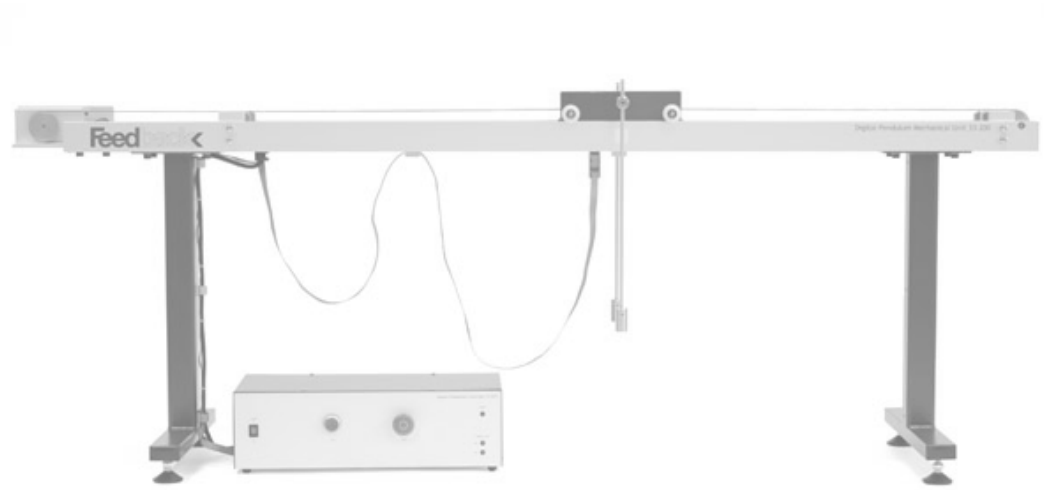
MEMÒRIA

CONTROL AMB APRENENTATGE PER REFORÇ D'UN PÈNDOL INVERTIT

TÈCNIQUES D'INTEL·LIGÈNCIA ARTIFICIAL I APLICACIONS PER A L'AUTOMATITZACIÓ



LAURA ARMENGOL

RICARD CASALS

GLÒRIA GARRIGA

ALBA RIBÓ

Q2 2017-2018

ÍNDEX

ÍNDEX DE FIGURES

1 INTRODUCCIÓ

Aquest treball es centra en la recerca i desenvolupament del control d'un pèndol invertit a partir de la tècnica d'intel·ligència artificial d'aprenentatge per reforç.

El projecte inclou la recerca realitzada sobre l'aprenentatge per reforç, on es compilen els coneixements necessaris adquirits durant l'elaboració del treball pel que fa a aquesta tècnica. En el projecte també es descriu el desenvolupament del controlador per mantenir el pèndol en posició invertida.

Finalment el treball compta amb 2 apartats de cloenda:l'avaluació dels resultats, on es justifica i es comprova els resultat obtinguts durant el desenvolupament; i les conclusions, on es defineix el compliment dels objectius plantejats a l'inici del treball.

A continuació, es planteja el problema i la planta amb que es treballa. També es detallen els objectius i la planificació del treball.

1.1 Control d'un pèndol invertit amb aprenentatge per reforç

Aquest treball recull l'estudi del control d'un pèndol invertit amb aprenentatge per reforç. El problema a resoldre consisteix en definir el controlador basat en tècniques d'aprenentatge per reforç (Q-learning) per aconseguir mantenir la maqueta del Feedback 33-005 en posició invertida.

Per tal de facilitar la tasca d'aprenentatge, és necessari la utilització d'un model de la maqueta. Un cop realitzat l'aprenentatge es comprovarà el funcionament del controlador amb la maqueta real. La maqueta és un sistema carro-pèndol que es detalla a continuació.

1.1.1 Sistema carro-pèndol

El sistema carro-pèndol consisteix en un barraarticulada (pèndol)i en un carro, de tal manera que el pèndol pot moure's només en un pla vertical. En la Figura 1 es mostra l'esquema del sistema.

Un dels problemes de control més comuns en el que s'utilitza el sistema carro-pèndol, és el pèndol invertit, que consisteix en mantenir el pèndol en un angle de 180º respecte el seu estat de repòs. Aquesta és l'aplicació que s'utilitzarà en aquest projecte, realitzant el controlador del sistema aplicant aprenentatge per reforç.

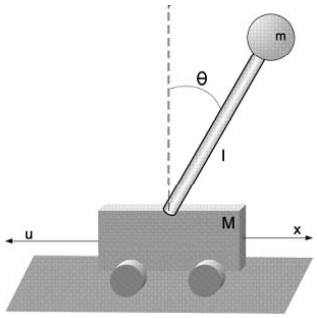


Figura 1: Esquema del sistema carro-pèndol

Les constants del sistema són les següents:

- m: massa del pèndol.
- M: massa del carro.
- L: longitud del pèndol.

Les variables del sistema són les següents:

- x: posició del carro.
- θ: angle del pèndol.

Les equacions, proporcionades pel fabricant, que defineixen el sistema carro-pèndol són les següents:

$$(m+M)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F$$
$$(I+ml^2)\ddot{\theta} - mgl\sin\theta + ml\ddot{x}\cos\theta + d\dot{\theta} = 0$$

Figura 2: Equacions del sistema carro-pèndol

1.1.2 Maqueta Feedback 33-005

La maqueta que s'utilitzarà per realitzar les proves del controlador és el Feedback 33-005. El qual està format per una estructura fixa que conté el carril i el carro mòbil amb un pèndol doble. Tal com es mostra en la Figura 3.

El carroes desplaça per un carril de longitud limitadaaccionat per un motor DC. En desplaçar-se el carro pel carril, el pèndol articulat en ell es balanceja, ja que al'estar articulat pot girar de forma lliure (rotant en l'eix perpendicular al moviment del carro).

Els paràmetres que defineixen el sistema carro-pèndol de la maqueta venen donats pel manual del fabricant, i són els que es mostren a la Figura 4.

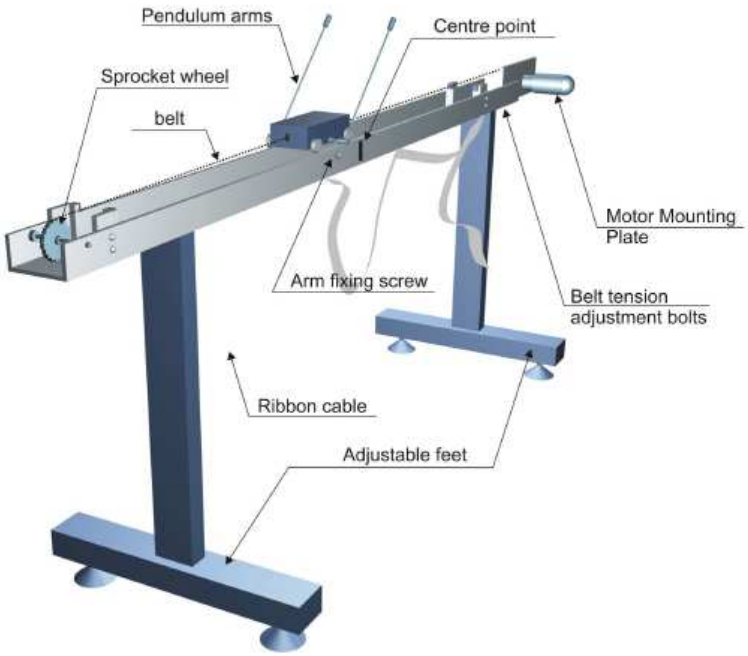


Figura 3: Esquema de la maqueta Feedback 33-005

Parameter	Value
g- *gravity*	*9.81 m/s²*
l- *pole length* *	*0.36 to 0.4 m - depending on the configuration*
M- *cart mass*	*2.4 kg*
m- *pole mass*	*0.23 kg*
I- *moment of interia of the pole*	*about 0.099 kg·m² - depends on the configuration*
b - *cart friction coefficient*	*0.05 Ns/m*
d - *pendulum damping coefficient*	*although negligible,necessary in the model- 0.005 Nms/rad*

Figura 4: Taula de paràmetres de la maqueta Feedbak 33-005

**En el nostre cas, s'ha configurat a la llargada del pèndol, l, a 0,36m.*

A més a més, dels paràmetres de la Figura 4, cal considerar que el rang de voltatge de control té un rang de -2,5V a 2,5V, el qual alimenta el motor, generant una força màxima de $\pm$20N.

També cal saber que la llargada del carril és d'1m. Al centre del carril s'hi troba els 0m i en els extrems els $\pm$0,5m.

A més, la maqueta proporciona l'angle del pèndol, en radians, i la posició del carro en el carril, en metres, amb un període d'actualització d'1ms.

1.2 Objectius

A continuació, els principals objectius del treball:

1- Conèixer l'aprenentatge per reforç, més concretament elQ-learning.
2- Adquirir model de simulació del Feedback 55-003, utilitzant algun model existent, que sigui el més semblant possibleal model real.
3- Programar i entrenar el controlador amb el model, amb la finalitat de que el controlador sigui capaç de mantenir el pèndol en posició vertical i el carro al centre del carril.
4- Correcte funcionament del sistema real amb el controlador, després d'aprendre amb el model.

1.3 Planificació del treball

Per a la distribució de tasques entre els membres del grup s'ha planificat un diagrama de Gantt, com es pot veure a la Figura 7.

El treball es divideix en 12 tasques diferents, la durada de cada tasca es defineix en funció de la seva carga de feina o com per exemple la redacció de la memòria, és una tasca que s'ha de fer paral·lelament a la resta. Moltes de les tasques han variat la seva durada respecte a la que s'havia definit de un principi.

La realització de les diferents tasques es marca per un sol integrant del grup, però en el seu desenvolupament també han ajudat i participat els altres integrants.

Setmanalment, s'ha realitzat unareunió amb els integrants per programar les tasques i fer el seguiment d'aquestes. Tot el que s'ha acordat durant les trobades s'ha posat per escrit en actes.

