Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Control Automático

Human in the loop usando Aprendizaje por Reforzamiento

Tesis que presenta

Carlos Armando Castillo Díaz

para obtener el Grado de

Maestro en Control Automático

Director de Tesis

Dr. Wen Yu Liu

Ciudad de México **Agosto de 2022**

Prefacio

En esta tesis se expondrá

Dedicatoria

Índice general

Índice de figuras

Índice de tablas

Símbolos

Sets

$\mathcal{S}$ Estados

Numbers and Arrays

x position
v velocity

Sets

a acceleration
t time
F force

Capítulo 1

Introducción

1.1. General

El aprendizaje por refuerzo (RL, por sus siglas en inglés) es una disciplina importante en la intersección del aprendizaje automático y el control [1]. El término parece haber sido acuñado por Minsky [2], e independientemente en la teoría del control por Waltz y Fu [3]. La primera investigación de aprendizaje automático que ahora se considera directamente relevante fue el jugador de damas de Samuel [4], que utilizó el aprendizaje de diferencia temporal para administrar la recompensa retrasada de la misma manera que se usa hoy. Por supuesto, el aprendizaje y el refuerzo se han estudiado en psicología durante casi un siglo, y ese trabajo ha tenido un impacto muy fuerte en el trabajo de IA/ingeniería. De hecho, uno podría considerar que todo el aprendizaje por refuerzo es simplemente la ingeniería inversa de ciertos procesos de aprendizaje psicológico y en la actualidad es usado mucho para la inteligencia artificial generalizada, los robots autónomos y los automóviles autónomos. En el aprendizaje por refuerzo, una política de control se refina con el tiempo, y se logra un mejor desempeño a través de la experiencia. El marco más común para RL es el proceso de decisión de Markov, donde la dinámica del sistema y la política de control se describen en un entorno probabilístico, de modo que la estocasticidad se integra en la dinámica del estado y la estrategia de actuación. De esta manera, las políticas de control son probabilísticas, promoviendo un equilibrio entre optimización y exploración.

El aprendizaje por refuerzo puede verse como parcialmente supervisado, ya que no siempre se sabe de inmediato si una acción de control fue efectiva o no. En RL, un agente promulga una política de control, y este agente solo puede recibir información parcial sobre la efectividad de su estrategia de control. Un desafío importante que aborda RL es el desarrollo de una función de valor, también conocida como función de calidad Q, que describe el valor o la calidad de estar en un estado particular y tomar una decisión de política de control particular. Con el tiempo, el agente aprende y refina esta función

Q, mejorando su capacidad para tomar buenas decisiones. En el ejemplo del ajedrez, un jugador experto comienza a tener intuición para una buena estrategia basada en la posición del tablero, que es una función de valor compleja sobre un espacio de estado de dimensiones extremadamente altas (es decir, el espacio de todas las configuraciones posibles del tablero). Q-learning es una estrategia de aprendizaje por refuerzo sin modelo, donde la función de valor se aprende de la experiencia. Recientemente, el aprendizaje profundo se ha aprovechado para mejorar drásticamente el proceso de Q-learning en situaciones donde los datos están fácilmente disponibles [5–8].

1.2. Motivación

La integración del control humano con controladores automáticos o tareas automatizadas es un área de investigación abierta. Por lo tanto, proponemos utilizar un conjunto de herramientas teóricas de control (como en [9]) para componer comandos de colocación de pies delanteros controlados por humanos con un controlador automático que evita la colocación inestable de pies para un robot de rescate cuadrúpedo.

Los investigadores en el campo HITLC se refieren a la interacción de control propuesta entre humanos y controladores como interacción de iniciativa mixta o humano en el circuito (p. ej., [10],[11]). También puede denominarse control compartido, como en [12], ya que ambos controladores (humano y controlador) actúan sobre el mismo sistema dinámico. Uno de los puntos fuertes del control presentado en [9] son los diferentes niveles de autonomía que resultan de la interacción del control humano y automático.

1.3. Abstract

Resultados 3 hojas

Capítulo 2

Aprendizaje por Reforzamiento

En el problema general del control de aprendizaje, el sistema de aprendizaje desempeña el papel de un controlador que selecciona acciones, y, de un conjunto de acciones posibles, y, para que sirvan como entradas de control para un proceso, como se muestra en la Figura 2.1. La salida del proceso, z, es el estado del proceso o, de manera más realista, una observación del estado del proceso obtenida mediante un conjunto de sensores. Guiado por la información de entrenamiento que se le proporciona, el controlador tiene que aprender a generar acciones de control apropiadas para realizar la tarea especificada por su entrada, x. Debido a la ausencia de rutas de retroalimentación en la Figura 2.1 y otras figuras en esta disertación, los controladores pueden parecer restringidos a lo que los teóricos del control llaman controladores de lazo abierto o de avance. Además de la especificación de la tarea, la entrada, x, al controlador también puede incluir retroalimentación del proceso actual y anterior.

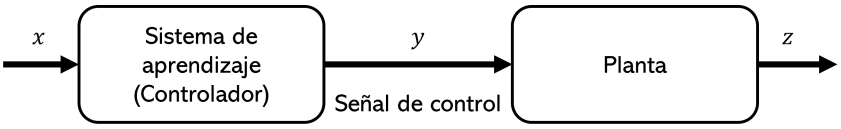

Figura 2.1: La configuración básica del problema para el control del aprendizaje.

El controlador debe aprender a controlar el proceso a través de la generación de señales de control apropiadas. Tiene que hacer esto utilizando la información que se le proporciona a través de la especificación de la tarea, la posible retroalimentación de los resultados del proceso actual y pasado y las acciones de control pasadas, y la información de capacitación. salidas, así como acciones de control previas, lo que permite un control de circuito cerrado.

El controlador implementa una función de control $F_W : X \to Y$, con el subíndice W que denota los parámetros del controlador que determinan qué función se calcula. W denota el búfer de memoria usado en el aprendizaje de memoria, las reglas en un sistema

de aprendizaje basado en reglas, el árbol de decisión de un controlador que usa métodos de árbol de decisión como ID3 [13].Aprender el comportamiento de control apropiado implica determinar W para que la función de control resultante, F_W.

2.1. Métodos de aprendizaje

Los métodos de aprendizaje utilizados para las tres clases de tareas de control del aprendizaje delineadas anteriormente reflejan diferencias en el contenido de la información de entrenamiento. Consideremos primero los problemas de control definidos como tareas de aprendizaje supervisado. Debido a que las acciones deseadas o los gradientes de error de acción están disponibles en las tareas de aprendizaje supervisado, ajustar los parámetros del controlador, W, para reducir los errores de acción es relativamente sencillo. Sin embargo, se requieren métodos de aprendizaje supervisado considerablemente sofisticados para asegurar que la función de control, F_W, exhibe propiedades de interpolación y extrapolación que implican una buena generalización del control. Varios investigadores han utilizado métodos de aprendizaje supervisado para entrenar a los controladores en tareas de aprendizaje supervisado [14, 15]. El experto (generalmente humano) proporciona un conjunto suficientemente grande de pares de entrenamiento que especifican las acciones deseadas para varias entradas del controlador, y el controlador está entrenado para producir la acción correspondiente para cada entrada. Uno de los primeros ejemplos del uso de este método es el entrenamiento de robots industriales para realizar operaciones repetitivas en una línea de montaje.

Los métodos para resolver problemas de control del aprendizaje que involucran el aprendizaje por refuerzo y el aprendizaje con un maestro distante son más complejos que los métodos aplicables a las tareas de aprendizaje supervisado. Es necesario un método para cerrar la brecha entre la forma en que la información de capacitación está disponible para el controlador (evaluaciones, objetivos distales, etc.) y la forma de información requerida para un control exitoso (acciones de control apropiadas). Los métodos indirectos implican la construcción de un modelo de la transformación de las acciones del controlador en evaluaciones u objetivos distantes y el uso del modelo para obtener información de entrenamiento para el controlador. Por otro lado, los métodos directos o sin modelo obtienen la información de entrenamiento necesaria al perturbar el proceso y observar el efecto sobre las evaluaciones o los resultados del proceso distal.

2.1.0.1. Métodos indirectos

Los métodos indirectos pueden usar modelos en al menos tres formas diferentes. En el control adaptativo indirecto convencional, se utiliza un modelo de proceso parametrizado como una representación matemática del proceso a partir del cual se puede obtener analíticamente una ley de control adecuada. Los parámetros del modelo de proceso se

adaptan en línea a través de una operación comúnmente conocida como identificación del sistema en la literatura de control. Debido a que la ley de control se deriva analíticamente usando el modelo actual, los métodos de control adaptativo indirecto difieren significativamente de los métodos de control de aprendizaje, que usan el modelo para obtener información para entrenar al controlador.

Un método indirecto también puede usar un modelo del proceso en la dirección "hacia adelante"para simular el comportamiento del proceso a lo largo del tiempo. Este es el enfoque que se usa con más frecuencia en los programas de juegos de IA ([4],[16]) en los que un modelo del juego se utiliza para generar árboles de búsqueda. Muchos algoritmos de búsqueda heurística se han desarrollado en AI [16] para este tipo de búsqueda. Claramente, estos algoritmos de búsqueda heurística también se pueden aplicar a problemas de control que no sean juegos y en situaciones en el que el modelo tiene que ser construido en línea El principal inconveniente de este enfoque es que el proceso de búsqueda directa es, en general, poco restringido y por lo tanto costoso en el termino computacional.

Un diagrama de bloques del método indirecto basado en gradientes para el control del aprendizaje. Este método se puede aplicar a problemas de control que involucran el aprendizaje con un maestro distante o el aprendizaje por refuerzo. Después de Jordan y Rumelhart [17] y Barto [18]. Esto se ilustra en la Figura 2.2.

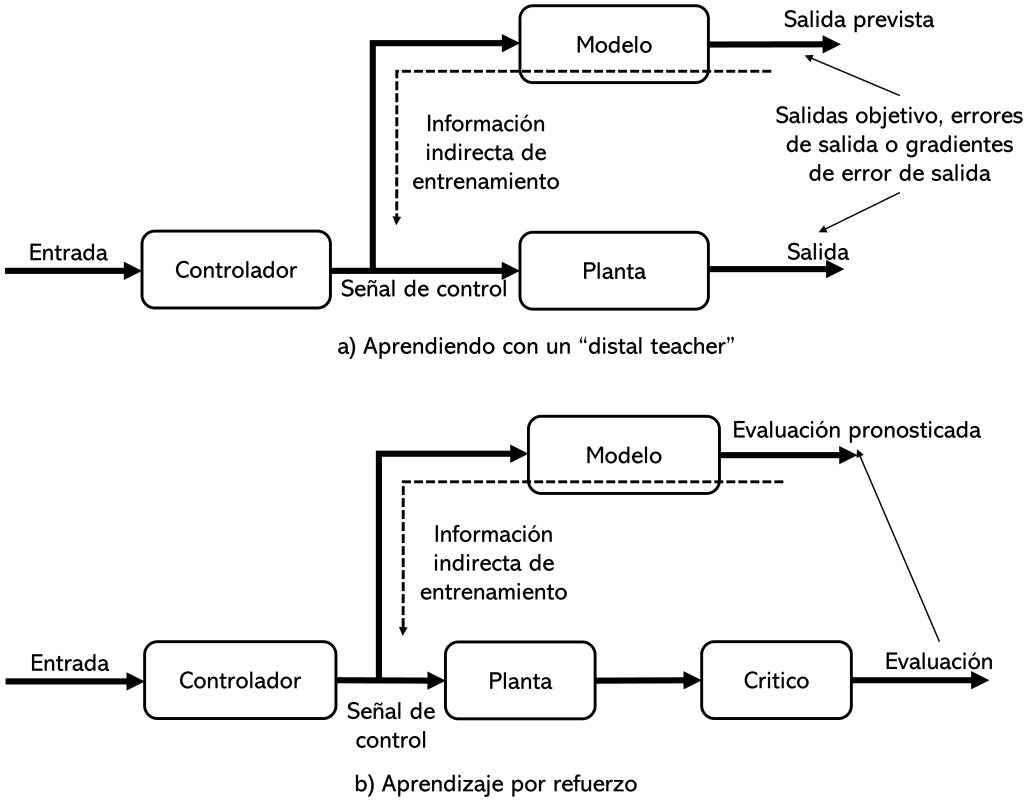

Figura 2.2: Un diagrama de bloques del método indirecto basado en gradientes para el control del aprendizaje. Este método se puede aplicar a problemas de control que involucran el aprendizaje con un maestro distante o el aprendizaje por refuerzo.

2.1.0.2. Métodos directos

En lugar de recurrir a la construcción de modelos, los métodos directos utilizan el propio proceso como fuente de datos de entrenamiento para entrenar al controlador. Para tareas de aprendizaje con objetivos distales, si la entrada de comando al controlador es la salida deseada del proceso, se ha propuesto como solución la identificación directa de un modelo inverso del proceso [19]. Tal método ha sido llamado ”modelado inverso directo”[20] en la literatura de aprendizaje conexionista y çoincidencia de entrada.[en] la literatura de control adaptativo [21]. Los datos de entrenamiento para el controlador se obtienen alimentando una variedad de señales de control al proceso y observando la salida del proceso resultante. Se utiliza un método de aprendizaje supervisado para entrenar el modelo inverso con la salida del proceso observado como entrada y las señales de control como las acciones deseadas, como se muestra en Figura 2.3. Una vez entrenado, el modelo inverso se puede usar como un controlador que produce una acción de control adecuada para cualquier salida de proceso deseada.

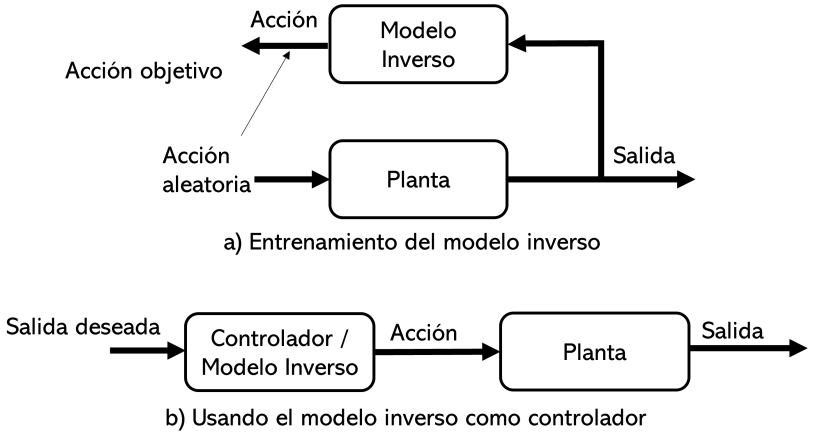

Figura 2.3: La parte (a) muestra la configuración utilizada para entrenar el modelo inverso, mientras que la parte (b) muestra cómo se utiliza el modelo inverso para controlar el proceso.[18]

2.1.0.3. Agente-Entorno

El problema del aprendizaje por refuerzo pretende ser un encuadre directo del problema del aprendizaje a partir de la interacción para lograr un objetivo. El agente es el sujeto del aprendizaje por refuerzo. Su funcionamiento consiste en leer el estado del entorno, realizar acciones sobre el entorno y leer las recompensas que producen estas acciones. El entorno es el objeto sobre el que opera el agente. El entorno recibe las acciones del agente y evoluciona. Su comportamiento suele ser desconocido y estocástico. Es el responsable de generar las recompensas asociadas a las acciones y cambios de estado.. El entorno también da lugar a recompensas, valores numéricos especiales que el agente trata de maximizar con el tiempo. Una especificación completa de un entorno define una tarea, una instancia del problema de aprendizaje por refuerzo.

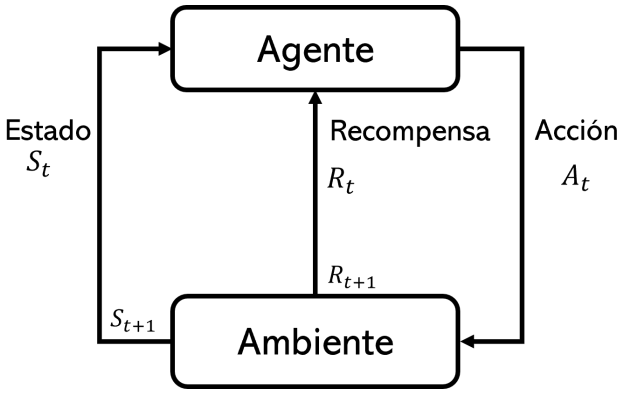

Figura 2.4: La interacción agente-entorno en el aprendizaje por refuerzo.

2.1.0.4. Proceso de Decision de Markov

Una tarea de aprendizaje por refuerzo que satisface la propiedad de Markov se denomina proceso de decisión de Markov o MDP. Si los espacios de estado y de acción son finitos, se denomina proceso de decisión de Markov finito (MDP finito).

Un MDP finito particular se define por sus conjuntos de estado y acción y por la dinámica de un solo paso del entorno. Dado cualquier estado y acción s y a, la probabilidad de cada par posible del siguiente estado y recompensa, s', r, se denota

$$p\left(s', r \mid s, a\right) = \Pr\left\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\right\} \tag{2.1}$$

Estas cantidades especifican completamente la dinámica de un MDP finito. La mayor parte de la teoría que presentamos en el resto de este libro asume implícitamente que el entorno es un MDP finito.

Dada la dinámica especificada por 2.1, se puede calcular cualquier otra cosa que desee saber sobre el entorno, como las recompensas esperadas para los pares estado-acción,

$$r(s,a) = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right] = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p\left(s', r \mid s, a\right),$$

las probabilidades de transición de estado,

$$p\left(s' \mid s, a\right) = \Pr\left\{S_{t+1} = s' \mid S_t = s, A_t = a\right\} = \sum_{r \in \mathcal{R}} p\left(s', r \mid s, a\right),$$

y las recompensas esperadas por los triples estado-acción-siguiente-estado,

$$r\left(s, a, s'\right) = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s'\right] = \frac{\sum_{r \in \mathcal{R}} r p\left(s', r \mid s, a\right)}{p\left(s' \mid s, a\right)}$$

2.2. Programación Dinámica

La programación dinámica es un enfoque basado en modelos para resolver problemas de aprendizaje por refuerzo [22]. Una forma de programación dinámica almacena el valor esperado de cada acción en cada estado. El acción-valor, Q, de una acción en un estado es la suma de la recompensa por realizar esa acción en ese estado más la recompensa futura esperada si la política descrita por action-values almacenados se sigue a partir de ese momento (Ecuación 2.2):

$$Q = (x_t, u_t) = E\left[r_t\left(x_t, u_t, X_{t+1}\right) + \gamma r_{t+1}\left(X_{t+1}, \underbrace{arg\,max}_{u_{t+2}} Q(X_{t+2}, u_{t+2}), X_{t+3}\right) + ...\right] \tag{2.2}$$

donde las variables aleatorias (probabilísticas) se denotan con letras mayúsculas. $arg\,max_u\, Q(x,u)$ es la acción con el valor más alto en el estado x, $max_u\, Q(x,u)$ es el valor de la acción de mayor valor en el estado x; esto se denomina valor del estado x.

2.2.0.1. Ecuación de Bellman y ecuación de optimización de Bellman

La programación dinámica es un método retrospectivo para encontrar el valor y la política óptimos. Por el contrario, el aprendizaje por refuerzo se ocupa de encontrar políticas óptimas basadas en la experiencia causal mediante la ejecución de decisiones secuenciales que mejoran las acciones de control basadas en los resultados observados del uso de una política actual. Este procedimiento requiere la derivación de métodos para encontrar valores óptimos y políticas óptimas que puedan ejecutarse en el tiempo. La clave de esto es la ecuación de Bellman. [1]

Para derivar métodos de avance en el tiempo para encontrar valores óptimos y políticas óptimas, establezca el horizonte de tiempo T en infinito y defina el costo de horizonte infinito

$$J_k = \sum_{i=0}^{\infty} \gamma^i r_{k+i} = \sum_{i=k}^{\infty} \gamma^{i-k} r_i \tag{2.3}$$

La función de valor de horizonte infinito asociada para la política $\pi(x,u)$ es

$$V^{\pi}(x) = E_{\pi}\left\{J_k \mid x_k = x\right\} = E_{\pi}\left\{\sum_{i=k}^{\infty} \gamma^{i-k} r_i \mid x_k = x\right\}. \tag{2.4}$$

2.2.0.2. Políticas y funciones de valor

Definición 2.2.1 (Política de Markov) . *Una política de Markov aleatoria y dependiente del tiempo es un conjunto de funciones* $(\pi_t)_{t\geq 0}$,

$$\pi_t : \mathcal{S} \times \mathcal{B}(\mathcal{A}) \to [0,1],$$

que satisface las siguientes condiciones.

- *Para todo* $x \in \mathcal{S}$ *y* $B \in \mathcal{B}(\mathcal{A})$, *la aplicación* $t \mapsto \pi_t(x,B)$ *es medible en* $[0,\infty)$.

- *Para todo $t \geq 0$ y $x \in \mathcal{S}$, la aplicación $B \mapsto \pi_t(x, B)$ es una medida de probabilidad en $\mathcal{B}(\mathcal{A})$ con $\pi_t(x, \mathcal{A}(x)) = 1$. Es $\pi_t(x, B)$ la probabilidad de que se elija una acción $a \in B$ cuando el proceso está en el estado $x \in \mathcal{S}$ en el tiempo $t \geq 0p$*

Tal política se llama estacionaria si $\pi_t(x, B) \equiv \pi(x, B)$ es independiente de t. Se llama determinista si para cada $t \geq 0$ y $x \in \mathcal{S}$ existe un $a \in \mathcal{A}(x)$ tal que $\pi_t(x, \cdot) = \delta_a$ es una medida de Dirac. Una política de Markov estacionaria determinista viene dada por una función

$$u : \mathcal{S} \to \mathcal{A} \tag{2.5}$$

con $u(x) \in \mathcal{A}(x)$ para todo $x \in \mathcal{S}$, asignando a cada estado una acción disponible de manera determinista. Denotamos el conjunto de todas las políticas de Markov aleatorias y dependientes del tiempo con Π y el conjunto de todas las políticas de Markov estacionarias deterministas con $\mathcal{U}$.

Remark 1 *Aquí, el término "Markov"se refiere al hecho de que la política es una función del estado real $x \in \mathcal{S}$ y no depende de la historia completa del proceso.*

Los algoritmos de aprendizaje por refuerzo implican la estimación de funciones de valor: funciones de estados (o de pares de estado-acción) que estiman qué tan bueno es para el agente estar en un estado dado (o qué tan bueno es realizar una acción dada en un determinado estado). La noción de "qué tan bueno.ªquí se define en términos de recompensas futuras que se pueden esperar o, para ser precisos, en términos de rendimiento esperado. Por supuesto, las recompensas que el agente puede esperar recibir en el futuro dependen de las acciones que tome. En consecuencia, las funciones de valor se definen con respecto a políticas particulares.

Una política, π, es un mapeo de cada estado, $s \in \mathcal{S}$, y acción, $a \in \mathcal{A}(s)$, a la probabilidad $\pi(a \mid s)$ de realizar la acción a cuando se está en el estado s. De manera informal, el valor de un estado s bajo una política π, denotado $v_\pi(s)$, es el rendimiento esperado cuando comienza en s y sigue π a partir de entonces. Para los MDP, podemos definir $v_\pi(s)$ formalmente como

$$v_\pi(s) = \mathbb{E}_\pi \left[G_t \mid S_t = s \right] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

donde $\mathbb{E}_\pi[\cdot]$ denota el valor esperado de una variable aleatoria dado que el agente sigue la política π, y t es cualquier paso de tiempo. Tenga en cuenta que el valor del estado terminal, si lo hay, siempre es cero. Llamamos a la función v_π la función de valor de estado para la política π.

De manera similar, definimos el valor de realizar la acción a en el estado s bajo una política π, denotada $q_\pi(s, a)$, como el rendimiento esperado a partir de s, tomando la

acción a, y luego siguiendo la política π :

$$q_\pi(s,a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

Llamamos a q_π la función de valor de acción para la política π.

2.3. Aprendizaje de diferencias temporales

El aprendizaje de diferencia temporal (TD) es un enfoque para aprender a predecir una cantidad que depende de los valores futuros de una señal dada. El nombre TD se deriva de su uso de cambios o diferencias en las predicciones sobre pasos de tiempo sucesivos para impulsar el proceso de aprendizaje. La predicción en cualquier paso de tiempo dado se actualiza para acercarla a la predicción de la misma cantidad en el siguiente paso de tiempo. Es un proceso de aprendizaje supervisado en el que la señal de entrenamiento para una predicción es una predicción futura. Los algoritmos TD a menudo se usan en el aprendizaje por refuerzo para predecir una medida de la cantidad total de recompensa esperada en el futuro, pero también se pueden usar para predecir otras cantidades. También se han desarrollado algoritmos TD de tiempo continuo.

2.4. Q-Learning

La programación dinámica es un enfoque basado en modelos para resolver problemas de aprendizaje por refuerzo [22]. Una forma de programación dinámica almacena el valor esperado de cada acción en cada estado. El action-value $\mathcal{Q}$, de una acción en un estado es la suma de la recompensa por realizar esa acción en ese estado más la recompensa futura esperada si la política descrita por los valores de acción almacenados se sigue desde luego en:

$$\begin{aligned}\mathcal{Q}\left(x_t, u_t\right) = E\big[r_t\left(x_t, u_t,\ \ X_{t+1}\right) \\ +\gamma r_{t+1}\left(X_{t+1}, \arg\max_{u_{t+1}} \mathcal{Q}\left(X_{t+1}, u_{t+1}\right), X_{t+2}\right) \\ +\gamma^2 r_{t+2}\left(X_{t+2}, \arg\max_{u_{t+2}} \mathcal{Q}\left(X_{t+2}, u_{t+2}\right), X_{t+3}\right) + \ldots\big]\end{aligned} \tag{2.6}$$

donde las variables aleatorias (probabilísticas) se denotan con letras mayúsculas. $\arg\max_u \mathcal{Q}(x,u)$ es la acción con el valor más alto en el estado $x \cdot \max_u \mathcal{Q}(x,u)$ es el valor de la acción de mayor valor en el estado x; esto se denomina valor del estado x. Los

valores de acción se denotan $\mathcal{Q}^*$ si satisfacen la ecuación de optimización de Bellman:

$$\mathcal{Q}^*(x_t, u_t) = E\left[r_t(x_t, u_t, X_{t+1}) + \gamma \max_{u_{t+1}} \mathcal{Q}^*(X_{t+1}, u_{t+1})\right] \forall x_t, u_t \tag{2.7}$$

Se garantiza que ejecutar $\arg_{\text{máx}_u} \mathcal{Q}^*(x, u)\ldots$ es una política óptima. En este marco, el problema de encontrar una política óptima se transforma en la búsqueda de $\mathcal{Q}^*$. El enfoque de programación dinámica encuentra $\mathcal{Q}^*$ iterativamente a través de un modelo dinámico directo.

$\mathcal{Q}$-Learning es un enfoque sin modelo para encontrar $\mathcal{Q}^*$ [23]. En Q-learning, la experiencia del mundo real ocupa el lugar del modelo dinámico: los valores esperados de las acciones en los estados se actualizan a medida que se ejecutan las acciones y se pueden medir los efectos. En $\mathcal{Q}$-aprendizaje de un paso, los valores de acción se actualizan mediante la ecuación de actualización de $\mathcal{Q}$ de un paso:

$$\mathcal{Q}(x_t, u_t) \xleftarrow{\alpha} r(x_t, u_t, x_{t+1}) + \gamma \max_{u_{t+1}} \mathcal{Q}(x_{t+1}, u_{t+1}) \tag{2.8}$$

donde α es una tasa de aprendizaje (o tamaño de paso) entre 0 y 1 que controla la convergencia. La flecha en la Ecuación 2.8 es el operador de movimiento hacia, no debe confundirse con la implicación lógica. La operación $A \xleftarrow{\alpha} B$ es equivalente a mover A hacia B en proporción a α.A y B son escalares o vectores. Si no se muestra α, es equivalente a 1 .

$$\begin{aligned} A \xleftarrow{\alpha} B, \text{ is equivalent to,} \\ A := (1-\alpha)A + \alpha B, \quad \alpha \in [0, 1] \end{aligned} \tag{2.9}$$

Bajo la actualización de $\mathcal{Q}$-learning de un paso, los valores de acción están garantizados con probabilidad 1 de converger a valores de acción óptimos ($\mathcal{Q}^*$) bajo las siguientes condiciones [24]:

- 1. cada acción se ejecuta en cada estado un número infinito de veces;
- α se reduce con un programa adecuado; y
- Los valores de acción se almacenan perfectamente (como en una tabla).

