

Sistemi di Controllo Multivariabile

Controllo in retroazione di stato di un robot manipolatore PUMA

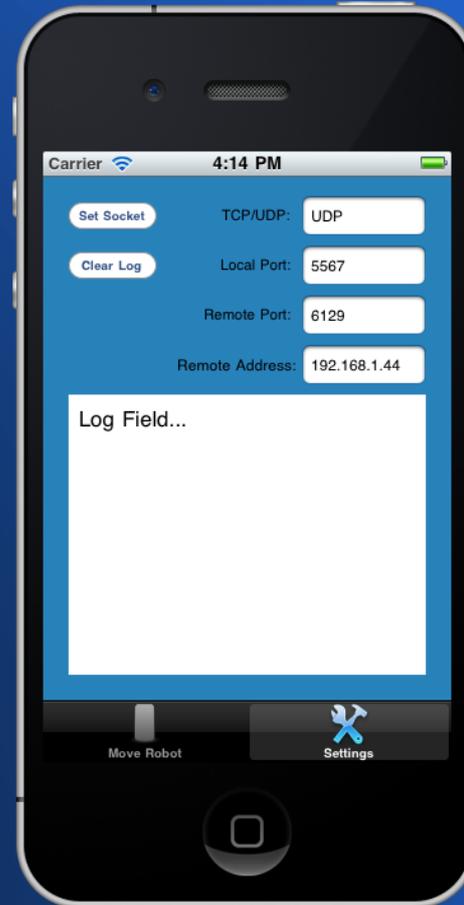
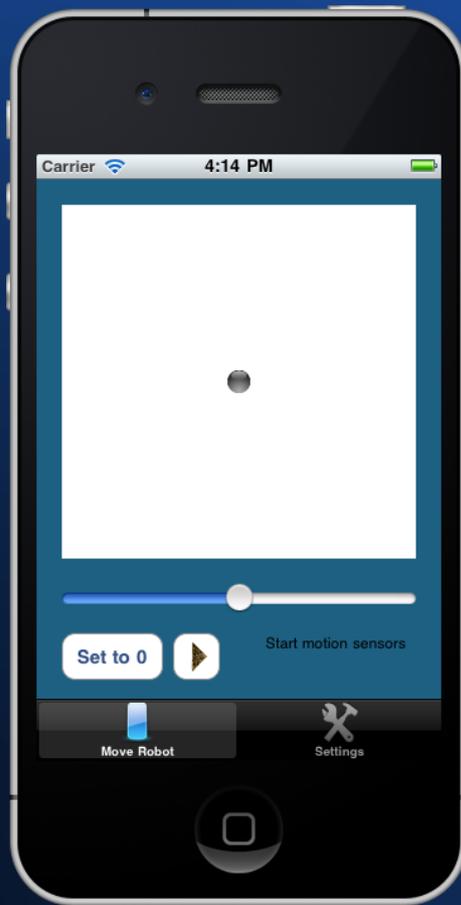
Carmine Dario Bellicoso M58/028

Andrea Gerardo Barbato M58/036

Processo implementato

- Robot PUMA
- Riferimento nello spazio operativo generato tramite iPhone
- CLIK : Closed Loop Inverse Kinematics
- DAMA^{ROB}: Dario and Marco's Robotics Toolbox
- Controllo in retroazione di stato
 - Allocazione poli
 - LQR
- Invio di segnali ad Arduino e realizzazione fisica del robot