Bloc	***Tasca***	***Integrant***	***Dedicació (h)***
Recerca	Aprenentatge per reforç i Controlador Q-learning	Alba, Glòria	10, 5
	Model de simulació	Ricard	15
	Descripció Feedback 33-005	Laura	13
Desenvolupament	Codi aprenentatge	Alba	17
	Enllaçar codi amb el model	Ricard	10
	Entrenar el controlador	Alba, Ricard	15, 15
	Proves amb el sistema real	Glòria	15
	Millores de programari	Glòria	10
	Avaluació dels resultats	Glòria, Laura	5,5
Memòria	Diagrama de Gantt	Glòria	5
	Redacció de la memòria	Laura	32

Figura 5: Repartició de tasques

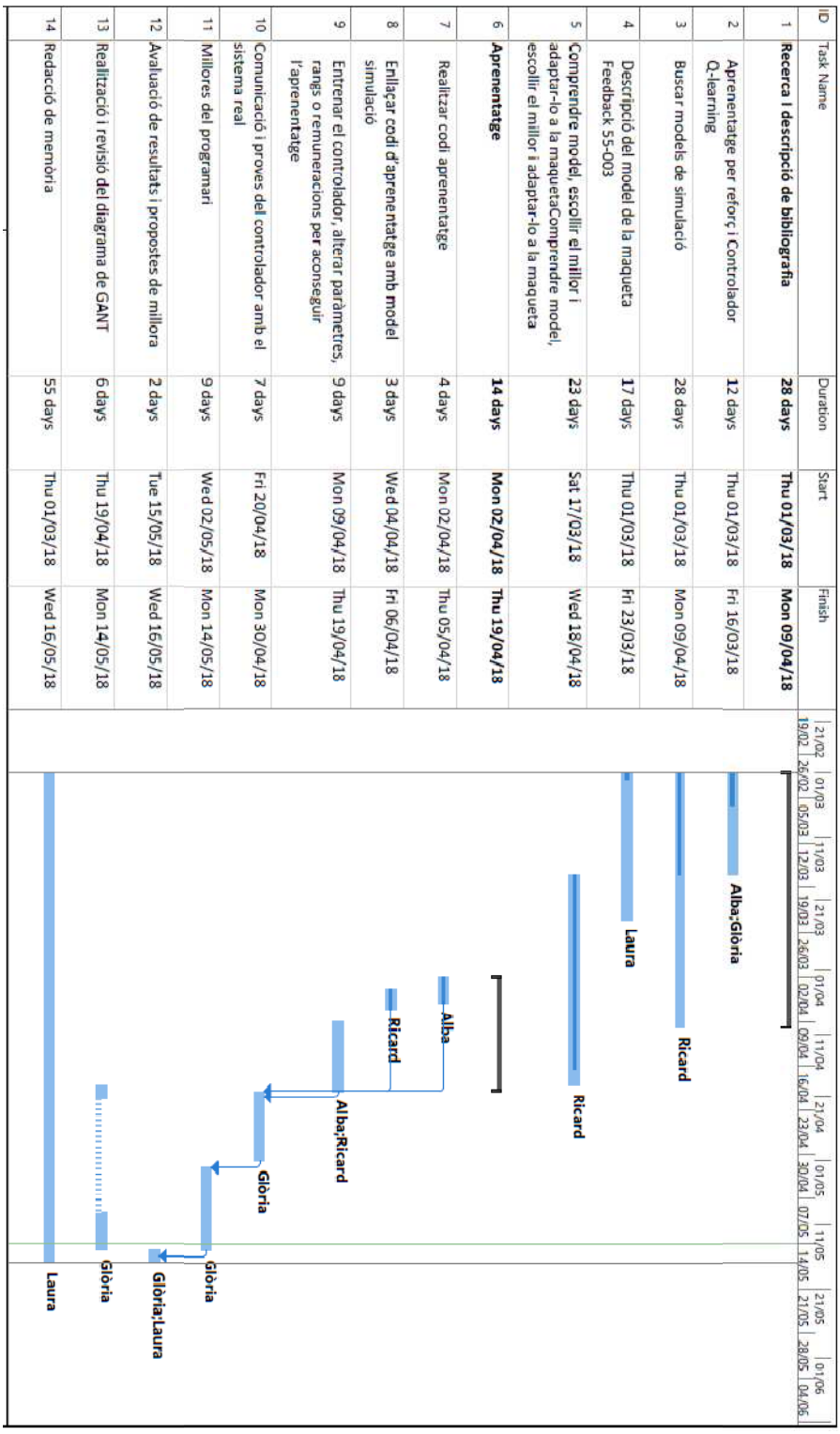


Figura 6: Diagrama de Gantt

2 APRENENTATGE PER REFORÇ

L'aprenentatge per reforç estaria situat entre l'aprenentatge supervisat (on es sap què és el que es vol aprendre) i l'aprenentatge no supervisat (on a partir de dades s'extreuen conclusions). L'aprenentatge per reforç consisteix en aprendre mitjançant un procés de presa de decisions on el feedback és limitat. És a dir, al controlador no se li pot passar la solució ja que es desconeix, però se li pot indicar si la decisió pressa és aportar bons resultats o no. Aquest feedback consisteix en una recompensa que es percep després de prendre cada acció en un estat determinat. En un principi,l'aprenent (controlador) no sap quina acció és millor prendre en cada estat, sinó que ho anirà descobrint en funció de quines accions aportin una major recompensa (o remuneració) a mesura que les vagi provant.

Un dels reptes en l'aprenentatge per reforç, és que l'aprenent ha d'explotar els coneixements que va aprenent (escollir quines accions tenen una recompensa major per tal d'arribar a l'objectiu) a la vegada que ha d'explorar per tal d'identificar accions encara no provades que podrien conduir a una millor solució.

2.1 Elements principals de l'aprenentatge per reforç

- **Agent:** es refereix a l'aprenent. En el cas del projecte que es desenvolupa en aquests document seria el programa que governa el voltatge de control (controlador).
- **Entorn:**es refereix a l'ambient on l'agent pot interactuar. L'entorn pot presentar diferents estats i l'agent ha de prendre la decisió de quina acció prendre en l'estat en que es troba. En aquest projecte, l'entorn seria el sistema carro–pèndol, incloent les posicions i velocitats d'ambdós elements.
- **Política (π):**es refereix al comportament de l'agent en un moment determinat (quina acció prendre en cada estat). En determinats casos pot ser representat mitjançant una taula o una funció.
- **Funció de recompensa (r):**defineix l'objectiu: enllaça parelles d'estat – acció amb un sol valor (la recompensa) que indica la conveniència de l'estat al que han portat. La funció de recompensa defineix quines són les bones i les males accions.
- **Funció de valor (V):**similar a la funció de recompensa, però mentre que en la recompensa sol s'avalua el següent estat immediat, la funció de valor especifica el queés bo allarg termini (valorant les recompenses que pot acumular partint des d'un estat específic).

2.2 Algoritmes per l'aprenentatge per reforç

Existeixen tres classes fonamentals de mètodes per resoldre els problemes d'aprenentatge per reforç: Programació dinàmica (*Dynamic Programming – DP*), mètodes de Monte Carlo i aprenentatge amb diferència temporal (*Temporal Difference – TD*). A continuació, es comenten els principals avantatges i inconvenients dels tres mètodes.

Els mètodes de programació dinàmica fan possible l'aprenentatge pas a pas (en cada acció aplicada) però requereixen un coneixement complet de l'entorn ja que és necessari treballar amb un model molt acurat: es necessita saber des d'un principi a quin estat s'arriba després d'aplicar qualsevol acció en qualsevol estat (el que vindria a ser un mapa complet de tots els estats i les accions que provoquen els canvis entre ells).

En la vida real és complicat trobar un sistema del que es pugui obtenir un model tant detallat. A més, també es pot donar el cas que el mapa sigui massa complicat. Per exemple, sí seriem capaços de fer un mapeig de tots els estats del 3 en ratlla, però el joc dels escacs comprèn tantes possibilitats que el mapa seria intractable.

Els mètodes de Monte Carlo no requereixen d'un coneixement complet de l'entorn, però l'aprenentatge no realitza pas a pas, sinó que es fa en completar un episodi sencer. Entenem com a episodi una simulació o execució completa, des del seu inici en les condicions inicials fins complir les condicions de finalització(quan s'ha complert l'objectiu, quan s'ha superat el temps màxim de simulació...). En un episodi es produeixen diversos passos (accions aplicades).

Els mètodes de diferència temporal són una combinació dels dos mètodes comentats anteriorment: són capaços d'aprendre directament de l'experiència, sense un mapa complet de tots els estats i les accions que els comuniquen, i a més poden aprendre pas a pas. Aquests mètodes són els més innovadors pel que fa a l'aprenentatge per reforç.

A continuació, es comenten tres algoritmes de diferència temporal.

2.2.1 Algoritme TD(0)

L'algoritme més simple, conegut com TD(0), Sutton (1988), serveix per resoldre problemes de predicció, en els quals s'avalua la funció de valor sota una política π coneguda.

L'algoritme es detalla en laFigura 7. Primerament, s'inicialitza la funció de valor (el valor de cada estat) de forma arbitrària, i es defineix la política π a ser avaluada. Per cada episodi, s'inicialitza l'estat (s) i es selecciona l'acció a

efectuar seguint la política escollida. En efectuar l'acció s'observa el nou estat on hem arribat i la recompensa rebuda. A continuació, s'actualitza el valor de l'estat anterior en funció de la recompensa i del valor de l'estat al que hem arribat. El procés de selecció de l'acció i l'actualització de la funció de valor segueix fins arribar a l'estat final o en complir la condició de finalització.

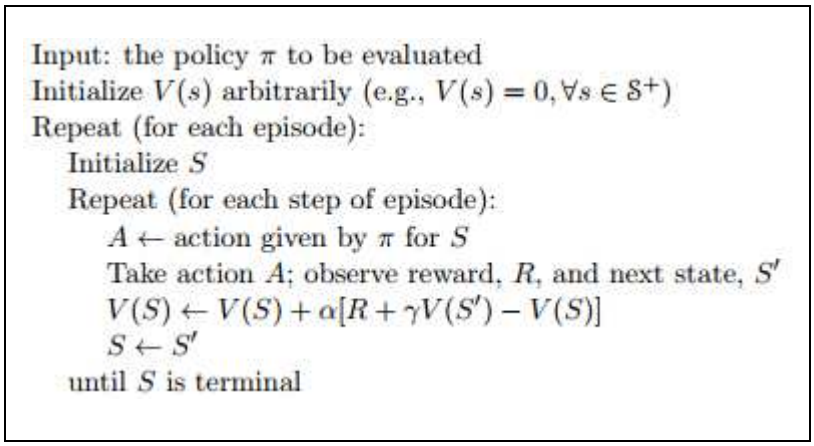


Figura 7: Algoritme TD(0)(font: Reinforcement Learning – An Introduction*)*

On:

- *S: és l'estat del qual partim.*
- *A: és l'acció presa en l'estat S.*
- *R : és la recompensa rebuda per passar a l'estat S'.*
- *S': és l'estat en que ens situem després de prendre l'acció A.*
- *V(S): és el valor de l'estat S.*
- *V(S'): és el valor de l'estat S'.*
- *π: és la política a avaluar.*
- *α: és el factor d'aprenentatge [0,1].*
- *γ: és el factor de descompte [0,1] Indica la importància de les remuneracions futures.*

Aquest algoritme no pretén trobar la política òptima, sinó trobar la funció de valor per una política determinada.

A continuació, es tractaran dos algoritmes utilitzats en problemes de control (on l'objectiu és trobar la política òptima): Q-learning i Sarsa.

2.2.2 Algoritme Q-learning

L'algoritme Q-learning, Watkins (1989), s'utilitza en problemes de control per tal de trobar la política òptima. Aquest algoritme és de tipus *off-policy,* que significa que per l'actualització de la funció de valor per un estat, no és necessari escollir l'acció a efectuar en el següent estat.

En l'algoritme Q-learning, la funció de valor no guarda el valor de cada estat (com en el cas de TD(0)) sinó que guarda el valor de cada parella estat-acció en la matriu Q. La política òptima consistirà en elegir l'acció que tingui un valor major per cada estat.