La verdadera convergencia hacia los valores de acción óptimos rara vez se puede lograr. En el uso práctico, el objetivo es que los valores de acción describan un controlador aceptable en un tiempo razonable.

2.5. Control de Robots mediante Aprendizaje por Reforzamiento

En este capítulo, se presentan las técnicas en el aprendizaje por reforzamiento y el modelo formal detrás del problema que resuelven: Tal como es el Proceso de Decisión de Markov. Además se considera el Proceso de Decisión de Markov desde el caso determinístico y estocástico, así como también su solución óptima y finalmente, llegar a una ecuación de recursividad de Bellman.

2.5.1. Control del Robot con Recompensas y Retornos

En el aprendizaje por reforzamiento, el propósito u objetivo del controlador, es el recibir una señal de recompensa que pasa del proceso al controlador. En cada lapso de tiempo, la recompensa es un simple número, $r_k \in \mathbb{R}$. En palabras simples, el objetivo del controlador será el de maximizar la cantidad total de recompensas que recibe, donde la recompensa no se maximiza de manera inmediata pero sí a largo plazo.

En general, se busca maximizar la esperanza del retorno, donde el retorno $R(x)$, se define como una función específica de la recompensa. El caso más simple del retorno es la suma de las recompensas:

$$R(x) = r_1 + r_2 + r_3 + \cdots + r_T \in \mathbb{R}$$

Donde T es el tiempo final. Este enfoque tiene mucho sentido en aplicaciones donde existe un tiempo final, que es, cuando la interacción entre el controlador y el proceso termina en una subsecuencia, llamada episodios. Cada episodio termina en un estado especial llamado estado terminal, seguido de una reinicialización a los estados iniciales prefijados o a una muestra de una distribución estándar de los estados iniciales.

Por otra parte, en muchos casos la interacción entre el controlador y el proceso no termina de manera natural en algún episodio, sino que se sigue continuamente sin límite. Por ejemplo, este sería el caso de una tarea de control de procesos continuo, o una aplicación a un robot con larga vida útil. Se le conoce como tareas continuas. La ecuación (3.1) resulta problemática para tareas continuas ya que el tiempo final sería $T = \infty$, y el retorno, que es lo que tratamos de maximizar, podría fácilmente ser infinito. (Por ejemplo: suponer que el controlador recibe una recompensa de $+1$ en cada instante de tiempo). Entonces el concepto adicional que necesitamos es el de descuento. En este enfoque el controlador trata de seleccionar acciones tal que la suma de las recompensas con descuento que recibe en el futuro sea maximizada. En particular se escoge u_k para

maximizar la esperanza del retorno con descuento:

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

donde γ es un parámetro, $0 \leq \gamma \leq 1$, llamado factor de descuento y la ecuación (3.2) es ligeramente más compleja conceptualmente pero mucho más simple matemáticamente. El factor de descuento determina los valores presentes de las futuras recompensas: por ejemplo, una recompensa recibida en el tiempo k en el futuro sólo tendrá un valor de γ^{k-1} veces lo que valdría si fuera recibida de manera inmediata. Si $\gamma < 1$, la suma infinita tiene ahora un valor finito, siempre y cuando la secuencia de la recompensa $\{R\}$ esté acotada. Si$\gamma = 0$, al agente se le conoce como miope ya que sólo se enfoca en maximizar la recompensa inmediata: su objetivo en este caso es aprender como escoger u_k para maximizar sólo r_{k+1}. Si cada acción del controlador tuviera una influencia sólo en la recompensa inmediata y no de las futuras recompensas, entonces el controlador miope podría maximizar (3.2) de manera separada únicamente maximizando cada recompensa inmediata. Pero en general, el sólo maximizar la recompensa inmediata puede reducir el acceso a futuras recompensas de modo que el retorno puede reducirse. Mientras γ se aproxime a 1, el objetivo toma en consideración las recompensas futuras con más fuerza, el controlador llega a ser más clarividente.

2.5.2. Control del Robot en el Proceso de Decisión de Markov

En particular, formalmente definimos la propiedad del ambiente y sus señales de estado conocida como la propiedad de Markov. Se supone que existe un número finito de estados y recompensas, lo cual permite trabajar en términos de sumas y probabilidades en lugar de integrales y densidad de probabilidades, pero el argumento fácilmente puede ser extendido a incluir recompensas y estados continuos. 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 distribución de probabilidad:

$$\Pr\left\{r_{k+1} = r, x_{k+1} = x' \mid x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k\right\},$$

para todo x', r, y todos los posibles valores de los eventos pasados, entradas, salidas y recompensas: $x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k$. 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:

$$\Pr\left\{r_{k+1} = r, x_{k+1} = x' \mid x_k, u_k\right\}.$$

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, si y sólo si la ecuación (3.3) es igual a la ecuación (3.4) para todo x', r y las historias de $x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k$. En este caso, se dice que el ambiente y las tareas tienen la propiedad de Markov.

Los problemas de DP y RL se pueden formalizar con la ayuda de MDP (Puterman, 1994). Primero presentamos el caso más simple de MDP con transiciones de estado deterministas. Luego, extendemos la teoría al caso estocástico.

2.5.3. Control de un Robot en el Caso Determinístico

Un MDP 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 $f : X \times U \to X$:

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

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

$$r_{k+1} = \rho\left(x_k, u_k\right),$$

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

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

$$u_k = h\left(x_k\right).$$

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}. Esta es la propiedad de Markov que es esencial para proporcionar las garantías teorícas sobre los algoritmos DP/RL. Algunos procesos de desición de Markov tienen estados terminales que, una vez que son alcanzados ya no los pueden dejar; y todas las recompensas recibidas en el estado terminal son 0. En la literatura del RL frecuentemente se usan ensayos o episodios para referirse a trayectorias que empiezan en un estado inicial y finalizan en un estado terminal.

2.5.3.1. Optimización en el Caso Determinístico

En DP y RL, el objetivo es encontrar una política óptima que maximice el retorno desde cualquier estado inicial x_0. El retorno es una agregación acumulativa de las recompensas a lo largo de las trayectorias empezando desde x_0. Representa de forma concisa la recompensa obtenida por el controlador a largo plazo. Diferentes tipos de retornos existen, dependiendo de la manera en que se acumula la recompensa. (Bertsekas y Tsitsiklis, 1996; Kaelbling 1996). El objetivo del aprendizaje en los procesos de decisión de Markov MDP es acumular recompensas. Si el controlador sólo tomara en cuenta la recompensa inmediata, un simple criterio de optimización sería optimizar (R). Sin embargo, hay varias maneras de tomar esto en cuenta. Existen básicamente tres modelos de optimización en los MDP que son suficientes para cubrir la mayoría de los enfoques en la literatura.

Horizonte infinito:

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

donde $\gamma \in [0,1)$ es el factor de descuento y $x_{k+1} = f\left(x_k, h\left(x_k\right)\right)$ para $k \geq 0$. El factor de descuento puede ser interpretado intuitivamente como que tan clarividente es el controlador al considerar sus recompensas, o también como una manera de tomar en cuenta la creciente incertidumbre sobre las recompensas futuras. Desde un punto de vista matemático, el descuento asegura que el retorno siempre estará acotado si las recompensas son acotadas. El objetivo es por lo tanto maximizar el desempeño a largo plazo (retornos), mientras sólo usamos la realimentación del desempeño inmediato (recompensa). Esto nos lleva al conocido reto de recompensas con retardo (Sutton y Barto, 1998): donde las acciones tomadas en el presente afectan el potencial de lograr una buena recompensa en el futuro, pero las recompensas inmediatas no proveen información sobre los efectos a largo plazo.

Igualmente, otro tipo de retornos pueden ser definidos, como el retorno sin descuento, obtenido al dejar γ con un valor de 1 en la ecuación (3.5), 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:

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

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 trayectorias finitas de longitud K (el horizonte), en lugar de trayectorias infinitas. De hecho, el retorno de

horizonte finito con descuento puede ser definido como:

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

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 utilizaremos el retorno con descuento de Horizonte Infinito (3.5) por que cuenta con muchas propiedades teóricas y aunque es ligeramente más complejo conceptualmente, también mucho más simple matemáticamente. En particular, para este tipo de retornos, bajo ciertas suposiciones técnicas existe por lo menos una política óptima determinista estacionaria $h^* : X \to U$ (Bertsekas y Shreve, 1978, Capítulo 9). En contraste con el caso del horizonte finito, las políticas óptimas dependen en general del paso de muestreo k, i.e., no son estacionarias (Bertsekas, 2005a, Capítulo 1). Mientras que el factor de descuento γ puede ser considerado teóricamente como una parte del problema, en la práctica, se debe elegir un buen valor de γ. Escoger γ a menudo implica una compensación entre la calidad de la solución y la tasa de convergencia del algortimo DP/RL, por las siguientes razones: algunos algoritmos importantes convergen más rápido cuando γ es más pequeño (este es el caso de iteraciones basadas en modelo). Sin embargo, si γ es muy pequeño, la solución puede llegar a ser no satisfactoria por que no toma suficientemente en cuenta las recompensas obtenidas después de una gran cantidad de pasos. Asimismo, se puede interpretar γ de varias maneras: puede ser visto como un factor de interés, una probabilidad de que exista otro paso, o como un truco matemático para acotar la suma infinita. Además el modelo con descuento es matemáticamente más tratable que el modelo del horizonte finito. Esta es una de las razones por la cual este modelo ha recibido gran atención.

2.5.3.2. Funciones Valor y Ecuación de Bellman

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). Frecuentemente en la literatura el nombre de funciones valor se utiliza tanto para funciones Q como para funciones V, en este texto se utilizará el nombre de función Q y función V para diferenciarlos claramente uno del otro. Primero definiremos las funciones Q y más adelante se explicarán las funciones V.

La función Q que está definida como $Q^h : X \times U \to \mathbb{R}$ de una política h nos da como resultado el retorno obtenido cuando empezamos desde un estado dado, aplicando una acción dada, y siguiendo una política h por lo tanto tenemos:

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

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,

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

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

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

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

$$Q^*(x,u) = \max_h Q^h(x,u).$$

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

$$h^*(x) \in \arg\max_u Q^*(x,u),$$

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

$$h(x) \in \arg\max_u Q(x,u),$$

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 (3.8) para calcular una política en Q^*

Las funciones Q^h y Q^* se caracterizan recursivamente por la ecuación de Bellman, y que además son de importancia central para los algoritmos de valor de iteración y de política de iteración. La ecuación de Bellman para Q^h, nos 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:

$$Q^h(x,u) = \rho(x,u) + \gamma Q^h(f(x,u), h(f(x,u)))$$

Esta ecuación de Bellman puede ser obtenida de la ecuación (3.6) como sigue a continuación:

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

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

$$Q^*(x,u) = \rho(x,u) + \gamma \max_{\dot{u}} Q^*\left(f(x,u), u'\right)\Big).$$

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

$$V^h(x) = R^h(x) = Q^h(x, h(x)).$$

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:

$$V^*(x) = \max_h V^h(x) = \max_u Q^*(x,u).$$

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

que satisface:

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

Usando esta formula es más difícil que usar la ecuación (3.8); en particular, un modelo de MDP se necesita en la forma de la dinámica f y la recompensa ρ. Debido a que la función Q también depende de la acción, esta ya incluye la información sobre la calidad de la transición. Por el contrario, la función V sólo describe la calidad del estado, y para inferir sobre la calidad de las transiciones, estas deben tenerse en cuenta de manera explícita. Esto es lo que vemos en la ecuación (3.14), y esto explica por que es más difícil calcular políticas desde las funciones V. Debido a estas diferencias las funciones Q se preferirán a las funciones V, aunque sea más costoso que representar funciones V, ya que la función Q depende tanto de x y u.

Las funciones V^h y V^* satisfacen las siguientes ecuaciones de Bellman, que son similares a las ecuaciones (3.10) y (3.11):

$$V^h(x) = \rho(x, h(x)) + \gamma V^h(f(x, h(x)))$$
$$V^*(x) = \max_u \left[\rho(x,u) + \gamma V^*(f(x,u))\right]$$

2.5.4. Control de un Robot en el Caso Estocástico

Una tarea del aprendizaje por reforzamiento que satisface la propiedad de Markov se le conoce como Proceso de decisión de Markov MDP. Si el espacio de estados y las acciones son finitos entonces se le conoce como proceso de decisión de Markov finito (MDP finito). MDP finito son muy importantes en la teoría del aprendizaje por reforzamiento. Dada cualquier estado y acción, x y u, la probabilidad de que el siguiente estado sea x' es:

$$p\left(x_{k+1} \mid x_k, u_k\right) = \tilde{f}\left(x_k, u_k, x'\right) = \Pr\left\{x_{k+1} = x' \mid x_k = x, u_k = u\right\}.$$

Estas cantidades se llaman probabilidades de transición. Similarmente dado cualquier estado o acción actual, x, u, junto con el siguiente estado, x, la esperanza del valor de la siguiente recompensa es:

$$r_{k+1} = \tilde{\rho}\left(x_k, u_k, x_{k+1}\right) = E\left\{r_{k+1} \mid x_k = x, u_k = u, x_{k+1} = x'\right\}.$$

Estas cantidades $p\left(x_{k+1} \mid x_k, u_k\right)$ y $\tilde{\rho}\left(x_k, u_k, x_{k+1}\right)$, especifican completamente los aspectos más importantes de la dinámica de un MDP finito.

2.5.4.0.1. Funciones de Valor y Ecuación de Bellman Uno de los objetivos en el aprendizaje por reforzamiento es estimar qué tan bueno es estar en un estado (o estar en un estado y realizar una acción). La noción de qué "tan bueno"se define en términos de las futuras recompensas o la esperanza de las recompensas acumuladas que son representadas

como funciones de valor. La función valor de un estado x denotado por $V^h(x)$, representa la esperanza total de la recompensa acumulada que el agente puede recibir iniciando en el estado xy siguiendo la política h. De manera similar, la función valor del estado x, tomando la acción u, denotado por $Q^h(x,u)$ representa la esperanza total de la recompensa acumulada que el agente puede recibir iniciando en el estado x, tomando la acción u y siguiendo la política h. La idea es encontrar una política que produzca el máximo de la función valor en lugar del máximo de una recompensa inmediata. Las recompensas son dadas por el proceso, pero las funciones valor necesitan ser estimadas (aprendidas) con la experiencia [5]. En consecuencia, las funciones valor son definidas con respecto a políticas particulares. Recordar que una política h, es un mapeo de cada estado $x \in X$, y acción $u \in U(x)$, a una probabilidad h de tomar una acción u encontrándose en el estado x. De manera informal, el valor de un estado x bajo la política h, denotado por $V^h(x)$, es la esperanza del retorno cuando se inicia en el estado x y se sigue h. Para MDP, podemos definir formalmente $V^h(x)$ como:

$$V^h(x) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x\right],$$

donde $E[\cdot]$, representa la esperanza del valor dado cuando el agente sigue una política h, y k es cualquier paso de tiempo. Llamamos a la función $V^h(x)$ la función del valor estado para la política h.

Una propiedad fundamental de las funciones de valor es que satisfacen ciertas propiedades de recursividad. Para cualquier política h y cualquier estado x la expresión de la ecuación (3.17) puede ser definida recursivamente en términos de la ecuación de Bellman (1957):

$$\begin{aligned}
V^h(x) &= E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x\right] \\
V^h(x) &= E\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \cdots \mid x_k = x\right] \\
V^h(x) &= E\left[r_1 + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \mid x_k = x\right] \\
V^h(x) &= E\left[\tilde{\rho}\left(x, u, x'\right) + \gamma E\left[\sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \mid x_{k+1} = x'\right] \mid x_k = x\right] \\
V^h(x) &= E\left[\tilde{\rho}\left(x, u, x'\right) + \gamma V^h\left(x'\right)\right]
\end{aligned}$$

donde está implícito que las acciones u, son tomadas del conjunto $U(x)$, y los siguientes estados $\dot{x}$, son tomados del conjunto de estados X. La ecuación (3.18) es la ecuación de

Bellman para V. Expresa una relación entre el valor del estado y el valor de su estado sucesor.

Similarmente, se define el valor de tomar una acción u en el estado x bajo la política h, denotado por $Q^h(x,u)$, como la esperanza del retorno iniciando en x, tomando la acción u, y siguiendo la política h.

$$Q^h(x,u) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x, u_k = u\right],$$

donde Q^h se le conoce como función valor-acción para la política h. (Sutton and Barto 2012).

Haciendo un análisis de recursividad como el anterior podemos derivar una ecuación equivalente para $Q^h(x,u)$:

$$Q^h(x,u) = E\left[\tilde{\rho}\left(x,u,x'\right) + \gamma Q^h\left(x', h\left(x'\right)\right)\right]$$

2.5.4.0.2. Ecuación de Optimización de Bellman En la práctica, las mejores políticas son obtenidas por aquellas que producen la esperanza de la recompensa acumulada más grande. Una política h se considera mejor o igual que otra política h si la esperanza de los retornos es más grande o igual que los de h para todos los estados.

$h \geq h'$ si $V^h(x) \geq V^{h'}$ para todo $x \in X$.

Existe al menos una política que es mejor o igual que todas las demás políticas, llamada política óptima, aunque podría existir más de una, representamos todas las políticas óptimas como h^*. Su función valor conocida como función de valor óptimo, representado por V^*y definida como:

$$V^*(x) = \max_h V^h(x),$$

para todo $x \in X$.

Las políticas óptimas también comparten la misma función de valor-acción representada por Q^* y definida como:

$$Q^*(x,u) = \max_h Q^h(x,u).$$

Para todo $x \in X$ y $u \in U(x)$, para el par estado-acción (x,u), donde esta función da la esperanza del retorno al tomar la acción u en el estado x y luego seguir una política óptima. Entonces, se puede escribir Q^* en términos de V^* de la siguiente manera:

$$Q^*(x,u) = E\left[\tilde{\rho}\left(x,u,x'\right) + \gamma V^*\left(x_{k+1}\right) \mid x_k = x, u_k = u\right].$$

Considerando las ecuaciones (3.19) y (3.20), las funciones de valor óptimo pueden ser expresadas recursivamente con la ecuación de optimización de Bellman de la siguiente

manera:

$$V^*(x) = \max_u Q^h(x,u)$$
$$V^*(x) = \max_u E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x, u_k = u\right]$$
$$V^*(x) = \max_u E\left[r_{k+1} + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \mid x_k = x, u_k = u\right]$$
$$V^*(x) = \max_u E\left[\tilde{\rho}\left(x,u,x'\right) + \gamma V^*\left(x_{k+1}\right) \mid x_k = x, u_k = u\right]$$
$$V^*(x) = \max_u E\left[\tilde{\rho}\left(x,u,x'\right) + \gamma V^*\left(x'\right)\right]$$

y similarmente para Q :

$$Q^*(x,u) = E\left[r_{k+1} + \gamma \max_{u'} Q^*\left(x_{k+1}, u'\right) \mid x_k = x, u_k = u\right]$$
$$Q^*(x,u) = E\left[\tilde{\rho}\left(x,u,x'\right) + \gamma \max_{u'} Q^*\left(x', u'\right)\right].$$

Uno de los principales avances en la investigación de métodos de aprendizaje por refuerzo, fue el de la introducción de los métodos de diferencia temporal (TD), los cuales son una clase de procedimientos de aprendizaje incremental especializados en problemas de predicción. Dichos métodos son conducidos por el error o diferencia entre predicciones sucesivas temporales de los estados. El aprendizaje ocurre en cada momento que se produce un cambio en la predicción al paso del tiempo.

El método más simple conocido como TD (0) actualiza la estimación de la función de valor, después de ir del estado x al estado x_{k+1} y de recibir la recompensa r, utilizando la siguiente regla:

$$V_{k+1}(x) \rightarrow V_k(x) + \alpha\left[r + \gamma V_k\left(x'\right) - V_k(x)\right]$$

Varios métodos populares tales como Q-Learning, Sarsa y los métodos de Actor-Crítico fueron desarrollados basándose en dicha regla.

Aunque el aprendizaje por reforzamiento es una estrategia difícil para resolver el problema de aprendizaje automático, promete el desarrollo de sistemas computacionales capaces de auto-mejorar sus desempeños, lo cual es un objetivo principal de la comunidad de investigadores en inteligencia artificial. Dos de las aplicaciones más impresionantes de estos métodos son el jugador TD-Gammon (Gerald Tesauro, IBM, 1992), y el algoritmo de control de helicóptero. El primer trabajo inyectó nuevo interés en el estudio de los métodos de aprendizaje por refuerzo; mientras que el segundo trabajo es una de las mejores aplicaciones del mundo real que se han logrado con los métodos de RL.

2.5.5. Control de un Robot con Q-Learning

Q-Learning es un algoritmo de control por diferencia temporal off -policy que aproxima directamente la función de valor estado-acción óptima, independientemente de la política seguida. Es uno de los algoritmos de aprendizaje por refuerzo más populares. El algoritmo del Esquema 1. muestra los pasos a seguir de manera secuencial. Si en el límite los valores de las acciones para todos los pares de estado-acción son actualizados un número infinito de veces, con un valor decreciente de α entonces el algoritmo converge a Q con probabilidad 1 .

Algoritmo Q-Learning

1. Inicializar $Q(x, u)$ arbitrariamente, para cada episodio de entrenamiento hacer
2. inicializar x
3. repetir para cada paso del episodio
4. escoger u desde x usando la política derivada de Q (ej..u ambiciosa)
5. realizar la acción u, observar r, x'
6. $Q(x, u) \leftarrow Q(x, u) + \alpha \left[r + \gamma \, \text{máx}_u \, Q\left(x', u\right) - Q(x, u)\right]$
7. $x < x'$
8. hasta x es terminal

Capítulo 3

Human in the Loop Control

Human-in-the-loop es el término que se utiliza a menudo en la literatura sobre teoría de control para describir la participación del ser humano en los sistemas físicos como las redes neuronales [25–27], las redes difusas [28, 29] y el aprendizaje por refuerzo [1].

Human-in-the-loop se refiere particularmente a una situación en la que un sistema o una máquina están controlados, total o parcialmente, por un humano. Human-in-the-loop también puede significar que el humano es monitoreado o incluso controlado por una máquina, al que nos referimos como pasivo. En la configuración humana activa, el humano observa la salida del sistema a través de, por ejemplo, una pantalla en la que puede ver toda la información necesaria para actualizar sus acciones de control o sus decisiones. Esta es la arquitectura típica de un control realimentado (véase figura 3.1). Por lo tanto, el HITL puede modelarse como un sistema de entrada-salida, de manera similar a cualquier sistema dinámico. Esto ha llevado al desarrollo de varios modelos dinámicos que imitan el comportamiento humano.

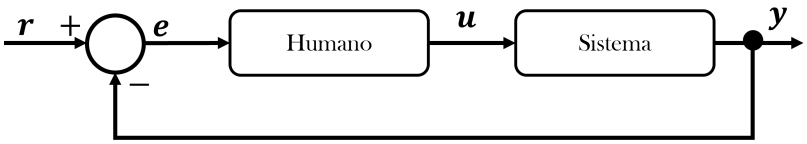

Figura 3.1: Human in the loop control

3.1. Trabajo Relacionado

Durante la primera década del siglo XXI de las interacciones humano-robot (HRI), se discutieron temas sobre la definición, taxonomía y modelos. De acuerdo con la definición presentada en 1992 por el Grupo de Desarrollo Curricular de la Asociación de Maquinaria de Computación (ACM) y el Grupo de Interés Especial sobre

Interacción Computadora-Humana (SIGCHI): "La interacción persona-computadora es una disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para uso humano y con el estudio de los principales fenómenos que los rodean "[30]. El robot se ajusta a la definición de los sistemas informáticos y, por lo tanto, la interacción humano-robot (HRI) podría considerarse como un subconjunto del área de interacción humano-computadora (HCI)[31–34].

Existe un trabajo significativo en la literatura sobre esquemas de HITLCPs[35, 36] y sistemas robóticos[10, 37–43]. El tema de HITL se ha tratado con cierta generalidad, aunque sin una perspectiva integral de sistemas y controles. En [44] se propone una taxonomía de "Human In The Loop Control Physical Systems" (HITLCPS), (vease figura Figura 3.2). Es posible organizar las aplicaciones HiTL existentes en tres tipos: ***i)*** aplicaciones donde los humanos controlan directamente el sistema, ***ii)*** aplicaciones donde el sistema monitorea pasivamente a los humanos y toma las acciones apropiadas, y ***iii)*** un híbrido de ***i)*** y ***ii)***.

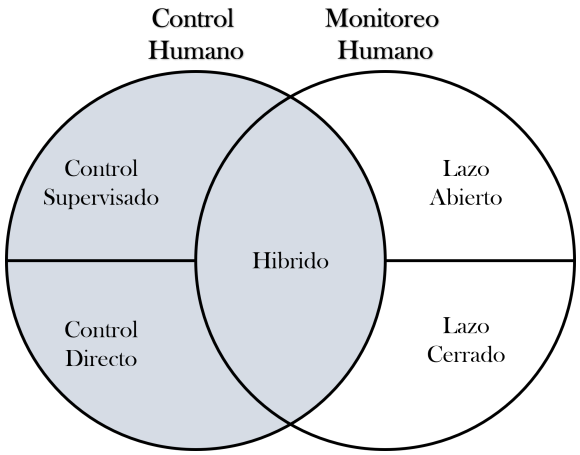

Figura 3.2: Taxonomía de aplicaciones de HITL[44]

La investigación sobre el modelado humano comenzó utilizando el concepto de describir la función del comportamiento humano en los trabajos de Tustin [45]. El modelo cuasi-lineal es uno de los primeros modelos humanos[46] propuestos por McRuer y Krendel [47], consta de una función descriptiva y una señal remanente que explica el comportamiento no lineal. McRuer y Krendel[48] ofrecen una descripción general de este modelo. En algunas aplicaciones donde el comportamiento lineal puede ser dominante, la parte no lineal de este modelo puede ignorarse y el compensador de tipo adelanto-retraso resultante se utiliza en el análisis de estabilidad de lazo cerrado [49]. El modelo crossover, propuesto por McRuer y Graham [50], es un modelo humano prominente en aeronáutica[51, 52].

3.2. Beneficios

Cada paso que incorpora la interacción humana exige que el sistema esté diseñado para ser entendido por los humanos para tomar la siguiente acción, y que exista alguna agencia humana para determinar los pasos críticos.
. El valor de los sistemas HITL no radica únicamente en la eficiencia o la corrección, sino también en la preferencia y la agencia humanas, ya estos sistemas ponen a los seres humanos en el circuito de decisiones.

Las estrategias de diseño HITL a menudo pueden mejorar el rendimiento del sistema en comparación con los sistemas totalmente automatizados y totalmente manuales. Esto se alinea con la noción de que un sistema híbrido no puede funcionar peor que los sistemas completamente automatizados, es decir, según lo permita el diseño, el ser humano puede ceder al resto del sistema siempre que lo desee

3.3. Retos y contribuciones

Los sistemas de HITLC ofrecen oportunidades interesantes para una amplia gama de aplicaciones incluida la gestión de energía [53], el cuidado de la salud [54] y los sistemas de automóviles [55]. Aunque tener a HITL tiene su ventaja, modelar los comportamientos humanos es extremadamente desafiante debido al complejo aspecto fisiológico, psicológico y de comportamiento de los seres humanos.