Generazione del riferimento tramite iPhone

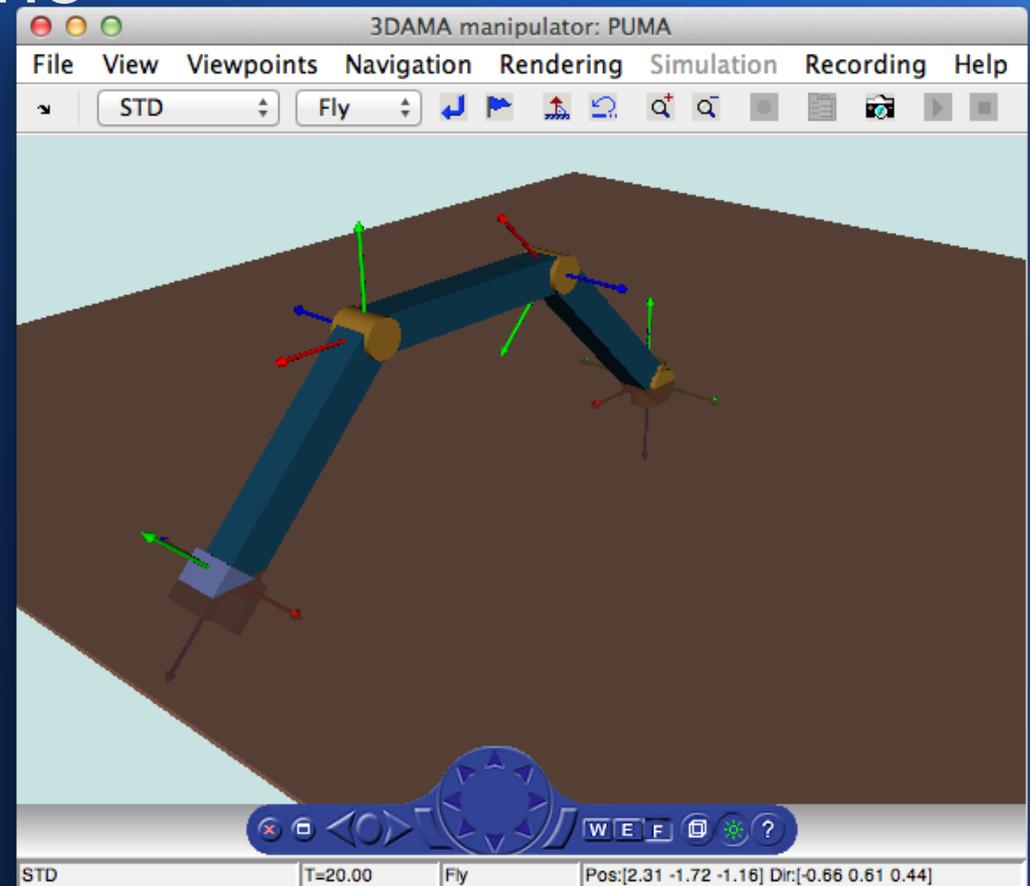


DAMA^{ROB}

- Robotics toolbox scritto in Matlab/Simulink
 - Programmazione Object-Oriented
 - Modello cinematico e dinamico di un manipolatore in forma simbolica
 - Modello VR per animazioni 3D

Modello Cinematico PUMA

- Posizionamento terne



Modello Cinematico PUMA

- Matrice di rotazione tra terna base e terna end effector

$$\begin{pmatrix} -\sin(\text{th1}) & -\sin(\text{th2} - \text{th3} + \text{th4}) \cos(\text{th1}) & \cos(\text{th2} - \text{th3} + \text{th4}) \cos(\text{th1}) \\ -\cos(\text{th1}) & \sin(\text{th2} - \text{th3} + \text{th4}) \sin(\text{th1}) & -\cos(\text{th2} - \text{th3} + \text{th4}) \sin(\text{th1}) \\ 0 & -\cos(\text{th2} - \text{th3} + \text{th4}) & -\sin(\text{th2} - \text{th3} + \text{th4}) \end{pmatrix}$$

