac{\sum_{i=1}^{4} K_s \omega^2}{m} \\
\dot{p} = \frac{(I_y - I_z)}{I_x} qr + \frac{\left((\omega_1^2 - \omega_3^2) sin(\alpha) + (\omega_4^2 - \omega_2^2) cos(\alpha)\right) D K_s}{I_x} \\
\dot{q} = \frac{(I_z - I_x)}{I_y} pr + \frac{\left((\omega_1^2 - \omega_3^2) cos(\alpha) + (\omega_4^2 - \omega_2^2) sin(\alpha)\right) D K_s}{I_x} \\
\dot{r} = \frac{(I_y - I_x)}{I_z} pq + \frac{(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) K_s L}{I_z} \\
\dot{\phi} = p + q\, tan(\theta) sin(\phi) + r\, tan(\theta) cos(\phi); \\
\dot{\theta} = q\, tan(\theta) sin(\phi) - r\, sin(\phi); \\
\dot{\psi} = \frac{sin(\phi)}{cos(\theta)} q + \frac{cos(\phi)}{cos(\theta)} r \\
\dot{x} = u\, cos(\theta) sin(\psi) + v\left(sin(\theta) sin(\phi) cos(\psi) - cos(\phi) sin(\psi)\right) + w\left(sin(\theta) cos(\phi) cos(\psi) + sin(\phi) sin(\psi)\right); \\
\dot{y} = u\, cos(\theta) sin(\psi) + v\left(sin(\theta) sin(\phi) sin(\psi) + cos(\phi) cos(\psi)\right) + w\left(sin(\theta) cos(\phi) sin(\psi) - sin(\phi) cos(\psi)\right); \\
\dot{z} = -u\, sin(\theta) + v\, cos(\theta) sin(\phi) + w\, cos(\theta) cos(\phi)
\end{gathered}
\tag{3.1.1}
$$

3.2 MODELLO NON LINEARE MATLAB

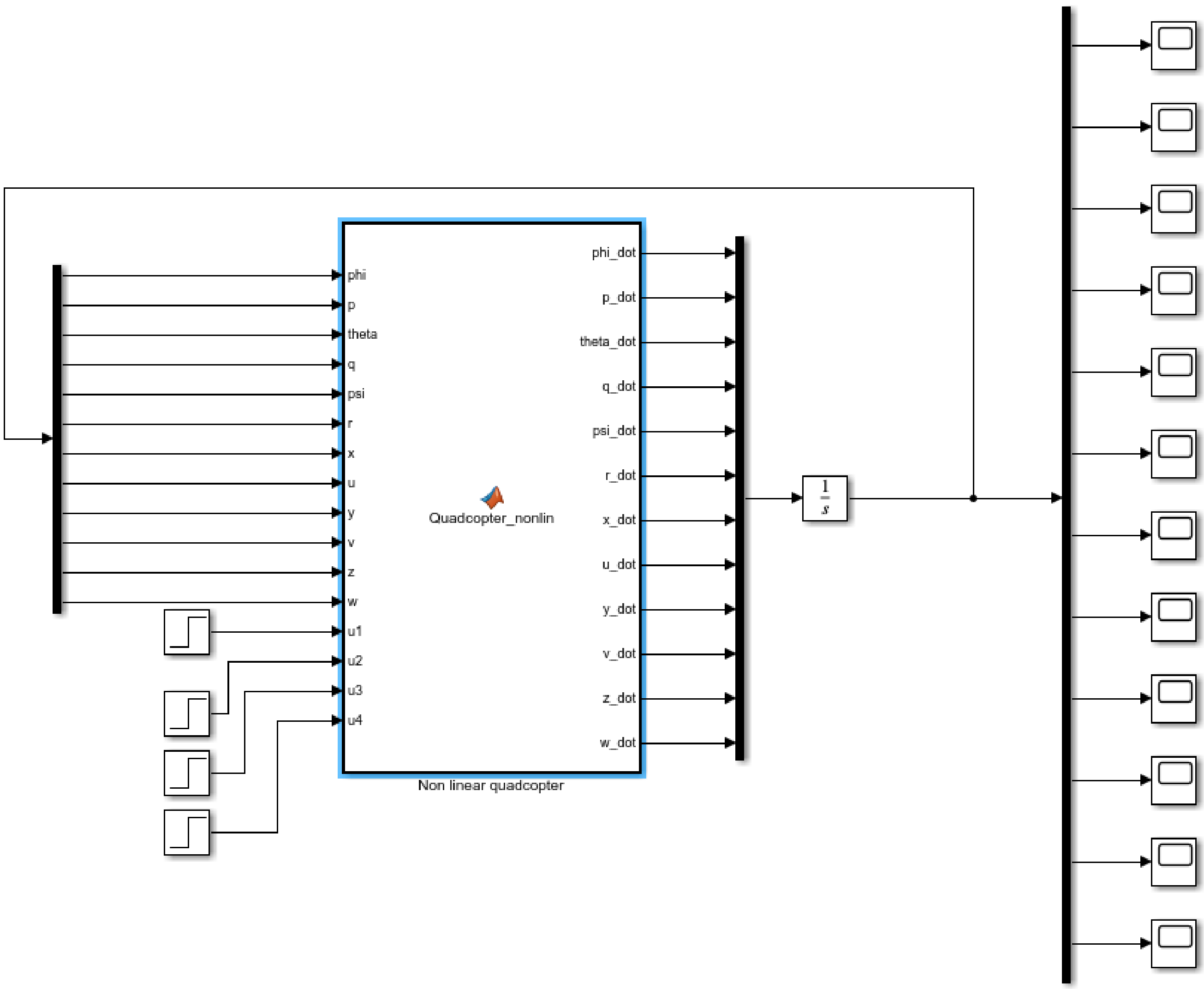


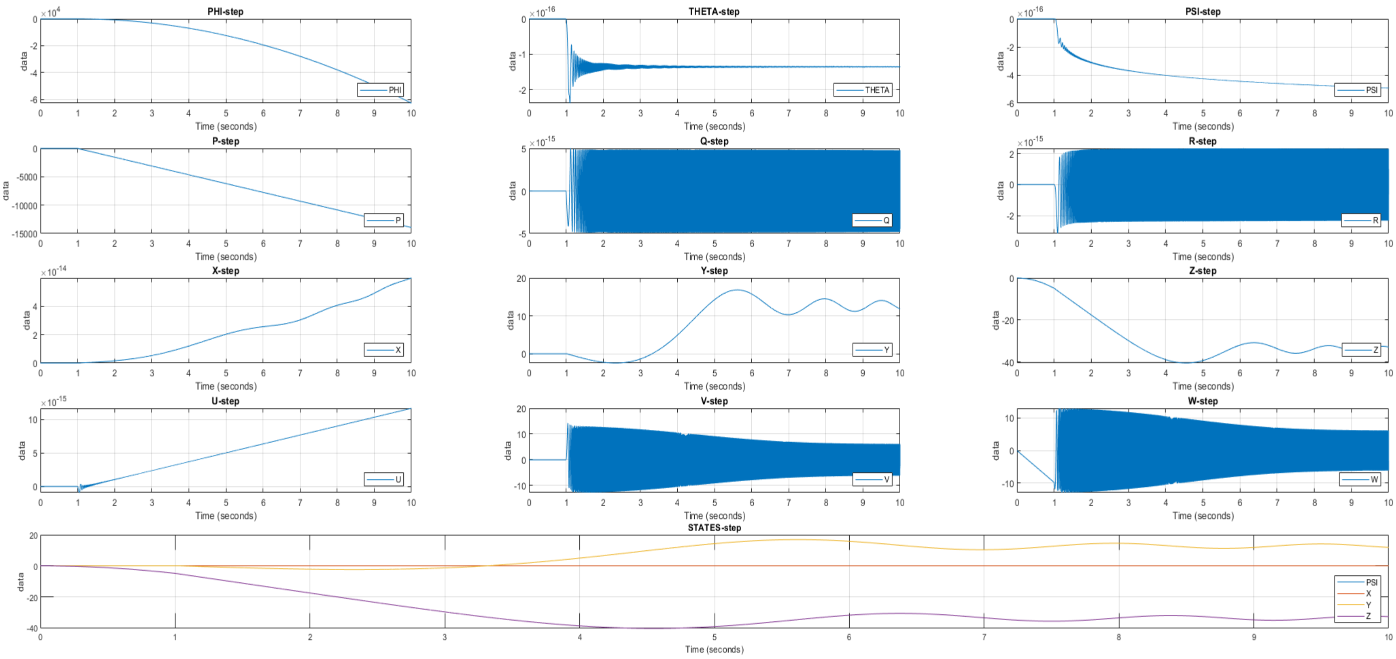
Figura 11: Schema Simulink non lineare

```
function [ phi_dot, p_dot, theta_dot, q_dot, psi_dot, r_dot, x_dot, u_dot, y_dot, v_dot,  z_dot, w_dot]= ...
       Quadcopter_nonlin( phi, p, theta, q, psi, r, x, u, y, v, z, w, u1, u2, u3, u4)

m_body=1.5; %massa del body
m_motore=0.05; %massa dei motori
m_bracci=0.0356; %massa dei bracci
m_gimbal=0.3; %massa della gimbal

m=m_gimbal+m_motore+m_bracci+m_body %massa complessiva del drone

r_gimbal=0.05; %raggio della sfera che rappresenta la gimbal
s_x_body=0.17; %semiasse x dell'ellissoide che rappresenta il body del drone
s_y_body=0.09; %semiasse y dell'ellissoide che rappresenta il body del drone
s_z_body=0.045; %semiasse z dell'ellissoide che rappresenta il body del drone

d_body=0.335;  %lunghezza dei bracci
d_gimbal=0.115;  %lunghezza della gimbal

Ks=0.9; %coefficiente portanza
g=9.81; %forza di gravità
alpha=pi/4 %angolo di assetto

l=0.1; %lunghezza elica
passo=4.7; %passo elica

%Calcolo baricentro
Z=(m_gimbal*d_gimbal)/(4*m_motore+m_body+4*m_bracci+m_gimbal);

%Calcolo momento di inerzia asse X
Ix=4*((m_motore)*(d_body^(2))*((cos(alpha))^(2))+(1/3)*m_bracci*(d_body^(2))*((sin(alpha))^(2)))...

+(m_body/5)*(s_y_body^(2)+s_z_body^(2))+[4*(m_motore+m_bracci)+m_body]*(Z^(2))+(2/5)*m_gimbal*r_gimbal^(2)...
    +m_gimbal*((d_gimbal-Z)^(2))

%Calcolo momento di inerzia asse y
Iy=4*((m_motore)*(d_body^(2))*((sin(alpha))^(2))+(1/3)*m_bracci*(d_body^(2))*((cos(alpha))^(2)))...

+(m_body/5)*(s_x_body^(2)+s_z_body^(2))+[4*(m_motore+m_bracci)+m_body]*(Z^(2))+(2/5)*m_gimbal*r_gimbal^(2)...
    +m_gimbal*((d_gimbal-Z)^(2))

%Calcolo momento di inerzia asse Z
Iz=4*((m_motore)*(d_body^(2))+(1/3)*m_bracci*(d_body^(2)))+(m_body/5)*(s_x_body^(2)+s_y_body^(2))...
    +(2/5)*m_gimbal*(r_gimbal^(2))

%Forze di Thrust
F1=Ks*(((u1)^(2))-((u3)^(2)));
F2=Ks*(((u2)^(2))-((u4)^(2)));
F4=Ks*(((u4)^(2))-((u2)^(2)));

%Equazione differenziali velocità angoli p,q,r
p_dot=((q*r)*((Iy-Iz)/(Ix)))+(((d_body)/(Ix))*(F1*sin(alpha)+F4*cos(alpha)))
q_dot=((p*r)*((Iz-Ix)/(Iy)))+(((d_body)/(Iy))*(F1*cos(alpha)+F2*sin(alpha)))
```

```
r_dot=((p*q)*((Ix-Iy)/(Iz)))+((Ks*l)/(Iz))*((((u4)^(2))+((u2)^(2)))-((u1)^(2))-((u3)^(2)))

%Equazione differenziali posizioni angolari rollio-roll-phi,beccheggio-pitch-theta, imbardata-yaw-psi
phi_dot=p+q*sin(phi)*tan(theta)+r*tan(theta)
theta_dot=(cos(theta))*q-sin(phi)*r
psi_dot=q*((sin(psi))/(cos(theta)))+r*((cos(psi))/(cos(theta)))

%Equazione differenziali velocità traslazionali u,v,w
u_dot=-q*w+r*v-g*sin(theta)
v_dot=-r*u+p*w+g*cos(theta)*sin(phi)
w_dot=-p*v+q*u-g*cos(theta)*cos(phi)-(Ks*(((u4)^(2))+((u2)^(2))+(u1)^(2)+((u3)^(2))))/m

%Equazione differenziali spostamento x,y,z
x_dot= u*(cos(theta)*cos(psi))+v*(sin(theta)*sin(phi)*cos(psi)-cos(phi)*sin(psi))+...
    +w*(sin(theta)*cos(phi)*cos(psi)+sin(phi)*sin(psi))
y_dot=u*(cos(theta)*sin(psi))+v*(sin(theta)*sin(phi)*sin(psi)+cos(phi)*cos(psi))+...
    +w*(sin(theta)*cos(phi)*sin(psi)-sin(phi)*cos(psi))
z_dot= -u*sin(theta)+v*(cos(theta)*sin(phi))+w*(cos(theta)*cos(phi))

end
```

Applicando un **ingresso** a **gradino**, ovvero ponendo la velocità **dell'i-esima elica** $\boldsymbol{\omega_i}$pari a un **gradino** $\boldsymbol{\delta_{-1}(t)}$ costante a partire da un certo istante modificandone l'**ampiezza** è possibile ottenere ad esempio il seguente risultato

- $\boldsymbol{u = [5\delta_{-1}(t) \quad 10\delta_{-1}(t) \quad 10(t) \quad 5\delta_{-1}(t)]}$



	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PHI	5.6921	9.9095	9.9095	0	0	62763
THETA	0.034073	5.5797	5.5797	76.06	0	2.3789e-16
PSI	4.8088	9	9	0.1945	0	4.9379e-16
P	7.2	9.82	9.82	0	0	13947
Q	0.047059	10	10	9.7455	112.79	5.0323e-15
R	0.0048336	10	10	170.1	366.09	3.1556e-15
X	6.2229	9.8713	9.8713	0	4.4537e-31	5.9908e-14
Y	1.0771	9.9415	9.9507	42.421	20.56	16.851
Z	2.168	9.5461	9.5461	23.102	0	40.371
U	7.0495	9.8117	9.8255	0.017255	7.8108	1.1662e-14
V	0.00087925	10	10	796.08	1004.7	14.278
W	0.48951	10	10	115.48	215.95	12.963

3.3 VELOCITÀ ANGOLARE IN CASO DI HOVERING

Figura 12

L'obbiettivo del controllo di un **drone** è far sì che abbia una **posizione fissa** nello **spazio**, ovvero che si trovi nella posizione di **hovering** in cui la **velocità angolare** di ogni **i-esimo rotore** $\boldsymbol{\omega_i}$ sia in grado di generare una **portanza** $\boldsymbol{P}$ in grado di opporsi la **forza peso** $\boldsymbol{mg}$ del **drone** e allo stesso tempo deve impedire che non ci siano degli squilibri nei **momenti** $\boldsymbol{M_x}, \boldsymbol{M_y}, \boldsymbol{M_z}$.

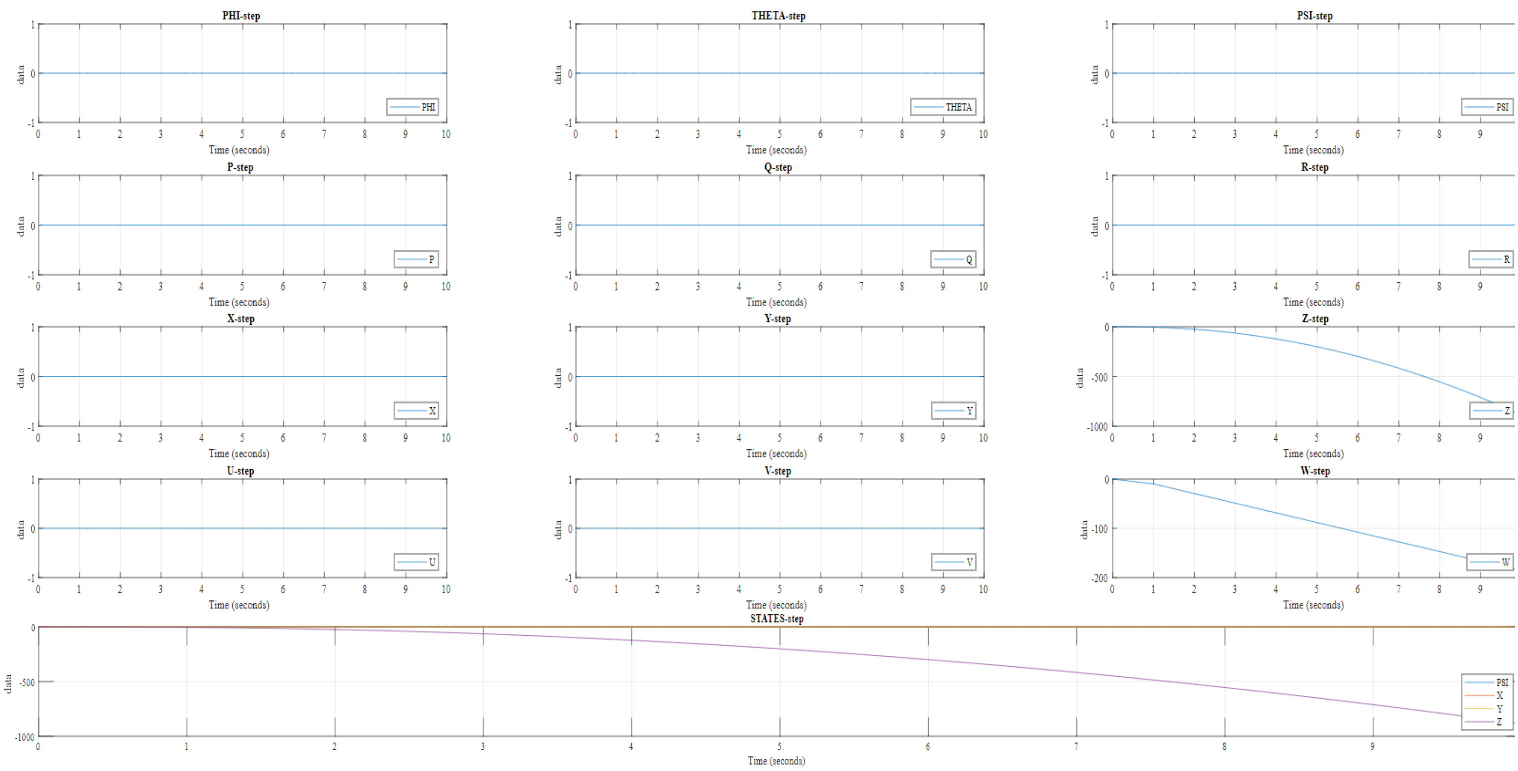
Per raggiungere questo scopo bisognerà far sì che le seguenti **variabili** di **stato** siano pari a:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \\ u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.3.1}$$