1. *La necesidad de una comprensión integral del espectro completo de tipos de HITLC* Se ha realizado un esfuerzo muy limitado en esta dirección. En [44] se presenta una taxonomía de las aplicaciones que involucran humanos. La taxonomía se basa en los roles humanos en una aplicación determinada. Las aplicaciones basadas en taxonomía deben incluir varias características clave que distinguen las diferentes aplicaciones y el rol humano en esta aplicación.

    a) *Nivel de inteligencia del sistema bajo control.* El aspecto conductual del operador dependerá del nivel de inteligencia del sistema bajo control y de la capacidad del sistema para ejecutar decisiones de forma autónoma. Agrupar las aplicaciones con el mismo nivel de inteligencia ayudará a identificar patrones de comportamiento similares de los operadores humanos. Definir las características y los límites de sus niveles no es una tarea fácil, sin embargo, es una característica clave esencial que define el comportamiento del modelo humano en la tarea de control.

    b) *Capacidad de control del sistema.* Los sistemas de ingeniería tienen diferentes grados de controlabilidad basados, por ejemplo, en la naturaleza no holonómica

del sistema. Los sistemas mecánicos no holonómicos, como los robots móviles con ruedas, los automóviles, los vehículos submarinos autónomos, los vehículos aéreos no tripulados, los robots poco accionados, no pueden moverse en una dirección arbitraria en su espacio de configuración. El grado de controlabilidad a menudo se define como la energía de entrada mínima para cambiar los estados del sistema [56–58].

c) *La resiliencia y robustez del sistema.* Un sistema resiliente [59] es un sistema que se adapta a la incertidumbre cambiando su método de operaciones mientras continúa funcionando[60]. Sin embargo, un sistema robusto es el sistema que continúa funcionando en presencia de incertidumbre limitada sin ningún cambio en el sistema original[61]. El grado de robustez y resistencia del sistema controlado afecta el comportamiento humano mientras controla el sistema[62]. Por tanto, es lógico clasificar los sistemas en función de su grado de robustez y resiliencia.

d) *Habilidades necesarias para operar el sistema.* El comportamiento humano, como operador o controlador en un escenario particular, dependerá del tiempo de contacto entre el ser humano y el sistema controlado[63].

2. *La necesidad de extensiones a la identificación del sistema u otras técnicas para derivar modelos de comportamientos humanos.* Capturar el comportamiento humano ampliando la identificación del sistema u otras técnicas de modelado es extremadamente difícil debido a los complejos aspectos fisiológicos, psicológicos y conductuales de los seres humanos. Además, el nivel de modelado depende de los requisitos de la aplicación. Aunque los requisitos son diferentes para diferentes aplicaciones, una parte significativa de las aplicaciones de HITL tienen que abordar algunos desafíos comunes, por ejemplo, umbrales y parámetros específicos del usuario, cambio de comportamiento humano a lo largo del tiempo y tecnología de detección requerida para detectar el valor apropiado, aspectos del comportamiento humano. Necesitamos modelar el comportamiento humano para un gran número de aplicaciones antes de que surjan teorías y principios generales para abordar estos problemas.Los sistemas CPS robustos probablemente requerirán modelos predictivos para evitar problemas antes de que ocurran, por lo que también se requieren avances en el control predictivo del modelo estocástico [64] [65].

3. *Determinar cómo incorporar modelos de comportamiento humano a la metodología formal de control de realimentación.* Incluso si tenemos un modelo de comportamiento humano, no está claro dónde colocar el modelo para çada.ªplicación[53, 66]. Tomando como ejemplos: a)Human-in-the-plant, b)Human-in-the-controller, c)Human-machine-symbols, d)Human-in-loops

3.4. Retrasos de tiempo en sistemas de HITLC

En muchos sistemas dinámicos, existe demora en la adquisición de información, la toma de decisiones y la ejecución de decisiones [67] que contribuyen a que los eventos no sucedan simultáneamente. Los sistemas con retrasos existen ampliamente en ingeniería, biología, física, investigación de operaciones y economía [67].

En [68], se ilustran los retrasos en el cerebro humano y sus efectos. Según el ejemplo de Stepan, las vibraciones existen en cada cuerpo humano, por ejemplo, durante el equilibrio. Los seres humanos sanos podrían suprimir fácilmente las vibraciones y mantener la estabilidad. Sin embargo, debido al mal funcionamiento del sistema neural, un retraso incrementado [69] podría causar cambios inmanejables en la fase de las señales neurales. Se comenta ampliamente que el temblor en los dedos, el brazo y el cuerpo; dificultades para equilibrar; el mayor peligro de caída para las personas mayores e incluso los trastornos del movimiento en el caso de episodios de epilepsia, esclerosis múltiple, enfermedad de Parkinson, etc., se deben en parte al aumento anormal del retraso en el sistema neural humano [68].

La existencia de demoras no contribuye necesariamente a la inestabilidad de un sistema. En algunos casos, la presencia de retrasos podría ayudar a estabilizar el sistema [67]. En [70], un controlador está diseñado para estabilizar el sistema de agentes múltiples introduciendo retrasos intencionalmente en el controlador. Las discusiones sobre los efectos estabilizadores del retraso se pueden encontrar en [67].

3.5. Modelado del humano

3.5.1. Modelado en el dominio de la frecuencia del operador humano

Modelar al operador humano como un conjunto de ecuaciones diferenciales de coeficiente constante lineal sugiere representar al humano como una función de transferencia. Este enfoque, generalizado para describir descripciones de funciones, captó la atención de algunos de los primeros y más influyentes ingenieros de control manual [47].

La figura 12.2 muestra una función descriptiva que representa al operador o controlador humano en una tarea de seguimiento de una entrada y una salida (SISO). Aquí la representación de la función de transferencia del ser humano se ha generalizado como una función descriptiva cuasi-lineal [71] mediante la adición de una señal remanente.ªditiva , $n_e t$). Esta señal representa la parte de la señal de error del sistema $e(t)$ inexplicable por el comportamiento del operador lineal, y no correlacionada linealmente con la entrada

del sistema $c(t)$.

Las mediciones espectrales del remanente se fusionaron mejor cuando se supuso que el remanente se inyectaba en el error mostrado $e(t)$ en lugar de la salida del operador $\&(t)$. Por esta razón, la porción remanente de la función descriptiva cuasi-lineal se muestra casi universalmente con remanente inyectado por error.

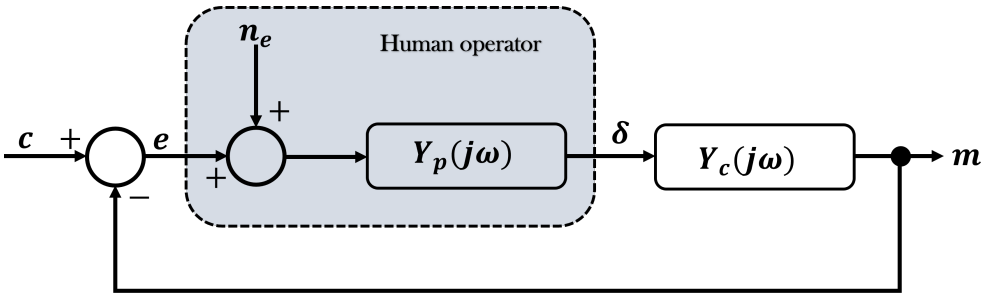

Figura 3.3: Una representación cuasi-lineal de función descriptiva del operador humano[50]

La identificación en el dominio de la frecuencia del operador humano que describe funciones en tareas simples de seguimiento de laboratorio se ha estudiado activamente durante las últimas tres décadas [72]. En estos experimentos, las funciones descriptivas identificadas fueron $Y_c(jw)$ y $\phi_{n_e n_e}(w)$, donde la última cantidad se define como la densidad espectral de potencia de la señal remanente $n_e(t)$.El elemento o planta controlada era miembro de un conjunto de dinámicas estereotipadas de elementos controlados, i.e., $Y_c(s) = \frac{K_c}{s_k}$, $k = 0, 1, 2$, y las entradas o las perturbaciones eran señales de aparición aleatoria, a menudo generadas como sumas de sinusoides. Los resultados llevaron a uno de los primeros modelos de ingeniería verdaderos del operador humano, denominado Çrossover model". Çrossover model"se puede definir como un modelo de la combinación del humano / planta (loop transmision) para tareas compensatorias que establece que la transmisión en realimentacion en sistemas SISO controlados manualmente se puede aproximar mediante un integrador y un retardo de tiempo alrededor de la frecuencia de cruce.

3.5.2. Modelado en el dominio del tiempo del operador humano

El advenimiento de una técnica de síntesis de control en el dominio del tiempo a mediados de la década de 1960, conocida como diseño lineal cuadrático gaussiano (LQG) condujo a un poderoso modelo del operador humano llamado Modelo de Control Óptimo (MCO). Este modelo se diferencia de los definidos anteriormente porque es algorítmico y se basa en un procedimiento de optimización en el dominio del tiempo. El modelo es algorítmico porque la especificación cuantitativa de ciertas limitaciones del procesamiento

de la información del operador humano, como la relación señal-ruido en las variables observadas y de control y los retrasos de tiempo sensorial-motor, junto con una función objetiva que se supone que el ser humano está minimizando en la tarea en cuestión, puede conducir al cálculo directo de la dinámica y el remanente del operador humano lineal. Además, esta capacidad algorítmica no se limita a los sistemas SISO, sino que también se puede extender al control humano de sistemas de múltiples entradas y múltiples salidas (MIMO).

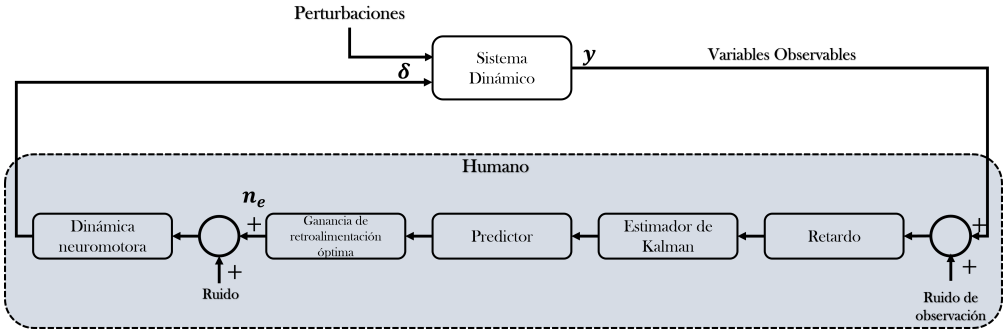

Figura 3.4: El modelo de control óptimo del operador humano[50]

La figura 12.4 muestra la estructura básica de OCM. Centrándonos por el momento en un sistema SISO, los elementos de la figura 12.4, desde el retardo de tiempo hasta la dinámica neuromuscular, forman la dinámica del operador humano. Estrictamente hablando, el OCM nunca es un modelo SISO, porque una hipótesis fundamental en la formulación del modelo es que, si se muestra una variable $d(t)$ al operador, su derivada en el tiempo $\dot{d}(t)$ también se detecta, ambas señales se corrompen con ruido de observación blanco. Por lo tanto, en su forma más simple, $y(t)$ en la figura 12.4 es un vector de columna cuyos dos elementos son la señal mostrada y su derivada en el tiempo. Asimismo, $v_y(t)$ en la figura 12.4 es también un vector de columna cuyos elementos son las señales de ruido de observación. Se supone que las covarianzas de estas señales de ruido de observación escalan con las covarianzas de las señales visualizadas / observadas $y(t)$ y $\dot{y}(t)$. Además del ruido de observación, también se supone que la señal $u_c(t)$ está corrompida con ruido de "motor". Este ruido proporciona predicciones de rendimiento que son más realistas que las que se producen cuando no hay ruido del motor.

La figura 12.4 muestra la estructura básica de OCM. Centrándonos por el momento en un sistema SISO, los elementos de la figura 12.4, desde el retardo de tiempo hasta la dinámica neuromuscular, forman la dinámica del operador humano. Estrictamente hablando, el OCM nunca es un modelo SISO, porque una hipótesis fundamental en la formulación del modelo es que, si se muestra una variable $d(t)$ al operador, su derivada en el tiempo $\dot{d}(t)$ también se detecta, ambas señales se corrompen con ruido

de observación blanco. Por lo tanto, en su forma más simple, $y(t)$ en la figura 12.4 es un vector de columna cuyos dos elementos son la señal mostrada y su derivada en el tiempo. Asimismo, $v_y(t)$ en la figura 12.4 es también un vector de columna cuyos elementos son las señales de ruido de observación. Se supone que las covarianzas de estas señales de ruido de observación escalan con las covarianzas de las señales visualizadas / observadas $y(t)$ y $\dot{y}(t)$. Además del ruido de observación, también se supone que la señal $u_c(t)$ está corrompida con ruido de "motor". Este ruido proporciona predicciones de rendimiento que son más realistas que las que se producen cuando no hay ruido del motor.

3.5.3. Quasi-linear model

El modelo cuasi-lineal se desarrolló a partir del hecho de que la mayoría de los sistemas no lineales tienen respuestas similares a entradas específicas en comparación con las respuestas de sistemas lineales equivalentes a las mismas entradas. Para una combinación de entrada-sistema no lineal dada, la respuesta del sistema no lineal se puede dividir en dos partes; un componente que corresponde a la respuesta de un elemento lineal equivalente impulsado por esa entrada y una cantidad adicional, denominada remanente, que representa la diferencia entre la respuesta del elemento lineal real y equivalente [48, 73].

$$F_H(s) = K_H \cdot \frac{T_L s + 1}{T_I s + 1} \cdot \frac{1}{\frac{s^2}{\omega_N^2} + \frac{2\zeta_N}{\omega_N} s + 1} \cdot e^{-\tau s}$$

Aquí, K es la ganancia humana, T es el retraso debido al tiempo de reacción humano, T_L es la constante de tiempo de espera, T_I es la constante de tiempo de retraso y T_N es la constante de dinámica neuromotora. Este modelo también se conoce como modelo cruzado ya que el desempeño del ser humano basado en este modelo depende de la frecuencia cruzada ω_c. A esta frecuencia, la función de transferencia de lazo abierto satisface. Este modelo también se conoce como crossover model.

El *crossover model* se basa en el siguiente hecho comprobable experimentalmente: En un diagrama de Bode que representa la transmisión en lazo $Y_p(j\omega) \cdot Y_p(jc)$ del sistema, como se muestra en la Figura 12.2 , el ser humano adopta características dinámicas $Y_p(j\omega)$ para que

$$Y_p(j\omega) \cdot Y_c(j\omega) \approx \frac{\omega_c e^{-\tau_e \omega}}{j\omega} \tag{3.1}$$

La *crossover frequency*, ω_c se define como la frecuencia donde $\|Y_p Y_c(j\omega)\| = 1.0$. La ecuación 3.1 es válida en un amplio rango de frecuencias (1 a 1.5 décadas) alrededor de la frecuencia de cruce ω_c. El factor τ_e, referido como un retraso de tiempo efectivo,

representa el efecto acumulativo de los retrasos de tiempo reales en el sistema de procesamiento de información humana (por ejemplo, tiempos de detección visual, tiempos de conducción neural, etc.), los efectos de baja frecuencia de la dinámica del operador humano de frecuencia hisber (por ejemplo, , dinámica de actuación muscular), y dinámica de frecuencia más alta en el propio elemento controlado. Aquí, "frecuencia más alta- se refiere a frecuencias que están por encima de ω_c.

Asociado con la ecuación 3.1 es un modelo de $\Phi_{n_e n_{e(\omega)}}$ la densidad espectral de potencia del remanente inyectado por error. Una vez más, una amplia evidencia experimental sugiere la siguiente forma:

$$\Phi_{n_e n_e}(\omega) \approx \frac{Re^{-2}}{\omega^2 + \omega_R^2} \tag{3.2}$$

La razón para comenzar esta discusión con el modelo cruzado es que es básico para el modelado de control manual. Cualquier modelo válido del operador humano en tareas continuas con entradas de apariencia aleatoria debe exhibir las características de la Ecuación 3.1.

La transmisión en realimentación prescrita por la ecuación 3.1 es similar a la que seleccionaría un diseñador de sistemas de control experimentado en una síntesis en el dominio de la frecuencia de un sistema de control con un elemento de compensación inanimado y requisitos de rendimiento similares a los del sistema controlado manualmente [74].

Objetivo de diseño: El objetivo del controlador de bucle externo específico de la tarea es encontrar los valores óptimos de los parámetros de impedancia prescritos $\bar{B}, \bar{K}$, la ganancia humana K_h (O $\bar{M}$ si $K_h = 1$), y la entrada auxiliar $\bar{l}(x_d)$ en 4.21 para minimizar el esfuerzo de control humano f_h y optimizar el rendimiento del seguimiento en función de la tarea.

3.6. Método de control de bucle externo específico de la tarea: Método LQR

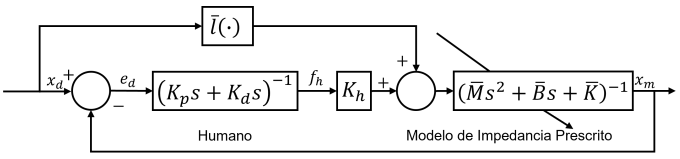

Figura 3.5: Interfaz hombre-robot en el bucle externo específico de la tarea

El diagrama de bloques del controlador de tareas de bucle externo se muestra en la figura 2 y se muestra en detalle en la figura 5. Como se muestra en la figura 5, además del bucle de impedancia adaptativa que especifica los parámetros de impedancia óptimos, una entrada auxiliar de realimentación y se emplea una ganancia de fuerza humana para ayudar al ser humano a minimizar el error de seguimiento. El término *feedforward* $\bar{l}(x_d)$ en 4.21 está diseñado para hacer que el error de seguimiento de estado estacionario llegue a cero. La ganancia humana K_h y los valores óptimos de los parámetros de impedancia prescritos $\bar{K}$ y $\bar{B}$ en 4.21 se determinan para minimizar el esfuerzo humano y el error de seguimiento para una tarea determinada.

A continuación, se muestra cómo el problema de encontrar los valores óptimos de $\bar{B}, \bar{K}$ y K_h se transforma en un problema LQR, y cómo estos parámetros se obtienen mediante resolver una ecuación algebraica de Riccati (ARE). Definir el error de seguimiento

$$e_d = x_d - x_m \in \mathbb{R}^n \tag{3.3}$$

y

$$\bar{e}_d = \left[e_d^T \dot{e}_d^T\right]^T = \bar{x}_d - \bar{x} \in \mathbb{R}^{2n} \tag{3.4}$$

con

$$\bar{x} = \left[\begin{array}{cc} x_m^T & \dot{x}_m^T \end{array}\right]^T \in \mathbb{R}^{2n} \tag{3.5}$$

y

$$\bar{x}_d = \left[x_d^T \dot{x}_d^T\right]^T \in \mathbb{R}^{2n}. \tag{3.6}$$

En función de este error de seguimiento, defina el índice de rendimiento

$$J = \int_t^\infty \left(\bar{e}_d^T Q_d \bar{e}_d + f_h^T Q_h f_h + u_e^T R u_e\right) d\tau \tag{3.7}$$

donde $Q_d = Q_d^T > 0, Q_h = Q_h^T > 0, R = R^T > 0$, y u_e es la entrada de control de retroalimentación que depende linealmente del error de seguimiento $\bar{e}_d$ y el esfuerzo humano f_h. Entonces

$$u_e = K_1 \bar{e}_d + K_2 f_h. \tag{3.8}$$

En el Teorema 2 se muestra que la entrada de control 3.8 tiene dos componentes. El primer componente, es decir, K_1 sintoniza los parámetros de impedancia prescritos $\bar{B}$ y $\bar{K}$ y el segundo componente, es decir, K_2 sintoniza la ganancia de control humano K_h (o $\bar{M}$ y $K_h = 1$).

Remark 3: Teniendo en cuenta que al minimizar el índice de rendimiento 3.7, ambos errores de seguimiento $\bar{e}_d$ y esfuerzo humano f_h se minimizan. Al definir el estado

aumentado

$$X = \begin{bmatrix} \bar{e}_d \\ f_h \end{bmatrix} \in \mathbb{R}^{3n} \tag{3.9}$$

el índice de rendimiento 3.7 se puede escribir como

$$J = \int_t^\infty \left(X^T Q X + u_e^T R u_e \right) d\tau \tag{3.10}$$

donde $Q = \text{diag}\left(Q_d, Q_h\right)$ y $u_e = KX$ con $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$. Ahora se dan las dinámicas del sistema con el estado aumentado 3.9. Utilizando 4.21, se tiene

$$\dot{\bar{x}} = \begin{bmatrix} 0 & I_{n\times n} \\ 0 & 0 \end{bmatrix} \bar{x} + \begin{bmatrix} 0 \\ I_{n\times n} \end{bmatrix} u \equiv A_q \bar{x} + B_q u \tag{3.11}$$

donde $\bar{x}$ se define en 3.5, y

$$u = \bar{M}^{-1}\left(-K_q \bar{x} + K_h f_h\right) + \bar{M}^{-1}\bar{l}\left(x_d\right) \tag{3.12}$$

con

$$K_q = \begin{bmatrix} \bar{K} & \bar{B} \end{bmatrix} \tag{3.13}$$

donde $K_q \in \mathbb{R}^{n\times 2n}, \bar{B}, \bar{K}$, y $\bar{M}$ son el modelo de impedancia prescrito en 4.21. Por otro lado, basado en el modelo humano 4.39, tenemos

$$\left(K_d s + K_p\right) f = k_e e_d \tag{3.14}$$

que se puede escribir en el dominio del tiempo como

$$K_d \dot{f}_h + K_p f_h = k_e e_d \tag{3.15}$$

o equivalente

$$\dot{f}_h = -K_d^{-1} K_p f_h + k_e K_{d,0} \bar{e}_d \equiv A_h f_h + E_h \bar{e}_d \tag{3.16}$$

donde $K_{d,0} = \begin{bmatrix} K_d^{-1} & 0 \end{bmatrix} \in \mathbb{R}^{n\times 2n}$ y $\bar{e}_d$ se define en 3.4.

El siguiente teorema muestra cómo el problema de encontrar los parámetros óptimos del modelo de impedancia prescrito y la ganancia humana se obtienen resolviendo un problema LQR.

Observación 4: Teniendo en cuenta que en [75], la función de transferencia humana de e_d a f_h se consideró como

$$G(s) = \frac{K_d s + K_p}{Ts + 1} \tag{3.17}$$

Para este caso, A_h y B_h en 3.16 convertirse $A_h = -T^{-1}$ y $E_h = T^{-1}\begin{bmatrix} K_p & K_d \end{bmatrix}$

Teorema 2: Considere el modelo de impedancia del robot prescrito 4.21. Con base en la dinámica en 3.11 y 3.16, defina las matrices aumentadas A y B por

$$A = \begin{bmatrix} A_q & 0 \\ E_h & A_h \end{bmatrix}, B = \begin{bmatrix} B_q \\ 0 \end{bmatrix} \tag{3.18}$$

Definimos

$$K = \begin{bmatrix} K_q & K_h \end{bmatrix} \in \mathbb{R}^{n \times 3n} \tag{3.19}$$

como la matriz de los parámetros de impedancia y la ganancia humana. Entonces, el valor óptimo de K que minimiza el índice de rendimiento 3.7 está dado por

$$K = -\bar{M} R^{-1} B^T P \tag{3.20}$$

donde P es la solución del ARE

$$0 = A^T P + PA + PBR^{-1}B^T P + Q \tag{3.21}$$

Entonces, el control de retroalimentación óptimo está dado por

$$u_e = \bar{M}^{-1}\bar{K}e_d + \bar{M}^{-1}\bar{B}\dot{e}_d + \bar{M}^{-1}K_h f_h \tag{3.22}$$

Prueba: Manipulando 3.12 da

$$\begin{aligned} u &= \bar{M}^{-1}\left(K_q \bar{e}_d + K_h f_h\right) + M^{-1}\left(\bar{l}\left(x_d\right) - K_q \bar{x}_d\right) \\ &\equiv u_e + u_d \end{aligned} \tag{3.23}$$

donde $\bar{e}_d$ y $\bar{x}_d$ se definen en 3.4 y 3.6, y

$$u_e = \bar{M}^{-1}\left(K_q \bar{e}_d + K_h f_h\right) \tag{3.24}$$

es una entrada de control de retroalimentación, y

$$u_d = M^{-1}\left(\bar{l}\left(x_d\right) - K_q \bar{x}_d\right) \tag{3.25}$$

es una entrada de control feedforward. El estado estacionario o el término feedforward se utiliza para garantizar un seguimiento perfecto. Es decir, en el estado estacionario se tiene

$$\dot{\bar{x}}_d = A_q \bar{x}_d + B_q u_d \tag{3.26}$$

donde $\bar{x}_d$ se define en 3.6. Por lo tanto

$$\bar{l}\left(x_d\right) = \bar{M}u_d + K_q \bar{x}_d = \bar{M}B_q^{-1}\left(\dot{\bar{x}}_d - A_q \bar{x}_d\right) + K_q \bar{x}_d. \tag{3.27}$$

Tomando la derivada de $\bar{e}_d$ y usando 3.11 y 3.26, y algunas manipulaciones da

$$\dot{\bar{e}}_d = A_q \bar{e}_d + B_q u_e. \tag{3.28}$$

Usando el estado aumentado 3.9, y usando 3.16 y 3.28 uno tiene

$$\begin{aligned} \dot{X} &= \begin{bmatrix} \dot{\bar{e}}_d \\ \dot{f}_h \end{bmatrix} = \begin{bmatrix} A_q & 0 \\ E_h & A_h \end{bmatrix} \begin{bmatrix} \bar{e}_d \\ f_h \end{bmatrix} + \begin{bmatrix} B_q \\ 0 \end{bmatrix} u_e \\ &\equiv AX + Bu_e \end{aligned} \tag{3.29}$$

La entrada de control u_e en términos del estado aumentado se puede escribir como

$$u_e = \bar{M}^{-1} \left(K_q \bar{e}_d + K_h f_h \right) = \bar{M}^{-1} K X. \tag{3.30}$$

Encontrar el control de retroalimentación óptimo 3.30 para minimizar el índice de desempeño 3.7 sujeto al sistema aumentado 3.29 es un problema LQR y su solución está dada por [76]

$$u_e^* = -R^{-1} B^T P X \tag{3.31}$$

donde P es la solución a la ecuación de Riccati 3.21. Igualando los lados derechos de 3.30 y 3.31 se obtiene

$$K = \begin{bmatrix} K_q & K_h \end{bmatrix} = -\bar{M} R^{-1} B^T P \tag{3.32}$$

Esto completa la prueba.

Observación 5: El vector K definido en 3.19 incluye ambos parámetros 3.13 del modelo de impedancia del robot y la ganancia K_h de la fuerza humana. Por lo tanto, la solución al problema LQR formulado proporciona los valores óptimos de los parámetros del modelo de impedancia prescritos y la ganancia de la fuerza del operador humano. Si la ganancia humana no se puede aumentar para una aplicación HITL específica, es decir, si $K_h = 1$, entonces, según 3.30, se puede establecer el coeficiente de f_h en la entrada de control como $\bar{M}^{-1}$ y luego busque $\bar{M}$ en lugar de K_h. Es decir, si $K_h = 1$ y $\bar{M}$ son desconocidos, entonces 3.32 se convierte en $K = \left[\bar{M}^{-1} K_q \bar{M}^{-1} \right] = -R^{-1} B^T P$, lo que da parámetros desconocidos del modelo de impedancia 4.21.

Observación 6: El diseño de control de bucle externo consta de dos componentes: 1) un componente de impedancia adaptable que encuentra los valores óptimos de los parámetros 3.13 del modelo de impedancia prescrito y 2) un componente de asistencia que incluye la ganancia de fuerza humana K_h y el término feedforward $\bar{l}\left(x_d\right)$ para ayudar al humano a minimizar el error de seguimiento.

3.7. Aprendizaje de los parámetros óptimos del modelo de impedancia prescrito mediante el aprendizaje por refuerzo integral

Resolver 3.21 requiere el conocimiento de la matriz A en 3.18 y consecuentemente el conocimiento del modelo humano. Se han desarrollado varios algoritmos de RL sin modelo para resolver el control óptimo de sistemas lineales sin necesidad de ningún conocimiento de la dinámica del sistema [77–82]. En este escrito, se utiliza el algoritmo integral RL (IRL) fuera de política [79–82] para resolver el problema LQR dado. El IRL es un algoritmo iterativo de iteración de políticas para resolver 3.21 que consta de dos pasos de iteración: 1) evaluación de políticas y 2) mejora de políticas. En el paso de evaluación de la política, la función de valor relacionada con una política fija se evalúa utilizando una ecuación IRL Bellman [ver 3.34] que no involucra la dinámica del sistema. En el paso de mejora de políticas, se encuentra una política mejorada utilizando el valor obtenido en el paso de evaluación de políticas.