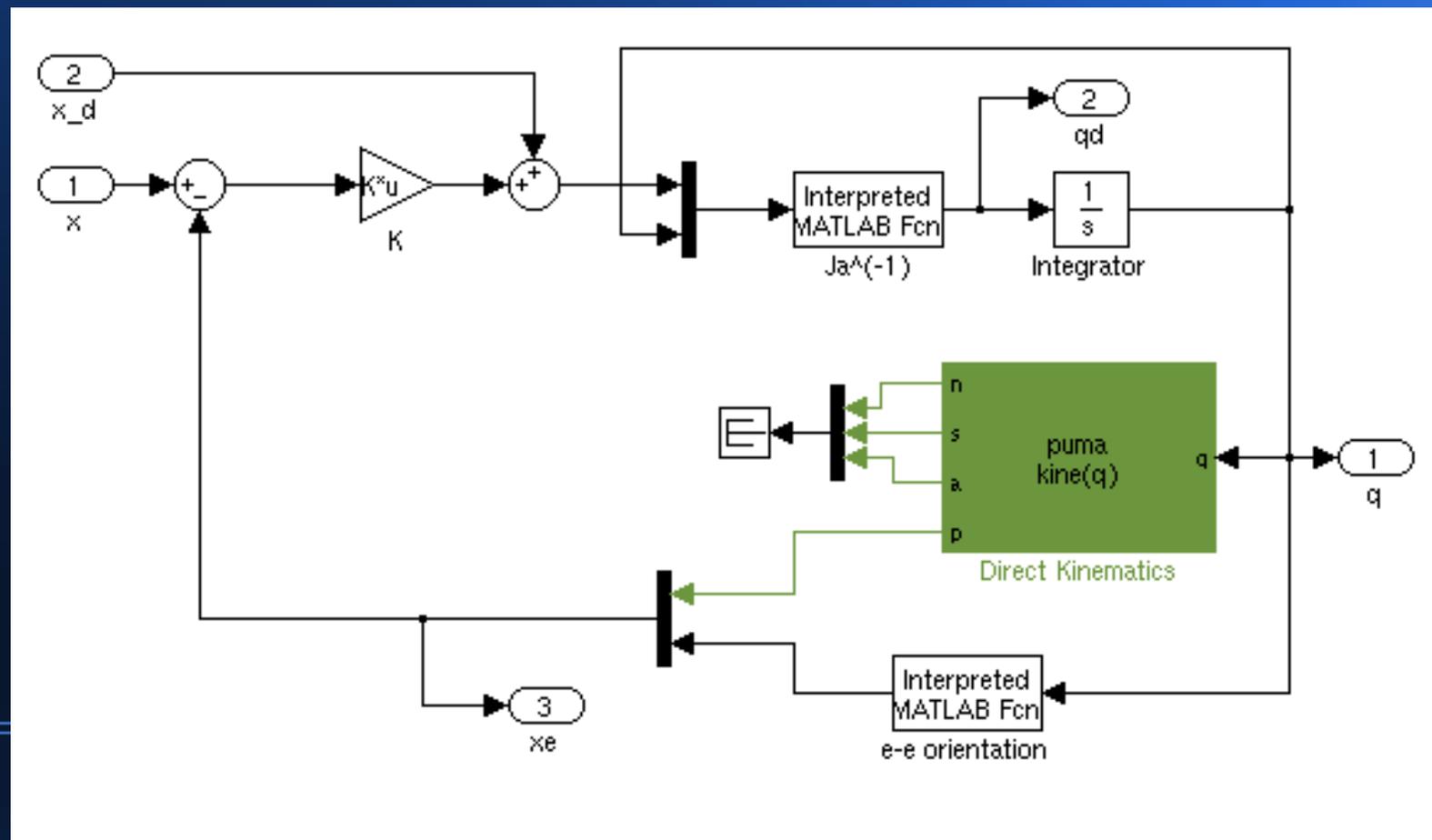
Modello Cinematico PUMA

- Posizione dell'end effector rispetto alla terna base

$$\begin{pmatrix} \frac{\cos(\theta_1) (\cos(\theta_2 - \theta_3) + \cos(\theta_2 - \theta_3 + \theta_4) + \cos(\theta_2))}{2} \\ - \frac{\sin(\theta_1) (\cos(\theta_2 - \theta_3) + \cos(\theta_2 - \theta_3 + \theta_4) + \cos(\theta_2))}{2} \\ - \frac{\sin(\theta_2 - \theta_3)}{2} \quad - \frac{\sin(\theta_2 - \theta_3 + \theta_4)}{2} \quad - \frac{\sin(\theta_2)}{2} \end{pmatrix}$$

CLIK

- Inversione cinematica del primo ordine tramite inversa dello Jacobiano analitico



Modello Dinamico PUMA

- Equazioni del modello dinamico ricavate dal metodo di Lagrange:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}_v\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$
$$\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T = [\vartheta_1 \ \vartheta_2 \ \vartheta_3 \ \vartheta_4]^T$$

Feedback Linearization: Inverse Dynamics

- Si sceglie un ingresso tale da compensare le non linearità del modello:

$$u = B(q)y + n(q, \dot{q})$$

- Sostituendo nel modello, si ha:

$$\ddot{q} = y$$

- Il modello ottenuto è lineare e disaccoppiato. Il manipolatore, sotto l'azione della dinamica inversa, risulta essere un doppio integratore sotto l'azione dell'ingresso ausiliario y

Modello lineare nell'errore della traiettoria

- Partendo dal modello linearizzato, si vuole trovare una equazione differenziale che rappresenti la dinamica dell'errore tra le traiettorie desiderate e quelle effettive nello stato
- Su questo modello verrà poi implementato il controllo, il cui compito sarà quello di portare asintoticamente a zero lo stato

Modello lineare nell'errore della traiettoria

- Scegliendo opportunamente l'ingresso y e sostituendo si ha:

$$y = -\dot{q} - q + r$$

$$\ddot{q} + \dot{q} + q = r$$

Modello lineare nell'errore della traiettoria

- A questo punto, è sufficiente scegliere r in modo tale da ottenere il modello il cui stato rappresenta l'errore di traiettoria:

$$\mathbf{r} = \ddot{\mathbf{q}}_d + \dot{\mathbf{q}}_d + \mathbf{q}_d - \mathbf{w}$$

$$\ddot{\tilde{\mathbf{q}}} + \dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{q}} = \mathbf{w}$$

Modello lineare nell'errore della traiettoria

- Scegliendo come stato il vettore:

$$\mathbf{x} = [q \ \dot{q}]^T$$

il modello i-s-u risulta:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{I}_8$$