Questo risultato si ottiene quando:

$$\bar{u} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{gm}{4K_s}}\delta_1(t) \\ \sqrt{\frac{gm}{4K_s}}\delta_1(t) \\ \sqrt{\frac{gm}{4K_s}}\delta_1(t) \\ \sqrt{\frac{gm}{4K_s}}\delta_1(t) \end{bmatrix} = \begin{bmatrix} 2.266773036719821\delta_1(t) \\ 2.266773036719821\delta_1(t) \\ 2.266773036719821\delta_1(t) \\ 2.266773036719821\delta_1(t) \end{bmatrix} \tag{3.3.2}$$

Applicando questo ingresso si ottengono i seguenti risultati:



	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PHI	0	0	0	Inf	0	0
THETA	0	0	0	Inf	0	0
PSI	0	0	0	Inf	0	0
P	0	0	0	Inf	0	0
Q	0	0	0	Inf	0	0
R	0	0	0	Inf	0	0
X	0	0	0	Inf	0	0
Y	0	0	0	Inf	0	0
Z	6.0456	9.9037	9.9037	0	0	887.81
U	0	0	0	Inf	0	0
V	0	0	0	Inf	0	0
W	7.6	9.81	9.81	0	0	186.39

Per stampare le seguenti risposte sono stati utilizzati i seguenti comandi:

```
PHI_step = out.PHI; % Salvattaggio uscita PHI
THETA_step = out.THETA; % Salvattaggio uscita THETA
PSI_step = out.PSI; % Salvattaggio uscita PSI

P_step = out.P; % Salvattaggio uscita P
Q_step = out.Q; % Salvattaggio uscita Q
R_step = out.R; % Salvattaggio uscita R

X_step = out.X; % Salvattaggio uscita X
Y_step = out.Y; % Salvattaggio uscita Y
Z_step = out.Z; % Salvattaggio uscita Z

U_step = out.U; % Salvattaggio uscita U
V_step = out.V; % Salvattaggio uscita V
W_step = out.W; % Salvattaggio uscita W

figure('Name','Uscita Drone')

% Figura uscita PHI
subplot(5,3,1);
plot(PHI_step)
grid on
title(['PHI-step'])
legend('PHI', 'Location','southeast')

% Figura uscita THETA
subplot(5,3,2);
plot(THETA_step)
grid on
title(['THETA-step'])
legend('THETA', 'Location','southeast')

% Figura uscita PSI
subplot(5,3,3);
plot(PSI_step)
grid on
title(['PSI-step'])
legend('PSI', 'Location','southeast')

% Figura uscita P
subplot(5,3,4);
plot(P_step)
grid on
title(['P-step'])
legend('P', 'Location','southeast')

% Figura uscita Q
subplot(5,3,5);
plot(Q_step)
grid on
title(['Q-step'])
legend('Q', 'Location','southeast')
```

```
% Figura uscita R
subplot(5,3,6);
plot(R_step)
grid on
title(['R-step'])
legend('R', 'Location','southeast')

% Figura uscita X
subplot(5,3,7);
plot(X_step)
grid on
title(['X-step'])
legend('X', 'Location','southeast')

% Figura uscita Y
subplot(5,3,8);
plot(Y_step)
grid on
title(['Y-step'])
legend('Y', 'Location','southeast')

% Figura uscita Z
subplot(5,3,9);
plot(Z_step)
grid on
title(['Z-step'])
legend('Z', 'Location','southeast')

% Figura uscita U
subplot(5,3,10);
plot(U_step)
grid on
title(['U-step'])
legend('U', 'Location','southeast')

% Figura uscita V
subplot(5,3,11);
plot(V_step)
grid on
title(['V-step'])
legend('V', 'Location','southeast')

% Figura uscita W
subplot(5,3,12);
plot(W_step)
grid on
title(['W-step'])
legend('W', 'Location','southeast')
% Figura confronto uscite tutte gli stati
subplot(5,3,[13,14,15]);
plot(PHI_step)
hold on
plot(THETA_step)
plot(PSI_step)
plot(P_step)
plot(Q_step)
plot(R_step)
plot(X_step)
plot(Y_step)
plot(Z_step)
plot(U_step)
plot(V_step)
plot(W_step)
title(['Comparazione'])
legend('PHI','THETA','PSI','P','Q','R', 'X','Y', 'Z','U','V','W', 'Location','southeast')

% Salvataggio parametri uscite tutte gli PHI, THETA, PHI
PHI_step_info = stepinfo(PHI_step.Data,PHI_step.Time);
THETA_step_info = stepinfo(THETA_step.Data,THETA_step.Time);
PSI_step_info = stepinfo(PSI_step.Data,PSI_step.Time);

% Salvataggio parametri uscite tutte gli P, Q, R
P_step_info = stepinfo(P_step.Data,P_step.Time);
Q_step_info = stepinfo(Q_step.Data,Q_step.Time);
R_step_info = stepinfo(R_step.Data,R_step.Time);
```

```
% Salvataggio parametri uscite tutte gli X, Y, Z
X_step_info = stepinfo(X_step.Data,X_step.Time);
Y_step_info = stepinfo(Y_step.Data,Y_step.Time);
Z_step_info = stepinfo(Z_step.Data,Z_step.Time);

% Salvataggio parametri uscite tutte gli PHI, THETA, PHI
U_step_info = stepinfo(U_step.Data,U_step.Time);
V_step_info = stepinfo(V_step.Data,V_step.Time);
W_step_info = stepinfo(W_step.Data,W_step.Time);

%Crezione tabella parametri risposte stati
stepInfoTable = struct2table([PHI_step_info THETA_step_info PSI_step_info...
    P_step_info Q_step_info R_step_info...
    X_step_info Y_step_info Z_step_info...
    U_step_info V_step_info W_step_info]);

stepInfoTable = removevars(stepInfoTable,{...
    'SettlingMin','SettlingMax','PeakTime'});

stepInfoTable.Properties.RowNames = {'PHI','THETA','PSI',...
    'P','Q','R',...
    'X','Y', 'Z'...
    'U','V','W',};
stepInfoTable
```

3.4 LINEARIZZAZIONE

Il **modello** del **drone** che fino ad ora è stato trattato è **non lineare** su cui però non è possibile applicare un **controllo lineare** di alcun tipo, per questo motivo verrà **linearizzato** intorno ad un **punto di equilibrio**.

Figura 13

Il **punto di equilibrio** che verrà considerato in questo caso è la **posizione** di **hovering** in cui si ha un **volo livellato** in cui non si hanno **rotazioni** (come **imbardata**, **beccheggio**, **rollio**) né **traslazioni** (lungo gli **assi** $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$) e la **quota** rimane costante. In questo caso la **quota** viene fissata a **0** (con **0** non si intende ovviamente il drone appoggiato al suolo ma sospeso in aria come mostrato in **Figura 13**), ovvero:

$$\boldsymbol{x_0} = \begin{bmatrix} \phi_0 \\ \theta_0 \\ \psi_0 \\ p_0 \\ q_0 \\ r_0 \\ x_0 \\ y_0 \\ z_0 \\ u_0 \\ v_0 \\ w_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.4.1}$$

Questo **punto** di **equilibrio** si ottiene applicando l'**ingresso** $\overline{\boldsymbol{u}}$ visto nel capitolo precedente:

$$\overline{\boldsymbol{u}} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{gm}{4K_s}}\delta_1(t) \\ \sqrt{\frac{gm}{4K_s}}\delta_1(t) \\ \sqrt{\frac{gm}{4K_s}}\delta_1(t) \\ \sqrt{\frac{gm}{4K_s}}\delta_1(t) \end{bmatrix} = \begin{bmatrix} 2.2667730367198 21\delta_1(t) \\ 2.266773036719821\delta_1(t) \\ 2.266773036719821\delta_1(t) \\ 2.266773036719821\delta_1(t) \end{bmatrix} \tag{3.3.2}$$

Verranno utilizzate le seguenti istruzione per linearizzare il sistema:

```
clc
close all
clear all

load('massa_totale.mat') % caricamento massa totale drone
x0=zeros(12,1); % vettore punto di equlibrio
Ks=0.9; % coefficiente di portanza
g=9.81; % costante gravitazionale
w0=sqrt((g*m)/(4*Ks)) %ingresso per hovering
u0=w0*ones(4,1); % definzione vettore u_bar
[X,U,Y,DX,options]=trim('Copy_of_Quadcopter_linearization',x0,u0,[],[1,2,3,4,5,6,7,8,9,10,11,12],3) % cerca punto di equilibrio e ingresso u_bar
[A,B,C_R,D_R]=linmod('Copy_of_Quadcopter_linearization',X,U) %linearizza il sistema intorno al punto di equilibrio in funzione di X e di U
```

Ottenendo le seguenti matrici:

- ***A*:**

A

12x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	-1.0388e-17	0	0	3.3373e-17	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	1.0388e-17	0	0	1.0000	0	-2.0778e-17	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	9.6720e-09	9.6614e-17	1.0000	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	-1.6530e-34	0	1.2952e-25	0	4.2113e-17	0	0	1.0000	0	-9.6720e-09	0	3.3373e-17
8	0	0	-9.8100	-1.2819e-25	0	-4.2113e-17	0	0	0	-1.0388e-17	0	-9.6614e-17
9	-1.2942e-25	0	0	0	2.0162e-24	0	0	9.6720e-09	0	1.0000	0	-2.0778e-17
10	9.8100	1.3066e-25	0	0	0	-1.6098e-24	0	1.0388e-17	0	0	0	6.7630e-16
11	-4.2113e-17	0	-1.6087e-24	0	0	0	0	-3.3373e-17	0	2.0778e-17	0	1.0000
12	0	0	0	0	0	0	0	0	0	0	0	0

- ***B*:**

B

12x4 double

	1	2	3	4
1	0	0	0	0
2	-0.9222	0.6011	0.5823	1.2918
3	0	0	0	0
4	-0.7081	-0.4616	0.4471	-0.9919
5	0	0	0	0
6	0.2051	-0.1337	0.1295	0.2873
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0.0426	0.0278	0.0269	-0.0597

- **C_r:**

C_R

12x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	0	1

- D_r:

12x4 double

	1	2	3	4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0

Con le seguenti istruzioni elimino i canali di **uscita** che non sono rilevati ai fini del controllo:

```
C1=C_R(5,:) % Salvo il canale di uscita PSI
C2=C_R(7,:) % Salvo il canale di uscita X
C3=C_R(9,:) % Salvo il canale di uscita Y
C4=C_R(11,:) % Salvo il canale di uscita Z
C=[C1;C2;C3;C4] % Creo dei canali di uscita solo per le variabili di interesse PSI,X,Y,Z

D1=D_R(5,:); % Salvo il canale di uscita PSI
D2=D_R(7,:); % Salvo il canale di uscita X
D3=D_R(9,:); % Salvo il canale di uscita Y
D4=D_R(11,:); % Salvo il canale di uscita Z
D=[D1;D2;D3;D4] % Creo dei canali di uscita solo per le variabili di interesse PSI,X,Y,Z
```

Ottenendo le seguenti matrici:

- C:

C

4x12 double

1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0

- D:

D

4x4 double

	1	2	3	4
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

Utilizzando il seguente comando è possibile visualizzare nel dettaglio il **sistema linearizzato:**

```
sysDrone=ss(A,B,C,D)
```

```
sysDrone =

  A = 
               x1          x2          x3          x4          x5          x6          x7          x8          x9         x10         x11         x12
   x1           0           1  -1.039e-17           0           0   3.337e-17           0           0           0           0           0           0
   x2           0           0           0           0           0           0           0           0           0           0           0           0
   x3   1.039e-17           0           0           1           0  -2.078e-17           0           0           0           0           0           0
   x4           0           0           0           0           0           0           0           0           0           0           0           0
   x5           0           0           0   9.672e-09   9.661e-17           1           0           0           0           0           0           0
   x6           0           0           0           0           0           0           0           0           0           0           0           0
   x7  -1.653e-34           0   1.295e-25           0   4.211e-17           0           0           1           0  -9.672e-09           0   3.337e-17
   x8           0           0       -9.81  -1.282e-25           0  -4.211e-17           0           0           0  -1.039e-17           0  -9.661e-17
   x9  -1.294e-25           0           0           0   2.016e-24           0           0   9.672e-09           0           1           0  -2.078e-17
  x10        9.81   1.307e-25           0           0           0    -1.61e-24           0   1.039e-17           0           0           0   6.763e-16
  x11  -4.211e-17           0  -1.609e-24           0           0           0           0  -3.337e-17           0   2.078e-17           0           1
  x12           0           0           0           0           0           0           0           0           0           0           0           0
```

```
B =
            u1         u2         u3         u4
  x1         0          0          0          0
  x2   -0.9222     0.6011     0.5823      1.292
  x3         0          0          0          0
  x4   -0.7081    -0.4616     0.4471    -0.9919
  x5         0          0          0          0
  x6    0.2051    -0.1337     0.1295     0.2873
  x7         0          0          0          0
  x8         0          0          0          0
  x9         0          0          0          0
  x10        0          0          0          0
  x11        0          0          0          0
  x12   0.0426    0.02777     0.0269   -0.05968


C =
       x1   x2   x3   x4   x5   x6   x7   x8   x9  x10  x11  x12
   y1   0    0    0    0    1    0    0    0    0    0    0    0
   y2   0    0    0    0    0    0    1    0    0    0    0    0
   y3   0    0    0    0    0    0    0    0    1    0    0    0
   y4   0    0    0    0    0    0    0    0    0    0    1    0

D =
       u1  u2  u3  u4
   y1   0   0   0   0
   y2   0   0   0   0
   y3   0   0   0   0
   y4   0   0   0   0
```

Con le seguenti istruzioni è stato quindi possibile ottenere il modello linearizzato del drone, ovvero:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases} \tag{3.3.2}$$

Autovalori matrice $\boldsymbol{A}$:

```
>> eig(A)

ans =

   1.0e-16 *

   0.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.1039i
   0.0000 - 0.1039i
   0.0000 + 0.1039i
   0.0000 - 0.1039i
   0.0000 + 0.0000i
   0.9661 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
   0.0000 + 0.0000i
```

4.1 RETROZIONE DELLO STATO CON ALLOCAZIONE DEGLI AUTOVALORI

In questo capitolo verrà stabilito se il **sistema linearizzato** del **drone** rispetta o meno le condizioni per essere controllato tramite la **retroazione statica** dello **stato** con l'**allocazione** degli **autovalori** o **pole placement.**
Per poter essere applicata la **retroazione statica** dello **stato** il **sistema** definito dall'equazione (3.3.2) deve rispettare una delle due **proprietà**:

- La coppia $(\boldsymbol{A}, \boldsymbol{B})$ è completamente controllabile
- La coppia $(\boldsymbol{A}, \boldsymbol{B})$ è stabilizzabile

Per stabilire se rispetti o meno una di queste due **proprietà** calcolo la **matrice** di **controllabilità** $\mathcal{C}$, il suo **rango** e la **forma canonica** di **controllo** di **Kallmann** con i seguente comandi:

```
%% check per la controllabilità %%

Co=ctrb(A,B) %Calcolo matrice di controllabilità
r_Co=rank(Co) %Calcolo del rango matrice di controllabilità

[Abar,Bbar,Cbar,T,K]=ctrbf(A,B,C) %Calcolo della forma canonica di Kallmann di controllabilità
```

Ottenendo i seguenti risultati:

- <u>Rango della matrice di controllabilità</u> $\mathcal{C}$:

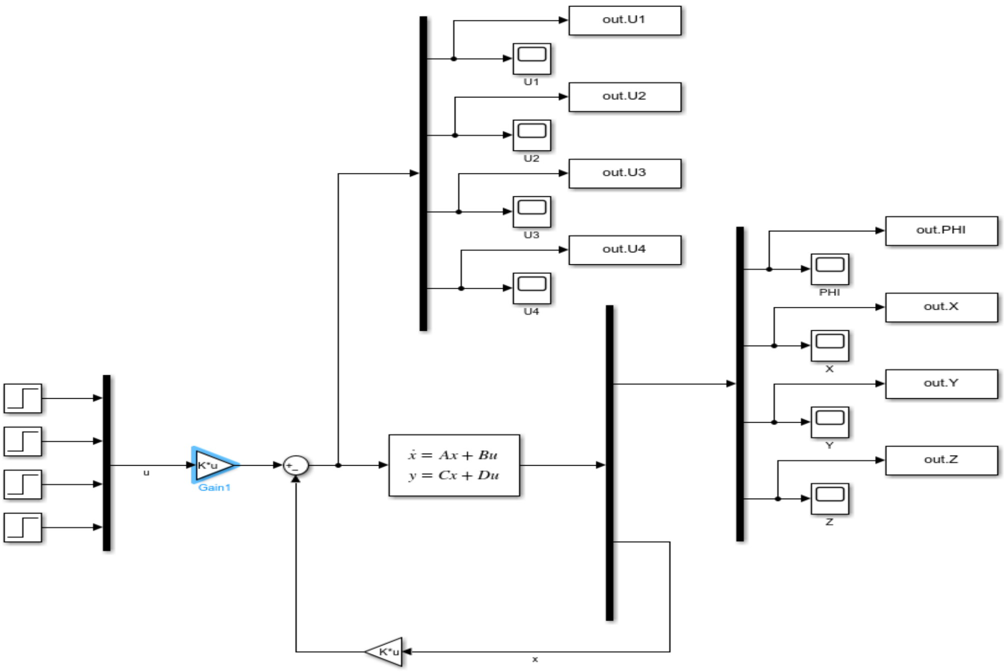
```
r_Co =

    12
```

Avendo la **matrice** di **controllabilità** $\mathcal{C}$ **rango pieno,** ovvero $rank(\mathcal{C}) = n$ il sistema sarà **completamente controllabile** e quindi è possibile **allocare** gli **autovalori** con la seguente **legge** di **controllo:**

$$u = -Kx + K_{ff}r \tag{4.1.1}$$

Dove:

- x: sono gli **stati** da **retroazionare**
- K: **matrice** che compie un'azione di **feedback** permette di modificare il **transitorio** del **sistema**
- K_{ff}: **matrice** che compie un'azione di **feed-forward** permette di modificare il **regime** del **sistema**



Utilizzando la **legge** di **controllo** sarà possibile trasformare la **matrice dinamica** A in $(A - BK)$ e in questo modo sarà possibile allocare a piacimento gli **autovalori** del sistema.
Essendo il sistema di ordine 12 si è optato per un'allocazione degli autovalori a **due poli dominanti** mentre i restanti **poli** verranno allocati ad una decade di distanza.