Para garantizar una exploración suficiente del espacio de estado, que es crucial para una convergencia adecuada a la función de valor óptimo, se agrega a la entrada de control un pequeño ruido de sondeo exploratorio que consta de sinusoides de frecuencias variables para satisfacer cualitativamente la excitación persistente (PE) [83, 84]. Considere el sistema 3.29 explorado por una señal de prueba variable en el tiempo conocida e_τ

$$\dot{X} = AX + B\left[u_e + e_\tau\right]. \tag{3.33}$$

La ecuación IRL Bellman [79, 80] utiliza solo la información proporcionada al medir el estado del sistema y una integral de la función de utilidad en intervalos de refuerzo finitos para evaluar una política de control. La ecuación de IRL Bellman para el problema de LQR dado para el sistema 3.33 incluido el ruido de sondeo se proporciona, para el intervalo de tiempo $\Delta t > 0$, mediante [82]

$$\begin{aligned} X(t)^T PX(t) + \int_t^{t+\Delta t} \left[2X(t)^T PBe_\tau\right] d\tau \\ = \int_t^{t+\Delta t} \left[X(t)^T QX(t) + u_e^T Ru_e\right] d\tau \\ + X(t+\Delta t)^T PX(t+\Delta t) \end{aligned} \tag{3.34}$$

Esta ecuación contiene explícitamente el ruido de sondeo y se denomina ecuación de Bellman fuera de la política. Usando 3.34 para el paso de evaluación de políticas y una ley de actualización en forma de 3.31 para encontrar una política mejorada, se obtiene el siguiente algoritmo exploratorio basado en IRL para resolver 3.21.

Teniendo en cuenta que la señal de sondeo e_τ en 3.33 debe aplicarse durante el aprendizaje para asegurar la convergencia del Algoritmo 1. Sin embargo, después de la convergencia, el ruido de sondeo ya no es necesario y puede eliminarse.

Observación 7: Teniendo en cuenta que para el problema LQR, dado que el sistema es lineal y la función de rendimiento es cuadrática, la solución óptima es única y se encuentra resolviendo el ARE 3.21. En [81] y [82] se muestra que el algoritmo IRL 1 fuera de política converge a la solución óptima global encontrada al resolver el ARE 3.21, siempre que el ruido de sondeo sea PE. La inclusión explícita del ruido de sondeo en la ecuación IRL Bellman 3.34 significa que el algoritmo converge sin sesgo, como se muestra en [82].

Observación 8: La solución para P^i en el paso de evaluación de políticas 3.35 generalmente se lleva a cabo en un sentido de mínimos cuadrados (LSs). De hecho, 3.35 es una ecuación escalar y P^i es una matriz $n \times n$ simétrica con $n(n+1)/2$ elementos independientes y por lo tanto al menos $n(n+1/2$ se requieren conjuntos de datos antes de que 3.35 pueda resolverse usando LS. En consecuencia, la complejidad computacional de calcular P^i depende del tamaño del sistema.

Observación 9: Teniendo en cuenta que el Algoritmo 1 resuelve el ARE 3.21 y no requiere el conocimiento de la matriz A que contiene el conocimiento de la dinámica humana. De hecho, la información de A está integrada en la medición en línea de los datos del sistema.

Algorithm 1 Algoritmo IRL en línea para diseño de control de bucle externo

Inicialización: Comience con una entrada de control admisible $u^0 = K_1^0 X$
Evaluación de políticas: Dada una política de control u^i, encuentre P^i usando la ecuación de Bellman fuera de la política

$$\begin{aligned} X(t)^T P^i X(t) + \int_t^{t+\Delta t} \left[2X(t)^T P^i B e_\tau\right] d\tau \\ = \int_t^{t+\Delta t} \left[X(t)^T Q X(t) + u_e^T R u_e\right] d\tau \\ + X(t+\Delta t)^T P^i X(t+\Delta t) \end{aligned} \tag{3.35}$$

Mejora de la política: actualizar la entrada de control usando

$$u_e^{i+1} = -R^{-1} B_1^T P^i X. \tag{3.36}$$

Capítulo 4

Inner Loop

4.1. Aprendizaje por refuerzo para regulador cuadrático lineal de tiempo continuo

El control de retroalimentación óptimo es fundamentalmente un problema de retroceso en el tiempo, ya que para planificar nuestras acciones de control primero debemos mirar hacia adelante a los objetivos finales que queremos lograr al final.

El regulador cuadrático lineal (LQR) es uno de los métodos más básicos y poderosos para diseñar sistemas de control de retroalimentación. En este capítulo derivamos el LQR utilizando dos enfoques: el principio mínimo y la programación dinámica. Vemos que el diseño óptimo de los controles de retroalimentación es fundamentalmente un problema retrospectivo. El principio mínimo muestra que el diseño de retroalimentación LQR se basa en la solución de una ecuación matricial conocida como ecuación algebraica de Riccati (ARE), que requiere un conocimiento completo de la dinámica del sistema. La programación dinámica es una solución inversa para el control óptimo que no se puede implementar avanzando en tiempo real. Luego revelamos que una técnica de aprendizaje automático conocida como aprendizaje por refuerzo permite resolver el diseño LQR sin resolver el ARE y sin conocer la dinámica completa del sistema.

La importancia del aprendizaje por refuerzo es que proporciona un método avanzado en el tiempo para aprender controles óptimos en línea en tiempo real mediante la observación de datos medidos a partir de las entradas y salidas del sistema.

El regulador cuadrático lineal (LQR) es un pilar en el diseño de sistemas de control de retroalimentación y fue presentado por Rudolph Kalman en 1960 en su artículo [85]. En ese artículo, Kalman mostró la importancia de usar una descripción de espacio de estado para capturar el comportamiento interno del sistema en lugar de la descripción de función de transferencia clásica que solo captura el comportamiento de entrada-salida.

Además, introdujo la noción de que el diseño óptimo para minimizar una función de rendimiento prescrita (que podría ser, por ejemplo, una energía) brinda una técnica para obtener un rendimiento garantizado en términos de estabilidad, que no es fácil de obtener utilizando funciones de transferencia. En su artículo, presentó un método de diseño sencillo para obtener el LQR, que es un controlador de retroalimentación que minimiza el índice de rendimiento. El diseño LQR se basa en resolver una ecuación cuadrática matricial conocida como ecuación de Riccati

4.1.1. El regulador cuadrático lineal (LQR)

El regulador cuadrático lineal (LQR) se basa en la dinámica del sistema lineal y un índice de rendimiento cuadrático. Deje que una planta o sistema se describa mediante la dinámica lineal del espacio de estado invariante en el tiempo

$$\dot{x} = Ax + Bu \tag{4.1}$$

con estado $x(t) \in R^n$ y control $u(t) \in R^m$. Se supone que la matriz de dinámica del sistema A y la matriz de entrada B se conocen para el diseño LQR. Definir un índice de rendimiento cuadrático de horizonte infinito

$$V(x(t)) = \int_t^\infty r(x,u)d\tau = \frac{1}{2}\int_t^\infty \left(x^T Q x + u^T R u\right) d\tau \tag{4.2}$$

con $Q = Q^T \geq 0, R = R^T > 0$. Este índice de rendimiento incluye la energía $x^T Q x$ del estado y la energía $u^T R u$ de la entrada. (Recuerde que la energía es una forma cuadrática, por ejemplo, la energía cinética del movimiento con velocidad $v(t)$ es $K = \frac{1}{2}mv^2$.) Las matrices Q, R son parámetros de diseño que selecciona el ingeniero de diseño para equilibrar la ponderación de la energía de estado y la energía de control. Los próximos ejemplos aclararán esto.

Problema de control óptimo. El problema del regulador cuadrático lineal es encontrar la entrada de control $u(t)$ que minimice el índice de rendimiento $V(x(t))$ a medida que el estado se mueve a lo largo de las trayectorias prescritas por la dinámica (4.1).

Estabilidad y Detectabilidad. Se dice que el par de matrices (A, B) es estabilizable si existe una entrada de control $u(t)$ tal que el estado tiende a cero, es decir, $x(t) \to 0$ cuando el tiempo $t \to \infty$ en (4.1). Definiendo una salida $y = \sqrt{Q}x$ para la dinámica (4.1), se dice que el par $(A, \sqrt{Q})$ es detectable si la salida $y(t) \to 0$ como $t \to \infty$ implica que el estado completo $x(t) \to 0$. La condición de detectabilidad significa que todas las excursiones de estado que se alejan de cero son finalmente perceptibles a través del índice

de rendimiento. Controlabilidad y Observabilidad.

Optimalidad versus estabilidad. Una entrada de control continua $u(t)$ que minimiza $V(x(t))$ se llama óptima. Un requisito más moderado es que $u(t)$ sea estabilizador. Esta estabilidad significa que $u(t)$ aplicado a la dinámica (4.1) da como resultado un estado $x(t)$ que tiende a cero con el tiempo t.

Supongamos que $u(t)$ es continuo y óptimo. Entonces $u(t)$ produce un valor mínimo de $V(x(t))$ en (4.2. Entonces la integral infinita $V(x(t))$ toma un valor finito y el integrando es continuo. Por lo tanto, el integrando $x^TQx+u^TRu = y^Ty+u^TRu$ tiende a cero. Como $R>0$, tanto $y(t)=\sqrt{Q}x$ como $u(t)$ llegan a cero con el tiempo. Esto implica que ingrese $u(t)\to 0$, y $x(t)\to 0$ si $(A,\sqrt{Q})$ es detectable. En consecuencia, la optimalidad implica estabilidad en estas condiciones.
Admisibilidad. Se dice que la entrada de control $u(t)$ es admisible si es continua, estabiliza el sistema (4.1) y produce un valor finito de $V(x(t))$ en (4.2).
Función de valor. Si $u(t)$ es cualquier control estabilizador, entonces la función asociada $V(x(t))$ definida por (4.2) es finita para la dinámica (4.1) y se conoce como la función de valor. Representa $V(x(t))$ calculado a lo largo de las trayectorias $x(t)$ dadas por (4.1) cuando la entrada es $u(t)$. Se denomina como el valor de usar ese control.
Hay dos enfoques básicos para resolver el problema de control óptimo y encontrar el control LQR óptimo: el Principio Mínimo y la Programación Dinámica. Ahora consideraremos ambos.

4.1.1.1. LQR por principio mínimo

Para encontrar el LQR por el Principio Mínimo, el primer paso es diferenciar el índice de desempeño.
Fórmula de Leibniz. Recuerda que, dada una función

$$\Phi(z,t)=\int_{\alpha(t)}^{\beta(t)}\varphi(z,t)dz \tag{4.3}$$

su derivada temporal viene dada por la fórmula de Leibniz

$$\frac{d}{dt}\Phi(x,t)=\beta'\varphi(\beta,t)-\alpha'\varphi(\alpha,t)+\int_{\alpha}^{\beta}\frac{\partial}{\partial t}\varphi(z,t)dz \tag{4.4}$$

donde prima denota derivada del tiempo.
Bellman Equation Usando la fórmula de Leibniz (4.4) sobre (4.2) se obtiene

$$\dot{V}(x(t))=-\frac{1}{2}\left(x(t)^TQx(t)+u(t)^TRu(t)\right) \tag{4.5}$$

Por otro lado, usando la regla de la cadena se puede expresar $\dot{V}(x(t))$ en términos de $\dot{x}$ de modo que

$$\dot{V}(x(t)) + \frac{1}{2}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right) = \left(\frac{\partial V}{\partial x}\right)^T \dot{x} + \frac{1}{2}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right) = 0 \tag{4.6}$$

Este objeto es lo suficientemente importante como para merecer su propio nombre. En consecuencia, definiendo la función hamiltoniana $H(x, \nabla V, u)$ y sustituyendo de (4.1) se escribe la ecuación de Bellman

$$H(x, \nabla V, u) \equiv \nabla V^T(x(t))(Ax(t) + Bu(t)) + \frac{1}{2}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right) = 0 \tag{4.7}$$

donde el gradiente se define como $\nabla(\mathrm{V(x)}) = \partial \mathrm{V}/\partial x$. Esta es una ecuación diferencial parcial (PDE). Su condición inicial es $V(x(0)) = 0$ con $t = 0$ el tiempo inicial.

La función hamiltoniana es la cantidad central en la dinámica hamiltoniana en física. Combina la dinámica del sistema (4.1) y los requisitos de desempeño (4.2) en un solo objeto. En Control Óptimo, generalmente no se requiere que la Función Hamiltoniana sea igual a cero [Lewis opt]. Sin embargo, si es igual a cero, entonces las ecuaciones (4.7) y la combinación de (4.1) y (4.2) son equivalentes. Un resultado formal muestra que, dado un control estabilizador continuo $u(t)$, el $V(x(t))$ encontrado al resolver la EDP (4.7) es igual a $V(x(t))$ encontrado al integrar la dinámica (4.1) para un horizonte de tiempo infinito y evaluar (4.2).

El punto es que al resolver PDE (4.7) no necesitamos simular el sistema (4.1) sobre un horizonte de tiempo infinito para evaluar su desempeño, o para controlar el sistema actual correspondiente a (4.1) en tiempo real. tiempo en un horizonte de tiempo infinito. Esto destaca la extrema importancia de (4.7) y justifica darle un nombre: la Ecuación de Bellman [Bellman]. La importancia de la ecuación de Bellman no se explota en el procedimiento de solución LQR estándar. Veremos en apartados posteriores que es la base del Aprendizaje por Refuerzo.

Hamilton-Jacobi-Bellman (HJB) Equation La condición de estacionariedad establece que el control óptimo se encuentra minimizando $H(x, \nabla V, u)$. Este es un caso especial del Principio Mínimo de Pontryagin. Para minimizar (4.7) las condiciones de estacionariedad se requiere que

$$\frac{\partial H(x, \nabla V, u)}{\partial u} = B^T \nabla V(x(t)) + Ru(t) = 0 \tag{4.8}$$

de modo que

$$u(t) = -R^{-1}B^T\nabla V(x(t)) \tag{4.9}$$

Ahora sustituya este control en la ecuación de Bellman (4.7) para obtener la ecuación de Hamilton-Jacobi Bellman (HJB)

$$\nabla V^T(x(t))\left(Ax(t) - BR^{-1}B^T\nabla V(x(t))\right) + \frac{1}{2}\left(x(t)^TQx(t) + \nabla V^T\left(x(t)BR^{-1}B^T\nabla V(x(t))\right)\right) = 0$$

o

$$\nabla V^T(x(t))Ax(t) + \frac{1}{2}x(t)^TQx(t) - \frac{1}{2}\nabla V^T(x(t))BR^{-1}B^T\nabla V(x(t)) = 0 \tag{4.10}$$

La ecuación HJB es una PDE cuadrática con $V(x(0)) = 0$ que debe resolverse para $V(x(t))$. Su condición inicial es $V(x(0)) = 0$ con $t = 0$ el tiempo inicial.

Nótese que el hessiano está dado por

$$\frac{\partial^2 H(x, \nabla V, u)}{\partial u^2} = R > 0 \tag{4.11}$$

de manera que la condición de estacionariedad arroja un valor mínimo de $H(x, \nabla V,\, u)$, no un valor máximo.

Un teorema dice que si la ecuación HJB (4.10) se resuelve para una solución continua $V^*(x(t))$, entonces el control (4.9) es continuo y minimiza (4.2). Además, el valor mínimo de (4.2) viene dado por $V^*(x(t))$. Finalmente, como $V^*(x(t))$ es finito, el control (4.9) estabiliza el sistema (4.1).

Solución LQR y ecuación algebraica de Riccati (ARE) Para el LQR, el método de barrido [86] establece que la función de valor tiene la forma cuadrática especial

$$\mathrm{V}(x(t)) = \frac{1}{2}x(t)^TPx(t) \tag{4.12}$$

para alguna matriz $P = P^T$. En consecuencia, la PDE HJB se puede simplificar. Nótese que en este caso $\nabla V = \partial V/\partial x = Px$, de modo que sustituyendo (4.12) en (4.10) se obtiene

$$x(t)^TPAx(t) + \frac{1}{2}x(t)^TQx(t) - \frac{1}{2}x^T(t)PBR^{-1}B^TPx(t) = 0 \tag{4.13}$$

o

$$\frac{1}{2}x(t)^T\left(A^TP + PA + Q - PBR^{-1}B^TP\right)x(t) = 0 \tag{4.14}$$

(Tenga en cuenta que $x^TPAx = \frac{1}{2}x^T\left(PA + A^TP\right)x$.)

Ahora asumimos que esta ecuación se cumple para todas las condiciones iniciales $x(0)$ en (4.1) y, por lo tanto, para todas las trayectorias de estado $x(t)$. Entonces se tiene la

Ecuación Algebraica de Riccati (ARE)

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{4.15}$$

Regulador cuadrático lineal

Ecuación algebraica de Riccati (ARE)

$$A^T P + PA + Q - PBR^{-1}B^T P = 0$$

LQR Optimal Feedback Gain

$$u = -R^{-1}B^T Px \equiv -Kx$$

Así, dada la forma cuadrática (4.12), la PDE HJB (4.10) se ha convertido en la ecuación matricial cuadrática (4.15). Tenga en cuenta que si (4.12) se cumple, de acuerdo con (4.9), el control óptimo LQR es

$$u(t) = -R^{-1}B^T Px(t) \tag{4.16}$$

Nuestros resultados se resumen en la Tabla 2.1.1 y en el siguiente resultado, que consideramos suficientemente importante para formular como Teorema. Comprender la prueba es un factor importante para comprender el LQR.

Teorema 2.1.1. Regulador cuadrático lineal. Dada la dinámica lineal (4.1) y el índice de rendimiento cuadrático (4.2), suponga que (A, B) es estabilizable y $(A, \sqrt{Q})$ es detectable. Entonces el ARE (2.1.17) tiene una única solución definida positiva $P = P^T > 0$ y el control dado por la retroalimentación de estado (2.1.18) minimiza el índice de rendimiento (4.2). Además, el valor mínimo de (4.2) viene dado por $V = \frac{1}{2}x^T Px$. Finalmente, la dinámica de lazo cerrado

$$\dot{x} = Ax + Bu = (A - BK)x \tag{4.17}$$

son estables.

4.1.1.2. LQR mediante Programación Dinámica

Acabamos de derivar el LQR usando el Principio Mínimo. Una segunda forma de derivar el LQR es usando Programación Dinámica (DP) [86]. DP se basa en el Principio de Optimalidad.

Bellman's Principle of Optimality Una de las propiedades más fundamentales de una política de control de optimización es la siguiente.

Principio de Optimalidad de Bellman. Una política óptima tiene la propiedad de que no importa cuál haya sido la decisión anterior (es decir, los controles), las decisiones

restantes deben constituir una política óptima con respecto al estado resultante de esas decisiones anteriores.

El principio de optimización de Bellman implica que las estrategias de control óptimas deben determinarse trabajando hacia atrás desde la etapa final. De hecho, el problema de control óptimo es inherentemente un problema de retroceso en el tiempo. Veremos que, de hecho, el Principio de Optimalidad sirve para limitar el número de estrategias de control potencialmente óptimas que deben investigarse.

Dada la dinámica del sistema (4.1),

$$\dot{x} = Ax + Bu \tag{4.18}$$

definir el índice de rendimiento cuadrático de horizonte finito o costo

$$V(x(t), t) = \phi(x(T), T) + \frac{1}{2}\int_t^T \left(x^T Q x + u^T R u\right) d\tau \tag{4.19}$$

donde el intervalo de tiempo de interés es el intervalo finito $[t, T]$ y $\phi(x(T), T)$ es una ponderación de estado final que debe ser pequeña en el tiempo final T. Por ejemplo $\phi(x(T), T) = 1/2x^T(T)\,\mathrm{Px}(T)$ con matriz de ponderación $P = P^T > 0$ busca minimizar la energía del estado final $x(T).V(x(t))$ se conoce como el costo de ida desde el momento t. Tenga en cuenta que ahora $V(x, t)$ es una función explícita del tiempo t.

Problema de control óptimo de horizonte finito: El problema del regulador cuadrático lineal de horizonte finito consiste en encontrar la entrada de control $u(t)$ en el intervalo de tiempo $[t, T]$ que minimiza el índice de rendimiento $V(x(t), t)$ en (4.19) a medida que el estado se mueve a lo largo de las trayectorias prescritas por la dinámica (4.18).

Para descubrir el Principio de Optimalidad de Bellman para este problema, considere el tiempo actual t y un tiempo futuro $t+\Delta t$ cercano a t. Entonces el costo de ida se puede escribir como

$$V(x(t), t) = \phi(x(T), T) + \frac{1}{2}\int_{t+\Delta t}^T \left(x^T Q x + u^T R u\right) d\tau + \frac{1}{2}\int_t^{t+\Delta t} \left(x^T Q x + u^T R u\right) d\tau$$

o

$$V(x(t), t) = \frac{1}{2}\int_t^{t+\Delta t} \left(x^T Q x + u^T R u\right) d\tau + V(x(t+\Delta t), t+\Delta t)$$

donde $x + \Delta x$ es el estado en el momento $t + \Delta t$ cuando se usan las corrientes $x(t)$ y $u(t)$ en (4.18). Tenga en cuenta que, a primer orden

$$\Delta x = (Ax + Bu)\Delta t$$

Denote con un 'asterisco' el costo óptimo $V^*(x,t)$ para ir. La ecuación (2.1.24) describe todos los costos posibles para ir desde el tiempo t hasta el tiempo final T. De acuerdo con el Principio de Optimalidad de Bellman, sin embargo, los únicos candidatos para el costo óptimo $V^*(x,t)$ son los costos $V(x,t)$ que son óptimos desde el tiempo $t+\Delta t$ a T. Supongamos que el costo óptimo $V^*(x+\Delta x, t+\Delta t)$ se conoce para todos los posibles $x+\Delta x$. Supongamos también que el control óptimo ha sido determinado en el intervalo $[t+\Delta t, T]$. Entonces, se obtiene el Principio de Optimalidad para sistemas de tiempo continuo

$$V^*(x(t),t) = u(\tau); \text{mín}\, \tau \leq t+\Delta t \left[\frac{1}{2}\int_t^{t+\Delta t} \left(x^T Q x + u^T R u\right) d\tau + V^*(x(t+\Delta t), t+\Delta t)\right]$$

Time-Varying Hamilton-Jacobi Bellman Equation Para resolver el problema LQR utilizando el principio de optimización, realice una expansión en serie de Taylor de $V^*(x(t+\Delta t), t+\Delta t)$ sobre (x,t) para aproximar (2.1 .26) por

$$V^*(x,t) = u(\tau); t \leq \tau \leq t+\Delta t \left[\frac{1}{2}\left(x^T Q x + u^T R u\right)\Delta t + V^*(x,t) + \left(\frac{\partial V^*}{\partial x}\right)^T \Delta x + \frac{\partial V^*}{\partial t}\Delta t\right]$$

Ahora usa (2.1.25) y observa que $V^*(x,t)$ y $V_t^*\Delta t$ son independientes de $u(\tau); t \leq \tau \leq t+\Delta t$ escribir

$$V^*(x,t) = V^*(x,t) + V_t^*\Delta t + u(\tau); t \leq \tau \leq t+\Delta t \left[\frac{1}{2}\left(x^T Q x + u^T R u\right)\Delta t + \left(\frac{\partial V^*}{\partial x}\right)^T (Ax+Bu)\Delta t\right]$$

o

$$-V_t^*\Delta t = u(\tau); t \leq \tau \leq t+\Delta t \left[\frac{1}{2}\left(x^T Q x + u^T R u\right)\Delta t + \left(\frac{\partial V^*}{\partial x}\right)^T (Ax+Bu)\Delta t\right]$$

Haciendo $\Delta t \to 0$ finalmente se obtiene la ecuación de Hamilton-Jacobi-Bellman (que varía en el tiempo)

$$-\frac{\partial V^*}{\partial t} = u(t)\left[\frac{\text{mín}}{2}\left[x^T Q x + u^T R u\right) + \left(\frac{\partial V^*}{\partial x}\right)^T (Ax+Bu)\right]$$

Esta ecuación HJB es una ecuación diferencial parcial para el costo óptimo $V^*(x,t)$. La ecuación se desarrolla hacia atrás en el tiempo, debido al signo menos en el lado izquierdo. La condición de contorno es la ponderación de estado final $V^*(x(T),T) = \phi(x(T),T)$.

Defina la función hamiltoniana (4.7)

$$H(x, \nabla V, u, t) \equiv \nabla V^T (Ax + Bu) + \frac{1}{2}\left(x^T Q x + u^T R u\right)$$

Tenga en cuenta que ahora la función hamiltoniana es una función explícita del tiempo t. Ahora, uno puede escribir la ecuación HJB como la PDE hacia atrás en el tiempo

$$-\frac{\partial V^*}{\partial t} = u(t)\left[H\left(x, \nabla V^*, u, t\right)\right]$$

Es importante señalar el siguiente hecho. En la derivación del principio mínimo, se requiere que la función hamiltoniana sea igual a cero en la ecuación de Bellman (4.7). Por el contrario, en la derivación de programación dinámica, no se menciona en absoluto que la función hamiltoniana es igual a cero.

Ejercicio 2.1.1. Principio Mínimo HJB y Programación Dinámica HJB. La ecuación HJB encontrada usando el Principio Mínimo es (4.10). La ecuación HJB variable en el tiempo que se encuentra usando Programación Dinámica es (2.1.30). Relaciona los dos escribiendo (2.1.30) en términos de (4.10).

Solución LQR de horizonte finito mediante programación dinámica Para especializar la ecuación HJB al caso LQR, recuerde que en el LQR se tiene la forma cuadrática (4.12) para el costo. Eso es

$$V(x(t), t) = \frac{1}{2} x(t)^T P(t) x(t)$$

donde ahora $P(t)$ varía en el tiempo. El hamiltoniano es ahora

$$H(x, \nabla V, u, t) \equiv x^T P(Ax + Bu) + \frac{1}{2}\left(x^T Q x + u^T R u\right)$$

y el mínimo sobre $u(t)$ requerido en (2.1.32) se obtiene calculando $\partial H/\partial u = 0$ de modo que

$$u(t) = -R^{-1} B^T P(t) x(t) \equiv -K(t) x(t)$$

Y poniendo esto en (2.1.34) para escribir (2.1.32) como

$$-\frac{\partial V^*}{\partial t} = -\frac{1}{2} X^T \dot{P} X = \frac{1}{2} X^T \left(A^T P + PA + Q - PBR^{-1} B^T P\right) x$$

Dado que esto es válido para todas las trayectorias $x(t)$, se tiene el ARE variable en el tiempo

Regulador cuadrático lineal de horizonte finito

Ecuación algebraica de Riccati variable en el tiempo (ARE)

$$-\dot{P} = A^T P + PA + Q - PBR^{-1}B^T P, \quad P(T) \text{ given}$$

Ganancia de retroalimentación óptima variable en el tiempo LQR

$$u = -R^{-1}B^T P(t)x(t) \equiv -K(t)x(t)$$

$$-\dot{P} = A^T P + PA + Q - PBR^{-1}B^T P$$

Debido al signo menos, se desarrolla hacia atrás en el tiempo. Comparar con ARE (2.1.17). En consonancia con (2.1.33), suponga que la ponderación del estado final tiene la forma

$$\phi(x(T), T) = \frac{1}{2}x(T)^T P(T)x(T)$$

para alguna matriz definida positiva dada $P(T)$. Entonces la condición terminal para (2.1.37) es $P(T)$.

El control óptimo LQR variable en el tiempo se encuentra resolviendo primero (2.1.37) y luego calculando (2.1.35) como se resume en la Tabla 2.1.2.

Tenga en cuenta que si el horizonte de tiempo T en (4.19) es infinito, entonces la solución de estado estacionario P en (2.1.38) tiene $\dot{P}(t) = 0$, de modo que satisface el ARE de la Tabla 2.1.1. De hecho, bajo las condiciones del Teorema 2.1.1, (2.1.38) tiene una solución de estado estacionario finito para cada $P(T)$.

