

Robòtica industrial i de serveis.

Projecte final

Implementació d'estructures de control utilitzant la Robotics Toolbox.

Guillem Casado, Albert Llebaria

Gener 2019

1 Objectius

L'objectiu d'aquest projecte és implementar dues estructures de control per al moviment d'un robot. Aquestes són:

- Controlador PD amb compensació de gravetat en l'espai d'articulacions.
- Control de dinàmica inversa en l'espai d'articulacions.

Per tal d'assolir-ho, s'ha de dur a terme els següents punts:

- Per cada estratègia de control, implementar el sistema mitjançant SIMULINK.
- Utilitzar la llibreria Robotic Toolbox per a la realització del projecte.
- Utilitzar el robot PUMA560, ja que es disposen les seves característiques (inèrcia, coriolis...) per facilitar la implementació.
- Proporcionar un script per configurar paràmetres com ara les constants dels controladors, configuracions inicial i final de les articulacions del robot etc.
- Mostrar una animació del robot movent-se de posició inicial a final en cada execució del SIMULINK de cada estratègia de control. Això permet comprovar que la implementació és correcta.

2 Controlador PD amb compensació de gravetat en l'espai d'articulacions

2.1 Estudi del model teòric a implementar

Per aquesta estructura de control, es desitja assolir una determinada configuració d'articulacions, q_d . D'aquesta manera, l'error serà $\tilde{q} = q_d - q$. El model del robot a controlar és $B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F\dot{q} + g(q) = u$. Donada la no linealitat d'aquest, es fa ús del mètode de Lyapunov per tal de garantir que la llei de control porti al robot fins a la configuració desitjada d'una manera asimptòticament estable. La funció de

Lyapunov $V(q, \dot{q})$ per aquest model és:

$$V(q, \dot{q}) = \frac{1}{2} \dot{q}^T B(q, \dot{q}) + \frac{1}{2} \tilde{q}^T K_p \tilde{q} > 0, \forall \dot{q}, q \neq 0$$

Si $\dot{V}(q, \dot{q}) < 0$, llavors es pot garantir l'estabilitat. La derivada de la funció de Lyapunov resulta de la següent manera:

$$\dot{V}(q, \dot{q}) = -\dot{q}^T F(\dot{q}) + \dot{q}^T [u - g(q) - K_p \tilde{q}]$$

Donat que el primer terme de la derivada és sempre negatiu, una possibilitat per assegurar l'estabilitat és que el segon terme sigui zero. D'aquesta manera, es decideix que la llei de control sigui:

$$u = g(q) + K_p \tilde{q}$$

Finalment, per accelerar la convergència, s'afegeix un terme derivatiu. La llei de control final, queda doncs, expressada de la forma següent:

$$u = g(q) + K_p \tilde{q} - K_D \dot{q}$$

Un cop definides les equacions del sistema de control, el sistema expressat en blocs resulta com el mostrat en la figura 1.

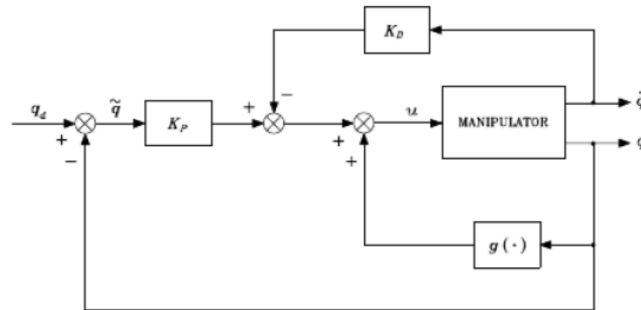


Figura 1: Esquema de blocs del sistema.

2.2 Implementació del controlador amb Matlab

Un cop definit el senyal de control i el diagrama de blocs, es procedeix a implementar-ho en el SIMULINK de Matlab. El resultat del diagrama de blocs es mostra en la figura 2.