Aquest algoritme és capaç d'aprendre la funció de valor Q òptima independentment de la política que es segueixi en l'aprenentatge, sempre que el valor de totes les parelles estat-acció s'actualitzi freqüentment. La política d'aprenentatge pot consistir en aplicar directament la política derivada de la Q actual (la política òptima), o pot contenir factors d'exploració per tal de maximitzar els estats visitats.

L'algoritme es detalla en laFigura 8. Primerament, s'inicialitza la matriu Q de forma arbitrària, i per cada episodi es selecciona l'estat inicial (s) i s'escull l'acció a efectuar seguint la política d'exploració. En efectuar l'acció s'observa el nou estat on hem arribat i la recompensa rebuda. A continuació, s'actualitza el valor de la parella estat-acció anteriors. El procés de selecció de l'acció i actualització de la funció de valor Q segueix fins arribar a l'estat final o en complir-se la condició de finalització.

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(terminal\text{-}state,\cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
    $S \leftarrow S'$
  until S is terminal

*Figura 8: Algoritme Q-learning (font:*Reinforcement Learning – An Introduction*)*

On:

- *S: és l'estat actual.*
- *A: és l'acció presa en l'estat S.*
- *R : és la recompensa rebuda per passar a l'estat S'.*
- *S': és l'estat següent, en que ens situem després de prendre l'acció A en l'estat actual S.*
- *Q(S,A): és el valor d'efectuar l'acció A en l'estat S.*
- *max $_{a'}$ Q(S',a): és el màxim valor en l'estat S' (entre els valors corresponents a cada una de les accions possibles en l'estat S', es pren el màxim).*
- *α: és el factor d'aprenentatge [0,1] .*
- *γ: és el factor de descompte [0,1] Indica la importància de les remuneracions futures.*

A continuació s'analitza més detalladament el mecanisme d'actualització de la funció de valor Q, que es mostra en la Figura 9amb subíndexs temporals.

$$Q_{k+1}(s_t,a_t) = Q_k(s_t,a_t) + \alpha\left(r_t + \gamma \max_a Q_k(s_{t+1},a) - Q_k(s_t,a_t)\right)$$

Figura 9: Actualització de la funció de valor Q en l'algoritme Q-learning

Observem com, primerament, s'està en el estat s_t, en el qual es pren l'acció a_t. Un cop efectuada aquesta acció, s'arriba al estat s_{t+1}. En aquest punt, s'actualitza la funció de valor per la parella s_t i a_t, considerant el màxim valor

previst per $Q(s_{t+1}, a)$. És a dir, per actualitzar el valor de $Q(s_t, a_t)$ no es té en compte l'acció presa en l'estat s_{t+1}, sinó que s'actualitza amb el valor de la parella $Q(s_{t+1}, a)$ que aporti un valor màxim.

2.2.3 Algoritme SARSA

Rummery and Niranjan (1994) **S**tate–**A**ction–**R**eward–**S**tate–**A**ction.

Aquest algoritme és de tipus *on-policy*, ja que, s'actualitza la funció de valor (Q) per la parella estat-acció actuals en funció de la parella estat-acció següent.

L'algoritme d'aprenentatge de SARSA (Figura 10) varia lleugerament respecte el Q-learning, ja que, en aquest cas, l'elecció de l'acció a efectuar es fa previ a l'actualització de la funció de valor.

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(terminal\text{-}state, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

*Figura 10: Algoritme SARSA (font:*Reinforcement Learning – An Introduction*)*

On:

- *S: és l'estat actual.*
- *A: és l'acció presa en l'estat S.*
- *R : és la recompensa rebuda per passar a l'estat S'.*
- *S': és l'estat següent, en que ens situem després de prendre l'acció A en l'estat actual S.*
- *A' és l'acció presa en l'estat S'.*
- *Q(S,A): és el valor d'efectuar l'acció A en l'estat S.*
- *Q(S',A'): és el valor d'efectuar l'acció A' en l'estat S'.*
- *α: és el factor d'aprenentatge [0,1].*
- *γ: és el factor de descompte [0,1] Indica la importància de les remuneracions futures.*

A continuació s'analitza més detalladament el mecanisme d'actualització de la funció de valor Q de l'algoritme SARSA, que es mostra en la Figura 11amb subíndexs temporals, comparant-lo amb l'algoritme Q-learning.

$$Q_{t+1}(s_t,a_t) = Q_t(s_t,a_t) + \alpha\left(r_t + \gamma Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)\right)$$

Figura 11: Actualització de la funció de valor Q en l'algoritme SARSA

En l'algoritme SARSA, sí es considera el valor de $Q(s_{t+1}, a_{t+1})$ per l'actualització de $Q(s_t,a_t)$. És a dir, primerament, s'està en l'estat s_t, en el qual es pren l'acció a_t. Un cop efectuada aquesta acció, s'arriba al'estat s_{t+1}, on s'efectua l'acció a_{t+1}.

Llavors s'actualitza $Q(s_t,a_t)$ tenint en compte el valor de $Q(s_{t+1}, a_{t+1})$ per al càlcul de $Q(s_t, a_t)$.

La diferència entre els algoritmes Q-learning i SARSA radica en el terme $maxQ(s_{t+1}, a)$ o $Q(s_{t+1}, a_{t+1})$. En el Q-learning sempre es pren el màxim valor del següent estat, mentre que SARSA pren el valor de la següent parella estat-acció.

S'ha de tenir en compte, que l'elecció de l'acció a efectuar es fa mitjançant una política que deriva de la funció de valor Q juntament amb una política exploratòria (per tal de visitar tots els estats sovintment). Per tant, el valor de $Q(s_{t+1}, a_{t+1})$ no té perquè coincidir amb $maxQ(s_{t+1}, a)$. Aquí radica la diferència dels dos algoritmes.

3 CONTROLADOR Q-LEARNING PEL SISTEMA CARRO-PÈNDOL

El controlador aplicat al sistema és una variació de l'algoritme Q-learning, adaptat al sistema carro-pèndol.L'Algoritme Q-learning típices basa en l'aprenentatge que es realitza en diferents "episodis", tots ells partint de l'estat inicial, i finalitzant un cop s'arriba a l'estat objectiu.

En el cas del pèndul invertit, apareix una lleugera variació, ja que en l'estat inicial el pèndul estarà invertit aproximadament 180º de la seva situació de repòs (en l'instant t=0 subjectat per algú de l'equip amb angle proper als 0º) i la tasca del controlador serà mantenir-lo en aquesta posició, sense allunyar-se de la posició central del seu recorregut, per tant l'objectiu serà que es quedi el màxim de temps possible en posició invertida (la posició inicial). En el treball s'ha considerat que un temps de 10s en posició invertida és suficient per suposar que, partint de l'estat inicial en qüestió, ha après el suficient per aguantar-se invertit un període més llarg de temps.

En definitiva, els diferents "episodis" de l'aprenentatge, es desenvoluparan començant en un estat amb el pèndol invertit i acabant quan aquest hagi caigut o quan hagi aguantat 10s en la posició invertida. Es considera que el pèndol ha caigut quan l'angle del pèndol cau fora del rang de [-10º, 10º] o quan la posició del carro arriba al límit del seu recorregut (-0,5m o +0,5m).

Realitzar l'aprenentatge amb la maqueta real podria ser molt costós pel que fa a temps i dedicació, ja que, en cada episodi s'hauria de moure el pèndol en les seves condicions inicials. Així doncs, primerament es realitza un aprenentatge amb el model, on la posició inicial serà centrada en el carril i l'angle inicial estarà entre els -2º i els 2º (invertit).

L'aprenentatge finalitza quan 10 vegades seguides s'assoleixen 10s de simulació (manté el pèndol 10s en posició invertida en 10 episodis consecutius). En aquest cas es considera que el controlador ha completat l'aprenentatge (és capaç de mantenir el pèndol invertit durant llargs períodes de temps en diferents angles inicials).

A continuació, es plantegen les diferents consideracions que s'han pres per implementar el controlador Q-learning pel que fa als estats, les accions, les remuneracions, l'actualització de la matriu Q, i la política a seguir en l'aprenentatge.

3.1 Definició dels estats: Discretització dels valors

El sistema carro-pèndol treballa amb senyals analògiques, que presenten valors continus al llarg del temps. En canvi l'algoritme Q-learning treballa amb estats discrets. Per poder aplicar el controlador sobre el sistema s'han definit diferents criteris per poder discretitzar les senyals del sistema carro-pèndol en estats.

Un estat concret es compon per la posició del carro, l'angle del pèndol, el sentit de la velocitat angular del pèndol i el sentit de la velocitat del carro. Com s'ha comentat, però, en comptes de treballar amb el valor real d'aquests paràmetres, es treballa amb els valors discretitzats en rangs.

3.1.1 Posició del carro

Per a la discretització de la posició s'ha dividit el carril on es desplaça el carro de 1 m de llargada en 5 parts.

A la Figura 12es pot veure els 5 rangs de valors en que es divideix el carril, en groc la part central que va des del els 150mm fins els -150mm. En taronja, els rangs que, a continuació dels de la zona groga, arriben fins a 350mm i -350mm. En vermell els rangs que arriben fins a -500mm i 500mm.

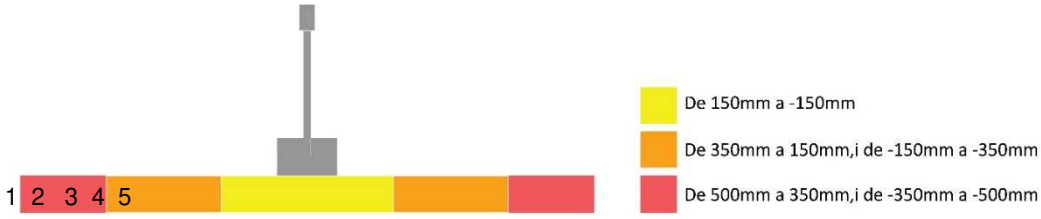


Figura 12: Divisió de la posició del carro

3.1.2 Angle del pèndol

Per a la discretització dels angles s'ha tingut en compte que la situació amb el pèndol perfectament invertit correspon als 0º, i la simulació s'executa dins els rang de [-10º a 10º], en sortir d'aquest rang la simulació es para i comença un nou episodi. L'arc de circumferència dels -10º als 10º s'ha dividit en 7 segments o rangs.

En la Figura 13es veuen els rangs en que s'ha discretitzat l'angle del pèndol. En verd es poden veure la zona que engloba els angles compresos entre -0,5º i 0,5º. En groc es troben els angles que parteixen dels angles anteriors i arriben fins a -2,5º i 2,5º. En taronja queden els que arriben fins a -5,5º i 5,5º. Per últim, en vermell, els que arriben fins a 10º i -10º.