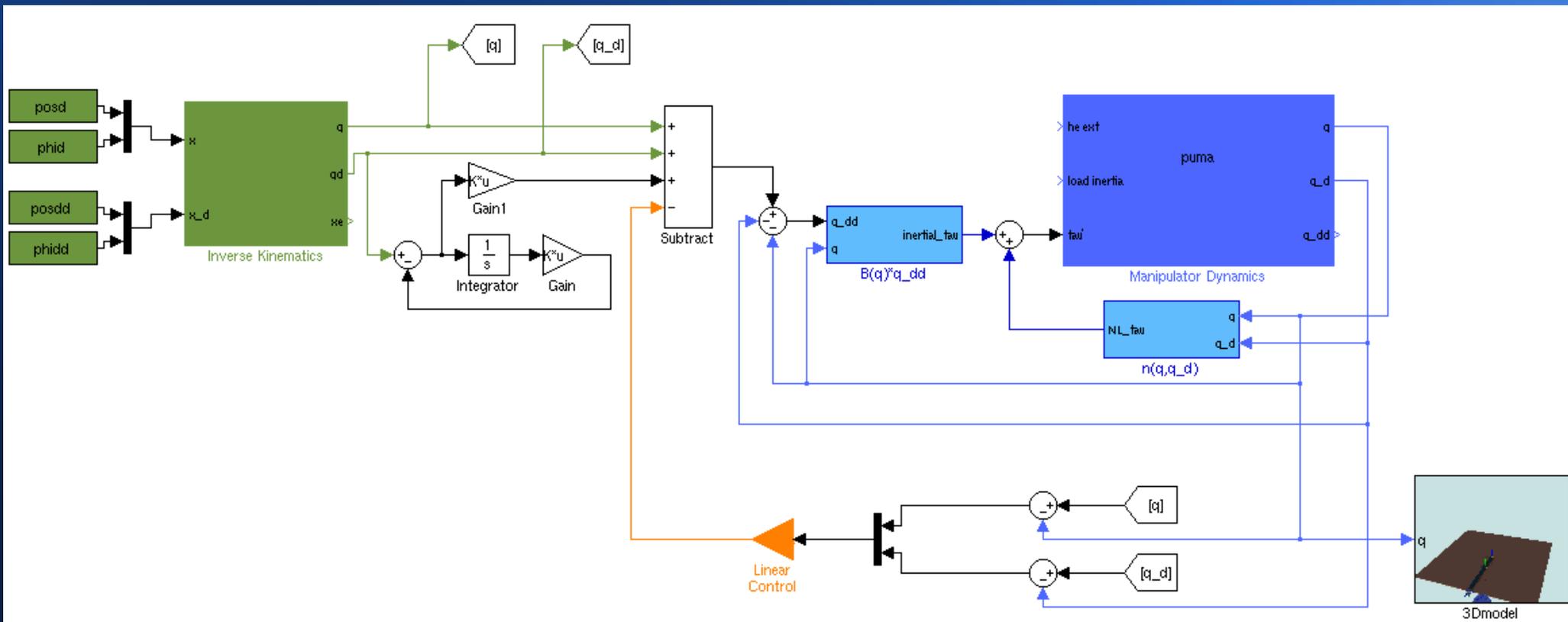
Controllabilità

- Prima di implementare l'allocazione dei poli, è necessario verificare la controllabilità del sistema ottenuto. E' bastato valutare in Matlab il rango della matrice di controllabilità:

$$C = [B \ AB \ A^2B \ A^3B \ A^4B \ A^5B \ A^6B \ A^7B]$$

- Essendo il rango di C pari a 8, il sistema risulta completamente controllabile. Tutte le dinamiche instabili potranno essere quindi stabilizzate dal sistema di controllo in retroazione di stato. Ciò discende dal fatto che la controllabilità implica la stabilizzabilità

Schema generale di controllo



Allocazione dei poli

- Utilizzando il comando place di Matlab, si è scelto un controllore k tale da piazzare gli autovalori del sistema a ciclo chiuso in:

$$\lambda = [-10 \quad -10 \quad -10 \quad -15 \quad -15 \quad -15 \quad -15 \quad -15]^T$$