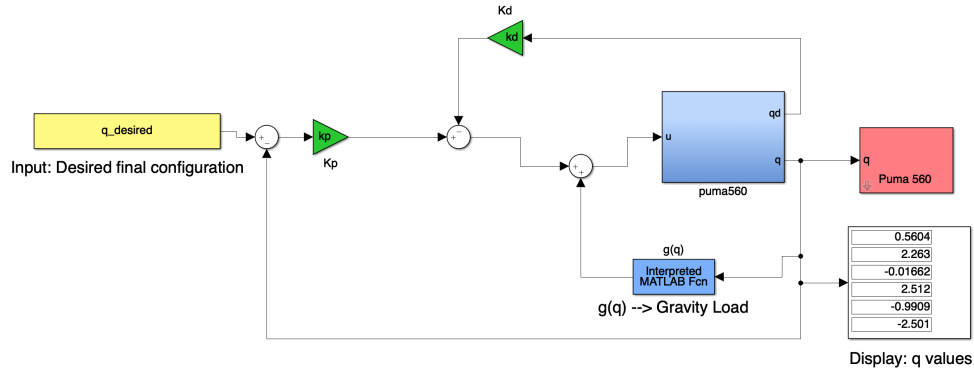


Figura 2: Esquema de blocs del sistema en Simulink.

Els valors de q_d , K_p i K_d es defineixen en el script Matlab d'inicialització. El bloc $g(q)$ correspon a la compensació de gravetat donada una configuració d'articulacions. Donat que el PUMA560 ja disposa de totes les seves característiques, es pot obtenir el valor del bloc simplement amb la comanda $p560.gravload(q)$. El bloc del manipulador, rep un senyal de control que correspon al parell que han de realitzar les articulacions. La sortida d'aquest, correspon al valor de q i \dot{q} que s'utilitzen per al següent cicle de control. Per aconseguir-los, el que realitza el bloc manipulador és obtenir l'acceleració \ddot{q} de cada articulació a partir del parell donat i, posteriorment s'integra una vegada per obtenir \dot{q} i dues per trobar q . Cal comentar que per optimitzar l'acceleració de la convergència que proporciona el terme derivatiu, s'ha cregut convenient incorporar un saturador que evités que aquest prengués valors de magnitud massa elevada. Aquest fet podria comportar que el sistema s'inestabilitzés. Així doncs, aquest saturador ha permès augmentar el valor de la K_d sense córrer el risc comentat. Finalment, un últim bloc es fa servir per visualitzar els resultats. Per un moviment de $q_{inicial} = [0, 0, 0, 0, 0, 0]$ a $q_d = [\pi/4, 3\pi/4, 0, 3\pi/4, -\pi/4, -3\pi/4]$, els resultats obtinguts són els que es mostren en la figures 3 i 4, que corresponen al robot amb la configuració inicial i final, respectivament.

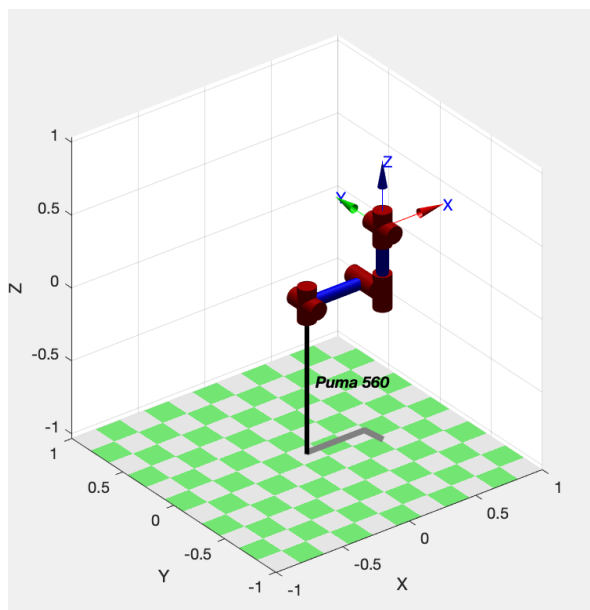


Figura 3: Posició inicial del PUMA560.

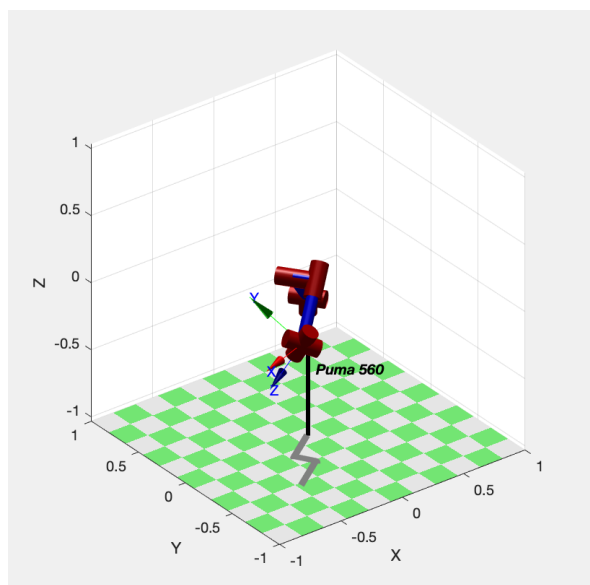


Figura 4: Posició final del PUMA560, un cop l'error ha arribat a zero.