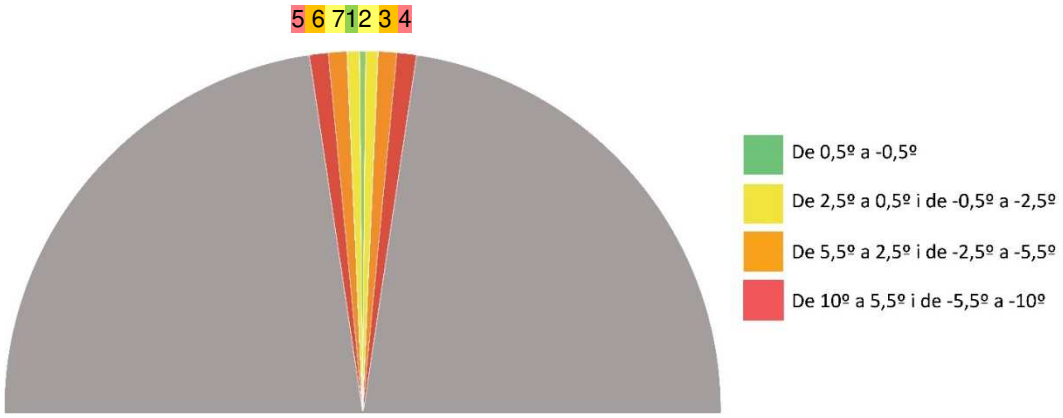
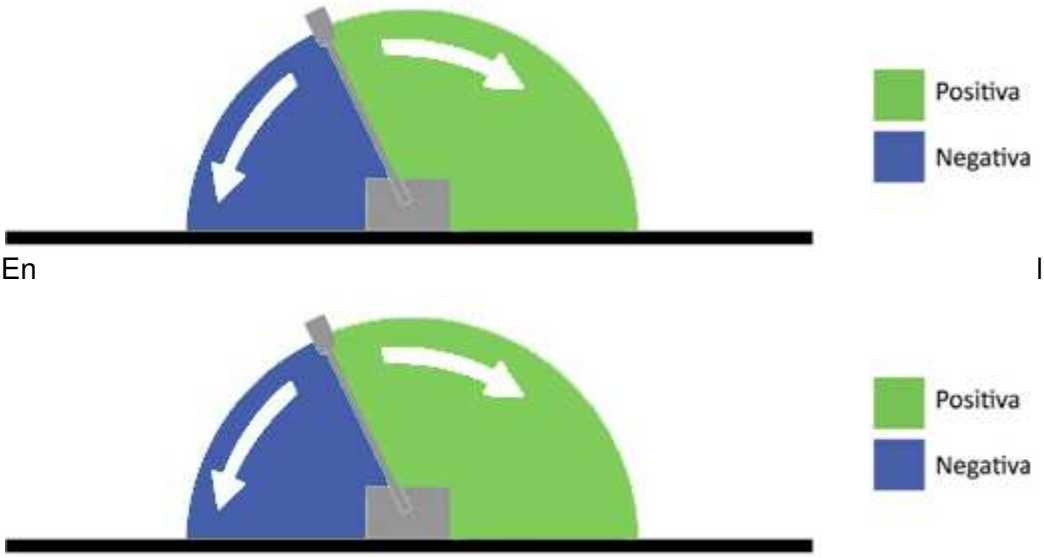


Figura 13:Divisió dels angles

3.1.3 Velocitat angular del pèndol

Per a poder definir en quin sentit esta girant el pèndol en l'estat en que es troba el sistema, s'ha agafatel signe de la velocitat angular (positiva o negativa). Per fer-ho es compara l'angle de la mostra anterior amb l'angle de la mostra actual.



En la

Figura 14es pot veure com en verd el pèndol giraria en direcció positiva i en blau negativa (la posició en que figura el pèndol seria la posició de la mostra anterior, si en la mostra actual el pèndol està en la zona verda, la velocitat angular serà positiva, si està en la zona blava, serà negativa).

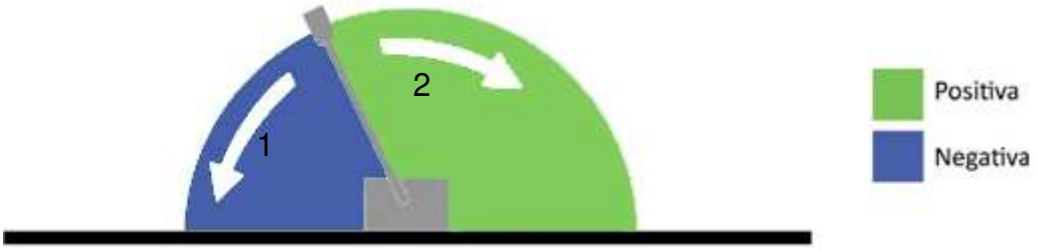


Figura 14:Direcció de gir del pèndol

3.1.4 Velocitat lineal del carro

Per definir el sentit del desplaçament del carro, s'ha tingut en compte el signe dela velocitat lineal. Es parteix de que la posició és de 0m en el centre del recorregut, -0,5m al topall esquerra i 0,5m al topall de la dreta.La velocitat serà positiva si la posició del carro és major en l'instant actual que en l'anterior, per contra, si la posició actual és menor que l'anterior, la velocitat és negativa.

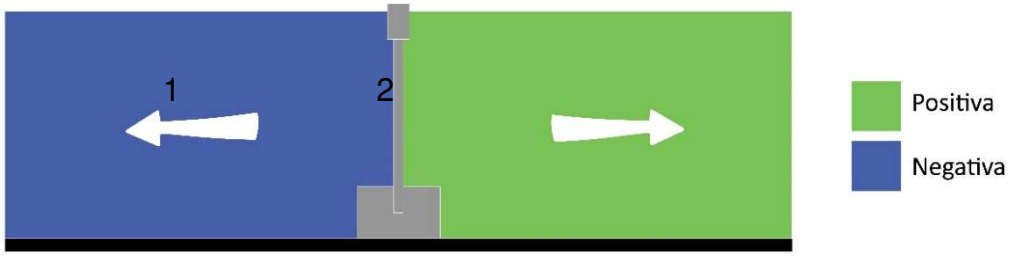


Figura 15: Direcció del carro

3.2 Acció: Voltatge de control

Com s'ha comentat anteriorment, el rang de voltatge de control és una senyal analògica des de -2,5V fins a 2,5V. Les accions del controlador Q-learning, però, són discretes i finites, així doncs s'ha optat per tenir 11 possibles accions, de -2,5V amb increments de 0,5V fins als 2,5V: {-2,5 -2 -1,5 -1 -0,5 0 0,5 1 1,5 2 2,5}V.

3.3 Remuneracions

La remuneració és el reforç que es dóna a una parella estat-acció. El reforç es calcula per l'estat previ en funció de l'estat actual al que s'ha arribat aplicant l'acció escollida en l'estat previ.

Si l'estat actual es considera bo, el reforç per la parella estat-acció prèvia serà de 0, si per contra l'estat actual es considera dolent, el reforç per la parella estat-acció prèvia serà negatiu: -1.

Es considera un estat dolent si és es troba en els rangs extrems de posició del carro i/o de l'angle del pèndol: posició major que 350mm o menor que -350mm i/o angle major que 5,5º o menor que -5,5º.

Cal dir que aquest ha estat un dels punts crítics per aconseguir l'aprenentatge, havent provat també reforços menys dràstics i també recompensar amb reforç positiu el fet d'aguantar un temps elevat en posició invertida. Al final les remuneracions més simples explicades prèviament són les que han donat millors resultats (assolir l'aprenentatge).

3.4 Actualització de la matriu Q

En l'algoritme Q-learning la matriu Q recull la informació del "valor" de cada parella estat-acció.

En l'inici de l'aprenentatge la matriu s'emplena de 0, i es va actualitzant cada 1ms. En finalitzar un episodi la matriu Q es guarda, i serà la matriu Q inicial pel proper episodi.

En el controlador desenvolupat la matriu Q té 5 dimensions, les quatre primeres corresponen a l'estat del sistema (posició, angle, velocitat angular i velocitat lineal), i la última correspon a les accions que es poden prendre en cada estat. En concret la mida de la matriu Q en cada dimensió correspon als rangs comentats en la discretització i en el nombre d'accions possibles: (5, 7, 2, 2, 11).

Per actualitzar la matriu Q, el controlador llegeix les variables del sistema (angle del pèndol i posició del carro) cada 1ms, i així també cada 1ms s'actualitza la posició de la matriu Q corresponent ala parella estat-acció prèvia en funció de la Q que tenia anteriorment aquella parella, la remuneració que se li ha donat i el màxim valor de Q de l'estat actual aplicant qualsevol acció.

En la Figura 16es pot veure el codi d'actualització de Q en l'algoritme Q-learning:

```
Q(x_prev, angle_prev, vel_ang_prev, vel_x_prev, accio_prev)=
(1-a)*Q (x_prev, angle_prev, vel_ang_prev, vel_x_prev,accio_prev)
+a*(r+g*max(Q(x_mat,angle_mat,vel_ang_mat,vel_x_mat,:)));
```

Figura 16: Codi d'actualització de Q en l'algoritmeQ-learning

x_prev= Valor discret posició estat previ.
angle_prev= Valor discret angle previ.
vel_ang_prev =Valor discret direcció gir prèvia.
vel_x_prev=Valor discret direcció carro prèvia.
accio_prev = Acció presa en l'estat previ.
x_mat=Valor discret de la posició actual.
angle_mat= Valor discret de l'angle actual.
vel_ang_mat= Valor discret direcció gir actual.
vel_x_mat= Valor discret direcció carro actual.
a=Paràmetre Alpha (α).
g=Paràmetre Gamma (γ).
r=Remuneració.

Com s'ha comentat, cada 1ms el sistema actualitza el seu estat i el programa actualitzaria la matriu "*Q*" de l'estat previ, en que es té en compte el valor que ja estava guardat a la matriu (amb un factor 1-α) i se li sumael resultat de haver realitzat aquesta acció (amb un factor α) i el màxim valor en l'estat actual (amb un factor α*γ) .

Els paràmetres constants de la funció, α i γ (a i g) es mantindran sempre en els mateixos valors:a=0,4 i g=0,8.

Un valor de α de 0,4 permet conservar l'aprenentatge realitzant fins al moment (té en compte els històrics, és necessari que una acció sigui dolenta de forma repetida per tal d'assolir un valor baix). Un valor de gamma de 0.8 dóna una certa importància als possibles estat futurs propers. És a dir, per un valor de gamma de 0 no es tindria en compte el valor dels estats futurs. Mentre que per gamma igual a 1, tots el valors dels estats futurs repercutiria al valor a actualitzar.

3.5 Política d'aprenentatge

En l'aprenentatge no es segueix una política òptima totalment que seria aplicar sempre l'acció que té un valor major (o una d'aleatòria entre les que tenen el valor major), sinó que també es combina amb una política exploratòria, exigint executar una acció si en l'estat actual aquesta s'ha executat menys d'un 2% de les vegades.

Per aconseguir controlar l'exploració, paral·lelament a la matriu "*Q",* es va omplint una matriu *"Q_num",* que té les mateixes dimensions que Q i també s'inicialitza a zeros al començar l'aprenentatge. Aquesta matriu recollirà per cada parella estat-acció la quantitat de vegades que s'ha visitat (sumant 1 a cada vegada que es visita).

Així doncs, previ a l'elecció de la millor acció a efectuar es comprova si en l'estat actual alguna s'ha visitat menys del 2% de les vegades que s'ha visitat aquell mateix estat i s'executa (en cas d'haver més d'una acció executada menys del 2% de les vegades se n'elegeix una a l'atzar). En cas que totes les accions s'hagin executat almenys un 2% de les vegades, s'escull l'acció que en la matriu Q presenta un valor major per l'estat actual.