4.1.2. Aprendizaje por refuerzo para LQR

La solución LQR requiere que uno resuelva el ARE en la Tabla 2.1.1 y luego calcule la ganancia óptima allí. El ARE es una ecuación matricial que se resuelve fácilmente usando, por ejemplo, la rutina de MATLAB lqr(A, B, Q, R). Un inconveniente importante de este procedimiento de solución es que se debe conocer la información dinámica completa (A, B) en (4.18) para resolver el ARE. Esto requiere la identificación del sistema o, en el caso de las aeronaves, pruebas exhaustivas en el túnel de viento. Los enfoques de programación dinámica como los de la tabla 2.1.2 requieren la solución de ecuaciones diferenciales hacia atrás en el tiempo y no pueden implementarse causalmente avanzando en el tiempo.

El aprendizaje por refuerzo (RL) es un método de aprendizaje automático de la informática que se puede adaptar para aprender la solución al problema LQR en el tiempo midiendo los datos de entrada y salida del sistema que están disponibles en tiempo real a medida que evoluciona la dinámica. La dinámica de sistemas no tiene por qué ser conocida

en RL. Aprendizaje reforzado. El precepto de RL es aplicar una política de control de prueba al sistema, evaluar el resultado del desempeño y, en base a esa evaluación, actualizar el control para mejorar el desempeño.

Desarrollar un enfoque de aprendizaje por refuerzo completo para el diseño de LQR que evite resolver la ecuación HJB (4.10) (equivalentemente, el ARE en la Tabla 2.1.1) y encuentre el control óptimo sin conocer la dinámica (A, B) por Para medir los datos disponibles en tiempo real, necesitamos:

Tres pasos para el aprendizaje por refuerzo:

- Primero, tenemos que evitar resolver la ecuación HJB (4.10). Esto se hace mediante iteración de políticas.

- En segundo lugar, debemos mostrar cómo eliminar la dinámica del sistema en la ecuación de Bellman (4.7). Esto se hace mediante el Aprendizaje por Refuerzo Integral.

- En tercer lugar, debemos mostrar cómo implementar un procedimiento de solución en línea en tiempo real. Esto se hace mediante la aproximación de función de valor.

4.1.2.1. Iteración de políticas para LQR

La clave para desarrollar métodos de Aprendizaje por Refuerzo para LQR es la ecuación de Bellman (4.7) derivada usando el Principio Mínimo. Allí, se requiere que la función hamiltoniana sea igual a cero para proporcionar una ecuación diferencial equivalente a la función de costo (4.2), que está en forma integral. Recuerde que el enfoque de programación dinámica en la Sección 2.1.2 en ninguna parte menciona que la función hamiltoniana debe ser igual a cero.

Iteración de políticas

Establecer $j = 0$. Seleccione una política de control inicial estabilizadora $u^0(t)$.
Iteración j

1. Evaluación de políticas. Dada la política de control $u^j(t)$ resuelva la ecuación PDE Bellman para $V^j(x(t))$

$$H\left(x, \nabla V^j, u\right) \equiv \nabla V^{jT}\left(Ax + Bu^j\right) + \frac{1}{2}\left(x^T Q x + u^{jT} R u^j\right) = 0, \quad V^j(0) = 0$$

2. Política de mejora. Actualizar la política de control usando

$$u^{j+1} = -R^{-1}B^T \nabla V^j$$

Si $V^j = V^{j-1}$ detener. De lo contrario, configure $j = j + 1$ y vaya a 1.

Solución de iteración de políticas de la ecuación HJB

Aquí cumplimos el paso 1 de RL. El ARE de la tabla 2.1.1 es un caso especial de la ecuación HJB (4.10). Para evitar resolver la ecuación HJB, considere la ecuación de Bellman (4.7) y el cálculo de control (4.9):

$$H(x, \nabla V, u) \equiv \nabla V^T(x(t))(Ax(t) + Bu(t)) + \frac{1}{2}\left(x(t)^T Q x(t) + u(t)^T R u(t)\right) = 0$$
$$u(t) = -R^{-1}B^T \nabla V(x(t))$$

Ahora, considere el siguiente procedimiento iterativo. Seleccione un control estabilizador $u(t)$. Resuelve la ecuación de Bellman (2.2.3) para $V(x(t))$. Luego actualice el control de acuerdo con (2.2.4). Repetir. Es decir, resuelva repetidamente (2.2.3) seguido de (2.2.4). Entonces se puede demostrar que este procedimiento iterativo converge a la solución de la ecuación HJB (4.10).

Este intercalado repetido de la ecuación de Bellman (2.2.3) seguido de la actualización del control (2.2.4) se conoce como Iteración de políticas y se detalla como Algoritmo 2.2.1. En este algoritmo, el subíndice j denota el número de iteración. Tenga en cuenta que si el algoritmo de iteración de políticas converge, entonces $u^{j+1} = u^j$. Entonces, al poner (2.2.2) en (2.2.1) se obtiene nada más que la ecuación HJB (4.10).

Según (2.2.2), el control $u^{j+1}(t) = -R^{-1}B^T \nabla V^j(x(t))$. En consecuencia, el control se da en función del estado $x(t)$; por lo tanto, es un control de retroalimentación de estado.

Política de control. Se dice que la entrada es una Política de control si se da en función del estado. Eso es $u(t) = h(x(t))$ para alguna función, posiblemente no lineal,

$h(.)$ Obviamente, el control $u(t)$ varía con el tiempo porque depende del estado $x(t)$. Sin embargo, la política en sí varía con el tiempo solo si la función $h(.)$ depende explícitamente del tiempo. En la Tabla 2.1.1, la política (2.1.18) es invariable en el tiempo, mientras que en la Tabla 2.1.2 la política (2.1.39) es variable en el tiempo. Además, la política solo cambia cuando cambia la función $h(.)$. Por ejemplo, al realizar (2.2.2) en el algoritmo de iteración de políticas 2.2.1.

Es importante señalar que, en el Algoritmo de iteración de políticas 2.2.1, dada una política de control $u^j(t)$, al resolver (2.2.1) se obtiene la función de valor $V^j(x)$ que es igual al valor de la integral (4.2) cuando la entrada es $u^j(t)$. De ahí que se llame Evaluación de Políticas. Se puede mostrar que la actualización de control (2.2.2) (consulte el siguiente teorema) da como resultado una política mejorada en la iteración j y, por lo tanto, se conoce como Mejora de la política.

Teorema 2.2.1. Iteración de políticas. Deje que la política de control inicial $u^0(t)$ en el Algoritmo 2.1.1 se estabilice. Entonces, la política de control $u^j(t)$ en cada iteración se estabiliza. Además, $u^{j+1}(t)$ tiene un valor $V^{j+1}(x)$ menor que el valor $V^j(x)$ de la política $u^j(t)$. Finalmente, el algoritmo de iteración de políticas converge monótonamente a la solución $V(x)$ de la ecuación HJB (4.10).

Ecuación de Lyapunov y algoritmo de Kleinman Nuestro objetivo es evitar la solución del ARE en la Tabla 2.1.1 y encontrar el LQR sin conocer la dinámica (A, B) midiendo los datos disponibles en tiempo real. El Paso 1 de RL es evitar la solución del ARE especializando el Algoritmo de Iteración de Políticas 2.2.1 al caso LQR. Asumiendo la forma cuadrática de la función de costo LQR (4.12), la ecuación de Bellman (2.2.3) se convierte en

$$H(x, \nabla V, u) = \frac{1}{2}\left[x^T P(Ax + Bu) + (Ax + Bu)^T Px + x^T Qx + u^T Ru\right] = 0$$

y el cálculo de control (2.2.4) se convierte en

$$u = -R^{-1}B^T Px \equiv -Kx$$

Ahora, considere la retroalimentación de estado $u = -Kx$ para cualquier matriz de ganancia K, no necesariamente la retroalimentación dada por (2.2.6). Poniendo $u = -Kx$ en (2.2.5) se obtiene

$$H(x, \nabla V, -Kx) = \frac{1}{2}\left[x^T P(A - BK)x + x^T (A - BK)^T Px + x^T Qx + x^T K^T RKx\right] = 0$$

Algoritmo 2.2.2 Algoritmo de Kleinman

Establecer $j = 0$. Seleccione una ganancia de retroalimentación inicial estabilizadora K^0.
Iteration j

1. Evaluación de políticas. Dada la ganancia K_j resuelve la ecuación de Lyapunov

$$\left(A - BK^j\right)^T P^j + P^j \left(A - BK^j\right) + Q + K^{jT} R K^j = 0$$

2. Mejora de políticas. Actualice la ganancia de retroalimentación usando

$$u^{j+1} = -K^{j+1}x = -R^{-1}B^T P^j x$$

Si $P^j = P^{j-1}$ detener. De lo contrario, establezca $j = j + 1$ y vaya a 1 .

Dado que esto es válido para todas las trayectorias $x(t)$, se obtiene la ecuación de Lyapunov

$$(A - BK)^T P + P(A - BK) + Q + K^T R K = 0$$

Se puede demostrar que la ecuación de Lyapunov tiene una única solución definida positiva P si y solo si $(A - BK)$ es asintóticamente estable (por ejemplo, tiene todos los polos estrictamente en el semiplano izquierdo).

Por lo tanto, para el caso LQR, el Algoritmo de iteración de políticas 2.2.1 intercala (2.2.10) y (2.2.6), como se detalla en el Algoritmo 2.2.2, que fue desarrollado por primera vez por David Kleinman en 1963 en []. Tenga en cuenta que el valor (4.2) de la política $u^j(x(t))$ es $V^j(x(t)) = 1/2x(t)^T P^j x(t)$. Dado que el algoritmo de Kleinman se basa en soluciones repetidas de la ecuación de Lyapunov, la ganancia de retroalimentación inicial K^0 debe ser estable.

Tenga en cuenta que si el algoritmo de Kleinman converge, entonces $K^{j+1} = K^j$. Entonces, al poner (2.2.9) en (2.2.8) no se obtiene nada más que la ecuación ARE de la tabla 2.1.1. Además, de acuerdo con el Teorema 2.2.1, si la ganancia de retroalimentación inicial en el Algoritmo 2.2.2 se estabiliza, entonces se estabiliza la ganancia K^j en cada iteración. Además, en cada iteración $P^{j+1} < P^j$. Finalmente, el algoritmo converge monótonamente a la única solución definida positiva P del ARE (2.1.17).

4.1.2.2. Aprendizaje Integral por Refuerzo

El algoritmo de iteración de políticas 2.2.1 evita la solución de la ecuación HJB (4.10) resolviendo repetidamente la ecuación de Bellman (2.2.1) y realizando actualizaciones de políticas de control (2.2.2). Es decir, la PDE HJB de segundo orden se resuelve

resolviendo repetidamente la PDE Bellman de primer orden más simple. Sin embargo, resolver una PDE no es fácil. Mostramos cómo especializar la iteración de políticas en el caso LQR y obtener el algoritmo 2.2.2 de Kleinman, que evita resolver el ARE y no resuelve las PDE, sino que solo resuelve repetidamente una ecuación lineal de Lyapunov. Desafortunadamente, resolver la ecuación de Lyapunov requiere información completa sobre la dinámica del sistema (A, B).

Ahora, logramos el Paso 2 de RL mostrando cómo escribir otra ecuación que es equivalente a la ecuación de Bellman (2.2.1) pero que no es una PDE y no contiene la dinámica del sistema (A, B)

Considere el índice de rendimiento (4.2). Derivando $V(x(t))$ se obtiene la ecuación PDE Bellman (4.7)/(2.2.3). Considere, en cambio, cortar la cola de (4.2) y escribir la función de valor como

$$V(x(t)) = \int_t^\infty r(x,u)d\tau = \frac{1}{2}\int_t^\infty \left(x^TQx + u^TRu\right)d\tau$$
$$= \frac{1}{2}\int_t^{t+T} \left(x^TQx + u^TRu\right)d\tau + \frac{1}{2}\int_{t+T}^\infty \left(x^TQx + u^TRu\right)d\tau$$

Ahora tenga en cuenta que, debido al límite superior infinito, el último término de esta ecuación no es más que el valor futuro $V(x(t+T))$. En consecuencia, obtenemos la ecuación de Bellman del aprendizaje por refuerzo integral (IRL)

$$V(x(t)) = \frac{1}{2}\int_t^{t+T} \left(x^TQx + u^TRu\right)d\tau + V(x(t+T))$$

Esta es una ecuación en diferencias para el valor $V(x(t))$. La cantidad

$$\rho(t, t+T)) = \frac{1}{2}\int_t^{t+T} \left(x^TQx + u^TRu\right)d\tau = \int_t^{t+T} r(x,u)d\tau$$

Algoritmo 2.2.3 Iteración de la política de aprendizaje por refuerzo integral (IRL)

Establecer $j = 0$. Seleccione una política de control inicial estabilizadora $u^0(t)$.
Iteration j

1. Evaluación de políticas. Dada la política de control $u^j(t)$, resuelva la ecuación diferencial de IRL Bellman para $V^j(x(t))$

$$V^j(x(t)) = \frac{1}{2}\int_t^{t+T} \left(x^T Q x + u^{jT} R u^j\right) d\tau + V^j(x(t+T))$$

3. Mejora de políticas. Actualizar la política de control usando

$$u^{j+1} = -R^{-1}B^T \nabla V^j$$

Si $V^j = V^{j-1}$ detener. De lo contrario, establezca $j = j + 1$ y vaya a 1 .

se conoce como Refuerzo Integral en el intervalo de tiempo $(t, t+T)$. Es el valor del índice de rendimiento (4.2) en el intervalo de tiempo $(t, t+T)$ para cualquier política estabilizadora $u(x(t))$ aplicada en ese intervalo.

Lema 2.2.2 Aprendizaje por refuerzo integral (IRL) Ecuación de Bellman La función de valor $V(x(t))$ que se encuentra resolviendo la ecuación diferencial de Bellman en la vida real (2.2.13) es la misma que la función de valor obtenida resolviendo la PDE de Bellman (2.1.7)/(2.2.3)ps

Prueba: La PDE de Bellman (2.2.3) y (4.5) son iguales. Integre (4.5) para obtener

$$\int_t^{t+T} \frac{d}{d\tau} V(x(\tau))d\tau = V(x(t+T)) - V(x(t)) = -\frac{1}{2}\int_t^{t+T} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau)\right) d\tau$$

que es lo mismo que la ecuación IRL Bellman (2.2.13).

De acuerdo con este teorema, está justificado reemplazar la ecuación de Bellman PDE (2.2.1) en el Algoritmo de iteración de políticas 2.2.1 por la ecuación de diferencia de Bellman IRL más simple (2.2.13) para obtener el Algoritmo de iteración de políticas IRL 2.2.3.

Iteración de políticas para el caso LQR

Para el caso LQR, la función de valor tiene la forma cuadrática (4.12), o

$$\mathrm{V}(x(t)) = \frac{1}{2} x(t)^T \,\mathrm{P}\mathrm{x}(t)$$

Algoritmo 2.2.4 Iteración de política IRL para *LQR*

Set $j = 0$. Select initial P^0 so that $u^0 = -K^0 x = -R^{-1}B^T P^0 x$ is stabilizing.
Iteration j

1. Evaluación de políticas. Dada la política de control $u^j(t)$ resuelva la ecuación de diferencias de IRL Bellman para P^j

$$x(t)^T P^j x(t) = \frac{1}{2}\int_t^{t+T} \left(x^T Q x + u^{jT} R u^j\right) d\tau + x(t+T)^T P^j x(t+T)$$

2. Mejora de políticas. Actualizar la política de control usando

$$u^{j+1} = -K^{j+1}x = -R^{-1}B^T P^j x$$

Si $P^j = P^{j-1}$ detener. De lo contrario, configure $j = j + 1$ y vaya a 1.

Luego, podemos escribir el Algoritmo de iteración de valor 2.3.1 para el LQR como Algoritmo 2.3.2

4.1.2.3. IRL en tiempo real por aproximación de función de valor

Nuestro objetivo final aquí es desarrollar un enfoque de aprendizaje por refuerzo completo para el diseño LQR que evite resolver la ecuación HJB (4.10) (equivalentemente, el ARE en la Tabla 2.1.1), y encuentre el control óptimo sin conocer la dinámica (A, B) midiendo los datos disponibles en tiempo real. Hemos mostrado en el Paso 1 de RL cómo evitar resolver la ecuación HJB (equivalentemente, el ARE) mediante el uso del Algoritmo de iteración de políticas 2.2.1 que resuelve repetidamente la ecuación PDE de Bellman (2.2.1) en su lugar. Luego, en el Paso 2 de RL, mostramos cómo evitar resolver la PDE de Bellman, que requiere el conocimiento de la dinámica (A, B), resolviendo la ecuación de diferencia de Bellman de IRL en el Algoritmo de iteración de políticas de IRL 2.2.3, que no necesita información dinámica.

Ahora, cumplimos el Paso 3 de RL y mostramos cómo desarrollar un algoritmo en tiempo real que se puede implementar en línea en el tiempo sin conocer la dinámica (A, B) midiendo los datos disponibles en tiempo real. Esto se logra mediante la aproximación de la función de valor $V(x(t))$ en (2.2.15).

Aproximación de funciones.
Un resultado básico en la aproximación de funciones es el siguiente.

Teorema de aproximación de Weierstrass. Cualquier función continua de valor real $f(x)$ de un escalar x se puede aproximar de acuerdo con

$$f(x) = \sum_{\ell=1}^{L} w_\ell \phi_\ell(x) + \varepsilon(x)$$

para coeficientes adecuados w_ℓ, donde $\{\phi_\ell(x)\}$ es un conjunto base de polinomios en x. Es decir, $\{\phi_\ell(x)\} = \{1, x, x^2, x^3, x^4, \ldots\}$. Además, a medida que el número de polinomios L tiende a infinito, el error de aproximación $\varepsilon(x)$ tiende a cero uniformemente (por ejemplo, independientemente del valor de x).

De hecho, (2.2.18) no es más que los primeros términos de una serie de Taylor de $f(x)$. Los resultados de aproximación más elaborados se basan, por ejemplo, en redes neuronales [], donde una función vectorial uniforme $f(x) : R^n \to R^p$ con componentes de valor real $f_i(x)$ se aproxima a un conjunto compacto $\|x\| \leq R, R > 0$ como

$$f_i(x) = \sum_{\ell=1}^{L} (w_{ii}\phi_\ell(x) + w_{i0}) + \varepsilon_i(x)$$

donde $\{\phi_t(x)\}$ se conocen como funciones de activación. Definir una matriz de coeficientes $W^T = [w_{i\ell}] \in R^{p\times L}$ permite escribir

$$f(x) = W^T\phi(x) + \varepsilon(x)$$

[Hornik y Stinchcomb, Sandberg] demostraron que, para funciones de activación adecuadamente elegidas, el error de aproximación $\varepsilon(x) = [\varepsilon_i(x)]$ está acotado en un conjunto compacto . Además, a medida que el número de unidades de capa oculta L tiende a infinito, $\varepsilon(x)$ tiende a cero. De hecho, las funciones de activación deben elegirse para que sean un conjunto base para $f(x)$. Las funciones de activación populares típicas incluyen sigmoids, $\tanh(x)$, funciones de base radial, etc.

Iteración de políticas IRL en tiempo real por aproximación de función de valor (VFA)

Supongamos que escribimos la función de valor usando la ecuación de aproximación de función de valor (VFA)

$$V(x(t)) = W^T\phi(x(t))$$

con $\phi(x)$ un vector de función de activación adecuadamente elegido. Dado que la función de valor es un escalar, $W^T = [w_{il}] \in R^{1\times L}$ y el vector de peso W es un vector de longitud L. Sustituya esta ecuación VFA en la ecuación IRL Bellman (2.2.15) en el Algoritmo de iteración de políticas IRL 2.2.3 para obtener

$$W^{jT}\phi(x(t)) = \frac{1}{2}\int_t^{t+T} \left(x^TQx + u^{jT}Ru^j\right) d\tau + W^{jT}\phi(x(t+T))$$

o

$$W^{jT}\left[\phi(x(t)) - \phi(x(t+T)] = \frac{1}{2}\int_t^{t+T}\left(x^TQx + u^{jT}Ru^j\right)d\tau = \rho(t,t+T)\right)$$

Esta es una ecuación lineal con pesos desconocidos W^j. Es decir, VFA nos ha permitido transformar la ecuación en diferencias (2.2.15) en una ecuación lineal que es mucho más fácil de resolver. Además, usando VFA uno tiene

$$\nabla V(x(t)) = \frac{\partial \phi^T(x(t))}{\partial x}W$$

por lo que la actualización de control (2.2.16) se puede escribir en términos del vector de peso.

La forma final de iteración de política IRL en tiempo real se proporciona como algoritmo 2.2.5. En resumen, la iteración de políticas nos ha permitido evitar resolver la ecuación PDE HJB de segundo orden (4.10) resolviendo repetidamente la ecuación Bellman de PDE de primer orden en el algoritmo de iteración de políticas 2.2.1. Luego, IRL ha transformado esto en el Algoritmo de iteración de políticas de IRL 2.2.3, que en su lugar requiere una solución repetida de la ecuación de diferencias de Bellman de IRL (2.2.15). Finalmente, VFA ha transformado esto en el Algoritmo de iteración de políticas de IRL en línea 2.2.5, con una ecuación de Bellman de IRL (2.2.26) que es lineal en los pesos. Este algoritmo se implementa en línea hacia adelante en el tiempo midiendo los datos $x(t), u(t)$ a lo largo de las trayectorias del sistema y utilizándolos para resolver (2.2.26) los pesos W^j.

Defina la diferencia de la función de activación en el momento t como

$$\Delta\phi(x(t)) = \phi(x(t)) - \phi(x(t+T)$$

Algoritmo 2.2.5 Iteración de políticas de aprendizaje por refuerzo integral (IRL) en tiempo real

Iteration j

1. Policy Evaluation. Given control policy $u^j(t)$ solve the IRL Bellman weight equation for W^j

$$W^{jT}\left[\phi(x(t)) - \phi(x(t+T)] = \frac{1}{2}\int_t^{t+T}\left(x^TQx + u^{jT}Ru^j\right)d\tau = \rho(t,t+T)\right)$$

3. Mejora de políticas. Actualizar la política de control usando

$$u^{j+1} = -R^{-1}B^T\frac{\partial\phi^T}{\partial x}W^j$$

Si $V^j = V^{j-1}$ detener. De lo contrario, establezca $j = j+1$ y vaya a 1 .

$$W^{jT}\Delta\phi(x(t)) = \frac{1}{2}\int_t^{t+T}\left(x^TQx + u^{jT}Ru^j\right)d\tau = \rho(t,t+T)\right)$$

El problema con la ecuación de peso IRL (2.2.26) es que es una ecuación escalar para los pesos W^j, que generalmente se dan como un vector con entradas L. Sin embargo, existen varios mecanismos simples para resolver W^j usando datos $x(t), u(t)$ medidos en línea en tiempo real. De hecho, esta ecuación está en la forma estándar de la ecuación de identificación del sistema en el control adaptativo. Como tal, se resuelve fácilmente utilizando algoritmos de mínimos cuadrados recursivos (RLS) []. De hecho, $\Delta\phi(x(t))$ en (2.2.28) se denomina vector de regresión en la identificación del sistema.

Example 2.2.1. Solución de mínimos cuadrados por lotes de la ecuación de peso de Bellman IRL para LQR

Considere el caso LQR lineal e invariante en el tiempo con un vector de estado bidimensional $x(t) \in R^2$. Entonces la función de valor viene dada por (4.12) o

$$V(x) = x^T(t)Px(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}^T \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

donde la matriz de pesos 2×2 $P = P^T > 0$ es una constante. Como P es simétrico, hay 3 incógnitas y se puede escribir

$$V(x) = \begin{bmatrix} p_{11} & p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} x_1^2(t) \\ 2x_1(t)x_2(t) \\ x_2^2(t) \end{bmatrix} \equiv W^T\phi(x)$$

Se ve que en VFA (2.2.21) un conjunto adecuado de activaciones para LQR son los polinomios en el vector $x(t)$. Además, el vector de pesos W es constante y consta de las entradas de la matriz P. Aquí, W es un vector de longitud 3.

Ahora la diferencia de la función de activación se puede escribir como

$$\Delta\phi(x(t)) = \phi(x(t)) - \phi\left(x(t+T)\right) = \begin{bmatrix} x_1^2(t) \\ 2x_1(t)x_2(t) \\ x_2^2(t) \end{bmatrix} - \begin{bmatrix} x_1^2(t+T) \\ 2x_1(t+T)x_2(t+T) \\ x_2^2(t+T) \end{bmatrix}$$

que es el vector polinomial evaluado en $x(t)$ menos el vector polinomial evaluado en $x(t+T)$. Esto se puede calcular directamente si se mide $x(t), x(t+T)$ en cada paso de tiempo.

Ahora toma datos de tres pasos de tiempo y usa (2.2.26) para escribir

$$W^{jT}[\Delta\phi(x(t))] \equiv W^{jT}[\phi(x(t)) - \phi(x(t+T))] = \int_t^{t+T} \left(Q(x) + u_k^T R u_k\right) dt = \rho(t, t+T)$$

$$W^{jT}[\Delta\phi(x(t+T))] \equiv W^{jT}[\phi(x(t+T)) - \phi(x(t+2T))] = \int_{t+T}^{t+2T} \left(Q(x) + u_k^T R u_k\right) dt = \rho(t+T, t+2T)$$

$$W^{jT}[\Delta\phi(x(t+2T))] \equiv W^{jT}[\phi(x(t+2T)) - \phi(x(t+3T))] = \int_{t+2T}^{t+3T} \left(Q(x) + u_k^T R u_k\right) dt = \rho(t+2T, t+3T)$$

Tenga en cuenta que la matriz de peso W^j es la misma en las tres ecuaciones y recójalas en la ecuación matricial única

$$W^{jT}[\Delta\phi(x(t))\Delta\phi(x(t+T))\Delta\phi(x(t+2T))] = [\rho(t, t+T)\rho(t+T, t+2T)\rho(t+2T, t+3T)]$$

Mediante definiciones adecuadas, escribe esto como

$$W^{jT} M(t, t+2T) = \mathrm{P}(t, t+2T)$$

Estas son las ecuaciones normales de mínimos cuadrados. Tenga en cuenta que $M(t, t+2T)$ es una matriz de 3×3, por lo que W^j se puede encontrar mediante una solución de mínimos cuadrados por lotes. De hecho, si $M(t, t+2T)$ tiene 3 columnas independientes, entonces se puede invertir para obtener W^j.

Suponga que el vector de peso tiene W^j tiene entradas L, con L el número de unidades de capa oculta, p. el número de entradas en el vector de activación $\phi(x)$. Entonces se

requieren al menos L ecuaciones de la forma (2.2.26) utilizando datos recopilados de L pasos de tiempo, y las ecuaciones normales de mínimos cuadrados se convierten en

$$W^{jT} M(t, t+(L-1)T) = \mathrm{P}(t, t+(L-1)T)$$

donde $M(t, t+(L-1)T)$ es una matriz $L \times L$. Si esta matriz es invertible, entonces se pueden resolver las ecuaciones normales de mínimos cuadrados para W^j. Esto sucede si las salidas $\Delta\phi(x(t))$ de las unidades de la capa oculta son persistentemente emocionantes.

Persistencia de excitación. La serie $\Delta\phi(x(t))$ es persistentemente excitante (PE) si

$$\sum_{\ell=0}^{\Lambda-1} \Delta\phi(x(t+\ell T))[\Delta\phi(x(t+\ell T))]^T > 0$$

durante cierto número de pasos de tiempo Λ.

