Reinforcement Learning Compensation based PD Control for a Double Inverted Pendulum

G. Puriel-Gil, W. Yu, and H. Sossa

***Abstract*—In this paper, we present a Control Algorithm based on Reinforcement Learning for a double inverted pendulum on a cart. By implementing the Q-Learning techniques in the PD control scheme, the second pendulum (top pendulum) is enabled to improve its performance. In a first step, Q-Learning is used so that the control can balance the second pendulum towards its inverted vertical position, while the first pendulum has no restrictions on its movement and also the car remains in a range of ± 1 meter in its displacement. In a second step, we combine hybrid techniques of Q-Learning and PD control, in a system that has had changes in its parameters and in its initial conditions. Then, with the hybrid control, we obtain better results than using the controllers individually. Finally, the simulation results show the effectiveness of the proposed controller.**

***Index Terms*—Reinforcement Learning, Q-Learning, Double Inverted Pendulum.**

I. Introducción

Recientemente se ha tenido un interés en el aprendizaje por reforzamiento, el cual se ha convertido en una de las áreas con mayores frutos dentro del aprendizaje para máquinas donde se pueden encontrar muchas aplicaciones en control óptimo y en la robótica. Desde un punto de vista práctico, el objetivo principal es aprender como asignar estados a las acciones, mientras se intenta maximizar una señal de recompensa [1]. Otro reto que se encuentra de manera frecuente en robótica es la generación de buenas recompensas que especifiquen de manera precisa la interacción entre el proceso y el controlador [2]. Por ejemplo, en la práctica se necesitan recompensas que lleven a la planta o proceso rápidamente hacia el objetivo, este objetivo se le conoce como recompensa audaz y representa una gran contribución en el aprendizaje por reforzamiento. El especificar buenas recompensas en el área de la robótica requiere bastante experiencia y siempre es difícil de obtener en la práctica [3]. El término aprendizaje para máquinas se refiere al conjunto de métodos computacionales o algoritmos que son diseñados para que un sistema computacional sea capaz de mejorar su desempeño, usando su experiencia obtenida a lo largo del tiempo. Dependiendo del tipo de experiencia se pueden mencionar tres tipos de aprendizaje para máquinas: aprendizaje por reforzamiento, aprendizaje supervisado y aprendizaje no supervisado.

En el aprendizaje por reforzamiento la idea central es que un agente aprenda a alcanzar un objetivo basado en su experiencia con el medio ambiente [6]. Este problema se resuelve de manera habitual, usando procesos de decisión de Markov (PDM). Sin embargo, en el problema del aprendizaje por reforzamiento, ni las transiciones de probabilidad de los estados o la función de recompensa se conocen, lo que impide usar algoritmos valor de iteración o de política de iteración. Entonces, lo que se puede hacer, es que el agente interactúe con el ambiente de manera repetida y así se obtenga información de las recompensas obtenidas y los estados visitados después de realizar acciones en diferentes estados. Usando los datos recolectados por el agente, hay varios algoritmos que pueden ser utilizados para resolver el problema del aprendizaje por reforzamiento [5]. El aprendizaje por reforzamiento se enfoca en el uso de técnicas clásicas de control óptimo a modelos que aprenden a través de interacción repetida con el ambiente [2] y [4]. Además de la relación que tiene con el control óptimo, también existe una relación cercana con la programación dinámica y la simulación de optimización [7].

Guillermo Puriel-Gil and Wen Yu are with the Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Av. IPN 2508, México City, 07360, México. gpuriel@ctrl.cinvestav.mx. yuw@ctrl.cinvestav.mx. H Sossa is with the Centro de Investigación en Computación del Instinto Politécnico Nacional and with the Tecnológico de Monterrey, Unidad Guadalajara, humbertososa@gmail.com.

II. Control PD con Compensación

El aprendizaje por reforzamiento, desde el punto de vista de la robótica, difiere considerablemente de los casos clásicos del aprendizaje por reforzamiento. Por ejemplo, en el caso de la robótica, un problema es considerado de alta dimensión si tenemos más de 10 acciones y estados continuos [4] y [2].

La dinámica de un manipulador rígido serial de n enlaces puede ser definido como en [9].

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F\dot{q} = \tau \tag{1}$$

aquí $q \in \Re^n$ representa la posición del eslabón, $\dot{q} \in \Re^n$ representa la velocidad del eslabón, $M(q) \in \Re^{n\times n}$ es la matriz de inercia, $C(q,\dot{q}) \in \Re^{n\times n}$ representa la matriz de fuerza centripeta y de coriolis, $G(q) \in \Re^n$ es el vector de gravedad, $F \in \Re^{n\times n}$ es una matriz diagonal que representa los términos de fricción, y finalmente $\tau \in \Re^n$ es el vector de entradas de control. Es bien conocido que el control PD con términos de fricción más la compensación de gravedad puede alcanzar la estabilidad asintótica [8]. El uso de las redes neuronales para compensar los términos no lineales de la dinámica de un robot se pueden encontrar en [10] y [11]. Si G y F se conocen, entonces es posible usar una red neuronal para aproximarlos de la siguiente manera:

$$G + F - f = \widehat{W}_t \sigma(x) + \eta \tag{2}$$

donde η se define como la cota del error modelado, $\eta^T \Lambda_1 \eta \leq \overline{\eta}$, Λ_1 es una matriz tal que $\Lambda_1 = \Lambda_1^T > 0$, $\sigma(x)$ se le conoce como la función de activación, x representa las entradas de la red neuronal.