En l'avaluació del controlador, la política que es segueix sí és la òptima.

Un cop s'escull l'acció a executar, aquesta es traspassa a voltatge, i els valors previs s'igualaran als valors actuals.

4 APRENENTATGE AMB EL MODEL DEL SISTEMA

En aquest apartat es comenta el model obtingut del sistema i l'aprenentatge realitzat amb ell.

4.1 Model en Simulink del sistema

Per tal de simular el comportament de la maqueta, s'ha utilitzat un bloc de Simulink (Matlab) proporcionat pel fabricant. Aquest bloc representa les equacions del sistema carro-pèndol, tal com es mostra en la Figura 17.A més a més, s'han introduït els paràmetres de la maqueta Feedback 33-005per tal de modelitzar de la forma més acurada possible el sistema real.

El bloc del modelés un sistema SIMO- Single Input Múltiple Output, on l'entrada és el voltatge de control i les sortides són l'angle del braç del pèndol i la posició del carro sobre el carril, simulant d'aquesta manera el sistema de la maqueta real.

Igual que en el sistema real, l'angle del braç té les unitats en radians, i els 0 radians coincideix amb la posició invertida del pèndol; i la posició del carro és mesurada en metres i els 0metres coincideixen amb la posició central del carro en el carril.

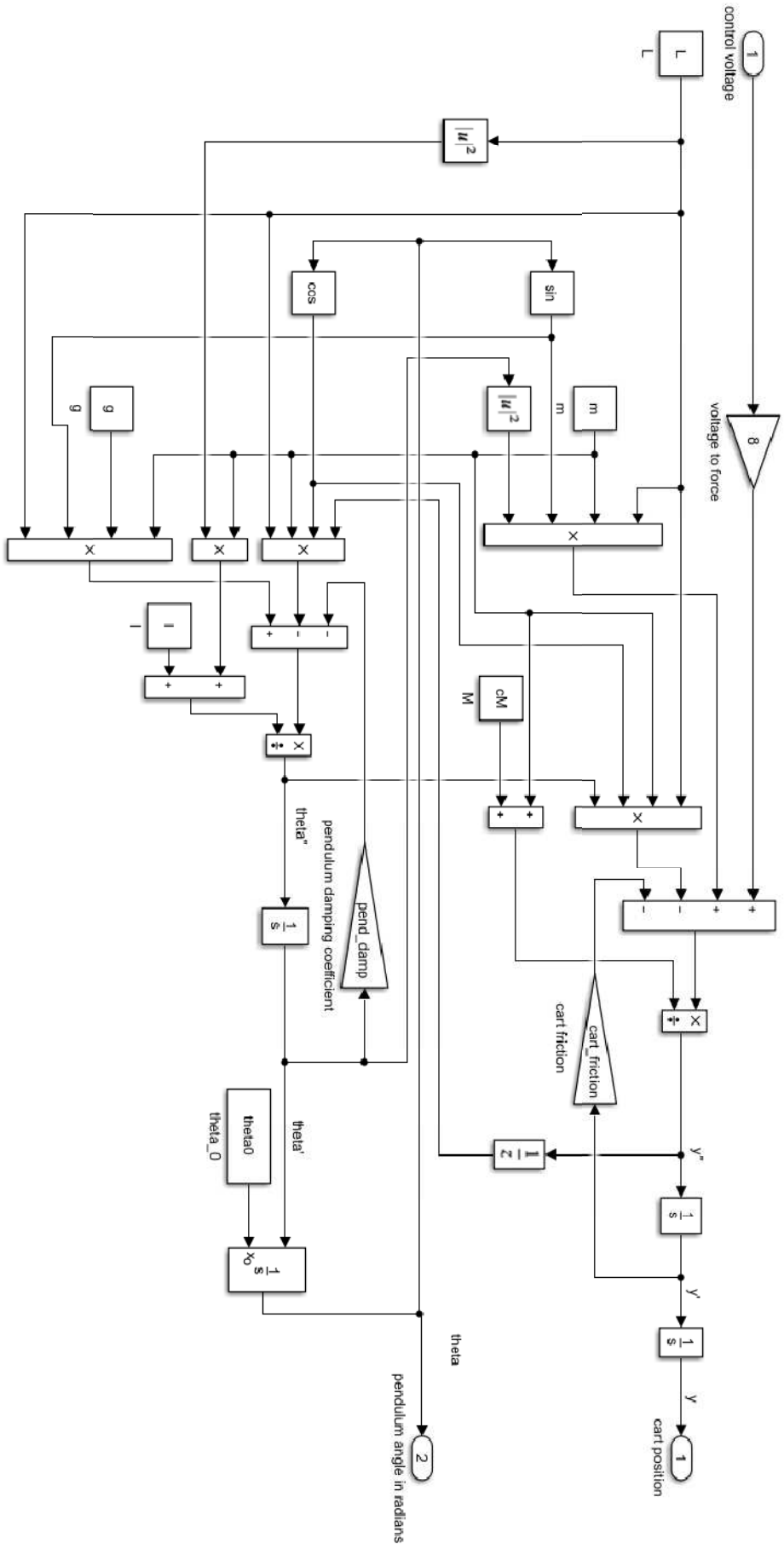


Figura 17: Bloc del model amb Simulink

4.2 Incorporació del Q-learning en el model

En el Simulink s'ha incorporat el codi del controlador Q-learning en una "MATLAB Function", que té com a entrades la posició en x i l'angle del pèndol (que rep del model) i la sortida és el voltatge de control (que s'envia al model). També s'ha definit que la simulació s'aturi quan la posició del carro sobrepasi els limits del carril o quan l'angle sigui superior a un angle de 10º o inferior a -10º, situació que s'ha consideratque el sistema ja no serà capaç de recuperar-se.

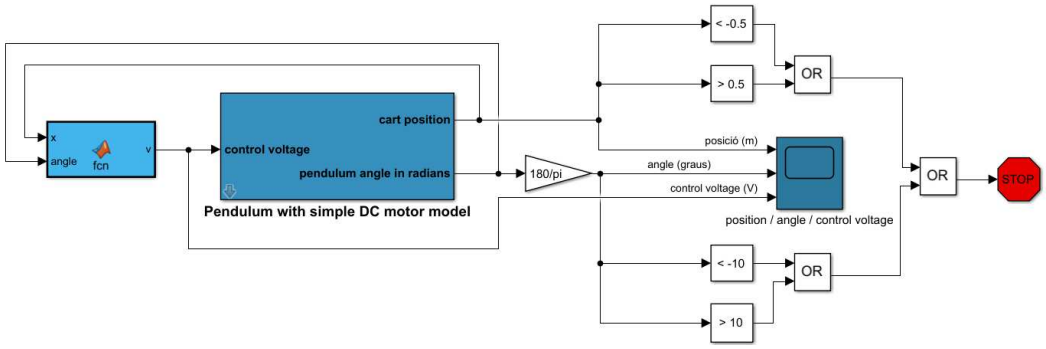


Figura 18: Model amb controlador des de Simulink

El bloc "*Pendulum with simple DC motor model*" conté el model en Simulink que s'ha mostrat anteriorment. Mitjançant el formulari que es mostra al'entrar dins del bloc del model, Figura 19, s'introdueixen els paràmetres del sistema real.

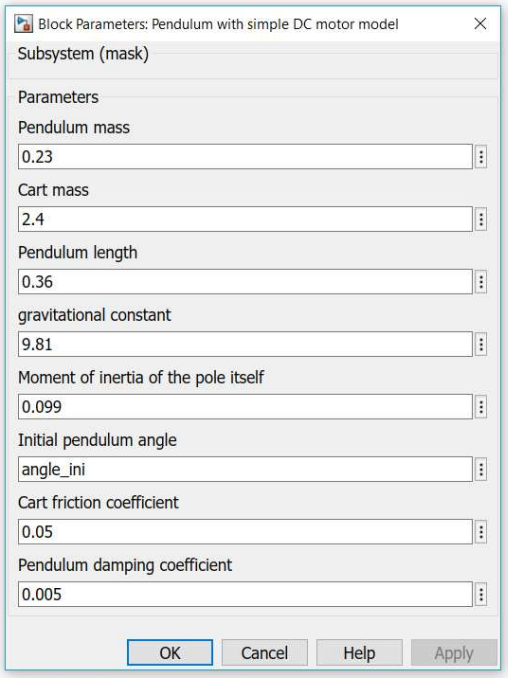


Figura 19: Paràmetres entrats al model

En la funció que conté el controlador Q-learning es defineixen de forma persistent les variables que ha de mantenir en memòria entre diferents crides de

la funció (dins la mateixa simulació). Per tal de conservar el valor de Q i Q_num entre diferents crides al Simulink, aquestes matrius s'importen i s'exporten al WorkSpace base amb les funcions "*evalin"* i "*assignin"*.

En el scope, es mostra la gràfica dels valors de la posició del carro, l'angle del pèndol i el voltatge de sortida.

Per poder realitzar diverses repeticions (episodis) de la simulació, s'ha dissenyat un programa (*Script*) que crida repetidament el programa en Simulink alterant l'angle inicial (angle_ini) entre -2º i 2º. Aquest*Script* també indica si el controlador ha d'aprendre o no i en cas afirmatiu amb quina exploració.

Només en el cas que es simulin 10 iteracions o episodis consecutius en que el pèndol es mantingui 10 segons invertit, el *Script* que crida el Simulink donarà per finalitzat l'aprenentatge. En aquesta situació s'entendrà que el controlador ja ha après suficient com per mantenir el pèndol invertit per qualsevol angle inicial entre -2º i 2º.

Cal considerar que el sistema carro-pèndol amb les discretitzacions necessàries per definir els estats no és del tot determinista. És a dir, aplicar una acció en un mateix estat no sempre ens portarà al mateix estat següent.

En la Figura 21es mostren un exemple d'aquest cas, en que s'aplica una acció (un voltatge de control) per invertir els sentit de les velocitats lineal i angular. El primer cop que s'aplica l'acció es frenen les inèrcies (sense modificar l'estat) i el segon cop sí es canvien els sentits de gir i de la direcció del carro, i per tant canvia d'estat.

Tenint en compte que el sistema no es determinista, la convergència de Q per obtenir la política òptima no està assegurada, així doncs, es poden donar casos en que l'aprenentatge no sigui fructuós, encara que es repeteixin un gran nombre d'episodis.

Per tal d'evitar realitzar infinits episodis sense arribar a trobar la política òptima, es defineix el nombre màxims d'episodis d'una prova en 2000. Si passats 2000 episodis no ha aconseguit aguantar en 10 episodis consecutius 10s el pèndol invertit, es comença una nova prova inicialitzant les matrius Q i Q_num a zeros.

En la Figura 21es mostra pseudocodi que conté la crida de la simulació en cada un dels episodis juntament amb el Q-learning que conté la simulació.