Claramente, si $\Delta\phi(x(t))$ es PE para Λ, entonces la matriz $M(t, t+(\Lambda-1))M^T(t, t+(\Lambda-1))$ no es singular y las ecuaciones normales de mínimos cuadrados (2.2.34) sobre Λ pasos de tiempo se pueden resolver para W^j usando la solución de mínimos cuadrados por lotes

$$W^{jT} = \mathrm{P}(t, t+(\Lambda-1)T)M^T(t, t+(\Lambda-1)) \left[M(t, t+(\Lambda-1))M^T(t, t+(\Lambda-1))\right]^{-1}$$

Obviamente, se requiere que $\Lambda \geq L$. Considere el modelo dinámico del robot manipulador en el espacio cartesiano [87]

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + F_c(\dot{q}) + G(q) + \tau_d = \tau + K_h f_h \tag{4.20}$$

con $M = J^{-T}M^*J^{-1}$, $C = J^{-T}\left(C^* - M^*J^{-1}J\right)J^{-1}$, $M = J^{-T}M^*J^{-1}$, $F_c = J^{-T}F^*$, $G = J^{-T}G^*$, y $\tau = J^{-T}\tau^*$, donde $q \in \mathbb{R}^n$ es el vector de coordenadas conjuntas generalizadas, n es el numero de juntas, $x \in \mathbb{R}^n$ es la posición cartesiana del efector final, la fuerza de entrada de control es $\tau = J^{-T}\tau^*$ con τ^* es el vector de pares generalizados que actúan en las articulaciones, $M^* \in R^{n\times n}$ es la matriz de masa definida positiva simétrica (inercia), $C^*(q,\dot{q})\dot{q} \in \mathbb{R}^{n\times 1}$ es el vector de Coriolis y las fuerzas centrípetas, $F_c^*(\dot{q}) \in \mathbb{R}^{n\times 1}$ es el término de fricción de Coulomb, $G^*(q) \in \mathbb{R}^{n\times 1}$ es el vector de pares gravitacionales, τ_d es una perturbación general no lineal, f_h es el esfuerzo de control humano, K_h es una ganancia y J es la matriz jacobiana.

Considere el modelo de impedancia del robot prescrito

$$\bar{M}\ddot{x}_m + \bar{B}\dot{x}_m + \bar{K}x_m = K_h f_h + \bar{l}\left(x_d\right) \equiv l\left(f_h, x_d\right) \tag{4.21}$$

en el espacio cartesiano, donde x_m es la salida del modelo de impedancia del robot prescrito, $\bar{M}, \bar{B}$, and $\bar{K}$ son las matrices deseadas de parámetros de inercia,

amortiguamiento y rigidez, respectivamente.La entrada auxiliar $\bar{l}\,(x_d)$ es una entrada dependiente de la trayectoria.

Objetivo de diseño: El objetivo es diseñar la fuerza τ en 4.20 para hacer la dinámica del robot desconocido **??** de la fuerza humana f_h a las coordenadas cartesianas x se comportan como el modelo de impedancia del robot prescrito 4.21. Es decir, se desea hacer que el siguiente error de seguimiento del modelo llegue a cero:

$$e = x_m - x \tag{4.22}$$

Teniendo en cuenta que esto no es un error de seguimiento de trayectoria. Por lo tanto, este es un diseño que sigue un modelo, y no un diseño que sigue una trayectoria, en contraste con la mayoría de los trabajos sobre control de par de robots [88–93]. No se requiere información de tareas en esta sección. Todos los detalles específicos de la tarea se tienen en cuenta en la siguiente sección.

Ahora se requiere diseñar un par de control τ para hacer que el robot se comporte como el modelo de impedancia del robot prescrito 4.21.

Considere el par de control

$$\tau = \hat{W}^T \phi\left(\hat{V}^T z\right) + K_v r - v(t) - K_h f \tag{4.23}$$

Donde $v(t)$ es una señal de refuerzo a especificar, K_v es la ganancia de control, y

$$r = \dot{e} + \Lambda_1 e + \Lambda_2 \varepsilon \tag{4.24}$$

es el error del modo deslizante con

$$\varepsilon = \int_0^t e(\tau) d\tau \tag{4.25}$$

Finalmente

$$\hat{h}(z) = \hat{W}^T \phi\left(\hat{V}^T z\right) \tag{4.26}$$

es una NN con $z = [q, \dot{q}, \dot{x}_m, \ddot{x}_m, e, \dot{e}, \varepsilon]^T$ la entrada a la NN, $\hat{W}$ y $\hat{V}$ los pesos NN, y $\phi(z)$ el vector de funciones de activación. Como se muestra en la demostración del Teorema 1, el controlador NN en 4.23 se usa para compensar la función desconocida del robot h definida comoo

$$\begin{aligned} h(z) = M(q)\left(\ddot{x}_m + \Lambda_1 \dot{e} + \Lambda_2 e\right) + C(q, \dot{q})\left(\dot{x}_m + \Lambda_1 e + \Lambda_2 \varepsilon\right) \\ + F_c(\dot{q}) + G(q) \end{aligned} \tag{4.27}$$

La propiedad de aproximación universal NN especifica que cualquier función continua desconocida se puede aproximar en un conjunto compacto utilizando un NN de dos capas

con cualquier precisión arbitraria [87]. Es decir, para la función continua $h(z)$ en un conjunto compacto $z \in \Omega$, uno tiene

$$h(z) = W^T \phi\left(V^T z\right) + \varepsilon(z) \tag{4.28}$$

Donde V es una matriz de pesos de la primera capa, W es una matriz de pesos de la segunda capa, y ε es el error de aproximación funcional NN. Los vectores de peso ideal W y V son desconocidos y se aproximan en línea. Por lo tanto, $h(z)$ se aproxima como 4.26 con $\hat{W}$ y $\hat{V}$ las estimaciones de W y V, respectivamente. Definir

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \tag{4.29}$$

y $\hat{Z}$ equivalentemente.

*Suposición 1:*Los pesos ideales de NN están acotados por un escalar constante, de modo que

$$\|Z\| \leq Z_B. \tag{4.30}$$

El siguiente teorema muestra que la entrada de control propuesta τ dada por 4.23 garantiza la acotación del error de seguimiento del modelo e y los pesos NN.

Teorema 1: Considere la dinámica del manipulador del robot **??** y el modelo de impedancia del robot prescrito 4.21. Deje que la entrada de control se elija como 4.23. Deje que la suposición 1 se mantenga. Sea la regla de actualización para los pesos de NN dada por

$$\dot{\hat{W}} = F\hat{\phi} r^T - F\hat{\phi}'\hat{V}^T z r^T - kF\|r\|\hat{W} \tag{4.31}$$

$$\dot{\hat{V}} = Gz\left(\hat{\phi}'\hat{W} r\right)^T r^T - kG\|r\|\hat{V} \tag{4.32}$$

donde $\hat{\phi} = \phi\left(\hat{V}^T z\right), \hat{\phi}' = d\phi(y)/\, dy|_{y=\hat{V}^T z}, F = F^T > 0$, $G = G^T > 0$, y $k > 0$ es un pequeño parámetro de diseño. Sea el término de robustecimiento

$$v(t) = -K_z\left(\|\hat{Z}\| + Z_B\right) \tag{4.33}$$

donde $K_z > 0$. Entonces, $e(t)$ en 4.22 y los pesos estimados de NN están uniformemente acotados en última instancia.

Prueba: Derivando 4.22 con respecto al tiempo se tiene $\dot{e} = \dot{x}_m - \dot{x}$, o equivalente $\dot{x} = \dot{x}_m - \dot{e}$. Diferenciando $\dot{x}$ da $\ddot{x} = \ddot{x}_m - \ddot{e}$. Considerando el error de seguimiento

del modo deslizante r definida en 4.24, se tiene $\dot{e} = r - \Lambda_1 e - \Lambda_2 \varepsilon$. Diferenciando $\dot{e}$ da $\ddot{e} = \dot{r} - \Lambda_1 \dot{e} - \Lambda_2 e$. Usando estas expresiones en **??** se obtiene

$$\begin{aligned} M(q)\left(\ddot{x}_m - (\dot{r} - \Lambda_1 \dot{e} - \Lambda_2 e)\right) + C(q,\dot{q})\left(\dot{x}_m - (r - \Lambda_1 e - \Lambda_2 \varepsilon)\right) \\ + F_c(\dot{q}) + G(q) + \tau_d = \tau + K_h f_h \end{aligned} \tag{4.34}$$

Esto da la dinámica de error de modo deslizante

$$\begin{aligned} M(q)\dot{r} = -C(q,\dot{q})r + h\left(q,\dot{q},\dot{x}_m,\ddot{x}_m,e,\dot{e},\varepsilon\right) \\ + \tau_d - \tau - K_h f_h \end{aligned} \tag{4.35}$$

con h definido en 4.27. Se asume que la dinámica del manipulador del robot **??** es desconocida y, por lo tanto, h en 4.35 se desconoce y se aproxima en línea mediante 4.26. Entonces, la dinámica del error filtrado en bucle cerrado 4.35 se convierte en

$$M(q)\dot{r} = -C(q,\dot{q})r + \hat{W}^T \phi\left(\hat{V}^T z\right) + \tau_d - \tau - K_h f_h + \tilde{h} \tag{4.36}$$

donde $\tilde{h} = h - \hat{h}$ es el error de estimación.

Sustituyendo τ de 4.23 en 4.36 da

$$M(q)\dot{r} = -C(q,\dot{q})r - K_v r + \tau_d + \tilde{h} + v(t) \tag{4.37}$$

El resto de la prueba es igual a [87] y, por lo tanto, solo se describe aquí. Una función de Lyapunov se define como

$$L = \frac{1}{2} r^T M(q) r + \operatorname{tr}\left(\tilde{W}^T F^{-1} \tilde{W}\right) + \operatorname{tr}\left(\tilde{V}^T F^{-1} \tilde{V}\right) \tag{4.38}$$

donde los errores de estimación de peso son $\tilde{W} = W - \hat{W}$ y $\tilde{V} = V - \hat{V}$, y se muestra usando 4.31,4.32,4.33 y 4.36 que la derivada de la función de Lyapunov es negativa fuera de un conjunto compacto. Esto garantiza la acotación del error de seguimiento filtrado r así como los pesos NN. Los límites específicos de r y los pesos NN se dan en [87].

Teniendo en cuenta que el controlador propuesto 4.23 se compone de cuatro partes. La primera parte es un compensador no lineal que consta de un controlador NN para compensar la función desconocida h definida en 4.27. La segunda parte del controlador es un controlador proporcional integral derivado estabilizador que estabiliza el error de seguimiento del modelo e. La tercera parte es un término robusto que está diseñado para lograr robustez frente a las incertidumbres. Finalmente, la última parte se usa para compensar la entrada humana $K_h f_h$.

La Fig. 4 muestra el esquema detallado del controlador de bucle interno propuesto. Llamamos a este modelo control neuroadaptativo de referencia porque el controlador adaptativo NN hace que la dinámica del robot se comporte como el modelo de impedancia prescrito 4.21. Esto contrasta con el trabajo de control de par NN, que busca hacer que el movimiento del robot $x(t)$ siga una trayectoria prescrita.

Observación 2: la función $h(z)$ en 4.27 no contiene los parámetros $\bar{M}, \bar{D}$ y $\bar{K}$ del modelo de impedancia del robot prescrito 4.21. Esto significa que la NN no necesita estimar el modelo de impedancia. Esto contrasta con los métodos que diseñan el par τ en **??** para garantizar el seguimiento de una trayectoria deseada, y exigen que la trayectoria y la dinámica del error de seguimiento sigan un modelo de impedancia prescrito [88–93]. Nuestro diseño del controlador de bucle interno específico del robot es independiente de los objetivos de cualquier tarea.

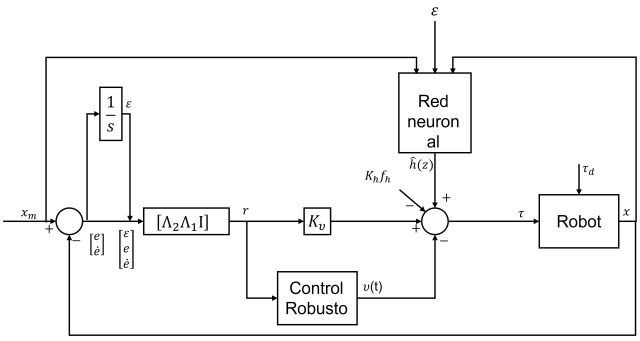

Figura 4.1: Controlador de bucle interno neuroadaptativo de referencia de modelo

4.2. Control PID usando como Compensación el Aprendizaje por Reforzamiento.

Hoy en día, la idea del control de robots manipuladores ha atraído la atención de la comunidad científica de robótica. Un punto de interés se ubica en el diseño de sistemas de control para robots manipuladores en aplicaciones industriales tales como el estibar cargamento, ensamblaje, traslado, pintado de objetos, etc. Puesto que los robots industriales son capaces de realizar correctamente una variedad, a simple vista parecería innecesario desarrollar investigación sobre el tema de control de robots manipuladores. Sin embargo, es importante resaltar que la ejecución de la tarea encomendada al robot requiere alto desempeño y exactitud en sus movimientos.

El diseño de nuevos esquemas de control requiere de grandes retos teóricos que mejoran sustancialmente problemas de origen práctico. Además, su estudio resulta indispensable

en aplicaciones que no pueden ser llevadas a cabo por medio de algoritmos de control tradicionales. De ahí que resulta muy importante y de gran interés el diseño de nuevas técnicas de control que solventen las necesidades de tener un control con mayor robustez. Este capítulo está destinado a presentar un algoritmo de control basado en la técnica del aprendizaje por reforzamiento, y compararlo contra una familia de controladores clásicos basados en la metodología de modelo de energía.

El control de posición o regulación de control de robots manipuladores es un caso particular de control de movimiento en el cual no hay una referencia variante en el tiempo que el robot haga seguimiento como en el caso de control de trayectoria, más bien, es un punto constante en el tiempo al que se le denomina posición deseada o set point. El objetivo de control es posicionar el extremo final del robot en ese punto y que permanezca ahí de manera indefinida. Por supuesto, para propósitos prácticos, una vez que el extremo final del robot alcanza el punto deseado, deberá pasar uno o más periodos de muestreo para cambiar de valor el punto deseado, entonces, el actual punto deseado, tomará el papel de condición inicial y el extremo final del robot se moverá al nuevo punto deseado, y así sucesivamente. Este concepto da la posibilidad de interpolar punto a punto para que el robot pueda seguir la trayectoria a través de un esquema de control de posición con puntos cercanos entre sí. En control automático se le conoce como control punto a punto.

4.2.1. Control PID en caso de Regulación

El problema de control de posición o regulación consiste en mover el extremo final del robot manipulador desde cualquier posición inicial hacia una posición deseada. Esto significa que la i- ésima articulación del robot deberá moverse hacia la respectiva i-ésima posición deseada. Un ejemplo ilustrativo se muestra en la figura (5.1) donde el robot parte de la posición de reposo (posición de casa) para llegar a la configuración deseada permaneciendo indefinidamente en el punto de equilibrio. Formalmente, el objetivo de control de posición está determinado por encontrar una ley de control τ que proporciona los pares aplicados a las articulaciones o servomotores del robot, de tal forma que la posición actual del robot $q(t)$ y la velocidad articular de movimiento $\dot{q}(t)$ tiende asintóticamente hacia la posición deseada q_dy velocidad cero, respectivamente, sin importar las condiciones iniciales. Es decir,

$$\lim_{t\to\infty} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} q_d \\ 0 \end{bmatrix}.$$

Nótese que en la figura (5.1) el robot se encuentra en su punto de equilibrio estable, lo que significa que el objetivo de control (5.1) se cumplió sin depender de las condiciones iniciales, entonces la posición deseada se alcanza, por lo que la posición del extremo final del robot permanece constante $q(t) = q_d$ y por lo tanto la velocidad de movimiento es

cero ($\dot{q} = 0$).

Finalmente, poder decir que un algoritmo de control de posición o regulación es una formulación cuya principal característica es generar un atractor en la ecuación de lazo cerrado formada por el modelo dinámico del robot manipulador y la estructura matemática del algoritmo de control. Lo anterior significa que el punto de equilibrio sea asintóticamente estable. La importancia de esta problemática radica en proponer estrategias de control que

El desempeño de un algoritmo de control se refiere a realizar de manera correcta y exacta la tarea programada al robot, lo que lo habilita a llevar a cabo diversas aplicaciones de control punto a punto. Por lo tanto, el espectro de aplicaciones comerciales, domesticas, científicas e industriales se incrementa.

4.2.2. Control PID con compensación QL

Considérese el modelo dinámico que describe el comportamiento de un robot manipulador de n grados de libertad

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

en términos del vector de estado $\left[q^T, \dot{q}^T\right]^T$

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1}[\tau - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)] \end{bmatrix},$$

Figura 5.1: En la figura de la izquierda vemos la condición inicial en la posición de equilibrio estable (posición de casa). En la figura de la derecha vemos la condición final deseada.

donde $q\dot{q}\ddot{q} \in \mathbb{R}^n$ denotan la posición, velocidad y aceleración articular, respectivamente, $\tau \in \mathbb{R}^n$ es un vector de fuerzas y pares aplicados en las uniones mediante los actuadores, $M(q) \in \mathbb{R}^{n\times n}$ es la matriz de inercia, $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ es el vector de fuerzas centrifugas y de coriolis, $G(q) \in \mathbb{R}^n$ es el vector de pares gravitacionales, y $F(\dot{q})$ representa la función de coulomb, y se representa de la forma:

$$F(\dot{q}) = B_{f1}\dot{q} + B_{f2}\,\mathrm{sign}(\dot{q}),$$

donde B_{f1} y B_{f2} son matrices positivas $\in \mathbb{R}^{n\times n}$, por simplicidad usaremos el modelo siguiente:

$$F(\dot{q}) = B_{f1}\dot{q}.$$

El objetivo de control se puede definir formalmente de la siguiente manera: dada la

posición angular deseada $q_d \in \mathbb{R}^n$ constante para todo $t \geq 0$, el problema es diseñar una ley de control τ tal que la posición angular del robot $q(t)$ se aproxima a $q_d \in \mathbb{R}^n$ asintóticamente, es decir:

$$\lim_{t\to\infty} \|\tilde{q}\| = 0.$$

El vector $q_d \in \mathbb{R}^n$ es la posición articular deseada, y el vector $\tilde{q} = q_d - q \in \mathbb{R}^n$ es el vector de error de posición.

4.2.3. Ley de Control

La ley de control PID+QL puede expresarse de la siguiente manera:

$$\tau = K_p\tilde{q} + K_d\dot{\tilde{q}} + K_i \int_0^t \tilde{q}(\psi)d\psi + u_r,$$

donde las matrices de diseño $K_p, K_d, K_i \in \mathbb{R}^{n\times n}$ llamadas respectivamente las ganancias proporcional, derivativa e integral, son matrices simétricas y definidas positivas convencionalmente elegidas. $u_r \in \mathbb{R}^n$ es el algoritmo de control llamado Q-Learning, que tiene la forma:

$$u_r = \beta(-\operatorname{sign}(\dot{q}) + \varepsilon),$$

donde β es una matriz diagonal y definida positiva seleccionada por el diseñador, y ε representa el error de aproximación de la función $\Phi(\dot{q})$ con la función $\operatorname{sign}(\dot{q})$, donde la función signo se representa de la siguiente manera:

$$\Phi(\dot{q}) = \operatorname{sign}(\dot{q}) = \begin{cases} 1 \text{ si } \dot{q} > 0 \\ 0 \text{ si } \dot{q} = 0 \\ -1 \text{ si } \dot{q} < 0 \end{cases}$$

El vector $\operatorname{sign}(\dot{q})$ está definido por $\operatorname{sign}(\dot{q}) = [\operatorname{sign}(\dot{q}_1), \ldots, \operatorname{sign}(\dot{q}_n)]^T$.

La acción integral del controlador PID+QL introduce una nueva variable de estado adicional que será denotada por ξ y cuya derivada temporal es $\dot{\xi} = K_i\tilde{q}$. Además, en el caso de regulación $\dot{q}_d = 0, \tilde{q} = -\dot{q}$, entonces, la ley de control PID+QL en el caso de regulación puede expresarse por medio de las ecuaciones siguientes:

$$\begin{aligned} \tau &= K_p\tilde{q} - K_d\dot{q} + \xi + \beta(-\operatorname{sign}(\dot{q}) + \varepsilon) \\ \dot{\xi} &= K_i\tilde{q} \end{aligned}$$

4.2.4. Ecuación en malla cerrada

La ecuación que describe el comportamiento en malla cerrada se obtiene al combinar las ecuaciones (5.2) y (5.3).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = K_p\tilde{q} - K_d\dot{q} + \xi + \beta(-\operatorname{sign}(\dot{q}) + \varepsilon),$$

despejando $\ddot{q}$ se tiene

$$\begin{aligned}\ddot{q} &= M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + K_i\xi + \beta(-\operatorname{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right],\\ \dot{\xi} &= K_i\tilde{q}\end{aligned}$$

la cual puede expresarse en términos del vector de estado $\left[\xi^T, \tilde{q}^T, \dot{q}^T\right]^T$ como:

$$\frac{d}{dt}\begin{bmatrix}\xi\\ \tilde{q}\\ \dot{q}\end{bmatrix} = \begin{bmatrix} K_i\tilde{q} \\ -\dot{q} \\ M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + \xi + \beta(-\operatorname{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right]\end{bmatrix}$$

los equilibrios tienen la forma

$$\begin{bmatrix}\xi\\ \tilde{q}\\ \dot{q}\end{bmatrix} = \begin{bmatrix} G\left(q_d\right) \\ 0 \\ 0\end{bmatrix}.$$

El equilibrio anterior puede trasladarse al origen mediante el siguiente cambio de variable $\tilde{\xi} = \xi - G\left(q_d\right)$, y la malla cerrada podrá expresarse en términos del vector de estado $\left[\tilde{\xi}^T, \tilde{q}^T, \dot{q}^T\right]^T$ como:

$$\frac{d}{dt}\begin{bmatrix}\tilde{\xi}\\ \tilde{q}\\ \dot{q}\end{bmatrix} = \begin{bmatrix} K_i\tilde{q} \\ -\dot{q} \\ M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} - G\left(q_d\right) + \beta(-\operatorname{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right]\end{bmatrix}.$$

Nótese que la ecuación anterior es autónoma y su único equilibrio es el origen $\left[\tilde{\xi}^T, \tilde{q}^T, \dot{q}^T\right]^T = 0 \in \mathbb{R}^{3n}$.

4.2.5. Prueba de Estabilidad

Para estudiar la estabilidad del origen del espacio de estados, se propone la siguiente función candidata de Lyapunov:

$$\begin{aligned} V(\tilde{\xi},\tilde{q},\dot{q}) =& \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_p \tilde{q} + U\left(q_d - q\right) - k_u + \\ &+ \tilde{q}^T G\left(q_d\right) + \tilde{q}^T\tilde{\xi} + \frac{3}{2}G\left(q_d\right)^T K_p^{-1} G\left(q_d\right) + \frac{\alpha}{2}\tilde{\xi}^T K_i^{-1}\tilde{\xi} + \\ &- \alpha\dot{q}^T M(q)\tilde{q} + \frac{\alpha}{2}\tilde{q}^T\left[K_d + B_{f1}\right]\tilde{q} + \alpha K_i^{-1}\int_0^t \Phi(\dot{q})d\xi \end{aligned}$$

donde $U\left(q_d - q\right)$ denota, como ya es costumbre, la energía potencial del robot, $k_u = \mathrm{mi}_q\left\{U\left(q_d - q\right)\right\}$, que se agrega de modo que $V(0) = 0$, y α es una constante positiva que satisface condiciones bien definidas para que la función candidata de Lyapunov sea definida positiva.

Se probará que la función candidata de Lyapunov es definida positiva, $V(\tilde{\xi},\tilde{q},\dot{q}) \geq 0$.

El término $\frac{1}{2}\tilde{q}^T K_p \tilde{q}$ los dividiremos en tres partes, y la función candidata de Lyapunov la dividiremos en cuatro partes $V(\tilde{\xi},\tilde{q},\dot{q}) = \sum_{i=1}^{4} V(\tilde{\xi},\tilde{q},\dot{q})_i$.

$$\begin{aligned} V(\tilde{\xi},\tilde{q},\dot{q})_1 &= \frac{1}{6}\tilde{q}^T K_p \tilde{q} + \tilde{q}^T G\left(q_d\right) + \frac{3}{2}G\left(q_d\right)^T K_p^{-1} G\left(q_d\right) \\ V(\tilde{\xi},\tilde{q},\dot{q})_2 &= \frac{1}{6}\tilde{q}^T K_p \tilde{q} + \tilde{q}^T\tilde{\xi} + \frac{\alpha}{2}\tilde{\xi}^T K_i^{-1}\tilde{\xi} \\ V(\tilde{\xi},\tilde{q},\dot{q})_3 &= \frac{1}{6}\tilde{q}^T K_p \tilde{q} - \alpha\dot{q}^T M(q)\tilde{q} + \frac{1}{2}\dot{q}^T M(q)\dot{q} \\ V(\tilde{\xi},\tilde{q},\dot{q})_4 &= U\left(q_d - q\right) - k_u + \frac{\alpha}{2}\tilde{q}^T\left[K_d + B_{f1}\right]\tilde{q} + \alpha K_i^{-1}\int_0^t \Phi(\dot{q})d\tilde{\xi} \end{aligned}$$

Del primer término $V(\tilde{\xi},\tilde{q},\dot{q})_1$ expresado en forma matricial podemos facílmente ver que si $K_p > 0$, entonces $V(\tilde{\xi},\tilde{q},\dot{q})_1$ es semidefinida positiva.

$$V(\tilde{\xi},\tilde{q},\dot{q})_1 = \frac{1}{2}\begin{bmatrix} \tilde{q} \\ G\left(q_d\right) \end{bmatrix}\begin{bmatrix} \frac{1}{3}K_p & I \\ I & 3K_p^{-1} \end{bmatrix}\begin{bmatrix} \tilde{q} \\ G\left(q_d\right) \end{bmatrix} \geq 0.$$

Del segundo término $V(\tilde{\xi},\tilde{q},\dot{q})_2$ expresado en forma matricial obtenemos la primera condición de α para que la función sea definida positiva.

$$V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{2}\begin{bmatrix} \tilde{q} \\ \tilde{\xi} \end{bmatrix}\begin{bmatrix} \frac{1}{3}K_p & I \\ I & \alpha K_i^{-1} \end{bmatrix}\begin{bmatrix} \tilde{q} \\ \tilde{\xi} \end{bmatrix},$$

usando el criterio de Sylvester, para probar que la matriz es definida positiva, el

determinante debe ser positivo, por lo cual tenemos:

por lo tanto si hacemos que α cumpla con la restricción anterior, entonces $V(\tilde{\xi},\tilde{q},\dot{q})_2$ es definida positiva.

Del tercer término $V(\tilde{\xi},\tilde{q},\dot{q})_3$ expresado en forma matricial obtenemos la segunda condición de α para que la función sea definida positiva.

$$V(\tilde{\xi},\tilde{q},\dot{q})_3 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}\begin{bmatrix}\frac{1}{3}K_p & -\alpha M(q)\\ -\alpha M(q) & M(q)\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix},$$

usando nuevamente el criterio de Sylvester, para probar que la matriz es definida positiva, el determinante debe ser positivo, por lo cual tenemos:,

$$V(\tilde{\xi},\tilde{q},\dot{q})_3 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}\begin{bmatrix}\frac{1}{3}\lambda_{\text{mín}}(K_p) & -\alpha\lambda_{\text{máx}}(M)\\ -\alpha\lambda_{\text{máx}}(M) & \lambda_{\text{mín}}(M)\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}$$

$$V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \tilde{\xi}\end{bmatrix}\begin{bmatrix}\frac{1}{3}\lambda_{\text{mín}}(K_p) & 1\\ 1 & \alpha\lambda_{\text{mín}}\left(K_i^{-1}\right)\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \tilde{\xi}\end{bmatrix}$$

$$\det = \frac{1}{3}\lambda_{\text{mín}}(K_p)\,\alpha\lambda_{\text{mín}}\left(K_i^{-1}\right) - 1 \geq 0$$

$$\frac{\alpha}{3}\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}\left(K_i^{-1}\right) \geq 1$$

$$\alpha \geq \frac{3}{\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}\left(K_i^{-1}\right)}$$

$$\det = \frac{1}{3}\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}(M) - \alpha^2\lambda_{\max}(M)^2 \geq$$

$$\frac{1}{3}\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}(M) \geq \alpha^2\lambda_{\max}(M)^2$$

$$\frac{\frac{1}{3}\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}(M)}{\lambda_{\max}(M)^2} \geq \alpha^2$$

$$\frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}(M)}}{\lambda_{\max}(M)} \geq \alpha$$

por lo tanto si hacemos que α cumpla con la restricción anterior, entonces $V(\tilde{\xi},\tilde{q},\dot{q})_3$ es definida positiva.