La scelta degli **autovalori** è stata basata sulla volontà di rispettare determinate specifiche **dinamiche** ovvero:

- <u>Percentage Overshoot o Percentuale di Sovraelongazione o</u> PO: inferiore al 3%
- <u>Settling Time o Tempo di assestamento</u> t_s: inferiore al $4s$

Queste due grandezze sono definite dalle seguenti espressioni:

$$\boldsymbol{PO = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}} \tag{4.1.1}$$

$$\boldsymbol{t_s \cong \frac{4}{\zeta\omega_n}} \tag{4.1.2}$$

Da queste relazioni si possono ricavare dei valori fondamentali per il calcolo della coppia di **poli dominanti**, ovvero:

- <u>Pulsazione naturale</u> $\boldsymbol{\omega_n}$:

$$\boldsymbol{\zeta = \frac{\left(\left|log\left(\frac{PO}{100}\right)\right|\right)}{\sqrt{\pi^2 + log^2\left(\frac{PO}{100}\right)}}} \tag{4.1.3}$$

- <u>Smorzamento</u> $\boldsymbol{\zeta}$:

$$\boldsymbol{t_s \cong \frac{4}{\zeta t_s}} \tag{4.1.4}$$

$$\boldsymbol{p_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}} \tag{4.1.5}$$

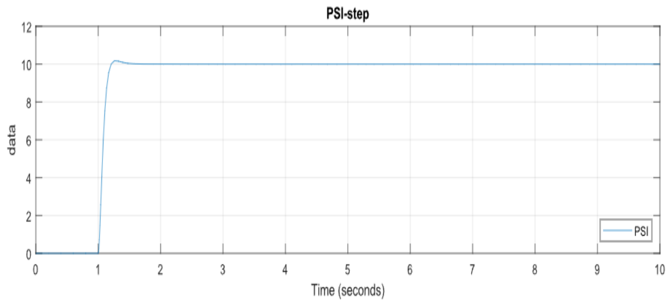
Sono stati allocati gli autovalori con i seguenti comandi:

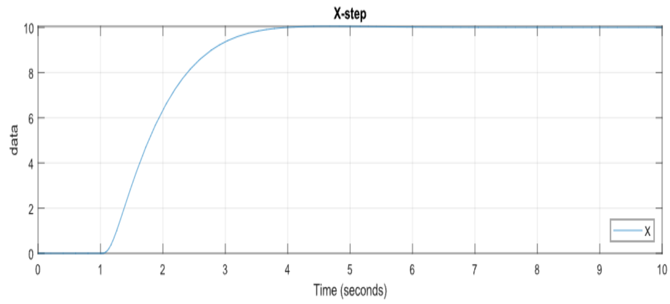
```
T_sett=4 % Settling Time
PO=0.3 % Percentage Overshoot
zita=abs(log(PO/100))/(sqrt(pi^(2)+log(PO/100)^(2))) % Calcolo Smorzamento

wn=4/(zita*T_sett) % Pulsazione naturale

p1= -wn*zita + 1j*wn*sqrt(1-zita^(2)) % calcolo primo polo

p2 = 10*real(p1) % calcolo terzo polo

p_star = [ p1 p1' p2 p2-1 p2-2 p2-3 p2-4 p2-5 p2-6 p2-7 p2-8 p2-9]; % definizione poli

K=place(A,B,p_star); % Allocazione autovalori

sys_alloc=ss(A-B*K,B,C-D*K,D); % Sistema retroazionato
sort(eig(A-B*K)) % Autovalori sistema retroazionato
Kfb=dcgain(sys_alloc) % Guadagno sistema
Kff=pinv(Kfb) % Calcolo matrice precompensatore
```

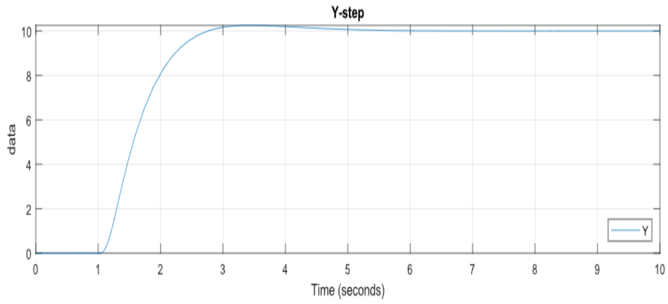
Si otterranno i seguenti autovalori della matrice $(\boldsymbol{A - BK})$:

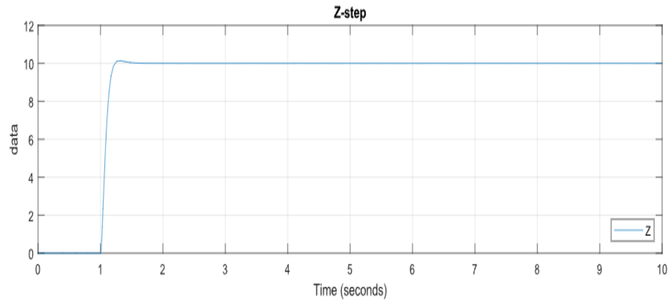
```
>> sort(eig(A-B*K))

ans =

  -1.3333 - 0.7211i
  -1.3333 + 0.7211i
 -13.3333 + 0.0000i
 -14.3333 + 0.0000i
 -15.3333 + 0.0000i
 -16.3333 + 0.0000i
 -17.3333 + 0.0000i
 -18.3333 + 0.0000i
 -19.3333 + 0.0000i
 -20.3333 + 0.0000i
 -21.3333 + 0.0000i
 -22.3333 + 0.0000i
```

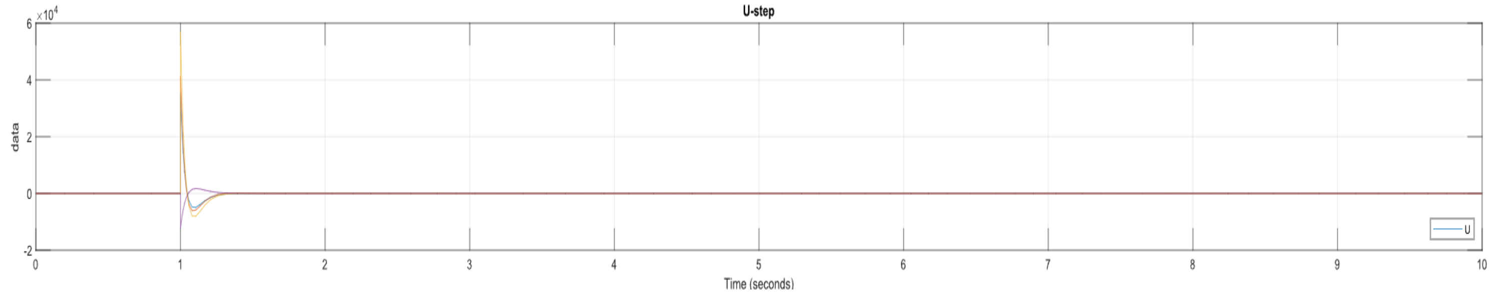
Ottenendo i seguenti risultati:











	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	0.12361	1.19	1.19	1.7547	0	10.176
X	1.5202	3.4594	3.4594	0.49459	1.4005e-29	10.049
Y	1.0223	3.9985	3.9985	2.5443	2.2019e-29	10.254
Z	0.1328	1.2035	1.2035	1.3465	0	10.135

In alternativa al metodo di **allocazione** degli **autovalori** a **due poli dominanti** si può usare il **metodo ITAE** esso cerca di **minimizzare** il seguente integrale:

$$\int_0^\infty t|e(t)|dt \tag{4.1.6}$$

Dove:

- $e(t)$: è l'**errore** tra un **riferimento** $r(t)$ a **gradino** $\delta_{-1}(t)$ e l'**uscita** $y(t)$ $e(t) = r(t) - y(t)$

I poli vengono scelti utilizzando la seguente funzione:

```
function pITA = ITAEpol(wn,n)
%PITA Summary of this function goes here
%   Detailed explanation goes here
switch n
    case(1)
        pITA=roots([1 wn]);
    case(2)
        pITA=roots([1 1.4*wn wn^2]);
    case(3)
        pITA=roots([1 1.75*wn 2.15*wn^2 wn^3]);
    case(4)
        pITA=roots([1 2.1*wn 3.4*wn^2 2.7*wn^3 wn^4]);
    case(5)
        pITA=roots([1 2.8*wn 5.0*wn^2 5.5*wn^3 3.4*wn^4 wn^5]);
    case(6)
        pITA=roots([1 3.25*wn 6.6*wn^2 8.6*wn^3 7.45*wn^4 3.95*wn^5 wn^6]);
    otherwise
        pITA=[];
```

```
        warning('CASE NON CONSIDERED IN ITAEPOL')
end
end
```

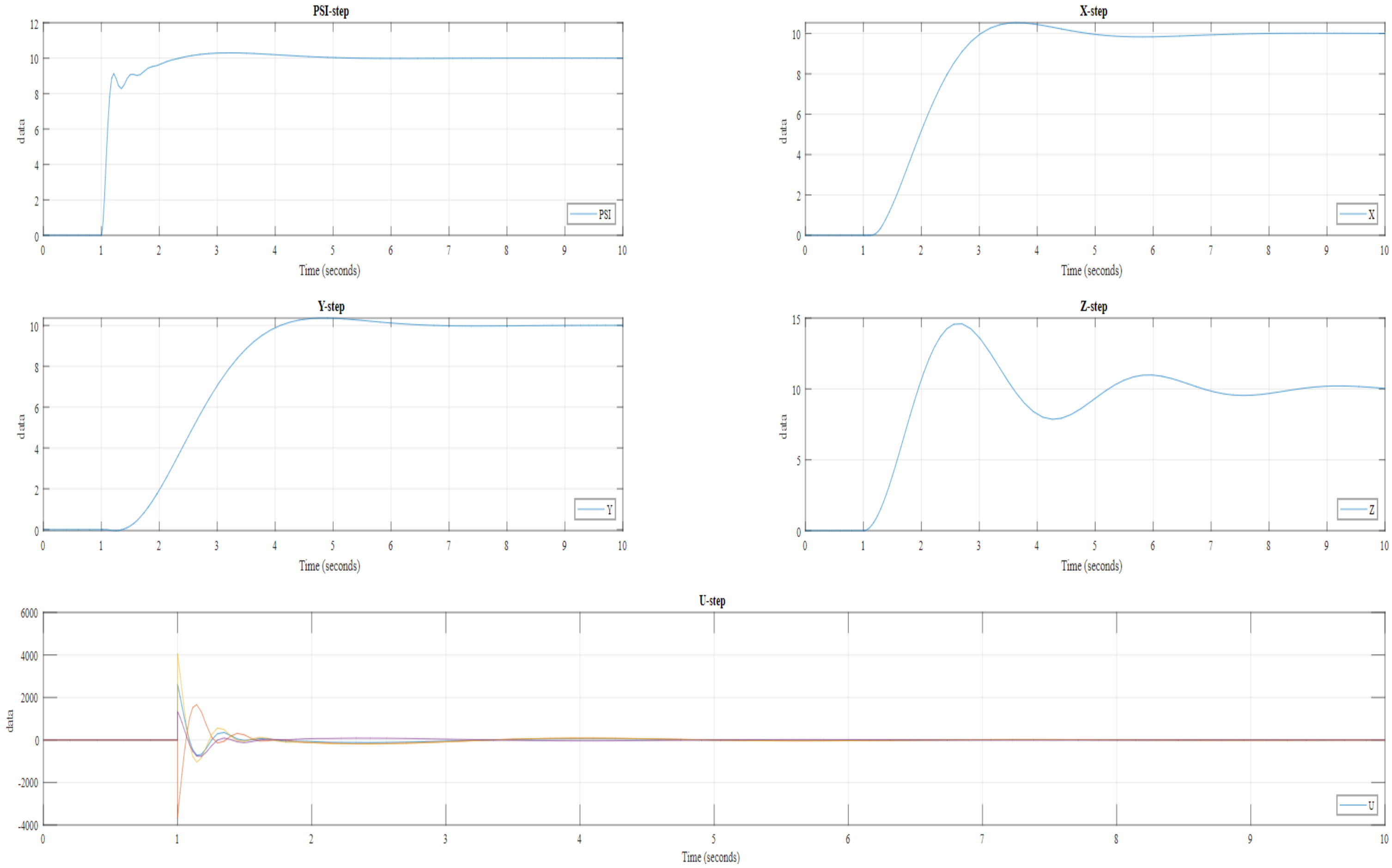
Eseguendo le seguenti istruzioni è possibile allocare i **poli** con il **metodo ITAE:**

```
p_6=ITAEpol(wn,6) % Scelta 6 poli ITAE

p_6=p_6' % Trasposizione vettore poli

p_ITAE=[p_6 10*p_6(1,1) 10*p_6(1,2) 10*p_6(1,3) 10*p_6(1,4) 10*p_6(1,5) 10*p_6(1,6)] % Definizione poli ITAE successivi al 6 ad una decade di distanza

K_ITAE=place(A,B,p_ITAE); % Allocazione poli

sys_alloc=ss(A-B*K_ITAE,B,C-D*K_ITAE,D); % Sistema Retroazionato
sort(eig(A-B*K_ITAE)); % Autovalori sistema retrozionato
Kfb_ITAE=dcgain(sys_alloc) % Calcolo guadagno sistema
Kff_ITAE=pinv(Kfb_ITAE) % Calcolo matrice precompesatore
```

Si otterranno i seguenti autovalori della matrice $(\boldsymbol{A}-\boldsymbol{BK})$:

```
>> sort(eig(A-B*K_ITAE))

ans =

  -1.1136 - 0.4354i
  -1.1136 + 0.4354i
  -0.8799 - 1.1866i
  -0.8799 + 1.1866i
  -0.4697 - 1.9150i
  -0.4697 + 1.9150i
 -11.1358 - 4.3542i
 -11.1358 + 4.3542i
  -8.7995 -11.8663i
  -8.7995 +11.8663i
  -4.6969 -19.1504i
  -4.6969 +19.1504i
```

Ottenendo i seguenti risultati:



	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	0.16169	3.9816	3.9816	2.9871	0	10.299
X	1.2571	4.4598	4.4599	5.2949	0.014354	10.533
Y	1.7723	5.6573	5.661	3.5537	0.49675	10.359
Z	0.63509	8.2547	8.2547	45.324	0	14.594

4.2 OSSERVATORE CON ALLOCAZIONE DEGLI AUTOVALORI

Nella realtà non è sempre possibile accedere agli **stati** del **sistema** per cui è fondamentale dotare il **controllore** di un **osservatore** in grado di **stimare** la **variabile vettoriale** di **stato** $\boldsymbol{x}$ tramite le uniche variabile note l'**ingresso** di **controllo** $\boldsymbol{u}(\boldsymbol{t})$ e **l'uscita** $\boldsymbol{y}(\boldsymbol{t})$ e sarà possibile tramite il principio di separazione progettare separatamente **retroazione** dello stato e **osservatore.**
Per poter essere applicato l'**osservatore** di **stato** $\widehat{\boldsymbol{x}}$, il **sistema** definito dall'equazione (3.3.2) deve rispettare una delle due **proprietà**:

- <u>La coppia $(\boldsymbol{A}, \boldsymbol{C})$ è completamente osservabile</u>
- <u>La coppia $(\boldsymbol{A}, \boldsymbol{C})$ è rilevabile</u>

Per stabilire se rispetti o meno una di queste due **proprietà** calcolo la **matrice** di **controllabilità** $\boldsymbol{\mathcal{O}}$ e il suo **rango** con i seguenti comandi:

```
%% check per la osservabilità %%

Ob=obsv(A,C) %Calcolo matrice di osservabilità
r_Ob=rank(Ob) %Calcolo del rango matrice di osservabilità
```

Ottenendo i seguenti risultati:

- <u>Rango della matrice di osservabilità</u> $\boldsymbol{\mathcal{O}}$**:**

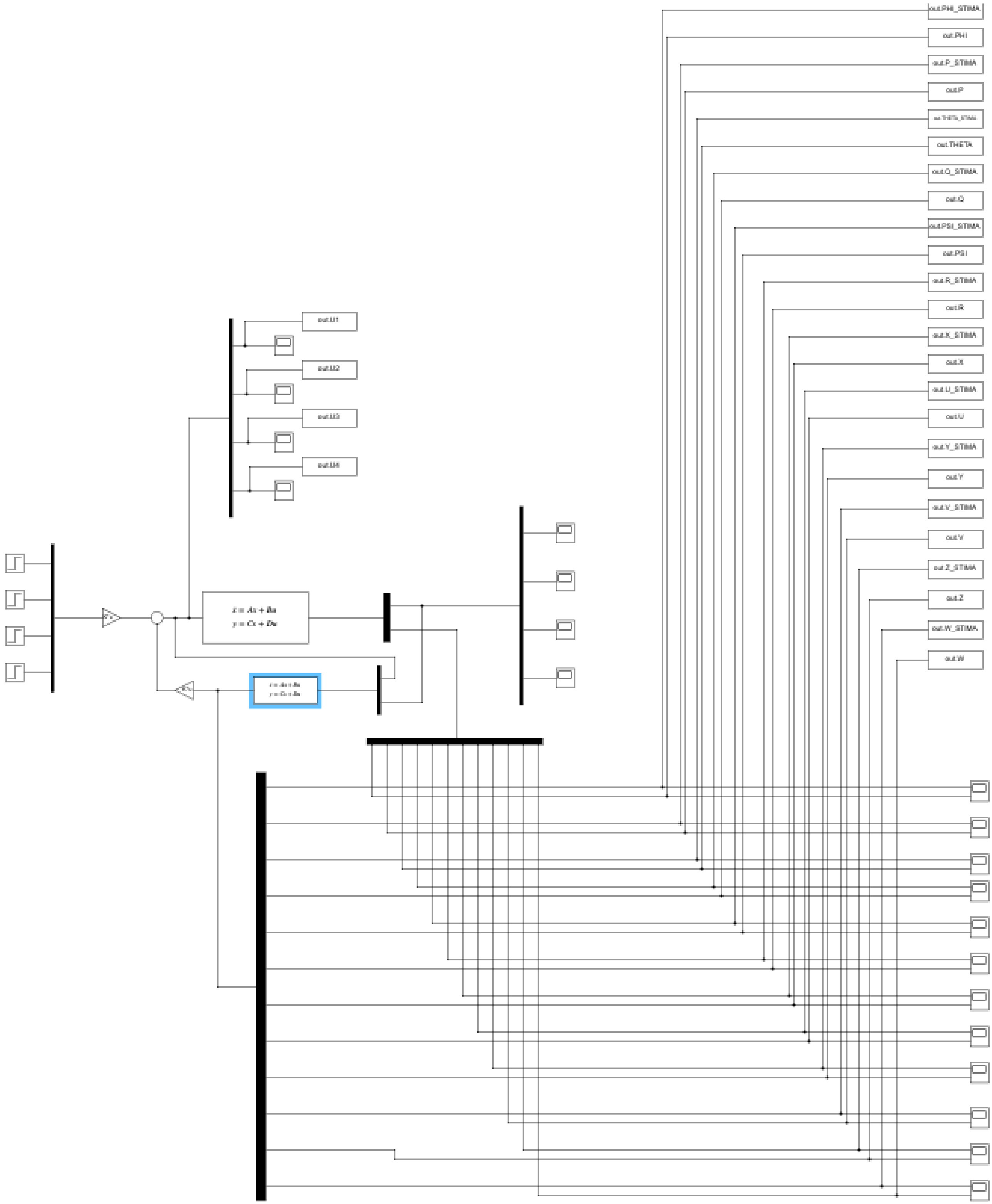
```
r_Ob =

    12
```

Avendo la **matrice** di **controllabilità** $\boldsymbol{\mathcal{O}}$ **rango pieno,** ovvero $\boldsymbol{rank}(\boldsymbol{\mathcal{O}}) = \boldsymbol{n}$ il sistema sarà **completamente osservabile** sarà quindi possibile utilizzare l'**osservatore** di stato che è retto dalla seguente equazione

$$\dot{\widehat{\boldsymbol{x}}} = \boldsymbol{A}\widehat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\widehat{\boldsymbol{x}} - \boldsymbol{D}\boldsymbol{u}) \quad (4.2.1)$$

Dove:

- $\boldsymbol{A}\widehat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u}$: è la **replica** dell'**impianto** $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$
- $\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\widehat{\boldsymbol{x}} - \boldsymbol{D}\boldsymbol{u})$: è il **termine correttivo**



Può essere definito l'**errore** di **stima** come:

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}} - \dot{\hat{\boldsymbol{x}}} \tag{4.2.2}$$

Si può dimostrare che esso è possibile definirlo anche come:

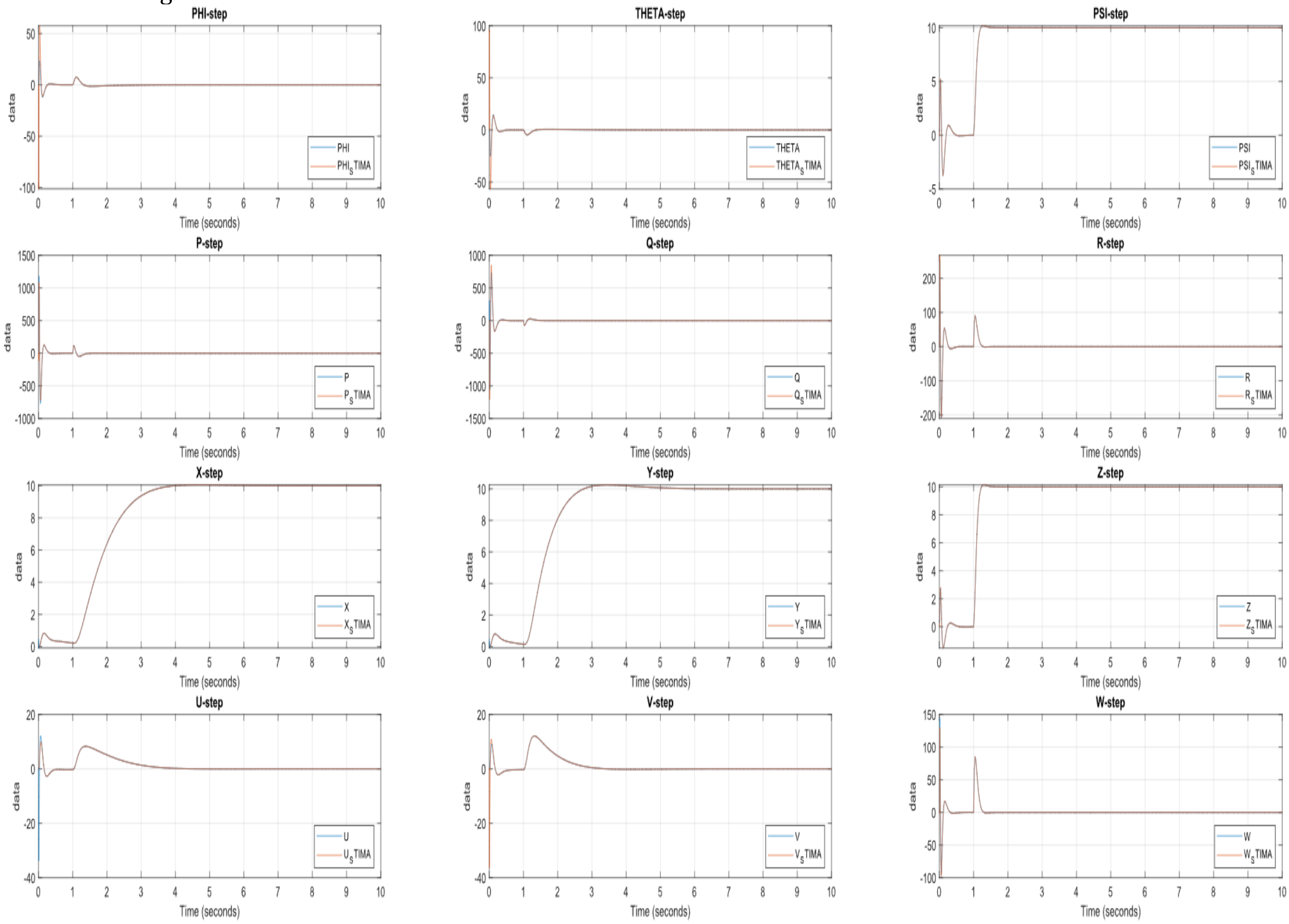
$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{LC})\boldsymbol{e} \tag{4.2.3}$$

Utilizzando la matrice $\boldsymbol{L}$ che è un **parametro** di **progetto** sarà possibile trasformare la **matrice** $(\boldsymbol{A} - \boldsymbol{LC})$ e scegliendo in maniera opportuna gli **autovalori dell'osservatore** sarà possibile scegliere in modo tale da far si che **l'evoluzione libera** dell'**errore** $\boldsymbol{e}$ converga a zero in modo tale che la sua **derivata** sia **nulla** a **regime** facendo si che la **stima** e lo **stato reale** siano uguali ossia, $\dot{\boldsymbol{x}} = \dot{\hat{\boldsymbol{x}}}$**.**

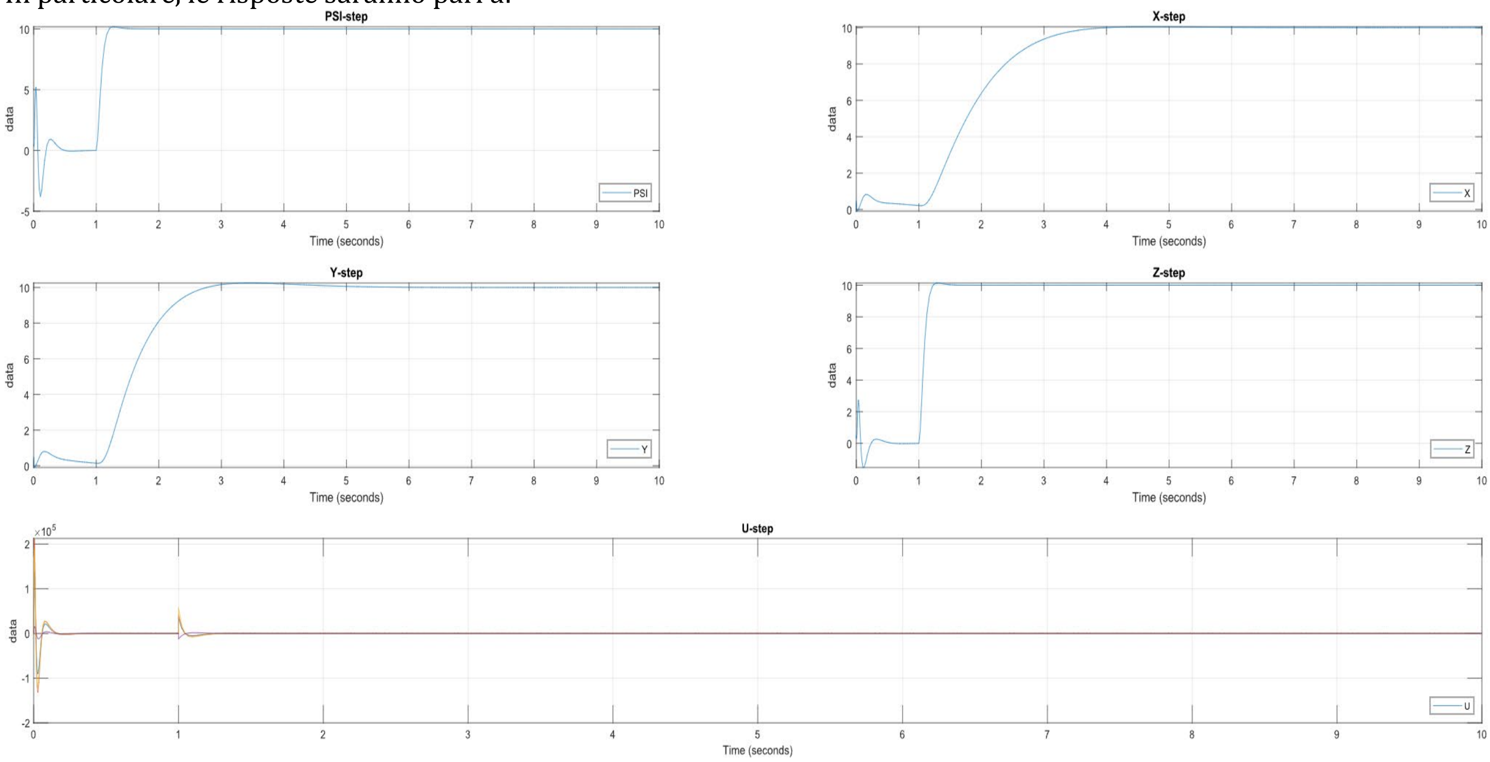
Per far si che i poli dell'**osservatore** non influenzino quelli della **retroazione** dello **stato** verranno posti a mezza decade di distanza utilizzando le seguenti istruzioni:

```
p_obs= 5*p_star % Scelta poli osservatore
L= place(A',C', p_obs)' % Allocazione poli osservatore
x0_obs= 0.5*[1 1 1 1 1 1 1 1 1 1 1 1 ] % Condizione iniziale osservatore
```

Ottenendo i seguenti risultati:



In particolare, le risposte saranno pari a:

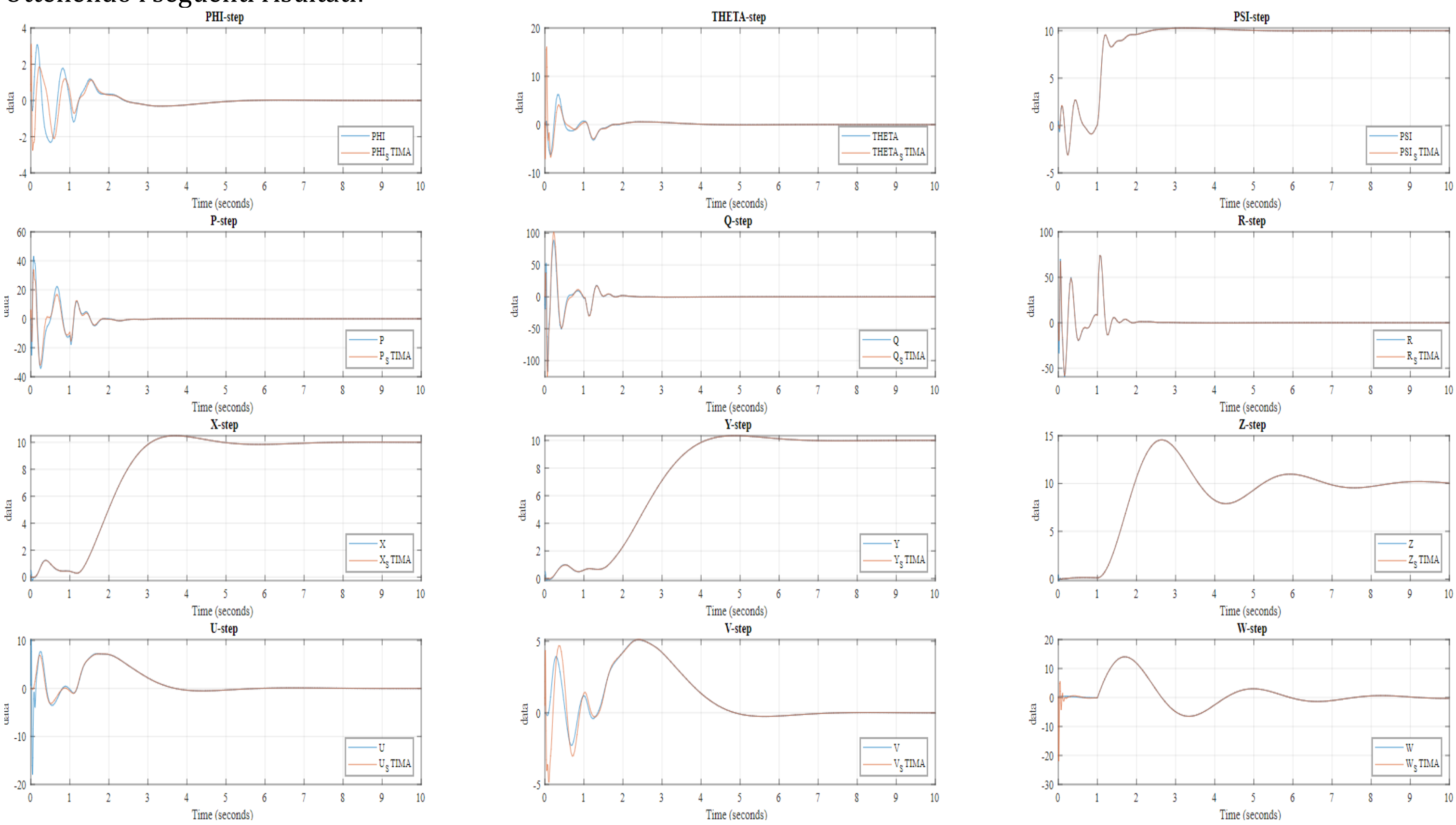


	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	1.1371	1.1833	1.1903	1.8169	37.882	10.182
X	1.5367	3.4523	3.4554	0.48101	0.94267	10.048
Y	1.031	3.9595	3.973	2.4893	0.97055	10.249
Z	1.1438	1.2024	1.2047	1.3848	15.323	10.139

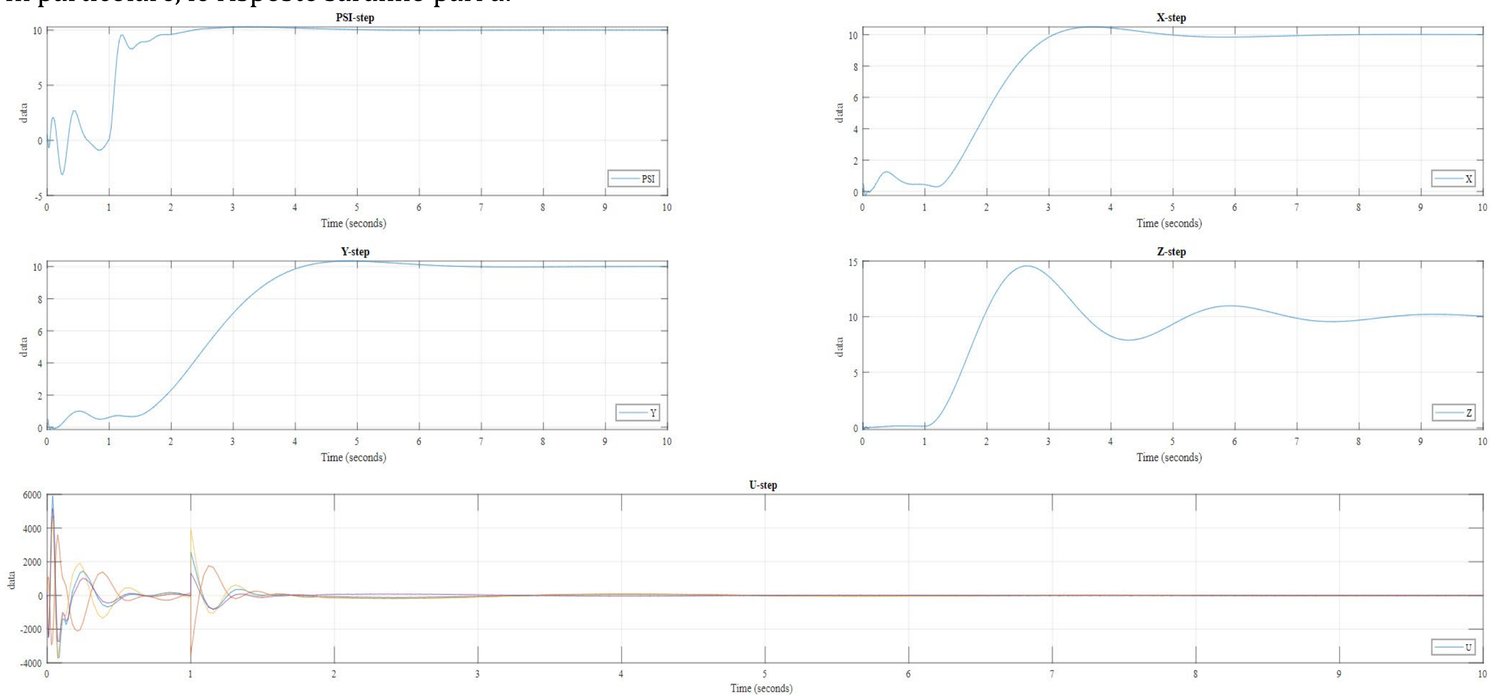
In alternativa al metodo di **allocazione** degli **autovalori** a **due poli dominanti** si può usare il **metodo ITAE** con le seguenti istruzioni:

```
p_obs_ITAE= 5*p_ITAE  % Scelta poli osservatore ITAE
L_ITAE= place(A',C', p_obs_ITAE)' % Allocazione poli osservatore ITAE
x0_obs_ITAE= 0.5*[1 1 1 1 1 1 1 1 1 1 1 1 ] % Condizione iniziale osservatore
```