$$X = \left[q, \dot{q}, q^d, \dot{q}^d\right]^T \tag{3}$$

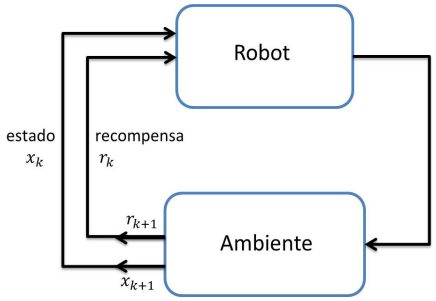


Fig. 1. Modelo del aprendizaje por reforzamiento

f representa las incertidumbres. El Control PD neuronal es:

$$\tau = -K_v r - \widehat{W}_t \sigma(x) \tag{4}$$

así el error auxiliar r se representa de la siguiente manera:

$$\begin{gathered} r = \overline{x}_2 + \Lambda \overline{x}_1 \quad \Lambda = \Lambda^T > 0 \\ \overline{x}_1 = x_1 - x_1^d, \quad \overline{x}_2 = x_2 - x_2^d \\ x_1 = q, x_2 = \dot{q}, x_1^d = q^d, x_2^d = \dot{q}^d. \end{gathered} \tag{5}$$

Se conoce que hay varios problemas que aparecen en el control PD con compensación de redes neuronales, tal como los mínimos locales, la velocidad de convergencia y el tipo de estructura etc. En este artículo se pretende usar un algoritmo de comportamiento humano, como lo es el aprendizaje por reforzamiento [22], para poder resolver estos problemas y poder asegurar un buen desempeño.

El control PD más compensación del aprendizaje por reforzamiento se define de la siguiente manera:

$$\tau = K_p e + K_d \dot{e} + u_r \tag{6}$$

donde $e = q - q^d$, u_r es el torque generado por el aprendizaje por reforzamiento.

III. Aprendizaje por Reforzamiento con Compensación u_r

El problema del aprendizaje por reforzamiento se encuentra en tener una buena interacción con el ambiente (proceso) para así alcanzar las metas y objetivos. El robot o agente conocido como el que toma las decisiones, es el que aprende en base a la prueba y el error para seleccionar una política óptima [7]. En el caso del ambiente o proceso, que comprende todo lo que rodea al robot y que interactúa con él. Finalmente, con esta interacción continua el robot aprende como tomar acciones y al mismo tiempo el ambiente responde a las acciones con recompensas a cada acción tomada y dejando al robot en una nueva situación o estado [19]. La Figura (1) muestra una representación completa del proceso, del ambiente y de cómo interactúan entre ellos. Un proceso de decisión de Markov para estados y acciones finitos se puede ver de la siguiente manera: En el caso discreto $k = 1, 2, \cdots$,el controlador (robot/agente) observa el estado x_k entonces realiza una acción u_k y recibe la recompensa u_k y el siguiente estado x_{k+1}. Entonces el objetivo en el aprendizaje por reforzamiento no solo es maximizar la recompensa inmediata si no también la recompensa a largo plazo la cual es representada de la siguiente manera:

$$r_1 + r_2 + r_3 + \cdots + r_T, \tag{7}$$

Donde T representa el tiempo final, pero la ecuación anterior tiene el problema de que puede crecer de manera infinita, por lo cual es mejor usar una ecuación con descuento la cual de manera conceptual es más compleja pero de manera matemática resulta mucho más simple ya que reduce la contribución de las recompensas mientras el tiempo k se incrementa.,

$$R(x) = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{k+1}, \tag{8}$$

donde γ es un parámetro, $0 \leq \gamma \leq 1$ conocido como factor de descuento.

Consideramos como el ambiente podría responder en el tiempo $k+1$ a la acción tomada en el tiempo k. En el caso causal más general, esta respuesta podría depender de todo lo sucedido antes. En este caso, la dinámica puede ser definida solamente especificando la completa distribución de probabilidad:

$$\begin{gathered} \Pr\{r_{k+1} = r, x_{k+1} = x' \mid \\ x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k\}, \end{gathered} \tag{9}$$

para todo x' , r, y todos los posibles valores de los eventos pasados: $x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k$. Si el estado tiene la propiedad de Markov, entonces la respuesta del ambiente al tiempo $k+1$ dependerá únicamente del estado y la acción tomada al tiempo k, por lo que en este caso la dinámica del ambiente puede ser definida únicamente por:

$$\Pr\{r_{k+1} = r, x_{k+1} = x' \mid x_k, u_k\}, \tag{10}$$

para todo x', r, x_k, y u_k. En otras palabras, el estado tiene la propiedad de Markov, y es un estado de Markov [21], si y sólo si la ecuación (9) es igual a la ecuación (10) para todo x', r y las historias de $x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k$. Un PDM determinístico está definido por el espacio de estados X del proceso, el espacio de acción U del controlador, la función de transición f del proceso (la cual describe cómo cambia el estado como resultado de las acciones de control), y la función de recompensa ρ (la cual evalúa el desempeño del control inmediato). Como un resultado de la acción u_k aplicada en el estado x_k, en el tiempo discreto k, el estado cambia a x_{k+1}, de acuerdo a la función de transición $f : X \times U \rightarrow X$:

$$x_{k+1} = f(x_k, u_k). \tag{11}$$

Al mismo tiempo, el controlador recibe la señal de recompensa escalar r_{k+1}, de acuerdo con la función de recompensa ρ : $X \times U \rightarrow \mathbb{R}$:

$$r_{k+1} = \rho(x_k, u_k), \tag{12}$$

donde se asume que $\| \rho \|_\infty = \sup_{x,u} \mid \rho(x,u) \mid$ es finita. La recompensa evalua el efecto inmediato de la accion u_k, conocida como la transición de x_k a x_{k+1}, pero en general, no dice nada sobre los efectos a largo plazo.

El controlador escoge acciones de acuerdo con su política $h : X \rightarrow U$, usando:

$$u_k = h(x_k). \tag{13}$$

Dados f y ρ, el estado actual x_k y la acción actual u_k son suficientes para determinar tanto el siguiente estado x_{k+1} como la recompensa r_{k+1}. Existen básicamente tres modelos de optimización en los (PDM) que son suficientes para cubrir la mayoría de los enfoques en la literatura.

Horizonte Infinito:

$$R(x_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, h(x_k)), \tag{14}$$

donde $\gamma \in [0, 1)$, es el factor de descuento y $x_{k+1} = f(x_k, h(x_k))$ para $k \geq 0$. Otro tipo de retornos pueden ser defindos como lo es el retorno sin descuento, obtenido al dejar γ con un valor de 1 en la ecuación (14), donde simplemente suma las recompensas sin realizar algún descuento. Desafortunadamente, el retorno horizonte finito sin descuento frecuentmente no está acotado. Una alternativa es usar el retorno *Horizonte Finito Promedio*:

$$\lim_{k \to \infty} \frac{1}{k} \sum_{k=0}^{k} \rho(x_k, h(x_k)), \tag{15}$$

el cual es acotado en muchos casos. Retornos de horizonte finito, pueden ser obtenidos mediante la acumulación de las recompensas a lo largo de las treyectorias finitas de longitud K (el horizonte), en lugar de trayectorias infinitas. De hecho, el retorno de *Horizonte Finito* con descuento puede ser definido como:

$$\sum_{k=0}^{K} \gamma^k \rho(x_k, h(x_k)). \tag{16}$$

El retorno sin descuento ($\gamma = 1$), puede ser usado de manera más fácil en el caso del horizonte finito, que está acotado cuando las recompensas son acotadas. En este trabajo principalmente se utilizará el retorno con descuento de horizonte infinito (14). Una manera conveniente de caracterizar las políticas es por medio de sus funciones valor. Dos tipos de funciones valor existen: funciones de valor estado-acción (funciones Q) y funciones de valor estado (funciones V). La función Q que está definida como $Q^h : X \times U \to \mathbb{R}$ de una política h da como resultado el retorno obtenido cuando se empieza desde un estado dado, aplicando una acción dada, y siguiendo una política h por lo tanto se tiene:

$$Q^h(x, u) = \rho(x, u) + \gamma R^h(f(x, u)), \tag{17}$$

donde $R^h(f(x, u))$ es el retorno del siguiente estado $f(x, u)$. Esta representación de la fórmula puede ser obtenida si se escribe $Q^h(x, u)$ de manera explícita como una suma de las recompensas con descuento obtenido, tomando la acción u en el estado x y siguiendo la política h

$$Q^h(x, u) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k), \tag{18}$$

donde $(x_0, u_0) = (x, u)$, $x_{k+1} = f(x_k, u_k)$, para $k \geq 0$, y $u_k = h(x_k)$, para $k \geq 1$. Entonces se puede separar el primer termino de la sumatoria:

$$\begin{aligned}
Q^h(x, u) &= \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k) \\
Q^h(x, u) &= \gamma^0 r_1 + \gamma^1 r_2 + \gamma^2 r_3 + \cdots \\
Q^h(x, u) &= r_1 + \gamma^1 r_2 + \gamma^2 r_3 + \cdots \\
Q^h(x, u) &= r_1 + \sum_{k=1}^{\infty} \gamma^k r_{k+1} \\
Q^h(x, u) &= \rho(x, u) + \sum_{k=1}^{\infty} \gamma^k \rho(x_k, u_k) \\
Q^h(x, u) &= \rho(x, u) + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} \rho(x_k, h(x_k)) \\
Q^h(x, u) &= \rho(x, u) + \gamma R^h(x_{k+1}) \\
Q^h(x, u) &= \rho(x, u) + \gamma R^h(f(x, u)).
\end{aligned} \tag{19}$$