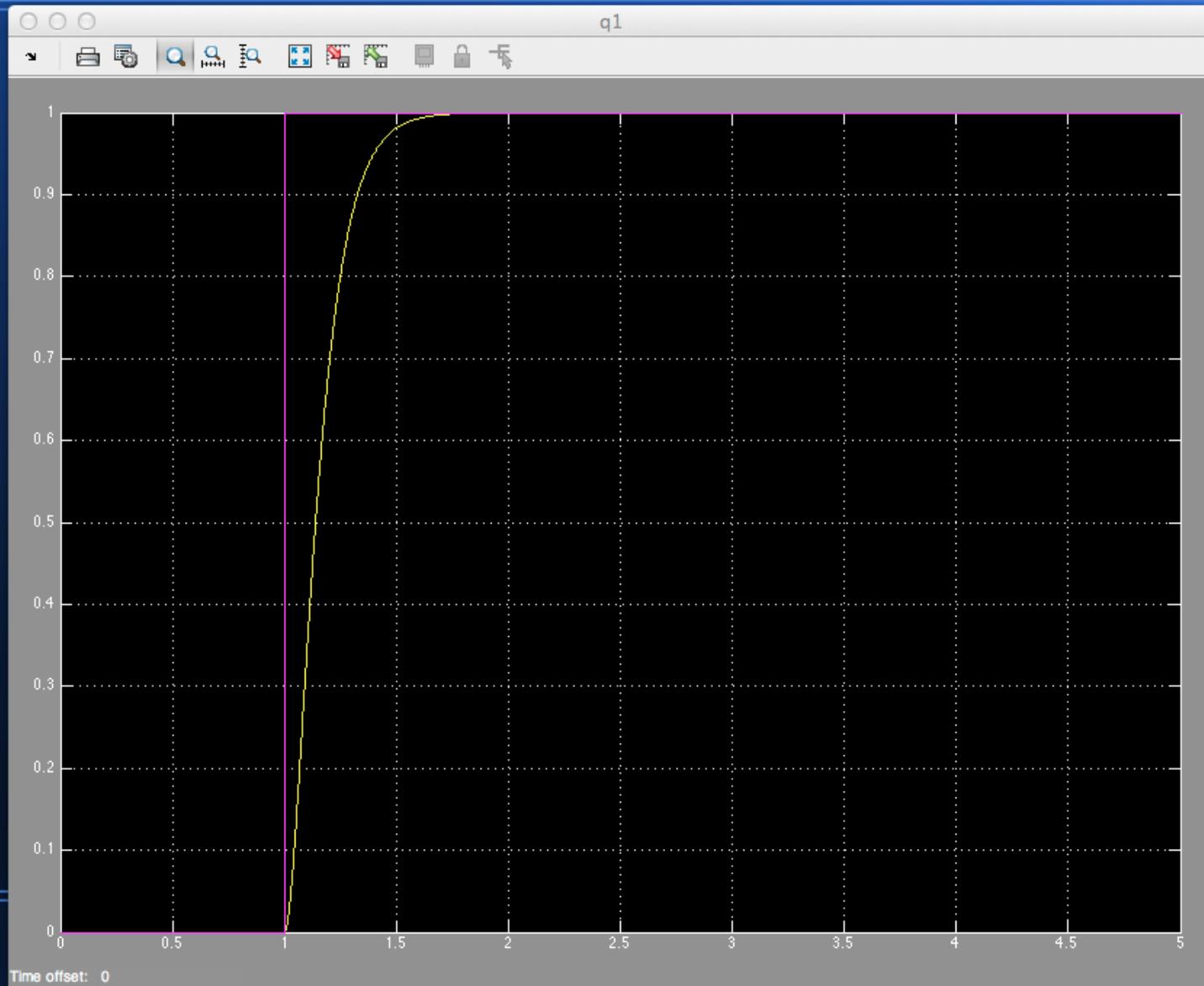
- Il controllore risulta essere:

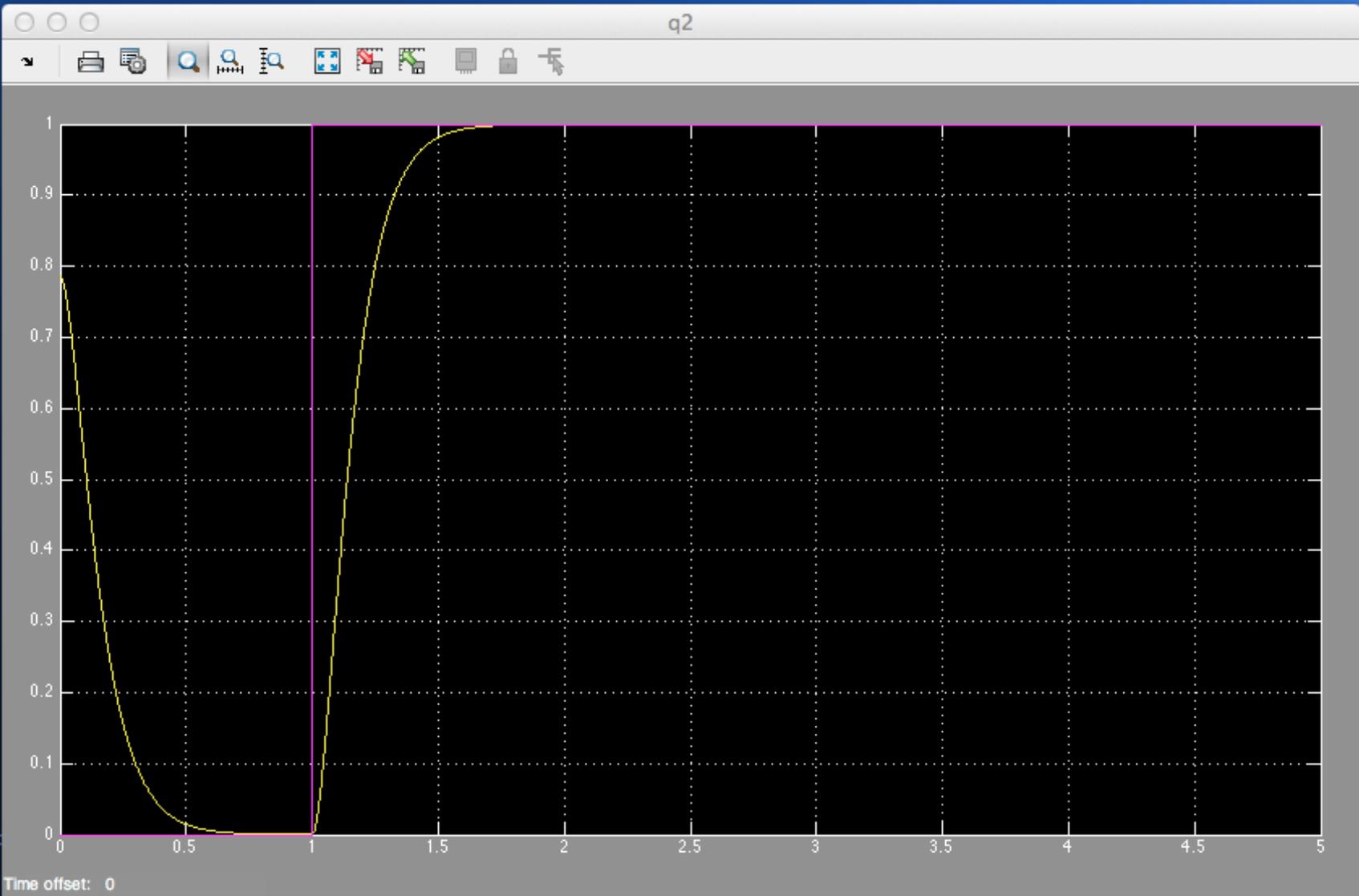
$$k = \begin{bmatrix} -149 & 0 & 0 & 0 & -24 & 0 & 0 & 0 \\ 0 & -149 & 0 & 0 & 0 & -24 & 0 & 0 \\ 0 & 0 & -149 & 0 & 0 & 0 & -24 & 0 \\ 0 & 0 & 0 & -149 & 0 & 0 & 0 & -24 \end{bmatrix}$$