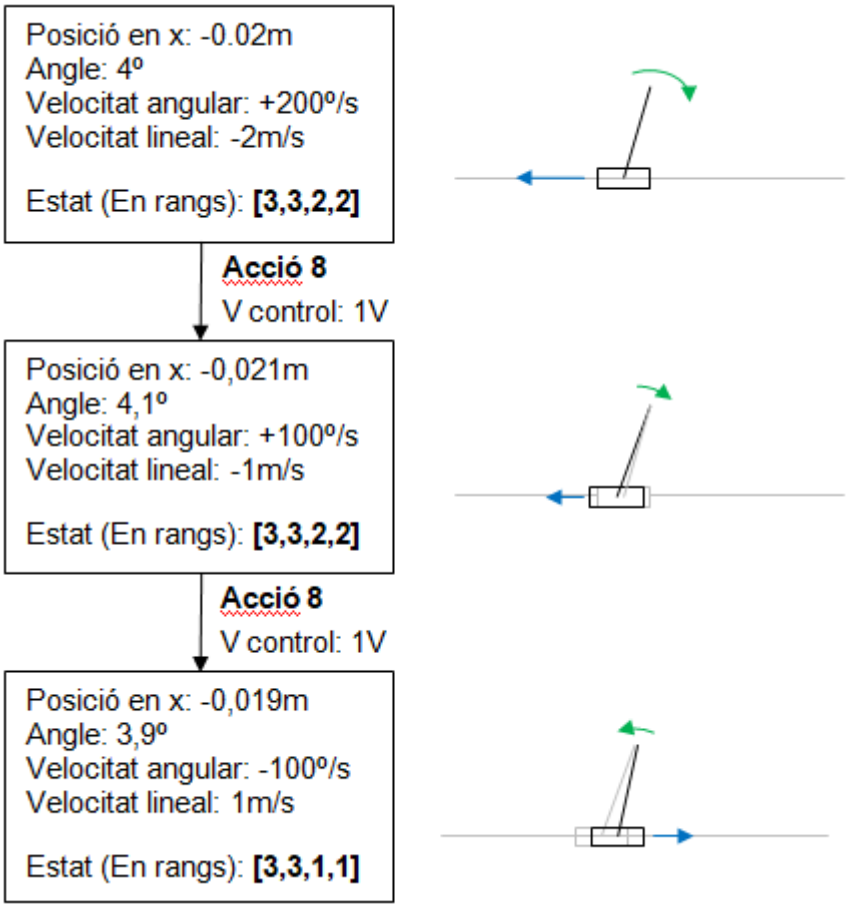


Figura 20: Exemplificació del no determinisme del sistema amb discretitzacions
Executar una mateixa acció en un mateix estat no porta al mateix estat següent

APRENENTATGE:

Repetir proves

Inicialitzar Q i Q_num (zeros) i paràmetres d'aprenentatge alpha(α) i gamma(γ).

N_episodis=0 (conta el nº d'episodis de la prova)

N_episodis_10s=0 (conta el nº d'episodis consecutius que s'arriba als 10s de simulació)

Repetir episodis

Determinar aleatòriament l'angle inicial entre -2º i 2º

t=0 (conta el temps de simulació)

Inicia simulació(en Simulink)

Executar simulació, obtenir posició del carro (x) i angle del pèndol (angle).

Cada 1ms executar algoritme Q-learning(Dins MatlabFunction de Simulink)**:**

t=t+0,001

Discretitzar:x, angle, velocitat angular i velocitat lineal →(estat actual)

Assignar la remuneració per arribar a l'estat actual (r)

Actualitzar el valor en Q per l'estat previ (si no estem en l'estat inicial):

Q(estat previ, acció prèvia) =

(1-α)*Q(estat previ, acció prèvia)

+α*(r+γ*max(Q(estat actual, :))

Escollir l'acció a efectuar (acció actual)

Correspondència entre acció actual {1-11} i Voltatge {-2,5 – 2,5}

estat previ = estat actual

acció prèvia = acció actual

Sortida Q-learning: Voltatge

Fi simulació si t=10s o angle<-10º o angle>10º o x<-0,5m o x>0,5m

N_episodis=N_episodis+1

Si t=10s

N_episodis_10s=N_episodis_10s+1

Sinó

N_episodis_10s=0

Fi Si

Fins que N_episodis==2000 o N_Episodis_10s==10,

Fins N_Episodis_10s==10

Figura 21: Pseudocodi d'aprenentatge

4.3 Resultats de l'aprenentatge amb el model

Les series de simulacions s'han realitzat de 2000 episodis de 10 segons com a màxim cada una (el temps de cada episodi és el temps que el sistema aguanta dins els límits, si arriba a 10s la simulació es dóna com a bona i es para per començar el següent episodi). En cas d'haver realitzat 2000 episodis i no haver complert l'objectiu d'aprenentatge es reinicialitza la matriu Q i Q_num a zeros i es comença una prova l'aprenentatge des de zero. Pot ser que degut a les aleatorietats, s'hagi arribat a un estat de Q en que sigui més difícil l'aprenentatge que en l'estat inicial de Q amb tot de zeros.

A la Figura 22: Simulació al inici del aprenentatgees veu una gràfica de sortida extreta al inici de l'aprenentatge. S'observa que la simulació no arriba als 3 segons ja que s'arriba als límits. A més a més, es veu que el voltatge de sortida no segueix cap patró regular en el conjunt de la simulació.

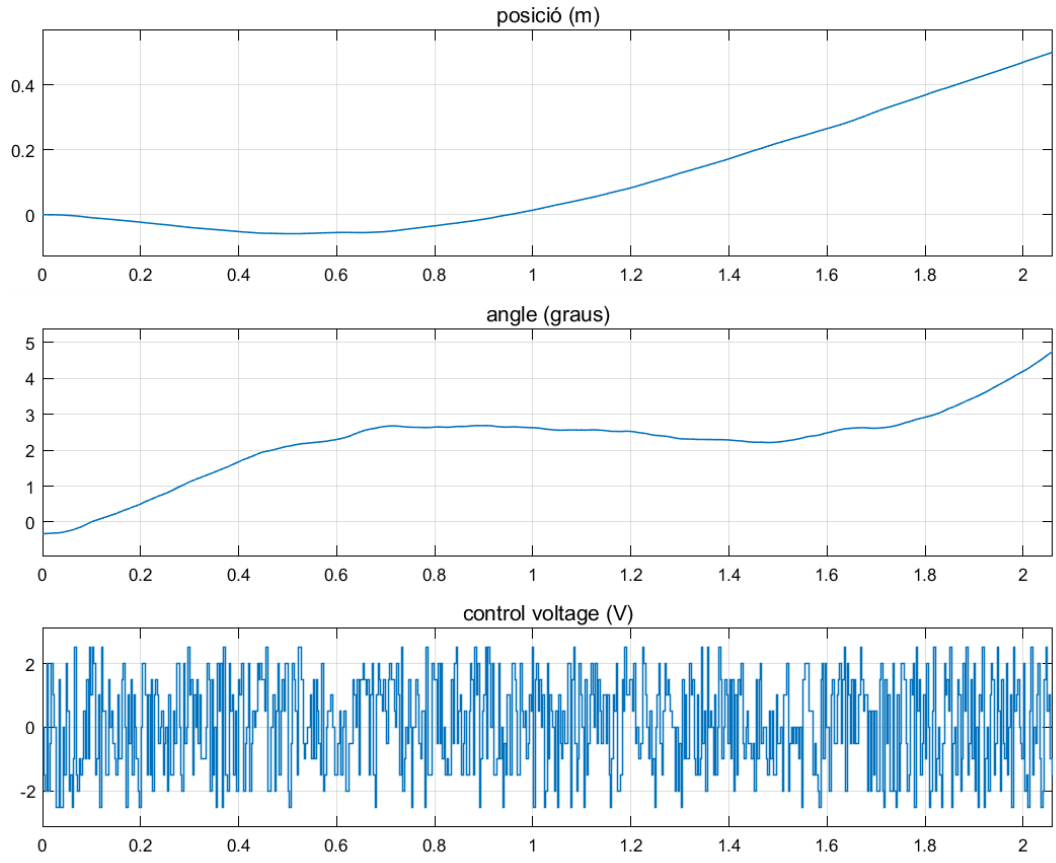


Figura 22: Simulació al inici del aprenentatge

Després de diverses iteracions es veu que el sistema ja ha arribat a l'objectiu desitjat i el controlador ja ha aprés suficient com per mantenir el pèndol 10 segons en la posició invertida.En concret l'aprenentatge s'assoleix en l'episodi

614 de la quarta prova d'aprenentatge (des de l'episodi 604 aguanta 10s de forma consecutiva en posició vertical).

Amb la política derivada de la Q de l'episodi 614, comprovem com el controlador és capaç de mantenir el pèndol en posició invertida per més temps que 10s. En la Figura 23:Simulació del sistema avaluant la matriu Q apresa la simulació s'atura als 60s, però pel pèndul segueix estant en posició invertida.

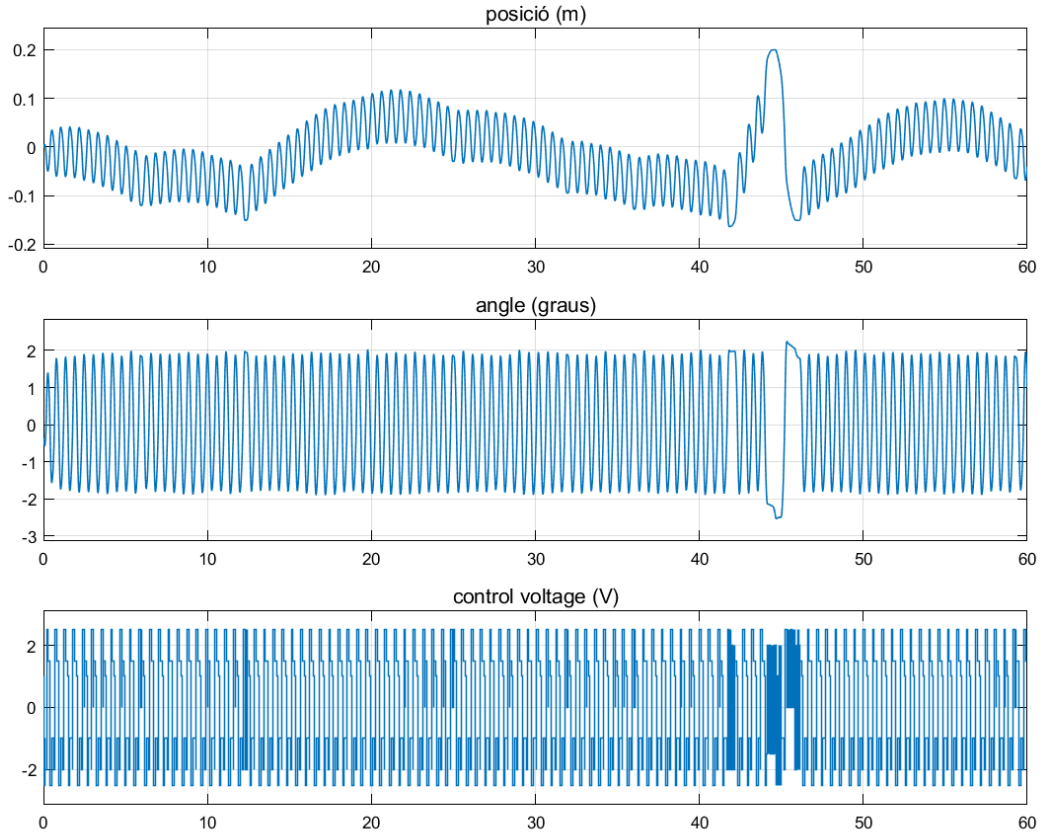


Figura 23:Simulació del sistema avaluant la matriu Q apresa

A més a més, si ens fixem en una regió més petita de temps, observemcom el voltatge de control del pèndol segueix un patró relacionat amb la posició i l'angle del pèndol:

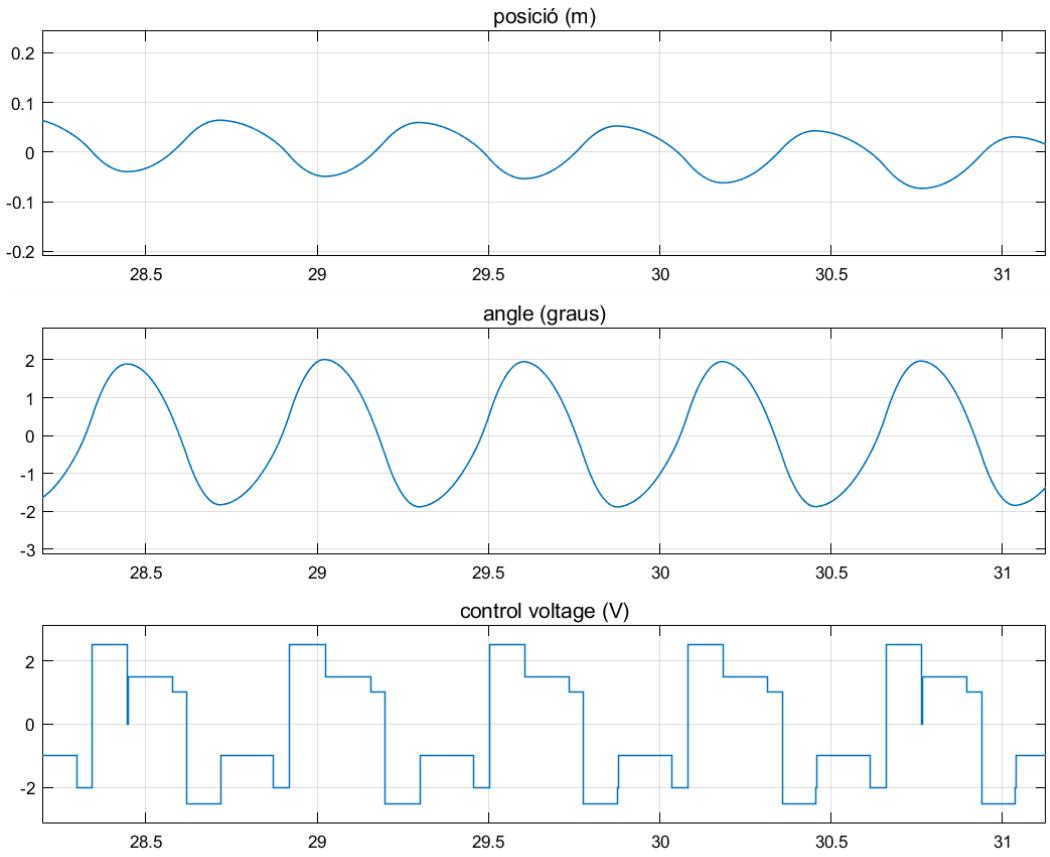


Figura 24: Simulació del sistema avaluant la matriu Q apresa (detall)

Així doncs, s'ha assolit l'aprenentatge i un cop ha après, el controlador no solament és capaç de mantenir el pèndol en posició invertida durant 10s sinó que és capaç de corregir-se i sembla que es podria mantenir invertit de forma indefinida.

5 FUNCIONAMENT DEL CONTROLADOR AMB LA MAQUETA

A continuació, es presenta el funcionament del controlador Q-learning amb el sistema real, introduint el programa amb Simulink que comunica amb la maqueta Feedback 33-005.

5.1 Programa en Simulink

Per la comunicació amb la maqueta s'han utilitzat els blocs de PCI1711 de Feedback per Simulink facilitats en el Laboratori de Control de la UPC.

Com es veu a la Figura 25, les variables d'entrada d'adquisició (Inputs) són la posició del carro i l'angle del pèndul. Com s'ha comentat anteriorment, per poder emprar la funció de Q-learning s'ha utilitzat el bloc de fcn (Matlab function) de Simulink. Les sortides del bloc són la consigna de voltatge i la Q actualitzada (en cas de realitzar aprenentatge).

Ha estat necessari incloure el bloc "Memory" que permet retroalimentar la funció durant l'aprenentatge amb la maqueta (La Q actualitzada en el step *t* passa a ser la Q d'entrada en l'step *t+1*). Al mateix temps, en finalitzar la simulació, la Q de sortida es guarda al Workspace com a variable "Q_maqueta".

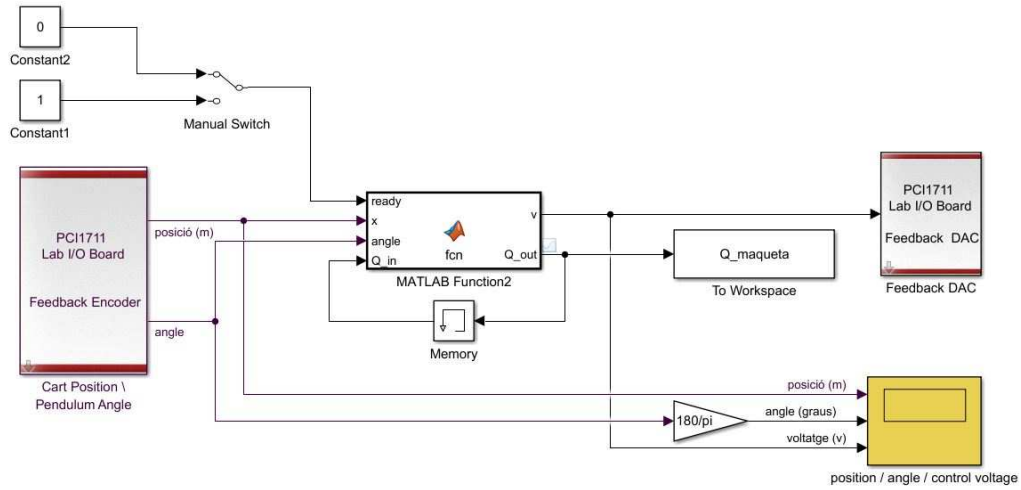


Figura 25: Simulink adaptat a la maqueta

Un dels problemes que s'han resolt utilitzant un "Manual Switch" ha estat la necessitat d'activar el control i l'aprenentatge un cop el sistema es troba en les condicions desitjades de posició del carro i angle del pèndol. Si no s'utilitzava el switch manual, a la maqueta s'hi aplicaven consignes de control mentre encara no havíem situat el pèndol en les condicions inicials.

A part, s'ha inclòs la conversió de radians a graus de l'angle del pèndul per mostrar-ho de forma més intuïtiva en el scope.

5.2 Avaluació de la política obtinguda amb el model

Inicialment, es van fer proves amb la Q obtinguda del model en Simulink. Com es mostra a laFigura 26, la simulació de la maqueta arriba aguantar 4segons.

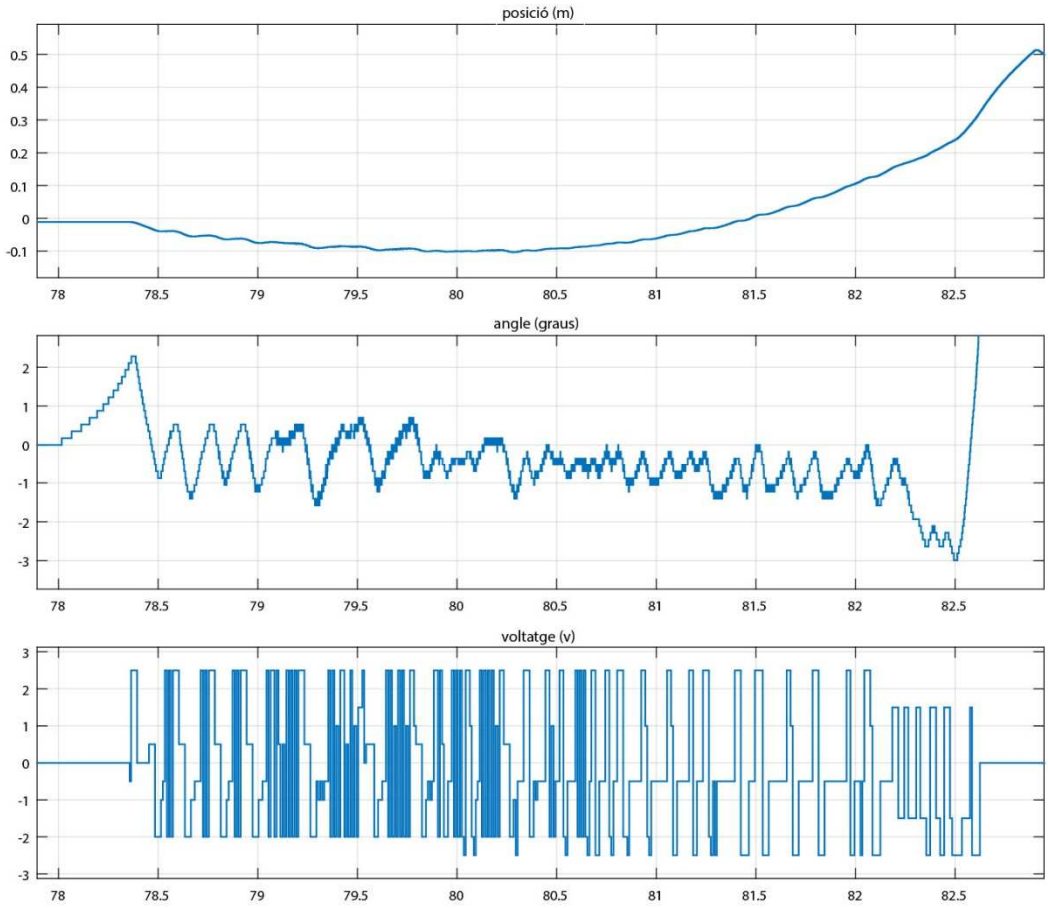


Figura 26: Gràfiques del model amb la Q apresa

Es va decidir realitzar aprenentatge amb la maqueta al comprovar que la política obtinguda amb el model no era capaç de mantenir el pèndol invertit durant llargues estones de temps.

5.3 Resultats de l'aprenentatge amb la maqueta

Els paràmetres de l'aprenentatge són iguals que en l'aprenentatge amb el model a excepció de la política seguida, en aquest cas serà la política òptima,

seleccionant sempre la millor opció. Això permet comprovar les millores del controlador en el mateix aprenentatge, ja que està seguint la política òptima.