La función óptima Q se define como la mejor función Q que puede ser obtenida por cualquier política de la siguiente manera:

$$Q^*(x, u) = \max_h Q^h(x, u) \tag{20}$$

Cualquier política h^* que selecione en cada estado una acción que genere el valor más grande de la funcion óptima Q :

$$h^*(x) \in \arg\max_u Q^*(x, u), \tag{21}$$

es óptima (nos dice que maximiza el retorno). En general, para una función Q dada, tener una política h que satisface

$$h(x) \in \arg\max_u Q(x, u), \tag{22}$$

se le conoce como una acción ambiciosa en Q. Entonces para encontrar una política óptima basta con encontrar una Q^* y después aplicar la ecuación (21) para calcular una política en Q^*. La ecuación de Bellman para Q^h dice que el valor de tomar una acción u en el estado x bajo la política h es igual a la suma de las recompensas inmediatas y el valor descontado alcanzado por h en el siguiente estado:

$$Q^h(x, u) = \rho(x, u) + \gamma Q^h(f(x, u), h(f(x, u))) \tag{23}$$

Esta ecuación de Bellman puede ser obtenida de la ecuación (17) que sigue a continuación:

$$\begin{aligned}
Q^h(x, u) &= r_1 + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \\
Q^h(x, u) &= r_1 + \gamma \left[r_2 + \gamma \sum_{k=2}^{\infty} \gamma^{k-2} r_{k+1} \right] \\
Q^h(x, u) &= \rho(x, u) + \\
&\gamma \left[\rho(f(x, u), h(f(x, u))) + \gamma \sum_{k=2}^{\infty} \gamma^{k-2} \rho(x_k, h(x_k)) \right] \\
Q^h(x, u) &= \rho(x, u) + \gamma Q^h(f(x, u), h(f(x, u))),
\end{aligned} \tag{24}$$

donde $(x_0, u_0) = (x, u)$, $x_{k+1} = f(x_k, u_k)$ para $k \geq 0$, y $u_k = h(x_k)$ para $k \geq 1$.

La ecuación de Bellman que representa a Q^*, donde establece que el valor óptimo de la acción u tomada en el estado x es igual a la suma de las recompensas inmediatas y al valor óptimo con descuento obtenido por la mejor acción en el siguiente estado:

$$Q^*(x,u) = \rho(x,u) + \gamma \max_{u'} Q^*(f(x,u),u')), \tag{25}$$

La función $V^h : X \rightarrow \mathbb{R}$ de una política h es el retorno obtenido empezando desde un estado particular y siguiendo la política h :

$$V^h(x) = R^h(x) = Q^h(x,h(x)). \tag{26}$$

La función óptima V se obtiene como la mejor función V que puede ser obtenida por cualquier política, y puede ser calculada desde la función óptima Q:

$$V^*(x) = \max_h V^h(x) = \max_u Q^*(x,u), \tag{27}$$

Y finalmente una polítca óptima h^* puede ser calculada desde V^*, usando el hecho de que satisface:

$$h^*(x) \in \arg\max_u \left[\rho(x,u) + \gamma V^*(f(x,u))\right]. \tag{28}$$

Al tomar esta fórmula es más difícil que usar la ecuación (21); en particular, un modelo de PDM se necesita en la forma de la dinámica f y la recompensa ρ. Las funciones V V^h y V^* satisface las siguientes ecuaciones de Bellman, que son similares a las ecuaciones (23) y (25):

$$V^h(x) = \rho(x,h(x)) + \gamma V^h(f(x,h(x))), \tag{29}$$

$$V^*(x) \max_u \left[\rho(x,u) + \gamma V^*(f(x,u))\right]. \tag{30}$$

Se utilizará el método de diferencias temporales (**DT**) para estimar la función Q en linea. Por lo tanto la ecuación del aprendizaje Q está dada de la siguiente forma:

$$\begin{array}{l} Q^{(k+1)}(x_k,u_k) = Q^{(k)}(x_k,u_k) + \\ +\alpha \left[\begin{array}{c} R^{(k)} + \gamma \max_{u_{k+1}} Q^{(k)}(x_{k+1},u_{k+1}) - \\ -Q^{(k)}(x_k,u_k) \end{array} \right], \end{array} \tag{31}$$

donde x_k, u_k son el estado y la acción en el timepo k y $R^{(k)}$ la recompensa en el paso de tiempo k, $R^{(k+1)} = r(x_k,u_k)$. Entonces α es la tasa del aprendizaje entre $(0 < \alpha \leq 1)$, γ es el factor de descuento entre $(0 \leq \gamma \leq 1)$.

$$\pi = \beta \arg\max_{u_k} \left[Q^{(k+1)}(x_k,u_k)\right], \tag{32}$$

β es una constante ($\beta > 0$).

Una de las condiciones de convergencia del algoritmo para que Q sea Q^* es que todos los estados sean visitados un número infinito de veces y que además α debe decaer de manera adecuada [18].

IV. Aplicación

El péndulo doble invertido sobre el carro Figura (2) plantea un problema de control desafiante. Además es una de las herramientas más atractivas para probar leyes de control lineal y no lineal.