Ottenendo i seguenti risultati:



In particolare, le risposte saranno pari a:



	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	1.1018	3.5456	3.9495	2.8586	31.169	10.287
X	2.423	4.4618	4.4722	4.8693	2.5556	10.49
Y	3.053	5.6468	5.6581	3.3946	1.4655	10.343
Z	0.65067	8.2515	8.2563	44.972	1.523	14.559

4.3 IMPIANTO AUMENTATO

Il **sistema** di **controllo** trattato nei capitoli precedenti non garantisce di essere un **controllo robusto,** infatti se il **modello** del **sistema** fosse affetto da **incertezza** ci potrebbero essere degli scostamenti delle **posizioni** delle **radici** del **polinomio caratteristico** delle **matrici** del **controllore** che potrebbero far sì che si trovino al di là dell'**asse immaginario**, ovvero $(\boldsymbol{A}+\boldsymbol{LC})$ e $(\boldsymbol{A}+\boldsymbol{BK})$.
Per risolvere questo problema si opta per un controllo utilizzando un **sistema aumentato**, che sarà un **sistema** caratterizzato dalla presenza di una retroazione dell'**uscita** $\boldsymbol{y(t)}$ e di un **blocco** di **integratori** $\frac{\boldsymbol{I}}{\boldsymbol{s}}$ sul **ramo** di **azione**.
La presenza dell'azione **integrale** garantisce:

- <u>Errore a regime nullo in presenza di riferimento a gradino</u>
- <u>Astatismo rispetto a disturbi costanti all'ingresso o all'uscita dell'impianto</u>

Per poter applicare questa tipologia di impianto il **sistema** deve soddisfare le seguenti due proprietà:

- <u>La coppia $(\boldsymbol{A_{aug}}, \boldsymbol{B_{aug}})$ è stabilizzabile</u>: dove:
  - $\boldsymbol{A_{aug}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ -\boldsymbol{C} & \boldsymbol{0} \end{bmatrix}$
  - $\boldsymbol{B_{aug}} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix}$
- <u>La coppia $(\boldsymbol{A}, \boldsymbol{C})$ è rilevabile</u>

Avendo già verificato la **seconda proprietà** e avendo stabilito che la **coppia** $(\boldsymbol{A}, \boldsymbol{C})$ è **completamente osservabile**, non manca che verificare la **prima proprietà** ovvero che la **coppia** $(\boldsymbol{A_{aug}}, \boldsymbol{B_{aug}})$ è **stabilizzabile** che verrà fatto con i seguenti comandi:

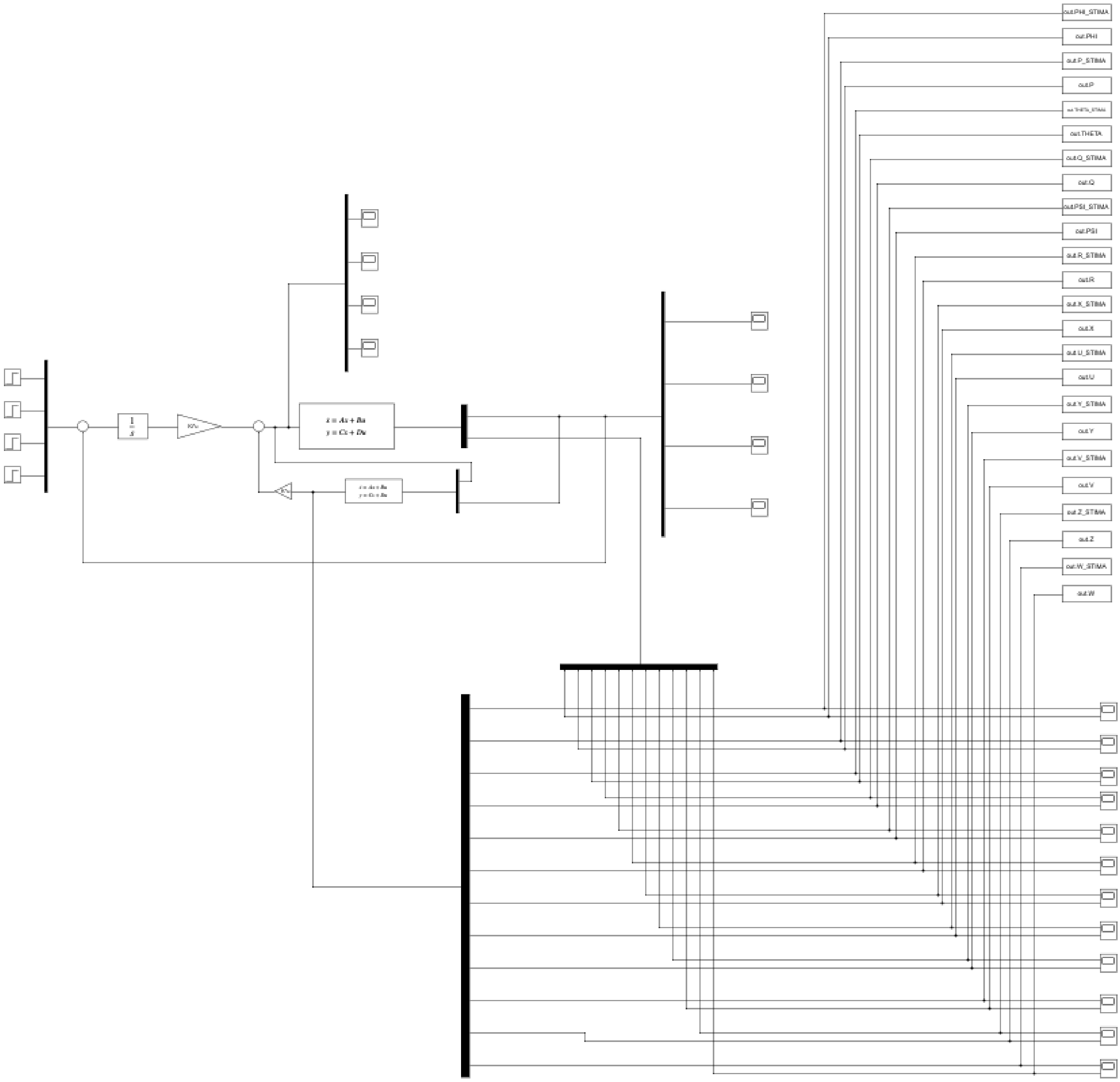
```
A_aug = [A zeros(12,4); -C zeros(4,4)] %  Matrice A sistema aumetato
B_aug = [B; zeros(4,4)] % Matrice B sistema aumetato
C_aug = [C zeros(4,4)] % Matrice C sistema aumetato
D_aug = D %  Matrice D sistema aumetato

Sys_aug = ss(A_aug,B_aug,C_aug,D_aug) % Sistema aumetato

Ctr_aug = ctrb(A_aug,B_aug) % Matrice di controllabilità sistema aumentato
rho_Ctr_aug = rank(Ctr_aug) % Calcolo rango matrice di controllabilità sistema aumentato
```

Ottenendo i seguenti risultati:

- <u>Rango della matrice di controllabilità</u> $\boldsymbol{\mathcal{C}_{aug}}$:

```
rho_Ctr_aug =

    16
```

Essendo la **matrice di controllabilità** $\boldsymbol{\mathcal{C}_{aug}}$ di **rango pieno** sarà possibile applicare questa tipologia di controllo.



La **legge** di **controllo** sarà definita dalla seguente espressione:

$$\boldsymbol{u(t) = -K_P x(t) - K_I x_I} \tag{4.3.1}$$

Dove:

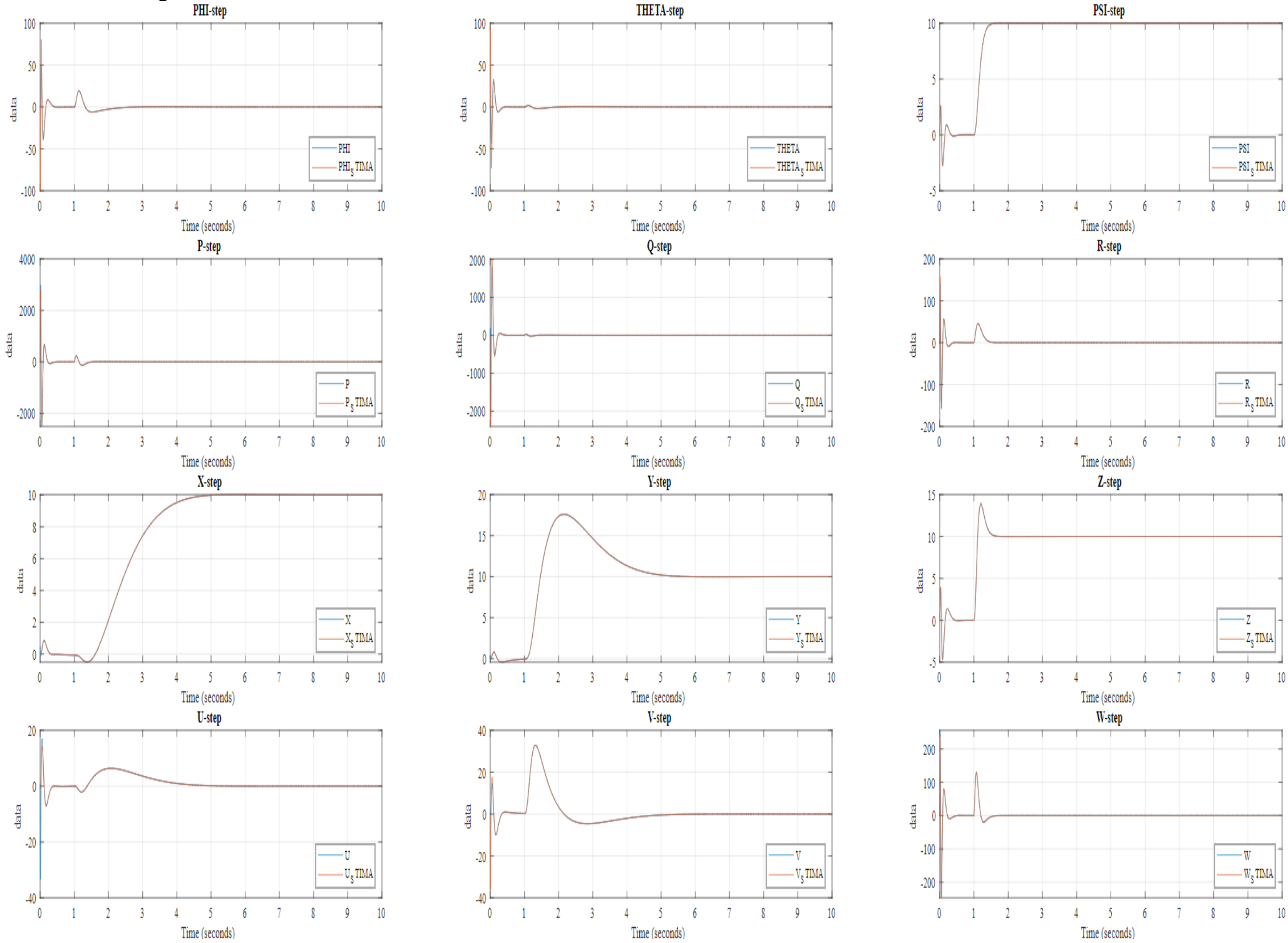
- <u>**Guadagno proporzionale**</u> $\boldsymbol{K_p}$**:** è la matrice $\boldsymbol{K}$ della **retroazione** di **stato**
- <u>**Guadagno integrale**</u> $\boldsymbol{K_I}$
- <u>**Variabile di stato integrale**</u> $\boldsymbol{x_I(t)}$**:** rappresenta l'**errore a regime** tra il **riferimento a gradino** $\boldsymbol{r(t) = \delta_{-1}(t)}$ e l'**uscita** $\boldsymbol{y(t)}$

$$\boldsymbol{\dot{x}_I(t) = \dot{r}(t) - \dot{y}(t)} \tag{4.1.4}$$

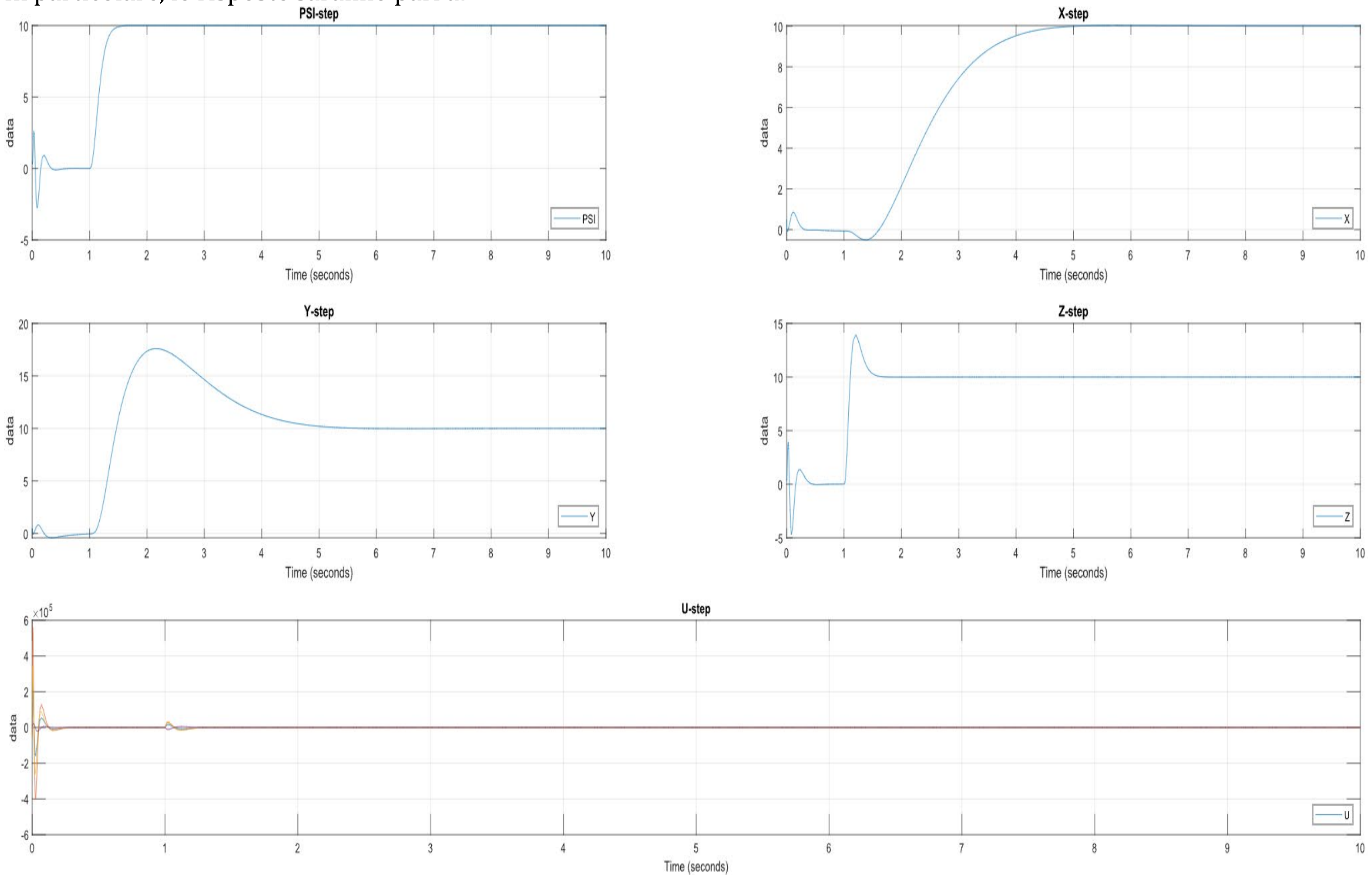
Con le seguenti istruzioni sarà possibile implementare l'**impianto aumentato** con **allocazione** degli **autovalori** con il **metodo** dei **due poli domninati:**

```
p_aug = [p_star 20*real(p1) 20*real(p1)-1 20*real(p1)-2 20*real(p1)-3] % Poli del Sistema Aumentato
K_aug = place(A_aug,B_aug,p_aug) % Allocazione poli sistema aumentato
Kp = K_aug(:,1:12) % Definizione Guadagno proporzianle
Ki = K_aug(:,13:16) % Definizione Guadagno integrale
```

Ottenendo i seguenti risultati:



In particolare, le risposte saranno pari a:

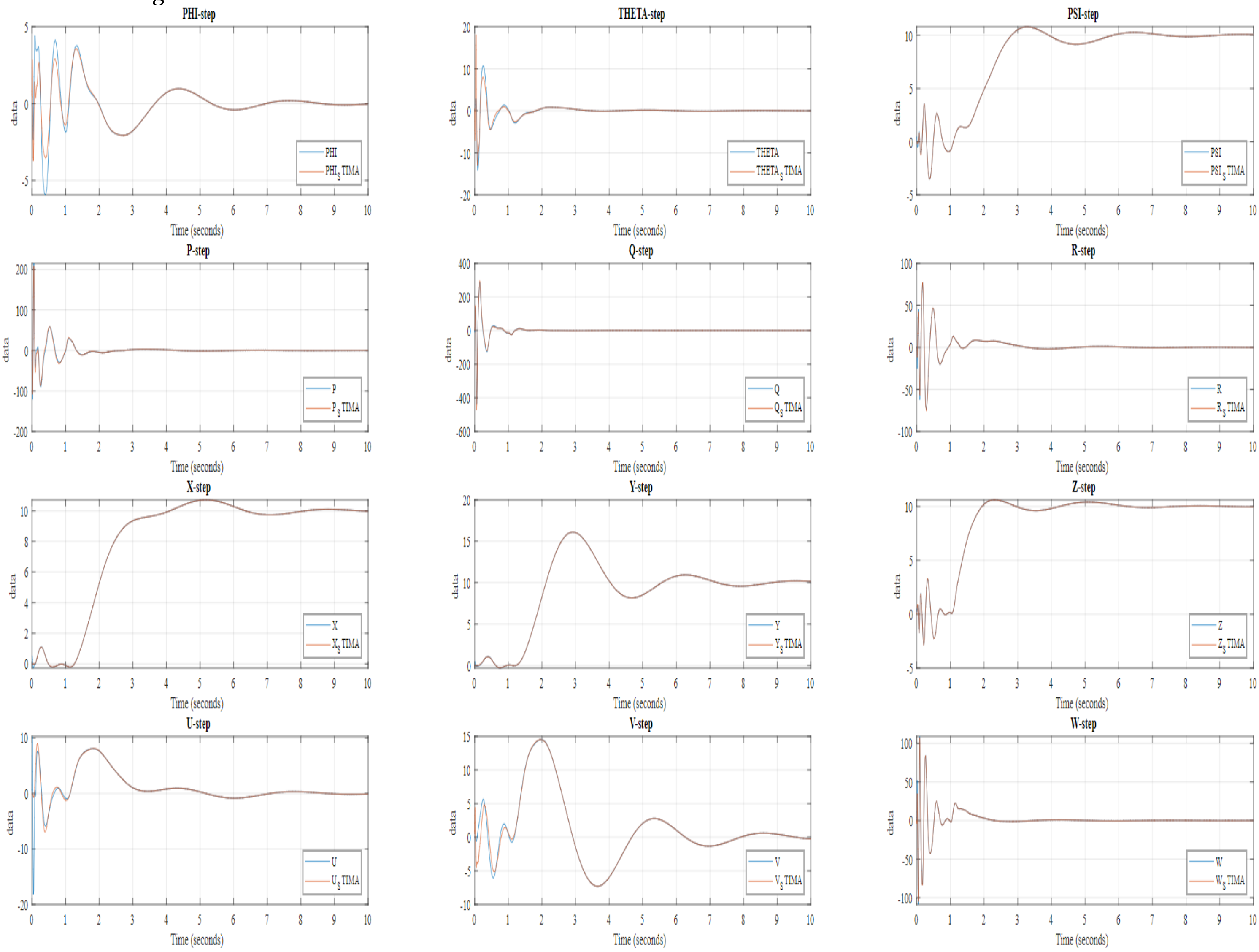


	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	1.3015	1.4235	1.4412	0.022592	27.694	10.002
X	1.7888	4.3454	4.3645	0.31583	5.0828	10.031
Y	0.26909	4.9905	5.0077	75.993	4.1862	17.599
Z	1.0906	1.5044	1.5355	39.577	46.805	13.957

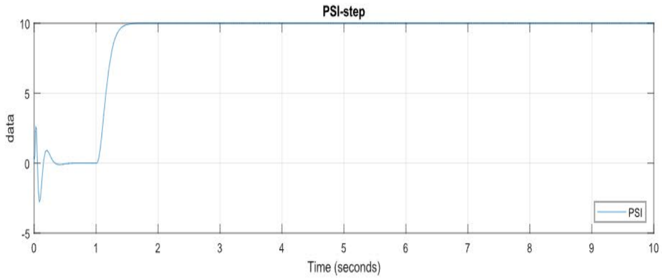
Con le seguenti istruzioni sarà possibile implementare l'**impianto aumentato** con **allocazione** degli **autovalori** con il **metodo** dei **ITAE:**

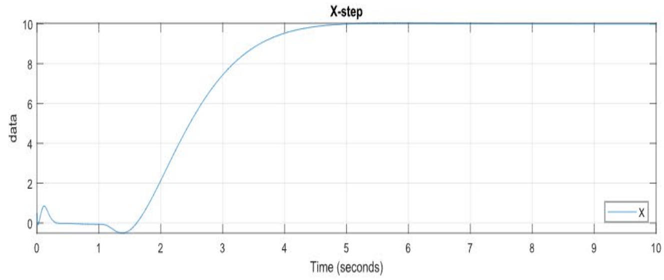
```
p_aug_ITAE=[p_ITAE 20*p_6(1,1) 20*p_6(1,2) 20*p_6(1,3) 20*p_6(1,4)] % Poli del Sistema Aumentato ITAE
K_aug_ITAE=place(A_aug,B_aug,p_aug_ITAE); % Allocazione poli sistema aumentato ITAE
Kp_aug_ITAE = K_aug_ITAE(:,1:12) % Definizione Guadagno proporzionale ITAE
Ki_aug_ITAE = K_aug_ITAE(:,13:16) % Definizione Guadagno integrale ITAE
```

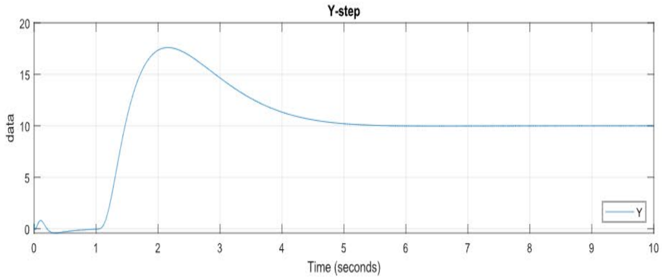
Ottenendo i seguenti risultati:

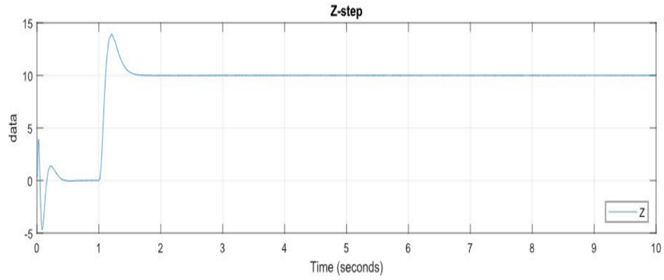


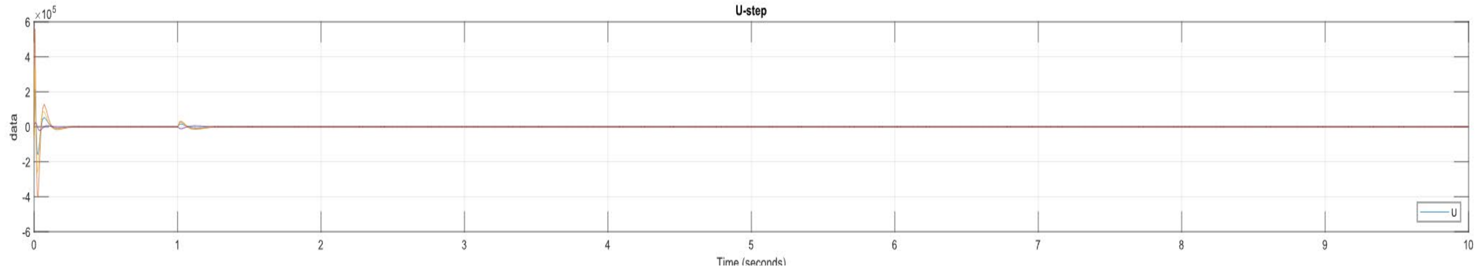
In particolare, le risposte saranno pari a:











	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	2.4215	5.5995	6.5885	7.3104	35.182	10.791
X	2.5353	7.4072	7.429	7.4052	2.7206	10.716
Y	1.7008	8.7178	8.7283	58.948	3.0592	16.109
Z	1.6368	5.7598	5.8818	6.5901	29.255	10.628

5.1 CONTROLLO OTTIMO LQ

In questo capitolo verrà applicata la **retroazione** dello **stato** tramite il **controllo ottimo LQ** o **Lineare Quadratico** esso si basa sul **principio del massimo** o di **Pontryagin** ma a differenza di quest'ultimo, da cui si riesce **legge** di **controllo ottima** a **ciclo aperto**, mentre il **controllo ottimo LQ** riesce a generare una **legge** di **controllo** a **ciclo chiuso**

Questa **legge** di **controllo ottimo** a **ciclo chiuso** $\boldsymbol{u(t)}$ si ottiene risolvendo il seguente problema di **minimizzazione,** ovvero sarà quella **legge** di **controllo ottima** $\boldsymbol{u^*(t)}$ che **minimizza** il seguente **funzionale di costo**:

$$\boldsymbol{J(x_0, u) = \int_0^{\infty} x^T Q x + u^T R u \, dt} \tag{5.1.1}$$

Dove:

- $\boldsymbol{Q \geq 0}$**:** è una **matrice** che permette di **minimizzare** le **traiettorie** degli **stati** $\boldsymbol{x(t)}$ del **sistema**
- $\boldsymbol{R > 0}$**:** è una **matrice** che permette di **minimizzare** le **leggi** di **controllo stati** $\boldsymbol{u(t)}$ del **sistema**

La legge di **controllo ottima** $\boldsymbol{u^*(t)}$ che si ottiene risolvendo questo problema sarà la seguente:

$$\boldsymbol{u(t) = -R^{-1} B^T \overline{P} x(t) = -K x(t)} \tag{5.1.2}$$

Dove:

- $\boldsymbol{P > 0}$**:** si ottiene risolvendo la seguente equazioni **Algebrica** di **Riccati:**

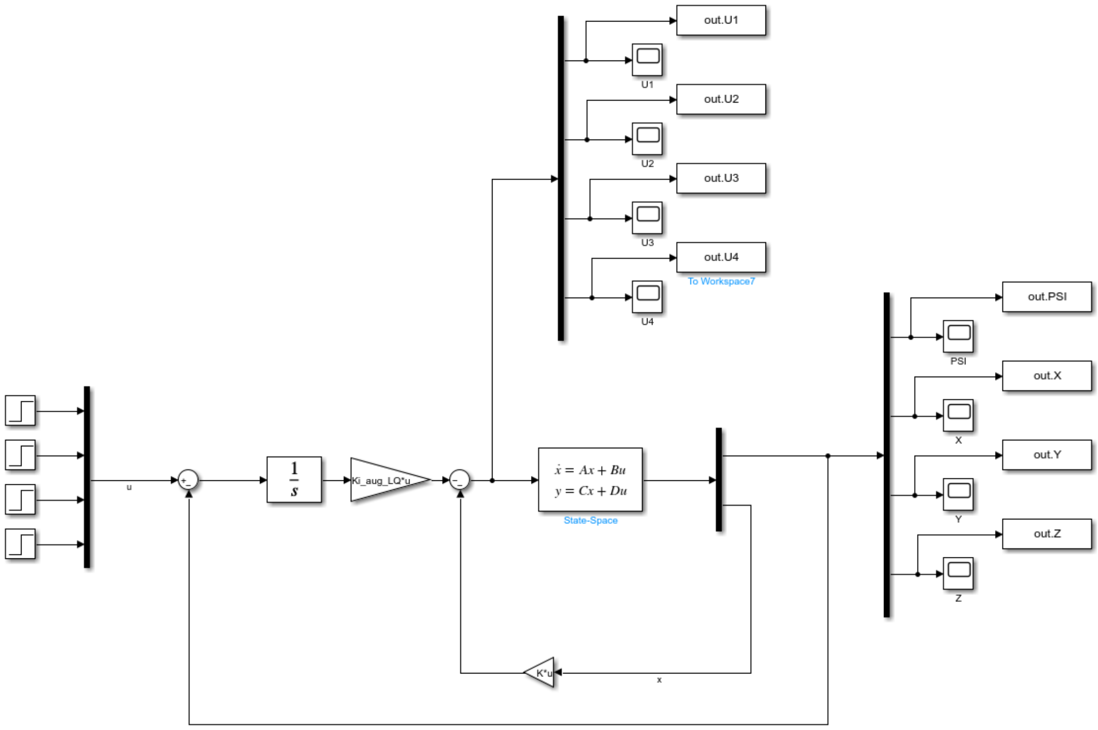
$$\boldsymbol{A^T \overline{P} + \overline{P} A^T + Q - \overline{P} B R^{-1} B^T \overline{P} = 0} \tag{5.1.3}$$

A questi risultati si può giungere se e soltanto se si rispettano le seguenti due **proprietà**:

- <u>La coppia $(\boldsymbol{A}, \boldsymbol{B})$ è stabilizzabile</u>
- <u>La coppia $(\boldsymbol{A}, \boldsymbol{C})$ è rilevabile</u>

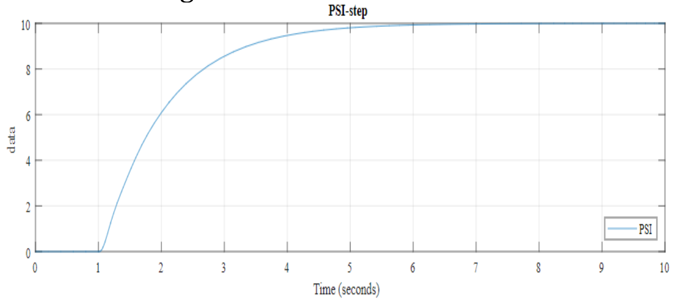
Altrimenti il **funzionale** di **costo** $\boldsymbol{J}(\boldsymbol{x_0}, \boldsymbol{u})$ definito nella equazione (5.1.1) **divergerebbe** e non sarebbe possibile risolvere la (5.1.3) ma bisognerebbe risolvere l'**equazione differenziale** di **Riccati.**
Essendo $(\boldsymbol{A}, \boldsymbol{B})$ **completamente controllabile** ed essendo $(\boldsymbol{A}, \boldsymbol{C})$ **completamente osservabile** è possibile applicare il **controllo LQ** ed avendo già costatato che il **sistema** rispetta le **proprietà** per l'applicazione dell'**impianto aumentato**, ovvero:

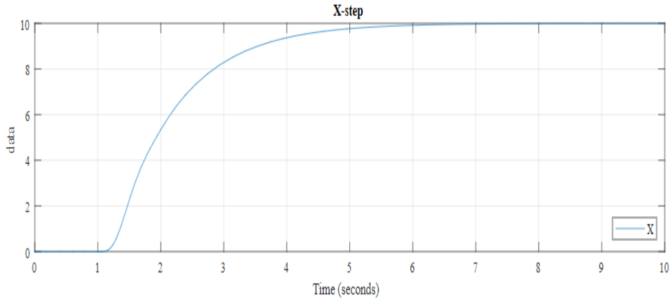
- <u>La coppia $(\boldsymbol{A_{aug}}, \boldsymbol{B_{aug}})$ è stabilizzabile</u>
- <u>La coppia $(\boldsymbol{A}, \boldsymbol{C})$ è rilevabile</u>

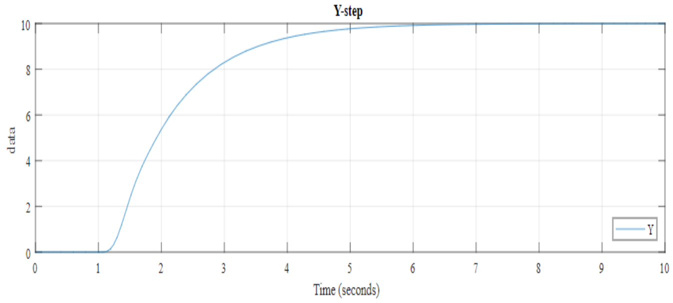


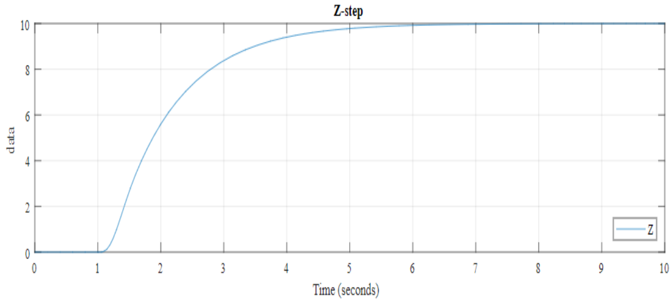

```
Q = 10*(C'*C) % Definizione matrice Q
R = 1e-5*diag([1 1 1 1]) % Definizione matrice Q
Q_aug = [Q zeros(12,4); zeros(4,12) eye(4)*10]; % Definizione matrice Q Aumentata
K_LQ_aug=lqr(A_aug,B_aug,Q_aug,R) % Calcolo matrice K
Kp_aug_LQ=K_LQ_aug(:, 1:12) % Guagnado proporzianale
Ki_aug_LQ=K_LQ_aug(:, 13:16) % Guagnado integrale
```

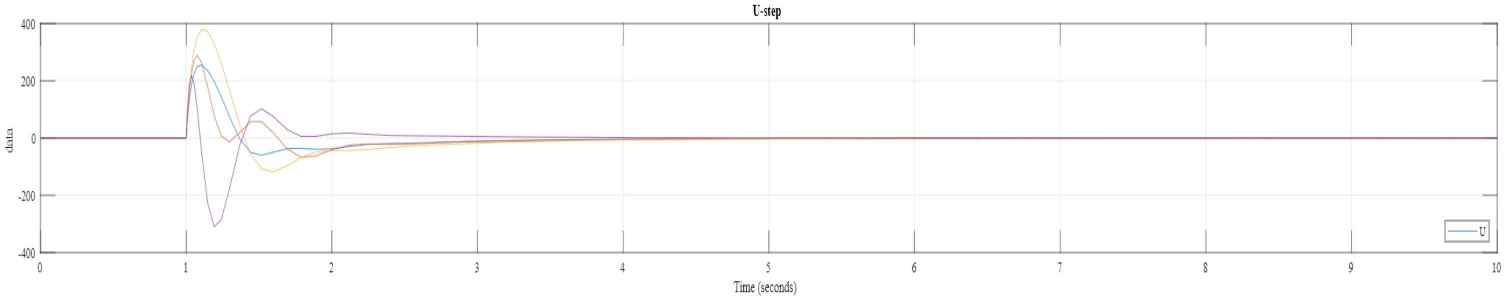
Ottenendo i seguenti risultati:











	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	2.1973	4.9656	4.9656	0	0	9.9986
X	2.1816	5.1395	5.1395	0	0	9.9984
Y	2.1873	5.1337	5.1337	0	0	9.9984
Z	2.1862	5.0848	5.0848	0	5.0405e-30	9.9985