Finalmente, es fácil ver que $V(\tilde{\xi},\tilde{q},\dot{q})_4 \geq 0$.

Si hacemos cumplir a α las restricciones anteriores, tenemos:

$$\frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}(M)}}{\lambda_{\max}(M)} \geq \alpha \geq \frac{3}{\lambda_{\text{mín}}(K_p)\,\lambda_{\text{mín}}\left(K_i^{-1}\right)},$$

donde vemos que si K_p es suficientemente grande o K_i suficientemente pequeño, entonces $V(\tilde{\xi}, \tilde{q}, \dot{q})$ es definida positiva semiglobalmente.

La derivada temporal de $V(\tilde{\xi}, \tilde{q}, \dot{q})$ a lo largo de las trayectorias del sistema en lazo cerrado y usando $\frac{d}{dt}\int_0^t \Phi(\dot{q})d\tilde{\xi} = \frac{\partial \int_0^t \Phi(\dot{q})d\tilde{\xi}}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial t} = \dot{\tilde{\xi}}^T\Phi(\dot{q})$, tenemos:

$$\begin{aligned}\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) =& \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} + \tilde{q}^T K_p\dot{\tilde{q}} + \dot{q}G(q) + \dot{\tilde{q}}^T G\left(q_d\right) + \\ &+ \dot{\tilde{q}}^T\tilde{\xi} + \tilde{q}^T\dot{\tilde{\xi}} + \alpha\tilde{\xi}^T K_i^{-1}\dot{\tilde{\xi}} - \alpha\dot{\tilde{q}}^T M(q)\dot{q} - \alpha\tilde{q}^T\dot{M}(q)\dot{q} + \\ &- \alpha\tilde{q}^T M(q)\ddot{q} + \alpha\tilde{q}^T\left[K_d + B_{f1}\right]\dot{\tilde{q}} + \alpha\dot{\tilde{\xi}}^T K_i^{-1}\Phi(\dot{q})\end{aligned}$$

sustituyendo $\dot{\tilde{\xi}}, \dot{\tilde{q}}$, y $\ddot{q}$ en la ecuación anterior resulta:

$$\begin{aligned}\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) =& \dot{q}^T M(q)M(q)^{-1}\left[\begin{array}{c} K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} - G\left(q_d\right) + \\ +\beta(-\operatorname{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\end{array}\right] \\ &+ \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} + \tilde{q}^T K_p[-\dot{q}] + \dot{q}G(q) + [-\dot{q}]^T G\left(q_d\right) + \\ &+ [-\dot{q}]^T\tilde{\xi} + \tilde{q}^T\left[K_i\tilde{q}\right] + \alpha\tilde{\xi}^T K_i^{-1}\left[K_i\tilde{q}\right] - \alpha[-\dot{q}]^T M(q)\dot{q} - \alpha\tilde{q}^T\dot{M}(q)\dot{q} + \\ &- \alpha\tilde{q}^T M(q)M(q)^{-1}\left[\begin{array}{c} K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} - G\left(q_d\right) + \\ +\beta(-\operatorname{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\end{array}\right] \\ &+ \alpha\tilde{q}^T\left[K_d + B_{f1}\right][-\dot{q}] + \alpha\left[\tilde{q}^T K_i\right]K_i^{-1}\Phi(\dot{q})\end{aligned}$$

reagrupando términos, usando la propiedad de antisimetría $\frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \dot{q}^T C(q,\dot{q})\dot{q} = 0$, la igualdad $M(\dot{q}) = C(q,\dot{q}) + C(q,\dot{q})^T$ y simplificando, tenemos lo siguiente:

$$\begin{aligned}\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) =& -\dot{q}^T\left[K_d - \alpha M(q)\right]\dot{q} - \tilde{q}^T\left[\alpha K_p - K_i\right]\tilde{q} + \\ &- \alpha\tilde{q}^T C(q,\dot{q})^T\dot{q} - \alpha\tilde{q}^T\left[G\left(q_d\right) - G(q)\right] + \\ &+ \dot{q}^T[\beta(-\operatorname{sign}(\dot{q}) + \varepsilon) - F(\dot{q})].\end{aligned}$$

Normando la función de Lyapunov.

$$\begin{aligned}-\dot{q}^T\left[K_d - \alpha M(q)\right]\dot{q} \leq& -\left[\lambda_{\text{mín}}\left(K_d\right) - \alpha\lambda_{\text{máx}}(M)\right]\|\dot{q}\|_2^2 \\ -\tilde{q}^T\left[\alpha K_p - K_i\right]\tilde{q} \leq& -\left[\alpha\lambda_{\text{mín}}\left(K_p\right) - \lambda_{\max}\left(K_i\right)\right]\|\tilde{q}\|_2^2,\end{aligned}$$

además, usando las propiedades

$$\begin{aligned}\|C(x,y)z\| &\leq k_{C1}\|y\|\|z\| \\ \|G\left(q_d\right) - G(q)\| &\leq k_g\|x - y\|\end{aligned}$$

los siguientes términos cumplen con:

$$\begin{aligned} -\alpha\tilde{q}^T C(q,\dot{q})^T\dot{q} &\leq \alpha k_{C1}\|\tilde{q}\|_2\|\dot{q}\|_2^2 \\ -\alpha\tilde{q}^T\left[G\left(q_d\right)-G(q)\right] &\leq \alpha k_g\|\tilde{q}\|_2^2 \end{aligned}$$

finalmente, usando $\dot{q}^T\operatorname{sign}(\dot{q}) = \|\dot{q}\|_1$, con $\|\dot{q}\|_1 = |q_1| + |q_2| + \cdots + |q_m| = \sum_{i=1}^{m}\|\dot{q}_i\|_1$, donde $\|\cdot\|_1$ es la norma 1 , el valor absoluto.

$$\begin{aligned} -\dot{q}^T\beta\operatorname{sign}(\dot{q}) &\leq -\lambda_{\text{mín}}(\beta)\|\dot{q}\|_1 \\ \dot{q}^T\beta\varepsilon &\leq \varepsilon\lambda_{\text{mín}}(\beta)\|\dot{q}\|_1, \\ -\dot{q}^T F(\dot{q}) &\leq \lambda_{\text{máx}}\left(B_{f1}\right)\|\dot{q}\|_1 \end{aligned}$$

después de mayorar la función de Lyapunov, tenemos:

$$\begin{aligned} \dot{V}(\tilde{\xi},\tilde{q},\dot{q}) \leq &-\left[\lambda_{\text{mín}}\left(K_d\right)-\alpha\lambda_{\text{máx}}(M)\right]\|\dot{q}\|_2^2-\left[\alpha\lambda_{\text{mín}}\left(K_p\right)-\lambda_{\max}\left(K_i\right)\right]\|\tilde{q}\|_2^2, \\ &\alpha k_{C1}\|\tilde{q}\|_2\|\dot{q}\|_2^2+\alpha k_g\|\tilde{q}\|_2^2 \\ &-\lambda_{\text{mín}}(\beta)\|\dot{q}\|_1+\varepsilon\lambda_{\text{mín}}(\beta)\|\dot{q}\|_1+\lambda_{\text{máx}}\left(B_{f1}\right)\|\dot{q}\|_1 \end{aligned}$$

agrupando,

$$\begin{aligned} \dot{V}(\tilde{\xi},\tilde{q},\dot{q}) \leq &-\left[\lambda_{\text{mín}}\left(K_d\right)-\alpha\lambda_{\max}(M)-\alpha k_{C1}\|\tilde{q}\|_2\right]\|\dot{q}\|_2^2 \\ &-\left[\alpha\lambda_{\text{mín}}\left(K_p\right)-\lambda_{\max}\left(K_i\right)-\alpha k_g\right]\|\tilde{q}\|_2^2 \\ &-\left[\lambda_{\text{mín}}(\beta)-\varepsilon\lambda_{\text{mín}}(\beta)-\lambda_{\max}\left(B_{f1}\right)\right]\|\dot{q}\|_1 \end{aligned}$$

si elegimos la norma del error de posición de la siguiente manera $\|\tilde{q}\|_2$

$$\|\tilde{q}\|_2 \leq \frac{\lambda_{\text{máx}}(M)}{\alpha k_{C1}},$$

entonces, tomando el primer término de $\dot{V}(\tilde{\xi},\tilde{q},\dot{q})$, se obtiene la siguiente relación:

$$\lambda_{\text{mín}}\left(K_d\right)-\alpha\lambda_{\max}(M)-\alpha k_{C1}\|\tilde{q}\|_2 > 0$$

usando

$\|\tilde{q}\|_2 \leq \frac{\lambda_{\max}(M)}{\alpha k_{C1}}$ y $\alpha = \frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}}{\lambda_{\max}(M)}$

$$\begin{aligned}
&\lambda_{\text{mín}}\left(K_d\right) - \alpha\lambda_{\max}(M) - \alpha k_{C1}\frac{\lambda_{\max}(M)}{\alpha k_{C1}} \geq 0\\
&\lambda_{\text{mín}}\left(K_d\right) - \alpha\lambda_{\max}(M) - \lambda_{\max}(M) \geq 0\\
&\lambda_{\text{mín}}\left(K_d\right) \geq \alpha\lambda_{\max}(M) + \lambda_{\max}(M)\\
&\lambda_{\text{mín}}\left(K_d\right) \geq \frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}\left(K_p\right)\lambda_{\text{mín}}(M)}}{\lambda_{\max}(M)}\lambda_{\max}(M) + \lambda_{\max}(M)\\
&\lambda_{\text{mín}}\left(K_d\right) \geq \sqrt{\frac{1}{3}\lambda_{\min}\left(K_p\right)\lambda_{\text{mín}}(M)} + \lambda_{\max}(M)\\
&\lambda_{\min}\left(K_d\right) \geq \eta + \lambda_{\max}(M)
\end{aligned}$$

donde $\eta = \sqrt{\frac{1}{3}\lambda_{\text{mín}}\left(K_p\right)\lambda_{\text{mín}}(M)}$

Ahora, si tomamos el segundo término de $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q})$, y usando $\alpha = \frac{3}{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}\left(K_i^{-1}\right)}$ tenemos:

$$\begin{aligned}
&\alpha\lambda_{\min}\left(K_p\right) - \lambda_{\max}\left(K_i\right) - \alpha k_g \geq 0\\
&\alpha\lambda_{\min}\left(K_p\right) \geq \lambda_{\max}\left(K_i\right) + \alpha k_g\\
&\lambda_{\text{mín}}\left(K_p\right) \geq \frac{1}{\alpha}\lambda_{\max}\left(K_i\right) + k_g\\
&\lambda_{\min}\left(K_p\right) \geq \frac{\lambda_{\text{mín}}\left(K_p\right)\lambda_{\text{mín}}\left(K_i^{-1}\right)\lambda_{\max}\left(K_i\right)}{3} + k_g\\
&\text{si } \lambda_{\text{mín}}\left(K_i^{-1}\right) = \frac{1}{\lambda_{\max}\left(K_i\right)}\\
&\lambda_{\text{mín}}\left(K_p\right) \geq \frac{1}{3}\lambda_{\min}\left(K_p\right) + k_g\\
&\lambda_{\text{mín}}\left(K_p\right) \geq \frac{3}{2}k_g
\end{aligned}$$

Si tomamos la relación (5.5) encontramos el valor mínimo para K_i

$$\frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}\left(K_p\right)\lambda_{\text{mín}}(M(q))}}{\lambda_{\max}(M)} \geq \alpha \geq \frac{3}{\lambda_{\text{mín}}\left(K_p\right)\lambda_{\text{mín}}\left(K_i^{-1}\right)}$$

$$\frac{\eta}{\lambda_{\max}(M)} \geq \frac{3}{\lambda_{\text{mín}}\left(K_p\right)\lambda_{\text{mín}}\left(K_i^{-1}\right)}$$

$$\frac{\eta\lambda_{\text{mín}}\left(K_p\right)}{3\lambda_{\max}(M)} \geq \frac{1}{\lambda_{\text{mín}}\left(K_i^{-1}\right)}$$

si nuevamente tomamos $\lambda_{\text{mín}}\left(K_i^{-1}\right) = \dfrac{1}{\lambda_{\max}\left(K_i\right)}$

$$\eta\frac{\lambda_{\text{mín}}\left(K_p\right)}{3\lambda_{\max}(M)} \geq \lambda_{\max}\left(K_i\right).$$

Por lo tanto, si elegimos la siguiente sintonización para las ganancias K_p, K_d, K_i y β

entonces,

$$\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) \leq 0$$

4.2.6. Estabilidad asintótica

Definamos una bola $\sum$ de radio $\sigma > 0$ centrada en el origen del espacio de estados del tipo:

$$\sum \left\{ \tilde{q} : \|\tilde{q}\| \leq \frac{\lambda_{\text{máx}}(M)}{\alpha k_{C1}} = \sigma \right\}.$$

Luego si $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q})$ es semidefinida negativa en la bola $\sum$. Entonces existe una bola $\sum$ de radio $\sigma > 0$ centrada en el origen del espacio de estados que satisface que $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) \leq 0$.

Haciendo uso del principio de invariancia de Barbashin-Krassovskii-La Salle para sistemas dinámicos autónomos definimos primero Ω como:

$$\lambda_{\text{mín}}\left(K_p\right) \geq \frac{3}{2}k_g$$

$$\lambda_{\text{mín}}\left(K_d\right) \geq \eta + \lambda_{\text{máx}}(M)$$

$$\lambda_{\max}\left(K_i\right) \leq \eta\frac{\lambda_{\min}\left(K_p\right)}{3\lambda_{\max}(M)}$$

$$\lambda_{\min}(\beta) \geq \lambda_{\min}(\beta)\varepsilon + \lambda_{\max}\left(B_{f1}\right)$$

$$\Omega = \left\{ x(t) = [\tilde{\xi}, \tilde{q}, \dot{q}] \in \mathbb{R}^{3n} : \dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) = 0 \right\}$$

Para la derivada de la función de Lyapunov se cumple que:

$$\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) = 0.$$

Entonces, para una solución $x(t)$ contenida en Ω para todo $t \geq 0$, es necesario y suficiente que el error de regulación sea cero, es decir, $\tilde{q} = \dot{q} = 0$ para todo $t \geq 0$.

Por lo tanto también se cumple que $\ddot{q} = 0$ para todo $t \geq 0$.

Entonces, concluimos que del sistema en lazo cerrado (5.4), si la solución $x(t) \in \Omega$ para todo $t \geq 0$, entonces $G(q) = G(q_d) = \tilde{\xi} + G(q_d)$ y $\dot{\tilde{\xi}} = 0$. Esto implica que $\tilde{\xi} = 0$ para todo $t \geq 0$. De modo que $x(t) = [\tilde{\xi}, \tilde{q}, \dot{q}] = 0 \in \mathbb{R}^{3n}$ es sólo la condición inicial en Ω para la cuál $x(t) \in \Omega$ para todo $t \geq 0$. De aquí se concluye finalmente que el origen de la ecuación de malla cerrada (5.4) es un equilibrio asintoticamente estable en forma local.

Finalmente, con estos resultados se puede formular el siguiente teorema:

Teorema 5.1 Dada la dinámica del robot (5.2) controlada por el la ley de control PID $+QL$ (5.3), entonces el sistema en lazo cerrado (5.4) es semiglobalmente asintóticamente estable en el punto de equilibrio:

$$x = \left[\xi^T - G(q_d), \tilde{q}^T, \dot{q}^T\right]^T = 0 \in \mathbb{R}^{3n}$$

Con las siguientes condiciones para las ganancias:

$$\lambda_{\text{mín}}(K_p) \geq \frac{3}{2} k_g$$
$$\lambda_{\text{mín}}(K_d) \geq \eta + \lambda_{\max}(M)$$
$$\lambda_{\max}(K_i) \leq \eta \frac{\lambda_{\text{mín}}(K_p)}{3\lambda_{\text{máx}}(M)}$$
$$\lambda_{\text{mín}}(\beta) \geq \varepsilon \lambda_{\text{mín}}(\beta) + \lambda_{\max}(B_{f1})$$

$\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}, k_g$ satisfacen la condición de Lipschitz.

La dinámica del modelo humano y la interacción del robot y el ser humano se consideran en este diseño de control de bucle externo. Se mostró en [94] que la dinámica humana cambia durante el proceso de aprendizaje de tareas. Después del aprendizaje, un operador humano experto se caracteriza por una característica de transferencia lineal simple. Por lo tanto, se supone que el modelo de impedancia humana es

$$(K_d s + K_p) f_h = k_e e_d \tag{4.39}$$

donde K_d, K_p, y k_e son ganancias desconocidas. Estos parámetros varían de un individuo a otro y dependen de la tarea específica.

Capítulo 5

Simulaciones y Experimentos

Un péndulo es un sistema mecánico clásico para probar nuevas ideas en el campo del control inteligente [56]. Tiene la ventaja de ser un mecanismo relativamente simple por un lado y un sistema que contiene puntos de inestabilidad por el otro. Los péndulos se utilizan ampliamente como estándar para comparar dos algoritmos de control [65] y como hardware para implementarlos. Una de las otras cualidades de este dispositivo es que su dinámica no es lineal, similar a la de los robots de un solo grado de libertad. Los algoritmos utilizados para controlar el péndulo se pueden adaptar para controlar otros mecanismos más complejos [70]. Hay varias formas de implementar estos procesos de aprendizaje.

Descripción y planteamiento del problema. El péndulo es un servo mecanismo que consta de una base sobre el cual el péndulo rota los 360° de libertad y en nuestro caso puede girar libremente sin restricciones. El péndulo iniciará en su condición de equilibrio estable y con velocidad cero y se desea que el péndulo alcance la vertical invertida. Las condiciones iniciales están dadas de la siguiente manera:

$$\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

La ecuación dinámica que describe el comportamiento del péndulo está representada de la siguiente manera:

$$ml^2\ddot{\theta} + b\dot{\theta} + mgl\sin(\theta) = \tau,$$

tomando como variables de estado $q = \theta$, y $\dot{q} = \dot{\theta}$

$$\ddot{q} = \frac{\tau}{ml^2} - \frac{b}{ml^2}\dot{q} - \frac{g}{l}\sin(q).$$

Diseño de la ley de control El algoritmo Q-Learning requiere que el péndulo alcance la posición vertical superior, por lo que se propone como objetivo primordial diseñar un algoritmo de control que lleve al péndulo a la condición deseada $q_d = \pi$.

La posición del péndulo está limitada a $q \in [-\pi, \pi]$rad, la velocidad está restringida a $\dot{q} \in [-\pi, \pi]$rad/s.

Se genera una discretización del espacio de estados x_d basado en las posibles posiciones y velocidades en las que el péndulo podría estar. Las posiciones se definen como $z_1 = [-\pi, \pi]$ discretizado en pasos de 0,01 los cual produce 629 estados. Las velocidades se definen como $z_2 = [-\pi, \pi]$ discretizado en pasos de 0,1 lo cual produce 63 estados. La discretización del espacio de estados tendrá $629 \times 63 = 39627$ filas que representan todas las posibles combinaciones de posiciones y velocidades que podría tomar el péndulo. Además, se tendrán dos columnas una para definir posición y otra para definir velocidad. Finalmente, se tiene una discretización del espacio de estados $x_d \in \mathbb{R}^{39627\times 2}$.

El error está definido como:

$$e_x = x_d - x_s$$

y lo que se busca es la fila con el error más cercano a cero, $\min e_x$. Las acciones están definidas como el par aplicado al péndulo

$$u = \{-1, 0, 1\},$$

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

$$u_r = \arg\max_u \left[Q_{t+1}\left(e_{x_t}, u_t\right)\right],$$

La matriz Q se inicializa en ceros y el algoritmo Q-Learning está definido de la siguiente forma:

$$Q_{t+1}\left(e_{x_t}, u_t\right) = Q_t\left(e_{x_t}, u_t\right) + \alpha\left[r_{t+1} + \gamma \max_{u'} Q_t\left(e_{x_{t+1}}, u'\right) - Q_t\left(e_{x_t}, u_t\right)\right],$$

la recompensa estará definida de la forma:

$$r_{t+1} = -|\tilde{q}|^2 - 0,25|\dot{q}|^2,$$

donde $\tilde{q} = q_d - q$, y la mayor recompensa se produce cuando el error $\tilde{q}$ y la velocidad $\dot{q}$ son cero, es decir, cuando el péndulo se encuentra sobre la vertical superior y su velocidad es cero. Además, en la función de recompensa, la velocidad se encuentra escalada por un factor de 0,25 con la finalidad de no castigar al algoritmo de control a cambios bruscos de velocidad.

Para mostrar la efectividad del desempeño del controlador se realizaron las simulaciones bajo la plataforma Matlab . En la figura(5.1, 5.2, 5.3, 5.4) se presentan las gráficas de salida para el ángulo deposición y velocidad. En la posición observamos un atrayectoria suave que alcanza

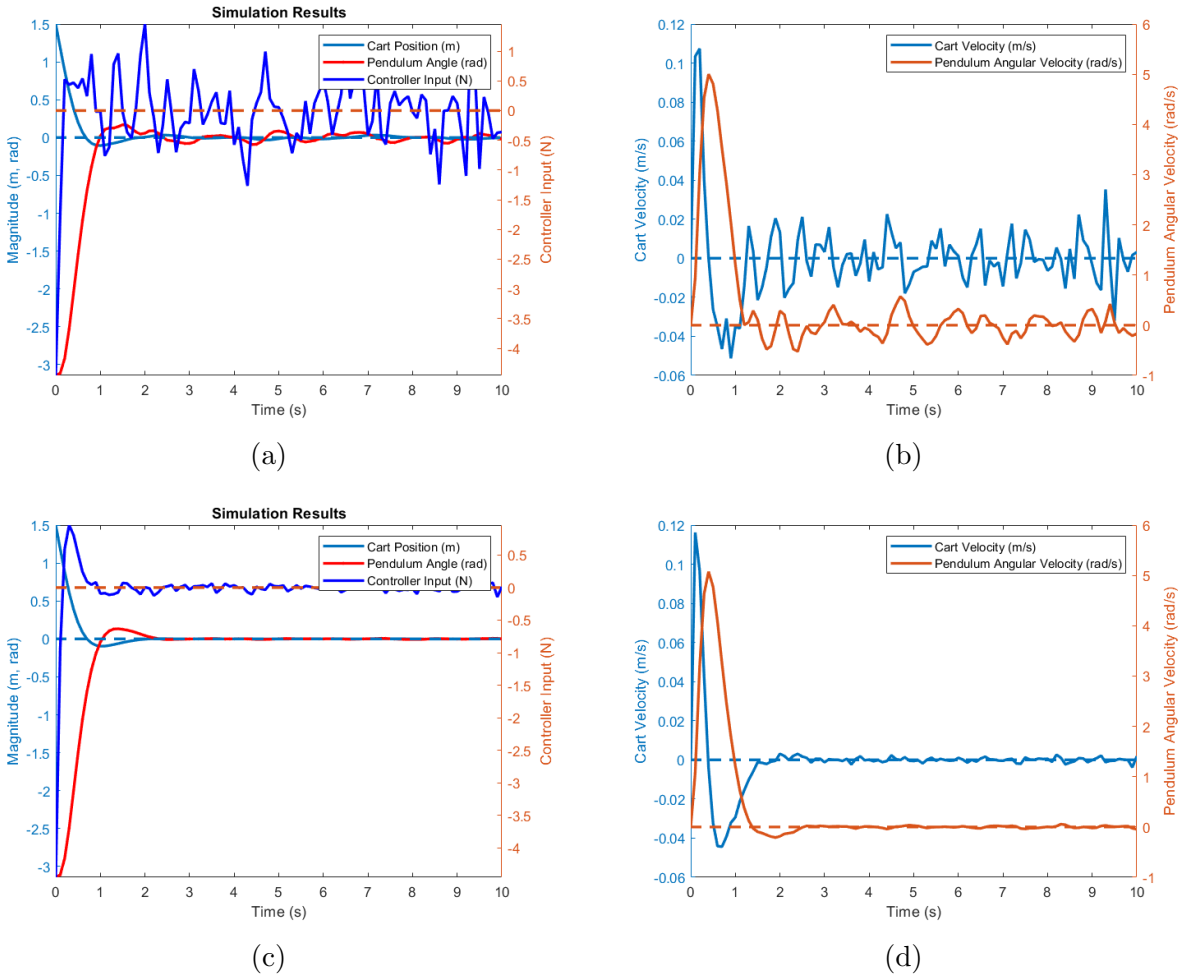

Figura 5.1: Gráficas de posición, ángulo y control, usando el controlador PID con y sin perturbaciones

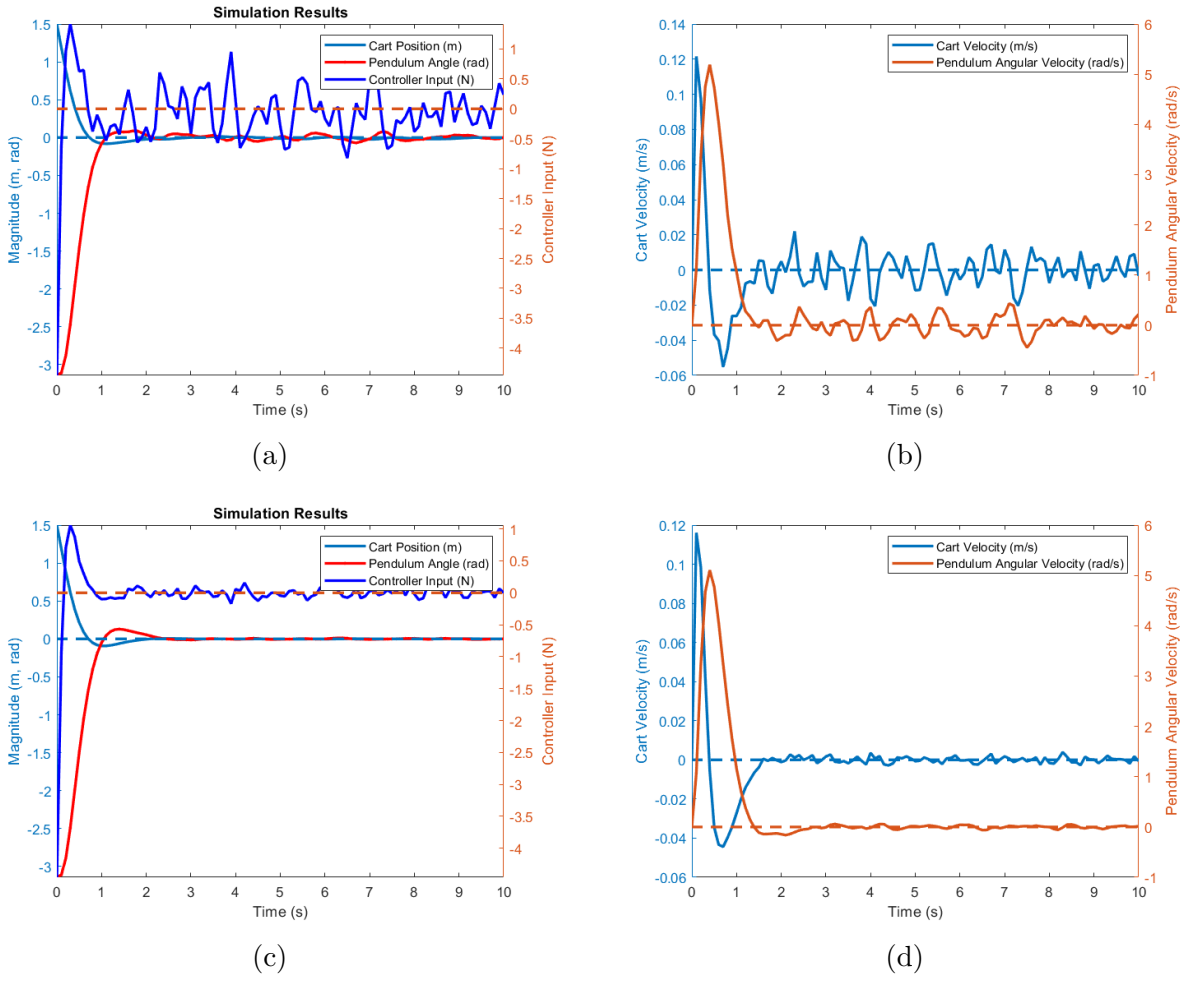

Figura 5.2: Gráficas de posición, ángulo y control, usando el controlador IRL con y sin perturbaciones