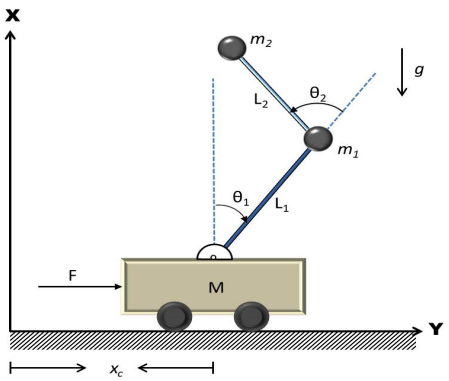


Fig. 2. Doble péndulo invertido sobre un móvil

1) Descripción y planteamiento del problema: El doble péndulo invertido en el carro, es una extensión del péndulo invertido en el carro. El objetivo es estabilizar el segundo péndulo (péndulo superior) sobre la vertical, mientras se mantiene una posición deseada sobre el desplazamiento del carro. La dificultad para controlar el doble péndulo invertido se encuentra en que la acción de la fuerza solo será aplicada en el carro, mientras que los péndulos no están actuados, además del hecho de que es un sistema caótico por naturaleza. El mecanismo está formado por tres cuerpos rígidos, como se muestra en la Figura (2) un carro de masa M, acoplado a través de una articulación de rotación, a una barra con masa m_1, y longitud l_1. A su vez la primera barra esta acoplada, en el otro extremo y también a través de una articulación de rotación, a una segunda barra de masa m_2 y longitud l_2.

Planteamiento del problema: Se desea que el segundo péndulo se mantenga en la posición vertical invertida sin restricción del primer pendulo, mientras el carro mantiene una restricción impuesta de encontrarse en un rango de ±1 metro desde su punto de origen referencial ($x_{c\_inicial} = 0$). Las condiciones iniciales son que el primer péndulo (péndulo inferior) empiece en su equilibrio inestable (origen) y con velocidad cero, mientras que el segundo péndulo (péndulo superior) podrá tener una condición inicial de ±5°, donde θ_1 y $\dot{\theta}_1$ representan la posición y velocidad del primer péndulo, θ_2 y $\dot{\theta}_2$, representa la posición y velocidad del segundo péndulo y finalmente x_c y $\dot{x}_c$ son la posición y velocidad del carro.

2) Diseño del control: El método del aprendizaje por reforzamiento como lo es Q-Learning (QL) no necesita de la dinámica del sistema. Los estados del algoritmo Q-Learning estarán definidos como una suma cuadrática del error de posición y de la velocidad, más una combinación de la suma de las velocidades del primer y segundo péndulo

$$e_x = e^2_{posicion} + e^2_{velocidad}, = f(\theta_1,\dot{\theta}_1,\theta_2,\dot{\theta}_2,x_c,\dot{x}_c,x^d,\dot{x}^d), \tag{33}$$

donde

$$\begin{array}{l} e_{posicion} = x^d - (x_c + \sin(\theta_1+\theta_2)) \\ e_{velocidad} = \dot{x}^d - (\dot{x}_c + (\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2)) \end{array} \tag{34}$$

$$\begin{array}{l} x^d = [-\pi,\pi] \\ \dot{x}^d = [-\pi,\pi] \end{array}, \tag{35}$$

y las acciones están definidas como la fuerza aplicada al móvil

$$u = \{-1, 1\}, \tag{36}$$

entonces se tiene finalmente un algoritmo de aprendizaje incremental que se realiza en línea, donde su versión en línea está dado por:

$$Q^{(k+1)}(e_{x_i}, u_j) = Q^{(k)}(e_{x_i}, u_j) + \\ +\alpha \left[r_{k+1} + \gamma \max_{u \in U} Q^{(k)}(e_{x_{inew}}, u_{j_{new}}) - Q^{(k)}(e_{x_i}, u_j) \right], \tag{37}$$

la recompensa estará definida para el péndulo superior

$$r_{1_{k+1}} = -|x_c + \sin(\theta_1 + \theta_2)|^2 + \\ -0.25 \left| \dot{x}_c + (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2) \right|^2, \tag{38}$$

y nuestro objetivo es encontrar una política f (ley de control) que maximice el retorno esperado.

$$f = \arg\max_u \left[Q^{(k+1)}(e_{x_i}, u_j) \right]. \tag{39}$$

3) Resultados del aprendizaje: Para mostrar la efectividad del desempeño del controlador se realizarán las simulaciones bajo la plataforma Matlab. Los parámetros utilizados tanto en el doble péndulo invertido como en el aprendizaje por reforzamiento se muestran en la Tabla 1.

TABLA I
PARÁMETROS DEL DOBLE PÉNDULO Y DEL CONTROLADOR QL

Parámetros	Descripción	Valor
m	Masa del péndulo 1	0.3 kg
m	Masa del péndulo 2	0.2kg
M	Masa del carro	0.8kg
l	Longitud del péndulo 1	0.5 m
l	Longitud del péndulo 2	0.4m
g	Gravedad	9.8 m/s^2
α	Tasa de aprendizaje	0.99
γ	Factor de descuento	0.9
f	Acciones de entrada	$\{-100, 100\}N$

La Tabla 2 muestra los parámetros tanto de la planta como del controlador al agregar el control PD a nuestra matriz Q previamente calculada con los parámetros iniciales de la Tabla 1.