A la Figura 27, es pot observar una de les gràfiques extretes durant el període d'aprenentatge on el pèndol arriba a aguantar més de 10 segons invertit.

Amb l'aprenentatge realitzat amb la maqueta, el pèndol es manté sempre molt proper als 0º, i la posició del carro pràcticament centrada en els 0m durant 12s. passat aquest temps, el pèndol es desestabilitza, i cau.

L'execució durant aquests 12s és inclús millor que l'efectuada amb el model, on la posició del carro variava de forma més brusca. Tot hi això, amb el model, en cas de sortir de la zona central, el controlador era capaç de recuperar-se i per això la simulació podia aguantar més temps.

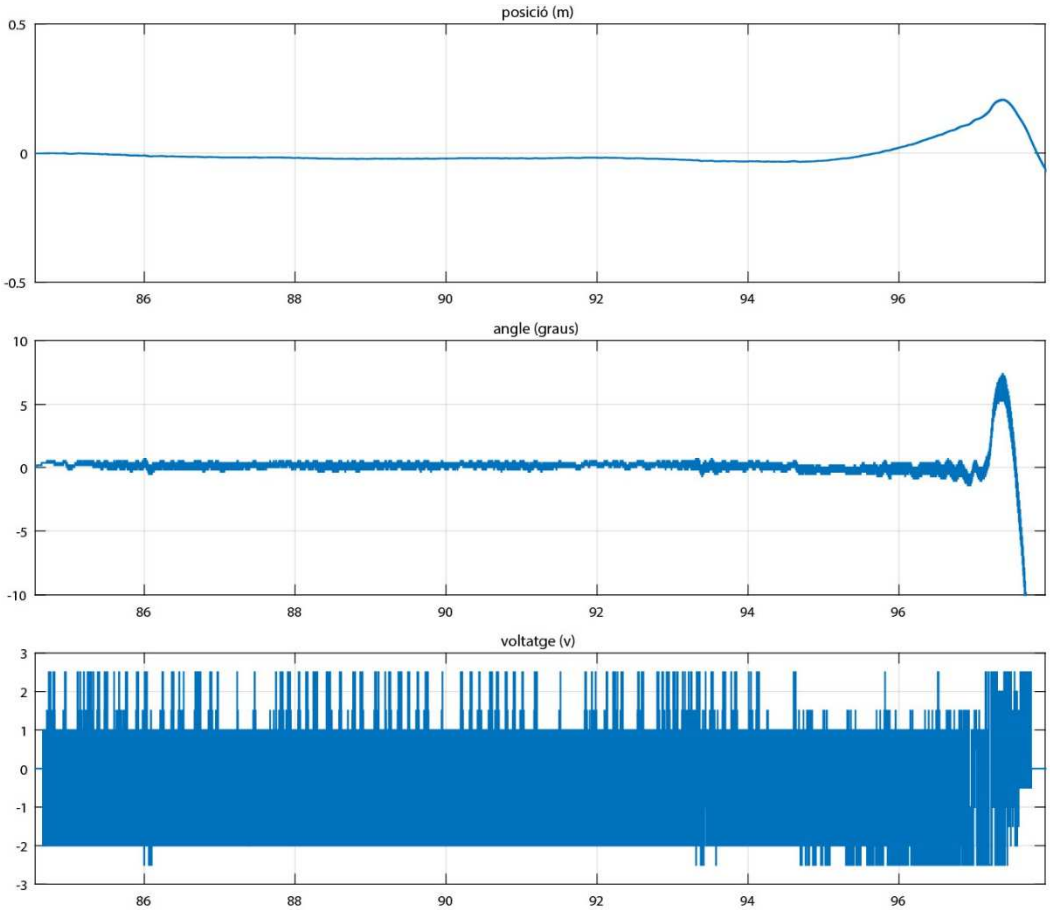


Figura 27:Gràfiques obtingudes amb el controlador i maqueta

A la Figura 28, es pot veure en detall un segon de la simulació representada a la Figura 27. S'observa com l'angle del pèndol varia molt poc en cada moment de la simulació, ja que la acció de control es manté en valors alts molt alternats.

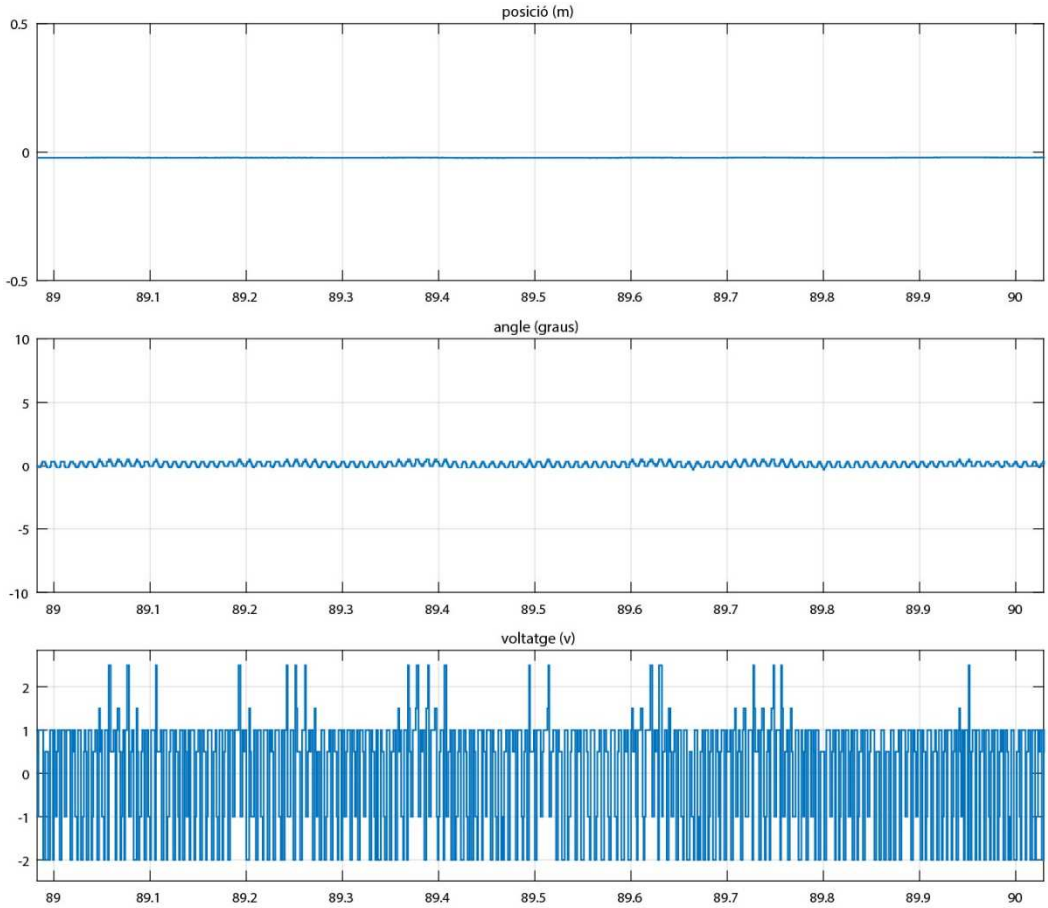


Figura 28: 1 segon de simulació de l'aprenentatge amb la maqueta

6 RESULTATS

A continuació es presenta un breu resum dels resultats de l'aprenentatge amb el model i l'aprenentatge amb la maqueta real.

Pel que fa a l'aprenentatge amb el model, els resultats han estat molt bons, s'ha assolit l'aprenentatge i a més la política apresa(que s'obté directament de la última matriu Q de l'aprenentatge) permet mantenir el pèndol invertit 1 min en repetits casos amb diferents angles inicials (entre -2º i 2º).

Pel que fa a l'aplicació del controlador en el sistema real, podem dir que la política que funcionava amb el model no funciona suficientment bé en el sistema real. Durant els 4s inicials sí es capaç d'aguantar el pèndol en posició invertida, però amb una tendència de desplaçar el carro cap a l'esquerra.

Per tal de millorar l'execució del controlador en el sistema real s'ha realitzat aprenentatge amb la maqueta, a partir de la matriu Q apresa amb la simulació. Després de varis episodis d'aprenentatge (20 aproximadament), les accions preses per el controlador han començat a millorar i finalment s'ha aconseguir que el pèndol arribi a aguantar més de 10 segons invertit.

7 CONCLUSIONS

Com a conclusions s'avaluarà l'assoliment dels objectius. Primerament, podem dir que ens hem familiaritzat amb l'aprenentatge per reforç, concretament amb el Q-learning, i hem après el seu funcionament. Així doncs, hem assolit el primer i bàsic objectiu del treball.

El segon objectiu era adquirir el model de simulació de la maqueta del pèndol invertit del laboratori. Aquest objectiu també s'ha complert, tenint un model amb Simulink que ens ha permès realitzar l'aprenentatge de forma àgil, sense necessitat d'estar presencialment en el laboratori posant el sistema real en condicions inicials en cada etapa.

Pel que fa al tercer objectiu, aquest incloïa la programació i entrenament del controlador Q-learning per tal que aprengués a mantenir el pèndol del model en posició invertida i amb el carro en la zona central del carril. Aquest objectiu també s'ha assolit de forma satisfactòria, ja que, el controlador ha après una política capaç de mantenir el pèndol de la simulació invertit en simulacions de 1min.

Pel que fa a l'últim objectiu, el d'aconseguir un correcte funcionament del controlador amb la maqueta, podem dir que em obtingut resultats satisfactoris. Posant en marxa la maqueta amb la política apresa en la simulació permet aguantar el pèndol en posició invertida (entre -10º i 10º) durant 4-5 segons, però degut a desajustos mecànics o pertorbacions de l'entorn la maqueta no es comporta exactament igual que el model, això comporta que la política apresa no sigui vàlida per la maqueta, fet que fa necessari l'aprenentatge amb ella.

Alrealitzar un aprenentatge a partir de la maqueta, el controlador ha començat a millorar, i finalment el pèndol s'ha mantingut més de 10 segons invertit. Degut que les proves amb la maqueta comporten molt de temps, s'ha decidit donar per vàlid el fet que arribés aguantar durant 13 segons. Però creiem que si s'invertís més temps en realitzar les proves d'aprenentatge amb la maqueta, el controlador seria capaç de mantenir el pèndol invertit durant tot el temps de simulació, a més, seria més tolerant pel que fa a les condicions inicials.

8 BIBLIOGRAFIA

- **Sutton, Richard S.; Barto, Andrew G.. (1998). *Reinforcement Learning: Introduction, An*. MIT Press. <http://www.myilibrary.com?ID=209678>**
  (Disponible en el directori de la UPC)

- **Marco Wiering and Martijn van Otterlo (Eds.)(2012). *Reinforcement Learning State-of-the-Art*. doi: 10.1007/978-3-642-27645-3**
  (Disponible en el directori de la UPC)

- http://www.cs.us.es/~fsancho/?e=109Fernando Sancho Caparrini (Professor) – Aprenentatge per reforç: algoritme Q-learning

- http://mnemstudio.org/path-finding-Q-learning-tutorial.htm John McCullock - Q-learning, step by step tutorial