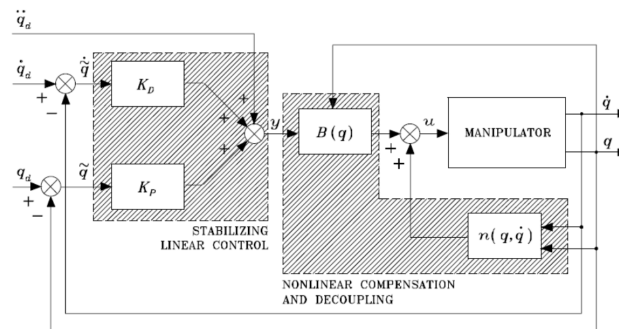
3 Control de dinàmica inversa en l'espai d'articulacions

3.1 Estudi del model teòric a implementar

Per aquesta estructura de control, s'ha de seguir una trajectòria desitjada, q_d , \dot{q}_d i \ddot{q}_d . La dinàmica del manipulador ve descrita per la següent equació:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F\dot{q} + g(q) = u$$

The diagram illustrates a closed-loop control system for a manipulator. The input y is processed by a block $B(q)$. The output of $B(q)$ is summed with a feedback signal from a block $n(q, \dot{q})$ to produce the control signal u . This signal u is then fed into the MANIPULATOR block. The MANIPULATOR block has two outputs: q and \dot{q} . The output q is fed back to the block $n(q, \dot{q})$. The output \dot{q} is integrated to produce q , which is then integrated again to produce \dot{q} , which is the final output of the system.



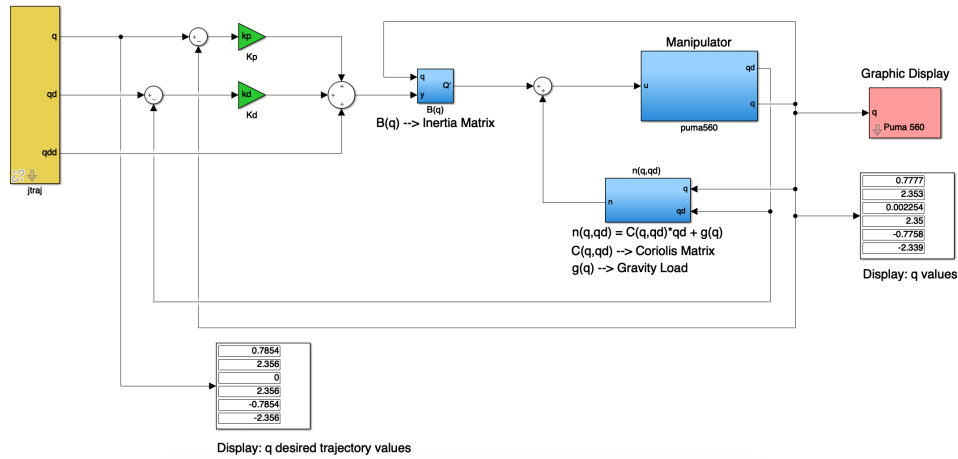


Figura 7: Esquema de blocs del sistema en Simulink.

En aquesta segona estructura de control també s'aprofiten les característiques i funcionalitats que proporciona el model del PUMA560. Concretament dins el bloc $n(q, \dot{q})$ es criden funcions com $p560.gravload(q)$ que, com s'ha comentat, proporciona la càrrega que provoca la gravetat per a una configuració determinada i $p560.coriolis(q, \dot{q})$ que retorna la matriu de forces centrífugues i de Coriolis del sistema. En relació al bloc $B(q)$, s'ha emprat la funció $p560.inertia(q)$ que proporciona la matriu d'inèrcies del sistema. El bloc que representa el manipulador està definit de la mateixa manera que en el cas anterior.