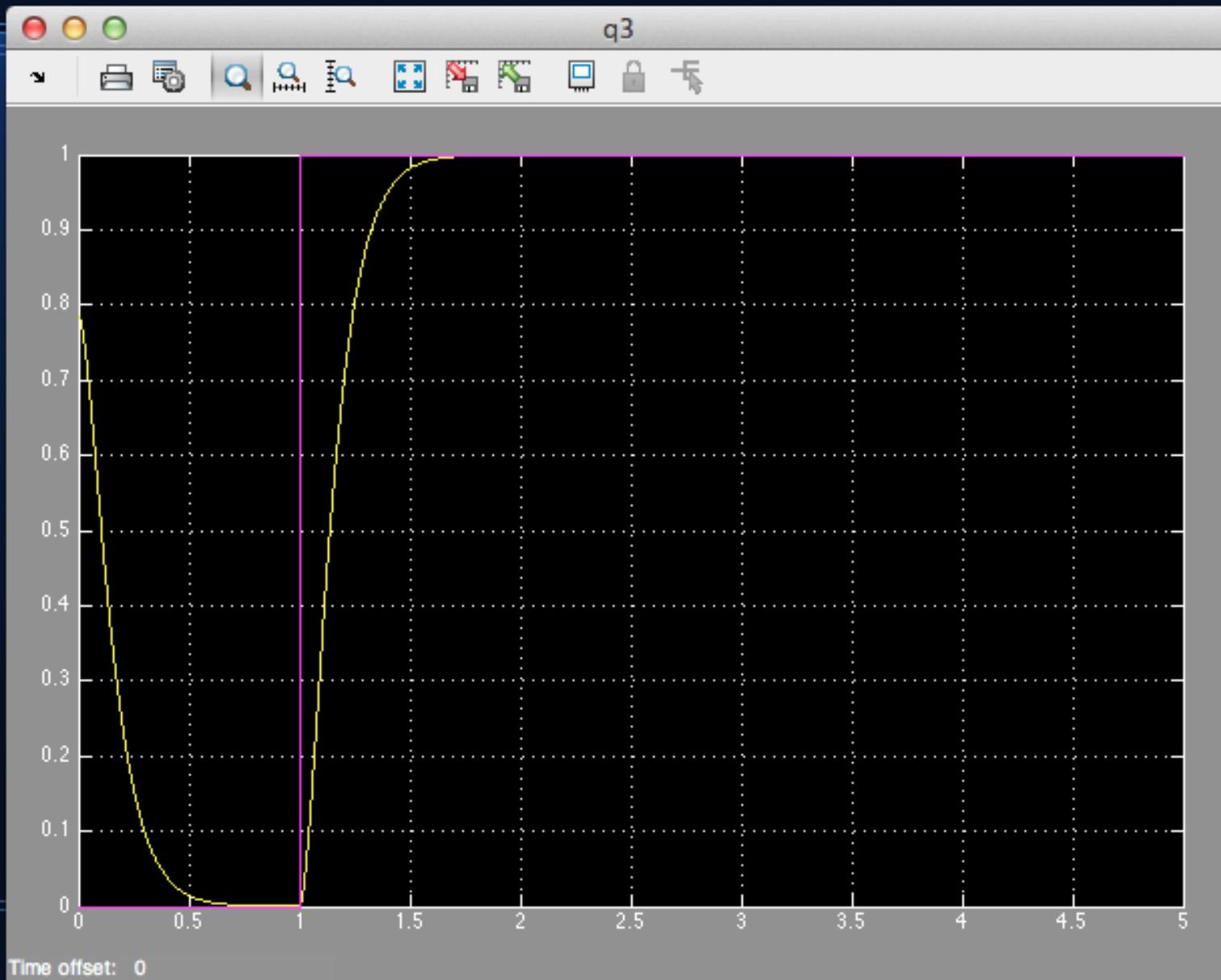
Allocazione dei poli

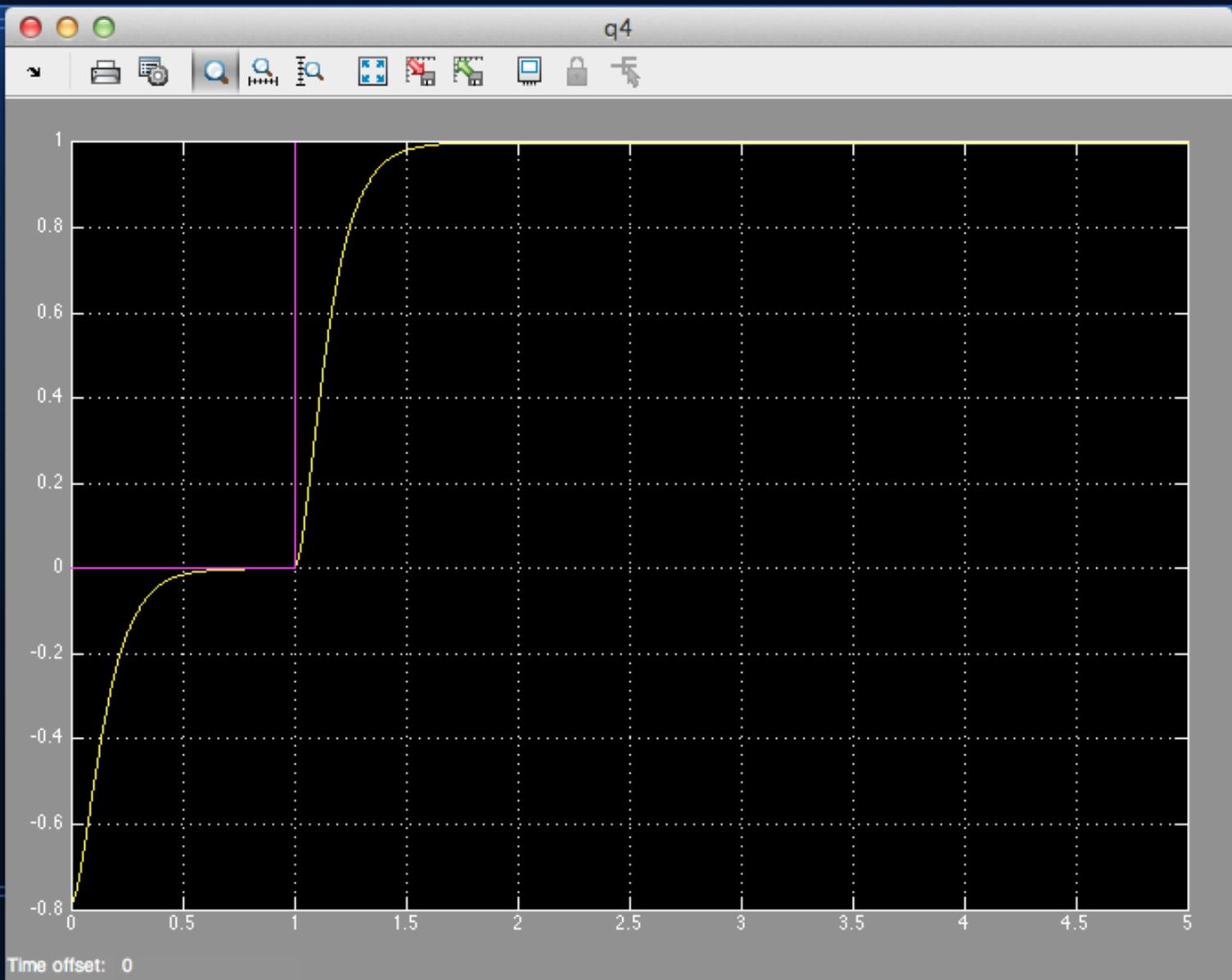
- Si vuole ora mostrare l'andamento del vettore delle variabili di giunto q in risposta ad un ingresso a gradino unitario. Come si vedrà dai grafici, il sistema risulta effettivamente stabilizzato, per cui partendo da una condizione iniziale non nulla i giunti tenderanno ad assumere la posizione di riposo ($q = 0$), per poi seguire l'ingresso a gradino