TABLA II
PARÁMETROS DEL PÉNDULO Y EL CONTROLADOR QL+PD

Parámetros	Descripción	Valor
m	Masa del péndulo 1	0.5 kg
m	Masa del péndulo 2	0.5kg
M	Masa del carro	1.2kg
l	Longitud del péndulo 1	0.5 m
l	Longitud del péndulo 2	0.5m
g	Gravedad	9.8 m/s^2
α	Tasa de aprendizaje	0.99
γ	Factor de descuento	0.9
K_p	Ganancia Proporcional	10
K_d	Ganancia Derivativa	155
β	Ganancia control QL	100
Γ	Conjunto de acciones	$\{-1, 1\}N$

El control híbrido final está definido de la siguiente manera:

$$f = K_p e_{posicion} + K_d e_{velocidad} + \beta u_r. \tag{40}$$

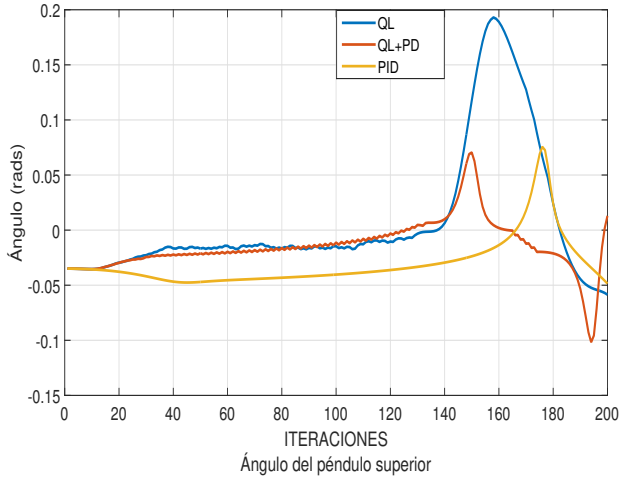


Fig. 3. Posición del péndulo superior

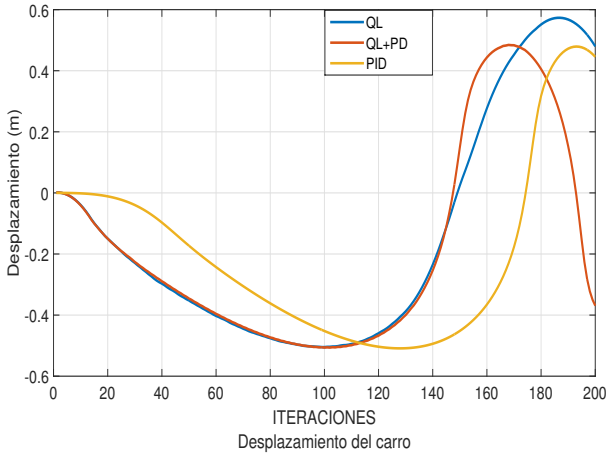


Fig. 4. Desplazamiento del carro

Las ganancias del controlador PD fueron ajustadas de tal manera que no causara una inestabilidad en el péndulo superior, y así que el control PD sólo proporcionará la energía suficiente para reducir el error de posición y el del velocidad evitando así la deriva de los péndulos y del carro mismo. Finalmente, en el control Q-Learning sólo una ganacia β fue incrementada a la entrada, βu_r, mientras se mantiene la misma matriz $Q^{(k)}(e_{s_i}, u_{r_j})$.

Las Figuras (3) y (4), muestran el desempeño del algoritmo propuesto. Para el cálculo de las ganancias del controlador PID se realizaron modificaciones sucesivas en los parámetros de control hasta conseguir los valores más óptimos en función de la comparación entre los controladores QL y QL+PD, resultando en: $kp = 9, kd = 160, ki = 6$, donde se pudo observar, que el mejor desempeño fue del controlador QL+PD, demostrando que mantiene el péndulo superior sobre la vertical con un ángulo de desplazamiento mucho más pequeño en comparación con los otros controladores. La Figura (4) muestra como la restricción del desplazamiento del carro se mantiene en el rango de ±1 a partir de su punto de origen

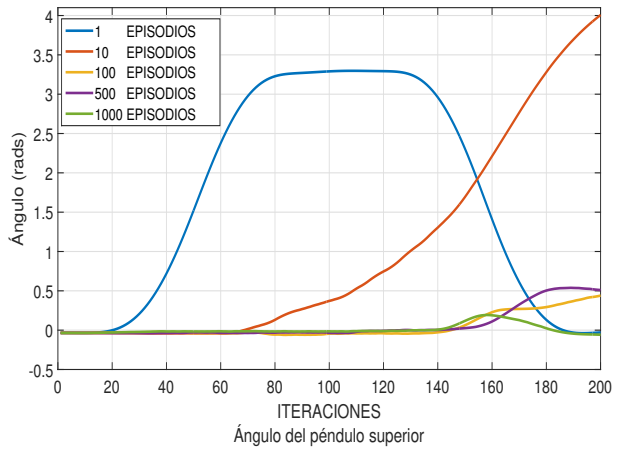


Fig. 5. Posición del péndulo superior en los episodios 1, 10, 100, 500 y 1000

referencial, arrojando la respuesta esperada. La Figura (5) muestra la evolución del aprendizaje del ángulo de salida del péndulo superior θ_2, para el episodios 1, el episodio 10, el episodio 100, el episodio 500 y finalmente el episodio 1000, esto con la intención de revelar como va mejorando el aprendizaje a lo largo del tiempo.