Finalment, per tal de aportar el senyal d'entrada s'ha optat per un bloc generador de trajectòria. Aquest bloc rep com a paràmetres la posició inicial i final així com el temps de duració màxim de la trajectòria i l'interval de temps al que ha d'anar generant les dades.

A continuació es mostren les figures 8 i 9 que representen la configuració inicial del robot ($q_{inicial} = [0, 0, 0, 0, 0, 0]$) i a la que arriba aquest un com realitzada tota la trajectòria ($q_d = [\pi/4, 3 * \pi/4, 0, 3 * \pi/4, -\pi/4, -3 * \pi/4]$).

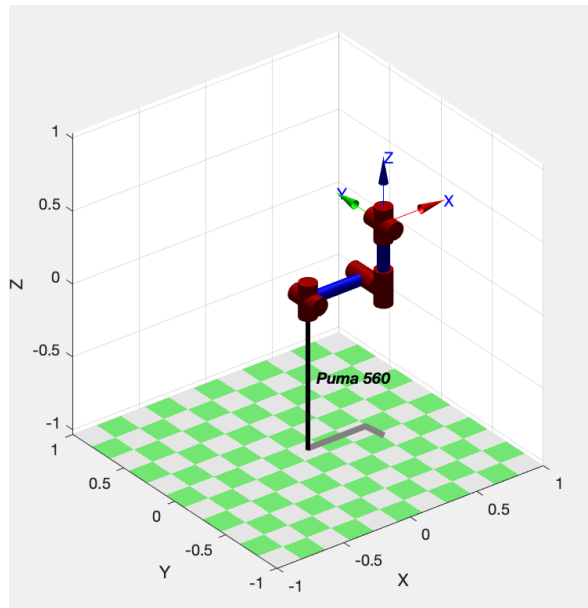


Figura 8: Posició inicial del PUMA560.

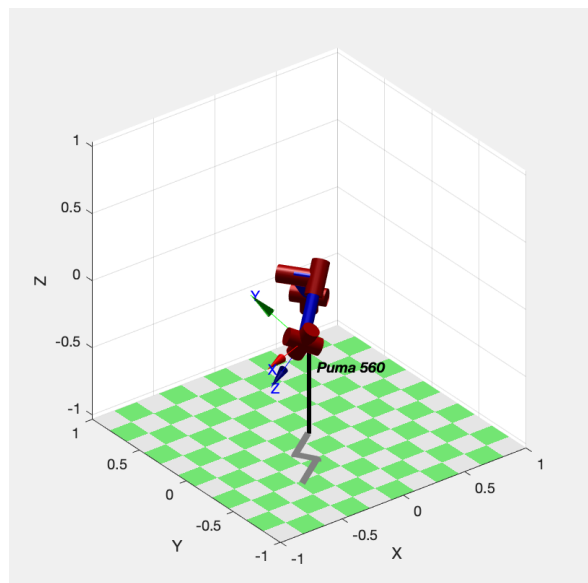


Figura 9: Posició final del PUMA560, un cop realitzada tota la trajectòria.

Com es pot comprovar, la configuració inicial i final del robot es la mateixa que en el primer control. Donat que en aquest segon cas, també s'assoleix aquesta última posició i s'ha comprovat com es desenvolupa correctament el seguiment de la trajectòria, es dona per vàlid el funcionament i resultats de la implementació d'aquesta segona estructura de control.

Referències

- [1] Alireza. Izadbakhsh. *Closed-Form Dynamic Model of PUMA 560 Robot Arm*. Garmsar. Iran. Febrer 2009.

- [2] Atta Oveisi, Miad Jarrahi, Mohammad Gudarzi and Mohammad Mahdi Mohammadi *Genetic Algorithm and Adaptive Model Reference Controller in Tracking Problem of PUMA 560 Arm Robot*. Teheran. Iran. Febrer 2013.
- [3] B.Morales and R.Carelli. *Robot control with inverse dynamics and non-linear gains*. San Juan. Argentina.
- [4] Reham H. Mohammed, Fahmy Bendary and Kamel Elserafi. *Trajectory Tracking Control for Robot Manipulator using Fractional Order-Fuzzy-PID Controller*. Egipte. Gener 2019.
- [5] Peter Corke: Robotics Toolbox for MATLAB,
<http://petercorke.com/wordpress/toolboxes/robotics-toolbox>