Allocazione dei poli – risposta al gradino









Controllo ottimo LQR

- Il controllo ottimo LQR permette non solo di stabilizzare il sistema in retroazione ma anche di minimizzare un indice di costo, minimizzando ad esempio l'energia dell'azione di controllo. La soluzione del problema LQR su orizzonte infinito è stata affrontata tramite matrice Hamiltoniana. Successivamente per verificare i risultati è stato utilizzato il comando Matlab `lqr`

Selezione delle matrici di peso

- Per affrontare il problema LQR sono state selezionate le seguenti matrici:

$$Q = 1000 * \text{eye}(8)$$

$$R = 100 * \text{eye}(4)$$

- Condizione sufficiente affinché l'azione di controllo riesca a minimizzare l'indice è che R sia definita positiva.

Matrice Hamiltoniana

- Affinchè l'equazione algebrica di Riccati abbia soluzione P^* , l'Hamiltoniana non deve possedere autovalori sull'asse immaginario. Tramite Matlab gli autovalori di H risultano essere:

$$\lambda(H) = \begin{bmatrix} -2.746 \\ 2.746 \\ 1.2077 \\ -1.2077 \\ -2.746 \\ 2.746 \\ 2.746 \\ 2.746 \\ -2.746 \\ -2.746 \\ -1.2077 \\ -1.2077 \\ -1.2077 \\ 1.2077 \\ 1.2077 \\ 1.2077 \end{bmatrix}$$

- Gli autovalori di H , come previsto, risultano essere a simmetria quadrante. Nessuno di essi inoltre ha parte reale nulla. È possibile quindi, tramite un cambiamento di base, esprimere H in forma diagonale, con le forme di Jordan relative agli autovalori stabili ed instabili di H sulla diagonale

- La trasformazione di base U risulta simbolicamente:

$$U = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix}$$

dove la prima colonna di U è composta dagli autovettori di H relativi agli autovalori stabili di H , la seconda colonna è composta dagli autovettori di H relativi agli autovalori instabili di H . Per ricavare tali autovettori in modo più comodo che col comando `eig` di Matlab, è stata scritta una funzione Matlab che riordina autovalori ed autovettori di una matrice quadrata qualsiasi in modo da ottenere prima quelli stabili e poi quelli instabili

Soluzione dell'ARE

- A questo punto, si ottiene subito:

$$k = R^{-1}B^T P^* = R^{-1}B^T U_{21}U_{11}^{-1}$$

$$k = \begin{bmatrix} -2.3166 & 0 & 0 & 0 & -2.9539 & 0 & 0 & 0 \\ 0 & -2.3166 & 0 & 0 & 0 & -2.9539 & 0 & 0 \\ 0 & 0 & -2.3166 & 0 & 0 & 0 & -2.9539 & 0 \\ 0 & 0 & 0 & -2.3166 & 0 & 0 & 0 & -2.9539 \end{bmatrix}$$

Controllo LQR – risposta al gradino

- Come fatto per il controllo in allocazione dei poli, verranno mostrati i grafici della risposta al gradino unitario. Si osserva da subito che l'evoluzione dello stato del sistema sotto l'azione del controllo ottimo u^* è più lenta di quella nel caso di controllo in allocazione dei poli. Questo perchè gli autovalori stabili di H , che coincidono con gli autovalori della matrice dinamica $A-Bk$ sono più in bassa frequenza rispetto al caso precedente.