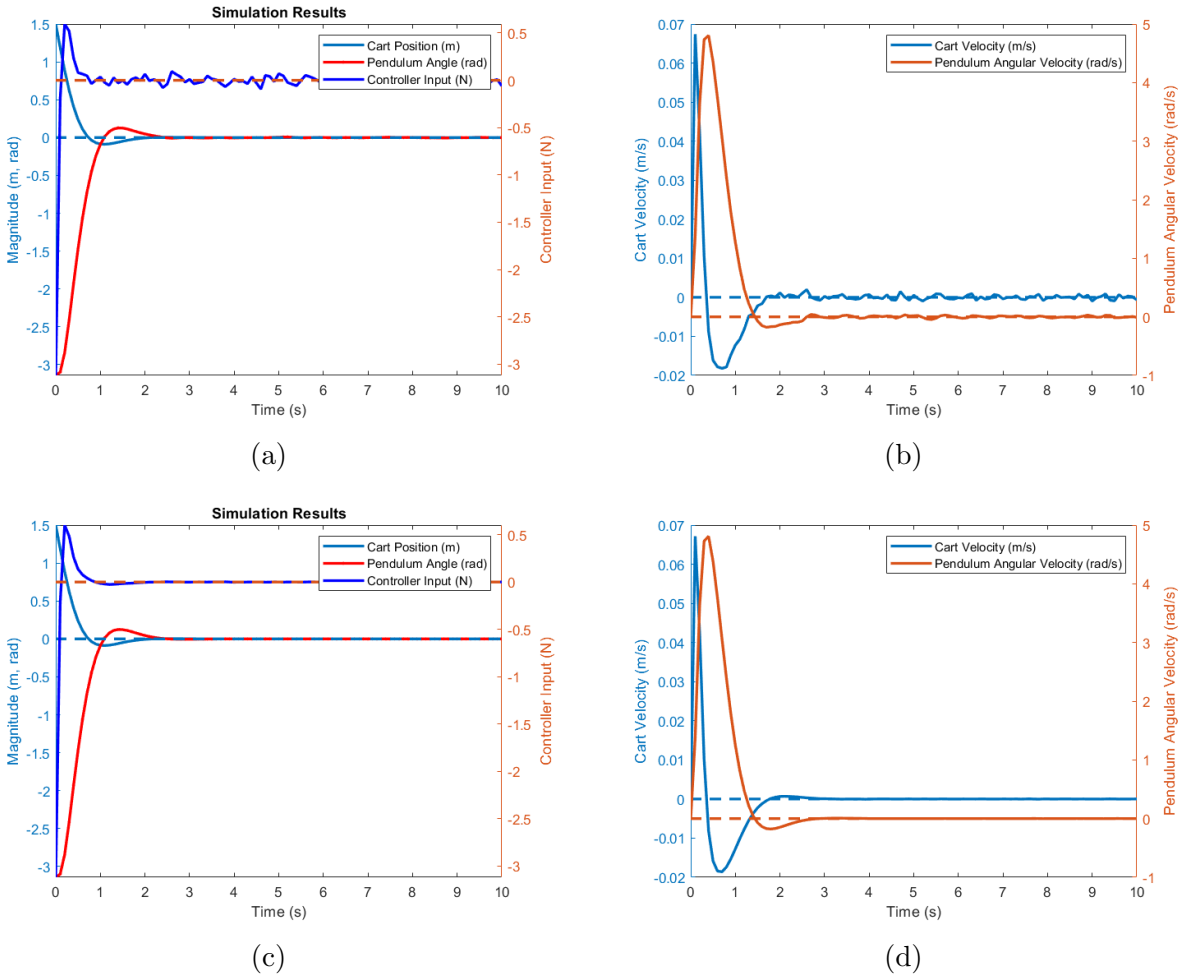

Figura 5.3: Gráficas de posición, ángulo y control, usando el controlador PID junto con el IRL como compensador con y sin perturbaciones

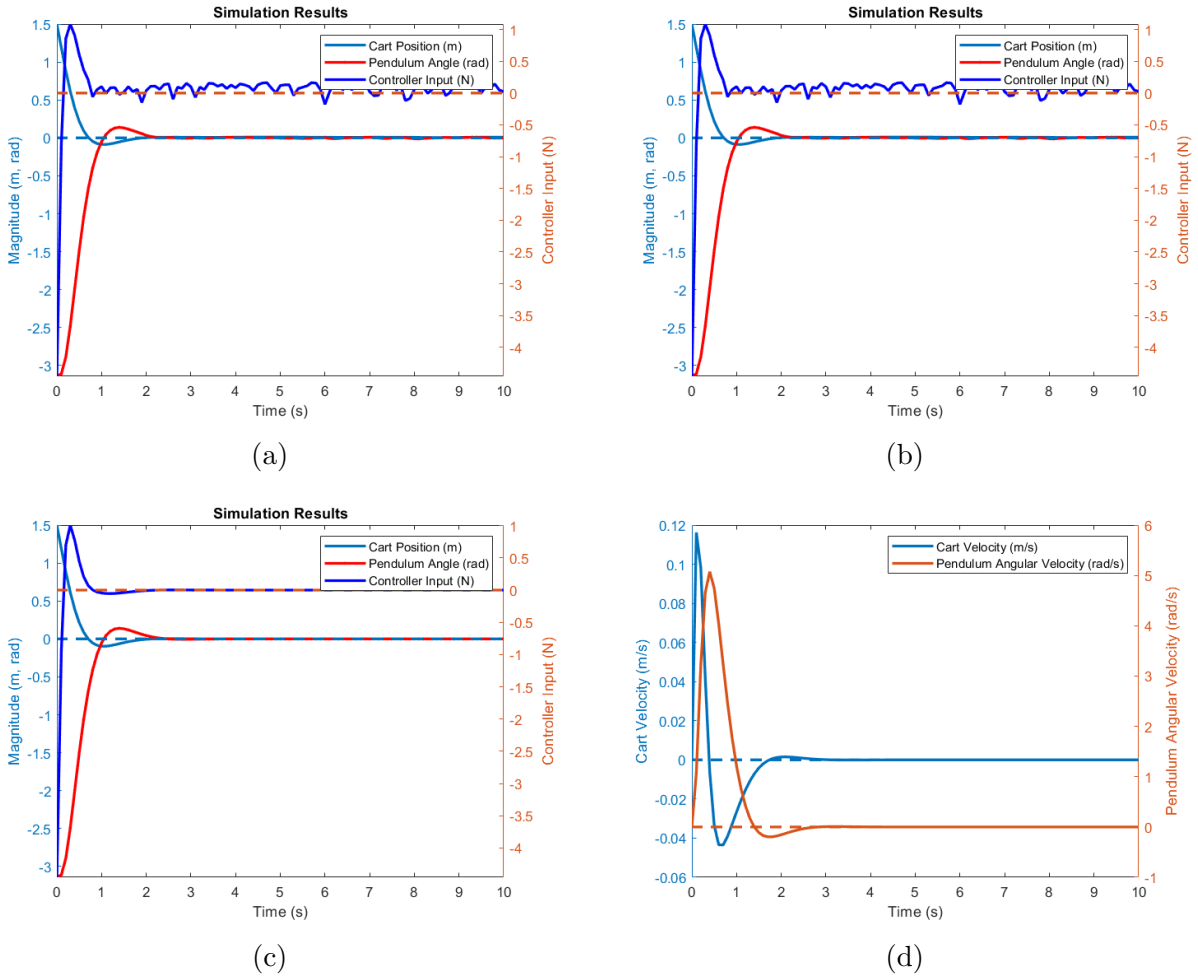

Figura 5.4: Gráficas de posición, ángulo y control, teniendo en cuenta a HITL

Capítulo 6

Conclusiones

Lo expuesto a lo largo de este trabajo permite arribar a las siguientes conclusiones:

Se presenta un nuevo método de diseño de control HRI inspirado en los estudios de factores humanos. La estructura de control propuesta tiene dos lazos de control. El primer bucle es un bucle de control interno en el cual, se diseñó un control PID con compensación Q-Learning basado en la teoría del aprendizaje por reforzamiento y el control clásico, y se realizaron pruebas de control en casos de estudio. Este controlador sirvió de base para generar las primeras pruebas de estabilidad local y estabilidad asintótica.

La ventaja de este algoritmo de control es que no necesita del conocimiento del modelo dinámico del sistema a controlar, lo cual resulta de lo más favorable al momento de seleccionar un esquema de control, debido a que el control es mucho más simple al usuario, y sin necesidad de conocer los parámetros del sistema. Otra ventaja, es que este algoritmo híbrido brinda robustez ante perturbaciones generadas de forma externa, y que no fueron presentadas durante su aprendizaje, siempre y cuando la ganancia del aprendizaje por refuerzo sea mayor a la perturbación más la dinámica a compensar.

Por lo tanto, las prestaciones de este controlador son satisfactorias y la combinación del control híbrido trabajó mejor de manera conjunta que de forma separada.

En cuanto al control PID con compensación IRL, se presenta una sintonización explícita de las ganancias del controlador, donde el valor máximo de la ganancia integral se da de forma explícita, y así se evitan problemas debido a valores muy grandes de la ganancia integral al momento de cancelar el error en estado estacionario. La principal contribución de este controlador es que la ganancia del aprendizaje por reforzamiento se utiliza para cancelar la dinámica del robot, dando mejores resultados en su forma híbrida PID con compensación IRL que de forma individual. Además, se presenta la prueba de que el sistema en lazo cerrado es semiglobal asintóticamente estable.

El segundo bucle es un bucle específico de la tarea que incluye al ser humano, al robot y su interacción y encuentra los parámetros óptimos de los parámetros de impedancia prescritos para ayudar al ser humano a realizar la tarea con menos esfuerzo y un rendimiento óptimo.

En concreto, las principales contribuciones de esta tesis fueron:

- **Garantías teóricas de finalización de tareas de bajo nivel:** esto se probó teóricamente para el marco de control predictivo del modelo presentado utilizando el análisis de Lyapunov, y el control proporciona robustez a la par con el control óptimo de horizonte infinito.
- **Posibilidad de completar tareas de alto nivel para operadores humanos:** a través de tareas de ejemplo, estudios de operadores humanos y aplicaciones, el marco de control proporcionó a los operadores humanos la libertad de llevar a cabo aspectos de alto nivel de la tarea a pesar de compartir el control del sistema. Los operadores pudieron llevar a cabo tareas de control compartidas para escenarios de robots individuales y múltiples y robots enjambre.
- **Autonomía de deslizamiento basada en el desempeño de tareas de bajo nivel:** la influencia del control humano se ponderó más cuando el operador realiza adecuadamente la tarea de bajo nivel y se ponderó más bajo cuando el operador tiene dificultades. De esta manera, hemos desarrollado un marco de control cooperativo humano-robot donde el control automático complementa las habilidades del operador humano cuando es necesario.

Apéndice A

Modelos

A.1. Sistema carro-péndulo

La dinámica del sistema carro-péndulo se expresa por

$$M(q)\dot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu \tag{A.1}$$

donde $q \in \mathbb{R}^n$ representan las posiciones de cada junta, $\dot{q} \in \mathbb{R}^n$ representa la velocidad de las juntas, $M(q) \in \mathbb{R}^{n\times n}$ es la matriz de inercia, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ es la matriz de Coriolis, $G(q)$ es el vector de pares gravitacionales, $F \in \mathbb{R}^{n\times n}$ representa la fricción no lineal y τ es el par aplicado en cada junta.

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{m\,sin(q)(g\,cos(q) - l\dot{q})}{M+m-m\,cos(q)^2} \\ \frac{sin(q)(g(M+m) - l\,m\,cos(q)\dot{q}))}{l(M+m-m\,cos(q))} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{M+m-m\,cos(q)^2} \\ \frac{cos(q)}{l(M+m-m\,cos(q))} \end{bmatrix} u \tag{A.2}$$

Donde:

$$\begin{aligned} x &= (x_c, \dot{x}_c, q, \dot{q}_c)^T \\ u &= F \end{aligned} \tag{A.3}$$

El vector de estado es $x = (x_c, \dot{x}_c, q, \dot{q})^\top$ y $u = F$ es la entrada de control.

A.2. Parametros usados

La Tabla ?? muestra los valores de los parámetros de control del algoritmo PDI+IRL

Parametro	Descripción	Valor
m	Masa de péndulo	$0.3kg$
g	Gravedad	$9.81m/s^2$
M	Masa del carro	$2kg$
l	Longitud del péndulo	$0.4m$
K_p	Ganancia proporcional	1.8
K_i	Ganancia integral	-2.4
K_d	Ganancia derivatival	-35

Bibliografía

[1] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998.

[2] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[3] M. Waltz and K. Fu. A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control*, 10(4):390–398, 1965.

[4] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

[5] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[8] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016.

[9] Rahul Chipalkatty and Magnus Egerstedt. Human-in-the-loop: Terminal constraint receding horizon control with human inputs. In *2010 IEEE International Conference on Robotics and Automation*, pages 2712–2717, May 2010.

[10] Michael A. Goodrich and Alan C. Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, January 2007.

[11] J.E. Allen, C.I. Guinn, and E. Horvtz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, Sep. 1999.

[12] P. Griffiths and R.B. Gillespie. Shared control between human and machine: haptic display of automation during manual control of vehicle heading. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.*, pages 358–366, March 2004.

[13] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.

[14] Bernard Widrow. Pattern-recognizing control systems. *Compurter and Information Sciences*, 1964.

[15] Ronald J Williams. *Reinforcement-learning connectionist systems.* College of Computer Science, Northeastern University, 1987.

[16] Avron Barr, Edward A Feigenbaum, and Paul R Cohen. *The handbook of artificial intelligence*, volume 1. William Kaufmann, 1981.

[17] Michael I. Jordan and David E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354, July 1992.

[18] Andrew G Barto. *Some learning tasks from a control perspective.* CRC Press, 2018.

[19] B Widrow. Adaptive control by inverse modeling. In *Twelfth Asilomar Conference on Circuits, Systems, and Computers, 1979*, 1979.

[20] David A. Rosenbaum, Kate M. Chapman, Chase J. Coelho, Lanyun Gong, and Breanna E. Studenka. Choosing actions. *Frontiers in Psychology*, 4, 2013.

[21] Graham C. (Graham Clifford) Goodwin. *Adaptive filtering prediction and control / Graham C. Goodwin and Kwai Sang Sin.* Prentice-Hall information and system sciences series. Prentice-Hall, Englewood Cliffs, N.J, 1984.

[22] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[23] Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

[24] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, May 1992.

[25] Hamidreza Modares, Isura Ranatunga, Bakur AlQaudi, Frank L. Lewis, and Dan O. Popa. Intelligent human–robot interaction systems using reinforcement learning and neural networks. In *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*, pages 153–176. Springer International Publishing, 2017.

[26] Tao Zhang and M. Nakamura. Neural network-based hybrid human-in-the-loop control for meal assistance orthosis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(1):64–75, March 2006.

[27] Guohuai Lin, Hongyi Li, Hui Ma, Deyin Yao, and Renquan Lu. Human-in-the-loop consensus control for nonlinear multi-agent systems with actuator faults. *IEEE/CAA Journal of Automatica Sinica*, pages 1–12, 2020.

[28] Lucian Busoniu. *Reinforcement learning and dynamic programming using function approximators*. CRC Press, Boca Raton, FL, 2010.

[29] David Luviano and Wen Yu. Continuous-time path planning for multi-agents with fuzzy reinforcement learning. *Journal of Intelligent & Fuzzy Systems*, 33:491–501, 2017. 1.

[30] Thomas Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. *ACM SIGCHI Curricula for Human-Computer Interaction*. Association for Computing Machinery, January 1992.

[31] Adolfo Perrusquía and Wen Yu. Human-in-the-loop control using euler angles. *Journal of Intelligent & Robotic Systems*, 97(2):271–285, Feb 2020.

[32] Luka Peternel, Tadej Petrič, and Jan Babič. Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1497–1502, May 2015.

[33] Rahul Chipalkatty, Hannes Daepp, Magnus Egerstedt, and Wayne Book. Human-in-the-loop: Mpc for shared control of a quadruped rescue robot. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4556–4561, Sep. 2011.

[34] Mathew DeDonato, Velin Dimitrov, Ruixiang Du, Ryan Giovacchini, Kevin Knoedler, Xianchao Long, Felipe Polido, Michael A. Gennert, Taşkın Padır, Siyuan Feng, Hirotaka Moriguchi, Eric Whitman, X. Xinjilefu, and Christopher G. Atkeson. Human-in-the-loop control of a humanoid robot for disaster response: A report from the darpa robotics challenge trials. *J. Field Robot.*, 32(2):275–292, March 2015.

[35] Ashwin P. Dani, Iman Salehi, Ghananeel Rotithor, Daniel Trombetta, and Harish Ravichandar. Human-in-the-loop robot control for human-robot collaboration: Human intention estimation and safe trajectory tracking control for collaborative tasks. *IEEE Control Systems Magazine*, 40(6):29–56, Dec 2020.

[36] Tansel Yucelen, Yildiray Yildiz, Rifat Sipahi, Ehsan Yousefi, and Nhan Nguyen. Stability limit of human-in-the-loop model reference adaptive control architectures. *International Journal of Control*, 91(10):2314–2331, 2018.

[37] Hongyi Liu and Lihui Wang. Human motion prediction for human-robot collaboration. *Journal of Manufacturing Systems*, 44:287–294, 2017. Special Issue on Latest advancements in manufacturing systems at NAMRC 45.

[38] Roberto Meattini, Davide Chiaravalli, Gianluca Palli, and Claudio Melchiorri. semg-based human-in-the-loop control of elbow assistive robots for physical tasks and muscle strength training. *IEEE Robotics and Automation Letters*, 5(4):5795–5802, Oct 2020.

[39] I.R. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. *IEEE Pervasive Computing*, 4(1):72–79, Jan 2005.

[40] Rachel Schlossman, Minkyu Kim, Ufuk Topcu, and Luis Sentis. Toward achieving formal guarantees for human-aware controllers in human-robot interactions. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7770–7776, Nov 2019.

[41] Tatsuya Teramae, Koji Ishihara, Jan Babič, Jun Morimoto, and Erhan Oztop. Human-in-the-loop control and task learning for pneumatically actuated muscle based robots. *Frontiers in Neurorobotics*, 12:71, 2018.

[42] Dong Wei, Zhijun Li, Qiang Wei, Hang Su, Bo Song, Wei He, and Jianqiang Li. Human-in-the-loop control strategy of unilateral exoskeleton robots for gait rehabilitation. *IEEE Transactions on Cognitive and Developmental Systems*, 13(1):57–66, March 2021.

[43] Juanjuan Zhang, Pieter Fiers, Kirby A. Witte, Rachel W. Jackson, Katherine L. Poggensee, Christopher G. Atkeson, and Steven H. Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2017.

[44] David Sousa Nunes, Pei Zhang, and Jorge Sá Silva. A survey on human-in-the-loop applications towards an internet of all. *IEEE Communications Surveys Tutorials*, 17(2):944–965, Secondquarter 2015.

[45] A. Tustin. The nature of the operator's response in manual control, and its implications for controller design. *Journal of the Institution of Electrical Engineers - Part IIA: Automatic Regulators and Servo Mechanisms*, 94(2):190–206, May 1947.

[46] Katherine Driggs-Campbell, Victor Shia, and Ruzena Bajcsy. Improved driver modeling for human-in-the-loop vehicular control. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1654–1661, May 2015.

[47] Duane T McRuer and Ezra S Krendel. Dynamic response of human operators. Technical report, KELSEY-HAYES CO INGLEWOOD CA CONTROL SPECIALISTS DIV, 1957.

[48] Duane T McRuer and Ezra S Krendel. Mathematical models of human pilot behavior. Technical report, ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT NEUILLY-SUR-SEINE (FRANCE), 1974.

[49] T. PETER NEAL and ROGERS E. SMITH. A flying qualities criterion for the design of fighter flight-control systems. *Journal of Aircraft*, 8(10):803–809, October 1971.

[50] W. S. Levine. *Control system applications*. CRC Press, Boca Raton, Fla, 2000.

[51] Martin R. Cacan, Mark Costello, Michael Ward, Edward Scheuermann, and Michael Shurtliff. Human-in-the-loop control of guided airdrop systems. *Aerospace Science and Technology*, 84:1141–1149, 2019.

[52] Dr. John K. Hawley. *PATRIOT WARS Automation and the Patriot Air and Missile Defense System*. Center for a New American Security, 2017.

[53] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. The smart thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, page 211–224, New York, NY, USA, 2010. Association for Computing Machinery.

[54] Matthew Kay, Eun Kyoung Choe, Jesse Shepherd, Benjamin Greenstein, Nathaniel Watson, Sunny Consolvo, and Julie A. Kientz. Lullaby. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*. ACM Press, 2012.

[55] G. Burnham, Jinbom Seo, and G. Bekey. Identification of human driver models in car following. *IEEE Transactions on Automatic Control*, 19(6):911–915, December 1974.

[56] C. N. Viswanathan, R. W. Longman, and P. W. Likins. A degree of controllability definition - fundamental concepts and application to modal systems. *Journal of Guidance, Control, and Dynamics*, 7(2):222–230, March 1984.

[57] Peter Burgmeier. Degrees of controllability. In *Operations Research '91*, pages 182–185. Physica-Verlag HD, 1992.

[58] Haemin Lee and Youngjin Park. Degree of controllability for linear unstable systems. *Journal of Vibration and Control*, 22(7):1928–1934, August 2014.

[59] Reza Arghandeh, Alexandra von Meier, Laura Mehrmanesh, and Lamine Mili. On the definition of cyber-physical resilience in power systems. *Renewable and Sustainable Energy Reviews*, 58:1060–1069, May 2016.

[60] Alexander A. Ganin, Emanuele Massaro, Alexander Gutfraind, Nicolas Steen, Jeffrey M. Keisler, Alexander Kott, Rami Mangoubi, and Igor Linkov. Operational resilience: concepts, design and analysis. *Scientific Reports*, 6(1), January 2016.

[61] Dwight Read. SOME OBSERVATIONS ON RESILIENCE AND ROBUSTNESS IN HUMAN SYSTEMS. *Cybernetics & Systems*, 36(8):773–802, December 2005.

[62] Giliberto Capano and Jun Jie Woo. Resilience and robustness in policy design: a critical appraisal. *Policy Sciences*, 50(3):399–426, January 2017.

[63] Paulo Leitão, Stamatis Karnouskos, Luis Ribeiro, Jay Lee, Thomas Strasser, and Armando W. Colombo. Smart agents in industrial cyber–physical systems. *Proceedings of the IEEE*, 104(5):1086–1101, May 2016.

[64] Steven Carr, Nils Jansen, Ralf Wimmer, Jie Fu, and Ufuk Topcu. Human-in-the-loop synthesis for partially observable markov decision processes. In *2018 Annual American Control Conference (ACC)*, pages 762–769, June 2018.

[65] Chi-Pang Lam and S. Shankar Sastry. A pomdp framework for human-in-the-loop system. In *53rd IEEE Conference on Decision and Control*, pages 6031–6036, Dec 2014.

[66] Tariq Samad. Human-in-the-loop control: Applications and categorization. *IFAC-PapersOnLine*, 53(5):311–317, 2020. 3rd IFAC Workshop on Cyber-Physical & Human Systems CPHS 2020.

[67] Rifat Sipahi, Silviu-iulian Niculescu, Chaouki T. Abdallah, Wim Michiels, and Keqin Gu. Stability and stabilization of systems with time delay. *IEEE Control Systems Magazine*, 31(1):38–65, Feb 2011.

[68] Gabor Stepan. Delay effects in brain dynamics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1891):1059–1062, February 2009.

[69] Albert Goldbeter. Modelling biochemical oscillations and cellular rhythms. *Current Science*, 73(11):933–939, 1997.

[70] Adrián Ramírez and Rifat Sipahi. Design of a delay-based controller for fast stabilization in a network system with input delays via the lambert w function 1. *Procedia IUTAM*, 22:83–90, 2017.

[71] Dunstan Graham and Duane T McRuer. *Analysis of nonlinear control systems*. Wiley, 1961.

[72] Duane T McRuer. Human pilot dynamics in compensatory systems. Technical report, SYSTEMS TECHNOLOGY INC HAWTHORNE CA, 1965.

[73] D.T. McRuer and H.R. Jex. A review of quasi-linear pilot models. *IEEE Transactions on Human Factors in Electronics*, HFE-8(3):231–249, Sep. 1967.

[74] Norman Nise. *Control systems engineering*. Benjamin/Cummings Pub. Co, Redwood City, Calif, 1992.

[75] Satoshi Suzuki and Katsuhisa Furuta. Adaptive impedance control to enhance human skill on a haptic interface system. *Journal of Control Science and Engineering*, 2012:1–10, 2012.

[76] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal Control*. John Wiley & Sons, Chichester, England, 3 edition, January 2012.

[77] Bahare Kiumarsi, Frank. L. Lewis, Mohammad-Bagher Naghibi-Sistani, and Ali Karimpour. Optimal tracking control of unknown discrete-time linear systems using input-output measured data. *IEEE Transactions on Cybernetics*, 45(12):2770–2779, 2015.

[78] Hamidreza Modares and Frank L. Lewis. Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning. *IEEE Transactions on Automatic Control*, 59(11):3051–3056, Nov 2014.

[79] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F.L. Lewis. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 45(2):477–484, 2009.

[80] Draguna Vrabie and Frank Lewis. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. *Neural Networks*, 22(3):237–246, 2009. Goal-Directed Neural Systems.

[81] Yu Jiang and Zhong-Ping Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, 2012.

[82] Jae Young Lee, Jin Bae Park, and Yoon Ho Choi. Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5):916–932, 2015.

[83] Hamidreza Modares, Frank L. Lewis, and Mohammad-Bagher Naghibi-Sistani. Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. *Automatica*, 50(1):193–202, 2014.

[84] Hamidreza Modares, Frank L. Lewis, and Mohammad-Bagher Naghibi-Sistani. Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 24(10):1513–1525, 2013.

[85] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[86] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[87] Frank L Lewis, Darren M Dawson, and Chaouki T Abdallah. *Robot manipulator control: theory and practice*. CRC Press, 2003.

[88] SS Ge, CC Hang, LC Woon, and XQ Chen. Impedance control of robot manipulators using adaptive neural networks. *International Journal of Intelligent Control and Systems*, 2(3):433–452, 1998.

[89] Guozheng Xu and Aiguo Song. Adaptive impedance control based on dynamic recurrent fuzzy neural network for upper-limb rehabilitation robot. In *2009 IEEE International Conference on Control and Automation*. IEEE, December 2009.

[90] L. Huang, S.S. Ge, and T.H. Lee. Neural network based adaptive impedance control of constrained robots. In *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, pages 615–619, 2002.

[91] Elena Gribovskaya, Abderrahmane Kheddar, and Aude Billard. Motion learning and adaptive impedance for robot control during physical interaction with humans. In *2011 IEEE International Conference on Robotics and Automation*, pages 4326–4332, 2011.

[92] Shahid Hussain, Sheng Q. Xie, and Prashant K. Jamwal. Adaptive impedance control of a robotic orthosis for gait rehabilitation. *IEEE Transactions on Cybernetics*, 43(3):1025–1034, 2013.

[93] Seul Jung and T.C. Hsia. Neural network impedance force control of robot manipulator. *IEEE Transactions on Industrial Electronics*, 45(3):451–461, 1998.

[94] Katsuhisa Furuta, Yuya Kado, and Shinya Shiratori. Assisting control in human adaptive mechatronics—single ball juggling—. In *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pages 545–550. IEEE, 2006.