Índices de desempeño: Con la finalidad de cuantificar el comportamiento de los controladores, utilizamos los criterios integrales conocidos como Integral del error absoluto *(IAE)* e Integral del tiempo por el error absoluto *(ITAE)*:

$$IAE = \int_0^\infty \mid e(t) \mid dt \tag{41}$$

$$ITAE = \int_0^\infty t \mid e(t) \mid dt \tag{42}$$

donde el error está dado por $e(t) = \theta_d - \theta_2$

Tabla 3. Índices de desempeño

	IAE	ITAE
QL	7.688	0.0231
PID	7.136	0.0214
QL+PD	4.241	0.0127

Se realizó la comparación entre el aprendizaje por reforzamiento QL, el control QL+PD y el control PID, donde la Tabla 3 muestra que los valores más pequeños en los índices de desempeño son para el controlador QL+PD, lo cual nos dice que el ángulo de salida del péndulo superior se mantiene sobre la referencia (vertical invertida) por más tiempo en comparación con los otros controladores, además en este controlador QL+PD propuesto se aprecian ventajas como por ejemplo: la respuesta híbrida [20] trabaja mucho mejor que un controlador PID, así mismo, la sintonización del control PD resulta mucho más simple ya que el aprendizaje por reforzamiento absorbe toda la dinámica del sistema, dejando que el control PD sólo se encargue de mantener el error del ángulo del péndulo superior en cero. La desventaja que se presenta es que el tiempo que tarda en aprender el algoritmo de aprendizaje por reforzamiento es de aproximadamente 1hr con 30 minutos lo que representa 1000 episodios de 200 Iteraciones por episodio.

V. Conclusión

Se observa que controlar una planta tal como el doble péndulo invertido sobre el carro resulta una tarea más complicada para el algoritmo Q-Learning por que el diseño de la recompensa tiene que ser más ingenioso y deber estar en función de la proyección que el péndulo superior tiene sobre el eje de las abscisas. Además, la forma de discretizar el espacio de trabajo no sólo está en función de los dos ángulos, sino también en función de la velocidad del carro. Finalmente, las graficas muestran el desempeño del ángulo θ_2 y del desplazamiento del carro x, para el contolador QL, el control híbrido QL+PD, y para el control PID, donde se aprecia que el algoritmo de aprendizaje por reforzamiento más el control PD trabajan muy bien de manera cooperativa dando mejores resultados en su forma híbrida, que de manera individual.

Agradecimientos

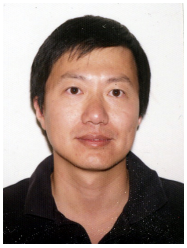
Los autores agradecen el apoyo por parte del CINVESTAV-IPN y el IPN para llevar acabo esta investigación. Así mismo, H Sossa agradece el apoyo económico por parte del IPN y del CONACYT bajo los otorgamientos SIP-IPN 20190007 y CONACYT 65 (Frontiers of Science). G Puriel agradece al CONACYT por la beca otorgada para la realización de sus estudios de doctorado.

Referencias

[1] R. S. Sutton and A. G. Barto, “Reinforcement learning: an introduction”, The MIT Press, March 1998. ISBN 0262193981
[2] Deisenroth, M. Peter, G. Neumann, and J. Peters, “A survey on policy search for robotics”, Foundations and Trends in Robotics vol. 2, pp. 1-142, 2013.
[3] A. S. Polydoros and L. Nalpantidis, “Survey of Model-Based Reinforcement Learning: Applications on Robotics,” Journal of Intelligent & Robotic Systems, vol. 86, pp. 153-173, 2017.
[4] J. J. Kober, A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey”, The International Journal of Robotics Research vol. 32, pp. 1238-1274, 2013.
[5] L. P. Kaelbling , M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey.” Journal of artificial intelligence research vol. 4 , pp. 237-285, 1996.
[6] T. Moerland, J. Broekens, and C. M. Jonker, “Emotion in Reinforcement Learning Agents and Robots: A Survey.” arXiv preprint arXiv:1705.05172 , 2017 .
[7] M. Ghavamzadeh, S. Mannor, J. Pineau and A. Tamar. “Bayesian reinforcement learning: A survey”, Foundations and Trends in Machine Learning, vol. 8, No. 5-6, pp.359-483, 2015.
[8] R. Kelly and V. Santibáñez, “Control de Movimiento de Robots Manipuladores”, Pearson Prentice Hall, 2003.
[9] M.W. Spong and M. Vidyasagar, “Robot Dynamics and Control,” John Wiley & Sons Inc.,Canada, 1989.
[10] F.L. Lewis, A. Yesildirek and K.Liu, “Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance,” IEEE Trans. on Neural Networks, vol.7, No.2, pp. 388-399, 1996.
[11] F.L. Lewis, “Neural Network Control of Robot Manipulators,” IEEE Expert, vol.11, No.2, pp. 64-75, 1996.
[12] Y. Zheng, S. Luo, and Z. Lv, “Control double inverted pendulum by reinforcement learning with double cmac network”, Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 4. IEEE, 2006.