5.2 CONTROLLO OTTIMO LQG

Il **controllo ottimo LQ** necessita della conoscenza degli **stati** del **sistema** per cui è fondamentale dotare il **controllore** di un **osservatore,** ma se si utilizzasse un osservatore tradizionale si perderebbe l'**ottimalità** del **sistema** per cui sarà necessario applicare utilizzare un **osservatore ottimo** ovvero il **filtro** di **Kallmann** che non solo permetterà di stimare in maniera **ottimale** lo **stato** ma permetterà anche di reiettare i **rumori** di **processo** $\boldsymbol{v_x}$ e i **rumori** di **misura** $\boldsymbol{v_y}$.
Infatti realisticamente il sistema avrà la seguente forma:

$$\begin{cases} \dot{x} = Ax + Bu + v_x; \\ y = Cx + Du + v_y; \end{cases} \quad x(t_0) = x_0 \tag{5.2.1}$$

Dove:

- <u>Rumore di processo</u> $\boldsymbol{v_x}$: è un **segnale vettoriale bianco gaussiano** a **media nulla** $E[v_x] = 0$
- <u>Rumore di misura</u> $\boldsymbol{v_y}$: è un **segnale vettoriale bianco gaussiano** a **media nulla** $E[v_y] = 0$

Essendo lo **stato** $\boldsymbol{x(t)}$ dipendente da un **segnale vettoriale stocastico** sarà anch'esso un **segnale stocastico** per cui non sarà possibile utilizzare il **problema** di **minimizzazione** del **funzionale di costo** $\boldsymbol{J(x_0, u)}$ definito nell'espressione (5.1.1) inquanto era un problema deterministico per cui si dovrà **minimizzare** la **potenza** della **media** del **funzionale** di **costo:**

$$J(x_0, u) = \lim_{T \to \infty} \frac{1}{T} \; E\left[\int_0^T x^T Q x + u^T R u \; dt\right] \tag{5.2.2}$$

Per poter applicare il **controllo LQG** il sistema deve rispettare contemporaneamente delle **proprietà** legate al **controllo LQ** e contemporaneamente delle **proprietà** legate al **filtro di Kallmann**, ovvero:

- <u>Controllo LQ</u>:
    - La **coppia** $(\boldsymbol{A}, \boldsymbol{H})$ deve essere **rilevabile**
    - La **coppia** $(\boldsymbol{A}, \boldsymbol{B})$ deve essere **stabilizzabile**
    - $\boldsymbol{Q} = \boldsymbol{H}^T\boldsymbol{H} \geq \boldsymbol{0}$
    - $\boldsymbol{R} > \boldsymbol{0}$
- <u>Filtro di Kallmann</u>:
    - La **coppia** $(\boldsymbol{A}, \boldsymbol{M})$ deve essere **stabilizzabile**
    - La **coppia** $(\boldsymbol{A}, \boldsymbol{C})$ deve essere **rilevabile**
    - $\boldsymbol{E}[\boldsymbol{v}_x] = \boldsymbol{0}$
    - $\boldsymbol{E}[\boldsymbol{v}_y] = \boldsymbol{0}$
    - $\boldsymbol{E}[\boldsymbol{v}_x\boldsymbol{v}_y^T] = \boldsymbol{0}$
    - $\boldsymbol{E}[\boldsymbol{v}_x\boldsymbol{v}_x^T] = \widetilde{\boldsymbol{Q}} = \boldsymbol{M}\boldsymbol{M}^T \geq \boldsymbol{0}$
    - $\boldsymbol{E}[\boldsymbol{v}_y\boldsymbol{v}_x^T] = \widetilde{\boldsymbol{R}} > \boldsymbol{0}$

Il **Filtro** di **Kallmann** sarà definito dalla seguente equazione:

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\overline{\boldsymbol{K}}\hat{\boldsymbol{x}} + \overline{\boldsymbol{L}}(\boldsymbol{y} - \boldsymbol{C}\hat{\boldsymbol{x}}) \tag{5.2.3}$$

Dove la **matrice** $\boldsymbol{L}$ è pari a

$$\overline{\boldsymbol{L}} = \overline{\boldsymbol{\Sigma}}\boldsymbol{C}\widetilde{\boldsymbol{R}}^{-1} \tag{5.2.4}$$

Dove $\overline{\boldsymbol{\Sigma}}$ si ricava dalla seguente **equazione** di **Algebrica** di **Riccati**

$$\mathbf{A}\overline{\boldsymbol{\Sigma}} + \overline{\boldsymbol{\Sigma}}\boldsymbol{A}^T + \widetilde{\boldsymbol{Q}} - \overline{\boldsymbol{\Sigma}}\boldsymbol{C}^T\widetilde{\boldsymbol{R}}^{-1}\boldsymbol{C}\overline{\boldsymbol{\Sigma}} = \boldsymbol{0} \tag{5.2.4}$$

Con le seguenti istruzioni verrà definito il **Filtro** di **Kallmann**

```
T_noise= 0.003; % Tempo di campionamento rumore
proc_std3=0.01; % Scarto quadratico medio del rumore di processo
meas_std3=0.01; % Scarto quadratico medio del rumore di misura

proc_var=(proc_std3/3)^(2); % Varianza del rumore di processo
meas_var=(meas_std3/3)^(2); % Varianza del rumore di processo

Qt=diag([1 1 1 1 1 1 1 1 1 1 1 1]*proc_var); % Definizione Q tilde
Rt=eye(4)*meas_var;  % Definizione Q tilde
Rank(Qt);  % Test di Sylvestre
Rank(Rt);  % Test di Sylvestre
```

Essendo la **coppia** $(\boldsymbol{A}, \boldsymbol{C})$ **completamente osservabile** ed essendosi verificate tutte le ipotesi del **Filtro** di **Kallmann** in quanto:

```
>> rank(Qt)

ans =

    12

>> rank(Rt)

ans =

     4
```

Manca da verificare se la **coppia** $(\boldsymbol{A}, \boldsymbol{M})$ sia **stabilizzabile**, lo si può fare con le seguenti istruzioni:

```
M=sqrt(Qt) % Definizione M
CO_2=rank(ctrb(A,M))  % Check stabilizzabilità
```

Ottenendo il seguente risultato:

```
CO_2 =

    12
```

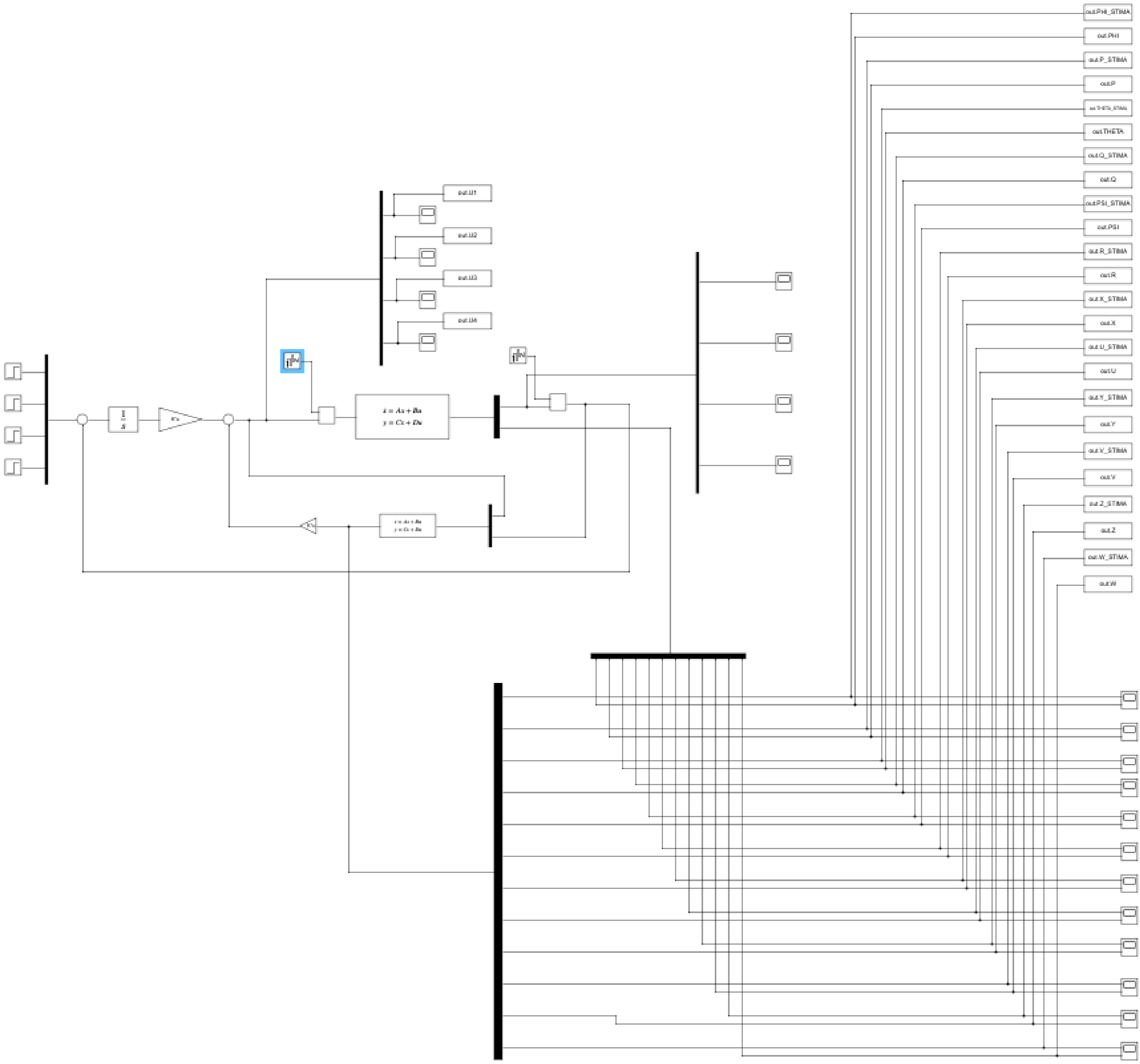
Per cui è possibile definire la **matrice** $\boldsymbol{L}$ di **Kallmann**, con la seguente istruzione:

```
L_Kalman=lqr(A',C',Qt,Rt)' % Calcolo matrice L Kallmann
```

Si utilizzeranno diverse matrici $\boldsymbol{Q}$ ed $\boldsymbol{R}$ per il **controllo LQ.**

```
%% LQ 0.5
Q_05=diag([1 10 1 10 1 10 1 10 1 10 1 10]);
R_05=diag([1 1 1 1]*0.5);
H_05=sqrt(Q_05);
rank(obsv(A,H_05))
Q_aug_05 = [Q_05 zeros(12,4); zeros(4,12) eye(4)*10];
K_LQ_aug_05=lqr(A_aug,B_aug,Q_aug_05,R)
Kp_aug_LQ_05=K_LQ_aug_05(:, 1:12)
Ki_aug_LQ_05=K_LQ_aug_05(:, 13:16)

%% LQ 100
Q_100=diag([100 0.1 100 0.1 100 0.1 100 0.1 100 0.1 100 1]);
R_100=diag([1 1 1 1]*100);
H_100=sqrt(Q_100);
rank(obsv(A,H_100))
Q_aug_100 = [Q_100 zeros(12,4); zeros(4,12) eye(4)*10];
K_LQ_aug_100=lqr(A_aug,B_aug,Q_aug_100,R)
Kp_aug_LQ_100=K_LQ_aug_100(:, 1:12)
Ki_aug_LQ_100=K_LQ_aug_100(:, 13:16)

%% LQ scelta dei pesi
U_1_max=0.8;
U_2_max=0.1;
U_3_max=1.6;
U_4_max=0.7;
PHI_max=0.01;
P_max=0.05;
THETA_max=0.01;
Q_max=0.05;
PSI_max=1;
R_max=0.5;
X_max=1;
U_max=0.8;
X_max=1;
V_max=1;
Z_max=1;
W_max=0.15;

Q_W=diag([PHI_max P_max THETA_max Q_max PSI_max R_max X_max U_max X_max V_max Z_max W_max]);
R_W=diag([U_1_max U_2_max U_3_max U_4_max]);
H=sqrt(Q_W);
rank(obsv(A,H))

Q_aug_W = [Q_W zeros(12,4); zeros(4,12) eye(4)*10];
K_LQ_aug_W=lqr(A_aug,B_aug,Q_aug_W,R)
Kp_aug_LQ_W=K_LQ_aug_W(:, 1:12)
Ki_aug_LQ_W=K_LQ_aug_W(:, 13:16)
```

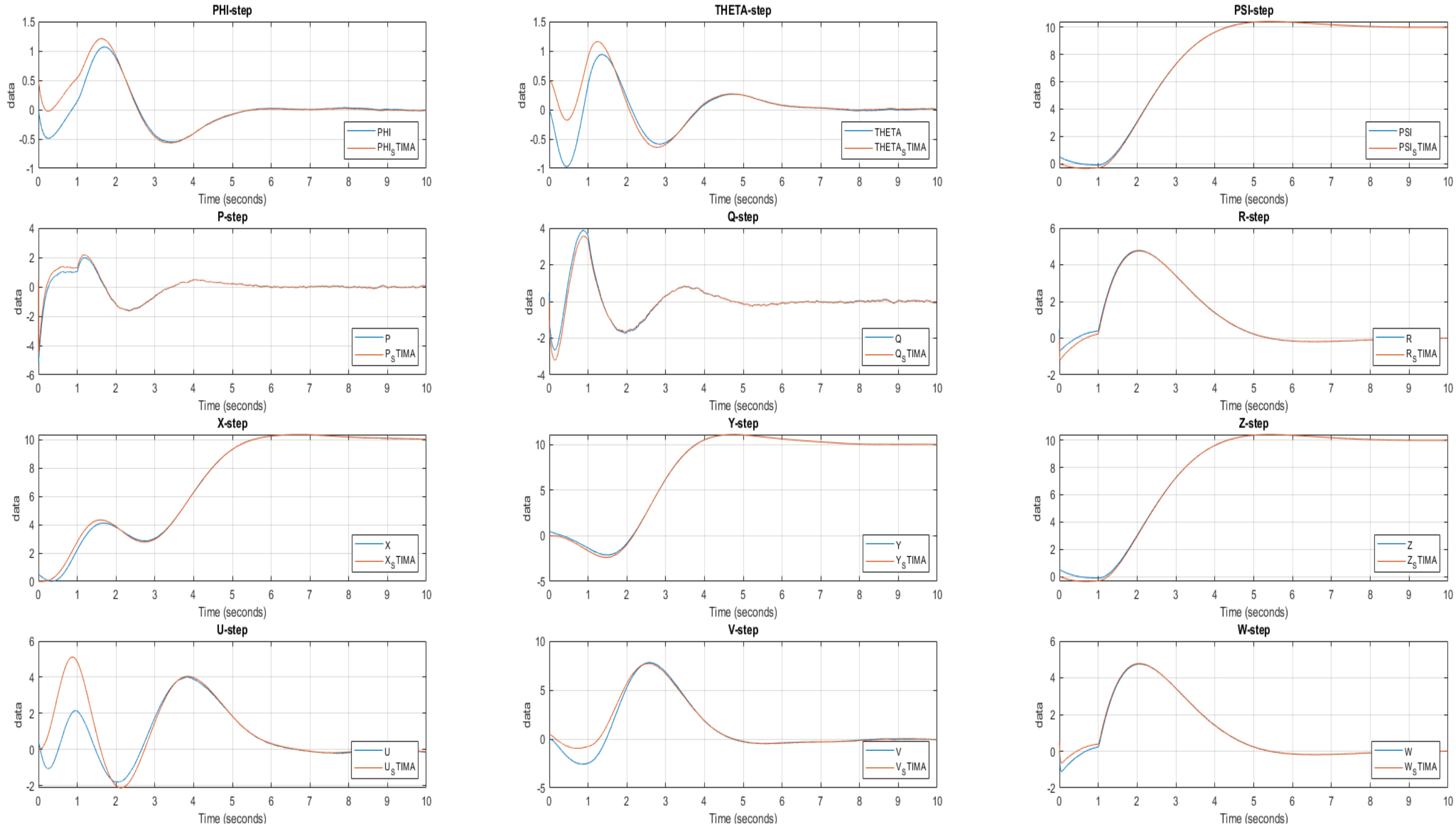
Tutte rispetteranno le proprietà per applicare il **controllo ottimo.**

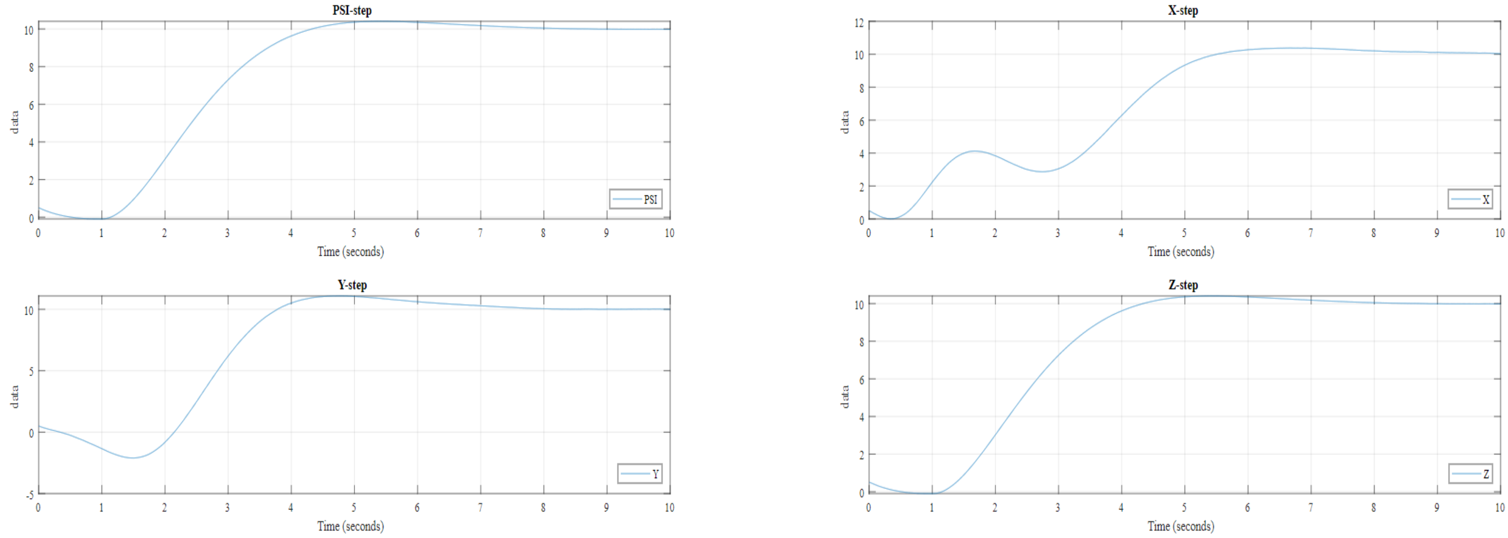


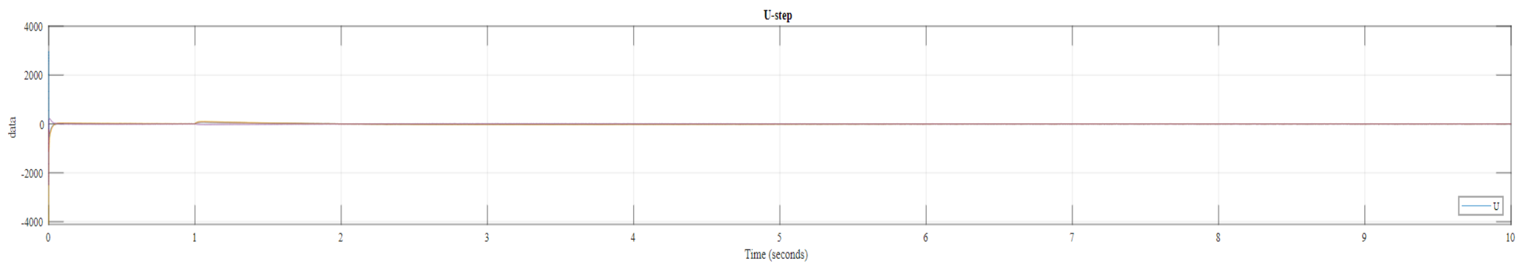
out.U1
out.U2
out.U3
out.U4
$\dot{x} = Ax + Bu$
$y = Cx + Du$
out.PHI_STIMA
out.PHI
out.P_STIMA
out.P
out.THETA
out.Q_STIMA
out.Q
out.PSI_STIMA
out.PSI
out.R_STIMA
out.R
out.X_STIMA
out.X
out.U_STIMA
out.U
out.Y_STIMA
out.Y
out.V_STIMA
out.V
out.Z_STIMA
out.Z
out.W_STIMA
out.W