[13] S. Hosokawa and K. Nakano, “A reward allocation method for reinforcement learning in stabilizing control of T-inverted pendulum” , Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference on. IEEE, 2012.
[14] S. Hosokawa, J. Kato and K. Nakano, “A reward allocation method for reinforcement learning in stabilizing control tasks” , Artificial Life and Robotics, vol. 19, No.2, pp. 109-114, 2014.
[15] S. Mahadevan, “Average reward reinforcement learning: Foundations, algorithms, and empirical results,” Machine learning, vol. 22, No.1 , pp.159-195, 1996.
[16] Y. Zheng, S. w. Luo and Z. Lv, “Active exploration planning in reinforcement learning for Inverted Pendulum system control,” Machine Learning and Cybernetics, 2006 International Conference on. IEEE, 2006.
[17] W. Linglin, L. Yongxin and Z. Xiaoke, “Design of reinforce learning control algorithm and verified in inverted pendulum.” Control Conference (CCC), 2015 34th Chinese. IEEE, 2015.
[18] M. Hehn, R. D. Andrea, “A Flying inverted pendulum,” Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
[19] R. Figueroa, A. Faust, P. Cruz , L. Tapia and R. Fierro, “Reinforcement learning for balancing a Flying inverted pendulum,” Intelligent Control and Automation (WCICA), 2014 11th World Congress on. IEEE, 2014.
[20] S. Bongain, M. Jamett ”Electrohydraulic Active Suspension Fuzzy-Neural Based Control System” IEEE Latin America Transaction, vol. 17, no. 9, sep. 2018.
[21] L. Carvalho, J. M. Palma, A. P. C. Goncalves, C. Duran “Vehicles Following problem: A Control Approach for Uncertain Systems with Lossy Networks” IEEE Latin America Transactions, vol. 17, no. 9, sep. 2018.
[22] Z. Huang, J. Liu, Z. Li and C. Y. Su, “Adaptive impedance control of robotic exoskeletons using reinforcement learning.” Advanced Robotics and Mechatronics (ICARM), International Conference on. IEEE, 2016.
[23] Kiumarsi, Bahare, et al. ”Optimal and autonomous control using reinforcement learning: A survey.” IEEE transactions on neural networks and learning systems 29.6 (2018): 2042-2062.
[24] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.
[25] Goyal, Parul, Hasmat Malik, and Rajneesh Sharma. ”Application of Evolutionary Reinforcement Learning (ERL) Approach in Control Domain: A Review.” Smart Innovations in Communication and Computational Sciences. Springer, Singapore, 2019. 273-288.

Guillermo Puriel Gil Recibió el grado de Ingeniero Electrónico del Instituto Tecnológico de Veracruz (ITV) Veracruz, México en el 2006, y el grado de Maestro en Ciencias en Ingeniería Eléctrica con especialización en Mecatrónica del Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN) Ciudad de México, México 2011. Actualmente estudia el Doctorado en Ciencias en el departamento de Control Automático en el Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN). Sus áreas de interés son la robótica móvil, el control de robots y el aprendizaje por reforzamiento.



Wen Yu Recibió el grado de Licenciatura en control automático de la universidad Tsinghua, Beijing, China en 1990 y los grados de Maestro en Ciencias y Doctorado, ambos en Ingeniería Eléctrica de la Northeastern University, Shenyang, China, en 1992 y 1995, respectivamente.

De 1995 a 1996, él fue profesor en el Departamento de Control Automático de la Northeastern University, Shenyang, China. Desde 1996, ha estado en el Centro de Investigación y Estudios Avanzados, Instituto Politécnico Nacional (CINVESTAV-IPN), Ciudad de México, México, donde actualmente es profesor en el Departamento de Control Automático. De 2002 a 2003, ocupó cargos de investigación en el Instituto Mexicano del Petróleo. Fue investigador visitante senior en Queen’s University Belfast, Belfast, Reino Unido, de 2006 a 2007, y profesor asociado de la Universidad de California, Santa Cruz, de 2009 a 2010. También es profesor visitante en la Northeastern University en China desde 2006.

El Dr. Yu es editor asociado del Journal of Intelligent and Fuzzy Systems y miembro de la Academia Mexicana de Ciencias.

Humberto Sossa es Ingeniero en Comunicaciones y Electrónica por la Universidad de Guadalajara en 1981. Obtuvo los grados de Maestro en Ciencias con especialidad en Ingeniería Eléctrica en el Centro de Investigación y de Estudios Avanzados del IPN en 1987 y de Doctor del Instituto Politécnico de Grenoble, Francia en 1992.

Es Profesor de tiempo completo del Instituto Politécnico Nacional desde 1987. Es el Jefe del Laboratorio de Robótica y Mecatrónica del Centro de Investigación en Computación del IPN. Es autor de tres libros y co-editor de 20 libros. Es autor y co-autor de más de 450 artículos en revistas, capítulos de libros, congresos nacionales e internacionales y reportes técnicos. Es autor de 8 patentes. Sus intereses de investigación son en Inteligencia Artificial y Aprendizaje para Máquinas y sus aplicaciones en el control de robots y el manejo de información para la toma de decisiones.

El Dr. Sossa es miembro del Sistema Nacional de Investigadores nivel 3 y de la Academia Mexicana de Ciencias. Es miembro de la ACM y de la Academia de Ingeniería (AI).