Si otterranno i seguenti risultati:

- **LQ 0.5:**

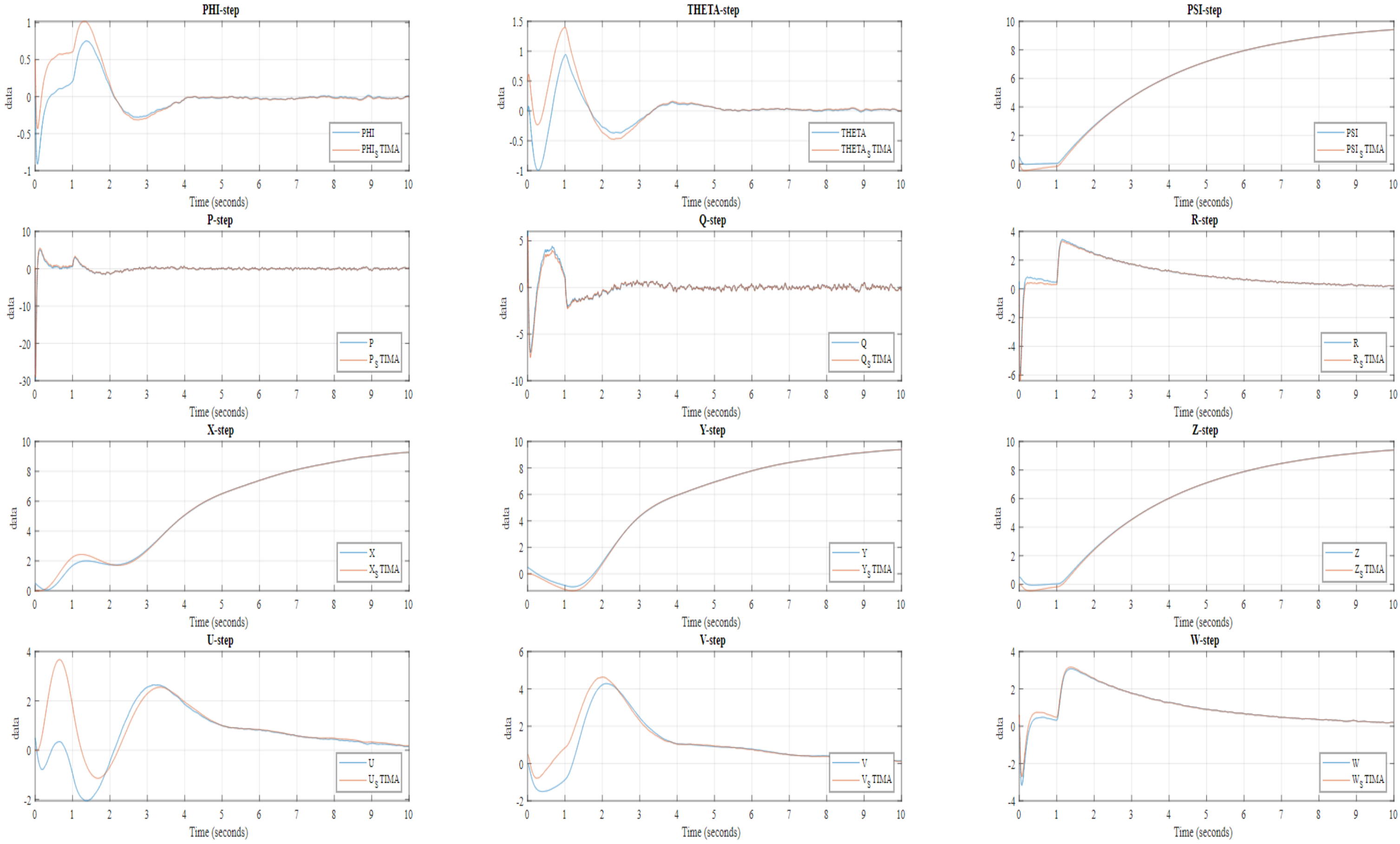


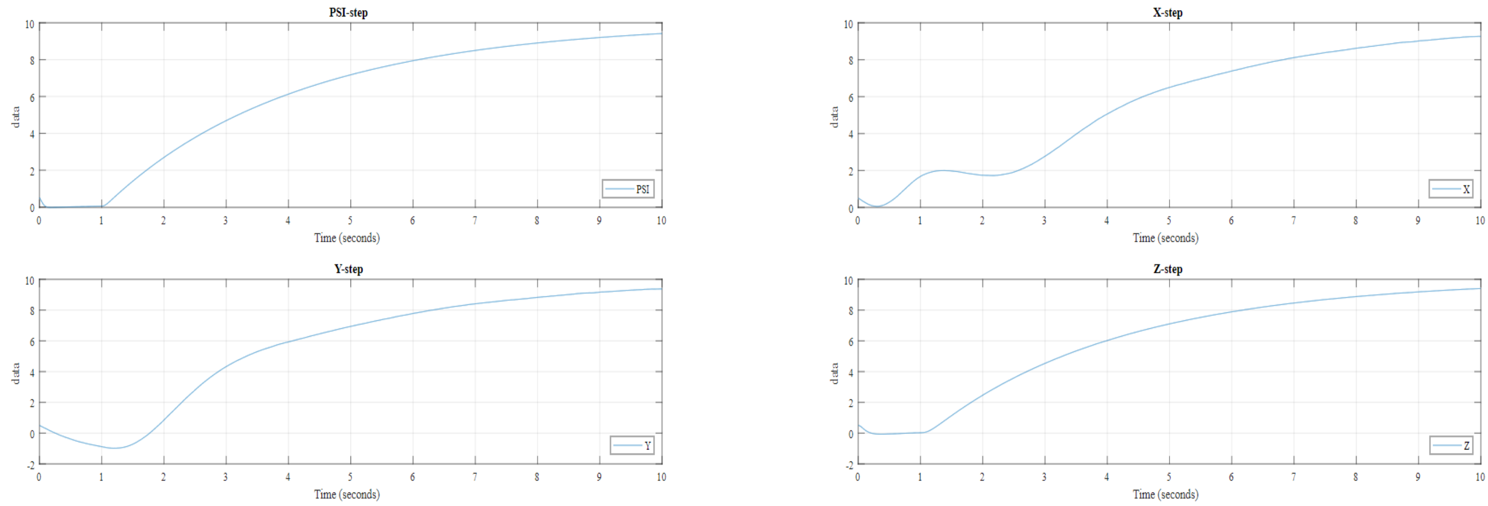


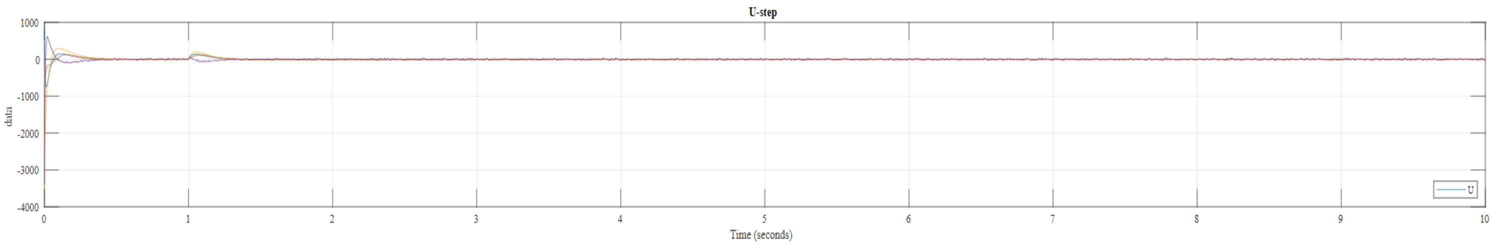


	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	2.1055	6.9778	6.9974	4.28	0.93307	10.409
X	4.097	7.7808	7.7805	3.3937	0	10.378
Y	1.2021	7.164	7.3068	10.901	21.072	11.088
Z	2.1062	6.9827	7.0007	4.2644	0.97432	10.409

- **LQ 100:**

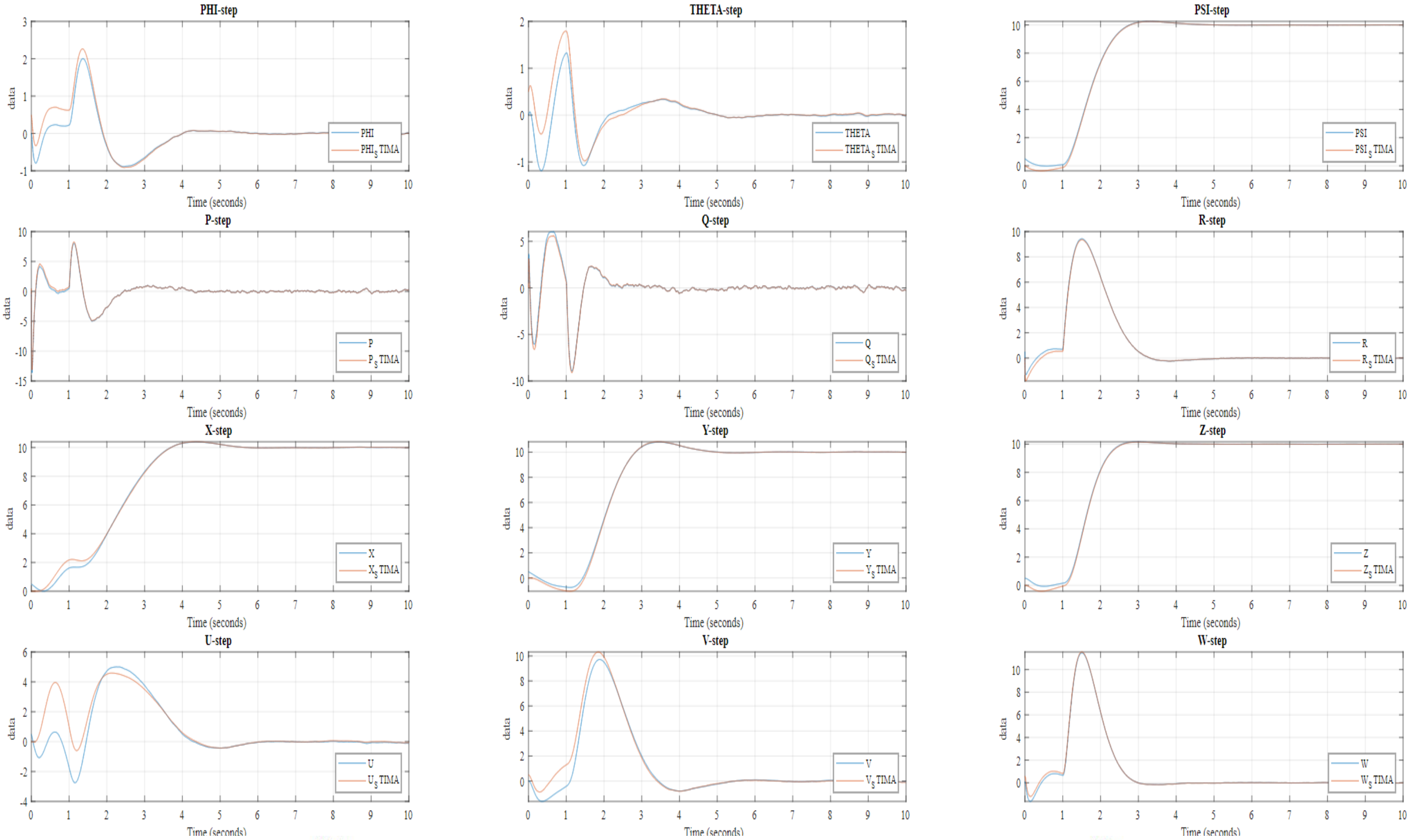


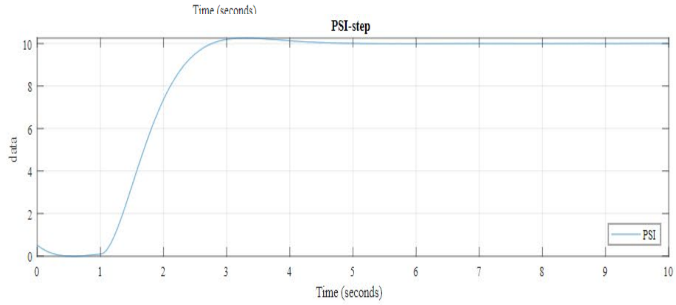


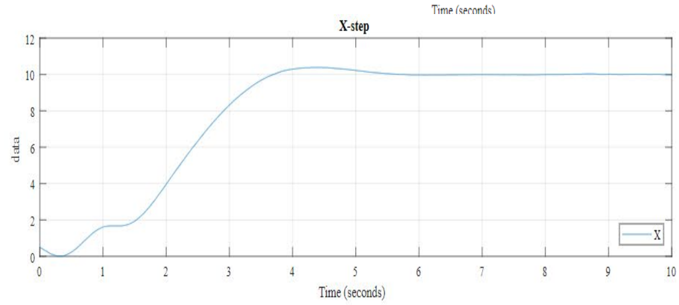


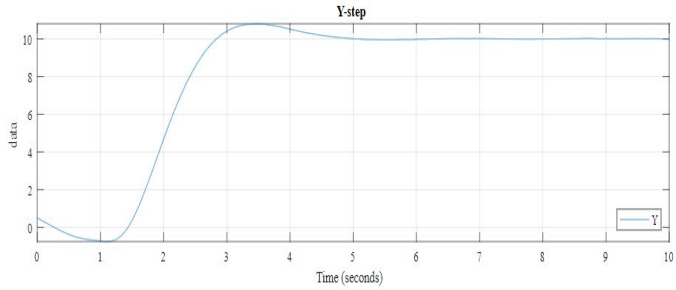
	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	1.1022	3.6922	3.6934	2.5733	0.15052	10.255
X	2.4644	5.1146	5.1145	4.1111	0	10.391
Y	0.99505	4.4828	4.5171	8.2231	7.5681	10.804
Z	0.91014	2.4296	2.4304	1.6827	0.66377	10.165

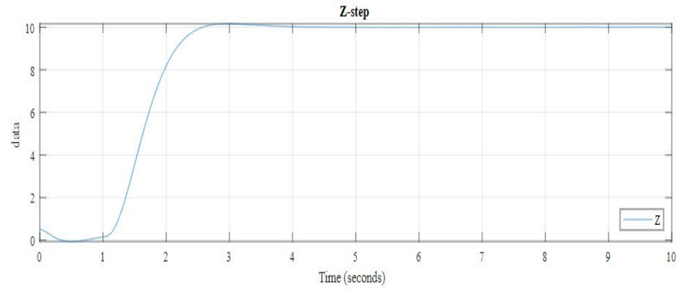
- **LQ scelta dei pesi:**

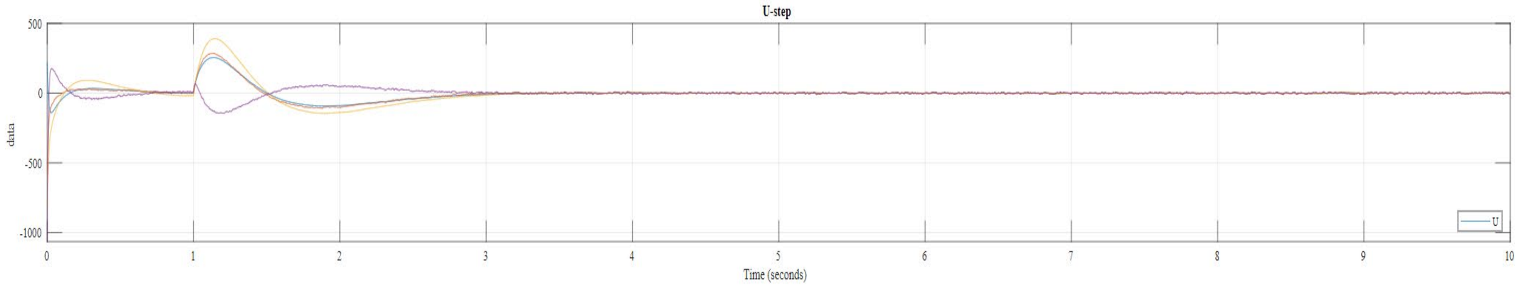












	RiseTime	TransientTime	SettlingTime	Overshoot	Undershoot	Peak
PSI	5.6134	9.1114	9.1134	0	0.22454	9.4176
X	6.6912	9.2255	9.2214	0	0	9.2672
Y	5.0459	9.0087	9.0981	0	10.5	9.3763
Z	5.5699	9.1231	9.1303	0	0.69874	9.4009